

REPUBLIQUE DU CAMEROUN

\*\*\*\*\*

Paix-Travail-Patrie

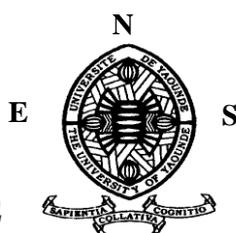
\*\*\*\*\*

UNIVERSITE DE YAOUNDE I

\*\*\*\*\*

ECOLE NORMALE SUPERIEURE  
DE YAOUNDE I

\*\*\*\*\*



REPUBLIC OF CAMEROON

\*\*\*\*\*

Peace-Work-Fatherland

\*\*\*\*\*

UNIVERSITY OF YAOUNDE I

\*\*\*\*\*

HIGHER TEACHER TRAINING  
COLLEGE OF YAOUNDE I

\*\*\*\*\*

DEPARTEMENT DE PHYSIQUE

*DEPARTMENT OF PHYSICS*

**ANALYSE SPECTRALE ET  
MODELISATION 2D : ESTIMATION DES  
PARAMÈTRES DE PRODUCTION D'UNE  
TEXTURE SYNTHÉTISÉE D'IMAGE PAR  
UN CODE MATLAB OPTIMISÉ**

Mémoire présenté en vue de l'obtention  
du diplôme de Professeur de l'Enseignement Secondaire  
Deuxième grade (DIPES II)

*Par:*

**KOLOKO DEUBOUE MARC SOREL**

*TITULAIRE DU DIPES I - Licencié en physique*

**Matricule: 02Y484**

*Sous la direction de :*

**Prof. NDOUGSA MBARGA Théophile**

**Maître de Conférences**

**Année académique 2015 – 2016**

---

## **DEDICACES**

---

Je dédie ce mémoire à ma famille.

---

## REMERCIEMENTS

---

Je tiens à remercier sincèrement :

- mon directeur de mémoire le **Professeur NDOUGSA MBARGA Théophile**, Maître de conférences à l'Ecole Normale Supérieure de Yaoundé 1 pour son investissement total et personnel dans la réalisation de ce mémoire. Le Professeur m'a aidé à mieux comprendre la philosophie de la recherche et à devenir plus systématique dans mon travail ;

- toute **l'équipe pédagogique de l'Ecole Normale Supérieure de Yaoundé 1** et particulièrement **les enseignants du département de physique**, pour avoir développé en moi grâce à leurs enseignements de qualité, des compétences pédagogiques en théorie et en pratique dans ce domaine scientifique ;

- les **membres du Jury** pour l'honneur qu'ils m'accordent en acceptant d'évaluer ce travail ;

- les aînés du laboratoire **NGOH Jean Daniel** et **YANDJIMAIN Justine** pour leurs disponibilités, leurs encouragements, leurs conseils et leurs remarques constructives qui ont été pour nous une source de motivation et ont enrichi notre travail ;

- ma famille qui m'a soutenu sans relâche durant toutes ces années d'études et en particulier : mon père **DEUBOUE Raphaël**, ma mère **KAMENI Thérèse**, mes mamans **NGUILIMENI Carole** et **EMAKOA Clotilde**, mes oncles **Dr. NZOMO Joseph**, **TOUKAM Michel**, **TCHAMDA Raphaël** et **KOUECHOU Michel**, sans oublier ma grande sœur **KUOKAM Anne** et mes frères **SIEWE Pierre**, **TCHEUNTA Francis**, **BODA Merlin**, **HAMACHI Brice**, **DEUBOUE GUY** et **YOMENI Cédric** ;

- mes amis et camarades de promotion qui m'ont accompagné durant ce cycle de formation notamment : **BIKEK Gervais**, **HOUPA Sonia**, **BIYA Arthur**, **SIEWE Titus** et tous les membres du C.12.

---

## TABLE DES MATIERES

---

<b><u>DEDICACE</u></b> .....	ii
<b><u>REMERCIEMENTS</u></b> .....	iii
<b><u>TABLE DES MATIERES</u></b> .....	iv
<b><u>LISTE DES FIGURES</u></b> .....	vi
<b><u>LISTE DES TABLEAUX</u></b> .....	vii
<b><u>GLOSSAIRE DES TERMES</u></b> .....	ix
<b><u>RESUME</u></b> .....	x
<b><u>ABSTRACT</u></b> .....	xi
<b><u>INTRODUCTION GENERALE</u></b> .....	1
<b><u>REVUE DE LA LITTERATURE</u></b> .....	4
<u>Introduction</u> .....	5
<b><u>1.1. Définition de la texture</u></b> .....	5
<b><u>1.2. Techniques d'analyse de la texture</u></b> .....	6
<u>1.2.1. Modèle structurelles</u> .....	6
<u>1.2.2. Analyse spectrale</u> .....	7
<u>1.2.3. Analyse spatio-temporelle</u> .....	7
<u>1.2.4. Approche mutli-niveaux</u> .....	7
<u>1.2.5. Géométrie fractale</u> .....	7
<u>1.2.6. Modélisation</u> .....	7
<u>1.2.7. Matrices de co-occurrence</u> .....	8
<b><u>1.3. Description de la méthode par analyse spectrale</u></b> .....	8
<u>1.3.1. Autocovariance et autocorrélation</u> .....	8
<u>1.3.2. Méthode classique</u> .....	9
<u>1.3.3. Méthode paramétrique</u> .....	10
<b><u>1.4. Définitions des concepts utilisés</u></b> .....	11
<u>1.4.1. Processus stochastique</u> .....	11
<u>1.4.2. Processus auto-normal fini tore</u> .....	11
<b><u>1.5. Modèle autorégressif AR</u></b> .....	12
<u>1.5.1. Intérêt du modèle</u> .....	12
<u>1.5.2. Représentation du modèle autorégressif 2D</u> .....	12
<u>Conclusion</u> .....	14

<u>METHODOLOGIE ET PROGRAMMATION</u> .....	15
<u>Introduction</u> .....	16
<b><u>2.1. Méthode du maximum de vraisemblance</u></b> .....	16
<b><u>2.2. Système de voisinage considéré</u></b> .....	17
<b><u>2.3. Nombre de paramètres à estimer</u></b> .....	18
<b><u>2.4. Optimisation</u></b> .....	19
<b><u>2.3. Optimisation du code Matlab</u></b> .....	20
<u>Conclusion</u> .....	21
<u>RESULTATS ET DISCUSSIONS</u> .....	22
<u>Introduction</u> .....	23
<b><u>3.1. Image test</u></b> .....	24
<b><u>3.2. Présentation des résultats et discussions</u></b> .....	24
3.2.1. Paramètres et ordres de l'hypothèse isotrope de la texture d'une image statique .....	30
3.2.2. Paramètres et ordres du modèle anisotrope de la texture de l'image statique .....	34
<b><u>3.3. Analyse globale des deux hypothèses isotrope et anisotrope de la texture d'une image statique d'une image statique</u></b> .....	34
<b><u>3.4. Apport pédagogique et intérêt didactique du mémoire</u></b> .....	34
3.4.1. Apport pédagogique .....	34
3.4.2. Intérêt didactique du mémoire .....	35
<u>Conclusion</u> .....	35
<u>CONCLUSION GENERALE ET PERSPECTIVES</u> .....	36
<u>BIBLIOGRAPHIE GENERALE</u> .....	38

---

## LISTE DES FIGURES

---

- Figure 1.1- Textures – Le motif brique correspond à une structure régulière alors que l’herbe est plutôt aléatoire.
- Figure 1.2- Configuration de voisinage d’un processus AR.
- Figure 1.3- Représentation d’une image en 2D.
- Figure 2.1- Exemples de voisinage pour 4 ordres différents.
- Figure 3.1- Exemple d’images acquises en statique.

---

## LISTE DES TABLEAUX

---

<u>Tableau 2.1</u> : Ensemble de voisins symétriques pour un modèle AR NC jusqu'à l'ordre 9 – la configuration des pixels est donnée dans la figure 2.1.....	18
<u>Tableau 2.2</u> : Nombre de paramètres à estimer sous l'hypothèse isotrope et anisotrope jusqu'à l'ordre 9 .....	19
<u>Tableau 3.1</u> : Paramètres estimés du code non optimisé sous l'hypothèse isotrope de la texture d'une image statique jusqu'à l'ordre 6 .....	24
<u>Tableau 3.2</u> : Paramètres estimés du code de Maya sous l'hypothèse isotrope de la texture d'une image statique jusqu'à l'ordre 9.....	25
<u>Tableau 3.3</u> : Paramètres estimés du code optimisé sous l'hypothèse isotrope de la texture d'une image statique jusqu'à l'ordre 9.....	25
<u>Tableau 3.4</u> : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 1 .....	26
<u>Tableau 3.5</u> : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 2 .....	26
<u>Tableau 3.6</u> : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 3 .....	27
<u>Tableau 3.7</u> : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 4 .....	27
<u>Tableau 3.8</u> : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 5 .....	27
<u>Tableau 3.9</u> : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 6 .....	28
<u>Tableau 3.10</u> : Paramètres estimés sous l'hypothèse isotrope des deux codes (Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 7 .....	28
<u>Tableau 3.11</u> : Paramètres estimés sous l'hypothèse isotrope des deux codes (Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 8 .....	28
<u>Tableau 3.12</u> : Paramètres estimés sous l'hypothèse isotrope des deux codes (Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 9 .....	29
<u>Tableau 3.13</u> : Paramètres estimés du code non optimisé sous l'hypothèse anisotrope de la texture d'image statique jusqu'à l'ordre 6 .....	30
<u>Tableau 3.14</u> : Paramètres estimés du code de Maya sous l'hypothèse anisotrope de la texture d'une image statique jusqu'à l'ordre 7 .....	30
<u>Tableau 3.15</u> : Paramètres estimés du code optimisé sous l'hypothèse anisotrope de la texture d'une image statique jusqu'à l'ordre 7 .....	31
<u>Tableau 3.16</u> : Paramètres estimés sous l'hypothèse anisotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 1 .....	32

<u>Tableau 3.17</u> : Paramètres estimés sous l'hypothèse anisotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 2 .....	32
<u>Tableau 3.18</u> : Paramètres estimés sous l'hypothèse anisotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 3 .....	32
<u>Tableau 3.19</u> : Paramètres estimés sous l'hypothèse anisotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 4 .....	33
<u>Tableau 3.20</u> : Paramètres estimés sous l'hypothèse anisotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 5 .....	33
<u>Tableau 3.21</u> : Paramètres estimés sous l'hypothèse anisotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 6 .....	33
<u>Tableau 3.22</u> : Paramètres estimés sous l'hypothèse anisotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 7 .....	34

---

## GLOSSAIRE DES TERMES

---

**Chaussée** : Partie de la voie publique aménagée pour la circulation.

**Code informatique** : Ensemble d'instructions conçues et programmées par l'homme et qui sont exécutées par un logiciel de simulation numérique pour un but précis.

**Estimation** : Evaluation.

**Fissure** : petite crevasse.

**Matlab** : Logiciel de simulation numérique.

**Modélisation d'un système** : Représentation d'un système sous la base d'hypothèses simplificatrices afin de l'implémenter dans les codes informatiques pour des simulations.

**Pixels** : ensemble de points lumineux formant une image.

**Texture d'image** : disposition des composants de l'image.

**Optimisation d'un code informatique** : Amélioration de la rapidité d'exécution des instructions du code informatique.

---

## RESUME

---

L'un des défauts déterminants pour décider de la rénovation d'une chaussée est la fissure [Chambon, 2010]. La rénovation efficace de la couche d'usure de la chaussée plutôt que sa réparation complète permettrait d'optimiser la gestion des infrastructures routières des pays Africains. Ceci contribuerait à augmenter la part des financements alloués à la croissance et au développement. Pour rendre la détection de fissures plus confortable et moins dangereuse pour les personnes qui la réalisent, mais aussi plus efficace et moins coûteuse, des systèmes d'acquisition d'images et de détection automatique de défauts ont été conçus. Parmi les méthodes de détection des fissures conceptualisées et implémentées, la méthode par filtrage adapté combinée à une segmentation par modélisation markovienne [Chambon, 2010] donne des résultats intéressants. Toutes fois, ces résultats présentent encore de fausses détections et des détections incomplètes [Maya, 2011]. On peut penser que ces défauts proviennent du fait que la texture particulière de la chaussée n'est pas bien prise en compte par les modèles utilisés [Maya, 2011]. Des solutions pour améliorer les résultats portent dans leur majorité sur la production d'une texture synthétisée proche de l'originale et utilisent une modélisation basée sur les modèles AR 2D et Markovien [Chambon, 2010]. Plusieurs codes sont déjà écrits et certains figurent dans la bibliothèque de Matlab [Maya, 2011]. Ce travail porte sur l'optimisation du code Matlab basé sur la rapidité du calcul du critère à minimiser pour un système de voisinage non causal d'ordre 9. Le code optimisé va permettre une estimation rapide des paramètres nécessaires à la production d'une texture synthétisée.

La première partie est consacrée à la revue de la littérature des notions liées à la modélisation de la texture de la chaussée. Nous présentons aussi le modèle autorégressif non causal 2D.

Dans la deuxième partie, nous nous intéressons à la méthodologie nécessaire à l'estimation des paramètres, mais aussi à l'algorithme lié à la programmation. Pour y arriver, nous avons opté pour la méthode du maximum de vraisemblance. Ceci nous amène à un problème d'optimisation non-linéaire. L'optimisation de notre code est faite pour un modèle autorégressif 2D. Nous utilisons ici la proposition de Kamina [kamina, 2011] pour un système de voisinage d'ordre 9.

La troisième partie est réservée aux résultats et discussions. Nous travaillons avec deux hypothèses de texture : isotrope et anisotrope. Pour une texture d'image statique (acquise par un appareil photographique), nous confrontons les résultats numériques de notre code optimisé aux résultats numériques d'un modèle autorégressif non causal 2D (code non optimisé d'ordre 6) et du code de Maya (ordre 9) [MAYA, 2011]. L'analyse des résultats obtenus nous fait constater que la texture isotrope d'ordre 4 est la meilleure pour une image statique. A la fin de cette partie, nous présentons l'apport pédagogique et l'intérêt didactique de ce mémoire.

