

Thèse de doctorat

Pour obtenir le grade de :
Docteur de l'université de Paris 5 René Descartes.

Discipline : Informatique

Présentée par : Benoît DOLEZ

Labellisation d'images par méthodes fractales

Présentée le :

Nicole Vincent	Directrice de thèse
Marinette Revenu	Rapporteur
Christine Fernandez	Rapporteur
Jacques Levy Vehel	Examineur
Georges Stamon	Examineur
Noël RICHARD	Examineur
Daniel Duclos	Examineur
Thomas Devichi	Invité

1	RESUME	4
2	INTRODUCTION	6
2.1	Introduction générale	6
2.2	Sujet et Contexte Industriel	8
2.3	Plan de la thèse	12
3	ETAT DE L'ART, ANALYSE CRITIQUE ET OUTILS	13
3.1	Introduction	13
3.2	Etat de l'art sur la labellisation	13
3.2.1	Définition d'une zone homogène	13
3.2.2	Croissance, réunion et partage de régions étiquetées	14
3.2.3	Segmentation et approche MDL	15
3.3	Etat de l'art sur les zones texturées	17
3.3.1	Introduction et discussion sur la définition d'une texture	18
3.3.2	Matrice de cooccurrence	21
3.3.3	Filtres de Gabor	24
3.3.4	Les Moments	26
3.3.5	Filtres de Laws	27
3.3.6	Caractéristiques fréquentielles de la texture	29
3.3.7	Détection de zones artificielles	31
3.3.8	Colorisation d'image monochrome	32
3.4	Opérateurs fractals	35
3.4.1	Les Fractales, historique et applications	36
3.4.2	Synthèse et suréchantillonnage	40
3.4.3	Dimension fractale	43
3.4.4	Dimension fractale directionnelle par ondelettes	51
3.4.5	Lacunarité	56
3.4.6	Multi fractales	57
3.4.7	D's C : 3.4.7 Apport dans le domaine des multi fractales	177
3.4.8	Compression de données image	67
3.5	Conclusion	69
4	APPRENTISSAGE DE CONCEPTS	70
4.1	Position du problème	70
4.2	Construction d'une transformation contractante	72
4.2.1	Principe de la compression fractale	72
4.2.2	Construction d'une transformation contractante et compression de l'image	74
4.3	Mise en place d'une méthode d'apprentissage de concepts	85
4.3.1	Première approche	85
4.3.2	Méthode de comparaison d'imagettes	88
4.3.3	Prise en compte des dépendances ranges-domaines	90

4.3.4	Apprentissage par seuil adaptatif	94
4.3.5	Approche par sélection des ranges	104
4.4	Conclusion	108
5	RECONNAISSANCE ET OPTIMISATION	112
5.1	Position du problème, introduction	112
5.2	Mise en place d'une méthode de reconnaissance	113
5.2.1	Introduction et approche préliminaire	113
5.2.2	Contexte et carte de richesse	115
5.2.3	Conclusion	121
5.3	Outils, optimisation et dimensionnement de l'apprentissage	121
5.3.1	Introduction	121
5.3.2	Appariement d'images	121
5.3.3	Mise à jour de masque par valeur min/max locale	125
5.3.4	Séparation naturel / artificiel	126
5.3.5	Optimisation dimensionnement de l'apprentissage	135
5.3.6	Conclusion sur les outils et l'optimisation	142
5.4	Évaluation	143
5.4.1	Introduction	143
5.4.2	Reconnaissance	143
5.4.3	Extraction de points marginaux	159
5.4.4	Conclusion de l'évaluation	162
5.5	Conclusion	163
6	CONCLUSION ET PERSPECTIVES	164
6.1	Conclusion	164
6.2	Perspectives	165
6.2.1	Streaming vidéo	165
6.2.2	Optimisation sur la matrice de reconstruction	165
6.2.3	Adaptation de l'apprentissage aux zones naturelles	166
6.2.4	Identification d'objets	166
7	BIBLIOGRAPHIE	169

1 Résumé

Résumé :

L'objectif de notre étude est de générer une carte d'étiquettes reliant chaque partie de l'image à un concept appris. Il s'agit de se positionner dans le cadre d'une compréhension macroscopique de l'image. Cette compréhension permet de répondre à des problématiques souvent situées en amont de chaîne algorithmique, telle que l'estimation de fond, dans le cadre de la détection de cible, ou la réalité augmentée. L'étiquetage de régions d'une image, ou labellisation, présente plusieurs difficultés liées aux variabilités des informations à traiter et de leurs contextes. Labelliser une région équivaut à la relier à un concept connu. Les principales problématiques portent sur les variabilités présentes au niveau inter et intra concept. La première dépend de la manière et de la finesse avec lesquelles nous définissons ce que nous voulons reconnaître. La seconde témoigne d'une variabilité des paramètres intervenant dans un même concept. A cela nous devons ajouter l'influence du contexte sur ce que nous cherchons à caractériser. Classiquement, cette influence extérieure proviendra de l'éclairage de la scène observée ou des paramètres atmosphériques. Notre méthode repose sur une approche statistique et sur un apprentissage de blocs image caractéristiques. L'approche statistique nous permet de caractériser les régions naturelles, végétales de l'image. Les zones géométriques, telles que les véhicules ou les bâtiments, sont d'avantage caractérisables par leur structure et les sous parties qui les composent. Nous avons choisi de traiter ces deux aspects en nous appuyant sur les outils fractals. En effet, les opérateurs fractals statistiques sont adaptés à la modélisation de végétation et les méthodes de compression fractale utilisent les blocs image à différentes échelles, prenant également en compte les variabilités en rotation et éclairage, pour synthétiser l'information. Les systèmes de fonctions itérés et un système de notation nous permettent d'extraire les imquettes les plus représentatives d'une image. L'aspect composite des sous parties d'un concept/objet est pris en compte lors de l'apprentissage et de la reconnaissance. Ceci nous permet une détection robuste dans des contextes image parfois complexes. Nous commençons par étudier le contexte industriel actuel et les applications connexes à notre étude. Puis nous analysons l'existant algorithmique présent dans ce domaine. Nous explicitons ensuite notre méthode et la caractérisons en termes de performances. Pour conclure, nous effectuons un récapitulatif de nos travaux et posons les perspectives applicables à notre étude.

Abstract:

The aim of our study is to create a label map which links every part of the image to a learnt concept. This task is included into the scheme of macroscopic image understanding. This understanding allow to answer classical problems, generally located at the very beginning of an image algorithmic process, such as background estimation in the scheme of target detection, or augmented reality. Image region labelling presents many difficulties because of the variability of available information and its context. The variability of the information may come from intra and inter concept variability. The first one depends on the way and sharpness we have described every concept. The second one shows the variability of the parameters contributing to a given concept. Moreover we must take the influence of the context on what we are trying to characterize. Usually, this external influence comes from the scene lighting or atmospheric parameters. Our method is based on a statistical approach and a characteristic image block learning. The statistical approach allows characterising the natural and vegetable regions of the image. Geometric zones, such as vehicles and buildings, are characterisable by their structure or parts they are composed by. We have chosen to deal with both of these aspects by using fractal tools. Actually, statistic fractal operators suit to vegetation modelling and fractal compression methods use multiscale image blocks and take orientation and lighting parameters into account in order to synthesize information. Iterated Function Systems and a score counting process allow extracting the most representative blocks of an image. The composite aspect of parts of a concept is taken into account during the learning and recognition process. This ensures a robust detection in complex image context. We first study the present industrial context and the related application to our study. Then we analyse the existing algorithms in this domain. Later we explain our method and characterise it in terms of performance. Finally we sum our work up and propose perspectives to our study.

2 Introduction

2.1 Introduction générale

La labellisation de données est une tâche que nous effectuons plus ou moins consciemment dans la vie de tous les jours. Elle consiste à isoler un signal parmi d'autres, auditif, olfactif ou visuel par exemple, et à lui attribuer une étiquette de manière à synthétiser l'information. Ainsi nous sommes capable de reconnaître un morceau de musique dans un bruit de conversation, ou de localiser des véhicules, des bâtiments ou des personnes, par rapport à notre position actuelle. Notons qu'il n'est pas nécessaire d'avoir une interprétation fine de la scène pour bien la comprendre. Dans le cas d'obstacles, une localisation grossière suffit souvent pour pouvoir le contourner. De même, il n'est pas nécessaire de dénombrer, séparer des objets dans une scène pour les labelliser. Une groupe d'objets peut être traité et labellisé comme tel plutôt que chaque objet individuellement. On observe notamment ce type d'approche au niveau cortical. Notre action se situe alors essentiellement dans le déplacement de notre point d'intérêt. Ainsi, en reprenant un contexte visuel, nous déplaçons notre regard d'une région de la scène perçue à une autre sans pour autant connaître le nombre précis d'objets présents [VIT06]. Nous considérons un pâté de maisons ou une foule d'individus dans son ensemble, du moins tout d'abord, avant d'éventuellement porter notre attention sur un point plus spécifique de la scène observée. L'intérêt d'étiqueter l'information au niveau macroscopique est dans un premier temps de la synthétiser. Par la suite, cette représentation permet de faire le lien entre l'appris, nos connaissances accumulées dans le passé, et l'information actuellement présente. Par exemple, la labellisation d'un véhicule présent dans la scène, nous permet d'extraire de forts a priori tels que des estimations, grossières mais immédiates, de taille, masse, orientation et vitesse. Nous sommes ensuite capable de porter une attention plus prolongée sur un point particulier de la scène, pour en extraire des informations plus détaillées et éventuellement aboutir à une identification complète de l'objet observé.

Les priors extraits d'une labellisation macroscopique, même s'ils peuvent être approximatifs, sont la plupart du temps suffisants pour prendre des décisions cohérentes et rapides[LOR97]. En effet, nous n'avons pas besoin de connaître le type précis d'un véhicule ni son orientation ou vitesse exactes pour pouvoir mettre en œuvre une stratégie d'évitement. Le phénomène d'étiquetage s'inscrit donc de manière naturelle et intrinsèque dans un processus décisionnel où l'information, dans les connaissances acquises et dans la scène présente, est particulièrement riche. Le résultat de la prise de décision peut varier. Il peut s'agir d'effectuer un étiquetage plus précis à l'intérieur d'une zone déjà labellisée. La labellisation est alors arborescente et nous passons alors d'un phénomène de reconnaissance à une identification de l'observé. Certaines régions de la scène perçue peuvent être caractérisées comme peu ou non porteuses d'information. Savoir qu'une zone est peu porteuse d'information, relativement à un problème donné, est une information importante en soi puisqu'elle permet de répartir le temps de traitement sur les autres régions. Dans un cadre routier, l'attention doit être portée sur la route et les autres véhicules, sur la signalisation, et moins sur les régions contenant de la végétation, du ciel, des bâtiments, etc. Une première labellisation permet ainsi de définir des priorités en fonction des ressources disponibles, comme le temps de calcul par exemple. Cet

ordonancement est particulièrement important quand le traitement des données présentes s'inscrit dans un cadre temps réel, et où une décision rapide et cohérente par rapport au contexte est primordiale. Nous aurons l'occasion de reparler de ce point quand nous évoquerons le thème de la veille militaire (voir section 2.1). La labellisation peut également aider à la prise en charge des « phénomènes de surprise ». Un tel phénomène apparaît quand la différence entre le prior sur une partie de la scène et le posterior, est jugée importante. Cet écart témoigne d'une incompréhension locale de la scène. La gestion de cette incompréhension dépendra de l'application. Dans un cadre de veille militaire ou un cadre médical [MGA95], ce type de stratégie est parfois utilisé pour la détection de cible ou de tumeur. Une première labellisation macroscopique de l'image extrait les zones principales à traiter, les étiquettes « mères ». Une seconde labellisation, plus fine, peut ensuite être lancée à l'intérieur des premières zones étiquetées pour extraire les étiquettes « filles ». Une différence de diagnostic entre une ou plusieurs étiquettes filles et leur étiquette mère est représentative d'une cible potentielle, un char camouflé sur fond végétal, ou une tumeur sur un tissu.

Nous appliquerons la labellisation sur des images en niveaux de gris, issues du spectre visible. Les données que nous serons amené à gérer seront des scènes comportant des régions naturelles, typiquement de la végétation, et des régions artificielles, telles que des bâtiments ou des véhicules. Les régions artificielles ont souvent pour principale caractéristique la présence de motifs géométriques, éventuellement répétés sur l'ensemble de l'objet observé. Ces motifs seront soumis à la variabilité de la scène en terme de luminosité, de contraste, de rotation et d'échelle. Les propriétés des fractales nous ont paru adaptées pour répondre à nos objectifs. Historiquement, les fractales ont permis de répondre efficacement à la modélisation d'objets naturels. Elles sont essentiellement connues du grand public par leurs méthodes de synthèse d'images de fougères, de flocons de neige ou de formes imaginaires. Quelques exemples de formes fractionnaires sont donnés dans la Figure 2-0. Pour notre part, ce sont essentiellement les attributs statistiques et de compression de données des fractales qui nous intéresseront. D'une part, les outils statistiques, fondés sur les travaux de Mandelbrot [MAN83], fournissent des mesures discriminantes pour les textures naturelles présentes sur une image. D'autre part, concernant la compression fractale, les travaux de Jacquin [JAC90] basés sur les systèmes de fonctions itérés ont montré la capacité de ce type de méthode pour gérer les redondances d'images à différentes échelles, orientations et conditions d'éclairage. Cet aspect sera la base d'un chapitre de notre étude. Nous expliquerons comment nous sommes inspiré de ce type de compression pour mettre en relief les blocs les plus représentatifs d'une image en termes de redondances, et comment nous l'avons transformé en un processus d'apprentissage. La partie statistique sera utilisée conjointement à des opérateurs plus classiques pour nous permettre de séparer l'artificiel du naturel dans une image à traiter.

Nous venons de voir la problématique théorique ainsi que les objectifs techniques. Nous allons maintenant présenter plus en détail le sujet et le contexte industriel de notre étude. Nous expliquerons ensuite plus précisément ce que sont les fractales avant de présenter un état de l'art sur la labellisation et les méthodes classiques d'apprentissage de textures. Puis nous traiterons de nos travaux à proprement parler. Nous évaluerons les résultats obtenus par notre méthode dans un contexte de reconnaissance d'objets avant de conclure et proposer des perspectives.

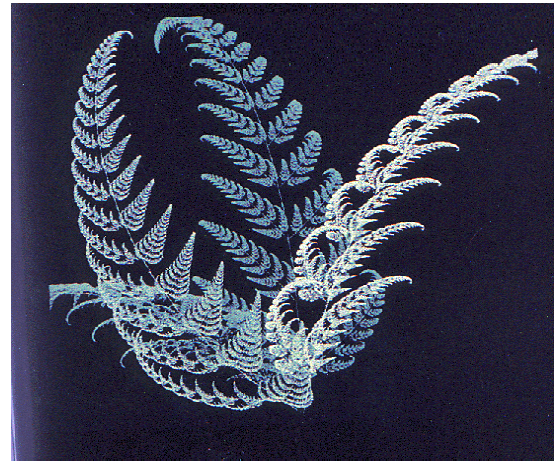
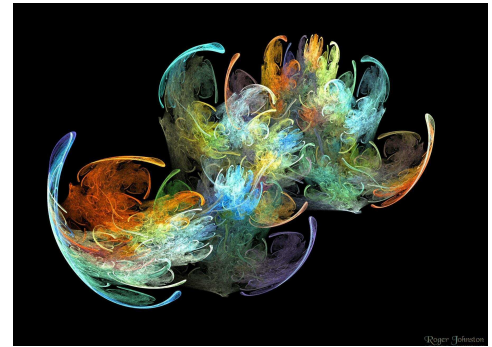


Figure 2-0 Exemples d'objets fractionnaires naturels (à gauche) et artificiels (à droite) [Sources des images : Net 1, Net 2, Net 3]

2.2 Sujet et Contexte Industriel

Notre objectif dans ce travail est de réaliser une labellisation d'une image. Les données traitées seront des images sol/sol ou air/sol basse altitude dans le spectre visible ou proche infra-rouge (IR). L'étude s'inscrit dans le large cadre de la compréhension de scènes complexes. La complexité d'interprétation d'une scène peut relever de différents critères. Il peut s'agir d'un point de vue formel d'une variabilité importante des modèles statistiques présents. On retrouve ce cas lorsque les niveaux de gris des régions d'une image suivent des lois de probabilité différentes. Pour une même loi de répartition, on peut également être soumis à une variabilité importante des paramètres de cette loi. Enfin un élément important à prendre en compte est l'influence de l'éclairage de la scène sur les objets qu'on observe. La complexité peut également être conceptuelle, elle fait alors appel à des notions différentes comme des véhicules, des bâtiments, des personnes ou de la végétation. Notre but sera d'apprendre un certain nombre de ces notions et d'en extraire les éléments représentatifs pour générer une base de connaissances de manière automatique. Enfin notre méthode devra être capable de retrouver ces notions au sein de nouvelles scènes. Son efficacité devra être quantifiée, tant en terme de fiabilité qu'en terme de consommation de ressources de manière à pouvoir la situer par rapport au matériel existant et à venir, et aux besoins de l'industriel.



Figure 2-0 Image de veille marine. A gauche, baigneurs et bateaux à moteur. A droite un voilier en bordure de flaque solaire en contraste négatif.

Les domaines industriels applicatifs liés à la labellisation sont nombreux. Nous citerons notamment ceux de la veille (notamment veille terrestre, les veilles marine ou aérienne posant des problèmes très spécifiques), et de la réalité augmentée. Dans le premier cas, une indentification des régions de l'image permet d'obtenir une estimation du fond. Nous utilisons ici l'expression « estimation de fond » par opposition à la détection de cibles de petite taille : nous voulons découper l'image en régions pour en avoir une vue macroscopique. Cette séparation des données de l'image peut être utilisée dans le but d'augmenter les performances d'une chaîne algorithmique de détection de cible, en se plaçant en amont de celle-ci. Dans le cas où l'environnement dans lequel évolue la cible est très variable, une détection efficace peut s'avérer difficile à mettre en œuvre. Si le fond n'est pas connu, une solution possible est de relaxer les contraintes de l'algorithme pour trouver la cible, par exemple les seuils de détection. Cette stratégie peut alors aboutir à un nombre important de fausses détections. Formellement, l'estimation de fond permet de passer de l'estimation d'une probabilité $P(\text{détection})$ sans a priori, à celle d'une probabilité $P(\text{détection} | \text{fond})$ avec a priori. Dans la pratique la seconde approche donnera des valeurs de probabilité supérieures à celles de la première. En effet, une évaluation du fond permet l'adaptation des paramètres des algorithmes en aval de la scène, à l'environnement de la cible.

Nous avons parlé plus haut d'une autre application de la labellisation des zones d'une image. Il s'agissait de la réalité augmentée. Dans ce contexte, le but est de faciliter la tâche de l'utilisateur, soit en lui présentant différemment l'information, comme c'est le cas dans la colorisation d'images monochromes, soit en ajoutant explicitement des données, par exemple en surimprimant des données géographiques sur un afficheur. L'utilisateur posté sur un terrain qu'il ne connaît pas peut ainsi voir cette lacune comblée, du moins en partie, par ce type d'approche. La Figure 2-0 illustre ce procédé, les images sont issues de l'article [HOL01] qui traite du prototype MARS du centre de recherche naval à l'Université de Columbia. L'opérateur peut accéder au plan des bâtiments dans lesquels il évolue. En extérieur, chaque bâtiment est étiqueté par son nom et sa fonction.

(a)

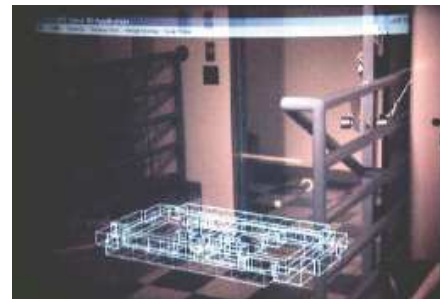


Vue extérieure du prototype MARS. Les bâtiments du campus sont labellisés par leur fonctionnalité (ici, « philosophie » et « chapelle St Paul »).

(b)



Vue détaillée (a) et filtrée (b) d'un bâtiment.



Exemple de guidage d'un utilisateur en intérieur par le prototype MARS.

Figure 2-0 Application de réalité augmentée développée sur le prototype MARS ([HOL01], Columbia University, Naval Research Laboratory).

La réalité augmentée telle qu'elle est présentée dans la Figure 2-0 nécessite une information fine sur la scène observée, telle que des données géographiques ou de cadastre. Or celles-ci ne seront pas toujours disponibles et ce manque rendrait la labellisation impossible. Il existe cependant une application moins contrainte dans le domaine de la réalité augmentée, et se trouvant être d'actualité en terme de compromis performances / besoin de l'opérationnel. Il s'agit de la colorisation de scènes monochromes en couleurs artificielles, notamment dans le cadre de la vision nocturne.

Nous estimons que cette application est peu contrainte dans le sens où le concept caractérisant chaque pixel n'est pas une notion complexe (un véhicule ou objet naturel) mais une couleur. Il n'y a pas de décision automatique à l'issue d'une telle application, il s'agit uniquement de présenter l'information différemment à l'opérateur.

Le but fonctionnel de cette application est d'augmenter la capacité de l'utilisateur à discerner des objets et de diminuer sa fatigue oculaire et psychologique (perte de sommeil et maux de tête lors d'utilisations prolongées). Dans [TOE04], Alexander Toet liste les principales gênes de l'opérateur et propose une solution technique de colorisation d'une scène monochrome pour répondre à ce besoin. Ce type d'approche sera vu plus en détail dans l'état de l'art. La Figure 2-0 illustre un résultat obtenu dans le cadre de cette application.



Figure 2-0 (application militaire Danoise de [TOE04]) : colorisation d'images monochromes, IR bande 2 (thermique). A droite l'image de référence (couleurs et niveaux de gris) qui a servi à coloriser la scène.

Dans ce type d'approche les textures sont classiquement modélisées par un modèle statistique. Or l'interprétation d'une texture, vue localement, peut varier en fonction de son contexte / voisinage immédiat dans l'image. Nous retrouvons fréquemment ce problème, par exemple, quand nous sommes confronté à une région présentant un niveau de gris homogène. Pour labelliser de manière fiable une telle zone, il convient de prendre en compte son voisinage pour lever l'ambiguïté. La Figure 2-0 illustre ce phénomène.

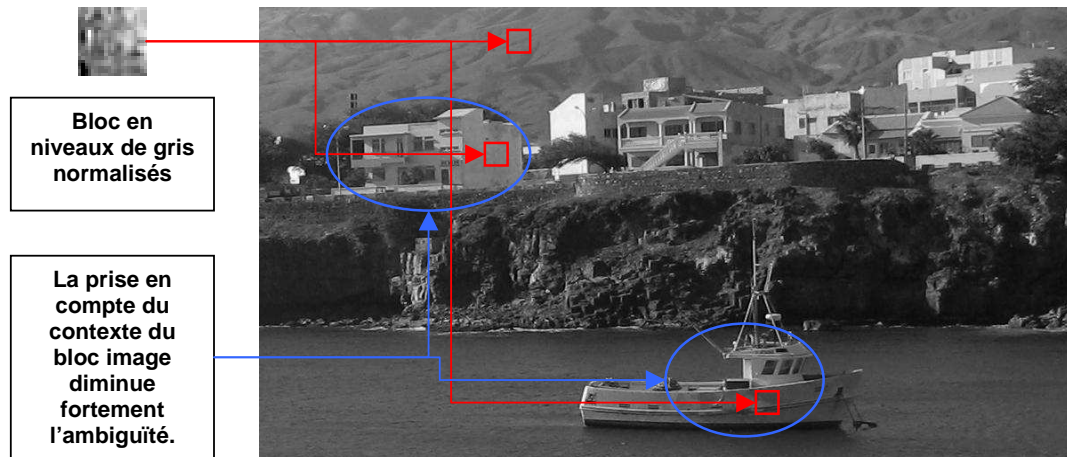


Figure 2-0 Exemple d'ambiguïté d'attribution d'un bloc image parmi trois notions différentes, montagne, bâtiment, bateau.

Les textures sont souvent considérées comme des modèles statistiques. C'est-à-dire que toute fenêtre glissante est sensée renvoyer des attributs caractéristiques similaires à chacun de ses pas. Pourtant bien souvent dans la pratique, l'étiquetage d'une zone de l'image doit faire appel à des données complémentaires, par exemple de voisinage sur l'image, pour obtenir une réponse pertinente. Dans la Figure 2-0, il est difficile de décider à quelle partie de l'image appartient le petit bloc image donné à gauche de la figure, surtout quand les conditions d'éclairage peuvent varier d'une scène à l'autre. Dans le cas présent, le bloc est extrait du bateau au premier plan. Pour pouvoir le reconnaître et

l'étiqueter en tant que navire, et s'appuyer sur des mesures fiables, l'extraction de blocs contenus dans le voisinage immédiat serait nécessaire. Dans la figure, ce voisinage est représenté par les ellipses bleues. Cet aspect composite, de bloc images ou de mesures caractéristiques, devra être présent dans la partie reconnaissance de notre méthode et par conséquent de manière symétrique dans l'étape d'apprentissage. Pour ce faire, nous nous sommes positionné dans la suite des travaux réalisés par Audrey Séropian [SER03, SER05]. La méthode exposée permet l'identification de scripteurs et d'alphabets, latin, arabe ou cyrillique, en se basant sur la compression fractale proposée en 1990 [JAC90], [JAC92]. Le principe était d'établir une base de connaissance de blocs image représentatifs du scripteur ou de l'alphabet en question, de manière à pouvoir le reconnaître automatiquement dans d'autres documents manuscrits. Les données manipulées étaient alors contraintes en termes de géométrie et d'amplitude du signal puisqu'il s'agissait de lignes d'écriture sur des images binaires, écriture noire sur support blanc. Nous allons mettre en œuvre une méthode permettant d'appliquer ce principe à des images de scènes quelconques dans le spectre visible.

2.3 Plan de la thèse

Nous allons commencer notre étude par un état de l'art sur les différentes thématiques liées à nos objectifs. Celles-ci étant assez vastes, nous ciblerons quelques notions essentielles. L'action de labelliser une image correspond à en séparer des portions et à les identifier. Notre état de l'art recouvrera dans ce sens les notions de segmentation, puis la caractérisation de régions texturées par différents opérateurs, pour finalement approfondir un historique sur les opérateurs fractals. Nous aborderons ensuite le cœur de notre étude et détaillerons comment nous nous sommes inspiré de la compression fractale pour construire un processus d'apprentissage de concept. Nous expliquerons comment nous avons élaboré un système de notation permettant de mettre en relief les imagerie les plus représentatives en termes de redondances à différentes échelles sur l'objet à apprendre. Nous nous pencherons ensuite sur la mise en application des données apprises en proposant une méthode de reconnaissance. Celle-ci tiendra compte de l'aspect composite des concepts appris en les considérant comme des sacs de mots. Un objet sera considéré comme reconnu si plusieurs de ses imagerie caractéristiques sont présentes dans une zone restreinte de l'image. Nous évaluerons ensuite les performances des applications de nos travaux sur une base d'images publiques [Net 4] et nous dégagerons les avantages et les limites des méthodes proposées. Enfin nous exposerons nos conclusions et proposerons les perspectives possibles.

3 Etat de l'art, analyse critique et outils

3.1 Introduction

Dans ce chapitre, nous effectuons un état de l'art de l'existant technique et applicatif des méthodologies utilisées dans l'étiquetage de zones image. Il est difficile de parler d'étiquetage sans parler de segmentation. En effet, les problèmes d'identification et de séparation de zones sur une image, même s'ils ne sont pas duaux, sont du moins fortement liés : bien identifier une zone commence par bien la délimiter. A l'inverse, délimiter spatialement des régions différentes sur une image implique, sans obligatoirement les avoir identifiées individuellement, que nous sachions suffisamment les caractériser pour les reconnaître comme différentes et de contenu homogène.

Dans un premier temps, nous nous intéresserons à certaines techniques de segmentation. Nous verrons quelles peuvent être les approches qui permettent de délimiter et affiner une carte de labels. Dans un second temps, nous nous pencherons sur des attributs caractérisant des textures. Nous verrons comment ils permettent de définir deux régions homogènes voisines dans l'image. Nous constaterons que la définition même d'homogénéité n'est pas triviale. Enfin nous détaillerons les opérateurs fractals, de leur historique jusqu'à leurs applications dans des domaines aussi différents que la compression d'image ou la caractérisation de textures.

3.2 Etat de l'art sur la labellisation

3.2.1 Définition d'une zone homogène

L'exemple le plus connu de zone homogène en traitement d'image est l'homogénéité en niveau de gris. Même dans ce contexte restreint il n'est pas toujours évident de choisir quand l'homogénéité entre deux régions voisines est brisée et qu'il faut en déduire une frontière. En effet l'homogénéité peut dépendre du contraste global de l'image, du bruit de capteur, etc. Elle n'est de plus, pas limitée au niveau de gris. On peut effectivement considérer que deux régions sont homogènes par rapport à un opérateur ou une méthode de reconnaissance donnés. Par exemple, la réunion de deux régions ayant des niveaux de gris différents pourra être considérée comme homogène relativement à un opérateur fréquentiel si elles utilisent les mêmes fréquences. De plus, notons que l'homogénéité implique de manière évidente une relation de voisinage. Il s'agit donc d'une réponse « proche » spatialement dans le repère image, et dans les valeurs retournées par un opérateur, en chaque pixel de l'image. La notion de « proche » n'est pas triviale à définir et représente le cœur du problème. Plus formellement, dans [MAI99], Henry Maître définit une région homogène R_i d'une image ($i \in [1..nb \text{ Régions}]$) comme un ensemble de pixels qui vérifie un prédicat. Nous rappelons qu'un prédicat P est une proposition logique dont la valeur dépend de son argument. Les prédicats les plus utilisés sont présentés dans le Tableau 3-0.

Méthode	Définition du prédicat
Contraste sur la région	$P(R_i) = true \Leftrightarrow \max_{R_i} [I(x, y)] - \min_{R_i} [I(x, y)] < \sigma$
Ecart type de la région	$P(R_i) = true \Leftrightarrow \sqrt{\frac{1}{Card(R_i)} \sum_{R_i} [I(x, y) - \bar{I}_{R_i}]^2} < \sigma$ où \bar{I}_{R_i} représente le niveau de gris moyen de la région R_i .
Distance interquartile sur la région	$P(R_i) = true \Leftrightarrow Q_3(I) - Q_1(I) < \sigma$ où Q_1 et Q_3 sont les 1 ^{er} et 3 ^{ème} quartiles des valeurs de I , Q_2 étant la médiane de I .
Différences limitées	$P(R_i) = true \Leftrightarrow \forall (x, y)(x', y') \text{ voisins}, I(x, y) - I(x', y') < \sigma$ dans R_i .
Entropie	$P(R_i) = true \Leftrightarrow - \sum_{n/p(n) \neq 0} p(n) \log(p(n)) < \sigma$ où $p(n)$ est la fréquence d'apparition du niveau de gris n dans la région R

Tableau 3-0 Prédicats classiquement utilisés pour définir une zone homogène sur une image [MAI99].

Dans le cadre d'une labellisation où chaque pixel appartient à une seule classe, l'étiquetage complet de l'image I constitue une partition $\Pi = \{R_i \mid i = 1..N_\Pi\}$ de l'image.

$$\Pi \text{ vérifie } \begin{cases} \forall i, j \in \Pi & R_i \cap R_j = \emptyset \\ \cup_i R_i = I \end{cases} (*)$$

Pour un prédicat donné, il existe un grand nombre de partitions possibles. Il suffit par exemple de diviser une région existante en deux sous régions pour que chacune continue à vérifier le prédicat et que Π vérifie la condition (*) sur l'ensemble de l'image. Des contraintes peuvent être ajoutées pour guider le choix à travers l'ensemble des solutions possibles. Parmi les plus communes, on retrouve un critère sur la taille de chaque région (à maximiser), le nombre total de régions (à minimiser), et la similarité entre régions selon un critère choisi (à maximiser). Cette distance entre régions peut être définie de plusieurs manières. Il peut s'agir, par exemple, de l'écart entre les paramètres statistiques des régions, moyennant un modèle a priori. Il existe plusieurs méthodes pour générer une partition, telles que la croissance, la réunion ou le partage de régions.

3.2.2 Croissance, réunion et partage de régions étiquetées

La méthode de **croissance de région** (*region growing*) prend comme point de départ un certain nombre de germes. Un germe est une zone de l'image où le prédicat choisi est réalisé sans aucune ambiguïté. Les régions vont croître à partir de ces germes. Il peut arriver qu'un pixel puisse être affecté avec une égale confiance à deux régions déjà définies. Généralement l'affectation aléatoire de ce pixel à l'une ou l'autre région sera sans conséquence pour la suite du traitement. Un point sensible est la fusion de régions adjacentes. En effet, fusionner deux régions équivaut à définir une tolérance sur la différence d'adéquation qu'elles ont par rapport à un modèle. Définir cette tolérance n'est souvent pas trivial car nous pouvons rapidement aboutir à une sur ou sous segmentation.

La fusion peut être effectuée soit directement lors de la croissance de région, soit en acceptant dans un premier temps la sursegmentation et en fusionnant les régions dans une phase postérieure à part. Cette méthode n'aboutit pas strictement à une partition puisque l'union de toutes les régions ne recouvre pas systématiquement l'image [ADA94], en effet la croissance de région est initialisée par des graines. Les régions issues de ces graines peuvent ne pas s'étendre jusqu'à un recouvrement complet de l'image. Dans cette approche, certains pixels de l'image peuvent n'appartenir à aucune région obtenue par croissance. Ils appartiennent alors par défaut à la l'étiquette/classe « vide ».

La **réunion de régions** (*region merging*) se rapproche par certains points de la croissance de région en ce sens qu'on peut la considérer comme une croissance de région où chaque pixel est une graine. Nous partons de petites régions, par exemple 2x2 pixels, sur l'image et elles sont fusionnées tant que le prédicat P est vérifié sur l'ensemble des régions de l'image, notamment sur les nouvelles régions issues d'une réunion de régions [ZHU96].

L'approche par **partage de région** (*region splitting*) part de l'image entière à laquelle on affecte initialement une seule étiquette. Tant qu'une région de la partition ne vérifie pas le prédicat, et que la division de la région concernée a encore un sens, région suffisamment grande par exemple, alors on divise la région en plusieurs parties, classiquement deux ou quatre (selon les axes horizontaux et/ou verticaux).

Une méthode tirant parti des deux précédentes est l'**approche pyramidale**. Celle-ci consiste à construire plusieurs graphe d'adjacence de régions et à les organiser de manière pyramidale (Figure 3-0). A l'intérieur d'un même niveau de la pyramide, les régions ont une taille sensiblement identique [REV03], [COC95]. La segmentation par approche pyramidale revient à effectuer un simplification de graphe (recherche de « stable ») tout en prenant en compte les données contenues dans l'image (vérification du prédicat). A chaque itération, un ensemble de sommets « survivants » doit être calculé selon des propriétés d'adjacence entre sommets d'un même niveau, et de connexité entre niveaux de la pyramide. Des méthodes de construction rapide de segmentations pyramidales sont proposées dans [TAR75], [NAJ06] et [HOR76].

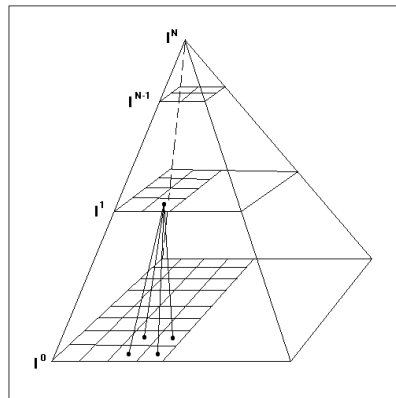


Figure 3-0 Représentation pyramidale des régions d'une image. Le niveau 0 correspond à l'image entière (image issue de [Net 6])

3.2.3 Segmentation et approche MDL

L'approche **MDL** (*Minimum Description Length*) a été introduite par Rissanen en 1978 [RIS78]. C'est un principe important de la théorie de l'information et de l'apprentissage.

Elle repose sur le postulat que toute information peut être représentée par un mot (k-uplet de symboles, une suite de bits par exemple) de taille minimum. Ce principe a été appliqué au domaine de la segmentation d'image dans [LEC89], [ZHU96]. Les données à compresser dans la segmentation sont par exemple la description du polygone délimitant chaque région ainsi que les paramètres caractérisant la répartition des niveaux de gris des pixels de la région (relativement à un modèle statistique donné). Dans [GAL04], Frédéric Galland propose une méthode de segmentation d'image utilisant les MDL dans le cadre d'une grille de contours actifs. Le principe général de sa méthode est la déformation itérative de la grille par déplacement de nœuds et fusion des régions (suppression de segment dans la grille). La grille de contours actifs ne peut que se simplifier (en terme de complexité de graphe), l'ajout de nœud ou de région n'est pas possible dans l'état de la méthode proposée par l'article. Une énergie globale régit l'acceptation ou le refus d'une modification de la grille. Elle est composée d'une énergie interne liée au contenu des régions, c'est-à-dire la répartition des valeurs des niveaux de gris des pixels, et d'une énergie externe dépendant de la complexité de la grille en tant que graphe. L'avantage de ce type de méthode est que les paramètres utilisés sont soit de haut niveau, pour le choix du modèle statistique a priori, soit suffisamment souples pour ne pas rendre la méthode instable à de petits changements de valeurs, c'est le cas pour la taille de la grille d'origine qui est entièrement déformable et dont les zones polygonales peuvent fusionner et éventuellement être incluses les unes dans les autres. Une des applications de cette méthode a été la segmentation automatique d'images SAR. Le modèle de répartition des valeurs de niveaux de gris des pixels pour une image radar est connu, il s'agit d'un speckle, modélisé par une loi gamma.

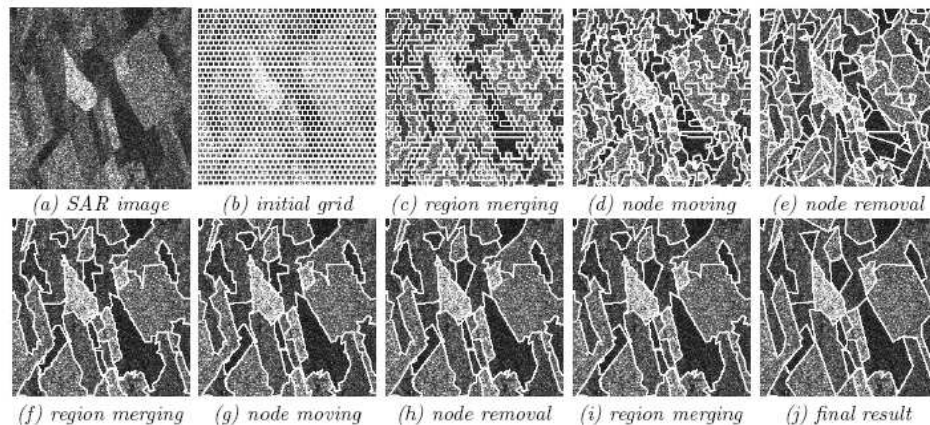


Figure 3-0 (issue de [GAL04]) Etapes de la segmentation par grilles de contours actifs selon la méthode de F. Galland. On part d'une fine (sursegmentation), puis alternativement les nœuds sont déplacés et les régions fusionnées jusqu'à stabilisation.

La Figure 3-0 montre les différentes étapes présentes dans la méthode de Galland. L'image a) représente l'image SAR initiale. En b) une grille de contour actif a été ajoutée, elle est régulière par défaut. Vient alors une étape de fusion de régions (c), puis les opérations sur les nœuds : déplacement (d), suppression des nœuds inutiles (e), et la fusion de régions (f). Les étapes (g) à (i) refont une itération sur les déplacements de nœuds et la fusion de régions. Le résultat final de la segmentation est visible en (j).

Dans [DEL06], la méthode de Galland a été perfectionnée et généralisée à une approche sans modèle a priori sur la répartition des niveaux de gris de chaque région. Les histogrammes des régions segmentées à une itération donnée sont représentés par

des fonctions étagées. C'est sous cette forme qu'ils sont utilisés pour les calculs d'énergie. La répartition des niveaux de gris dans les régions peut alors être définie par des modèles très divers. La Figure 3-0 montre un exemple de modélisation d'histogramme par fonction étagée.

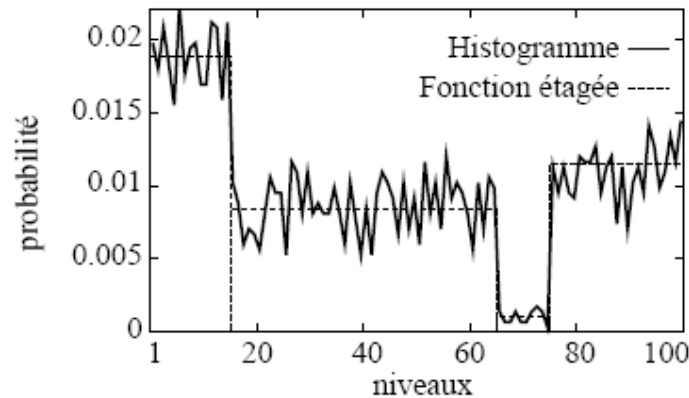


Figure 3-0 Exemple de simplification d'histogramme en 4 classes.

Le paramètre à prendre en compte est alors le nombre « d'étages » que l'on souhaite dans la représentation de l'histogramme (dans la Figure 3-0 : quatre étages). Cette approche permet, de plus, de prendre en compte des images qui reposent sur plusieurs modèles. Dans le cas de l'imagerie radar, nous savons que nous avons affaire à du signal type speckle, donc modélisé par une loi gamma. De même dans une image en IR 3 (thermique), les régions observées peuvent être essentiellement homogène sur certains capteurs. Dans ces deux derniers cas, la méthode par histogramme étagé n'apporterait pas un gain significatif de performance. Par contre, dans une image naturelle en domaine visible, les textures présentes, et donc les modèles qui les caractérisent, peuvent être très variées et la généralisation proposée par [DEL06] aurait un intérêt certain.

Nous nous sommes penché dans cette section sur la définition de l'homogénéité de régions image et sur les principales méthodes pour délimiter spatialement ces régions. Après ce regroupement de données en groupes homogènes, nous allons maintenant nous pencher sur les approches les plus connues pour identifier ces régions par des attributs texture.

3.3 Etat de l'art sur les zones texturées

Dans cette section, nous nous intéressons à la caractérisation de zones texturées. Le point de départ est donc une zone image spatialement délimitée et le but est de caractériser son contenu. Cette étape pourra faire appel à des outils très variés basés sur les statistiques de différents ordres sur les pixels, sur des aspects fréquentiels ou structurels. Nous étudierons une partie de ces outils et nous détaillerons pour chacun d'eux les avantages et inconvénients relativement à la problématique de caractérisation de textures.

3.3.1 Introduction et discussion sur la définition d'une texture

Définir ce qu'est une texture n'est pas aisé. Cette notion est liée d'une part à la notion d'homogénéité, que nous avons abordée au chapitre 3.2.1 et qui assure la séparabilité des différentes textures dans une image, d'autre part à la notion de caractérisation et de discrimination. Nous voulons en effet dans certains cas être capable de reconnaître une même texture dans différentes scènes avec, par exemple, des conditions d'éclairage variables. Dans le spectre visible et sur des images monochromatiques, une zone homogène selon les niveaux de gris est peu porteuse d'information en terme de texture. Cette zone n'est caractérisée que par la moyenne de ses niveaux de gris, elle-même dépendante directement des conditions d'éclairage. Ainsi lorsqu'on veut faire l'étude texturale d'une image, il convient dans un premier temps de se poser la question si cette image contient de la texture. Considérons la Figure 3-0 (images issues de la base d'images ROBIN, robin.inrialpes.fr), elle représente des scènes dans le spectre visible et IR (bande 3 / thermique). Un même objet peut être porteur d'information texturale dans un spectre et pas dans un autre. C'est le cas notamment de la végétation. Dans cette figure, on constate que la végétation en domaine visible est fortement texturée alors qu'elle n'est qu'une zone homogène gris clair en IR3.

Un autre paramètre essentiel intervenant dans la texture est l'échelle d'observation [COC95]. Prenons le cas d'une ville. Un piéton y percevra un nombre important de zones texturées : murs de briques, bitume, béton, etc. Par contre une vue aérienne de cette même ville peut permettre de la considérer comme un seul et même objet, référencé par la texture « tissu urbain ». Considérons un autre exemple : un mur de briques. Une observation à courte distance nous fera considérer l'intérieur des briques comme une texture particulière (ou région homogène) et les séparations entre les briques comme un autre élément de la scène. A l'inverse, une observation plus globale du mur met en évidence la redondance du motif « brique » pour former le mur dans son intégralité.



Figure 3-0 Illustration du changement de textures pour une même scène, observée en visible en haut et en infrarouge en bas (bande IR3). Les rectangles correspondent à la vérité terrain (détection de personnes) liée à la base d'images (ROBIN).

Un autre paramètre intervenant dans l'interprétation d'une texture peut être son voisinage dans l'image. Une texture seule, isolée, peut ne rien signifier, ou bien être ambiguë entre plusieurs interprétations possibles. Des informations comme les textures voisines, les a priori sur l'image elle-même et la scène sont autant d'éléments qui peuvent guider l'interprétation.

Notons également que certaines textures peuvent avoir une dynamique importante au fil du temps, comme c'est le cas pour la mer par exemple. L'aspect de l'eau peut varier [GOU07] fortement d'une scène à l'autre (la densité de vagues, leur taille, le nombre d'ondes, leurs directions imposées par les courants maritimes, qui composent localement le fluide). Cette dynamique et la réflectance en font de plus un matériau à apparence très variable selon les conditions d'illumination.

Parmi les nombreux documents traitant de la texture, certains la caractérisent, d'autres permettent de la synthétiser. Réussir l'une de ces deux tâches n'implique pas nécessairement de parvenir à la seconde. Nous pouvons par exemple caractériser une texture par un ensemble d'opérateurs très différents. Nous pouvons alors dire que nous avons réussi dans ce cas à définir une région dans un espace qui semble approprié pour représenter la texture considérée. Mais cela n'implique pas pour autant qu'on sache synthétiser la texture en question. A l'inverse on peut être en mesure de synthétiser une texture de manière réaliste et avoir des difficultés à la reconnaître de manière fiable. Dans [SOA01], l'auteur reconstruit une texture à partir d'un échantillon en se basant sur un LDS (Linear Dynamic System) qui peut être vu comme un filtre de Kalman sur une séquence d'images (voir Figure 3-0). Autre exemple : il est aisé de générer l'image d'une texture d'eau à partir d'un ensemble d'ondes dont les paramètres sont tirés aléatoirement. Il est en revanche beaucoup plus difficile de faire le travail inverse, à savoir retrouver le nombre des ondes du fluide considéré et leurs paramètres à partir d'une suite d'images donnée.

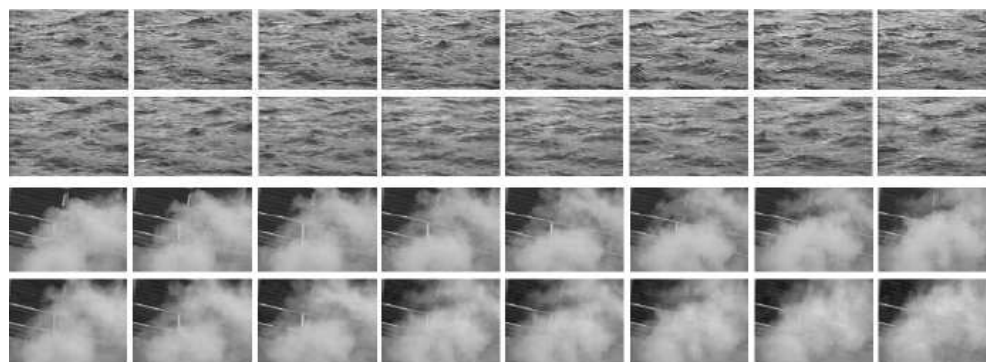


Figure 3-0 Textures dynamiques reconstruites à partir de LDS (source : [SOA01]).

Les méthodes de reconnaissance de textures sont classiquement regroupées en structurelle, statistique, approche modèle, et approche transformée.

Dans l'**approche structurelle**, on distingue deux éléments principaux [HAR79], d'une part, les primitives à utiliser, d'autre part, les relations spatiales qui lient les primitives entre elles. Hormis l'aspect analyse, plutôt adaptée aux textures macroscopiques, cette approche peut se montrer efficace pour une tâche de synthèse de textures. Un autre avantage de ce type de méthode est qu'elle permet d'avoir une bonne vision symbolique de la texture étudiée.

Dans l'**approche statistique**, la texture est vue comme la réalisation d'un processus stochastique. Le but est alors d'en extraire des attributs statistiques. Les données sources de ces attributs peuvent être les pixels eux-mêmes (cas le plus classique, premier ordre) ou les couples de pixels, comme c'est le cas pour la matrice de cooccurrence (mesure du second ordre) [HAR79]. Les mesures du second ordre ont montré leur efficacité dans le domaine de la discrimination de texture par rapport à une perception humaine [JUL75]. La matrice de cooccurrence a été utilisée dans l'imagerie biomédicales dans [LER93, STR95] et a montré son efficacité par rapport à des méthodes de transformée par ondelettes dans le domaine de la classification de textures [VAL98].

L'**approche modèle** repose sur des modèles stochastiques (parmi lesquels on retrouve les fractales). Les paramètres du modèle sont estimés et utilisés pour l'analyse de la texture. Dans la pratique, ces méthodes sont relativement coûteuses en temps de calcul. Les méthodes fractales se sont montrées adaptées à la description de phénomènes naturels comme le précisent [PEN84, CHA95 et KAP95].

Dans l'approche s'appuyant sur les **transformées**, on cherche à représenter la texture dans une autre base que le domaine spatial de l'image. Le but est alors le plus souvent de trouver des vecteurs de la base qui soient les plus porteurs d'information pour reconnaître la texture. On retrouve notamment dans cette catégorie les méthodes s'appuyant sur la transformée de Fourier, les filtres de Gabor et les ondelettes.

Ces différentes familles ne sont pas disjointes. Une méthode de caractérisation de texture peut en effet appartenir à plusieurs de ces familles. On peut par exemple avoir une approche *statistique* après avoir exprimé le signal dans un espace particulier moyennant une *transformée* donnée.

Nous allons maintenant voir quelques opérateurs classiques et étudier leurs limites et domaines d'application.

3.3.2 Matrice de cooccurrence

La matrice de cooccurrence est un outil d'ordre deux (on ne parle pas ici d'ordre deux au sens moment comme la variance, mais du fait qu'on mesure des relations entre objets qui vont par *paires*, en l'occurrence des couples de pixels). Dans une zone donnée de l'image, la matrice de cooccurrence mesure le nombre de fois où un prédicat liant deux pixels est vérifié. Classiquement, ce prédicat vaut vrai quand $I(p_1) = I(p_2)$, où $I(p_i)$ est le niveau de gris du pixel p_i . Dans le cas où on considère 256 niveaux de gris, la matrice de cooccurrence est donc de taille 256x256. Les couples (p_1, p_2) sont généralement liés par un vecteur déplacement T , par exemple $T = (0, +1)$ représente des couples où p_1 est le pixel actuel et p_2 son voisin de droite. Le plus souvent on choisira de sous échantillonner à quelques niveaux de gris (4 à 8 selon [COC95]).

De par sa définition, la matrice de cooccurrence, dans l'état où nous l'avons décrite, est sensible à un certain nombre de paramètres fortement variables.

Les normes couleurs utilisées vont du bleu foncé pour les valeurs les plus faibles, au rouge foncé pour les valeurs les plus élevées.



Figure 3-0 Echelle de couleurs utilisées dans ce document. Les valeurs faibles sont en bleu et les valeurs élevées sont en rouge foncé.

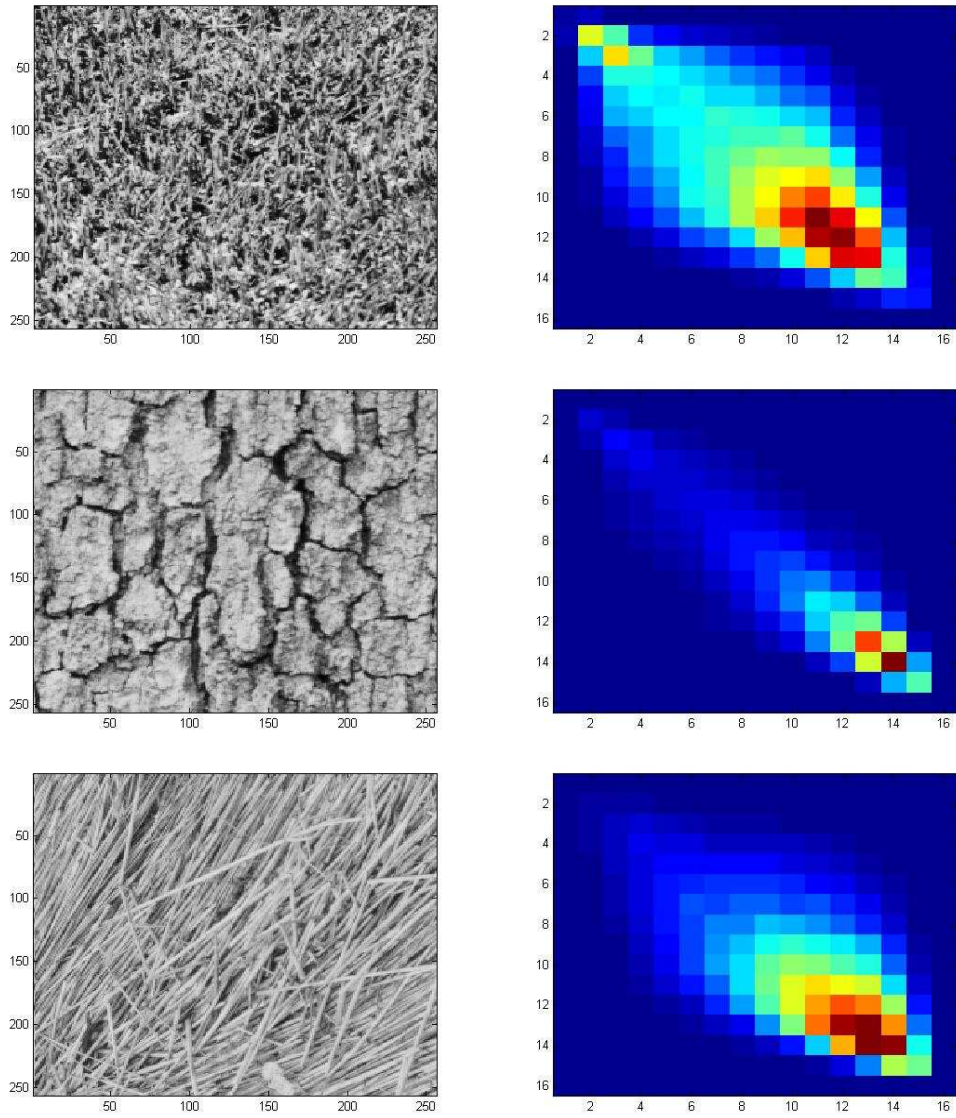


Figure 3-0 A gauche, échantillons d'images de la base Brodatz, à droite : leurs matrices de cooccurrence sur 16 niveaux de gris.

Un changement en luminosité d'une texture se traduit par une translation des valeurs non nulles dans la matrice de cooccurrence le long de la diagonale, vers le coin bas droit si on augmente la quantité de lumière, vers le coin haut gauche si on la diminue.

L'augmentation du contraste de la texture aura pour effet de « dilater / éclater » plus ou moins la signature de la texture dans la matrice de cooccurrence. On en entend par « signature » de la texture, relativement à la matrice de cooccurrence, l'ensemble des éléments non nuls de cette matrice (tache rouge-orange-verte dans la Figure 3-0).

Le changement d'échelle de la texture peut être assimilé à un moyennage des valeurs de l'image suivi d'un sous échantillonnage. Cela se traduira sur l'image par un ramassement de la signature vers ses valeurs les plus élevées (représentées en rouge dans la Figure 3-0).

La matrice de cooccurrence, par sa dépendance au vecteur T , dépend fortement de l'orientation. Pour être indépendant de ce paramètre, il est par exemple possible de calculer plusieurs matrices selon différentes orientations pour T et de prendre par exemple celle qui possède la plus grande énergie. Une autre possibilité pour obtenir l'indépendance en rotation est d'estimer l'orientation, de la tester et d'effectuer un recalage en rotation en prétraitement (voir chapitre 5.3.2.2).

Il est possible de définir une distance entre deux textures à partir de leurs matrices de cooccurrence. Pour ce faire, il suffit de calculer la distance euclidienne, en considérant une matrice de cooccurrence comme un vecteur d'un espace à N^2 dimensions, N étant le nombre de niveaux de gris utilisés. Dans le cas où les matrices concernent des textures dont les dimensions image sont différentes, on peut soit normaliser les valeurs des matrices de cooccurrence entre 0 et 1, soit considérer uniquement des matrices binarisées (avec un seuil à 0 : le couple de pixels apparaît au moins une fois dans l'image considérée). On ne compare alors que les taches binaires constituées par les signatures des textures. L'intérêt de cette approche est qu'on peut, lors de l'apprentissage, considérer des échantillons de texture de grande taille. Nous avons alors une vue statistiquement significative de l'ensemble des couples de pixels possibles. Le masque représentant la version binarisée de la matrice de cooccurrence constitue alors une enveloppe maximale. Lors de la reconnaissance, des échantillons de plus petite taille seront utilisés. Leurs signatures dans la matrice de cooccurrence binarisée devraient alors être incluses dans la signature issue de l'apprentissage, pour peu que les textures correspondent.

On définit un score de ressemblance entre deux textures comme le nombre d'éléments positifs du masque correspondant au ET logique entre les matrices binarisées (normalisé par le nombre d'éléments positif du masque correspondant à la matrice binarisée issue de l'apprentissage). Ce score est représentatif (relativement aux matrices de cooccurrence) de la proximité entre une texture apprise et un échantillon de texture à reconnaître. En effet, un score proche de 1 signifie un bon matching entre les textures apprise et reconnue. A l'inverse, une valeur nulle implique que les signatures n'ont aucun élément en commun. Un score nettement supérieur à 1 signifie que l'apprentissage . Au final, à partir d'une matrice de cooccurrence d'une texture apprise et d'une image représentant une scène composée, on est en mesure de générer une carte de distance (de taille égale à l'image d'entrée) par rapport à la texture apprise. Ceci se fait en parcourant l'image d'entrée par une fenêtre glissante. A chaque pas, on compare la matrice de cooccurrence du contenu de la fenêtre avec celle issue de l'apprentissage et on en déduit la distance comme définie plus haut.

Nous avons implémenté et testé cette méthode pour la reconnaissance de forêt sur une vidéo proche IR. Ici l'apprentissage est effectué sur la première image de la vidéo sur une zone forestière au premier plan. A chaque nouvelle image, on établit la carte de distance locale comme définie plus haut. Le résultat fourni en sortie est une vidéo dont la moitié supérieure de chaque image représente la vidéo d'entrée et la moitié inférieure représente la carte de distance entre la forêt apprise et la réponse locale en chaque point de l'image d'entrée. Concernant les normes des couleurs, le rouge correspond à une correspondance forte (distance proche de zéro), le bleu une correspondance faible.

La vidéo complète peut être trouvée à l'adresse suivante :

http://www.math-info.univ-paris5.fr/~dolez/these/resultats/reco_foret_mat_cooc.avi

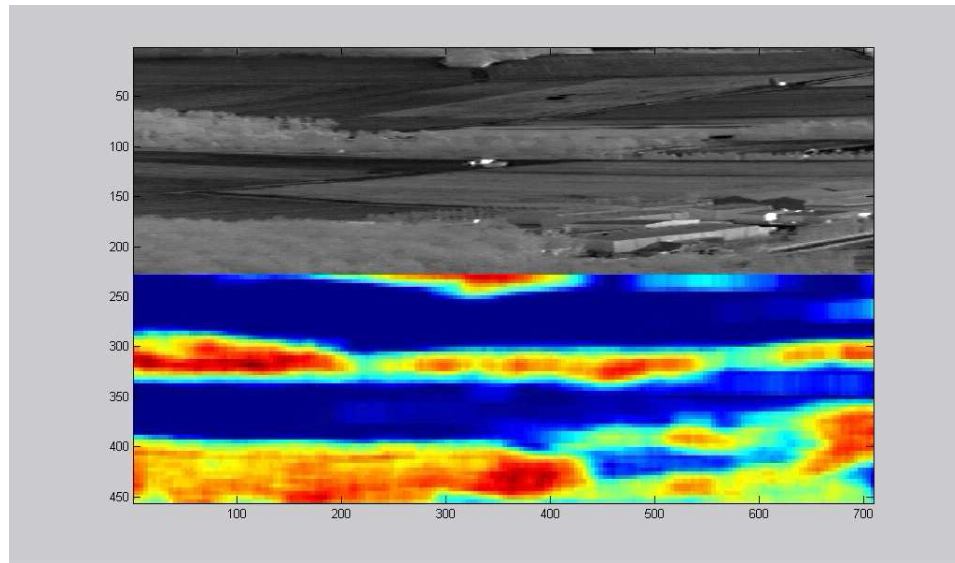


Figure 3-0 Détection de forêt sur une séquence IR hélicoptère. La partie supérieure de l'image représente la vidéo d'entrée. La partie inférieure de l'image représente la carte de score local par rapport à un apprentissage sur des zones de forêt. Une couleur jaune-rouge représente un score élevé.

3.3.3 Filtres de Gabor

Dans [JAI90], l'auteur propose de caractériser une texture sur une image par des filtres de Gabor. D'un point de vue cognitif, l'auteur souligne que ce type de méthode rejoint le mécanisme du système visuel humain de par son approche multifiltre. A chaque filtre correspondent une échelle et un angle de rotation. D'un point de vue perceptif, le système visuel humain différencie avec précision des orientations à partir de 5° . On rappelle qu'un filtre de Gabor correspond à une sinusoïde 2D orientée et de fréquence donnée, pondérée par une enveloppe gaussienne. Après application du filtre sur une image, nous obtenons une carte d'énergie de même dimension que l'image en question. Les énergies locales les plus fortes correspondent aux zones de l'image répondant fortement au filtre appliqué.

Pour paramétrer le filtre, on considère que la texture à reconnaître est comprise dans une fenêtre carrée de côté N_c , où N_c est une puissance de 2. Les fréquences radiales correspondent à $1\sqrt{2}$, $2\sqrt{2}$, $4\sqrt{2}$ et $(N_c/4)\sqrt{2}$ cycles par largeur d'image. Les orientations des filtres se font tous les 45° , et les bandes passantes toutes les octaves, ce qui correspond aux sensibilités physiologiques pour la bande passante. L'auteur rappelle que la bande passante, en octaves, d'une fréquence f_1 à une fréquence f_2 est donnée par $\log_2(f_1/f_2)$. Concernant les orientations, l'œil humain regroupe les textures par paquets d'orientations de 5° (les 45° sont donc a priori insuffisants, mais faire tous les 5° prendrait plus de temps de calcul). Les petites fréquences radiales ($1\sqrt{2}$ et $2\sqrt{2}$) sont laissées de côté pour économiser du temps de calcul.

L'auteur dispose alors d'un ensemble de filtres. Pour choisir les filtres qui caractérisent une texture, l'auteur démontre dans un premier temps que les filtres utilisés constituent une base orthogonale à partir de laquelle il peut représenter relativement fidèlement

l'image. C'est-à-dire qu'il est possible de reconstruire fidèlement l'image si on dispose pour chaque filtre, vecteur unitaire de la base, du coefficient qui lui est associé. Nommons $\hat{s}(x,y)$ une reconstruction partielle de l'image à partir d'un sous ensemble d'images filtrées, et $s(x,y)$ la reconstruction complète à partir de toutes les images filtrées. L'auteur définit le coefficient de détermination R^2 , basé sur la somme des carrés des différences entre l'image reconstruite partiellement et la « reconstruction complète » de l'image, tel que R^2 tende vers 1 quand $\hat{s}(x,y)$ tend vers $s(x,y)$.

$$R^2 = 1 - \frac{SSE}{SSTOT} = 1 - \frac{\sum_{x,y} [\hat{s}(x,y) - s(x,y)]^2}{\sum_{x,y} s(x,y)^2}$$

A partir de là, il constitue un ensemble de filtres, par un algorithme *forward*, jusqu'à satisfaire la contrainte $R^2 > 0,95$. Une fois les filtres trouvés, l'auteur les applique à l'image. Les réponses de l'application de ces filtres sont ensuite utilisées dans la méthode de clusterisation présentée dans [JA188] pour segmenter l'image en question. La Figure 3-0 illustre un exemple d'extraction de texture par cette méthode.

