

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université Mouloud MAMMERRI, Tizi-Ouzou**



**Faculté de Génie Electrique et d'Informatique**  
**Département d'Automatique**

## *Mémoire de Fin d'Etudes*

En vue de l'obtention du diplôme

*D'Ingénieur d'Etat en Automatique*

## *Thème*

Application du modèle  
autorégressif à la segmentation  
d'images texturées

*Proposé par :*

Mr HAMMOUCHE. K

*Présenté par :*

SFIHI Hayet  
KEFANE Hassina  
SBILI Lila

Soutenu le : 14 /07/2010

*Promotion 2010*

# SOMMAIRE

<b>Introduction Générale</b> .....	01
------------------------------------	----

## ***Chapitre I : Segmentation d'images texturées***

I.1 Introduction .....	03
I.2 Définition de la texture .....	03
I.2.1 Approche perceptuelle .....	03
I.2.2 Approche formelle .....	05
I.3 Différents types de textures .....	05
I.3.1 Textures structurelles .....	05
I.3.2 Textures aléatoires .....	06
I.3.3 Textures directionnelles .....	07
I.4 Définition du motif d'une texture.....	07
I.5 Finesse d'une texture .....	08
I.6 Méthodes d'analyse de la texture .....	08
I.6.1 Méthodes structurelles .....	09
I.6.2 Méthodes spatio-fréquentielles.....	11
I.6.3 Méthodes statistiques .....	12
I.6.4 Méthodes basées sur un modèle .....	15
I.7 Segmentation d'images.....	16
I.7.1 Définition .....	17
I.7.2 Approches de segmentation d'image.....	18
I.7.2.1 approche région .....	19
1) Segmentation par division de régions (split) quad-tree .....	19
2) Segmentation par fusion .....	20
3) Segmentation par division-fusion (split and merge) .....	20

I.7.2.2 Segmentation par contours .....	21
1) Etape de détection des points des contours .....	21
2) Etape d'obtention des contours.....	21
I.7.2.3 Segmentation par classification .....	22
1) Classification supervisée .....	22
2) Classification non supervisée .....	22
I.8 Conclusion .....	25

## ***Chapitre II : Modélisation et méthodes d'estimations du modèle AR***

II.1 Introduction.....	26
II.2 Modélisation .....	26
II.2.1 Modélisation non paramétrique.....	26
II.2.2 Modélisation paramétrique .....	26
II.2.2.1 L'intérêt d'une modélisation paramétrique .....	27
II.2.2.2 Etapes de la modélisation paramétrique.....	27
II.3 Modèles rationnels linéaires .....	29
II.3.1 Modèle ARMA.....	29
II.3.2 Modèle à moyenne glissante (MA) .....	30
II.3.3 Modèle AR.....	31
II.4 Utilisation des modèles AR .....	32
II.5 Estimation des coefficients AR.....	33
II.5.1 Equations de Yule Walker .....	34
II.5.1.1 Algorithme de Levinson-Durbin.....	37
II.5.1.2 Algorithme de Burg .....	38
II.5.1.3 Méthode de covariance.....	41
II.5.1.4 Méthode des moindres carrés .....	42
II.6 Ordre du modèle AR .....	44

II.7 Modélisation AR monodimensionnel d'image .....	45
II.7.1 Ordre du Parcours .....	45
II.7.2 Les différents types de courbes de remplissage d'esp .....	46
II.8 Modèle autorégressif bidimensionnel .....	49
II.8.1 Définition .....	49
II.8.2 Causalité des systèmes bidimensionnels .....	50
II.8.3 Modèles stationnaires .....	52
II.9 Estimation des paramètres .....	53
II.9.1 Ecriture matricielle des équations normales .....	55
II.9.2 Cas d'un modèle causal .....	56
II.10 Conclusion .....	60

### ***Chapitre III : Tests et résultats***

III.1 Introduction .....	61
III.2 Présentation des résultats .....	63
III.3 Interprétation .....	76
III.4 Conclusion .....	82

<b>Conclusion Générale.....</b>	<b>83</b>
---------------------------------	-----------

### **Bibliographie**

## Résumé

Le traitement et l'analyse d'image sont nés de la nécessité de remplacer l'observateur humain par la machine, à présent l'analyse d'image est allée au-delà de cette idée, des solutions ont été proposées pour traiter des problèmes correspondant à des situations auxquelles l'observateur humain n'avait jamais été confronté.

La qualité de l'interprétation d'une image dépend fortement de celle de la segmentation. Malgré la grande diversité de méthodes, les résultats de segmentation restent moyens et varient en fonction de la technique choisie.

Cette thèse s'inscrit dans le cadre d'application d'un modèle autorégressif à la segmentation d'images texturées. Ce modèle considéré comme un filtre tous pôles consiste à balayer l'image ligne par ligne puis regrouper les pixels parcourus dans une séquence monodimensionnelle. Son principe est de prévoir les attributs d'un pixel en fonction des pixels précédemment rencontrés aux cours du balayage.

L'image est une représentation bidimensionnelle, vue la complexité de cette représentation et pour ce rapprocher du cas 2D nous avons fait appel aux différents types de parcours, dans notre application nous nous sommes limités aux deux types Vidéo et Hilbert.

Les résultats sont évalués sur des images synthétiques et des images réelles.

## **Introduction générale**

Parmi les perspectives offertes par l'informatique, le traitement d'images suscite un intérêt de plus en plus croissant à mesure que l'image s'impose comme un support et une source d'information privilégiée. La multiplication des dispositifs d'acquisition conduit à la production d'un grand nombre d'images qu'il devient difficile d'exploiter manuellement. Le traitement d'images s'inscrit alors dans un processus préliminaire destiné à extraire les informations pertinentes contenues dans une image en vue de leur interprétation, leur stockage et leur transmission.

Le traitement et l'analyse d'image sont nés de la nécessité de remplacer l'observateur humain par la machine. A présent, l'analyse d'image est allée au delà de cette idée, des solutions ont été proposées pour traiter des problèmes correspondant à des situations auxquelles l'observateur humain n'avait jamais été confronté, c'est le cas notamment des images non visibles telles que les images radar et les images acoustique. Ainsi le traitement d'image intervient dans plusieurs activités humaines, telles que la médecine, l'agriculture, l'industrie...etc. Tous ces domaines, bien que très différent, font souvent appel aux mêmes techniques de traitement de l'image.

Parmi ces techniques, la segmentation est sans doute l'une des plus étudiées, de part l'importance de son rôle. Elle est définie comme un processus d'extraction des différents objets constituant une image. Cette tâche bien que relevée avec facilité par le système visuel humain, est en réalité complexe et reste un véritable défi pour la communauté du traitement de l'image malgré plusieurs décennies de recherche.

Cependant, la nature et la complexité des images ont poussé les chercheurs à prendre en considération l'information texturale.

La segmentation des images texturées nécessite une analyse de la texture de l'image en question afin d'en extraire ses caractéristiques. Il existe plusieurs méthodes d'analyses de la texture, ces méthodes peuvent être classées en quatre approches essentielles qui sont : L'approche statistique, l'approche structurelle, l'approche spatio-fréquentielle et l'approche par modélisation.

Nous nous sommes intéressés dans notre travail à la modélisation AR, en considérant l'image comme une séquence temporelle de niveaux de gris.

Le cadre général dans lequel s'inscrit notre mémoire est celui de la segmentation d'images texturées basée sur la modélisation AR. Pour cela on a structuré notre travail en trois chapitres.

Dans le premier chapitre, nous introduisons la notion de texture, sa définition et ses diverses formulations. Nous présentons également les différentes approches d'analyse de la texture, puis nous exposerons les différentes méthodes de segmentation d'images texturées.

Dans le deuxième chapitre, nous présenterons la modélisation et l'analyse du modèle autorégressif 1D, 2D ainsi son application à l'extraction des attributs de texture. Nous définissons également les différents types de parcours permettant de passer d'une image bidimensionnelle en une séquence monodimensionnelle.

Le troisième chapitre englobe les tests et résultats obtenus par l'approche adoptée.

Enfin, nous terminerons par une conclusion générale qui énoncera les résultats et les perspectives de notre travail.

# *Chapitre I*

---

## *Segmentation d'images texturées*

## I.1 Introduction

La texture joue un rôle important dans l'analyse, l'interprétation et la synthèse d'une grande variété d'images, dans les différents domaines (médical, géographique, etc. . . .). Plusieurs travaux de recherche ont montré que l'être humain dans sa perception visuelle distingue les objets en s'appuyant sur la notion de texture.

Dans le cas de segmentation d'une image, au sens de texture, on parle de segmentation d'images texturées. Il est nécessaire de classer les pixels de cette image, en se basant sur l'extraction d'attributs permettant de caractériser les différentes textures de l'image.

Dans ce chapitre, nous introduisons les notions de la texture et de la segmentation, ainsi que leurs différentes approches.

## I.2 Définition de la texture [1]

Le traitement d'images naturelles met en évidence la nécessité de prendre en compte pour une interprétation postérieure, des régions non uniformes au sens du niveau de gris. Ces régions sont dites texturées.

La notion de texture est un concept assez flou, et que la définition diffère selon les points de vue. Nous distinguons deux approches:

- *L'approche perceptuelle.*
- *L'approche formelle.*

### I.2.1 Approche perceptuelle

La notion de texture est abordée ici au travers de sa perception. Cette approche consiste à décrire une texture à partir de propriétés perceptuelles telles que la complexité ou la granularité.

Le mot texture (de latin *textura*) a tout d'abord désigné l'arrangement des fils d'un matériau tissé. A partir de VI<sup>ème</sup> siècle, la texture définit plus généralement l'arrangement des éléments d'une matière, elle caractérise sa structure. Une texture représente une variation d'un ou plusieurs paramètres caractéristiques d'un objet, à une petite échelle devant la taille de ce dernier. La prise en compte de la taille est importante : les fils d'un tissu définissent la texture

d'un vêtement à l'échelle humaine, mais en effectuant un zoom sur ce dernier, le fil devient le matériau dont les fibres forment la texture.

Unser [2] définit la texture comme suit :

*«Une texture est une région d'une image pour laquelle il est possible de définir une fenêtre de dimension minimale, telle qu'une observation au travers de celle-ci se traduit par une perception ( impression) visuelle identique pour toutes les translations possibles de cette fenêtre à l'intérieure de la région considérée ».*

Cette propriété d'invariance par translation de perception visuelle d'une texture est exploitée dans la plupart des méthodes de classification à partir d'attributs de texture [3], [4], [5], [6].

Selon Picard [7], il est difficile d'établir une définition de la texture. Il est cependant possible de dégager trois propriétés essentielles:

- La complexité d'une texture est difficilement quantifiable.
- Une texture est caractérisée par des variations importantes dans les hautes fréquences. Une zone uniforme ou quasiment lisse n'est pas une texture pour Picard.
- La notion de texture n'est valable que pour un ensemble de résolution et d'observation donné.

Selon Zhang [8], la notion de texture découle de la nécessité de décrire des informations d'une image de la manière la plus simple possible : *« une texture peut être considérée comme une façon de décrire les corrélations entre les différentes zones d'une région de l'image ».*

Notons que cette simplicité de description s'accompagne généralement de l'emploi de termes 'vague' dont la complexité est difficilement quantifiable. La texture permet de décrire des zones complexes de l'image composées de motifs élémentaires plus ou moins faciles à déterminer.

En résumé, les principales propriétés visuelles d'une texture que l'on dégage de la littérature sont les suivantes :

- Une texture ne peut être définie que sur un voisinage dont la taille dépend de la taille des motifs qui la composent.
- La notion de texture est liée à la résolution de l'observation. En effet, une région peut apparaître texturée à une certaine échelle et uniforme à une autre.

Ces propriétés visuelles sont importantes lors de la caractérisation d'une texture. Plusieurs paramètres de texture découlant de ces propriétés (granularité, caractères déterministe) permettent de discriminer différentes textures.

### **I.2.2 Approche formelle [1]**

L'approche formelle consiste à déterminer les éléments fondamentaux composant une texture et à décrire leurs interactions.

La définition du dictionnaire indique que la texture est la répartition spatiale d'un motif de base dans diverses directions de l'espace.

Selon Julesz [9], la texture apparaît comme la répartition spatiale régulière d'un unique motif géométrique appelé « Texton ». Il faut noter que cette définition n'est valable que pour les textures déterministes.

Selon Haralick [10], une texture peut être décomposée suivant deux dimensions.

La première concerne la description des primitives tonales qui la composent et la seconde définit les interactions entre ces propriétés tonales.

## **I.3 Différents types de textures [1]**

Il existe en générale trois types de textures :

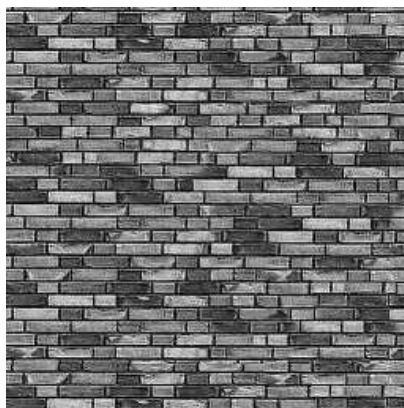
### **I.3.1 Textures structurelles**

Ces textures aussi nommées macrotextures présentent une structure spatiale régulière pouvant être générées par la répétition d'un motif ou d'une primitive. Une description structurale de la texture consistera alors à repérer dans l'image les éléments de base ou motifs, ainsi que leur arrangement.

Elles peuvent également correspondre à des images de synthèse. Ce type exige que l'image soit formée d'une texture régulière et se heurte à de nombreux problèmes. Tout d'abord, il est généralement difficile de repérer les motifs élémentaires, d'autant plus que les propriétés de ces motifs ne sont pas nécessairement constantes dans l'image (variation de l'éclairage). Un autre problème est dû à la répartition des primitives qui n'est pas forcément

régulière (un tissu avec des trames plus serrées par endroits), ce qui entraîne l'introduction d'un élément aléatoire.

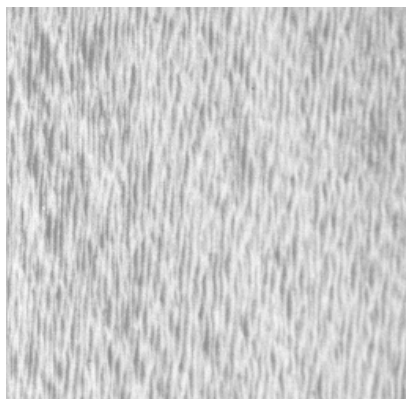
Notons que les textures parfaitement régulières existent rarement dans la réalité donc leur utilisation se limite à des applications très précises. Elles représentent en générale un milieu artificiel, par exemple un mur de briques (Fig.I.1).



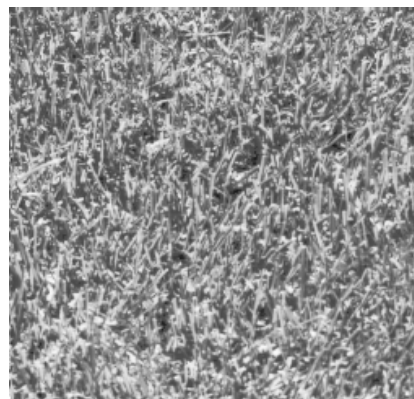
**Fig.I.1** : Mur de briques

### I.3.2 Textures aléatoires

Ces textures nommées aussi microscopiques sont décrites comme la réalisation d'un processus stochastique bidimensionnel où il est difficile d'isoler des motifs élémentaires. Ces textures présentent un aspect désordonné, mais sont néanmoins homogènes. Ce type de texture ne comporte pas des contours forts et possède au contraire une structure aléatoire par exemple l'eau, l'herbe...etc. (Fig.I.2 et Fig.I.3). Ces textures sont bien adaptées aux textures naturelles.



**Fig.I.2** : Eau



**Fig.I.3** : Herbe

### I.3.3 Textures directionnelles

Ces textures sont nommées aussi hybrides. Elles présentent un arrangement de structures élémentaires orientées, l'orientation est la propriété prédominante de ce type de textures qui ne sont pas totalement aléatoires.



Fig.I.4 Tissu

### I.4 Définition du motif d'une texture

L'approche formelle tente de définir les composantes d'une texture ainsi que leurs interactions. L'interaction entre ces motifs décrit la fréquence d'apparition du motif ainsi que sa localisation. Selon Haralick [10] et Francos [11], nous définissons le motif d'une texture en prenant en compte trois composantes :

- Le motif de base : c'est la plus petite structure spatiale de la texture.
- La transformation du motif de base : elle caractérise d'éventuelles déformations géométriques et des modifications des niveaux de gris du motif au sein de la même texture.
- La loi de probabilité de la transformation du motif: elle permet de définir la transformation d'un motif donné. Nous définissons donc le motif ou la primitive d'une texture de la façon suivante :

*« Le motif d'une texture est la plus petite structure spatiale contenue dans une texture dont la forme est régie par la même loi de probabilité au sein d'une texture ».*

## I.5 Finesse d'une texture

La finesse d'une texture est une propriété perceptuelle importante puisqu'elle caractérise la taille du motif d'une texture.

Selon Gagalowicz [12], une texture fine est caractérisée par une fonction d'autocorrélation tendant vers zéro rapidement en fonction de la distance de déplacement. Cette même fonction pour une texture grossière (primitives de grande taille) décroît beaucoup plus lentement. Dans le cas des textures déterministes, cette analyse ne doit se faire que sur la période de la texture considérée.

Cette définition caractérise la variation de la corrélation entre pixels plus ou moins proches pour une texture donnée.

Nous définissons la finesse d'une texture comme la caractérisation de la taille du motif élémentaire composant une texture à une résolution donnée. La résolution est liée à la distance d'observation d'une texture ou à la taille de la fenêtre d'analyse de celle-ci. La finesse contrairement au caractère déterministe ou aléatoire d'une texture, varie en fonction de la résolution de l'observation.

## I.6 Méthodes d'analyse de la texture [13]

Le but de l'analyse de texture est de formaliser les descriptifs de la texture par des paramètres mathématiques qui serviraient à l'identifier. En effet, pour un humain, une image n'est qu'une collection non structurée de pixels. Il possède des facultés visuelles qui lui permettent de regrouper les pixels de l'image qu'il perçoit en différentes régions. Plusieurs facteurs sont liés à ces facultés de regroupement dont les plus importantes sont la connexité, la similarité et la proximité.

L'absence d'une définition formelle et précise de la texture a conduit à l'élaboration d'une multitude de méthodes d'analyse de texture. Le choix d'une méthode de caractérisation d'une texture est étroitement lié à l'application visée et à la nature de la texture à discriminer.

Une multitude de méthodes, de variantes et de combinaisons de méthodes ont déjà été proposées dans la littérature. Celles ci peuvent être résumées dans l'organigramme suivant (Fig.I.5).

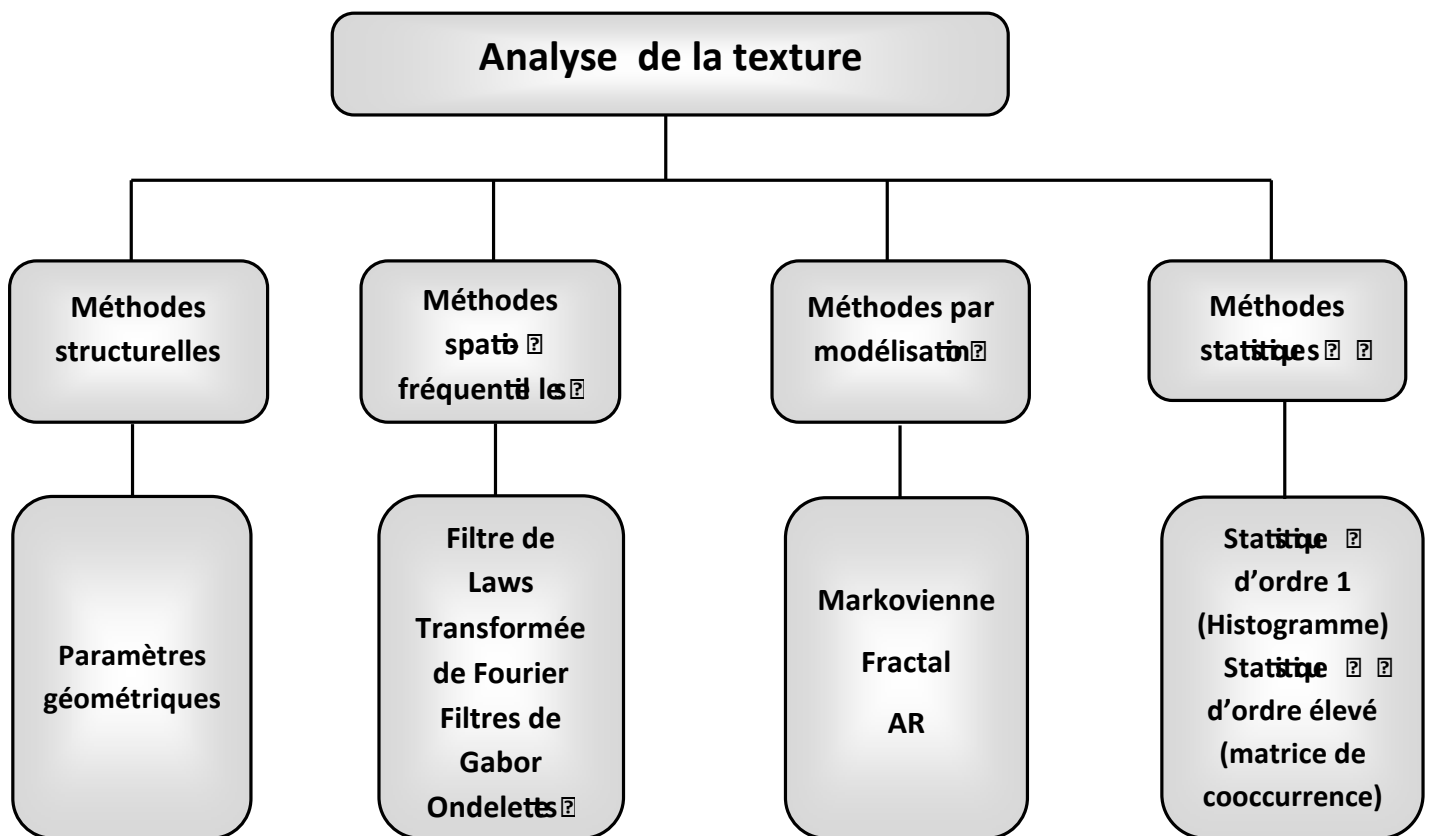


Fig.I.5 : Méthodes d'analyse de la texture

### I.6.1 Méthodes structurelles [14]

Ces méthodes tiennent compte de l'information structurelle et contextuelle d'une forme et sont particulièrement bien adaptées aux textures macroscopiques, où une structure spatiale forte se dégage par l'intermédiaire d'un motif plus ou moins répétitif.

Cette approche ne sera pas plus commentée, étant donné la diversité des techniques employées très liées à la nature des images. Le trait caractéristique de ces méthodes est que toutes se déroulent en deux étapes, l'identification des éléments constitutifs, puis la définition des règles de placement.