**Mots clés : Modélisation - Texture - Méthode du maximum de vraisemblance - Paramètres - Code Matlab optimisé.**

---

## ABSTRACT

---

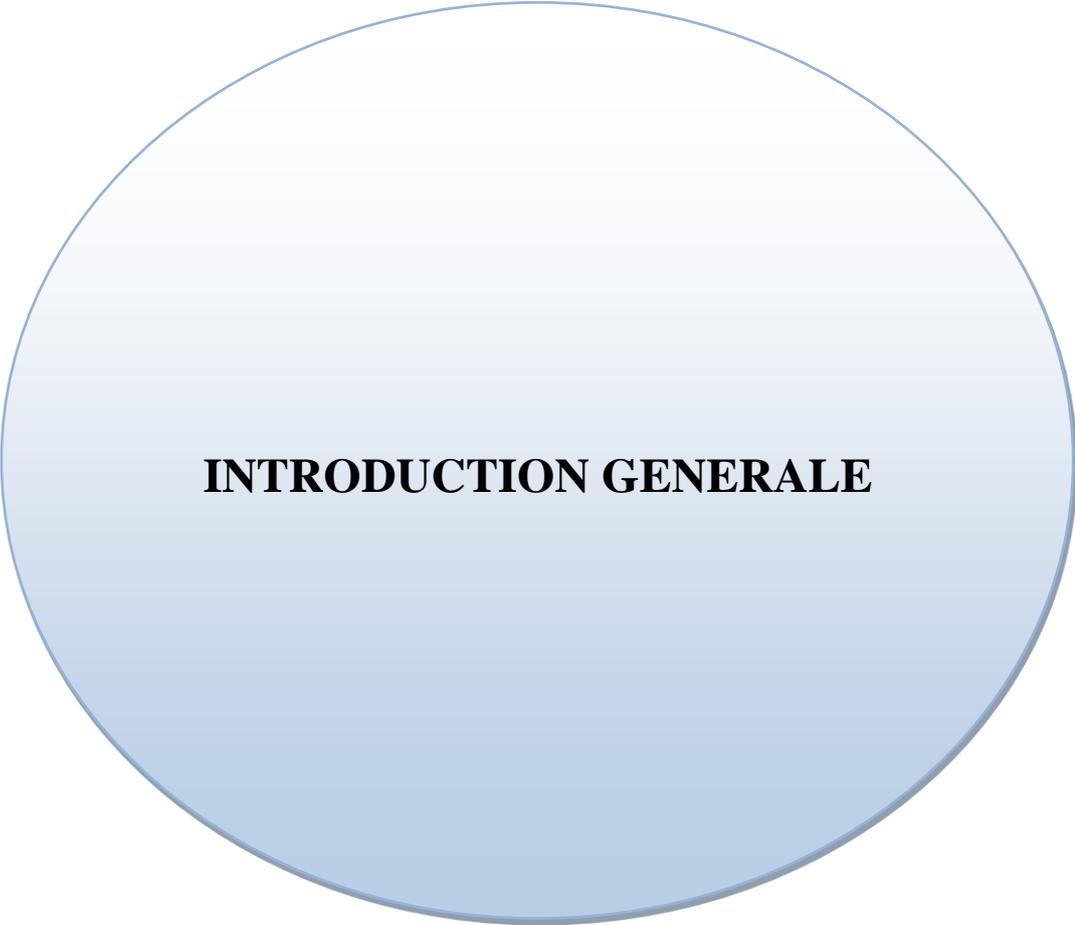
To decide the renovation of the roadway, one important element is the fissure. Efficient renovation of roadway layer rather than complete reparation would permit the optimization of African countries road infrastructure. With that, more finances are destined for growth and development. To render fissure detection for humans more comfortable and less expensive, the automatic fissure detection systems on image have been conceived. Among those systems, we have the method using combination of adapted filter and segmentation by Markovien modelisation [Chambon, 2010] which gives interesting results. But, these results still give counter faults detections [Maya, 2011]. One might think that these faults observed arise from the bad taking into account of roadway texture model. The solutions proposed to ameliorate the results are based on the production of image texture base on AR 2D Model and Markovien Model [Chambon, 2010]. Many codes have been written and another exists in Matlab library. The recent publications on the modelisation of an image texture are based on an optimization of Matlab code based on the rapidity of the calculation of criterion to minimize for non-causal neighboring system of order 9. The Matlab optimized code will permit a rapid calculation of estimation parameters necessary for the production of synthesized texture

The first part talks about literature review of notions linked to the modelisation of the texture of the roadway and the presentation of the autoregressive non causal 2D model.

In the second part, we are interested in the methodology necessary for the estimation of parameters, but also to the algorithm linked to the programming. To arrive of that, we opted the maximum likelihood method. It lead us to a non-linear optimization problem. Thanks to the Kamina proposition [Kamina, 2011] used to optimize the Matlab code by non-causal 2D autoregressive method. We realize optimization of parameter estimation for a neighboring system of order 9.

The last part was reserved for results and discussions. For a static image texture, we confronted numerical results of our optimized code to numerical results of a non-causal 2D autoregressive model (no optimized code order 6), and those of the Maya code for a system neighboring to non-causal of order 9 [Maya, 2011]. We analyzed results of our confrontation to predict for each texture model (isotropy and anisotropy), the parameters and the orders which fit to a texture synthesized production very identical to the original. And the best model texture is isotropy and the order is 4. The end of this part is reserved for pedagogies contribution and didactic interest of this work.

**Keywords:** Modelisation - Texture - The maximum likelihood method – Parameters - Optimized Matlab code.



# **INTRODUCTION GENERALE**

---

# INTRODUCTION GENERALE

---

## 1- Contexte de l'étude

Pour réaliser l'optimisation de la gestion des infrastructures routières, il ressort des travaux qu'il faut s'appuyer sur la fissure, défaut le plus présent et déterminant pour décider de la rénovation d'une chaussée. En effet, ces fissures sont susceptibles de laisser pénétrer l'eau en profondeur et de détériorer la structure de la chaussée, par exemple la saison pluvieuse avec les pluies. Elles sont aussi parfois révélatrices d'un défaut de la structure de la chaussée. Si les informations géométriques et les informations liées au profil longitudinal et au profil transversal de la chaussée sont régulièrement collectées, on peut programmer efficacement la rénovation suffisamment tôt.

## 2- Problématique

Parmi les méthodes de détection des fissures conceptualisées et implémentées, la méthode par filtrage adapté combinée à une segmentation par modélisation markovienne [Chambon, 2010] donne des résultats intéressants. Toutes fois, ces résultats présentent encore de fausses détections et des détections incomplètes [Maya, 2011]. On peut penser que ces défauts proviennent du fait que la texture particulière de la chaussée n'est pas bien prise en compte par les modèles utilisés.

## 3- Objectifs

Ce travail tente de proposer une solution pour améliorer les résultats obtenus en cherchant à prendre en compte la texture de la chaussée. C'est la raison pour laquelle nous avons choisi d'optimiser le code Matlab qui estime les paramètres nécessaires à la production d'une image synthétisée de la chaussée. Ainsi, en utilisant la méthode du maximum de vraisemblance [Chellappa, 1983] dans un modèle autorégressif non-causal 2D, nous obtenons une fonction non-linéaire. Pour la minimiser, nous utilisons l'optimisation non-linéaire. Cette optimisation est faite pour un système de voisinage non causal d'ordre 9. On utilise la proposition d'optimisation de Kamina [Kamina, 2011] pour un système de voisinage non causal d'ordre 6. Les résultats numériques obtenus sont confrontés sous l'hypothèse (isotrope ou anisotrope) de la texture d'une image statique aux résultats numériques du code non optimisé d'une part ; à ceux du code de Maya [MAYA, 2011]. Le système de voisinage est de 6 pour le modèle et 9 pour le code de Maya.

## **4- Plan**

Dans la suite de ce mémoire, premièrement, nous présentons la revue de la littérature des notions liées à la modélisation de la texture de la chaussée. Nous nous attardons aussi sur le modèle autorégressif non causal 2D. Deuxièmement, nous nous intéressons à la méthodologie nécessaire à l'estimation des paramètres, mais aussi à l'algorithme lié à la programmation. Troisièmement, nous confrontons nos résultats à ceux d'un modèle autorégressif non causal 2D (code non optimisé) et du code de Maya [Maya, 2011]. Ceci est fait sous l'hypothèse isotrope et anisotrope de la texture d'une image statique. Nous analysons les résultats pour choisir les paramètres et l'ordre qui conviennent le mieux à une production de texture synthétisée identique proche de l'originale. A la fin de cette partie, nous donnons l'apport pédagogique et l'intérêt didactique de ce mémoire. Une conclusion générale suivie d'une perspective d'un travail prochain va boucler notre travail.



**REVUE DE LA LITTERATURE**

## REVUE DE LA LITTÉRATURE

---

### Introduction

L'étude des textures rencontrées dans les images est un thème qui suscite beaucoup d'intérêt dans la communauté du traitement d'image. Il s'agit d'un domaine de recherche encore largement ouvert, si l'on en croit le nombre de publications récentes portant sur ce sujet que l'on peut trouver dans la littérature. Il existe une grande diversité dans les approches utilisées pour décrire ou identifier une texture, toutes aussi variées que la diversité des textures que l'on peut rencontrer dans la nature. Il existe aussi différents types d'applications de l'analyse de texture. En effet, nous pouvons citer la reconnaissance de texture afin d'effectuer une classification ou une segmentation :

- la classification qui a pour but de fournir une image classifiée où chaque région uniformément texturée est assignée à la classe de texture à laquelle elle appartient et ce suivant un certain critère de similitude.
- La segmentation qui consiste à détecter les frontières entre deux textures adjacentes dans la même image.

La modélisation de la texture est une autre application possible. Elle consiste à déterminer les modèles de distribution propres à chaque texture. Ces modèles peuvent être aussi bien employés pour estimer les paramètres nécessaires à la synthèse que pour la reconnaissance de texture.

Dans la première partie de ce chapitre, nous donnons une présentation de la notion de texture en traitement d'images. Ensuite, nous décrivons les concepts des méthodes d'analyse de texture [Paquis, 1996]. Puis, nous détaillerons la méthode choisie pour notre travail qui est l'analyse spectrale. Enfin, nous présentons le modèle autorégressif 2D. Le but de cette démarche consiste à définir une méthode d'analyse de la texture qui soit adaptée à l'étude de la corrélation dans la texture des images des surfaces de chaussées.

### 1.1- Définition de la texture

Plusieurs définitions de la notion de texture ont été présentées au cours des vingt dernières années. Ces définitions sont rarement génériques et portent plutôt sur un des aspects ou une des applications de la notion de la texture [Oukil, 2007].

En pratique, on peut distinguer deux grandes classes de textures, qui correspondent à deux niveaux de perception cf. figure 1.1 :

- **les textures régulières**, dans lesquelles la périodicité du motif est évidente, telle que les grilles, les murs, les tissus, etc.
- **les textures aléatoires** pour lesquelles la distribution des intensités n'est l'objet d'aucune régularité apparente telle que le sable, les nuages, l'herbe, etc.



(a) brique

(b) herbe

**Figure 1.1- Textures – Le motif brique correspond à une structure régulière alors que l’herbe est plutôt aléatoire.**

La première famille est bien décrite par des approches fréquentielles ou des approches structurelles dans lesquelles on associera un motif et des règles de placement sur un pavage régulier. La seconde est plutôt décrite par des lois statistiques, une description spectrale en termes de densité de puissance, des propriétés de corrélation ou d’isotropie. Mais une texture n’est jamais strictement périodique ni totalement aléatoire et les deux modèles se complètent naturellement. L’une des difficultés est de les mêler dans des modèles capables de s’adapter à la variabilité des textures étudiées [Oukil, 2007].

Dans ce mémoire, on travaille avec la texture des surfaces de chaussées. Cette texture admet certes comme primitive évidente le granulat mais il est instable au niveau de la forme, de la taille et de son homogénéité en termes de niveau de gris. Donc, on considère cette texture plutôt avec la seconde approche.

## 1.2- Techniques d’analyse de texture

Le but de l’analyse des textures est de décrire ces textures de façon formelle, en utilisant une formation mathématique. Une multitude de méthodes, de variantes et de combinaisons de méthodes ont déjà été proposés dans la littérature et prouvées en pratique. Dans la suite on va résumer le principe des méthodes étudiées par Stéphane PAQUIS dans sa thèse au sein de l’IFSTTAR (Institut Français des sciences et technologies des transports, de l’Aménagement et des Réseaux) [Paquis, 1996].

### 1.2.1- Méthodes structurelles

Cette approche suppose que les surfaces étudiées sont composées d’éléments de bases arrangés selon une loi de placement. Les étapes d’analyse sont d’abord l’identification des éléments constitutifs, puis la définition des règles de placement. Dans [Paquis, 1996], il a été montré que cette méthode est applicable à l’étude de texture régulière donc elle ne s’adapte pas aux textures que nous étudions du fait qu’un grand nombre de facteurs aléatoires sont à considérer.

### **1.2.2- Analyse spectrale**

Cette analyse permet de mettre en évidence des correspondances entre les domaines spectral et spatial. Ainsi, le spectre d'une texture régulière de structure grossière (respectivement fine) est concentré dans les basses (respectivement hautes) fréquences. On obtient le spectre usuellement en utilisant la transformée de Fourier 2D. Cette analyse a montré l'absence de la périodicité et de la direction privilégiée au sein des textures de chaussées. Ainsi, elle a confirmé le caractère aléatoire de ces textures.