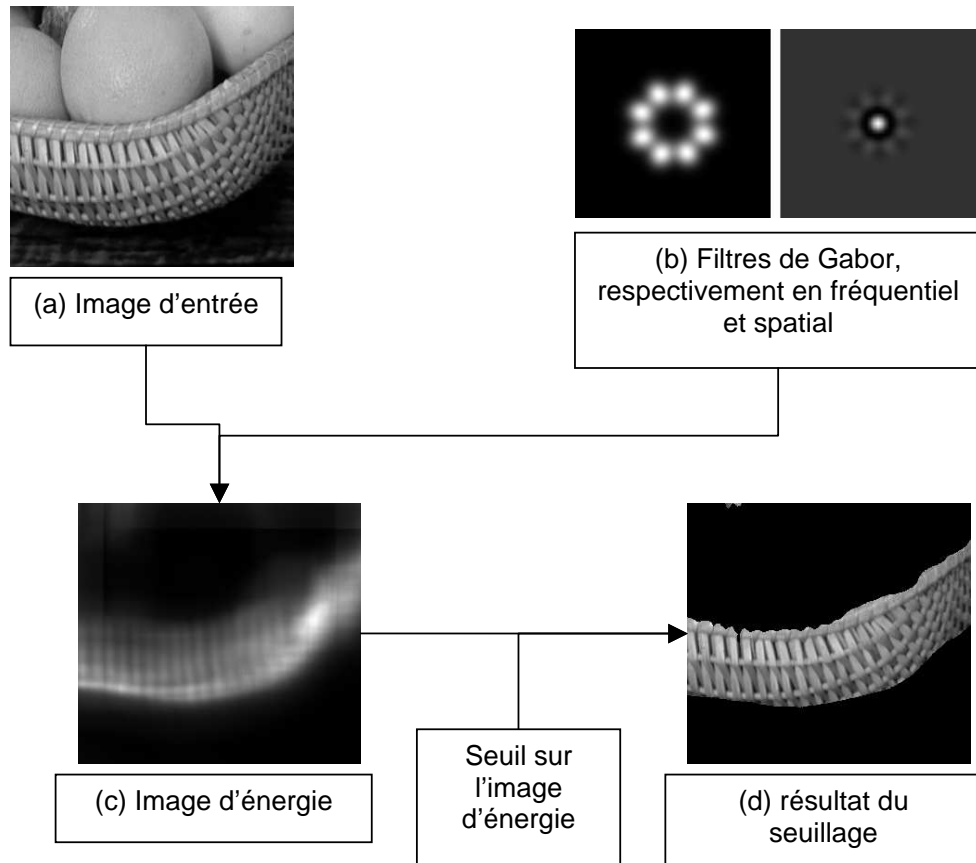
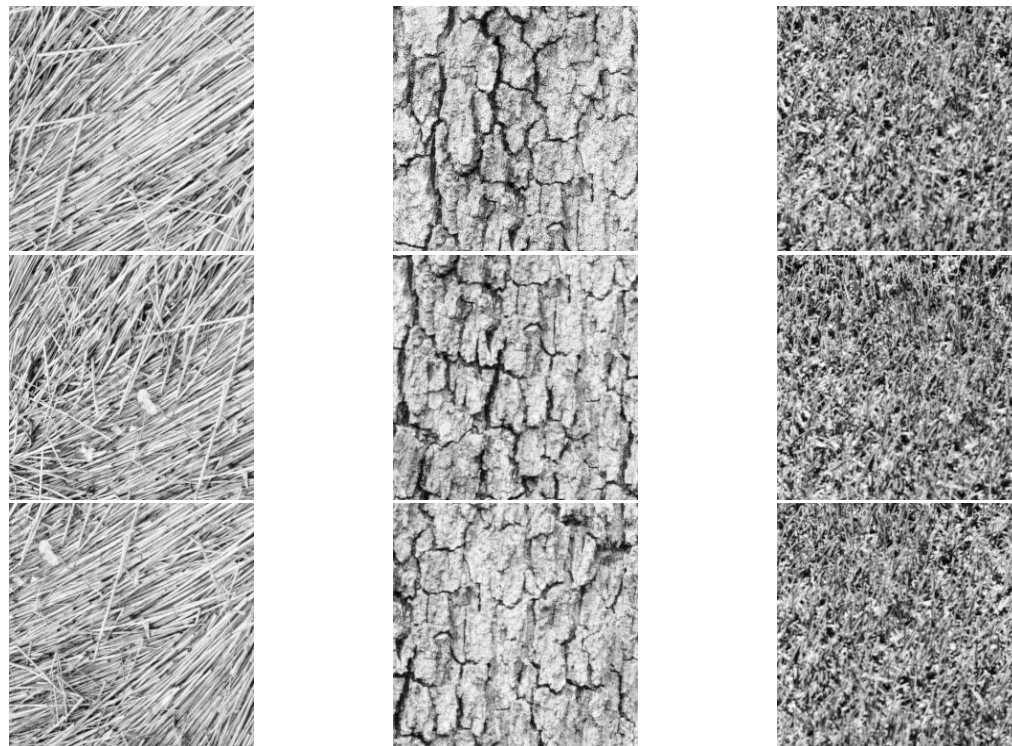


Figure 3-0 (a) Image d'origine, (b) filtre appliqué (représentation fréquentielle et spatiale), (c) image d'énergie calculée à partir de la réponse du filtre, résultat de l'image d'énergie seuillée, masque appliqué à l'image d'origine.

Les filtres de Gabor, utilisés tels quels, permettent d'obtenir une segmentation efficace d'une zone de l'image mais à condition de connaître précisément l'échelle de l'objet à reconnaître, l'orientation et les fréquences qui rentrent en ligne de compte pour le caractériser. Ils paraissent donc appropriés pour séparer des régions, mais moins pour les apprendre et les identifier sur de nouvelles images.

3.3.4 Les Moments

Les moments sont des mesures très utilisées pour caractériser des textures. Il existe des moments statistiques (moyenne, écart-type). Ils permettent d'avoir une approche sur la variabilité des valeurs des pixels de l'image considérée. Les moments de Zernike et de Legendre permettent d'exprimer l'image dans une base orthogonale. Ce qui signifie que si on calcule un nombre suffisant de ces moments, nous serions en mesure de pouvoir reconstruire l'image. Parmi les revues traitant des moments, on citera [TUC94].



ET	Sk	Kt
36.13	-0.67	2.85
35.76	-0.64	2.81
36.91	-0.62	2.73

ET	Sk	Kt
46.94	-0.94	3.06
47.80	-0.90	2.93
41.43	-1.05	3.64

ET	Sk	Kt
49.66	-0.30	2.32
51.20	-0.27	2.23
49.13	-0.31	2.32

Figure 3-0 Mesures de moments d'ordre sur des échantillons de textures issues de la base Brodatz. Chaque colonne d'images correspond à une même texture. On dispose de 3 exemplaires de chaque texture. Les colonnes de chaque tableau, en bas, représentent respectivement les réponses des opérateurs écart type, skewness et kurtosis pour chaque échantillon.

La Figure 3-0 illustre une caractérisation de texture par moments d'ordre 2 3 4, respectivement écart type, asymétrie (*skewness*) et kurtosis. On a appliqué les mesures sur 3 textures différentes, chaque texture étant ici représentée par 3 exemplaires.

Lorsque nous observons les mesures calculées ici, il est important de noter, d'une part la stabilité d'un échantillon à l'autre, d'autre part la séparabilité des modèles de texture (discrimination inter classes). Dans le cas présent, nous avons toujours au moins un attribut qui permet de séparer les classes entre elles. Prenons les échantillons d'écorces de bois et d'herbe (respectivement seconde et troisième colonne d'images), l'écart type semble insuffisant pour discriminer ces deux classes. Par contre les deux autres attributs permettent de lever l'ambiguïté avec notamment une asymétrie (*skewness*) proche de 1 pour l'écorce et d'approximativement -0,3 pour l'herbe.

3.3.5 Filtres de Laws

Dans le cadre des méthodes de caractérisation de textures statistiques, nous pouvons citer les filtres de Laws [LAW79]. Ces filtres 2D sont construits par convolution de filtres 1D. Sur une fenêtre 1D de 5 pixels, ces filtres 1D ont la configuration donnée dans le Tableau 3-0.

L5	1 4 6 4 1
E5	-1 -2 0 2 1
S5	-1 0 2 0 1
W5	-1 2 0 -2 1
R5	1 -4 6 -4 1

Tableau 3-0 Configuration des filtres 1D de Laws en monodimensionnel.

Les lettres L, E, S, W, R sont indiquées pour Level, Edge, Spot, Wave, et Ripple. En convoluant ces 5 filtres entre eux deux à deux, nous obtenons 25 filtres 2D. Ces derniers sont répertoriés dans le Tableau 3-0.

L5L5	E5L5	S5L5	W5L5	R5L5
L5E5	E5E5	S5E5	W5E5	R5E5
L5S5	E5S5	S5S5	W5S5	R5S5
L5W5	E5W5	S5W5	W5W5	R5W5
L5R5	E5R5	S5R5	W5R5	R5R5

Tableau 3-0 Configuration des filtres de Laws en bidimensionnel. Ces filtres 2D sont des convolutions des filtres 1D.

La Figure 3-0 illustre des réponses à des filtres de Laws sur des scènes complexes en domaine visible. Dans l'exemple présenté, la roche et la végétation renvoient des réponses approximativement similaires.

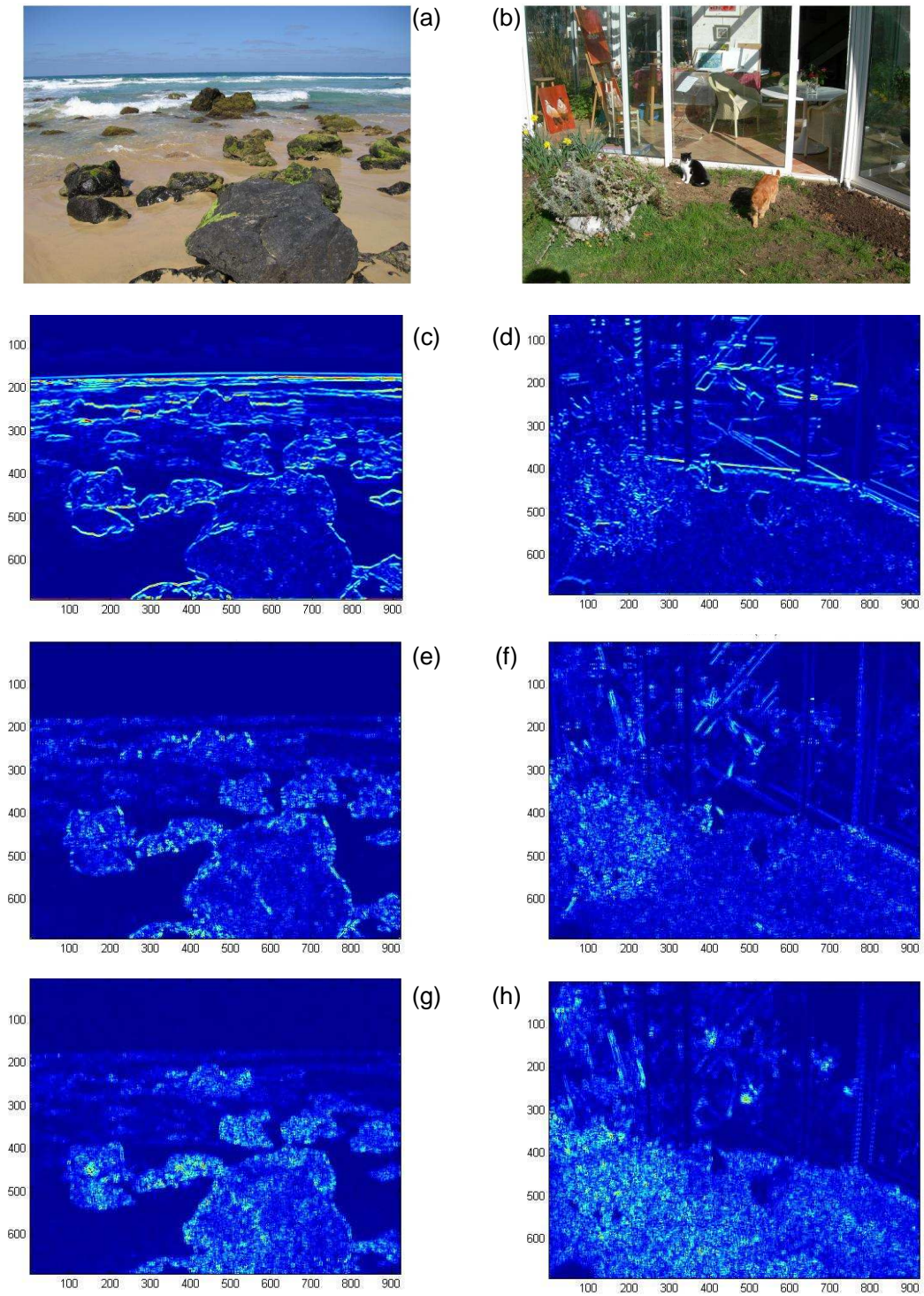


Figure 3-0 (a) et (b) sont les images d'entrée. (c), (e), (g) (respectivement (d), (f), (h)) sont les réponses des filtres de Laws L5E5, S5E5 et R5S5 à l'image (a) (respectivement image (b)) en niveaux de gris. Le premier filtre met en relief les contours à forte luminosité, les deux autres portent sur les points isolés et les crêtes (ridges).

De par leur nature, ces filtres sont fortement liés à l'échelle et à l'orientation de la texture observée. Certains sont même directement dépendants du niveau de gris moyen. Ils pourraient en revanche servir de point d'entrée pour des méthodes de segmentation automatique. Pour prendre en charge plusieurs échelles, il est possible de définir les L5 E5 S5 W5 R5 pour des tailles autres que 5. Concernant la dépendance en rotation, on peut utiliser les filtres du Tableau 3-0 en leur appliquant des rotations selon un pas angulaire constant. Ces opérations sont coûteuses en temps de calcul et la levée de la dépendance de paramètres tels que l'échelle et la rotation n'est alors effectuée que par le parcours d'un nombre important de valeurs pour ces paramètres.

3.3.6 Caractéristiques fréquentielles de la texture

Le spectre de Fourier permet d'observer l'image dans l'espace des fréquences. Différentes mesures peuvent être extraites de cette représentation, telles que la variance du spectre, son énergie, sa valeur moyenne ou encore son amplitude maximale. Ces mesures sont récapitulées dans le Tableau 3-0.

Nature du paramètre	Expression
Variance du spectre de Fourier	$\text{varSpecFourier}(I) = \text{var}(F'(I))$
Energie du spectre de Fourier	$\text{enerSpecFourier}(I) = \sqrt{\sum \text{norm}(F'(I))^2}$
Moyenne du spectre de Fourier	$\text{moySpecFourier}(I) = \text{moy}[\text{norm}(F'(I))]$
Amplitude max du spectre de Fourier	$\text{maxSpecFourier}(I) = \max[\text{norm}(F'(I))]$

Tableau 3-0 Quelques mesures calculées à partir du spectre de Fourier. I est l'image d'entrée, $F'(I)$ est la transformée de Fourier de l'image I , dans laquelle on a annulé la valeur centrale du spectre de Fourier de I noté $F(I)$. $\text{norm}(\cdot)$ prend une matrice de complexes en entrée et renvoie une matrice de même taille, où chaque élément est la norme du nombre complexe correspondant dans la matrice d'entrée.

où $F'(I)$ est la transformée de Fourier de l'image I , dans laquelle on a annulé la valeur centrale du spectre de Fourier de I noté $F(I)$. La Figure 3-0 illustre les réponses obtenues classiquement avec ce type d'opérateurs sur une scène complexe (rade de Toulon).

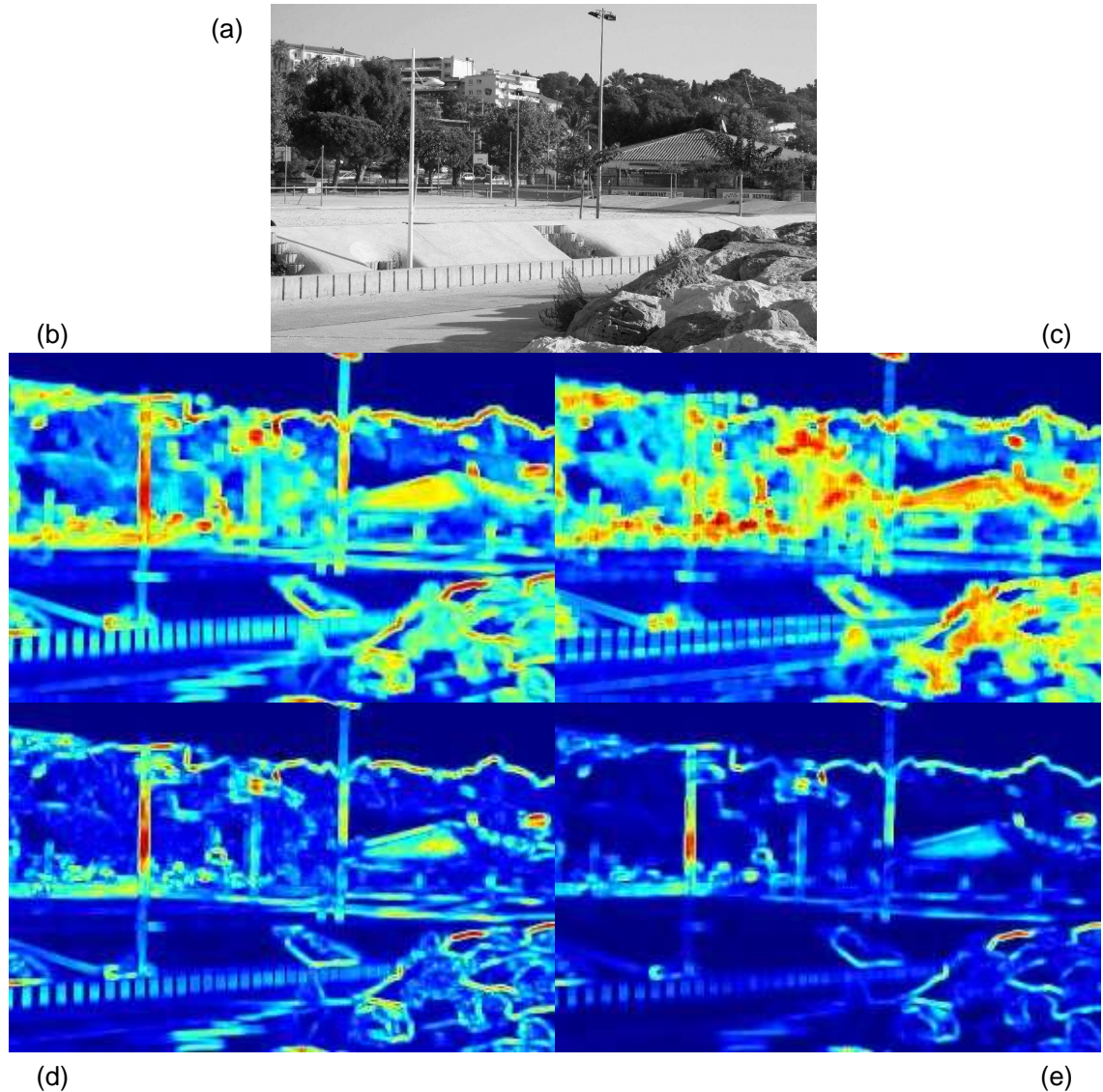


Figure 3-0 Image originale (a), énergie amplitude (b), amplitude moyenne (c), amplitude max (d), variance de l'amplitude (e) sur le spectre de Fourier.

Les opérateurs basés sur le module du spectre de Fourier présentent un bon pouvoir de discrimination de certaines textures pour une même scène. Ils produisent des réponses homogènes sur des parties de l'image telles que le toit du bâtiment sur la droite (Figure 3-0, image (a) à (e)), ou la végétation sur la moitié haute de l'image. Bien que ces opérateurs soient indépendants du niveau de gris moyen, ils dépendent néanmoins du contraste et de l'échelle de l'imagette traitée.

Dans certaines situations, la notion de texture est difficile à définir, notamment quand les objets à reconnaître sont caractérisés par leur aspect structurel/composite. Ce problème se rencontre fréquemment en présence d'objets artificiels, géométriques. Les méthodes pour labelliser une texture « bâtiments » dans une scène urbaine par exemple, peuvent être assez spécifiques et différentes de celles vues jusqu'à présent. Nous allons nous pencher sur ce type de problématique.

3.3.7 Détection de zones artificielles

C'est le cas quand nous voulons caractériser des bâtiments en vue rapprochée par exemple. Les objets rencontrés peuvent être relativement divers, contenir d'autres objets (panneaux publicitaires, vitrines). L'aspect composite de l'objet à reconnaître, plus que son aspect statistique local, devient alors prépondérant. Si, en plus de pouvoir reconnaître le modèle/concept « bâtiment », nous voulons pouvoir séparer les instances présentes dans la scène (bâtiment1, bâtiment2), la tâche peut devenir ardue. Une approche pour résoudre ce problème est de caractériser un bâtiment comme une zone comportant de forts gradients (aspect variationnel de l'intensité de lumière) et la présence de segments de droites (aspect géométrique local), regroupés de manière structurale en terme de parallélisme et perpendicularité (aspect géométrique composite). Ce type de méthode est utilisé dans [YLI02]. Les caractéristiques que nous venons de citer permettent d'établir, sur la scène observée, des clusters de zones correspondant à des bâtiments. La séparabilité entre les instances présentes est assurée par une information de teinte à proximité des segments de droite extraits (image couleur). La Figure 3-0 illustre une application de ce type d'approche. Nous reparlerons au chapitre 2 de l'importance de l'aspect composite d'une texture et présenterons une méthode pour extraire cette information de manière automatisée.

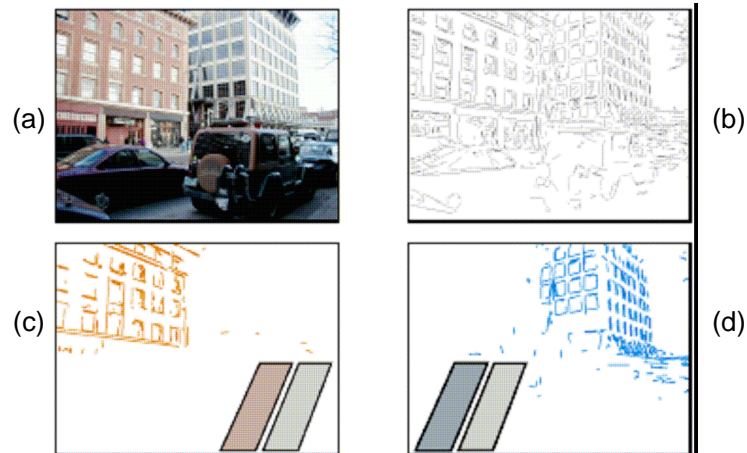


Figure 3-0 (issue de [YLI02]) Extraction de buildings par recherche de segments cohérents en couleur et parallélisme et perpendicularité (relativement à la perspective de la scène). De haut en bas et de gauche à droite : l'image d'origine (a), le résultat de l'extraction de contours (b), résultat de détection de l'immeuble de gauche (c) et de droite (d). La séparation des deux immeubles est effectuée à partir de l'information de teinte de l'image couleur.

Dans sa thèse, Yves Caron [CAR04] rappelle que la séparation entre les parties manufacturées et naturelles de l'image peut se faire via le calcul de la dimension fractale et l'adaptation de ce modèle à la zone image locale considérée. Une dimension fractale comprise en 2 et 2,5 avec une erreur importante par rapport au modèle fractal représente une zone artificielle. Le modèle utilisé est une droite dans un graphe log-log, se référer à la section 3.4.3 pour plus de détails.

Une fois qu'un certain nombre de textures ont pu être caractérisées, que la séparation entre le naturel et l'artificiel a été effectuée, il convient de savoir ce que va devenir l'information extraite. Une application répondant aux besoins actuels du domaine militaire et se trouvant être réalisable dans des contraintes de temps réel est la colorisation

d'image monochrome. C'est cette problématique qui sera abordée dans la prochaine section.

3.3.8 Colorisation d'image monochrome

L'étude de la texture a abouti ces dernières années dans une application moins contrainte que la reconnaissance complète d'un ensemble de textures, à savoir la colorisation d'images monochromes. De récents travaux ont été effectués par le Netherlands Organisation for Applied Scientific Research (TNO) en collaboration avec l'armée Danoise pour appliquer cette approche à leur équipement de vision de nuit IL (*lumière intensifiée*) des fantassins. Lors d'utilisations prolongées, ce type d'équipement pouvait occasionner une fatigue oculaire et psychologique (problèmes de sommeil, maux de tête) à l'utilisateur. Plus que l'aspect cosmétique, ce système a pour but de résoudre ces inconvénients en utilisant la majeure partie du spectre visuel humain (Figure 3-0). En effet, si nous considérons une lunette à vision de nuit au phosphore 43, l'information transmise à l'opérateur ne recouvre que 20% de son spectre visible (520 à 550nm pour le phosphore P43, contre 400 à 700nm pour le spectre humain). L'utilisateur doit donc traiter tout ce qu'il voit (bâtiments, obstacle, véhicules) avec une partie très restreinte de son spectre, comparée à une scène « habituelle » en plein jour. La dissociation entre objets devient alors difficile. Ce phénomène est illustré en Figure 3-0. Dans [TOE04], Alexander Toet propose une méthode de colorisation d'image monochrome en se basant sur une image couleur présentant une scène conceptuellement proche de l'image à coloriser. Après une étape intermédiaire de changement de repère de couleurs, l'auteur applique des opérateurs locaux sur l'image monochrome et sur l'image couleur transformée sur un seul canal. Les pixels de l'image de nuit sont colorisés artificiellement avec des coordonnées couleurs de pixels de l'image couleur qui ont renvoyé des réponses identiques, relativement aux opérateurs choisis. L'avantage de cette approche est que la texture n'est pas apprise. Seul un contexte est donné en entrée (une ou plusieurs images couleur, de jour). La méthode ne fait que mettre en correspondance des éléments (couleurs) présents (dans le cas de l'image de jour) ou considérés comme variables cachées (dans l'image à coloriser). Les liens entre les éléments de ces deux ensembles sont des attributs texture locaux.

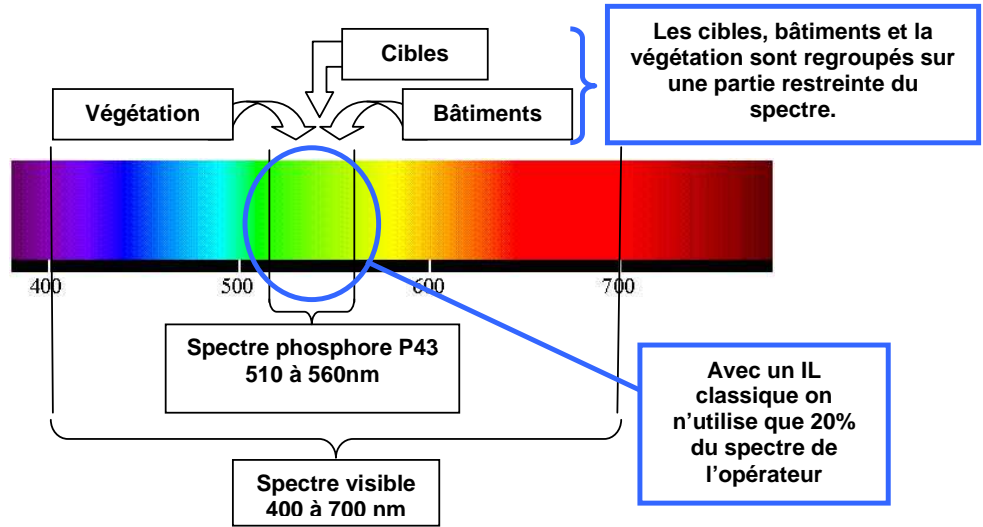


Figure 3-0 Comparaison du spectre d'une vision de nuit au phosphore avec le spectre visible humain.

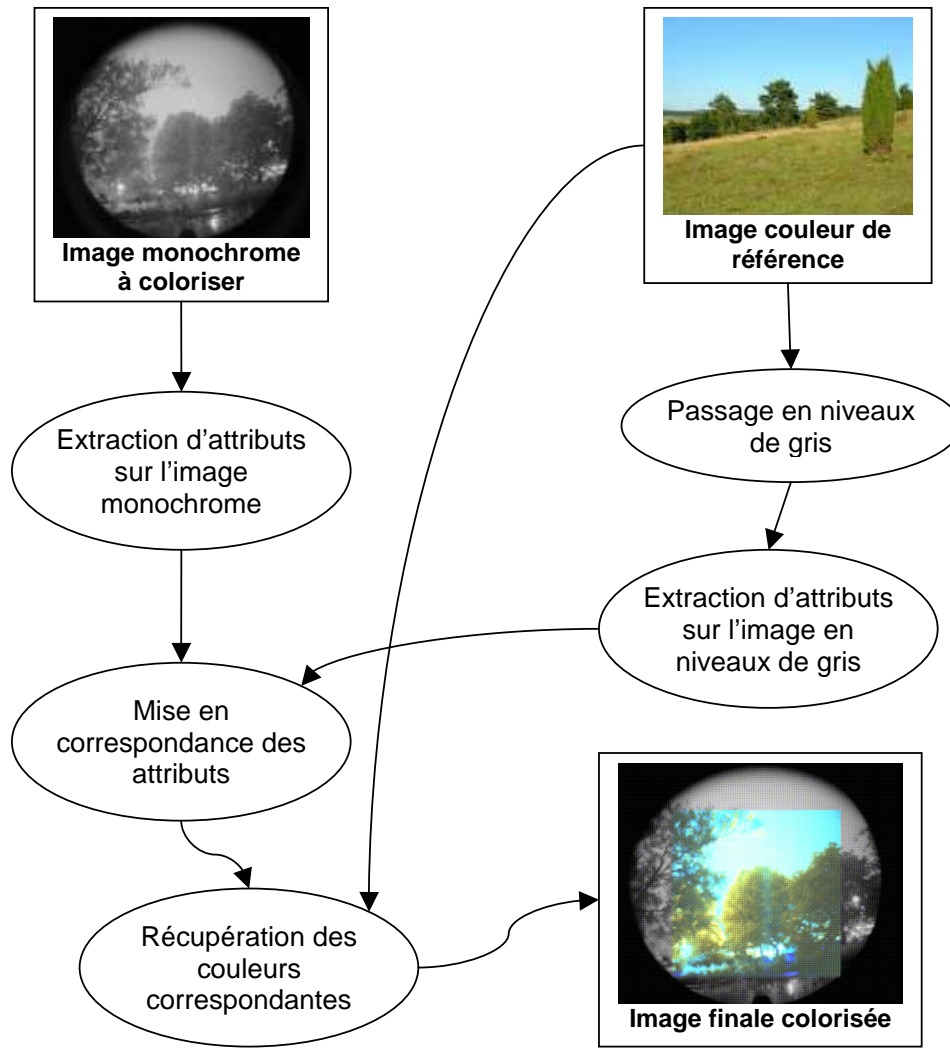


Figure 3-0 Méthode de colorisation d'image monochrome en se basant sur une image couleur.

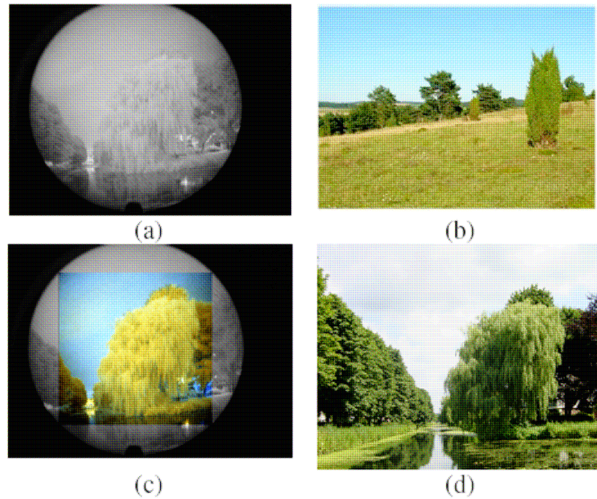


Figure 3-0 (issue de [TOE05]) : colorisation d'image monochrome en lumière intensifiée. Image de nuit monochrome (a), image couleur de référence (b), image de nuit colorisée (c), image la même scène vue de jour (d) à titre comparatif.

Le diagramme de la méthode est donné en Figure 3-0. La méthode prend en entrée le flux d'images monochromes à coloriser, une image d'une scène couleur, et une liste d'opérateurs. Les mesures des opérateurs sont extraites des images monochromes et de l'image couleur. Après une mise en correspondance des attributs texture extraits de ces deux sources, l'image monochrome est colorisée avec les couleurs des éléments de l'image couleur qui ont renvoyé localement des valeurs proches (dans la version monochrome de l'image couleur).

Une implémentation de la méthode a été effectuée. Nous avons pu remarquer que la méthode montrait ses limites en présence de formes géométriques complexes. Celles-ci sont très hétérogènes d'un point de vue modèles statistiques. Dans les images proposées dans l'article, soit il n'y a pas de zone artificielle, soit celles-ci sont vues de très près. Dans ce dernier cas, une zone artificielle est alors soit homogène, soit un contour (la plupart du temps une ligne droite). Par contre la colorisation en présence de végétation et ciel par exemple fonctionne relativement bien visuellement. Elle permet un gain en confort pour l'utilisateur tout en restant relativement économique en temps de calcul.

Pour combler les lacunes dues à la présence de zones artificielles, une solution pourrait être d'utiliser en aval une méthode de séparation naturel/artificiel, comme proposé dans [CAR04], ou la méthode simple développée en 5.3.4.

Nous avons vu dans les sections précédentes différentes méthodes pour caractériser la texture sur des images. Nous allons maintenant nous intéresser plus spécifiquement aux opérateurs fractals et quantifier leurs avantages et limites par rapport aux opérateurs vus précédemment.

3.4 Opérateurs fractals

Dans cette section nous présentons un ensemble d'opérateurs basés sur les fractales. Nous verrons dans un premier temps l'opérateur le plus utilisé dans ce domaine, à savoir

la dimension fractale. Nous constaterons que ces applications peuvent être très diverses en fonction des connaissances a priori dont nous disposons sur la scène observée. Nous nous intéresserons ensuite à la mesure de lacunarité. Cette mesure est complémentaire de la dimension fractale dans le sens où elle permet de distinguer certains objets possédant une même dimension. Nous nous pencherons également sur les multifractales. Elles permettent d'observer suivant l'échelle, l'évolution de la dimension fractale d'un objet non fractionnaire.

3.4.1 Les Fractales, historique et applications

Dans toute étude, il convient de situer rigoureusement les besoins et l'outil qui va y répondre. Dans un premier temps nous présenterons un rapide historique des fractales, et identifierons les besoins qui ont participé à la naissance des outils relatifs à ce domaine. Ensuite, nous examinerons ce que ces outils sont devenus jusqu'à nos jours et quelles sont les applications qui les emploient.

Historiquement le terme *fractal* a été créé en 1974 par Benoît Mandelbrot (1924, -). Il désigne des objets qui ont pour propriété une redondance dans les motifs qui les caractérisent. Nous voyons les fractales apparaître dans des domaines très variés. Elles sont communément connues pour leur aptitude à générer des formes artistiques et naturelles. On en trouve de nombreux exemples sur des couvertures d'ouvrages traitant du sujet [BAR88, MAN95], ou dans les bases d'images se référant aux fractales. Même si les fractales ont plus d'une trentaine d'années, le besoin d'un tel outil s'est fait sentir bien plus tôt dans le domaine de la cartographie et de la physique. Dans son ouvrage *Les Atomes* de 1913 [PER13], Jean Perrin souligne que ces domaines sont parfois mal représentés par la géométrie Euclidienne et le calcul différentiel. Vers 1920, Norbert Wiener reprend la suite des travaux de Perrin et s'en inspire pour construire le modèle probabiliste du mouvement brownien. Les résultats de ses travaux et de ses nombreuses publications (plus de 300) sont regroupés dans des ouvrages comme [WIE66]. En 1961, Lewis Fry Richardson établit empiriquement la relation qu'il y a entre la longueur de la côte d'un pays et le pas de mesure utilisé pour calculer cette longueur (publications regroupées dans [RIC93]). Le principe consiste à choisir une mesure étalon, par exemple un segment d'un kilomètre, et à observer la longueur de la côte en nombre de segments mis bout à bout. La Figure 3-0 illustre ce phénomène. La longueur de la côte augmente au fur et à mesure que l'échelle de mesure, autrement dit la longueur du segment étalon, diminue. Si on observe l'évolution de la longueur de la côte en fonction de la longueur du segment utilisé, pour différents pays, nous obtenons les graphes présentés en Figure 3-0. Richardson n'avait pas eu l'occasion de publier ses résultats avant sa mort. Mandelbrot les retrouve en 1967 et commence alors à introduire la notion de dimension fractale.

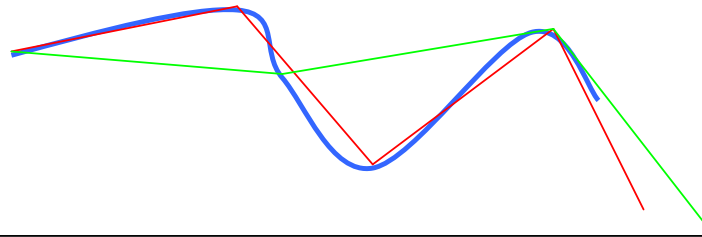


Figure 3-0 Mesure d'une courbe par un segment de longueur fixe. Ici deux longueurs sont présentées, rouge et verte, pour mesurer la longueur de la courbe bleue. Plus le segment est petit, plus la mesure de la longueur de la courbe est précise. L'évolution de la longueur estimée de la courbe par rapport à la longueur du segment informe sur sa dimension fractale, définie plus loin.

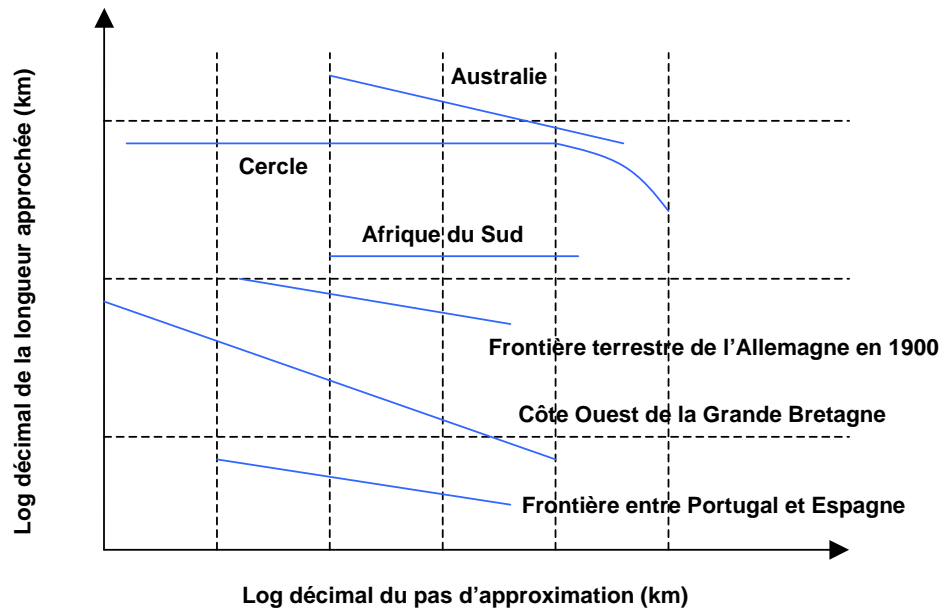


Figure 3-0 Evolution de la longueur de frontières territoriales en fonction de la mesure utilisée (graphe issu de [MAN95]).

Par la suite, l'application des fractales a commencé à se diversifier. Les fractales ont été utilisées dans le domaine artistique [BAR88]. Le principe de base de la génération d'objets fractionnaires est le choix d'une forme de base et d'un ensemble de transformations. L'application itérative de ces transformations produit une forme qui a pour propriété d'être similaire à elle-même à plusieurs échelles d'observation. De plus amples détails sur cet aspect seront donnés dans la suite du document. En traitement d'image, elles sont considérées comme pertinentes pour la reconnaissance de textures naturelles [MAN83, PEN84, KEN92]. En 1990, Jacquin propose d'utiliser les fractales pour en faire une méthode de compression d'image avec perte [JAC90] et [JAC92]. Par la suite, les fractales permettent des mesures sur des phénomènes naturels comme les modèles cristallins [NAR99], les coraux, les nuages [VEN04], un flocon de neige, la forme

des pays, etc. Dans le domaine industriel les fractales ont trouvé leur place dans le monde médical et les applications militaires. Dans [KEN92, NAK93, BUT94], elles sont utilisées pour la détection de pathologies liées au rythme cardiaque. Dans [GOL02], l'auteur souligne que bien que nous considérions communément le rythme cardiaque sur de petites périodes, classiquement une à deux minutes, celui d'un cœur sain, même pour un individu au repos, est loin d'être régulier et parfaitement périodique. Les mesures de loi puissance et opérateurs fractionnaires permettent d'avoir une nouvelle approche sur ces phénomènes biologiques et pourraient aider à prévenir la surveillance des processus de vieillissement et de dégradation du système cardiovasculaire. Dans la Figure 3-0, le rythme cardiaque de 4 patients est mesuré au repos sur une période d'une demi heure. Un seul parmi les 4 patients est sain.

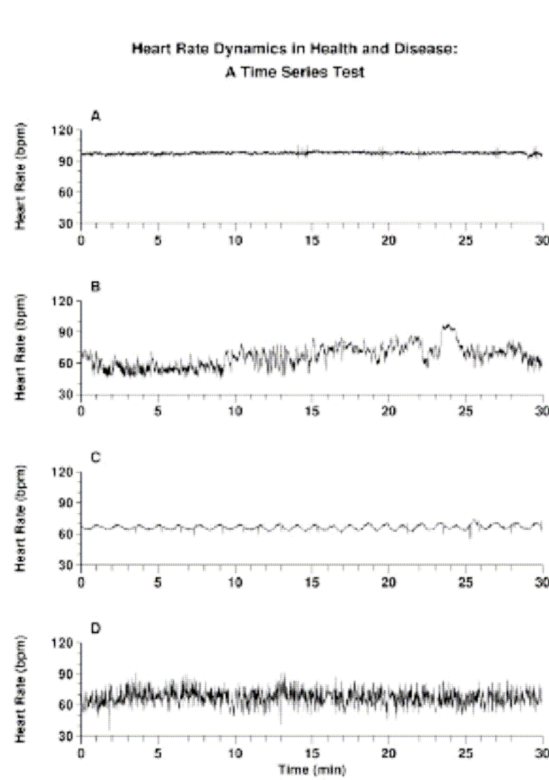


Figure 3-0 Rythme cardiaque de 4 patients au repos, sur 30 minutes. Le patient sain est le patient B. Son rythme suit un modèle de mouvement brownien fractionnaire.

Les patients A et C ont des rythmes hyper stables (forme sinusoïdale), leur rythme ne s'adapte pas aux différents phénomènes physiologiques qui interviennent au cours de la durée de la mesure. D souffre d'arythmie, son rythme cardiaque semble complètement aléatoire et non corrélé par rapport aux mesures passées. B est le patient sain, son rythme se modifie et ressemble à un mouvement brownien fractionnaire. Dans ce type de mouvement l'état futur peut varier fortement mais sa valeur dépend toujours de la valeur actuelle (additif par rapport à l'état $n - 1$).

Dans [ZHE99] et [SAR99] les mesures fractales sont utilisées pour la détection du cancer du sein. Elles servent dans le premier cas à mettre de côté les zones qui ne

peuvent pas contenir de cancer, et dans le second à détecter les zones de micro calcifications, significatives par leur absence de similarité par rapport au reste de l'image. On les retrouve également dans le traitement de l'épilepsie [EST99], le séquençage de l'ADN [BER96], la reconstruction 3D [SUM99], la neurologie [HEN01], et la prévention de maladies osseuses [SHA99], [PAR99]. En biométrie, les fractales ont été utilisées pour identifier des individus à partir de leur façon de marcher. [SAG06] utilise une représentation fréquentielle de la marche et les autosimilarités pour faire du suivi d'une personne en présence d'occultations, dues aux croisements avec d'autres individus. Ce principe de recherche d'autosimilarité dans la démarche d'une personne est également repris dans [ABD03]. Les fractales ont été utilisées par [WAN00] dans le domaine de la reconnaissance vocale. Dans son article, un processus Gaussien en $1/f$ est utilisé (un modèle mathématique pour les processus autosimilaires issus des fractales) pour modéliser le bruit et la voix, qui sont ensuite séparés par un classifieur Bayésien.

Dans le domaine militaire, les fractales sont utilisées dans le débruitage d'image SAR (*Synthetic Aperture Radar*) comme illustré dans la Figure 3-0 (données issues de la bibliothèque *FracLab*, <http://complex.futurs.inria.fr/FracLab>). Ce débruitage s'opère en effectuant une estimation et un contrôle de la régularité de la texture via les coefficients d'ondelettes [LEV97], [LEG03]. Dans la Figure 3-0, l'image de gauche représente la donnée image brute issue du radar, à droite l'image radar débruitée par méthode multifractale. Alors que l'image brute semble difficile à exploiter de manière immédiate, un croisement de fleuves formant un « V » inversé apparaît sur l'image débruitée.

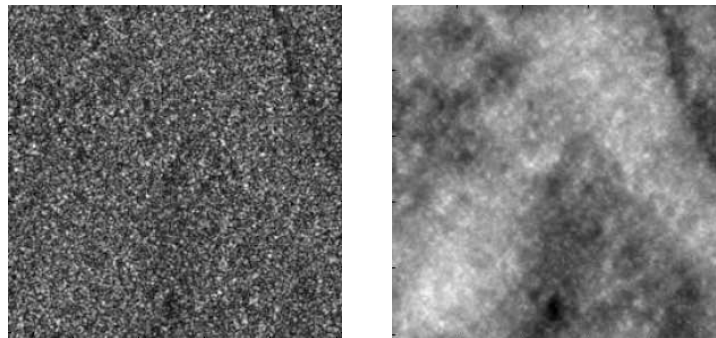


Figure 3-0 à gauche l'image SAR d'entrée, à droite l'image débruitée issue de l'étude multifractale (source : FracLab).

Les fractales ont été utilisées dans la détection d'objets en mouvement dans des vidéos comme proposé dans [ALL06]. La Figure 3-0 est issue de l'article en question et illustre cette méthode. L'image de gauche (a) montre le résultat de la détection de mouvement avec une fenêtre temporelle centrée de 9 images pour le calcul de l'erreur fractale (définie plus loin), l'image du centre (b) montre la même méthode avec une fenêtre non centrée de 4 images, enfin l'image de droite (c) illustre le résultat par différence temporelle non centrée entre les images.

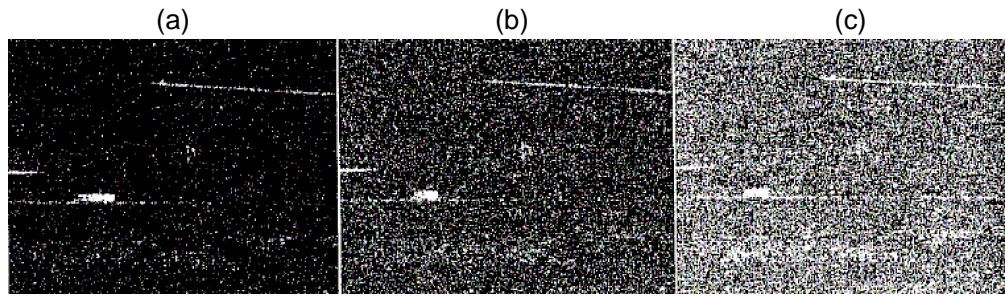


Figure 3-0 (issue de [ALL06]) Résultat de détection de cible par mesure de l'erreur fractale. Erreur fractale en utilisant une fenêtre temporelle centrée de longueur 9 images (a), erreur fractale utilisant une fenêtre non centrée de longueur 4 images (b), différence inter images, non centrée sur 4 images (c).

Nous venons de voir un historique des fractales ainsi que quelques exemples d'applications dans des domaines divers. Nous allons maintenant nous y intéresser d'un point de vue plus technique en commençant l'aspect le plus connu du monde des fractales, à savoir la partie synthèse.

3.4.2 Synthèse et suréchantillonnage

Les fractales sont essentiellement connues du grand public par leur faculté à générer des formes étonnantes. Même si nous n'allons pas directement utiliser la partie synthèse des fractales dans notre étude, il était difficile de passer à côté de ce point. Les phénomènes de synthèse d'objets fractals sont la plupart du temps utilisés en une ou deux dimensions mais leur formalisme permet de les utiliser dans un espace à n -dimensions. Il s'agit d'un processus itératif la plupart du temps. A chaque itération, l'évolution peut se faire de manière déterministe ou en introduisant une partie stochastique. Un flocon de neige est un exemple naturel de fractale déterministe : il est autosimilaire et deux instances de flocons ont une forme pratiquement identique. Un phénomène naturel peut être fractal et comporter une partie aléatoire dans sa construction. La Figure 3-0 montre un « paysage fractal » dépourvu d'aléatoire, vu de haut et de $3/4$. Cette forme est basée sur le triangle de Sierpinsky. Plus bas, la Figure 3-0 illustre une figure basée sur la même méthode de construction mais en introduisant de l'aléatoire. Cette seconde représentation semble plus naturelle que la première et permet de générer et stocker des cartes de paysage de grande taille à un faible coût en terme d'occupation mémoire. C'est une des raisons pour lesquelles les fractales sont communément utilisées dans la modélisation et synthèse de terrains (dans des simulateurs de vol par exemple). L'aspect fractionnaire permet de plus de choisir le niveau de détail auquel on se situe. Pour une vue du terrain à haute altitude, on ne fera que peu d'itérations mais sur une grande surface (maillage grossier), pour une représentation plus proche, on effectuera plus d'itérations mais sur une zone plus ciblée.

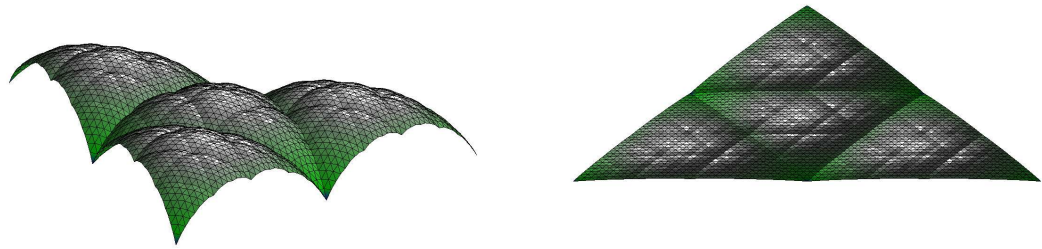


Figure 3-0 Paysage généré de manière déterministe, vue de côté et vue de haut.

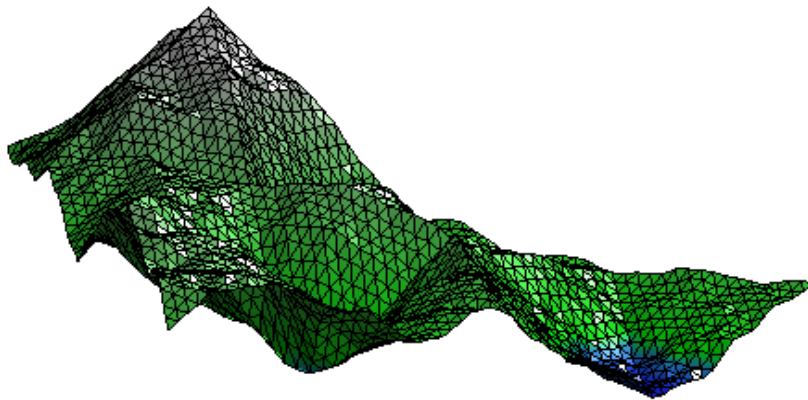


Figure 3-0 Exemple de paysage fractal de mêmes paramètres mais ajout du caractère stochastique dans la génération.

Quand nous voyons le terrain d'aspect naturel généré ci-dessus, nous pourrions être tenté de poser la question : « dans le cas où nous observons un paysage réel, de montagne par exemple, et que nous choisissons de le modéliser par un modèle fractal, supposons qu'une partie seulement de ce paysage soit observable (pour cause de brouillard ou par simple occultation par un relief montagneux), est-ce que les mesures fractales que nous pouvons faire sur la partie observable nous permettraient de générer une bonne approximation de la partie non observable du terrain ? ». La réponse est *non*. En effet, il existe une infinité de reliefs possibles ayant en commun la partie observable et possédant les mêmes caractéristiques fractionnaires (dimension fractale et autres opérateurs que nous verrons par la suite). Une application moins ambitieuse mais réalisable et plus axée sur le cosmétique serait d'augmenter artificiellement le niveau de détail d'une carte de hauteur. Par exemple, supposons qu'on connaisse un relief montagneux avec un pas d'échantillonnage de 20m, nous serions en mesure de générer *une* carte (parmi l'infinité de cartes possibles) représentant le même terrain avec un pas de 5m. Ici nous ne créons pas d'information, nous ne faisons que générer des données cohérentes (d'un point de vue mesure fractale) avec ce qui est observable.

Cet effet de conservation de l'irrégularité d'une carte de hauteur peut s'appliquer à une image 2D en niveaux de gris ou en couleur (on répète alors le même procédé sur les trois

canaux rouge, vert, bleu). Un exemple est donné à la Figure 3-0, appliqué dans le domaine de la photo numérique. Nous disposons d'une photo d'un œil prise avec une taille de 106x40 pixels dont nous souhaitons augmenter la résolution de façon artificielle (ici 450% de la taille). Cette figure illustre le résultat de quelques interpolations : par plus proches voisins, interpolation bilinéaire, bicubique et fractale. Les plus proches voisins aboutissent à un aspect pixellisé de l'image. L'interpolation bilinéaire donne une image plus nette mais introduit un phénomène de flou, notamment dans les zones où un des canaux fait « une bosse », par exemple des veinules dans le blanc de l'œil ou les cils (sur la moitié droite de l'image). Ceux-ci gardent leur aspect légèrement pixellisé et flou. L'interpolation bicubique gère mieux ce type de phénomène mais les zones où les contrastes sont plus fins, telles que l'iris, le blanc de l'œil et le coin entre l'œil et le nez, manquent encore de netteté. L'interpolation fractale donne les meilleurs résultats dans ces zones, hormis les cils où nous notons la présence d'artéfacts.

Les fractales ne se limitent pas à rendre possible la génération d'images synthétiques ou de phénomènes naturels. Leur étude a permis d'élaborer également un ensemble d'outils mathématiques qui permettent de caractériser certains signaux. Comme l'indique [COC95], « ...il n'est pas nécessaire que les objets soient fractals pour être caractérisés par des attributs fractals. ». Des objets non fractionnaires renvoient des résultats représentatifs quand ils sont observés par des outils fractals (mesures statistiques, redondance de motifs géométriques). Le champ des domaines d'application est vaste. Nous rassemblons une liste non exhaustive d'applications possibles :

- Dans le domaine médical, la mesure fractionnaire d'un rythme cardiaque peut donner des informations sur l'éventuelle présence d'une pathologie. [KEN92, NAK93, BUT94]
- L'amélioration d'image, le suréchantillonnage [SEC03, PRI98, WIT95]
- La compression d'image. Ce thème sera largement plus détaillé plus loin dans un chapitre de ce document.
- L'estimation de scène, indice sur la géométrie d'une scène
- La synthèse de terrains naturels.

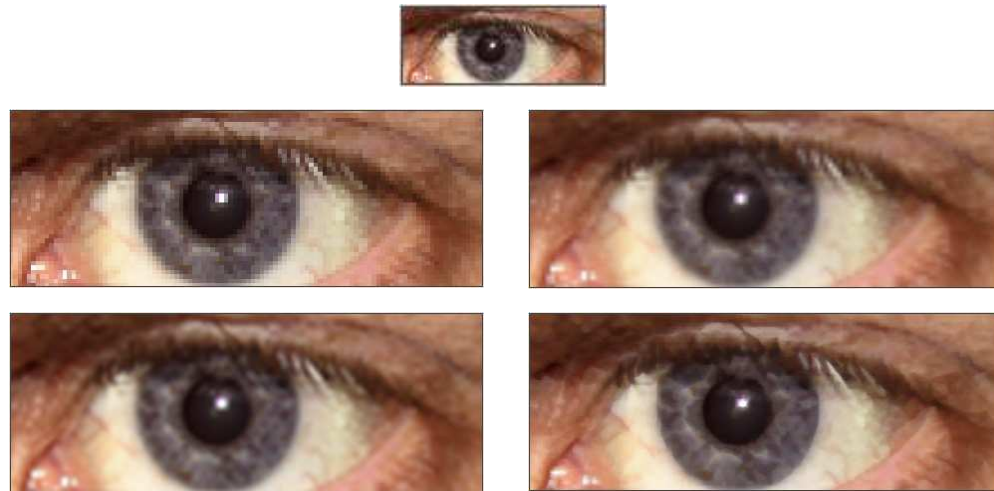


Figure 3-0 De haut en bas, l'image d'origine à agrandir, interpolation par plus proches voisins, bilinéaire, bicubique, fractale.

3.4.3 Dimension fractale

La dimension fractale est une mesure qui repose sur le curieux postulat qu'un objet peut être de dimension non entière. Elle est née du besoin de caractériser des objets dont la mesure variait en fonction de l'échelle d'observation. Prenons l'exemple bien connu de la côte d'un pays. Nous cherchons à déterminer sa longueur. Une méthode simple est de prendre un bâton de longueur ε , 1km pour commencer, et de compter combien de fois nous pouvons le placer (« bout à bout ») le long de la côte. Nous faisons ensuite varier ε (500m, 250m, 100m, 10m, 1m, ...), et nous obtenons à chaque fois une nouvelle mesure de la longueur de la côte. Nous constatons que celle-ci augmente au fur et à mesure que ε diminue. En effet, de « petits » ε permettent de passer dans des recoins plus fins que des ε de taille plus importante. A chaque ε , nous associons donc une longueur de côte $L(\varepsilon)$. Le même type de résultat pourrait être constaté en remplaçant les bâtons par une série de disques non recouvrants de rayon ε ou des boîtes carrées de côté ε (Figure 3-0). Dans chaque cas, nous comptons le nombre d'éléments atomiques utilisés (un segment de droite, un disque, une boîte) pour recouvrir l'objet considéré. Plusieurs variantes existent [PEL484].

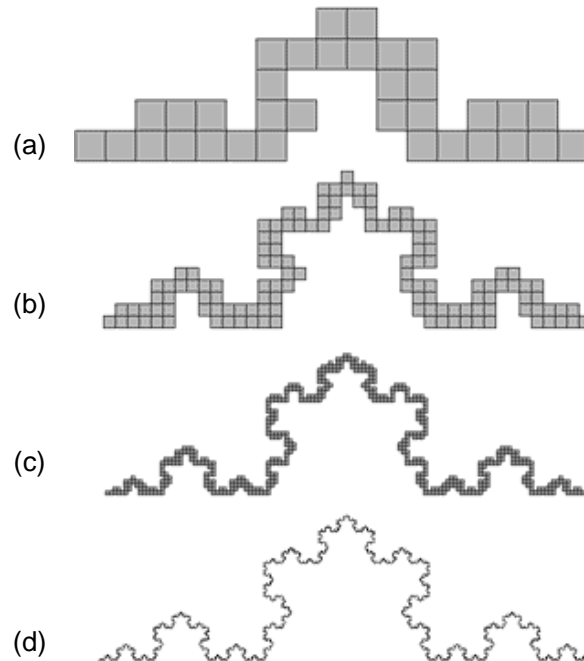


Figure 3-0 Estimation de la dimension fractale du flocon de von Koch par la méthode des boîtes. Images (a) à (d) : le nombre de boîtes utilisées pour recouvrir le flocon augmente au fur et à mesure que la taille des boîtes diminue.

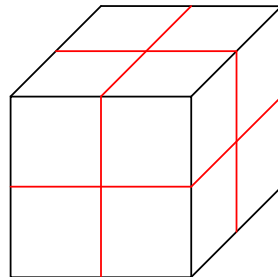
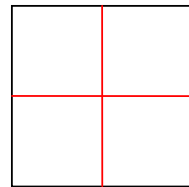
Mandelbrot a démontré que pour beaucoup de lignes côtières nous avons la relation

$$L(\varepsilon) = F \cdot \varepsilon^{1-D}$$

où F et D sont des constantes caractéristiques de la côte. D est appelée la dimension fractale de la côte. Pour une courbe fractale, si on regarde l'évolution de la longueur de la courbe $L(\varepsilon)$ en fonction de ε dans un graphe log-log, on observe qu'elle

peut être par approximation une droite de pente $1 - D$. Pour une courbe qui est une droite on a $D = 1$.

La dimension fractale nous informe sur la relation entre le nombre de sous parties similaires et leurs tailles successives (de plus en plus petites au fil des échelles). Cela permet d'obtenir une notion de dimension, éventuellement non entière, et pourtant cohérente avec l'approche que nous en avons communément. En effet, plus la dimension fractale d'une forme est proche de 2, plus cette forme va tendre vers une surface, plus cette valeur sera proche de 3, plus elle se rapprochera d'un volume. Si nous appliquons cette mesure au flocon de von Koch, nous obtenons une valeur d'approximativement 1,26, ce qui correspond à son processus de formation (un segment de taille L , divisé en 4 parties de taille $L/3$), d'où $D = \frac{\log(4)}{\log(3)} = 1,26$.



Cas 1D : Nous considérons une droite qu'on découpe en N parts égales, le ratio entre la droite d'origine et ses sous parties est $r = 1/N$ et $N.r^{(D=1)} = 1$.

Cas 2D : Nous considérons un carré qu'on divise en N carrés de même taille, $r = 1/N^{1/2}$ et $N.r^{(D=2)} = 1$

Cas 3D : on considère un cube qu'on divise en N cubes de même taille, $r = 1/N^{1/3}$ et $N.r^{(D=3)} = 1$

Cas N D : pour un objet à N dimensions on a $N.r^D = 1$ ce qui nous amène à $D = \frac{\log(N)}{\log(1/r)}$.

Figure 3-0 Définition intuitive de la dimension fractale pour des ensembles autosimilaires et lien avec la notion usuelle de dimension.

La dimension fractale peut être généralisée à autre chose que la mesure de distance dans le domaine spatial. Nous pouvons également nous en servir pour caractériser le comportement d'une mesure en fonction de l'échelle d'observation. Dans le cas où l'objet observé est fractal (au sens de la mesure choisie), l'observation de cette évolution est représentée par une droite de pente D dans un graphe log-log, avec

$$D = \lim_{\varepsilon \rightarrow 0} \frac{\log(\mu(X_\varepsilon))}{\log(\varepsilon)}$$

où $\mu(X)$ représente une mesure sur les données X , et ε représente l'échelle d'observation. X_ε représente les données X observées à l'échelle ε . Une approximation de D est classiquement faite en calculant la pente de la droite $\mu(X_\varepsilon)$ en fonction de ε dans un graphe log-log par une régression linéaire. Dans son article de 1984 [PEN84], Alex P. Pentland présente différentes approches de l'utilisation de la dimension fractale. Il l'emploie dans le cadre de la détection de contours, la segmentation automatique et l'estimation de la forme, moyennant certaines hypothèses sur la texture considérée. La dimension fractale est alors appliquée pour mesurer l'évolution de la distance entre des niveaux de gris, distance en amplitude pour des couples de pixels, en fonction de la distance spatiale séparant ces pixels. Nous avons alors :

$\mu(X_\varepsilon)$ est la moyenne des valeurs absolues de la différence des niveaux de gris entre deux pixels p_1 et p_2 de X_ε (classiquement $X_\varepsilon \in [0..255]$), ε est la distance spatiale euclidienne entre p_1 et p_2 . Pentland montre que la dimension fractale est invariante aux changements d'échelle et aux transformations linéaires sur les données observées (ici, des échantillons locaux de textures).

La Figure 3-0 représente une vue aérienne de la baie de San Francisco (a) et la carte de valeur représentant sa dimension fractale locale (b), calculée sur une fenêtre glissante de 16x16 pixels. A partir de cette carte de dimension fractale locale, l'auteur propose une méthode de segmentation automatique en appliquant la méthode de seuillage de Otsu [OTS79] à l'histogramme des valeurs de la dimension locale de l'image (cf. Figure 3-0). La séparation en deux sous régions peut être itérée plusieurs fois de manière à obtenir une décomposition hiérarchique de l'image.

(a)



(b)

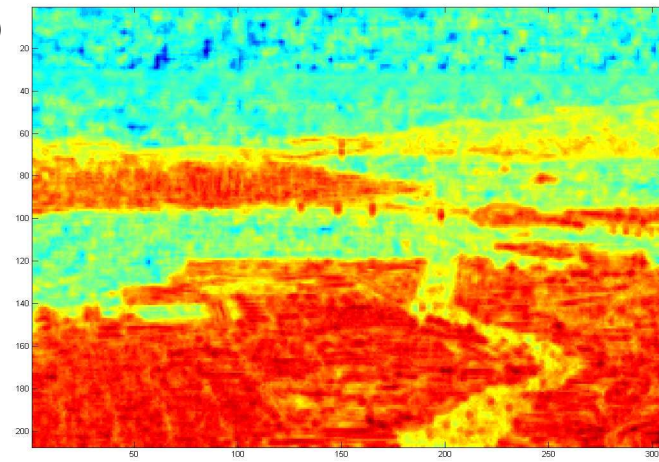


Figure 3-0 Calcul de la dimension fractale sur une vue aérienne de la baie de San Francisco.

