

الجمهورية الجزائرية الديمقراطية الشعبية  
République algérienne démocratique et populaire  
وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique  
المركز الجامعي لعين تموشنت  
Centre Universitaire Belhadj Bouchaib d'Ain-Temouchent  
Institut de Technologie  
Département de Génie Electrique



Projet de fin d'études  
Pour l'obtention du diplôme de Master en :  
Domaine: TECHNOLOGIE  
Filière : Télécommunication  
Spécialité : Réseaux de Télécommunication

Thème

Application d'une Nouvelle Approche de Filtrage Non-Linéaire d'image  
Numérique sur un système embarqué

**Présenté Par :**

- 1) MALEK Mokhtaria
- 2) MALEK Naouel

**Devant le jury composé de :**

Souiki Sihem	MCB	C.U.B.B (Ain Temouchent)	Présidente
BOUTKHIL Malika	MAA	C.U.B.B (Ain Temouchent)	Encadrante
BEDDAD Boucif	Doctorant	U.D.L.S.B.A (sidi bel abbes)	Co-encadrant
AYECHE Choukria	Pr	C.U.B.B (Ain Temouchent)	Examineur

*Année universitaire 2018/2019*

# بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

" وَقُلْ اَعْمَلُوا فَسَيَرَى اللَّهُ عَمَلَكُمْ وَرَسُولُهُ

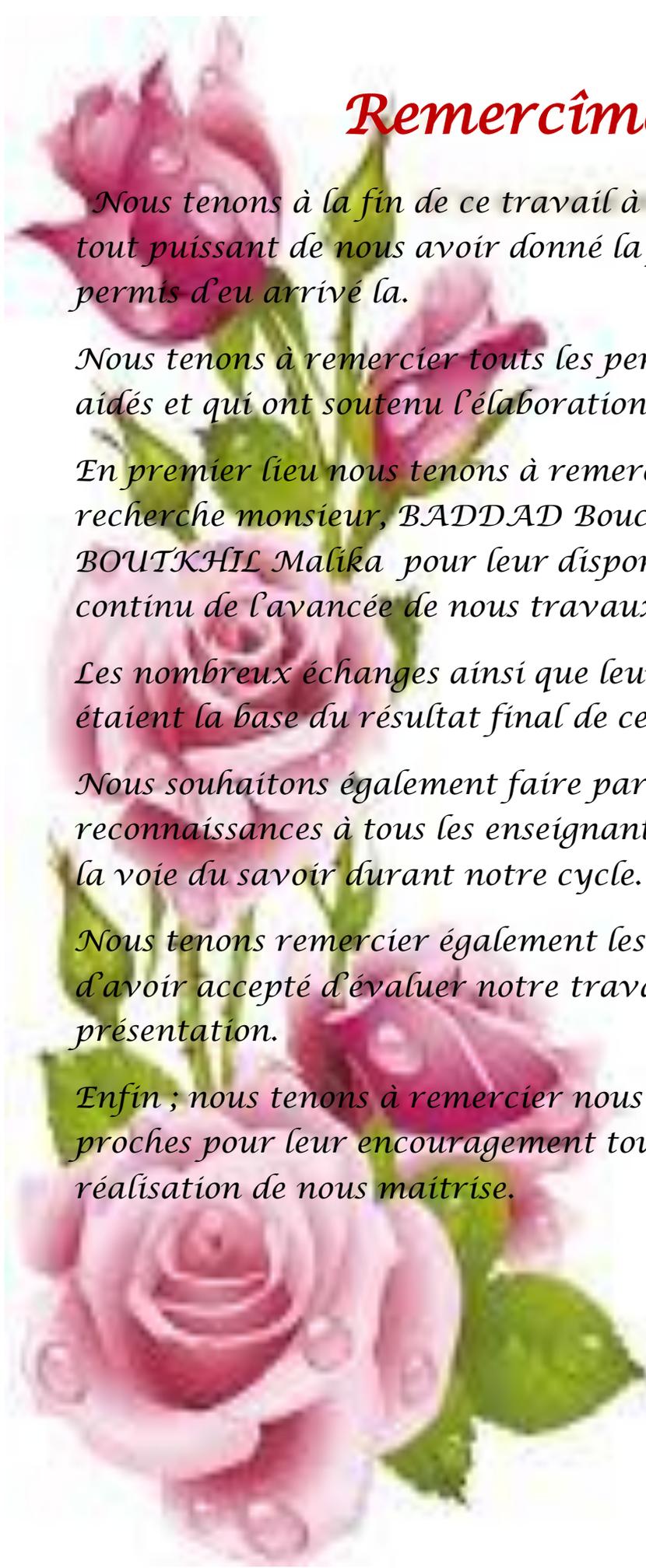
وَالْمُؤْمِنُونَ وَسَتُرَدُّونَ إِلَىٰ عَالَمِ الْعَذَابِ وَالشَّهَادَةِ

-التوبة: 105 -

فَيُنَبِّئُكُمْ بِمَا كُنْتُمْ تَعْمَلُونَ "

ويقول العماد الأصفهاني

{إني رأيت أنه لا يكتب أحد كتابا في يومه إلا قال فيه عنه: لو غير هذا لكان أحسن، ولو زيد لكان يستحسن ، ولو قدم هذا لكان أفضل، ولو ترك هذا لكان أجمل، وهذا من أعظم العبر، وهو دليل على استيلاء النقص على جملة البشر}.

A bouquet of pink roses with green leaves is positioned on the left side of the page. The roses are in various stages of bloom, with some fully open and others as buds. The background is a light, soft-focus white.

## Remercîment

*Nous tenons à la fin de ce travail à remercier ALLAH le tout puissant de nous avoir donné la foi et de nous avoir permis d'en arriver là.*

*Nous tenons à remercier toutes les personnes qui nous ont aidés et qui ont soutenu l'élaboration de ce mémoire.*

*En premier lieu nous tenons à remercier notre directeur de recherche monsieur, BADDAD Boucif et madame, BOUTKHIL Malika pour leur disponibilité et leur suivi continu de l'avancée de nos travaux.*

*Les nombreux échanges ainsi que leurs précieux conseils étaient la base du résultat final de ce travail.*

*Nous souhaitons également faire part de nos reconnaissances à tous les enseignants qui nous ont éclairés la voie du savoir durant notre cycle.*

*Nous tenons remercier également les membres du jury d'avoir accepté d'évaluer notre travail et assister à sa présentation.*

*Enfin ; nous tenons à remercier nos camarades et nos proches pour leur encouragement tout au long de la réalisation de notre maîtrise.*

## *Dédicace*

*Ce que personne ne peut compenser les sacrifices qu'ils ont consentis pour mon éducation et pour mon bien être qu'ont j'amaï cessé de me soutenir moralement et matériellement pour que je puisse mes études et avoir une bonne formation et surtout être le meilleur :*