### **1.2.3- Analyse spatio-fréquentielle.**

Cette technique offre la possibilité de mettre en évidence des propriétés locales sur une zone d'intérêt dans une image. Le principe d'extraction de données spatio-fréquentielles des textures consiste en la convolution de l'image avec un banc de filtres prédéfinis. A partir de ces résultats, un calcul de statistiques locales est effectué et le vecteur de paramètres texturés est obtenu par combinaison de ces statistiques. Les bancs usuels sont les masques de Laws et les filtres de Gabor 2D. Cette analyse est particulièrement efficace pour l'étude de textures périodiques.

### **1.2.4- Approche multi-niveaux**

L'analyse d'image à différentes échelles est née de la nature même des images qui contiennent des objets de diverses tailles placés à diverses distances de l'observateur. Le principe de cette approche est d'étudier la texture à différents niveaux de résolution. La transformée en ondelettes et le modèle pyramidal sont les deux procédés les plus utilisés dans l'analyse multi-niveaux. Cette approche donne des résultats intéressants pour la discrimination des 4 types de chaussées distingués par les gestionnaires de la route.

### **1.2.5- Géométrie fractale**

En analyse de texture, la dimension fractale, qui est une mesure du degré d'irrégularité d'un objet, décrit une certaine propriété de la texture. Le modèle fractal est basé essentiellement sur l'estimation par des méthodes spatiales de la dimension fractale de la surface représentant les niveaux de gris de l'image. Cette approche n'a pas été adoptée pour la discrimination des textures de chaussées parce que ces textures sont probablement trop similaires du point de vue de leur granularité.

### **1.2.6- Modélisation**

Cette méthode suppose que l'image, présentée par l'ensemble de niveaux de gris, est une réalisation d'un champ de variables aléatoires. L'étude statistique des relations entre un pixel et son voisinage permet de définir un modèle de la texture. Les attributs sont alors les coefficients du modèle de la texture, dont le nombre est égal à la taille de voisinage considéré. Les deux modèles les plus utilisés sont : les modèles autorégressifs (AR) et les champs de Markov. D'après [Paquis, 1996], ces modèles ne sont pas efficaces pour la macro-texture. Nous détaillons le modèle autorégressif 2D à la fin de ce chapitre.

### 1.2.7- Matrices de co-occurrence

Les matrices de co-occurrence constituent une représentation simple des relations spatiales inter-pixels. Elles permettent d'estimer les statistiques d'ordre 2 de l'image. Ces statistiques sont accumulées en un ensemble de matrices 2D, dont chacune mesure la dépendance spatiale de deux niveaux de gris, étant donné un vecteur de déplacement. On peut dériver de la matrice de co-occurrence des caractéristiques de la texture, comme l'énergie, l'entropie, le contraste, l'homogénéité et la corrélation. Cette approche est adaptée aux textures de chaussées mais l'inconvénient est le choix de l'élément structurant qui nécessite une connaissance a priori sur la texture.

### 1.3- Description de la méthode par analyse spectrale

Il existe deux grandes approches pour l'analyse spectrale. La première contient des méthodes dites classiques ou non-paramétriques qui sont fondées sur le périodogramme. La seconde classe, dite des méthodes non-classiques ou paramétriques, est de supposer que les données suivent un modèle, ce qui fournit un moyen de paramétrer le spectre à réduire ainsi le problème de l'estimation spectrale à celui de l'estimation des paramètres dans le modèle supposé [Lawrence, 1987]. Avant de détailler ces deux méthodes, on donne une définition de la notion d'auto-covariance qui sera utilisée dans les sections suivantes.

#### 1.3.1- Autocovariance et autocorrélation

Dans la nature, quelle que soit la variable observée, sa variabilité dans un voisinage proche sera généralement moindre que sur une plus grande étendue : le tout est de savoir à quelle distance cette variabilité est significative. On comprend donc la nécessité de mettre au point des tests statistiques prenant en compte cet aspect. Si nous considérons une variable  $x(m, n)$  avec  $m \in [1, M]$  et  $n \in [1, N]$ , dont la valeur est fonction de sa position  $m$  et  $n$  dans l'espace ou dans le temps, l'auto-covariance se définit comme la covariance entre la variable  $x(m, n)$  et elle-même après un décalage entier  $(k, l)$ . Si l'on note  $\bar{x}$ , la moyenne des  $x(m, n)$ , l'auto-covariance est donnée par :

$$Cov(k, l) = (MN)^{-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (x(m, n) - \bar{x})(x(m+k, n+l) - \bar{x}) \quad (1.1)$$

Et l'autocorrélation est donnée par :

$$C(k, l) = \frac{Cov(k, l)}{\sigma(k)\sigma(l)} \quad (1.2)$$

Le coefficient de corrélation est compris entre -1 et 1. Il est égal à 1 dans le cas où l'une des variables est fonction affine croissante de l'autre variable, à -1 dans le cas où la fonction affine est décroissante. Les valeurs intermédiaires renseignent sur le degré de dépendance linéaire entre les deux variables. Plus le coefficient est proche des valeurs extrêmes -1 et 1, plus la corrélation entre les variables est forte. Une corrélation égale à 0 signifie que les variables sont linéairement indépendantes.

### 1.3.2- Méthode classique

La transformation usuelle pour obtenir le spectre est la transformée de Fourier bidimensionnelle (TF 2D). Basiquement, les résultats de la TF nous informent sur le contenu fréquentiel global à partir duquel certains paramètres caractéristiques texturaux peuvent être extraits. Donc, cette méthode se base sur la décomposition du processus étudié en fonctions périodiques. Cette décomposition exprime la structuration du processus et donc ses échelles d'auto-covariance.

#### Transformée de Fourier Discrète (TFD)

L'idée de décomposer un signal en fonctions périodiques revient au mathématicien français Joseph Fourier (1768-1830). Le principe en est que tout signal, continu ou discret, peut être résumé en une série de fonctions périodiques de base. L'expression de la TFD 1D est :

$$F(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f(n) e^{-2i\pi \frac{kn}{N}} \quad (1.3)$$

où  $F(k)$  est la transformée de Fourier du signal 1D  $f(n)$ .

La transformée de Fourier discrète inverse (TFDI) étant :

$$F(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} f(k) e^{2i\pi \frac{kn}{N}} \quad (1.4)$$

La transformée de Fourier discrète bidimensionnelle (TFD 2D) est définie par :

$$F(k, l) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-i2\pi(\frac{km}{M} + \frac{ln}{N})} \quad (1.5)$$

et sa réciproque par :

$$F(m, n) = \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} f(k, l) e^{i2\pi(\frac{km}{M} + \frac{ln}{N})} \quad (1.6)$$

#### Périodogramme

La méthode du périodogramme a été introduite par Schuster en 1898. Il est l'estimateur le plus utilisé de sa simplicité de mise en œuvre. Le périodogramme généralisé bidimensionnel est défini par :

$$\hat{S}(k, l) = \frac{1}{MN} \left| \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) e^{-i2\pi(\frac{km}{M} + \frac{ln}{N})} \right|^2 \quad (1.7)$$

où :

-  $x(m, n)$  est le signal 2D avec  $m \in [1, M]$  et  $n \in [1, N]$ .

-  $\hat{S}(k, l)$  est la densité spectrale estimée.

Une propriété intéressante du périodogramme est qu'il présente une relation directe avec la fonction d'auto-covariance. En effet, le périodogramme est lui-même la transformée de Fourier de l'auto-covariance empirique. Ce qui veut dire qu'il existe une relation connue et réversible entre les deux. Ils sont donc mathématiquement équivalents bien qu'ils expriment l'information de façon différente et qu'ils n'aient pas les mêmes propriétés statistiques. Le périodogramme 2D est traditionnellement modifié de façon à avoir les valeurs  $(k, l) = (0, 0)$  situées au centre. Il est symétrique par rapport à l'origine. Le périodogramme décroît rapidement quand il y a une corrélation entre les pixels et décroît lentement quand il n'y a pas de corrélation entre les pixels. Dans le cas de bruit blanc où il n'y a aucune corrélation, la densité spectrale de puissance est plate.

### **Périodogramme modifié**

La forte variance des résultats obtenus par des méthodes d'analyse utilisant le périodogramme a motivé le développement des méthodes modifiées qui ont une variance plus faible mais au détriment d'une résolution réduite. Parmi ces méthodes, nous pouvons citer les méthodes présentées dans [Lawrence, 1997] :

- la méthode de Daniel qui lisse le périodogramme en faisant la moyenne sur des bancs de fréquences adjacentes ;
- la méthode de Bartlett qui consiste à moyenniser plusieurs périodogramme calculés sur différents segments du signal ;
- la méthode de Welch qui est une extension de la méthode de Bartlett.

Plus précisément, le périodogramme de Welch est paramétré par :

- la longueur des segments : si on prend une petite longueur de segments, on obtient un plus grand nombre de segments à moyenniser et par conséquent un spectre estimé avec une plus petite variance, mais cela va diminuer la résolution fréquentielle. Cependant, si on augmente la longueur des segments, on augmente aussi la résolution fréquentielle tout en augmentant la variance ;
- le taux de recouvrement : en permettant aux séquences de se recouvrir, il est possible d'augmenter le nombre ou la longueur des séquences qui sont moyennées, pour arriver à un compromis entre la réduction de la variance et une meilleure résolution ;
- la forme de la fenêtre : la fenêtre rectangulaire est la fenêtre la plus simple qui est utilisée. Dans certains cas, l'utilisation d'autres fenêtres de troncature adoucies (telles que les fenêtres de Hamming, Hannig, Bartlett ...) permet d'améliorer la sensibilité au détriment de la résolution.

Dans la suite de notre travail, nous considérons la méthode de Welch qui est la méthode la plus générale.

### **1.3.3- Méthode paramétrique**

Dans une approche paramétrique, la première étape consiste à choisir un modèle approprié pour le processus. Ce choix est basé sur une connaissance a priori sur la manière dont le processus est généré. Les modèles les plus utilisés sont les modèles autorégressifs (AR), à moyenne ajustée (MA) et autorégressifs à moyenne ajustée (ARMA). Une fois le modèle choisi, la seconde étape est d'estimer les paramètres du modèle à partir des données disponibles. La dernière étape est d'estimer le spectre à partir des paramètres estimés.

L'intérêt d'une modélisation paramétrique est double : d'une part, elle permet de réduire l'espace de représentation, d'autre part, elle permet d'extraire de façon plus fine certaines informations. Dans notre travail, on représente la texture de la chaussée par un modèle autorégressif (AR). Ce modèle permet d'estimer les paramètres nécessaires pour caractériser l'apparence de la texture au moyen des statistiques du second ordre.

## 1.4- Définitions des concepts utilisés

### 1.4.1- Processus stochastiques

**Définition** : Un processus stochastique  $X = (X_t)_{t \in T}$  est une famille de variables aléatoires  $X_t$  indexées par un ensemble  $T$ .

En général  $T = \mathbb{R}$  ou  $\mathbb{R}_+$  et on considère que le processus est indexé par le temps  $t$ . Si  $t \in E$  est un ensemble fini, le processus est un vecteur aléatoire. Si  $t \in T = \mathbb{N}$ , alors le processus est une suite de variables aléatoire. Plus généralement quand  $T \subset \mathbb{Z}$ , le processus est dit discret. Ce processus est un moyen de décrire un phénomène physique quelconque à partir d'observations statistiques. Il permet d'effectuer la prédiction ainsi que l'estimation des paramètres du spectre spectrale. Nous nous intéressons aux processus qui reposent sur un nombre fini de paramètres, pour leur simplicité d'implantation. Les trois formes les plus répandues sont les modèles autorégressifs (AR), les modèles à moyenne ajustée (MA) ou les modèles autorégressifs à moyenne ajustée (ARMA). Notre étude utilise le modèle autorégressif. Il permet d'estimer une valeur par combinaison linéaire des valeurs qui lui sont voisines comme on verra dans le chapitre suivant.

### 1.4.2- Processus auto-normal fini tore

Comme décrit dans [Besag, 1974], on considère qu'un processus

$\{X(m, n) : m, n = 0, \pm 1, \dots, \pm \infty\}$  est auto-normal quand la distribution conjoint des variables du site est normale. Ce processus a une structure de probabilité conditionnelle :

$$E(X(m, n) | \text{toutes les autres valeurs de sites}) = \mu + \sum_{k,l} \lambda_{k,l} (X(m-k, n-l) - \mu) \quad (1.8)$$

où  $\mu$  est la moyenne et  $\lambda_{k,l} = 0$  sauf si  $k$  et  $l$  sont voisins.

La fonction de densité conjointe est donnée par :

$$P(x) = (2\pi\sigma^2)^{\frac{1}{2n}} |B|^{\frac{1}{2}} e^{\left(\frac{-1}{2}\sigma^{-2}(s-\mu l^T B(x-\mu l))\right)} \quad (1.9)$$

où  $l$  est un vecteur unitaire,  $B$  est une matrice symétrique définie positive dont la diagonale l'est unitaire et les autres éléments valent  $-\lambda_{k,l}$  (les paramètres ou coefficients à estimer).

Pour simplifier, on considère seulement un treillis rectangulaire avec des sites définis par des paires  $(m, n)$  sur une région définie (de taille  $M \times N$ ).

$$\{X(m, n) : m = 0, 1, \dots, M-1; n = 0, 1, \dots, N-1\} \quad (1.10)$$

Pour éviter les problèmes aux bords on travaille avec un tore (c'est-à-dire que les bords horizontaux et verticaux sont connectés de façon que le voisinage soit défini et tout point). La fonction de densité spectrale de ce processus décrit dans [Besag, 1975] est donnée par :

$$g_{pq} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \gamma_{mn} e^{-i(\omega_p m + \eta_p n)} \quad (1.11)$$

où  $\omega_p = \frac{2\pi p}{m}$ ,  $\eta_p = \frac{2\pi q}{m}$  avec  $p$  et  $q \in \mathbb{N}$  et  $\{\gamma_{mn}: m = 0, 1, \dots, M-1; n = 0, 1, \dots, N-1\}$  est l'autocovariance théorique de ce processus.