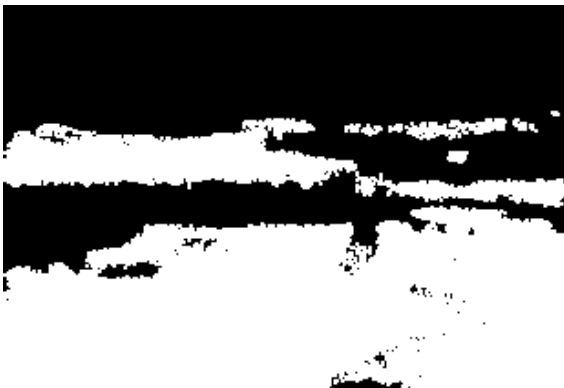
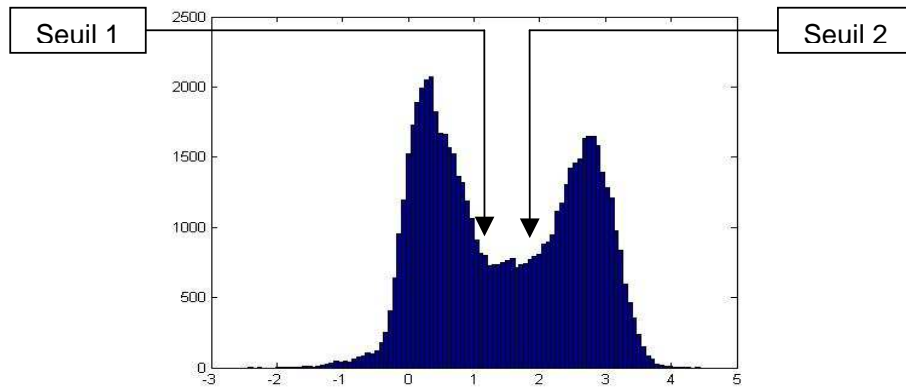
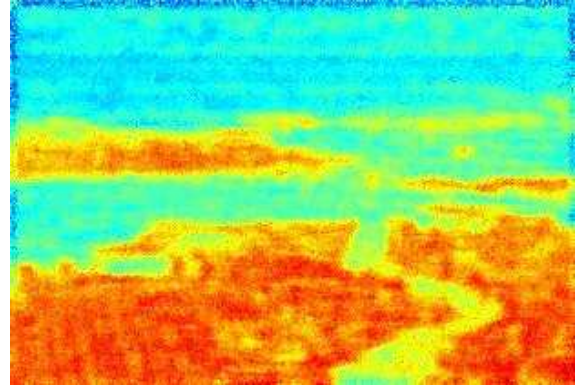


Figure 3-0 En haut à gauche et à droite, image d'origine de la baie de San Francisco et réponse de la dimension fractale. Au centre, histogramme des valeurs de la dimension fractale. En bas à gauche et droite, résultats du seuillage à 1,2 et 1,6 (cf. histogramme). Avec un seuil de 1,2 nous perdons le fleuve nous extrayons la côte proche de l'horizon, avec un seuil de 1,6 le fleuve ressort mieux mais nous perdons une partie de la côte à l'horizon.

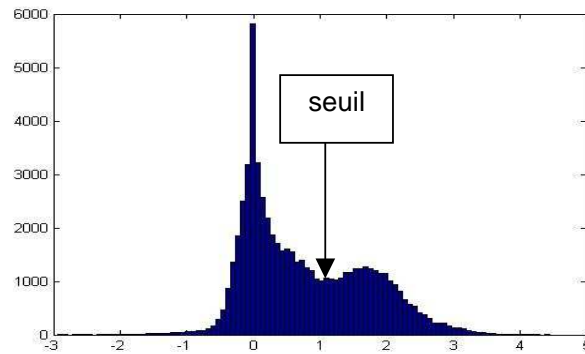
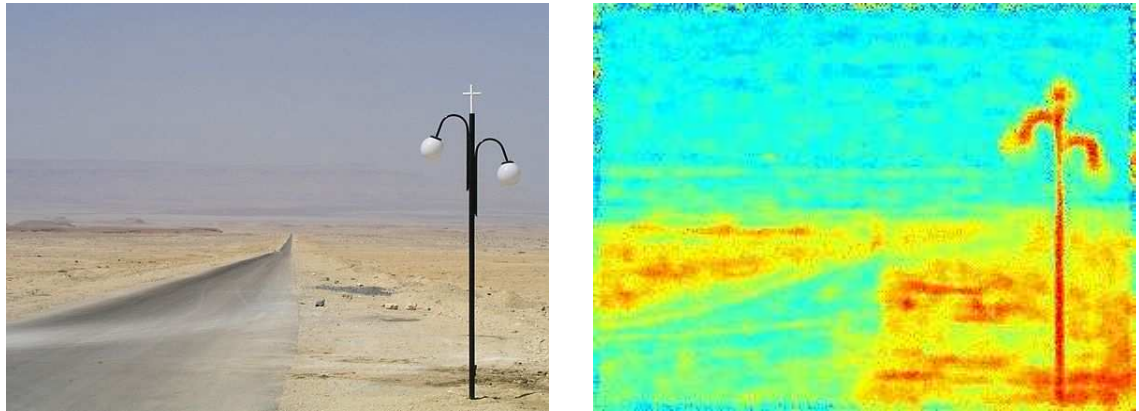


Figure 3-0 En haut à gauche, l'image originale de route au milieu du désert, en haut à droite, la réponse de la dimension fractale. Au centre, l'histogramme des valeurs de la dimension fractale et la valeur de seuil (1). En bas à gauche et à droite respectivement la réponse de la dimension fractale seuillée, et seuillée suivie d'une ouverture par masque 2x2 et suppression des petites régions.

Pentland propose également d'utiliser la dimension fractale pour faire une estimation de la forme 3D d'une scène ou une estimation de l'angle entre l'axe optique du capteur et la normale au plan moyen d'une texture. L'hypothèse principale pour que cette technique fonctionne est que la texture traitée soit isotrope (on obtient une erreur importante dans le cas où cette hypothèse n'est pas valide). Une application possible est l'estimation d'angle

de prise de vue d'une photo aérienne : le tissu urbain, s'il est vu d'une hauteur suffisamment importante peut être considéré comme isotrope. Plus on tend vers l'horizon, plus la texture du tissu urbain est « adoucie ». La pente de l'évolution de la dimension fractale du premier plan jusqu'à l'horizon permet d'estimer l'angle de prise de vue de l'image.

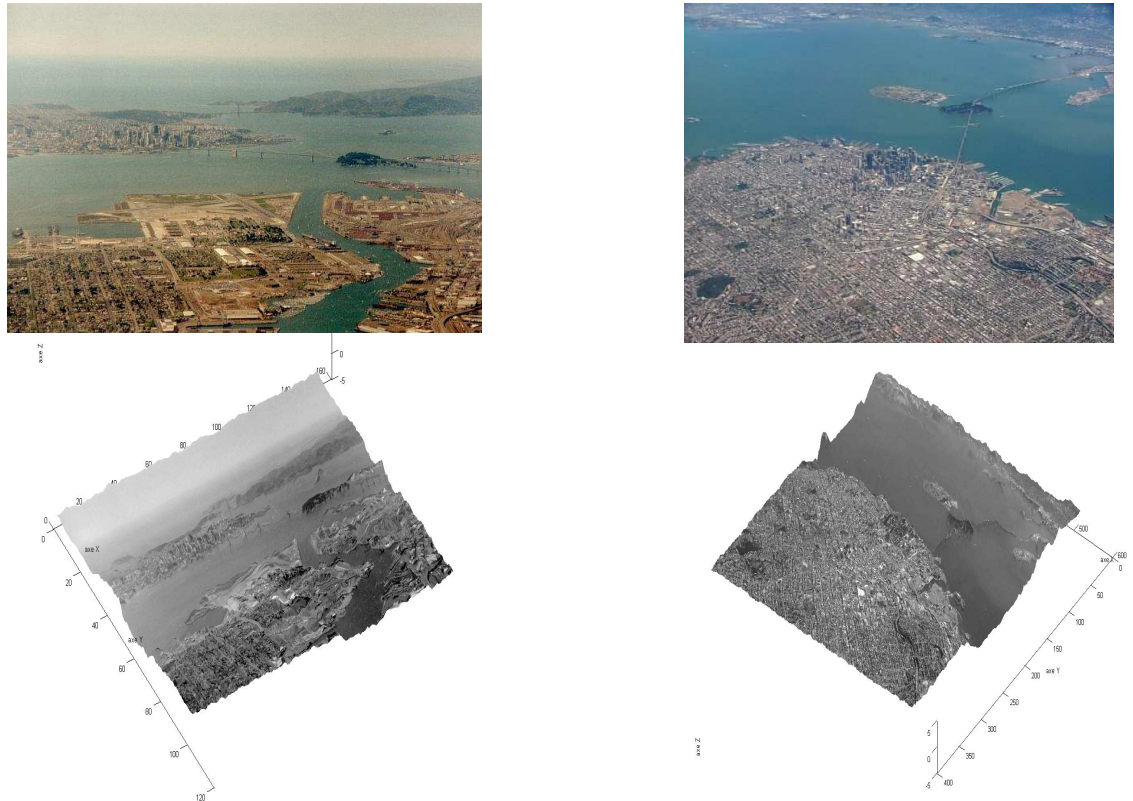


Figure 3-0 Reconstruction 3D de vue aérienne à partir de la dimension fractale.

Quand le sol est plus proche, la scène devient plus riche et l'hypothèse d'isotropie sur l'ensemble de l'image n'est plus applicable. On ne peut donc pas en déduire une orientation globale de l'image en tant que texture. Par contre, si on considère une partie locale de l'image, alors la propriété reste vérifiée. La Figure 3-0 illustre ce type de phénomène. Une portion de végétation est sélectionnée. Celle-ci est située sur un relief et s'éloigne vers l'horizon. La dimension fractale locale a été définie sur tout l'échantillon. On note une diminution de la dimension fractale locale sur cette portion de texture lorsqu'on considère des valeurs correspondant à des données en premier plan et des valeurs correspondant à des données proches de l'horizon et situées à une grande distance par rapport à l'emplacement de prise de vue. La dimension fractale offre donc des résultats intéressants quand elle est appliquée dans les conditions prescrites par l'auteur (notamment, vue aérienne : Figure 3-0, ou peu de textures différentes sur une même image : Figure 3-0).

Sur une image basse altitude avec des textures diverses, la dimension fractale locale prend des valeurs hétérogènes. Cependant localement, à l'intérieur d'une région représentant une même texture, l'information fournie par la dimension fractale reste

pertinente. La Figure 3-0 illustre le comportement de la dimension fractale sur une zone forestière. Plus la texture s'éloigne du plan capteur, plus son irrégularité diminue, et avec elle sa dimension fractale locale. Cependant l'information extraite d'une telle image par la dimension fractale n'est pas exploitable globalement sur toute l'image, mais uniquement sur des zones isolées et déjà segmentées. La dimension fractale locale sur les falaises ou le village, au centre de la photo de la Figure 3-0, prend des valeurs élevées, par rapport aux valeurs prises sur une zone forestière. Cela ne signifie pas que le village, ou la falaise, sont proches du plan capteur, mais que l'irrégularité locale des niveaux de gris est particulièrement importante dans ces zones de l'image. Par contre, si on calcule la dimension fractale locale pour une zone forestière d'une part, et pour une zone urbaine d'autre part, bien que les valeurs moyennes soient différentes, la même « pente de prise de vue », illustrée en Figure 3-0, peut se retrouver pour ces deux zones.

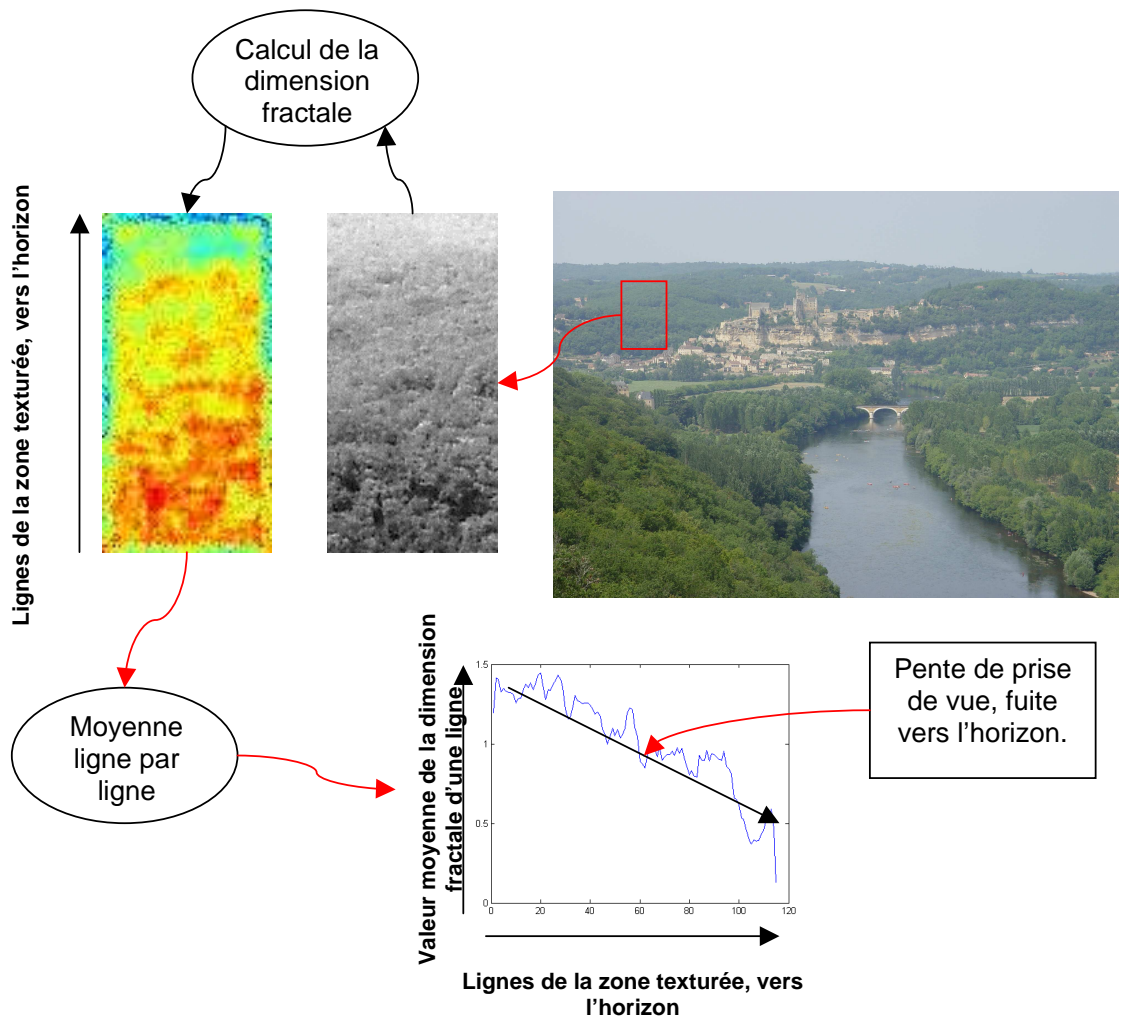


Figure 3-0 Exemple d'orientation locale de texture dans une scène complexe.

Remarque

La dimension fractale est une mesure qui peut se révéler coûteuse en temps de calcul à cause du nombre de couples de pixels à traiter par imagerie. Nous avons testé une amélioration simple de ce calcul en ne sélectionnant aléatoirement qu'une partie des

couples de pixels pour une imagerie donnée. La Figure 3-0 illustre la précision dans l'estimation de la dimension fractale, en la calculant selon un pourcentage plus ou moins important de couples de pixels. Dans cet exemple l'estimation de la dimension fractale vaut 3,1.

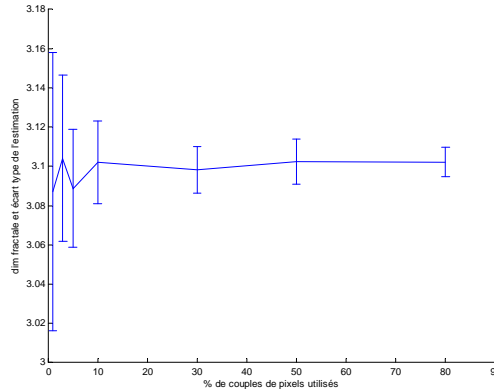


Figure 3-0 Exemple d'estimation, et écart type de l'estimation, de la dimension fractale par méthode stochastique pour une imagerie pour 1 à 80% des couples de pixels possibles.

Le calcul de la dimension fractale locale par méthode stochastique permet d'avoir une bonne estimation de la valeur obtenue par la méthode exhaustive (100% des couples) en diminuant de manière significative le nombre de couples de pixels pris en compte. Cette flexibilité permet d'adapter cet opérateur à la puissance de calcul disponible, c'est-à-dire au matériel.

3.4.4 Dimension fractale directionnelle par ondelettes

L'aspect orientation peut être pris en compte dans les mesures fractionnaires. Une approche parfois utilisée est de faire des coupes de la texture selon différentes orientations (on obtient alors plusieurs signaux 1D) et de calculer la dimension fractale 1D sur chacun des segments. L'histogramme des réponses fractionnaires des coupes constitue une signature de la texture.

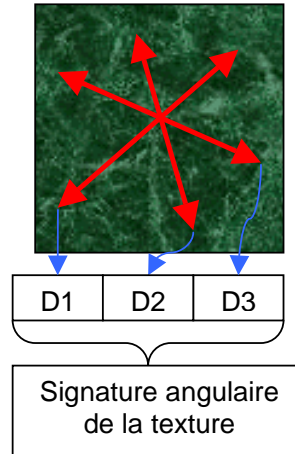


Figure 3-0 Signature d'une texture par calcul de la dimension fractale 1D sur des coupes suivant différentes orientations.

L'avantage de cette approche est qu'on extrait de l'information sur l'orientation de la texture. L'inconvénient est que dans le cas d'une texture bruitée, cette méthode peut s'avérer sensible au positionnement des coupes et au bruit de l'image, vue la faible quantité d'information portée par chaque coupe. Une solution plus robuste a été proposée dans [ESP98]. L'auteur se sert de la transformée en ondelettes pour calculer la dimension fractale d'une image selon les orientations de la décomposition (classiquement horizontale, verticale et diagonale). Une texture est alors caractérisée par un triplet d'attributs fractionnaires, un attribut par orientation. Ces attributs sont utilisés pour indexer les textures (base Brodatz) mais également pour la détection de cibles. Pour une orientation $\theta \in \{horizontal, vertical, diagonal\}$ donnée et un niveau d'ondelettes $\nu \in \{0, \dots, \log_2 m\}$, l'attribut local au pixel p de voisinage $N(p)$ dans la décomposition en ondelettes W est défini comme suit.

$$D_{\theta, \nu}(p) = \frac{\log^+ \left(\sum_{u \in W_{\theta, \nu}[N(p)]} |u| \sqrt{2^{-\nu}} \right)}{\log^+(2^\nu)}$$

où $W_{\theta, \nu}[N(p)]$ représente le voisinage du pixel p dans la transformation en ondelettes de niveau ν et d'orientation θ .

Une fenêtre 16x16 ou 32x32 est classiquement utilisée selon la taille des zones texturées à séparer sur une image composite.

La Figure 3-0 présente des segmentations issues de la dimension fractale par ondelettes (donc sur 3 attributs) et à titre comparatif une segmentation faite par filtres de Gabor avec 4 orientations. Les attributs fractionnaires permettent une meilleure séparation des limites entre textures et une plus grande séparabilité interclasses. En effet, certaines segmentations par filtres de Gabor tendent à fusionner des textures trop proches.

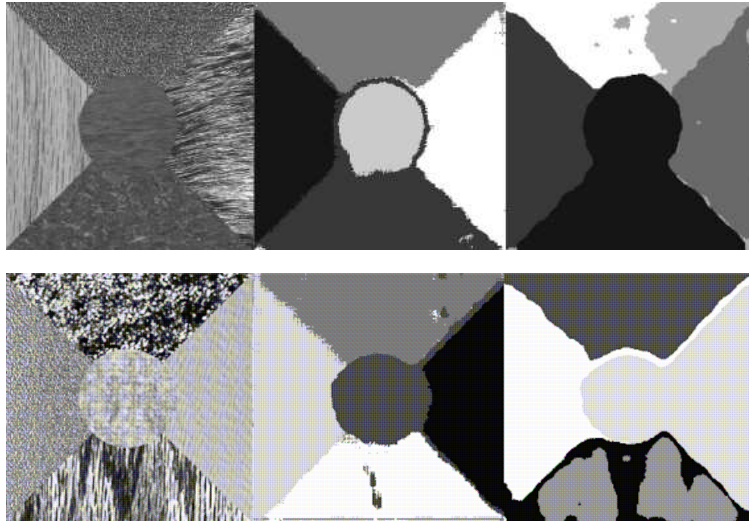


Figure 3-0 Exemple de segmentations obtenues à partir de la méthode par dimension fractale par ondelettes sur la colonne centrale et par filtres de Gabor sur 4 orientations sur la colonne de droite à titre comparatif.

La Figure 3-0 illustre l'application de [ESP98] à la détection de cibles sur des images infrarouges issues de la base publique Chinalake. Les cibles sont détectées en utilisant trois niveaux d'ondelettes. Les fenêtres sont 4x4, 8x8, 16x16. L'hypothèse faite est que les cibles ont une signature fractale élevée sur toutes les orientations.

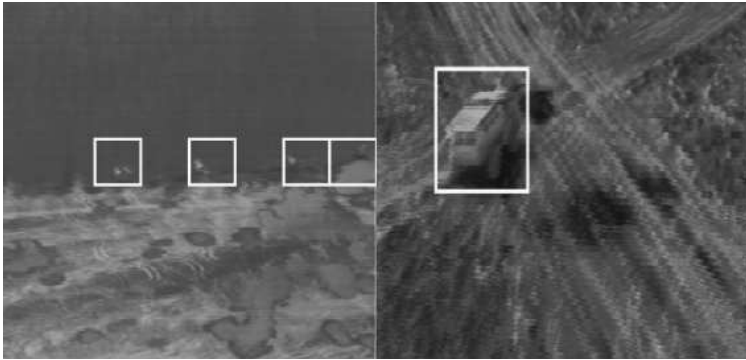
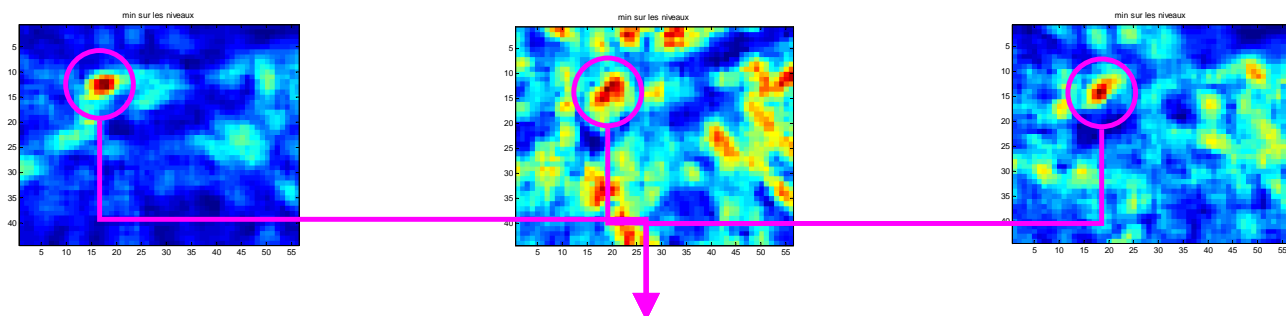
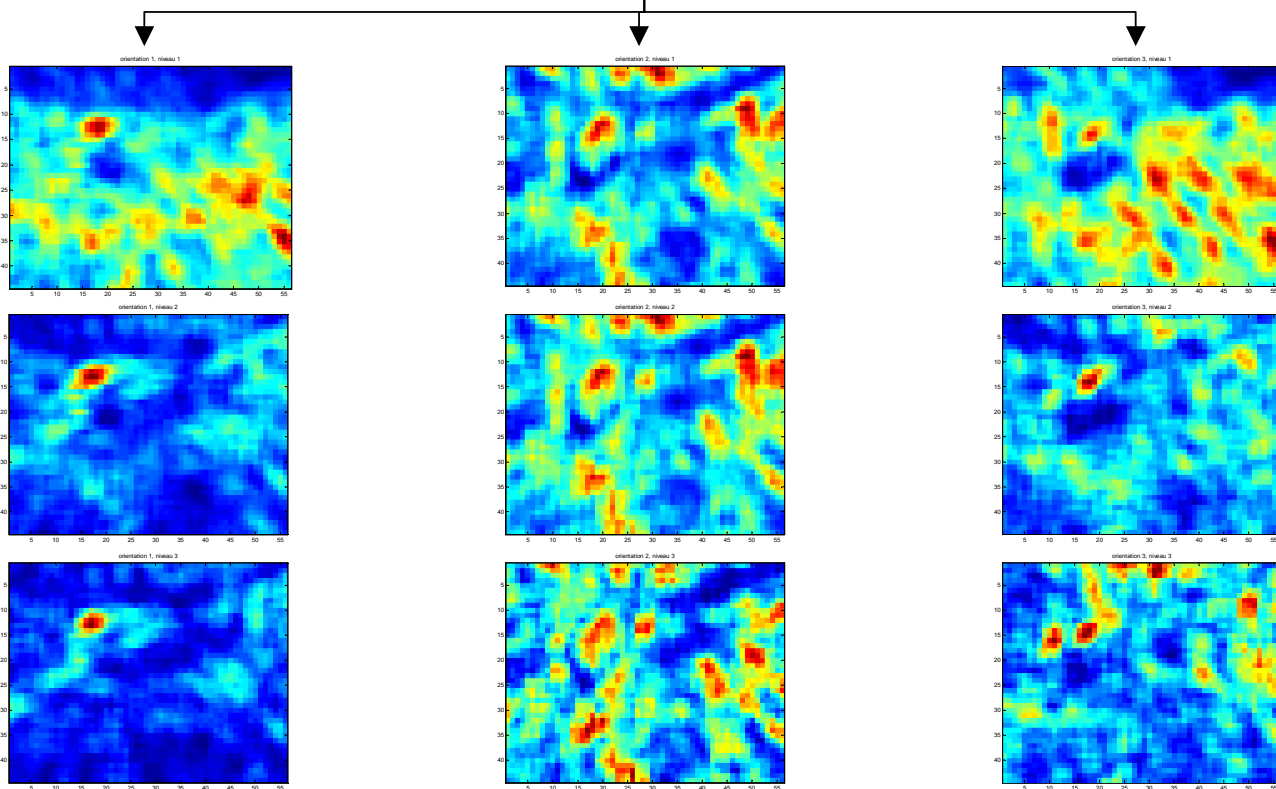


Figure 3-0 Application de [ESP98] à la détection de cibles (images issues de la base Chinalake).



La cible a une dimension fractale élevée sur les trois orientations

Figure 3-0 En haut, l'image initiale. Affichage de la réponse de la méthode sur 3 orientations (une orientation par colonne) et 3 niveaux de décomposition (lignes 1 à 3). La dernière ligne représente le minimum sur les différents niveaux, seuls les éléments à valeur élevée et constante persistent.

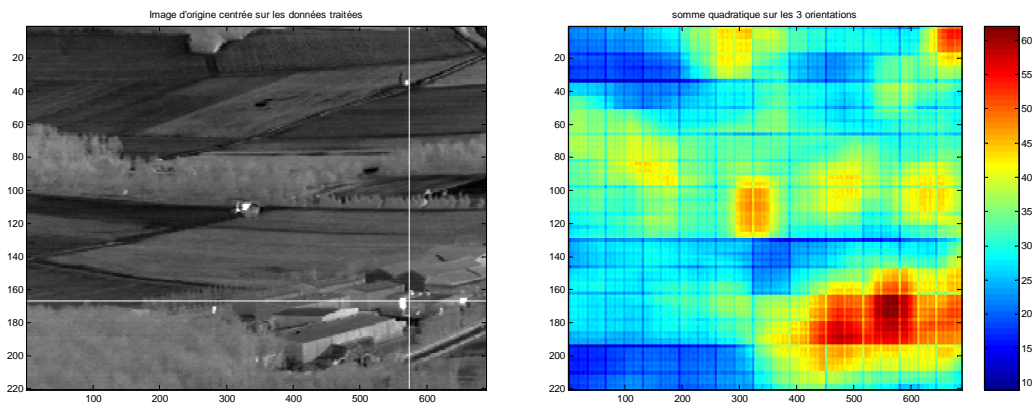


Figure 3-0 Application à des images infrarouges de la méthode présentée dans [ESP98].

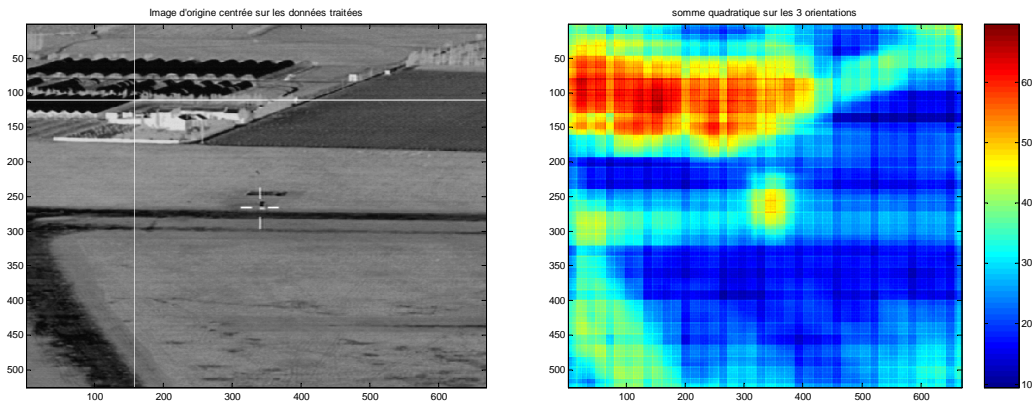
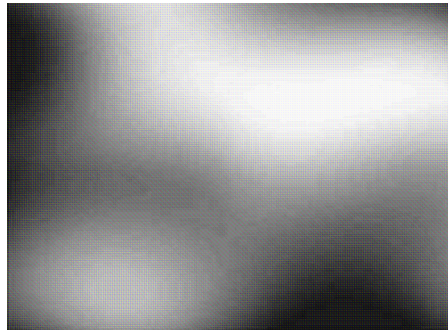


Figure 3-0 Application héliportée de la méthode présentée dans [ESP98].

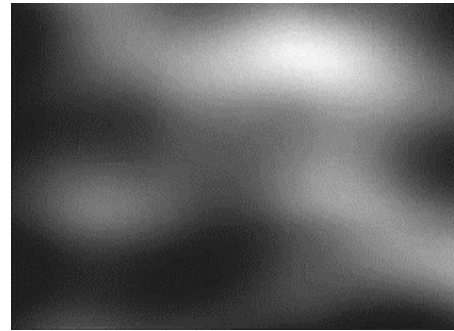
La Figure 3-0 et la Figure 3-0 illustrent la mise en application de la méthode de [ESP98] sur des images IR héliportées. L'opérateur renvoie globalement des valeurs faibles pour les zones naturelles (voir les champs sur la seconde image notamment) et des valeurs élevées sur les zones présentant un véhicule ou des bâtiments. Nous notons que la présence du réticule au centre de la seconde image fait également réagir l'opérateur, mais avec une valeur moyenne. En effet ces forts contrastes ne seront pas persistants sur les derniers niveaux de la décomposition, contrairement à ceux des bâtiments.

3.4.5 Lacunarité

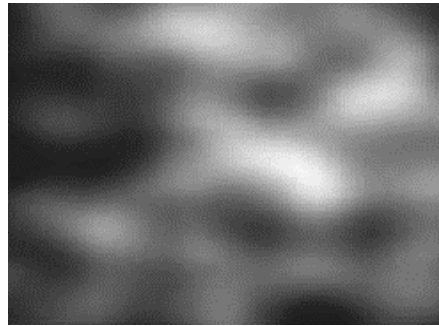
Selon [COC95], « la lacunarité est un paramètre du second ordre qui permet de mesurer la géométrie d'une région relativement au reste de l'image ». Etymologiquement, elle établit un lien entre le nombre de trous dans une texture et la répartition de ces trous. Elle permet de différencier des zones de textures différentes qui renvoient une même signature fractale. La Figure 3-0 illustre ce phénomène à travers trois textures synthétiques de même dimension fractale ($D = 2,5$) et de lacunarités différentes.



Lacunarité = 2



Lacunarité = 3



Lacunarité = 5

Figure 3-0 Plusieurs textures possédant la même dimension fractale et une lacunarité différente.

La lacunarité peut être définie comme le rapport entre deux moments statistiques dépendant d'une notion de masse que nous définissons ci-après. La masse est à définir selon l'application.

$$\Lambda(r) = \frac{Z_2}{Z_1^2} = \frac{\sum_M M^2 \cdot Q(M, r)}{\left[\sum_M M \cdot Q(M, r) \right]^2}$$

Une autre définition est donnée dans [BAR88] :

$$\Lambda(r) = \frac{Z_2 - Z_1^2}{Z_1^2}$$

où $Q(M, r)$ représente la fréquence d'apparition des éléments de masse M pour une taille d'élément r donnée.

Dans [DON00], l'auteur propose de définir une masse comme le nombre de boîtes nécessaires pour couvrir l'amplitude des niveaux de gris présents dans une fenêtre glissante. Le nombre de fenêtres d'une même masse permet de calculer le facteur $Q(M, r)$. La Figure 3-0 illustre ce découpage utilisé dans [DON00] suivant le plan image et l'axe correspondant aux niveaux de gris.

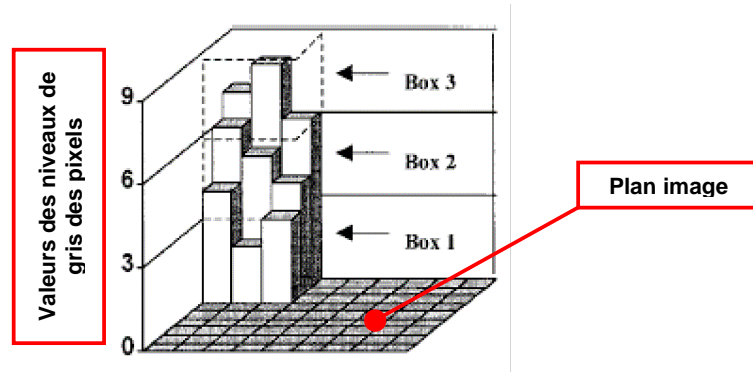


Figure 3-0 Illustration des boîtes utilisées dans [DON00].
Illustration extraite de l'article.

3.4.6 Multi fractales

Nous avons vu plus haut la définition de la dimension fractale, ainsi que quelques applications possibles. Nous avons constaté que pour une même texture, la dimension fractale changeait en fonction de la distance d'observation (cf. Figure 3-0). Cette variation en fonction de la distance d'observation varie d'une texture à l'autre. L'observation de cette variabilité peut donner des informations supplémentaires pour discriminer des textures qui auraient à l'origine des dimensions fractionnaires identiques. Ces observations sur plusieurs échelles, de la dimension fractale portent le nom de multifractale. En 1984, S. Peleg [PEL84] propose une étude basée sur ce procédé. Considérons la texture ou l'image comme une surface. Le principe de cette méthode est de calculer l'évolution du volume compris entre deux surfaces / nappes u (*upper*) et b (*bottom*), situées respectivement au-dessus et en dessous de la surface à étudier, en fonction d'une échelle d'observation ε . La Figure 3-0 illustre ce phénomène en considérant un signal monodimensionnel. Ces différentes échelles sont modélisées en effectuant, pour chaque nouvel ε , une érosion sur $b_{\varepsilon-1}$ et une dilatation sur $u_{\varepsilon-1}$. L'élément structurant est un masque 3x3, chacun de ses éléments valant 1. A l'origine, aucune dilatation ou érosion n'est encore effectuée, le volume entre u et b est donc nul et $u_0 = b_0 = I$, I étant la texture à étudier. Les cartes de valeurs u et b suivantes sont calculées de la manière suivante :

$$u_{\varepsilon}(i, j) = \max \left\{ u_{\varepsilon-1}(i, j) + 1, \max_{|(m,n)-(i,j)| \leq 1} u_{\varepsilon-1}(m, n) \right\}$$

$$b_{\varepsilon}(i, j) = \min \left\{ b_{\varepsilon-1}(i, j) - 1, \min_{|(m,n)-(i,j)| \leq 1} b_{\varepsilon-1}(m, n) \right\}$$

Le volume à l'échelle d'observation ε vaut

$$v_{\varepsilon} = \sum_{i,j} (u_{\varepsilon}(i, j) - b_{\varepsilon}(i, j))$$

La méthode correspondant à la définition originale de la dimension fractale voudrait que nous définissions la valeur observée A suivant les différentes échelles par

$$A(\varepsilon) = v_{\varepsilon} / 2\varepsilon$$

C'est-à-dire que nous mesurons le nombre de boules nécessaires pour couvrir l'espace à l'échelle ε . Cette approche est conservée dans [CAR90]. Peleg choisit d'observer la mesure suivante, celle-ci étant selon lui plus adaptée aux objets non fractionnaires :

$$A(\varepsilon) = \frac{(v_{\varepsilon} - v_{\varepsilon-1})}{2}$$

La Figure 3-0 illustre le calcul de u et b ainsi que l'évolution du volume compris entre ces deux surfaces sur un signal monodimensionnel.

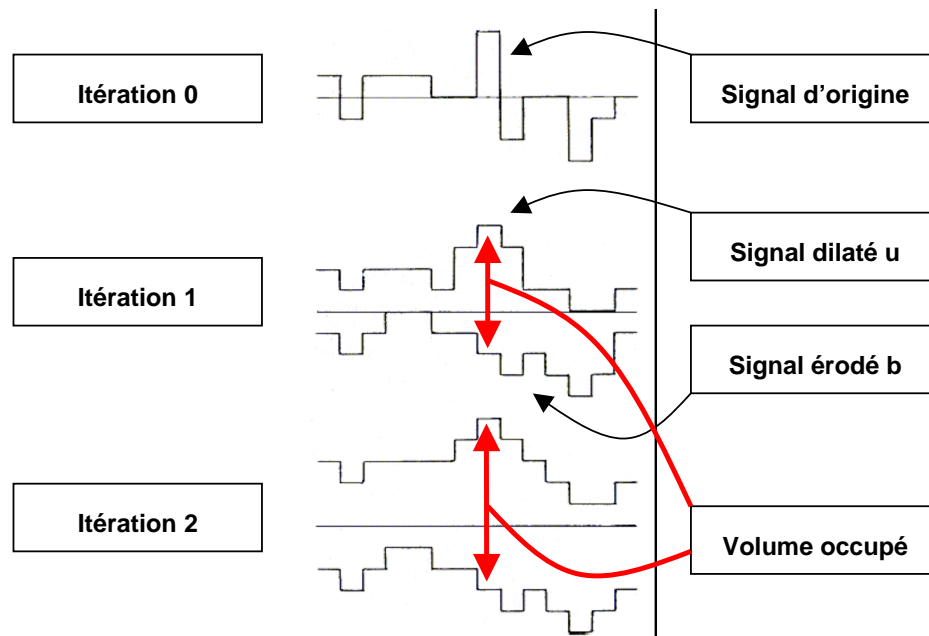


Figure 3-0 Exemple 1D d'évolution du volume compris entre les surfaces u et b définies dans [PEL84]. $V(1)=47$, $v(2)=78$, d'où $L(1)=(47-0)/2=23.5$, et $L(2)=(78-47)/2=15.5$.

L'évolution de $A(\varepsilon)$ dans un graphe log-log pour un objet fractal, correspond à une droite de pente $2-D$, où D est la dimension fractale de l'objet. La pente de la droite

associée à $A(\varepsilon)$ est également porteuse d'information. Elle est notée $S(\varepsilon)$ et est calculée par régression linéaire sur des triplets de couples

$$\begin{bmatrix} \log(\varepsilon - 1), \log(A(\varepsilon - 1)) \\ \log(\varepsilon), \log(A(\varepsilon)) \\ \log(\varepsilon + 1), \log(A(\varepsilon + 1)) \end{bmatrix}$$

Pour les objets fractionnaires, la mesure de $S(\varepsilon)$ est constante et vaut $2 - D$.

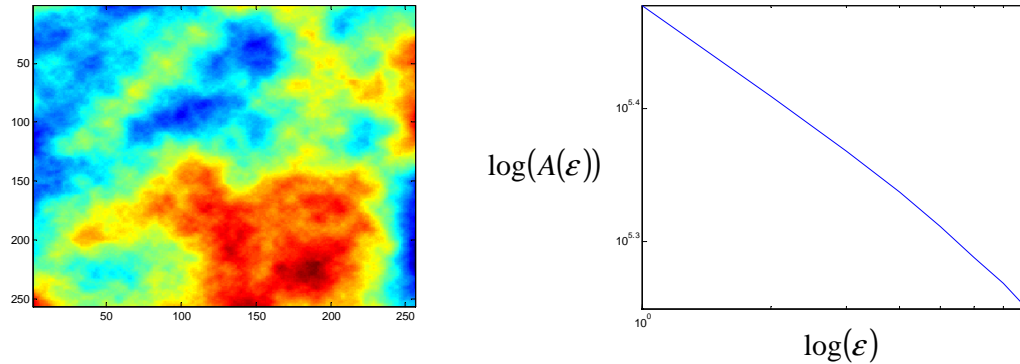


Figure 3-0 Brouillard fractal et courbe de Peleg associée.

Pour le brouillard fractal donné en Figure 3-0, nous obtenons pour le graphe $A(\varepsilon)$ en fonction de ε dans un repère log-log, une droite de pente $-0,3392$. Nous avons donc $-0,3392 = 2 - D$, soit $D = 1,6608$.

A partir des courbes $S(\varepsilon)$ en fonction de ε , Peleg définit une distance entre textures. L'expression de celle-ci se rapproche d'une distance quadratique où chaque élément est pondéré par un log. Une première distance entre deux textures i et j est donc définie par

$$D(i, j) = \sum_{\varepsilon} [S_i(\varepsilon) - S_j(\varepsilon)]^2 \cdot \log\left(\frac{\varepsilon + 1/2}{\varepsilon - 1/2}\right)$$

Dans [MAN83], Mandelbrot critique le fait que ce type d'approche soit asymétrique. Prenons la courbe présentée en Figure 3-0. On souhaite estimer la longueur de la frontière qui sépare les parties 1 et 2 de la figure. Une méthode possible est de choisir des boules d'un diamètre donné et de compter combien de boules on peut placer le long de la frontière. Cette opération est faite du côté partie 1 de la frontière et du côté partie 2.

Le diamètre choisi d'une boule correspond à la finesse de la mesure. Si on choisit des boules de petit diamètre ($\varnothing < 1$), on obtient la même estimation de la longueur de frontière. En effet, le recouvrement créé par les boules de petit diamètre sera adapté aux deux « fissures » qui entaillent la « partie 2 » de la figure. Par contre, dans le cas où on choisit des boules de diamètre plus important, l'estimation peut varier. Si on remplit la « partie 2 » avec des boules de grand diamètre (boules rouges sur la figure), l'estimation

de la longueur de la séparation sera proche de la réalité, puisque les « fissures » sépareront les boules. Dans le cas où nous remplissons la « partie 1 » avec des boules de fort diamètre, les boules ne peuvent pas entrer dans les « fissures » et l'espace à l'intérieur de ces dernières est alors complètement ignoré. Dans ce dernier cas, la frontière est considérée comme une droite, et l'estimation de sa longueur est complètement fautive.

On a donc dans la « partie 2 », une estimation de la longueur de la séparation (« de la côte » si on reprend l'application classique d'estimation de la longueur de côte d'un pays), qui devient de plus en plus grossière quand on augmente le diamètre des boules (c'est-à-dire l'échelle d'observation), ce qui est un comportement tout à fait normal. Par contre dans le cas de la « partie 1 », au-delà d'un certain diamètre des boules (c'est-à-dire au-delà d'une certaine finesse de mesure), l'estimation passe de « grossière » à « complètement fautive ». Il y a donc asymétrie dans le comportement de l'estimation de la valeur recherchée, selon le point de vue choisi (partie 1 ou partie 2).

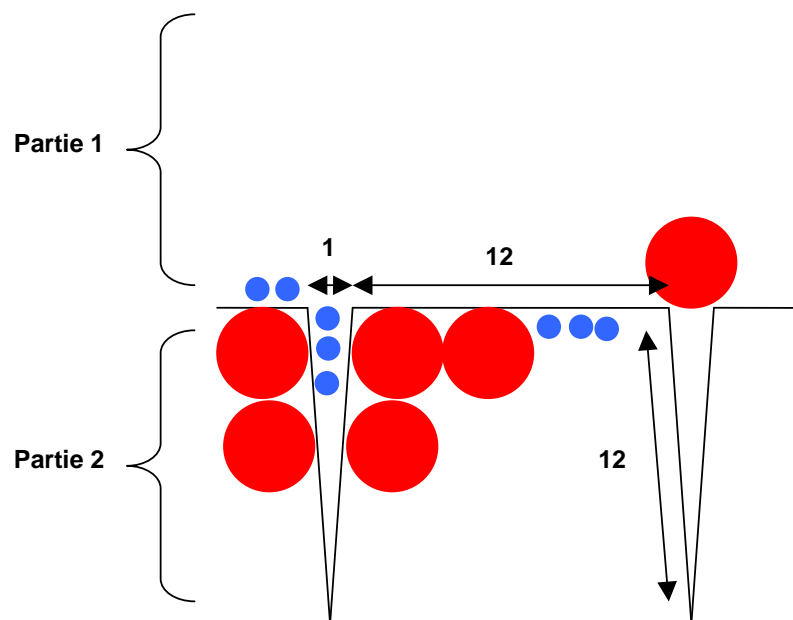


Figure 3-0 Illustration de la dissymétrie de mesure entre la partie 1 et la partie 2. Selon la finesse de la mesure choisie (taille du diamètre des boules) et surtout si la mesure est estimée d'après la partie 1 ou la partie 2, la ligne centrale est estimée soit à une longueur de 68, soit avec une longueur de 20.

Peleg propose de mettre à profit cette dissymétrie et de l'appliquer à des textures. Les calculs des volumes $v(\varepsilon)$ entre les surfaces u , b et g , et les aires $A(\varepsilon)$ sont alors redéfinis.

$$v_{\varepsilon}^{+} = \sum_{i,j} (u_{\varepsilon}(i, j) - g(i, j))$$

$$v_{\varepsilon}^{-} = \sum_{i,j} (g(i, j) - b_{\varepsilon}(i, j))$$

Avec ces nouvelles définitions, nous passons donc de la notion de « volume » seule, aux notions de « volume haut » et « volume bas » (respectivement v_{ε}^+ et v_{ε}^-).

$$A^+(\varepsilon) = v_{\varepsilon}^+ - v_{\varepsilon-1}^+$$

$$A^-(\varepsilon) = v_{\varepsilon}^- - v_{\varepsilon-1}^-$$

Les $S^+(\varepsilon)$ et $S^-(\varepsilon)$ sont calculés de la même manière que $S(\varepsilon)$ mais à partir de $A^+(\varepsilon)$ et $A^-(\varepsilon)$ respectivement. Il en découle alors une nouvelle mesure de distance entre textures, où chaque élément est à nouveau pondéré par un log.

$$D'(i, j) = \sum_{\varepsilon} \left\{ \left[(S_i^+(\varepsilon) - S_j^+(\varepsilon))^2 + (S_i^-(\varepsilon) - S_j^-(\varepsilon))^2 \right] \log \left(\frac{\varepsilon + 1/2}{\varepsilon - 1/2} \right) \right\}$$

L'auteur applique ces mesures à la reconnaissance de texture. Pour cela, un certain nombre d'échantillons extraits d'images de la base Brodatz sont extraits (Seafan, Burlap, Ice, Pig, Raffia, Mica). Pour chaque échantillon, la distance par rapport à tous les autres échantillons est calculée. Sur les textures choisies, les échantillons Pigskin et Ice étaient mal classifiés en utilisant la première distance. Avec la seconde distance D' , 100% des échantillons sont classifiés correctement.

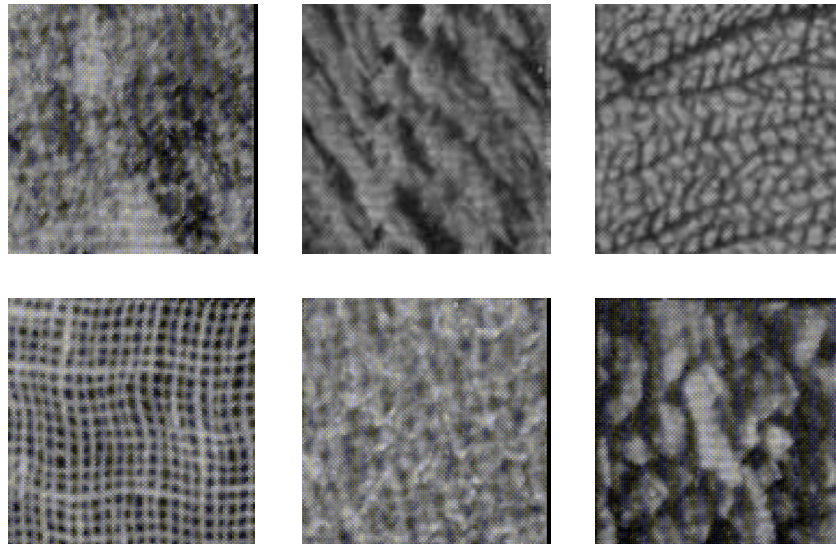


Figure 3-0 Echantillons de textures issues de la base Brodatz : Ice, Raffia, Seafan, Burlap, Pigskin, Mica.

Sur la Figure 3-0, nous observons que pour une scène à prise de vue fixe, c'est-à-dire pas de changement du fond, la courbe S- reste inchangée quand les véhicules présents sont regroupés ou dispersés. Par contre, lorsque les véhicules sont rapprochés, la courbe

S+ prend des valeurs plus élevées. Le même constat est fait dans la Figure 3-0, où un seul véhicule s'éloigne du groupe. Plus le véhicule s'éloigne plus la courbe S+ globale de la scène se décale vers des valeurs faibles. Nous noterons que le déplacement d'un seul véhicule produit un déplacement de la courbe plus faible que le déplacement de tous les véhicules. Par contre, dans les deux cas, nous observons une variation de la courbe dans le même sens quand les véhicules convergent (resp. divergent) les uns vers les (resp. des) autres.

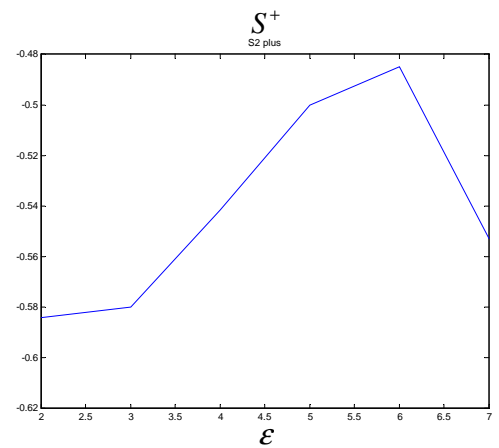
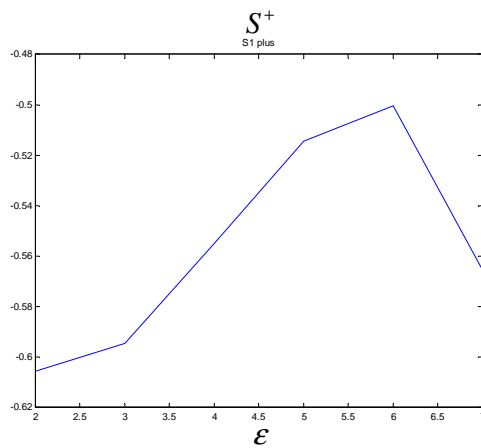
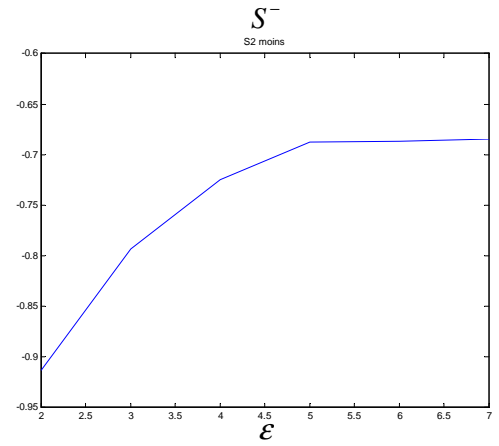
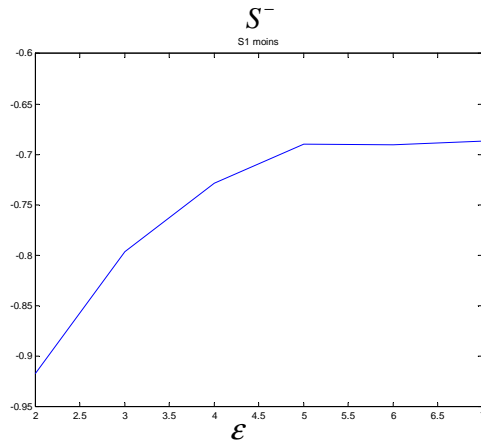


Figure 3-0 Indice de dispersion d'objets (ici des chars) dans une scène en domaine visible. La signature multi fractale S^- ne change pas d'une image à l'autre (représentation du fond de la scène, selon l'auteur). Par contre, lorsque l'on regroupe les chars, l'ensemble des valeurs de la courbe S^+ augmentent.

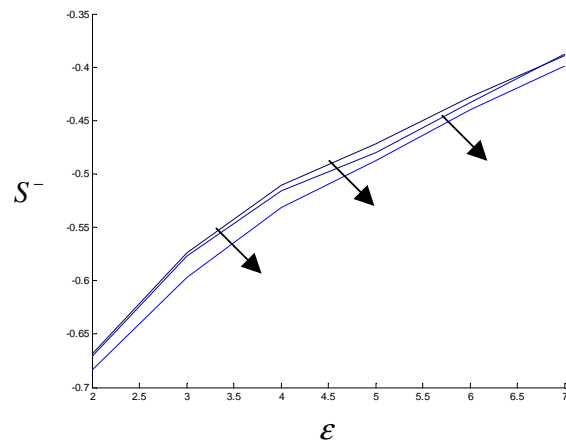


Figure 3-0 Evolution de la courbe S+ en fonction de l'éloignement d'un seul véhicule (déplacement vers la partie en haut à droite de l'image). Au fur et à mesure que le véhicule s'éloigne, la courbe prend des valeurs plus faibles.

Les deux variantes de [PEL84] sont résumées dans les diagrammes suivants.

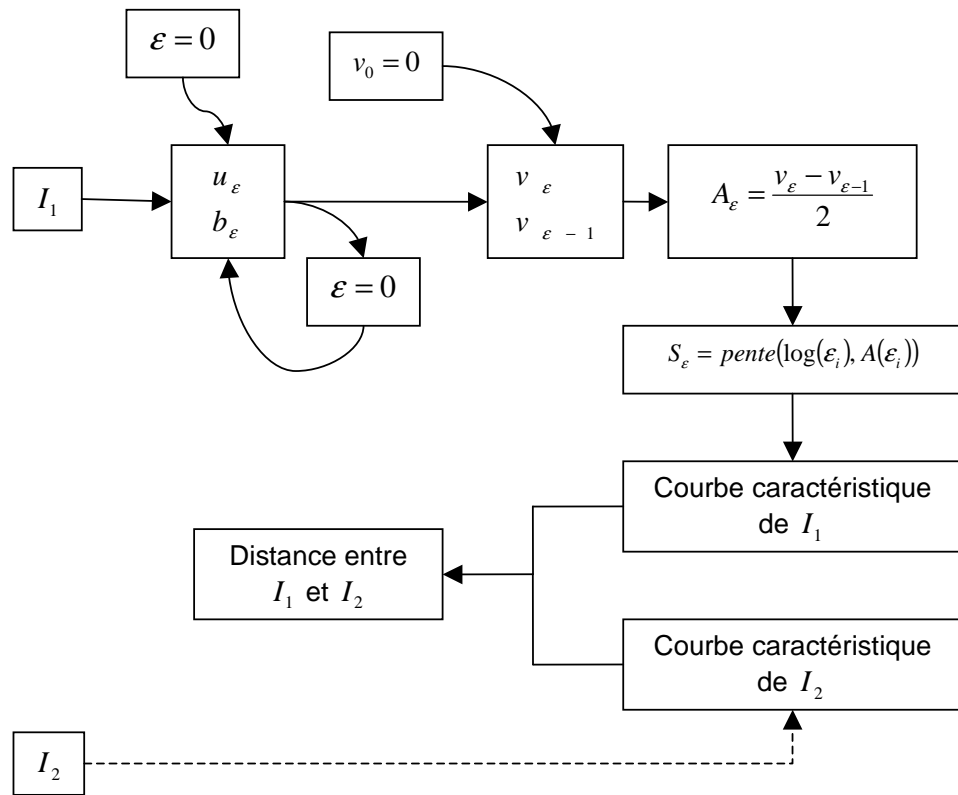


Figure 3-0 Diagramme de la première méthode de Peleg pour deux images I_1 et I_2 .

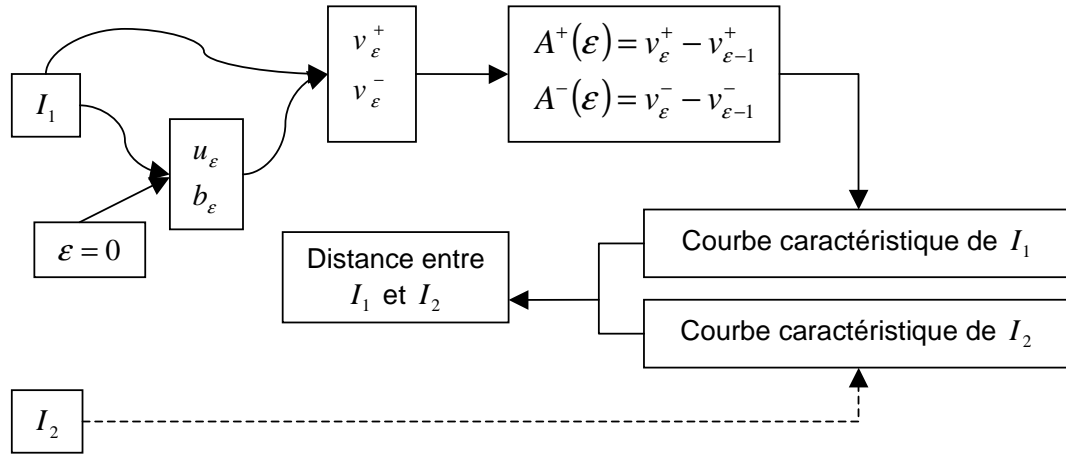


Figure 3-0 Diagramme de la seconde méthode de Peleg

Segmentation automatique par multifractales

Nous avons vu que les courbes fournies par l'étude multifractale d'une texture permettaient, dans une certaine mesure, de la caractériser en se basant sur une étude multiéchelles. Si on applique le calcul de courbes multifractales à chaque position d'une fenêtre glissante, et qu'on considère chaque courbe comme un vecteur d'attribut, on peut alors appliquer une méthode de clusterisation pour segmenter l'image. La segmentation par k-moyennes a été appliquée en utilisant la distance définie dans [PEL84]. La Figure 3-0 illustre un résultat de segmentation automatique en 2 classes par k-moyennes sur des courbes de Peleg.

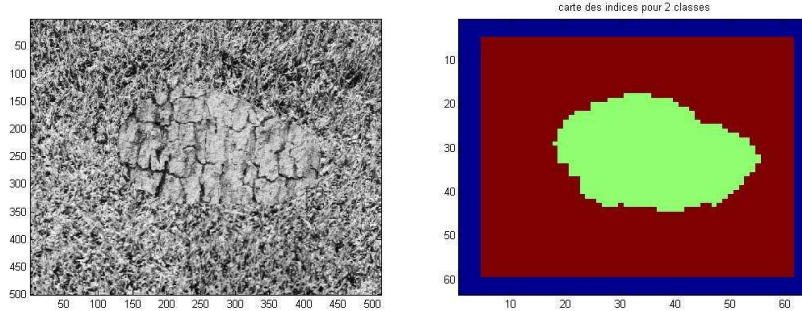


Figure 3-0 Exemple de segmentation par multi fractales. A gauche, une image synthétique composée de deux textures Brodatz. A droite le résultat de segmentation par multi fractales.

Le résultat sur cette image comportant deux classes de textures semble intéressant, par contre la méthode montre ses limites quand le nombre de classes est plus grand. La Figure 3-0 illustre une segmentation sur 4 classes pour une image composée de 4 textures issues de la base Brodatz.

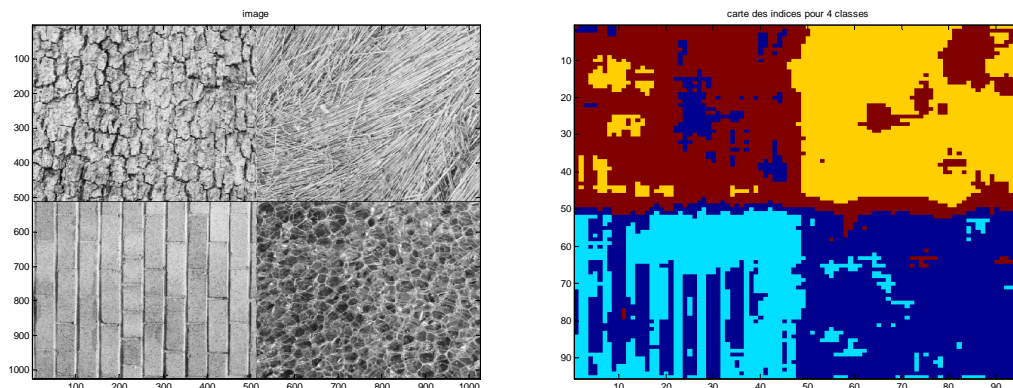


Figure 3-0 Segmentation par multifractales. L'image d'entrée (a) est composée de 4 images de textures issues de la base Brodatz. (b) représente le résultat de la segmentation de (a) par k-moyennes sur des données multifractales sur 4 classes.

Les multifractales sont un outil qui peut être utilisé dans des applications diverses. Les résultats qu'elles fournissent sont intéressants mais elles montrent leurs limites en termes de discrimination au delà d'un certain nombre de textures présentes sur l'image. Les résultats obtenus dans [PEL84] ont cependant démontré leur efficacité pour la classification de textures pour des *échantillons de grande taille* (images Brodatz 512x512).

Nous avons vu quelques outils statistiques basés sur les fractales, nous allons maintenant nous intéresser à une tout autre application des fractales, à savoir la compression de données.

3.4.7 Compression de données image

Une application répandue des fractales est la compression de données, et plus particulièrement d'images en ce qui nous concerne. L'idée de base de cette méthode de compression est de trouver des similarités dans l'image à différentes échelles. Cette application étant un axe majeur de nos travaux, elle sera détaillée plus loin dans le document. Nous ne pouvons cependant pas terminer notre état de l'art sans explorer les bases de cet outil.

Le principe de la compression fractale repose sur la recherche de similarités dans l'image à compresser. La mise en évidence de similarités dans les données permet de choisir la partie d'information à mémoriser dans un contexte de compression. La Figure 3-0 représente la génération du triangle de Sierpinsky à partir d'une image quelconque, ici un cheval. Dans cette image, chaque « triangle » est similaire à chacune des trois sous parties qui le composent. Quelle que soit l'image d'origine (ici un cheval), la suite des images constituées converge vers une image unique (le triangle de Sierpinsky) au fil des transformations de chaque triangle en trois « parties ».

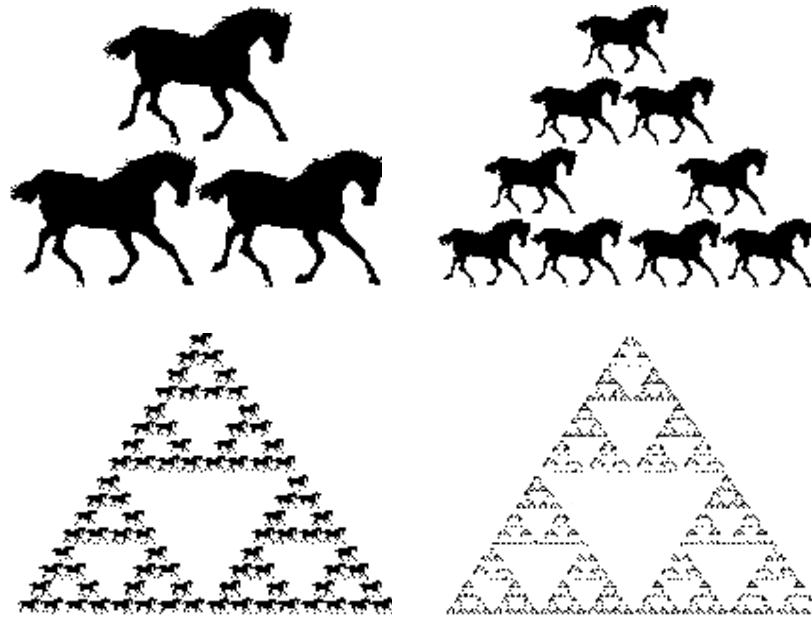


Figure 3-0 Exemple de génération du triangle de Sierpinsky à partir d'un motif d'origine, ici un cheval. Quel que soit le motif de base, l'ensemble des transformations contractantes font toujours converger la suite d'images vers une figure unique.

Le triangle de Sierpinski est une image purement fractale, c'est-à-dire autosimilaire, ce qui n'est pas le cas d'une image quelconque. Dans le cas d'une image quelconque, on cherche les similarités entre les régions de l'image (classiquement des imasettes rectangulaires) à différentes échelles, et modifications des niveaux de gris près. La Figure 3-0 illustre une similarité entre deux imasettes de tailles différentes sur l'image de Lena.

Dans ce type de compression, aucune information pixellique n'est stockée. On ne mémorise que les transformations liant des couples d'imasettes. Une transformation est classiquement composée de la position de chacune des imasettes du couple en question, du facteur d'échelle et d'une transformation linéaire des niveaux de gris de la première imasette pour reconstruire au mieux la seconde imasette.



Figure 3-0 Exemple de similarité sur l'image de Lena

La décompression s'effectue de manière itérative. On part d'une image quelconque, de même dimension que l'image compressée (l'image initiale peut être une image monochrome de niveau de gris à 128, une scène naturelle, etc.). Par applications successives des transformations mémorisées lors de la compression, l'image va se transformer et converger vers l'image compressée.

Bien que cette recherche de similarités dans l'image ait pour application majeure la compression de données images, les propriétés de la compression fractale ont été utilisées également dans le domaine du débruitage comme c'est le cas dans [GHA03]. L'extraction d'éléments similaires de différentes tailles permet de diminuer le bruit de manière significative en préservant au mieux les contours. Une autre application possible de la compression fractale est le streaming vidéo. Une méthode de streaming vidéo à base d'ondelettes et de fractales a été proposée dans [ZHA97]. Une application plus connexe à notre étude a été la segmentation automatique [YIN97].