Notons qu'en principe, ces méthodes utilisent des techniques d'autocorrélation pour retrouver le placement dans la texture de la primitive extraite, afin d'en déduire une règle de placement.

### ▼ Fonction d'autocorrélation

La fonction d'autocorrélation permet de mettre en avant les dépendances spatiales existant entre certaines paires de pixels et d'obtenir de multiples informations perceptuelles sur la texture

Cette fonction mesure une distance entre une image originale et sa tradlatée. A partir de cette mesure, on peut estimer la taille des éléments d'une texture présente dans une image. Sur une fenêtre de taille  $(2W+1)*(2W+1)$ , centrée sur le pixel de coordonnées  $(j,k)$  et pour une translation  $(d_1, d_2)$ , la fonction d'autocorrélation peut être calculée de la manière suivante :

$$c(d_1, d_2, j, k) = \frac{\sum_{m=j-W}^{j+W} \sum_{n=k-W}^{k+W} g(m, n) * g(m - d_1, n - d_2)}{\sum_{m=j-W}^{j+W} \sum_{n=k-W}^{k+W} [g(m, n)]^2}$$

où

$$d_1, d_2 = 0, \pm 1, \pm 2, \dots, \pm T \Leftrightarrow -T \leq d_1, d_2 \leq T$$

avec

$T$  un entier positif et  $g(m, n)$  le niveau de gris du pixel  $(m, n)$ .

La fonction d'autocorrélation contient deux sortes d'informations qui sont résumées comme suit :

Une texture fine a une fonction d'autocorrélation qui décroît rapidement comparativement à celle d'une texture grossière. Si les primitives qui constituent la texture sont grandes, alors les petites translations ne produisent pas de grande différence dans la comparaison. Par conséquent, la fonction d'autocorrélation va montrer une décroissance lente. Une répétitivité des primitives entraîne une périodicité de la fonction d'autocorrélation.

A partir de cette fonction, la signature d'éléments de la texture peut être calculée de la manière suivante :

$$C(j, k) = \sum_{d_1=-T}^T \sum_{d_2=-T}^T d_1^2 d_2^2 C(d_1, d_2, j, k).$$

Plus  $C(j, k)$  augmente, plus la texture est grossière.

### I.6.2 Méthodes spatio-fréquentielles [15]

Cette approche exploite en même temps la représentation spatiale et la représentation fréquentielle de l'image. Une texture douce, pour laquelle on ne détecte des variations significatives du niveau de gris que sur un nombre important de points image, aura un spectre de puissance caractérisé par de fortes amplitudes dans le domaine des basses fréquences. Par contre pour une texture grossière et granuleuse, dans laquelle les variations du niveau de gris sont brutales et rapprochées, le spectre de puissance présente les plus fortes amplitudes dans le domaine des hautes fréquences. Elles font appel entre autre aux transformées en ondelettes et aux filtres de Gabor.

#### ▼ Filtres de Gabor

Le filtre de Gabor permet de filtrer l'image dans une direction et à une fréquence donnée. Il constitue un excellent compromis espace fréquence pour la représentation du contenu des images. Ce filtre à une base sinusoïdale modulée par une gaussienne.

*Sinus / cosinus* : analyse en fréquence

*Gaussienne* : différents niveaux de lissage (échelles)



Gaussienne \* sinus/cosinus = Gabor

Fig.I.6 : Représentation d'un filtre de Gabor

C'est donc un filtre passe bande ayant un gabarit gaussien. Il est très répandu par sa propriété de résolution optimale conjointe en fréquence et en temps. La fonction de filtre de Gabor est donnée par :

$$f_{Gb} = g(x', y') \exp[2\pi j(m_0(x - x_0)^2 + \nu_0(y - y_0)^2)]$$

avec

$$g(x', y') = \exp\left[-\frac{1}{2}\left(\frac{x'^2}{s_m^2} + \frac{y'^2}{s_v^2}\right)\right]$$

où

$s_m$  et  $s_v$  sont des constantes d'espace de l'enveloppe gaussienne qui déterminent l'étendue de l'onde dans les axes  $x$  et  $y$  respectivement. Le point  $(x_0, y_0)$  définit le point d'origine où s'applique la fonction  $f_{Gb}$ . En ce point, la fonction  $f_{Gb}$  est maximale.

$m_0$  et  $\nu_0$  sont des paramètres de position.

Le filtre de Gabor fait partie d'une famille de techniques d'analyse et de caractérisation de texture qui décrivent le rendu visuel par un mélange de signaux de fréquence, d'amplitude et de direction différentes. Il permet l'extraction directe de caractéristiques de texture localisées en fréquence et en orientation, en utilisant le principe de fenêtres gaussiennes, qui glissent sur l'image en capturant ses propriétés locales. Il s'appuie sur un banc de filtres très sélectifs en fréquence et en orientation.

Généralement, ces bancs analysent 6 directions polaires (pas de 30°) et 5 échelles (1,2,4,8,16) [Dunn et Higgins,1995] [Poursaberi et Araabi,2006].

### I.6.3 Méthodes statistiques

Dans ces méthodes, la texture est considérée comme la réalisation d'un processus stochastique stationnaire, où des paramètres statistiques sont estimés. Suivant la modalité des images à étudier, la signature la plus discriminante de la texture est à rechercher dans des méthodes qui exploitent directement les propriétés statistiques de la texture (statistique d'ordre 1, matrice de cooccurrence).

### ▼ Matrices de cooccurrences

Cette méthode a été introduite par HARALICK. Elle constitue l'une des méthodes les plus utilisées dans l'analyse de texture.

L'idée de cette méthode est d'identifier les répétitions des niveaux de gris de deux pixels séparés selon une direction  $\theta$  et une distance  $d$  souvent égale à 1 et les directions les plus utilisées sont  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  et  $135^\circ$ .

La matrice de cooccurrence notée  $M$  d'une région  $R$  pour une translation  $t$  est définie comme suit :

$$\left\{ M(i, j, d, \theta) = \text{card} \left\{ (m, m+t) \in R^2 \mid g_m = i, g_{m+t} = j \right\} \right\}$$

avec  $m$  représente un pixel.

C'est une matrice de taille  $N_g \times N_g$  ou  $N_g$  étant le niveau de gris maximale de l'image.

Quatorze indices ont été extraits de la matrice de cooccurrence pour définir une texture.

Les 5 paramètres les plus connus sont :

- **Contraste**

$$\text{Cont}(d, q) = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i-j)^2 M(i, j, d, q)^2$$

Chaque terme de la matrice  $M$  est pondéré par sa distance à la diagonale. On obtient un indice correspondant à la notion usuelle de contraste, il est élevé quand les termes éloignés de la diagonale de la matrice sont élevés, c'est-à-dire quand on passe souvent d'un pixel très clair à un pixel très foncé ou inversement.

- **Entropie**

$$\text{ENT}(d, q) = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} M(i, j, d, q) \log M(i, j, d, q)$$

L'entropie est faible si on a souvent le même nombre de pixels, forte si chaque couple est peu représenté. Elle fournit un indicateur de désordre que peut présenter une texture.

- **Corrélation**

$$Cor(d, q) = \frac{\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i - m_x)(j - m_y) M(i, j, d, q)}{s_x s_y}$$

$$M_x(i) = \sum_{j=0}^{N_g-1} M(i, j)$$

$$M_y(j) = \sum_{i=0}^{N_g-1} M(i, j)$$

$$m_x(i) = \sum_{j=0}^{N_g-1} j M_x(j)$$

$$m_y(j) = \sum_{i=0}^{N_g-1} i M_y(i)$$

$$s_x^2 = \sum_{i=0}^{N_g-1} (i - m_x)^2 M_x(i)$$

$$s_y^2 = \sum_{j=0}^{N_g-1} (j - m_y)^2 M_y(j)$$

Ce paramètre quantifie la dépendance directionnelle des niveaux de gris, il atteint ses plus grandes valeurs lorsque  $q$  est proche de l'orientation des lignes de la texture.

- **Homogénéité locale**

$$H(d, q) = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{1}{1 + (i - j)^2} M(i, j, d, q)$$

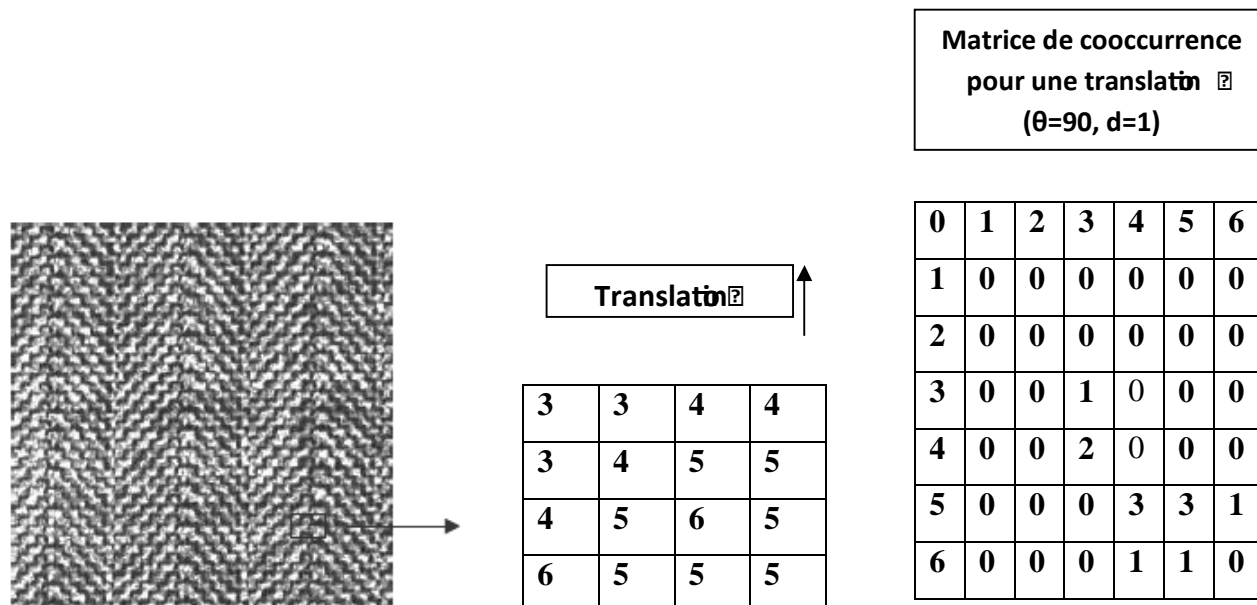
Il reflète l'existence des plages uniforme de texture et affecte un poids de plus en plus faible au fur à mesure qu'on s'éloigne de la diagonale principale. Plus la valeur de H est élevée plus la texture est grossière.

- **Variance**

$$Var(d, q) = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i - j)^2 M(i, j, d, q)$$

Ce paramètre mesure la somme des fréquences pour différents écarts des niveaux de gris.

La figure (I.7) illustre le principe de calcul d'une matrice de cooccurrence.



**Fig.I.7:** Tissu extrait de la banque de textures de Brodatz.  
(16 niveaux de gris)

#### I.6.4 Méthodes basées sur un modèle [16]

Le but de ces méthodes est d'obtenir un modèle générateur de la texture. Les paramètres de ces modèles permettent alors de caractériser ou synthétiser une texture. Les modèles les plus connus sont les modèles markoviens, les modèles fractals et modèles auto régressifs (AR).

Notre intérêt dans cette thèse est les modèles AR.

##### ✓ Modèle Auto régressif

On peut caractériser la dépendance entre les niveaux de gris d'une texture par un modèle auto régressif en considérant l'image comme une séquence temporelle de niveaux de gris dont la valeur à tout instant dépend des valeurs précédemment rencontrées au cours d'un balayage ligne par ligne. Un tel modèle permet de prévoir la valeur du niveau de gris d'un pixel en fonction des niveaux de gris de ses voisins.

Pour un modèle autorégressif à moyenne ajustée (ARMA), l'image est décrite par la formule :

$$x(i, j) = \sum_{(k,l) \in D_s} a_{k,l} x(i-k, j-l) + b_{0,0} e(i, j) + \sum_{(k,l) \in D_e} b_{k,l} e(i-k, j-l)$$

où

$D_s$  représente le domaine de prédiction lié à la sortie du filtre,  $D_e$  celui lié à l'entrée du filtre et  $e$  décrit un processus de bruit.

- Si les coefficients  $b_{k,l}$  sont nuls pour tout  $(k,l)$  appartenant à  $D_e$ , alors le processus est AR.
- Si les coefficients  $a_{k,l}$  sont nuls pour tout  $(k,l)$  appartenant à  $D_s$ , alors le processus est MA.

Le choix des domaines  $D_s$  et  $D_e$  incombe à l'utilisateur. Dans de nombreux cas on choisit  $D_s = D_e = D$ . Afin de conserver aux modèles ARMA une bonne localité, il est souhaitable que le nombre de termes dans  $D$  soit faible, mais pour bien représenter des signaux à longue périodicité, il vaut mieux qu'il soit grand. Ces modèles expriment naturellement une dépendance causale.

Dans le cas d'un modèle AR, les coefficients  $a(k,l)$  sont pris comme attributs de texture. Leur estimation fera l'objet du prochain chapitre.

## I.7 Segmentation d'images [17]

La segmentation d'image est une étape essentielle dans un système de traitement d'images. Elle partitionne l'image en un ensemble de régions pour pouvoir interpréter leur contenu. Pour segmenter une image en régions homogènes au sens de la texture, il est nécessaire de classer les pixels de l'image, la difficulté repose dans l'élaboration des critères définissant les objets à segmenter. Souvent des descripteurs statistiques de région sont utilisés pour obtenir des régions ayant des propriétés statistiques particulières. De nombreux algorithmes ont ainsi été proposés durant les dernières décennies. Ils sont basés sur différentes approches : contour, région, texture.

### I.7.1 Définition

La segmentation d'images peut être considérée comme un traitement de bas niveau, elle a pour but de permettre l'extraction d'éléments de l'image. Elle n'est généralement qu'une première étape essentielle dans le processus d'interprétation d'une scène.

Elle effectue une partition de l'image  $I$  en un certain nombre de régions disjointes, si  $I$  est une image composée de  $N$  sous-ensembles  $(I_1, I_2, \dots, I_N)$  formant une partition et  $p$  un prédicat d'uniformité, alors :

1.  $\bigcup_{i=1}^n I_i = I$ ;
2.  $\forall (i, j), i \neq j, I_i \cap I_j = \emptyset$ ;
3.  $\forall I_i, P(I_i) = \text{vrai}$ ;
4.  $\forall (i, j), I_i$  spatialement adjacent à  $I_j, p(I_i \cup I_j) = \text{faux}$ ;

**ū** La première condition implique que chaque pixel de l'image doit appartenir à une région  $I_i$ , et l'union de toutes les régions donne l'image originale.

**ū** La deuxième condition implique qu'un pixel ne doit pas appartenir à deux régions différentes.

**ū** La troisième condition stipule que chaque région doit respecter un prédicat d'uniformité.

**ū** La dernière condition implique la non réalisation de ce même prédicat pour la réunion de deux régions adjacentes.

Le résultat de la segmentation est une image dans laquelle une étiquette est attribuée à chaque pixel. L'étiquette d'un pixel correspond au numéro de la région à laquelle il appartient.

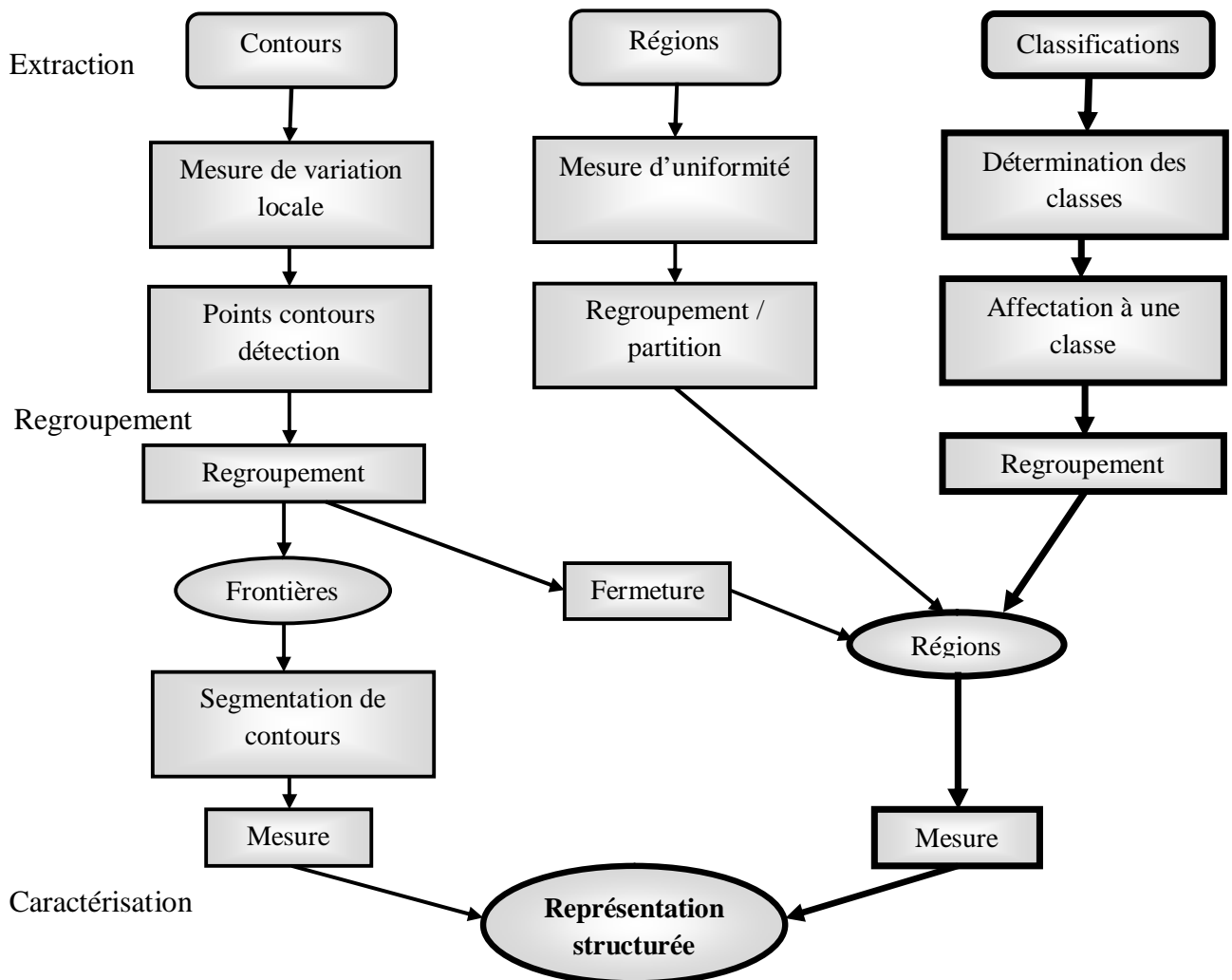
Pour une image texturée, une région  $I_i$  est considérée uniforme si elle est constituée de pixels ayant des attributs de textures identiques. Dans ce cas, la segmentation d'images texturées revient à identifier dans cette image les régions qui possèdent une propriété de texture uniforme.

**I.7.2 Approches de segmentation d'image [18], [1], [19]**

Les méthodes de segmentation d'images sont très nombreuses, on peut les classer en trois grandes classes :

- segmentation basée sur les régions (approche région).
- segmentation basée sur les contours (approche contours).
- segmentation par classification des pixels (approche classification).

Les différentes approches de la segmentation sont résumées dans l'organigramme (Fig.1.1) suivant :



**Fig.I.8 :** Approches de segmentation d'image

### I.7.2.1 Approche région

Cette méthode considère une région comme un ensemble de pixels connexes dont les attributs de texture sont proches les uns des autres. Elle procède par un balayage de l'image afin de construire les régions et fait intervenir l'information texturale pour la prise de décision. Trois approches sont distinguées :

- La segmentation par division de régions.
- La segmentation par croissance de régions.
- La segmentation par division-fusion de régions.

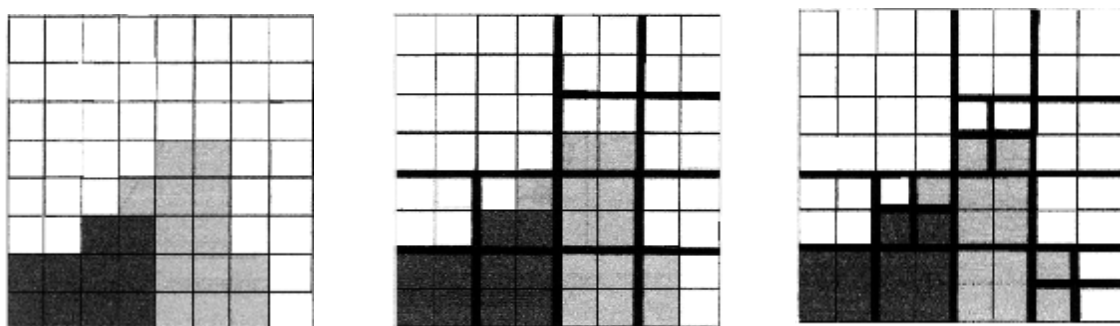
#### 1) Segmentation par division de régions (split) quad-tree

L'approche par division de régions consiste à diviser l'image originale en régions homogènes au sens d'un critère donné (uniformité des attributs de texture). Dans ces techniques, on divise l'image en quatre blocs de taille réduite à chaque fois. Trop de divisions amènent à travailler sur une image où les différences entre pixels ne sont plus significatives. Ce processus de division peut être représenté par un arbre quaternaire ou quad-tree.

C'est une arborescence dont la racine est l'image tout entière et dont les nœuds possèdent quatre fils. Chaque fils est un quadrant du nœud président. L'algorithme de découpage récursif s'arrête lorsque toutes les régions respectent le prédicat de l'homogénéité.

A cause des divisions en quatre sous régions, cette méthode est plutôt adaptée à des images carrées ayant un nombre de lignes et de colonnes égales à une puissance de deux et dans lesquelles les régions sont de formes rectangulaires. D'autre part, cette méthode a tendance à faire apparaître des effets de blocs.

L'exemple de la figure (I.9) illustre le principe de la division de régions.