*A mes très chers parents mon père et ma mère qui à toujours était à mes cotés.*

*A mes frères Mohamed, Bigad, Abdenour, Abdelkader.*

*A mon encadreur monsieur BEDDAD Boucif et madame BOUTKHIL Malika qui m'a facilité le chemin de mes études par ses précieux conseils et orientation bénéfiques.*

*A toute ma grande famille.*

*A mon binôme et mon cousine qui était à coté de moi au long de mon école primaire jusqu'un ce moment.*

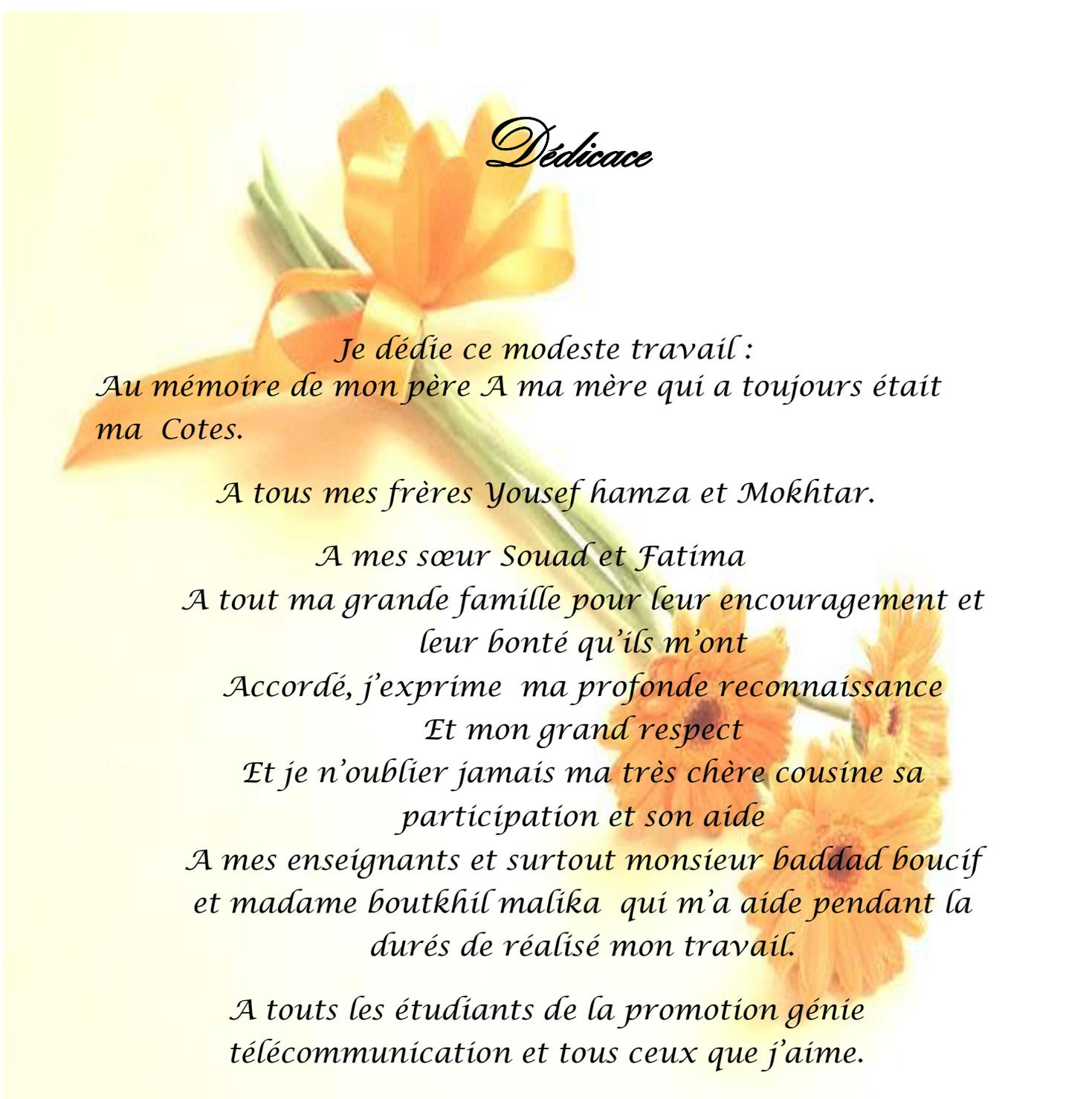
*A tous les étudiants de la promotion Gene télécommunication.*

*A ceux qui sont la source de mon inspiration et mon courage.*

*A tous ceux qui voulaient et me souhaitaient du succès.*

*Malek naouel*





## *Dédicace*

*Je dédie ce modeste travail :  
Au mémoire de mon père A ma mère qui a toujours était  
ma Cotes.*

*A tous mes frères Yousef hamza et Mokhtar.*

*A mes sœur Souad et Fatima  
A tout ma grande famille pour leur encouragement et  
leur bonté qu'ils m'ont  
Accordé, j'exprime ma profonde reconnaissance  
Et mon grand respect  
Et je n'oublier jamais ma très chère cousine sa  
participation et son aide  
A mes enseignants et surtout monsieur baddad boucif  
et madame boutkhil malika qui m'a aide pendant la  
durés de réalisé mon travail.*

*A tous les étudiants de la promotion génie  
télécommunication et tous ceux que j'aime.*

*« MALEK MOKHTARIA »*

# SOMMAIRE

Résumé

Liste des Figures

Liste des tableaux

Introduction générale ..... 01

## **Chapitre I: Etat de l'art des Méthodes de Filtrage d'images**

**I.1** Introduction.....04

**I.2** Définition de l'image.....04

**I.2.1** image Numérique.....04

**I.2.2** Construction d'une image Numérique.....05

**I. 3** Type et Codage de l'image Numérique.....06

**I.3.1** Image binaire .....06

**I.3.2** Image en niveau de gris.....06

**I.3.3** Image couleur (R G B).....07

**I.4** Caractéristiques de l'image Numérique.....08

**I.4.1** Pixel.....08

**I. 4.2** Taille de l'image.....08

**I.4.3** Résolution.....09

**I.4.4** Luminance.....09

**I.4.5** Contraste.....09

**I.4.6** Histogramme.....09

**I.4.7** Contour et texture .....10

**I.4.8** bruit.....10

**I.4.9** Niveau de Gris.....10

**I.5** Les différents Formats d'images.....11

**I.5.1** Format Vectorielle.....11

<b>I.5.2</b> Format Matricielle.....	11
<b>I.6</b> Obtention des données images.....	11
<b>I.7</b> Système de traitement d'image .....	12
<b>I.8</b> Traitement numérique des images.....	13
<b>I.8.1</b> Filtrage Numérique .....	13
<b>I.8.1.1</b> Filtres linéaires .....	13
<b>I.8.1.2</b> Filtres non linéaire.....	14
<b>I.9</b> Domaines d'application.....	14
<b>I.10</b> Conclusion.....	14
 <b>Chapitre II : Développement d'une technique efficace de réduction du bruit gaussien et impulsif.</b>	
<b>II.1</b> Introduction.....	16
<b>II.2</b> Dégradation d'image.....	16
<b>II.2.1</b> Bruit gaussien .....	16
<b>II.2.2</b> Bruit impulsif.....	17
<b>II.3</b> Débruitage d'image.....	17
<b>II.4</b> Critère d'analyse pour le débruitage d'image.....	17
<b>II.4.1</b> Critère Subjective. ....	17
<b>II.4.2</b> Critère objective .....	17
<b>II.5</b> Méthodes de filtrage non-linéaire d'image .....	18
<b>II.5.1</b> Méthode de EFPGF.....	18
<b>II.5.2</b> Méthode de MAUTMPF.....	20
<b>II.5.3</b> Filtre NAGAO Modifié.....	22
<b>II.5.4</b> Filtre Médian Standard.....	23
<b>II.5.5</b> Nouvelle Approche Proposé.....	24
<b>II.6</b> Résultats de Débruitage.....	25
<b>II.7</b> Représentation des résultats.....	28