3.5 Conclusion

Dans ce chapitre nous avons fait un état de l'art de quelques méthodes de segmentation. Nous avons mis en relief le lien fort entre la séparation des données image et leur identification. Nous avons ensuite exploré un ensemble d'opérateurs permettant de caractériser les zones texturées. Finalement nous avons approfondi notre prospection sur les outils fractionnaires. La partie statistique de ces outils a montré son efficacité dans la gestion de textures naturelles. Nous avons vu également que les fractales permettaient d'effectuer une compression de données avec pertes. Ce dernier point va être détaillé dans le chapitre suivant. Nous expliquerons comment nous nous sommes inspiré de ce type de compression pour en faire un processus d'apprentissage de concepts.

4 Apprentissage de Concepts

4.1 Position du problème

Nous avons vu plus haut qu'une texture peut être beaucoup de choses. Elle peut être modélisée fidèlement par une fonction périodique ou par un objet composite plus complexe. Apprendre une texture ou un concept composé de plusieurs textures, peut être équivalent à extraire les éléments de base qui le composent et caractériser le placement des composants les uns par rapport aux autres. La recherche de similarités dans l'image peut nous aider à comprendre la constitution de la texture considérée en mettant en évidence les redondances présentes dans l'image. Une similarité peut être relativement immédiate pour l'œil humain si les objets comparés sont quasi identiques, mais ce type de redondance est trop restrictif pour apparaître suffisamment souvent et donc nous apporter une information concluante. Dans la courbe de von Koch (Figure 4-0), les motifs se répètent indéfiniment à des échelles différentes. De plus, la transformation contractante générant ce flocon contient une partie rotation qu'il faut prendre en compte si l'on veut retrouver l'élément original.

Caractériser une texture par ce type d'approche revient à définir précisément ce que l'on entend par redondance et les degrés de liberté qu'on introduit volontairement pour pouvoir généraliser sous forme apprentissage (notamment la rotation pour la courbe de von Koch). Dans la Figure 4-0, l'élément de base est un segment. Il est transformé en 4 sous segments égaux, chacun de longueur $1/3$ par rapport au segment d'origine.

La procédure de construction d'un tel flocon est la suivante :

1. On prend un segment de longueur L
2. On le transforme en 4 sous segments de longueur $L/3$, disposés comme présenté dans la Figure 4-0
3. Répéter récursivement les étapes 1 et 2 pour tous les segments présents dans la figure.

Caractériser cet objet revient, d'après ce que nous venons de dire, à retrouver la transformation qui doit être appliquée à chaque itération et qui sert à générer la forme de la Figure 4-0. On a déjà mentionné que la dimension fractale de cet objet est

$$D = \frac{\log(4)}{\log(3)} = 1,26.$$

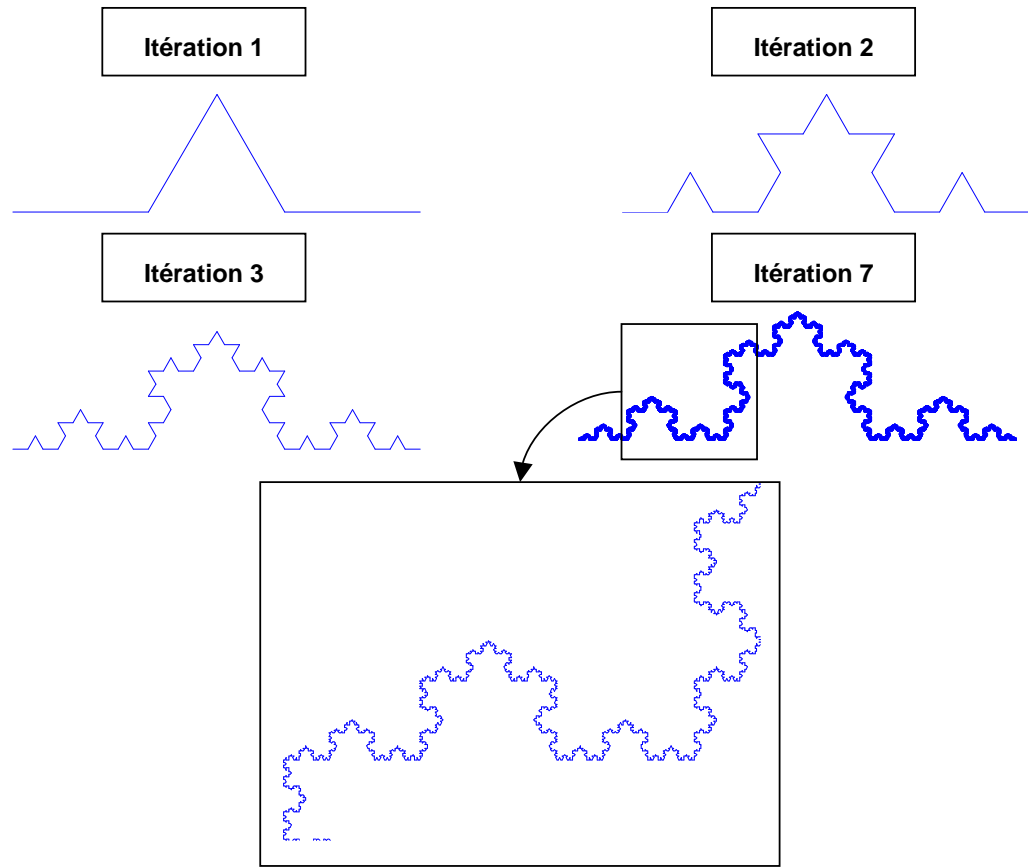


Figure 4-0 Génération de la courbe de von Koch pour les itérations 1,2,3 et 7. Un zoom sur l'itération 7 montre la présence des motifs à différentes échelles.

Dans des images en niveaux de gris, les transformations contractantes comportent une partie géométrique (changement d'échelle, isométrie), comme pour les segments du flocon de von Koch, mais également une partie sur le niveau de gris de l'objet à manipuler (classiquement : un bloc image). Nous voulons mettre en relief des redondances dans l'image au travers de ces transformations. Nous rappelons que nous souhaitons que ces redondances soient robustes aux paramètres variant fréquemment d'une scène à l'autre, à savoir le contraste, la rotation, le décalage en luminosité, etc. La construction de telles transformations est prise en compte dans des domaines comme les systèmes de fonctions itérés (IFS : *iterated function systems*) et, par extension, la compression fractale. Notre étude s'est donc naturellement orientée vers ce type d'outils. Nous allons voir le principe global de la compression fractale, puis nous montrerons comment nous nous sommes inspiré de cette méthode pour construire une méthode de calcul de score affecté aux éléments de l'image.

Version de Nicole toto

Notre objectif est de reconnaître différentes zones dans l'image. Ces zones sont souvent caractérisées par une texture et sont donc ces textures que nous voulons étudier. Les niveaux de détails ne sont pas connus et ce sont ces éléments que nous voulons mettre en évidence. Nous faisons l'hypothèse que les zones significatives sont celles qui

se retrouvent à différentes échelles dans l'image et que l'on retient dans le processus de compression fractale. Après avoir précisé le principe de la compression fractale nous montrerons comment un concept, c'est-à-dire un ensemble de structure globale caractéristique peut être appris à partir de l'extraction de certaines imagerie. Nous nous attarderons sur les différentes stratégies que nous avons élaborées pour résumer l'information contenue dans une image. Ce sont ces imagerie fondamentales que nous utiliserons dans le chapitre 5 pour reconnaître les concepts. Ce chapitre peut donc être vu comme une phase d'apprentissage que nous exploiterons dans le chapitre suivant.

4.2 Construction d'une transformation contractante

Dans cette section, nous allons nous intéresser au principe de la compression fractale. Nous le définirons d'un point de vue théorique dans un premier temps, puis nous nous pencherons sur sa mise en œuvre, à savoir la construction d'une transformation contractante permettant de mettre en évidence des redondances dans une image.

4.2.1 Principe de la compression fractale

Le principe de la compression fractale repose sur le théorème du point fixe. Celui-ci assure la convergence d'une suite d'images vers une limite, une image unique : l'image compressée.

Théorème : Théorème du point fixe pour une application contractante : Soient E un espace métrique complet (non vide) et f une application contractante de E dans E (de rapport de contractance k , avec $k < 1$). Alors il existe un point fixe unique x^* de f dans E , c'est-à-dire tel que $f(x^*) = x^*$. De plus toute suite d'éléments de E vérifiant la formule de récurrence $x_{n+1} = f(x_n)$ converge vers x^* , avec $x_0 \in E$.

Définition : Une application f de \mathfrak{X} dans \mathfrak{X} est une application **contractante** k -lipschitzienne avec $k \in [0,1[$, c'est-à-dire que l'application f est contractante si

$$\exists k \in [0,1[\ / \ \forall (x, y) \in \mathfrak{X}^2, |f(x) - f(y)| \leq k \cdot |x - y|$$

et

plus généralement si

$$\forall (x, y) \in E^2, d_F(f(x), f(y)) \leq k \cdot d_E(x, y)$$

où (E, d_E) et (F, d_F) sont des espaces métriques et $f : E \rightarrow F$.

Cette convergence peut s'observer par exemple sur une suite, $(x_n)_{n \in \mathbb{N}}$

$$x_0 = a \in \mathfrak{X}$$

$$x_{n+1} = \frac{1}{2} \cdot x_n + 10$$

qui converge vers 20 quelle que soit la valeur initiale x_0 , ou sur une image, comme l'illustre la Figure 3-0 avec le triangle de Sierpinsky. En partant d'un motif quelconque, ici un cheval, l'image converge vers une forme unique, le triangle. Cet objet fractal possède une propriété forte d'*autosimilarité*. C'est-à-dire qu'il est composé d'un même motif d'origine à plusieurs échelles d'observation. Étymologiquement le mot autosimilaire vient de l'anglais *self-similar*. Il n'a pas encore été accepté par l'Académie. Autant il est aisé de construire une image d'un objet fractionnaire à partir d'un modèle fixé a priori, autant il

peut être difficile d'effectuer l'opération inverse, c'est-à-dire, trouver les applications *contractantes* qui ont permis de générer l'objet.

Les images fractales sont la plupart du temps générées de manière artificielle. A partir d'une image initiale et d'un ensemble de transformations, on crée l'image synthétique. La grande majorité des images réelles, issues d'un capteur, ne sont pas fractales. Il est alors difficile de partir de l'image pour trouver les transformations contractantes qui pourraient la générer (pour peu que de telles transformations existent). Le but, dans ce cas, est de trouver une transformation contractante dont le point fixe soit une image qui soit la plus proche possible visuellement de l'image d'origine. La compression fractale est donc une méthode de compression avec perte. La compression fractale repose sur les systèmes de fonctions itérées ou IFS (Iterated Function System). Dans ce type de compression, aucune information pixellique n'est mémorisée, seules les transformations qui lient des parties similaires de l'image sont stockées. Un IFS est un ensemble de fonctions contractantes $T_i : M \rightarrow M$ dans un espace métrique E . On définit une fonction également contractante sur l'ensemble des parties compactes non vides de E muni de la distance de Hausdorff.

$$T = \bigcup_{i=1}^N T_i : P(E) \rightarrow P(E)$$

Le théorème du point fixe assure l'existence et l'unicité d'un sous-ensemble $F \subset M$ tel que $T(F) = F$. F est appelé attracteur de l'IFS. Dans le cadre de la compression d'image, il est très rare qu'une image soit identique à elle-même (dans sa totalité) à plusieurs échelles. Pour cette raison on utilise le principe d'IFS partitionné ou PIFS (Partitioned IFS) : on partitionne l'image, puis pour chaque élément de la partition on veut trouver une zone de l'image qui lui corresponde à une transformation contractante près. Dans notre cas, la partition sera fournie par une simple grille. Ces éléments seront appelés des *ranges*. Nous les avons choisis carrés. Les zones de l'image à faire correspondre avec chaque range seront appelées *domains*.

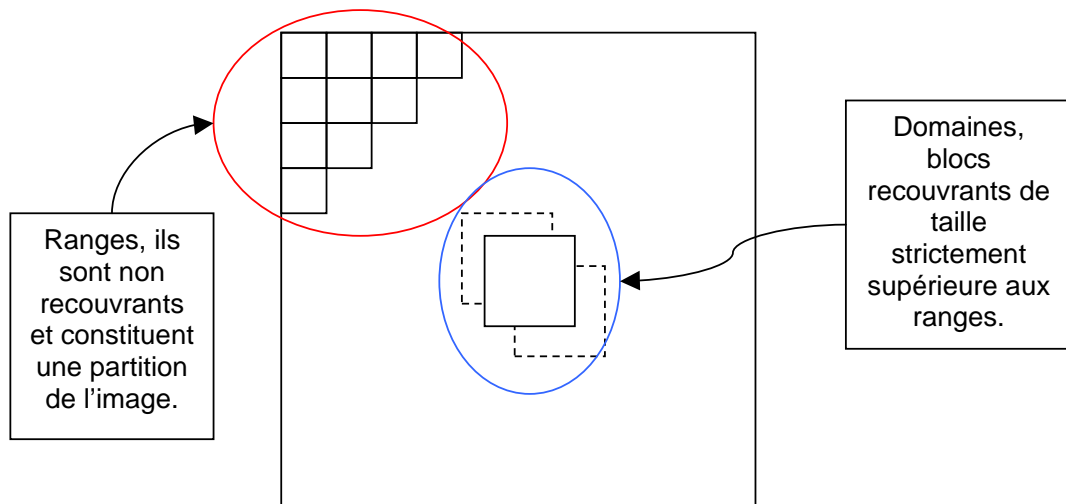


Figure 4-0 Exemple d'autosimilarité sur l'image de Lena.

La phase de décompression de l'image correspond à l'application répétée des transformations composant le PIFS. L'image initiale doit avoir les mêmes dimensions que l'image compressée mais son contenu peut être quelconque. A chaque itération, l'image

est transformée et se rapproche visuellement de l'image compressée (convergence due à l'ensemble des transformations). On stoppe les itérations quand on estime que l'image reconstruite est suffisamment « bonne » ou qu'il n'y a plus de changement d'une itération à la suivante. De nombreux articles sont déjà parus dans ce domaine dans les vingt dernières années. Notre problème n'est pas celui de la compression d'image mais nous nous intéressons à la technique de construction des transformées. Certains des travaux traitent plus spécifiquement de la taille mémoire occupée par l'image compressée [HSU02], d'autres s'orientent vers l'optimisation de la recherche de mise en correspondance entre des éléments similaires dans l'image ([LAI03], [HAM96], [WOH95], [DIS06], [BAR94]): un problème majeur dans la compression fractale est, comme on le verra plus loin, le parcours de l'ensemble des domaines. Ainsi même si la décompression d'une image est très rapide, le temps de compression peut rapidement exploser si on utilise un parcours exhaustif simple. Nous venons de considérer la compression fractale d'un point de vue théorique, nous allons maintenant nous intéresser à sa mise en œuvre.

4.2.2 Construction d'une transformation contractante et compression de l'image

La compression fractale d'une image consiste à trouver les similitudes entre les différentes parties de celle-ci moyennant certaines transformations. Ces dernières portent notamment sur la géométrie et les modifications des niveaux de gris des parties de l'image considérées. Nous rappelons que les régions de grandes tailles sont appelées domaines et que les régions de petite taille sont appelées ranges. Dans la suite du document, quand aucune indication ne sera précisée, les domaines et ranges seront des imageries carrées. Cette représentation n'est pas la seule possible, il en existe d'autres (Figure 4-0), mais elle est la plus utilisée et fournit la matière suffisante pour ce que nous voulons traiter dans cette section.

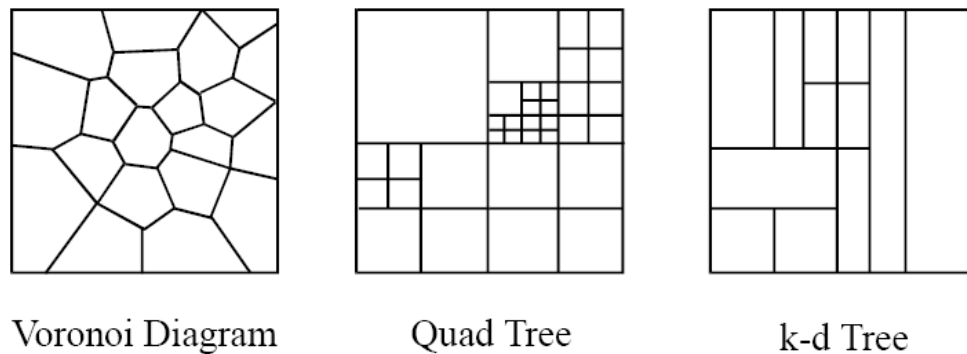


Figure 4-0 Différents types de partitions d'image utilisées dans la compression fractale.

Les modifications des niveaux de gris sont classiquement modélisées par une transformation linéaire appliquée aux niveaux de gris des pixels du domaine pour obtenir les niveaux de gris des pixels du range. Pour assurer la propriété de contractance de la transformation, une condition suffisante est que le facteur d'échelle entre les ranges et les domaines, ainsi que le paramètre de contraste soient strictement plus petits que un.

Un choix classiquement retenu pour rechercher et modéliser les transformations consiste à trouver, pour chaque range R et domaine D , les paramètres qui correspondent le mieux à l'expression suivante :

$$R \cong iso(res(D,s),iIso)c+l$$

où $red(D,s)$ est le résultat d'une homothétie (réduction) de D par un facteur s , $iso(B,iIso)$ est l'application d'une isométrie référencée par $iIso$ sur le bloc image B . c et l sont les paramètres de contraste et luminosité et peuvent être calculés par la méthode des moindres carrés entre les niveaux de gris des pixels du range et les niveaux de gris des pixels correspondant du domaine transformé. Dans le cas de blocs image carrés, nous limiterons, dans un premier temps, les isométries considérées aux huit possibilités de rotations et symétries ($iIso \in [1..8]$). Ces isométries utilisées dans [JAC90] et dans la plupart des publications utilisant des imagerie carrées, sont représentées dans le Tableau 4-0. B représente un bloc image carré de côté N .

Identité	$B'(i,j)=B(i,j)$
Symétrie horizontale	$B'(i,j)=B(N+1-i,j)$
Symétrie verticale	$B'(i,j)=B(i,N+1-j)$
Rotation de π	$B'(i,j)=B(N+1-i,N+1-j)$
Symétrie verticale + rotation de $\pi/2$	$B'(i,j)=B(j,i)$
Rotation de $-\pi/2$	$B'(i,j)=B(N+1-j,i)$
Rotation de $\pi/2$	$B'(i,j)=B(j,N+1-i)$
Symétrie verticale et rotation de $-\pi/2$	$B'(i,j)=B(N+1-j,N+1-i)$

Tableau 4-0 : Tableau récapitulatif des 8 isométries utilisées dans la méthode de Jacquin [JAC90]. Le bloc B est transformé en B'.

Des transformations plus complexes pourraient être considérées [LEE96], mais la combinatoire de recherche de la « bonne » transformation peut rapidement exploser. En effet, bien que certains paramètres comme le contraste et la luminosité soient calculables directement de manière analytique, d'autres devront être parcourus sur l'ensemble de leurs valeurs possibles ou être approximés de manière itérative. Si nous voulons utiliser les PIFS uniquement pour compresser une image (tout en préservant une bonne reconstruction de celle-ci), certaines contraintes peuvent être imposées sur les degrés de liberté. Le facteur d'échelle peut par exemple être fixé à 0,5 comme l'explique Jacquin dans [JAC90] et [JAC92]. De même, le facteur de rétrécissement en niveaux de gris pourra être fixé à 0,75, le paramètre l sera alors égal à la différence des niveaux de gris moyens des blocs domaine et range.

En plus des paramètres de contraste et luminosité, il est également possible d'introduire une pente (un « tilt ») sur les niveaux de gris, relativement aux axes

horizontaux et verticaux de l'image considérée. Si on considère que les imagerie I_1 et I_2 sont vues comme la concaténation de leurs lignes respectives, alors trouver les tilts des niveaux de gris entre deux imagerie suivant les axes verticaux et horizontaux revient à résoudre (par la méthode des moindres carrés par exemple) l'égalité suivante :

$$I_2 = l + tilt_j \cdot liste_j + tilt_i \cdot liste_i + c \cdot I_1$$

avec

$$liste_i = [1,1,1,1,2,2,2,2,\dots,N,N,N,N\dots],$$

$$liste_j = [1,2,\dots,N,1,2,\dots,N,\dots]$$

Où l est la luminosité, c le contraste, $tilt_i$ et $tilt_j$ des scalaires représentant la pente appliquée sur l'axe des niveaux de gris, respectivement selon les axes verticaux et horizontaux de l'imagerie carrée de côté N pixels.

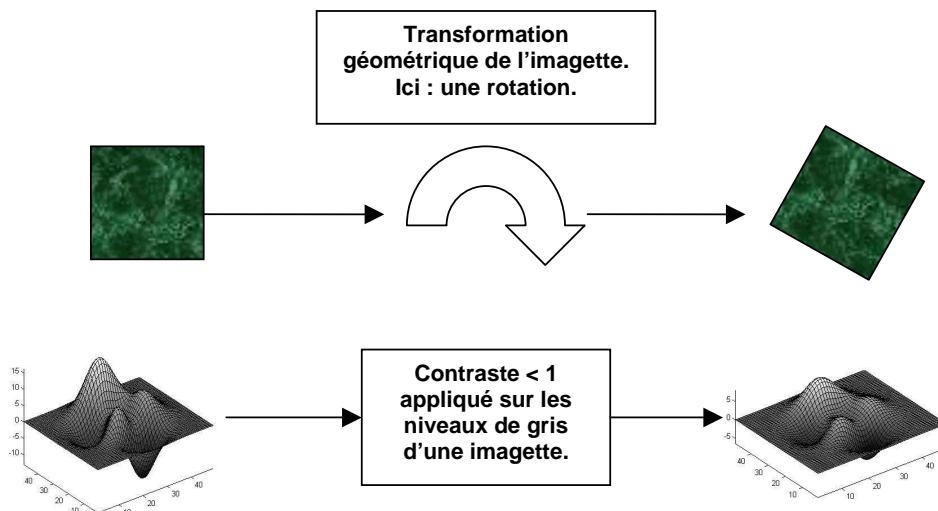


Figure 4-0 Exemple d'opérations de transformation géométriques et sur les niveaux de gris d'une imagerie.

On dit que le domaine D code le range R si la correspondance entre D et R , par $T = (D, R, iso, s, c, l)$, est suffisamment bonne, c'est-à-dire que les imagerie $T(D)$ et R peuvent être considérées comme similaires selon un critère donné: PSNR, RMS, etc. Comme il est possible d'utiliser plusieurs méthodes de mesure de la qualité de reconstruction, nous appellerons « mesure d'erreur de reconstruction » la fonction représentant le PSNR ou le RMS ou toute autre fonction choisie. Une liste non exhaustive de ces fonctions est définie à la section 4.3.2.

Plusieurs méthodes de compression avec des partitionnements adaptatifs existent dans la littérature (Figure 4-0), basées entre autre, sur les quadrees, la triangulation de Voronoï, les k-d trees [BEN90]. Le but de ces partitionnements est de partager l'image en éléments homogènes, susceptibles de matcher les uns avec les autres, tant en terme de forme (à une transformation près) que d'un point de vue contenu (statistiques sur les pixels inclus dans la région).

Dans notre étude, nous avons choisi de nous limiter à un partitionnement fixe de l'image. Nous démontrerons plus loin la robustesse de notre approche aux différents partitionnements en grille (décalage de la grille et recouvrement des blocs).

Une méthode plus coûteuse en temps de calcul mais permettant de faire abstraction du nombre limité de rotations (multiples de $\pi/2$) est de considérer non plus des imagerie carrées mais uniquement le plus grand disque inscrit dans ce carré. On applique alors l'algorithme fourni dans [LAG98] pour calculer l'angle qui minimise un critère d'erreur (à définir, le RMS par exemple) entre le disque de l'imagerie I_1 et celui de I_2 .

Maintenant que nous disposons d'outils pour définir et comparer deux blocs range/domaine de l'image, nous devons mettre en place une stratégie de parcours et de mise en correspondance de ces couples. Cette stratégie est étroitement liée au processus d'apprentissage que nous détaillons plus loin. Pour le moment nous allons présenter une stratégie simple pour clore cette présentation sur la compression fractale.

La **compression** consiste à définir les ensembles des ranges et domaines, une mesure d'erreur de reconstruction entre deux imagerie (le RMS par exemple), les transformations contractantes qui lient les ranges et domaines ainsi que la valeur de la mesure d'erreur pour chaque couple range/domaine (appelée *erreur de reconstruction*), et un seuil relatif à cette mesure. Les ensembles ranges et domaines sont créés à partir des partitionnements définis plus haut. Le seuil permet de dire si la correspondance (matching) entre deux imagerie range/domaine est bonne ou non. Pour chaque range, nous devons trouver au moins un domaine qui lui corresponde, c'est-à-dire qui conduise à une erreur de reconstruction inférieure au seuil choisi. Quand nous comparons l'erreur de reconstruction à ce seuil, nous considérons toujours la meilleure transformation possible (isométrie, contraste, luminosité, etc.). Nous verrons plus loin que le choix de cette mesure d'erreur est important et peut dépendre des imagerie que nous voulons caractériser : imagerie à contenu plutôt géométrique (bâtiment, véhicule) ou plutôt statistique (végétation). Un algorithme de compression fractale est indiqué à la Figure 4-0.

```

Entrées : les ranges R, les domaines D
Sorties : les transformations T liant un range à un domaine

Pour chaque range Ri
    bDomaineTrouvé ← FAUX
    Pour chaque domaine Di & tant que bDomaineTrouvé == FAUX
        Mettre le domaine à l'échelle du range
        Pour chaque isométrie
            & tant que bDomaineTrouvé == FAUX
                Appliquer l'isométrie au domaine réduit → D'i
                Calculer le meilleur couple (c,l) pour matcher
                    D'i avec Ri
                D''i ← D'i*c + l
                erreur ← RMS(D''i,Ri)
                Si erreur<=seuil_erreur alors
                    Mémoriser T(Ri,Di)
                    bDomaineTrouvé ← VRAI
                Fin si
            Fin pour chaque isométrie
        Fin pour chaque domaine
    Fin pour chaque range

```

Figure 4-0 Pseudo-code de compression fractale d'image.

Nous disposons pour chaque range, en fin d'exécution de cet algorithme, du premier domaine suivant le parcours utilisé qui a satisfait la condition de passage sous le seuil de l'erreur de reconstruction ainsi que de la transformation qui les fait correspondre. La stratégie de parcours des domaines est à définir, il s'agit d'un point important qui sera traité lors de la phase d'optimisation. Pour le moment, nous pouvons par exemple considérer le parcours simple et gourmand consistant à parcourir les domaines de haut en bas et de gauche à droite dans le repère image.

Nous noterons qu'un seuil élevé provoquera une compression rapide mais de mauvaise qualité : un seuil infini revient à associer tous les ranges avec le premier domaine rencontré, seul le contraste et la luminosité sont pertinents. A l'inverse un seuil nul implique un parcours exhaustif, pour chaque range, de tous les domaines possibles. La compression sera alors optimale, relativement aux paramètres a priori : partitionnement, transformations, mesure de l'erreur de reconstruction. Dans le pseudo code ci-dessus, nous remarquons qu'il est possible de gagner en temps de calcul en inversant les boucles « pour chaque range » et « pour chaque domaine », ainsi que la boucle de transformations du domaine concerné. Pour un domaine donné, chaque transformation n'est alors effectuée qu'une seule fois.

La **décompression** est bien plus directe, elle ne comporte pas de phase de recherche. Comme expliqué plus haut, la compression fractale ne stocke aucune donnée pixellique, uniquement des transformations, c'est-à-dire l'ensemble de définition et la formule de la transformation. La Figure 4-0 illustre les données classiquement stockées dans une compression fractale.

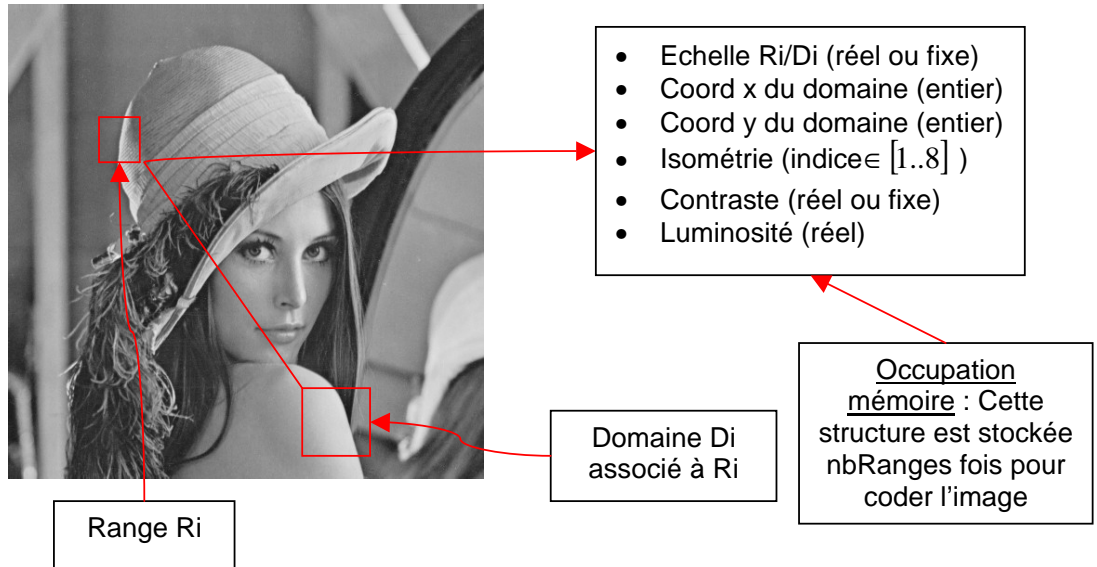


Figure 4-0 Structure classique des données stockées dans une compression fractale.

La décompression de l'image consiste à appliquer un certain nombre de fois les transformations extraites dans la partie compression. Concrètement, nous remplaçons le contenu de chaque range par le bloc image situé aux coordonnées du domaine correspondant, après application de la transformation liant le range et le domaine, calculée lors de la compression. Au fur et à mesure que les transformations seront appliquées à l'image, la suite des images constituées converge vers l'image compressée. L'image initiale peut donc être une scène quelconque. La Figure 4-0 illustre ce phénomène de convergence. Ici l'image est stable après 4 itérations.

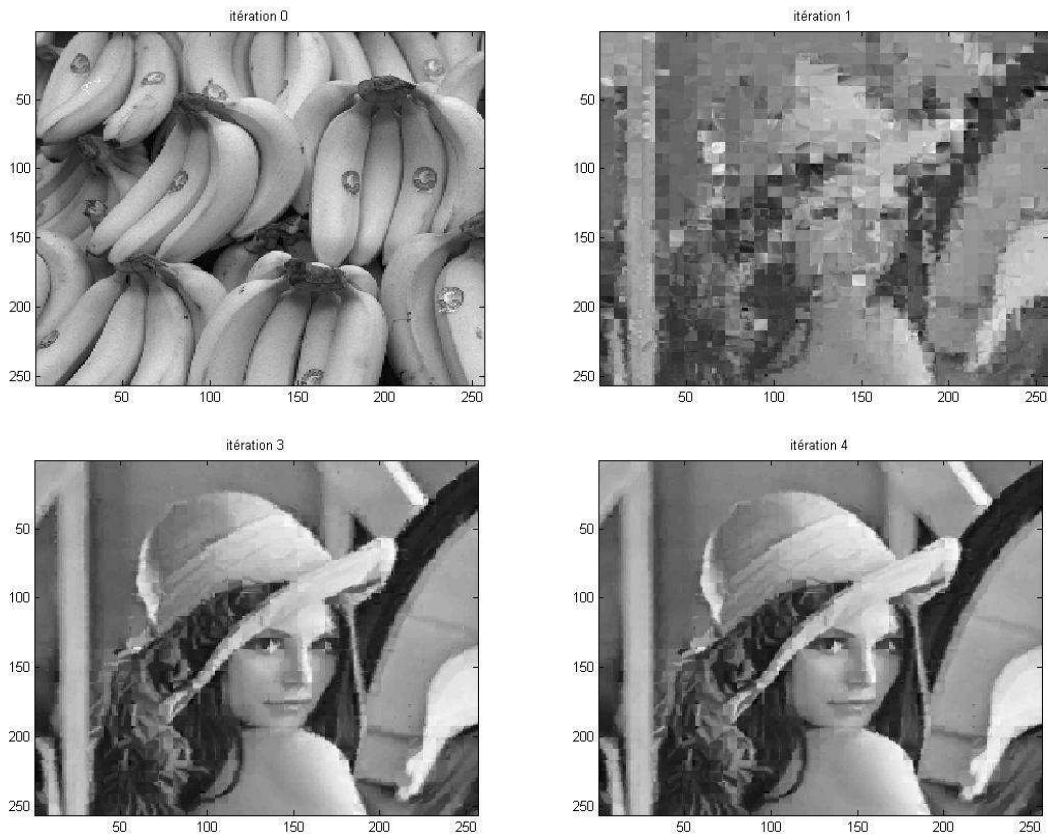


Figure 4-0 Reconstruction de Lena, itération après itération. L'image d'initialisation est une image issue de la base d'images "Chicago".

Plus l'image initiale (itération 0) est proche de l'image compressée, moins on aura besoin d'itérer l'ensemble des transformations pour atteindre l'image compressée. Le nombre d'itérations peut être fixe (souvent après 3 itérations l'image a fortement convergé, elle évolue peu après 6 à 8 itérations) ou adapté en fonction du changement de l'image d'une itération à l'autre. En observant les deux algorithmes compression / décompression nous constatons le décalage important en coût de calcul qu'il peut y avoir entre les deux. En effet, la phase de compression peut rapidement tendre vers le parcours exhaustif de couples range/domaine. Au contraire, la décompression est immédiate, nous ne faisons qu'appliquer les transformations trouvées lors de la compression. Un algorithme de décompression fractale est indiqué à la Figure 4-0.

Entrées : les transformations T liant un domaine à chaque range, vDim (vecteur 2D contenant la taille de l'image à décompressée)

Sorties : l'image I décompressée

I ← image quelconque de dimension vDim

Pour chaque itération

Pour chaque range Ri

 Di ← I(coord du domaine correspondant à Ri)

 D''i ← T[Di,Ri](Di)

 Coller D''i aux coordonnées de Ri

Fin pour chaque range

Fin pour chaque itération

Figure 4-0 Algorithme de décompression fractale.

La taille et la qualité finale de l'image décompressée par rapport à son apparence originale avant compression dépend de plusieurs paramètres, ceux-ci étant parfois liés les uns aux autres.

La partition est un paramètre important. Elle intervient sur la qualité de l'image dans le sens où si elle est bien adaptée à l'image, un nombre élevé de couples ranges/domaines seront susceptibles de correspondre avec une très bonne qualité de reconstruction. Le partitionnement intervient également sur la taille des données compressées. En effet il n'est pas toujours évident de caractériser et trouver ce qu'est un partitionnement « bien adapté à l'image » a priori. Pour cette raison, une solution peut être de choisir un partitionnement plus fin, ce qui correspond dans le cas d'un partitionnement en grille carrée, à passer par exemple de blocs image de taille 8x8 pixels à des blocs de taille 4x4 pixels. Plus les blocs (ranges/domaines) qu'on doit faire correspondre sont de petite taille, plus il est aisé d'avoir une bonne reconstruction, et ce avec des transformations simples. Pour des images 4x4, des isométries avec des rotations de 90° sont amplement suffisantes. Dans le cas 32x32, des rotations plus fines peuvent être envisagées. On peut prendre en compte les transformations supplémentaires sur les niveaux de gris, autres que le gain et l'offset (contraste et luminosité), la pente (« tilt ») des niveaux de gris par exemple. Et dans le cas extrême 1x1, purement théorique (puisque sans intérêt d'un point de vue compression), les transformations géométriques deviennent obsolètes.

Le temps accordé à la recherche de couples ranges/domaines, ainsi que la stratégie de parcours de ces couples influent également sur l'image compressée. Dans le cas le moins contraint au niveau temporel, nous parcourons tous les couples ranges/domaines. La qualité de reconstruction est alors optimale relativement au partitionnement et aux transformations choisies. De nombreux articles traitent des techniques d'optimisation. Dans [LAI03], chaque bloc image B est représenté par un scalaire

$$\bar{B} = \|B\|^2 - \frac{1}{n} \cdot \sum_{i=1..n} B_i,$$

où $\|\cdot\|^2$ est la norme 2 et n le nombre de pixels de B .

Pour coder le range R avec le domaine D , Lai propose de ne parcourir que les domaines vérifiant les deux conditions suivantes :

- $\overline{R} - \overline{D} \leq d_{\min}$, d_{\min} étant un seuil qui reste à définir.
- $\sqrt{\frac{\overline{R}}{\overline{D}}}$ est supérieur à un seuil donné. Ce seuil représente un contraste minimum de R par rapport à D .

Hamzaoui, dans son papier de 1996 [HAM96], propose de regrouper les blocs à l'aide d'une carte de Kohonen. Ce principe est repris dans [MAR98], qui le combine avec la transformée en cosinus discret (DCT). La DCT avait déjà été utilisée dans les méthodes d'optimisation de compression fractale par Wohlberg en 1995 [WOH95]. Dans [DIS06], l'auteur propose de situer chaque bloc image par rapport à l'erreur relative à un « range moyen ». Cette erreur est utilisée pour placer les blocs dans un KD-Tree qui servira à trouver rapidement un domaine susceptible de coder un range donné.

Les transformations autorisées paramètrent aussi la qualité de l'image compressée. Elles représentent les degrés de liberté pour la comparaison des blocs image. Plus elles sont nombreuses et riches plus elles permettent de faire correspondre des blocs image entre eux. Pour autant, établir un ensemble riche de transformations pour permettre de coder au mieux un range par un domaine donné n'est pas forcément bénéfique à tous points de vue. La reconstruction de l'image sera meilleure, cependant nous émettrons quelques réserves.

Premièrement, un ensemble de transformations avec un cardinal élevé fait baisser le taux de compression de l'image. En effet, pour chacune d'elles il faut stocker les paramètres pour chaque range de l'image, ce qui augmente la taille de mémoire unitaire occupée par un range (cf. structure représentée en Figure 4-0).

Deuxièmement, dans le cas où les transformations deviennent particulièrement souples et nombreuses, nous nous trouvons dans une situation où n'importe quel domaine peut coder n'importe quel range. Dans la suite du document, nous montrerons que nous utilisons la compression fractale pour extraire des imageries représentatives de l'image. Si pratiquement tous les domaines peuvent coder tous les ranges, il devient alors difficile de mettre des éléments en valeur par rapport aux autres de manière pertinente.

Nous disposons maintenant de suffisamment d'outils pour savoir si des ranges et domaines peuvent être considérés comme similaires. Nous allons définir une première notion de score pour chaque domaine. Le score d'un domaine est égal, dans cette section, au nombre de ranges qu'il peut coder relativement à un seuil.

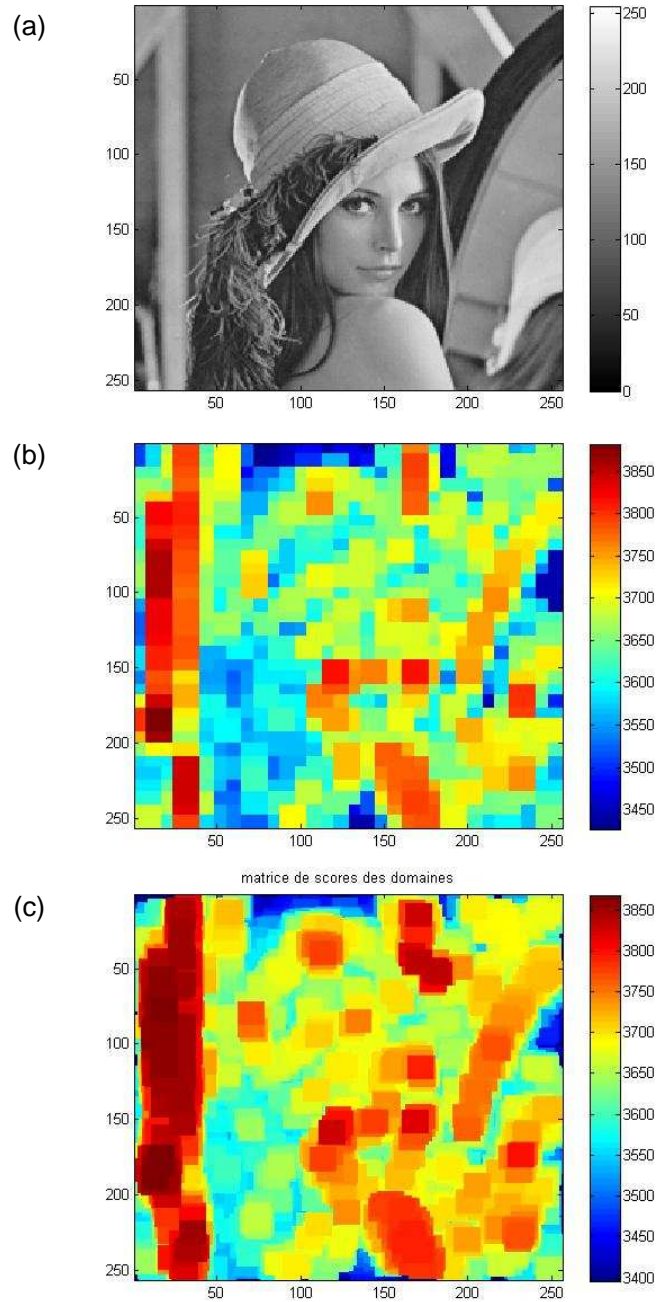


Figure 4-0 Image de Lena (a), carte des scores des domaines (16x16) avec un pas de parcours de 8 pixels (b), avec un pas de 1 pixel (c). Le niveau retenu dans les zones de superposition est le niveau maximum.

La Figure 4-0 illustre la compression de l'image de Lena (256x256, image (a)), et les cartes des scores des domaines (16x16), respectivement avec un pas de parcours de 8 et 1 pixels (image (b) et (c)). Les ranges sont de taille 8x8 et non recouvrants. Nous remarquons une certaine stabilité dans les scores, tant au niveau de leurs valeurs (entre 3400 et 3850 dans les deux cas), que dans leur répartition spatiale : les valeurs élevées demeurent dans les mêmes régions d'une carte à l'autre. Ici, les domaines qui ont les scores les plus élevés sont les parties horizontales/verticales (« bords droits »), les bords du chapeau, du miroir, et les blocs homogènes.

La carte avec un pas de 1 pixel pour les domaines correspond à un parcours exhaustif. Le temps de calcul est approximativement 64 fois plus long avec un pas de 1 pixel qu'avec un pas de 8 pixels. Le comparatif entre ces deux parcours nous permet d'avoir une idée de la stabilité de la carte des scores en fonction de la finesse du parcours des domaines. Une autre question que nous sommes en droit de nous poser avec cette première méthode est la stabilité des scores en fonction de la partition. Est-ce que les scores restent stables si on décale la partition par exemple ? Les modifications de partition en taille de range ne seront pas traitées dans cette section, elles seront l'objet de discussions plus loin dans ce document.

Le calcul de score a donc été effectué en décalant d'un demi range la partition d'origine.

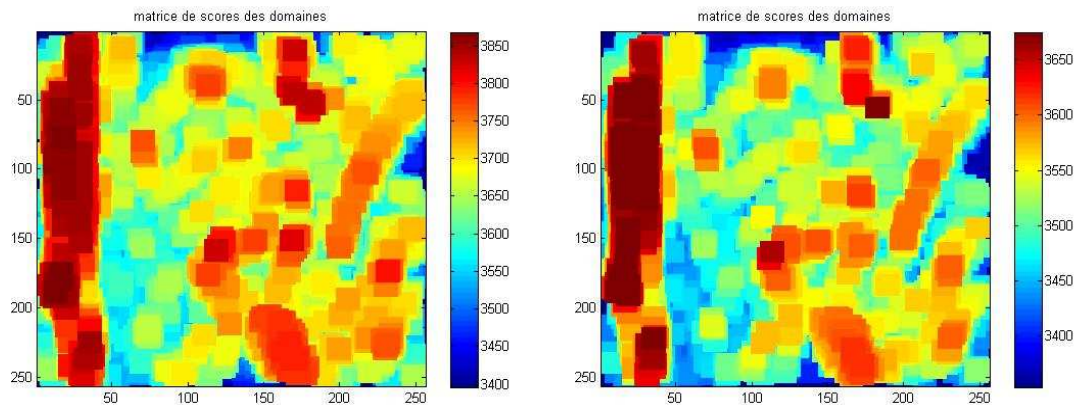


Figure 4-0 Cartes des scores avec partition d'origine (image 1) et partition décalée d'un demi range (image 2). Les deux cartes ont été calculées avec un pas de parcours de 1 pixel pour les domaines (parcours exhaustif).

La Figure 4-0 illustre l'impact d'une modification de la partition par décalage des ranges. En observant les échelles de valeurs, nous constatons que les deux partitions fournissent des scores sensiblement identiques (valeurs légèrement plus élevées pour la première partition).

Nous avons vu dans cette section comment compresser et décompresser une image en associant les ranges et domaines de celle-ci. Nous avons constaté que certaines parties de l'image pouvaient être considérées comme redondantes. Nous allons expliquer dans la suite de ce document comment ces redondances nous servent de base pour mettre en place une méthode de caractérisation de textures.

4.3 Mise en place d'une méthode d'apprentissage de concepts

Lors de la compression fractale d'une image, en fonction du seuil de tolérance sur l'erreur de reconstruction range/domaine, nous remarquons que certains domaines sont plus utilisés que d'autres pour compresser l'image. Cela est dû au fait qu'ils sont présents plusieurs fois dans l'image et à plusieurs échelles. Si nous considérons qu'un élément redondant dans un signal est représentatif de ce dernier, alors il peut être intéressant de mémoriser ces éléments caractéristiques (leur contenu ou leurs attributs) pour pouvoir retrouver ce signal dans un autre contexte. Dans notre cas, le signal en question est une image, un concept ou une texture qu'on souhaite apprendre et les éléments caractéristiques sont des blocs de l'image. Pour pouvoir dire que des imajettes sont représentatives en terme de redondance, il faut pouvoir leur attribuer un score de manière à les ordonner et mettre en relief les plus intéressantes. Nous avons vu à la section précédente une première approche pour le calcul simple d'un score. Nous testerons dans cette section une seconde approche d'extraction d'éléments redondants. Nous nous intéresserons ensuite aux méthodes de comparaison d'imajettes. Ce point constitue une zone sensible de la chaîne algorithmique que nous allons mettre en œuvre au fil de ce document. Nous nous pencherons ensuite sur les dépendances entre couples range/domaine pour quantifier de manière robuste la redondance des éléments que nous voulons extraire. Nous verrons ensuite les problèmes de seuillage. La première approche décrite en début de chapitre nous montrera les limitations d'une méthode par seuil fixe sur la qualité de la mesure d'erreur de reconstruction. Une méthode par seuil adaptatif sera donc mise en œuvre pour résoudre ce problème. Enfin, après avoir étudié l'extraction de domaines redondants, nous nous placerons dans une optique d'extraction de ranges redondants et comparerons l'efficacité, en termes de pertinence sur l'apprentissage, chacun de ces choix.

La section suivante définit une seconde méthode de calcul de score de blocs par application d'un seuil fixé a priori sur l'erreur de reconstruction entre un range et un domaine. Nous verrons quels sont les éléments d'apprentissage importants qui sont mis en évidence par cette approche ainsi que ses limites.

4.3.1 Première approche

La manière la plus intuitive pour définir le score d'un domaine est celle que nous avons introduite à la fin du paragraphe 4.2. Elle consiste à compter combien chaque domaine peut coder de ranges selon un seuil et une mesure d'erreur de reconstruction définis a priori. On rappelle qu'à chaque fois, on considère la meilleure transformation possible pour un couple range/domaine donné. La Figure 4-0 illustre les cartes de scores des domaines sur Lena avec des seuils respectifs de 1 et 2 selon l'erreur de reconstruction définie comme étant la distance euclidienne pixel à pixel, normalisée par le nombre de pixels de chaque imajette. Le second seuil étant moins contraignant, nous observons naturellement une augmentation globale des valeurs min et max de l'intervalle de scores : les intervalles sont passés de [3400,3900] à [4800 5400]. Les domaines de l'image possédant un fort score dans la première estimation (seuil=1) restent pour la plupart en tête de liste. Des domaines qui avaient des scores de valeur faible semblent être davantage marginalisés. Par contre les domaines de scores intermédiaires ont des comportements plus divers. Certains se rapprochent des hauts scores, d'autres se sont rapprochés de la borne inférieure de l'intervalle de valeurs. La maîtrise de l'impact du seuil sur le score des domaines sera un paramètre déterminant dans la suite de notre étude.

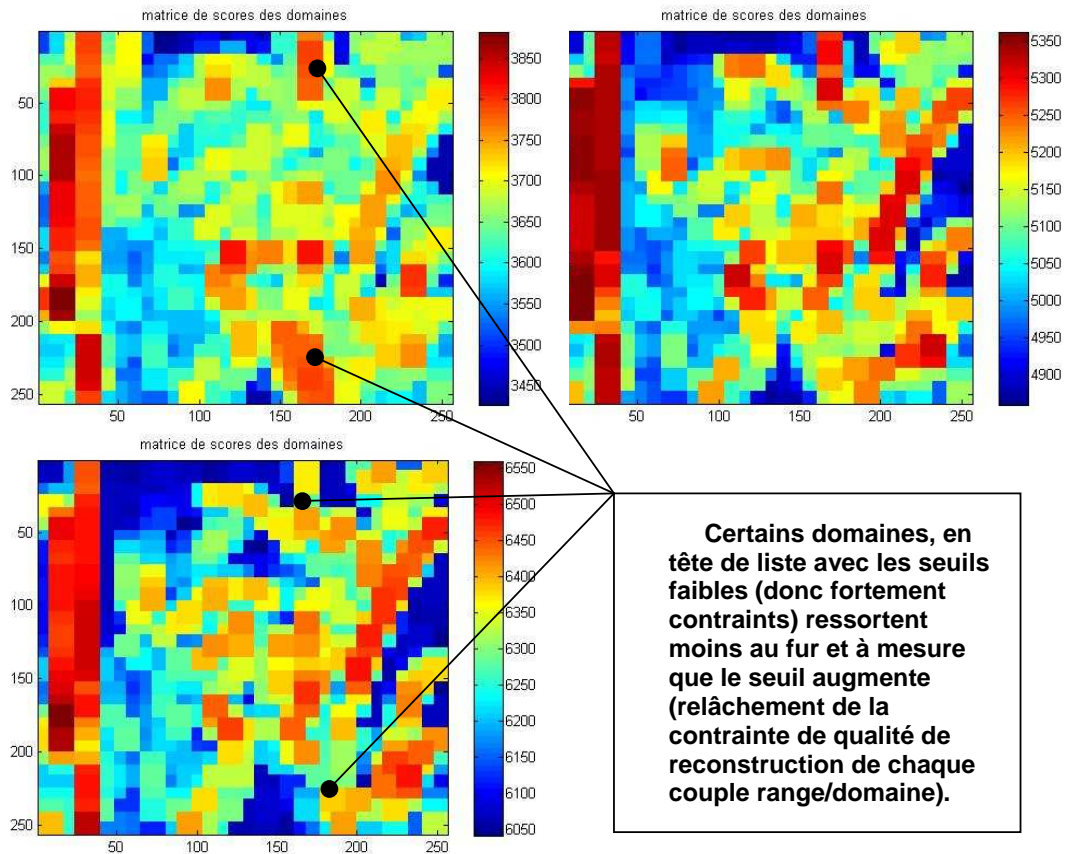


Figure 4-0 Cartes des scores des domaines pour des seuils de respectivement 1 (haut gauche), 2 (haut droite) et 3 (bas gauche) sur la mesure d'erreur.

La Figure 4-0 donne une représentation des scores pour les deux cartes calculées. Augmenter le seuil tend à aplatir l'histogramme des scores. Dans le cas d'un seuil supérieur ou égal au pire appariement range/domaine de l'image, chaque domaine marque un point pour tous les ranges, l'histogramme est alors parfaitement plat (et $score = nbRanges$).

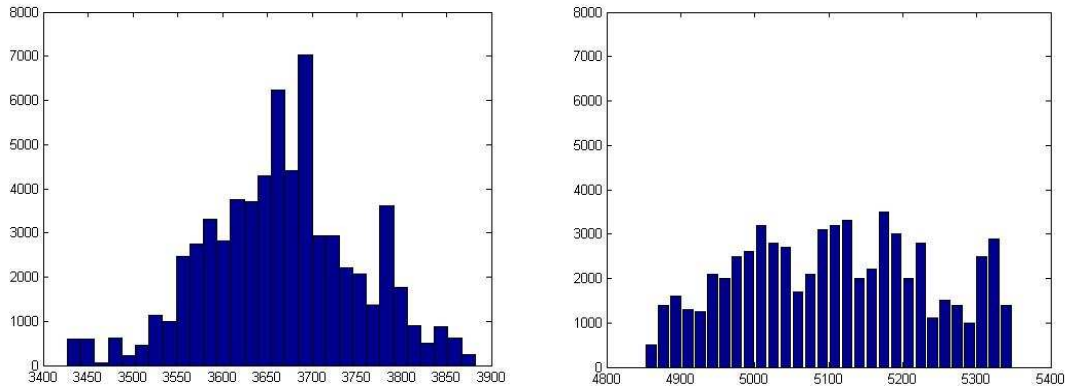


Figure 4-0 Histogramme des valeurs des scores pour les seuils 1 et 2.

Dans des tests antérieurs, non présentés dans ce document, un score global avait été testé où la notion de seuil était absente. Le score, pour chaque domaine, était la somme de toutes les erreurs de reconstruction de tous les ranges, en considérant pour chaque range la meilleure transformation possible. Malgré l'avantage certain de ne pas avoir de seuil, l'inconvénient principal est que nous faisons intervenir dans le score des ranges très différents du domaine considéré. Un domaine très bon pour reconstruire plusieurs ranges pouvait se retrouver pénalisé par la mauvaise reconstruction de quelques ranges. A l'inverse, un domaine produisant une reconstruction très moyenne d'un grand nombre de ranges pouvait se retrouver avec un score équivalent ou meilleur. Cette piste a rapidement été abandonnée.

Revenons à notre proposition utilisant un seuil. Son premier avantage est sa simplicité, tant d'un point de vue conception que sur la vision qu'elle nous permet d'avoir sur les données. Elle permet, de plus, d'avoir une première estimation de notre approche et d'étudier la stabilité d'un certain nombre de paramètres, comme nous l'avons vu en Figure 4-0 et Figure 4-0. Elle permet de sélectionner LE « meilleur domaine », relativement au seuil, partition, erreur de reconstruction, ...choisis. En revanche, cette méthode ne peut pas directement être appliquée à la sélection de plusieurs « meilleurs blocs ». En effet, pour le moment, des domaines quasi identiques vont avoir un score proche, comme c'est le cas pour des domaines spatialement très proches ou visuellement semblables. Si nous sélectionnons uniquement les domaines qui ont les scores les plus élevés, nous allons donc nous retrouver avec des groupes d'images similaires. Dans l'image de Lena, cela reviendrait à extraire plusieurs domaines ressemblant à l'épaule, le coin de miroir, une partie de la bordure à gauche de l'image, etc. Cette extraction ne serait pas aberrante en soi, nous aurions en quelque sorte des « clusters » d'images représentatives, mais dans l'absolu, particulièrement dans le cadre d'un apprentissage, nous ne souhaitons pas retenir les clusters eux-mêmes mais plutôt leurs « barycentres ». Dans l'état actuel de notre méthode de calcul de score, rien n'empêche deux domaines, visuellement identiques ou non, de marquer des points avec un même range, pour peu que le seuil le permette.

Nous discuterons plus loin de cette dépendance de reconstruction d'un même range par plusieurs domaines, et nous proposerons une méthode pour remédier à ce problème. Mais avant cela, comme nous sommes amené à souvent comparer des couples d'images et à décider si elles correspondent ou non relativement à un seuil donné, nous allons détailler dans la section suivante quelques opérateurs de mesures de distances ou erreurs de reconstruction entre images.

4.3.2 Méthodes de comparaison d'images

Nous venons de voir dans une première approche que nous pouvons mettre en relief des éléments représentatifs dans une image en nous basant sur une erreur de reconstruction entre des images et sur un seuil donné. Détaillons un peu cette mesure d'erreur. Tout d'abord un seuil n'a de sens que par rapport à la cohérence des données manipulées et surtout à la définition de l'erreur. Certaines erreurs mettront en relief principalement la ressemblance statistique, géométrique ou fréquentielle d'images. Il convient de connaître un ensemble suffisamment fourni de mesures de l'erreur de reconstruction, et pour chaque famille, savoir ce que l'erreur permet de quantifier, autrement dit, à quel aspect des données elle s'adresse (répartition spatiale, en niveaux de gris, richesse d'histogramme, etc.)

Ainsi, le couple {méthode de calcul de l'erreur, seuil sur l'erreur} nous permet de donner un sens à la question « est-ce que le matching entre ces deux images est bon ? ». Un problème central dans la comparaison d'images est la taille des données manipulées. Nous entendons ici par « images » des images de taille relativement petite, à savoir de 8x8 à 32x32 pixels. Des outils classiquement utilisés pour quantifier la ressemblance entre deux images sont le SSE (*Sum of Squared Error*) et le RMS (*Root Mean Square*) aussi appelé « valeur efficace » dont voici les définitions. Nous considérons des images carrées de côté N, les images contiennent N² pixels. La division de certaines distances par N² permet d'avoir une mesure homogène quand on veut comparer des couples d'images de tailles différentes.

$$SSE(I_1, I_2) = \sum_{i,j} [I_1(i, j) - I_2(i, j)]^2$$

$$RMS(I_1, I_2) = \sqrt{\frac{1}{N^2} \sum_{i,j} [I_1(i, j) - I_2(i, j)]^2} = \sqrt{\frac{1}{N^2} SSE(I_1, I_2)}$$

Une mesure très utilisée en compression d'image est le PSNR (*Peak Signal to Noise Ratio*) dont voici la définition :

$$PSNR(I_1, I_2) = 10 \log_{10} \left(\frac{d_{\max}^2}{EQM(I_1, I_2)} \right)$$

où

$$EQM(I_1, I_2) = \frac{1}{N^2} SSE(I_1, I_2) \text{ et } d_{\max} \text{ est la dynamique des images (i.e. 255).}$$

Dans [ROS06], Christophe Rosenberger présente un ensemble de mesures de comparaison d'images.

$$D1(I_1, I_2) = \frac{1}{N^2} \sum_{i,j=0}^{N-1} \|I_1(i, j) - I_2(i, j)\| \text{ également appelée } \textit{City Block Distance}, \text{ divisée}$$

par le nombre de pixels des images.

$$D2(I_1, I_2) = \frac{1}{N^2} \sum_{i,j=0}^{N-1} (I_1(i, j) - I_2(i, j))^2, \text{ une distance Euclidienne.}$$

$$D3(I_1, I_2) = \frac{1}{N^2} \sqrt[p]{\sum_{i,j=0}^{N-1} |I_1(i, j) - I_2(i, j)|^p}$$

D3 est une distance de Minkowski. Notons que pour $p=1$, nous retrouvons la distance *city block*, pour $p=2$ nous avons une distance Euclidienne.

$$D4,5(I_1, I_2) = \sqrt{\frac{1}{2(N-w)^2} \sum_{i,j=(w+1)/2}^{N-(w-1)/2} d(I_1, I_2, i, j)^2 + d(I_2, I_1, i, j)^2}$$

Ici $d(I_1, I_2, i, j)$ est une distance ou une pseudo distance (distance non symétrique) entre deux imageries I_1 et I_2 , relativement à un décalage (i, j) dans le voisinage w autour du centre des imageries. Elle est calculée pixel à pixel, dans un voisinage carré de côté 3 ou 5, autrement dit $w \in \{3, 5\}$, pour les distances $D4$ et $D5$ respectivement. Cette mesure est donc plus robuste à de petits décalages entre imageries similaires.

$$D6(I_1, I_2) = \frac{1}{R} \sum_{r=0}^R \sum_{i,j=0}^{N-1} \frac{(I_1^{(r)}(i, j) - I_2^{(r)}(i, j))^{1/2}}{N_r^2 \times 2^{2r+1}}$$

D6 représente une distance définie $R+1$ résolutions. $I^{(r)}$ représente l'image I à la résolution r .

$$C1(I_1, I_2) = \frac{\sum_{i,j=0}^{N-1} I_1(i, j)^2}{\sum_{i,j=0}^{N-1} I_2(i, j)^2}$$
 représente un rapport d'énergies entre I_1 et I_2 .

C2 représente une mesure de corrélation.

$$C2(I_1, I_2) = \frac{\sum_{i,j=0}^{N-1} I_1(i, j) I_2(i, j)}{\sum_{i,j=0}^{N-1} I_1(i, j)^2}$$

$$C3(I_1, I_2) = \frac{1}{N^2} \sum_{i,j=0}^{N-1} \left(1 - \frac{2 \cdot \min[I_1(i, j), I_2(i, j)]}{I_1(i, j) + I_2(i, j)} \right)$$

représente une erreur moyenne normalisée.

La comparaison par SSIM (*Structural SIMilarity*) [WAN04] permet d'avoir une comparaison d'images qui soit proche de la perception humaine. Dans cette méthode, I_1 et I_2 subissent un prétraitement. Elles sont filtrées par une gaussienne. La taille de cette dernière est définie selon la taille des images à comparer. La distance entre I_1 et I_2 est alors définie par

$$DSSIM(I_1, I_2) = \frac{2\mu_1\mu_2 + C1}{(\mu_2^2 + \mu_1^2 + C1)} \frac{2\rho + C2}{(\sigma_2^2 + \sigma_1^2 + C2)}$$

où ρ est l'inter corrélation entre I_1 et I_2 , c'est-à-dire :

$\rho(f(x), g(x)) = \int f(t).g(x+t)dt$. L'inter corrélation est également appelée *sliding dot product* (produit scalaire glissant), c'est l'équivalent d'une convolution mais sans inversion $(x-t)$.

$$C1 = (0.01 * 255)^2 = 6.50$$

$$C2 = (0.03 * 255)^2$$

μ_k et σ_k sont les moyennes et variances des niveaux de gris des images I_k

D'autres distances de comparaison d'image sont proposées dans [WAE07].

Dans le cas où l'imagette contient une texture naturelle, les méthodes de comparaison pixel à pixel perdent leur sens. L'œil retient essentiellement l'aspect statistique de l'imagette (granularité, contraste, symétrie, aspect périodique, etc.) et c'est alors ce type de mesures qu'il faudrait utiliser. En toute rigueur, pour savoir si les opérateurs que nous venons de définir sont applicables, lors de nos évaluations nous devons d'abord décider si une imagette est naturelle ou artificielle. Dans l'éventualité où nous avons à comparer une image naturelle et une imagette artificielle, le choix dépendra de l'application. Nous pouvons par exemple calculer les deux distances et choisir la plus pénalisante.

4.3.3 Prise en compte des dépendances ranges-domaines

Dans l'approche décrite en 4.3.1, nous avons autorisé qu'un même range puisse augmenter le score de plusieurs domaines. Des domaines visuellement proches auront alors des scores similaires. Donc si nous choisissons de stocker dans une base d'apprentissage les domaines qui ont les scores les plus élevés, nous risquons potentiellement d'avoir une redondance d'information (et ce n'est pas ce que nous souhaitons dans notre base d'apprentissage). De plus, si nous limitons le nombre d'imagettes à mémoriser, ces imagettes redondantes occuperont peut-être les places d'imagettes pertinentes.

Dans cette section, nous proposons donc une première solution à ce problème. Le principe est que, lorsqu'un range est codé par un domaine, il est considéré comme « *bloqué* », c'est-à-dire qu'il ne pourra plus donner de points à d'autres domaines, il ne sert qu'une seule fois. Nous remarquons alors que, pour un range donné, l'ordre de parcours des domaines est primordial. En effet, si le seuil sur l'erreur de reconstruction est relativement élevé, le premier domaine venu sera en mesure de coder le range et de le bloquer, prenant alors la main sur d'autres domaines de l'image qui auraient pu obtenir une reconstruction bien meilleure de ce range. De plus, l'ordre de parcours par défaut est le parcours ligne à ligne de l'image, les domaines situés en haut à gauche bénéficieraient alors d'un privilège qui serait uniquement basé sur leur position absolue dans l'image, ce qui n'est pas pertinent. Il faut donc définir un ordre de parcours cohérent par rapport à la nature des domaines. Cet ordre de parcours peut être défini par rapport à un (ou plusieurs) attribut(s) sur les imagettes considérées (ranges et domaines). Ces attributs doivent être indépendants de la taille de l'imagette à laquelle ils s'appliquent et des autres paramètres liés à la compression fractale (contraste, luminosité, etc.). Pour chaque range, nous parcourons les domaines par distance croissante aux attributs du range. Selon [HAS04], l'entropie d'un bloc image est une mesure sur laquelle nous pouvons nous appuyer pour caractériser la quantité d'information portée par ce bloc. Notre méthode consistera donc à parcourir les domaines par entropie croissante. A chaque fois que le

domaine considéré peut coder un range relativement à un seuil de reconstruction donné, ce range sera bloqué et non disponible pour les domaines suivants.

Cette méthode nous permet d'avoir une première approche sur la gestion de la dépendance entre les ranges et les domaines. Elle nous assure que nous n'allons potentiellement pas mémoriser des domaines voisins, recouvrants et/ou identiques. En revanche, son côté « straight-forward » (blocage d'un range par le premier domaine pouvant le coder) empêche une vision globale des reconstructions possibles des couples range/domaine. La sensibilité à la stratégie de parcours et au seuil choisis est donc importante.

