

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari MOUMEDIENE
(USTHB)

Faculté d'Electronique et d'Informatique



MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER

En : ELECTRONIQUE

Spécialité : Traitement du Signal et des Images

Par : **BENELKADI, Née BOUTELDJA Samia**

Sujet

***Analyse, segmentation, classification de textures
basées sur les modèles autorégressifs***

Soutenu le 29/04/ 2008, devant le jury composé de :

Y. SMARA	Professeur	USTHB	Président
A. OUKIL	Maître de Conférence	USTHB	Directeur de thèse
T. IFTENE	Maître de Recherche	ASAL	Examineur
S. LARABI	Maître de conférences	USTHB	Examineur

Table des matières

TABLE DES MATIERES.....	I
LISTE DES FIGURES	III
LISTE DES TABLEAUX.....	IV
INTRODUCTION GENERALE.....	1
CHAPITRE 1 TEXTURE ET ANALYSE TEXTURALE.....	3
I.1 NOTION DE TEXTURE.....	4
I.1.1 DEFINITION DE LA TEXTURE.	4
I.1.2 TYPES DE TEXTURES.....	4
I.1.3 ATTRIBUTS VISUELS DE LA TEXTURE	5
I.1.4 INVARIANCE PAR TRANSLATION	6
I.2 APPROCHES D'ANALYSE DE TEXTURE.....	7
I.2.1 APPROCHES STATISTIQUES.....	7
<i>Méthode de matrice de cooccurrence</i>	7
I.2.2 LES APPROCHES STRUCTURALES.....	8
<i>La morphologie mathématique.....</i>	8
<i>Méthode de longueurs de plages.....</i>	9
I.2.3 APPROCHES PAR MODELISATION	10
I.2.3.1 <i>Modèle fractal.....</i>	10
I.2.3.2 <i>Modèle Markovien.....</i>	11
I.2.3.3 <i>Modèle autorégressif.....</i>	12
I.2.4 APPROCHE PAR TRANSFORMATION.....	12
I.2.4.1 <i>La transformée de Fourier.....</i>	13
I.2.4.2. <i>La transformée de Gabor.....</i>	13
CONCLUSION.....	14
CHAPITRE 2 MODELISATION DE TEXTURE BASEE SUR LES MODELES AUTOREGRESSIFS	16
II.1 LE MODELE AR 1D	18
II.2 LE MODELE AR-2D.....	21
II.2.1 LE SUPPORT DE PREDICTION	21
a) <i>Support de prédiction non-causal d'ordre (M,N).....</i>	22
b) <i>Support de prédiction causal d'ordre (M,N).....</i>	22
c) <i>Support de prédiction semi-causal d'ordre (M,N).....</i>	23
II.3 ESTIMATION DES PARAMETRES D'UN MODELE AR APPLIQUE A LA TEXTURE.....	24
II.3.1. MODELE AUTOREGRESSIF MONODIMENSIONNEL.....	25
II.3.1.1 <i>Les équations de Yules-Walker.....</i>	25
• L'algorithme de Levinson-Durbin 1D.....	27
II.3.1.2 <i>Méthode des moindres carrés.....</i>	28
II.3.1.3 <i>Application</i>	29
II.3.1.3.1 Cas d'un support causal QP.....	30
a) Estimation des paramètres par Levinson 1D	30
Effet de l'utilisation de l'algorithme de levinson 1D.....	33
R. Estimation des paramètres par la méthode des moindres carrés.....	33
II.3.1.3.2 Cas d'un support non causal.....	34
II.3.2. MODELE AUTOREGRESSIF BIDIMENSIONNEL.....	36
• L'algorithme de Levinson 2D	37
II.3.2.1 <i>Application</i>	40

II.3.3 AUTRES METHODES D'ESTIMATION (LE MAXIMUM DE VRAISEMBLANCE).....	41
CONCLUSION	41
CHAPITRE 3 SYNTHÈSE DE TEXTURE ET CLASSIFICATION.....	43
III.1 SYNTHÈSE DE TEXTURE.....	43
III.1.1 SYNTHÈSE BASÉE SUR LES PARAMÈTRES ESTIMÉS PAR LEVINSON 1D.	43
III.1.2 SYNTHÈSE BASÉE SUR LES PARAMÈTRES ESTIMÉS PAR LEVINSON 2D.	44
III.1.3 SYNTHÈSE BASÉE SUR LES PARAMÈTRES ESTIMÉS PAR LES MOINDRES CARRÉS.	45
III.1.4 SYNTHÈSE DE TEXTURES OBEISSANT À UN MODÈLE AR NON CAUSAL.....	48
III.2 CLASSIFICATION DES IMAGES TEXTURÉES.	53
III.2.1 PRINCIPE	53
III.2.2 SCHEMA DE CLASSIFICATION.....	54
III.2.3 RESULTATS DE CLASSIFICATION	55
III.2.3.1 Equations de Yules-Walker 1D (Levinson 1D).....	55
III.2.3.2 Equations de Yules-Walker 2D (Levinson 2D).....	56
III.2.3.3 Moindres carrés support causal.....	56
III.2.3.4 Moindres carrés support non causal.....	57
CONCLUSION	61
CONCLUSION GÉNÉRALE.....	64
BIBLIOGRAPHIE	66

Liste des figures

Fig. I.1. Textures structurales-----	5
Fig. I.2. Textures aléatoires.-----	5
Fig. I.3. Influence de la fenêtre d'analyse-----	7
Fig. I.4. Textures fractales -----	10
Fig. I.5. Voisinage du champ de Markov.-----	11
Fig. I.6. Cliques associés à l'ordre 2. -----	12
Fig. II.1 Prédiction progressive de $y(n)$ à partir de p échantillons antérieurs-----	19
Fig. II.2 modèle AR d'ordre p -----	20
Fig. II.3 Filtre linéaire pour générer (a) le processus y à partir d'un bruit blanc.(b) le filtre inverse -----	20
Fig. II.4. Voisinage non causal-----	22
Fig.II.5. (a) Voisinage quart de plan, (b) Voisinage DPNS-----	23
Fig. II.6 Voisinage semi causal-----	24
Fig. II.7. Images de Brodatz. -----	30
Fig.II.8 Tracés des coefficients pour une fenêtre 10×10 -----	31
Fig.II.9. Variance en fonction de l'ordre du modèle.-----	32
Fig. II.10. Temps de calcul des paramètres AR 1D en fonction de l'ordre -----	33
Fig. II.11. Evolution de la variance en fonction de l'ordre. Taille d'échantillon 100×100 -----	34
Fig. II.12. Voisinage non causal d'ordre 3×3 -----	35
Fig. II.13. Evolution de la variance en fonction de l'ordre. -----	36
Fig. II.14. Représentation multi canal d'une image.-----	38
Fig. II.15. Evolution de la variance par Levinson 2D. -----	40
Fig. III.1. Textures synthétisées par les paramètres estimés par Levinson 1D-----	44
Fig. III.2. Textures synthétisées à partir des paramètres estimés par Levinson 2D. -----	45
Fig.III.3. Textures synthétisées pour différents ordres par la méthode des moindres carrés, support quart de plan. Taille d'échantillon 50×50 -----	46
Figure III.4. TF des textures originales et synthétisées par la méthode des moindres carrés (support causal) -----	48
Fig. III.5. Textures synthétisées par l'échantillonneur GMRF direct -----	51
Fig. III.5. Textures synthétisées par l'échantillonneur GMRF direct (suite)-----	52
Fig. III.6. Histogrammes des textures originales et synthétisées-----	52
Fig. III.6. Histogrammes des textures originales et synthétisées (suite)-----	53
Fig. III.7. Images à segmenter-----	55
Fig.III.8. Image1 classifiée par levinson1D. ordre 3×3 , et 15×15 (taille d'échantillon 32×32)-----	55
Fig.III.9. Image1 classifiée par levinson2D. ordre 3×3 , et 15×15 (taille d'échantillon 32×32)-----	56
Fig.III.10. Image1 classifiée par la méthode des moindres carrés (support QP). Ordre 3×3 , et 7×7 (taille d'échantillon 32×32) -----	56
Fig.III.11. Image1 classifiée par la méthode des moindres carrés (support QP). Ordre 3×3 , et 7×7 (taille d'échantillon 16×16) -----	57
Fig.III.12. Image2 classifiée par la méthode des moindres carrés (support causal). Ordre 3×3 , et 7×7 (taille d'échantillon 16×16 , et 32×32)-----	57
Fig.III.13. Classification de 'image1' par la méthode des MC, support non causal -----	58
Fig.III.14. Classification de 'image2' par la méthode des MC, support non causal -----	58
Fig.III.15. Image optique-----	59
Fig.III.16. Classification de l'image optique. Ordre 5×5 , taille d'échantillon 32×32 . -----	59
Fig.III.17 Image SAR-----	60
Fig.III.18. Image SAR classifiée avec les ordres 5×5 , 9×9 (taille d'échantillon 16×16 , 32×32)-----	60

Liste des tableaux

<i>Tableau II.1 Coefficients AR 1D pour une fenêtre 10x10</i>	30
<i>Tableau II.2 Evolution de la variance en fonction de l'ordre</i>	31
<i>Tableau II.3. Paramètres estimés avec deux portions de textures différentes</i>	32
<i>Tableau II.4 Evolution de la variance</i>	33
<i>Tableau II.5 Evolution de la variance en fonction de l'ordre.</i>	35
<i>Tableau II.6 Evolution de la variance</i>	40
<i>Tableau III.1 Paramètres estimés par moindres carrés de chevron originale et synthétisée (ordre 7x7, taille d'échantillon 50x50)</i>	47
<i>Tableau III.2 Pourcentage d'identification des classes pour l'image classifiée avec ordre 9x9, taille d'échantillon 32x32</i>	61

Introduction générale

La texture est quasi omniprésente dans les images numériques représentant des scènes réelles. Bien qu'il n'existe pas de définition unique de la texture dans la littérature de l'imagerie numérique, celle-ci est facilement perçue par l'être humain, c'est une source riche d'informations visuelles et contribue à l'apparence globale des objets.

La texture peut être définie par la distribution spatiale des niveaux de gris dans l'image [Har, 73]. Cette distribution est due aux variations physiques de la scène (grains de sable, vagues sur l'eau, etc.).

La modélisation de texture consiste justement à rechercher les modèles de distribution propre à chaque texture. La texture est alors considérée comme la réalisation d'un processus aléatoire dont on cherche à déterminer le modèle. Ce modèle aura un double objectif : décrire la texture et la synthétiser. Un modèle de texture idéal est celui qui est capable de capturer les caractéristiques uniques d'une texture particulière. Il devrait, alors, permettre de régénérer la texture à partir de l'ensemble des paramètres du modèle.

On peut distinguer deux grandes familles de modèles, les modèles paramétriques, et les modèles non paramétriques. Les premiers reposent sur la détermination des paramètres du modèle. Une fois ces derniers estimés, ils seront utilisés pour la synthèse ou la classification [Kou, 2007]. Les modèles autorégressifs, Markoviens et fractals sont les modèles paramétriques les plus utilisés. Quand aux modèles non paramétriques, ils ne sont pas assujettis à l'établissement d'un modèle [Kou, 2007].

Le modèle autorégressif a largement été utilisé en traitement du signal. On considère, alors, que la valeur d'une variable aléatoire à un instant donné peut être obtenue par une sommation pondérée des valeurs précédentes. Dans le cas de l'imagerie, on suppose que l'intensité d'un pixel donné peut s'exprimer en fonction des intensités des pixels qui lui sont voisins. On cherche, en fait, à modéliser la distribution spatiale des intensités par une dépendance linéaire.

Dans ce contexte, nous nous sommes intéressés à l'étude et la mise en œuvre des modèles autorégressifs (AR) dans le but de caractériser, de synthétiser et d'identifier les images texturées, en considérant la texture comme étant un processus AR monodimensionnel d'une part, et un processus AR bidimensionnel d'autre part. Pour ce faire, nous avons mis en œuvre trois méthodes différentes d'estimation des paramètres du modèle AR : les équations de Yules-Walker 1D, les équations de Yules-Walker 2D et la méthode des moindres carrés. Nous avons considéré pour ceux-ci deux supports de prédiction : le support causal quart de plan et le support non causal. Nous avons, ensuite, synthétisé des textures et classifié des images texturées en utilisant les paramètres estimés. Par ailleurs, nous avons évalué les

effets de la méthode d'estimation, de l'ordre du modèle, ainsi que l'effet du type de support de prédiction sur les résultats de synthèse et de classification.

Ainsi, ce mémoire est organisé de la manière suivante :

Dans le premier chapitre, nous définissons quelques notions élémentaires de la texture, et nous présentons une synthèse résumant différentes approches d'analyse texturale.

Le deuxième chapitre est consacré au modèle autorégressif, où nous rappelons les concepts et les définitions mathématiques de base utilisés dans la modélisation AR 1D et 2D. Par ailleurs, nous explicitons et développons les différentes procédures d'estimation des paramètres du modèle AR appliqué à la texture, et nous présentons les algorithmes récursifs de Levinson 1D dans le cas monodimensionnel, et l'algorithme de Levinson 2D basé sur une approche multi canal pour le cas bidimensionnel. Ces algorithmes permettent de déterminer les paramètres du modèle AR de façon récursive, et d'améliorer, ainsi, le temps d'estimation de ces paramètres.

Les résultats de la mise en œuvre des différentes méthodes d'estimation sont donnés dans le troisième chapitre. Les paramètres estimés sont employés dans des procédures de synthèse, et les textures synthétisées sont comparées aux textures modélisées. La synthèse des textures obéissant à un modèle AR non causal est réalisée à l'aide de l'algorithme GMRF direct (Gaussian Markov Random Field).

D'autre part, nous présentons dans ce chapitre les résultats de classification des images texturées en employant les paramètres AR estimés. Les approches mises en œuvre sont testées sur des textures de Brodatz, et ensuite appliquées sur des images satellitaires.

Chapitre 1 Texture et Analyse Texturale

L'analyse de textures est un thème d'intérêt dans la communauté du traitement du signal et des images. C'est un domaine de recherche encore largement ouvert, si l'on en juge par le nombre de travaux sur ce sujet [Che, 88], [Bou, 90], [Cha, 90], [Men, 92], [Bad, 93], [Dil, 93], [Elf, 94], [Rup, 99], [Alan,2002],. Le but de l'analyse de texture est de formaliser le descriptif de la texture par des paramètres mathématiques qui serviront à l'identifier.

L'analyse de texture est utilisée dans des domaines de plus en plus variés. Une application immédiate de l'analyse de texture est l'identification des régions de l'image en utilisant des propriétés de texture, on parle alors de *la classification de texture*. Le but de la classification de texture est alors de produire une carte où chacune des régions texturées est identifiée à la classe de texture à laquelle elle appartient. Une deuxième application de l'analyse de texture est *la segmentation de texture*. Le but de la segmentation est d'obtenir la carte de frontières d'une image.

L'analyse de texture est souvent employée pour la compression d'image. Les images dépeignant des scènes naturelles contiennent souvent de grandes régions texturées, telles qu'une forêt, ou une plage de sable, etc. Puisque les textures contiennent souvent l'information significative à haute fréquence, elles ne sont pas bien compressées par les techniques de transformation telles que le JPEG (Joint Photographic Experts Group). En segmentant ces régions texturées dans une étape de prétraitement, cela les rend compressibles par une technique de synthèse de texture [Wei,2001].

La texture joue un rôle important, aussi bien, en infographie où le but est de synthétiser des surfaces qui sont aussi réalistes que possible, qu'en restauration d'images en permettant de remplacer les parties dégradées ou manquantes par une version synthétisée générée à partir du modèle textural élaboré.

Les techniques d'analyse de texture sont employées dans plusieurs applications dont le diagnostic médical et la télédétection. En général, ces applications impliquent l'extraction automatique des

caractéristiques de l'image, qui sont alors employées pour une variété de tâches de classification, telles que la distinction de tissu normal d'un tissu pathologique, et l'identification des couvertures terrestres.

Dans ce qui suit, nous donnons les notions élémentaires de texture, puis nous exposons une synthèse sur les méthodes d'analyse texturale les plus employées.

I.1 Notion de texture

I.1.1 Définition de la texture.

Dans le traitement d'images, la texture est considérée comme une caractéristique très importante. C'est le terme utilisé pour caractériser la distribution spatiale des éléments d'une surface donnée d'une image. On peut définir plusieurs propriétés de la texture : uniformité, densité, rugosité, régularité, linéarité, fréquence, phase, directivité, grossièreté, aspect aléatoire, finesse, douceur, granulation, etc... [Mat, 98].

Bien qu'il n'y ait aucune définition stricte de la *texture*, celle-ci est facilement perçue par l'être humain, c'est une source riche d'informations visuelles sur les objets. Dans le langage courant, on lui associe souvent un adjectif : texture granuleuse de la peau d'orange, texture lisse de la cerise, etc. Dans l'album de Brodatz [Bro, 66], nous pouvons trouver une grande collection de textures.

I.1.2 Types de textures.

On peut classer les textures en deux grandes catégories, correspondant à deux niveaux de perception :

a) *Textures structurelles (macro textures)* : Elles se présentent sous un aspect régulier avec une répétition de motifs élémentaires. La répartition spatiale de ces motifs de base suit des règles de directions et de placement. Exemple : peau de lézard, mur de brique (Figure I.1). Cette catégorie de textures a engendré les méthodes d'analyse structurelles.

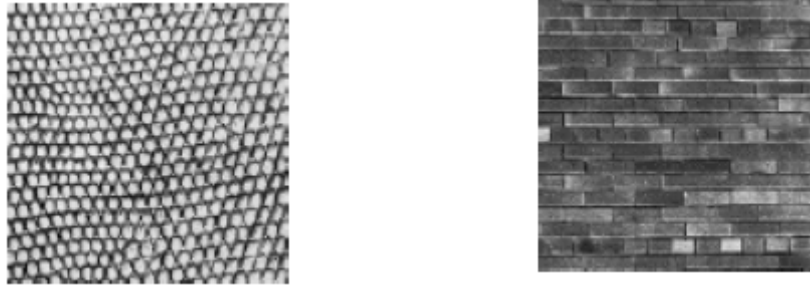


Fig. I.1. Textures structurées

b) *Textures aléatoires (micro textures)* : Contrairement aux textures structurées, les textures aléatoires présentent des primitives microscopiques distribuées de manière désordonnée tout en apparaissant globalement homogènes. Exemple : texture "sable", "herbe", "tissu" (Fig. I.2). En associant l'image à un processus aléatoire bidimensionnel, cette catégorie de textures a engendré les méthodes d'analyse statistiques.

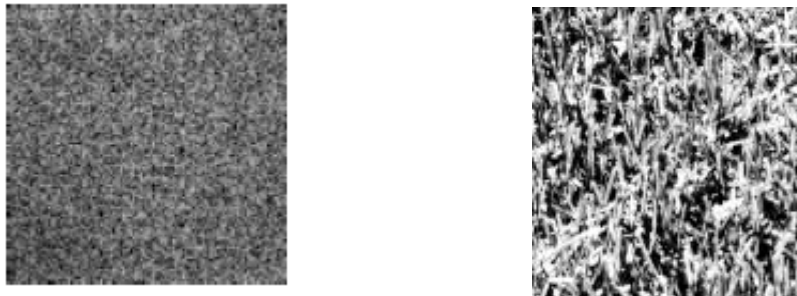


Fig. I.2. Textures aléatoires.

Dans la réalité, il existe peu de textures totalement structurées ou aléatoires. Ainsi ; La plupart des textures n'appartiennent pas exclusivement à l'une de ces deux catégories, mais se positionnent entre les deux, ce qui explique la difficulté de l'analyse texturale.

I.1.3 Attributs visuels de la texture

Diverses expériences ont été faites afin de mettre en évidence la perception visuelle humaine des textures. On peut citer par exemple les expériences psycho visuelles de Julesz [Jul,62], et de Gagalowicz [Gag.83]. Pour discuter le travail pilote de Julesz, nous devons d'abord définir ce que sont les statistiques spatiales du premier et du deuxième ordre.

(i) Statistiques de premier ordre : mesurent la probabilité d'observer un niveau de gris à un endroit aléatoirement choisi dans l'image. Des statistiques de premier ordre peuvent être calculées à partir de l'histogramme des niveaux de gris de l'image. Celles-ci dépendent seulement des valeurs des pixels et

non pas de l'interaction ou la cooccurrence des valeurs voisines des pixels. La moyenne de l'image est un exemple de statistiques du premier ordre.

(ii) Statistiques de second ordre : sont définies comme la probabilité d'observer une paire de niveaux de gris se produisant à des endroits et à une orientation aléatoire dans l'image.

B.Julesz [Jul,62] a mené des expériences dans lesquelles une image composée de différentes textures est présentée aux observateurs pendant une période très brève (environ 100 ms). Les observateurs doivent retrouver les différentes textures composant l'image. Ces tests ont pour but de démontrer la complexité du traitement effectué par le système visuel sans qu'une analyse détaillée, engageant une procédure de mémorisation ou de reconnaissance, ne soit effectuée.

Les résultats de ces expériences, sur la discrimination pré attentive des textures, ont conduit à la fameuse conjoncture de B.Julesz selon laquelle : « un observateur humain est incapable de faire la distinction entre des textures ayant des statistiques d'ordre deux égales, et différentes pour les ordres supérieurs ».

Dans [Jul,83], Julesz remet en cause cette conjoncture, en donnant un contre exemple. Il avança une autre hypothèse, qui suppose que le système visuel pré attentif opère comme un détecteur d'éléments fondamentaux de structures locales nommées « Textons ». Ces derniers sont considérés comme étant des structures géométriques locales comprenant aussi des terminaisons et des croisements. La détection de ces structures locales donne une possibilité de discrimination des textures ayant des distributions du deuxième ordre identiques.

I.1.4 Invariance par translation

L'invariance par translation est une propriété qui a une influence directe sur l'analyse de la texture. Elle peut être vérifiée si la fenêtre d'analyse est pertinente au regard de la texture considérée. Le fait d'analyser de petites parties d'une texture au lieu de la texture toute entière peut conduire à l'étude de zones ne possédant pas les mêmes caractéristiques texturales. Considérons les textures de la figure I.3. Dans l'image (a) ; les deux fenêtres fournissent la même information texturale, tandis que dans l'image (b), les deux fenêtres ne sont pas cohérentes puisqu'on ne retrouve pas les mêmes caractéristiques texturales [Seb, 2001].

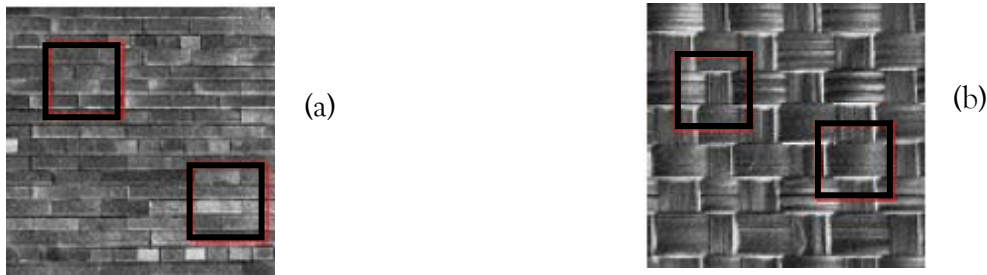


Fig. 1.3. Influence de la fenêtre d'analyse

I.2 Approches d'analyse de texture

Les approches en analyse de texture peuvent être regroupées en quatre grandes familles :

- Les approches statistiques.
- Les approches structurales.
- Les approches basées sur la modélisation.
- Les approches basées sur les transformations

I.2.1 Approches statistiques.

Les méthodes statistiques, comme les matrices de cooccurrence [Har,73], [Har,79], [Elf,94], s'adaptent à l'étude des textures naturelles, et particulièrement aux textures microscopiques. Ces méthodes exploitent les paramètres statistiques qui modélisent la distribution et les relations entre les niveaux de gris de l'image.

Méthode de matrice de cooccurrence

La matrice de cooccurrence est une matrice de taille $N \times N$ où N est le nombre de niveaux de gris dans l'image, elle est donc de taille 256×256 pour une image de 256 niveaux de gris. En pratique, on réduit ce nombre à 8 ou 16 ou 32. Pour un déplacement (translation) d , un élément (i,j) de la matrice $(i, j : 0 \leq NG_{\max})$ est défini par le nombre de pixels de l'image de niveau de gris i situés à 'd' d'un pixel de niveau de gris j . Considérons par exemple l'image suivante, notée 'I', avec trois niveaux de gris : $z_1=0, z_2=1, z_3=2$.

$$I = \begin{bmatrix} 0 & 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 & 1 \\ 2 & 2 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Pour le déplacement $d(1,1)$ comme un pixel à droite et un pixel en bas, on aboutit à la matrice de cooccurrence 3×3 suivante :

$$A = \begin{bmatrix} 4 & 2 & 1 \\ 2 & 3 & 2 \\ 0 & 2 & 0 \end{bmatrix}$$

Les matrices de cooccurrence mettent en évidence les relations qui existent entre les pixels, à la fois par un aspect local et un aspect spatial (le déplacement). Cependant, ceci n'est vrai que si un grand nombre de matrices est calculé. Si on se limite au calcul de quelques matrices en ne considérant qu'une direction et quelques déplacements, une grosse partie de l'information sera perdue.