<b>II.8</b> Etude Subjective.....	31
<b>II.8.1</b> Bruit Gaussien.....	31
<b>II.8.2</b> Bruit impulsif.....	32
<b>II.9</b> Conclusion.....	33
 <b>Chapitre III : étude générale de la Plate-forme TMS320C6713DSK</b>	
<b>III.1</b> Introduction .....	35
<b>III.2</b> Présentation de TMS320C6713.....	35
<b>III.2.1</b> La famille TMS320C .....	35
<b>III.2.2</b> Les différentes Familles de C6x .....	36
<b>III.2.2.1</b> TMS320C62X.....	36
<b>III.2.2.2</b> TMS320C64X.....	36
<b>III.2.2.3</b> TMS320C67X.....	36
<b>III.2.3</b> Caractéristique principales du TMS320C6713 .....	37
<b>III.2.4</b> Architecture du TMS320C6713 .....	37
<b>III.2.4.1</b> Unité centrale de traitement (CPU).....	37
<b>III.2.4.1.1</b> Registres.....	39
<b>III.2.4.2</b> Les périphériques du TMS320C6713.....	40
<b>III.2.4.3</b> La structure de la mémoire .....	40
<b>III.3</b> Présentation du Code Composer Studio.....	41
<b>III.3.1</b> environnement de développement DSP.....	42
<b>III.4</b> Conclusion .....	43
 <b>Chapitre IV : Application de l’algorithme proposé sur TMS320C6713DSK</b>	
<b>IV.1</b> Introduction.....	45
<b>IV.2</b> Présentation du Model Simulink Proposé pour l’implémentation.....	45
<b>IV.3</b> Simulation et résultats obtenus à l’aide de Matlab Simulink.....	46
<b>IV.4</b> Implémentation des Algorithmes sur TMS320C6713 .....	47

<b>IV.4.1</b> Instruction pour la connexion de la carte .....	47
<b>IV.4.2</b> Les paramètres de configuration.....	48
<b>IV.4.3</b> Création d'un projet en « C/C++ » à partir d'un model Simulink.....	49
<b>IV.4.4</b> Différentes opérations pour l'implémentation surTMS320C6713 DSK.....	50
<b>IV.4.5</b> Résultats obtenu à l'aide du Code Composer Studio CCS .....	51
<b>IV.4.5.1</b> l'image original à l'aide du CCS .....	52
<b>IV.4.5.2</b> l'image bruitée à l'aide du CCS .....	52
<b>IV.4.6</b> conclusion .....	53
Conclusion général.....	55

## **Résumé :**

Le filtrage et le rehaussement d'images est un champ important qui est employé infiniment dans le traitement d'images, où des images sont corrompues par différents types de bruit. Pour cela, des méthodes et des techniques doivent être employées pour supprimer ces bruits. Dans notre étude cinq méthodes de filtrages sont utilisés, à savoir : Médian Standard, NAGAO Modifié, EFPGE, MAUTMPF, une méthode proposé. Sur différentes natures d'images affectées par deux types de bruits impulsif et gaussien afin d'étudier leur performance. Les résultats obtenus sont présentés et discutés en termes de qualité visuelle et implémenté sur une carte TMS320C6713 DSK à base de DSP.

**Mots clés :** Traitement numérique d'image, bruit, filtrage, TMS320C6713 DSK, Code Composer Studio CCS, MATLAB, DSP

## **ملخص:**

تعد معالجة الصور الرقمية وتحسينها من أهم الحقول المهمة والمستخدمة في العديد من الفروع الخاصة بمعالجة الصور, ولأهمية الصور في عدة مجالات في الحياة, ولكونها تتعرض عند التعامل معها أو إرسالها عبر قناة معينة لكثير من الضوضاء الداخلة إليها. في دراستنا نستخدم خمس طرائق و تقنيات على النحو التالي: المتوسط, مرشح ناجارو, والتقنية المقترحة على أنواع مختلفة من الصور ذات ضوضاء مختلفة لدراسة أدائها يتم عرض النتائج في الجداول الإحصائية والرسوم البيانية وسوف ننظر في جودة الصورة, تم نقوم بتطبيق هذه التقنيات على DSPTMS320C6713 على أساس DSP

**الكلمات المفتاحية :** معالجة الصور, TMS320C6713 DSK, DSP Code Composer Studio CCS, ضوضاء, ترشيح

## **Abstract :**

Image filtering and enhancement is an important field that is used infinitely in image processing, where images are corrupted by many kinds of noises. Different methods and techniques should be used to remove these noises. In our study five methods are used such as: Median, Nagao, EFPGE, MAUTMPF and proposed technique on different kinds of images, affected by different noises in order to study their performance. The results obtained are presented and discussed in terms of the visual quality. And implement these algorithms on a TMS320C6713 DSK board based on DSP.

**Keywords:** image processing, noise, denoising, TMS320C6713 DSK, CCS Code Composer Studio, MATLAB, digital signal processor DSP.