Pour que ce processus soit auto-normal fini il faut que sa fonction génératrice d'autocovariance (la transformée en  $Z$  de la fonction d'autocovariance) soit égale à :

$$f(\omega, \eta) = K \left\{ 1 - \sum_k \sum_l \lambda_{k,l} (\cos(\omega_p k + \eta_p l)) \right\}^{-1} \quad (1.12)$$

avec les conditions suivantes :

1.  $\lambda_{k,l} = 0$  sauf si  $0 \leq k \leq M-1, 0 \leq l \leq N-1$  ;
2.  $\lambda_{0,0} = 0$  ;
3.  $\lambda_{-k,-l} = \lambda_{k,l}$  ;
4.  $\sum_k \sum_l \lambda_{k,l} (\cos(\omega_p k + \eta_p l)) < 1$  pour  $0 \leq \omega, \eta \leq 2\pi$

où  $K$  est une constante positive liée à la variance de  $X_{m,n}$ , c'est-à-dire :

$$K = \sigma^2 = \gamma_{00} - \sum_k \sum_l \lambda_{k,l} \gamma_{kl} \quad (1.13)$$

et  $KB^{-1}$  la matrice de dispersion de  $X(m, n)$ .

## 1.5- Modèle auto-régressif AR

### 1.5.1- Intérêt du modèle AR

De nombreuses publications ont été réalisées dans le cadre de l'étude de la texture par un modèle AR 2D comme [Tugnait, 1993 ; Chellappa, 1983 ; Joshi.M, 2009 ; Deguchi, 1986]. Récemment, des modèles non causaux ont suscité beaucoup d'intérêt parce que ces modèles sont un choix plus naturel pour de nombreuses applications en traitement d'images. Avant qu'un modèle autorégressif non causal (ARNC) puisse être adapté à l'image donnée, deux problèmes doivent être abordés. Il faut déterminer, d'une part, le meilleur voisinage du modèle et, d'autre part, une méthode pour estimer les paramètres du modèle. Nous allons introduire le modèle AR 2D en commençant par présenter ces configurations possibles. L'estimation des paramètres se fera dans le chapitre suivant.

### 1.5.2- Représentation du modèle autorégressif 2D

Un des aspects le plus important dans une image en niveaux de gris est la dépendance statistique des niveaux de gris des pixels. Un des modèles qui présente cette dépendance est le modèle auto-régressif à deux dimensions. Ce modèle caractérise la dépendance statistique en

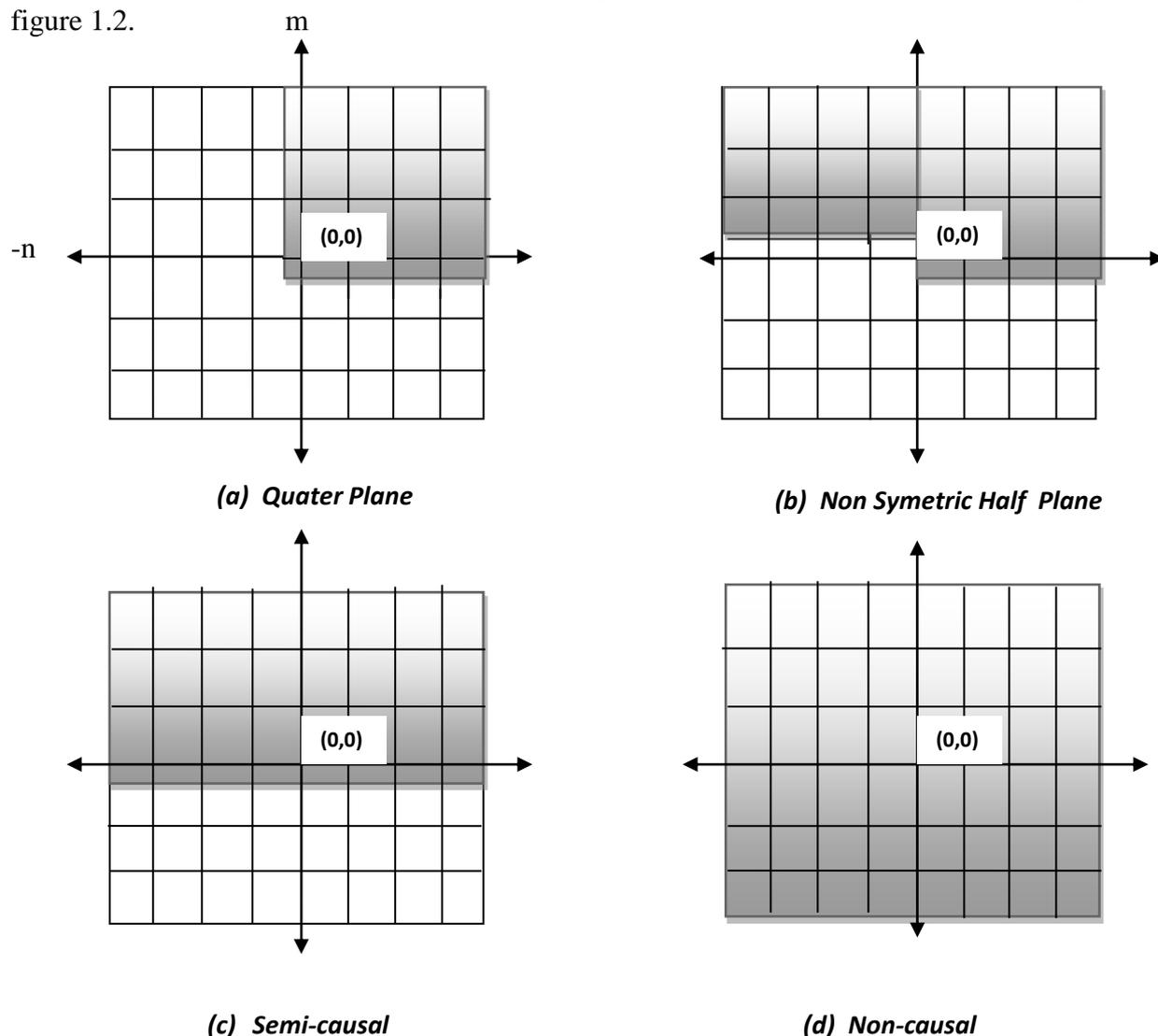
représentant l'intensité d'un pixel comme une combinaison linéaire 2D de l'intensité de ses voisins et un bruit additif :

$$x(m, n) = \sum_{k,l \in N} \lambda_{k,l} x[m - k, n - l] + e(m, n) \text{ avec } (k, l) \neq (0, 0) \quad (1.14)$$

avec :

- $\lambda_{k,l}$ , l'ensemble des coefficients de prédictions ;
- $e(m, n)$ , un bruit blanc gaussien ;
- $k, l \in N$  avec  $N$ , le voisinage considéré.

Il existe trois configurations de voisinage d'un processus AR [Oukil, 2007] illustrées par la figure 1.2.



**Figure 1.2-** Configuration de voisinage d'un processus AR – Les Pixels considérés pour chaque type de voisinage sont les points de rencontre (pixel) des traits verticaux et horizontaux de la fenêtre.

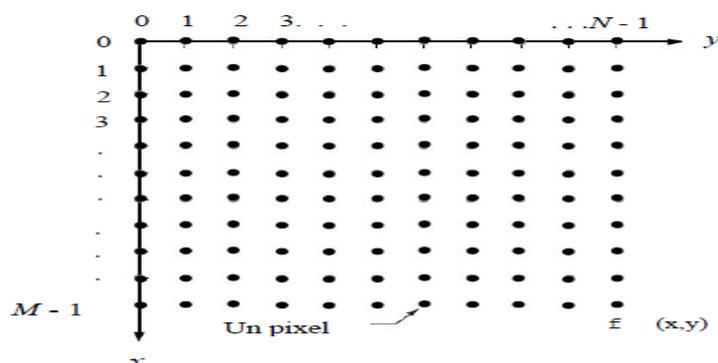


Figure 1.3- Représentation d'une image en 2D. Les points foncés sont les pixels.

### Modèle causal

La valeur prédite ne dépend que du passé. Comme une image n'a pas de définition temporelle, mais plutôt une définition spatiale, la forme causale n'a pas de signification physique [Deguchi, 1986]. Son avantage provient de la simplicité d'estimation des paramètres. Il existe deux grandes formes de voisinage causal. La première est le Quater Plane où dans l'équation (1.14),  $m \in [0, M]$  et  $n \in [0, N]$ . La seconde se nomme le *NonSymmetric Half Plane* et correspond à  $m \in [0, M]$  et  $n \in [0, N] \cup m \in [-M, 0]$  et  $n \in [1, N]$ .

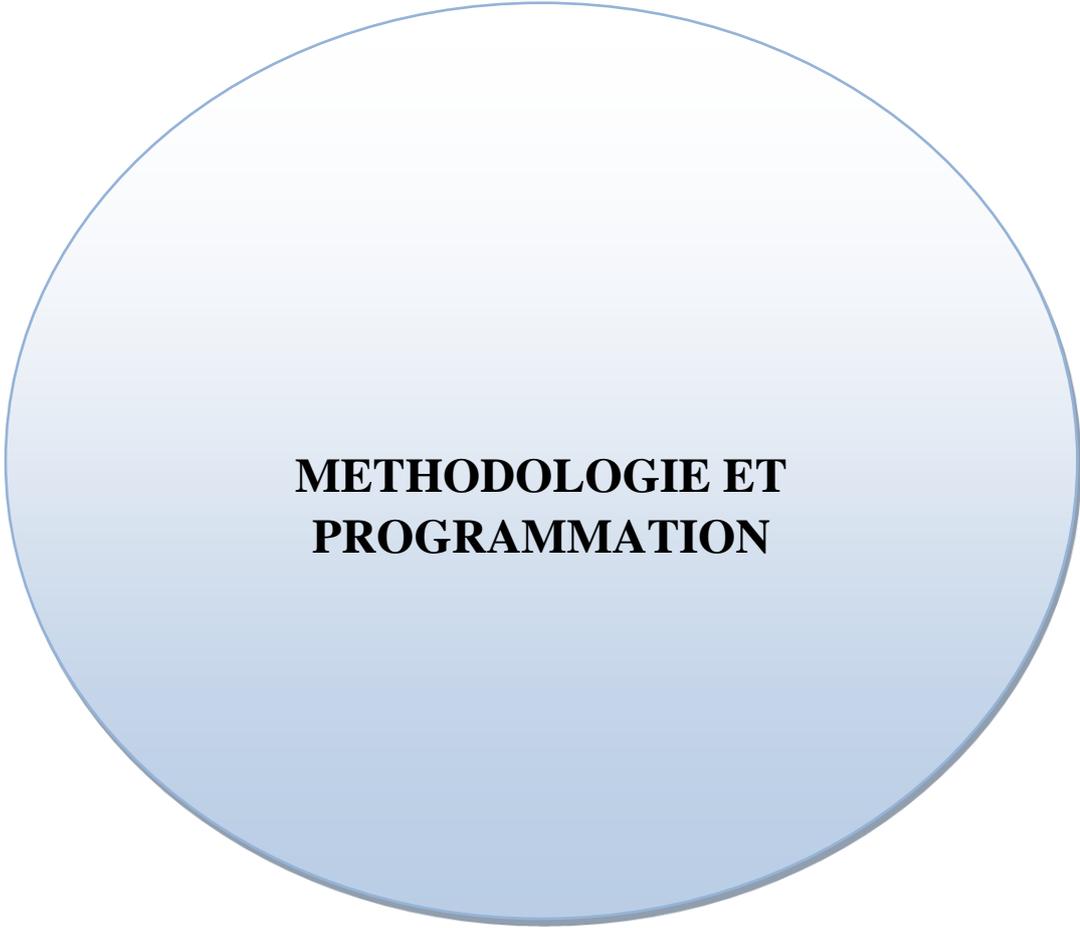
### Modèle semi-causal

Le modèle est causal et non-causal à la fois. La figure (1.2, c) illustre ce voisinage défini pour  $m \in [-M, M]$  et  $n \in [0, N]$ .

### Modèle non-causal

Le voisinage non-causal est le plus général. Il permet de n'avoir aucune contrainte sur la forme du voisinage. Cependant, l'estimation des paramètres non-causaux est plus compliquée. La figure (1.2, d) représente un voisinage non-causal défini par  $m \in [-M, M]$  et  $n \in [-N, N]$ . Après avoir vu les différents types de voisinage, nous travaillons avec le modèle non causal parce qu'il est le plus adapté pour représenter l'existence de la corrélation entre les lignes d'une image et entre les pixels d'une même ligne ainsi que la dépendance spatiale.

**Conclusion:** Dans ce chapitre, nous avons commencé par montrer le caractère complexe de la notion de texture et les techniques utilisées pour analyser la texture des surfaces de chaussées. Puis, nous avons détaillé les deux approches de l'analyse spectrale qui sont utilisées dans notre travail : classique et paramétrique. Enfin, nous avons présenté le modèle auto-régressif dans son entièreté. Le chapitre suivant aborde la manière d'estimer les paramètres d'un modèle autorégressif et sa mise en œuvre pour l'obtention d'un code optimisé de calcul.



**METHODOLOGIE ET  
PROGRAMMATION**

---

**METHODOLOGIE ET PROGRAMMATION**


---

**Introduction**

L'estimation des paramètres AR peut s'effectuer selon de nombreuses méthodes comme : le maximum de vraisemblance [Chellappa, 1983], les moindres carrés et l'erreur quadratique moyenne (Yule-Walker) [Oukil, 2007]. L'estimateur de moindres carrés est consistant et efficace pour estimer les paramètres du modèle AR 2D causal, mais cet estimateur n'est plus consistant dans le cas de non-causalité. Dans ce dernier cas, on utilise le maximum de vraisemblance. La modélisation consiste à associer à une image un modèle, représenté par un vecteur de paramètres  $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_p]$ . Ainsi, nous modélisons la texture avec un modèle AR non-causal 2D et pour estimer les paramètres de ce modèle, nous suivons la méthode proposée dans [Besag, 1975].