Les matrices de cooccurrence contiennent une masse d'information trop importante. On extrait de ces matrices quatorze indices (attributs), prenant en compte l'ensemble de la matrice. Ces indices ont été définis par Haralick [Har,73], ils représentent des descriptifs de la texture. Ces attributs peuvent être appliqués sur une image, avec une fenêtre d'analyse prédéfinie pour générer une image de texture caractérisant un aspect particulier de la texture.

Cependant, les images générées par ces attributs présentent un certain degré de corrélation entre eux, on restreint l'utilisation au sept attributs qui donnent les images les plus décorréliées. [Har, 79]. Les indices retenus sont : la moyenne, la variance, l'énergie ou le second moment, le contraste ou l'inertie, l'entropie, la corrélation, l'homogénéité locale. Tous ces attributs sont définis pour une valeur de déplacement d .

I.2.2 Les approches structurales.

Les méthodes structurales décrivent la texture par une extraction de primitives, et une détermination des règles de placement des primitives. Ces méthodes s'adaptent à l'analyse de textures structurées qui présentent une organisation purement déterministe. C'est le cas de la morphologie mathématique [Ser,82], et de la méthode de longueurs de plages [Gall, 75].

La morphologie mathématique.

Fondée par Matheron et Serra [Ser,82], la morphologie mathématique fournit des outils puissants à toutes les disciplines qu'intéresse l'analyse quantitative d'images.

L'idée de base des méthodes morphologiques est de comparer l'image que l'on veut analyser à un objet de forme connue appelé élément structurant. Cet élément structurant est une forme géométrique qui peut être : un disque, un segment, un carré, un hexagone ou toute autre forme géométrique. Il est porteur d'une information minimum : une origine (centre du cercle, extrémité du segment, coin d'un carré ou d'un rectangle, etc), et une dimension (rayon du centre, longueur du segment, etc).

Une transformation morphologique est un moyen de réaliser des inventaires structuraux de façon systématique. Elle procède en choisissant, indépendamment de l'ensemble $\{Ie(x, y)\}$ qui représente la fenêtre de l'image à traiter, un autre ensemble Y centré à son origine et de forme géométrique donnée, (l'élément structurant). Ensuite, on déplace cet élément de façon à ce que son origine passe par tous les points de l'espace image. A chaque position, on se pose une question relative à l'union, à l'intersection et à l'inclusion de Y avec $\{Ie(x, y)\}$ ou dans $\{Ie(x, y)\}$. L'ensemble obtenu par les centres de l'élément Y correspond aux réponses de la transformée de l'ensemble $\{Ie(x, y)\}$.

Ces transformations morphologiques peuvent être appliquées sur des images en niveaux de gris ou sur des images binaires. Les principaux opérateurs morphologiques sont : l'érosion, la dilatation, l'ouverture et la fermeture.

Méthode de longueurs de plages.

Une plage de niveaux de gris (ou iso-segment) est un ensemble connexe et linéaire de pixels ayant le même niveau de gris [Gall, 75]. La longueur d'une plage (run length) est le nombre de pixels dans cet ensemble. La méthode des plages consiste à détecter des plages présentes dans l'image dans une direction donnée, et à effectuer des statistiques sur leurs longueurs.

On définit, ainsi, plusieurs matrices correspondant chacune à une direction des plages dans l'image. Cette direction est fixée par un angle θ défini par rapport à un bord de l'image. La matrice $M(i, j, \theta)$ est constituée par les éléments $f(i, j, \theta)$ qui indiquent le nombre d'occurrences dans l'image d'une plage de longueur j et de niveau de gris i orientée suivant l'angle θ .

Cinq paramètres caractéristiques sont dérivés de cette matrice qui sont : iso-segments courts, iso-segments longs, distribution des niveaux de gris, distribution des iso-segments, pourcentage des iso-segments. En calculant les cinq paramètres à partir des quatre matrices $M(i, j, \theta_i)$; $i=0^\circ, 45^\circ, 90^\circ$ et 135° , on obtient une signature texturale de 20 paramètres(5x4) [Bel,98], qui peut être exploitée dans un procédé de classification ou de segmentation.

I.2.3 Approches par modélisation

L'analyse de texture basée sur les modèles interprète la texture comme la réalisation de certaines lois de probabilité sur un champ aléatoire (modèle stochastique). Elle repose sur l'hypothèse d'existence d'une dépendance entre un pixel et les pixels qui lui sont voisins. Cette dépendance (fonction à trouver) est utilisée pour synthétiser des textures. Ainsi ; le niveau de gris de chaque pixel peut être le résultat d'une somme pondérée des points qui lui sont voisins. Les coefficients de pondération représentent, alors, les paramètres du modèle de texture. Une fois ces paramètres estimés, ils sont utilisés pour la synthèse, ou la classification. Ces modèles permettent une représentation concise des textures en ne prenant en compte, théoriquement qu'un nombre petit de paramètres. Parmi ces modèles, on cite :

- a) Le modèle Markovien [Che, 88], [Gho,93], [Di,93], [Hid,96].
- b) Le modèle AR 1D et 2D [Che,85], [Fra,88], [Bad,93], [Kie,2001],
- c) Le modèle fractal [Man, 82], [Cha,2001].

I.2.3.1 Modèle fractal

Mandelbrot [Man, 82] est celui qui a proposé la géométrie fractale, et est le premier à remarquer son existence dans le monde naturel.

Un fractal n'est qu'une reproduction itérative d'un patron de base selon des règles mathématiques. Selon la complexité des lois, on peut générer des images ayant une géométrie simple ou complexe, c'est le cas des images générées par ordinateur.

Une texture fractale représente un exemple d'une texture aléatoire. Elle est caractérisée par le fait qu'elle se ressemble à n'importe quelle échelle de décimation. Cette caractéristique est nommée la dimension fractale. Une estimation de cette dimension est nécessaire pour synthétiser une telle texture. Malheureusement, il est difficile d'estimer cette dimension pour tout type de texture. La figure (I.4) montre deux exemples de textures fractales [Cha, 2001].



Fig. I.4. Textures fractales

I.2.3.2 Modèle Markovien

Les modèles de Markov ont suscité de nombreux travaux en traitement d'images ces dernières années. Le modèle Markovien a été appliqué dans diverses applications de traitement d'image comme la synthèse [Raj,93], [Myu,2001], la classification de texture [Jen,94], la segmentation d'image [Dil,93],[San,97], la restauration [Alan,2002], et la compression d'image [Bad,93].

La propriété fondamentale des champs de Markov appliquée au traitement d'images, est que l'on suppose que la probabilité conditionnelle de la valeur estimée d'un pixel donné, sachant l'état de tous les autres pixels de l'image est égale à la probabilité conditionnelle aux états des pixels appartenant à son voisinage immédiat, soit :

$$P(X_s = x_s | X_r = x_r, r \in S, r \neq s) = P(X_s = x_s | X_r = x_r, r \in V_s) \quad (I.1)$$

Où : $S = \{s_1, s_2, \dots, s_n\}$ L'ensemble des sites (pixels).

X : une variable aléatoire associée au pixel $s \in S$

X_s : la valeur de la variable aléatoire X au site s .

V_s : est le voisinage du pixel s .

La taille et la forme du voisinage sont définies par l'ordre du modèle. Les voisinages d'ordre 1, 2, 3 et 4 sont représentés dans la figure (I.5), où les pixels d'ordre 1 sont libellés '1', et ceux du 2ème ordre sont libellés '2' et '3', etc .

	4	3	4	
4	2	1	2	4
3	1	x	1	3
4	2	1	2	4
	4	3	4	

Fig. I.5. Voisinage du champ de Markov.

Grâce au théorème d'Hammersley Clifford, nous pouvons obtenir les relations qui simplifient la conception de champs de Markov. Basé sur ce théorème les champs de Gibbs sont les plus souvent utilisés lorsqu'il faut implanter des champs de Markov.

La distribution de Gibbs s'écrit :

$$P(X = x) = \frac{1}{Z} e^{-U(x)} \quad (I.2)$$

Où $Z = \sum e^{-U(x)}$ est un facteur de normalisation (appelé aussi fonction de partition). $U(x) = \sum V_c(x)$ est une fonction d'énergie définie sur un ensemble de pixels appelés cliques. Le terme $V_c(x)$ est le potentiel associé à chaque clique c . Pour le voisinage d'ordre 2, les cliques possibles sont:

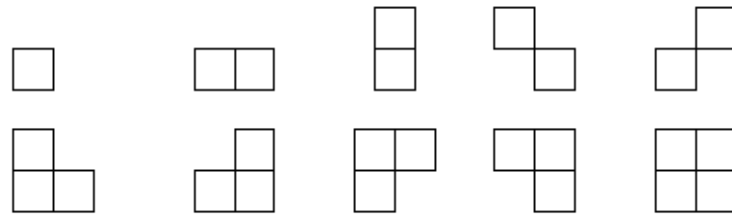


Fig. 1.6. Cliques associés à l'ordre 2.

I.2.3.3 Modèle autorégressif

Le modèle autorégressif est basé sur l'hypothèse de l'existence d'interactions locales entre les pixels de l'image. Pour un pixel donné $y(i,j)$, cette interaction se traduit sous la forme d'une combinaison linéaire des pixels voisins, soit :

$$y(i, j) = \sum_{(m,n) \in D} a(m, n) y(i - m, j - n) + u(i, j) \quad (I.3)$$

Où $a(m,n)$: les paramètres du modèle.

D : le voisinage du pixel $y(i,j)$.

$u(i,j)$: bruit blanc de variance σ^2 .

Les coefficients de pondération $a(i,j)$ peuvent être déterminés par plusieurs méthodes telles que les moindres carrés et le maximum de vraisemblance. Ce modèle a été employé dans de nombreuses applications de traitement d'images. De plus amples détails sur ce modèle seront présentés dans les chapitres suivants.

I.2.4 Approche par transformation

L'analyse basée sur les transformations, comme les transformations de Fourier, Gabor, et Ondelettes, représentent l'image dans un espace dont les axes ont une interprétation étroitement liée aux caractéristiques de la texture, comme la fréquence ou la taille [Liu,90].

I.2.4.1 La transformée de Fourier

La transformée de Fourier (TF) permet d'extraire les caractéristiques fréquentielles de la texture. Pour une image I de taille $M \times N$, la TF est donnée par l'expression suivante :

$$F(x, y) = \frac{1}{M \times N} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} I(k, l) e^{2\pi i \left(\frac{k}{N} x + \frac{l}{M} y \right)} \quad (I.4)$$

La transformée de Fourier d'une image représente la décomposition de l'image originale en terme de composantes de fréquences spatiales qui expriment les transitions des niveaux de gris dans le domaine spatial [Bel, 98]. L'image peut être reconstituée à partir de sa transformée inverse dont l'expression est la suivante :

$$I(k, l) = \frac{1}{M \times N} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u, v) e^{2\pi i \left(\frac{uk}{N} + \frac{vl}{M} \right)} \quad (I.5)$$

La quantité $|F(u, v)|^2$ est appelée spectre de puissance.

On peut extraire des attributs de texture à l'aide du spectre de puissance et la phase de la transformée de Fourier. Liu et Al [Liu,90] ont proposé 28 paramètres calculés à partir des spectres de puissance et de la phase.

Bien que la transformée de Fourier soit porteuse de précieuses informations sur le signal analysé, elle ne permet pas de localiser les portions du signal dans lesquelles les variations sont rapides, ni celles où elles sont lentes. Gabor a suggéré de rendre locale l'analyse de Fourier, en s'aidant de fenêtres. Une fenêtre est une fonction régulière, lentement variable, et bien localisée. En multipliant le signal étudié par une fenêtre, on en obtient une version « locale », dont on peut déterminer le contenu fréquentiel par analyse de Fourier classique. On renouvelle alors l'opération en déplaçant la fenêtre d'analyse. L'ensemble de ces transformées de Fourier ainsi localisées forme la transformée de Gabor du signal, et fournit donc une analyse fréquentielle locale.

I.2.4.2. La transformée de Gabor

Les fonctions de Gabor bidimensionnelles ou les Gaussiennes lissées font partie des méthodes d'analyse fréquentielle [REE, 93]. En effet, Gabor propose un procédé d'analyse d'un ensemble de signaux élémentaires qui sont des oscillations harmoniques modulées par une impulsion de probabilité

gaussienne. Ces signaux élémentaires sont localisés en temps (au-delà en espace pour l'image) et en fréquence, et ont la propriété de minimiser la dispersion en temps et en fréquence [DU,90]. Par ailleurs, les études récentes sur la vision indiquent que la représentation d'une image dans le contexte visuel nécessite les deux variables : spatiale et fréquentielle pour sa description. Les filtres de Gabor fournissent cette possibilité de dépendance des deux variables simultanément [Bel,98].

Les fonctions de Gabor sont définies comme des sinusoides modulées par une gaussienne. La forme mathématique d'une fonction de Gabor dans l'espace à deux dimensions est [Bel,98] :

$$G(x, y / X_c, Y_c, W, \theta, \varphi) = \exp\left[-\frac{(x - X_c)^2 (y - Y_c)^2}{2\sigma^2}\right] \cdot \sin[W(x \cos \theta - y \sin \theta) + \varphi] \quad (I.6)$$

Où :

X_c, Y_c sont les coordonnées du centre du filtre.

W et θ sont respectivement la fréquence centrale et l'angle d'orientation du filtre.

φ est la phase de l'onde plane.

σ^2 est la largeur gaussienne(écart type)

De même que la transformée de Fourier, le signal peut être reconstruit à partir de sa transformée de Gabor.

Conclusion

Dans ce chapitre nous avons présenté succinctement les méthodes d'analyse de texture les plus employées. L'étude bibliographique que nous avons menée sur la texture et l'analyse texturale nous a permis d'approcher les méthodes de traitement d'images texturées.

Dans ce travail, nous nous sommes proposés d'étudier et de développer une méthode d'analyse qu'est la modélisation AR, du fait qu'elle est largement employée pour modéliser les textures naturelles [Chel,85], [Ran,85], [Fran,89], [Cha,90], [Men,92], [Weu,92], [Bad,93]. [Jit,93], [Tak,97], [Ala,98], [Kie,2001]. Les paramètres du modèle AR estimés seront utilisés dans la synthèse, pour générer une texture similaire à la texture originale. Grâce à la synthèse, on évaluera le modèle, en vue de l'employer dans un processus de segmentation des images issues de l'album de Brodatz, et des images satellitaires.

Nous présentons dans le chapitre suivant la théorie liée au modèle autorégressif, ainsi que les méthodes d'estimation des paramètres de ce modèle.

Chapitre 2 Modélisation de texture basée sur les modèles autorégressifs

Le modèle autorégressif bidimensionnel est une extension naturelle du modèle AR monodimensionnel. Il a été étudié et utilisé dans la modélisation des signaux bidimensionnels, en particulier les textures naturelles [Che, 85], [Chel,85], [Ran,85], [Fra,88], [Fran,89], [Cha,90], [Yus, 90], [Men,92], [Weu,92], [Bad,93]. [Jit,93], [Mit,93], [Sta, 94], [Ala, 96], [Tak,97], [Ala,98], [Kie,2001]. L'utilité de la modélisation AR 2D est illustrée par des expériences qui montrent qu'il est possible de créer des textures synthétisées ayant des propriétés visuelles similaires à des textures naturelles, de segmenter, de classifier, de compresser des images texturées, et de restaurer des images détériorées.

Si nous considérons l'image comme la réalisation d'un champ aléatoire bidimensionnel $\{y(i,j)\}$, il est donc possible de la représenter par un modèle auto régressif 2D. Le modèle AR 2D suppose que la relation entre un pixel de coordonnées (i,j) et son voisinage est décrite par l'équation aux différences suivante [Ala,98] :

$$y(i, j) = \sum_{(m,n) \in D} \theta_{m,n} y(i-m, j-n) + e(i, j) \quad (\text{II.1})$$

Où $\theta_{m,n}$ sont les coefficients du modèle. D , appelé voisinage ou support de prédiction, est constitué de l'ensemble des pixels utilisés dans la prédiction de la valeur du pixel courant $y(i,j)$. Il peut avoir différentes formes : causale, non causale, ou semi causale. $e(i, j)$ est l'erreur d'estimation. Le nombre de pixels utilisés pour l'estimation du pixel $y(i,j)$ est appelé l'ordre du modèle, noté p . Un modèle AR2D possède $p+1$ paramètres à estimer : p coefficients et la variance de l'erreur d'estimation. Ces paramètres peuvent être estimés par différentes méthodes comme les équations de Yules-Walker, la méthode des moindres carrés, et le maximum de vraisemblance.

Comme mentionné précédemment, les textures sont généralement classées selon deux catégories : structurelles et stochastiques. Cependant, la plupart des textures naturelles n'appartiennent ni à l'une ni

à l'autre de ces deux catégories, mais se situent entre les deux. Francos [Fra, 88] présente une méthode d'analyse de texture plus générale adaptée aux deux catégories. Les composants périodiques et directionnels de la texture sont identifiés, paramétrés via une analyse spectrale, et les composants statistiques sont modélisés via un modèle AR 2D semi causal. Bader [Bad,93] a effectué une comparaison entre des textures synthétisées, en se basant sur les coefficients déterminés par la méthode des moindres carrés d'une part, et la méthode du maximum de vraisemblance (MV) d'autre part. Il conclut que la méthode du maximum de vraisemblance induit une légère amélioration par rapport à la méthode des moindres carrés, du moment que les textures synthétisées par le MV sont visuellement plus proches des textures naturelles, et que la variance est plus petite.

Dans [Ben, 98], Bennett et Khotanzad ont étudié la capacité des modèles SAR (modèle AR non causal) dans la modélisation et la synthèse des images multi spectrales. Kie [Kie,2001] a introduit une nouvelle classe de modèles : les modèles GCAR (Generalized Circular Auto Regressive Model), ceux-ci sont une généralisation du modèle AR circulaire. Cette classe de modèles est définie par un voisinage non causal circulaire ou elliptique où les coefficients AR sont égaux si les pixels se trouvent sur le même cercle ou ellipse. Selon ses résultats, l'auteur conclut que les modèles GCAR permettent de représenter un bon nombre de textures possédants une structure directionnelle, comme la texture "Bois", avec un nombre moindre de paramètres que le modèle AR non causal ordinaire ou les champs de Markov.

Dans [Par, 2001], Parada, Javier et Koppen ont construit un système de retrait de textures d'une base de données accessible via le web. Une de ses caractéristiques principales est la génération de la texture désirée dans le cas où elle n'est pas disponible dans la base de données. Le processus de génération est implémenté en utilisant un modèle AR 2D de façon interactive, où l'utilisateur assiste le système dans la synthèse de la texture désirée.

Le modèle AR 2D a été employé aussi, pour la segmentation et la classification : [Bou,90], [Men,92], [Ala,96], [Cla,97], [Pie, 97], [Ala,98]. Dans [Lia, 92], Lia, Qin et Tan ont considéré le modèle SAR pour classifier et segmenter des textures naturelles et des images médicales, avec un taux de classification de 98% pour les textures naturelles de Brodatz, et 88% pour les images médicales. Dans [Tar, 97], Tardif a employé le modèle AR 2D dans la modélisation et la segmentation des images IRM (images à résonance magnétique) pour la détection des tissus anormaux. Il a étendu son application en modélisant les images IRM par un modèle AR 3D, en proposant aussi, une segmentation tridimensionnelle de ces images. Bouman [Bou, 90] a proposé une méthode non supervisée d'estimation à la fois du nombre et du type

de textures dans une image. Cette méthode a été utilisée pour segmenter une variété d'images synthétisées et aériennes, en modélisant les textures par un modèle AR Gaussien.

En supposant que la texture suit un modèle Gaussien Markovien, et en vue d'une classification, Chellapa et coauteurs [che,85] ont utilisé deux méthodes pour extraire les paramètres du modèle : la méthode des moindres carrés (MC), et la méthode des statistiques suffisantes. Des classificateurs à distance minimale simple ont été utilisés avec les deux ensembles de paramètres, et ont donné un taux de classification de 99% pour les MC et 92% pour la deuxième méthode.

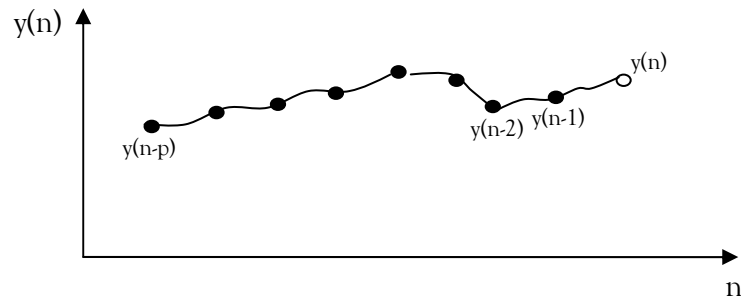
Le modèle AR 2D a été employé dans la compression d'images. Dans [Bad, 93], Bader et coauteurs ont étudié l'effet de la méthode d'estimation des paramètres du modèle sur le résultat de la compression. Les résultats obtenus ont montré que les paramètres estimés par la méthode de maximum de vraisemblance présentent un meilleur taux de compression que les paramètres estimés par la méthode des moindres carrés. Dans d'autres articles comme dans [Tiw, 2003], Tiwari et coauteurs ont mené des tests en vue de comparer l'effet de la forme du voisinage sur le résultat de la compression. Ils ont conclu que le voisinage semi causal donne de meilleurs résultats de compression que le voisinage quart de plan.

Dans les travaux traitant la modélisation AR 2D, différents supports et différentes méthodes d'estimation ont été proposés. Néanmoins, peu d'études comparatives ont été menées, et les résultats de synthèse obtenus sont limités à certains types de textures. Dans ce contexte, les travaux de recherche que nous avons entrepris visent à répertorier et mettre en œuvre les différentes méthodes d'estimation en considérant diverses hypothèses de départ et en les testant sur différentes textures.

Dans ce qui suit, nous rappelons les concepts et les définitions mathématiques de base utilisés pour la modélisation AR 1D et 2D. Nous passons en revue, les différents types de causalité et les méthodes d'estimation des paramètres AR, et nous présentons les algorithmes de Levinson monodimensionnel et bidimensionnel, permettant de déterminer les paramètres du modèle AR de façon récursive.

II.1 Le modèle AR 1D

Dans la modélisation AR 1D, une valeur $y(n)$ d'un processus aléatoire stationnaire y est supposée être prédictible en fonction d'un certain nombre de valeurs antérieures observées du processus, lorsqu'il s'agit de la prédiction progressive. La prédiction rétrograde permet de prédire $y(n)$ à partir de valeurs futurs du processus. Figure II.1.

Fig. II.1 Prédiction progressive de $y(n)$ à partir de p échantillons antérieurs

Considérons la prédiction progressive de la valeur $y(n)$, qui est obtenue par une sommation pondérée des valeurs antérieures $y(n-1)$, $y(n-2)$, ..., $y(n-p)$. [Mak, 75] :

$$\hat{y}(n) = -\sum_{i=1}^p \theta_i y(n-i) \quad (\text{II.2})$$

L'équation (II.2) est connue sous l'appellation de « série temporelle auto régressive ». $\hat{y}(n)$ est la meilleure estimation de $y(n)$ basée sur les p échantillons passés. On définit la différence entre la valeur $y(n)$ et la valeur estimée de $y(n)$ comme une erreur de prédiction, notée $e(n)$.

$$e(n) = y(n) + \sum_{i=1}^p \theta_i y(n-i) \quad (\text{II.3})$$

La TZ de l'équation (II.3) donne alors :

$$Y(z)(1 + \sum_{i=1}^p \theta_i z^{-i}) = E(z) \quad (\text{II.4})$$

$$Y(z) = \frac{E(z)}{1 + \sum_{i=1}^p \theta_i z^{-i}} = H(z)E(z) \quad (\text{II.5})$$

Les coefficients (θ_i) , $i=1, \dots, p$ constituent les coefficients de prédiction, et représentent les paramètres du modèle autorégressif. La modélisation AR passe, donc, par la détermination de ces paramètres (θ_i) qui minimisent la puissance (la variance) de l'erreur d'estimation.