# Liste Des Figures

## Chapitre(I) : Etat de l'art du traitement des images

Figure(01) : représentation du pixel.....	05
Figure(02) : exemple de quantification sur 8bit, 4bit, 2bit.....	06
Figure(03) : image en noire et blanc.....	06
Figure(04) : image en niveau de gris.....	07
Figure(05) : image en couleur.....	07
Figure(06) : variation du nombre de pixel en réalité lors de l'affichage à l'écran la taille est divisé par 4 à chaque étape.....	08
Figure(07) : tracé de l'histogramme .....	10
Figure(08) : Echantillonnage d'une image.....	12
Figure(09) : Quantification d'une image.....	12

## Chapitre II : Développement d'une technique efficace de réduction du bruit gaussien et impulsif.

Figure(01) : Architecture du système.....	19
Figure(02) : Organigramme d'algorithme MAUTMPF.....	21
Figure(03) : Les 9 fenêtres du filtre NAGAO.....	22
Figure(04) : Mise en œuvre du filtre Médian.....	23
Figure(05) : Les zones de voisinage .....	24
Figure(06) : Représentation de PSNR des images cameraman débruitées en fonction de la variance de bruit gaussien par différentes techniques de débruitage .....	28
Figure(07) : Représentation de PSNR des images cameraman débruitées en fonction de la pourcentage de bruit impulsif par différentes techniques de débruitage.....	28
Figure(08) : Représentation de SNR des images cameraman débruitées en fonction de la pourcentage de bruit gaussien par différentes techniques de débruitage.....	29
Figure(09) : Représentation de SNR des images cameraman débruitées en fonction de la pourcentage de bruit impulsif par différentes techniques de débruitage.....	29
Figure(10) : Représentation de MSE des images cameraman débruitées en fonction de la pourcentage de bruit gaussien par différentes techniques de débruitage.....	30

Figure(11) : Représentation de MSE des images cameraman débruitées en fonction de la pourcentage de bruit impulsif par différentes techniques de débruitage.....	30
Figure(12) : Image cameraman bruitée par bruit gaussien avec une variance $\sigma=0.05$ d'ébruitée par déferent techniques.....	31
Figure(13) : Image cameraman bruitée par bruit impulsif avec une pourcentage 50% d'ébruitée par déferent techniques.....	32

### **Chapitre III : étude générale de la Plate-forme TMS320C6713DSK**

Figure(01) : La famille 320C .....	35
Figure(02) : Bloc diagramme simplifie de la famille TMS320C67xx.....	37
Figure(03) : Structure du système du développement du TMS320C6713.....	41
Figure(04) : Capture d'écran du code Composer Studio.....	43

### **Chapitre IV : Application de l'algorithme proposé sur TMS320C6713DSK**

Figure (01) : Architecture du Model Simulink proposé .....	46
Figure(02) : Le Block du C6713.....	47
Figure (03): Vérification de la bonne marche de la carte DSK.....	48
Figure (04) : Fenêtre de la configuration des Paramètres.....	49
Figure (05) : Paramètre du Link for CCS.....	49
Figure(06) : Le passage du Simulink au « C/C++ ».....	50
Figure (07) : Création du projet Filtre Non-Linéaire développé sur CCS.....	51
Figure (08): Capture d'écran de l'image original à l'aide du CCS. ....	52
Figure (09) : Capture d'écran de l'image bruitée à l'aide du CCS.....	52
Figure (10) : Capture d'écran CCS de l'image résultante (après l'utilisation du filtre proposé).....	53

# Liste Des Tableaux

## **Chapitre(I) : Etat de l'Art des Méthodes de Filtrage d'images**

Tableau (01) : Table histogramme.....	10
---------------------------------------	----

## **Chapitre II : Développement d'une technique efficace de réduction du bruit gaussien et impulsif.**

Tableau (01) : Résultat de débruitage (PSNR [dB]) sur différentes images de tests par toutes les méthodes de débruitage étudiées.....	25
---	----

Tableau (02) : Résultat de débruitage (SNR) sur différentes images de tests par toutes les méthodes de débruitage étudiées.....	26
---	----

Tableau (03) : Résultat de débruitage (MSE) sur différentes images de tests par toutes les méthodes de débruitage étudiées.....	27
---	----

## **Chapitre III : étude générale de la Plate-forme TMS320C6713DSK**

Tableau (01) : Principale caractéristiques des trois plateformes de DSP.....	36
--	----

Tableau (02) : Bloc diagramme simplifié de la famille TMS320C6713.....	38
--	----

Tableau (03) : Le rôle des unités fonctionnelles.....	39
---	----

## **Chapitre IV : Application de l'algorithme proposé sur TMS320C6713DSK**

Tableau (01) : Différents Simulink Blocksets Utilisés.....	45
--	----

# Abréviation et Acronyme

**RGB** : image couleur (rouge, vert, bleu).

**3D** : trois dimensions (x, y, z).

**MBP** : (bitmap) image matricielle.

**JPEG** : Joint Photographique Experts Group.

**MSE** : Mean Squar Error (erreur quadratique moyenne.)

**PSNR** : Peak Signal to Noise Ratio (rapport signal sur bruit en pic.)

**SNR** : rapport signal sur bruit.

**FFPGF** : Enhanced Fuzzy Peer Group Filter (Filter Pair Group Flue Amélioré.)

**MAUTMPF**: Morphological based Adaptive Unsymmetrical Trimmed Mid-Point Filters.

**DSP**: digital signal processors.

**DSK**: TMS320C6713 starts kit.

**GPS**: le system de positionnement global.

**VLIW**: very-long-instruction word.

**JTAG**: joint team action group.

**CPU**: central processor unit.

**DMA**: direct memory access.

**EDMA**: Enhanced direct memory access.

**HPI**: host port interface (interface central port).

**McBSP** : Multichannel Buffered serial port.

**CAN** : convertisseur Analogique Numérique.

**CNA** : Convertisseur Analogique Numérique.

**LED** : light –emitting diodes (diode électro-luminescente ).

**DIP**: Dual in-line pickage (double paquet en ligne).

**DRAM** : Dynamic Random-Access Memory (Mémoire dynamique Access Aléation).

**CCS** : Code Composes Studio.

# Introduction Générale

---

## Introduction Générale

Les images numériques jouent un rôle important dans les applications de la vie quotidienne telles que la télévision par satellite, l'imagerie par résonance magnétique, la tomographie par ordinateur, etc., ainsi que dans les domaines de la recherche et de la technologie tels que les systèmes d'information géographique et l'astronomie. Technique dans laquelle les données d'une image sont numérisées et diverses opérations mathématiques sont appliquées aux données, généralement avec un ordinateur numérique, afin de créer une image améliorée qui soit plus utile ou plus agréable pour un observateur humain, ou pour effectuer certaines opérations. Les tâches d'interprétation et de reconnaissance généralement effectuées par des humains, également appelées traitement d'image.

Le traitement d'image est un domaine très vaste. En effet, on peut parler de restauration, de compression ou encore de modification, mais l'amélioration de l'image est essentiellement obtenue par ce que l'on appelle une opération de filtrage. De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes.

Même avec les technologies les plus récentes cependant, il existe de nombreuses situations où les dégradations de la scène originale sont trop importantes pour l'application visée. Ceci peut être dû, par exemple, aux conditions difficiles d'acquisition rencontrées en imagerie médicale, en astronomie ou dans le domaine militaire entre autres. Lorsqu'il est impossible d'améliorer le capteur, soit pour des raisons de coût, soit à cause de limites physique, le filtrage des images devient nécessaire pour améliorer la qualité visuelle de celles-ci, aussi pour augmenter les performances des traitements ultérieurs.

Dans ce cadre s'inscrit ce projet qui porte sur le développement d'une application permettant de concevoir une méthode de filtrage Non-Linéaire d'images numériques.