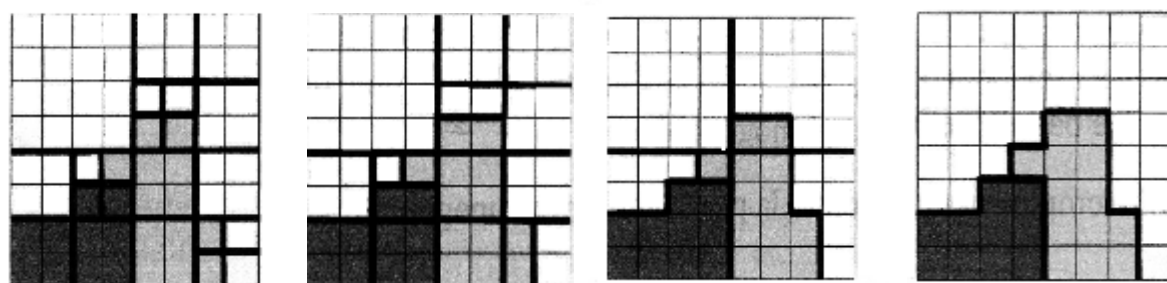


**Fig.I.9:** Division de régions

## 2) Segmentation par fusion

Développée initialement par Muerle et Allen 1968 [MuAl68], ce type d'approche consiste à choisir, d'abord, des germes de régions, puis à faire croître chaque germe en intégrant progressivement des pixels voisins à ces germes. Cette méthode présente des analogies avec le système de vision humaine puisque quand on regarde une image, on commence toujours par un point particulier, puis on cherche ensuite à reconnaître l'objet autour de ce point. Ici, l'image est d'abord divisée en cellules de  $K * K$  pixels ( $K = 2, 4$  ou  $8$ ). Ensuite, on balaye l'image de gauche à droite et de haut en bas. Dès qu'on rencontre une cellule comme le germe d'une nouvelle région, on teste la similitude entre la cellule en cours et la cellule germe pour chaque cellule adjacente à cette région.

L'exemple de la figure (I.10) montre le principe de la fusion de régions.



**Fig.I.10 : Segmentation par Fusion**

## 3) Segmentation par division-fusion (split and merge)

La segmentation par division-fusion regroupe les deux approches précédentes. Tout d'abord, l'image est divisée en régions homogènes qui respectent des critères globaux, puis les régions adjacentes qui répondent à des critères locaux sont fusionnées. Certains blocs adjacents dans l'image présentent des caractéristiques de texture identiques, c'est pourquoi ces couples de blocs sont par la suite fusionnés.

Le processus de segmentation comporte deux phases, dans la première phase, l'algorithme split and merge est appliqué en tenant compte que des niveaux de gris. Dans la deuxième phase, les régions formées lors de la première phase sont fusionnées si leurs attributs de texture sont proches. Les attributs utilisés sont ceux extraits à partir de la matrice de cooccurrence. L'analyse s'arrête lorsqu'il n'existe pas de couples qui respectent le prédicat de fusion.

### I.7.2.2 Segmentation par contours

L'extraction de contours joue un rôle primordial dans tout système de vision par ordinateur. Beaucoup d'efforts ont été fait pour extraire les contours d'une image et plusieurs techniques ont été proposées à ce jour. Les contours correspondent généralement à des changements brusques de propriétés physiques ou géométriques de l'image perçue et forment ainsi des attributs très important pour l'analyse.

Dans la plupart des techniques développées à ce jour, plusieurs opérateurs locaux (directionnel ou pas) de dérivation du premier ordre et du second ordre sont utilisées. Ils sont alors suivis respectivement d'une recherche de maxima locaux et de passage par zéro.

La segmentation par détection de contours s'appuie sur deux étapes essentielles.

#### 1) Etape de calcul du gradient

C'est une étape qui consiste à appliquer les opérateurs différentiels sur tous les points de l'image. On considère l'image comme une fonction  $I$  à deux dimensions. La dérivée première de  $I$  en un point  $M(x, y)$  connue sous le nom de **gradient**, est exprimée comme suit :

$$\nabla I(x, y) = \frac{\partial I}{\partial x}(x, y) + \frac{\partial I}{\partial y}(x, y)$$

On calcule le gradient en chaque point de l'image, puis on construit l'image de la norme gradient. On s'intéresse également au calcul des dérivées secondes, nommées **laplacien** qui est une fonction scalaire en chaque point de l'image. Exprimée comme suit :

$$\nabla^2 I(x, y) = \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2}$$

#### 2) Etape d'obtention des contours

Une fois que les vecteurs gradients ou laplacien sont calculés, les contours sont obtenus, soit en cherchant les points qui présentent un fort gradient qui correspondent aux maxima locaux de la norme gradient, ou on cherche les points de passage par zéro du laplacien.

### I.7.2.3 Segmentation par classification [20]

La classification est un processus qui consiste à regrouper des objets ou individus en classes ou groupes tel que les objets d'une même classe soient les plus similaires (même caractéristiques) possibles, alors que les objets issus de classes différentes soient les plus dissimilaires.

En segmentation d'images texturées, les objets correspondent aux pixels et les caractéristiques aux attributs de texture.

#### 1) Classification supervisée

Lorsque les différentes textures de l'image sont connues et l'appartenance de certains pixels (prototypes) à ces textures est connue a priori, alors la classification est dite supervisée.

Dans ce cas, elle consiste à construire à partir de ces pixels (prototypes) une fonction d'identification ou de discrimination pour les autres pixels. Cette fonction d'identification réalise un découpage de l'espace de représentation. A chaque zone de ce découpage est affectée une classe de texture a priori. Les autres pixels sont ensuite classifiés en fonction de leur position dans l'espace des paramètres. Cette classification concerne les méthodes bayésiennes, telle que la méthode des k plus proches voisins.

La méthode de K plus proche voisins se base sur le principe de comparaison c'est-à-dire on compare le vecteur d'attribut de chaque pixel à ceux des K plus proches voisins en termes d'attribut puis on lui affecte le numéro de la classe des pixels voisins qui dominant.

#### 2) Classification non supervisée

La classification non supervisée, telle qu'elle est définie en analyse de données, est un processus de regroupement d'éléments en un certain nombre de classes. Ce regroupement peut se faire soit sous forme de partition (on parle alors de classification par partitionnement) soit sous forme d'hierarchie (on parle alors de classification hiérarchique).

Le processus de classification est souvent rapproché à celui de la segmentation : le but de la segmentation est d'opérer une partition de l'image en zones connexes et homogènes sans aucune sémantique, alors que celui de la classification est de déterminer pour chaque pixel ou groupe de pixels de l'image à traiter, la classe de référence à laquelle il appartient. Ainsi, en segmentation la classification non supervisée est fréquemment employée lors d'une première

étape pour réaliser un partitionnement de l'espace des intensités des pixels (qu'il s'agisse de niveaux de gris, de couleurs ou de réponses spectrales). Puisqu'elle ignore complètement les notions de voisinage des pixels, une seconde étape d'analyse spatiale (étiquetage en composantes connexe par exemple) est ensuite nécessaire pour identifier les régions issues de la segmentation. La segmentation est alors un processus itératif, et à chaque itération, les classes obtenues peuvent être assimilées à des zones d'influence dans le contexte de la morphologie mathématique. Ce parallèle nous permet de bénéficier des algorithmes efficaces proposés dans ce domaine.

Dans le cadre non supervisé, les distributions a priori du champ aléatoire modélisant une image ne sont pas connues. Il faut donc une étape préliminaire afin d'estimer les paramètres relatifs à ces distributions.

Cette méthode se divise en deux familles :

- ▼ *Les méthodes statistiques.*
- ▼ *Les méthodes métriques.*

### Ø Méthodes statistiques

Les méthodes statistiques consistent à rechercher les modes de la fonction de densité de probabilité (*fdp*) sous jacents à la distribution de l'ensemble de pixels à classifier. En effet, les centres des classes correspondent aux régions de l'espace multidimensionnel caractérisées par une forte concentration des données. Ceci se traduit par des fortes valeurs de la *fdp* dans ces régions alors que les régions constituées de peu de données correspondent aux faibles valeurs de la *fdp*.

### Ø Méthodes métriques

Parmi les méthodes de classification non supervisées les plus connues on peut citer la méthode des « centres mobiles » ou la méthode des « K-means ». Dans cette méthode, chaque classes  $C_k$  ( $k = 1, \mathbf{K}, \mathbf{K}$ ), ( $k$  étant le nombre total des classes) est représenté par son centre de gravité noté  $m_k$  tel que  $m_k = [m_1, m_2, m_3, \mathbf{K}, m_{np}]^t$ ,  $np$  est le nombre d'attributs de texture.

On note par  $t_i = (t_{i1}, t_{i2}, \mathbf{L}, t_{inp})^t$  le vecteur caractéristique du pixel «  $i$  »,  $t_{ij}$  constitue alors l'attribut «  $j$  » du pixel «  $i$  ». Avec  $j = 1, \mathbf{K}, np$ .

- **Algorithme des K-means [14],[21]**

**Etape 1 :** Initialisation des centres de classes

Initialiser au hasard ou manuellement les  $N_c$  centres de gravité  $[m_1, m_2, m_3, \mathbf{K}, m_{N_c}]$  correspondant aux  $N_c$  classes.

**Etape 2 :** Affectation

1- affecter chaque pixel «  $i$  » à une classe  $C_k$ , dont le centre est  $m_k$ .

Un pixel «  $i$  » appartient à la classe  $C_k$  de centre  $m_k$  si et seulement si  $\|t_i - m_k\|$  est minimale pour tout  $k = 1, \mathbf{K}, N_c$ .

2- mettre à jour la position du centre de gravité  $m_k$  de la classe  $C_k$ .

$$m_{kj} = \frac{1}{N_k} \sum_{i \in C_k} t_{ij}$$

où  $N_k$  est le nombre de pixels de la classe  $C_k$ .

**Etape 3 :** Aller à l'étape 2 si les centres des classes changent entre deux itérations successives ou si un nombre maximal d'itérations  $n$  est pas atteint.

L'algorithme des K-means nécessite la connaissance du nombre de classes  $k$  et le choix des centres initiaux influe beaucoup sur la rapidité de convergence ainsi que sur le résultat final de la classification. Il est à noter que lorsque deux centres de gravité sont proches, les deux classes associées ne sont pas fusionnées. D'autre part, un centre de gravité isolé dans l'espace des paramètres peut être associé à une classe qui ne contient aucun pixel.

Malgré ces inconvénients l'algorithme des k-means reste très utilisé en pratique.

## I.8 Conclusion

Nous avons évoqué dans ce chapitre quelques notions sur la texture d'images. Il y'a lieu de retenir que la texture permet de caractériser l'état d'une surface dans une image et qu'on peut la définir pour chaque pixel en fonction des niveaux de gris de ses pixels voisins. Il est également important de noter que plusieurs méthodes d'analyse de la texture ont été développées. Parmi ces méthodes, nous avons retenu celle qui est basée sur les modèles auto régressifs que nous développerons plus en détails dans les chapitres qui suivent. Nous avons présenté également quelques généralités sur la segmentation d'images et nous avons adapté dans notre travail l'approche de la segmentation basée sur la classification. Celle-ci comprend deux étapes.

Les algorithmes de classification étant par essence très variés, nous avons fait le choix de nous intéresser ici à l'un des exemples les plus connus qui est l'algorithme des K-means.

# *Chapitre I*

---

## *Segmentation d'images texturées*

## II.1 Introduction

L'automatique consiste en l'étude des systèmes réels des différentes disciplines scientifiques (Electronique, mécanique, thermique, chimie, écologie, biologie, économie, sociologie, physique...), en vue de l'analyse, de la prédiction, de la surveillance, de la commande, et/ou de l'optimisation des systèmes. La condition nécessaire pour cela est l'obtention d'un modèle mathématique du système réel (on réalise une modélisation).

Dans ce chapitre, nous présentons un panorama des méthodes de modélisation et d'identification paramétrique des modèles AR 1D et 2D.

## II.2 Modélisation [22]

La modélisation réalisée à partir d'un comportement du système et /ou de lois physiques, consiste à déterminer la structure des équations qui régissent le comportement de ce système, et aussi, à fixer, a priori la valeur de ses paramètres (longueurs, masses, inerties, capacités, résistances, frottements...). Mais, il est souvent impossible d'obtenir une connaissance a priori complète et précise de tous les paramètres du modèle.

On distingue deux types de modélisation:

- Modélisation non paramétrique.
- Modélisation paramétrique.

### II.2.1 Modélisation non paramétrique

Ce type de modélisation peut être obtenu par une analyse fréquentielle ou par la réponse impulsionnelle du système. Généralement il consiste à estimer sa densité spectrale.

### II.2.2 Modélisation paramétrique

La modélisation d'un signal par des techniques paramétriques nécessite une intervention humaine sur le modèle. Il faut, dans un premier temps, définir le modèle le mieux adapté au signal à modéliser.

Elle consiste à associer à un signal un modèle, représenté par un vecteur paramètre  $q = [q_1, q_2, \dots, q_p]^T$  censé représenter au mieux le signal considéré. Etant donné que l'on choisit a priori un modèle pour le signal, cela signifie qu'en général on possède des informations a priori sur le signal lui-même qui permettent de sélectionner tel ou tel modèle. Il s'ensuit que le choix d'un modèle plutôt que d'un autre requiert au préalable une analyse du signal.

### II.2.2.1 L'intérêt d'une modélisation paramétrique

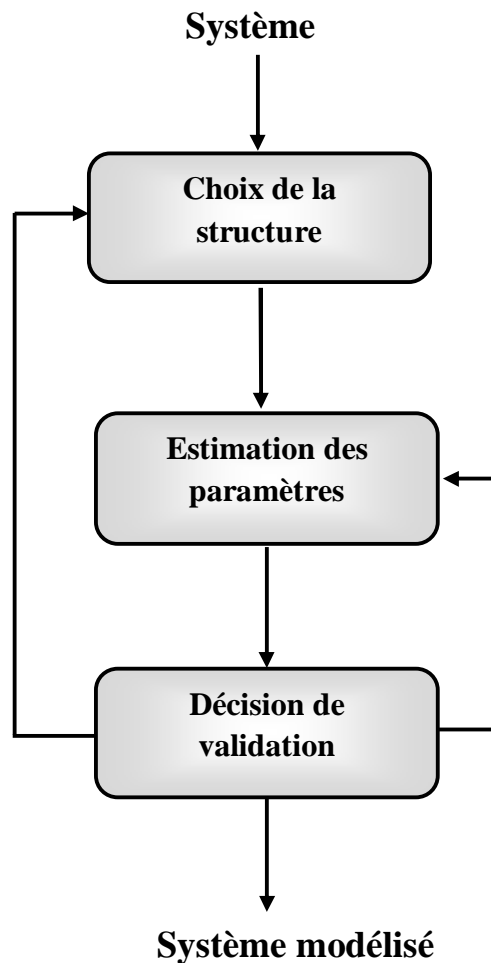
Elle permet de réduire l'espace de représentation. On peut ainsi représenter un ensemble de  $N$  échantillons par un vecteur de dimension  $p \ll N$ . Ceci est particulièrement intéressant dans de nombreuses applications :

- **Classification** : lorsque l'on désire classer des signaux suivant un certain nombre de classes, on substitue au signal ses paramètres et on classe suivant ces paramètres.
- **Détection** : si l'on veut détecter les ruptures dans un modèle, il suffit de suivre un petit nombre de paramètres pour déterminer l'instant de rupture.
- **Transmission-Compression** : au lieu de transmettre un signal, on ne transmet que les paramètres de modèle associé.

Elle permet d'extraire de façon plus fine certaines informations. C'est par exemple le cas en analyse spectrale où la modélisation permet d'estimer avec une meilleure résolution le spectre d'un signal.

### II.2.2.2 Etapes de la modélisation paramétrique [23]

La modélisation paramétrique des signaux (et des systèmes) se compose des trois principales étapes représenté sur la figure (II.1) :



**Fig.II.1:** Etape de la modélisation paramétrique.

- **Choix de la structure du modèle**

Ce choix est le résultat d'un compromis entre la complexité du modèle (ARMA ou AR) et de son aptitude à représenter correctement le système dans le domaine de l'étude.

- **Estimation des paramètres**

Les paramètres du modèle sont estimés de façon à minimiser un critère représentatif de l'équivalence signal-modèle.

- **Test de validation du modèle**

L'importance de cette étape réside dans la vérification de l'aptitude du modèle obtenu à représenter la classe de signaux considérés (vérification du choix de la structure et la méthode d'estimation). On suppose que le modèle estimé est correct et on utilise les propriétés théoriques du modèle pour extraire l'information utile.

### II.3 Modèles linéaires [24], [22], [25]

#### II.3.1 Modèle ARMA

Un modèle général Auto-Regressif à Moyenne Ajustée d'ordre  $(p, q)$  ou ARMA  $(p, q)$ , (Auto **R**egressive **M**oving **A**verage), est composé de deux parties, une partie autorégressive AR et une partie mobile MA.

Un modèle ARMA peut être défini par la relation suivante :

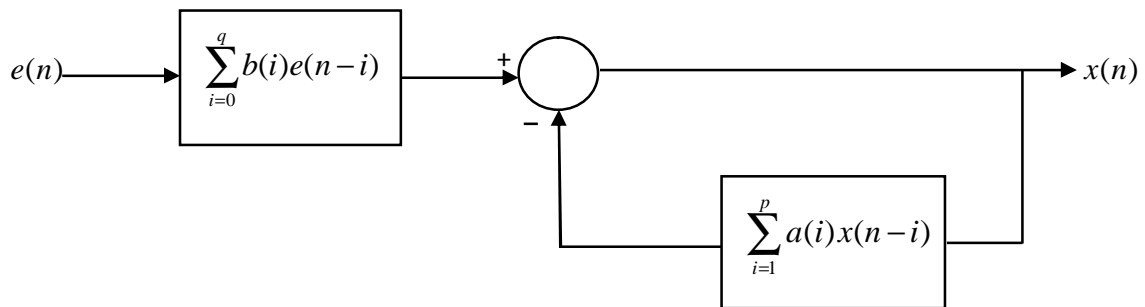
$$x(n) = -\sum_{i=1}^p a(i)x(n-i) + \sum_{i=0}^q b(i)e(n-i) \tag{II.1}$$

$e(n)$  : l'entrée du filtre ARMA, considérée comme un bruit blanc.

$x(n)$  : la sortie du modèle ARMA.

$a(i), b(i)$  : les paramètres du modèle ARMA.

Le modèle ARMA peut être représenté par le schéma bloc de la figure (II.2).



**Fig.II.2** : Schéma bloc de la représentation temporelle du modèle ARMA.

Pour  $a(0) = 0$  le modèle est non causal.

Soit  $X(z)$ , la transformée en  $z$  de la séquence  $x(n)$  définie par :

$$X(z) = \sum_{n=-\infty}^{+\infty} x(n)z^{-n} \tag{II.2}$$

sachant que

$$TZ \quad x(n-1) = z^{-1} X(z) \tag{II.3}$$

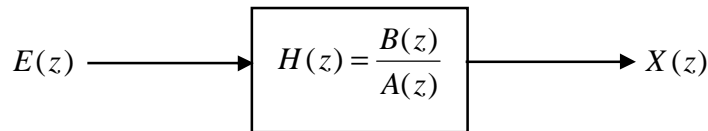
la TZ de l'équation (II.1) nous donne

$$X(z) + a_1 z^{-1} X(z) + \dots + a_p z^{-p} X(z) = b_0 E(z) + b_1 z^{-1} E(z) + \dots + b_q z^{-q} E(z)$$

soit :

$$\frac{X(z)}{E(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_q z^{-q}}{1 + a_1 z^{-1} + \dots + a_p z^{-p}} = H(z) \tag{II.4}$$

$H(z)$  est la fonction de transfert en  $(z)$  du modèle qui peut être représenté par la figure (II.3) suivante :



**Fig.II.3 :** Schéma bloc de la représentation spectrale du modèle ARMA.

On peut interpréter le modèle ARMA comme un filtre de fonction de transfert  $H(z)$  excité par une entrée  $E(z)$  et délivrant à sa sortie un signal  $X(z)$ . Il correspond au filtre numérique à réponse impulsionnelle infinie. Pour assurer que le processus  $x(n)$  soit stationnaire, il faut que  $H(z)$  soit stable et causal, c'est-à-dire que  $A(z)$  ait ses zéros à l'intérieur du cercle unité.

### II.3.2 Modèle à moyenne glissante (MA)

Une séquence aléatoire  $x(n)$  est appelée processus à moyenne ajustée (MA) d'ordre  $q$  lorsqu'on peut l'écrire sous forme d'une moyenne pondérée courante de variables aléatoires non-corrélés.

$$x(n) = \sum_{i=0}^q b(i)e(n-i) \tag{II.5}$$

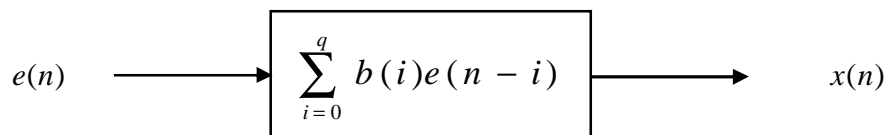
où :

$b(i)$  : sont les paramètres du modèle MA

$q$  : est l'ordre du modèle MA

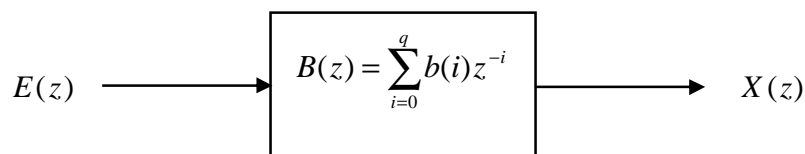
$e(n)$  : est un bruit blanc centré de variance  $\sigma^2$ .

Dans cette partie, les  $a(i)$  sont nuls  $a(i) = 0 \quad i \neq 0$



**Fig.II.4 :** Schéma bloc de la représentation temporelle du modèle MA

Dans le domaine spectral la fonction de transfert est alors  $H(z) = B(z)$  et  $A(z) = 1$ . Le modèle MA est dit : modèle tout zéro, car ce dernier sera caractérisé par un polynôme ayant que des zéros. C'est aussi un filtre stable.

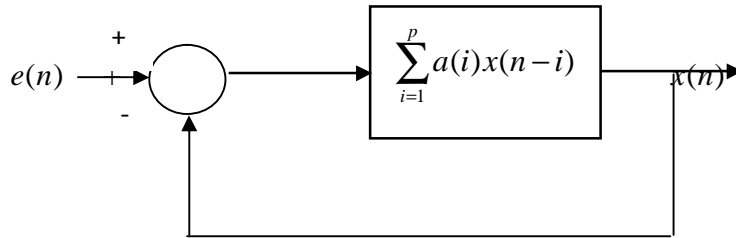


**Fig.II.5 :** Schéma bloc de la représentation spectrale du modèle MA

### II.3.3 Modèle AR

Un modèle autorégressif est un filtre tout-pôle (composé uniquement de pôles) au travers duquel passe un bruit blanc. Dans le cas où les coefficients  $b(i)$  du modèle ARMA sont nuls, on aboutit au modèle AR défini par la relation suivante :

$$x(n) = - \sum_{i=1}^p a(i)x(n-i) + e(n) \tag{II.6}$$



**Fig.II.6 :** Schéma bloc d'une représentation temporelle du modèle AR.

$a(i) : i = 1, \dots, p$  : sont les paramètres du modèle AR.

$p$  : L'ordre du modèle AR.

La fonction de transfert d'un filtre autorégressif d'ordre  $p$  est donnée par :

$$H(z) = \frac{1}{1 + \sum_{i=1}^p a(i)z^{-i}} \quad (\text{II.7})$$

Dans ce cas, le polynôme  $B(z)$  se réduit à une constante  $B(z)=1$  et la fonction de transfert  $H(z)$  ne contient que des pôles, d'où l'appellation du modèle tout-pôles.