Une solution serait de connaître pour chaque couple range/domaine (ou le plus grand nombre de couples possibles) l'erreur de reconstruction en considérant la meilleure transformation liant chaque couple. Cette vue plus globale des blocs et des transformations qui les lient permettrait un choix de seuil plus approprié. La section suivante définit plus précisément l'erreur de reconstruction liant les couples range/domaine au travers d'une matrice de reconstruction de dimension $nbDomains \times nbRanges$.

4.3.3.1 Matrice d'erreur de reconstruction

Nous venons d'étudier une méthode pour traiter le problème de la dépendance dans le calcul du score des domaines. Mais cette méthode repose encore sur un seuil unique qui détermine si un domaine code ou non un range. Le problème dans le choix du seuil est que pour un concept dont nous voulons extraire les imagerie caractéristiques, l'intervalle $[min, max]$ de l'erreur de reconstruction est très variable. Il dépend de la mesure d'erreur choisie et de la nature des imagerie présentes dans l'image. Par exemple un RMS aura un grand débattement de valeur pour une image comportant des zones géométriques et aura des valeurs moins variables pour des textures naturelles type feuillage. Il paraît alors difficile d'établir un seuil dans l'absolu. Une estimation globale de l'erreur est nécessaire pour espérer faire un choix de seuil approprié. Pour illustrer le problème et mieux comprendre la solution proposée pour résoudre ce type de problème nous allons visualiser les données mises en jeu.

Pour cela nous allons calculer une matrice d'erreur de reconstruction ranges/domaines. Elle permettra entre autres, de calculer rapidement le score de chaque domaine relativement à un seuil donné. Cette matrice nous donnera une vue globale de la « compressibilité » d'une image donc de la facilité qu'il y a à apprendre les caractères spécifiques. Par exemple, si pour un nombre restreint de domaines, nous constatons une erreur faible pour l'ensemble des ranges, cela signifie que l'image est codée fidèlement uniquement avec ces quelques domaines. Cette matrice d'erreur de reconstruction est de dimension $nbDomains \times nbRanges$ et chaque élément contient l'erreur de reconstruction liant un couple $[range, domaine]$. Une matrice complètement renseignée correspond donc à un parcours exhaustif des couples.

Une seconde étape à réaliser avant d'élaborer une stratégie de seuillage cohérente est de savoir à quel type de données nous avons affaire. Une méthode pour observer la répartition de l'erreur de reconstruction globale de l'image est d'afficher la matrice d'erreur en triant les valeurs suivant les domaines, les ranges, et les deux, à savoir par range puis par pire domaine (domaine ayant obtenu la plus grande erreur de reconstruction pour le range considéré). La Figure 4-0 représente la matrice d'erreur de reconstruction de Lena. La matrice triée selon des domaines, Figure 4-0-b, nous permet d'observer pour chaque range, quelle est l'évolution de son erreur de reconstruction en allant « de son pire à son meilleur domaine ». Un range peut être difficile à coder par la majorité des domaines, et bien reconstruit avec très peu d'entre eux. D'autres seront faciles à coder par un grand nombre de domaines. La matrice triée selon les ranges, Figure 4-0-c, nous indique, pour chaque domaine, selon le seuil qu'on choisira, le nombre potentiel de ranges qu'il pourrait

coder. Enfin la matrice triée selon les ranges et pires domaines, Figure 4-0-d, nous donne un aperçu général de la surface codable en fonction d'un seuil donné. Si nous voulions coder toute l'image, c'est-à-dire que chaque range ait au moins un domaine qui le code, alors nous devrions choisir un seuil vers 4 (bleu ciel sur l'échelle de couleur des erreurs). Ce seuil suffisamment relaxé permet que les ranges localisés sur la gauche de cette matrice aient au moins un domaine correspondant. Nous pouvons cependant observer que la majeure partie des ranges ont une erreur bien plus faible et que, moyennant un seuil de 1 (bleu foncé), nous pourrions coder environ 80% de l'image. Ce qui correspondrait mieux à un juste compromis entre qualité de sélection des blocs (c'est-à-dire un seuil suffisamment élevé pour que tous les domaines ne marquent pas des points avec tous les ranges), et la quantité de ranges codés.

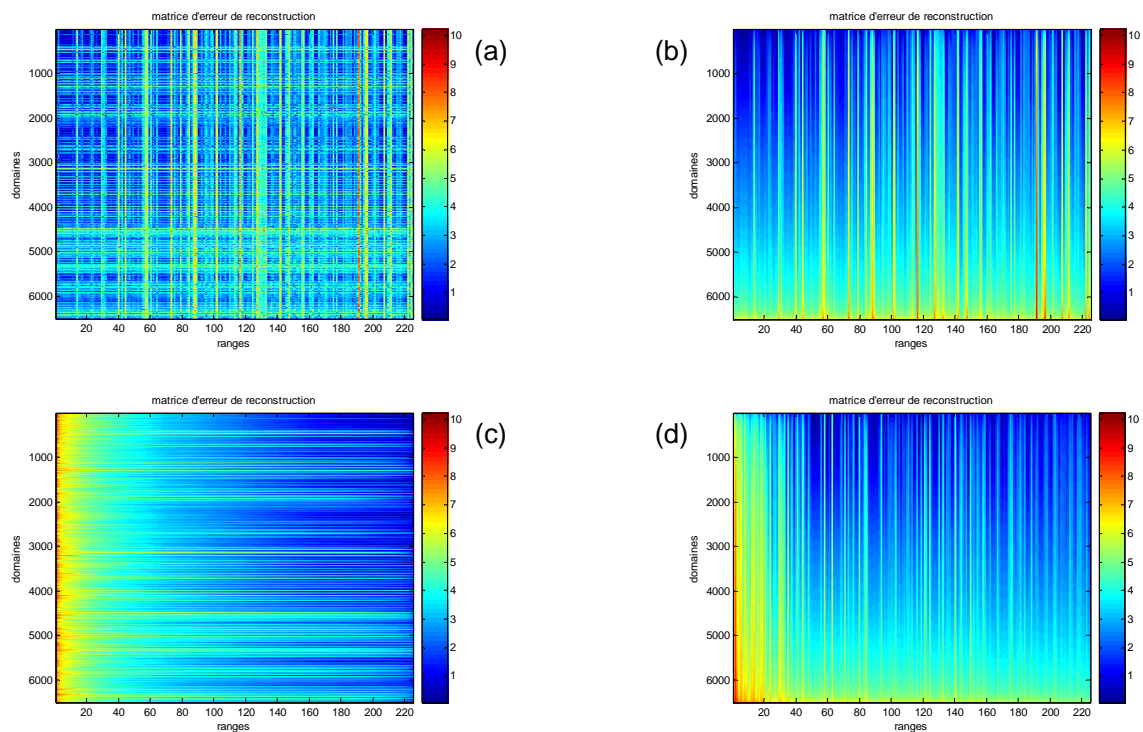


Figure 4-0 Matrice d'erreur de reconstruction de Lena (haut gauche), triée par ranges-colonne (haut droite), triée par domaines-ligne (bas gauche), triée par ranges puis par pires domaines (bas droite). L'axe vertical représente les domaines (un domaine par ligne), l'axe horizontal les ranges (un range par colonne).

Dans la matrice d'erreur en Figure 4-0-b, nous constatons que certains ranges semblent particulièrement difficiles à coder. Autrement dit, pour un grand nombre de domaines, ils ont une erreur de reconstruction élevée. Nous reconnaissons ces ranges par les grandes colonnes oranges, voire rouges dont ils sont à l'origine, en Figure 4-0-b. Les quatre « pires ranges », par erreur décroissante, sont les ranges numéros 191, 116, 73, 88. La Figure 4-0 permet de localiser ces ranges sur l'image d'origine. Le fait qu'ils ressortent autant par cette méthode de visualisation ne signifie pas qu'ils ne soient pas codables avec une bonne qualité, mais simplement que si des domaines peuvent les coder avec un seuil bas, alors ces domaines ne sont pas nombreux.

Observons les erreurs de reconstruction des domaines les plus adaptés aux ranges 191,116, 73, 88. La Figure 4-0 représente l'évolution de l'erreur de reconstruction, les erreurs ont été triées par ordre croissant, des domaines les plus adaptés à ces ranges. Le range 191, dont l'erreur est la plus élevée, peut être codé par un domaine avec une très bonne qualité, à savoir un seuil inférieur à 1,3. A l'inverse le range 116, localisé à la jointure d'un oeil et d'une mèche de cheveux n'est codable qu'à partir d'un seuil de 3,5. Cela signifie que, relativement à l'erreur choisie, un seuil inférieur à 3,5 et les transformations considérées, ce range ne serait jamais similaire à aucun autre. Autrement dit il est « unique ». Nous verrons plus loin comment ces éléments uniques seront gérés. Ils pourront être soit considérés comme non représentatifs car non redondants, soit, au contraire, importants car discriminants.

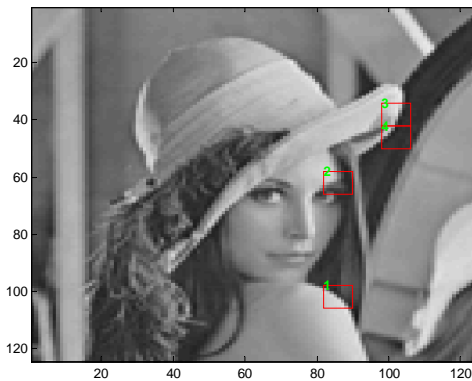


Figure 4-0 Localisation des ranges d'erreur moyenne la plus élevée sur l'ensemble de leurs domaines. Il s'agit des ranges 191,116, 73, 88, dont l'erreur triée est visible en Figure 4-0-b.

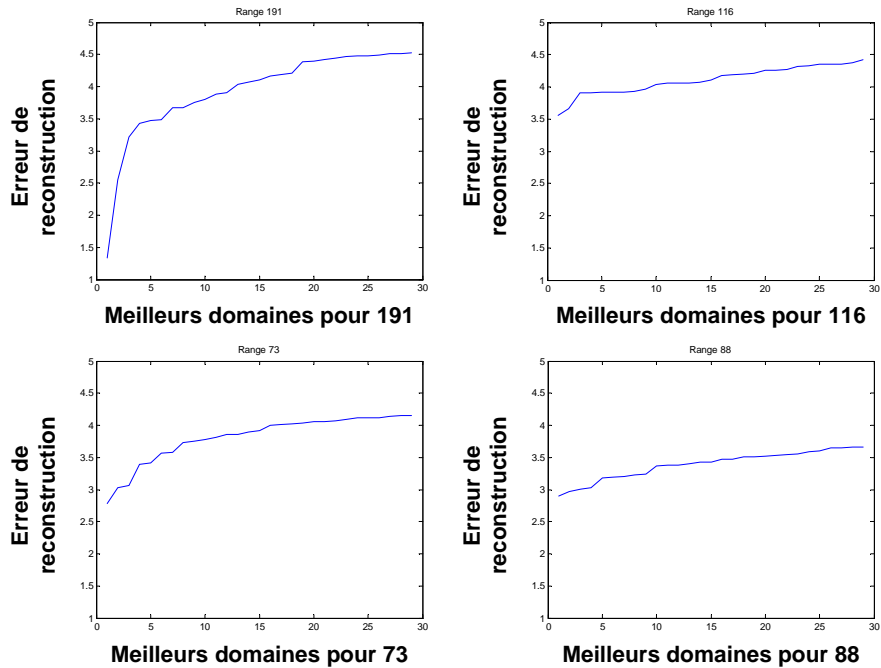


Figure 4-0 Erreurs de reconstruction, triées par ordre croissant, des meilleurs domaines pour la reconstruction des ranges 191,116, 73, 88.

4.3.4 Apprentissage par seuil adaptatif

Jusqu'à présent, nous avons vu des méthodes utilisant un seuil fixé a priori, choisi par l'utilisateur d'après la matrice d'erreur de reconstruction. Nous savons qu'un seuil élevé va diminuer la représentativité de chaque domaine puisqu'un grand nombre d'entre eux pourront marquer des points pour la majeure partie des ranges disponibles. Au contraire, un seuil bas va privilégier la qualité de reconstruction des ranges, mais ne permettra probablement pas de tous les reconstruire. Une partie de l'image sera alors laissée non traitée. Nous allons proposer une méthode permettant de faire un compromis entre la quantité des ranges codés et la qualité de reconstruction de chacun. Pour ce faire, nous allons relâcher la contrainte de qualité au fur et à mesure que les ranges seront codés, jusqu'à atteindre un pourcentage suffisant de ranges codés. Or, comme les ranges correspondent à une partition de l'image, coder un pourcentage donné de ranges est équivalent à coder ce même pourcentage de la surface totale de l'image d'origine. Nous avons donc remplacé le seuil sur l'erreur par un seuil sur la surface de l'image à coder. La notion de seuil peut même totalement disparaître si on considère que toute l'image doit être apprise et donc la surface à apprendre vaut 100%.

Nous avons donc une répartition de l'erreur de reconstruction sur l'ensemble des couples range-domaine. Pour le seuil adaptatif, nous allons établir une liste des valeurs qu'il prendra au fil de l'extraction des éléments représentatifs. Le but de ces seuils est de traiter les domaines à sélectionner par groupes homogènes, d'un point de vue qualité de reconstruction. Les deux questions concernant cette liste sont, d'une part, comment répartir les seuils dans l'intervalle couvert par l'ensemble des erreurs de reconstruction, d'autre part, une fois la liste établie, quand passer d'un seuil à son successeur. Autrement dit à quel moment, dans la boucle de sélection itérative, allons nous relâcher la contrainte de qualité pour coder une plus grande quantité de ranges.

4.3.4.1 Seuil à pas constant et progression du seuil

Une première approche simple est de choisir des valeurs de seuil équiréparties, à pas constant, sur l'intervalle [min max] des valeurs prises par l'erreur de reconstruction des couples range/domaine. Nous devons fournir en entrée le nombre de seuils à retenir. Dans le cas de l'image Lena, l'erreur de reconstruction va de 0,1 à 10,3. Une répartition régulière sur 5 niveaux nous donne donc la liste de seuils suivants : {0,11; 2,66; 4,21; 7,75; 10,30}.

Une fois cette liste définie, nous devons établir quand le seuil va changer de valeur, autrement dit, quelle est sa stratégie de progression. La méthode la plus simple est de faire évoluer le seuil à chaque passage dans la boucle. Nous appellerons cette méthode « seuil forcé ». Une autre possibilité est de choisir pour le seuil, l'erreur minimale présente dans la matrice de reconstruction d'un couple de range/domaine qui n'a pas encore servi. Le souci avec cette approche, telle qu'elle est présentée, est que nous aurons souvent une valeur unique d'erreur de reconstruction pour chaque couple range/domaine. Nous pourrions obtenir plusieurs valeurs très proches, mais pas strictement égales, il n'y aurait alors qu'un seul couple dans la compétition à chaque étape, ce qui aurait peu d'intérêt. Pour pouvoir appliquer efficacement cette seconde approche, il convient de regrouper (*clusteriser*) les valeurs de la matrice de reconstruction. C'est-à-dire que nous allons affecter à chaque élément de la matrice la valeur la plus proche de la liste de seuils. Dans l'exemple de l'image de Lena avec des valeurs d'erreurs comprises dans l'intervalle [0,1; 10,3], nous transformons la matrice pour que ses valeurs appartiennent à {0,11; 2,66; 4,21; 7,75; 10,30} (la méthode de transformation de la matrice est donnée plus bas). De cette manière, nous créons des « sac de couples » qui pourront être compétitifs entre eux lors de la sélection, et la sélection du seuil par valeur minimale prend alors un sens. La Figure 4-0 donne l'algorithme de sélection par seuils équirépartis. La différence de progression du seuil se situe dans le choix de la fonction *avancerSeuil(liste_seuils)*.

```
Entrées : liste_seuils, M : la matrice d'erreurs  
Sorties : liste de domaines représentatifs  
  
tabRangesCodes ← false(nbRanges)  
listeDomainesRepr ← ensemble vide  
Tant que % de surface non couverte  
    Seuil = avancerSeuil(liste_seuils)  
    M_bin ← (M < seuil)  
    scoresDomaines = sum(m_bin, 2);  
    (somme uniquement sur les ranges encore libres)  
    [scoreMax, iScoreMax] = max(scoresDomaines);  
    ajouter(listeDomainesRepr, iScoreMax)  
    mettre à jour tabRangesCodes (ceux de iScoreMax)  
    en déduire le % de surface codée  
Fin tant que
```

Figure 4-0 Algorithme de sélection de range par seuils équirépartis.

Un regroupement des valeurs de la matrice se fait selon la formule suivante :

$$M_{clust} = \min(M) + (\max(M) - \min(M)) \cdot \left\lfloor \frac{M - \min(M)}{\max(M) - \min(M)} \cdot (n_{clust} - 1) \right\rfloor / (n_{clust} - 1)$$

où n_{clust} est le nombre de clusters désirés

M et M_{clust} sont respectivement la matrice d'erreur de reconstruction et sa version clusterisée.
 $\lfloor . \rfloor$ désigne la partie entière.

Considérons les deux stratégies d'évolution du seuil.

Dans la stratégie de sélection du seuil par minimum de la matrice de reconstruction (clusterisée), le seuil n'évolue pas tant qu'il reste au moins un couple range/domaine dans l'intervalle $[liste_seuils(i), liste_seuils(i+1)]$ considéré. Comme nous commençons la sélection par les seuils les plus contraints en terme de qualité de reconstruction, cela veut dire que nous allons mémoriser en moyenne plus de blocs qu'avec la méthode par seuil forcé et que l'aspect qualité de reconstruction sera plus présent.

La méthode par seuil forcé est la plus compétitive dans la mesure où il n'y aura, la plupart du temps, qu'un seul bloc élu à chaque itération. Un couple range/domaine dont l'erreur de reconstruction se trouve dans l'intervalle $[liste_seuils(i), liste_seuils(i+1)]$ pourra participer aux sélections correspondant aux intervalles $[liste_seuils(i+1), liste_seuils(i+2)]$ et suivants, mais il sera en compétition avec un plus grand nombre de participants. Il n'est pas exclu de la course mais sa probabilité de sélection diminue.

Remarque sur le calcul de la surface codée : Le pourcentage de surface codée par un domaine peut être calculé de différentes manières. La plus simple est de compter le nombre de ranges codés. En effet, dans le cas classiquement adopté où les ranges constituent une partition de l'image, le rapport entre le nombre de ranges codés et le nombre total de ranges donne directement le pourcentage de la surface codée. Nous pouvons également prendre certaines libertés sur la répartition des ranges. Nous pouvons par exemple autoriser leur recouvrement partiel en faisant une « seconde partition » de l'image, décalée d'un demi range en hauteur et largeur. Ceci permet d'avoir une plus grande souplesse dans la sélection des éléments redondants de l'image, dans la mesure où chaque domaine pourra augmenter son score parmi un plus grand nombre de ranges disponibles. Dans ce cas, le pourcentage de surface image codée sera alors calculé au niveau pixel, entre ceux appartenant aux ranges codés lors des itérations précédentes et ceux correspondant aux ranges codés par le domaine actuel.

Nous avons déjà vu quelques approches différentes pour extraire des blocs représentatifs d'une image. Certaines sont basées sur un seuil fixé, d'autres sur un seuil dynamique calculé en fonction de l'intervalle occupé par l'erreur de reconstruction. Nous voulons être capable de comparer ces méthodes et pouvoir quantifier leur efficacité. La section suivante propose de remplir cette tâche au travers d'une reconstruction de l'image par les domaines sélectionnés.

4.3.4.2 Comparaison des méthodes d'extractions par reconstruction de l'image

Pour évaluer les différentes approches de sélection, nous allons utiliser une reconstruction de l'image d'origine. La compression fractale utilise la totalité des domaines disponibles dans l'image et ne stocke en mémoire que des transformations entre blocs, alors que nous proposons à l'inverse, de ne stocker aucune transformation

mais uniquement une quantité limitée de blocs image issus du processus d'apprentissage que nous avons proposé. Le but futur est de pouvoir utiliser ces blocs image dans une phase de reconnaissance d'objets, et non une reconstruction exacte d'une image particulière. La mesure d'évaluation que nous allons développer dans ce paragraphe ne sera pas là pour situer la qualité de la reconstruction par rapport aux méthodes de compression existantes, mais plutôt pour comparer nos solutions entre elles, et quantifier la pertinence de la représentativité des imagerie extraites.

Pour reconstruire une image à partir d'un ensemble d'imagerie, nous faisons correspondre les ranges de l'image d'origine avec les imagerie en question, considérant la meilleure transformation parmi celles utilisées lors de la compression. Le résultat de l'imagerie transformée est alors plaqué sur l'image à reconstruire à la position du range en question. Le principe se rapproche en ce sens de la méthode de décompression, à ceci près que nous n'itérons pas et que nous nous limitons à un nombre restreint d'imagerie (nous rappelons que dans la compression fractale standard, nous pouvons utiliser tous les domaines de l'image pour coder les ranges, ce qui n'est pas le cas ici). Les reconstructions peuvent être sensiblement améliorées en effectuant une seconde passe sur l'image à reconstruire. Pour chaque imagerie extraite de l'apprentissage, nous effectuons une intercorrélacion normalisée entre l'image d'origine et l'imagerie en question. Les positions correspondant à des corrélacions normalisées de valeur absolue supérieures à un seuil donné (classiquement 0,8) sont testées de la même manière que les positions des ranges. Cette seconde étape est optionnelle. Dans nos tests, nous avons observé un gain de quelques dixièmes sur le PSNR des images reconstruites par rapport à l'étape principale seule décrite plus haut.

Le Tableau 4-0 illustre l'évolution du nombre de domaines extraits et la qualité de l'image reconstruite en fonction du nombre de clusters choisi. Dans le cas où nous regroupons les valeurs de la matrice d'erreur de reconstruction de Lena sur 5 clusters (générés par un découpage équiréparti de l'intervalle de l'erreur), nous obtenons 4 domaines, un quasi homogène, deux contours obliques et une séparation verticale. Le PSNR résultant de l'image reconstruite est de 25. Si nous passons de 5 à 6 clusters, nous obtenons alors 12 domaines (toujours en considérant la stratégie d'évolution du seuil par seuil min), plus diversifiés. Le PSNR obtenu est alors de 27. La Figure 4-0 montre les domaines obtenus dans ces deux cas.

% de surface imposé	Nb de clusters	Type de progression du seuil	Nb de domaines obtenus	Qualité de reconstruction obtenue (psnr)
80%	5	par seuil min	4	25
80%	6	par seuil min	12	27

Tableau 4-0 Tableau Comparatif du nombre de domaines obtenus et de la qualité de reconstruction obtenue en fonction du nombre de clusters choisi.

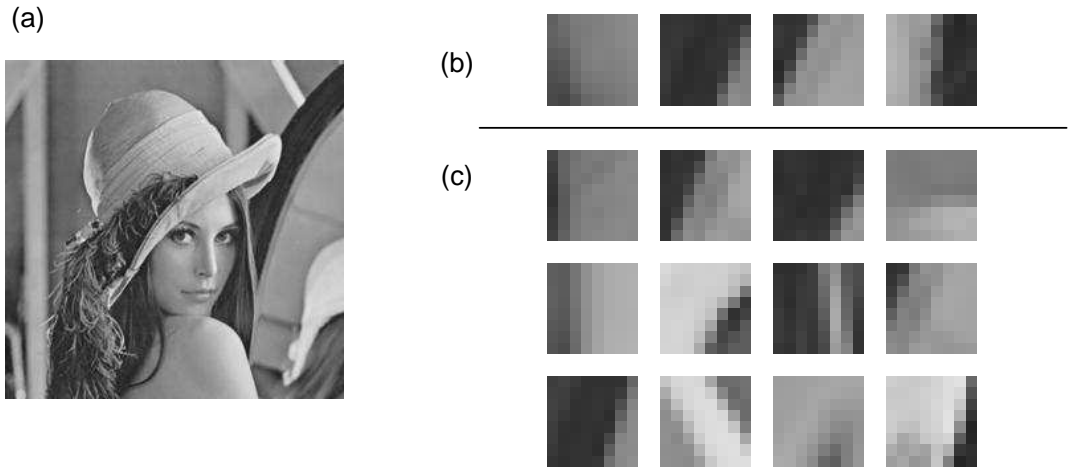


Figure 4-0 Image initiale (a). Ensemble de 4 et 12 domaines extraits respectivement à partir de 5 et 6 clusters (respectivement (b) et (c)).

Nous remarquons que nous n'obtenons pas nécessairement autant de blocs que de clusters. Ceci est dû à la contrainte de surface à apprendre, il est possible d'atteindre le pourcentage de surface requis sans pour autant atteindre le dernier cluster, qui regroupe les couples de range/domaine ayant une forte erreur de reconstruction.

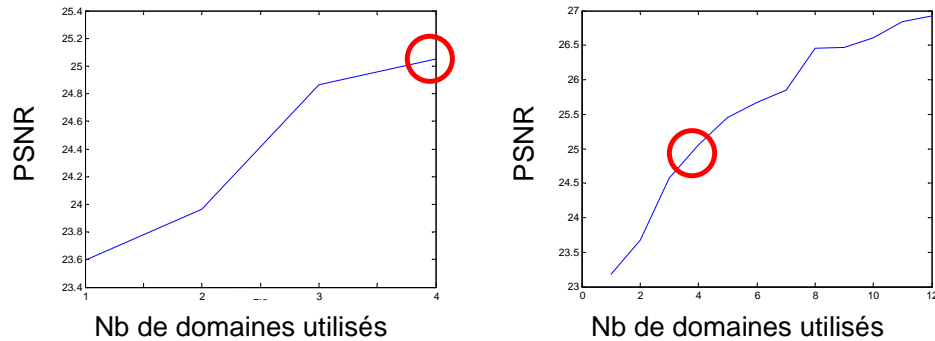


Figure 4-0 Evolution du PSNR de l'image reconstruite au fur et à mesure de l'utilisation des domaines appris. A gauche avec 5 domaines, à droite avec 12 domaines. En rouge, nous notons une même qualité de reconstruction (PSNR=25) pour un même nombre de domaines utilisés (4).

Dans l'exemple précédent nous avons extrait 4 et 12 domaines en utilisant respectivement 5 et 6 clusters pour la matrice de reconstruction. Nous rappelons que l'ordre dans lequel ces domaines ont été extraits est important. Il correspond au relâchement de la contrainte de qualité locale de reconstruction (dans le but d'augmenter la surface codée de l'image). La Figure 4-0 illustre le PSNR de l'image reconstruite en utilisant 1, puis 2, puis n domaines, en respectant leur ordre d'extraction. La courbe de gauche correspond à la reconstruction de l'image avec 4 domaines (5 clusters) et la courbe de droite la reconstruction avec 12 domaines (6 clusters). Les cercles rouges représentent le PSNR pour des reconstructions avec 4 domaines. Pour 4 domaines utilisés, nous parvenons dans les deux reconstructions à un PSNR de 25. Nous pouvons

remarquer sur la Figure 4-0 que les 4 premiers domaines de la liste de 12 domaines, issus de la première extraction, sont visuellement différents des 4 issus de la première extraction. Pourtant en terme de quantité d'information, si nous nous fions au PSNR, nous obtenons des PSNR comparables en chaque point (c'est-à-dire aux abscisses 1 à 4). Ceci nous conforte dans la faculté de notre méthode à extraire la même « quantité d'information » pour un même nombre de domaines.

Nous pouvons également observer la variation du PSNR lors de l'introduction d'un nouveau domaine. Une pente importante entre les introductions des domaines D_i, D_{i+1} est représentative d'un apport important d'information du domaine D_{i+1} par rapport à son prédécesseur. Sur l'exemple présenté, notamment sur la courbe de droite, plus un domaine se trouve en queue de liste (domaine 5 à 12), moins la pente est importante donc plus l'information ajoutée est mineure. Ceci nous conforte dans le fait que notre méthode concentre l'essentiel de l'information sur les premiers domaines extraits et complète l'apprentissage avec d'autres domaines en relâchant la contrainte de qualité, jusqu'à satisfaction de la contrainte de la surface couverte.

4.3.4.3 Seuil par histogramme cumulé

Dans l'approche avec seuil dynamique, nous n'avons utilisé jusqu'à présent que la méthode avec équirépartition sur l'intervalle de l'erreur de reconstruction. Un point faible de cette méthode est qu'elle ne tient pas compte de la répartition de l'erreur sur l'ensemble des couples range/domaine. Selon la nature de l'image, il peut arriver que certains intervalles $[listeSeuils_i, listeSeuils_{i+1}]$ contiennent bien moins de couples que d'autres. La Figure 4-0 illustre une répartition typique de l'erreur de reconstruction des couples range/domaine d'une image. Les couples d'erreur faible sont généralement minoritaires et permettent une reconstruction locale de l'image mais avec une très bonne qualité. A l'inverse, des domaines avec une erreur élevée sont plus présents. Ils représentent soit des couples dont les éléments n'ont rien à voir ensemble, dont le range peut être codé par un autre domaine, soit des éléments singuliers, c'est-à-dire qui ne sont similaires à aucun autre dans l'image.

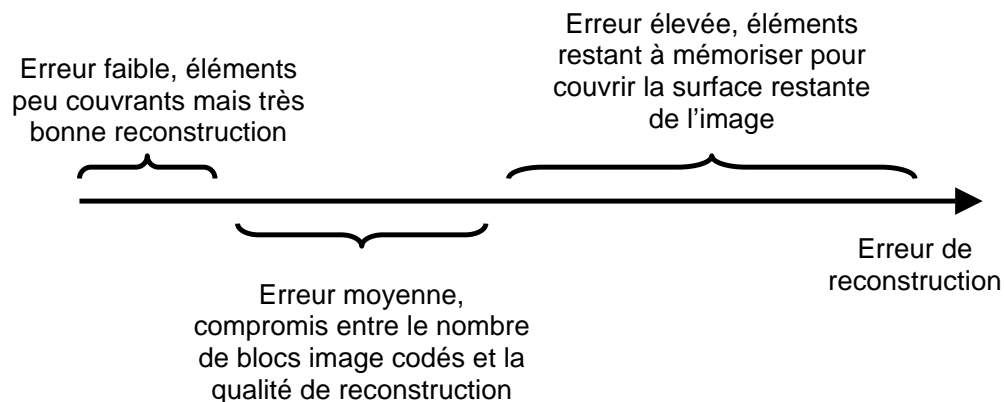


Figure 4-0 Illustration de la répartition typique de l'erreur de reconstruction sur l'ensemble des couples range/domaine.

Nous allons maintenant affiner la méthode vue dans la section précédente. L'atout de cette nouvelle approche est qu'elle tient compte de la répartition des valeurs de l'erreur

dans l'intervalle [min max], et non plus uniquement des limites de l'intervalle (nous rappelons que la liste de seuils était calculée en répartissant ces derniers de manière équirépartie entre les bornes [min max] de l'intervalle). Nous calculons l'histogramme cumulé des valeurs de l'erreur de reconstruction. Pour cela, il convient de définir les classes qui seront utilisées pour calculer l'histogramme en question. Le choix du nombre de classes de l'histogramme sera fixé à une valeur suffisamment petite par rapport au nombre de couples range/domaine, et suffisamment grande par rapport au nombre de clusters que nous souhaitons imposer sur les valeurs de la matrice de reconstruction. Nous avons généralement utilisé des histogrammes sur 100 à 200 classes (le nombre de classes ne change pas le résultat de manière significative). Un parcours de cet histogramme cumulé permet de fixer les seuils correspondant à des ensembles de couples de cardinal quasi constant.

Le choix du nombre de clusters doit être fixé a priori. Ce paramètre détermine la concision de l'information que l'on souhaite obtenir en sortie. Dans le cadre de la stratégie de parcours « seuil forcé », le nombre de domaines extraits est la plupart du temps égal ou proche du nombre de clusters. Dans la stratégie de « parcours par seuil min », le nombre de domaines sera supérieur au nombre de clusters. Dans tous les cas, plus le nombre de clusters est important, plus la reconstruction est fine et moins l'information est concise.

La Figure 4-0 illustre la répartition des groupes de couples ranges/domaines pour une répartition en 8 clusters, les seuils (séparant les clusters) sont représentés par les lignes verticales rouges. Nous constatons un resserrement des clusters à l'endroit où l'histogramme a la plus forte densité/pente sur l'histogramme cumulé. En effet, beaucoup de couples range/domaine possèdent une valeur de reconstruction entre 2 et 4,5 (RMS). De ce fait, un léger déplacement d'un seuil sur l'erreur de reconstruction dans l'intervalle [2 4,5] peut entraîner une importante variabilité dans le nombre de couples compris dans les clusters à gauche et à droite de ce seuil.

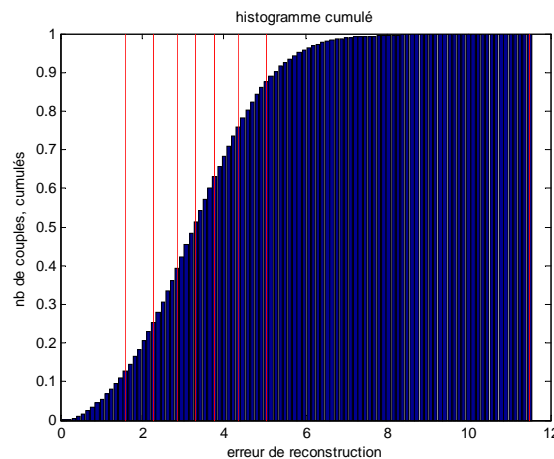


Figure 4-0 Histogramme cumulé de l'erreur de reconstruction de couples ranges/domaines de Lena, avec 8 clusters. Les clusters contiennent un nombre de couples quasi égal et les regroupent par erreur de reconstruction proche. Il y a donc une plus grande densité de clusters dans les parties où l'histogramme cumulé a une forte pente.

La Figure 4-0 montre l'évolution du PSNR de l'image reconstruite (Lena) en fonction du nombre de domaines extraits utilisés, en considérant respectivement une extraction de domaines par clusters équirépartis selon la valeur de l'erreur (courbe bleue), et selon le nombre d'individus dans chaque cluster (méthode par histogramme cumulé, courbe

verte). Si nous observons le PSNR de la reconstruction finale, la méthode par histogramme cumulé donne un résultat meilleur que la méthode à seuils équirépartis de l'erreur sur l'ensemble des domaines (donnés en abscisses). Ce gain de performance est particulièrement notable dans la partie gauche de la courbe. Sur le premier domaine, l'extraction par histogramme cumulé affiche une reconstruction de PSNR 27,6 contre un PSNR de 26 pour l'extraction par seuils équirépartis. L'aspect « synthétique » de l'extraction par histogramme cumulé est plus efficace. Enfin nous pouvons remarquer que les deux courbes convergent vers un PSNR commun proche de 30,3. Nous interprétons cette valeur comme la qualité de reconstruction limite atteignable relativement à la configuration choisie à l'origine (finesse du partitionnement, nombre et complexité des transformations, etc.).

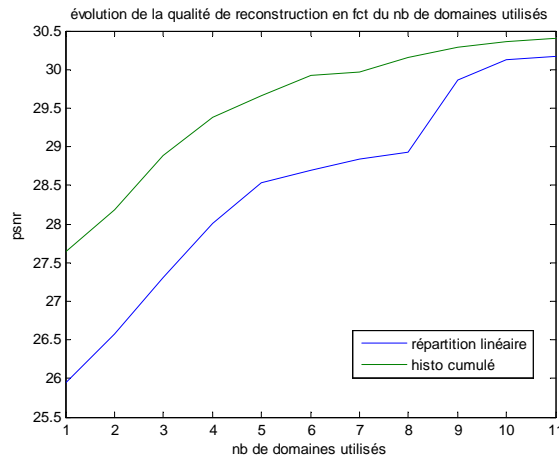


Figure 4-0 Evolution du PSNR au fur et à mesure qu'on prend en compte les domaines appris, extraits respectivement d'une équirépartition de l'erreur (courbe bleue), ou d'une répartition suivant l'histogramme cumulé de l'erreur des couples ranges/domaines (courbe verte).

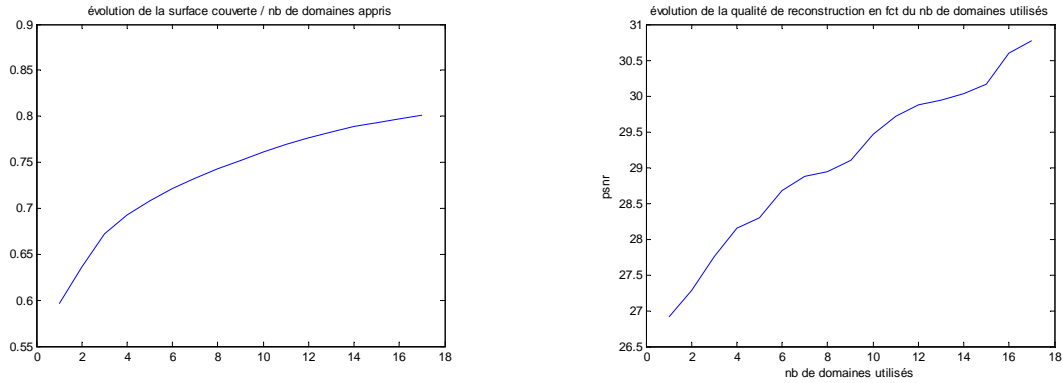


Figure 4-0 A droite, extraction de 12 domaines avec 10 clusters sur l'image Lena. A gauche, pourcentage cumulé de surface couverte lors de l'apprentissage par les différents domaines. Nous arrivons au 80% choisis en entrée d'apprentissage.

La Figure 4-0 illustre l'évolution du PSNR de l'image reconstruite (graphe de droite) ainsi que la surface apprise (graphe de gauche) au fur et à mesure que les domaines appris sont utilisés pour la reconstruction, dans leur ordre d'extraction.

Si nous effectuons une extraction sur 10 clusters avec la méthode de l'histogramme cumulé, nous obtenons 17 domaines (Figure 4-0). La reconstruction de l'image aboutit à un résultat à peine meilleur (PSNR=30.7) que celle avec 11 clusters vue auparavant. Ceci étaye l'hypothèse d'une reconstruction seuillée avec les conditions de test choisies.

Sur peu de clusters, 3, un seul domaine est extrait, avec une stratégie de parcours « par seuil min ». La qualité de reconstruction est alors mesurée par un PSNR égal à 26. Le PSNR passe à 26,9 si nous autorisons l'usage d'un domaine homogène ajouté artificiellement ou si nous tolérons un paramètre de contraste proche de zéro. Cet ajout d'un domaine artificiel est généralisable à l'ensemble de notre méthode, mais son apport (limité, +0,5 sur le PSNR) se ressent essentiellement quand le nombre de domaines extraits est particulièrement faible. Les transformations disponibles appliquées à ce domaine permettent de retrouver un certain nombre de bords (chapeau, miroir, visage, etc.). Concernant la qualité de l'image, un PSNR de 26 représente une qualité pauvre dans l'absolu. Cependant cette qualité de reconstruction est intéressante au regard du peu d'information que nous avons extrait, à savoir un domaine unique.

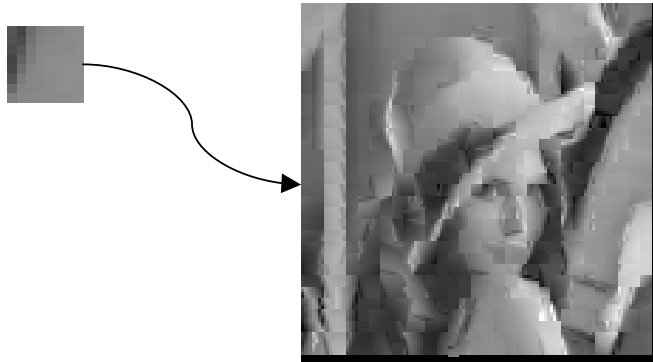


Figure 4-0 Reconstruction de Lena avec un seul domaine extrait par histogramme cumulé de l'erreur de reconstruction.

La courbe de gauche de la Figure 4-0 illustre l'évolution de la surface couverte en fonction du nombre de domaines extraits utilisés. Nous rappelons que les domaines extraits sont utilisés dans l'ordre de leur extraction. La courbe de droite de cette même figure illustre la qualité de reconstruction de l'image au fur et à mesure que nous prenons en compte de plus en plus de domaines. Ces deux courbes nous informent sur la *compressibilité* de l'image.

Nous disons qu'une image est fortement compressible si elle peut être représentée avec une bonne qualité, moyennant peu d'information stockée. Pour lier cette notion à la lecture de ces deux courbes, nous pouvons affirmer qu'une image est fortement compressible si la majeure partie de sa surface est restructurable avec peu de domaines et moyennant un PSNR supérieur à une valeur donnée. La Figure 4-0 illustre ce phénomène en présentant une forme générique de meilleure et pire courbe en terme de compressibilité de l'image. Une image comportant un nombre de similarités basées sur peu de domaines aura une courbe tendant vers 100% de surface couverte avec une erreur de reconstruction faible. A l'inverse, une image comportant énormément d'éléments singuliers sera associée à une courbe qui sera bien plus lente à atteindre un pourcentage élevé de surface couverte.

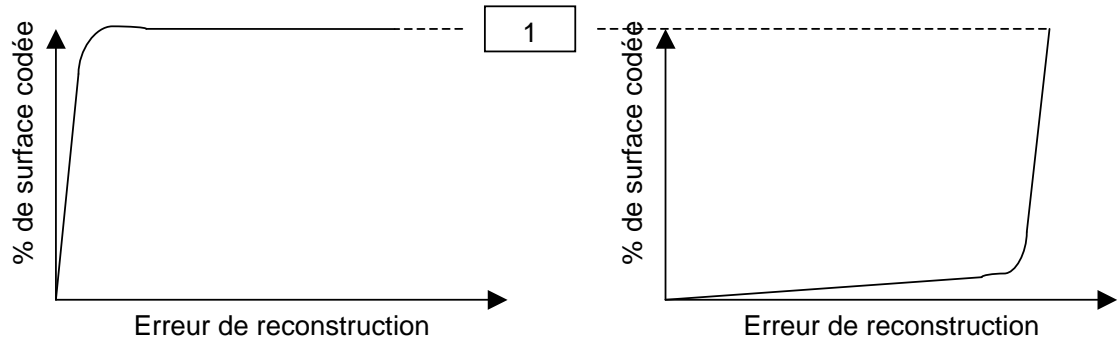


Figure 4-0 A gauche, Courbe idéale de « compressibilité » d'une image. La majeure partie de l'image peut être codée avec peu de domaines et avec une bonne qualité de reconstruction. L'image comporte un nombre important de similarités. A droite, la pire courbe. L'image comporte un grand nombre de blocs singuliers, il faut un seuil de reconstruction particulièrement élevé pour atteindre 100% de surface codée.

4.3.5 Approche par sélection des ranges

Jusqu'à présent nous nous sommes intéressé exclusivement à la sélection des domaines. C'est-à-dire que nous partions d'un ensemble d'individus restreint, les ranges, pour servir de base à l'établissement d'un score pour trouver les « meilleurs éléments » dans un ensemble bien plus grand, les domaines. Une autre approche consiste à utiliser les domaines cette fois comme éléments votants, et à élire des ranges. Autrement dit, l'ensemble des éléments à élire est le plus restreint (l'ensemble des ranges), et l'ensemble des éléments votant est le plus important (les domaines). Nous allons mettre en évidence les caractéristiques de cette nouvelle méthode de sélection et la comparer à la précédente.

Sélection de ranges représentatifs par inter corrélation normalisée 2D

Dans la section 4.3, nous expliquons comment l'inter corrélation normalisée peut être utilisée pour retrouver les occurrences proches d'une imagerie modèle sur une image entière. La valeur de cet opérateur renvoie un pic proche de 1 quand le signal local sur l'image complète correspond au modèle recherché. Cette représentation a l'avantage d'être indépendante du niveau de gris moyen et d'un facteur multiplicatif (le paramètre de contraste). Par expérience nous avons remarqué que des corrélations supérieures ou égales à 0,8 en valeur absolue correspondaient à une bonne correspondance visuelle (et vice versa). Nous avons donc mis en œuvre une méthode qui extrait les ranges représentatifs en utilisant cette propriété. Nous disposons en entrée d'une partition de l'image (éventuellement partition double, avec recouvrement partiel par demi côté de range), une liste d'échelles *côtéRange/côtéDomaine* et un nombre de ranges représentatifs souhaités. Cette liste peut contenir des échelles proches de 1, voire égale à 1. Dans le cas d'une échelle égale à 1, une liberté est prise par rapport à la contractance des transformations utilisées.

Le score d'un range est défini comme le nombre de pics dans le calcul de l'intercorrélation entre l'image dont nous voulons extraire les imageries caractéristiques et le range lui-même, en se restreignant aux pics supérieurs à 0,8, moyennant un test sur les 8 isométries, et en testant toutes les échelles demandées. Une autre définition

possible est de définir le score d'un range comme le nombre de pixels couverts par les domaines répondant positivement au seuil imposé. La Figure 4-0 illustre la différence entre ces deux méthodes de calcul de score. Dans le cas où le score est représenté par les pics de corrélation supérieurs à un seuil, l'aspect spatial des domaines répondant positivement n'est pas pris en compte. En effet, deux domaines proches spatialement et partiellement recouvrants compteront pour deux domaines complets dans le score, alors que dans la seconde approche, la partie commune aux domaines ne serait comptée qu'une seule fois. Cette seconde approche est plus intéressante mais consomme plus de temps de calcul (gestion de masque à chaque pic de corrélation).

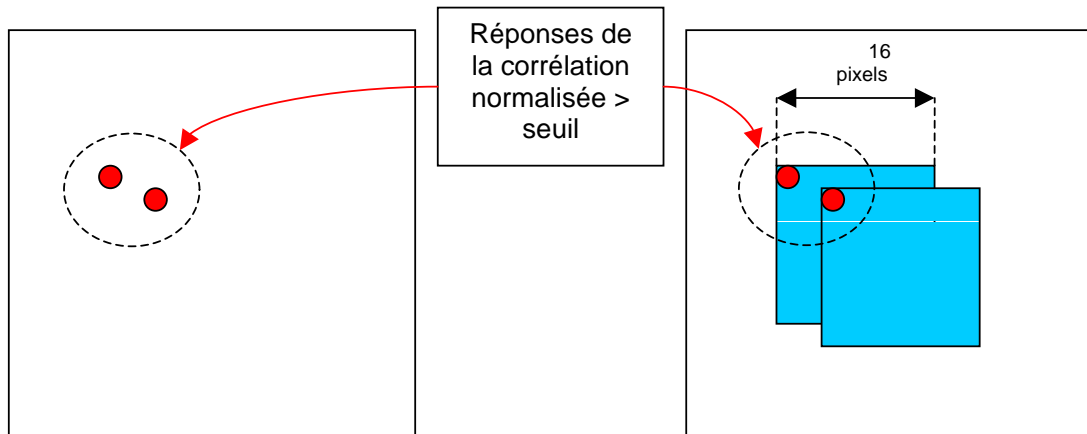


Figure 4-0 Les deux méthodes calcul de score pour la sélection de ranges par inter corrélation normalisée. A gauche, méthode par comptage des pics > seuil, le score vaut 2. A droite, méthode par nombre de pixels sur les domaines répondant au seuillage. Le score vaut alors $(16 \times 16) + (12 \times 2) + (4 \times 16) = 344$, les parties communes ne sont comptées qu'une seule fois.



Figure 4-0 Reconstruction type obtenue à partir de ranges extraits par inter corrélation normalisée. Le PSNR est de 21 avec des ranges 16x16 et des échelles range/domaine [1 1,5 2]. Nous remarquons que la géométrie est conservée mais que les parties fortement texturées sont aplanies, seule la géométrie globale de l'image est conservée. Cela est dû à la nature de l'opérateur choisi et à la taille de blocs image considérés.

Entrées : I , les ranges R , échelles range/domaine

Sorties : $\text{score}(R_i)$ pour chaque range R_i

Pour chaque range R_i

$P = \text{FAUX}(\text{dim}(I))$

Pour chaque échelle range/domaine : s

Pour chaque isométrie : t

$R_i' \leftarrow \text{agrandir}(R_i, s)$

$R_i' \leftarrow \text{appliquerIsométrie}(R_i', t)$

$C \leftarrow |\text{cornorm}(R_i', I)| > 0.8$

$P \leftarrow P \text{ OU } C$

Fin pour chaque

Fin pour chaque

$\text{score}(R_i) \leftarrow \text{card}[P(i, j) / P(i, j) = \text{VRAI}]$

Fin pour chaque

Figure 4-0 Algorithme de sélection de ranges représentatifs par inter corrélation normalisée seuillée.

La sélection des ranges les plus représentatifs se fait en prenant ceux qui ont le score le plus élevé, quelle que soit la méthode de calcul de score. L'algorithme de la méthode est décrit dans la Figure 4-0. La Figure 4-0 illustre la reconstruction de l'image Lena à partir de ranges de scores maximum, calculés à partir de l'inter corrélation normalisée 2D. Le PSNR est 21, les ranges considérés sont de taille 16x16 pixels, et les échelles range/domaine pour établir le score sont [1 1,5 2]. Nous remarquons que malgré un

PSNR de faible valeur, un nombre importants de détails ont été conservés, notamment les contours et les régions homogènes. Les régions texturées comme les plumes du chapeau, et les régions contenant beaucoup de détails, comme le visage, ont été aplanies et ont contribué à la dégradation du PSNR global de l'image.

4.4 Conclusion

Dans cette section nous avons détaillé la méthode d'apprentissage. La Figure 4-0 représente la chaîne de traitement complète de la phase d'apprentissage, à laquelle on a incorporé l'optimisation. Bien que les modules d'optimisation soient décrits plus loin dans le document (cf. chapitre 5), ce schéma nous permet d'avoir dès maintenant une vue d'ensemble sur les modules vus et à venir dans la phase d'apprentissage.

Etant donné un concept à apprendre, représenté dans une image, nous commençons par extraire les ranges et domaines. Les domaines peuvent éventuellement être sur plusieurs échelles, $x1,5 \times 2 \times 3$ par rapport à la taille des ranges. Nous rappelons qu'un domaine ne doit pas être de taille inférieure à un range si nous voulons respecter les contraintes des transformations contractantes de la compression fractale. Le module de filtrage permet, avant tout traitement, de supprimer des ranges et domaines que nous savons non pertinents a priori. Ce module a été utilisé notamment lors de l'apprentissage de volants, avions et hélicoptères, pour supprimer les imagettes correspondant au ciel. Sur des images terrestres, le filtrage peut être pris en charge par la séparation artificiel/naturel, assurée par la méthode décrite au chapitre 5.2. Les imagettes sont ensuite normalisées en niveaux de gris (entre 0 et 1), et en échelle. Cette étape est effectuée pour les ranges et les domaines. Les vecteurs de caractéristiques sont ensuite calculés. Pour chaque range, nous testons un pourcentage donné des domaines les plus proches, au sens de la distance par vecteur de caractéristiques. La répartition de l'ensemble des valeurs de l'erreur de reconstruction nous permet d'établir une liste de seuils (voir chapitre 4.3.4). La phase de clusterisation est optionnelle, selon la stratégie choisie pour l'évolution du seuil (cf. chapitre 4.3.4.1). Nous obtenons en sortie une liste d'imagettes représentatives, au sens des autosimilarités.

Version de Nicole toto

Ce chapitre s'est concentré sur l'extraction d'imagettes significatives d'une image dont on a montré, par leur capacité à reconstruire l'image, qu'elles contenaient la majeure partie de l'information. Nous allons maintenant tenter de tirer partie de cette extraction de connaissance pour modéliser non plus seulement une image mais un concept plus général que l'on pourra apprendre à partir d'un certain nombre d'images.

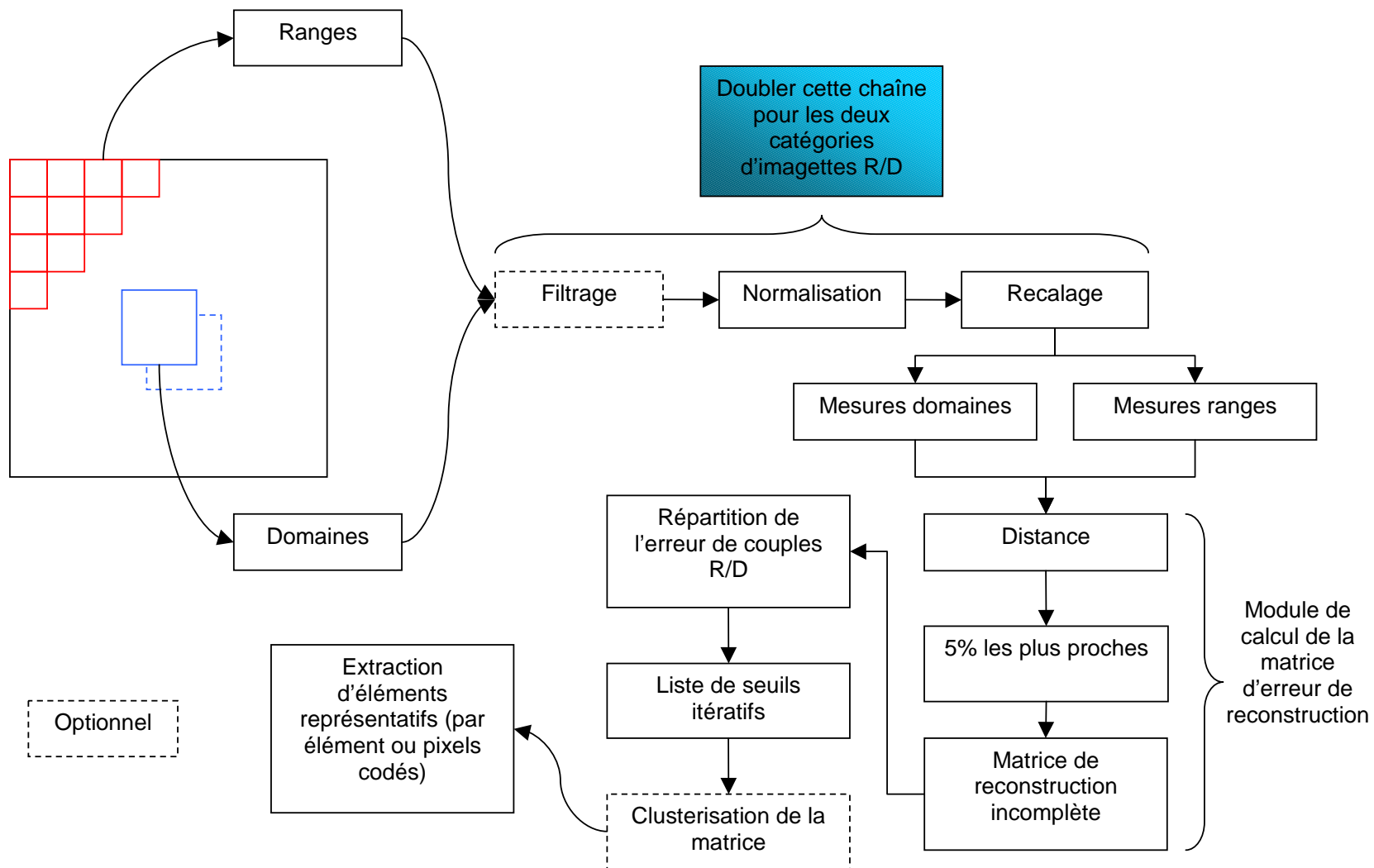


Figure 4-0 Chaînes de traitement de sélection de ranges représentatifs, phase d'apprentissage (schéma d'origine).

Dans cette section, nous avons détaillé une méthode d'apprentissage s'appuyant sur la compression fractale. Nous nous sommes penché dans un premier temps sur le principe de la compression fractale. Puis nous avons vu qu'elle pouvait mettre en évidence des redondances entre les imasettes présentes sur une image à étudier. Nous avons ensuite montré que ce type de compression pouvait être utilisé pour en faire un processus d'apprentissage. Une mesure sur les redondances entre les ranges et domaines de l'image permet d'établir un score et d'extraire les imasettes les plus représentatives.

Les principaux problèmes techniques posés ici sont la comparaison d'images pour en déduire une qualité de reconstruction locale et ensuite une mesure de représentativité de chaque imasette par rapport à l'image entière, tant en termes de qualité que de surface codée de l'image. Enfin, une fois les erreurs de reconstruction des couples range/domaine connues, nous avons proposé plusieurs méthodes de sélection des imasettes.

Dans la suite du document, nous nous intéresserons à l'utilisation des données apprises, à savoir la phase de reconnaissance à partir des imasettes extraites lors de l'apprentissage. La robustesse de l'ensemble de la chaîne apprentissage-reconnaissance ainsi que la quantité de données à traiter nous amèneront également à nous pencher sur la mise en œuvre de techniques d'optimisation.

5 Reconnaissance et optimisation

5.1 Position du problème, introduction

Nous avons proposé dans le chapitre précédent un processus d'apprentissage de concepts. Celui-ci repose sur l'extraction d'images caractéristiques du concept en question.

Notre but dans ce chapitre sera de mettre œuvre une méthode visant à utiliser les données extraites du processus d'apprentissage (i.e. les images représentatives) pour effectuer une reconnaissance des concepts appris. Lors de la phase d'apprentissage, nous avons mis l'accent sur le compromis surface apprise / qualité de reconstruction. L'intérêt était d'avoir une vue synthétique des concepts à apprendre et d'en extraire les éléments représentatifs qui le composent en mesurant leur taux de redondance dans le concept en question. Dans la phase de reconnaissance, nous allons utiliser cet aspect « composite » des concepts appris et nous en servir pour rendre notre méthode robuste à des environnements complexes.

Les données sur lesquelles nous allons appliquer les tests de reconnaissance sont des images issues de la base avions [Net 4] et des images au sol prises avec un appareil photo numérique [Net 5]. La base avions représente des avions vus de profil de différentes catégories, civile, militaire, et de différentes tailles, allant de l'avion de ligne au biplace à hélice. Certaines images ont été prises au sol, le but est alors de détecter l'avion en limitant les fausses alarmes dues aux bâtiments et véhicules qui se trouvent tout autour. D'autres images sont prises dans les airs, elles permettent d'aisément segmenter l'appareil, et nous les utiliserons pour l'apprentissage. La seconde base d'images contient des images prises au sol, en couleur, de jour, et converties en niveaux de gris. Elle contient des éléments variés de végétations, habitations et ponts.

Nous avons pu constater lors de la phase d'apprentissage que le nombre d'images considérées et les opérations que nous leur appliquons peuvent être coûteuses en temps de calcul. Nous serons confrontés à la même problématique lors de la phase de reconnaissance, celle-ci étant par certains points, symétrique à l'apprentissage. Pour fournir un résultat de reconnaissance en un temps raisonnable, une phase d'optimisation est nécessaire. Elle sera, de plus, applicable à l'apprentissage. Les optimisations utilisées seront d'ordre statistique et sémantique. La partie optimisation statistique a pour but de tenter d'apparier des images qui ont une forte probabilité de correspondre, relativement à des indications sur leurs attributs par exemple. L'optimisation sémantique effectue une séparation de l'image en amont de la chaîne algorithmique de reconnaissance. Cette séparation porte sur les classes « naturel » et « artificiel / manufacturé ». Nous avons pu constater dans le chapitre 2 que le traitement des zones manufacturées posait des problèmes différents des zones naturelles dans le cadre de la caractérisation de textures et de la comparaison d'images de petite taille. Les zones naturelles peuvent être parfois difficiles à caractériser dans des scènes très diverses, par contre dans une même image, elles possèdent des attributs statistiques qui restent localement stables. On peut ouvrir une parenthèse et noter que cela reste vrai si on considère une image multispectre. En multispectral, l'information portée par chaque pixel est très riche. La végétation à base de chlorophylle par exemple répond de manière très discriminante sur certaines bandes IR. Nous nous intéresserons brièvement à cet aspect dans ce chapitre, sans nous appesantir puisque notre problématique porte sur l'image monochromatique.

A l'inverse de la végétation, les zones manufacturées (notamment perçues à courte distance) sont caractérisables par leur hétérogénéité selon plusieurs prédicats associés à

des opérateurs décrits dans les chapitres 3.3.5, 3.3.6, 3.3.7. Pour classifier correctement de tels objets manufacturés, il est donc indispensable de prendre en compte leur contexte/voisinage et leur aspect composite. Dans notre processus de reconnaissance, nous allons donc commencer par effectuer une séparation entre naturel et manufacturé de manière à appliquer sur les imasettes des traitements appropriés qui fourniront des informations pertinentes. L'aspect composite des concepts appris est pris en charge par une carte de richesse des imasettes (d'un même concept) présentes localement pour reconstruire une zone de l'image testée.

Dans ce chapitre, nous étudierons donc une méthode de reconnaissance utilisant les données extraites de la phase d'apprentissage. Nous définirons la phase d'optimisation et montrerons comment celle-ci permet de dimensionner l'apprentissage en complexité de calcul et occupation mémoire. Enfin nous procéderons à une phase d'évaluation pour quantifier l'efficacité de notre approche et nous exposerons les conclusions qui en découlent.

5.2 Mise en place d'une méthode de reconnaissance

5.2.1 Introduction et approche préliminaire

Nous avons vu dans le chapitre 2 comment nous sommes parti de la compression fractale pour en faire une méthode d'apprentissage par extraction d'imasettes représentatives d'un concept. Nous sommes maintenant en mesure d'apprendre plusieurs concepts. A chacun d'eux nous pouvons faire correspondre une base d'imasettes caractéristiques. Nous allons maintenant voir comment mettre à profit ce qui a été « appris » et mettre en œuvre une méthode de reconnaissance basée sur cet apprentissage. Reconnaître un concept appris sur une nouvelle image équivaut à y retrouver les imasettes de la base d'apprentissage associée au concept. Plusieurs aspects pourront être pris en compte lors de la phase de reconnaissance.

Tout d'abord, nous considérerons chaque imasette apprise indépendamment et tenterons de la faire correspondre avec des parties de l'image de test. Cela pourra se faire soit par pattern matching, soit par similitude entre vecteurs d'attribut. Nous verrons plus loin comment l'utilisation combinée des attributs et du matching peut être mise à profit pour obtenir une version optimisée des phases de reconnaissance et d'apprentissage.

Ensuite, nous savons que les imasettes caractéristiques extraites sont liées entre elles par une contrainte de recouvrement de l'image apprise. Aussi sera-t-il important de tirer parti de l'aspect complémentaire qu'elles ont entre elles pour reconnaître de manière fiable les concepts choisis et éviter au maximum les fausses détections. Nous verrons enfin que, selon la manière dont nous gérons cette complémentarité, nous pourrons passer d'une phase de détection à une phase d'identification.

La Figure 5-0 illustre le principe de la reconnaissance. Il consiste à tenter de reconstruire l'image étudiée avec chacune des bases apprises. La meilleure reconstruction nous informe sur l'appartenance d'un objet de la scène à un concept appris, et sur la confiance que nous avons dans notre estimation. Nous disposerons alors pour cela de plusieurs informations, les imasettes utilisées, la qualité locale de reconstruction de chacune et les transformations que nous avons dû appliquer pour effectuer la correspondance entre imasettes (l'imasette extraite et la zone locale considérée de l'image testée). Certaines informations pourront être utilisées de manière statistique, pour caractériser ce que nous voulons reconnaître, comme les indices des imasettes utilisées pour reconstruire l'objet (information sur la structure de l'objet, on sait quelle partie du concept a servi à reconstruire quelle partie de l'image testée). D'autres informations peuvent être utilisées, au contraire, pour confirmer l'absence de l'objet. C'est le cas, entre autres, du paramètre de contraste. Un contraste proche de zéro aplatisse toute

image en une image quasi homogène. Si cette transformation est la meilleure possible pour cette base d'images à cet emplacement de l'image de test, alors il y a une forte probabilité pour que l'objet ne soit pas présent à l'endroit étudié.

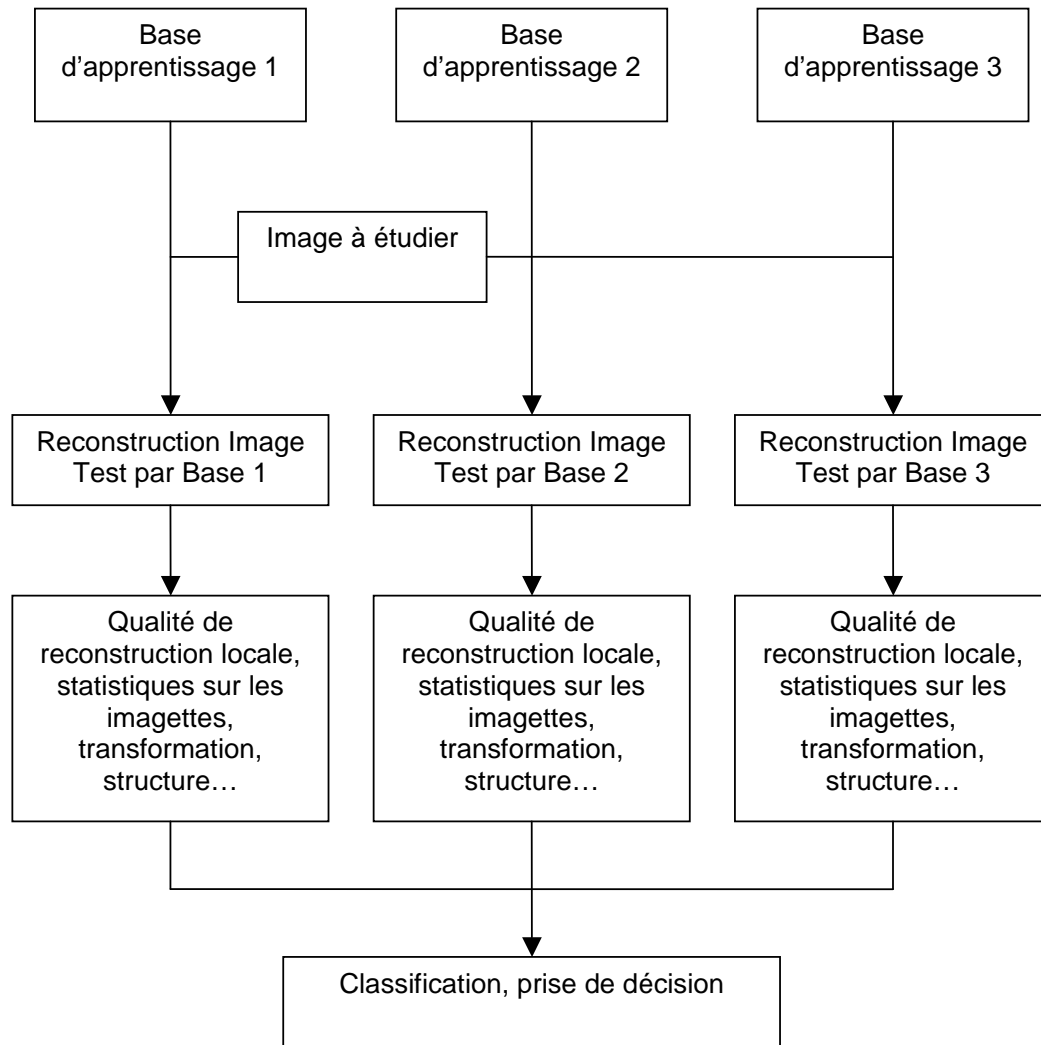
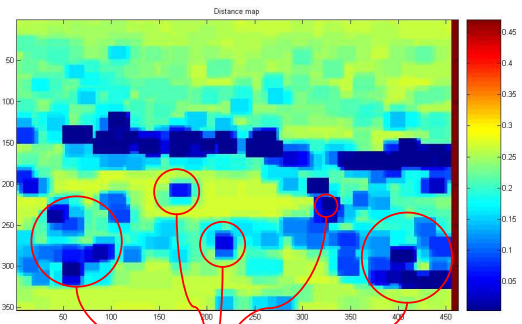
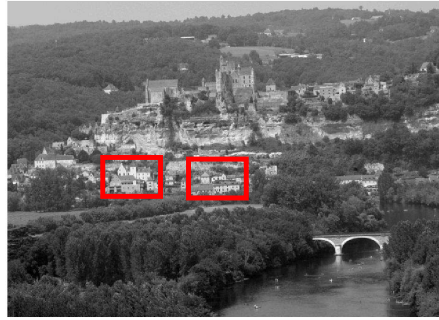


Figure 5-0 Schéma de traitement d'une image à partir des bases d'apprentissage pour en déduire une reconnaissance des portions de la scène observée.