Le premier chapitre est dédié au cadre d'application de ce mémoire. Il introduit les termes et concepts essentiels pour appréhender l'analyse et le traitement numérique d'image et les méthodes de filtrage non-linéaire.

Dans le chapitre deux nous présenterons une nouvelle approche efficace de réduction de bruit Et dans le troisième chapitre nous procéderons à la description de la partie logicielle (l'environnement de développement intègre CCS) et la partie matérielle sous forme d'une étude générale sur le TMS320C6713 DSK.

Le chapitre quatre est consacré à la présentation des résultats par l'algorithme que nous avons implémenté. Nous exposons également les différents cas particuliers que nous avons rencontrés à l'issue de cette implémentation. Nous terminons ce travail par une conclusion générale.

# CHAPITRE I :

**Etat de l'Art des Méthodes de Filtrage d'images**

## I.1 Introduction

Le filtrage d'image est un processus destiné à améliorer l'aspect visuel d'une image. L'objectif principal de l'amélioration des images est de traiter une image donnée de sorte que le résultat soit plus approprié que l'image d'origine pour une application spécifique. Il accentue ou rend plus nettes les caractéristiques de l'image, telles que les bords, les contours ou le contraste, afin de rendre un affichage graphique plus utile pour l'affichage et l'analyse. L'amélioration n'augmente pas l'information inhérente contenue des données, mais cela augmente la plage dynamique des caractéristiques choisies afin qu'elles puissent être facilement détectées. La plus grande difficulté en matière d'amélioration d'image consiste à quantifier le critère d'amélioration; par conséquent, un grand nombre de techniques d'amélioration d'image sont empiriques et nécessitent des procédures interactives pour obtenir des résultats satisfaisants. Les méthodes d'amélioration des images sont basées sur des techniques de domaine spatial ou fréquentiel. Les méthodes d'amélioration peuvent être classées en deux catégories:

- Méthodes de domaine spatial.
- Méthodes du domaine de fréquence.

Les méthodes de domaine spatial modifient directement les pixels de l'image pour obtenir l'amélioration souhaitée dans le domaine spatial. Les méthodes de domaine de fréquence effectuent les opérations d'amélioration de la transformée de Fourier discrète (DFT) d'une image dans le domaine de fréquence.

## I.2 Définition de l'image

L'image est une représentation visuelle d'une personne ou d'un objet par la peinture, dessin, photographie, film...etc. C'est aussi un ensemble structuré d'information qui après affichage sur l'écran, ont une signification pour l'œil humain. [1]

### I.2.1 Image Numérique:

L'image numérique est l'image dont la surface est divisée en élément de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter. [1]

Une image numérique est définie comme un signal fini bidimensionnel échantillonné à valeur quantifiée dans un certain espace de couleurs. Elle est constituée de points (pixel). Image numérique se dit (digital image) en anglais. Image numérique est définie par :

- Le nombre de pixels qui les composent en largeur et en hauteur.
- L'étendue de teintes de gris ou des couleurs que peut prendre chaque pixel. [2]

Toutes les données correspondant aux informations contenues dans l'image sont structurées d'une certaine façon afin de permettre leur stockage. Il existe un grand nombre de formats d'images, tous ces formats ne correspondent ni plus ni moins qu'à une structuration particulière des données concernant l'image.

L'image numérique, désignée aussi par le terme scène, possède un repère comme indiqué En figure I.1, il est différent de celui d'une fonction mathématique. Elle a une hauteur (H) Et une largeur (L).

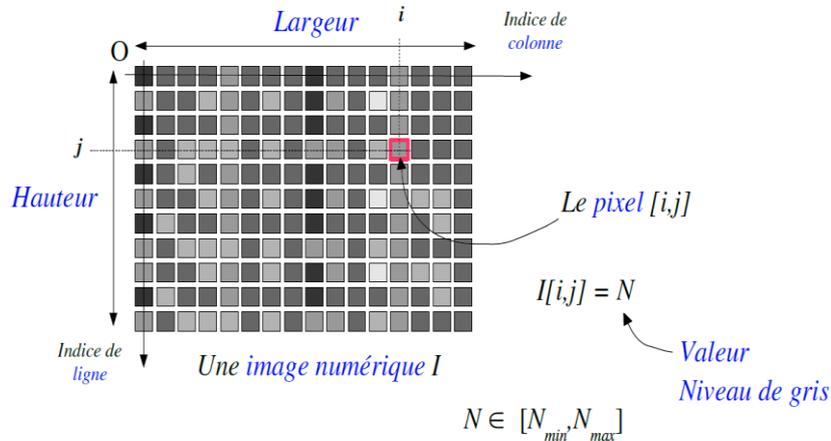
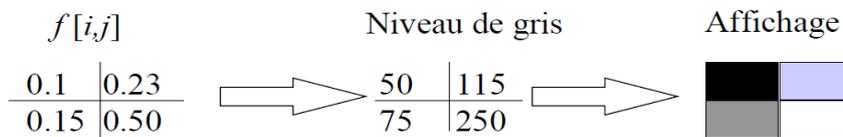


Figure I.1 : Représentation du pixel

**I.2.2 Constriction d'une image numérique:**

Pour visualiser une image, on remplit une région rectangulaire (Pixel) avec un niveau de gris (ou de couleur) correspondant à la valeur du pixel. En général les niveaux de gris (ou de couleur) utilisé pour la visualisation sont compris entre 0 et 255 (longueur fixe sur 8 bits).



- . CAN sur les systèmes d'acquisition d'images.
- . Codage de la valeur de chaque pixel sur N bits (En général 8 bits).

La quantification est la seconde étape nécessaire à la numérisation des signaux. Succédant à l'échantillonnage-blocage, elle permet le traitement numérique ou la mémorisation du signal. Son rôle est d'affecter une valeur de résolution finie à un échantillon dont l'amplitude est en théorie infiniment précise si l'on fait abstraction du bruit de fond propre au signal.

Quantifier un échantillon, c'est arrondir sa valeur à celle de l'échelon le plus proche sur une grille de niveaux. Lorsque les échelons sont à pas constant, la quantification est uniforme. La quantification est une opération non conservatrice car la précision originale ne peut être retrouvée après arrondi. Elle peut être vue comme la superposition d'une composante aléatoire sur l'amplitude de chaque échantillon original.



Figure I.2 : Exemple de quantification sur 8 bits, 4 bits, 2 bits

### I.3 Type et codage de l'image

L'image est une représentation visuelle d'une personne ou d'un objet par la peinture, dessin, photographie, film...etc. C'est aussi un ensemble structuré d'information qui après affichage sur l'écran, ont une signification pour l'œil humain. [1]

#### I.3.1 Image binaire:

Dans l'image binaire chaque pixel est soit blanc soit noir. Puisqu'il y a uniquement deux valeurs pour chaque pixel, un seul bit est utilisé pour le coder. Pour ce type de codage, chaque pixel est soit noir, soit blanc. Il faut un bit pour coder un Pixel (0 pour noir, 1 pour blanc). L'image de 10000 pixels codée occupe donc 10000 bits en mémoire. [3]



Figure I.3 : Représentation d'Image binaire

#### I.3.2 Image en niveau de gris :

Dans l'image en niveau de gris chaque pixel est un niveau de gris, allant de 0 (noir) à 255 (blanc). Cet intervalle de valeur signifie que chaque pixel est codé sur huit bits (un octet) 256 niveaux de gris suffisent pour la reconnaissance de la plus part des objets d'une scène. Si on code chaque pixel sur 2 bits on aura 4 possibilités (noir, gris foncé, gris clair, blanc), L'image codée sera très peu nuancée.