Après avoir étudié les modèles ARMA et MA plusieurs travaux ont comparé les performances obtenues par les types de modèle dans le traitement du signal. Vu à la non linéarité de la modélisation par la partie MA et le risque d'obtention de modèles instables par la partie ARMA, la partie AR a pris faveur.

## II.4 Utilisation des modèles AR

Les modèles AR sont très sollicités pour :

- La modélisation et l'identification des processus.
- Le calcul de la densité spectrale.
- La prédiction linéaire d'un signal.
- **Le calcul de la densité spectrale**

$x(n)$

La fonction d'autocorrélation du signal que nous supposons réel, stationnaire et de moyenne nulle est :

$$r_x(m) = E[x(n)x(n+m)] \quad (\text{II.8})$$

et qui est une fonction symétrique par rapport à l'origine

$$r_x(m) = r_x(-m) \quad (\text{II.9})$$

La transformée de Fourier de cette fonction d'autocorrélation est la densité spectrale qui est une fonction réelle positive ou nulle. Cette densité spectrale est la moyenne du module de la transformée de Fourier du signal  $x(n)$ , sa formule est donnée comme suit :

$$\left\{ \begin{array}{l} S_x(z) = \frac{\sigma_e^2}{A_p(z)A_p(z^{-1})} \\ \text{Avec} \\ A_p(z) = 1 - \sum_{i=1}^p a(i)z^{-i} \end{array} \right. \quad (\text{II.10})$$

- **La prédiction linéaire**

Il s'agit ici de modéliser le signal directement dans l'espace des temps et non pas dans l'espace des fréquences. L'astuce ici est d'utiliser un modèle particulier.

Si  $x(n)$  représente la série monodimensionnelle temporelle mesurée, alors le modèle est celui représenté par la formule (II.6).

On voit qu'un point de la série de données peut être ainsi prédit par une relation linéaire à partir des points précédents.

## II.5 Estimation des coefficients AR

Le problème de la modélisation AR revient à déterminer les coefficients du modèle ou du filtre tout-pôles dont on connaît le signal de sortie et celui de l'entrée. Il est par conséquent

nécessaire d'adopter un critère, afin de faire un choix parmi l'infinité de solution possible. Le critère classiquement utilisé est celui de la minimisation de l'énergie de l'erreur de prédiction.

Ce problème peut être résolu par la méthode des moindres carrés, de Levinson, de Burg ou de covariance.

### II.5.1 Equations de Yule Walker

On considère un processus  $x(n)$  comme la sortie d'un système linéaire et invariant dans le temps dont l'entrée  $e(n)$  est un bruit blanc stationnaire de moyenne nulle et de variance  $\sigma_e^2$ .

Dans le cas du modèle AR on a :

$$x(n) = -\sum_{i=1}^p a(i)x(n-i) + e(n) \quad (\text{II.11})$$

Le principe des équations de Yule Walker est d'exprimer les paramètres AR inconnus en fonction de la fonction d'autocorrélation  $r_x(m)$ .

- En multipliant ainsi les deux membres de l'équation (II.11) par  $x(n-m)$  et en prenant l'espérance, on obtient :

$$E[x(n)x(n-m)] = E[-\sum_{i=1}^p a(i)x(n-i)x(n-m)] + E[e(n)x(n-m)] \quad (\text{II.12})$$

$$E[x(n)x(n-m)] = -\sum_{i=1}^p a(i)E[x(n-i)x(n-m)] + E[e(n)x(n-m)] \quad (\text{II.13})$$

Or  $E[x(n)x(n-m)]$  n'est rien d'autre que l'autocorrélation du signal  $x(n)$  notée  $r_x(m)$

Dans le cas d'un signal ergodique, la fonction d'autocorrélation est :

$$r_x(m) = \frac{1}{N-m} \sum_{i=0}^{N-m} x(i)x(i+m) \quad (\text{II.14})$$

Comme les termes du bruit blancs sont indépendants les uns des autres et, de plus  $x(n-m)$  est indépendant de  $e(n)$  lorsque  $m$  est plus grand que zéro, on a :

Pour  $m \neq 0, E[e(n)x(n-m)] = 0$ . ( $e(n)$  est décorrélé avec le passé de  $x(n)$ ).

Par contre, pour  $m = 0$  on a :

$$E[e(n)x(n)] = E[e(n)(-\sum_{i=1}^p a(i)x(n-i) + e(n))] = \sum_{i=1}^p -a(i)E[e(n)x(n-i)] + E(e^2(n)) = 0 + \sigma_e^2 \quad (\text{II.15})$$

- En regroupant les deux cas précédents, on a pour  $m \geq 0$ .

$$r_x(m) = E[\sum_{i=1}^p -a(i)x(n-i)x(n-m)] + \sigma_e^2 d_m \quad (\text{II.16})$$

avec

$$d_m = \begin{cases} 1 & \text{si } m = 0 \\ 0 & \text{si } m \neq 0 \end{cases}$$

Par ailleurs, on a :

$$E[-\sum_{i=1}^p a(i)x(n-i)x(n-m)] = -\sum_{i=1}^p a(i)E[x(n)x(n-m+i)] = -\sum_{i=1}^p a(i)r_x(m-i) \quad (\text{II.17})$$

- En regroupant les deux termes de l'équation (II.17) on obtient les équations dites de Yule Walker :

$$r_x(m) = -\sum_{i=1}^p a(i)r_x(m-i) + \sigma_e^2 d_m \quad (\text{II.18})$$

$$\implies \sum_{i=0}^p a(i)r_x(m-i) = \sigma_e^2 d_m \quad (\text{II.19})$$

avec

$$a(0) = 1$$

Cette relation peut s'écrire sous forme matricielle comme suit :

$$\begin{pmatrix} r_x(0) & r_x(-1) & \mathbf{L} & r_x(-p) \\ r_x(1) & r_x(0) & \mathbf{L} & r_x(1-p) \\ \mathbf{M} & \mathbf{M} & \mathbf{L} & \mathbf{M} \\ r_x(p) & r_x(p-1) & \mathbf{L} & r_x(0) \end{pmatrix} \begin{pmatrix} 1 \\ a_1 \\ \mathbf{M} \\ a_p \end{pmatrix} = \begin{pmatrix} \mathbf{S}_e^2 \\ 0 \\ \mathbf{M} \\ 0 \end{pmatrix} \quad (\text{II.20})$$

Notons aussi que la fonction d'autocorrélation est paire. Pour  $m \neq 0$ , on a

$$r_x(m) = r_x(-m) = \sum_{i=1}^p a(i)r_x(|m|-i) + \mathbf{S}_e^2 d_m \quad (\text{II.21})$$

Pour déterminer les coefficients AR, il faut résoudre le système d'équations ainsi obtenu ou en éliminant la première ligne et la première colonne, le système devient :

$$Ra = -r \quad (\text{II.22})$$

$$R = \begin{pmatrix} r_x(0) & r_x(-1) & \mathbf{L} & r_x(-p+1) \\ r_x(1) & r_x(0) & \mathbf{L} & r_x(-p+2) \\ \mathbf{M} & \mathbf{M} & \mathbf{L} & \mathbf{M} \\ r_x(p-1) & r_x(p-2) & \mathbf{L} & r_x(0) \end{pmatrix} \quad (\text{II.23})$$

$$a = \begin{pmatrix} a_1 \\ a_2 \\ \mathbf{M} \\ a_p \end{pmatrix} \quad (\text{II.24})$$

$$r = \begin{pmatrix} r_x(1) \\ r_x(2) \\ \mathbf{M} \\ r_x(p) \end{pmatrix} \quad (\text{II.25})$$

Ce système étant constitué d'équations linéaires peut être résolu par des méthodes classiques (Gauss, Jordan, Jacobi, Gauss-Seidel, Cholesky, etc....). Cependant ces méthodes exigent un grand nombre d'opérations.

Etant donné que  $R$  est une matrice qui est :

- Définie positive
- Hermitienne  $r_x^*(m) = r_x(-m)$  (symétrique dans le cas des signaux réels)
- Toeplitz : les éléments de chaque diagonale (principale et secondaire) sont égaux.

La résolution du système se fait alors d'une manière efficace grâce à l'algorithme de Levinson-Durbin.

### II.5.1.1 Algorithme de Levinson-Durbin

Son principe consiste à exprimer les paramètres d'ordre  $k$  en fonction des paramètres d'ordre  $k-1$ .

Les paramètres  $a(i)$  et la variance de l'erreur sont déterminés d'une manière récursive

comme suit :

#### 1) Initialisation

$$k = 1$$

$$a_1[1] = \frac{-r_x(1)}{r_x(0)}; \quad s_1^2 = (1 - |a_1[1]|^2)r_x(0)$$

#### 2) Récursion

Pour  $k = 2, \dots, p$

$$a_k[k] = \frac{-1}{s_{k-1}^2} [r_x(k) + \sum_{l=1}^{k-1} a_{k-1}[l]r_x(k-l)]$$

$$a_k[l] = a_{k-1}[l] + a_k[k]a_{k-1}^*[k-l]$$

$$s_k^2 = (1 - |a_k[k]|^2)s_{k-1}^2$$

#### 3) Finalement, on déduit

$$s_p^2 = s_e^2 \quad \text{et} \quad a_p[1] = a(1)$$

**M**  
**M**  
**M**

$$a_p[p] = a(p)$$

**Remarques**

- ✓ Les coefficients  $a_k[k]$  sont appelés coefficients de réflexion.
- ✓ L'algorithme exige  $p^2$  opérations.
- ✓ Le meilleur prédicteur d'ordre  $p$  d'un processus AR ( $p$ ) est donné par les paramètres AR. Autrement dit celui qui rend l'erreur orthogonale aux données. En effet, pour minimiser l'erreur entre  $x(n)$  une combinaison linéaire des  $\{x(n-i)\}$ ,  $i = 1, \dots, p$ , il faut projeter orthogonalement  $x(n)$  sur l'espace engendré par  $\{x(n-i)\}$ ,  $i = 1, \dots, p$ .

De fait, l'erreur de prédiction devient orthogonale aux  $\{x(n-i)\}$ ,  $i = 1, \dots, p$ . Ceci peut se retrouver en écrivant que,  $\forall i \in [1, p]$ .

$$\left\{ E\{e(n)x^*(n-i)\} = E\left\{ \sum_{i=0}^p a(i)x(n-i)x^*(n-i) \right\} = \sum_{i=0}^p a(i)r_x(n-i) = 0 \right\} \quad (\text{II.26})$$

Au vu de (II.19), les coefficients optimaux  $a(i)$  sont donc ceux qui rendent l'erreur de prédiction  $e(n)$  orthogonale (non corrélée) aux données  $\{x(n-1), \dots, x(n-p)\}$ . L'innovation  $e(n)$  peut être vue comme l'information qui est contenue dans  $x(n)$  et qui n'était pas dans  $\{x(n-1), \dots, x(n-p)\}$ .

**II.5.1.2 Algorithme de Burg**

Les coefficients de réflexion obtenus par l'algorithme de Levinson-Durbin peuvent être supérieurs à 1, ce qui entraîne une instabilité du filtre AR. Afin d'éliminer cet inconvénient, Burg a proposé une autre approche appelée méthode de maximum d'entropie. Comme on l'a vu, si l'on suppose connus  $\{r_x(i)\}$   $i = 0, \dots, p$ , alors la fonction d'autocorrélation est implicitement extrapolée (via les équations de Yule-Walker) pour un processus AR.

Burg s'est alors posé la question suivante pour un processus quelconque : si l'on connaît  $\{r_x(i)\}$   $i = 0, \dots, p$ , comment est-il possible d'extrapoler  $r_x(p+1), r_x(p+2), \dots$  de manière à garantir le caractère semi-défini positif de la fonction de corrélation ? Il a alors montré que parmi toutes les extrapolations possibles, celle qui donne l'entropie maximale correspond à un processus AR. En d'autres termes, le processus AR est "le processus le plus aléatoire" qui permet d'extrapoler la fonction d'autocorrélation à partir de la connaissance de

$\{r_x(i)\} \quad i=0,\dots,p$ . Burg considère cette propriété importante dans la mesure où elle permet de faire le moins d'hypothèses possible (telle que corrélation nulle) sur la fonction d'autocorrélation en dehors de  $\{r_x(i)\} \quad i=0,\dots,p$ .

Du fait de l'analogie que nous venons de présenter, la méthode du maximum d'entropie est souvent assimilée à la modélisation AR.

L'algorithme de Burg utilise la contrainte de Levinson sur les coefficients autorégressifs et optimise les coefficients de réflexion par minimisation de la moyenne quadratique des erreurs de prédiction directe (ou progressive) et rétrograde.

$$E = \frac{1}{2} \sum_{n=p}^{N-1} \left| e_p^f(n) \right|^2 + \left| e_p^b(n) \right|^2 \quad (\text{II.27})$$

où

$e_p^f(n)$  représente l'erreur de prédiction directe à l'ordre  $p$ .

$$e_p^f(n) = x(n) - \hat{x}(n) \quad (\text{II.28})$$

avec

$$\hat{x}(n) = -\sum_{m=1}^p a(m)x(n-m) \quad (\text{II.29})$$

Burg a montré que :

$$e_p^f(n) = x(n) + \sum_{m=1}^p a_p[m]x(n-m) = e_{p-1}^f(n) + k_p e_{p-1}^b(n-1) \quad (\text{II.30})$$

$e_p^b(n)$  est l'erreur rétrograde définie comme étant l'erreur commise l'ors qu'un échantillon est prédit par une combinaison linéaire des échantillons postérieurs.

$$e_p^b(n) = x(n-p) + \sum_{m=1}^p a_p^*[m]x(n-p+m) = e_{p-1}^b(n-1) + k_p^* e_{p-1}^f(n) \quad (\text{II.31})$$

$$e_p^b(n) = x(n-p) - \hat{x}(n-p) \quad (\text{II.32})$$

avec

$$\hat{x}(n-p) = -\sum_{m=1}^p a_m^* x(n+m-p) \quad (\text{II.33})$$

$$k_p = \frac{-2 \sum_{n=p}^{N-1} e_{p-1}^f(n) e_{p-1}^{b*}(n-1)}{\sum_{n=p}^{N-1} |e_{p-1}^f(n)|^2 + |e_{p-1}^b(n-1)|^2} \quad (\text{II.34})$$

L'algorithme de Burg est résumé comme suit :

1) Initialisation

$$r_x(0) = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2$$

$$s_0 = r_x(0)$$

$$e_0^f(i) = x(i) \quad i = 1, 2, \dots, N-1$$

$$e_0^b(i) = x(i) \quad i = 0, 1, \dots, N-1$$

2) Calcul des coefficients de réflexion

Pour  $k = 1, 2, \dots, p$

$$a_k[k] = k_k = \frac{-2 \sum_{i=k}^{N-1} e_{k-1}^f(i) e_{k-1}^{b*}(i-1)}{\sum_{i=k}^{N-1} |e_{k-1}^f(i)|^2 + |e_{k-1}^b(i-1)|^2}$$

3) Calcul de la récursion de Levinson

$$a_k[j] = a_{k-1}[j] + a_k[k] a_{k-1}[k-j] \quad j = 1, \dots, k$$

$$s_k^2 = (1 - |a_k[k]|^2) s_{k-1}^2$$

4) Calcul des erreurs

$$e_k^f(i) = e_{k-1}^f(i) + a_k[k] e_{k-1}^b(i-1) \quad i = k+1, \dots, N-1$$

$$e_k^b(i) = e_{k-1}^b(i-1) + a_k[k] e_{k-1}^f(i) \quad i = k, \dots, N-2$$

### II.5.1.3 Méthode de covariance [24]

La méthode de Yule-Walker minimise l'erreur de prédiction sur l'ensemble des données  $x(n)$  ce qui oblige à compléter la séquence par des zéros au début et à la fin des données.

Le calcul des coefficients AR par la méthode de covariance consiste à minimiser l'erreur suivante :

$$\sum_{n=0}^{N+p-1} e^2(n) = \sum_{n=0}^{N+p-1} (x(n) + \sum_{i=1}^p a(i)x(n-i))^2 \tag{II.35}$$

Cette minimisation aboutit au système d'équations suivant :

$$\begin{pmatrix} C_x(1,1) & C_x(1,2) & \mathbf{K} & C_x(1,p) \\ C_x(2,1) & C_x(2,2) & \mathbf{K} & C_x(2,p) \\ \mathbf{M} & \mathbf{M} & \mathbf{K} & \mathbf{M} \\ C_x(p,1) & C_x(p,2) & \mathbf{K} & C_x(p,p) \end{pmatrix} \begin{pmatrix} a(1) \\ a(2) \\ \mathbf{M} \\ a(p) \end{pmatrix} = \begin{pmatrix} C_x(1,0) \\ C_x(2,0) \\ \mathbf{M} \\ C_x(p,0) \end{pmatrix} \tag{II.36}$$

où

$C_x(j,k)$  représente le coefficient de covariance du signal  $x(n)$ . Si  $x(n)$  est ergodique, on a :

$$C_x(j,k) = \frac{1}{N-p} \sum_{n=p}^{N-1} x(n-j)x(n-k) \tag{II.37}$$

$$0 \leq j \leq p \quad \text{et} \quad 0 \leq k \leq p$$

La résolution du système d'équations linéaires nous permet de déterminer les coefficients  $a(i)$  ( $i=1, \mathbf{K}, p$ ) La variance du bruit d'entrée  $s_e^2$  peut être estimée par :

$$s_e^2 = \min \left( \sum_{n=p}^{N-1} e^2(n) \right) = C_x(0,0) + \sum_{i=1}^p a(i)C_x(0,i) \tag{II.38}$$

La matrice de covariance n'est pas du type Toeplitz, par conséquent l'algorithme de Levinson-Durbin ne peut pas être appliqué pour résoudre le système.

Cette matrice est par contre symétrique définie positive, l'algorithme de Cholesky peut être utilisé pour la résolution du système.

Cependant, il y'a lieu de noter que la stabilité du filtre ainsi calculé par cette méthode n'est pas garanti.

Pour éviter cet inconvénient, la méthode de covariance modifiée propose de minimiser la somme des erreurs de prédiction directe et rétrograde comme dans l'algorithme de Burg.

$$\sum_{n=p}^{N-1} [e_p^f(n)]^2 + \sum_{n=0}^{N-1-p} [e_p^b(n)]^2 \quad (\text{II.39})$$

La minimisation de cette équation aboutit au même système d'équation précédent sauf que :

$$C_x(j,k) = \frac{1}{2(N-p)} \left[ \sum_{n=p}^{N-1} x(n-j)x(n-k) + \sum_{n=0}^{N-1-p} x(n+j)x(n+p) \right] \quad (\text{II.40})$$

#### II.5.1.4 Méthode des moindres carrés [24]

L'estimation des coefficients AR peut être aussi obtenue en minimisant l'erreur :

$$e(n) = x(n) + \sum_{i=1}^p a(i)x(n-i) \quad (\text{II.41})$$

Nous disposons initialement de  $N$  échantillons de  $x(n)$  avec  $n = 0, 1, \dots, N-1$ , pour calculer les coefficients  $a(i)$  ( $i = 1, \dots, p$ ), on considère différentes valeurs  $n = p, p+1, \dots, N-1$

On obtient ainsi le système d'équations suivant :

$$e(p) = x(p) + \sum_{i=1}^p a(i)x(p-i)$$

$$e(p+1) = x(p+1) + \sum_{i=1}^p a(i)x(p+1-i)$$

**M**

$$e(N-1) = x(N-1) + \sum_{i=1}^p a(i)x(N-1-i)$$

Ce système peut s'écrire sous forme matricielle :

$$e = x - XA \quad (\text{II.42})$$

avec

$$e = [e(p) \ e(p+1) \ \mathbf{K} \ e(N-1)]^T \quad \text{Le vecteur erreur.}$$

$$A = [a(1) \ a(2) \ \mathbf{K} \ a(p)]^T \quad \text{Le vecteur des coefficients AR qu'on cherche.}$$

$$x = [x(p) \ x(p+1) \ \mathbf{K} \ x(N-1)]^T \quad \text{Un vecteur d'échantillons de } x(n) \text{ .}$$

et  $X$  une matrice telle que :

$$X = - \begin{pmatrix} x(p-1) & x(p-2) & \mathbf{K} & x(0) \\ x(p) & x(p-1) & \mathbf{K} & x(1) \\ \mathbf{M} & \mathbf{M} & \mathbf{K} & \mathbf{M} \\ x(N-2) & x(N-3) & \mathbf{K} & x(N-1-p) \end{pmatrix} \quad (\text{II.43})$$

L'estimation du vecteur  $A$  consiste à minimiser l'erreur quadratique  $e^T e$ . Cela revient à résoudre :

$$\frac{\partial e^T e}{\partial A} = 0 \quad (\text{II.44})$$

ou à résoudre le système suivant :

$$X^T X A = X^T x \quad (\text{II.45})$$

Par l'une des méthodes de résolution des systèmes d'équations linéaires ou par inversion directe.

La solution dans ce cas est donnée par :

$$A = (X^T X)^{-1} X^T x \quad (\text{II.46})$$

avec

$$y = X^T X \quad (\text{II.47})$$

$$A = (X^T X)^{-1} X^T x \quad (\text{II.48})$$

## II.6 Ordre du modèle AR [24]

La modélisation AR nécessite la connaissance à priori de l'ordre  $p$  du modèle. Celui-ci peut être fixé arbitrairement, ou choisi de manière objective. Sa valeur conditionne la précision des coefficients ainsi que la qualité des résultats ultérieurs.

- Si l'ordre est trop faible, le modèle ne représentera pas les propriétés intrinsèques du modèle.
- S'il est trop élevé, le modèle représentera les propriétés qui sont dues aux bruits.

Plusieurs critères de sélection de l'ordre  $p$  du modèle AR ont été proposés. Ces critères sont généralement basés sur la variance de l'erreur de prédiction  $S_e^2(p)$  pour chaque valeur de  $p$ . Parmi ces critères, on peut citer :

- ▼ Critère FPE (Erreur de Prédiction Finale ou Final Prediction Error) :

$$FPE(p) = \frac{N+p}{N-p} S_e^2(p) \quad (\text{II.49})$$

où

$N$  représente le nombre d'échantillons du signal  $x(n)$ .

- ▼ Critère AIC (Akaike Information Criterion) :

$$AIC(p) = N \log(S_e^2(p)) + 2p \quad (\text{II.50})$$

C'est la loi la plus utilisée.

L'ordre  $p^*$  optimal est celui qui minimise l'un des deux critères.

## II.7 Modélisation AR monodimensionnelle d'image

La modélisation autorégressive considère l'image comme une séquence temporelle de niveaux de gris dont la valeur à tout instant dépend des valeurs précédemment rencontrées au cours d'un balayage ligne par ligne. Un tel modèle permet de prévoir la valeur du niveau de gris d'un point  $a(i, j)$  en fonction des niveaux de gris de ses voisins rencontrés au cours du balayage avant d'atteindre ce point.

### II.7.1 Ordre du Parcours

Le passage d'une image 2D  $x_{k1,k2}$  à une séquence  $x_k$  se fait généralement en concaténant les lignes ou les colonnes. C'est cette technique qu'on a utilisée jusqu'à présent. Cependant, d'autres parcours d'image sont possibles, notamment grâce aux courbes de remplissage d'espace SFC (Space Filling Curves) introduites par Giuseppe Peano en 1890.