**2.1- Méthode du maximum de vraisemblance**

Pour estimer les paramètres du processus (décrit dans le paragraphe 1.4.2), Besag [Besag, 1975] utilise l'estimation du maximum de vraisemblance qui l'amène à l'expression:

$$(2\pi K)^{\frac{1}{2}MN} |\mathbf{B}|^{\frac{1}{2}} e^{-\frac{1}{2} \mathbf{K}^{-1} (\mathbf{x} - \boldsymbol{\mu})' \mathbf{B} (\mathbf{x} - \boldsymbol{\mu})} \quad (2.1)$$

Le logarithme de cette fonction est donnée par :

$$-\frac{1}{2} MN \log(2\pi K) + \frac{1}{2} \log(|\mathbf{B}|) - \frac{1}{2} \mathbf{K}^{-1} (\mathbf{x} - \boldsymbol{\mu})' \mathbf{B} (\mathbf{x} - \boldsymbol{\mu}) \quad (2.2)$$

où K est donnée par l'équation (1.13). En examinant cette fonction, on obtient les valeurs estimées des paramètres notés :  $\hat{\boldsymbol{\mu}} = \bar{\mathbf{x}}$  où  $\bar{\mathbf{x}}$  est la moyenne de l'échantillon. La valeur estimée de K est donnée par :

$$\hat{\mathbf{K}} = (MN)^{-1} (\mathbf{x} - \boldsymbol{\mu})' \mathbf{B} (\mathbf{x} - \boldsymbol{\mu}) \quad (2.3)$$

En remplaçant dans l'équation (2.2), on obtient :

$$-\frac{1}{2} MN \log(2\pi) + \frac{1}{2} \log((MN)^{-1} (\mathbf{x} - \bar{\mathbf{x}} \mathbf{l})' \mathbf{B} (\mathbf{x} - \bar{\mathbf{x}} \mathbf{l})) + \frac{1}{2} \log(|\mathbf{B}|) - \frac{1}{2} MN \quad (2.4)$$

Et en divisant par  $\frac{1}{2} MN$  et en enlevant la constante on obtient :

$$\log \hat{\mathbf{K}} - (MN)^{-1} \log(|\mathbf{B}|) \quad (2.5)$$

On peut écrire  $\mathbf{K}$  comme dans l'équation (2.2) mais en utilisant l'autocovariance empirique de cette manière :

$$\hat{\mathbf{K}} = \mathbf{C}_{00} - \sum_k \sum_l \lambda_{k,l} \mathbf{C}_{k,l} \quad (2.6)$$

où  $\mathbf{C}_{k,l}$  l'autocovariance empirique des décalages  $k$  et  $l$ , et est donnée par l'équation (1.1). Ainsi, nous obtenons le critère suivant à minimiser :

$$\log(\mathbf{C}_{00} - \sum_k \sum_l \lambda_{k,l} \mathbf{C}_{k,l}) - (MN)^{-1} \log(|\mathbf{B}|) \quad (2.7)$$

Pour l'estimation par maximum de vraisemblance, on a besoin de calculer le déterminant de la matrice  $\mathbf{B}$ . Dans [Besag, 1975], on peut calculer  $|\mathbf{B}|$  de cette manière :

$$|\mathbf{B}| = \prod_{p=0}^{M-1} \prod_{q=0}^{N-1} (1 - \sum_k \sum_l \lambda_{k,l} (\cos(\omega_p k + \eta_q l))) \quad (2.8)$$

En remplaçant dans l'équation (2.7), on obtient : (2.9)

$$\log\left(\mathbf{C}_{00} - \sum_k \sum_l \lambda_{k,l} \mathbf{C}_{k,l}\right) - (MN)^{-1} \prod_{p=0}^{M-1} \prod_{q=0}^{N-1} \left\{ \log\left(1 - \sum_k \sum_l \lambda_{k,l} (\cos(\omega_p k + \eta_q l))\right) \right\}$$

où les paramètres doivent respecter les conditions suivantes :

1.  $\lambda_{k,l} = 0$  sauf si  $0 \leq k \leq M - 1$ ,  $0 \leq l \leq N - 1$  ;
2.  $\lambda_{0,0} = 0$  ;
3.  $\lambda_{-k,-l} = \lambda_{k,l}$  ;
4.  $\sum_k \sum_l \lambda_{k,l} (\cos(\omega_p k + \eta_q l)) \leq 1$  pour  $0 \leq \omega, \eta \leq 2\pi$

On remarque que cette fonction est non-linéaire et pour la minimiser il faut l'optimisation non-linéaire. Avant de minimiser ce critère, on doit déterminer le voisinage à considérer et par rapport à cela le nombre de paramètres à estimer. Puis, il faut choisir l'algorithme d'optimisation à utiliser.

## 2.2- Système de voisinage considéré

Dans notre travail, on utilise le voisinage non causal pour le modèle AR 2D. Dans le système de voisinage de premier ordre, aussi appelé le système de 4-voisins, chaque pixel a quatre voisins. Dans le système de voisinage de second ordre, aussi appelé le système de 8-voisinage, il y a huit voisins pour chaque site. La figure 2.1 nous donne un exemple des voisinages qui sont habituellement utilisés dans ce système non causal. Le cercle foncé représente l'élément alors que ses voisins sont gris. Généralement, un voisinage d'ordre 2 est suffisant pour tenir compte de la corrélation entre les pixels de l'image. En tenant compte de la troisième condition à respecter, le tableau 2.1 montre les pixels à considérer dans chaque ordre jusqu'à l'ordre 9 en respectant l'équation (2.10) [Besag, 1974] :

$$(\mathbf{m}, \mathbf{n}) \in \mathbf{N}_{\mathbf{m}, \mathbf{n}} \rightarrow (-\mathbf{m}, -\mathbf{n}) \notin \mathbf{N}_{\mathbf{m}, \mathbf{n}}; \mathbf{N} = \{(\mathbf{m}, \mathbf{n}) : (\mathbf{m}, \mathbf{n}) \in \mathbf{N}_{\mathbf{m}, \mathbf{n}} \text{ or } (-\mathbf{m}, -\mathbf{n}) \in \mathbf{N}_{\mathbf{m}, \mathbf{n}}\} \quad (2.10)$$

On désigne ce voisinage par l'ensemble de voisins symétriques.

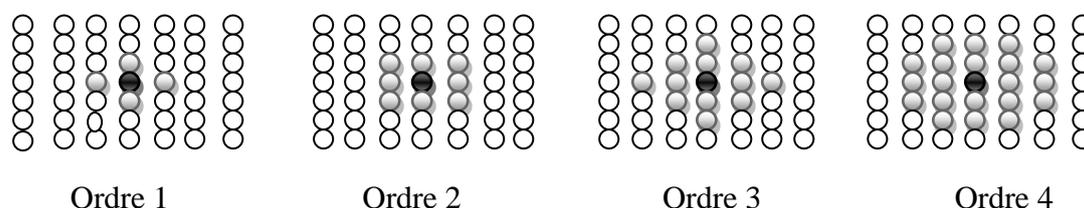


Figure 2.1- Exemples de voisinage pour 4 ordres différents.

Tableau 2.1- Ensemble de voisins symétriques pour un modèle AR NC jusqu'à l'ordre 9 – la configuration des pixels est donnée dans la figure 2.1.

ORDRE DU VOISINAGE	ENSEMBLE DE VOISINS SYMETRIQUES
$N_{s1}$	$(0,1), (1,0)$
$N_{s2}$	$N_{s1} \cup (1,1), (1,-1)$
$N_{s3}$	$N_{s2} \cup (0,2), (2,0)$
$N_{s4}$	$N_{s3} \cup (1,2), (2,1), (1,-2), (2,-1)$
$N_{s5}$	$N_{s4} \cup (2,2), (2,-2)$
$N_{s6}$	$N_{s5} \cup (0,3), (3,0)$
$N_{s7}$	$N_{s6} \cup (1,3), (3,1), (1,-3), (3,-1)$
$N_{s8}$	$N_{s7} \cup (2,3), (3,2), (2,-3), (3,-2)$
$N_{s9}$	$N_{s8} \cup (3,3), (3,-3)$

### 2.3- Nombre de paramètres à estimer

Le nombre de paramètres à estimer dépend du choix à réaliser pour la texture : elle est soit isotrope, soit anisotrope. Le nombre de paramètres dans le cas isotrope est égal à l'ordre choisi, mais dans le cas anisotrope plus l'ordre est grand, plus le nombre de paramètres augmente : soit par 2, soit par 4 et cela dépend si le voisinage de l'ordre contient 4 ou 8 pixels puisqu'on ajoute toujours le nombre de pixels de voisinage divisé par 2 en tenant compte de la condition

$\lambda_{-k,-l} = \lambda_{k,-l}$ . Dans le tableau 2.2, on montre le nombre de paramètres à estimer pour chaque ordre dans les deux cas.

**Tableau 2.2-** Nombre de paramètres à estimer sous l'hypothèse isotrope et anisotrope jusqu'à l'ordre 9

ORDRE	MODELE ISOTROPE	MODELE ANISOTROPE
1	1	2
2	2	4
3	3	6
4	4	10
5	5	12
6	6	14
7	7	18
8	8	22
9	9	24

## 2.4- Optimisation

Le critère à minimiser avec les conditions à respecter nous mettent devant un problème d'optimisation non linéaire avec contraintes (*Matlabmanual* en ligne).

**Définition :** L'optimisation non-linéaire avec contraintes consiste à trouver un vecteur  $x$  qui minimise la fonction scalaire non linéaire  $f(x)$  (que l'on appelle la fonction objet) en respectant des contraintes linéaires ou non linéaires ou les deux.  $t$  ou  $x$  appartenant à  $\mathbb{R}$  qui vérifie le système des contraintes s'appelle une solution réalisable.

$$\min_x f(x)$$

Sous les contraintes :

- contraintes non-linéaire d'inégalités et d'égalité :  $c(x) \leq 0, ceq(x) = 0$   
où  $c(x)$  et  $ceq(x)$  sont des fonctions qui retournent des vecteurs.
- contraintes linéaires d'inégalités et d'égalités :  $A \cdot x \leq B, Aeqx = Beq$
- bornes :  $lb \leq x \leq ub$   
où  $lb, Beq, lb$  et  $ub$  sont des vecteurs et  $A$  et  $Aeq$  des matrices.

La fonction Matlab adaptée à notre problème dans OptimizationToolbox est **fmincon**. Cette fonction est basée sur le gradient et propose quatre algorithmes pour résoudre le problème d'optimisation non-linéaire avec contraintes :

- L'algorithme *interior point -II* est particulièrement utile pour les problèmes parcimonieux à grande échelle, ainsi que pour les petits problèmes denses. Cet algorithme satisfait les contraintes de bornes à toutes itérations et peut récupérer à partir des résultats indéfinis ou infinis.

- L'algorithme **SQP – II** est utilisé pour les problèmes de petite ou moyenne échelle, donc il est plus rapide que le précédent mais moins précis.
- L'algorithme **active-sit– II** peut être utilisé avec un pas assez grand, ce qui rend cet algorithme plus rapide. Il est efficace sur certains problèmes avec des contraintes non lisses. Ce n'est pas un algorithme pour les grandes échelles.
- L'algorithme **trust-region-reflective-II** est utilisé pour les problèmes liés aux contraintes de bornes ou d'égalités linéaires uniquement, mais pas les deux. C'est un algorithme de grande échelle.

Normalement, il est recommandé de commencer par l'algorithme **interior point**. Puis, pour obtenir plus de rapidité d'exécution, dans le cas de problèmes petits ou moyens, on utilise l'algorithme **SQP** ou **active-sit**. On utilise l'algorithme **trust-region-reflective** quand on peut l'appliquer (c'est-à-dire quand on a des contraintes de bornes ou d'égalités linéaires uniquement).

En général, les étapes à suivre pour minimiser le critère sur Matlab sont :

- 1- Ecrire une fonction (*handlefunction*) qui prend la valeur du vecteur de paramètres comme entrée et donne les valeurs scalaires de la fonction objet, équation (2.7) et son gradient ;
- 2- Ecrire une fonction pour les contraintes. Dans notre cas, ce sont des contraintes linéaires que l'on obtient de la quatrième condition sur le critère qui donne  $|\sum_k \sum_l \lambda_{k,l}| \leq 1$ . Le nombre de conditions est donc  $2^n$  où n est le nombre de paramètres ;
- 3- Invoquer la routine d'optimisation avec contraintes (**fmincon** avec l'algorithme de **interior point**).

## 2.5- Optimisation du code Matlab

Pour résoudre un problème d'optimisation non-linéaire optimisant le code Matlab, plusieurs propositions ont été faites [Kamima, 2011]. Grâce à la proposition de Kamima [Kamima, 2011] utilisée pour optimiser le code Matlab par un modèle autorégressif non causal 2D dans ses travaux et ceci pour un système de voisinage d'ordre 6, nous réalisons l'optimisation d'estimation des paramètres pour un système de voisinage d'ordre 9.