En général on code chaque pixel sur 8 bits = 1 octet. On a alors 256 possibilités (on dit 256 niveaux de gris).

L'image de 10 000 pixels codée occupe alors 10 000 octets en mémoire. [3]



Figure I.4 : « image en niveau de gris »

### I.3.3 Image couleur (RGB) :

Chaque pixel possède une couleur décrite par la quantité de rouge (R), vert (G) et bleu (B). Chacune de ces trois composantes sont codée sur l'intervalle  $[0, 255]$ , ce qui donne  $255^3 = 16\,777\,216$  couleurs possibles. Il faut 24 bits pour coder un pixel.



Figure I.5: « image en couleur »

### Codage d'une image en couleurs 8 bits

Dans ce cas on attache une palette de 256 couleurs à l'image. Ces 256 couleurs sont choisies parmi les 16 millions de couleurs de la palette RVB. Pour chaque image le programme recherche les 256 couleurs les plus pertinentes. Chaque code (de 0 à 255) désigne une couleur.

## Codage d'une image en couleurs 24 bits

Il existe plusieurs modes de codage de la couleur. Le plus utilisé est le codage Rouge, Vert, Bleu (RVB). Chaque couleur est codée sur 1 octet = 8 bits. Chaque pixel sur 3 octets c'est à dire 24 bits : le rouge de 0 à 255, le vert de 0 à 255, le Bleu de 0 à 255. Le principe repose sur la synthèse additive des couleurs : on peut obtenir une couleur quelconque par addition de ces 3 couleurs primaires en proportions convenables. On obtient ainsi  $256 \times 256 \times 256 = 16777216$  (plus de 16 millions de couleurs différentes).

### I.4 Caractéristiques de l'image :

#### I.4.1 Pixel :

Une image est constituée d'un ensemble de points appelés pixels (Picture élément) représente ainsi le plus Petit élément constitutif d'une image numérique. La quantité d'information que véhicule chaque pixel Donne des nuances entre images monochromes et images couleurs. Pour les images 3D le «pixel» est alors appelé un voxel, et représente un volume élémentaire. Des exemples d'images de ce type se rencontrent dans les images médicales.

La taille du pixel définit la résolution par rapport à l'image analogique originale, c'est-à-dire la finesse de la grille. Plus la résolution baisse, plus le nombre de pixels dans l'image diminue, et plus la qualité de l'image numérique se dégrade. [4]

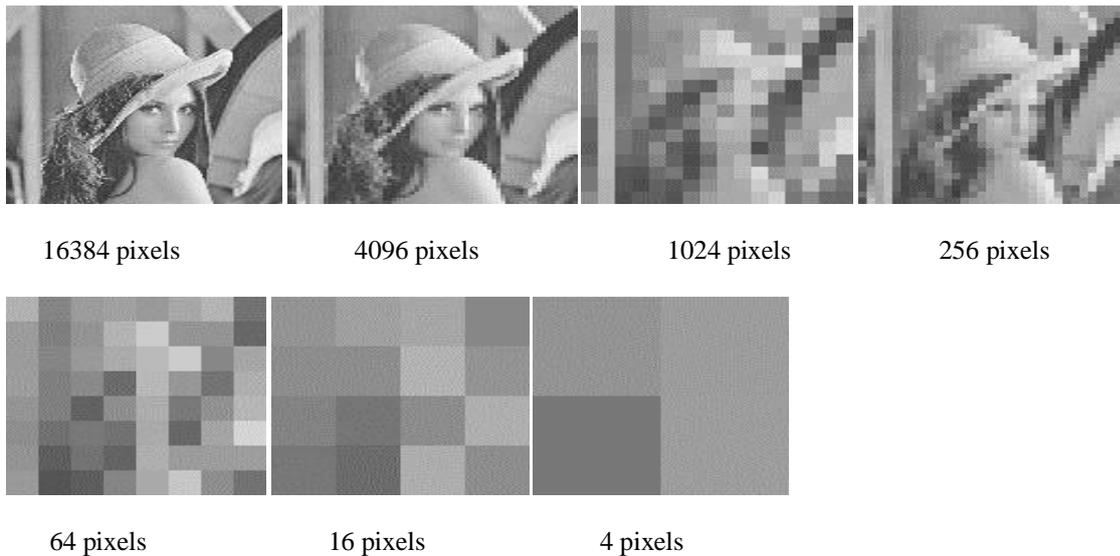


Figure I.6 : « variation du nombre de pixels. En réalité lors de l'affichage à l'écran la taille est divisée par 4 à chaque étape

#### I.4.2 Taille de l'image :

Etant donné que cette dernière est représentée sous forme d'une matrice dont les valeurs représentent l'intensité (pixels), le nombre de colonne (W) multiplié par le nombre de ligne (H) donne le nombre total de pixels dans l'image. [4]

Pour une image de 640x480 en couleur :

- ✓ Nombre de pixel =  $640 \times 480 = 307200$
- ✓ Poids de chaque pixel = 3 octets
- ✓ Le poids de l'image =  $307200 \times 3 = 921600$  octets = 900 Ko

### I.4.3 Résolution :

Une image est divisée en points ou pixels. Considérons une image de 10 cm sur 10 cm avec une résolution très faible de 10 pixels par cm. Elle est codée sur  $100 \times 100 = 10000$  pixels. Avec une résolution convenable de 100 pixels par cm (un pixel mesure 0,1 mm), elle serait codée sur  $1000 \times 1000 = 1000000$  pixels = 1 M pixels. Le symbole M signifiant million.

### I.4.4 Luminance :

C'est le degré de luminosité des points de l'image. Elle est définie ainsi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface. [5]

Bonne luminance se caractérise par :

- Des images brillantes.
- Un bon contraste.
- L'absence de parasites.

### I.1.4.5 Contraste :

C'est une propriété intrinsèque d'une image qui désigne et quantifie la différence entre les parties claires et foncées d'une image (elle différencie les couleurs claires des couleurs foncées).

En photographie on définit le contraste comme la différence entre la densité la plus forte et la plus faible d'une image. Le contrôle du contraste est un élément important de la pratique photographique. Le contraste final de l'image dépend à la fois du sujet, de la nature et du traitement du négatif et du positif.