Une courbe qui remplit l'espace est une courbe qui parcourt un espace multidimensionnel en passant, de point en point, et d'une manière unique par tous les points de cet espace. On s'intéressera en particulier à la différence topologique existant entre un espace multidimensionnel et sa représentation sous forme de courbe.

## II.7.2 Les différents types de courbes de remplissage d'espace

Le balayage d'une image 2D, suivant une courbe est un problème, car en parcourant l'image nous perdons une partie de son information spatiale et nous ajoutons du bruit. Pour cela plusieurs types de courbes ont été proposés à fin de balayer toute l'image et respecter la notion de voisinage.

- **Courbe de Zigzag scan**

Cette courbe respecte l'ordre canonique, mais des points voisins dans l'espace des composantes peuvent correspondre à des points très éloignés sur le parcours de la courbe, par conséquent, cette courbe présente peu d'intérêt du point de vue du traitement d'image.

- **Courbe de Regazzoni**

La structure en couches de cette courbe traduit la volonté de construire un ordre proche de celui que donnerait la distance. Cette courbe est utilisée beaucoup plus en filtrage.

- **Balayage Vidéo (lexicographique)**

On constate la très forte dissymétrie de cette courbe qui traduit le déséquilibre et la non préservation de la notion de voisinage.

- **Courbe entrelacement de bits**

Cette courbe se construit de manière récursive, ce qui lui confère une structure auto similaire. Notons que cette courbe est également utilisée dans l'espace 2D de l'image, notamment pour la construction quadtree lors de la décomposition d'une image en structure hiérarchique.

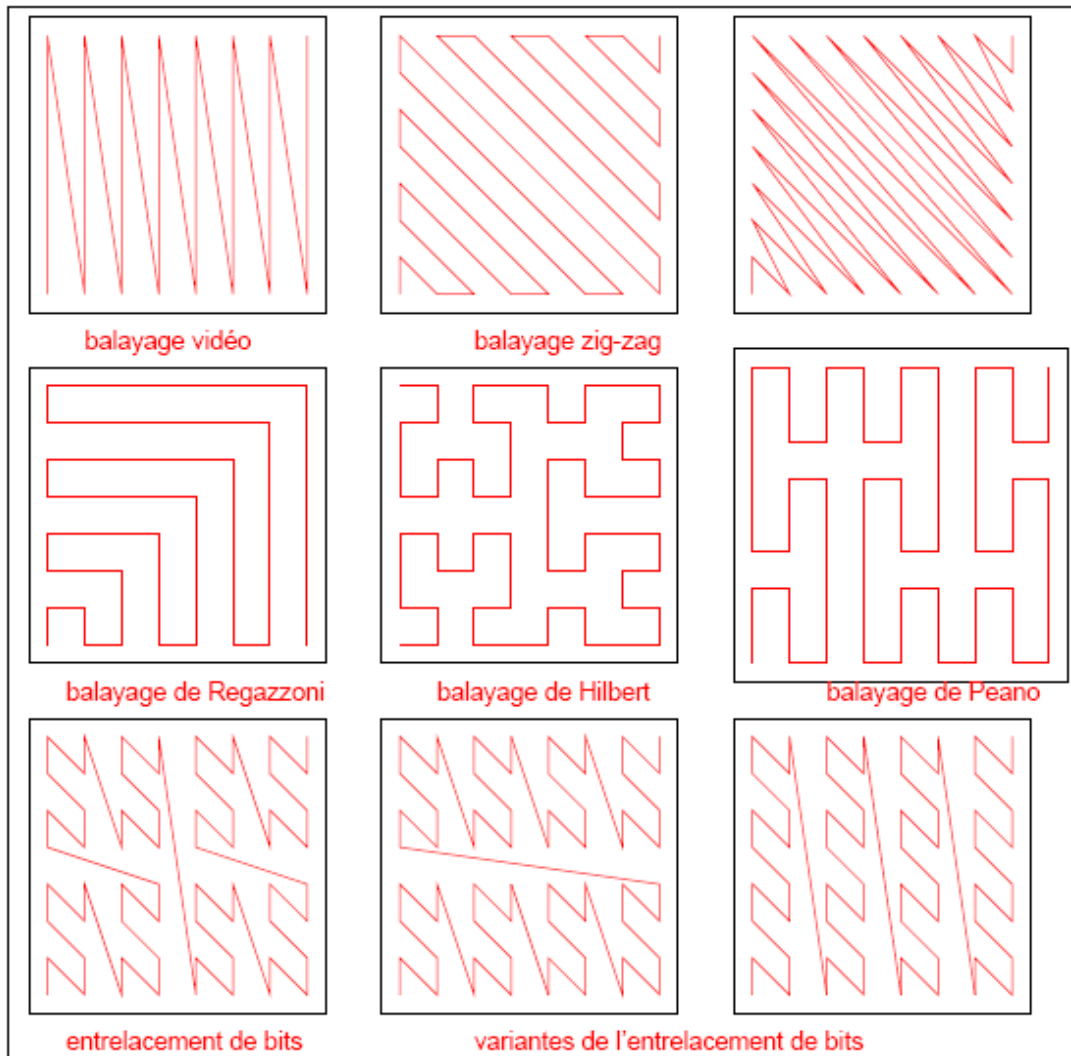
- **Courbe de Hilbert et de Peano**

Les courbes de Hilbert (souvent attribuées à Peano) et de Peano sont des SFC abondamment étudiées et souvent utilisées en traitement d'image pour permettre un parcours efficace du plan image.

Ces courbes présentent toutes les deux une propriété essentielle: le passage d'un point de la courbe à son voisin (sur la courbe) ne s'accompagne jamais d'un déplacement important dans l'espace des composantes, d'où la préservation de la topologie et d'information spatiale.

Il faut cependant noter que les ordres induits par ces courbes sont des ordres qui ne respectent pas l'ordre canonique.

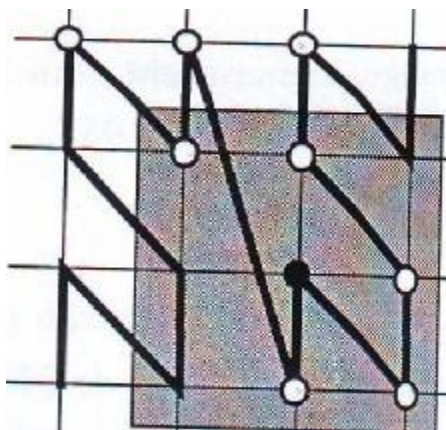
La figure ci-dessous résume les formes géométriques des différentes courbes de remplissage d'espace.



**Fig.II.7** : Représentation géométrique des SFC

Comme nous l'avons déjà évoqué, ces représentations scalaires s'accompagnent inévitablement de distorsions topologiques, c'est-à-dire la notion de voisinage est modifiée, et cela entraîne la perte de l'information spatiale et l'accroissement du bruit.

De manière à permettre une comparaison quantitative, un critère permettant la mesure de cette distorsion a été proposé. Ce critère est défini comme suit: étant donné le 8 voisinage d'un vecteur on calcule le pourcentage de ces voisins qui se retrouvent encore parmi les 8 voisins du pixel après codage sur la courbe, la figure (II.8) montre ce calcul.



**Fig.II.8** : critère de distorsion.

Le résultat final de cette étude est résumé dans le tableau suivant:

<b>Courbe</b>	<b>Taux de préservation du voisinage</b>
Hilbert	58%
Peano	57%
Entrelacement de bits	47%
Zigzag scan	27%
Regazzoni	26%
Lexicographique (vidéo)	25%

**Tableau II.1:** Taux de préservation topologique

## II.8 Modèle autorégressif bidimensionnel [26]

L'analyse des signaux aléatoires bidimensionnels est une extension directe de l'analyse des signaux monodimensionnels. Dans la première partie de ce chapitre nous avons vu la modélisation monodimensionnelle appliquée aux images en niveau de gris, et dans ce qui suit nous allons présenter quelques notions du modèle AR 2D.

Dans une image, les pixels sont sur une grille de points à 2D, il est alors plus naturel de baser le traitement sur une représentation bidimensionnelle, c'est-à-dire dans le domaine spatial. Cette représentation entraîne un choix sur l'ordonnement des différents pixels. Si, en effet la définition des notions de « passé », « présent » et « futur » qui caractérisent facilement la causalité ne présente aucune ambiguïté pour un signal monodimensionnel, ici, il y a lieu d'opérer un choix qui amène à définir plusieurs types de modèles qui auront une structure causale, semi causale et non causale.

### II.8.1 Définition

D'une manière générale un modèle autorégressif bidimensionnel est décrit par l'équation aux différences suivante :

$$x(i, j) = - \sum_{(m,n) \in D} a(m, n, i, j) x(i - m, j - n) + e(i, j) \quad (\text{II.51})$$

où

$e(i, j)$  : est l'entrée du modèle qui peut être un bruit blanc ou une séquence aléatoire de fonction de densité spectrale connue.

$a(m, n, i, j)$  : sont les coefficients du modèle.

$D$  : est la région de prédiction définie par l'ensemble des couples  $(m, n)$  qui dépendent du pixel  $(i, j)$ . L'ensemble  $D$  définit le type de causalité du modèle.

Par ailleurs la nature du champ d'entrée  $e(i, j)$  définit le type du modèle. Dans le cas où  $e(i, j)$  est un bruit blanc, le modèle (II.51) sera appelé modèle AR et dans le cas où  $e(i, j)$  est un bruit corrélé, le modèle sera alors un modèle ARMA. En outre si  $e(i, j)$  est une erreur de prédiction, le modèle (II.51) est appelé modèle de prédiction ou modèle à variance minimale.

### II.8.2 Causalité des systèmes bidimensionnels

Plusieurs types de supports finis  $D$  de prédiction causale peuvent être définis :

#### Ø Causalité quart de plan

La causalité quart de plan est la plus simple à envisager, car c'est une extension directe du concept de causalité dans le cas monodimensionnel, l'ensemble  $D$  est défini par :

$$D = \{m \geq 0, n \geq 0 \text{ et } m + n > 0\} \tag{II.52}$$

Pour un point de l'image  $x(i, j)$  on définit le passé et le futur comme cela est représenté sur la (figure II.9) l'ensemble  $D$  se trouve donc dans la zone « passé ».

#### Ø Causalité demi plan asymétrique

Cette définition de la causalité provient de la manière usuelle de lire et de stoker une image dans l'ordre lexicographique, c'est-à-dire dans le sens usuel de lecture, de gauche à droite et de haut en bas. L'ensemble  $D$  est défini par :

$$D = \{(m > 0, \forall n)\} \cup \{(m = 0, n > 0)\} \tag{II.53}$$

Ceci implique une autre définition du passé et du futur (figure II.10).

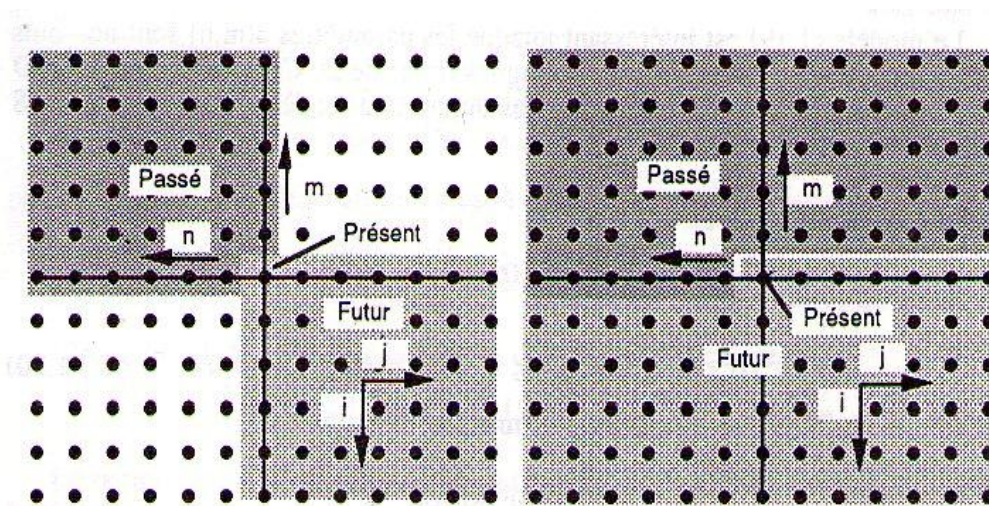


Fig.II.9 : Causalité quart de plan

Fig.II.10 : Causalité demi plan asymétrique

Ø Semi-causalité et non-causalité

Dans le cas bidimensionnel on peut se trouver dans la situation où le modèle est causal dans une direction et non causal dans l'autre, c'est ce qu'on appelle un modèle semi-causal (figure II.11), dans ce cas l'ensemble  $D$  est défini par:

$$D = \{(m > 0, \forall n)\} \cup \{(m = 0, n \neq 0)\} \tag{II.54}$$

Dans le cas non causal, le modèle n'est causal dans aucune direction (figure II.12). L'ensemble  $D$  est défini par :

$$D = \{\forall (m, n) \neq (0, 0)\} \tag{II.55}$$

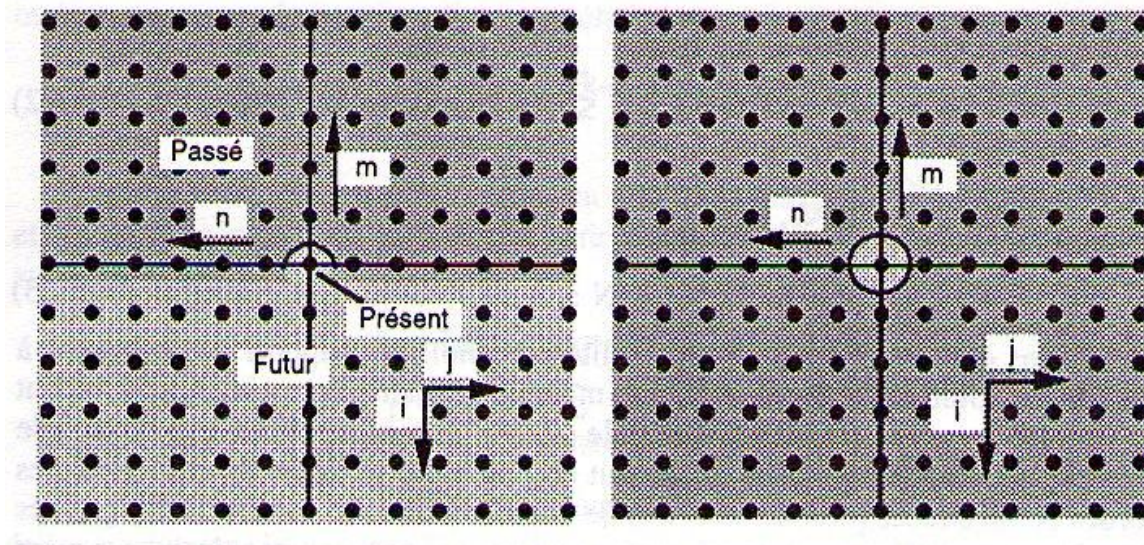


Fig.II.11 : Semi-causalité

Fig.II.12 : Non causalité

La causalité quart de plan paraît plus naturelle que la causalité demi plan asymétrique car c'est une extension directe du cas monodimensionnel, mais il est plus restrictif. La causalité demi-plan asymétrique est plus large et peut être obtenue à partir de la causalité monodimensionnelle par une simple transformation d'une grille de données en une séquence monodimensionnelle. Cette transformation est très importante car elle permet d'utiliser plusieurs résultats établis dans le cas monodimensionnel. Elle est définie de la manière suivante :

Soit  $x(i, j)$  un champ de données balayé ligne par ligne, à un point  $(i, j)$  on fait correspondre à un instant ou indice de position noté  $t$  tel que :

$$(i, j) \rightarrow t = i + jI \quad (\text{II.56})$$

avec

$$0 \leq i \leq I \quad \text{et} \quad 0 \leq j \leq J$$

$I$  : nombre de lignes

$J$  : nombre de colonnes.

### II.8.3 Modèles stationnaires

Dans le cas stationnaire le modèle (II.51) s'écrit :

$$x(i, j) = - \sum_{(m, n) \in S} a(m, n)x(i - m, j - n) + e(i, j) \quad (\text{II.57})$$

Les coefficients  $a(m, n)$  ne dépendent plus de la position du pixel  $(i, j)$  de l'image.

Cette hypothèse n'est souvent pas réalisable pour un nombre trop important de points de l'image. Dans de telles situations on fait un découpage de l'image en zones arbitraires et, sur chacune d'elles, on suppose que la stationnarité est satisfaite et par conséquent on estimera les paramètres du modèle AR pour chaque zone.

Le modèle (II.57) est intéressant lorsque les paramètres  $a(m, n)$  sont non nuls uniquement pour un nombre fini de couples  $(m, n)$  de  $D$ .

Ce sous ensemble de  $D$  noté  $S$  est appelé fenêtre de prédiction ou support du modèle.

La dimension de  $S$  définit l'ordre du modèle.

#### Exemples de supports

- Modèle causal quart de plan

$$S_0 = \{0 \leq m \leq M, 0 \leq n \leq N, (m, n) \neq (0, 0)\} \quad (\text{II.58})$$

Le couple  $(M, N)$  est appelé ordre du modèle.

- Modèle causal demi plan asymétrique.

$$S_1 = \{m = 0, 1 \leq n \leq N\} \cup \{1 \leq m \leq M, -N \leq n \leq N\} \quad (\text{II.59})$$

- Modèle semi-causal

$$S_2 = \{0 \leq m \leq M, -N \leq n \leq N, (m, n) \neq (0, 0)\} \quad (\text{II.60})$$

- Modèle non causal

$$S_3 = \{-M \leq m \leq M, -N \leq n \leq N, (m, n) \neq (0, 0)\} \quad (\text{II.61})$$

## II.9 Estimation des paramètres

Etant donné l'équation du modèle (II.57), pour un support  $S$  fixé on cherche à estimer les paramètres  $a(m, n)$  du modèle.

Pour cela considérons le cas du modèle à variance minimale ou du modèle de prédiction, c'est-à-dire le cas où  $e(i, j)$  est une erreur de prédiction. Le champ de données bidimensionnel  $x(i, j)$  est supposé aléatoire, gaussien, stationnaire, de moyenne nulle et de variance finie.

Les paramètres  $a(m, n)$  peuvent être calculés en minimisant la variance de l'erreur de prédiction  $e(i, j)$  :

$$E[e(i, j)^2] = E[(x(i, j) + \sum_{(m, n) \in S} a(m, n)x(i-m, j-n))^2] \quad (\text{II.62})$$

Par rapport aux paramètres  $a(m, n)$ , c'est-à-dire

$$\frac{\partial E[e(i, j)^2]}{\partial a(m, n)} = 0, \quad \forall (m, n) \in S \quad (\text{II.63})$$

L'équation (II.63) implique la relation d'orthogonalité suivante :

$$E[e(i, j)x(i - k, j - l)] = 0, \forall (k, l) \in S \quad (\text{II.64})$$

En remplaçant  $e(i, j)$  par sa valeur, on obtient le système d'équations normales habituelles :

$$E\left[\left(x(i, j) + \sum_{(m,n)} a(m, n)x(i - m, j - n)\right)x(i - k, j - l)\right] = 0, \forall (k, l) \in S \quad (\text{II.65})$$

Ou bien en termes de fonction d'autocorrélation :

$$r(k, l) + \sum_{(m,n) \in S} a(m, n)r(k - m, l - n) = 0, \quad \forall (k, l) \in S \quad (\text{II.66})$$

Les paramètres  $a(m, n)$  sont obtenus en résolvant le système d'équation (II.66) et la variance minimale de l'erreur sera alors calculée de la relation suivante :

$$s_e^2 = E[e(i, j)^2] = E[e(i, j)x(i, j)] = r(0, 0) + \sum_{(m,n) \in S} a(m, n)r(m, n) \quad (\text{II.67})$$

Le système d'équation (II.66) et l'équation (II.67) peuvent être regroupés sous la forme condensée suivante:

$$r(k, l) + \sum_{(m,n) \in S} a(m, n)r(k - m, l - n) = s_e^2 d(k, l), \quad \forall (k, l) \in S \quad (\text{II.68})$$

Suivant le choix de  $S$ , c'est-à-dire de la nature de prédicteur, le système linéaire (II.68) peut être résolu en utilisant différents algorithmes. Par ailleurs les propriétés du champ d'entrée  $e(i, j)$  sont aussi liées à la nature de la causalité. Pour le montrer calculons la fonction d'autocorrélation de  $e(i, j)$  pour les différentes géométries du support  $S$ , dans le cas d'un modèle à variance minimale, c'est-à-dire  $e(i, j)$  est une erreur de prédiction.

La fonction d'autocorrélation de l'entrée  $e(i, j)$  est donnée par :

$$r_e(k, l) = E[e(i, j)e(i - k, j - l)] \quad (\text{II.69})$$

$$r_e(k, l) = E[e(i, j) \{x(i - k, j - l) + \sum_{(m,n) \in S} a(m, n)x(i - k - m, j - l - n)\}] \quad \forall (k, l) \in S \quad (\text{II.70})$$

ou encore,

$$r_e(k, l) = E[e(i, j)x(i - k, j - l)] + \sum_{(m,n) \in S} a(m, n)E[e(i, j)x(i - k - m, j - l - n)] \quad \forall (k, l) \in S \quad (\text{II.71})$$

En utilisant la relation d'orthogonalité (II.67) on obtient les résultats suivants :

#### Modèle causale

$$r_e(k, l) = \sigma_e^2, \quad (k, l) = (0, 0) \quad (\text{II.72})$$

$$r_e(k, l) = 0, \quad \text{ailleurs} \quad (\text{II.73})$$

qui est la fonction d'autocorrélation d'un bruit blanc.

### II.9.1 Ecriture matricielle des équations normales

Le système d'équations linéaires (II.68) peut être écrit sous la forme matricielle suivante :

$$R_{MN} a = \sigma_e^2 \mathbf{1}_{MN} \quad (\text{II.74})$$

Où  $R_{MN}$  est la matrice d'autocorrélation de l'image,  $a$  est le vecteur paramètres et  $\mathbf{1}_{MN}$  est le vecteur unitaire c'est-à-dire un vecteur de dimension  $(M \times N)$  dont toutes les composantes sont nulles sauf une qui est égale à un. Dans le cas stationnaire, la matrice  $R_{MN}$  est une matrice symétrique de Toeplitz par bloc appelée aussi matrice Toeplitz bloc Toeplitz, (Akaike [1977]). Elle est définie par :

$$R_{MN} = \begin{pmatrix} R_0 & R_{-1} & \mathbf{K} & R_{-2N} \\ R_1 & & & \\ & & & R_{-1} \\ R_{2N} & \mathbf{K} & R_1 & R_0 \end{pmatrix} \quad (\text{II.75})$$

Les blocs  $R_K$  sont aussi constitués par des matrices de Toeplitz définies par :

$$R_K = \begin{pmatrix} r(0, k) & r(-1, k) & \mathbf{K} & r(-2M, k) \\ r(1, k) & & & \\ & & & r(-1, k) \\ r(2M, k) & & r(1, k) & r(0, k) \end{pmatrix} = R_{-k} \quad (\text{II.76})$$

Les dimensions et les valeurs des composantes des matrices  $R_{MN}$  et  $R_K$ , ainsi que celles des vecteurs  $\underline{a}$  et  $\underline{1}_{MN}$  sont fonction de la géométrie du support  $S$ , c'est-à-dire de la nature de la causalité du modèle.