Fausse alarmes potentielles

Figure 5-0 Résultat de détection locale de bâtiments. En haut, zones d'apprentissage de bâtiments. En bas, à gauche l'image de test, à droite les régions qui répondent favorablement à au moins une imagerie de bâtiment.

Dans les travaux de [VIN05], chaque document manuscrit était reconstruit à partir de chaque base apprise. La meilleure reconstruction permettait d'identifier le scripteur. Les documents étaient alors quasiment binaires. Dans notre cas, nous allons traiter des images en niveaux de gris. Pour conserver un degré de robustesse acceptable nous ajouterons une prise en compte des statistiques d'utilisation des imageries, de manière à supprimer les éventuelles fausses alarmes. Pour mettre en évidence ce dernier paramètre, nous avons lancé un apprentissage sur une zone contenant des bâtiments (cf. Figure 5-0). Les bâtiments étant de petites tailles dans ce premier test, nous avons extrait très peu d'imageries et celles-ci ont été uniquement caractérisées par des attributs statistiques (écart type, entropie). Dans un premier temps, nous avons établi une carte de distance locale entre l'image de test et la plus proche imagerie extraite des zones de bâtiments. Les fausses détections sont alors nombreuses.

5.2.2 Contexte et carte de richesse

Nous montrerons dans cette section comment la prise en compte de statistiques d'apparitions locales d'imageries, c'est-à-dire l'utilisation conjointe de plusieurs imageries de bâtiments pour reconstruire un même bâtiment, permettra de diminuer le nombre de fausses alarmes de manière significative.

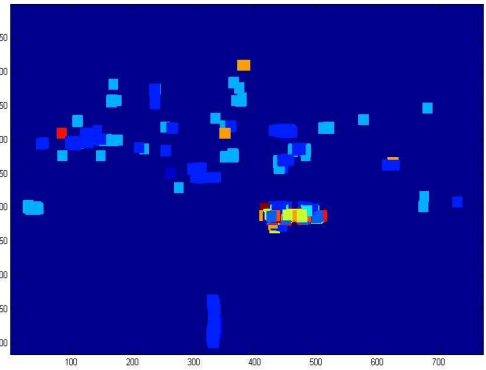
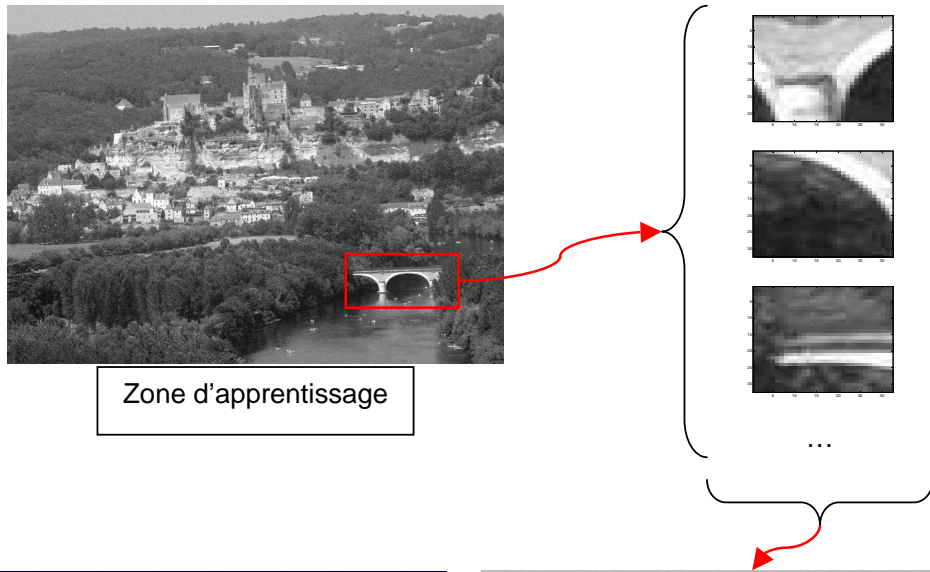
5.2.2.1 Contexte et utilisation des zones connexes

Nous avons vu plus haut que la détection d'un concept appris par simple distance locale par rapport à chacune des imagerie apprise menait à un nombre important de fausses alarmes. Nous savons que, la plupart du temps, un objet manufacturé est composé d'imagerie potentiellement hétérogènes d'un point de vue vecteur de caractéristiques. Par exemple, pour un même bâtiment, nous allons trouver des imagerie homogènes, des imagerie de coins et des imagerie de textures à proprement parler, telles qu'un toit en tuiles ou un mur de briques. Nous voulons tirer parti de cet aspect composite pour pouvoir détecter ce type d'objets dans son intégralité sur l'image et limiter par la même occasion le nombre de fausses détections. Autrement dit, nous considérons que nous avons détecté le concept appris si un nombre important de *différentes* imagerie de sa base d'apprentissage apparaissent dans une *zone restreinte* de l'image. La définition formelle de cette zone restreinte est un paramètre important et non trivial qui dépend du concept même. Pour le premier test traitant de l'aspect composite d'un concept, nous avons choisi de nous appuyer sur l'intercorrélation normalisée 2D, seuillée à 0,8 en valeur absolue. Cet outil possède des limitations qui sont rappelées au chapitre 5.3.2.1, cependant il nous permettra d'avoir une première approche intéressante dans un problème de détection ou de localisation d'un objet en nous fournissant une carte d'indices des imagerie recherchées sur l'image de test. A partir de cette carte, nous serons capable de considérer des zones connexes et de compter les apparitions de chaque imagerie apprise dans chacune d'elles. Pour chaque imagerie apprise, nous mettons à jour cette carte d'indices. Dans la carte d'indices des imagerie, l'indice de l'imagerie est associé à la position de chaque pixel recouvert par celle-ci et dont l'erreur de reconstruction locale est inférieure à la valeur actuelle du pixel. Nous disposons donc implicitement, pour chaque base d'apprentissage, de plusieurs cartes, de même taille que l'image testée : carte d'indices des imagerie, carte d'erreur de reconstruction locale, carte de contraste, etc. Ces cartes sont mises à jour localement à chaque correspondance entre une partie de l'image et une imagerie d'une base d'apprentissage. Dans le cadre de l'utilisation de zones connexes d'indices d'imagerie, l'erreur de reconstruction locale est représentée par l'intercorrélation.

Cette méthode de mise à jour des cartes de valeurs est détaillée en 5.3.3. Ceci correspond à n'écrire la valeur de la reconstruction que sur les pixels où la nouvelle erreur de reconstruction est plus petite que l'ancienne. Par défaut, les erreurs de reconstruction locale valent $+\infty$. Une erreur de reconstruction de zéro correspond à une reconstruction parfaite locale de l'image. Un indice de valeur zéro traduit qu'aucune imagerie n'a été affectée localement à cette partie de l'image et nous pouvons l'associer à ce que nous appellerons le « concept vide », ou « classe de rejet ». Considérons maintenant cette carte d'indices de classes. Si l'objet que nous cherchons apparaît dans la scène, alors il est représenté par une zone connexe d'indices, au sens *carteIndices* > 0 , et il doit utiliser un pourcentage important des imagerie apprise, idéalement la totalité. Pour détecter l'objet appris, nous recherchons donc les zones connexes d'indices non nuls, et comptons pour chacune d'elles le nombre d'indices différents utilisés. Les zones contenant une seule imagerie peuvent être ignorées de manière systématique, elles constituent des fausses alarmes. Ou alors cela impliquerait que l'apprentissage n'a renvoyé qu'une seule imagerie. Le concept appris ne serait alors pas composite, au sens des autosimilarités, et il n'y aurait donc pas d'intérêt à utiliser la méthode présentée.

Pour ce premier test sur l'aspect composite d'un concept, nous nous sommes positionné dans un cadre où l'appris est proche du testé. Nous avons pris deux images présentant la même scène avec un angle de vue légèrement différent et avec un niveau de zoom différent lors de l'acquisition de l'image. La Figure 5-0 illustre l'apprentissage d'un pont qui se traduit par l'extraction d'imagerie représentatives. Nous avons appliqué la méthode d'apprentissage du chapitre 4.3.3. La contrainte de la surface apprise n'étant pas prise en compte dans cet exemple d'apprentissage, nous obtenons un nombre important d'imagerie par rapport à la zone apprise. Une douzaine d'imagerie

caractéristiques sont sélectionnées dans la zone où se trouve le pont. Les isométries testées sont celles utilisées par [JAC92] (ces isométries sont définies dans la section regroupant les différents outils de localisation et transformations d'images, les isométries utilisées par Jacquin sont présentées en 5.3.2.2). Une reconnaissance de pont, dans une scène proche, est testée. Pour chaque zone de détection, nous disposons des images utilisées pour la reconstruire.



Carte de détection locale, les couleurs représentent les indices des imagettes. Une détection locale implique de nombreuses fausses alarmes.

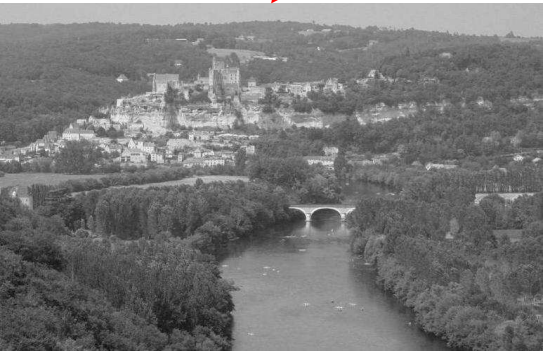


Image de test

La zone du pont utilise un grand nombre d'imagettes de la base « pont » pour se reconstruire.

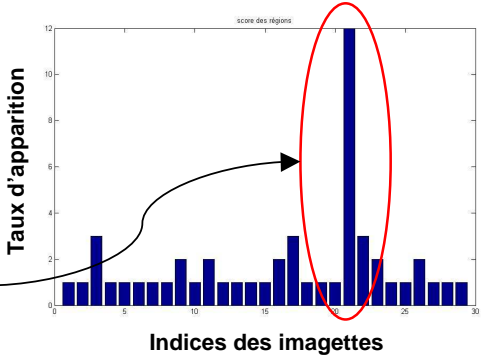


Figure 5-0 Mise en évidence de l'importance de l'aspect composite des imagettes d'une même classe, pour la suppression de fausses alarmes.

Un des problèmes de cette méthode est lié à l'utilisation de zones connexes. Nous pouvons avoir des zones qui conceptuellement ne devraient pas être connexes mais qui se retrouvent reliées sur la carte d'indices des imagettes. L'estimation de la classe correspondante peut alors être faussée. Par contre, dans le cas où les zones sont suffisamment éloignées, alors la détection d'un objet appris se fait de manière très nette.

L'histogramme de la Figure 5-0 représente le nombre d'imagettes différentes présentes dans chaque zone connexe. Il fait très nettement ressortir la zone connexe correspondant au pont. Un grand nombre d'imagettes de la base pont est nécessaire pour le reconstruire dans son intégralité. A l'opposé, les autres zones connexes extraites qui sont des fausses détections, forment des ensembles d'imagettes dont le cardinal est largement inférieur, au maximum 3. L'utilisation composite d'imagettes du pont a permis, d'une part de le détecter avec une forte discrimination (cf. histogramme), d'autre part de supprimer les fausses alarmes. La détection locale et l'usage multiple d'une seule et même imagette ne sont donc pas représentatifs d'une détection du concept appris. L'usage conjoint de plusieurs imagettes caractéristiques d'un concept dans une région restreinte de l'image, témoigne d'une forte probabilité de présence de l'objet en question.

La méthode que nous avons présentée dans cette section, bien que reposant sur des conditions simples, nous permet de nous rendre compte des points sensibles que nous aurons à résoudre pour obtenir une chaîne algorithmique complète. Tout d'abord, nous devons être capable de gérer des scènes bien plus hétérogènes, que l'appris et le testé qui ont été utilisés dans cette section. Cela implique, entre autres, l'incorporation d'une plus grande souplesse dans le calcul des isométries entre imagettes. Ce point sera traité en 5.3.2.2. De plus, nos prochains apprentissages seront basés sur la prise en compte de la contrainte de surface apprise définie en 4.3.4. Enfin, certaines zones qui ne sont pas connexes sur la vérité terrain peuvent se retrouver reliées avec cette méthode. Par contre, si nous obtenons une carte d'indices fiable, nous serions en mesure de prendre en compte l'aspect structurel d'un concept, et pas uniquement les statistiques d'apparition des imagettes apprises. Ceci impliquerait, pour le pont par exemple, de stocker d'une part qu'il y a deux fois plus d'arches que de piliers, mais également que les arches se trouvent au dessus des piliers. Le premier de ces points sera développé plus loin pour montrer comment nous pouvons utiliser l'apprentissage pour passer d'une méthode de détection à une méthode que l'on pourrait qualifier d'identification. En revanche, l'aspect purement structurel, reposant sur la géométrie de l'objet appris, ne sera pas développé dans cette étude et fera partie des perspectives possibles de nos travaux.

5.2.2.2 Carte de richesse à partir des imagettes apprises

Nous avons soulevé le problème de l'usage de zones connexes dans l'approche précédente. Un problème soulevé par ce type d'approche est que des régions candidates peuvent rassembler plusieurs objets lors de la détection.

Le but de la carte de richesse est d'utiliser la même information, à savoir la carte d'indices d'imagettes, mais de la présenter sous forme d'une surface continue, représentant le nombre d'imagettes différentes utilisées par unité de surface. Pour chaque base d'apprentissage, cette carte de richesse pourra être normalisée par le nombre d'imagettes de la base. Une telle carte est construite selon la définition suivante et possède l'avantage de ne pas être dépendante d'un seuil imposé a priori.

$$carteIndices(i, j) = \arg \min_k [cartesErreur_k(i, j)]$$

où k représente l'indice de l'imagette dans la base d'apprentissage,
 (i, j) représente la position du pixel correspondant dans l'image initiale.

Pour construire cette carte de richesse, nous définissons une fenêtre glissante sur la carte d'indices. A chaque position de la fenêtre, nous comptons le nombre d'indices d'imagettes différentes utilisées à l'intérieur de la fenêtre. Le point sensible de cette méthode est, comme pour chaque fois qu'on utilise une fenêtre glissante, la taille de la fenêtre. La fenêtre doit avoir une taille suffisamment grande pour contenir l'objet à détecter. Une taille trop grande augmentera le nombre de fausses alarmes potentielles.

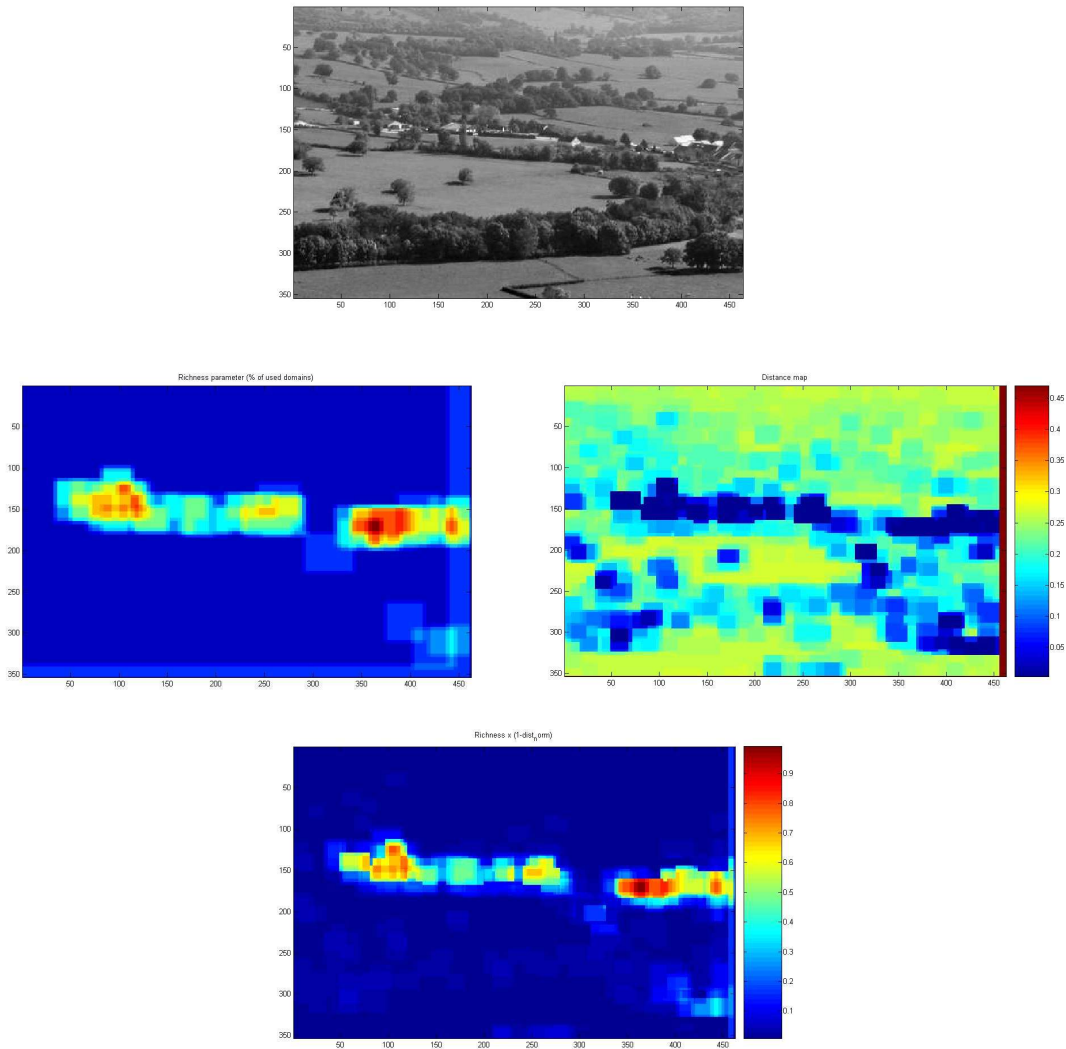


Figure 5-0 Exemple d'utilisation conjointe de la qualité de reconstruction et de l'aspect composite pour la suppression de fausses alarmes dans le domaine continu.

Nous pourrions supposer, quand nous observons la Figure 5-0, que la richesse pourrait suffire à la détection. La reconstruction locale est importante, car dans le cas où la taille de la fenêtre qui sert au calcul de la richesse, est trop importante, la reconstruction locale permet de conserver un masque qui épouse mieux le contour de l'objet à reconnaître.

L'avantage de la carte de richesse est sa simplicité et la concision de l'information qu'elle apporte. Son premier point faible est une dépendance par rapport à la taille de la fenêtre. Nous savons que cette lacune peut être comblée par d'autres paramètres de la reconnaissance.

Un autre problème est qu'un objet, localement, peut être construit d'un nombre restreint d'images représentatives. La carte de richesse est alors moins adaptée à ce genre de situation puisqu'elle renvoie une valeur faible alors que la procédure de

détection se situe à l'emplacement même de l'objet à détecter. Pour résoudre ce type de situation, la prise en compte de la structure de l'objet à reconnaître pourrait être nécessaire. Ce type d'approche ne sera pas détaillé dans ce document. Cela fera l'objet d'une partie des perspectives de nos travaux.

5.2.3 Conclusion

Dans une première approche pour effectuer la tâche de reconnaissance nous avons mis en évidence un certain nombre de problématiques. Parmi elles, nous avons constaté que l'utilisation individuelle des imagettes avec un simple bloc matching, produisait de nombreuses fausses alarmes. Notre réponse à cette problématique a été la prise en compte de l'aspect composite des imagettes pour la détection de l'objet appris. Nous avons utilisé dans un premier temps une carte de zones connexes. Puis nous avons établi une carte de richesse des imagettes, par base d'apprentissage, pour avoir une approche plus robuste. Finalement, nous avons montré que nous pouvions également faire de l'identification en se basant sur le taux d'apparition des imagettes.

Les problèmes apparus nous amènent à penser que plusieurs points doivent être approfondis. Dans la section suivante nous aborderons celui du matching et celui de l'optimisation des temps de calcul.

5.3 Outils, optimisation et dimensionnement de l'apprentissage

5.3.1 Introduction

Nous avons pu constater que nous étions souvent amené à apparier des imagettes pour tester leur ressemblance. A partir d'une imagette modèle, cela soulève essentiellement des problèmes de localisation et de recalage en rotation, sur une image entière. Ce sont ces deux points que nous allons traiter dans l'immédiat. De plus, le nombre de couples d'imagettes à apparier peut croître de manière significative quand la taille de l'image d'apprentissage, ou de reconnaissance, augmente. Nous allons donc également nous pencher, dans cette section, sur l'optimisation du traitement des données. Celle-ci se fera, d'une part par la séparation naturel/artificiel (optimisation portant sur la sémantique), d'autre part par des méthodes appropriées de recherche de couples d'imagettes.

5.3.2 Appariement d'imagettes

Un problème que nous rencontrons souvent dans nos algorithmes est la localisation d'imagettes, en particulier quand il s'agit de mettre en correspondance une imagette donnée avec celles présentes dans une image entière. Dans cette section nous allons discuter de quelques méthodes répondant à ce besoin.

5.3.2.1 Localisation

La méthode de base, quand le temps de calcul le permet, est **le parcours exhaustif** de l'image, en comparant les imagettes à toutes les positions possibles. Le point négatif de cette méthode est bien entendu la consommation en temps de calcul. Les avantages en sont la facilité de mise en œuvre et une exhaustivité de l'information extraite. La matrice d'erreur de reconstruction est alors complètement renseignée, donnant ainsi une vision globale de la reconstruction de chaque partie du concept. Cette méthode peut

servir de base, pour de petites images, pour comparer les temps de calcul et l'efficacité des méthodes d'optimisation.

Une autre méthode relativement proche du parcours exhaustif est l'**inter corrélation normalisée 2D [HAR92]**. A partir d'une image et d'une imagerie, elle renvoie une valeur dans l'intervalle $[-1, +1]$. Une valeur absolue proche de 1 témoigne d'une forte ressemblance entre l'imagerie et une portion locale de l'image. Une valeur de -1 signifie qu'on a retrouvé exactement le même signal mais en opposition de phase, ce qui correspond à l'imagerie dont les niveaux de gris auraient été multipliés par -1. Cette méthode est sensible à l'orientation de l'imagerie et à son échelle mais insensible au changement de luminosité et contraste. Un avantage de cette méthode est que la valeur renvoyée est aisément assimilable à un aspect perceptuel humain. Un seuillage de la valeur absolue de l'intercorrélation normalisée à 0,8 donne par exemple de très bons résultats sur des zones géométriques. Par contre, sur des zones artificielles, ce type d'opérateur n'est pas du tout approprié et les valeurs renvoyées ne sont pas significatives.

La méthode **Dynamic Time Warping (DTW)** a d'abord été développée pour les signaux monodimensionnels. Elle consiste à trouver les déformations nécessaires pour faire correspondre deux signaux, différents en échelle et amplitude. [BAE95], [BHA04].

Une autre technique très utilisée actuellement repose sur les **vecteurs SIFT**. Cette méthode commence par extraire des points d'intérêt de l'image en la convoluant par des gaussiennes de différentes tailles. Des caractéristiques (128) sont ensuite calculées sur ces points d'intérêts. Elles sont invariantes en orientation (recalage par rapport à l'orientation principale du gradient local) et en échelle (repérage par rapport aux tailles de gaussiennes utilisées) [LOW99], [LOW04]. Le souci connu des vecteurs SIFT est qu'ils peuvent ne fournir aucun point caractéristique sur certaines régions de l'image. Nous ne pourrions alors pas exploiter les redondances sur ces zones et la contrainte d'apprentissage sur le pourcentage de surface apprise ne serait alors pas toujours respectée.

Une autre approche consiste à calculer un **vecteur d'attributs** sur chacune des imageries présentes dans l'image de test et sur l'imagerie à retrouver. Si ces attributs sont pertinents, alors nous considérons qu'une proximité dans l'espace des attributs sera représentative d'une proximité dans l'espace des imageries avec une forte probabilité. C'est ce type d'approche que nous allons utiliser dans l'optimisation que nous proposons. Elle sera détaillée plus loin dans le document.

5.3.2.2 Recalage en rotation

La localisation d'une imagerie modèle par rapport à une image entière revient à effectuer un recalage en position. Nous allons maintenant traiter le recalage en rotation. Selon les types de recalage utilisés, il n'est pas toujours possible de dissocier orientation et position. Dans la méthode de [JAC90], huit isométries sont testées de manière exhaustive pour chaque couple range/domaine. Elles ont été définies dans le Tableau 4-0 (chapitre 4.2.2). Ce sont les isométries les plus classiquement utilisées.

Une autre méthode, plus coûteuse mais bien plus sensible consiste à trouver le minimum d'une fonction d'erreur entre les disques centraux des deux blocs image B_1 et B_2 . Le paramètre est l'angle de la rotation à appliquer sur le premier

$$\theta = \arg \min_{\theta} [\text{erreur}(\text{rot}(B_1, \theta'), B_2)]$$

Où *erreur* est une fonction de calcul d'erreur entre deux blocs image (en ne considérant cette fois-ci que les disques centraux des blocs considérés). Le calcul de la

minimisation est effectuée par la méthode décrite dans [LAG98]. La Figure 5-0 illustre un recalage de deux imagerie à partir de cette méthode.

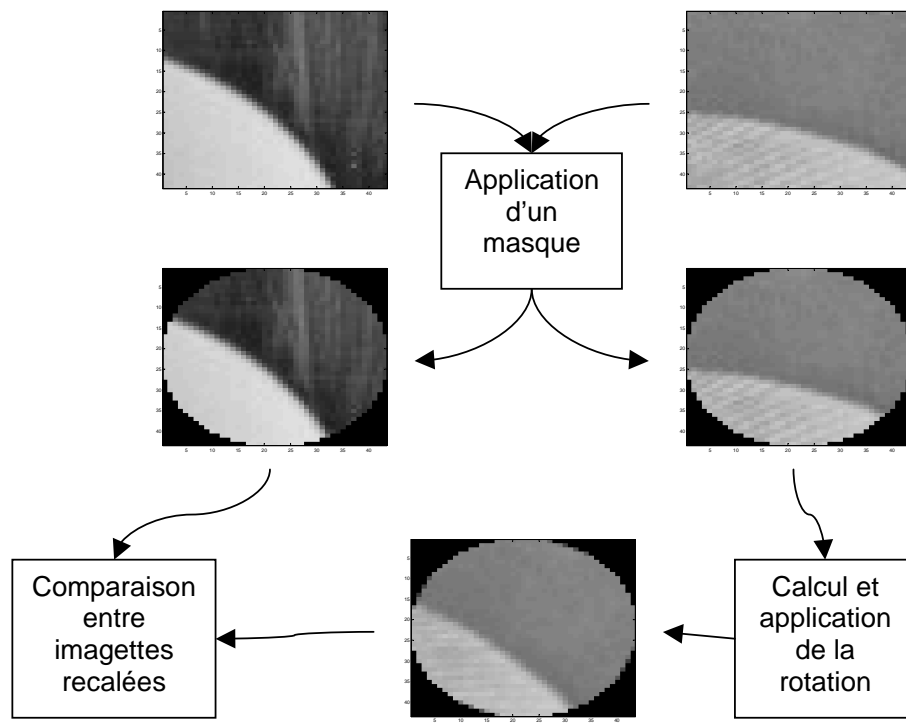


Figure 5-0 Exemple de recalage en rotation entre deux imagerie. Un masque circulaire est d'abord appliqué pour définir la zone sur laquelle l'erreur à minimiser est calculée, puis [LAG98] est appliqué pour trouver la meilleure orientation.

Nous verrons dans la section 5.3.4 comment ce recalage sera utilisé pour orienter toutes les imagerie dans une direction unique et ainsi permettre le calcul de certaines mesures géométriques sur les imagerie en question.

Méthode de recalage en rotation par inertie (TOTO ce n'est pas l'inertie !)

Nous pouvons recaler des imagerie entre elles en nous servant de leurs axes d'inertie. Considérons, comme précédemment, un masque circulaire M , de même taille que les imagerie à traiter, p est un pixel de niveau de gris p_{ng} et de coordonnées (p_x, p_y) .

$$q = \left(\frac{1}{n} \sum_{p \in M} p_x \cdot p_{ng}, \frac{1}{n} \sum_{p \in M} p_y \cdot p_{ng} \right)$$

où q représente les coordonnées du centre d'inertie et n est le nombre de pixels actifs du masque M . p_i contient la valeur du pixel situé aux coordonnées (p_{xi}, p_{yi}) . L'axe d'inertie passe par le centre de l'imagerie et le point q . Connaissant q nous pouvons recaler une imagerie soit par rapport à une orientation unique, en nous servant

de l'angle que fait l'axe d'inertie avec la verticale par exemple, soit recalcr des imageriees entre elles. Un exemple de recalcr est donn6 dans la Figure 5-0.

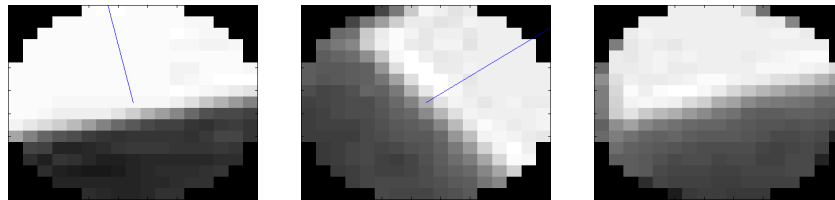


Figure 5-0 Recalcr d'imageriees par inertie. De gauche 6 droite, l'imageriee de r6f6rence, l'imageriee 6 recalcr, l'imageriee recalcr6e. Les lignes bleues repr6sentent les axes d'inertie.

Lors de nos tests, nous avons constat6 un rapport 15 entre la vitesse d'ex6cution de la m6thode par inertie et l'application de la fonction de minimisation d'erreur de [LAG98] 6 des imageriees en rotation.

M6thode de recalcr sur le spectre de Fourier

Une autre m6thode de recalcr en orientation est propos6e dans [JOS04]. Dans ce papier, l'auteur propose de repr6senter la texture dans l'espace de Fourier. Le niveau de gris moyen est supprim6 (i.e. on assigne z6ro 6 valeur centrale du spectre). L'orientation principale d'une texture appara6t de fa6on assez nette dans le spectre de Fourier sous la forme d'une raie dans le du spectre. Le principe est donc d'appliquer une Analyse en Composantes Principales (ACP) au spectre de Fourier de mani6re 6 extraire cette zone saillante. Dans son papier, l'auteur propose de seuiller par rapport 6 la valeur moyenne de la norme des 6l6ments (complexes) du spectre de Fourier. Ce seuil fonctionne avec des images peu bruit6es et r6solues, mais dans le cas images bruit6es, un seuil plus restrictif peut 6tre n6cessaire (95% de la valeur maximale par exemple), ou une ACP sur l'ensemble des individus pr6sents, pond6r6s par le log de leur valeur dans l'espace de Fourier, s'est av6r6 plus robuste dans ce cadre.

Un avantage significatif de cette approche est qu'en plus de l'orientation, nous pouvons nous servir du rapport des valeurs propres associ6es aux axes principaux. Cette derni6re valeur peut 6tre per6ue comme un taux de confiance dans la mesure de l'orientation, mais 6galement comme un indice sur l'isotropie de l'image observ6e. Dans le cas d'une texture parfaitement isotrope, les valeurs propres sont 6gales. Soient V_{\max} et V_{\min} les deux valeurs propres tri6es par ordre d6croissant. Le taux de confiance est calcul6 par

$$Confiance = \frac{V_{\max} - V_{\min}}{V_{\max} + V_{\min}}$$

Un probl6me classique lorsque l'on utilise le spectre de Fourier repose sur les effets introduits par les bords de l'image. Le spectre peut 6tre nettoy6 par une apolisation de l'image (multiplication de l'image par un \cos^2 en x, y normalis6s) : les bords tendent vers z6ro et le centre de l'image reste inchang6. Le r6sultat est un spectre de Fourier « moins bruit6 », les estimations orientation/confiance sur les images naturelles sont alors bien plus robustes.

Le recalcr en orientation vu plus haut propos6, issu de [LAG98], ne permet pas d'avoir un indice de confiance sur la diff6rence d'orientation obtenue. Par contre, dans les cas o6 les valeurs propres associ6es aux deux axes principaux sont quasi 6gales, la

méthode par ACP n'est pas capable de calculer de manière robuste une orientation alors que la méthode de [LAG98], plus coûteuse en temps de calcul, ne subit pas cet écueil.

Il peut arriver que deux instances d'un même modèle de texture soient disposées côte à côte sur l'image observée : deux champs de vigne sur une image aérienne par exemple. Ces deux objets vont renvoyer des signatures similaires selon certains opérateurs. Si on veut labelliser ces deux objets avec des étiquettes différentes, une méthode pour les séparer est de calculer l'orientation de la texture.

5.3.3 Mise à jour de masque par valeur min/max locale

Lors de la phase de reconnaissance, nous devons souvent mettre à jour des cartes de valeurs telles que la carte d'erreur de reconstruction locale, la carte d'indices et les cartes contenant les paramètres des transformations utilisées. Dans ce chapitre nous détaillons la méthode que nous avons préconisé pour mettre à jour ces différentes cartes. Ces dernières seront la plupart du temps de taille égale à celle de l'image traitée.

Ces mises à jour de cartes impliquent souvent des fenêtres recouvrantes. La question est de savoir comment modifier les valeurs dans les zones de recouvrement si les réponses des fenêtres recouvrantes sont différentes (ce qui est généralement le cas).

Nous partons d'une carte de référence, classiquement la carte d'erreur de reconstruction locale et d'un opérateur de comparaison, classiquement $<$ ou $>$, selon que nous cherchons à conserver les valeurs minimale ou maximale dans les zones de recouvrement entre positions de fenêtre. Les pixels à mettre à jour sont ceux présents dans la fenêtre à sa position actuelle et donc les valeurs associées dans une carte donnée (carte d'erreur de reconstruction par exemple) répondent positivement à travers l'opérateur choisi (« $>$ » dans le cas de la carte d'erreur de reconstruction, puisque nous cherchons à conserver les meilleures reconstructions, on « écrase » donc les valeurs supérieures à l'erreur actuelle).

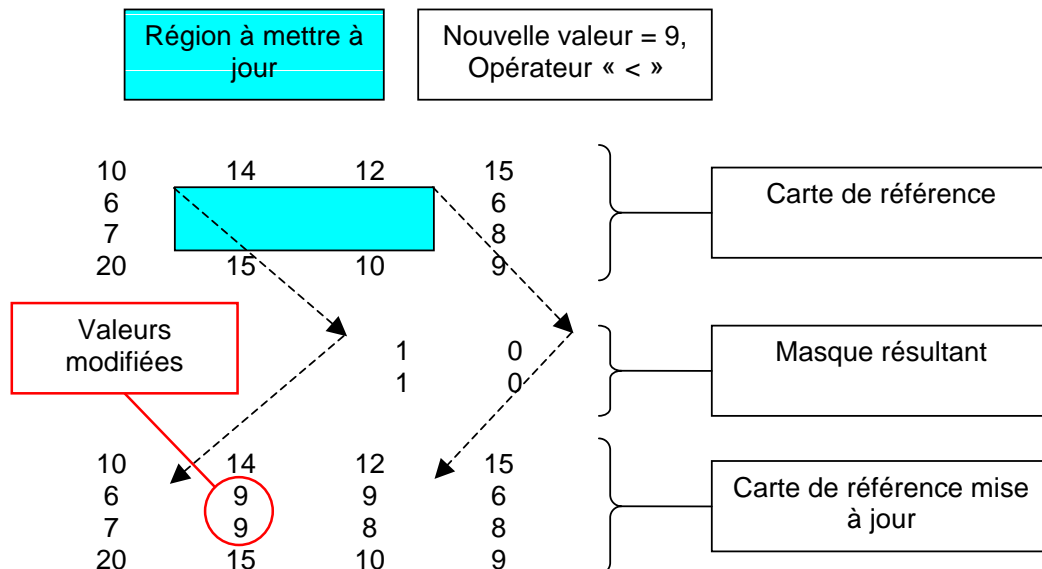


Figure 5-0 Illustration de mise à jour des différentes cartes par valeurs locales.

La Figure 5-0 illustre la mise à jour d'une région d'une carte de référence, représentée par le rectangle translucide bleu. Cette carte est par exemple une carte d'erreur de reconstruction locale. Le rectangle bleu représente la surface d'une imagerie tentant de matcher avec cette portion de l'image. Le « masque résultant » de la figure est calculé en listant les pixels dont la valeur dans la carte de référence respecte l'opérateur choisi (ici « > » à la nouvelle valeur, 9). Ce masque représente les pixels qui sont mis à jour dans la carte de référence ainsi que dans toutes les autres cartes associées telles que la carte de contraste et d'indices d'images.

Un point central dans le traitement d'images de scènes naturelles en extérieur est la séparation de zones naturelles et artificielles. Ces zones possèdent des caractéristiques différentes et il peut donc être intéressant de les séparer avant tout traitement ultérieur. Dans la prochaine section, nous aborderons donc le problème de la séparation entre le naturel et l'artificiel. Ce domaine étant vaste, nous ne testerons que quelques méthodes appropriées à nos besoins.

5.3.4 Séparation naturel / artificiel

Nous nous intéressons ici à la séparation entre naturel et artificiel. Nous avons pu constater qu'une partie des outils que nous avons explorés jusqu'à présent étaient plus adaptés à l'une de ces deux catégories. En effet, le naturel est plus aisément caractérisable par des mesures statistiques. À l'inverse, le manufacturé se distingue particulièrement par la présence de zones géométriques. Un problème de la séparation naturel/artificiel est que la définition sémantique et intuitive que nous en avons n'est pas toujours évidente à transcrire de manière formelle. En effet, un arbre est un objet naturel, mais est-ce qu'une rangée d'arbres, plantés par l'homme et parfaitement alignés, ne constitue pas une région manufacturée ? Une maison fait partie de l'ensemble des « bâtis », mais que dire d'un toit en chaume par exemple ? La classification en naturel ou artificiel est alors ambiguë. Dans [CAR04], Yves Caron s'intéresse à la séparation du naturel et de l'artificiel sur une image monochrome. Il rappelle qu'une séparation peut être effectuée en utilisant la dimension fractale. Dans [MAR84], les auteurs considèrent qu'un objet est naturel si sa dimension fractale est comprise entre 2 et 2,5. Dans [CAR90], les auteurs utilisent en plus l'erreur du modèle fractal pour caractériser le naturel et l'artificiel. Cette erreur est la moyenne des carrés des résidus de l'approximation linéaire effectuée lors de l'estimation de la dimension fractale. Un objet est alors classifié comme artificiel si sa dimension fractale est hors de l'intervalle [2, 2,5] défini dans [MAR84], et si son erreur par rapport au modèle est importante, supérieure à 90% dans [CAR90]. Dans ces articles, les auteurs ont appliqué leurs méthodes sur des prises de vue air/sol, perpendiculaires au sol. Ceci inclut l'imagerie aérienne et satellite. Ce type d'images possède l'avantage d'avoir très peu de perspective et donc une certaine constance sur le paramètre de profondeur, par rapport à l'axe du capteur. Nous serons amené à légèrement ajuster ces paramètres numériques, intervalle de dimension fractale et erreur modèle, pour traiter des images prises au sol et possédant une perspective très présente.

Nous avons ajouté un paramètre pour la détection des zones artificielles. Celui-ci consiste à estimer le rapport maximum entre les valeurs propres de la direction du gradient de la zone observée de l'image. Cette mesure prend une valeur élevée quand la région considérée possède un gradient élevé ou à forte directivité. La Figure 5-0 illustre le processus suivi pour extraire le naturel et l'artificiel sur les données.

Dans l'approche présentée en Figure 5-0, nous avons inclus un module de détection de zones d'ombres. La problématique de détection d'ombres, tout comme celles de la détection de reflets et de zones transparentes, sont des sujets difficiles à traiter puisque leurs solutions résident davantage dans la sémantique de l'objet dont l'ombre est projetée (et son contexte, conditions d'éclairage notamment) que dans des attributs statistiques de

l'image. Les ombres sont des éléments particuliers dans ce sens où elles peuvent, par exemple, représenter la silhouette d'un objet artificiel et être projetées sur une zone végétale. Classiquement, on observe que l'ombre d'un bâtiment est projetée sur de l'herbe. Une approche contours classifierait la zone comme un bâtiment, alors qu'une approche sur le contenu tendrait à la classifier comme du végétal. De plus, les zones ombrées sont des régions où le contraste est généralement faible et où l'information portée par le contenu est particulièrement faible (relativement au bruit de capteur). Pour toutes ces raisons, nous avons choisi de mettre en œuvre un module chargé de détecter les zones d'ombre et de les traiter à part dans la suite de la chaîne algorithmique.

Sans entrer dans des considérations géométriques complexes sur la localisation d'ombres, nous avons choisi de les caractériser formellement comme des zones de faibles amplitude et variabilité. Cela se traduit par l'usage de l'inverse du coefficient de variation [SOL98] de $(255 - I)$. A cela, nous ajoutons un seuil et supposons qu'une zone de luminosité moyenne supérieure à ce seuil ne peut en aucun cas être une ombre.

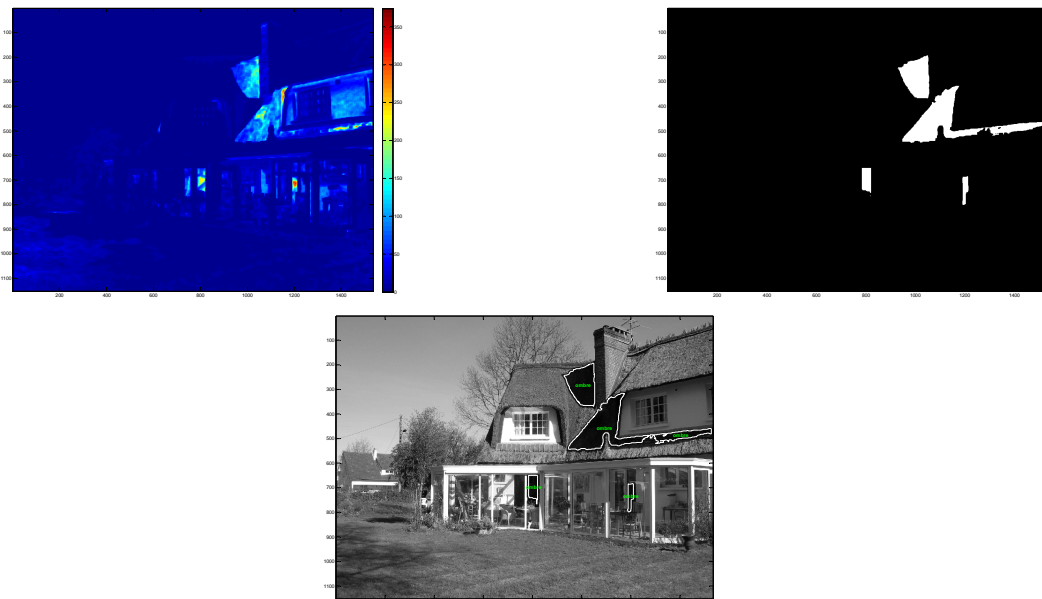


Figure 5-0 Exemple d'extraction de zones sombres et peu contrastées. En haut à gauche, carte du coefficient de variation local de $(255-I)$ où I est l'image initiale. A droite, version seuillée par hystérésis. En bas, détournage des régions sur l'image d'origine.

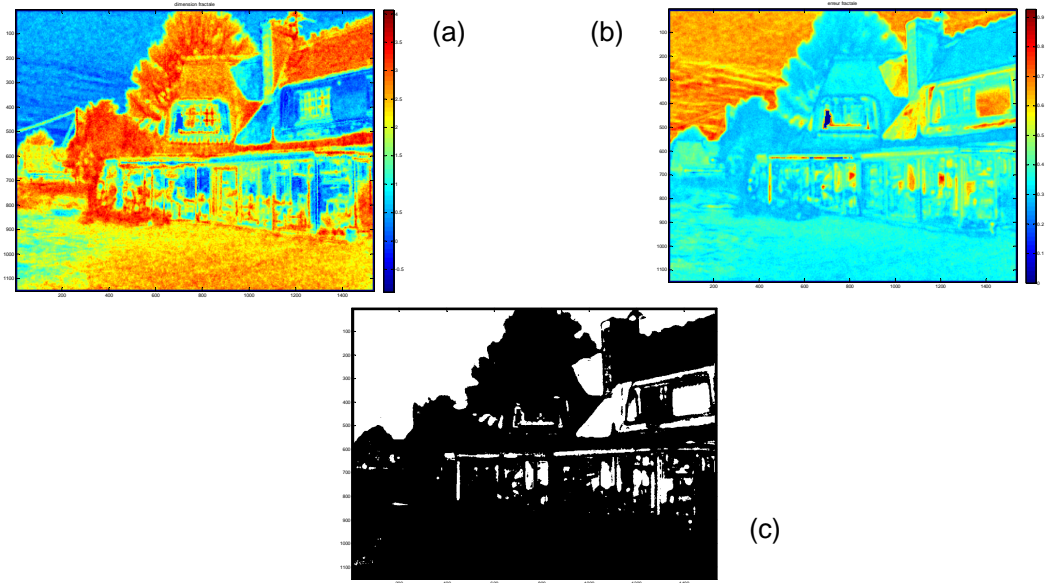


Figure 5-0 (a) : dimension fractale locale de la scène. (b) : erreur par rapport au modèle fractale. (c) : résultat de l'exploitation de (a) et (b) pour extraire l'artificiel (en blanc). Les textures homogènes comme le ciel sont considérées comme « artificiel ».

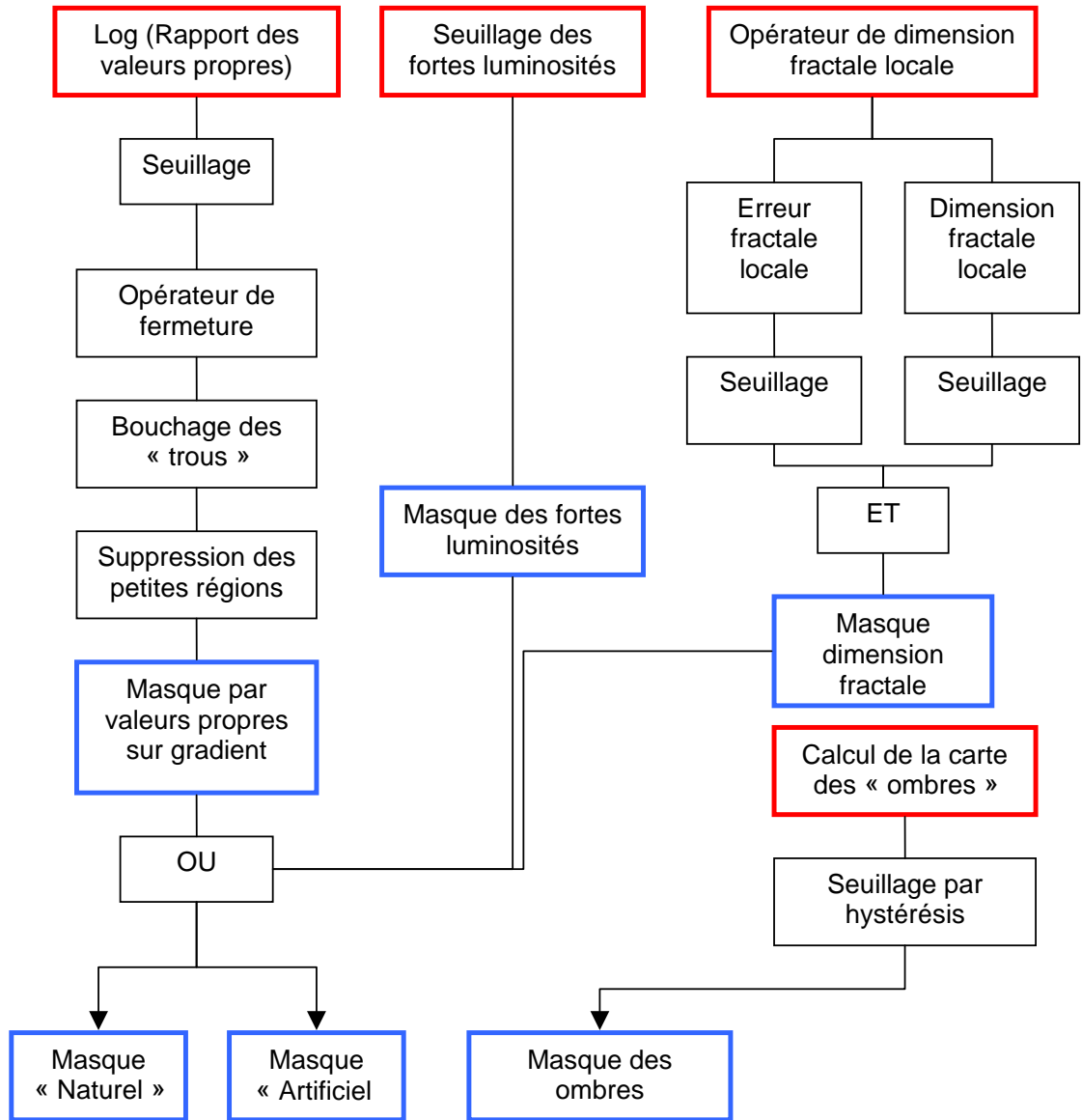


Figure 5-0 Schéma de séparation « naturel », « artificiel », « ombre ». Dans notre méthode nous avons utilisé conjointement les fractales, les valeurs propres sur le gradient de l'image et un seuillage par hystérésis pour séparer les zones naturelles et artificielles. Les zones d'ombres disposent de trop peu d'information pour être classées de manière fiable en « naturel » ou « artificiel ».

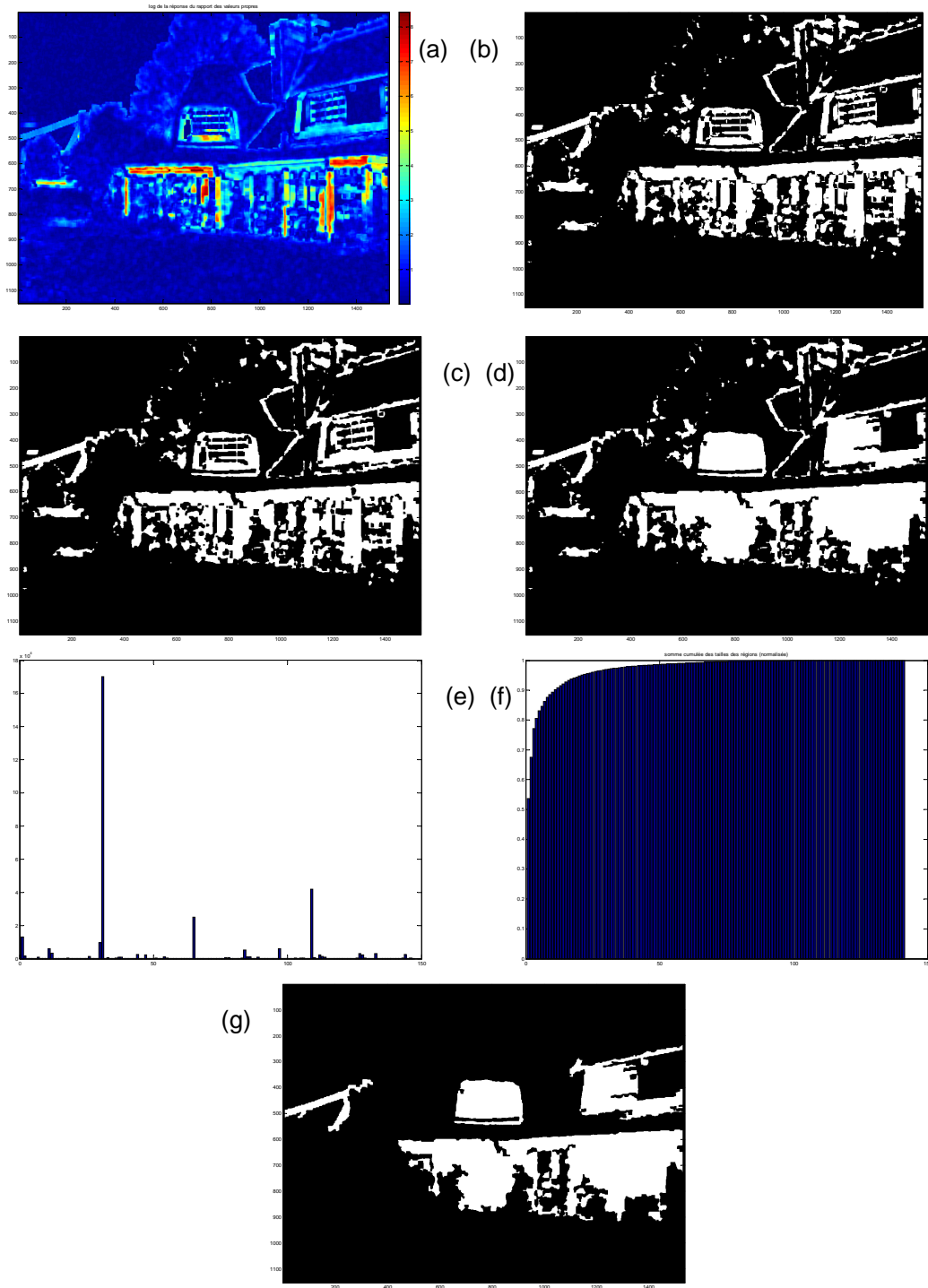


Figure 5-0 Séparation naturel/artificiel par log des valeurs propres du gradient de l'image. Carte du rapport des valeurs propres du gradient (a), résultat du seuillage de (a) à 1,5 : (b), masque après application morphologique de fermeture (c). La taille de l'opérateur est fixé en fonction de la taille de l'image. Masque après remplissage des trous (d). Taille des régions en nombre de pixels (e). Histogramme cumulé des tailles des régions (f). Zones conservées représentant 80% de la surface totale extraite (g). Nous récupérons les câbles, les fenêtres, et la véranda.

Même si nos objectifs portent sur des images en niveaux de gris, il peut être intéressant d'observer rapidement quel gain d'information nous aurions si nous avions les paramètres de plusieurs spectres à disposition (image couleur ou multispectrale). Ici encore, nous précisons que le traitement de la couleur est une tâche complexe que nous n'allons pas explorer en détail dans ce document. Cependant, dans le cadre industriel où nous nous trouvons l'imagerie multispectrale est de plus en plus présente, et quantifier l'apport d'information par plusieurs spectres, en plus de l'information texturale que nous allons extraire, nous a paru pertinent.

Dans certains contextes, il est nécessaire d'avoir une estimation macroscopique, globale et rapide de la scène. Cette rapidité peut être le résultat d'opérateurs simples sur la réponse des spectres en chaque pixel de l'image. Nous nous plaçons dans le cadre d'images couleur de jour, et faisons l'hypothèse (simpliste et contraignante) que la végétation est représentée par des zones vertes peu saturées. Pour cette raison, nous avons ici choisi de transformer, dans un premier temps, le triplet Rouge Vert Bleu (RVB) de chaque pixel par un triplet normalisé. L'avantage de cette représentation est qu'elle est indépendante du niveau de luminosité du pixel considéré.

$$RVB_{norm}(p) = \frac{RVB(p)}{|RVB(p)|}$$

où p est un pixel et $|v|$ est la norme du vecteur v .

Suivant notre hypothèse, une première estimation de la localisation de zone de végétation peut être effectuée en seuillant la composante verte normalisée.

Dans un second temps, toujours en considérant des couleurs normalisées, nous seuillons les pixels dont la seconde plus grande composante après le vert, (rouge ou bleue) est importante par rapport à la composante verte. Nous voulons ici supprimer les mélanges à prédominance vert+bleu ou vert+rouge.

Enfin, nous ne conservons que les pixels dont la saturation est inférieure à une valeur fixée a priori. L'effet ici est la suppression de zones fortement saturées, typiquement des peintures vertes sur des objets artificiels.

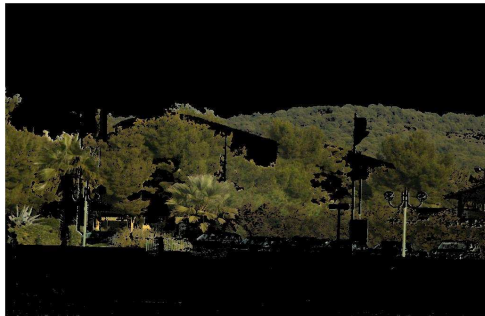
Cette méthode d'extraction de végétation par la couleur comporte trois seuils, et les opérateurs utilisés sont rudimentaires. Elle produit cependant des résultats intéressants et permet d'avoir une estimation de l'information disponible sur les trois spectres disponibles. La Figure 5-0 montre un exemple d'application de ces étapes sur une image couleur en plein jour. Le seuillage sur la composante verte normalisée seule permet un premier découpage de l'image. Certains éléments appartenant à des bâtiments ou des véhicules persistent. Les deux seuillage successifs sur la saturation et la valeur de la seconde plus forte composante permettent d'affiner la détection du végétal (toujours selon notre hypothèse simple définie plus haut). La Figure 5-0 montrent d'autres exemples suivant la même méthode (et les mêmes seuils).



(a)



(b)



(c)

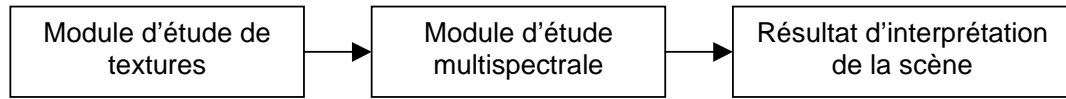
Figure 5-0 Image d'origine (a), détection de zones dont la couleur verte normalisée est majoritaire (b) et dont la composante rouge/bleue et la saturation sont inférieure à des seuils donnés (c).



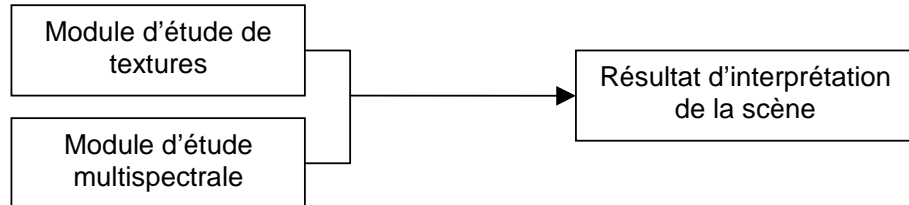
Figure 5-0 Test de séparation simple naturel/artificiel en s'appuyant uniquement sur la couleur (distribution des valeurs rouge,vert,bleu en normalisé).

Au regard de cette expérimentation très simple, nous sommes amené à penser que l'information spectrale contient une information riche sur l'image, avant même d'en étudier la texture. Même s'il est imparfait, un premier découpage de l'image en zones sémantiquement différentes peut être effectué (par exemple, séparation végétal/non végétal). Dans le cadre d'une application où le temps de réponse d'analyse de l'image est un paramètre important, un module d'étude multispectres en amont de chaîne de traitement serait un atout certain. Cette première classification de l'information présente dans l'image permet d'appliquer par la suite des traitements portants (par exemple) sur la texture de manière ciblée.

Stratégie (a) : étude de la texture, puis des spectres



Stratégie (b) : étude de la texture et des spectres en parallèle



Stratégie (c) : étude des spectres, puis de la texture

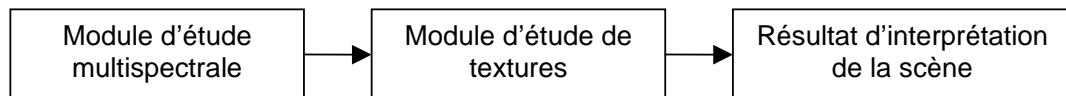


Figure 5-0 Différentes stratégies pour la gestion de la texture et du multispectral.

La Figure 5-0 illustre différentes stratégies d'ordonnement de la gestion de la texture et du multispectral. Si on se place dans le cas où le module de traitement en hyperspectral est plus rapide que l'étude de la texture (cas observé dans la pratique), la stratégie (c) est la plus avantageuse que les stratégies (a) et (b). (c) permettrait en effet de séparer rapidement l'image en plusieurs zones. Les traitements spécifiques de la texture seraient alors appliqués (ou non) à chacune de ces zones.

Dans cette section nous avons vu l'approche que nous avons choisie pour séparer le naturel de l'artificiel en nous fondant sur les fractales et la force de la directivité du gradient. Nous nous sommes ensuite penché sur les ombres. Nous nous sommes brièvement intéressé au problème de la couleur et à l'apport potentiel de plusieurs spectres. L'usage du multispectral paraît être un atout fort dans la reconnaissance du naturel, autant dans le domaine de la couleur RVB que dans des ensembles de spectres plus fournis (caractérisation de la chlorophylle par certaines bandes IR). Dans une chaîne comportant une partie analyse de texture et une partie analyse de spectres, cette dernière trouverait sans aucun doute sa place en amont de chaîne, présentant un découpage imparfait mais rapide de la scène. La texture pourrait agir en aval et produire des résultats plus spécifiques, locaux, sur la zone naturelle observée (densité, directivité). Des informations comme la dimension fractale ou la pente de la teinte peuvent également être utilisées pour obtenir une estimation du relief de la végétation (Figure 3-0).

Une fois l'image séparée en « naturel, artificiel, ombres », nous pouvons appliquer les procédures d'apprentissage/reconnaissance. Comme nous l'avons énoncé au chapitre 5.1, la phase d'apprentissage, mais surtout celle de reconnaissance, sont amenées à traiter et comparer un nombre important d'images. Elles sont donc par là-même coûteuses en temps de calcul et nécessitent la mise en œuvre d'une optimisation. Nous

détaillons dans la section suivante la méthode que nous avons suivie pour traiter les imagettes et optimiser les phases d'apprentissage et de reconnaissance.

5.3.5 Optimisation dimensionnement de l'apprentissage

Nous avons constaté qu'un point sensible de l'apprentissage par compression fractale était le grand nombre de couples d'imagettes à tester pour une correspondance éventuelle. Ce paramètre peut augmenter de façon significative avec la taille de l'image à apprendre et le pas de parcours choisi pour les domaines. De plus, la méthode de reconnaissance, que nous détaillerons plus loin, est le symétrique de la méthode d'apprentissage et conduira à des consommations de ressources du même ordre. Il nous a donc paru nécessaire de mettre en œuvre une méthode d'optimisation pour guider les deux phases, apprentissage et reconnaissance. Dans les deux cas, la méthode restera similaire à celle explicitée précédemment mais le module d'optimisation permettra de concentrer le temps de calcul sur un nombre limité de couples range/domaine, jugés comme étant les plus pertinents. Par cette contrainte ajoutée, la phase d'optimisation permettra de dimensionner l'apprentissage et la reconnaissance tant en terme de temps de calcul qu'au niveau de l'occupation mémoire. La maîtrise de ces deux paramètres pourrait s'avérer un atout lors d'un portage éventuel de nos outils sur du matériel embarqué.

L'apprentissage proposé est basé sur les valeurs contenues dans la matrice d'erreur de reconstruction. Celle-ci est de taille nbDomaines x nbRanges. Le calcul de l'ensemble de ses valeurs est coûteux et l'occupation mémoire potentiellement importante. Nous avons vu que cette matrice nous permettait de calculer une liste de seuils que nous appliquons itérativement, comme nous l'avons expliqué dans 4.2. Le but de ces seuils est de regrouper les couples d'imagettes par erreur de reconstruction « proche ». Une fois ces groupes constitués nous voulons extraire les blocs image les plus représentatifs. Nous pouvons optimiser le calcul de cette matrice soit en détériorant la qualité des valeurs contenues dans la matrice, soit en n'en calculant que la partie qui nous semble la plus pertinente.