### I.4.6 Histogramme :

Un histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image. Il fournit diverses informations comme les statistiques d'ordre (moyenne, variance,...), l'entropie, et peut permettre d'isoler des objets. [5]

Il peut être utile dans les cas suivant :

- Pour diminuer l'erreur de quantification.
- Pour comparer deux images obtenues sous des éclairages différents.
- Pour améliorer certaine propriété d'une image (rehaussement) pour en tirer des informations utiles.

3	4	5	6
1	1	2	4
3	2	2	7
1	6	1	6

**Image**

V	1	2	3	4	5	6	7
H(V)	4	3	2	2	1	3	1

Tableau I.1 : « Table Histogramme ».

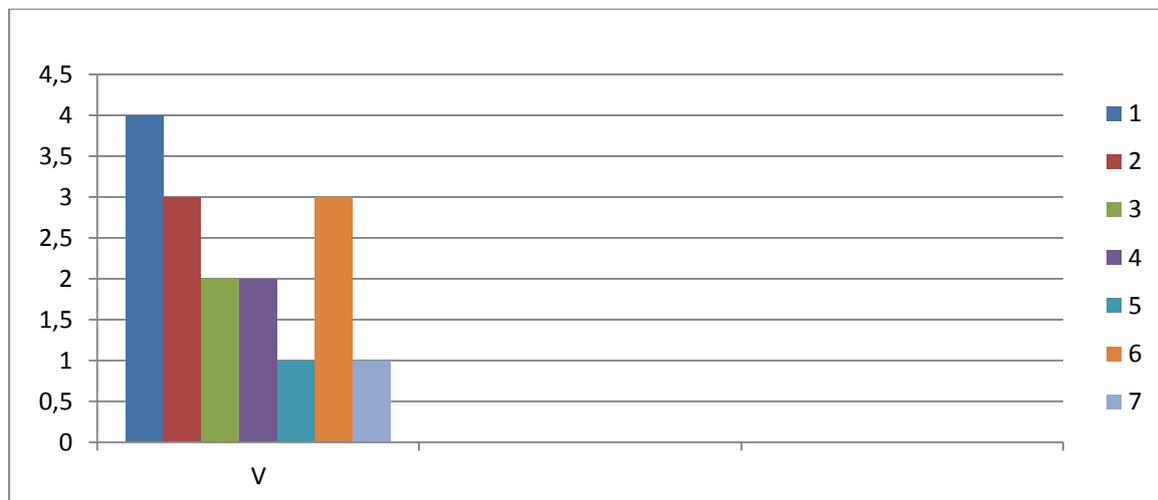


Figure I.7 : « Tracé de l'histogramme »

#### I.4.7 Contours et textures :

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes. [6]

#### I.4.8 Bruit :

La variation soudaine d'un pixel par rapport à ses voisins est un phénomène qu'on appelle bruit. Ce dernier peut provenir de l'éclairage, des dispositifs électroniques du capteur. [7]

#### I.4.9 Niveau de gris :

La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires, on peut donc dire que chaque pixel de l'image correspond à une quantité de lumière renvoyer comprise entre 0 et 255. [7]

## I.5 Les différents Formats d'images :

On peut classer les images en deux formats [3]

### I.5.1 Format vectorielle :

Dans une image vectorielle les données sont représentées par des formes géométriques simples qui sont décrites d'un point de vue mathématique. Par exemple, un cercle est décrit par une information du type (cercle, position du centre, rayon). Ces images sont essentiellement utilisées pour réaliser des schémas ou des plans. Et sont présentent deux Avantages : elles occupent peu de place en mémoire et peuvent être redimensionnées sans perte d'information.

### I.5.2 Format matricielle :

Une image matricielle est formée d'un tableau de points ou pixels. Plus la densité des points sont élevée, plus le nombre d'informations est grand et plus la résolution de l'image est élevée. Corrélativement la place occupée en mémoire et la durée de traitement seront d'autant plus grandes, les images vues sur un écran de télévision ou une photographie sont des images matricielles.

On obtient également des images matricielles à l'aide d'un appareil photo numérique, d'une caméra vidéo numérique ou d'un scanner. Parmi ces formats on peut citer :

**BMP (BitMap) :** Le format BMP est le format par défaut du logiciel Windows. C'est un Format matriciel, les images ne sont pas compressées son logiciel d'origine.

**JPEG (Joint Photographique Experts Group) :** Les images JPEG sont des images de 24 bits c'est-à dire qu'elles peuvent afficher un spectre de 16 millions de couleurs. C'est la meilleure qualité d'images disponible.

## I.6 Obtention des données images :

Afin de rendre une image utilisable par l'ordinateur, il est nécessaire d'effectuer une transformation, grâce à une procédure de numérisation qui consiste à transformer l'image de base en une mosaïque de points et les convertir en nombre binaire traitable par le système informatique « *Ordinateur* ». [7]

- ✓ Numériser une image consiste à transformer celle-ci de son état analogique (sous forme d'un signal continu) vers l'état numérique.

Cette opération se déroule en trois étapes :

### ➤ Echantillonnage :

Les processeurs d'échantillonnage consiste à prélever ponctuellement et régulièrement des valeurs d'amplitude du signal analogique. Conséquence de cette opération : le spectre du signal est dupliqué autour des multiples de la fréquence d'échantillonnage. C'est pourquoi la fréquence d'échantillonnage doit être au moins supérieure au double de la fréquence maximale du signal à convertir, afin d'éviter tout chevauchement de spectre. Si cette condition est vérifiée, l'échantillonnage n'a alors aucun rôle perturbation.

L'échantillonnage est qualifié d'idéal dès lors que l'on peut supposer ou approcher une prise instantanée des échantillons.



Figure I.8 : « Echantillonnage d'une image ».

➤ **Quantification :**

Elle consiste à effectuer une valeur numérique à chaque échantillon. la quantification peut être monochrome ou couleur.



Figure I.9 : « Quantification d'une image ».

➤ **Codage**

Le codage est une opération qui consiste à donner à chaque pixel de l'image un code qui représente une information complémentaire : sa couleur et sa précision. Ceci afin d'optimiser son stockage ou sa transmission.

## I.7 Système de traitement d'image :

Un système de traitement d'image est généralement composé des unités suivantes : [8]

### ❖ Acquisition et numérisation :

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation. Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux catégories principales : les caméras numériques et les scanners.

### ❖ Visualisation :

Tout système de traitement d'image est doté d'un dispositif de visualisation qui permet l'affichage des images. L'utilisation de différents types de reconstituteurs permet de transformer le signal numérique qu'est la matrice image en un signal analogique visible par l'œil de l'observateur. Pour cela, différents types de supports peuvent être employés : moniteur vidéo, clichés photographiques, impression sur papier. Dans tous les cas et pour chaque échantillon de l'image numérique, on recrée un nouvel élément d'image ou un nouveau pixel dont on choisit la forme de façon à reconstituer une image analogique qui soit la plus proche possible de l'image avant numérisation compte tenu des erreurs introduites lors de l'acquisition, de la numérisation et de la transmission .[9]

## I.8 traitement numérique des images :

Les techniques de traitement sont destinées à l'exploitation des informations contenues dans les images, ceci dans le but d'améliorer la qualité des images et de les rendre plus facilement interprétables, en d'autres termes elles permettent d'augmenter la qualité visuelle de l'image.