On trouve que pour calculer la deuxième partie du critère à optimiser sur Matlab :

$$F2 = (MN)^{-1} \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \{ \log(1 - \sum_k \sum_l \lambda_{k,l} (\cos(\omega_p k + \eta_q l)) ) \}, \quad (2.10)$$

il faut utiliser deux boucles for ( $p = 0, \dots, M - 1, q = 1, \dots, N - 1$ ). Cela demande un grand nombre d'itérations ( $M \times N$ ) pour chaque valeur du vecteur de paramètres. Pour cela, on obtient un code Matlab très lent. Pour améliorer la rapidité du calcul (optimiser le code), on peut écrire cette partie de la façon suivante [Kamina, 2011] :

$$\begin{aligned} P(p, q) &= \sum_k \sum_l \lambda_{k,l} (\cos(\omega_p k + \eta_q l)) \\ &= \frac{1}{2} \sum_{k=N_x}^{N_x} \sum_{l=-N_y}^{N_y} \lambda_{k,l} (e^{-j2\pi(\frac{kp}{M} + \frac{lq}{N})} + e^{j2\pi(\frac{kp}{M} + \frac{lq}{N})}) \end{aligned}$$

$$= \frac{1}{2} \sum_{k=0}^{N_x} \sum_{l=0}^{N_y} \lambda_{k,l} \left( e^{-j2\pi\left(\frac{kp}{M} + \frac{lq}{N}\right)} e^{j2\pi\left(\frac{N_x p}{M}\right)} + e^{j2\pi\left(\frac{kp}{M} + \frac{lq}{N}\right)} e^{-j2\pi\left(\frac{N_y q}{N}\right)} \right) \quad (2.11)$$

En comparant cette équation avec les équations (1.5) et (1.6), on trouve qu'on peut utiliser l'algorithme de la transformée de Fourier 2D (TF 2D) et l'algorithme de la transformée de Fourier inversée 2D (TFI 2D) pour calculer cette partie du critère. Ainsi, on obtient un code qui prend moins de temps pour estimer les paramètres. Supposons que l'ordre maximal qui peut être choisi dans notre code soit 9, alors le nombre de pixels à considérer dans le système de voisinage est 48. Donc, 24 paramètres inconnus à estimer :

$$\Lambda = [\lambda(k, l); -3 \leq k, l \leq 3; (k, l) \neq (0, 0)] \quad (2.12)$$

Pour déterminer la forme de la matrice des paramètres, on décompose l'équation (2.11) et en la comparant à la forme de la TF, on obtient la matrice M x N de la forme :

$\lambda(3,3)$	$\lambda(2,3)$	$\lambda(1,3)$	$\lambda(0,3)$	$\lambda(1,-3)$	$\lambda(2,-3)$	$\lambda(3,-3)$	0	...	0
$\lambda(3,2)$	$\lambda(2,2)$	$\lambda(1,2)$	$\lambda(0,2)$	$\lambda(1,-2)$	$\lambda(2,-2)$	$\lambda(3,-2)$	0	...	0
$\lambda(3,1)$	$\lambda(2,1)$	$\lambda(1,1)$	$\lambda(0,1)$	$\lambda(1,-1)$	$\lambda(2,-1)$	$\lambda(3,-1)$	0	...	0
$\lambda(3,0)$	$\lambda(2,0)$	$\lambda(1,0)$	$\lambda(0,0)$	$\lambda(1,0)$	$\lambda(2,0)$	$\lambda(3,0)$	0	...	0
$\lambda(3,-1)$	$\lambda(2,-1)$	$\lambda(1,-1)$	$\lambda(0,-1)$	$\lambda(1,1)$	$\lambda(2,1)$	$\lambda(3,1)$	0	...	0
$\lambda(3,-2)$	$\lambda(2,-2)$	$\lambda(1,-2)$	$\lambda(0,-2)$	$\lambda(1,2)$	$\lambda(2,2)$	$\lambda(3,2)$	0	...	0
$\lambda(3,-3)$	$\lambda(2,-3)$	$\lambda(1,-3)$	$\lambda(0,-3)$	$\lambda(1,3)$	$\lambda(2,3)$	$\lambda(3,3)$	0	...	0
0	0	0	0	0	0	0	0	...	0

En respectant les contraintes :  $\lambda_{(0,0)} = 0$  et  $\lambda_{(k,l)} = \lambda_{(-k,-l)}$ . (2.13)

**Conclusion:** Ce chapitre nous a permis d'estimer et de présenter les paramètres du code optimisé. Le chapitre suivant sera consacré aux résultats et à la discussion.



**RESULTATS ET DISCUSSION**

---

**RESULTATS ET DISCUSSIONS**

---

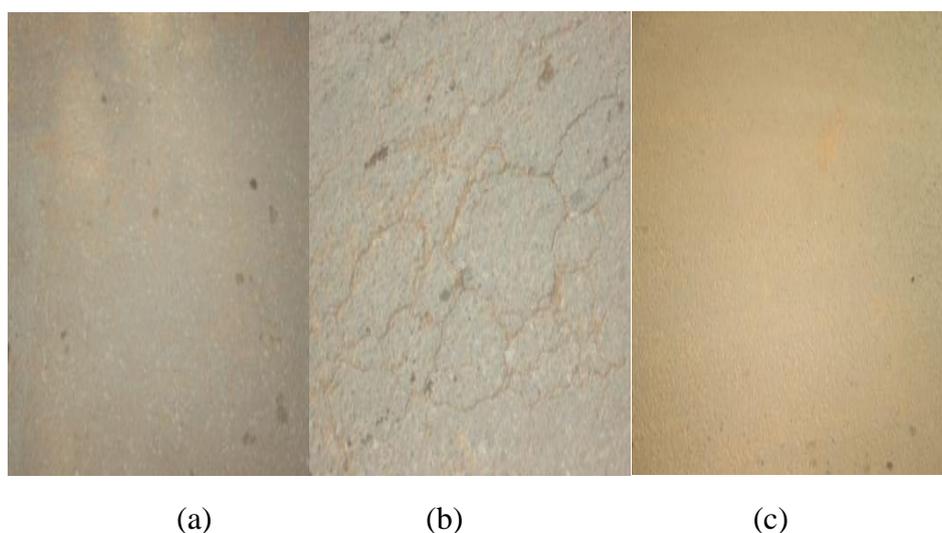
**Introduction**

Ici, nous présentons les résultats numériques obtenus de notre code optimisé d'ordres différents sous les deux hypothèses isotrope et anisotrope pour la texture d'une image statique. A la suite, nous comparons ces résultats aux résultats numériques du modèle autorégressif non causal 2D (code non optimisé) d'une part, aux résultats numériques du code de Maya [Maya, 2011] d'autre part.

**3.1- Image de test**

Dans son travail, Maya estime les paramètres obtenus de son code pour différentes textures d'images [Maya, 2011] et les compare à ceux du modèle autorégressif non causal 2D (code non optimisé). Dans notre mémoire, nous nous limitons à la texture d'une image statique car c'est celle qui est régulièrement utilisée pour la détection des fissures dans notre pays. Les images acquises en statique sont des images acquises sur route à partir d'un appareil photo numérique. Elles sont prises de telle sorte qu'il n'y ait pas de problèmes de luminosité, voir figure 3.1.

Nos trois images statiques (a, b, c) ont été prises par un appareil photo numérique de 18 Méga pixels successivement après 3m.



**Figure 3.1- Exemple d'images acquises en statique**

## 3.2- Présentation des résultats et discussions

### 3.2.1- Paramètres et ordres du modèle isotrope de la texture de l'image statique

Le tableau 3.1 montre les paramètres du modèle autorégressif non causal 2D (code non optimisé) du modèle isotrope de la texture d'une image statique.

Le tableau 3.2 présente les paramètres du code de Maya du modèle isotrope de la texture d'une image statique.

Le tableau 3.3 fait figurer les paramètres de notre code optimisé du isotrope de la texture d'une image statique.

**Tableau 3.1- Paramètres estimés du code non optimisé sous l'hypothèse isotrope de la texture d'une image statique jusqu'à l'ordre 6.**

VOISINAGE	PARAMETRES ESTIMES
$N_{s1}$	0.2315
$N_{s2}$	0.2344, -0.0155
$N_{s3}$	0.2232, 0, -0.0267
$N_{s4}$	0.2184, 0, 0, -0.0316
$N_{s5}$	0.2184, 0, 0, -0.0316, 0
$N_{s6}$	0.2184, 0, 0, -0.0316, 0, 0

**Tableau 3.2- Paramètres estimés du code de Mayasous l'hypothèse isotrope de la texture d'une image statique jusqu'à l'ordre 9.**

VOISINAGE	PARAMETRES ESTIMES
$N_{s1}$	0.2401
$N_{s2}$	0.2405, -0.001
$N_{s3}$	0.2398, -0.0003, -0.0001
$N_{s4}$	0.2390, -0.0001, -0.0012, -0.0024
$N_{s5}$	0.2184, 0, 0, -0.0316, 0
$N_{s6}$	0.2307, -0, -0.003, -0.0080, -0.0020, -0.0020
$N_{s7}$	0.2484, 0.00017, -0.00017, -0.00068, -0.0001, -0.0001, -0.0002
$N_{s8}$	0.2331, 0.0020, -0.0015, -0.0059, -0.0012, -0.0005, -0.0006, -0.0003
$N_{s9}$	0.1349, 0.0153, -0.0115, -0.0346, -0.0093, -0.0044, -0.0059, -0.0021, 0.0007

**Tableau 3.3- Paramètres estimés du code optimisé sous l'hypothèse isotrope de la texture d'une image statique jusqu'à l'ordre 9.**

VOISINAGE	PARAMETRES ESTIMES
$N_{s1}$	0.2499
$N_{s2}$	0.2498, -0
$N_{s3}$	0.2482, -0.0002, -0.005
$N_{s4}$	0.2425, -0.0004, -0.0012, -0.0045
$N_{s5}$	0.2355, -0.0003, -0.0010, -0.0100, -0.0005
$N_{s6}$	0.2317, -0, -0.0028, -0.0080, -0.0018, -0.0010
$N_{s7}$	0.2484, 0.0002, -0.0002, -0.0007, -0.0001, -0.0001, -0.0001
$N_{s8}$	0.2348, 0.0018, -0.0018, -0.0067, -0.0012, -0.0006, -0.0008, -0.0003
$N_{s9}$	0.1353, 0.0143, -0.0117, -0.0359, -0.0093, -0.0043, -0.0061, -0.0023, 0.0005

## Analyse des tableaux de l'hypothèse isotrope d'une texture d'image statique

Le coefficient le plus important des trois tableaux 3.1, 3.2, 3.3 dans tous les ordres est le premier coefficient. Il représente bien la dépendance entre chaque pixel et les 4 pixels de voisinage du premier ordre qui sont les plus proches. Les valeurs de paramètres obtenus de notre code optimisé (tableau 3.3) sont proches des valeurs des tableaux 3.1 et 3.2 jusqu'à l'ordre 6. C'est aussi le cas des ordres 7,8 et 9 pour les tableaux 3.2 et 3.3.

Ainsi donc, après le quatrième ordre, nous nous attendions à une production de texture synthétisée d'image sous l'hypothèse isotrope presque semblable et peu différente car les paramètres ne changent presque pas. Pour les trois codes, nous pourrions produire une texture synthétisée de la chaussée pour les ordres 4, 5 et 6 avec les trois codes. Le code de Maya et notre code optimisé aux ordres 7, 8, 9 pourraient produire des textures synthétisées voisines car les paramètres estimés ne diffèrent presque pas.

Le tableau 3.4 montre sous l'hypothèse isotrope, les paramètres estimés des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 1. Il ressort que le paramètre estimé est presque égal pour le code de Maya et notre code, mais un écart sensible existe avec celui du code non optimisé.

**Tableau 3.4 : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 1.**

Ordre 1	Paramètre estimé
Code non optimisé	0.2315
Code de Maya	0.2401
Code optimisé	0.2499

Le tableau 3.5 montre sous l'hypothèse isotrope, les paramètres estimés des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 2. Fort est de constater que les paramètres estimés sont pas très différents pour le code de Maya et notre code, mais un écart sensible existe avec ceux du code non optimisé.

**Tableau 3.5 : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 2.**

Ordre 2	Paramètres estimés
Code non optimisé	0.2344, -0.0155
Code de Maya	0.2405, -0.001
Code optimisé	0.2498, 0

Le tableau 3.6 montre sous l'hypothèse isotrope, les paramètres estimés des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 3. Nous notons ici pour les trois codes, des paramètres avec des écarts plus un moins important en fonction du nombre de chiffre après la virgule.

**Tableau 3.6 : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 3.**

Ordre 3	Paramètres estimés
Code non optimisé	0.2232, 0, -0.0267
Code de Maya	0.2398, -0.0003, -0.0001
Code optimisé	0.2482, -0.0002, -0.005

Le tableau 3.7 montre sous l'hypothèse isotrope, les paramètres estimés des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 4. Pour les trois codes, le tableau nous présente des paramètres qui nos pas des écarts importants les uns des autres.

**Tableau 3.7 : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 4.**

Ordre 4	Paramètres estimés
Code non optimisé	0.2184, 0, 0, -0.0316
Code de Maya	0.2390, -0.0001, -0.0012, -0.0024
Code optimisé	0.2425, -0.0004, -0.0012, -0.0045

Le tableau 3.8 montre sous l'hypothèse isotrope, les paramètres estimés des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 5. Nous constatons que les paramètres ne changent presque pas.

**Tableau 3.8 : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 5.**

Ordre 5	Paramètres estimés
Code non optimisé	0.2184, 0, 0, -0.0316, 0
Code de Maya	0.2184, 0, 0, -0.0316, 0
Code optimisé	0.2355, -0.0003, -0.0010, -0.0100, -0.0005

Le Tableau 3.9 présente sous l'hypothèse isotrope, les paramètres estimés des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 6. Les paramètres estimés une fois de plus ne changent pas trop.

**Tableau 3.9 : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 6.**

Ordre 6	Paramètres estimés
Code non optimisé	0.2184, 0, 0, -0.0316, 0, 0
Code de Maya	0.2307, -0, -0.003, -0.0080, -0.0020, -0.0020
Code optimisé	0.2317, -0, -0.0028, -0.0080, -0.0018, -0.0010

Pour les ordres 7, 8 et 9, les tableaux 3.10, 3.11, 3.12 présentent sous l'hypothèse isotrope, les paramètres estimés qui ne changent presque pas pour les deux codes (Maya et optimisé) de la texture d'une image statique pour l'ordre 6.