Le processus y peut donc être vu comme la sortie d'un filtre linéaire, de fonction de transfert $H(z)$. Figure.II.2 (a). $e(n)$ est vue comme la sortie du filtre inverse, appelé filtre de l'erreur de prédiction. Figure.II.2 (b).

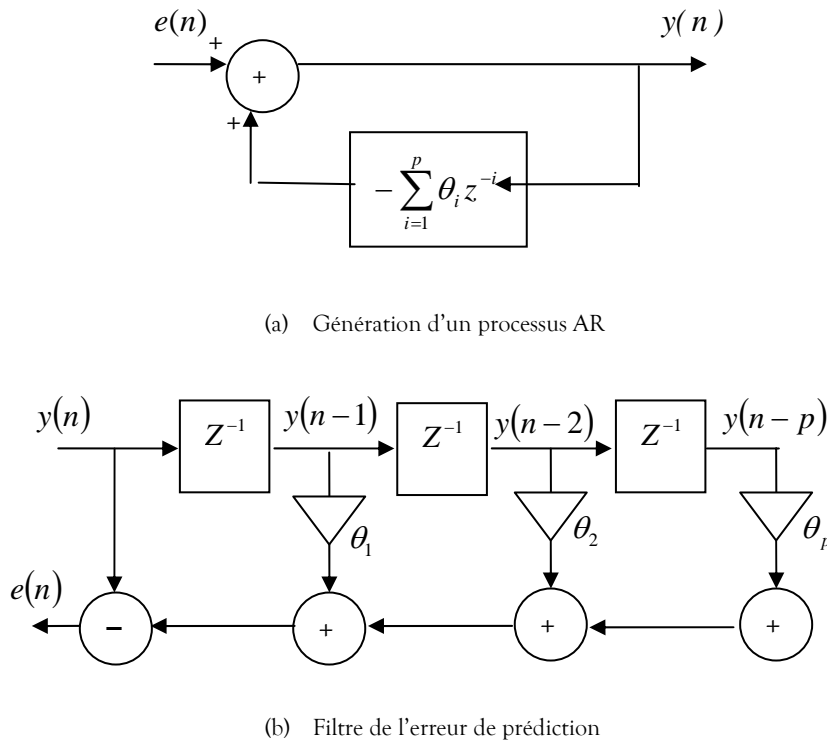


Fig. II.2 modèle AR d'ordre p

Par ailleurs, un processus AR stationnaire au sens large peut être représenté par la sortie d'un système linéaire, de fonction de transfert $H(z)$, excité par un bruit blanc $\omega(n)$ en entrée, de moyenne nulle et de variance σ^2 ($\xi(\omega(n)) = 0, \xi(\omega(n)\omega(m)) = \sigma^2 \delta_{nm}$). Inversement un processus AR stationnaire y peut être transformé en un bruit blanc en le passant à travers un filtre linéaire de fonction $1/H(z)$. On appelle ce filtre un filtre blanchisseur. Figure II.3.(a),(b) [Joh, 96]

$$y(n) = -\sum_{i=1}^p \theta_i y(n-i) + \omega(n) \tag{II.6}$$

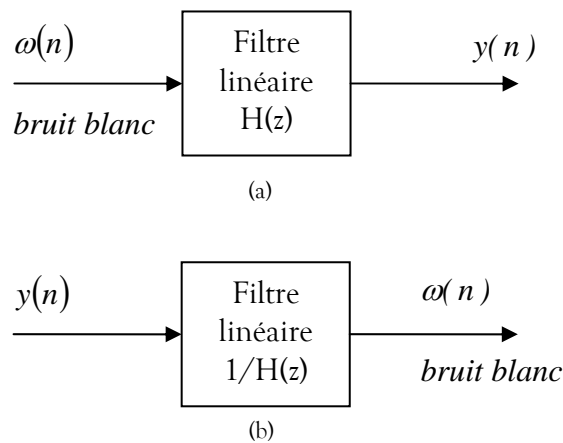


Fig. II.3 Filtre linéaire pour générer (a) le processus y à partir d'un bruit blanc.(b) le filtre inverse

lorsque l'ordre du modèle est grand, l'estimée $\hat{y}(n)$ devient une approximation plus proche de $y(n)$, et donc l'erreur d'estimation $e(n) = y(n) - \hat{y}(n)$ approche un bruit blanc [Guy, 2007], [Joh, 96]

II.2 Le modèle AR-2D.

Si nous considérons l'image comme la réalisation d'un champ aléatoire bidimensionnel $\{y(i,j)\}$, il est donc possible de la représenter par un modèle auto régressif 2D. Le modèle AR 2D suppose que la relation entre un pixel de coordonnées (i,j) et son voisinage est décrite par l'équation aux différences suivante [Ala, 98] :

$$y(i, j) = - \sum_{(m,n) \in D} \theta_{m,n} y(i-m, j-n) + e(i, j) \quad (\text{II.7})$$

$$y(i, j) = \hat{y}(i, j) + e(i, j) \quad (\text{II.8})$$

$\hat{y}(i, j)$ est la meilleure estimation de $y(i,j)$ basée sur les valeurs précédentes des observations.

Par la transformée en Z, on obtient :

$$Y(z_1, z_2)(1 + \sum_{(m,n) \in D} \theta_{m,n} z^{-m} z^{-n}) = E(z_1, z_2) \quad (\text{II.9})$$

$$\text{D'où :} \quad \frac{Y(z_1, z_2)}{E(z_1, z_2)} = \frac{1}{1 + \sum_{(m,n) \in D} \theta_{m,n} z^{-m} z^{-n}} \quad (\text{II.10})$$

L'ensemble $\{\theta_{m,n}\}$, $(m,n) \in D$, est l'ensemble des coefficients du modèle. D est appelé le support de prédiction, il est constitué de l'ensemble des pixels utilisés dans la prédiction de la valeur du pixel courant $y(i,j)$. $e(i,j)$ est l'erreur d'estimation. L'équation (II.10) représente la fonction de transfert du filtre AR bidimensionnel.

II.2.1 Le support de prédiction

Le support de prédiction D est défini par l'ensemble des couples (m,n) qui dépendent du pixel (i,j) . En traitement du signal monodimensionnel, l'ordonnancement d'une séquence suivant un indice est immédiat ainsi que la définition du passé et du futur d'un échantillon. Ce n'est plus le cas pour un signal 2D. Dans le cas 2D différents types de support de prédiction peuvent être envisagés : non causal, semi causal, et causal. Ainsi, l'ensemble D définit le type de causalité du modèle [Ala,98] .

En traitement d'images, seuls les supports de prédiction de dimensions finies présentent un intérêt pratique. Dans ce travail, nous n'avons considéré que les supports de forme rectangulaire. Ainsi, on définit l'ordre (M,N) d'un support, le couple d'entiers qui limite les dimensions du support suivant les deux axes. Un choix optimal de l'ordre du modèle sera fait pour garantir une meilleure modélisation.

a) Support de prédiction non-causal d'ordre (M,N)

Le voisinage non causal est le plus général. Il peut être associé au voisinage des champs de Markov. L'intensité du pixel dépend des niveaux de gris des pixels qui lui sont voisins dans toutes les directions. La figure II.4 représente un voisinage non causal dont voici l'expression mathématique :

$$y(i, j) = - \sum_{m=-M}^M \sum_{n=-N}^N \theta_{m,n} y(i-m, j-n) + e(i, j) \quad \text{ou} \quad (m, n) \neq (0,0) \quad (\text{II.11})$$

Le modèle AR non causal a été nommé quelquefois dans la littérature « Simultaneous AutoRegressive model (SAR) ». Ce modèle a fait l'objet d'études multiples [Bad,93], [Che,85] car, dans le cas où on suppose que le bruit est Gaussien, il offre la possibilité d'utiliser des algorithmes de synthèse relativement rapides.

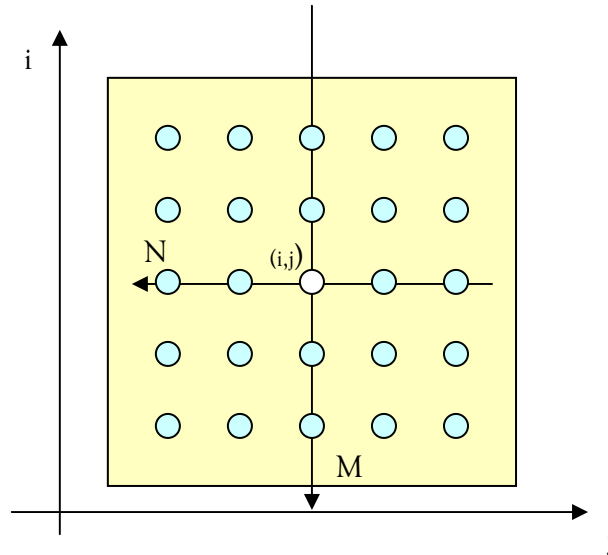


Fig. II.4. Voisinage non causal

b) Support de prédiction causal d'ordre (M,N)

La configuration causale est celle qui ressemble le plus à son homologue 1D. En effet l'effet récursif peut être appliqué aux deux dimensions. Les principaux supports qui ont été utilisés dans la littérature sont : le support Quart de Plan (QP) et le support Demi Plan Non Symétrique (DPNS).

- Le quart de plan « quarter plane QP »

$$y(i, j) = -\sum_{m=0}^M \sum_{n=0}^N \theta_{m,n} y(i-m, j-n) + e(i, j) \quad \text{ou} \quad (m, n) \neq (0,0) \quad (\text{II.12})$$

- Demi plan non symétrique DPNS. « non symmetric half plane »

$$y(i, j) = -\sum_{m=1}^M \sum_{n=-N}^N \theta_{m,n} y(i-m, j-n) + \sum_{n=1}^N \theta_{0,n} y(i, j-n) + e(i, j) \quad (\text{II.13})$$

Ce modèle a fait l'objet de nombreux travaux de recherche [Ala,85], [Fra,89], [Men,92], [Jit,93], [Xia,95], [Tak,97], [Kie,2001]. Les deux principaux avantages des modèles causaux sont d'offrir les possibilités, d'une part d'utiliser de nombreux algorithmes d'estimation de paramètres de modèles, comme les équations de Yules-Walker et les moindres carrés, et d'autre part, de pouvoir réaliser des traitements récursifs en employant les algorithmes de Levinson et Burg.

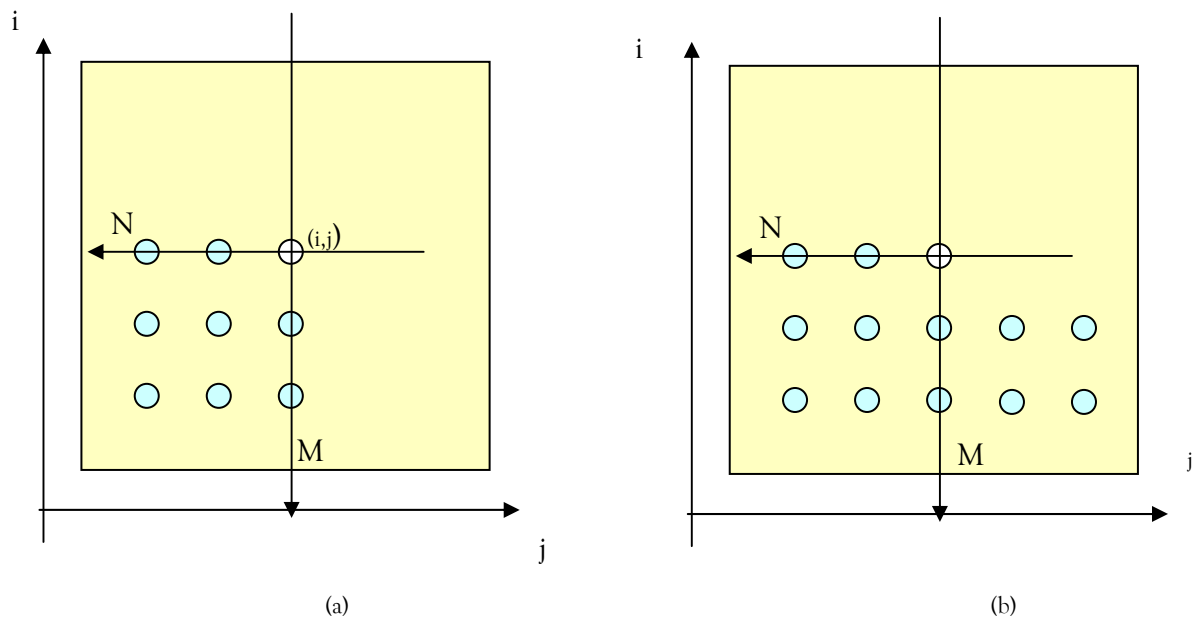


Fig.II.5. (a) Voisinage quart de plan, (b) Voisinage DPNS

c) Support de prédiction semi-causal d'ordre (M,N)

La figure II.6 montre un voisinage semi causal alors que l'équation (II.14) représente son expression mathématique.

$$y(i, j) = -\sum_{m=0}^M \sum_{n=-N}^N \theta_{m,n} y(i-m, j-n) + e(i, j) \quad \text{ou} \quad (m, n) \neq (0,0) \quad (\text{II.14})$$

Le modèle semi causal a fait l'objet de peu d'études [Jai81][Ran85] comparativement aux deux autres modèles, et il existe peu de méthodes d'estimation de paramètres. Cela peut s'expliquer de plusieurs manières. Tout d'abord, il est moins général que le modèle AR 2-D non causal. Du fait de sa semi causalité, il n'autorise que des traitements récursifs monodimensionnels, c.a.d ligne après ligne ou colonne après colonne. Il n'a pas de correspondance avec le modèle AR 1-D. En effet, les modèles AR 1-D causaux et non causaux ont été largement étudiés. De ce fait, des extensions au cas 2-D ont pu être réalisées, particulièrement en ce qui concerne les méthodes d'estimation des paramètres pour le modèle AR 2-D causal [Ala, 98].

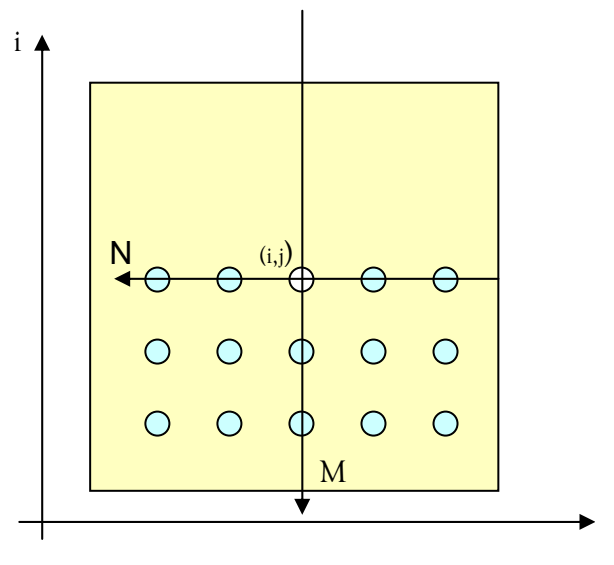


Fig. II.6 Voisinage semi causal

Pour notre part, nous avons opté pour le développement du modèle causal et du modèle non causal, car d'après la recherche bibliographique que nous avons menée, le premier a montré sa capacité de modéliser quelques textures naturelles [Ran,85], [Fra,88], [Fran,89], [Cha,90], [Xia,95], [Tak,97], [Ala,98]. En outre, il est simple à envisager et offre la possibilité d'utiliser des algorithmes récursifs d'estimation de paramètres. Quant au modèle non causal, il a lui aussi fait l'objet d'études multiples [Che,85], [Chel,85], [Ran,85], [Men,92], [Bad,93], [Jit,93], [Kie,2001], et permet une synthèse non récursive.

II.3 Estimation des paramètres d'un modèle AR appliqué à la texture.

L'estimation des paramètres AR peut s'effectuer selon de nombreuses méthodes : a) par les équations de Yules-Walker 1D et 2D représentant la relation entre la séquence d'autocorrélation du processus et les paramètres du modèle. B) la méthode des moindres carrés pour lesquelles l'estimation se fait en prenant en compte toutes les données de l'image. C) les méthodes fondées sur l'écriture des densités de

probabilité de l'image à étudier, comme celles basées sur le critère du maximum de vraisemblance. D) les méthodes adaptatives ; pour lesquelles l'estimation des paramètres est réalisée récursivement de site en site suivant un sens de parcours de l'image.

Nous présentons dans ce qui suit les méthodes que nous avons employées pour estimer les paramètres AR, en considérant la texture d'abord comme un processus auto régressif monodimensionnel (AR 1D) , puis comme un processus autorégressif bidimensionnel (AR 2D).

II.3.1. Modèle autorégressif monodimensionnel.

En mettant l'ensemble des pixels du support de prédiction D (de dimension MxN) bout à bout, on obtient un signal monodimensionnel qu'on appelle « vecteur voisinage ». L'image peut donc être modélisée par un modèle autorégressif monodimensionnel. Soit p la dimension du vecteur voisinage, le pixel à prédire $y(n)$ est lié à ses voisins par l'équation suivante :

$$y(n) = -\sum_{i=1}^p \theta_i y(n-i) + e(n) \quad (\text{II.15})$$

$$y(n) = \hat{y}(n) + e(n) \quad (\text{II.16})$$

Où $\hat{y}(n)$ est l'estimée de $y(n)$

II.3.1.1 Les équations de Yules-Walker

En multipliant l'équation (II.15) par $y(n-k)$, et en prenant l'espérance mathématique, on obtient :

$$\xi(y(n)y(n-k)) = -\sum_{i=1}^p \theta_i \xi(y(n-i)y(n-k)) + \xi(y(n-k)e(n)) \quad (\text{II.17})$$

$$R(k) = -\sum_{i=1}^p \theta_i R(k-i) + \xi[y(n-k)e(n)] \quad (\text{II.18})$$

Où $R(k)$ est l'autocorrélation du signal y . On suppose que le bruit $e(n)$ est décorrélé avec le passé de $y(n)$ [Ima]. En outre, le filtre est causal, on aura donc :

$$R(k) = -\sum_{i=1}^p \theta_i R(k-i), \quad \text{pour } k > 0 \quad (\text{II.19})$$

Pour $k=0$, si on multiplie l'équation (II.15) par $e(n)$ et on prend l'espérance mathématique, nous aurons :

$$\xi[y(n)e(n)] = \sigma^2 \quad (\text{II.20})$$

Et donc :

$$R(0) = -\sum_{i=1}^p \theta_i R(-i) + \sigma^2 \quad (\text{II.21})$$

En combinant les équations (II.19) et (II.21), nous obtenons :

$$R(k) = -\sum_{i=1}^p \theta_i R(k-i) + \sigma^2 \delta(k), \quad k \geq 0 \quad (\text{II.22})$$

$$\text{Avec :} \quad \delta(k) = \begin{cases} 1 & \text{si } k = 0 \\ 0 & \text{si } k \neq 0 \end{cases} \quad (\text{II.23})$$

L'équation (II.22) est connue sous le nom d'équation de Yules-Walker monodimensionnelle. Cette équation pour un support causal est donnée par l'expression matricielle suivante :

$$R\theta = R_0 \quad (\text{II.24})$$

D'où :

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \mathbf{M} \\ \theta_p \end{bmatrix} = \begin{bmatrix} R(0) & R(1) & R(2) & \Lambda & R(p-1) \\ R(1) & R(0) & R(1) & \Lambda & R(p-2) \\ R(2) & R(1) & R(0) & \Lambda & R(p-3) \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & & \\ R(p-1) & R(p-2) & R(p-3) & \Lambda & R(0) \end{bmatrix}^{-1} \cdot \begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \mathbf{M} \\ R(p) \end{bmatrix} \quad (\text{II.25})$$

La matrice R dans l'équation (II.25) est une matrice d'autocorrélation de forme Toeplitz. (Une matrice de Toeplitz est une matrice où tous les éléments dans chaque diagonale sont égaux). p est la dimension du vecteur voisinage, il représente l'ordre du modèle.

$R(k)$ est l'autocorrélation du processus y . Pour un processus stationnaire et ergodique l'autocorrélation statistique peut être calculée comme une autocorrélation spatiale. Ce qu'il n'est pas possible de faire pour un signal non stationnaire.

$$R(k) = \frac{1}{N} \sum_{n=1}^N y(n)y(n-k) \quad (\text{II.26})$$

Il existe par contre des processus qui sont localement stationnaires, et la théorie précédemment exposée peut alors être appliquée.

Les équations de Yules-Walker peuvent être résolues en inversant la matrice d'autocorrélation R . Cependant le coût de calcul d'une telle approche peut s'avérer gênant lorsque l'ordre du modèle est

important. Il est possible de résoudre ces équations dans le cas d'un support de prédiction causal en utilisant l'algorithme de Levinson-Durbin, avec un coût de calcul faible qui est de l'ordre $O(p^2)$ au lieu de $O(p^3)$, ou encore en utilisant l'algorithme de Burg.

- **L'algorithme de Levinson-Durbin 1D.**

C'est un algorithme qui permet de résoudre tout système du type $Ax = b$ avec A de forme Toeplitz, donc en particulier les équations de Yules-Walker.

En se servant du fait que la matrice d'autocorrélation dans l'équation (II.25) est de forme Toeplitz, et que le vecteur colonne du côté droit de cette équation comporte les mêmes éléments que ceux composant la matrice d'autocorrélation. Levinson et Durbin ont proposé un algorithme récursif pour résoudre ce type d'équations [Lev,47].

L'idée principale de la méthode de Levinson-Durbin est de procéder récursivement, en commençant par un modèle d'ordre 1 (un seul coefficient), et puis d'incrémenter l'ordre. La solution à l'ordre $p+1$ est obtenue en utilisant la solution à l'ordre p . L'algorithme de Levinson-Durbin se présente comme suit :

$$\begin{aligned}
 \sigma_0^2 &= E_0 = R(0) \\
 k_i &= - \left[R(i) + \sum_{j=0}^{i-1} \theta_j^{(i-1)} R(i-j) \right] / E_{i-1} \quad 1 \leq i \leq p \\
 \theta_i^{(i)} &= k_i \\
 \theta_j^{(i)} &= \theta_j^{(i-1)} + k_i \theta_{i-j}^{(i-1)} \quad , \quad 1 \leq j \leq i-1 \\
 \sigma_i^2 &= E_i = (1 - k_i^2) E_{i-1}
 \end{aligned} \tag{II.27}$$

Pour un modèle AR d'ordre p , L'équation (II.27) est résolue récursivement pour $i=1,2,\dots,p$. La solution finale est donnée par :

$$\theta_i = \theta_i^{(p)} \quad , \quad 1 \leq i \leq p$$

Dans cet algorithme, on passe des coefficients du modèle d'ordre i aux coefficients du modèle de l'ordre $i+1$, grâce au seul coefficient k_i dénommé *coefficient de corrélation partielle*. Ce coefficient représente la partie de la fonction d'autocorrélation que l'on peut déduire de l'observation de $i+1$ valeurs successives du signal après avoir retiré la partie de la fonction d'autocorrélation que l'on déduit de l'observation de i valeurs successives. Ainsi, si $y(n)$ est un signal modélisé par un modèle AR d'ordre p , toute la corrélation est obtenue dans l'observation de p valeurs successives du signal et $K_p=0$ [Max,87].

La dernière étape de l'algorithme montre que les coefficients de corrélation partielle sont de module inférieur à 1, puisque σ^2 est toujours positive.

L'algorithme de Levinson-Durbin incorpore le calcul de la puissance (la variance) de l'erreur E_i à chaque itération. Cette variance diminue ou reste la même lorsque l'ordre du modèle augmente. E_i n'est jamais négative naturellement puisque c'est une erreur quadratique. Nous devons donc avoir :

$$0 \leq E_i \leq E_{i-1}, \quad E_0 = R(0)$$

L'évolution de cette variance avec l'ordre p , donne un moyen de contrôle de l'ordre adapté au signal, on peut convenir d'arrêter la récurrence lorsque la décroissance de cette variance avec l'ordre devient nulle ou faible. Nous avons employé l'algorithme de Levinson pour estimer les paramètres du modèle AR avec un support quart de plan.

La méthode de résolution des équations de Yules-Walker par inversion matricielle ou par l'algorithme de Levinson peut conduire à une solution d'un filtre non stable. La méthode de Burg garantit un filtre stable, mais la solution des équations n'est pas exactement identique à la solution directe ou de Levinson. L'idée de Burg est de calculer directement à partir des données une estimée des coefficients de réflexion K_n et ce, sans passer par le calcul des autocorrélations [Guy, 2007].