Dans le domaine de la compression fractale, il existe de nombreux articles traitant de l'optimisation dans la recherche de similarités. Dans [HAS04] l'auteur démontre que les domaines les plus porteurs d'information sont la plupart du temps ceux dont l'entropie est la plus faible. La Figure 5-0 est extraite de l'article et représente les domaines de plus faible entropie. Nous pourrions parcourir les blocs soit directement par entropie croissante, soit par entropie proche de celle du range considéré, lorsque nous cherchons à l'apparier avec un domaine. Nous rappelons que l'entropie donne une mesure sur le désordre. Dans notre cas, il s'agira de mesurer le désordre dans l'histogramme de l'imagette considérée.

$$\text{entropie}(I) = - \sum_{i=1}^{NG} p_i \log(p_i)$$

avec

$$\sum_{i=1}^{NG} p_i = 1$$

où p_i est la probabilité d'apparition de la valeur i dans l'imagette I . NG est le nombre de niveaux de gris de l'image. Cet opérateur prend des valeurs faibles pour les zones homogènes de l'image et des valeurs fortes pour les régions qui utilisent un nombre important de niveaux de gris avec chacun un taux d'apparition significatif.



Figure 5-0 Les domaines de taille 8x8 (représentés en noirs) codant l'image Lena. Ils ont pour la plupart une entropie faible [HAS04].

D'autres méthodes d'optimisation sont basées sur une structure arborescente (toto REF), une clusterisation des imagettes en familles, une recherche à partir de l'erreur de reconstruction par rapport à un range moyen (toto REF). La plupart d'entre elles ne sont pas applicables à notre méthode d'apprentissage dans le sens où pour chaque range, elles recherchent un bon domaine quel qu'il soit, mais pas le domaine qui pourra servir à coder le plus de ranges possible, considérant une qualité de reconstruction donnée.

Le principe de l'optimisation par vecteurs d'attributs est de se positionner dans un cadre où la proposition suivante est vraie.

$$\forall(I_1, I_2), d_{E_1}(I_1, I_2) < \varepsilon_{E_1} \Rightarrow d_{E_2}(\text{mesures}[I_1], \text{mesures}[I_2]) < \varepsilon_{E_2}$$

où d_{E_1} et d_{E_2} représentent respectivement des distances dans l'espace des imagettes E_1 (où une dimension correspond à un pixel de d'une imagette) et l'espace des mesures sur les imagettes E_2 (où une dimension correspond à une mesure sur une imagette).

Autrement dit, nous voulons que la proximité de mesures dans un espace de caractéristiques E_1 nous donne des indications sur la qualité potentielle de matching dans l'espace des imagettes E_2 . E_1 est un espace de dimension égale au nombre d'attributs choisis, et E_2 est de dimension égale au nombre de pixels des imagettes I_1 et I_2 .

Dans le cas idéal, cette implication serait une équivalence. Nous n'aurions pas besoin de calculer de matching entre imagettes puisque la proximité dans l'espace de mesure serait alors directement représentative de la proximité dans l'espace des imagettes. Dans un cadre plus réaliste, nous voulons trouver des mesures d'attributs qui nous permettent d'affirmer qu'une distance trop importante dans l'espace des mesures implique une distance importante dans l'espace des imagettes. Dans ce cas, nous savons que le calcul de l'erreur de reconstruction du couple range domaine n'est pas nécessaire puisqu'il renverra une valeur élevée. Seuls les couples qui ont une probabilité forte d'obtenir un bon matching seront pris en considération.

Remarque : *Nous pouvons noter ici que le résultat du calcul de la liste de seuils itératifs pourra se trouver modifié. En effet, nous allons privilégier les couples qui ont une erreur de reconstruction faible, et donc par la même occasion, le paramètre qualité dans le compromis qualité / quantité de surface apprise de l'image sera privilégié.*

La méthode que nous avons élaborée consiste à définir un ensemble d'opérateurs respectant les invariants nécessaires à une transformation contractante et à les appliquer à chaque imagerie (range et domaine) pour leur associer un vecteur de caractéristiques. Pour que cette méthode soit intéressante et qu'elle permette réellement de gagner du temps, il faut trouver un bon compromis entre le temps de calcul consommé par les opérateurs et leur efficacité de filtrage. Dans le cas pratique, une proximité des vecteurs de caractéristiques ne sera pas automatiquement représentative d'une proximité dans l'espace des images. Si une telle implication était vérifiée, nous n'aurions alors pas besoin de faire l'opération de recalage (échelle, rotation) et le matching. Néanmoins des opérateurs appropriés permettront de guider la recherche des couples d'images et feront en sorte que l'implication soit vraie « suffisamment souvent ».

Dans le cas où nous choisissons d'optimiser le temps de calcul en détériorant la qualité de l'estimation de l'erreur de reconstruction, nous pouvons tenter de nous appuyer sur la distance entre les vecteurs de caractéristiques et la considérer comme une estimation de l'erreur de reconstruction. Nous devons alors avoir une approximation de la correspondance entre cette distance et l'erreur de reconstruction effective. La Figure 5-0 illustre cette correspondance sur une image d'avion civil. Nous avons tiré aléatoirement des couples range/domaine. Pour chaque couple, nous avons calculé la distance entre les vecteurs caractéristiques et l'erreur de reconstruction. Nous définirons plus loin le calcul des vecteurs caractéristiques. L'erreur et la distance ont été normalisées par rapport à leurs valeurs maximales rencontrées lors de l'expérimentation (les cas rencontrés étant supposés suffisamment exhaustifs). Le graphique de gauche nous informe sur le lien entre la distance vectorielle et l'erreur de reconstruction effective. Les segments de couleurs représentent une estimation linéaire par morceau de cette relation. Les segments rouges représentent une estimation linéaire classique sur le morceau considéré. Les segments verts représentent une estimation linéaire en rejetant 5% des valeurs les plus éloignées du modèle linéaire. L'histogramme de droite représente les erreurs de chaque modèle linéaire. Le second modèle accuse moins d'erreur sur tous les morceaux que le modèle linéaire utilisant 100% des mesures. Cependant, vue la dispersion par rapport aux modèles dans la partie centrale du graphique de gauche, il nous a paru incertain de s'appuyer sur ce type d'estimation pour obtenir une approximation de l'erreur de reconstruction.

Cependant, quand nous observons la première moitié du graphe (b) de la Figure 5-0, nous remarquons que pour de petites distances sur les vecteurs de caractéristiques, ces derniers sont alors relativement représentatifs de la future qualité de reconstruction d'un couple range/domaine considéré. Pour cette raison, nous avons opté pour une optimisation basée sur un remplissage partiel de la matrice de reconstruction plutôt que sur une estimation détériorée de l'erreur de reconstruction. Pour une imagerie de référence donnée, si nous considérons une liste d'images triées par ordre croissant selon la distance entre vecteurs caractéristiques, alors nous avons une forte probabilité d'avoir en tête de liste des images qui pourront matcher avec l'imagerie de référence.

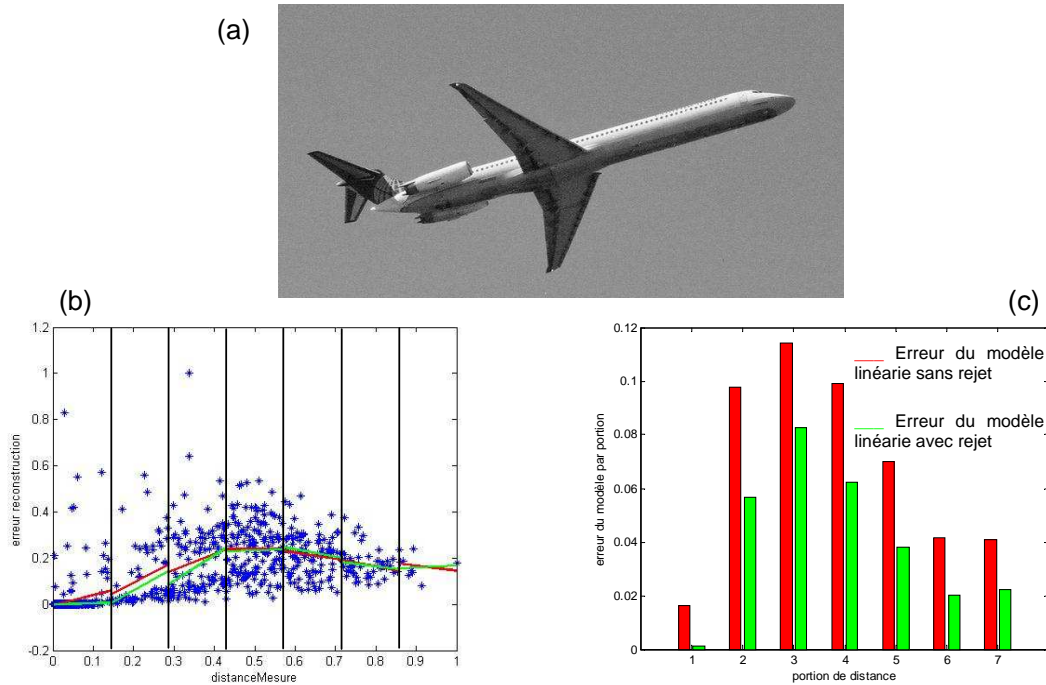


Figure 5-0 Estimation de l'erreur de reconstruction à l'aide d'une mesure de distance entre attributs. Image d'origine (a). Répartition de l'erreur de reconstruction en fonction de la distance entre attributs, ici entropie (b). Les segments rouges et verts dans (b) représentent une estimation linéaire par morceaux (ici 7 morceaux) respectivement sans et avec rejet des valeurs aberrantes. La méthode avec rejet (en vert sur (b) et (c)) produit un erreur plus faible que la méthode sans rejet des valeurs aberrantes (en rouge sur (b) et (c)) comme indiqué sur l'histogramme des erreurs sur chaque morceau (c).

La version optimisée de l'apprentissage consiste donc à définir un pourcentage de couples range/domaine à tester. Un pourcentage de 100% revient à utiliser la méthode exhaustive définie dans le chapitre 2. Pour chaque range, nous trions les domaines par distance croissante sur les vecteurs caractéristiques. Nous ne conservons que la tête de liste correspondant au pourcentage de couples choisis. Seules les imagerie de ces couples seront recalées en rotation et échelle pour ensuite calculer l'erreur de reconstruction du couple. Nous obtenons en sortie une matrice de reconstruction creuse, non pas au sens « remplie de zéros » mais plutôt « valeurs non renseignées ».

Remarque : Lorsque la matrice d'erreur de reconstruction est creuse, une représentation judicieuse lors de l'implémentation permettra de gagner avantageusement en mémoire occupée.

Nous avons vu que nous nous reposons sur des vecteurs de caractéristiques pour l'optimisation. Nous avons testé deux types d'attributs. Les premiers sont purement statistiques sur l'ensemble des pixels de l'imagerie. Ils ne tiennent absolument pas compte de la disposition spatiale des pixels. Les seconds sont des statistiques sur des quadrants de l'imagerie. Ils prennent donc en compte une partie de la géométrie de

l'imagette et requièrent un recalage dans une direction unique et commune à toutes les imagettes. Nous expliquerons plus loin comment nous avons procédé.

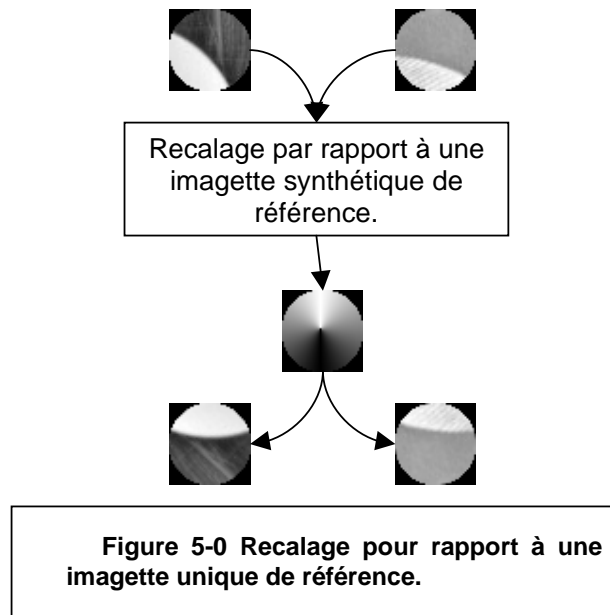
Calcul des vecteurs d'attributs

Un vecteur d'attributs proposant un bon compromis rapidité d'exécution / pertinence de la réponse est l'histogramme sur un nombre de classes limité, classiquement 8, sur l'imagette normalisée en niveaux de gris.

Un autre vecteur d'attributs statistiques que nous avons testé est une suite de caractéristiques telles que [entropie, écart type, dimension fractale]. Ces mesures sont normalisées par les valeurs min et max, rencontrées sur l'ensemble de l'apprentissage. Formellement il ne s'agit pas d'une normalisation, puisque nous pourrions potentiellement rencontrer des valeurs qui sortent de l'intervalle dans la phase de reconnaissance, mais dans tous les cas, si l'apprentissage est suffisamment fourni et est représentatif de ce qu'on souhaite reconnaître, l'intervalle appris permettra de calculer une mesure de distance, classiquement euclidienne, pertinente.

Optimisation avec recalage par rapport à une imagette artificielle unique

Si nous souhaitons extraire des informations sur la géométrie de l'imagette en nous basant sur ses quadrants, nous devons être en mesure de pouvoir la recaler selon une orientation unique. La prise en compte de mesures s'appuyant sur les quadrants permet la prise en compte de la géométrie grossière de l'imagette (ceci dépend du nombre de quadrants choisi), et de l'aspect statistique du contenu de ces quadrants. Nous pouvons ainsi, par ce type d'approche, obtenir un vecteur de caractéristiques robuste à de petits déplacements, tout en ayant une information relativement riche sur l'imagette. Le calcul de cette orientation unique peut se faire par ACP sur le spectre Fourier comme présenté dans [toto REF à définir] ou par recalage par rapport à une imagette-disque artificielle comme l'illustre la Figure 5-0.



L'avantage de recaler des imagettes selon une orientation unique est que nous pouvons calculer des attributs géométriques, dépendant de la position dans les imagettes recalées. La Figure 5-0 illustre ce type d'extraction.

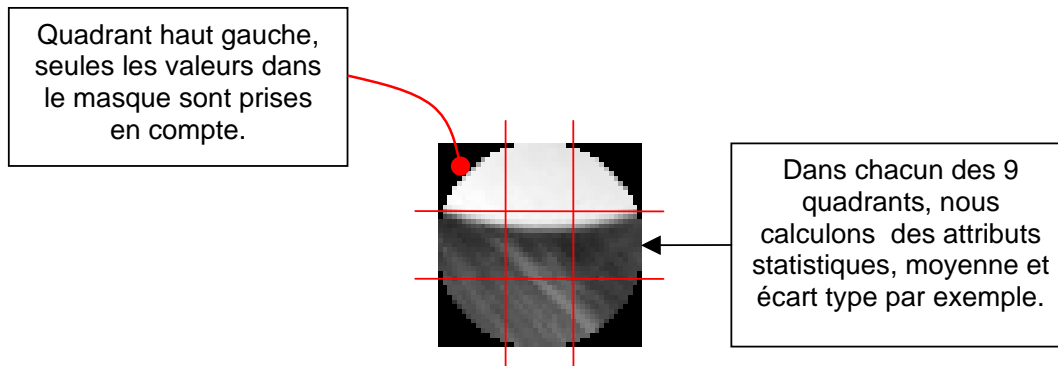


Figure 5-0 Exemple d'extraction d'attributs géométriques sur une imagerie recalée.

La Figure 5-0 illustre l'extraction de caractéristiques tirant parti de la géométrie de l'imagerie en se basant sur une orientation définie relativement à une imagerie artificielle unique. Les mesures effectuées sur des quadrants de l'imagerie, ceux-ci considérés dans un ordre prédéfini, prennent alors un sens et peuvent constituer une base fiable pour la construction d'un vecteur d'attributs.

Dans le cadre de l'apprentissage, l'optimisation consiste donc à cibler l'extraction de blocs représentatifs en se basant sur les couples qui ont une forte probabilité de matcher. Expliquons maintenant comment il est possible d'optimiser la partie reconnaissance en partant de la même approche.

Principe général de l'optimisation, côté reconnaissance

Dans la partie reconnaissance, nous effectuons la même extraction de mesures caractéristiques sur les domaines de l'image étudiée. Cette image ne comporte pas de range. Un filtrage peut être appliqué au préalable, comme pour la méthode d'apprentissage, pour supprimer par exemple les blocs considérés comme homogènes dans l'image (i.e. les blocs dont l'écart type des niveaux de gris est inférieur à un seuil donné).

Il existe plusieurs approches pour effectuer la reconnaissance de manière optimisée à partir des mesures stockées. Les deux que nous proposons sont représentées dans la Figure 5-0.

Nous proposons ces deux méthodes pour réaliser une optimisation. La première approche consiste, pour chaque domaine de l'image, à retrouver les blocs appris les plus proches. Cette approche s'inscrit dans un cadre à deux caractéristiques prédominantes. La première est l'importance du nombre de blocs appris pour tous les concepts. Pour chaque domaine de l'image de test, nous ne voulons pas traiter tous les blocs appris mais seulement les plus pertinents. La seconde caractéristique est qu'on se situe dans un contexte de compréhension globale de la scène. Nous voulons effectivement identifier toute l'image et relier chaque domaine à un concept appris. Dans cette approche, le but est de rapidement avoir une réponse globale pour chaque pixel de l'image. Ce dernier pointera alors par défaut vers le concept le plus proche. Dans cette approche, il n'y a pas de classe de rejet. Chaque pixel de l'image est relié au concept le plus proche. Ceci peut être problématique quand l'image contient un concept qui ne fait pas partie de l'apprentissage.

La seconde approche est symétrique de la première. Partant des blocs appris, nous voulons les tester en priorité sur les domaines de l'image qui ont le plus de chance de leur correspondre. Les éléments de l'image qui correspondent à un concept connu seront rapidement reconnus. Les éléments non appris seront potentiellement les plus distants des concepts, en termes de reconstruction locale et combinatoire d'imagettes, et resteront non attribués à aucun des concepts la majeure partie du temps. La classe de rejet est prise en charge de manière implicite en début de calcul. Nous entendons par là que les éléments non appris ne seront labellisés que très tard dans le processus de reconnaissance, c'est-à-dire quand le parcours aura duré suffisamment longtemps pour être proche du parcours exhaustif.

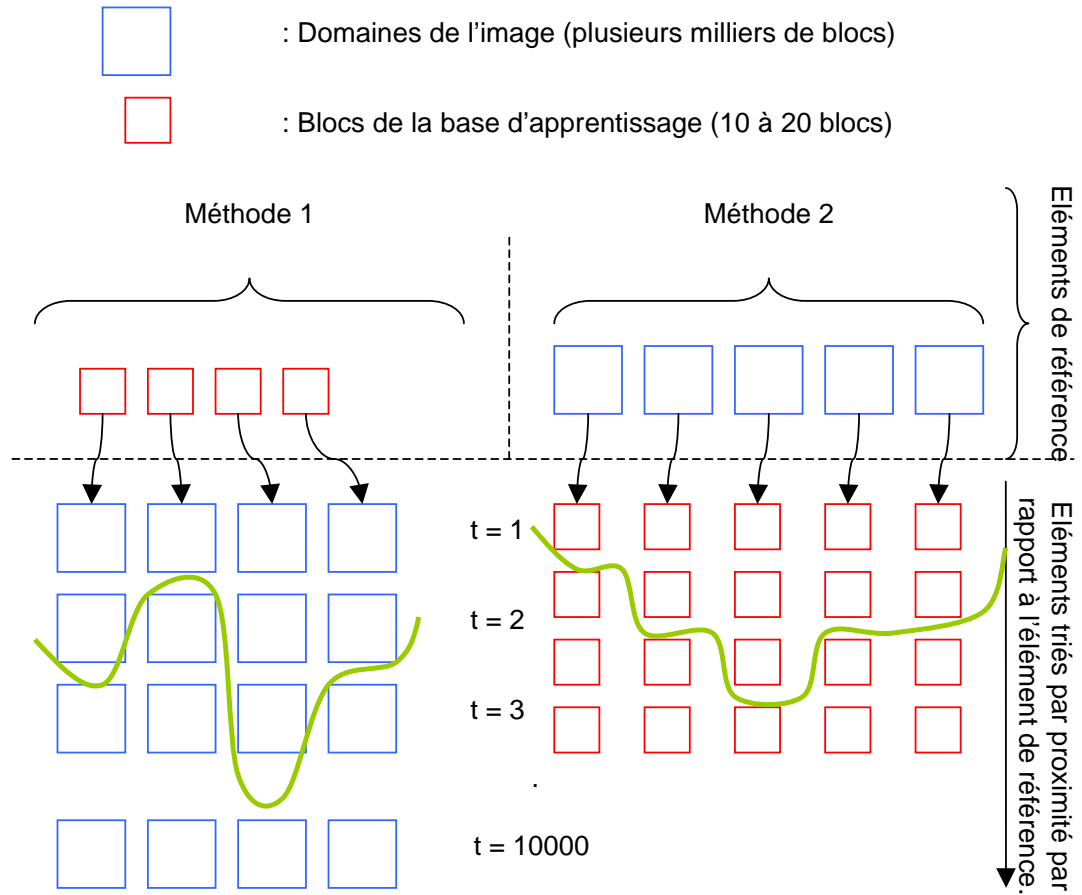


Figure 5-0 Illustration des deux méthodes de reconnaissance avec optimisation. Dans la première méthode chaque bloc appris pointe vers la liste triée des domaines de l'image de tests. Dans la seconde, chaque bloc de l'image de test pointe vers la liste triée des blocs appris. A chaque itération, nous conservons localement la réponse qui a obtenu le meilleur matching. Les courbes vertes représentent les réponses conservées à un instant donné.

Il reste un élément à figer pour que l'optimisation soit opérationnelle. Pour le moment, l'optimisation fait un parcours de même longueur que le parcours exhaustif des couples d'imagettes apprises et imagettes présentes sur l'image, hormis que ce parcours privilégie les « bons matchings » en début de liste. Nous devons donc obtenir une réponse pertinente et stable « relativement tôt » dans ce parcours optimisé. Une condition d'arrêt

pourrait être un délai fixé arbitrairement. Ce délai peut être déterminé par les besoins de l'application. Une application qui doit répondre rapidement imposera un délai court et disposera d'une réponse « en état » au moment de sa requête. Plus le délai est long, plus la réponse est fiable. Si la mesure statistique que nous utilisons est pertinente, c'est-à-dire si elle permet un tri des imagerie qui soit réellement pertinent (par rapport à une reconstruction effective de chaque imagerie), alors les différentes cartes fournies par la phase de reconstruction ne changeront plus à partir d'un certain temps puisque chaque tentative de nouvelle reconstruction locale produira une erreur supérieure à ce qui a déjà été testé.

Un autre paramètre que nous avons utilisé pour choisir l'ordre de parcours des domaines est le recouvrement par rapport au domaine voisin déjà parcouru. Deux domaines voisins ont un recouvrement fort. Si nous choisissons de parcourir les domaines uniquement dans l'ordre de proximité par rapport à un range donné, il arrivera souvent que la reconnaissance reste localisée dans des zones restreintes de l'image. Or si nous souhaitons avoir un aperçu rapide du résultat de la reconnaissance, la prise en compte du recouvrement comme second paramètre dans le tri de la liste de parcours des domaines pour un range donné permettra, pour ce range, le parcourir les domaines qui lui ressemblent le plus et qui le chevauchent peu.

Conclusion : L'optimisation permet de dimensionner l'apprentissage et la reconnaissance en termes de place mémoire et de temps de calcul. Cette souplesse de la méthode permet d'adapter notre algorithme à des contraintes variables d'un équipement à l'autre. De plus, lors de la phase de reconnaissance, si nous considérons un temps de calcul infini, la méthode reste alors équivalente d'un point de vue exhaustivité des images parcourues, mais les images les plus pertinentes seront traitées en priorité. Dans la pratique une réponse pertinente pourra donc être obtenue rapidement. La condition d'arrêt peut être fixée par une durée limite du traitement. L'avantage de ce type de stratégie est que, le jour où l'unité de calcul est remplacée par une plus rapide, le résultat du traitement sera affiné sans modifier ni l'algorithme, ni ses paramètres.

5.3.6 Conclusion sur les outils et l'optimisation

Dans cette section nous avons soulevé les différents problèmes relatifs aux outils que nous utilisons dans nos méthodes de reconnaissance et d'apprentissage. Au regard de la quantité d'images à traiter, nous nous sommes penché sur les questions d'optimisation.

Une première optimisation porte sur la séparation naturel / artificiel. Bien que ce traitement puisse être considéré comme une labellisation à part en tant que tel, nous l'utilisons comme technique d'optimisation pour diminuer le nombre de couples ranges/domaines à traiter dans l'image. Si nous cherchons à trouver un objet artificiel dans l'image, supprimer du traitement aval toutes les images comportant du naturel accélérera le processus de reconnaissance. Notons que la résolution de la problématique de séparation naturel/artificiel peut être facilitée par la présence de couleur dans l'image.

Les méthodes présentées ici travaillent sur des images en niveaux de gris. Seule la texture est utilisée pour séparer le naturel de l'artificiel.

La méthode proposée reste exhaustive si on la laisse tourner suffisamment longtemps, cependant l'optique utilisée ici est de donner une réponse pertinente le plus tôt possible. Si cette approche doit tourner sur du matériel amené à évoluer en terme de puissance de calcul, ce type de stratégie permet de bénéficier d'une augmentation de la précision sans modifier le logiciel embarqué.

5.4 Évaluation

5.4.1 Introduction

Nous avons détaillé précédemment une méthode basée sur la compression fractale, permettant l'extraction d'images caractéristiques d'une image. Nous avons ensuite abordé les problématiques d'optimisation et les différents outils nécessaires composant la méthode ou pouvant agir conjointement pour améliorer les performances en temps de calcul. Dans cette section, nous allons quantifier l'efficacité de notre approche en nous appuyant sur une base d'images publique. Nous nous intéresserons brièvement à des extensions de notre méthode. La première est l'extraction de points marginaux sur l'image. Notre système de notation donne un score élevé aux images les plus représentatives en termes de redondance. Nous avons voulu tester l'extraction d'images les moins redondantes. La seconde extension de notre approche est l'identification d'objets résolus (i.e. représentés sur le plan capteur par un « nombre important » de pixels). Une fois un objet localisé sur l'image, nous avons voulu savoir si notre méthode était capable de dire plus spécifiquement à quelle sous-catégorie l'objet appartient.

5.4.2 Reconnaissance

Nous nous plaçons ici dans le cadre où nous disposons en entrée de blocs image que nous avons extraits par la méthode détaillée dans la partie 4. Nous allons mettre l'apprentissage en application sur des données de test. Le but est de mettre en évidence l'efficacité de l'apprentissage et de montrer la pertinence des réponses fournies en phase de reconnaissance en s'appuyant sur les images apprises.

Une première problématique est de savoir si l'objet appris est présent dans l'image de test ou non, sans pour autant savoir où il se trouve précisément dans cette image (problème de détection de l'objet).

Une seconde problématique est, sachant a priori que l'objet est présent dans l'image, de savoir où il se trouve (problème de localisation de l'objet).

Enfin, une fois que la présence de l'objet est déterminée et qu'il est localisé sur l'image de test, une phase d'identification est possible. Il s'agit là de savoir à quelle sous-catégorie l'objet appartient (par exemple le modèle d'un avion). Nous allons tester notre méthode sur ces différentes problématiques.

La Figure 5-0 illustre un cas simple d'apprentissage/reconnaissance. Nous avons ici appris des blocs représentatifs d'un avion (a). Les images suivantes de la figure représentent, de haut en bas et de gauche à droite, l'image de test (b), la reconstruction de l'image de test (c) par les blocs extraits de l'image d'apprentissage, la carte des indices des blocs représentatifs (c) (à chaque couleur correspond un indice), la carte de contraste appliqué à chaque bloc représentatif pour reconstruire localement l'image de test (d) et la carte de richesse de images utilisées (f) pour la reconstruction. La méthode utilisée pour obtenir dans cette carte est celle décrite au paragraphe 5.3.3.

Bien qu'elle ne soit pas aussi porteuse d'information que la carte d'indices de blocs, la carte de contraste permet cependant d'avoir un aperçu d'où pourrait se trouver l'objet cherché s'il était présent dans la scène. En effet, bien que le contraste soit un paramètre laissé libre dans les transformations contractantes, un contraste élevé, relativement à un bloc appris, signifie que ce dernier a été peu modifié pour reconstruire localement l'image de test. Il peut donc effectivement s'agir de l'objet appris. De même, un contraste local très faible signifie que, moyennant tous les blocs appris disponibles, celui qui correspond le mieux a dû être fortement (ou complètement) « aplati » par un contraste particulièrement faible. L'objet cherché n'est alors potentiellement pas dans cette zone de l'image. Dans la Figure 5-0, le contraste local, relativement aux blocs appris, est plus élevé dans la région où il y a des avions. La carte la plus importante est celle contenant les indices des blocs représentatifs utilisés pour la reconstruction de l'image de test. Cette

carte permet de savoir quel bloc a été utilisé à quel endroit. Elle permet de calculer la carte de richesse, détaillée dans la section 5.2.2. La problématique de présence ou non de l'objet appris dans une image de test donnée peut être résolue par la présence simultanée d'un certain nombre de blocs représentatifs dans la scène. La problématique de localisation de l'objet dans la scène, sachant qu'il est présent, peut être résolue en seuillant la carte de richesse par rapport à la richesse maximale de l'image. Dans la Figure 5-0, nous sommes parti du même apprentissage que dans la Figure 5-0. Nous avons choisi de présenter les différences de localisation de l'objet en changeant le seuil appliqué à la carte de richesse. Un seuil élevé permet d'obtenir un masque plus resserré autour de l'objet à retrouver, mais il peut arriver que certaines instances soient mal détournées. Un seuil plus faible permet de mieux englober l'objet en question, mais le masque peut alors être plus grossier. Cette sensibilité est due en premier lieu à l'intervalle d'échelles autorisées lors de la phase de reconnaissance. En effet, si nous utilisons des échelles très petites, nos blocs appris seront alors fortement réduits en taille et pourront correspondre à beaucoup de zones de l'image testée. Dans un second lieu, cette sensibilité peut être imputée également (et surtout) aux limitations de la carte de richesse détaillées en 5.2.2. La Figure 5-0 reprend le même principe de reconnaissance. Nous remarquons que les régions mises en relief par la carte de richesse se trouvent au niveau de l'aile et de la queue de l'appareil. Cela est dû au fait que ces zones sont des intersections de régions de l'avion (fuselage + queue, fuselage + aile), on y trouve donc un nombre plus important de blocs représentatifs. Dans la Figure 5-0, nous avons testé nos méthodes d'apprentissage et de reconnaissance sur des ponts. Les blocs représentatifs extraits sont des portions d'arche et de pilier du pont, ce qui semble conceptuellement juste. Lors de la reconnaissance nous avons choisi de seuiller la carte de richesse quand au moins 80% des blocs appris avaient été utilisés. Le masque obtenu en sortie est un peu plus large qu'un masque idéal, cependant nous rappelons qu'aucune modification de la méthode n'a été apportée pour passer d'un apprentissage d'avion à un apprentissage de pont.

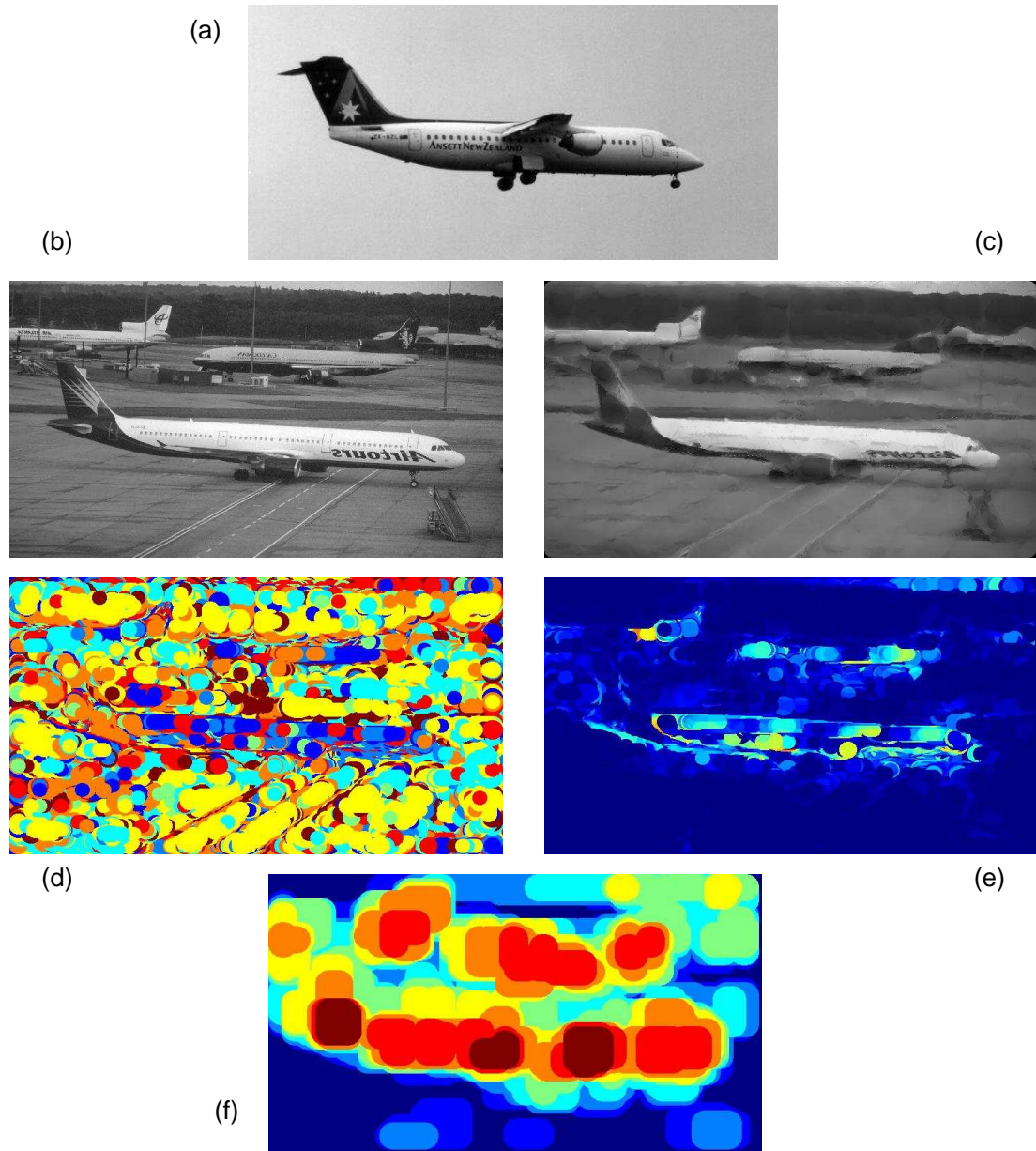


Figure 5-0 Exemple de reconnaissance d'avions à partir de la base « airplanes ».
 De haut en bas, et de gauche à droite : L'image d'apprentissage (a), une image de test (b), l'image de test reconstruite à partir des imagerie extraites de l'image d'apprentissage (c), indice des blocs représentatifs utilisés pour la reconstruction (d), carte de contraste (e) et carte de richesse (f).

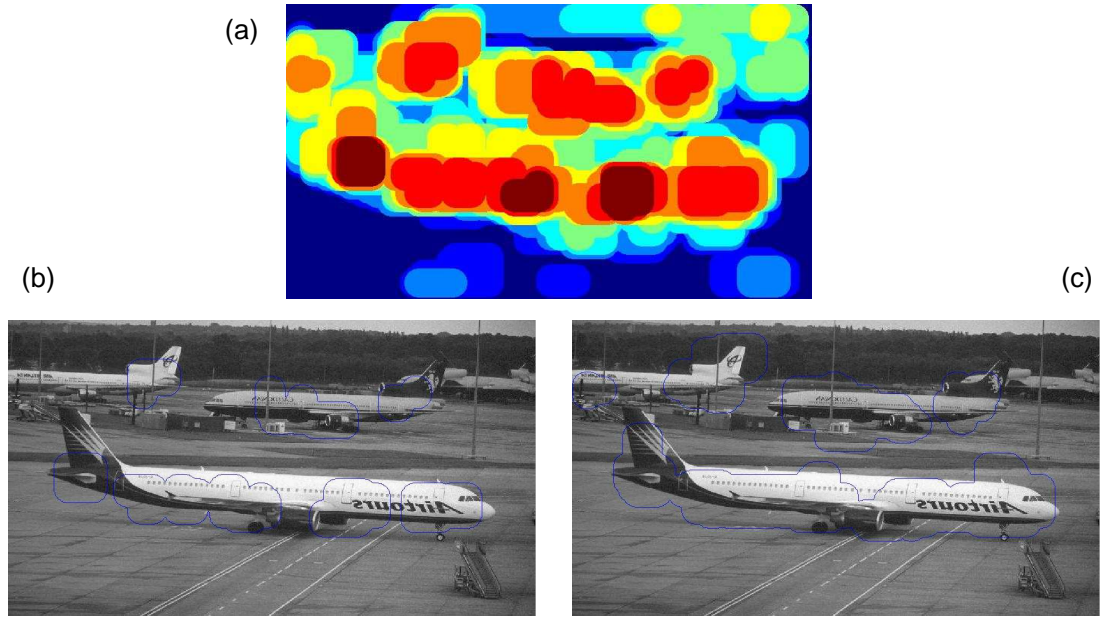


Figure 5-0 Carte de richesse (a), superposition de l'image de test avec les frontières de la carte de richesse, avec seuil = 5 et 6 respectivement sur la carte de richesse (respectivement (b) et (c)). Les trois avions sont détectés, mais sur la totalité de leur surface. Le fait que les masques « débordent » des appareils est dû au fait que les contours font partie des imagerie prises en compte lors de l'apprentissage.

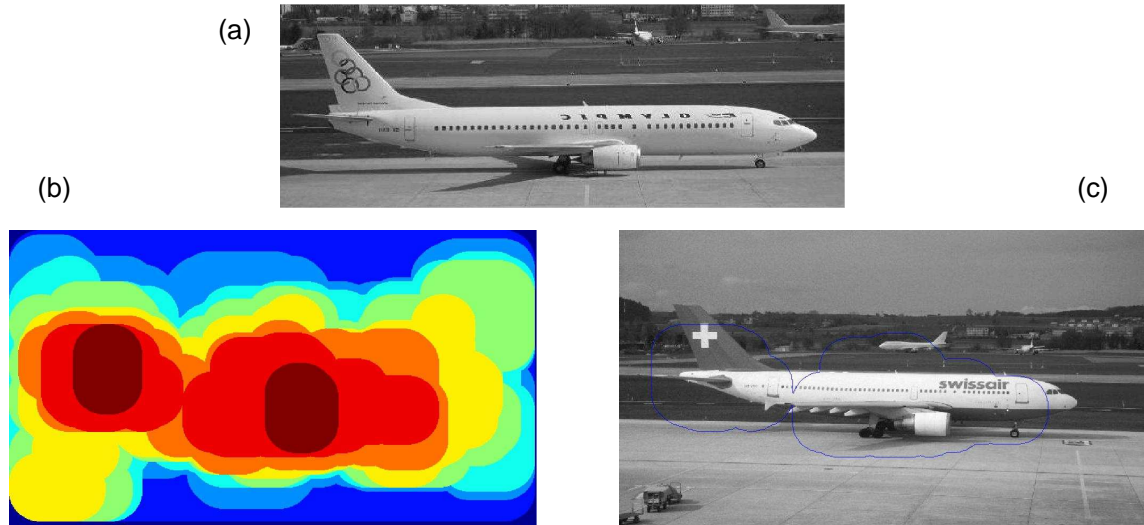


Figure 5-0 Apprentissage (a) et reconnaissance sur un appareil civils au sol (c). Le masque utilisé pour le calcul de la richesse (b) est relativement grand. Le rouge vif correspond à la valeur seuillée de la richesse, 6 imagettes représentatives sur 8 apprises.

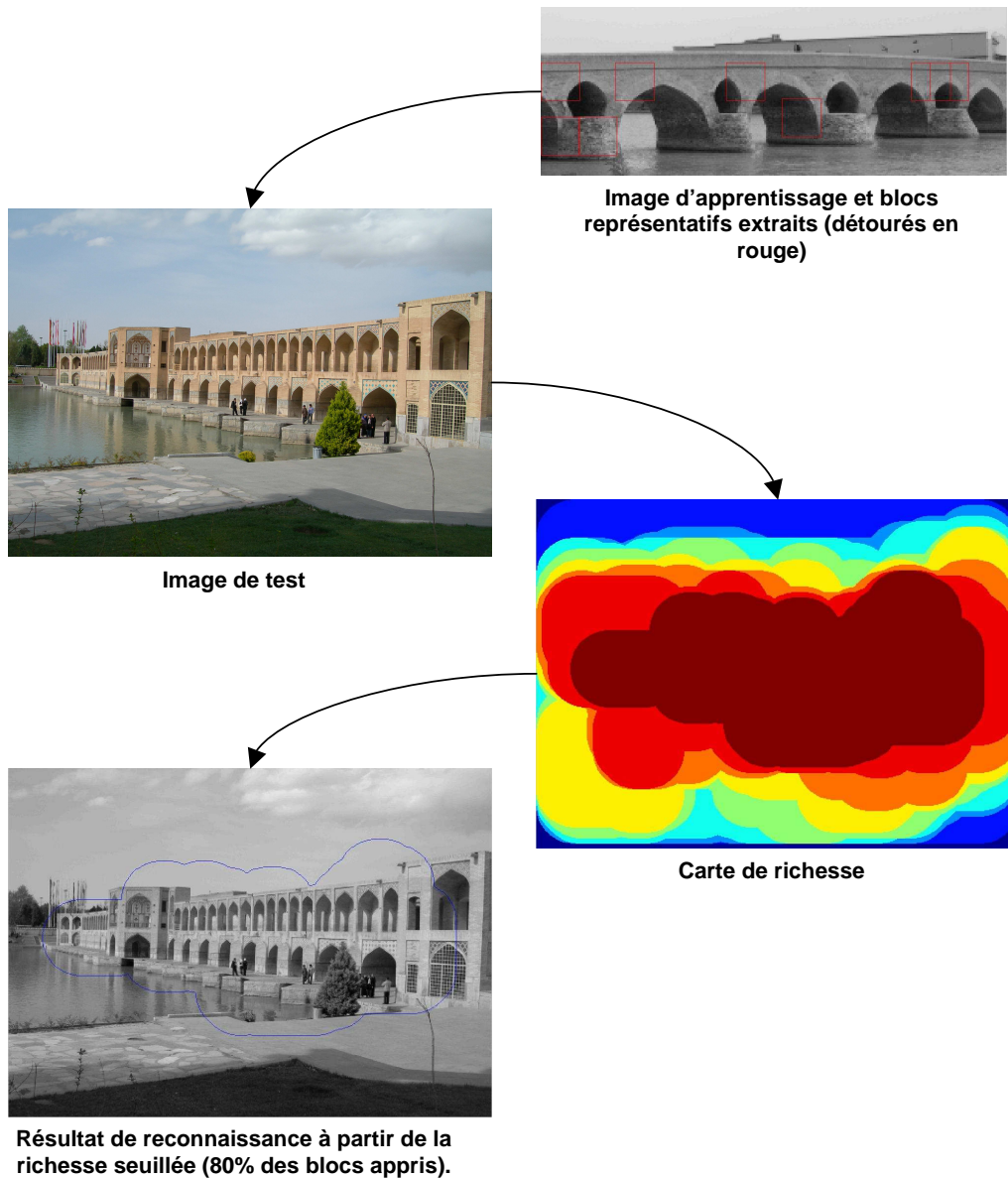


Figure 5-0 Extraction de blocs redondants et application de la méthode de reconnaissance sur des images de ponts (Ispahan, Iran).

Limite sur le paramètre de richesse

Bien que le calcul du paramètre de richesse soit aisé à mettre en œuvre, l'interprétation de son résultat reste limitée quant à la précision spatiale de la réponse (dans le repère image) et relativement à la complexité potentielle des objets traités. Ce paramètre représente uniquement le nombre d'images différentes par unité de surface. Cet opérateur peut s'avérer inapproprié dans certaines situations. Prenons le cas que l'on

peut considérer comme idéal, où l'apprentissage d'un avion de ligne nous renvoie les imageries suivantes : bout d'aile, hublots des passagers, réacteur, nez de l'appareil.

On peut supposer qu'un avion, de la queue au nez, serait représenté par une suite d'imageries : aile (pour l'aileron arrière), passagers, ... passagers, aile, réacteur, passagers, ... passagers, nez. L'opérateur de richesse renverrait alors une valeur élevée au centre de l'appareil (au niveau du réacteur) mais des valeurs faibles dans les zones « passagers, ... passagers ». Nous avons alors deux problèmes. Le premier est que la valeur élevée de l'opérateur est dans ce cas uniquement due à la structure même de l'objet, et non pas à la fiabilité de la méthode. Le second est que les zones ne comportant que des imageries « passagers » auraient une valeur de richesse faible. Une reconstruction locale juste se retrouverait alors pénalisée.

Dans le cas pratique, cette saillance de la richesse au niveau du réacteur apparaît effectivement sur la carte de richesse. Cependant dans le cas pratique, plus de 4 imageries sont nécessaires pour reconstruire un avion, la carte de richesse est donc exploitable. Néanmoins, on doit être conscient de cette limitation de l'opérateur de richesse. Il semble qu'une approche structurelle sur les imageries permettrait d'une part d'avoir un masque plus précis autour de l'objet à reconnaître, et permettrait d'autre part une diminution des fausses détections. Ce point sera détaillé dans les perspectives [toto REF chapitre sur les perspectives].

Pour extraire les imageries caractéristiques d'un ou plusieurs objets, la méthode reste inchangée. Il suffit de placer tous les objets dans une seule image et de lancer l'apprentissage par compression fractale. Les redondances seront alors mesurées sur l'ensemble des ranges et domaines de l'image contenant toutes les instances des objets à apprendre.

En changeant le nombre d'avions utilisés dans notre apprentissage nous avons obtenu 8 imageries caractéristiques pour un apprentissage sur un avion, des imageries de rayon 5 7 9 pixels, une matrice d'erreur de reconstruction clusterisée sur 8 groupes (une liste de 8 seuils) et une contrainte sur la surface apprise de 80%. La Figure 5-0 illustre cet apprentissage. Nous avons lancé un apprentissage dans les mêmes conditions, hormis le fait qu'il se faisait cette fois-ci sur 3 objets à la place d'un seul. Nous sommes alors passé de 8 imageries extraites à 7 imageries. La Figure 5-0 affiche les imageries extraites, la surface qu'elles couvrent ainsi que l'évolution de cette dernière. Dans les deux figures, dans le graphe d'évolution de la surface apprise, la ligne horizontale rouge représente la limite des 80% minimum imposés.

Pour évaluer notre méthode nous nous appuyons sur la base d'images publiques Airplanes [Net 4]. Nous avons ajouté une vérité terrain à chaque image, la base ne proposant par défaut que la boîte englobante de l'avion en premier plan. Cela pose plusieurs problèmes. D'une part on ne dispose que d'un seul avion référencé dans la vérité terrain alors que l'image peut en contenir plusieurs. Une méthode détectant tous les avions de l'image se trouverait pénalisée par cette vérité terrain, les avions détectés hors de la boîte englobante étant dans ce cas considérés comme fausses détections. D'autre part, la boîte englobante, notamment à cause de la queue de l'avion, est bien plus grande que l'objet qu'elle détecte. L'avion ne recouvre sur certaines images que 30% de la surface de la boîte englobante. Pour ces raisons, nous avons créé notre propre vérité terrain. Celle-ci consiste en un masque détournant précisément chaque avion présent dans la scène. Notre but sera de quantifier l'efficacité de notre méthode. Nous nous appuyerons pour cela sur les mesures de rappel et précision du masque de reconnaissance de notre méthode, relativement au masque de la vérité terrain.

De part la conception de notre approche, en particulier la carte de richesse (toto limitation de la carte de richesse), il ne sera la plupart du temps pas possible de recouvrir complètement le masque de la vérité terrain. Le paramètre de rappel ne sera donc pas très significatif. L'apport se situe essentiellement dans la précision de la réponse,

relativement à un contexte complexe. En effet, nous ne souhaitons pas détecter les véhicules et bâtiments qui se trouvent à proximité de l'avion.

Rappelons les définitions des notions de rappel et précision. Les masques issus de la reconnaissance et de la vérité terrain sont respectivement représentés par les variables : *masqueDetection* et *masqueVT*. L'opérateur de somme sur un masque renvoie le nombre d'éléments positifs du masque en question.

$$Precision = \frac{\sum masqueDetection \& masqueVT}{\sum masqueDetection}$$

$$Rappel = \frac{\sum masqueDetection \& masqueVT}{\sum masqueVT}$$

Le masque de détection est dans notre cas la carte des contrastes et richesses locales seuillée. La carte de contraste et richesse est le produit élément par élément de la carte de contraste et de la carte de richesse. Elle met en relief les zones où plusieurs imageries d'une même base d'apprentissage sont utilisées et où ces imageries ne sont pas « aplaties » par un contraste faible, relativement à l'image d'origine (stratégie appliquée déjà lors de l'apprentissage).

Pour calculer de manière automatique le seuil le plus approprié pour notre phase de reconnaissance, nous allons tracer les courbes rappel/précision sur un intervalle donné de seuils (classiquement 0 à 2, avec un pas de 0,1). Considérant les courbes rappel/précision sur l'ensemble des images testées, nous retiendrons le seuil qui maximise la précision.

Lors de nos premiers résultats de tests, nous avons constaté qu'une partie des cas de non détection était due à la valeur du seuil par rapport à l'intervalle [min max] des valeurs de la carte contraste/richeesse. C'est-à-dire que les blobs de valeurs importantes sur cette carte étaient bien positionnés sur les objets à détecter mais le seuil utilisé n'était pas compris dans l'intervalle [min max] des valeurs de la carte contraste/richeesse. Un seuillage relatif à la valeur maximale de la carte permettrait donc de détecter le ou les objets non détectés. Pour cette raison, nous avons également testé une détection sur une carte contraste/richeesse normalisée. Utiliser une carte normalisée, dans ce contexte, équivaut à inclure le prior « au moins un avion est présent dans la scène » (il y aura toujours une zone de la carte valant 1, ce qui est le seuil maximum) et à calculer sa localisation sur l'image. On se situe alors dans un contexte plus contraint que dans le cadre de la carte non normalisée, sans connaissance a priori.

Nous allons effectuer un ensemble de tests pour étudier le comportement de notre méthode (apprentissage et reconnaissance) en fonction de différents paramètres. Pour chaque test, calculerons les courbes de rappel et précision sur les détection avec et sans prior sur la présence d'un avion dans l'image, c'est-à-dire avec et sans normalisation de la carte contraste/richeesse. Un test est référencé par son identifiant et contient les conditions de tests (jeu de paramètres), les raisons de ce choix (c'est-à-dire, quel est le but précis de ce test), les résultats attendus, les résultats, et leur interprétation.

Test avec apprentissage sur 1 avion

- **numéro du test : 1**

- **paramètres** : rayons des domaines 9 11 13 pixels, 80% de couverture, 1 avion dans l'apprentissage.
- **raison du choix de ces paramètres** : les 80% permettent de s'assurer que nous n'allons pas extraire des éléments marginaux (cf. chapitre 5.4.3 sur l'extraction de points marginaux). Nous testerons par la suite un apprentissage sur plusieurs avions et nous comparerons les résultats des détections issues de ces deux apprentissages.
- **résultats attendus** : N/A. Premier test, première estimation des performances de la méthode.
- **les résultats** : cf. Figure 5-0.

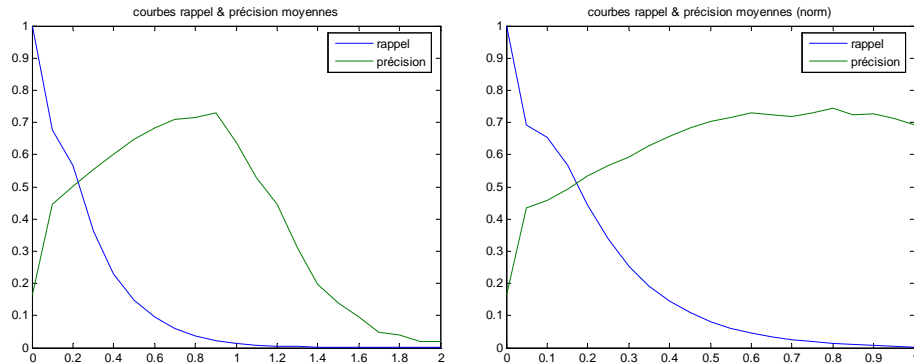


Figure 5-0 Courbe rappel précision sur carte contraste/richeesse classique (à gauche) et normalisée (à droite). Apprentissage sur 1 avion.

- **interprétations des résultats** : Les deux courbes donnent des précisions maximale sensiblement identiques 72%. La courbe de précision non normalisée chute à partir d'un seuil de 1 car il est trop élevé pour certaines images. Il n'y a alors aucune détection. Les courbes présentées étant des courbes moyennes, la valeur diminue lorsque le seuil devient trop contraignant pour un nombre important d'images testées. Pour un seul avion appris, les performances de détection et de localisation sont les mêmes avec et sans prior sur la présence éventuelle de l'objet dans la scène.

Test avec apprentissage sur 3 avions

- **numéro du test** : 2
- **paramètres** : idem test 1 mais apprentissage sur 3 avions.
- **raison du choix de ces paramètres** : Nous voulons mettre en évidence l'apport de l'apprentissage et vérifier par exemple qu'on ne passe pas de 10 à 30 imagettes en apprenant respectivement 1 et 3 avions. Le nombre d'imagettes est sensé rester assez stable.
- **résultats attendus** : nombre de blocs extraits proche de celui de l'apprentissage du test 1.
- **les résultats** : Les résultats de ce test sont représentés dans la Figure 5-0 pour la partie apprentissage et dans la Figure 5-0 pour la partie reconnaissance.

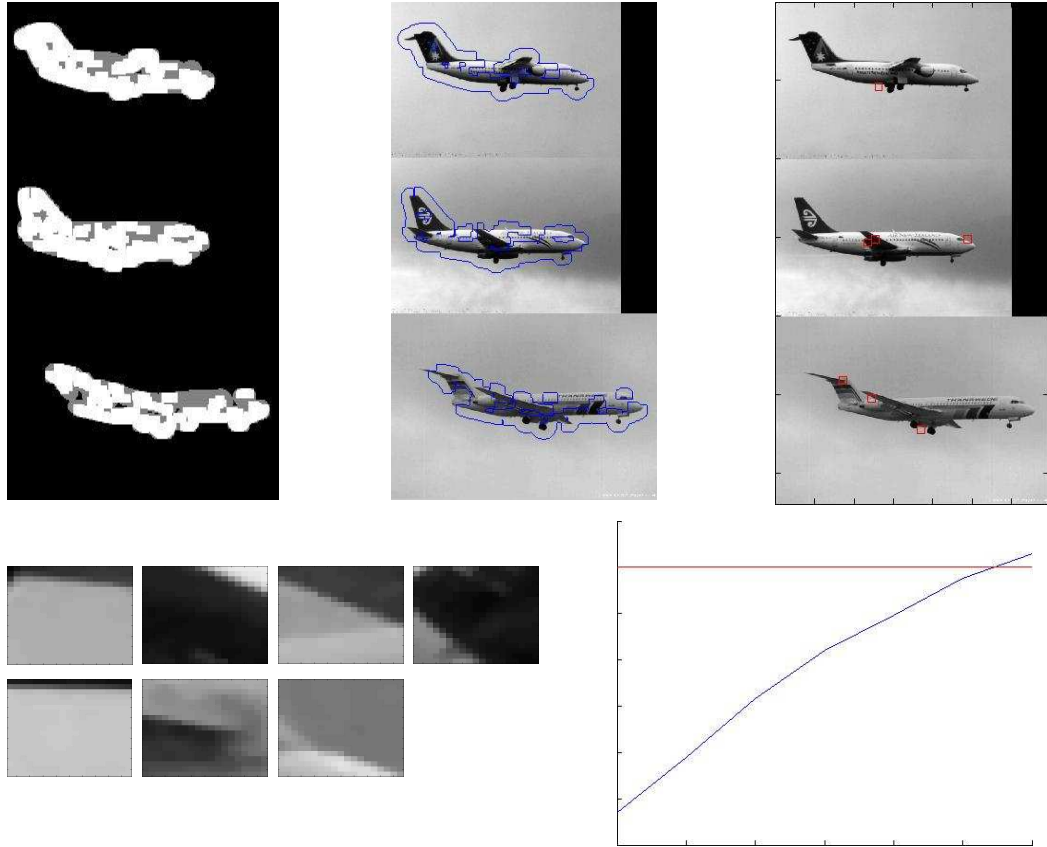


Figure 5-0 Apprentissage sur 3 avions, avec des domaines de rayon 9 11 13 pixels. De haut en bas et de gauche à droite : masque de la surface codée par les imagerie extraites, contour de ce masque sur l'image d'apprentissage, localisation des imagerie apprises, les imagerie zoomées, et la courbe de croissance de la surface codée à chaque nouvelle imagerie extraite (la ligne rouge représente le seuil imposé à 80%).

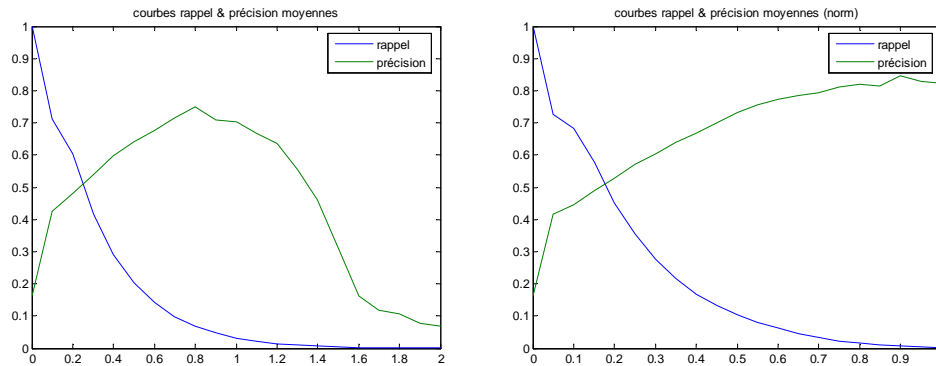


Figure 5-0 Courbes rappel & précision pour un apprentissage à 3 avions, seuillage par rapport à la carte richesse et contraste. A gauche, version non normalisée, à droite version normalisée.

- **interprétations des résultats** : Un seuil de 0,8 permet de détecter la présence d'un avion dans la scène, et de le situer, avec une probabilité de 75%. L'ajout d'un prior sur la présence d'un avion dans la scène permet de passer de 75% de « bonnes détections+localisations » à 85% de « bonnes localisations ».

Test avec apprentissage sur 3 avions, 95% de couverture

- **numéro du test** : 3
- **paramètres** : idem test 1 sauf 95% de couverture (à la place de 80%)
- **raison du choix de ces paramètres** : Une surface trop importante impliquera l'apprentissage éventuel d'images marginales (pas ou peu de similarités dans l'image). Nous voulons tester l'impact de l'augmentation de la surface apprise sur l'apprentissage et sur la phase de reconnaissance.
- **résultats attendus** : Nous nous attendons, lors de l'apprentissage, à extraire plus d'images pour couvrir la surface imposée. Lors de la reconnaissance, un apprentissage d'images plus spécifiques peut entraîner une baisse du seuil de détection maximale. En effet, comme nous rencontrerons probablement moins souvent les quelques images supplémentaires apprises, la richesse moyenne peut chuter. Les images supplémentaires ne permettraient d'obtenir localement 100% de richesse uniquement si nous trouvons le même avion, avec les mêmes logos par exemple.
- **les résultats** : Les résultats de l'apprentissage sont représentés dans la Figure 5-0, ceux de la reconnaissance dans la Figure 5-0.
- **interprétations des résultats** : Pour la partie apprentissage, nous constatons une augmentation du nombre d'images apprises. Nous sommes passé de 7 à 11 images entre 80% et 95% de surface apprise. Concernant la partie reconnaissance, dans le cas sans prior de présence de l'objet, le seuil est passé de 0,8 à 0,5, mais la probabilité de détection reste sensiblement la même. Lors de la reconnaissance avec prior, la localisation de l'avion est légèrement supérieure à la localisation de l'apprentissage à 80% de surface apprise. La précision atteinte est de 88%. L'ajout d'information par d'autres images a alors permis une meilleure discrimination entre l'objet à reconnaître et le fond.

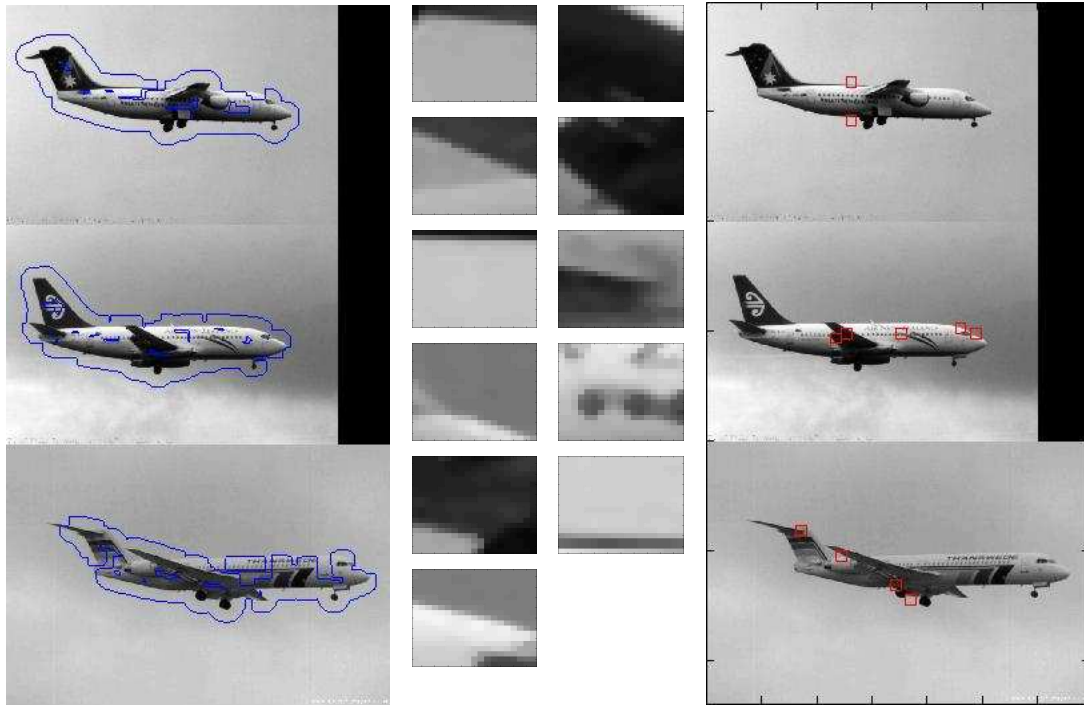
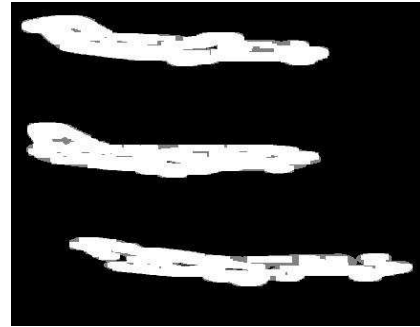
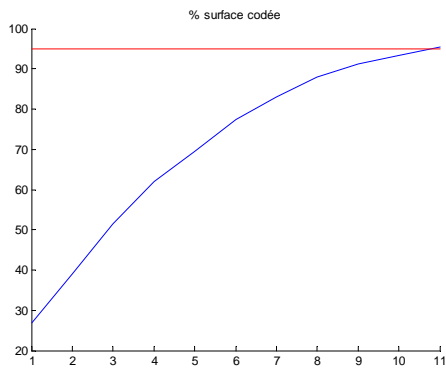


Figure 5-0 Courbe d'évolution de la surface couverte en fonction du nombre d'imagettes extraites

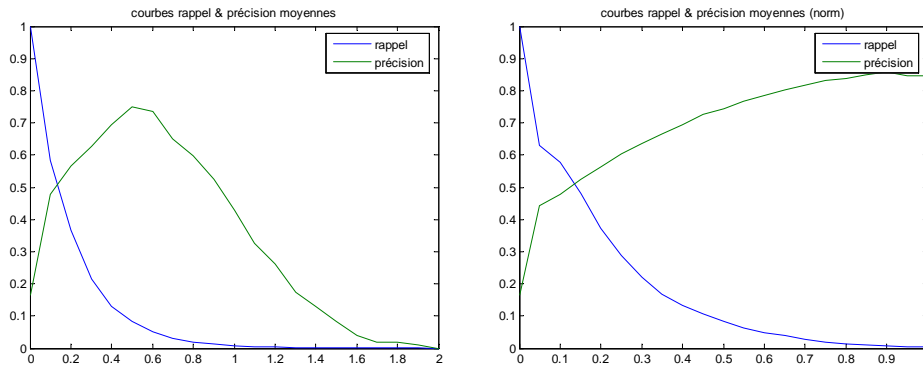


Figure 5-0 Courbes rappel et précision en classique et normalisé.

Apprentissage avec 13 15 17, 80% de surface couverte

- **numéro du test** : 4
- **paramètres** : idem test 1 sauf rayons des domaines 13 15 17
- **raison du choix de ces paramètres** : nous voulons tester si des imagettes d'une taille importante nuisent à la reconnaissance. Il y a un seuil où l'imagette contiendra trop d'information et où elle deviendra trop spécialisée par rapport à l'objet appris, sa posture, etc.
- **résultats attendus** : .