### II.9.2 Cas d'un modèle causal

#### ü Causalité quart de plan

Dans le cas où le support est un quart de plan noté  $S_0$ , l'équation (II.68) s'écrit :

$$r(k, l) + \sum_{m=0}^M \sum_{n=0}^N a(m, n) r(k - m, l - n) = s_e^2 d(k, l), \quad \forall (k, l) \in S_0 \quad (\text{II.77})$$

Dans ce cas la matrice  $R_{MN}$  sera donnée par :

$$R_{MN} = \begin{pmatrix} R_0 & R_{-1} & \mathbf{K} & R_{-N} \\ R_1 & & & \\ & & & R_{-1} \\ R_N & \mathbf{K} & R_1 & R_0 \end{pmatrix} \quad (\text{II.78})$$

où les matrices  $R_K$  sont définies par (II.76) avec la dimension  $(M+1)(M+1)$ , le vecteur paramètres  $\underline{a}$  est donné par :

$$\underline{a}^T = [\underline{a}_0^T \ \underline{a}_1^T \ \mathbf{K} \ \underline{a}_n^T \ \mathbf{K} \ \underline{a}_N^T] \quad (\text{II.79})$$

où

$$\underline{a}_0^T = [1 - a(1,0)\mathbf{K} - a(m,0)\mathbf{K} - a(M,0)], \quad a(0,0) = 1 \quad (\text{II.80})$$

$$\underline{a}_n^T = [-a(0,n)\mathbf{K} - a(m,n)\mathbf{K} - a(M,n)] \quad (\text{II.81})$$

et le vecteur  $\underline{1}_{MN}$  est donné par :

$$\underline{1}_{MN}^T = [\underline{1}_M^T \ \underline{0}^T \ \mathbf{K} \ \underline{0}^T \ \mathbf{K} \ \underline{0}^T] \quad (\text{II.82})$$

### ü Causalité quart de plan avec les autres quadrants

On a établi les équations normales en supposant que le support causal quart de plan est celui de premier quadrant. Il est possible d'établir des relations similaires pour les autres quadrants qui sont illustrés sur la figure (II.11). Si on définit les coefficients  $a(m,n)$  avec les supports du deuxième, du troisième et du quatrième quadrant, on aura :

**2<sup>ème</sup> Quadrant :**

$$a(m,n) = a^{(2)}(m,n), \quad \{(0 \leq m \leq M, -N \leq n \leq 0, (m,n) \neq (0,0))\} \quad (\text{II.83})$$

**3<sup>ème</sup> Quadrant**

$$a(m,n) = a^{(3)}(m,n), \quad \{(-M \leq m \leq 0, -N \leq n \leq 0, (m,n) \neq (0,0))\} \quad (\text{II.84})$$

**4<sup>ème</sup> Quadrant:**

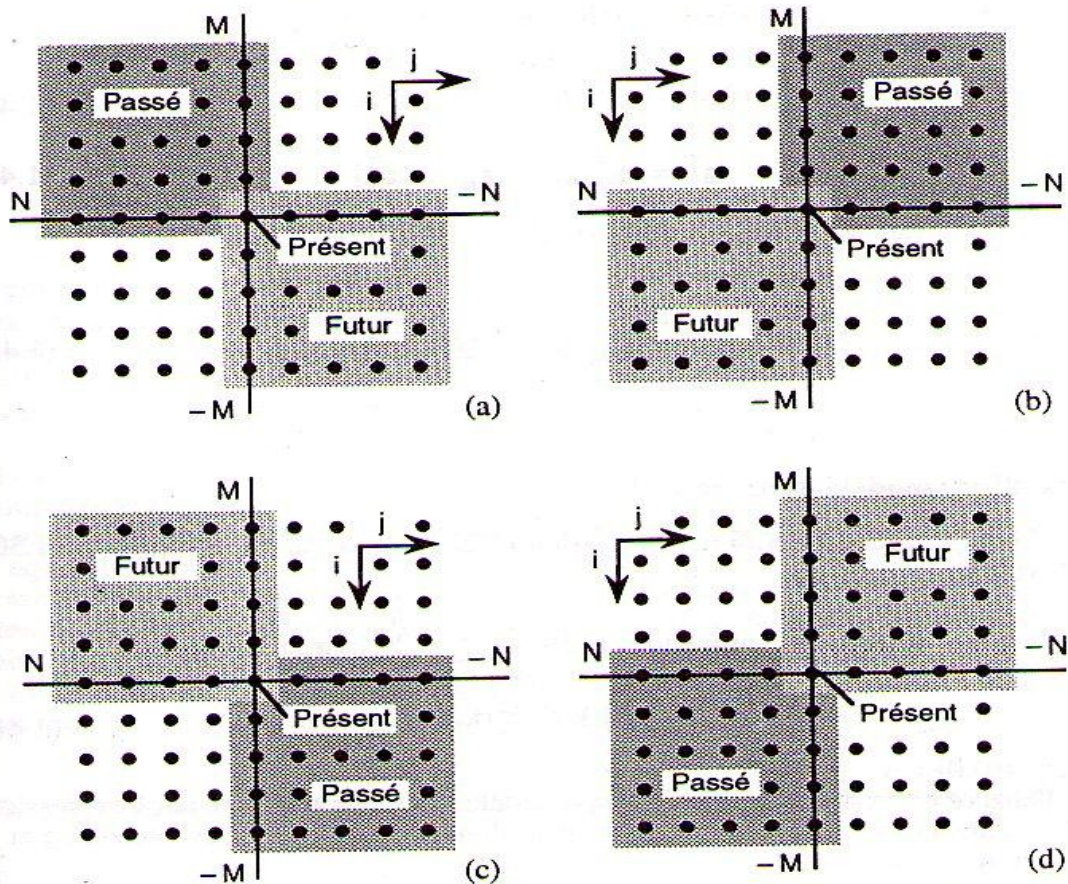
$$a(m,n) = a^{(4)}(m,n), \quad \{(-M \leq m \leq 0, 0 \leq n \leq N, (m,n) \neq (0,0))\} \quad (\text{II.85})$$

Les différents supports  $S_0$  de la causalité quart de plan dans les quatre quadrants sont illustrés par la (figure II.13) En suivant les étapes précédentes pour établir les équations normales relatives aux coefficients  $a(m,n)$  dans le premier quadrant, on déduit des équations normales des trois autres quadrants les relations suivantes :

$$a^{(1)}(m,n) = a^{(3)}(m,n) \tag{II.86}$$

et

$$a^{(2)}(m,n) = a^{(4)}(m,n) \tag{II.87}$$



**Fig .II.13 :** Causalité quart de plan dans chacun des quatre quadrants (a)1<sup>er</sup> quadrant, (b) 2<sup>ème</sup> quadrant, (c) 3<sup>ème</sup> quadrant, (d) 4<sup>ème</sup> quadrant.

• **Causalité demi-plan asymétrique**

Dans ce cas le support est défini par  $S_1$ . L'équation (II.68) peut s'écrire :

$$r(k,l) + \sum_{m=1}^M a(m,0)r(k-m,l) - \sum_{m=M}^M \sum_{n=1}^N a(m,n)r(k-m,l-n) = s_e^2 d(k,l), \forall (k,l) \in S_1 \quad (\text{II.88})$$

Dans ce cas la matrice  $R_{MN}$  sera donnée par :

$$R_{MN} = \begin{bmatrix} R'_0 & R'_{-1} & & R'_{-N} \\ R'_1 & R_0 & R_{-1} & R_{-N+1} \\ & & & \\ & R_1 & & R_{-1} \\ R'_N & R_{N-1} & R_1 & R_0 \end{bmatrix} \quad (\text{II.89})$$

où les matrices  $R_k$  sont définies par (II.76) et  $R'_k$  sont définies par la matrice suivante :

$$R'_k = \begin{bmatrix} r(M,k) & r(M-l,k) & r(-M,k) \\ r(2M,k) & r(2M-l,k) & r(0,k) \end{bmatrix} = R'_{-k} \quad (\text{II.90})$$

$$\underline{a}^T = [\underline{a}_0^T, \underline{a}_1^T, \mathbf{K}, \underline{a}_n^T \mathbf{K} \underline{a}_N^T] \quad (\text{II.91})$$

où

$$\underline{a}_0^T = [1 - a(1,0) \mathbf{K} - a(m,0) \mathbf{K} - a(M,0)] \quad , a(0,0) = 1 \quad (\text{II.92})$$

$$\underline{a}_n^T = [-a(-M,n) \mathbf{K} - a(0,n) \mathbf{K} - a(M,n)] \quad (\text{II.93})$$

et le vecteur  $\underline{1}_{MN}^T$  est donné par :

$$\underline{1}_{MN}^T = [\underline{1}_M^T \underline{0}^T, \mathbf{K} \underline{0}^T \mathbf{K} \underline{0}^T] \quad (\text{II.94})$$

$$\underline{1}_{MN}^T = [1 \ 0 \mathbf{K} \ 0 \mathbf{K} \ 0] \quad (\text{II.95})$$

## II.10 Conclusion

Dans ce chapitre nous avons donné un aperçu sur quelques concepts mathématiques de base concernant la modélisation d'image en niveaux de gris. Nous avons décrit les principales représentations des images sous forme d'un processus monodimensionnel ou sous forme d'un processus bidimensionnel. Dans le cas où l'image est considérée comme un processus 1D, celle-ci peut être modélisée par un modèle AR dont les coefficients peuvent être considérés comme des attributs de textures. Ils peuvent être estimés par plusieurs méthodes (Levinson-Durbin, Burg, Covariance, Moindres carrés). Dans le 2eme cas l'image peut être modélisée par un modèle AR 2D dont les coefficients sont considérés comme des attributs de texture et dépendant du support du modèle. Nous avons défini les différents types de causalité et donné les principales étapes pour la solution des équations normales.

# *Chapitre III*

---

## *Tests et résultats*

### III.1 Introduction

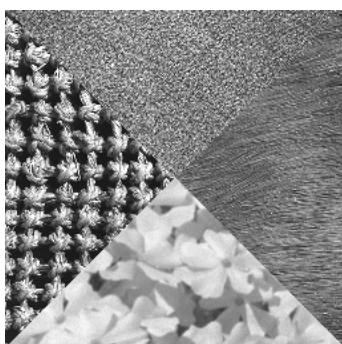
Nous allons présenter dans ce chapitre les résultats de la segmentation d'images texturées basée sur les modèles autorégressifs.

La programmation des différentes procédures a été mise en œuvre en utilisant le langage Matlab 7.5. Le matériel utilisé est un PC portable équipé d'un système d'exploitation: Windows Vista Edition familiale Premium. Les performances sont :

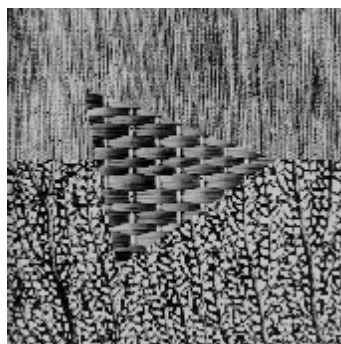
- Ø Un processeur : Intel®Pentium®processor T4200 (2.0GHz, 800MHz FSB).
- Ø Une mémoire vive de 2GB DDR3.

Afin d'évaluer les performances des techniques développées, les résultats sont illustrés sur deux sortes d'images :

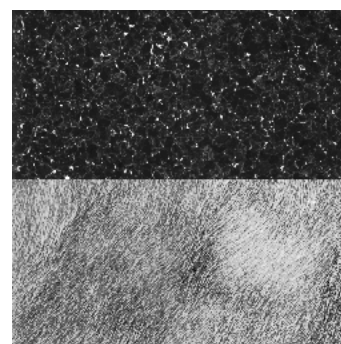
Trois images synthétiques test1, test2 et test3 tirées de l'album de Brodatz de taille 256\*256 pixels composées respectivement de 4, 3 et 2 régions texturées codées sur 8 bits.



**Test1**



**Test2**

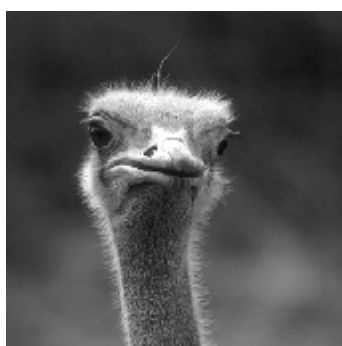


**Test3**

Trois images réelles test4, test5 et test6 composées de différentes régions uniformes et texturées. L'image test4 est de taille 282\*282 et les images test5 et test6 sont de taille 312\*312 pixels.



**Test4**



**Test5**



**Test6**

La démarche à suivre pour segmenter une image est la même quelque soit la méthode d'extraction d'attributs de la texture. Elle est résumée par l'algorithme ci-dessous.

**Etape 1 : Initialisation**

- 1- Fixer la taille de la fenêtre de voisinage ( $w$ ).
- 2- Fixer le nombre de classes de la texture ( $N_c$ ).
- 3- Fixer l'ordre  $P$  du modèle AR

**Etape 2 : Estimation des attributs de texture**

Calculer les attributs de la texture pour chaque pixel en utilisant une des méthodes d'estimation des coefficients AR vues dans le chapitre précédent à savoir l'algorithme de Levinson Durbin, l'algorithme de Burg, la méthode de covariance et celle des moindres carrés. Le calcul des attributs de chaque pixel par ces méthodes d'estimation est basé sur une séquence monovariante composée des niveaux de gris des pixels appartenant à la fenêtre de voisinage de taille  $(2W+1)^2$  centrée sur ce pixel. La transformation d'une partie de l'image en une séquence monovariante est effectuée suivant deux types de parcours (vidéo et Hilbert). On note par  $t_{ij} = (t_{i1}, t_{i2}, \dots, t_{ip})^t$  le vecteur caractéristique du pixel  $\langle i \rangle$ ,  $t_{ij}$  constitue alors l'attribut  $\langle j \rangle$  du pixel  $\langle i \rangle$ , avec  $j = 1, \dots, p$ . Ces attributs correspondent aux  $p$  coefficients AR.

**Etape 3 : Initialisation des centres de classes**

Initialiser au hasard ou manuellement les  $N_c$  centres de gravité  $[m_1, m_2, m_3, \mathbf{K}, m_{N_c}]$  correspondant aux  $N_c$  classes.

**Etape 4 : Affectation**

1- affecter chaque pixel  $\langle i \rangle$  à une classe  $C_k$ , dont le centre est  $m_k$ .

Un pixel  $\langle i \rangle$  appartient à la classe  $C_k$  de centre  $m_k$  si et seulement si  $\|t_i - m_k\|$  est minimale pour tout  $k = 1, \mathbf{K}, N_c$ .

2- mettre à jour la position du centre de gravité  $m_k$  de la classe  $C_k$ .

$$m_{kj} = \frac{1}{N_k} \sum_{i \in C_k} t_{ij}$$

où  $N_k$  est le nombre de pixels de la classe  $C_k$ .

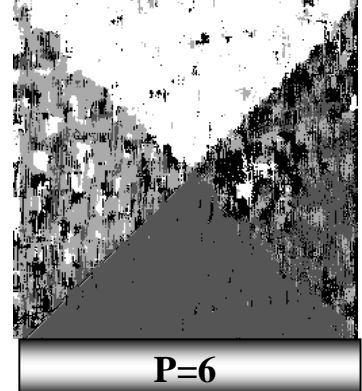
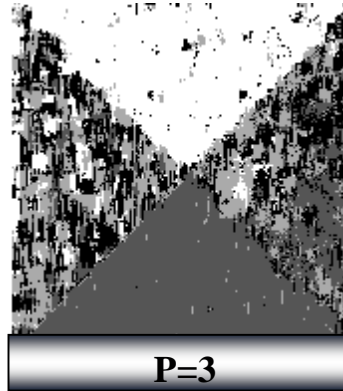
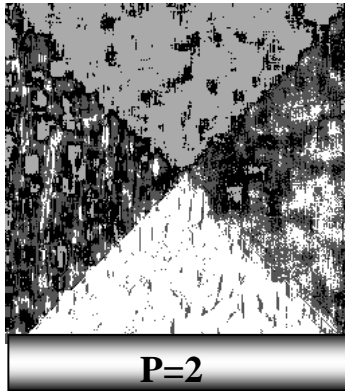
Les résultats de cette méthode dépendent principalement de la taille de la fenêtre de voisinage ( $W$ ), de l'ordre  $P$  du modèle AR donc des attributs de texture correspondant aux coefficients AR, ainsi que de la manière de déterminer ces coefficients. Pour les images synthétiques le nombre de classes  $N_c$  est fixé au nombre de textures correspondantes tandis que pour les images réelles le  $N_c$  est fixé à 3 pour test4 et test5, 2 pour test6. Nous allons donc analyser les différents résultats de la segmentation en fonction des paramètres  $W$ ,  $P$ , de la méthode d'estimation des coefficients AR et du type de balayage. Pour cela, nous allons baser notre analyse sur l'erreur de prédiction moyenne  $e^2$  et le temps de calcul.

### III.2 Présentation des résultats

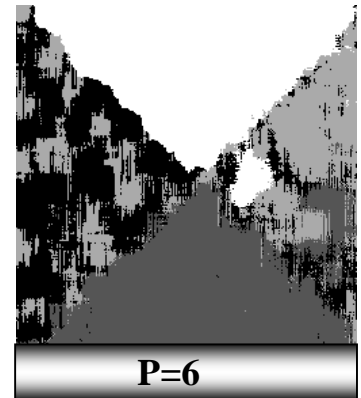
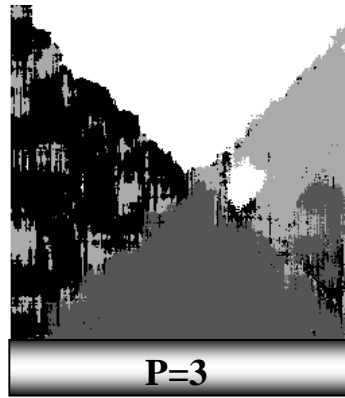
Afin d'analyser les performances de l'approche proposée, nous avons utilisé l'image test1 pour conduire tous nos tests.

Les figures (III.1) à (III.8) montrent les résultats de la segmentation de l'image synthétique (Test1) à base des méthodes de Levinson Durbin, Burg, Covariance et des Moindres carrés en faisant varier la taille de la fenêtre( $W$ ) et de l'ordre du modèle( $P$ )le balayage vidéo et Hilbert.

**W=4**



**W=8**

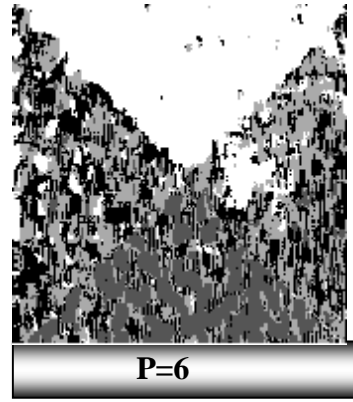
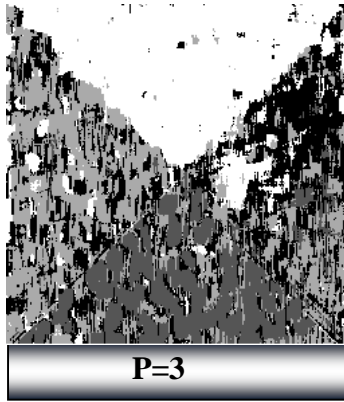
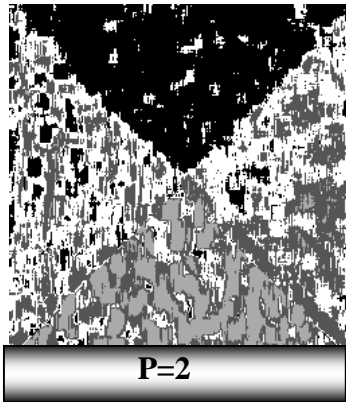


**W=16**

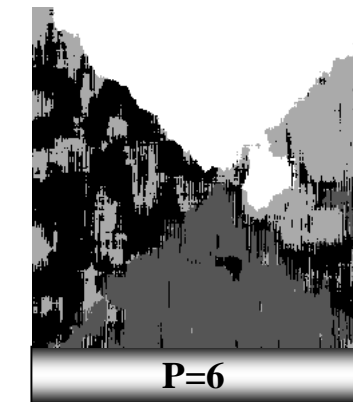
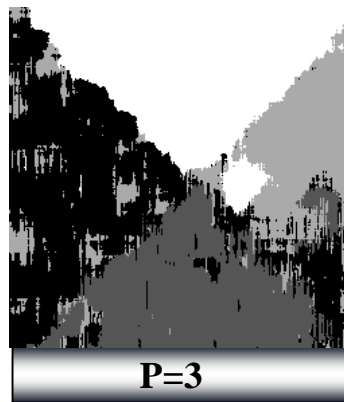


**Fig.III.1** : Segmentation de l'image Test1 par la méthode de Levinson Durbin et selon le balayage vidéo

W=4



W=8



W=16

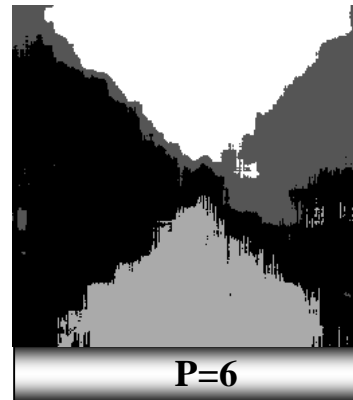
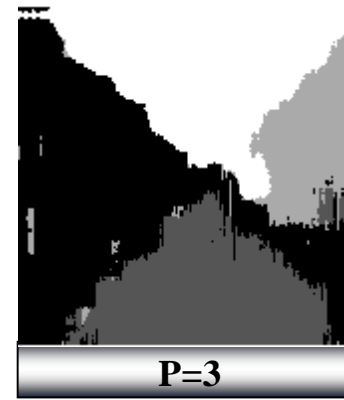
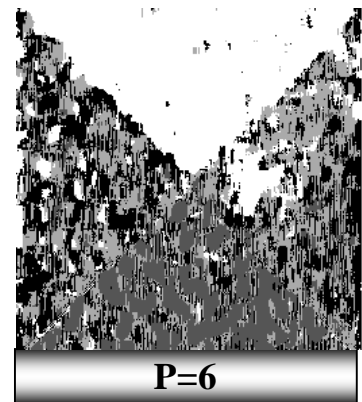
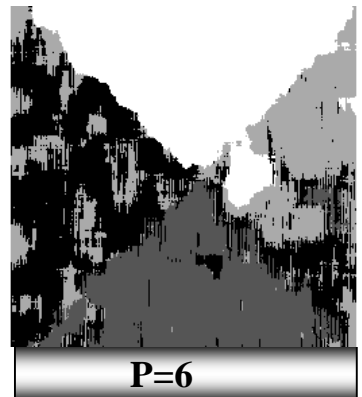
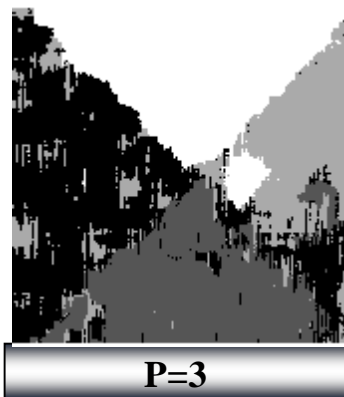
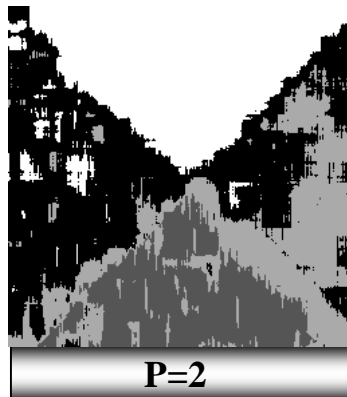