**Tableau 3.10 : Paramètres estimés sous l'hypothèse isotrope des deux codes (Maya et optimisé) de la texture d'une image statique pour l'ordre 7.**

Ordre 7	Paramètres estimés
Code de Maya	0.2484, 0.00017, -0.00017, -0.00068, -0.0001, -0.0001, -0.0002
Code optimisé	0.2484, 0.0002, -0.0002, -0.0007, -0.0001, -0.0001, -0.0001

**Tableau 3.11 : Paramètres estimés sous l'hypothèse isotrope des deux codes (Maya et optimisé) de la texture d'une image statique pour l'ordre 8.**

Ordre 8	Paramètres estimés
Code de Maya	0.2331, 0.0020, -0.0015, -0.0059, -0.0012, -0.0005, -0.0006, -0.0003
Code optimisé	0.2348, 0.0018, -0.0018, -0.0067, -0.0012, -0.0006, -0.0008, -0.0003

**Tableau 3.12 : Paramètres estimés sous l'hypothèse isotrope des deux codes (Maya et optimisé) de la texture d'une image statique pour l'ordre 9.**

Ordre 9	Paramètres estimés
Code de Maya	0.1349, 0.0153, -0.0115, -0.0346, -0.0093, -0.0044, -0.0059,-0.0021, 0.0007
Code optimisé	0.1353, 0.0143, -0.0117, -0.0359, -0.0093, -0.0043, -0.0061, -0.0023, 0.0005

### 3.2.2- Paramètres et ordres du modèle anisotrope de la texture de l'image statique

Les tableaux 3.13, 3.14 et 3.15 montrent les paramètres du modèle autorégressif non causal 2D (code non optimisé), du code de Maya et de notre code optimisé pour une modèle anisotrope de la texture d'une image statique.

**Tableau 3.13- Paramètres estimés du code non optimisé sous l'hypothèse anisotrope de la texture d'une image statique jusqu'à l'ordre 6.**

VOISINAGE	PARAMETRES ESTIMES
$N_{s1}$	0.3372, 0.1265
$N_{s2}$	0.3373, 0.137, -0.0013, -0.0251
$N_{s3}$	0.3687, 0.0962, 0.0007, -0.0027, -0.0282, -0.0011
$N_{s4}$	0.1542, 0.0745, 0.0324, 0.0062, -0.0583, 0.0140, -0.0046, -0.0000.0291, -0.0154, -0.0463
$N_{s5}$	0.3347, 0.0723, 0.0023, 0.0003, -0.0697, 0.0006, -0.0001, -0.0032, - 0.0010, -0.0147, -0.0003, -0.0006
$N_{s6}$	0.2845, 0.0860, 0.0102, 0.0011, -0.0605, 0.0002, -0.0008, -0.0115, -0.0051, -0.0326, -0.0018, -0.0029, 0.0001, -0.0023

**Tableau 3.14- Paramètres estimés du code de Maya sous l'hypothèse anisotrope de la texture d'une image statique jusqu'à l'ordre 7.**

VOISINAGE	PARAMETRES ESTIMES
$N_{s1}$	0.2315, 0.2097
$N_{s2}$	0.3822, 0.1043, -0.0003, -0.004
$N_{s3}$	0.3600, 0.092, 0.0007, -0.0025, -0.0291, -0.001
$N_{s4}$	0.2352, 0.1000, 0.0162, -0.0007, -0.0316, -0.0001, -0.0003, -0.0384, -0.0078, -0.0550
$N_{s5}$	0.3347, 0.0723, 0.0023, 0.0003, -0.0697, 0.0006, -0.0001, -0.0032, -0.0010, -0.0147, -0.0003, -0.0006
$N_{s6}$	0.0244, 0.0095, 0.0055, 0.0020, -0.0052, 0.0007, -0.0006, -0.0042, -0.0030, -0.0021, -0.0014, -0.0020, 0, 0.003
$N_{s7}$	0.3601, 0.00543, -0.0090, 0.0002, -0.0142, -0.0001, -0.0009, -0.0070, -0.0046, -0.0171, -0.0026, -0.0041, 0.0001, 0.0001, 0, -0.00056 -0.0001, -0.0022

**Tableau 3.15- Paramètres estimés du code optimisé sous l'hypothèse anisotrope de la texture d'une image statique jusqu'à l'ordre 7.**

VOISINAGE	PARAMETRES ESTIMES
$N_{s1}$	0.2330, 0.2670
$N_{s2}$	0.3928, 0.1023, -0.0009, -0.0036
$N_{s3}$	0.3666, 0.0901, 0.0007, -0.003, -0.0272, -0.0009
$N_{s4}$	0.2449, 0.1236, 0.0162, -0.0000.0015, -0.0375, -0.0006, -0.0007, -0.0184, -0.0099, -0.0460
$N_{s5}$	0.0568, 0.0343, 0.0174, 0.0036, -0.0203, 0.0016, -0.0020, -0.0126, -0.0088, -0.0213, -0.0056, -0.0091
$N_{s6}$	0.0144, 0.0086, 0.0045, 0.0010, -0.0052, 0.0005, -0.0005, -0.0032, -0.0022, -0.0054, -0.0014, -0.0023, 0, 0.003
$N_{s7}$	0.3740, 0.00643, -0.0090, 0.0002, -0.0150, -0.0002, -0.0003, -0.0068, -0.0036, -0.0185, -0.0016, -0.0038, 0, 0, 0, -0.0005, -0.0001, -0.0019
$N_{s8}$	
$N_{s9}$	

### **Analyse des tableaux de l'hypothèse anisotrope d'une texture d'image statique**

Les coefficients les plus importants des trois tableaux 3.13, 3.14, 3.15 dans tous les ordres possibles sont les deux premiers. Ils représentent bien les quatre pixels les plus proches. Les valeurs de ces deux coefficients ne sont pas proches, ce qui signifie que la texture de la chaussée dans l'image statique n'est pas isotrope. Nous nous arrêtons à l'ordre 7 en raison de la capacité de calculs de l'ordinateur utilisé. Il existe un écart relatif entre les valeurs de coefficients obtenus de notre code optimisé (tableau 3.6) et ceux du code non optimisé (tableau 3.4) pour tous les ordres. Par contre, les valeurs du code optimisé sont proches des valeurs du code de Maya (tableau 3.5) pour tous les ordres.

Pour tous les tableaux 3.16, 3.17, 3.18, 3.20, 3.21, 3.22 de l'ordre 1 jusqu'à l'ordre 7 sous l'hypothèse anisotrope, il n'y a pas de concordance entre les différents paramètres estimés du code non optimisé et ceux des deux autres codes (Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 6. Le tableau 3.19 présente une corrélation existante entre ces trois codes. Par contre, pour tous les ordres (1 à 7), les paramètres estimés du code optimisé et de celui de Maya sont similaires et semblables avec des fortes corrélations.

Ainsi, dans le modèle anisotrope, les paramètres du code optimisé et du code de Maya ne changent pas trop pour tous les ordres obtenus, donc nous pourrions avoir une production de texture synthétisée semblable pour ces ordres, et différentes avec le code non optimisé. Seul l'ordre 4 présente pour les trois codes, une corrélation existante. On pourrait donc obtenir une production de texture légèrement semblable et proche de l'originale pour cet ordre.

**Tableau 3.16 : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 1.**

Ordre 1	Paramètres estimés
Code non optimisé	0.3372, 0.1265
Code de Maya	0.2315, 0.2097
Code optimisé	0.2330, 0.2670

**Tableau 3.17 : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 2.**

Ordre 2	Paramètres estimés
Code non optimisé	0.3373, 0.137, -0.0013, -0.0251
Code de Maya	0.3373, 0.137, -0.0013, -0.0251
Code optimisé	0.3928, 0.1023, -0.0009, -0.0036

**Tableau 3.18 : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 3.**

Ordre 3	Paramètres estimés
Code non optimisé	0.3687, 0.0962, 0.0007, -0.0027, -0.0282, -0.0011
Code de Maya	0.3600, 0.092, 0.0007, -0.0025, -0.0291, -0.001
Code optimisé	0.3666, 0.0901, 0.0007, -0.003, -0.0272, -0.0009

**Tableau 3.19 : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 4.**

Ordre 4	Paramètres estimés
Code non optimisé	0.1542, 0.0745, 0.0324, 0.0062, -0.0583, 0.0140, -0.0046, -0.0000.0291, -0.0154, -0.0463
Code de Maya	0.2352, 0.1000, 0.0162, -0.0007, -0.0316, -0.0001, -0.0003, -0.0384, -0.0078, -0.0550
Code optimisé	0.2449, 0.1236, 0.0162, -0.0000.0015, -0.0375, -0.0006, -0.0007, -0.0184, -0.0099, -0.0460

**Tableau 3.20 : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 5.**

Ordre 5	Paramètres estimés
Code non optimisé	0.3347, 0.0723, 0.0023, 0.0003, -0.0697, 0.0006, -0.0001, -0.0032, -0.0010, -0.0147, -0.0003, -0.0006
Code de Maya	0.3347, 0.0723, 0.0023, 0.0003, -0.0697, 0.0006, -0.0001, -0.0032, -0.0010, -0.0147, -0.0003, -0.0006
Code optimisé	0.0568, 0.0343, 0.0174, 0.0036, -0.0203, 0.0016, -0.0020,-0.0126, -0.0088, -0.0213, -0.0056, -0.0091

**Tableau 3.21 : Paramètres estimés sous l'hypothèse isotrope des trois codes (non optimisé, Maya et optimisé) de la texture d'une image statique pour l'ordre 6.**

Ordre 6	Paramètres estimés
Code non optimisé	0.2845, 0.0860, 0.0102, 0.0011, -0.0605, 0.0002, -0.0008, -0.0115, -0.0051, -0.0326, -0.0018, -0.0029, 0.0001, -0.0023
Code de Maya	0.0244, 0.0095, 0.0055, 0.0020, -0.0052, 0.0007, -0.0006, -0.0042, -0.0030, -0.0021, -0.0014, -0.0020, 0, 0.003
Code optimisé	0.0144, 0.0086, 0.0045, 0.0010, -0.0052, 0.0005, -0.0005,-0.0032, -0.0022, -0.0054, -0.0014, -0.0023, 0, 0.003

**Tableau 3.22 : Paramètres estimés sous l’hypothèse isotrope des deux codes (Maya et optimisé) de la texture d’une image statique pour l’ordre 7.**

Ordre 7	Paramètres estimés
Code de Maya	0.3601, 0.00543, -0.0090, 0.0002, -0.0142, -0.0001, 0.0009, -0.0070, -0.0046, -0.0171, -0.0026, -0.0041, 0.0001, 0.0001, 0, -0.00056 -0.0001, -0.0022
Code optimisé	0.3740, 0.00643, -0.0090, 0.0002, -0.0150, -0.0002, 0.0003, -0.0068, -0.0036, -0.0185, -0.0016, -0.0038, 0, 0, 0, -0.0005, -0.0001, -0.0019

### **3.3- Analyse globale des deux hypothèses isotrope et anisotrope de la texture d’une image statique**

Il ressort donc de notre analyse que l’estimation des paramètres pour la production d’une texture synthétisée de chaussée de l’image statique doit être isotrope pour les ordres 4, 5 et 6 et pour le modèle anisotrope l’ordre 4 car nous obtenons des paramètres qui ne changent presque pas pour les trois codes (non optimisé, Maya et optimisé). Puisque l’ordre 4 se répète pour les deux modèles isotrope et anisotrope, c’est donc le meilleur ordre à retenir. Mais c’est plus pratique de travailler avec le modèle isotrope puisqu’il demande moins de paramètres à estimer (4 paramètres pour le modèle isotrope alors qu’il faut estimer 10 paramètres pour le modèle anisotrope). En plus, lorsque l’ordre du système de voisinage est élevé (9 pour notre code optimisé), les paramètres estimés ne permettent pas la production de la texture d’image proche de l’originale.

### **3.4- Apport pédagogique et intérêt didactique du mémoire**

Nous venons de montrer qu’il est possible de produire une texture synthétisée proche de l’originale pour une image statique. Les résultats de la détection automatique des défauts d’une texture peuvent ainsi être améliorés. Ces défauts de structure sont étudiés dans le domaine du génie civil. Dans cette dernière partie, nous présentons l’apport pédagogique de notre travail ainsi que l’intérêt didactique.

#### **3.4.1- Apport pédagogique**

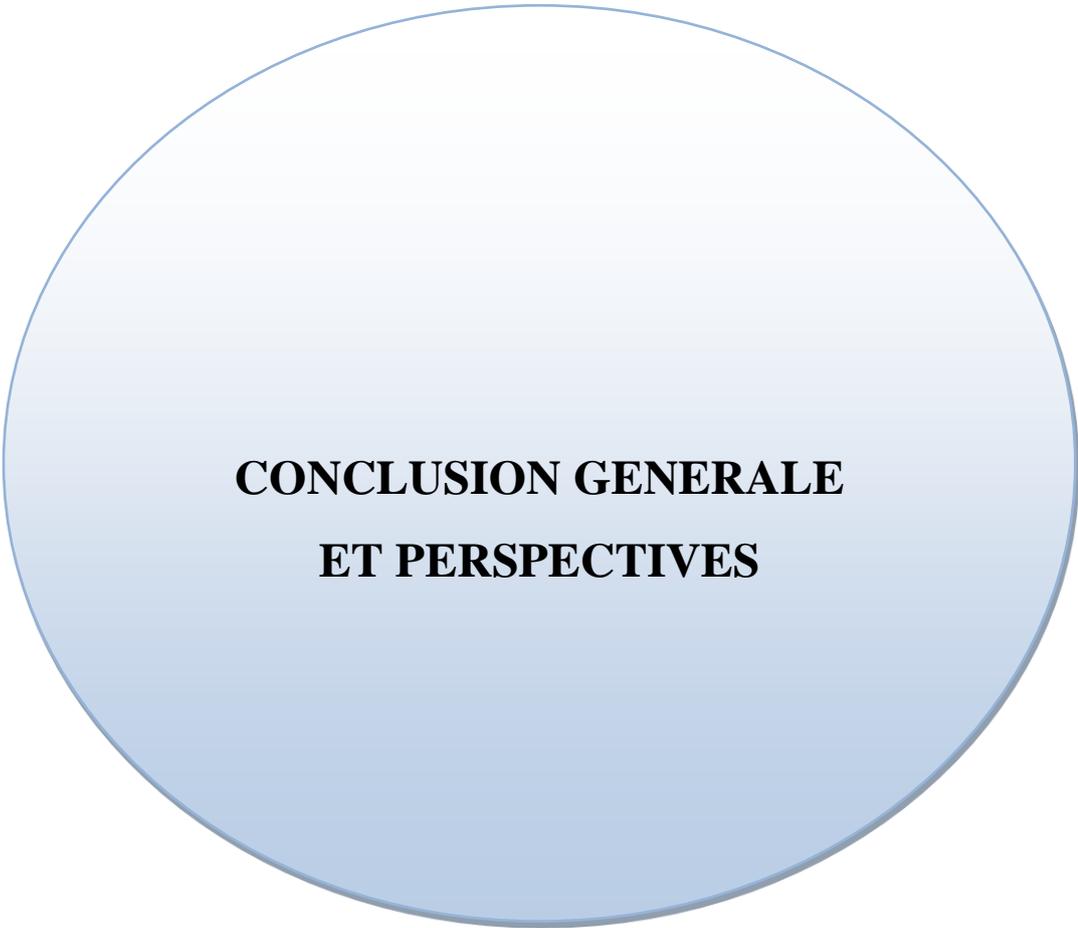
En travaillant dans le domaine du génie civil, nous montrons l’utilisation de la programmation en informatique dans les branches scientifiques. L’atteinte de nos objectifs nous a fait découvrir les branches du génie civil telles que (ingénierie de la structure de la matière, ingénierie du traitement des signaux et d’images, le génie mécanique appliqué aux ponts et routes. ...). Notons aussi pour ces branches que nous avons emmagasinés beaucoup de nouvelles connaissances.