II.3.1.2 Méthode des moindres carrés

Reprenons l'équation (II.3) :

$$e(n) = y(n) - \hat{y}(n) = y(n) + \sum_{i=1}^p \theta_i y(n-i) \quad (\text{II.28})$$

Soit E l'espérance de l'erreur quadratique.

$$E = \xi(e^2(n)) = \xi\left(y(n) + \sum_{i=1}^p \theta_i y(n-i)\right)^2 \quad (\text{II.29})$$

Par la méthode des moindres carrés, les paramètres θ_i sont obtenus en minimisant E par rapport à chacun des paramètres :

$$\frac{\partial E}{\partial \theta_i} = 0 \quad 1 \leq i \leq p$$

De (II.28) et (II.29), on obtient l'ensemble d'équations suivant :

$$\sum_{i=1}^p \theta_k \xi [(y(n-i)y(n-k))] = -\xi[y(n)y(n-k)] \quad 1 \leq k \leq p \quad (\text{II.30})$$

En supposant que $y(n)$ est stationnaire, on a :

$$\xi(y(n-i)y(n-k)) = R(k-i). \quad (\text{II.31})$$

Si on considère que le signal $\{y(s)\}$ n'est pas nul en dehors du support de prédiction, et que la minimisation de l'erreur quadratique est calculée sur l'ensemble de l'image, nous obtenons :

$$\theta = X^{-1}.X_0 = \left[\sum_{\Omega} g_n g_n^T \right]^{-1} \left(\sum_{\Omega} g_n y(n) \right) \quad (\text{II.32})$$

$$\sigma^2 = \frac{1}{M^2} \left[\sum_{\Omega} (y(n) - \theta g_n)^2 \right] \quad (\text{II.33})$$

M : étant la taille de l'image. Ω : étant l'ensemble des M^2 pixels de l'image, et $g_n = \text{Col}[y(n+i), 1 \leq i \leq p]$.

L'ensemble d'équations (II.32) est connu sous la dénomination « *équations normales* ». Elles sont formées d'un ensemble de p équations à p inconnus dont la résolution permet de trouver le jeu de coefficients $\{\theta_i\}$ minimisant l'erreur totale E .

La matrice X est une matrice d'autocorrélation qui n'a pas une forme de Toeplitz, mais elle est symétrique ($X_{i,j}=X_{j,i}$). Les coefficients peuvent être obtenus par différentes méthodes telles que la décomposition de Cholesky [Ran,94] ou par simple inversion de la matrice.

Nous avons utilisé cette méthode pour estimer les coefficients du modèle AR de la texture avec un support causal et non causal.

II.3.1.3 Application

Les méthodes d'analyse, de synthèse et de classification que nous avons développées ont été testées sur des images naturelles extraites de l'album de Brodatz [Bro, 66]. Ce sont des images de taille 512x512, d'où nous avons extraits des échantillons de taille 128x128. Nous avons représenté dans la figure II.7 quelques unes de ces textures.

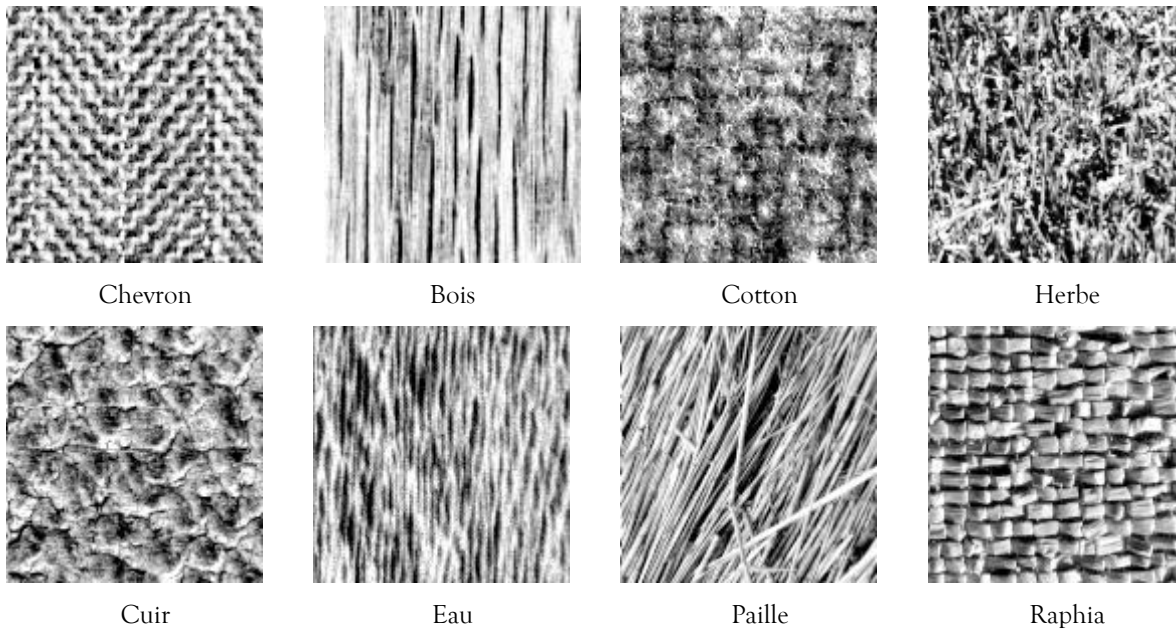


Fig. II.7. Images de Brodatz.

II.3.1.3.1 Cas d'un support causal QP.

Dans le cas d'un support causal, la relation entre le niveau de gris du pixel de coordonnées (i,j) et son voisinage est décrite par l'équation aux différences suivante :

$$y(i, j) = -\sum_{m=0}^M \sum_{n=0}^N \theta_{m,n} y(i-m, j-n) + e(i, j) \quad \text{ou} \quad (m,n) \neq (0,0) \quad (\text{II.34})$$

a) Estimation des paramètres par Levinson 1D

En utilisant une fenêtre d'analyse d'une taille donnée, celle-ci est parcourue ligne par ligne et les pixels sont mis bout à bout pour construire le vecteur voisinage monodimensionnel. Pour donner un ordre d'idée sur les paramètres AR, nous présentons dans le tableau II.1 les quinze premiers paramètres pour les textures 'Bois', 'Coton', et 'Herbe', en utilisant une fenêtre d'analyse 10x10. Figure II.8.

Paramètres	Bois	Coton	Herbe	Paramètres	Bois	Coton	Herbe
θ_1	0.9243	0.8562	0.7041	θ_9	0.0095	0.0019	0.0239
θ_2	-0.0282	0.0851	-0.0136	θ_{10}	0.3059	0.2300	0.5981
θ_3	0.0472	-0.0056	0.0784	θ_{11}	-0.300	-0.1632	-0.3947
θ_4	0.0192	0.0187	0.0219	θ_{12}	-0.0068	-0.0496	-0.0401
θ_5	0.0165	0.0050	0.0223	θ_{13}	0.0106	0.0082	-0.0404
θ_6	-0.0156	-0.0064	0.0150	θ_{14}	0.0088	-0.00006	-0.0153
θ_7	-0.0041	0.0165	0.0081	θ_{15}	0.0042	-0.0028	0.0180
θ_8	-0.0188	-0.0254	-0.0189				

Tableau II.1 Coefficients AR 1D pour une fenêtre 10x10

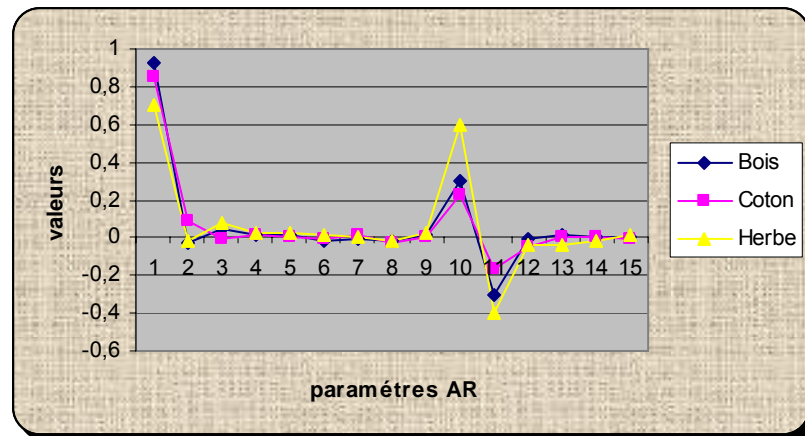


Fig.II.8 Tracés des coefficients pour une fenêtre 10x10

Pour toutes les textures, nous avons constaté que le coefficient le plus important correspond au pixel le plus proche du pixel à prédire, du fait que l'autocorrélation est plus grande. En effet, les valeurs des paramètres décroissent en s'éloignant du pixel à prédire. Néanmoins, nous remarquons un pic au dixième paramètre AR. Ce pic correspond au pixel le plus proche du pixel à prédire dans la direction verticale.

Variance de l'erreur d'estimation et choix de l'ordre du modèle.

La variance de l'erreur d'estimation dépend de la méthode d'estimation, de l'ordre du modèle (la taille de la fenêtre d'analyse), de la forme du voisinage, et bien évidemment de la texture analysée. La méthode d'estimation la plus précise est celle qui donne la plus petite variance pour un ordre donné. L'évolution de cette variance avec l'ordre, donne un moyen de contrôle de l'ordre adapté à la texture, nous pouvons dire que l'ordre est adéquat lorsque la décroissance de la variance de l'erreur au-delà de cet ordre devient nulle ou faible.

Nous avons tracé la variance de l'erreur en fonction de l'ordre pour les textures 'Cuir', 'Raphia', 'Coton', et 'Bois' (Tableau II.2). Pour calculer cette variance à un ordre donné il faut d'abord estimer pour chacune des textures de taille 128x128 les paramètres AR. Nous remarquons que la variance est grande mais diminue lorsqu'on augmente l'ordre.

Ordre	Cuir	Coton	Raphia	Bois
10x10	815,17	801,94	828,75	986,00
20x20	355,22	396,61	333,61	326,57
30x30	247,30	299,46	218,98	193,02
40x40	202,01	256,81	179,44	129,12

Tableau II.2 Evolution de la variance en fonction de l'ordre

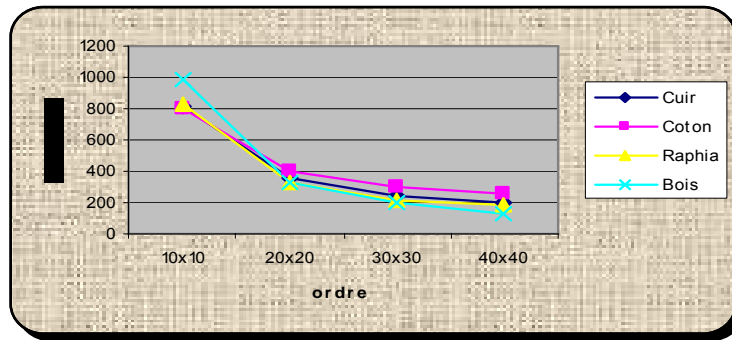


Fig.II.9. Variance en fonction de l'ordre du modèle.

Par ailleurs, rappelons que la modélisation AR suppose que le processus à analyser est stationnaire. Or les textures que nous avons à notre disposition ne le sont pas. De ce fait, la position de la fenêtre d'analyse conditionne l'estimation des paramètres. Ainsi, les paramètres estimés pour deux portions différentes de texture ne sont pas parfaitement égaux. Pour vérifier ceci nous avons calculé les paramètres des textures 'Bois' et 'Sable', avec un ordre 10x10 en prenant à chaque fois une fenêtre d'analyse différente. Le tableau (II.3) montre les résultats obtenus avec les 15 premiers paramètres.

Bois		Sable	
Portion1	Portion2	Portion1	Portion2
0.9277	0.9397	0.9318	0.9266
-0.0224	-0.0284	-0.0373	-0.0366
0.0422	0.0391	0.0273	0.0329
0.0169	0.0108	0.0051	0.0044
0.0098	-0.0021	0.0108	0.0094
-0.0068	0.0099	0.0012	0.0021
-0.0079	-0.0094	-0.0016	-0.0011
-0.0315	-0.0281	-0.0016	-0.0076
0.0176	0.0199	0.0416	0.0484
0.3171	0.3061	0.2712	0.2763
-0.3044	-0.2950	-0.2672	-0.2736
-0.0069	-0.0095	-0.0190	-0.0161
0.0124	0.0118	0.0022	-0.0025
0.0055	0.0026	-0.0021	0.0004
0.0032	0.0002	0.0059	0.0061

Tableau II.3. Paramètres estimés avec deux portions de textures différentes

Les résultats obtenus montrent que les différences sont petites, mais cumulées elles sont importantes, et vont affecter la précision du modèle. Pour éluder ce problème, nous avons pensé à introduire une modification à la procédure d'estimation de la fonction d'autocorrélation. Pour un ordre donné, son calcul se fait pour différentes positions à l'aide d'un estimateur biaisé, et la moyenne des autocorrélations est prise [Kou, 2007].

Effet de l'utilisation de l'algorithme de Levinson 1D

Pour mieux voir l'effet de l'utilisation de l'algorithme de Levinson sur le temps de calcul, nous avons tracé en fonction de l'ordre du modèle le temps écoulé dans l'estimation des paramètres pour une texture donnée (par exemple : la texture « coton »). Le temps de calcul dépend évidemment de la vitesse du processeur (dans notre cas un PIII avec 800 MHz).



Fig. II.10. Temps de calcul des paramètres AR 1D en fonction de l'ordre

Nous remarquons que l'algorithme de Levinson permet un gain de temps considérable.

b) Estimation des paramètres par la méthode des moindres carrés.

L'utilisation de l'algorithme des moindres carrés requiert de faire deux choix : le premier est lié à l'ordre (fenêtre d'analyse), et le deuxième concerne la taille de l'échantillon utilisé pour calculer la somme des carrés [Kou, 2007]. La fenêtre d'analyse doit être plus grande que la taille de la primitive afin de récupérer les caractéristiques statiques de la texture, et avoir des similitudes entre la texture originale et synthétisée. En outre, la taille de l'échantillon joue un rôle important, plus elle est importante mieux les textures sont modélisées.

Pour établir une comparaison avec l'approche précédente, nous donnons dans le Tableau.II.4 l'évolution de la variance de l'erreur en fonction de l'ordre pour les textures 'Bois', 'Coton', et 'Chevron'.

Ordre	Bois	Coton	Chevron
3x3	27.31	181.97	214.29
7x7	25.35	168.15	174.18
9x9	24.53	160.58	167.03
12x12	23.10	150.53	149.15

Tableau II.4 Evolution de la variance

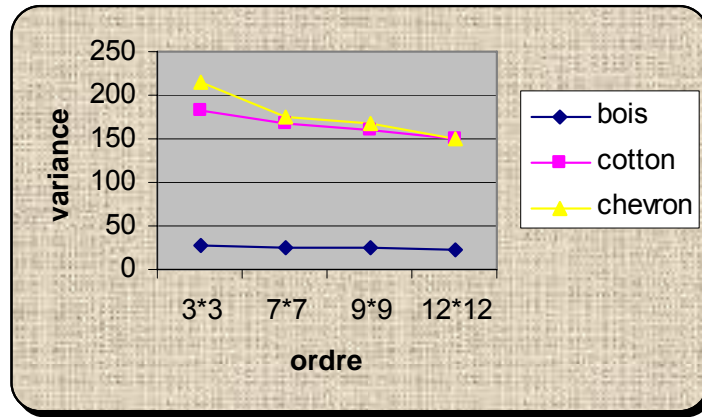


Fig. II.11. Evolution de la variance en fonction de l'ordre. Taille d'échantillon 100x100

Nous constatons que la variance dans ce cas, est beaucoup plus faible que dans l'approche précédente, et qu'elle se stabilise rapidement, ce qui indique que les résultats de la synthèse seront meilleurs et avec des ordres plus faibles.

II.3.1.3.2 Cas d'un support non causal

Le voisinage non causal est le plus général. L'intensité du pixel dépend des niveaux de gris des pixels qui lui sont voisins dans toutes les directions, comme le montre l'équation suivante :

$$y(i, j) = - \sum_{m=-M}^M \sum_{n=-N}^N \theta_{m,n} y(i-m, j-n) + e(i, j) \quad \text{ou} \quad (m, n) \neq (0,0) \quad (\text{II.35})$$

L'estimation des paramètres d'un modèle AR non causal peut se faire avec les équations de Yules-Walker 1D (Y-W), ainsi qu'avec la méthode des moindres carrés. Vu que la variance d'estimation calculée par la méthode des moindres carrés est beaucoup plus faible que par les équations de Y-W, nous avons alors utilisé les moindres carrés pour le support non causal.

En mettant l'ensemble des pixels du voisinage bout à bout, on obtient un vecteur voisinage monodimensionnel $\{y(s)\}$. Soit N' la dimension du vecteur voisinage, le pixel à prédire $y(s)$ est lié à ses voisins par l'équation suivante :

$$y(s) = - \sum_{r=1}^{N'} \theta_r y(s-r) + e(s) \quad (\text{II.36})$$

Cependant, pour une image de dimensions finies, le voisinage pour les pixels de bords n'est pas complet. Pour surmonter ce problème nous avons supposé une représentation torique de l'image [Bad,93], [Ala,98] :

$$y(s+r) = y((i, j) + (p, q)) = y[(p+i-1) \bmod T, (q+j-1) \bmod T] \quad (\text{II.37})$$

$$\text{si } (s+(p, q)) \notin [0, T-1] \times [0, T-1]$$

Où T est la taille de l'image.

L'hypothèse de la stationnarité de la texture implique un voisinage symétrique [Che,85], [Bad,93], [Rup,99]. En d'autres termes, les pixels séparés de la même distance du pixel à prédire auront le même coefficient AR (Figure II.12). Le nombre de paramètres est donc réduit : quatre paramètres à déterminer dans le cas d'un voisinage 3×3 .

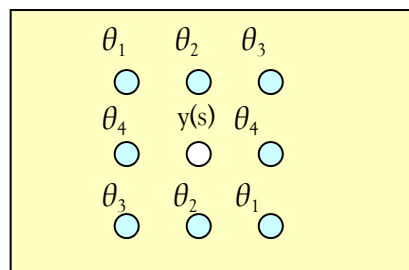


Fig. II.12. Voisinage non causal d'ordre 3×3

Nous avons représenté pour les mêmes textures que précédemment l'évolution de la variance de l'erreur en fonction de l'ordre (Tableau II.5). La figure II.13 fournit l'évolution de cette variance pour différentes textures. La variance de l'erreur de prédiction décroît jusqu'à une valeur asymptotique. Ainsi, travailler au-delà d'un certain ordre, qui sera déterminé empiriquement, n'apportera rien.

Ordre	Eau	Bois	Chevron	herbe	sable
3x3	18.31	15.6	111.43	521.2	95.58
5x5	17.65	15.36	103.29	451.22	84.38
7x7	17.52	15.20	102.2	436.72	82.70
9x9	17.49	15.15	101.59	434.9	82.09
11x11	17.43	15.10	101.81	435.61	82.15
13x13	17.34	15.01	101.41	434.35	81.98

Tableau II.5 Evolution de la variance en fonction de l'ordre.

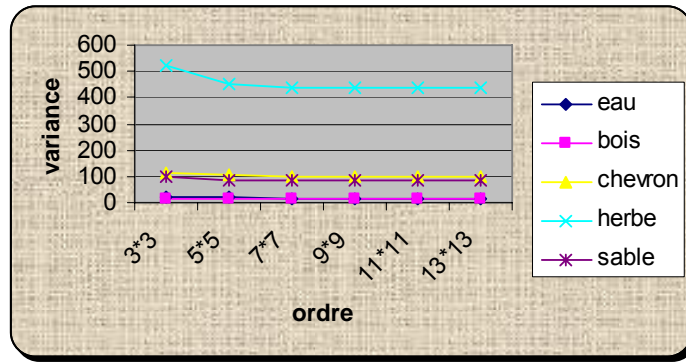


Fig. II.13. Evolution de la variance en fonction de l'ordre.

II.3.2. Modèle autorégressif bidimensionnel

Rappelons l'équation (II.1) représentant la relation entre un pixel de coordonnées (i,j) et son voisinage :

$$y(i, j) = -\sum_{(m,n) \in D} \theta_{m,n} y(i-m, j-n) + e(i, j) \tag{II.38}$$

D étant le support de prédiction. Les coefficients de prédiction du modèle $\{\theta_{m,n}\}, (m,n) \in D$ sont choisis, à l'instar du modèle AR 1D, de manière à minimiser l'erreur quadratique moyenne entre la valeur réelle et la valeur prédite du pixel. $e(i, j)$ est l'erreur d'estimation et est assimilée à un bruit blanc de variance σ^2 , et satisfait la condition d'orthogonalité suivante :

$$\xi[e(i, j)y(k, l)] = \sigma^2 \delta(i-k, j-l) \tag{II.39}$$

En multipliant l'équation (II.38) par $y(i-k, j-l)$, et en prenant l'espérance mathématique, on obtient les équations de Yules-Walker 2D :

$$R_y(k, l) + \sum_{(m,n) \in D} \theta_{m,n} R_y(k-m, l-n) = \begin{cases} 0 & (k, l) \in D \\ \sigma^2 & (k, l) = (0,0) \end{cases} \tag{II.40}$$

R_y Représente l'autocorrélation de l'image. Dans le cas d'un support Quart de plan (QP) de dimension (M,N) , les équations de Yules-Walker 2D s'écrivent sous la forme matricielle suivante :

$$\Re \theta = \Gamma \tag{II.41}$$

$$\begin{bmatrix} \mathfrak{R}[0] & \mathfrak{R}[-1] & \Lambda & \mathfrak{R}[-M] \\ \mathfrak{R}[1] & \mathbf{O} & \mathbf{O} & \\ \mathbf{M} & \mathbf{O} & \mathbf{O} & \mathfrak{R}[1] \\ \mathfrak{R}[M] & \Lambda & \mathfrak{R}[1] & \mathfrak{R}[0] \end{bmatrix} \begin{bmatrix} \theta[0] \\ \theta[1] \\ \mathbf{M} \\ \theta[M] \end{bmatrix} = \begin{bmatrix} \rho \\ 0 \\ \mathbf{M} \\ 0 \end{bmatrix} \quad (\text{II.42})$$

La matrice \mathfrak{R} est Toeplitz Bloc Toeplitz, avec :

$$\mathfrak{R}[l] = \begin{bmatrix} R_y(0,l) & R_y(-1,l) & \Lambda & R_y(-N,l) \\ R_y(1,l) & \mathbf{O} & \mathbf{O} & \mathbf{M} \\ \mathbf{M} & \mathbf{O} & \mathbf{O} & R_y(-1,l) \\ R_y(N,l) & \Lambda & R_y(1,l) & R_y(0,l) \end{bmatrix} \quad (\text{II.43})$$

Le vecteur des paramètres est donné par :

$$\theta = [\theta[0] \Lambda \theta[M]]^T \quad \theta[l] = [\theta_{0,l} \Lambda \theta_{N,l}] \quad (\text{II.44})$$

Avec $\theta_{0,0} = 1$

Et le vecteur ρ est donné par :

$$\rho = [\sigma^2 \quad 0 \quad \Lambda \quad 0]^T \quad (\text{II.45})$$

Les équations de Yules-Walker 2D entraînent l'utilisation d'une grande matrice d'autocorrélation $((M+1) \times (N+1), (M+1) \times (N+1))$. On a alors recours à l'utilisation de l'algorithme levinson2D qui permet de réduire le temps de calcul [Wig,65], [Mar,80], [The,81], [Kud,91], [Weu,92], [Law,2000].

- **L'algorithme de Levinson 2D**

L'algorithme de Levinson que nous avons utilisé a été proposé par W.Therrien [The,81],[The,89]. Il permet de lier le modèle autorégressif 2D au modèle AR 1D multi canal construit pour le support QP.

Considérons l'image comme la réalisation d'un processus AR 1D multi canal (Figure II.14). Pour un support QP de dimensions $M \times N$, chaque ligne de l'image représente un canal, les colonnes de l'image sont des vecteurs $Y[n]$ formés des éléments $y_j[n]$, $j=0,1,\dots,M$, $n=0,1,\dots,N$.