Fig.III.2 : Segmentation de l'image Test1 par la méthode de Burg et selon le balayage vidéo

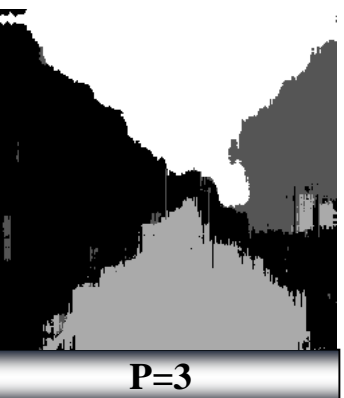
*W=4*



*W=8*

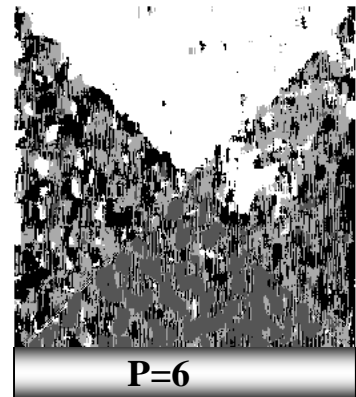
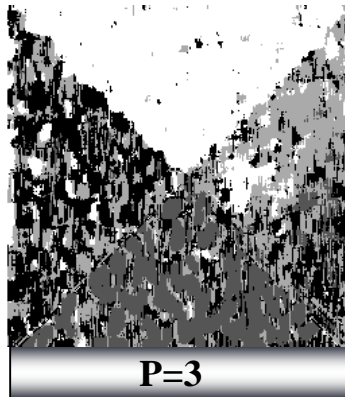
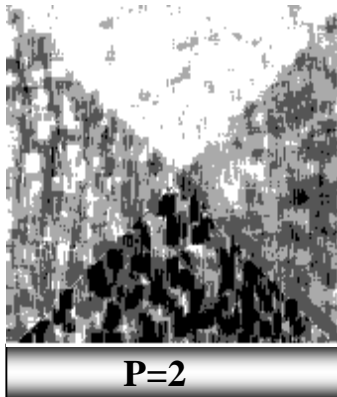


*W=16*

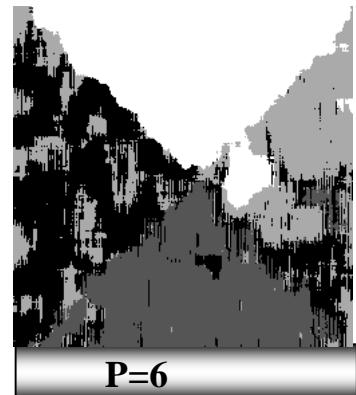
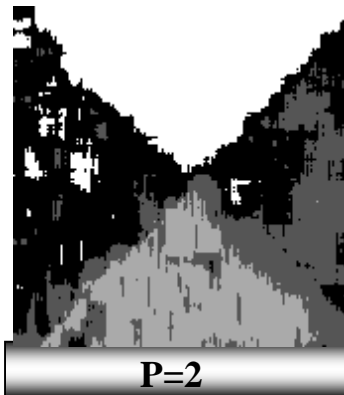


**Fig.III.3 :** Segmentation de l'image Test1 par la méthode de covariance et selon le balayage vidéo

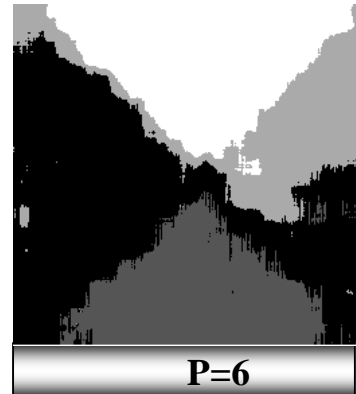
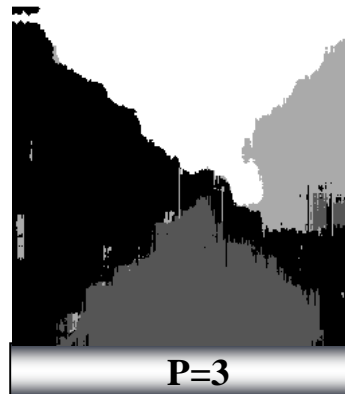
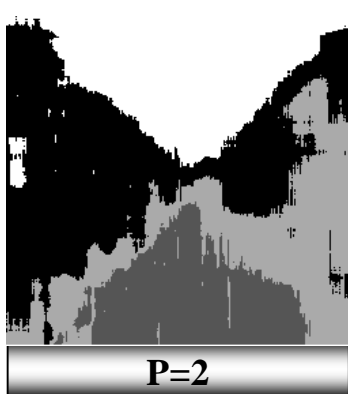
**W=4**



**W=8**

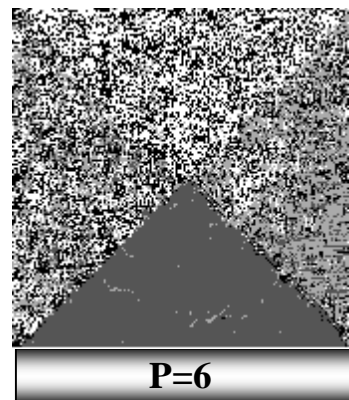
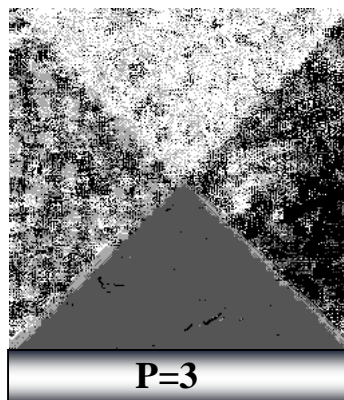
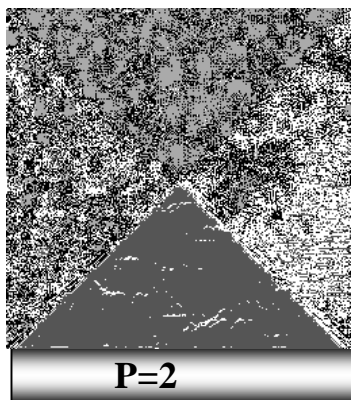


**W=16**

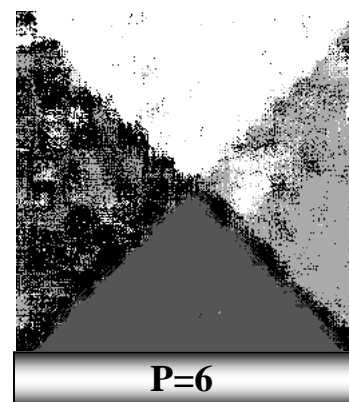
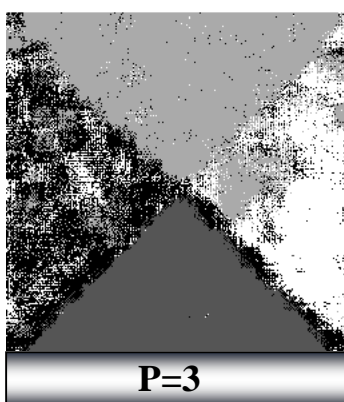
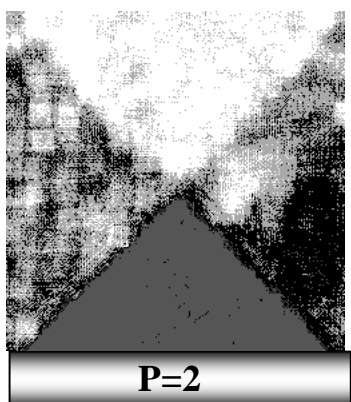


**Fig.III.4** : Segmentation de l'image Test1 par la méthode des moindres carrés et selon le balayage vidéo

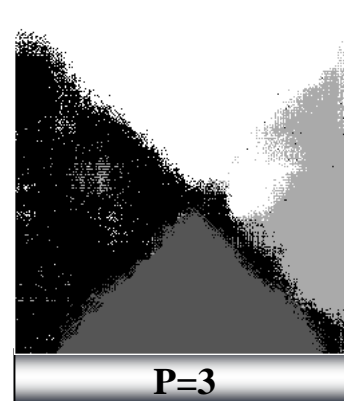
**W=4**



**W=8**

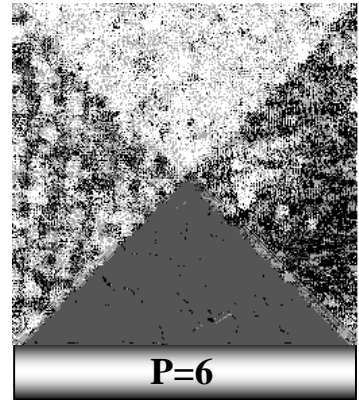
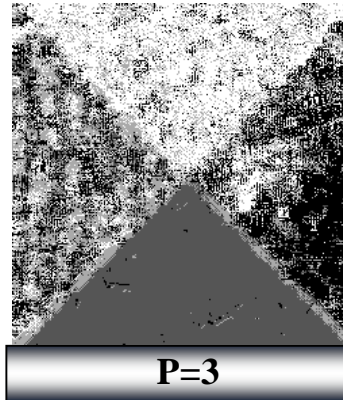
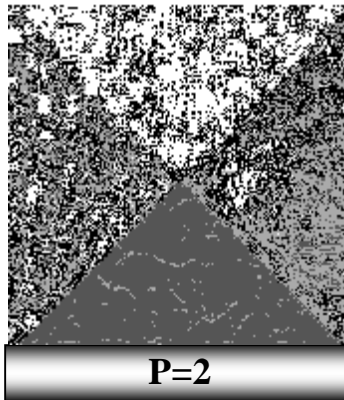


**W=16**

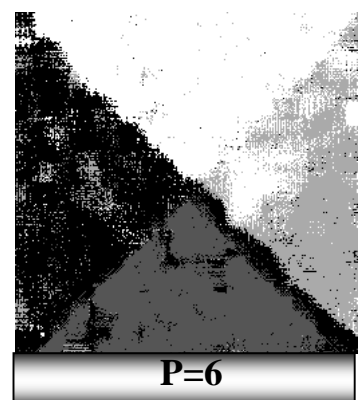
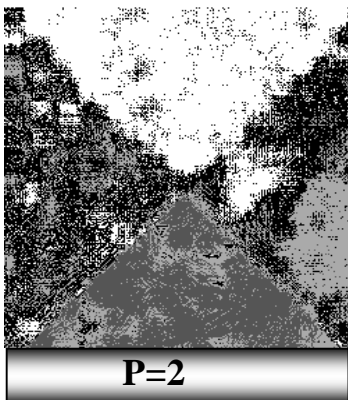


**Fig.III.5** : Segmentation de l'image Test1 par la méthode de Levinson Durbin et selon le balayage Hilbert

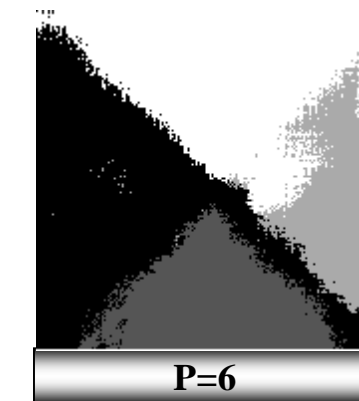
**W=4**



**W=8**

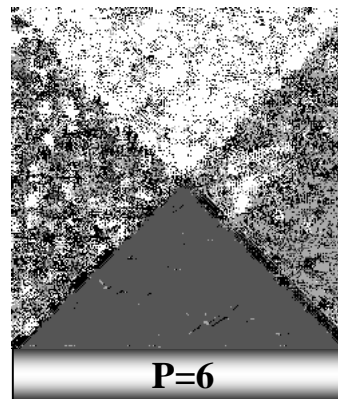
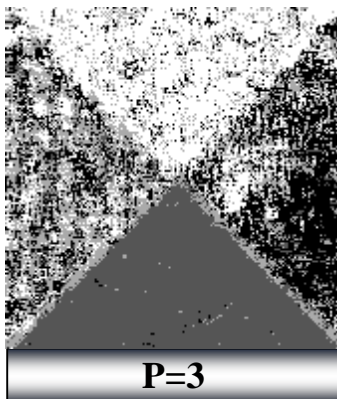
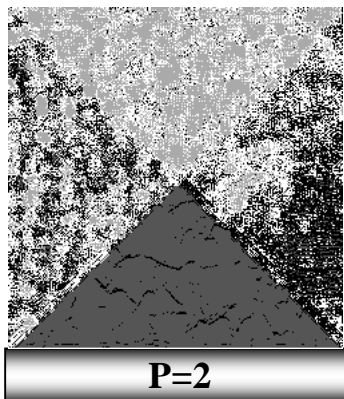


**W=16**



**Fig.III.6** : Segmentation de l'image Test1 par la méthode de Burg et selon le balayage Hilbert

**W=4**



**W=8**

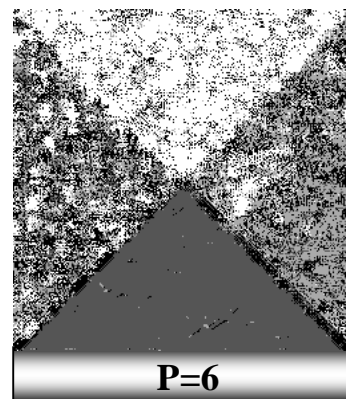
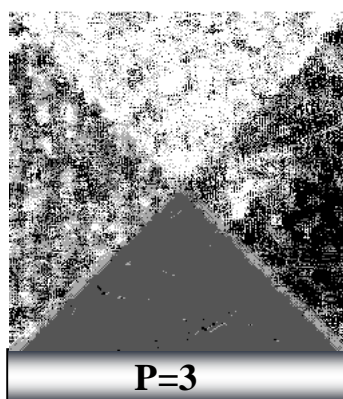
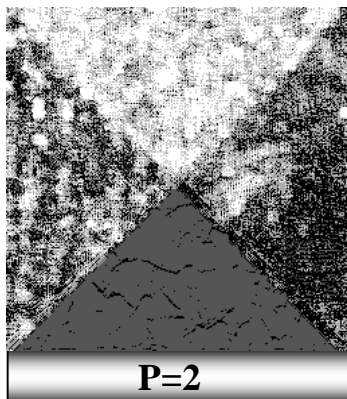


**W=16**



**Fig.III.7** : Segmentation de l'image Test1 par la méthode de covariance et selon le balayage Hilbert

**W=4**



**W=8**




**W=16**




**Fig.III.8** : Segmentation de l'image Test1 par la méthode des moindres carrés et selon le balayage Hilbert.

Les tableaux (III.1) et (III.2) donnent les différents temps d'exécution obtenus par les méthodes d'analyse de la texture selon les deux types de balayage et en fonction de la taille de la fenêtre de voisinage  $W$  et du nombre d'attributs  $P$  sur l'image test 1.

		<i>Temps d'exécution (secondes)</i> 			
<i>Taille de la fenêtre (W)</i>	<i>L'ordre du modèle (P)</i>	<i>Levinson</i>	<i>Covariance</i>	<i>Burg</i>	<i>Moindres carrés</i>
<b>5</b>	2	183,7099	157,2819	147,2121	130,0182
	3	257,9238	233,9695	229,8084	211,7617
	4	388,3928	311,3340	317,7018	291,6111
	5	413,2030	388,5594	396,9916	370,4068
	6	487,8742	463,359	480,7684	451,3283
<b>6</b>	2	98,7669	75,0258	74,4808	66,7752
	3	124,9961	105,8885	104,9026	95,8442
	4	165,9550	146,3709	149,7497	138,4397
	5	205,9297	189,0060	194,4955	182,6246
	6	246,8263	231,6933	238,1090	223,6607
<b>7</b>	2	188,0710	156,2598	148,4803	132,2828
	3	264,6130	237,2154	108,0665	99,2463
	4	166,7823	149,8797	151,5820	139,6526
	5	207,3021	191,0411	196,4211	182,8130
	6	244,7328	227,8469	236,0462	224,3240
<b>8</b>	2	107,5096	79,2280	73,0708	66,8145
	3	141,9043	112,6243	110,8682	103,3165
	4	182,3408	154,6042	155,0070	143,6446
	5	220,9916	193,1893	199,2897	185,7590
	6	259,1581	236,9154	241,1700	228,2425
<b>9</b>	2	111,5679	80,5206	76,4569	68,8543
	3	142,6816	114,0295	116,5856	103,7150
	4	183,0609	156,7748	157,4250	146,1149
	5	222,2980	202,4061	193,8258	193,9142
	6	255,5736	234,1572	239,7648	225,0916
<b>10</b>	2	107,5546	82,1932	72,2857	69,7634
	3	139,7408	115,5161	109,7335	102,7523
	4	179,5706	156,4314	153,2944	143,0560
	5	216,8288	196,4870	196,5092	187,7499
	6	255,0423	236,9020	239,7944	227,3393
<b>11</b>	2	143,7259	80,6971	73,7322	66,3566
	3	176,2603	119,1042	110,3966	103,2532
	4	214,0985	162,0980	152,6281	145,9269
	5	239,0145	193,3117	193,1543	186,1150
	6	289,7864	236,9569	239,1144	230,3802

<b>13</b>	2	137,1382	74,5654	69,7739	63,0182
	3	167,7977	120,4864	109,4507	107,3977
	4	198,5965	167,6560	156,0099	142,6552
	5	238,2982	196,7866	198,9559	188,6510
	6	276,7057	237,7057	307,3464	230,3002
<b>15</b>	2	135,3559	80,0597	76,9225	67,4299
	3	165,3540	120,6161	122,9274	116,1312
	4	212,3420	158,8475	158,7935	148,4732
	5	247,7994	204,1889	207,0010	200,5141
	6	288,6939	247,4401	248,8324	249,8111
<b>18</b>	2	181,7620	98,9927	83,3675	80,4164
	3	226,2587	139,1323	121,3813	119,2017
	4	263,0226	173,9978	170,1407	164,4777
	5	303,6137	232,6839	222,3378	215,1776
	6	314,9594	283,3499	266,7124	260,7406

**Tableau III.1** : Temps d'exécution en fonction de la taille de la fenêtre de voisinage, de l'ordre du modèle et de la méthode d'analyse de la texture.

		<b>Erreur de prédiction (<math>\epsilon \pm 003</math>)</b> 			
<i>Taille de la fenêtre (W)</i>	<i>L'ordre du modèle (P)</i>	<i>Levinson</i>	<i>Covariance</i>	<i>Burg</i>	<i>Moindres carrés</i>
<b>5</b>	2	2,3117	1,9579	1,9581	0,645
	3	2,2239	1,8445	1,8441	0,5187
	4	2,1767	1,7747	1,7739	0,4394
	5	2,1378	1,7117	1,7106	0,369
	6	2,11	1,664	1,6627	0,313
<b>6</b>	2	2,5417	2,2306	2,228	1,2169
	3	2,1126	1,8397	1,8398	0,7216
	4	2,0653	1,775	1,7751	0,6184
	5	2,0299	1,7218	1,7217	0,5331
	6	2,0048	1,6814	1,6813	0,465
<b>7</b>	2	2,1131	1,9218	1,9221	1,159
	3	2,0293	1,8242	1,8246	0,9489
	4	1,9824	1,764	1,7644	0,8201
	5	1,9474	1,7146	1,7151	0,7179
	6	1,9253	1,6803	1,6809	0,6369
<b>8</b>	2	2,0425	1,8931	1,8935	1,4563
	3	1,9617	1,8015	1,802	1,1954
	4	1,9182	1,7479	1,7485	1,0424
	5	1,8852	1,7038	1,7046	0,9213
	6	1,8637	1,6723	1,6732	0,8278

<b>9</b>	2	1,9871	1,8671	1,8674	1,7866
	3	1,9074	1,7787	1,7793	1,4658
	4	1,8634	1,7263	1,7271	1,2763
	5	1,8316	1,6856	1,6866	1,1342
	6	1,8127	1,6589	1,66	1,032
<b>10</b>	2	1,9551	1,8565	1,8568	2,1606
	3	1,8746	1,7688	1,7693	1,766
	4	1,8308	1,7181	1,7188	1,5335
	5	1,7994	1,6792	1,6802	1,3645
	6	1,7799	1,6529	1,6541	1,2435
<b>11</b>	2	1,9425	1,8601	1,8604	2,5955
	3	1,8614	1,773	1,7734	2,118
	4	1,8159	1,7214	1,7221	1,8277
	5	1,7829	1,682	1,6829	1,6173
	6	1,763	1,6564	1,6575	1,4703
<b>13</b>	2	1,9341	1,874	1,8742	3,605
	3	1,8557	1,7913	1,7917	2,9677
	4	1,8116	1,7428	1,7434	2,5702
	5	1,7775	1,704	1,7049	2,2644
	6	1,7556	1,6779	1,6789	2,0374
<b>15</b>	2	1,9248	1,879	1,8793	4,7574
	3	1,8477	1,7985	1,799	3,9393
	4	1,8038	1,7514	1,752	3,4347
	5	1,7707	1,7146	1,7155	3,0498
	6	1,7493	1,6899	1,6908	2,7559
<b>18</b>	2	1,8924	1,8601	1,8603	6,6578
	3	1,8179	1,7831	1,7836	5,5247
	4	1,7755	1,7384	1,7391	4,8451
	5	1,7438	1,7043	1,7051	4,3484
	6	1,7247	1,6828	1,6837	3,9838

**Tableau III.3 :** L'erreur de prédiction en fonction de la taille de la fenêtre de voisinage, de l'ordre du modèle et de la méthode d'analyse de la texture.