De même, la compréhension de la détection automatique des défauts des structures sur la texture d'une image et des méthodologies utilisées a facilité l'appréhension de la logistique du travail de terrain (matériels utilisés) et l'environnement adéquat à la collecte des données (la zone, la température, le taux d'humidité de l'air, le taux d'ensoleillement, le profil longitudinal et le profil transversal des routes, la période de l'année,...).

### 3.4.2- Intérêt didactique du mémoire

Du point de vue didactique, l'approche utilisée permet de montrer l'interconnexion entre les mathématiques, l'informatique, les sciences de l'ingénieur et particulièrement le génie civil. Ce domaine scientifique est appliqué à l'aide du traitement d'images dans le cadre de l'inventaire des défauts de structure de la matière grâce à une texture d'image, défauts susceptibles de détruire rapidement la structure édifiée (chaussée ayant des nids de poules, des ponts qui se brisent, des immeubles qui s'effondrent, des éboulements de terrains, des éruptions volcaniques, ...).

**Conclusion :** Dans ce chapitre, nous avons présenté les résultats numériques obtenus de notre code optimisé pour les ordres de 1 à 9 sous les deux hypothèses isotrope et anisotrope pour la texture d'une image statique. Par la suite, nous avons comparé nos résultats numériques à ceux du modèle autorégressif non causal 2D (code non optimisé) et du code de Maya [Maya, 2011]. Pour l'hypothèse isotrope, le coefficient le plus important des trois tableaux 3.1, 3.2, 3.3 dans tous les ordres est le premier coefficient. Il représente bien la dépendance entre chaque pixel et les 4 pixels de voisinage du premier ordre qui sont les plus proches. Les valeurs de paramètres obtenus de notre code optimisé (tableau 3.3) sont proches des valeurs des tableaux 3.1 et 3.2 jusqu'à l'ordre 6. C'est aussi le cas des ordres 7,8 et 9 pour les tableaux 3.2 et 3.3. Pour l'hypothèse anisotrope, les coefficients les plus importants des trois tableaux 3.13, 3.14, 3.15 dans tous les ordres possibles sont les deux premiers. Ils représentent bien les quatre pixels les plus proches. Les valeurs de ces deux coefficients ne sont pas proches. Il existe un écart relatif entre les valeurs de coefficients obtenus de notre code optimisé (tableau 3.6) et ceux du code non optimisé (tableau 3.4) pour tous les ordres. Par contre, les valeurs du code optimisé sont proches des valeurs du code de Maya (tableau 3.5) pour tous les ordres. Pour tous les tableaux 3.16, 3.17, 3.18, 3.20, 3.21, 3.22 de l'ordre 1 jusqu'à l'ordre 7 sous l'hypothèse anisotrope, il n'y a pas de concordance entre les différents paramètres estimés du code non optimisé et ceux des deux autres codes (Maya et optimisé) de la texture d'une image statique jusqu'à l'ordre 6. Le tableau 3.19 présente une corrélation existante entre ces trois codes. Par contre, pour tous les ordres (1 à 7), les paramètres estimés du code optimisé et de celui de Maya sont similaires et semblables avec des fortes corrélations. Après une étude globale des deux hypothèses, nous avons trouvé que les paramètres estimés pour les modèles isotrope et anisotrope de la texture d'une image statique, l'ordre 4 est le meilleur pour une production de texture synthétisée identique proche de l'originale. Mais dans la pratique, on doit travailler avec le modèle isotrope. Nous présentons à la fin de ce chapitre, l'apport pédagogique et l'intérêt didactique de ce mémoire.



**CONCLUSION GENERALE  
ET PERSPECTIVES**

---

## CONCLUSION GENERALE

---

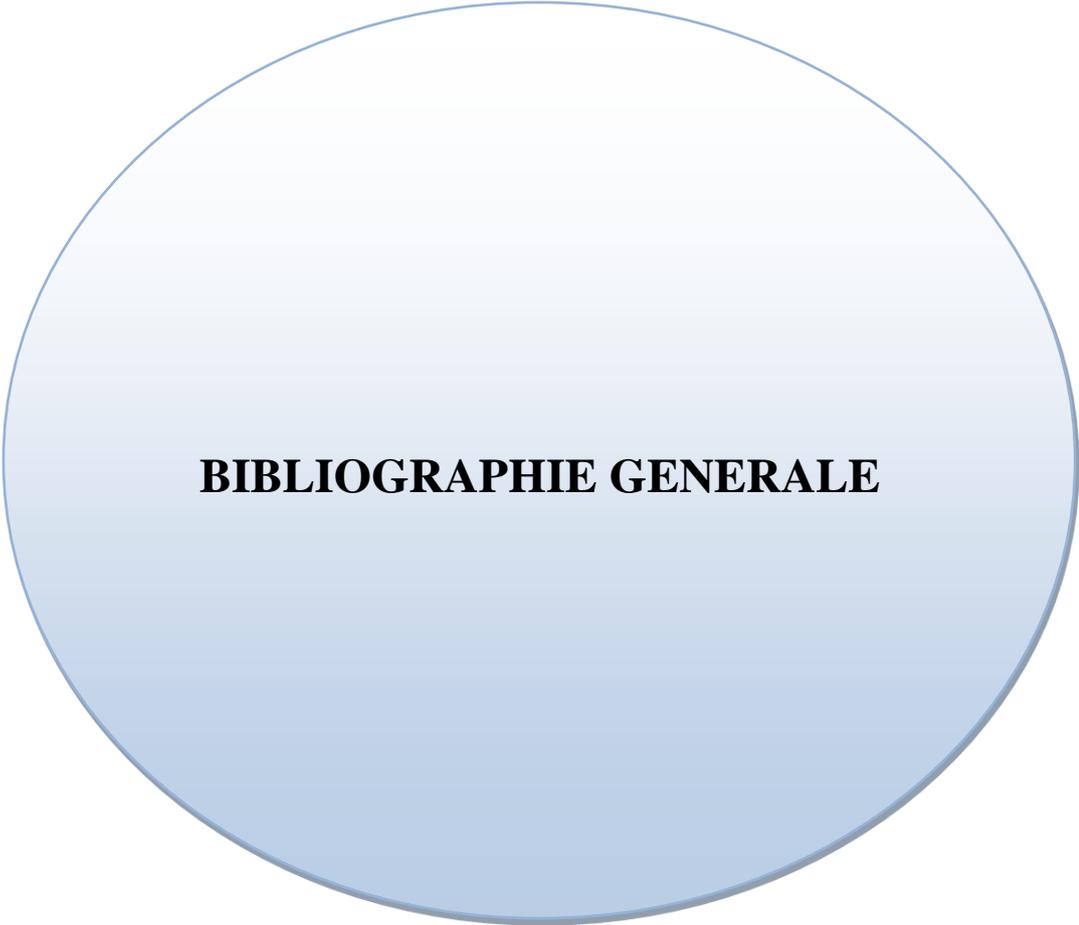
La prise en compte de la texture de la chaussée dans la détection des fissures sur image est un enjeu majeur pour l'optimisation de la gestion de l'infrastructure routière. Dans ce mémoire, l'objectif était d'estimer rapidement les paramètres nécessaires à la production d'une texture synthétisée à l'aide du code Matlab optimisé.

Pour situer le problème, nous avons analysé les connaissances existantes en matière d'estimation des paramètres à travers une étude bibliographique. Cette étude a permis de constater la complexité de l'estimation des paramètres à travers l'analyse spectrale 2D et le modèle autorégressif non causal 2D.

La phase la plus importante de ce travail a constitué à optimiser le code Matlab d'estimation des paramètres nécessaires pour la production de texture synthétisée d'image pour un système de voisinage non causal 2D d'ordre 9. Pour y arriver, nous avons opté pour la méthode du maximum de vraisemblance. Ceci nous a conduit à un problème d'optimisation non-linéaire. Grâce à la proposition de Kamima [Kamima, 2011] utilisée pour optimiser le code Matlab par un modèle autorégressif non causal 2D dans ses travaux pour un système de voisinage d'ordre 6, nous réalisons l'optimisation d'estimation des paramètres pour un système de voisinage d'ordre 9.

La phase du travail qui a suivi s'est appesantie sur les résultats et discussions. Ici, pour les hypothèses isotrope ou anisotrope de la texture d'une image statique, nous avons confronté les paramètres obtenus de notre code optimisé à ceux du modèle autorégressif non causal 2D (code non optimisé) et du code de Maya [Maya, 2011]. Nous avons obtenu une satisfaction relative et constaté que notre code optimisé est bien valide. L'analyse des confrontations nous montre que la meilleure texture synthétisée d'une image statique est celle d'un modèle isotrope pour l'ordre 4 car nous obtenons des paramètres qui ne changent presque pas pour les trois codes. En plus, nous constatons que les paramètres estimés pour des ordres élevés de voisinage non causal ne permettent pas la production d'une texture d'image statique proche de l'originale. A la fin de cette partie, nous donnons l'apport pédagogique et l'intérêt didactique de notre mémoire.

Pour un travail prochain, nous envisageons mener une étude sur la synthèse des textures des images statique et dynamique (image prise par un véhicule d'acquisition d'images en circulation normale) avec notre code optimisé. Nous travaillerons sous les hypothèses isotrope et anisotrope. Notre objectif sera de comparer les spectres des textures et les images synthétisées à ceux des codes non optimisé et de Maya. Un travail sur une modélisation 3D est aussi envisagé.



**BIBLIOGRAPHIE GENERALE**

---

## BIBLIOGRAPHIE GENERALE ET PERSPECTIVES

---

**CHAMBON, J et MOLIARD, J. M.** 2010. Rapport technique. Détection automatique de defaults dans des images de chaussées. Volume : 8 – 18.

**MAYA, A.** 2011. Caractérisation de la texture des chaussées par un modèle autoregressif non causal 2D (ARNC 2D). Mémoire de master Recherche. Ecole Centrale de Nantes, 1-Rue de la Noë – 44321 Nantes. Volume : 1 – 51.

**KAMINA, B.** 2011. Localisation à haute résolution des défauts mécaniques par interférométrie holographique. Mémoire de Magister. Université Mouloud Mammeri – Tizi – Ouzou, Algérie. Volume : 48 – 60.

**CHEPELLAPPA, R. et KASHYAP, R.**1983. Estimation of choice of Neighbors in Spatial Interaction Models of Images. Dans IEEE Trans Information Theory. Volume : 12 – 40.

**PAQUIS. S.** 1996. Reconnaissance des Surfaces de Chaussées par Analyse de Texture Image. Thèse de doctorat. Université Jean Monnet, Saint – Etienne, France. Volume : 41 – 90.

**OUKIL, A.** 2007. Analyse variographique, modélisation et synthèse de textures appliqués aux images numériques. Thèse de doctorat, université des Sciences et de la Technologie « HOUARI BOUMEDIENE », Algérie, 2007. Volume : 15 – 42.

**LAWRENCE, S. et MARPLE, Jr.** 1987. Chapitre Classical spectral estimation. Digital Spectral analysis with application. Prentice Hall Signal Processing Series. Volume : 8 – 20.

**TUGNAIT, J. K.** 1993. Texture Synthesis Using Assymmetric 2D non Causal AR Models. IEEE Processing Workshop on Higher – Order Statistics. Volume : 22 – 68.

**SUTAONE, M. ; JOSHI, M. ; Bartakke, P.** 2009. Texture representation using Autoregressive models. International conference on Advances in Computational Tools for Engineering Applications. Volume : 7 – 54.

**DEGUCHI, K.** 1986. Two – dimensional autoregressive model for analysis and synthesis of gray level texture. Science on form. Volume : 1- 5

**BESAG, J.** 1974. Spatial interaction and the statistical analysis of lattice systems. Journal of the the Royal statistical Society, Series B. Volume : 15 – 25.

**RANDEN, T. et HSØY, J.** 1994. Least squares image texture analysis and synthesis. Hog – Skolen in St Avenger Working. Volume : 12 - 18.

**BESAG, J. et MORAN, P.** 1975. On the Estimation and Testing of Spatial Interaction in Gaussian Lattice Processes. Biometrical. Volume : 9 – 19.

**CHEPELLAPPA, R. et KASHYAP, R.** 1985. Texture Synthésisusing 2-d non causal autoregressive models. Volume : 5 – 27.