Un signal multi canal $Y[n]$ avec $(M+1)$ canaux, obéissant à un modèle AR, satisfait l'équation suivante [Khu,93] :

$$Y[n] = -\sum_{i=1}^N A[i]Y[n-i] + e(n) \tag{II.46}$$

Avec :

$$Y[n] = [y_0[n] \quad y_1[n] \Lambda \quad y_M[n]]^T \tag{II.47}$$

Où $A[i]$ ($i=1,2, \dots, N$) sont les matrices de coefficients de prédiction de dimension $(M+1) \times (M+1)$. L'erreur de prédiction est donnée par :

$$e[n] = Y[n] - \hat{Y}[n] = Y[n] + \sum_{i=1}^N A[i]Y[n-i] \tag{II.48}$$

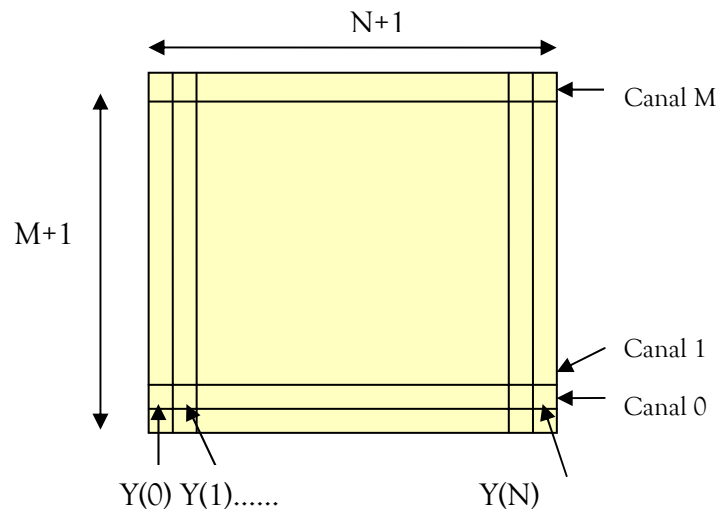


Fig. II.14. Représentation multi canal d'une image.

Les équations de Yules-Walker multi canal [Khu,93] sont donnés par :

$$R_m \cdot A = \Psi \quad (m \text{ pour multi canal}) \tag{II.49}$$

Avec :

$$R_m = \begin{bmatrix} R_m[0] & R_m[1] & \Lambda & R_m[N] \\ R_m[-1] & R_m[0] & \Lambda & R_m[N-1] \\ \text{M} & \text{M} & & \text{M} \\ R_m[-N] & R_m[-N+1] & \Lambda & R_m[0] \end{bmatrix} \tag{II.50}$$

et

$$R_m[k] = \begin{bmatrix} r_{00}[k] & r_{01}[k] & \dots & r_{0M}[k] \\ r_{10}[k] & M & & M \\ \vdots & \vdots & \ddots & \vdots \\ r_{M0}[k] & \Lambda & & r_{MM}[k] \end{bmatrix} \quad (\text{II.51})$$

Où :

$$r_{ij}[k] = E\{y_i[n].y_j[n-k]\} \quad (\text{II.52})$$

Le jeu de matrices $A[i]$ ($i=1,2, \dots, N$) est donné par :

$$A = [I \quad A[1] \quad \Lambda \quad A[N]]^T \quad (\text{II.53})$$

Et

$$\Psi = [P \quad 0 \quad \Lambda \quad 0]^T \quad (\text{II.54})$$

P : étant la matrice de covariance de l'erreur minimale.

0 : est une matrice nulle de dimension $(M+1) \times (M+1)$.

I : est la matrice identité de dimension $(M+1) \times (M+1)$.

La solution des équations de Yules-Walker multi canal peut être utilisée pour trouver les coefficients AR 2D en considérant que :

$$r_{ij}[k] = R_y[-k, i-j] \quad i, j=0, 1, \dots, M \quad (\text{II.55})$$

R_y représente l'autocorrélation de l'image.

Et en choisissant :

$$\rho = P\theta[0] \quad (\text{II.56})$$

On obtient :

$$\theta[k] = A^T[k]\theta[0] \quad k=1, 2, \dots, N \quad (\text{II.57})$$

Les équations (II.56) et (II.57) relient la prédiction linéaire 1D multi canal à la prédiction linéaire 2D, et conduisent à la résolution des équations de Yules-Walker 2D (éq. II.42) de manière récursive.

Les matrices de coefficients $A[i]$ ($i=1, 2, \dots, N$) sont déterminés récursivement grâce à l'algorithme de Levinson multi canal [The,89]. Cet algorithme a été repris par Marple [Mar,2000] , avec des simplifications additionnelles, et se présente comme suit :

Pour initialisation, on prend $P_1 = \mathfrak{R}[0]$, et $A_1[1] = I$

pour $1 \leq p \leq N-1$ *faire* :

$$\Delta_{p+1} = \mathfrak{R}[p+1] + \sum_{k=1}^p A_p[k] \mathfrak{R}[p+1-k] \quad (\text{II.58})$$

$$K_{p+1} = A_{p+1}[p+1] = -\Delta_{p+1} (JP_p^* J)^{-1} \quad (\text{II.59})$$

$$A_{p+1}[k] = A_p[k] + K_{p+1} (JA_p^*[p+1-k]J) \quad \text{pour } 1 \leq k \leq p \quad (\text{II.60})$$

$$P_{p+1} = (I - K_{p+1} (JK_{p+1}^* J)) P_p \quad (\text{II.61})$$

Avec

$$J = \begin{bmatrix} 0 & 0 & \Lambda & 1 \\ 0 & & 1 & 0 \\ M & N & & M \\ 1 & \Lambda & 0 & 0 \end{bmatrix}$$

Une fois les matrices de coefficients déterminés, les coefficients du modèle AR 2D sont calculés à partir des équations (II.56) et (II.57), et en normalisant $\theta[0]$ de manière à ce que $\theta_{0,0}$ soit égal à 1.

II.3.2.1 Application

Nous avons tracé, pour des textures différentes, la variation de la variance en fonction de l'ordre du modèle. Tableau II.6.

Nous notons que la variance de l'erreur est beaucoup plus grande que dans le cas 1D, et ceci pour toutes les textures. Ceci laisse présager que la qualité de synthèse sera moins bonne.

Ordre	Cuir	Coton	Raphia	Bois
10x10	1291,05	1214,79	1126,59	1376,84
20x20	538,72	459,46	437,06	317,00
30x30	318,00	331,36	242,05	183,21
40x40	250,15	266,29	191,34	137,82

Tableau II.6 Evolution de la variance

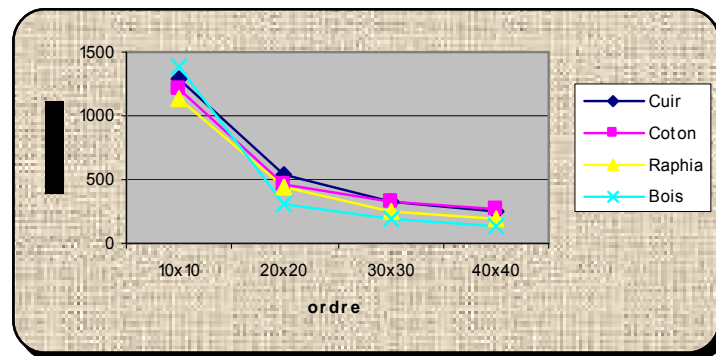


Fig. II.15. Evolution de la variance par Levinson 2D.

II.3.3 autres méthodes d'estimation (Le maximum de vraisemblance)

La méthode de maximum de vraisemblance est basée sur l'hypothèse que le processus à analyser possède une densité de probabilité connue a priori. On exprime alors, une probabilité conditionnelle pour ce processus dont l'optimisation fournit l'ensemble des paramètres du modèle associé au processus. En outre, cette méthode nécessite, pour une convergence rapide, une bonne initialisation des paramètres du modèle avant la procédure de minimisation. En effet un des défauts majeurs de cette méthode est le risque d'obtenir un maximum local de la probabilité conditionnelle [Ala, 98]. Par ailleurs, cette méthode exige un temps calculatoire excessive, ce qui la rend moins adoptée dans l'estimation des paramètres AR.

Conclusion

Quand on veut modéliser une texture par un modèle AR, on doit trouver les paramètres du modèle l'ayant engendré. Ces paramètres constituent des attributs texturaux qu'on peut utiliser éventuellement dans un procédé de synthèse, de segmentation ou de classification.

Lors de l'estimation des paramètres AR, plusieurs méthodes s'offrent à nous : les équations de Yules-Walker 1D et 2D, ainsi que la méthode des moindres carrés. En outre, plusieurs types de supports de prédiction peuvent être envisagés : causal, semi causal ou non causal.

Ainsi, il existe deux manières d'aborder le problème de la modélisation AR de textures :

La première est de considérer la texture comme un processus AR 1D en mettant bout à bout les pixels du support de prédiction. Dans ce cas, l'estimation des paramètres du modèle peut se faire grâce aux équations de Yules-Walker monodimensionnelles, qu'on peut résoudre de manière rapide grâce à l'algorithme de Levinson 1D, ou grâce à la méthode des moindres carrés aboutissant aux équations normales qui sont résolues par inversion de matrice.

La deuxième manière est de considérer la texture comme un processus AR 2D, aboutissant aux équations de Yules-Walker 2D. La matrice d'autocorrélation est de forme Toeplitz Bloc Toeplitz. L'algorithme de Levinson 2D permet de résoudre de façon récursive ces équations dans le cas d'un support quart de plan.

En modélisant la texture à travers un modèle paramétré qu'est le modèle AR on suppose que ce modèle est valable pour toute la texture. Ainsi, on se place implicitement dans un cadre stationnaire.

Mais dans la pratique, les variations statistiques des textures naturelles induisent des paramètres différents pour des portions différentes d'une même texture.

Dans le chapitre suivant, nous allons considérer le problème de la validation du modèle estimé pour une texture particulière, et ceci en adoptant deux solutions. La première est d'étudier le problème inverse qui consiste à synthétiser les textures à partir de leurs modèles estimés, et de comparer ensuite les textures synthétisées aux textures originales. La deuxième solution consiste à intégrer les paramètres estimés dans un processus de classification.

Chapitre 3 Synthèse de texture et classification

Nous avons présenté, dans le chapitre précédent, les algorithmes d'estimation des paramètres AR. Dans ce chapitre, et en vue de valider le modèle estimé nous allons intégrer les paramètres du modèle dans un processus de synthèse, et comparer ensuite, les textures synthétisées aux textures originales. Le deuxième moyen de validation consiste à intégrer ces paramètres dans un processus de classification.

III.1 Synthèse de texture

La synthèse est la génération d'une texture correspondant à la texture modélisée en utilisant les paramètres estimés du modèle. Le résultat de la synthèse dépend de plusieurs facteurs :

- Le choix de la méthode d'estimation des paramètres AR.
- Le choix de l'ordre du modèle.
- Le choix du support de prédiction (causal, non causal ...).
- La nature de la texture (homogène, non homogène, structurale, stochastique, etc).

La synthèse représente un outil d'évaluation du modèle et de la méthode d'estimation.

III.1.1 Synthèse basée sur les paramètres estimés par Levinson 1D.

Pour évaluer les paramètres estimés par l'algorithme de Levinson 1D, nous les avons utilisés pour synthétiser les textures ayant servi à les estimer. Avant de pouvoir entamer le processus de synthèse, il faut affecter des intensités aux premières lignes et aux premières colonnes. Nous avons affecté à ces lignes et colonnes les intensités de la texture modélisée [Kou, 2007]. Nous avons remarqué que pour des ordres faibles, la synthèse de textures ne présente aucune ressemblance avec les textures originales. C'est à partir d'une fenêtre 20x20 que la synthèse commence à donner des résultats proches de ceux désirés, car la variance de l'erreur se stabilise à partir de cet ordre. La figure III.1 représente deux exemples de textures synthétisées pour des fenêtres 20x20 et 40x40.

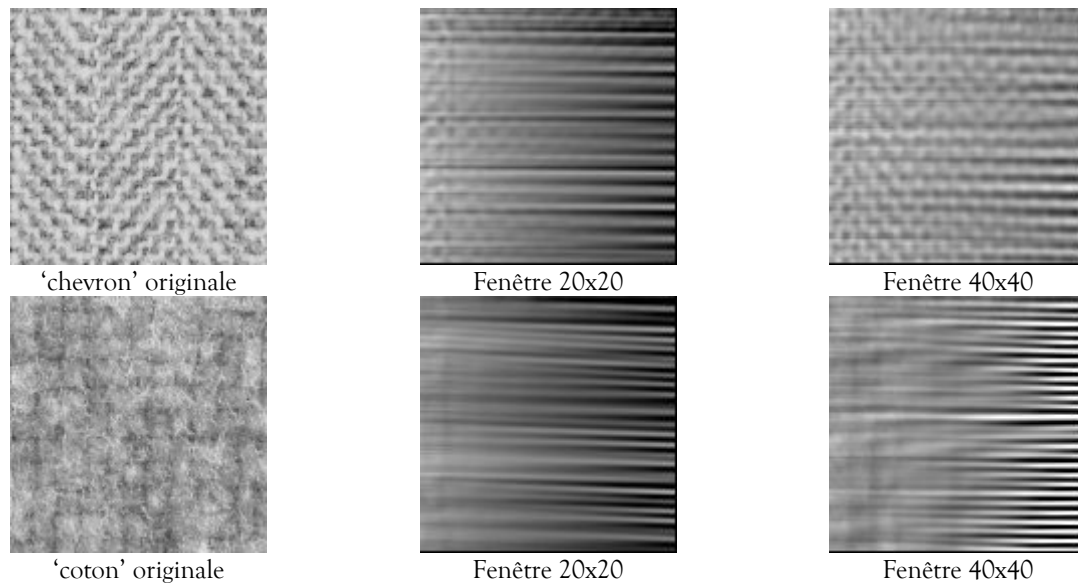


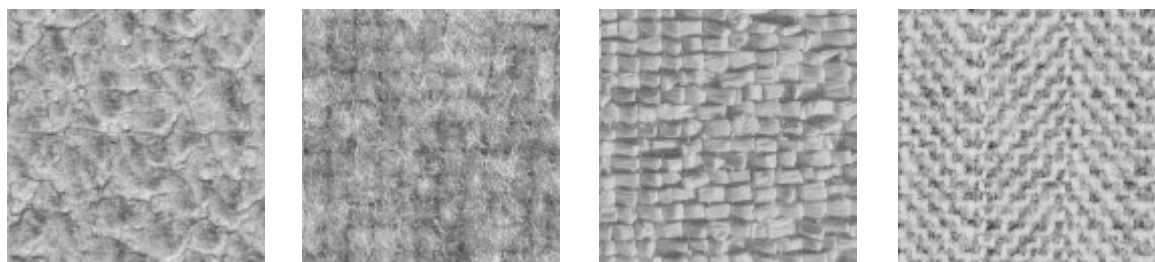
Fig. III.1. Textures synthétisées par les paramètres estimés par Levinson 1D

Ces résultats montrent qu'avec les paramètres estimés par l'algorithme de Levinson 1D, les textures synthétisées se dégradent progressivement. Cette dégradation est due à l'erreur d'estimation (Tableau II.2). En augmentant la fenêtre d'analyse de 20x20 à 40x40, la variance diminue et les erreurs se propagent moins vite. Même si on prend de plus grandes fenêtres, cet effet persiste induisant ainsi une imprécision de la modélisation et donc une mauvaise qualité d'images synthétisées, du fait que la variance de l'erreur se stabilise et ne diminue pas même pour des ordres importants.

III.1.2 Synthèse basée sur les paramètres estimés par Levinson 2D.

En utilisant l'algorithme de Levinson 2D, nous avons synthétisé quelques textures en considérant des fenêtres d'analyse de taille 20x20 et 40x40. Voir figure III.2

Nous remarquons que malgré l'augmentation de l'ordre du modèle, les résultats ne sont pas meilleurs. Ce qui laisse dire que la méthode d'estimation des paramètres de Yules-Walker 2D n'est pas adaptée à la synthèse.



Textures originales

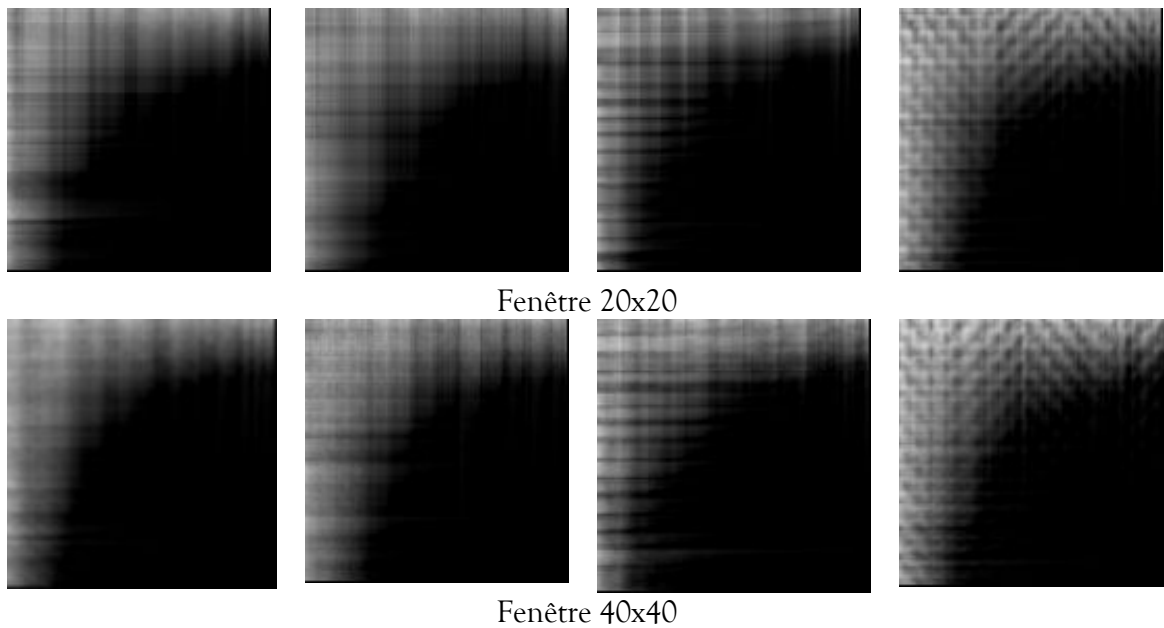
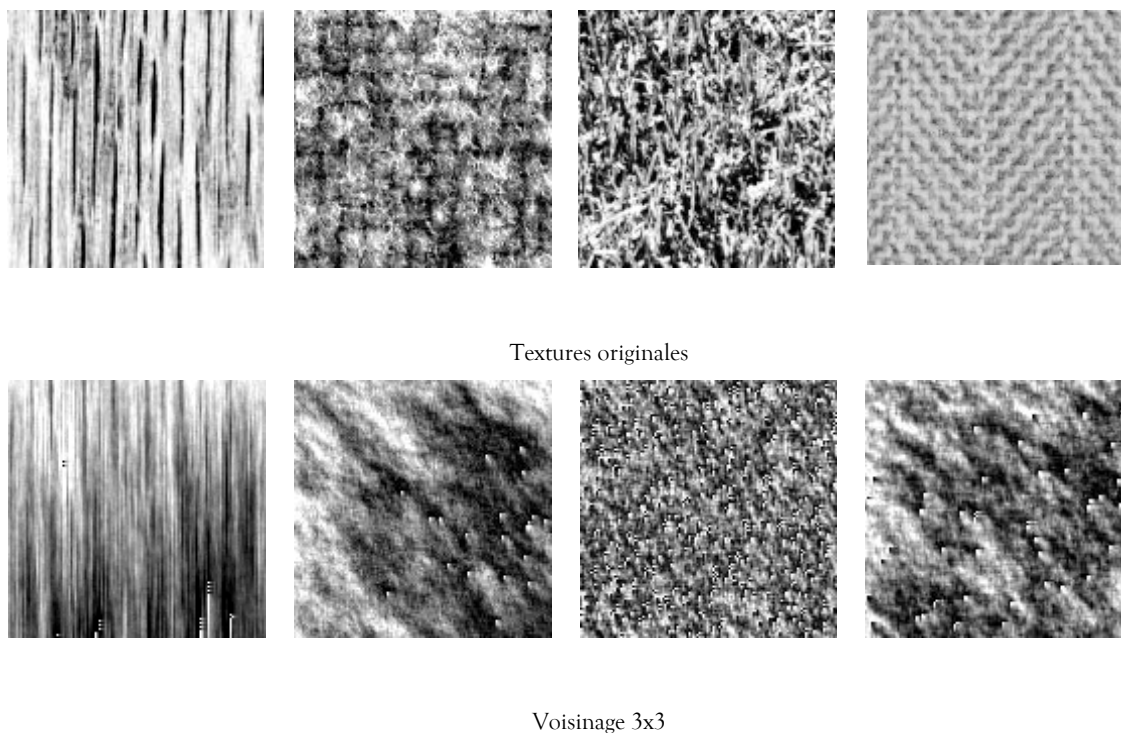


Fig. III.2. Textures synthétisées à partir des paramètres estimés par Levinson 2D.

III.1.3 Synthèse basée sur les paramètres estimés par les moindres carrés.

En utilisant les paramètres AR calculés par la méthode des moindres carrés, nous avons synthétisé les textures ‘Bois’, ‘Coton’, ‘Herbe’ et ‘Chevron’, avec des ordres différents (faibles dans l’ensemble), et une taille d’échantillon 50x50. Les résultats de synthèse obtenus sont représentés à la figure III.3.



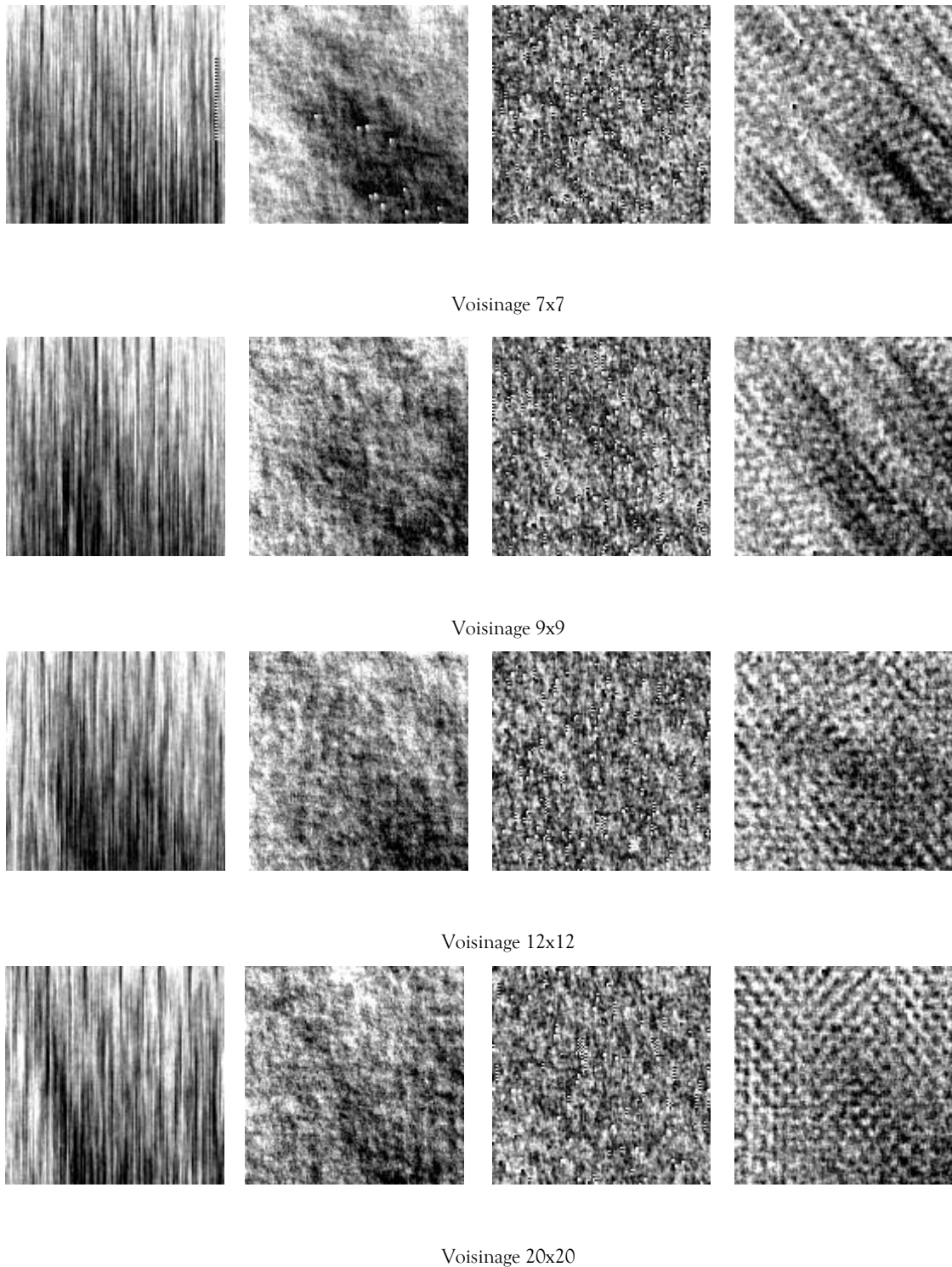


Fig.III.3. Textures synthétisées pour différents ordres par la méthode des moindres carrés, support quart de plan. Taille d'échantillon 50x50

A partir des résultats obtenus pour ces textures, nous pouvons avancer quelques observations :

Les textures synthétisées par la méthode des moindres carrés retiennent les caractéristiques des textures originales, contrairement aux deux méthodes précédemment testées, plus l'ordre est important et mieux les images sont synthétisées.