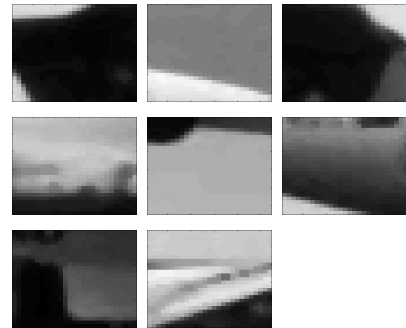
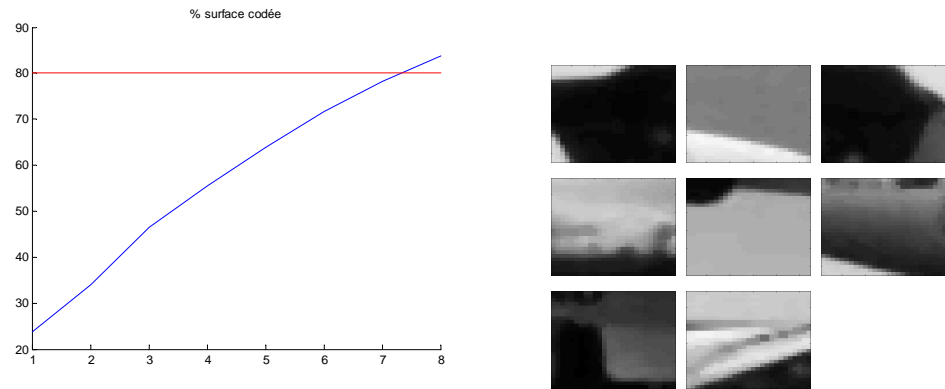
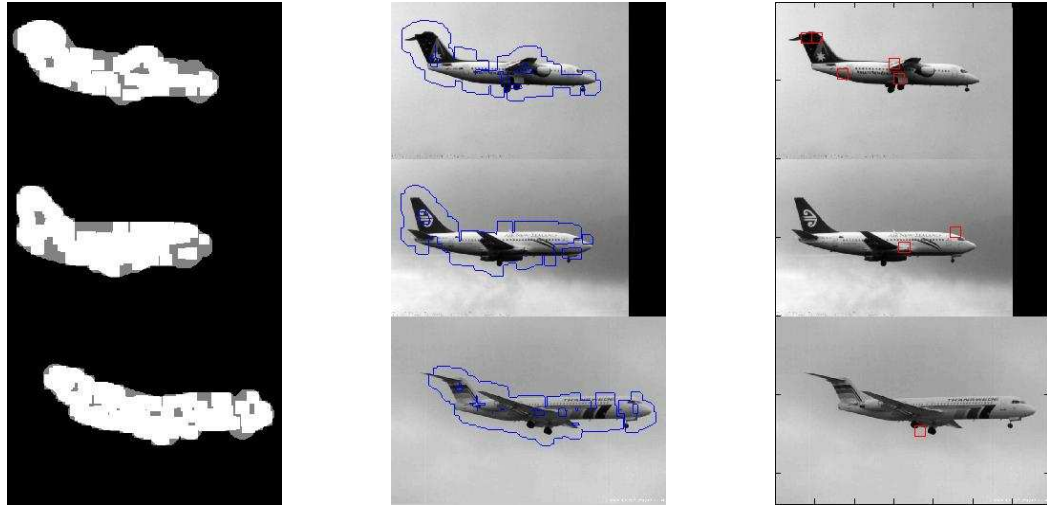


Figure 5-0 Apprentissage avec 13 15 17, 80% de surface couverte

Apprentissage 5 7 9, 80% :

- **numéro du test** : 5
- **paramètres** : idem test 1 sauf rayons des domaines 5 7 9
- **raison du choix de ces paramètres** .:
- **résultats attendus** .:
- **les résultats** : cf. courbes (Figure 5-0)

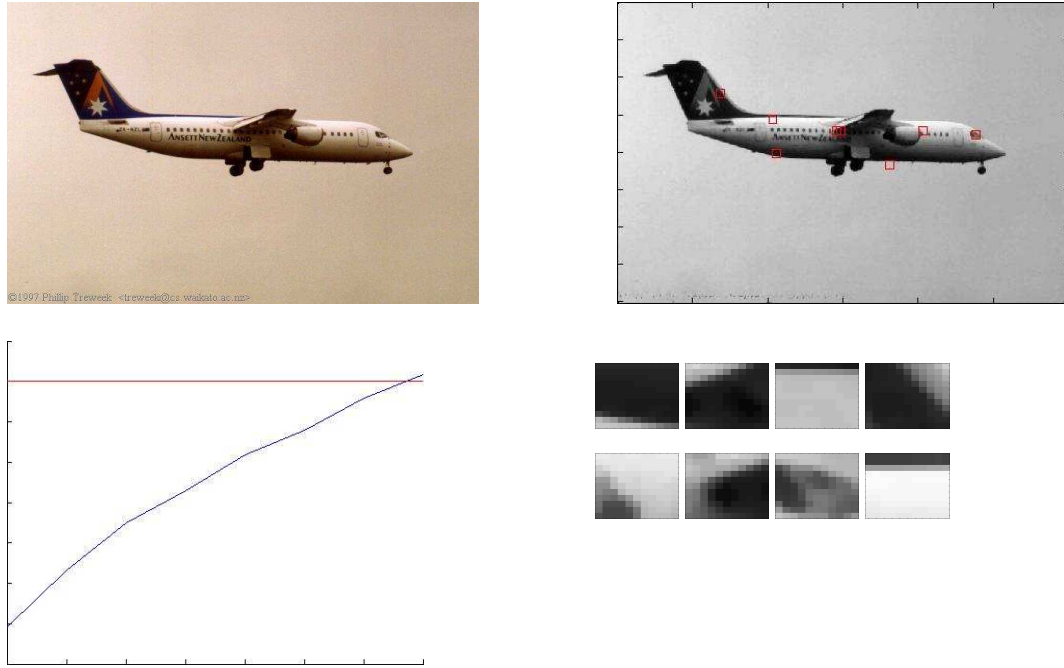


Figure 5-0 Apprentissage d'un avion, rayon = [5 7 9], 255 domaines traités par range appris, listes de 8 seuils, 8 domaines extraits pour 80% de couverture.

- **interprétations des résultats :**

Apprentissage 5 à 13, 80% :

- **numéro du test :** 6
- **paramètres :** idem test 1 sauf rayons des domaines : 5 à 13 pixels
- **raison du choix de ces paramètres :** nous voulons savoir si le fait de prendre l'intervalle d'échelle 5-13 donne un résultat meilleur ou moins bon que les bases d'apprentissage 5-9 et 9-13 séparées.
- **résultats attendus :** pour un objet parfaitement autosimilaire, la reconnaissance devrait être meilleure.

les résultats : Les résultats de la partie apprentissage sont regroupés dans la Figure 5-0, ceux de la partie reconnaissance dans la Figure 5-0.

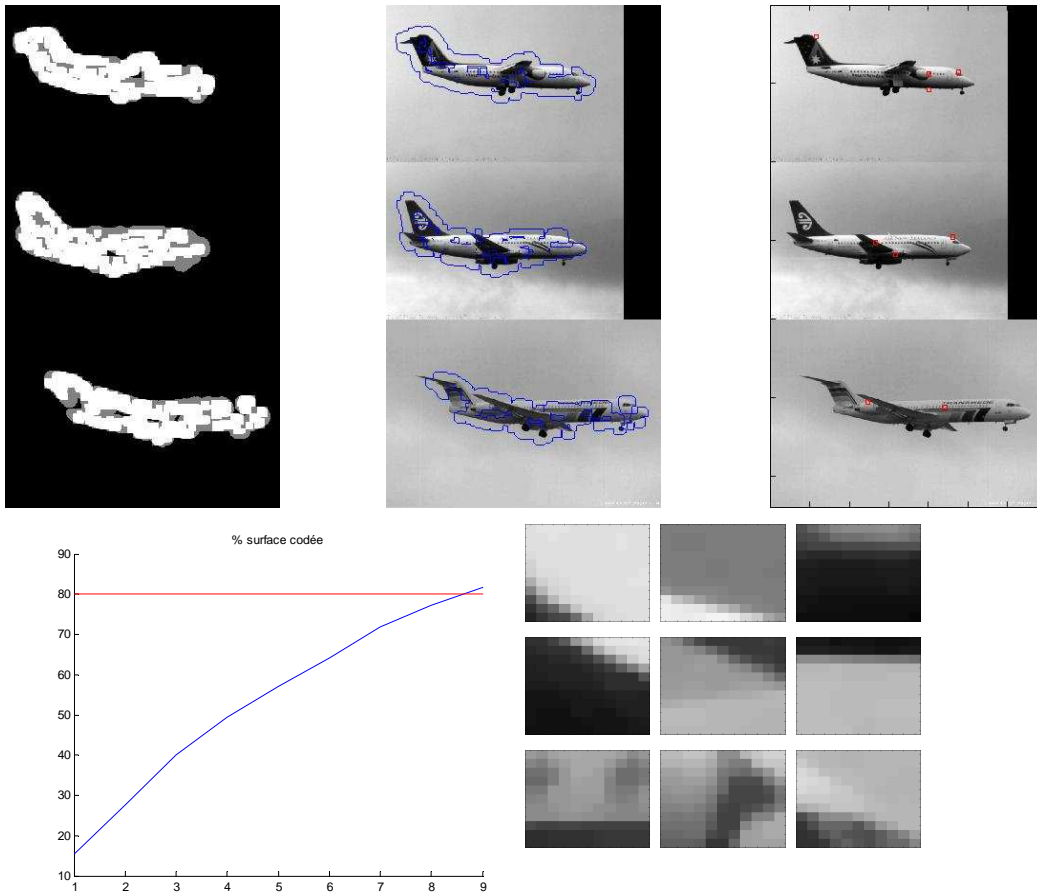


Figure 5-0 Apprentissage 5 à 13, 80%

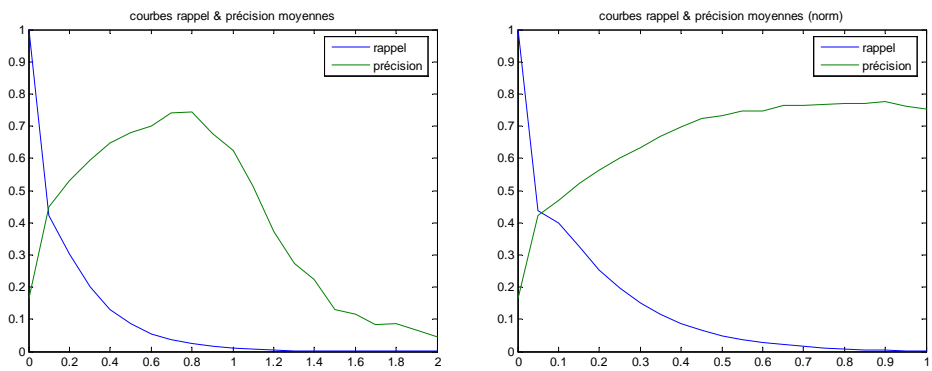


Figure 5-0 Evaluation pour 5 à 13

- **interprétations des résultats** : L'apprentissage sélectionne un nombre d'images proche de ceux vus précédemment. La partie détection (carte non normalisée) renvoie des valeurs identiques aux apprentissages précédents. Par contre la partie localisation (carte normalisée) n'est pas meilleure que celle sans prior. Notre interprétation est que pour un Intervalle d'échelles très grand, il y a beaucoup de réponses (matching possible entre des zones de l'image de test et les images utilisées) qui faussent partiellement la carte de richesse. Pour être robuste il faudrait soit être plus contraignant (intervalle d'échelles plus resserré) soit prendre en compte les échelles des images les unes relativement aux autres (dans l'apprentissage et la reconnaissance). On se rapproche à nouveau de l'aspect structurel (cf. dans le chapitre 5.4.2).

5.4.3 Extraction de points marginaux

Dans l'apprentissage que nous avons mis en œuvre dans le chapitre 4, les blocs représentatifs sont extraits du concept image à apprendre. Nous rappelons que la représentativité de ces blocs est définie par leur redondance dans l'image, relativement à différents critères (échelle, isométrie, contraste, luminosité et seuils par rapport à une mesure d'erreur de reconstruction). Une liste de seuils, définie en 4.3.4, a permis la prise en compte d'un compromis entre la qualité de reconstruction locale de chaque bloc image et la surface apprise/codée de l'image. A chaque fois que nous passons d'un seuil de la liste à son suivant, nous extrayons le bloc qui a le score le plus élevé. Nous avons choisi de tester une variante de notre approche en sélectionnant les blocs de scores les plus bas. Ceux-là représentent les zones de l'image les plus « marginales », ou « uniques » au sens des invariants et seuils définis lors de l'apprentissage. La Figure 5-0 illustre le résultat d'extraction de blocs image marginaux sur des images de la base « airplanes ». Les zones ici mises en relief sont le logo, le nom de la compagnie aérienne, et le train d'atterrissage de l'appareil.



Figure 5-0 Extraction des blocs les plus marginaux sur une image de la base « airplanes ». Ces blocs image sont uniques relativement aux invariants utilisés par la compression fractale et à la liste de seuils d'erreurs de reconstruction. Ici ils représentent le nom de la compagnie, le logo et le train avant de l'appareil.



Figure 5-0 Extraction des blocs marginaux sur le concorde. Le premier est le logo à l'arrière de l'appareil, puis viennent les blocs contenant le nom de la compagnie, sur l'avant, et enfin une partie du train d'atterrissage.

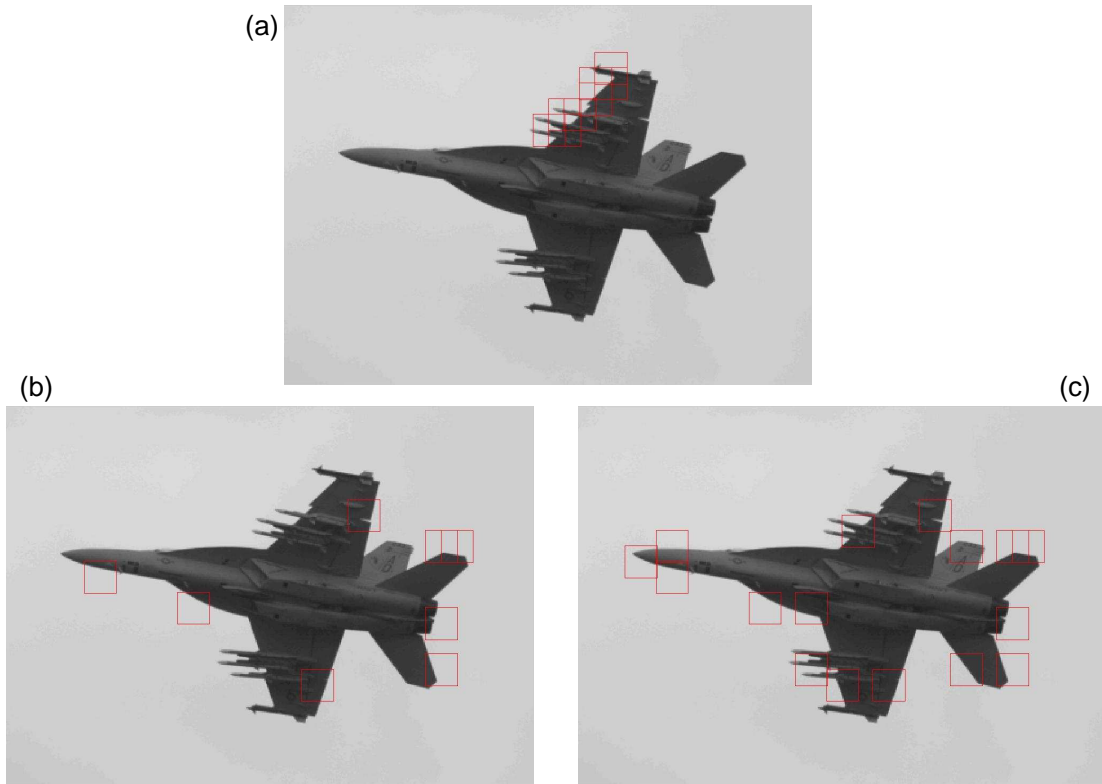


Figure 5-0 Extraction des blocs marginaux sur un F18 en (a). Les systèmes d'armes ne sont présents qu'une seule fois sur chaque aile de l'appareil et à une seule échelle. En (b) apprentissage/extraction des 8 imageries (et 8 clusters) les plus redondantes. En (c), extraction des imageries sur 8 clusters et avec contrainte sur la surface apprise minimum à 80%. Pour atteindre une telle surface, l'apprentissage spécifique des systèmes d'armes est nécessaire.

Dans ce test où nous avons utilisé notre méthode à contrario de son usage initial, à savoir extraire les blocs les plus marginaux plutôt que les plus redondants. Cette approche s'est révélée intéressante puisqu'elle a mis en relief des éléments conceptuellement saillants sur les objets observés, tels que les logos ou les systèmes d'armes. Nous noterons que ces blocs marginaux pourraient également servir à étayer l'identification. Tout d'abord, lorsqu'un objet à identifier est présenté, nous pourrions dans un premier temps, mettre à part les blocs marginaux, de manière à éviter des matchings des éléments tels que des logos ou les noms de compagnies aériennes. Dans un second temps, une fois que l'objet, un avion en l'occurrence, a été identifié, nous pourrions nous servir des points marginaux pour affiner la réponse et par exemple lancer une reconnaissance de logo ou un OCR pour dire à quelle compagnie l'avion appartient. Concrètement, la première réponse serait pour la Figure 5-0 « cet avion est un concorde », puis « ce concorde appartient à la British Airways ». Ce point n'a pas été implémenté et fera l'objet d'éventuelles perspectives.

L'extraction de points marginaux n'a pas fait l'objet de tests poussés. Il s'agit essentiellement de constats intéressants sur le comportement implicite de la méthode plutôt que d'un outil en soi.

5.4.4 Conclusion de l'évaluation

Dans cette section, nous avons testé l'efficacité de notre approche sur des images issues d'une base publique. Les contraintes sur l'opérateur de richesse (cf. chapitre 5.4.2) ne permettent pas l'obtention d'un masque recouvrant la totalité de l'objet à reconnaître. Une application spécifique à la reconnaissance d'avions de ligne obtiendrait probablement des résultats supérieurs à ceux que nous avons obtenus. Cependant nous avons pu constater une précision de 75% pour la détection et localisation de l'objet sans prior sur la scène et allant jusqu'à 88% avec prior dans des contextes images parfois complexes (présence de bâtiments et véhicules). Les masques obtenus en sortie de détection sont peu recouvrant par rapport à la vérité terrain (faible rappel lors des tests). Ceci est dû aux caractéristiques intrinsèques de la carte de richesse. La présence simultanée de plusieurs imagerie différentes, issues de l'apprentissage d'un même concept, participe à la détection et localisation de l'objet dans un environnement complexe, mais le revers de cet atout est que tout concept ne produit pas une richesse élevée en tout point de son image. La mesure de richesse locale sur des avions de lignes est élevée dans les zones présentant à la fois une partie de fuselage et d'aile, ou de queue de l'appareil. Les parties ne présentant que du fuselage vont utiliser peu d'imagerie et donc produire une richesse locale plus faible. Pour recouvrir tout l'avion, une approche plus évoluée serait nécessaire, comme par exemple celle utilisée dans [BUH07]. Notre méthode pourrait se situer en amont de méthodes telles que [BUH07] et leur fournir les graines pour la construction du graphe d'imagerie de l'objet à reconnaître. Enfin nous avons testé l'extraction de blocs marginaux. Ces blocs possèdent un score faible relativement aux similarités présentes dans l'image. Nous avons pu constater que ces blocs étaient situés essentiellement pour les avions de ligne à l'emplacement des logos, ou des systèmes d'armes pour les avions militaires. Nous reparlerons de ces blocs marginaux dans nos perspectives. Ils pourront en effet servir dans une future application d'identification, où une fois un avion détecté, on souhaite savoir plus spécifique à quelle catégorie ou compagnie il appartient.

5.5 Conclusion

Dans cette section, nous nous sommes intéressé à l'application des résultats de notre apprentissage lors de la phase de reconnaissance. Des approches préliminaires nous ont montré leurs limites et ont mis en évidence l'importance de la prise en compte de la richesse des imagerie utilisées pour la reconstruction d'un concept appris. Cette reconnaissance a été testée sur des imagerie d'avions et de ponts. Les masques obtenus ne recouvrent pas complètement l'objet, mais la méthode a montré une certaine robustesse quant au contexte complexe de l'objet dans l'image. Nous avons ensuite abordé une optimisation de notre méthode. L'optimisation est un module applicable à l'apprentissage et à la reconnaissance. Plusieurs variantes peuvent être appliquées à ce module d'optimisation. Une des approches testées permettait par exemple de prendre ou non en compte le recouvrement entre les domaines déjà traités d'une image, tout en conservant un tri sur les domaines les plus pertinents relativement aux imagerie caractéristiques extraites. Ce type d'ordonnement a permis d'obtenir un bon compromis entre la qualité des appariements effectués et la quantité de surface traitée sur l'image étudiée. Nous ajouterons que l'architecture de la méthode implémentée permet une modification locale de chaque module sans pour autant changer le déroulement global de la méthode. De plus dans le cadre d'une application embarquée et de matériel amené à évoluer en terme de puissance, ce type d'optimisation repose sur un parcours qui se veut exhaustif à l'origine mais dont les réponses les plus pertinentes sont placées en tête de file. Le jour où l'utilisateur fait évoluer son matériel, l'application et ses paramètres restent inchangés. Pour un même temps de calcul la réponse sera plus précise, ou une précision de reconnaissance équivalente pourra être obtenue en un temps plus court. Une partie de l'optimisation s'appuie comme nous venons de le voir sur l'ordonnement des couples d'imagerie, une partie additionnelle qui peut intervenir en amont de la méthode, est la séparation de l'image en naturel/artificiel. Cette tâche n'est pas aisée. Nous avons utilisé une méthode simple s'appuyant conjointement sur des opérateurs fractals et morphologiques. Lors de la phase d'évaluation nous avons quantifié les performances de notre méthode et classifié les cas d'erreurs. Les tests ont été effectués à partir d'une base d'images publiques. Des résultats intéressants ont été obtenus dans des contextes complexes, compte tenu de l'aspect non dédié de la méthode. Nous avons également testé l'extraction de blocs marginaux. Ceux-ci mettent en relief les parties d'un concept qui sont les moins redondants sur les images qui le représentent. Ces zones marginales pourraient permettre par la suite de passer d'une phase de détection à une phase d'identification des objets localisés sur l'image et ainsi extraire une information plus pointue de la scène. Ce point sera discuté dans les perspectives.

6 Conclusion et perspectives

6.1 Conclusion

L'objectif de cette thèse est, d'une part, pour l'industriel, d'avoir une veille sur les méthodes actuelles de traitement de la texture et des objets fortement résolus, d'autre part de proposer une chaîne algorithmique complète d'apprentissage/reconnaissance. Un état de l'art sur la texture et les outils fractionnaires nous a permis de situer l'existant, l'historique de ces domaines et de leurs champs d'applications. Nous nous sommes ensuite appuyé sur la compression fractale et les IFS pour mettre en œuvre une méthode originale d'apprentissage d'imagettes caractéristiques d'un concept.

Après un état de l'art sur la labellisation, nous avons proposé une méthode originale pour synthétiser l'information contenue dans une image. Notre approche extrait les imagettes caractéristiques de l'image considérée. Cette caractérisation est quantifiée par la redondance des imagettes présentes, et ce à différentes transformations près. Nous avons pu constater quelles imagettes extraites permettaient de représenter l'image de manière compacte, tant en terme de surface que de qualité. Les majeurs problèmes techniques rencontrés se sont trouvés au niveau de l'optimisation et de la comparaison entre imagettes. Ces dernières contenant peu d'information, la ressemblance apparente n'est pas toujours triviale à retranscrire formellement. Notre méthode faisant appel à un certain nombre de traitements, nous l'avons mise en œuvre en insistant sur l'aspect modulaire. Cela nous a permis d'insérer des outils qui se sont avérés indispensables dans le traitement de concepts de grande taille, tel que le module d'optimisation. Celui-ci fait le choix de privilégier le traitement des imagettes qui ont a priori la plus grande probabilité de correspondre. Initialement prévu pour accélérer l'apprentissage, il a également été appliqué à la phase de reconnaissance et d'identification. Tout en restant exhaustive sur une durée suffisamment longue, la version optimisée de notre méthode fournit alors plus tôt une réponse pertinente, comparée à une méthode de parcours stochastique des couples d'imagettes. La reconnaissance a montré une relative robustesse dans des environnements complexes. Nous avons pu noter que certains masques lors de la reconnaissance pouvaient être sensiblement plus larges qu'une segmentation parfaite de l'objet, sur la reconnaissance d'avion par exemple. Cependant la méthode mise en œuvre n'est pas liée à une application en particulier, comme nous l'avons montré en l'appliquant à une reconnaissance de ponts.

6.2 Perspectives

6.2.1 Streaming vidéo

Une lacune connue de la compression fractale est sa relative lenteur par rapport à des compressions type jpeg. A l'inverse, sa décompression est très rapide. Elle n'implique aucune opération trigonométrique et l'image converge à chaque itération vers l'image à reconstruire, si bien que si on exécute un streaming vidéo basé sur la compression fractale sur une machine de faible puissance, si cette dernière n'a pas le temps d'effectuer toutes les itérations avant la prochaine image de la vidéo, elle peut suspendre le calcul de l'image actuelle et fournir le résultat tel quel. Une machine de faible puissance pourra donc visionner la vidéo, seule la qualité sera limitée de manière implicite, mais pas la fluidité de la vidéo.

Dans le cas où la puissance viendrait encore à manquer, il serait possible de traiter une trame sur 2 par exemple. La convergence des fractales assurerait une interpolation jusqu'à l'image suivante, ce qui visuellement serait acceptable et qui serait cohérent par rapport au contenu de la vidéo.

Pourquoi il n'y aurait pas besoin de beaucoup d'itérations : parce que d'une image à l'autre, la plupart du temps, il y a très peu de changement. Si la scène est statique, une machine lente pourra profiter de la scène fixe (ou de portions fixes) pour améliorer la qualité de l'image au fil des trames.

Il serait également possible de reprendre le principe de keyframe, et de ne mettre à jour que les parties qui en ont besoin.

Déplacer la suite dans les perspectives ou supprimer. toto

L'émetteur peut ou non envoyer une image d'origine (*key frame*). Dans les deux cas le client peut reconstruire l'image, seulement, plus il aura en entrée une image proche de l'image à reconstruire, moins il aura besoin de faire d'itérations. On a donc un système qui pourrait naturellement s'adapter à la vitesse de traitement du client : un client qui peut faire 30 itérations entre chaque image pourra avoir une meilleure reconstruction qu'un client ne faisant que 5 itérations, mais dans les deux cas une information visuelle de base est disponible, seule la qualité changera.

Lors de la compression, à chaque nouvelle image à traiter, on pourrait se servir de l'image précédente et par défaut garder les correspondances ranges/domaines précédentes. Si ces correspondances sont suffisamment bonnes (erreur de reconstruction inférieure à un seuil fixé a priori) alors on ne recalcule pas le matching pour ce range.

6.2.2 Optimisation sur la matrice de reconstruction

Le remplissage complet de la matrice de reconstruction correspond à un parcours exhaustif des couples range/domaine. Nous avons vu au chapitre 5.3.4 une méthode d'optimisation qui avait pour but de concentrer le calcul uniquement sur les couples les plus proches en termes de vecteur d'attributs. Ceci aboutissait à un remplissage partiel de la matrice de reconstruction. Nous rappelons que la matrice sert dans un premier temps à calculer une liste de seuils. Puis ces seuils étaient appliqués aux erreurs de reconstruction pour calculer les scores de chaque imagerie représentative. Nous rappelons également la forme des boucles de calcul de correspondance range/domaine. Elles étaient du type « pour chaque domaine, tester les transformations » dans laquelle était incluse une boucle « pour chaque range, faire le test de matching ».

Supposons que nous disposons de la liste de seuils a priori, nous serions alors en mesure de « bloquer » un range pour un ou plusieurs domaines appartenant à un même cluster.

Supposons également, que nous disposons d'une mesure permettant de parcourir pour un range, les domaines par erreur croissante présumée.

Nous serions alors en mesure de calculer les imagerie représentatives au fur et à mesure du calcul de la matrice de reconstruction. Qui plus est, chaque range parcouru et participant au score d'un domaine pourrait être « bloqué » (chapitre 4.3.3) pour les autres domaines et ainsi, plus le parcours avancerait, moins il y aurait de calcul pour chaque domaine. Nous serions alors en mesure d'avoir des performances d'extraction proche d'un parcours exhaustif, en consommant bien moins de temps de calcul.

Pour arriver à ce résultat, nous devons vérifier les deux suppositions que nous avons énoncées plus haut.

La seconde est réalisable en s'appuyant sur les détails fournis au chapitre 5.3.4, Figure 2-0, où nous montrons qu'une mesure statistique en imagerie (léger en temps de calcul) permet d'avoir des informations sur l'erreur de reconstruction réelle d'une première imagerie par une seconde (calcul gourmand).

La première supposition peut être respectée en partant de l'hypothèse qu'un parcours d'une portion des couples ranges domaines permet d'avoir une estimation de l'intervalle [min max] de l'erreur de reconstruction ainsi qu'une estimation de sa répartition sur l'ensemble des couples. Les générations de liste de seuils par approche linéaire ou histogramme cumulé restent applicables.

En suivant cette approche, moyennant certaines hypothèses nous pourrions tendre vers une extraction d'imagerie représentatives proche de celle effectuées par un parcours exhaustif des couples range/domaine.

6.2.3 Adaptation de l'apprentissage aux zones naturelles

Nous avons pu constater que notre méthode d'apprentissage par compression fractale était surtout appropriée aux zones artificielles, contenant des objets géométriques. Dans le cas de régions naturelles, un matching sur des attributs de textures (fractionnaires par exemple) plutôt que sur les imagerie elles-mêmes serait plus approprié, et surtout bien plus rapide. Pour ce faire, nous devrions utiliser une méthode de séparation naturel/artificiel, éventuellement celle détaillée au chapitre 5.2, l'opérateur de matching (par imagerie ou par attributs) serait alors adapté en fonction de l'appartenance de la zone à apprendre à la classe naturelle ou artificielle. Dans le cas où la zone à apprendre contiendrait du naturel et de l'artificiel, nous aurions à la séparer en deux sous-zones. Deux compressions distinctes seraient alors appliquées.

6.2.4 Identification d'objets

Nous avons vu précédemment que la prise en compte de l'aspect composite d'un concept était un paramètre profitable à l'élimination des fausses détections. Jusqu'à présent, cette utilisation ne s'est faite que par l'intermédiaire d'une carte de richesse. C'est-à-dire que la réponse locale de la richesse est un scalaire. Nous avons évoqué lors de l'usage de zones connexes dans le paragraphe 5.2.2.1, que les statistiques d'apparition pourraient être un atout pour une éventuelle phase d'identification. C'est ce point que nous allons détailler dans cette partie.

Nous nous plaçons dans le cas où nous avons en entrée une zone déjà segmentée et dont nous voulons identifier le contenu. Par exemple, nous savons quand dans un masque binaire extrait d'une phase de détection, nous avons un avion, et nous voulons savoir de quel type d'avion il s'agit. Une telle phase de détection peut être effectuée, soit par un algorithme extérieur, dédié à l'application. Soit par la méthode que nous avons développée en 5.2.2.

Notre phase d'identification est basée sur les taux d'apparition des imagerie dans la zone à traiter. Une base d'imagerie caractéristiques est un ensemble de mots et nous

considérons que nous avons identifié le concept appris si nous avons, dans la zone segmentée, les mêmes mots et les mêmes taux d'apparition. Notons qu'il ne s'agit pas ici d'une définition de grammaire à proprement parlé puisqu'il n'y a pas de règles d'enchaînement des mots. Ici, il n'y a pas (encore) de structure sur les mots, seulement un comptage de leur présence. Nous verrons plus loin les limites que cette méthode et détaillerons des contraintes et avantages d'une méthode avec prise en compte de l'aspect structurel de l'objet appris.

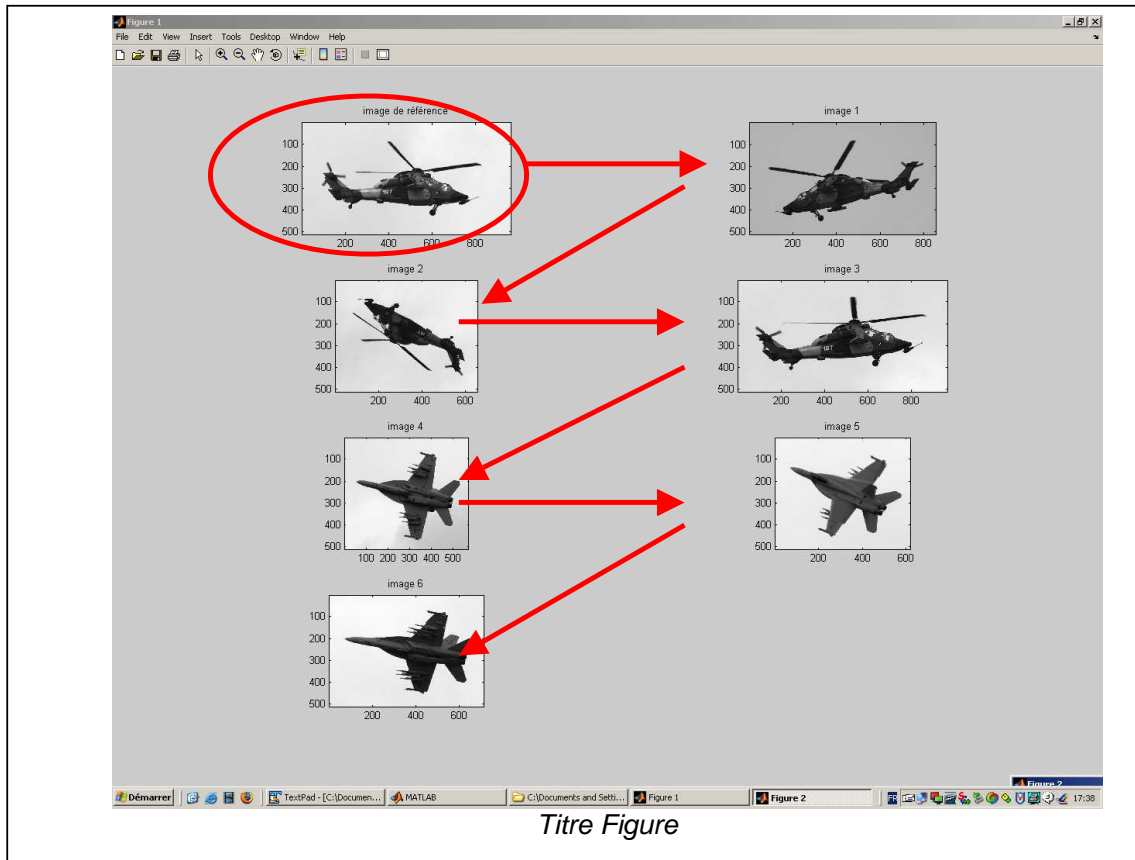
Nous disposons donc en entrée, d'un masque détournant l'objet à identifier, et des indices des imageries de utilisées dans ce masque, un indice par pixel image. Cette carte nous servira pour calculer les statistiques sur nos imageries. Nous pouvons calculer les taux d'apparition soit sur les imageries entières, nous ne comptabilisons sur la carte d'indices de pixels que les emplacement où il y a de la place pour toute l'imagerie ou un pourcentage suffisant de sa surface, soit en extrayant une information au niveau pixel sur la carte d'indice.

L'apprentissage prenant en compte la surface apprise au niveau pixel pour chaque bloc image, nous avons choisi la seconde approche. L'histogramme des indices des imageries est établi à l'intérieur du masque et sur l'image apprise. L'identification équivaut alors à une estimation de distance entre histogrammes. Le retour de l'identification est le concept appris étant le plus proche de l'image testée, relativement aux histogrammes d'apparitions d'imageries.

En modifiant de façon mineure la méthode, nous pouvons en faire une méthode d'indexation. Considérons en entrée une base d'imageries caractéristiques et un ensemble d'images de tests. Nous reconstruisons progressivement chacune des ces images à partir de la base d'apprentissage. Pour les méthodes de reconstruction, se référer aux techniques d'optimisation présentées en 5.3.4. A intervalle de temps régulier, nous pouvons consulter les cartes d'imageries se référant à chaque image de tests, c'est-à-dire les imageries qui ont servi à reconstruire cette image. La reconstruction converge et se stabilise après un parcours suffisant des domaines de l'image. La convergence de cette reconstruction se justifie en considérant que la carte d'indices des imageries n'est mise à jour que sur les pixels présentant une erreur de reconstruction supérieure à l'erreur de reconstruction actuelle.

Pour chaque image de test nous avons donc un vecteur de statistiques d'apparition des imageries de la base d'apprentissage choisie. Nous sommes alors en mesure de dire qu'une image est proche d'une autre, au sens d'une distance relative à la base choisie.

Pour illustrer ce phénomène, nous avons constitué une base d'imageries constituée à partir de l'apprentissage d'un avion de chasse. Nous avons ensuite reconstruit un ensemble d'images à partir de cette base d'imageries. Nous avons choisi une image et choisi d'afficher les images par distance croissante.



Approche exhaustive, approche stochastique : seulement 4% des imagerie nécessaire pour identifier la cible.

Isolation du fond pour économiser du temps de calcul. Dans notre cas, il s'agira d'avions et hélicoptères (Bourget 2007).

7 Bibliographie

- Net 1 : <http://www.journaldunet.com/science/science-et-nous/diaporamas/06/fractales/7.shtml>
- Net 2 : <http://www.cs.princeton.edu/introcs/22library/flaming-fractal.jpg>
- Net 3 : <http://www.mathcurve.com/fractals/fougere/fougere.shtml>
- Net 4 : <http://www.vision.caltech.edu/html-files/archive.html>
- Net 5 : http://www.math-info.univ-paris5.fr/~dolez/these/base_images/
- Net 6 : <http://www.cybergeu.eu/index8002.html?lang=fr>
- [ABD03] C. BenAbdelkadery, R. Cutlerz, and L. S. Davis, "Gait Recognition Using Image Self-Similarity", EURASIP Journal on Applied Signal Processing, pp. 1–14, 2004
- [ADA94] R. Adams, L. Bishof, "Seeded Region Growing". IEEE Trans. on PAMI, 16, pp. 641-647, 1994.
- [ALL06] Brian S. Allen, and [B50]E. David Jansing, "One dimensional fractal error for motion detection in an image sequence", Defense & Security Symposium, Orlando Florida conference 6234, SPIE, 2006.
- [BAE95] Y.J. Bae, M.C. Fairhurst, "Parallelism in dynamic time warping for automatic signature verification", Third International Conference on Document Analysis and Recognition (ICDAR), Vol. 1, p. 426, 1995.
- [BAR88] M. Barnsley, R. Devaney, B.B. Mandelbrot, H-O Peitgen, D. Saupe, R.F. Voss, "The Science of Fractal Images", Peitgen Saupe Editors, ISBN : 0-387-96608-0, 3-540-96608-0, 1988.
- [BAR94] K.U. Barthel, J. Schüttemeyer, T. Voyé and P. Noll, "A New Image Coding Technique Unifying Fractal and Transform Coding", IEEE International Conference on Image Processing (ICIP 94), 1994.
- [BEN90] J. L. Bentley, "K-d Trees for Semidynamic Point Sets", Proc. 6th Annual Symposium on Computational Geometry (SCG), pp. 187–197, 1990.
- [BER96] P. Bernaola-Galvan, R. Roldan, J. L. Oliver, "Compositional segmentation and long-range fractal correlations in DNA sequences", The American Physical Society, Physical Review E, Vol. 53, N°5, 1996.

- [BHA04] B. Bhanu, X. Zhou, "Face Recognition from Face Profile Using Dynamic Time Warping", 17th International Conference on Pattern Recognition (ICPR) – Vol. 4, pp. 499-502, 2004.
- [BUH07] J. M. Buhmann, "Complex Statistical Models for Object Recognition and Mass Spectrometry", Proc. Pattern Recognition in Information Systems (PRIS), pp. 3-8, 2007.
- [BUT94] G.C. Butler et al., "Fractal nature of short-term systolic BP and HR variability during lower body negative pressure". Am. J. Physiol, 1994.
- [CAR90] M. J. Carlotto, M. C. Stein, "A Method for Searching Artificial Objects on Planetary Surfaces", Journal of the British Interplanetary Society, Vol. 43, 1990.
- [CAR04] Y. Caron, Thèse, "Contribution de la loi de Zipf à l'analyse d'images", Université de Tours, 2004.
- [CHA95] B. Chaudhuri, N. Sarkar, "Texture Segmentation Using Fractal Dimension", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 17, N°1, pp.72-77, 1995.
- [COC95] J.P. Cocquerez et S. Philipp, "Analyse d'Images : filtrage et segmentation", ISBN : 2-225-84923-4, ISSN : 0992-5538, 1995.
- [CRO83] G. Cross and A. Jain, "Markov Random Field Texture Models", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 5, N°1, pp. 25-39, 1983.
- [DEL06] G. Delyon, P. Martin, P. Réfrégier, F. Guérault, "Segmentation d'images par minimisation de la complexité stochastique avec un modèle de bruit non paramétrique", Reconnaissance des Formes et Intelligence Artificielle (RFIA), 2006.
- [DIS06] R. Distasi, M. Nappi and D. Riccio, "A Range / Domain Approximation Error-Based Approach for Fractal Image Compression", IEEE Transactions on Image Processing, Vol 15, N°1, 2006.
- [DON00] P. Dong, "Test of a new lacunarity estimation method for image texture analysis", International Journal of Remote Sensing, Vol. 21, N°17, pp. 3369 – 3373, 2000.
- [ESP98] F. Espinal, T. Huntsberger, B. D. Jawerth, T. Kubota, "Wavelet-based fractal signature analysis for automatic target recognition", Society of Photo-Optical Instrumentation Engineers, Opt. Eng. 37(1), pp. 166–174, 1998.
- [EST99] R. Esteller[B51], G. Vachtsevanos, J. Echauz, T. Henry, P. Pennell, C. Epstein, R. Bakay†, C. Bowen and B. Litt, "Fractal Dimension Characterizes seizure onset in epileptic patients", International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Proc. IEEE Vol. 4, pp. 2343-2346, 1999.
- [GAL04] F. Galland, O. Germain, N. Berteaux, P. Réfrégier, « Partitionning into Homogeneous Regions Adapted to SAR Image », ESA, EUSC, 2004.

- [GHA03] M. Ghazel, G. H. Freeman, E. R. Vrscay, "Fractal Image Denoising", IEEE Transactions on Image Processing (ICIP), Vol. 12, N°12, 2003.
- [GOL02] Ary L. Goldberger, Luis A. N. Amaral, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, C.-K. Peng, and H. Eugene Stanley, « Fractal dynamics in physiology: Alterations with disease and aging », 2466–2472 PNAS vol. 99 suppl. 1, 19 Février 2002.
- [GOU07] J.F Goudou, « Apport de la Dimension Temporelle aux Traitements de Veille Infrarouge Marine », Mémoire de thèse, Télécom Paris, SAGEM DS, 2007.
- [HAG04] Caroline M. Hagerhall, Terry Purcell, Richard Taylor, « Fractal dimension of Landscape Silhouette Outlines as a Predictor of Landscape Preference », Journal of Environmental Psychology 24, 247–255, 2004.
- [HAM96] R. Hamzaoui, "Codebook clustering by self-organizing maps for fractal image compression", Institut für Informatik – Report 75, 1996.
- [HAR79] R.M. Haralick, « Statistical and structural approaches texture », Proc. IEEE, 67,5, pp. 786-804,1979.
- [HAR92] R. M. Haralick, L. G. Shapiro, "Computer and Robot Vision", Vol. 2, Addison-Wesley, pp. 316-317, 1992.
- [HAS04] M. Hassaballah, M.M. Makky, Y. B. Mahdy, "A Fast Fractal Image Compression Method Based Entropy", Electronic Letters on Computer Vision and Image Analysis 5(1):30-40, 2005.
- [HEN01] B.I. Henry, P.R. Hof, P. Rothnied, S.L. Wearned, "Fractal Analysis of Aggregates of Non-Uniformly Sized Particles : an Application to Macaque Monkey Cortical Pyramidal Neurons", World Scientific (Septembre), 2001.
- [HOL01] T. Hollerer, S. Feiner, D. Hallaway, B. Bell, M. Lanzagorta, D. Brown, S. Julier, Y. Baillot, L. Rosenblum, "User Interface Management Techniques for Collaborative Mobile Augmented Reality", Computers & Graphics 25, pp. 799–810, 2001.
- [HOR76] S. Horowitz, et T. Pavlidis, "Picture Segmentation by a Tree Traversal Algorithm", J. Ass. Comp. Mach, 23(2), pp. 368-388.
- [HSU02] H. T. Chang, "Gradient Match and Side Match Fractal Vector Quantizers for Images", IEEE Transactions on image processing, Vol. 11, N°1, 2002.
- [JAC90] A. Jacquin, "Fractal Image Coding Based on a Theory of Iterated Contractive Image Transformations", Proc. SPIE Visual Comm. And Image Proc., pp. 227-239, 1990.
- [JAC92] A. Jacquin, "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations", IEEE Transactions on Image Processing, Vol. 1, pp. 18-30, 1992.
- [JAI88] A.K. Jain et R.C. Dubes, "Algorithms for Clustering Data", Prentice-Hall, New Jersey, 1988.

- [JAI90] A .K Jain, F. Farrokhnia, "Unsupervised Texture Segmentation Using Gabor Filters", Proc. IEEE International Conference on Systems, Man and Cybernetics, pp. 14-19, 1990.
- [JOS04] B. Josso, D. R. Burton, M. J. Lalor, "Texture orientation and anisotropy calculation by Fourier transform and Principal Component Analysis", Coherent and Electro-optics Research Group, Liverpool John Moores University, 2004.
- [JUL75] B. Julesz, "Experiments in the Visual Perception of Texture", Scientific American, 232, 4, pp. 34-43, 1975.
- [KAP95] L. Kaplan, C-C. Kuo, "Texture Roughness Analysis and Synthesis via Extended Self-Similar (ESS) Model", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 17, N°11, pp.1043-1056, 1995.
- [KEN92] K. Arakawa, E. Krotkov, "Fractal Surface Reconstruction with Uncertainty Estimation: Modelling Natural Terrain", CMU-CS-92-194, 1992.
- [LAG98] Lagarias, J.C., J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," SIAM Journal of Optimization, Vol. 9, N°1, pp. 112-147, 1998.
- [LAI03] C. M. Lai; K. M. Lam; W. C. Siu, "A fast image coding based on kick-out and zero contrast conditions", IEEE Transactions on image processing, Vol. 12, N° 11, 2003.
- [LAW79] K. I. Laws, "Texture energy measures", DARPA Image Understanding Workshop, pp. 47-51, 1979.
- [LEC89] Y. Leclerc, "Constructing simple stable descriptors for image partitioning", Int. Journal of Computer Vision, Vol. 3, pp. 73-102, 1989.
- [LEE96] S. Y. Lee, K. Y. Chwa, J. Hahn, S. Y. Shin, « Image Morphing Using Deformation Techniques », Journal on Vision and Computer Animation (JVCA), 1996.
- [LEG03] P. Legrand, J. Lévy Véhel, "Local regularity-based image denoising", Proc. International Conference on Image Processing (ICIP), Vol. 3, pp. 377-380, 2003.
- [LER93] R. Lerski, K. Straughan, L. Shad, D. Boyce, S. Bluml, I. Zuna, "MR Image Texture Analysis – An Approach to Tissue Characterisation", Magnetic Resonance Imaging, Vol. 11, pp. 873-887, 1993.
- [LEV97] J. Lévy Véhel, B. Guiheneuf, "Multifractal Image Denoising", INRIA Rocquencourt, 1997.
- [LOR97] L. M. Lorigo, R. A. Brooks, W. E. L. Grimson, "Visually-Guided Obstacle Avoidance in Unstructured Environments", International Conference on Intelligent Robots and Systems, 1997. IROS apos;97., Proceedings of the 1997 IEEE/RSJ, Vol. 1, pp. 373-379, 1997.
- [LOW99] D. G. Lowe, "Object recognition from local scale-invariant features," Proceedings of International Conference on Computer Vision, pp. 1150-1157, 1999.

- [LOW04] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, Vol. 60, N°2, pp. 91-110, 2004.
- [MAI99] H. Maître, "Le Traitement des Image", Polycopier ENST, Février1999.
- [MAN77] B. B. Mandelbrot, "Fractals: Form, Chance and Dimension", San Francisco, CA: Freeman, 1977.
- [MAN83] B. B. Mandelbrot, "The Fractal Geometry of Nature", Freeman, San Francisco, 1983.
- [MAN95] B. B. Mandelbrot, « Les Objets Fractals, Forme, Hasard et Dimension », 4ème édition, éditions Flammarion, ISBN : 2-08-081301-3, 1995.
- [MAR84] D.M. Mark, P.B. Aronson, "Scale-Dependent Fractal Dimensions of Topographic Surfaces : an Empirical Investigation with Applications in Geomorphology and Computer Mapping", Mathematical Geology, Vol. 16 N° 7, 1984.
- [MAR98] M. R. J. Maria, S. Benoît, B. Macq, "Speeding up Fractal Image Coding by Combined DCT and Kohonen Neural Net Method", International Conference on Acoustics, Speech and Signal Processing (ICASSP), Proc. IEEE, pp. 1085-1088, 1998.
- [MGA95] O. L. Mangasarian, W. N. Street, W. H. Wolberg, "Breast Cancer Diagnosis and Prognosis via Linear Programming", Computational Science & Engineering, IEEE Vol. 2, N°3, pp. 70-73, 1995.
- [NAJ06] L. Najman, M. Couprie, "Building the Component Tree in Quasi-Linear Rime", IEEE Trans. on Image Processing, Vol. 15, N°11, pp. 3531-3539, 2006.
- [NAK93] Nakamura et al., "Autonomic Control of Heart Rate During Physical Exercise and Fractal Dimension of Heart Rate Variability", J. Appl. Physiol., 1993.
- [NAR99] S. S. Narine, A. G. Marangoni, "Fractal Nature of Fat Crystal Networks", 651X(99)00502-4, PHYSICAL REVIEW, VOLUME 59, NUMBER 2, février 1999.
- [OTS79] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, N° 1, pp. 62-66, 1979.
- [RIC93] L. F. Richardson, M. Oliver, H. Charnock, P. G. Drazin, J. C. R. Hunt, P. Smoker, P. Sutherland, M. Oliver, "The Collected Papers of Lewis Fry Richardson", Vol. 1: Meteorology and numerical analysis, Cambridge University Press, ISBN 978-0521382977. Vol. 2: Quantitative psychology and studies of conflict, ISBN 978-0521382984 778pp, 1993.
- [PAR99] I. Parkinson, N. Fazzalari, "Fractal Analysis of Trabecular Bone : a Standardised Methodology", Department of Pathology, University of Adelaide, South Australia, 1999.

- [PEL84] S. Peleg[B57], J. Naor, R. Hartley, D. Avnir, "Multiple Resolution Texture Analysis and Classification". IEEE Transactions On Pattern Analysis And Machine Intelligence (PAMI), Vol. 6, N°4, 1984.
- [PEN84] A. P. Pentland. "Fractal-Based Description of Natural Scenes", IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 6, N° 6, 1984.
- [PER13] J. Perrin, "Les Atomes", 1^{ère} édition Felix Alcan, 1913. Dernière édition : Gallimard, collection Champs, ISBN : 2-08-081225-4, 1991.
- [PRI98] J. R. Pricey, M. H. Hayes, "Fractal Interpolation of Images and Volumes", IEEE Proceedings of the 32nd Asilomar Conference on Signals, Systems, and Computers, Vol. 2, pp. 1698-1702, 1998.
- [REV03] M. Revenu, "La Segmentation d'Images", Polycopié Filière Informatique & Réseaux Spécialisation Imagerie Numérique, DEA IAA, 2002-2003.
- [RIS78] J. Rissanen, "Minimum Description Length Principle", Encyclopedia of Statistical Sciences, Vol. 3, pp. 523-527.
- [ROS06] C. Rosenberger, "Contribution à l'Evaluation d'Algorithmes de Traitement d'Images", Habilitation à Diriger des Recherches de l'Université d'Orléans, 2006.
- [SAG06] R. Sagawa, Y. Makihara, T. Echigo, Y. Yagi, "Matching Gait Image Sequences in the Frequency Domain for Tracking People at a Distance", Institute of Scientific and Industrial Research, Osaka University, ACCV, 2006.
- [SAR99] H. Sari-Sarraf, S. S. Gleason, R. M. Nishikawa, "Front-End Data Reduction in Computer-Aided Diagnosis of Mammograms : A Pilot Study », University of Chicago, SPIE, 1999.
- [SEC03] N. A. Secelean, "The Fractal Interpolation for Countable Systems of Data", Univ. Beograd. Publ. Elektrotehn. Fak. Ser. Mat. 14, pp. 11–19, 2003.
- [SER03] A. Séropian, "Analyse de Document et Identification de Scripteurs", Université de Toulon et duVar, UFR Sciences et Techniques, 2003.
- [SER05] N. Vincent, A. Seropian, G. Stamon, "Synthesis for Handwriting Analysis", Pattern Recognition Letters 26, pp. 267–275, 2005.
- [SHA99] S. Majumdar, J. Lin, T. Link, J. Millard, P. Augat, X. Ouyang, D. Newitt, R. Gould, M. Kothari, H. Genant, "Fractal analysis of radiographs: Assessment of Trabecular Bone Structure and Prediction of Elastic Modulus and Strength", American Association of Physicists in Medicine, S0094-2405~99!00207-2, 1999.
- [SOA01] S. Soatto, G. Doretto, Y. Wu, "Dynamic textures", Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV), Vol. 2, pp. 439–446, 2001.
- [SOL98] J.L. Solka, D.J. Marchette, B.C. Wallet, V.L. Irwin, G.W. Rogers, "Identification of Man-Made Regions in Unmanned Aerial Vehicle Imagery and Videos",

- IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, N° 8, 1998.
- [STR95] M. Strzelecki, "Segmentation of Textured Biomedical Images Using Neural Networks", PhD Thesis, Technical University of Łódź, Poland, 1995.
- [SUM99] R. M. Summers, L. M. Pusanik, J. D. Malley, J. M. Hoeg, "Fractal Analysis of Virtual Endoscopy Reconstructions", Proc. SPIE Vol. 3660, p. 258-269, Medical Imaging: Physiology and Function from Multidimensional Images, 1999.
- [TAR75] R.E. Tarjan, "Efficiency of a good but not linear set union algorithm", Journal of the ACM, Vol. 22, pp. 215-225, 1975.
- [TOE04] Alexander Toet, "Colorizing Grayscale Intensified Nightvision Images", Proc. of the Seventh International Conference on Information Fusion, Vol. 2, pp. 1066-1071, 2004.
- [TUC94] M. Tuceryan, "Moment Based Texture Segmentation", Pattern Recognition Letters, Vol. 15, pp. 659-668, 1994.
- [VAL98] K. Valkealathi, E. Oja, "Reduced Multidimensional Cooccurrence Histograms in Texture Classification", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 20, N° 1, pp. 90-94, 1998.
- [VEN04] V. Venema, C. Simmer, S. Crewell, "Iterative and Constrained Algorithms to Gcloud Fields with Measured Properties", Proc. of European Geosciences Union, Vol. 6, 2004.
- [VIT06] J. Vitay, "Emergence de Fonctions par l'Intéraction de Champs Neuronaux", NEUROSciences COMPUtationnelles, Vol. 1, 2006.
- [WAE07] S. de Waele, M. J. Verberne, "Coding Gain and Tuning for Parametrized Visual Quality Metrics", IEEE International Conference on Image Processing (ICIP), Vol. 6, pp. 461-464, 2007.
- [WAN00] W. Fan[B59], Z. Fang, W. Wenhui, "A Self-Adapting Endpoint Detection Algorithm for Speech Recognition in Noisy Environments Based on 1/f Process", International Symposium on Chinese Spoken Language Processing, pp. 327-330, 2000.
- [WAN04] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, "Image Quality Assessment: from Error Visibility to Structural Similarity," IEEE Transactions on Image Processing, Vol. 13, N°4, pp. 600-612, 20 04.
- [WIE66] N. Wiener, "Nonlinear Problems in Random Theory", Cambridge, MIT Press, 1966.
- [WIT95] C. M. Wittenbrink, "IFS Fractal interpolation for 2D and 3D Visualization", Proc. of IEEE Conference on Visualization, Vol. 6, pp. 77-84, ISBN:0-8186-7187-4, 1995.
- [WOH95] B.E. Wohlberg et G de Jager, "Fast Image Domain Fractal Compression by DCT Domain Block Matching", Electronics Letters , Vol. 31, pp. 869-870, 1995.

- [WOLXX] W. Schleifer[B60], M. Mannle, "Error Detection Based on the Self Similar Behavior of Network Traffic", Communications, IEEE Proceedings, Vol. 148, N° 1, pp. 38 – 42, 2001.
- [YIN97] Y. Zhang, Y. L. Yu, L. M. Po, "Wavelet Transform Based Variable Tree Size Fractal Video Coding", International Conference on Image Processing (ICIP), Vol.2, pp. 293 – 297, 1997.
- [YLI02] Y. Li, L. G. Shapiro, "Consistent Line Clusters for Building Recognition in CBIR", Proc. 16th International Conference on Pattern Recognition, Vol. 3, pp. 952-956, ISBN: 0-7695-1695-X, 2002.
- [ZHA97] Y. Zhang, L. M. Po, Y. L. Yu, "Wavelet Transform Based Variable Tree Size Fractal Video Coding", Proc. of IEEE International Conference on Image Processing (ICIP), 1997.
- [ZHE99] L. Zheng, A. K. Chan, G. McCord, S. Wu, J. S. Liu, "Detection of Cancerous Masses for Screening Mammography using DWT Based Multiresolution Markov Random Field", Scientific Committee on Antarctic Research (SCAR), 1999.
- [ZHU96] S. Zhu, A. Yuille, "Region Competition : Unifying Snakes, Region Growing and Bayes / MDL for Multiband Image Segmentation", IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), Vol. 18, pp. 884-900, 1996.

8 D's C : 3.4.7 Apport dans le domaine des multi fractales

Méthode par approximation et calcul global sur toute une image :

Si nous devons faire un nombre conséquent d'extractions de courbes multi-fractales locales sur une image par une fenêtre glissante, nous pouvons être amenés à effectuer des calculs redondants. En effet, si nous voulons analyser des zones polygonales quelconques, ou plus classiquement des fenêtres glissantes, il peut arriver que certaines dilations/érosions soient calculées plusieurs fois. Une approximation de l'estimation des courbes multi fractales peut être faite en calculant d'abord u et b sur toute l'image, puis après de calculer A et S en appliquant un masque sur u et b de la région correspondante sur l'image. Le souci principal est qu'au fur et à mesure des opérations morphologiques un extremum local peut voir sa valeur influencer des pixels de la zone étudiée, alors que lui-même n'est pas dans la zone. Dans le cas d'une étude sur n niveaux de dilatation/érosions, un extremum local modifiera les valeurs locales de u et b dans un voisinage de pixels alors que ce pixel correspondant à cet extremum peut ne pas se trouver dans la zone à étudier. Cet extremum, au niveau de dilatation/érosion le plus élevé, influencera les valeurs des pixels situés dans un rayon de n pixels autour de lui. La Figure 8-0 illustre ce type de phénomène. Le disque grisé représente la zone d'influence de l'extremum. Ce dernier se trouve à l'extérieur de la zone bleutée à étudier. Pourtant, une partie est commune à la région bleue et à la zone d'influence. Dans cette intersection, le calcul du volume entre les surfaces u et b sera faussé. L'étendue de cette erreur de calcul dépend de plusieurs paramètres. Le premier est le nombre de niveaux d'opérations morphologiques demandé. Plus ce niveau sera faible, moins un extremum local pourra étendre sa valeur sur un voisinage important. Ensuite, intervient la taille de la région commune par rapport à la région étudiée. Plus l'intersection est importante relativement à la zone d'étude, plus l'erreur a d'impact sur le résultat final. Intervient également la présence d'un autre extremum local situé dans la région à étudier. Si un pixel possède une valeur plus extrême que celui qui est externe à la zone, alors il peut prendre la main sur les pixels appartenant au voisinage commun de ces deux extrema locaux. La partie des pixels situés dans le voisinage de l'extremum interne à la zone ne sera alors pas faussée. De plus, la pondération des niveaux situés en fin de courbe pour les valeurs les plus élevées du ε (le log à la fin de la définition de la distance), permet d'atténuer une partie de la propagation de l'erreur. Enfin d'après [toto REF sur multi fractale entre deux textures] nous savons que la signature multi fractale entre deux textures, est la moyenne leurs signatures. Si nous utilisons une fenêtre glissante d'une texture à une autre, et qu'à chaque pas nous calculons la signature multi fractale du contenu de la fenêtre, alors nous observons un glissement continu des courbes de la première signature vers la seconde.

Nous venons de voir qu'il était possible de supprimer les redondances lors de multiples extractions de courbes multi fractales sur une image. La caractérisation de la perte de précision en calcul est difficile à formaliser puisqu'elle dépend d'un certain nombre de paramètres, non indépendants, et fortement variables. Globalement pour des régions de taille élevée par rapport au nombre de niveaux ε demandé (nous rappelons qu' ε et le nombre d'opérations morphologiques sont liés), le gain en temps de calcul est très significatif pour une imprécision mineure. Le facteur de gain en temps de calcul dépend

des recouvrements des zones. Pour une fenêtre glissante se déplaçant de 1 pixel à chaque pas, ce facteur est égal au nombre de pixels de la fenêtre (!).

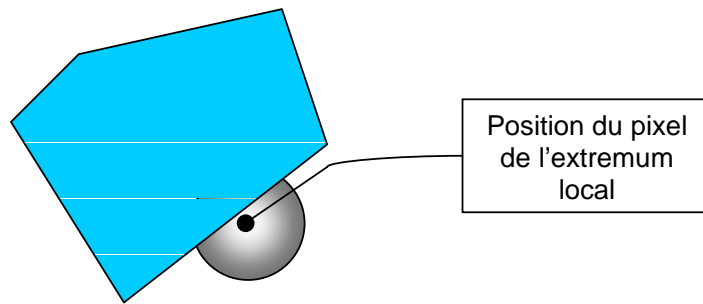


Figure 8-0 Exemple d'influence d'un maximum local

9 Ancienne évaluation

Détail de la procédure d'évaluation

Pour évaluer l'efficacité de notre reconnaissance, nous avons utilisé la vérité terrain fournie avec la base publique Airplanes. Cette vérité terrain se présente sous la forme d'une boîte englobante encadrant l'objet le plus en évidence sur l'image (c'est-à-dire en premier plan, ou au centre de l'image). Nous allouons un temps constant pour chaque image. Les couples d'imagette entre l'image à tester et les imagettes apprises sont alors traités. Nous calculons la carte de richesse et générons les blobs correspondant au seuillage de la carte de richesse par sa valeur maximale sur l'image. Si le barycentre du blob de plus grande taille, en nombre de pixels, est dans la boîte englobante alors la localisation est considérée comme valide, et non valide sinon. Dans le cas de mauvaise localisation, nous classerons les erreurs en différentes catégories pour les analyser et en retirer les lacunes ponctuelles de la chaîne algorithmique.

Nous avons testé les performances de notre méthode en changeant la taille des domaines utilisés lors de l'apprentissage et de la phase de reconnaissance. Le Tableau 9-0 récapitule les performances obtenues avec les différents paramètres testés. Celles-ci sont légèrement supérieures avec des imagettes de taille 9 11 13 pixels de rayon que pour les imagettes de rayon 5 6 9.

Nous avons également classé les erreurs en différentes catégories de manière à comprendre les points qui pourraient être améliorés dans notre approche (**toto ce point n'existe plus dans la nouvelle évaluation**). Dans la vérité terrain fournie avec la base Airplane, un seul avion est répertorié par image alors que cette dernière peut en comporter plusieurs. Or dans notre méthode de localisation nous choisissons de ne retenir que l'objet qui renvoie la richesse maximale et dont le blob correspondant à cette valeur maximale est le plus grand en nombre de pixels. Il pourra arriver dans la phase de reconnaissance que l'objet trouvé soit effectivement un avion mais pas essentiellement celui qui se trouve dans la vérité terrain. Ces résultats seront regroupés dans la colonne « autre avion ». Un cas plus pénalisant est celui où nous avons une mauvaise détection, l'objet trouvé n'est pas un avion (c'est cet objet qui a produit le plus grand blob de richesse maximale), mais plusieurs petits blobs de richesse maximale sont contenus dans l'avion à trouver. Il s'agit alors d'un problème classique de connexité de blobs, ou de taille la fenêtre servant à générer la carte de richesse. Nous rappelons qu'une petite fenêtre produira des localisations plus fines spatialement de l'objet à détecter, par contre elle sera plus sensiblement aux fausses détections puisque la richesse sera à chaque fois calculée sur des zones très locales. L'aspect global et composite de l'objet peut alors ne pas être pleinement exploité. Enfin la dernière catégorie d'erreur est le pire cas de détection : l'objet détecté n'est pas un avion et il n'y a pas de bobs dans aucun avion de l'image.

Taille Domaines	Bonne localisation	Autre avion	Plusieurs blobs dans avion	Dans les choux
5 7 9	---%	--%	--%	---%
9 11 13	---%	--%	--%	---%

Tableau 9-0 Récapitulatif des évaluations pour un apprentissage sur un seul avion

Taille Domaines	Bonne localisation	Autre avion	Plusieurs blobs dans avion	Dans les choux
5 7 9	83.3%	30%	30%	40%
9 11 13	87.4%	30%	30%	40%

Tableau 9-0 Tableau récapitulatif des évaluations pour un apprentissage sur 3 avions

Avec cette méthode d'évaluation, les détection sont meilleurs avec un seul avion qu'avec 3... aie aie aie