		<b>Temps d'exécution (seconde) <math>\bar{x}</math></b>			
<i>Taille de la fenêtre (W)</i>	<i>L'ordre du modèle (P)</i>	<i>Levinson</i>	<i>Covariance</i>	<i>Burg</i>	<i>Moindres carrés</i>
<b>4</b>	2	103,7708	77,1832	82,425	75,6662
	3	146,9874	122,5721	127,3846	116,0258
	4	180,1848	164,0355	171,6333	158,0575
	5	217,6059	201,4754	213,9069	192,2203
	6	251,2277	241,2067	253,4553	233,6766
<b>8</b>	2	111,7425	92,8306	88,3107	82,1969
	3	142,4495	125,0741	124,2556	115,7097
	4	177,4517	160,6747	166,3514	152,5943
	5	217,3985	207,8231	221,2384	200,0946
	6	261,4633	237,0683	242,1514	225,3419
<b>16</b>	2	293,2218	260,7414	255,1717	247,4326
	3	499,409	242,6589	218,1145	236,9176
	4	548,1638	354,1472	272,8126	309,8879
	5	364,2143	255,4722	249,7878	205,2884
	6	280,5641	260,0843	256,9146	240,533

**Tableau III.2 :** Temps d'exécution en fonction de la taille de la fenêtre de voisinage, de l'ordre du modèle et de la méthode d'analyse de la texture.

		<b>Erreur de prédiction (<math>\epsilon \pm 003</math>) <math>\bar{x}</math></b>			
<i>Taille de la fenêtre (W)</i>	<i>L'ordre du modèle (P)</i>	<i>Levinson</i>	<i>Covariance</i>	<i>Burg</i>	<i>Moindres carrés</i>
<b>4</b>	2	1,2271	1,0735	1,0581	0,4445
	3	1,1724	1,0122	0,9903	0,3803
	4	1,1509	0,9907	0,9621	0,3569
	5	1,1354	0,9774	0,9421	0,3437
	6	1,123	0,9702	0,9278	0,337
<b>8</b>	2	1,9644	1,7853	1,7855	1,1366
	3	1,841	1,641	1,6412	0,6658
	4	1,7875	1,5711	1,5709	0,6658
	5	1,7587	1,5278	1,5281	0,5713
	6	1,7397	1,4966	1,4978	0,5072
<b>16</b>	2	1.6119	1,5566	1,5565	2,0584
	3	1.9760	1,5307	1,5307	9,9116
	4	1,9750	1,5207	1,5007	7,9112
	5	0.6812	0,676	0,6703	2,311
	6	0,6535	0,6423	0,6427	2,0584

**Tableau III.4 :** L'erreur d'exécution en fonction de la taille de la fenêtre de voisinage, de l'ordre du modèle et de la méthode d'analyse de la texture.

### III.3 Interprétation

Nous constatons visuellement que quelque soit la méthode d'estimation, quelque soit P, quelque soit le type de balayage, les zones de même texture sont bien mises en évidence lorsque W est grand. Cependant ces zones sont mieux délimitées lorsque W est petit.

Cette constatation n'est pas toujours vérifiée lorsqu'on se réfère à l'erreur de prédiction.

Pour toutes les méthodes d'estimation, quelque soit W et le type de balayage, l'erreur de prédiction diminue lorsque P augmente. Cependant, il est très difficile de voir visuellement l'influence de P sur les images segmentées.

En comparant les méthodes d'estimation entre elles, il est également difficile de voir visuellement des différences notables entre les résultats.

Cependant, il est clair que d'un point de vue « erreur de prédiction », la méthode des moindres carrés est meilleure que les autres méthodes d'estimation. On peut également noter en tenant compte de ce paramètre que la méthode de Burg et de covariance sont comparables (l'erreur de prédiction est presque identique) et que la méthode de Levinson est la plus mauvaise.

On peut aussi voir que le balayage de Hilbert donne de meilleurs résultats que le balayage vidéo puisque pour un même W, même P et même méthode d'estimation, l'erreur de prédiction reste plus petite dans le cas du balayage Hilbert. Les zones de même texture sont mieux délimitées mais elles restent tout de même floues. Un filtrage peut alors facilement y remédier à ce problème.

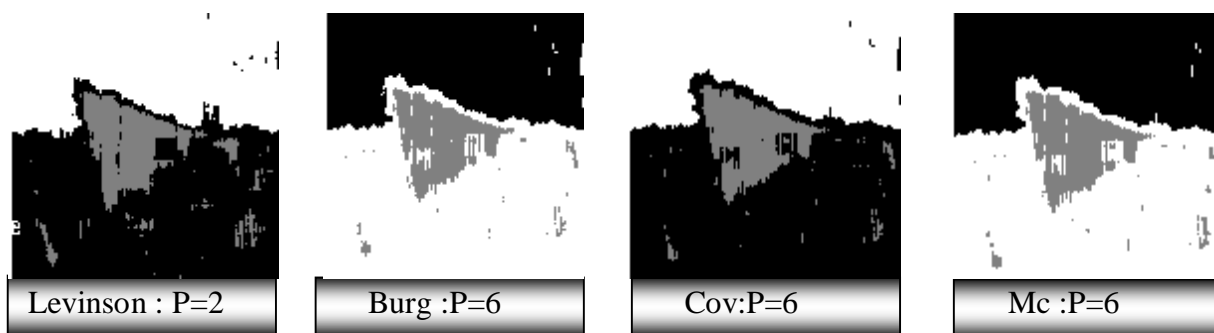
En ce qui concerne le temps d'exécution des différentes méthodes, celui-ci varie selon la méthode d'estimation utilisée, la taille de la fenêtre et l'ordre du modèle, plus W et P augmentent plus le temps de calcul nécessaire pour l'estimation des attributs de texture est élevé. Ceci est justifiable car plus W et P augmentent plus le nombre de pixels et la taille des systèmes à résoudre sont élevés. Cependant, la méthode des Moindres carrés est la plus rapide, elle fournit le meilleur temps d'exécution soit de 63.018 secondes alors que le temps le plus lent est obtenu par la méthode de Levinson soit de 137.138 secondes.

Il y a lieu de relever que dans les différents tests que nous avons effectué sur d'autres images, les résultats obtenus ont montré que la performance d'une méthode par rapport à une autre dépend de la nature de la texture qui compose l'image selon qu'elle est fine ou grossière, douce ou granulée. En effet pour des régions de texture fine, une petite fenêtre de voisinage suffit pour donner des résultats plus performants contrairement à des régions de texture grossière.

Les figures (III.9) à (III.13) montrent quelques résultats de la segmentation obtenus sur les autres images tests.

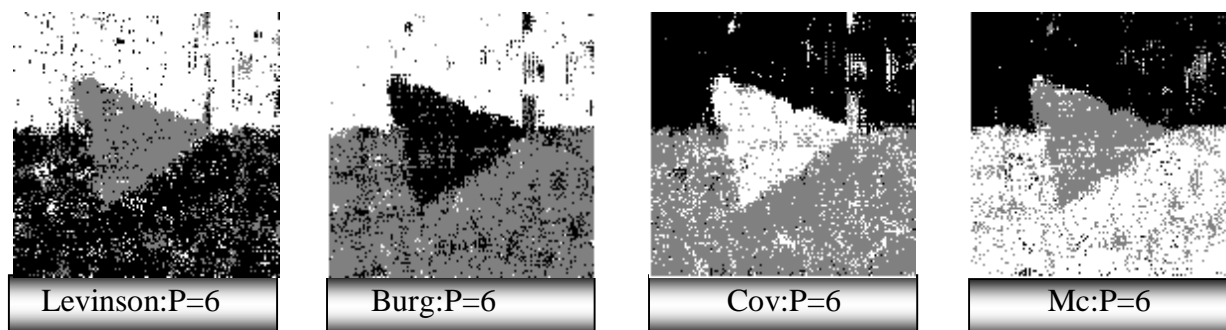
**Balayage vidéo**

$W=8$



**Balayage Hilbert**

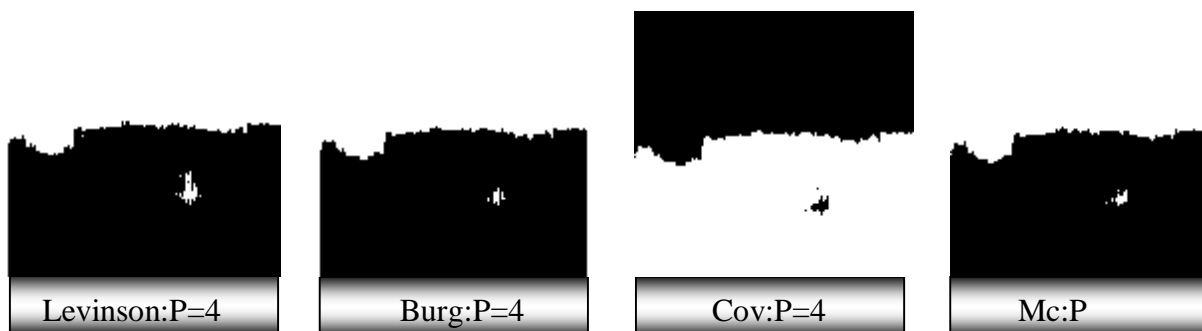
$W=8$



**Fig.III.9 :** Meilleurs résultats obtenus de la segmentation de l'image Test2 par les différentes méthodes et selon les deux types de balayage

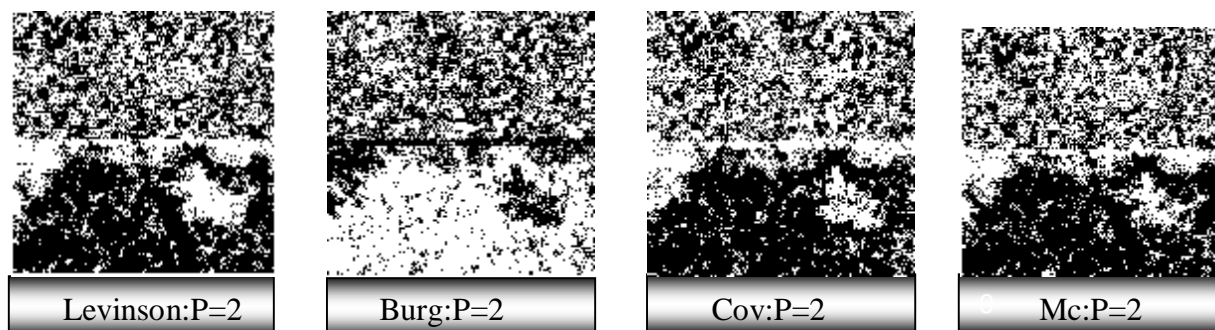
Balayage vidéo

$W=16$



Balayage Hilbert

$W=16$



**Fig.III.10** :Meilleurs résultats obtenus de la segmentation de l'image Test3 par les différentes méthodes et selon les deux types de balayage

**Balayage vidéo**

$W=4$



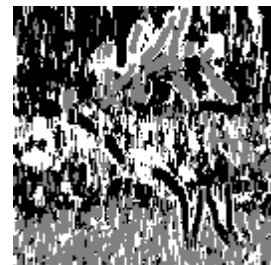
Levinson :P=4



Burg:P=3



Cov:P=3



Mc:P=3

**Balayage Hilbert**

$W=4$



Levinson:P=3



Burg:P=3



Cov:P=3

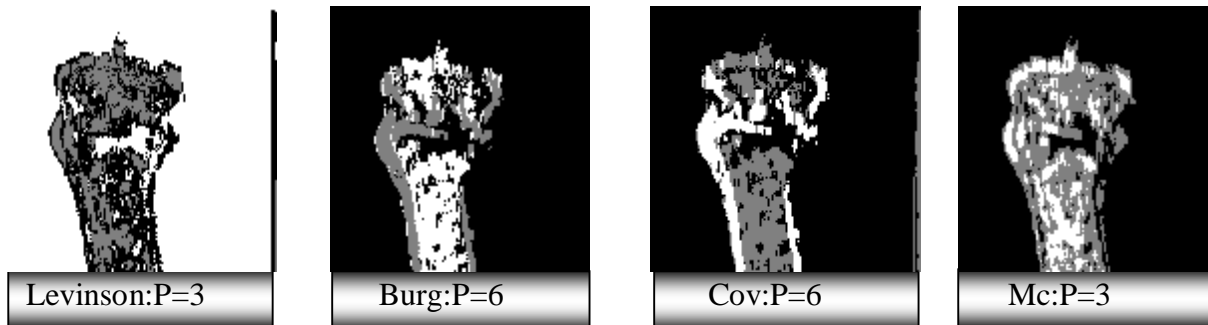


Mc:P=3

**Fig.III.11** :Meilleurs résultats obtenus de la segmentation de l'image Test4 par les différentes méthodes et selon les deux types de balayage

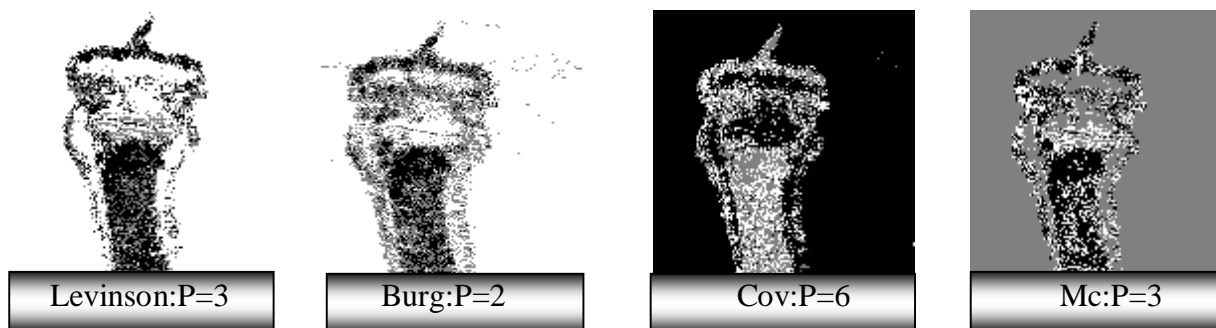
Balayage vidéo

$W=4$



Balayage Hilbert

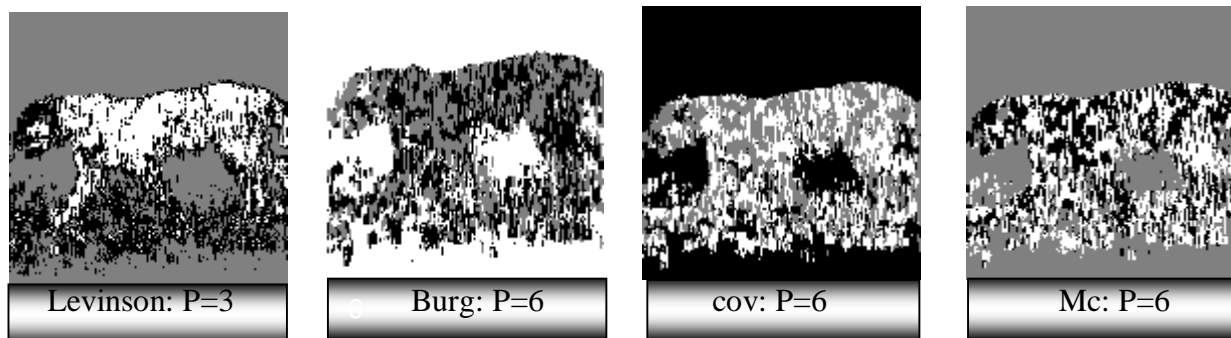
$W=4$



**Fig.III.12** : Meilleurs résultats obtenus de la segmentation de l'image Test5 par les différentes méthodes et selon les deux types de balayage

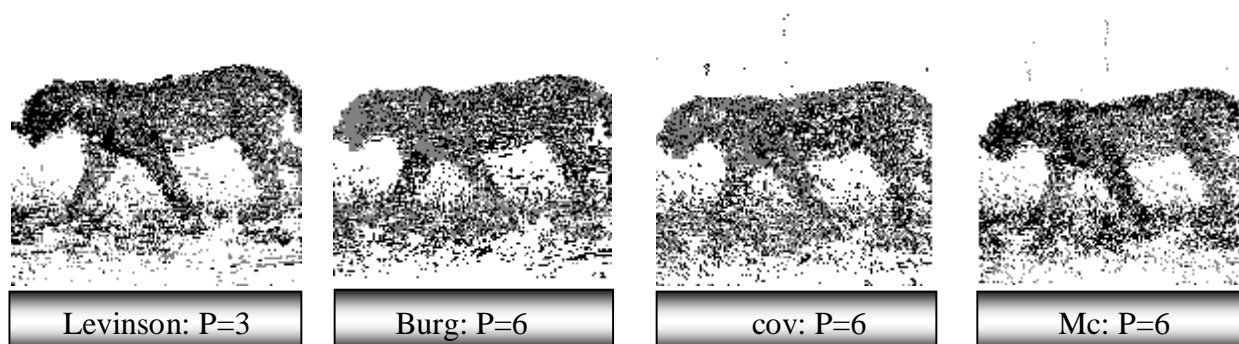
**Balayage vidéo**

***W=4***



**Balayage Hilbert**

***W=4***



**Fig.III.13 :** Meilleurs résultats obtenus de la segmentation de l'image Test6 par les différentes méthodes et selon les deux types de balayage

Comme on le remarque aussi dans ces résultats pris de chaque méthode d'estimation pour les différentes valeurs de la fenêtre de voisinage  $W$  et de l'ordre  $P$ , visuellement on peut dire que les images obtenues sont presque semblable autrement dit, l'aperçu de ces images ne montre pas l'influence de  $W$ , de  $P$  et de la méthode utilisée. Mais de point de vue rapidité on confirme ici que la méthode des moindres carrés est la plus performante. Par contre on peut voir clairement que le balayage Hilbert a fourni de bons résultats en comparant avec le balayage vidéo.

### **III.4 Conclusion**

Nous avons présenté dans ce chapitre, les résultats de segmentation obtenus par les différentes méthodes d'estimation en fonction des paramètres W,P et du type de parcours suivi pour le balayage de l'image. Nous avons constaté du point de vue rapidité que la méthode des moindres carrés est la plus performante parmi les méthodes d'estimation utilisées.

Le balayage Hilbert a fourni de meilleurs résultats puisque il permet de mettre en évidence les différentes zones texturées présentes dans l'image. Cependant, un filtrage de l'image segmentée ou des paramètres peut améliorer la qualité de la segmentation.

---

# *Conclusion Générale*

## **Conclusion générale**

Nous avons présenté dans ce mémoire une méthode de segmentation d'image texturée basée sur la modélisation autorégressive.

Notre méthode consiste à extraire dans un premier temps les attributs de texture pour chaque pixel en balayant localement l'image suivant deux types de parcours (vidéo et Hilbert). La classification des pixels est effectuée par la suite sur la base de ces attributs en utilisant la méthode classification des K-mens.

Une bonne description de l'information texturale nécessite un choix judicieux de plusieurs facteurs. Nous avons montré l'influence de la taille de la fenêtre d'analyse ainsi l'ordre du modèle sur le temps de classification, nous avons également montré que le choix du parcours joue un rôle prépondérant pour décrire les textures.

Les résultats expérimentaux portant sur les différents types d'images ont montré qu'aucune méthode d'estimation des paramètres AR ne peut être considérée meilleure que l'autre. En revanche l'utilisation du parcours Hilbert nous a permis d'améliorer les résultats obtenus. Nous avons constaté du point de vue rapidité que la méthode des moindres carrés est la plus performante parmi les méthodes d'estimation utilisées.

L'application des modèles AR sur les images texturées ne se limite pas seulement dans notre travail cependant, elle peut être étendue aux autres types de balayage et à l'estimation des paramètres AR 2D par l'algorithme de Levinson ou par la méthode des moindres carrés. Afin d'améliorer la qualité des résultats de la segmentation une étape de filtrage est envisagée.

En fin, nous souhaitons que notre travail servira de base pour la réalisation d'autres applications dans le domaine de traitement d'image

---

# *Bibliographie*

# *Bibliographie*

- [1] Christophe ROSENBERGER, Mise en Oeuvre d'un Système Adaptatif de Segmentation d'images. L'Université de RENNES 1.
- [2] M.Unser. Description statistique de la texture. PhD thesis, EPFL, Lansanne, 1984.
- [3] M.Unser. Texture classification and segmentation using walevet frames. IEEE Transactions on Image Prossing, 4(11): 1549-1560, November 1995.
- [4] T. Chang and C.C Jay Kuo. Texture analysis and classification with treestructured walevet transform. IEEE Transactions on Image Processing, 2(4):429-441, October 1993.
- [5] J.F. Haddon and J.F. Boyce. Texture classification of segmented regions of flir images using neural networks. In IEEE International Conference on Image Processing, volume3, Pages 660-664, AUSTIN, November 1994.
- [6] A. Laine and J. Fan. Texture classification by walevet packet signatures. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(15):1186-1191, January 1993.
- [7] R.W. Picard. A society of models for video and image libraries. IBM systems journal,3,1996.
- [8] T. Zhang. Issues in texture classification. CS328B, May 1996.
- [9] B. Julesz. Experiments in the visual perception of texture. Scientific software, 232:2-11, 1975.
- [10] R.M. Haralick. Statistical and structural approaches to texture. In proceedings of the IEEE, volume 69, pages 786-804, May 1979.
- [11] J.M. Francos, A.Z. Meiri, and B.Porat. A unified texture model based on a 2-d wold-like decomposition. IEEE Transaction on Signal Processing, 41: 2665-2678, August 1993.
- [12] A. Gagalowicz. Vers un modèle de textures. PhD thesis, Université Pierre et Marie CURIE, PARIS VI, May 1983.
- [13] Thèse Madjid MOGHRANI. Segmentation coopérative et adaptative d'images multicomposantes : Application aux images CASI.
- [14] Thèse K.Hammouche : Méthodes statistiques d'analyse de la texture appliquées à la segmentation d'image.
- [15] Alain Boucher.IFI. Cours Texture : Vision par ordinateur.

- [16] UNSER ,M. and DE COULON, F. Analyse des textures et segmentation.
- [17] Jean -Jaques ROUSSELLE. Les contours actifs, une méthode de segmentation. Application à l'imagerie médicale. Juillet 2003. Université de TOURS.
- [18] Catherine ACHARD. Cours traitement d'image 2006.
- [19] Henri MAÎTRE. La détection des contours dans les images.
- [20] Sébastien LEFEVRE. Une nouvelle approche pour la classification non supervisée en segmentation d'image (cours). LSIIT-CNRS/ Université de Strasbourg.
- [21] Classification par K-means.  
[Http://www.aiaccess.net/y tut liste descriptive.htm#K means](http://www.aiaccess.net/y_tut_liste_descriptive.htm#K_means)
- [22] Olivier Besson. Analyse Spectrale Paramétrique. 3<sup>ème</sup> année ENSICA.
- [23] I.D.LANDAU. Techniques de modélisations récursives pour l'analyse spectrale paramétrique adaptative. Laboratoire d'Automatique de Grenoble (UA 228, CNRS-INPG), GRECO Systèmes Adaptatifs (GR 69 CNRS), ENSIEG-INPG, BP 46, 38402. SAINT-MARTIN-D'HERES.
- [24] Mohamed NAJIM. Modélisation et identification en traitement du signal. Ingénieur ENSERB, Professeur à la Faculté des Sciences de Rabat. MASSON, Paris Milan Barcelone Mexico 1988.
- [25] <http://www.univ-rouen.fr/psi/heutte/rdp>
- [26] H.Youlal, M.J.Idrissi, M.Najim. Modélisation paramétrique en traitement d'images. MASSON, Paris Milan Barcelone 1994.