Il apparaît que la synthèse commence à donner des résultats proches à ceux désirés même avec des ordres faibles. Par contre, si l'ordre est petit comparativement avec la taille de la primitive de la texture, l'image est moins bien synthétisée. La texture « Herbe » en donne un exemple.

Le tracé de la variance propose le choix de l'ordre adapté à chaque texture. Les images de synthèse confirment ce choix. En effet, la texture 'Bois' synthétisée à des ordres élevés ne présente pas une réelle amélioration par rapport à celle obtenue avec un ordre 7x7. Par contre, les textures 'Chevron' et 'Coton' synthétisées ne commencent à ressembler aux textures originales qu'à partir de l'ordre 12x12, ordre au-delà duquel la variance se stabilise.

Pour valider les résultats obtenus, nous avons pensé à recalculer les paramètres à partir des textures synthétisées, et nous les avons comparés à ceux extraits des textures originales. Le tableau III.1 nous donne ces paramètres pour la texture « Chevron » originale et « Chevron » synthétisée, déjà représentée dans la figure III.3 (avec un voisinage 7x7).

Les résultats montrent clairement que les paramètres estimés à partir de la texture originale et synthétisée sont très proches. Ainsi, les paramètres estimés par la méthode des moindres carrés avec support causal Quart de Plan, permettent de générer des textures possédant les mêmes propriétés que les textures originales. Pour le vérifier, nous avons, représenté les transformées de Fourier des textures originales et synthétisées dans la figure III.4

Originale	Synthétisée	Originale (suite)	Synthétisée (suite)	Originale (suite)	Synthétisée (suite)
0.0219	0.0202	0.0022	-0.0067	0.084219	0.0904
0.0741	0.0719	0.0019	0.0009	-0.23827	-0.246
0.0458	0.0456	0.0036	0.0177	0.002207	0.0040
0.0087	0.0044	-0.0055	-0.0190	0.008137	0.0039
-0.0034	0.0064	-0.0093	-0.0030	0.025425	0.0302
0.0065	0.0054	-0.0162	-0.0004	-0.005515	0.0007
-0.0100	-0.0213	0.0154	0.0093	-0.047623	-0.0476
0.0179	0.0065	-0.0199	-0.0219	-0.298647	-0.2921
0.0135	0.0259	0.0070	0.0174	0.766644	0.7667
-0.00008	0.0025	0.0135	0.0046	0.009633	0.0184
-0.0019	-0.0068	-0.0381	-0.0222	0.002548	0.0014
0.0094	-0.0037	0.0539	0.0427	-0.011135	-0.0148
0.0098	0.0208	-0.0112	-0.0231	-0.030083	-0.0365
-0.0061	-0.0007	0.0114	0.0137	-0.024157	-0.0200
-0.0071	-0.0125	-0.0042	-0.0046	0.558034	0.5535
0.0086	0.0105	0.0041	-0.0007	variance=	variance=
		0.0015	-0.0036	174.18	164.82458

Tableau III.1 Paramètres estimés par moindres carrés de chevron originale et synthétisée (ordre 7x7, taille d'échantillon 50x50)

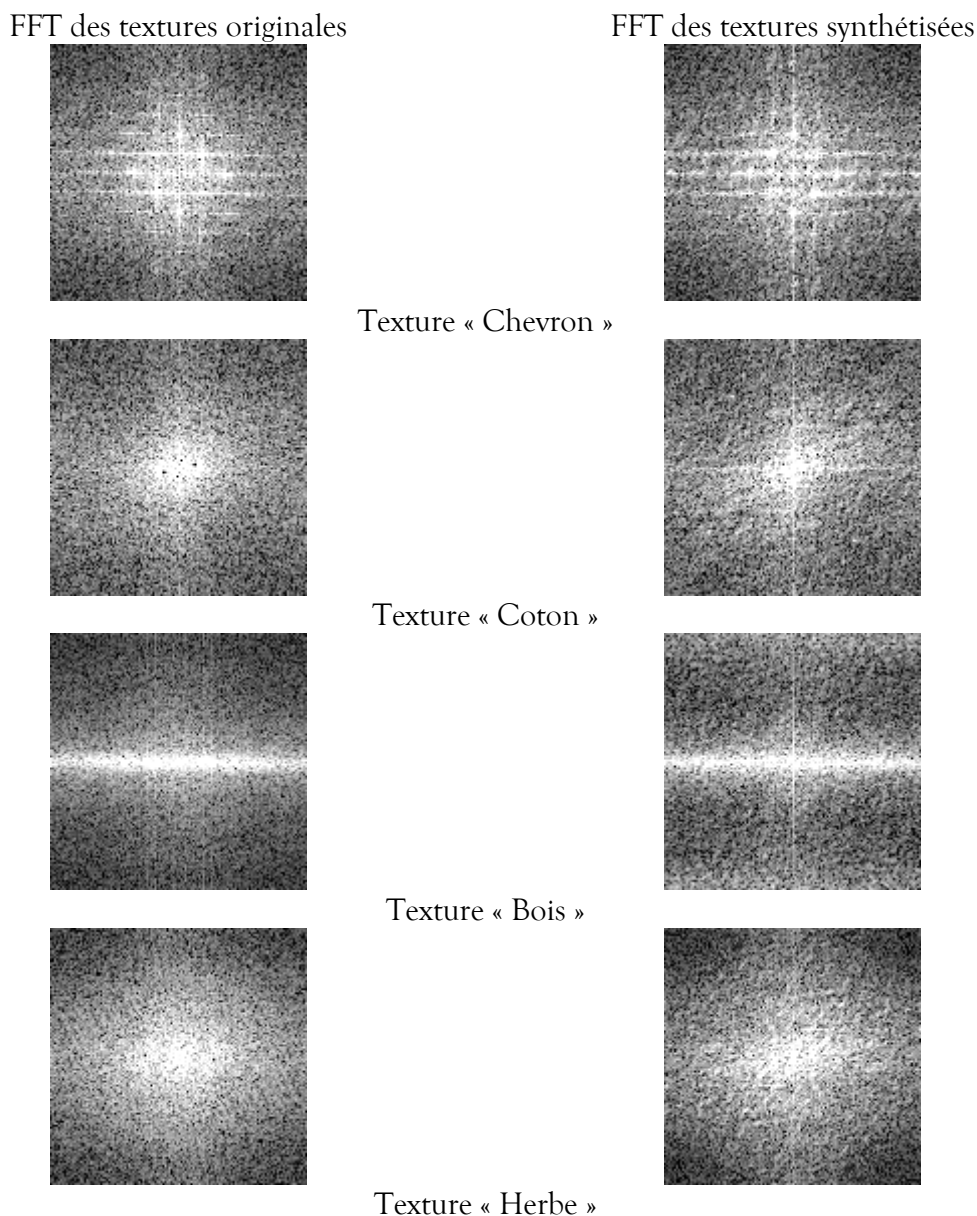


Figure III.4. TF des textures originales et synthétisées par la méthode des moindres carrés (support causal)

Nous remarquons que les TF des textures originales et synthétisées sont proches.

III.1.4 Synthèse de textures obéissant à un modèle AR non causal.

La synthèse de la texture dans le cas d'un support non causal ne peut pas se faire d'une manière récursive comme dans le cas causal. Le modèle AR 2D non causal est équivalent à un modèle Markovien Gaussien (GMRF :Gaussian Markov Random Field), [Che,85]. Pour générer une texture obéissant au modèle GMRF, deux algorithmes existent : l'échantillonneur GMRF itératif (Iterative GMRF sampler), et l'échantillonneur GMRF direct (Direct GMRF sampler).

L'échantillonneur GMRF itératif est similaire à l'échantillonneur de Gibbs. Mais ce dernier doit effectuer des centaines d'itérations avant d'aboutir à une texture stable. L'échantillonneur GMRF direct

(développé par Chellapa [Che,85]) fait appel à la transformée de Fourier 2D pour synthétiser une texture GMRF. C'est ce dernier que nous avons utilisé pour synthétiser des textures obéissantes à un modèle AR non causal.

Reprenons l'équation (II.36) avec l'erreur de prédiction assimilée à un bruit blanc Gaussien de moyenne nulle et de variance σ^2 :

$$y(s) = -\sum_{r=1}^{N'} \theta_r y(s-r) + \omega(s) \quad (\text{III.1})$$

Soit y le vecteur colonne qui contient tous les pixels de l'image. ω le vecteur colonne contenant toutes les valeurs de $\omega(s)$, et θ le vecteur colonne des paramètres AR du modèle :

$$y = [y_0, y_1, \dots, y_{(T-1)^2}]^t \quad (\text{III.2.a})$$

$$\omega = [\omega_0, \omega_1, \dots, \omega_{(T-1)^2}]^t \quad (\text{III.2.b})$$

$$\theta = [\theta_1, \theta_2, \dots, \theta_{N'}] \quad (\text{III.2.c})$$

Puisque le vecteur voisinage est symétrique, nous avons : ($\theta_r = \theta_{-r}$). Si on écrit l'équation précédente pour tous les pixels de l'image, nous obtiendrons l'équation matricielle suivante [Che,85] :

$$B(\theta)y = \omega \quad (\text{III.3})$$

Où B est une matrice bloc circulante de dimension $T^2 \times T^2$.

Prenons l'exemple d'un modèle d'ordre 3×3 , quatre paramètres sont à estimer ($\theta_1, \theta_2, \theta_3, \theta_4$). La matrice B pour ce modèle est donnée par :

$$B = \begin{bmatrix} B_{1,1} & B_{1,2} & \Lambda & B_{1,M} \\ B_{1,M} & B_{1,1} & \Lambda & B_{1,M-1} \\ M & M & & M \\ B_{1,2} & B_{1,3} & \Lambda & B_{1,1} \end{bmatrix} \quad (\text{III.4})$$

Où chaque élément $B_{i,j}$ est une matrice $T \times T$ circulante définie comme suit :

$$\begin{aligned} B_{1,1} &= \text{circulante}(1, -\theta_1, 0, 0, \dots, 0, -\theta_1) \\ B_{1,2} &= \text{circulante}(-\theta_2, -\theta_3, 0, 0, \dots, 0, -\theta_4) \\ B_{1,T} &= \text{circulante}(-\theta_2, -\theta_4, 0, 0, \dots, 0, -\theta_{31}) \\ B_{1,j} &= 0 \quad \text{pour } 2 < j < T \end{aligned} \quad (\text{III.5})$$

La matrice B est symétrique, et doit être définie positive (valeurs propres strictement positives).

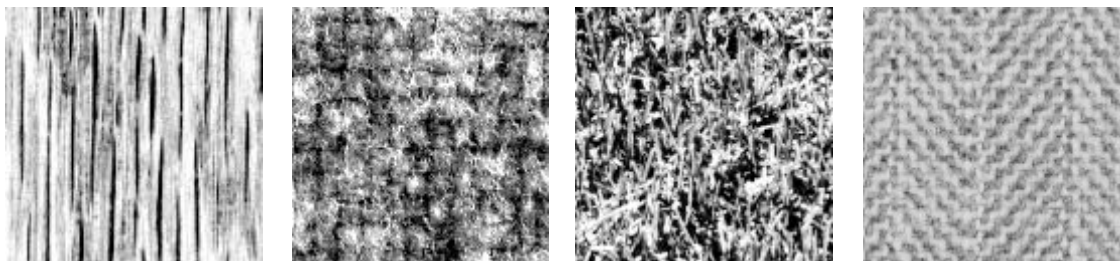
Soit A une matrice T×T, formée des éléments de la première ligne de la matrice B comme suit :
 $A_{i,j}=B(1,j+(i-1)T)$. Pour notre exemple, A est donnée par :

$$A = \begin{bmatrix} 1 & -\theta_1 & 0 & \Lambda & 0 & -\theta_1 \\ -\theta_2 & -\theta_3 & 0 & \Lambda & 0 & -\theta_4 \\ 0 & 0 & 0 & \Lambda & 0 & 0 \\ M & M & M & \Lambda & M & M \\ 0 & 0 & 0 & \Lambda & 0 & 0 \\ -\theta_2 & -\theta_4 & 0 & \Lambda & 0 & -\theta_3 \end{bmatrix} \quad (\text{III.6})$$

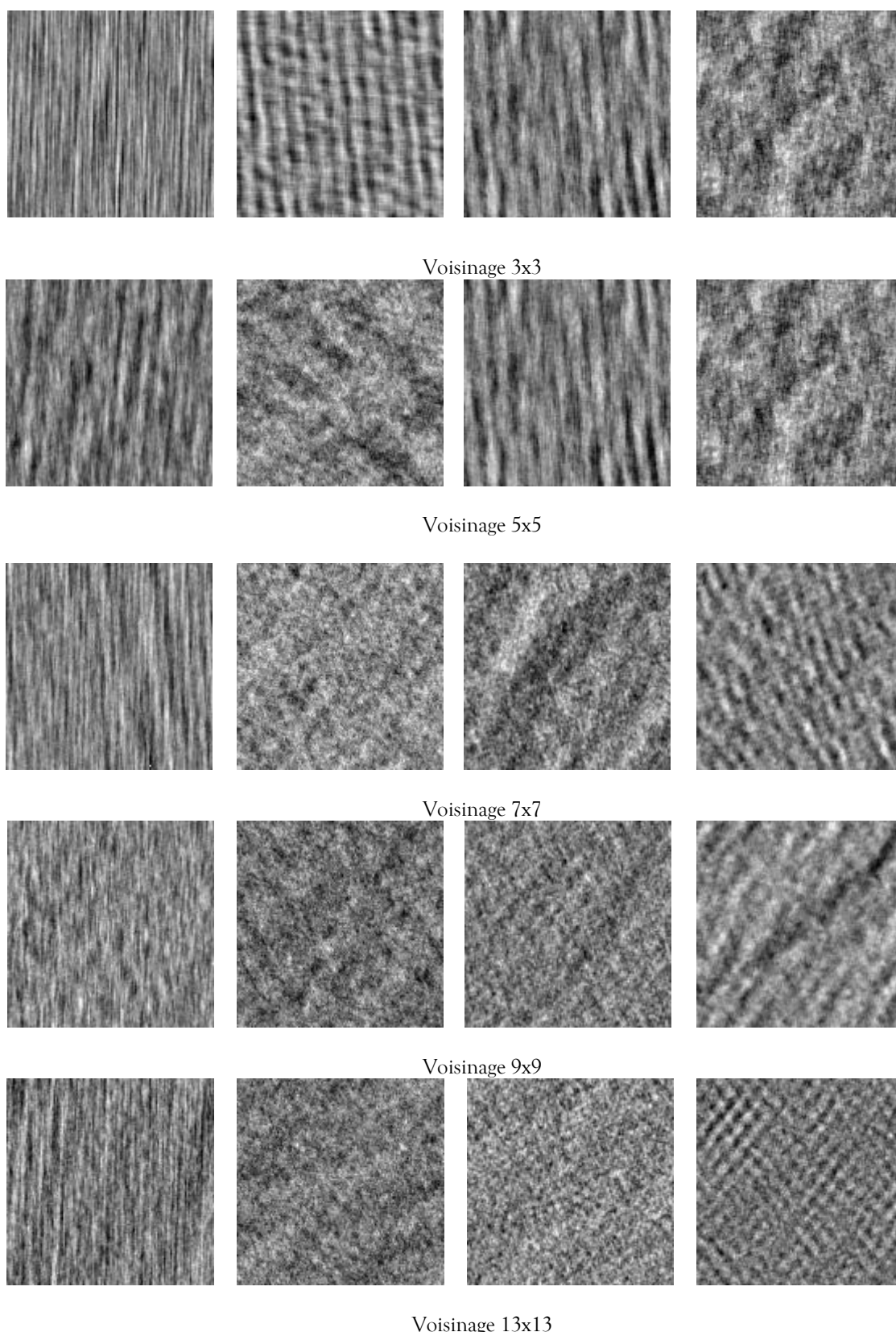
- *procédure de synthèse (Echantillonneur GMRF direct)*

1. générer une matrice η (T×T) composée d'un bruit blanc gaussien $N(0, \sigma^2)$.
2. calculer la FFT 2D de η et sauvegarder le résultat dans η .
3. calculer la FFT inverse 2D de A et sauvegarder le résultat dans A.
4. pour chaque élément (u,v) de la matrice η : $\eta(u,v) \leftarrow \frac{\eta(u,v)}{\sqrt{A(u,v)}}$
5. appliquer la FFT inverse 2D à η et sauvegarder le résultat dans η .
6. $\eta + \mu$ est une réalisation du modèle GMRF.

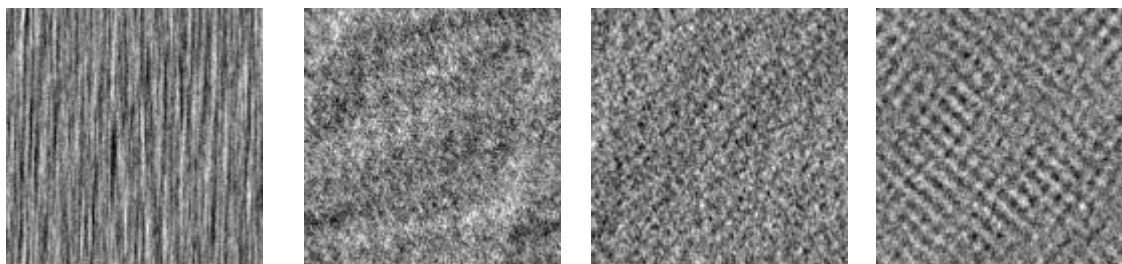
En utilisant l'échantillonneur GMRF direct, nous avons synthétisé différentes textures. La figure III.5 montre quelques résultats de synthèse.



Textures originales



Voisinage 13x13
 Fig. III.5. Textures synthétisées par l'échantillonneur GMRF direct



Voisinage 19x19

Fig. III.5. Textures synthétisées par l'échantillonneur GMRF direct (suite)

Au vu de ces résultats nous pouvons noter, comme dans le cas d'un modèle AR causal, que la qualité des images synthétisées dépend du voisinage. Nous remarquons, par contre, que les textures synthétisées n'ont restitué que l'aspect global des textures originales, mais elles sont loin de leur ressembler, du moment que nous ne pouvons pas visualiser les primitives de base. En les comparant à celles synthétisées par un support causal, elles sont moins bonnes malgré que la variance de l'erreur soit plus faible. Des résultats similaires sont obtenus par d'autres auteurs [Bad,93], [Ran,85], [Che,85].

Nous pouvons justifier ceci par la différence entre les méthodes de synthèse. La première est séquentielle. De ce fait, les premiers pixels sont bien synthétisés, et peu à peu l'erreur se propage, et affecte la synthèse des pixels éloignés. Quant à la deuxième méthode, c'est une approche globale, elle ne permet de capturer que la structure globale et prédominante de la structure.

En outre, la modélisation AR non causal suppose que la texture modélisée possède une distribution Gaussienne. Or les textures utilisées ne le sont pas. Dans l'image III.6, nous avons représenté les histogrammes des textures synthétisées et ceux des textures originales. Ils confirment que les textures synthétisées suivent une loi Gaussienne.

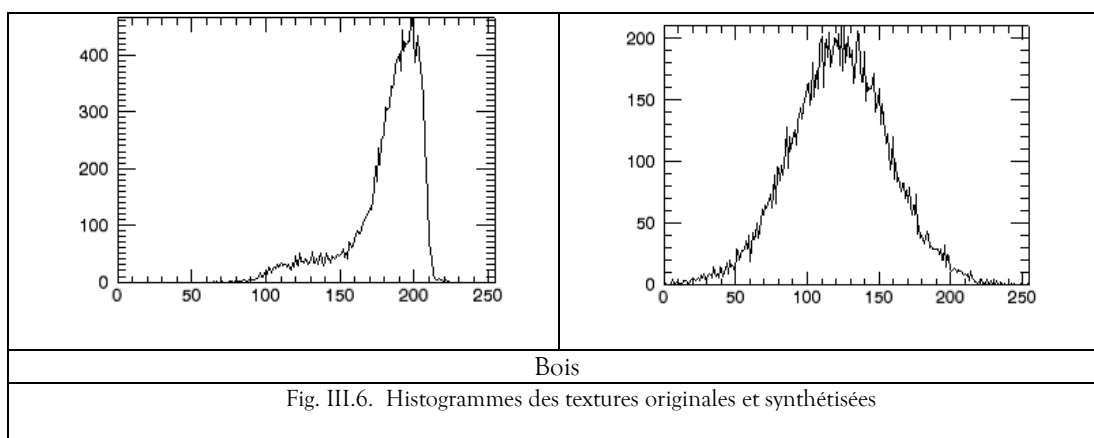


Fig. III.6. Histogrammes des textures originales et synthétisées

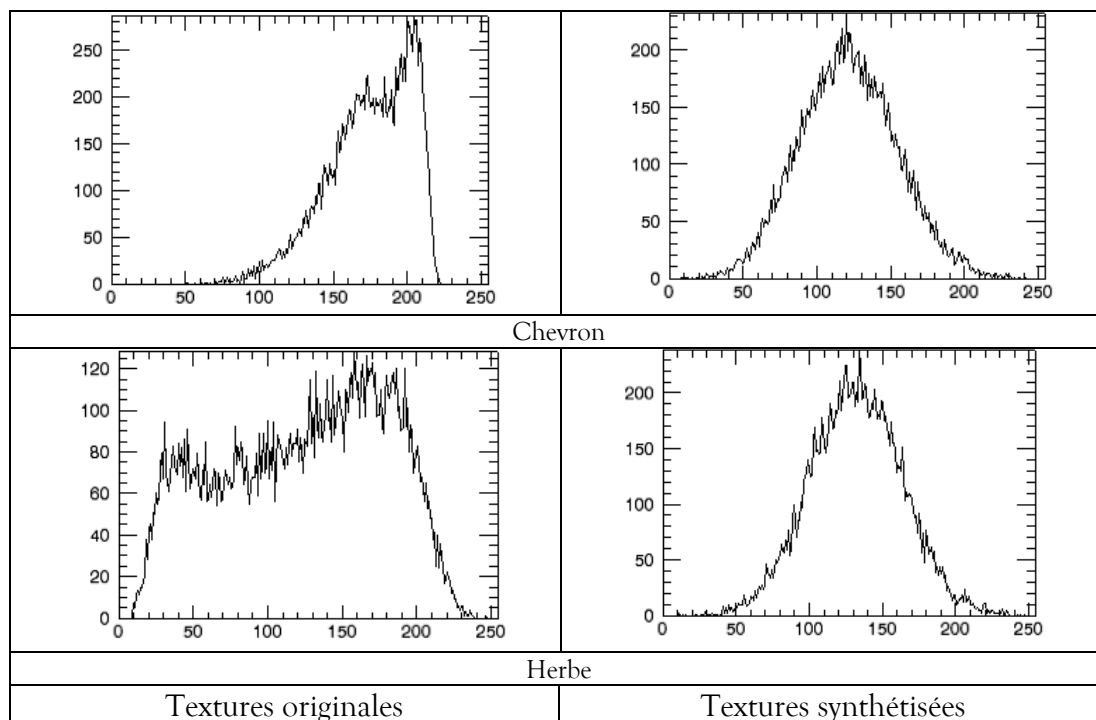


Fig. III.6. Histogrammes des textures originales et synthétisées (suite)

Dans ce paragraphe, nous nous sommes intéressés à l’aspect synthèse. Nous avons exploité la variance de l’erreur d’estimation pour déterminer l’ordre adéquat pour chaque texture, et nous avons comparé les performances de chaque méthode d’estimation par le biais de la synthèse. Il en est ressorti que la méthode d’estimation la plus efficace est la méthode des moindres carrés pour un support quart de plan. Un autre moyen de validation des paramètres consiste à les assimiler à des signatures spectrales et les incorporer dans un processus de classification.

III.2 Classification des images texturées.

III.2.1 Principe

La classification d’image consiste à partitionner l’image en régions homogènes. Une région est un ensemble de pixels connexes ayant des propriétés communes qui les différencient des pixels des régions voisines. Deux approches de classification existent : l’approche « frontière » et l’approche « région ».

L’approche « frontière » regroupe les techniques de détection de contours. Du fait que les contours dans une image sont rarement connexes, il faut procéder à une fermeture des contours pour réaliser une partition complète de l’image.

L'approche « région », fait référence à des groupements de pixels ayant des propriétés communes, chaque pixel étant affecté à une région unique.

Il existe deux types de procédés de classification dans l'approche région : la classification supervisée, et la classification non supervisée. La distinction entre les deux types réside dans le rôle joué par l'opérateur. Dans le cas non supervisé, l'opérateur n'a besoin d'aucune information *à priori* sur l'image à classifier. Soit, il existe une collection universelle de classes déjà définies indépendamment de l'image. Soit, le nombre de classes et leurs caractéristiques sont définis automatiquement durant le processus de classification. Par contre, la classification supervisée demande à l'opérateur de définir des zones de l'image comme étant des échantillons représentatifs des classes à extraire. La classification est donc précédée d'une phase d'apprentissage. Cette phase regroupe l'estimation des paramètres à partir de ces échantillons.

La phase d'apprentissage est suivie de la phase de classification, où chaque pixel est attribué à une classe donnée selon une règle de décision. La règle de décision définit la méthode de classification. Plusieurs méthodes de classification existent [Foo.,92], [Swa.78], [Bel, 98] :

- Classification par maximum de vraisemblance.
- Classification par le minimum de distance (méthode métrique).
- Classification par la distance de Mahalanobis. (Statistique).

III.2.2 Schéma de classification

Le schéma de classification que nous proposons se déroule en deux étapes. Dans un premier temps, il consiste à construire, pour toutes les classes de textures utilisées, les vecteurs caractéristiques pour un ordre donné. L'ensemble de ces vecteurs constitue la base d'apprentissage. Dans un second temps, l'image est balayée pixel par pixel, le vecteur caractéristique est estimé, et est comparé avec la base d'apprentissage. Le pixel est attribué à une classe donnée selon le minimum de distance euclidienne.

Pour construire la base d'apprentissage, nous avons choisi neuf textures de l'album de Brodatz, chacune de taille 128x128. Chaque texture est divisée en 16 sous images. Chaque classe est alors composée de 16 données d'apprentissage (32x32).

Pour chaque sous image, les paramètres AR sont estimés via les différentes méthodes d'estimation que nous avons développé, à savoir :

- Estimation par les équations de Yules-Walker 1D (support causal).
- Estimation par les équations de Yules-Walker 2D (support causal).

- Estimation par les moindres carrés (causal et non causal).

Le vecteur caractéristique se compose des paramètres AR estimés, ainsi que de la variance de l'erreur d'estimation normalisée par la variance de la texture à modéliser [chel,85]. Soit :

$$\text{Vecteur caractéristique} = \left\{ \theta_i, \frac{\sigma_e^2}{\sigma_y^2} \right\}. \quad (\text{III.7})$$

Pour étudier l'influence de la méthode d'estimation, l'ordre du modèle, le type de support de prédiction, ainsi que la taille de l'échantillon sur la qualité de la classification, nous avons mené un certain nombre de tests. Les images utilisées dans les tests sont représentées à la figure III.7.

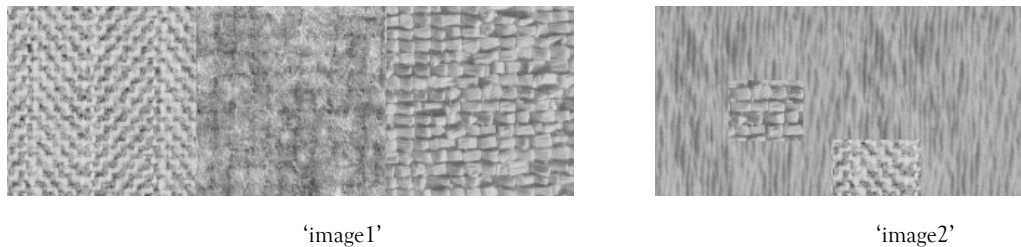


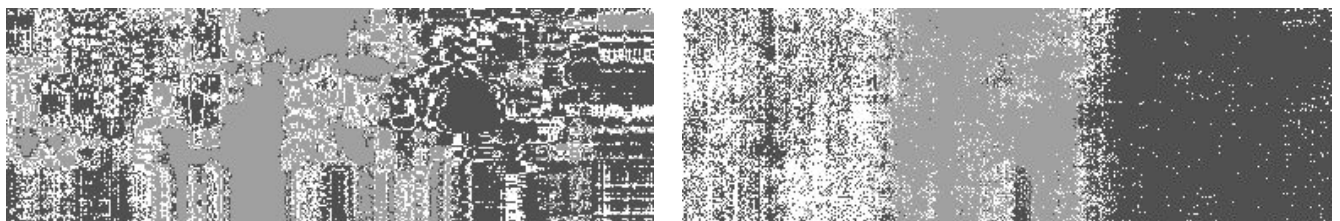
Fig. III.7. Images à segmenter

En employant à chaque fois une méthode d'estimation différente, nous avons classifié les images texturées, en considérant différentes tailles de voisinages, et différentes tailles d'échantillons. Nous avons calculé dans chaque cas le taux de bonne classification (Overall Accuracy), donné par le rapport de la somme des pixels bien classifiés sur la totalité des pixels de l'image.

III.2.3 Résultats de classification

III.2.3.1 Equations de Yules-Walker 1D (Levinson 1D).

L'image1 a été classifiée avec un ordre 3x3, puis 15x15, et une taille d'échantillon 32x32. La figure III.8 montre le résultat de segmentation, avec les taux de bonne classification.



AC=54,91 %
(a) Voisinage 3x3, taille d'échantillon 32x32

AC=77,38 %
(b) Voisinage 15x15, taille d'échantillon 32x32

Fig.III.8. Image1 classifiée par levinson1D. ordre3x3, et 15x15 (taille d'échantillon 32x32)

III.2.3.2 Equations de Yules-Walker 2D (Levinson 2D).

La même image a été classifiée en employant les paramètres estimés par les équations de Yules-Walker 2D, avec les mêmes ordres, et taille d'échantillon. Les résultats sont montrés à la figure III.9.

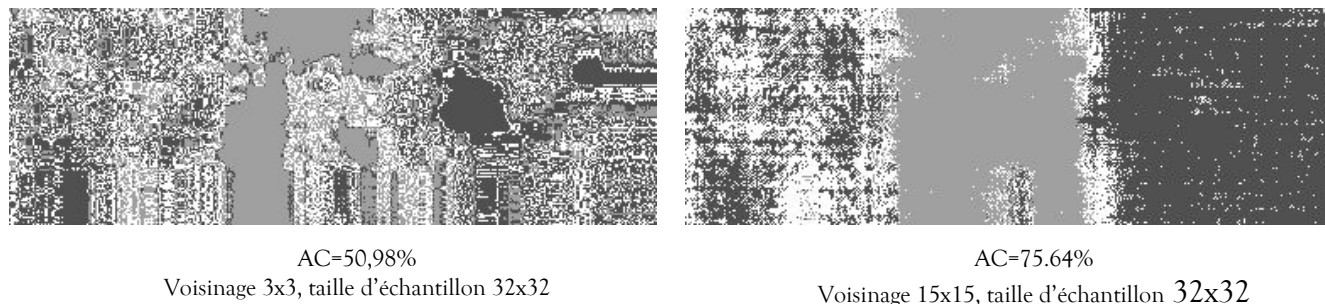


Fig.III.9. Image1 classifiée par levinson2D. ordre3x3, et 15x15 (taille d'échantillon 32x32)

Pour les deux méthodes, Les résultats obtenus montrent que l'ordre du modèle joue un rôle important sur la qualité de la segmentation. Cependant nous pouvons constater que l'augmentation considérable de l'ordre du modèle induisant ainsi un temps calculatoire très supérieur, n'a pas amélioré grandement la précision. Ces constatations ne font que confirmer les résultats de la synthèse.

III.2.3.3 Moindres carrés support causal

Les résultats de synthèse employant les paramètres estimés par les moindres carrés support causal, nous ont encouragé à utiliser des ordres faibles pour la segmentation, nous avons, ainsi, employé un modèle d'ordre 3x3, puis 7x7.

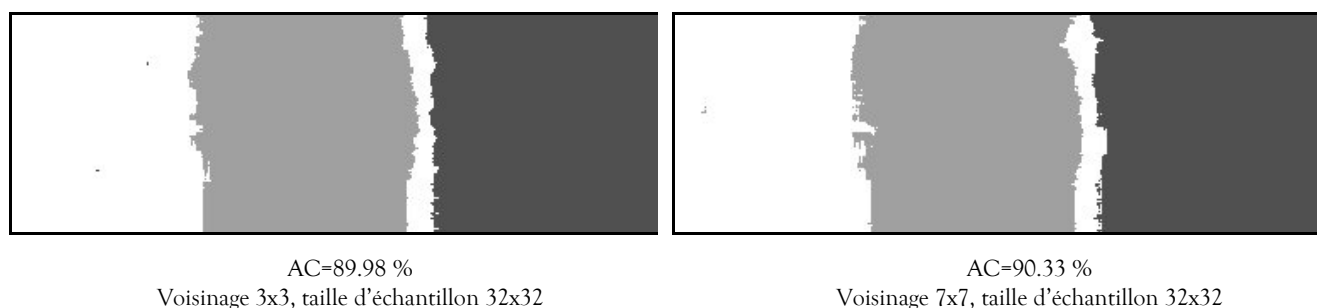


Fig.III.10. Image1 classifiée par la méthode des moindres carrés (support QP). Ordre3x3, et 7x7 (taille d'échantillon 32x32)

Nous constatons que par rapport aux deux autres approches précédemment évoquées, il y a un meilleur taux de reconnaissance. Ainsi, un modèle d'ordre 3x3 permet d'identifier les classes de textures approximativement avec la même précision qu'un modèle d'ordre 7x7. D'autre part, nous avons essayé

de diminuer la taille d'échantillon de 32x32 à 16x16, afin d'accélérer le processus en gardant la même qualité de segmentation. Fig.III.11

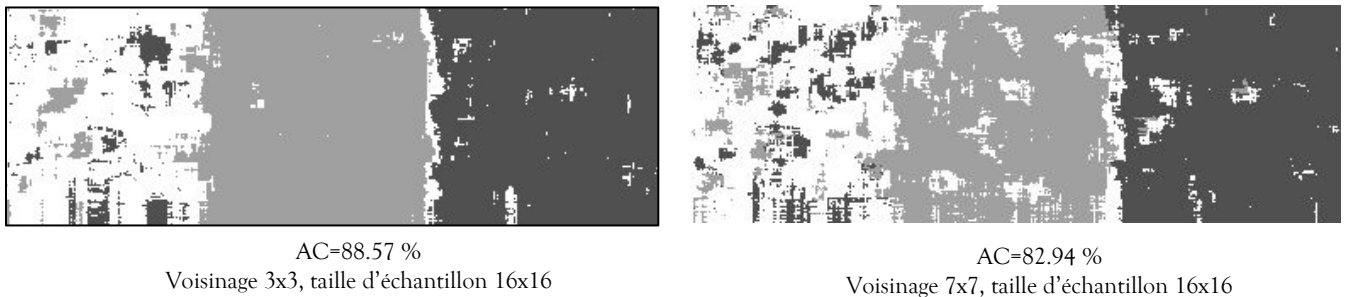


Fig.III.11. Image1 classifiée par la méthode des moindres carrés (support QP). Ordre 3x3, et 7x7 (taille d'échantillon 16x16)

Nous avons appliqué cette méthode pour segmenter l'image2. Les résultats représentés dans la figure III.12, indiquent que le meilleur taux de reconnaissance est obtenu avec un ordre 3x3 et une taille d'échantillon 16x16. Ceci laisse à penser que le modèle adéquat dépend de l'image à analyser.

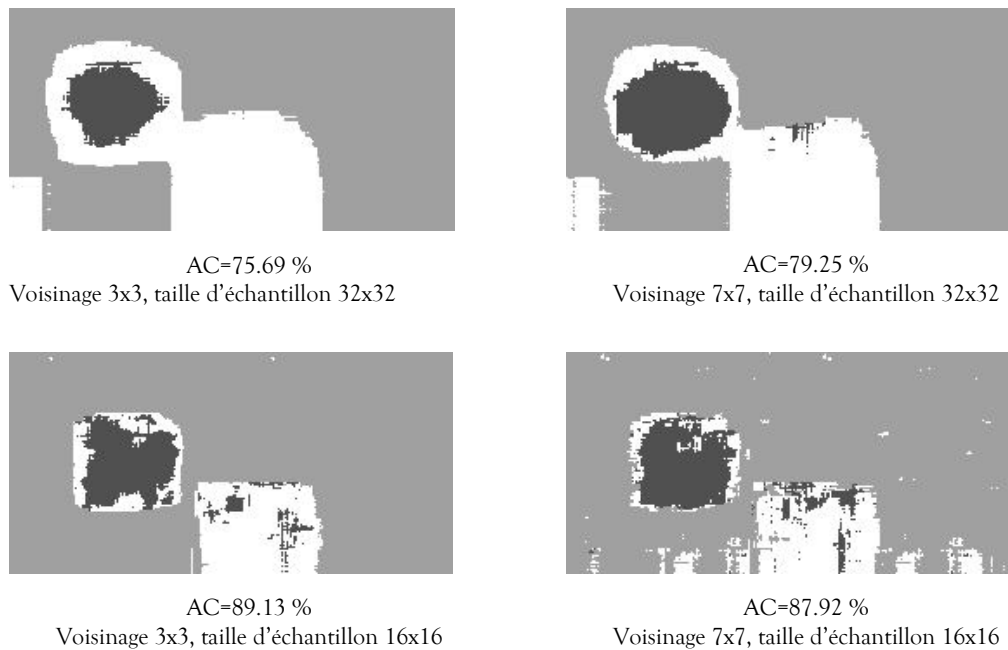


Fig.III.12. Image2 classifiée par la méthode des moindres carrés (support causal). Ordre 3x3, et 7x7 (taille d'échantillon 16x16, et 32x32)

III.2.3.4 Moindres carrés support non causal

En utilisant un support non causal, nous avons classifié les deux images avec des ordres 3x3 et 7x7, et taille d'échantillon 32x32 puis 16x16.

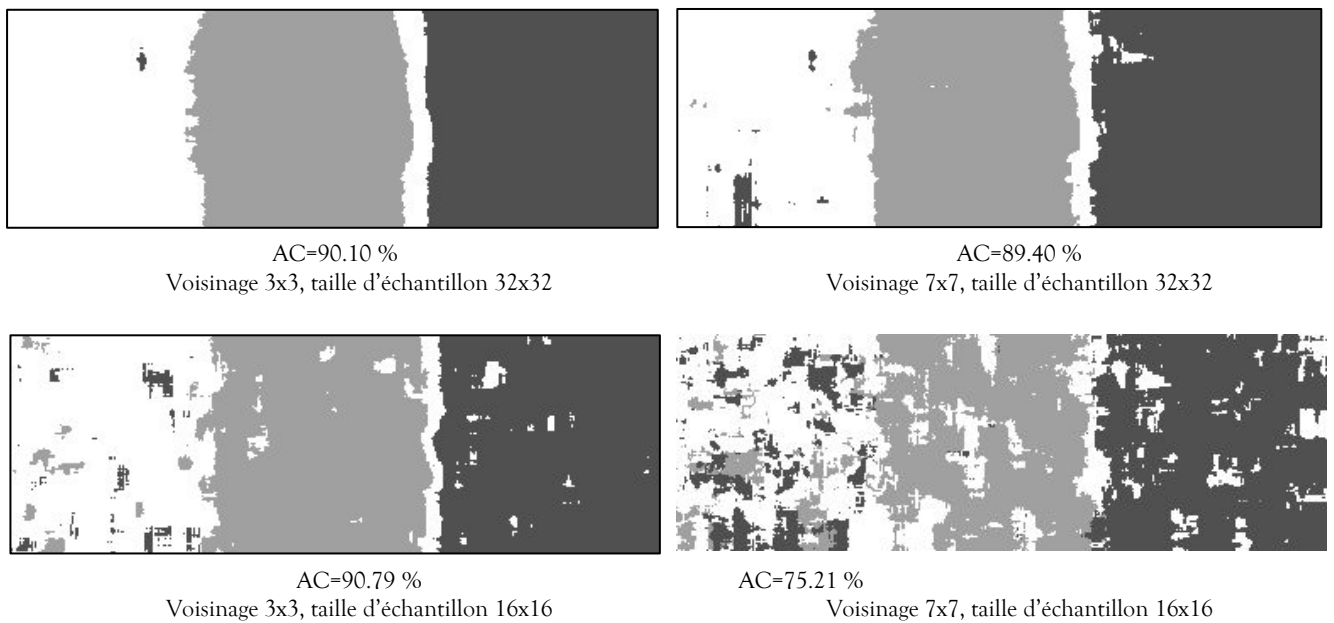


Fig.III.13. Classification de 'image1' par la méthode des MC, support non causal

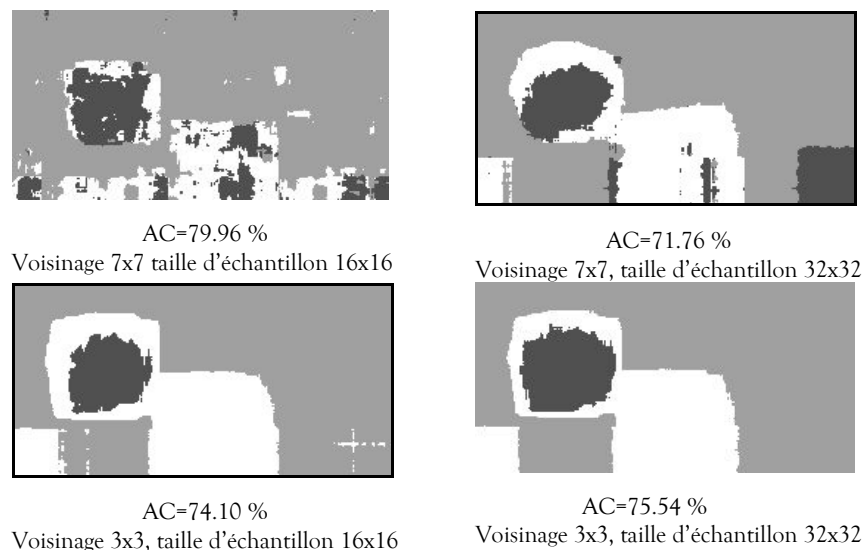


Fig.III.14. Classification de 'image2' par la méthode des MC, support non causal

Si nous comparons ces résultats avec ceux obtenus par les moindres carrés support causal, nous pouvons conclure que la segmentation par le support non causal présente une moins bonne précision. Nous avons, alors, tenté de classifier une image optique composée de sept bandes spectrales, en utilisant la méthode des moindres carrés, et un support causal. L'image utilisée est une image ETM7, acquise en 2001, avec une résolution de 30mx30m et représentant la zone d'Alger, dont nous avons extrait la composante principale d'une imagerie de taille 400x400 pixels, Figure III.15.

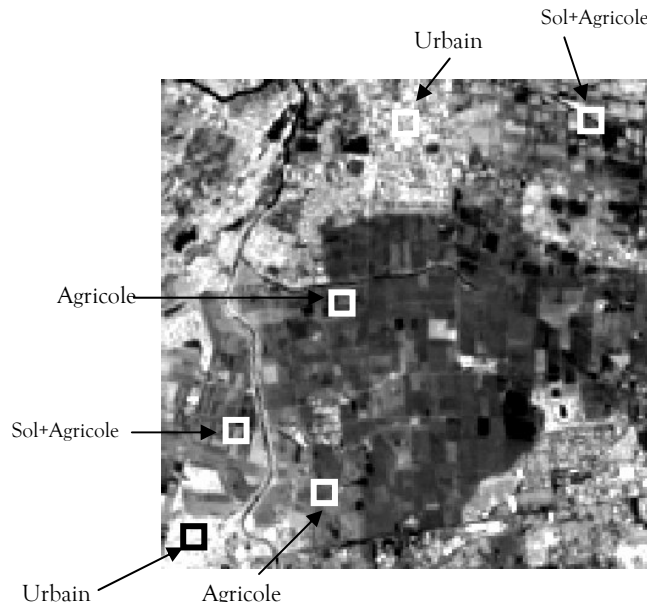


Fig.III.15. Image optique

Nous avons, en premier lieu, sélectionné trois classes et pour chacune, nous avons extrait deux échantillons. La première représente de l'urbain dense, la deuxième des terres agricoles, et la troisième un mélange de sol nu et de terres agricoles. Pour chaque classe, nous avons estimé les vecteurs caractéristiques représentés par les paramètres AR. Nous avons, ensuite, estimé les paramètres pour chaque pixel de l'image optique en considérant des ordres différents, et nous les avons comparés aux vecteurs caractéristiques. Le résultat obtenu avec un modèle d'ordre 5x5, et une taille d'échantillon 32x32 est représenté dans la figure III.16.

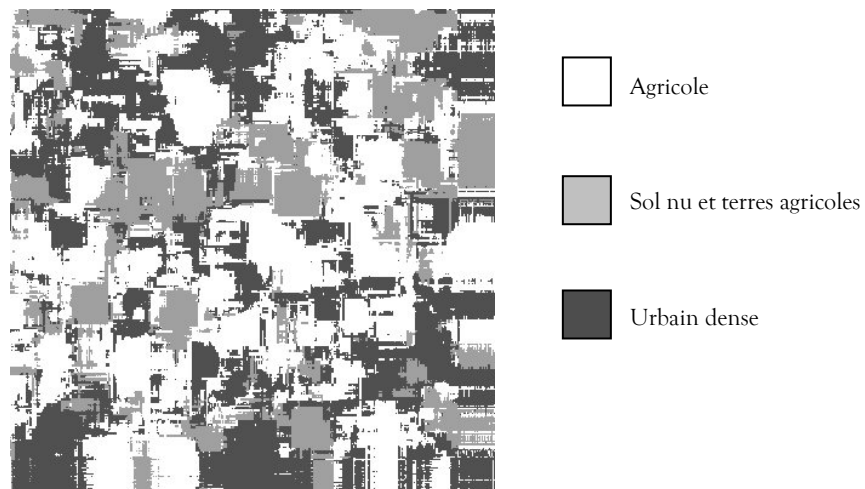


Fig.III.16. Classification de l'image optique. Ordre 5x5, taille d'échantillon 32x32.

Nous constatons une grande confusion entre les trois classes, et un effet de bloc très apparent. Nous avons classifié une image SAR de la même région (Radar ERS1, 1992, résolution 12,5mx12,5m), Figure III.17. D'où nous avons extrait aussi deux échantillons pour chaque classe. Les résultats de classification sont représentés à la figure III.18.

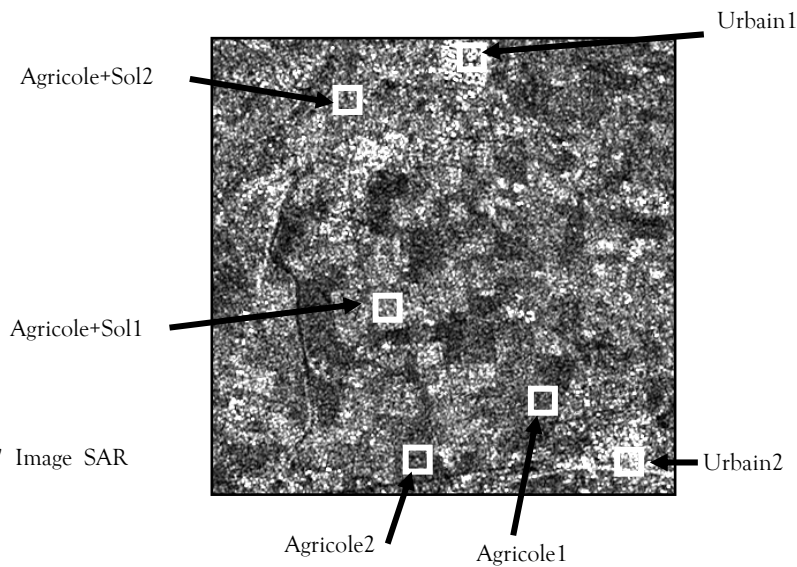


Fig.III.17 Image SAR

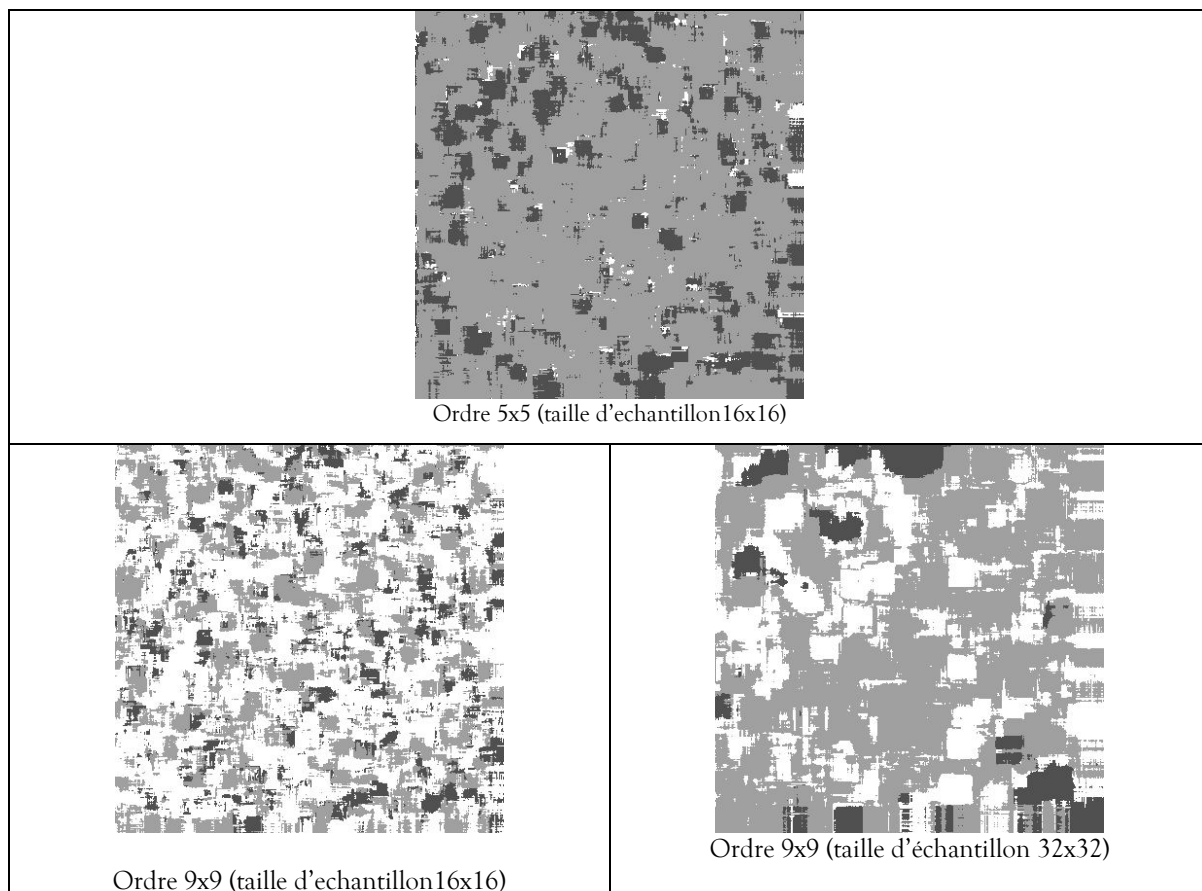


Fig.III.18. Image SAR classifiée avec les ordres 5x5, 9x9 (taille d'échantillon 16x16, 32x32)

Une comparaison entre les figures III.16 et III.18 permet de voir que par rapport à l'image optique, l'image SAR est plus exploitable pour la modélisation AR. En effet, la notion de texture y est plus palpable que dans une image optique.

Nous remarquons que le choix de l'ordre du modèle ainsi que la taille de l'échantillon doit se faire de façon judicieuse [Kou, 2007]. Pour les images classifiées avec une taille d'échantillon 16x16, cette taille étant petite comparativement à l'ordre du modèle, les classes ne sont pas bien identifiées. Nous remarquons une grande confusion entre les classes. L'image classifiée avec une taille d'échantillon 32x32 s'est révélée plus proche de la réalité du terrain. Néanmoins, nous constatons un effet de bloc qui apparaît, induisant ainsi une imprécision de la segmentation surtout aux frontières entre les classes. Pour valider les résultats de classification obtenus, nous avons à partir de l'image classifiée avec ordre 9x9, et taille d'échantillon 32x32, sélectionné un autre échantillon, situé à un autre emplacement et affecté au même thème. Nous avons, ensuite, reclassifié l'image avec ce nouvel échantillon tout en gardant les quatre autres, afin de voir si les pixels étaient identifiés de façon similaire. Les pourcentages d'identification obtenus sont représentés dans le tableau III.3.

classification	Agricole	Sol+agricole	Urbain
Image9x9, échantillon 32x32	40,88%	65,13%	30,80%

Tableau III.2 Pourcentage d'identification des classes pour l'image classifiée avec ordre 9x9, taille d'échantillon 32x32

A partir des résultats obtenus nous pouvons dire que le choix des échantillons est primordial et conditionne les résultats de classification. Les échantillons doivent être représentatifs du thème que l'on cherche à identifier. Ce choix ne peut être fait que sur la base d'une connaissance terrain, ou à partir d'une image déjà classifiée.

Conclusion

Dans ce chapitre, nous avons présenté les résultats de la mise en œuvre des modèles auto régressifs mono et bidimensionnels. Nous avons évalué différentes méthodes d'estimation des paramètres AR par la synthèse des textures, et dans des procédures de classification des images texturées.

D'après l'étude comparative que nous avons menée, il en est ressorti que le modèle AR1D, avec les paramètres estimés par la méthode des moindres carrés avec un support causal, a donné les meilleurs résultats de synthèse, même avec des ordres faibles, du fait que la méthode des moindres carrés est une

approche d'estimation globale basée sur la minimisation de l'erreur moyenne, et qu'un choix judicieux de la taille et de la forme du voisinage est essentiel dans l'obtention et l'interprétation des résultats.

Quand au modèle bidimensionnel, il nous a offert de moins bonnes performances comparées à son homologue monodimensionnel. En effet, les développements théoriques concernant cette classe de modèles ont porté surtout sur les problèmes de stabilité et de réalisation [Kou, 2007]. Ceci est du notamment aux problèmes algébriques de factorisation de polynômes à deux variables [Huan, 72], [Oztu, 92], [Ala, 2003].

Bien que la synthèse de texture par le modèle AR non causal employant la FFT, s'est révélée non fidèle à la texture originale, celle-ci possède l'avantage d'être employée avec n'importe quel ordre afin de produire une large gamme de textures synthétisées de manière homogène, et pouvant, de ce fait, servir pour plaquer des surfaces numériques.

Nous avons, ensuite, employé les paramètres AR, estimés par les différentes méthodes, dans un processus de classification des images mosaïques texturées. Les résultats obtenus confirment la performance de la méthode d'estimation des moindres carrés dans la reconnaissance des classes présentes dans l'image, comparativement aux équations de Yules-Walker. Notons que la qualité de la classification dépend de la nature du support de prédiction employé. Nous avons remarqué que le support causal présente, en général, une meilleure précision que le support non causal. Il faut noter quand même que, dans ce dernier cas, un nombre moindre de paramètres est requis. Nous avons, ensuite, classifié une image satellitaire en employant la méthode qui a donné les meilleurs résultats (méthode des moindres carrés, support causal). Les résultats de classification obtenus montrent que les modèles AR sont plus performants lorsque les textures mises en jeu sont stationnaires.

En effet, le problème auquel nous avons été confrontés est lié à la nature des textures analysées. Ainsi, le développement des modèles AR nécessite la condition de la stationnarité des textures, or les textures naturelles ne le sont pas. Pour éluder ce problème, nous avons apporté une modification dans le calcul de la fonction d'autocorrélation. Par ailleurs, et afin d'améliorer la qualité de la classification, nous avons recalculé les paramètres AR sur différentes fenêtres de la texture. Nous avons, aussi, incorporé le paramètre variance de l'erreur d'estimation dans les vecteurs caractéristiques pour affiner la discrimination des textures.

Conclusion générale

Le problème auquel nous nous sommes intéressés s'inscrit dans le cadre de la modélisation, la synthèse et la segmentation des textures. Notre centre d'intérêt a été la recherche d'une formulation mathématique permettant de caractériser les textures par le biais d'un modèle paramétrique, qu'est le modèle autorégressif.

Un modèle de texture idéal est celui qui est capable de capturer les caractéristiques uniques d'une texture particulière. Il devrait, alors, être capable de régénérer la texture à partir de l'ensemble des paramètres du modèle élaboré.

Au cours de ce travail, nous avons étudié et mis en œuvre les modèles AR1D, et AR2D, pour modéliser, synthétiser les textures naturelles, ainsi que pour classifier les images texturées.

Dans un premier temps, nous avons abordé le problème de l'estimation des paramètres du modèle AR. Nous avons, alors, mis en œuvre différents algorithmes d'estimation, en exploitant des algorithmes récursifs permettant de réduire le temps de calcul.

La question que nous nous sommes posés, ensuite, est relative à la validation d'un modèle estimé pour une texture particulière. Nous avons, alors, considéré le problème inverse qui consiste à synthétiser les textures à partir de leurs modèles estimés, et de comparer ensuite les textures synthétisées aux textures originales.

En employant les paramètres estimés par la méthode des moindres carrés et un support causal, nous avons pu générer des textures synthétisées retenant les caractéristiques des textures originales, et ceci avec des ordre faibles. Ce sont les meilleurs résultats de synthèse que nous avons pu obtenir. Ce qui nous a poussés à retenir cette méthode pour la classification des images satellitaires.

Nous avons, ensuite, appliqué les paramètres AR dans un processus de classification des images mosaïques texturées, et puis sur des images satellitaires. Par ailleurs, et afin d'améliorer la qualité de la classification, nous avons recalculé les paramètres AR sur différentes fenêtres de la texture. Nous avons, aussi, incorporé le paramètre variance de l'erreur d'estimation dans les vecteurs caractéristiques.

Le problème auquel nous avons été confrontés est lié à la nature des textures analysées. En effet, le développement théorique des modèles AR requiert l'hypothèse de la stationnarité des textures, et on suppose que la texture obéit à un modèle de distribution connue dont on cherche à estimer les paramètres. Or les textures naturelles ne le sont pas. Nous avons de ce fait, apporté une modification dans le calcul de la fonction d'autocorrélation, qui nous a permis d'améliorer les résultats.

A partir des différentes simulations réalisées, nous pouvons dire que le développement et l'application des modèles autorégressifs pour classier les images satellitaires se sont révélés difficile à mettre en œuvre. En effet, pour pouvoir estimer les paramètres du modèle relatifs à un pixel, nous devons considérer une zone représentative de la texture à laquelle le pixel appartient. Ceci n'est fructueux que si l'image contient des grandes zones texturées, entraînant d'un autre côté une perte de précision aux frontières entre les classes de textures.

A partir de ces constats, nous pouvons dire que quelque soit la texture analysée, les paramètres autorégressifs constituent des statistiques de second ordre. Les limitations du modèle AR sont celles de tout modèle dont l'ensemble des paramètres est une spécification au second ordre : difficulté de pouvoir synthétiser n'importe quelle texture ou séparer un ensemble de textures qui auraient des statistiques identiques à l'ordre deux et différentes à des ordres supérieurs.

Notons, en dernier, que de nombreux développements et extensions peuvent être envisagés, notamment le modèle autorégressif adaptatif pour palier au problème de l'erreur qui se propage, ainsi que le modèle autorégressif multi échelle pour pouvoir capturer les caractéristiques locales et globales des textures.

Bibliographie

- [Ala,98] Olivier Alata : (1998): Caractérisation de textures par coefficients de réflexion 2-D Application en classification et segmentation. Thèse de Doctorat, Université Bordeaux1. France.
- [Ala,96] O.Alata, P. Baylou, M. Najim (1996): Multiple resolution image segmentation using four QP supports of 2D autoregressive model. International conference on image processing, Vol 1, 16-19 Sept. pp 277-280.
- [Ala, 2003] Alata, O., Najim, M., Ramananjara, C. & Turcu, F (2003): Extension of Shur-Cohn stability test for 2D AR quarter plane model. IEEE transactions on information theory, 49(11), pp 3099-3106.
- [Alan,2002] Alan S, Willsky (2002): Multiresolution Markov models for signal and image processing. Proceeding of IEEE, Vol. 90, N° 8, pp 1398-1458.
- [Bad,93] David A.Bader, Joseph Jaja, Rama Chellapa (1993) :Scalable data parallel algorithms for texture synthesis and compression using Gibbs random fields. IEEE transactions on image processing, Vol 4, N° 10, pp 1456-1460.
- [Bel,98] Belhadj-Aissa.A (1998) : Contribution au système d'analyse et de traitement d'images satellitaires (L.A.T.I.S), analyse spatiale d'images satellitaires appliquées à la cartographie thématique. Thèse doctorat d'état, USTHB, Alger.
- [Ben, 98] Jesse Bennett and Alireza Khotanzad (1998): Multispectral random field models for synthesis and analysis of color images. IEEE transactions on pattern analysis and machine intelligence, Vol. 20, N°. 3, pp 327-332.
- [Bou,90] Charles Bouman and Bede Liu (1990): Unsupervised Estimation of Image Textures Using an Autoregressive Model. International conference on Acoustics, speech, and signal processing, April 3-6, 1990, Vol 4, pp 2141-2144.
- [Bro,66] Brodatz.P : textures :A photographic album for artists and designers. New York. Dover Publications.
- [Cha,2001] Chaur-Chin-Chen and Chien Chang Chen: Texture synthesis: a review and experiments. Journal of information science and engineering N° 19, pp 371-380.
- [Che,85] R.Chellapa & Rangasami.L.Kashyap (1985): Texture synthesis using 2D noncausal autoregressive models. IEEE Transactions on Acoustics, Speech, and Signal Processing , Vol 33 , pp 194-203.
- [Chel,85] Rama Chellapa. Shankar Chatterjee : Classification of texture using Gaussian Markov random field. IEEE transactions on acoustics, speech, and signal processing, Vol 33, No. 4, pp 959-963.

- [Che,88] Chee Sun Won & Haluk Derin (1988): Maximum Likelihood estimation of Gaussian MRF parameters. International conference on Acoustics, Speech and Signal processing, April 11-14, New York, Vol 88, pp 1040-1043.
- [Cla,97] Claude and A. Smolarz (1997): A new textured image segmentation algorithm by autoregressive modelling and multiscale block classification. Proceeding of Sixth IEEE international conference on image processing and its applications, 14-17 July Dublin, Ireland, pp 586-590.
- [Coc,95] Jean-Pierre COCQUEREZ et Sylvie PHILIPP (1995) : Analyse d'images, filtrage et segmentation. Paris. Masson, pp 4.
- [Dil,95] Dileep K.Panjwani and Glenn Healey (1995): Unsupervised segmentation of textured color images using Markov random field models. IEEE transactions on pattern analysis and machine intelligence, vol 17, N°10, pp 939-954.
- [DU,90] DU L. J (1990): Texture segmentation of SAR images using localised spatial filtering. In Proceedings of International Geoscience and Remote Sensing Symposium, Washington, DC. pp. 1983-1986.
- [Elf,94] Ibrahim M. Elfadel & Rosalind W.Picard (1994): Gibbs Random fields, cooccurrences, and texture modelling. IEEE transactions on pattern analysis and machine intelligence, Vol. 16, pp 24-37.
- [Foo, 92] Foody G .M., N.A. Campbell, N. M. Trodd T. F. Wood,(1992) : Derivation and applications of probabilistic measures of class membership from the maximum- likelihood classification. Photogrammetric Engineering and Remote Sensing, 8 (9), pp. 1335-1341.
- [Fran,89] Joseph M.Francois,A.Zvi.Meiri (1989):A 2D autoregressive finite support causal model for texture analysis and synthesis. Proceeding IEEE of international conference on Acoustics, speech and signal processing, May 23-26 Glasgow, Scotland, pp 1552-1555.
- [Fra,88] Joseph M.Francois,A.Zvi.Meiri (1988): A unified structural stochastic model for texture analysis and synthesis. Proceeding IEEE of international conference on pattern recognition, November 14-17, Rome, Italy , pp 41-45.
- [Gag,83] Gagalowicz, A. (1983). Vers un modèle de texture. Thèse de Doctorat d'état. Université de Paris VI. France.
- [Gall,75] M.M. Galloway (1975) : Texture analysis using gray level run lengths. Computer graphics and image processing, Vol 4, pp 172-179.
- [Guy,2007] Guy Almouzni (2007) : Traitement du signal, opérations, algorithmes. Polycopie de cours. Eisti. 2007. www.eisti.fr
- [Har,73] R.M Haralick, K.Shanmugan, I.Dinstein (1973): Textural features for image classification, IEEE Transactions on systems, Man, and cybernetics, Vol 8, pp.

601-621.

- [Har,79] R.M. Haralick, (1979): statistical and structural approaches to texture. Proceeding of the IEEE, vol 67, pp.786-804.
- [Hid,96] Hideki Noda & Mehdi N.Shirazi, Eiji Kawaguchi (1996): An MRF model based method for unsupervised textured image segmentation. Proceeding of IEEE 13th international conference on pattern recognition, pp 765-769.
- [Hua, 72] Huang, T.S (1972): stability of two dimensional recursive filters. IEEE Transactions on Audio and Electro acoustics. IEEE transactions on Audio and electroacoustics, Vol 20, N° 2, pp 63-158.
- [Jen,94] Jennifer Davidson, Ashit Talukda and Noel Cressis (1994): Texture analysis using partially ordered Markov models. Proceeding IEEE on image processing, pp 402-406, Austin, Texas, November 13-16.
- [Jit, 93] Jitendra K. Tugnait (1993): Texture synthesis using asymmetric 2D non causal AR models. IEEE signal processing workshop on higher order statistics, pp 71-75.
- [Joh, 96] John G. Proakis & Dimitris G.Manorass (1996) : Digital signal processing, principles, algorithms and applications. Prentice Hall. P 443.
- [Jul,62] Julesz B. Visual pattern discrimination. IRE Trans. Inform. Theory IT8, pp. 84-92.
- [Jul,83] B. Julesz & R. Bergen (1983): Textons, the fundamental elements in preattentive vision and perception of textures. Bell System Tech., 62(6)
- [Kan,2003] Rami Kanhouche : A modified Burg algorithm equivalent in results to Levinson algorithm.<http://www.cmla.ens-cachan.fr/fileadmin/Documentation/Prepublications/2003/CMLA2003-17.pdf>.
- [Khu,93] Gopinath R.Kuduvalli and Rangaraj M.Rangayyan (1993): An algorithm for direct computation of 2-D linear prediction coefficients. IEEE transactions on acoustics, speech and signal processing, Vol. 41, N°. 2, pp 996-1000.
- [Kie,2001] Kie B Eom (2001): Generalized circular auto regressive models for modelling isotropic and anisotropic textures. International conference on image processing, Vol 2, 7-10 October 2001, pp 129-132.
- [Kou, 2007] Kourgli A. (2007). Analyse variographique, modélisation et synthèse de textures appliqués aux images numériques. These de Doctorat d'état. USTHB. Algerie
- [Law,2000] S.Lawrance Marple Jr (2000): Two dimensional lattice linear prediction parameter estimation method and fast algorithm. IEEE signal processing letters, Vol. 7, No. 6, pp 164-168.

- [Lev,47] N.Levinson (1947):The Wiener RMS (root mean square) error criterion in filter design and prediction. J.Math, Phys.,Vol.25. N°4, pp 164-168.
- [Liu,90] Liu S.S &M.E Jernigan (1990): Texture analysis and discrimination in additive noise. Computer vision graphics and image processing Vol 49, pp 52-67.
- [Mak,75] John Makhoul (1975): Linear prediction: A tutorial review. Proceedings of the IEEE, Vol. 63, No. 4,pp 561-580.
- [Man, 82] Mandelbrot B.B, (1982) : The fractal geometry of nature. San Francisco, CA Freeman.
- [Mar,80] THOMAS L. MARZETTA.(1980)Two-Dimensional Linear Prediction: Autocorrelation Arrays, Minimum-Phase Prediction Error Filters, and Reflection Coefficient Arrays. IEEE transactions on acoustics, speech, and signal processing, Vol. 28, No. 6, pp 725-733.
- [Mat, 98] Andrzej Materka and Michael Strzelecki. : Texture Analysis Methods-A Review. Technical University of Lodz, Institute of Electronics, COST B11 report, Brussels 1998.
- [Max,87]. J.Max. Méthodes et techniques de Traitement du signal et applications aux mesures physiques. Tome II. Masson, pp144
- [Men, 92] Mengyang Liao, Jiamel Qin and Yanni Tan. (1992): Texture classification and segmentation using simultaneous autoregressive random model. Fifth annual IEEE Symposium on Computer-Based medical systems, 14-17 Jun 1992 , pp 398- 401.
- [Mit,93] Mitra Basu (1993): image synthesis using multiscale model. Systems. International Conference on Man and Cybernetics 17-20 Oct 1993, pp 517 - 522 .
- [Myu,2001] Myung-Hee Jung (2001): Contextual modelling and generation of texture observed in single and multichannel images. Korean Journal of Remote Sensing, Vol.17, No.4, pp.335-344.
- [Ozt, 92] Ozturk, Y & Abut, H (1992). A stabilisation algorithm for multichannel multidimensional linear prediction of imagery. IEEE transactions on image processing, Vol 1,pp 101-106.
- [Pat, 2001] Patricio Parada, Javier Ruiz-del-Solar, and Mario Koppen (2001) : Interactive Texture Synthesis.proceeding of the 11th international conference on image analysis and processing , ICIAP 2001, Sept. 26-28, Palermo, Italy, pp 434-439.
- [Pie, 97] Pierre Martin Tardif (1997) : Segmentation d'une séquence IRM. Rapport pré-doctoral. Université Laval Sainte-Foy (Québec), Canada
- [Raj,93] Raja Ghazi and Bernard C. Levy (1993): Phase transitions and multiscale

Markov random field : application to texture synthesis. IEEE conference record of the twenty- seventh Asilomar on signals, systems and computers, pp 1-5, Pacific Grove, California, October 28-November 1.

- [Ran,85] Trygve Randen & John Hakon Husvy (1985): Least squares image texture analysis and synthesis.Hogskolen I Stavanger, working paper 197, pp 1-10.
- [REE.,93] R.& J. M. Hans Du Buf (1993) : A review of recent texture segmentation and feature extraction techniques. C.V.G.I.P: Image Understanding .57 (3), pp. 359-372.
- [Rup,99] Rupert D. Paget (1999) : Nonparametric Markov Random Field Models for Natural Texture Images. These de Doctorat, université de Queensland,
- [Seb, 2001] Sébastien Mavromatis (2001): Analyse de texture et visualisation scientifique: Thèse de Doctorat. Université de la Méditerranée. France.
- [Ser,82] SERRA J (1982): Image analysis and mathematical morphology. Academic Press.
- [Sta,94] P.T.Stathaki & A.G.Constantinides (1994) : Autoregressive modeling through joint and weighted second and third order statistics. International Symposium on Speech, Image Processing and Neural Networks, April 1994, Hong Kong, pp 296-299.
- [Swa, 78] Swain P. H. & S. M. Davis ,(1978) : Remote sensing: the quantitative approach, Mc Graw Hill Inc, New York.
- [Tak,97] Takayuki Nakachi, Katsumi Yamashita and Nozomu Hamada (1997): Asymmetric half plane lattice modeling based on 2D Levinson algorithm. IEEE transactions on circuits and systems—II: analog and digital signal processing, Vol. 44, No. 10. pp 865-868.
- [The,81] Charles W.Therrien (1981) : Relations between 2D and Multicanal linear prediction. IEEE transactions on acoustics, speech, and signal processing, vol. Assp-29, No. 3. pp 454-456.
- [The,89] C. W. THERRIEN AND H. T. ELSHAER (1989) : A Direct Algorithm for Computing 2-D AR Power Spectrum Estimates. IEEE transactions on acoustics, speech, and signal processing, Vol. 37, No. 11 . pp 1795-1797.
- [Tiw, 2003] A.K. Tiwari and R.V.Raja (2003): A lattice Algorithm for lossless image compression using half plane. Conference on convergent technologies for Asia Pacific region. Vol 3, pp 899-903.

- [Wei,2001] Li-Yi Wei (2001) : Texture synthesis by fixed neighbourhood searching. These de Doctorat, department of electrical engineering . Standford University.
- [Weu,90] Weu-Hsien Fang,Andrew E-Yagle (1990): Discrete fast algorithms for two dimensionnal linear prediction on a polar raster. IEEE transactions on signal processing, Vol. 40, No. 6. pp 2017-2020.
- [WIG,65] R. A. Wiggins and E. A. Robinson (1965) : Recursive solution to the multichannel filtering problem; J. Geophysical Research, vol. 70, no. 8, pp. 1885-1891.
- [Xia,96] Xiang Liu and Mohammed Najim (1996): A Two Dimensional Fast Lattice Recursive Least Squares Algorithm. IEEE transactions on signal processing, Vol. 44, No. 10, pp 2557-2567.