### I.8.1 Filtrage Numérique :

Les images numériques telles qu'elles sont acquises, sont très souvent inexploitable pour le traitement d'images. Elles contiennent des signaux bruités. Pour remédier à cela, différents prétraitements pour l'amélioration ou la correction sont effectués. [2]

On peut scinder les filtres en deux grandes catégories :

#### I.8.1.1 Filtres linéaires :

Les filtres linéaires transforment un ensemble de données d'entrée en un ensemble de données de sortie par une convolution bidimensionnelle qui est une opération mathématique. Ils permettent de supprimer le bruit dans l'image. Chaque filtre a une taille  $N \times N$  avec  $N$  impair. [10] Les filtres linéaires les plus connus sont les filtres passe-bas, passe-haut.

### **I.8.1.2 Filtres non linéaire :**

Ils sont conçus pour régler les problèmes des filtres linéaires, Leur principe est le même que celui des filtres linéaires, il s'agit toujours de remplacer la valeur de chaque pixel par la valeur d'une fonction calculée dans son voisinage. La différence majeure, est que cette fonction n'est plus linéaire mais une fonction quelconque (elle peut inclure des opérateurs de comparaisons ou de classification). Les filtres non linéaires les plus connus sont le filtre médian. [8]

### **I.9. Domaines d'application :**

Le traitement d'images possède l'aspect multidisciplinaire. On trouve ses applications dans des domaines très variés tels que les télécommunications (T.V., vidéo, publicité,...), la médecine (radiographie, ultrasons,...), biologie, astronomie, géologie, l'industrie (robotique, sécurité), la météorologie, l'architecture, l'imprimerie, l'armement (application militaire). De nouvelles applications pratiques sont possibles aujourd'hui et touchent tous les domaines d'activités, tels que : métiers du spectacle, de la radio, créations artistiques,... [9]

### **I.10 Conclusion :**

Dans ce chapitre nous nous sommes intéressés aux généralités sur le traitement d'images. Nous avons présenté les concepts les plus importants liés au domaine du traitement d'images. Les différentes définitions qui y sont développées sont celles des connaissances élémentaires de cette discipline mais quand bien même elles seraient essentielles pour l'initiation aux traitements approfondis des images. Nous pouvons, à présent, aborder au prochain chapitre les techniques de filtrage d'images numériques, ainsi que les critères permettant d'analyser et d'étudier les images.

# CHAPITRE II :

**Développement d'une Technique  
efficace de réduction du bruit Gaussien et impulsif**

---

## II.1 Introduction

Les traitements de restauration sont souvent indispensables pour améliorer la qualité des images observées. Notre principal problème, que nous allons traiter, consistera à récupérer une image de bonne qualité, proche de l'image originale, à partir d'une image bruitée de mauvaise qualité.

Le présent chapitre a pour objet de présenter le vocabulaire et les concepts attachés au domaine du débruitage d'images, tout en donnant un aperçu sur quelques types de bruits dégradant les images, avec une brève description de quelques techniques classiques (standard) en l'occurrence les filtres, utilisées pour le débruitage des images. Nous terminons ce chapitre par la présentation et la discussion des résultats de simulation obtenus.

## II.2 Dégradation d'images Numérique

Dans la pratique, une image peut être dégradée par plusieurs types et formes de bruit, la source principale des bruits dans les images numériques survenues pendant l'acquisition (numérisation) et la transmission. Les performances des capteurs d'imagerie est affectée par une variété de facteurs, tel que les conditions environnementale durant l'acquisition, et aussi par les capteurs d'image eux-mêmes. Par exemple, dans l'acquisition des images avec une caméra, les niveaux de lumière et la température du capteur sont des facteurs principaux sur la quantité de bruit dans l'image capturée.

Les images sont aussi corrompues lors de la transmission principalement dû à l'interférence dans le canal utilisé pour la transmission, à titre d'exemple, une image transmise en utilisant un réseau sans fil peut-être endommagée en raison de l'éclairage ou d'autres perturbations atmosphériques. Donc, le bruit est un effet indésirable dans l'image, d'où la nécessité de le réduire ou le supprimer sans entraîner un changement significatif du détail et des caractéristiques de l'image [11].

Ainsi, il est important de connaître la nature du bruit contaminant l'image. En fait, le type de bruit le plus connu et le plus difficile à réduire, est le bruit blanc gaussien.

### II.2.1 Bruit blanc gaussien :

Le bruit blanc gaussien apparaît dans l'image comme une variation aléatoire autour des valeurs d'intensité. Il est modélisé par l'équation suivante:

$$F_x(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-m)^2}{2\sigma^2}} \quad (\text{II.1})$$

Où  $F_x(x)$  est la densité de probabilité gaussienne (unidimensionnelle),  $m$  et  $\sigma^2$  représentent respectivement la moyenne et la variance. On suppose que le bruit est de moyenne nulle et de Variance  $\sigma^2$ , donc la densité de probabilité devient [12]:

$$F_x(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} \quad (\text{II.2})$$

### II.2.2. Bruit poivre et sel :

Ce type de bruit est exprimé par le pourcentage du nombre total de pixels dans l'image, dont la valeur est 0 ou 255, Le bruit généré est distribué uniformément dans toute l'image [12].

### II.3 Débruitage d'images :

Le débruitage d'images a pour objectif principal d'atténuer, voire de supprimer dans les cas les plus favorables, les dégradations que subie une image. Le plus souvent, les dégradations considérées sont celles qui apparaissent au moment de l'acquisition de l'image, c'est-à-dire au moment de sa création. On pourrait y ajouter toutes les dégradations liées à la Transmission, au changement de format ou au stockage de l'image.

Le débruitage d'images Essaie donc de recréer, à partir de l'image dont on dispose, l'image telle que l'on devrait l'avoir en situation idéale. Ce qui est visé, c'est la fidélité de l'image obtenue à la scène qu'elle représente [13].Le besoin pour la suppression du bruit sans dégrader significativement les contours et autres composants de hautes fréquences de l'image, a donc motivé le développement des techniques efficaces qui lissent le bruit et conservent-les Contours.

### II.4. Critère d'analyse pour le débruitage d'images :

La pertinence des méthodes de débruitage dépend de deux critères, subjectif et objectif permettant d'évaluer l'efficacité de ces méthodes. Le critère subjectif représente l'aspect visuel.

#### I.4.1. Critère subjectif (aspect Visuel) :

L'œil humain est un outil essentiel pour apprécier la qualité d'une image. Il va permettre à l'utilisateur d'identifier le contenu des images, la netteté de celles-ci, la présence d'artefacts et la qualité des contours. Il est donc capital que les méthodes de débruitage prennent en compte le système optique humain. Cependant, cette évaluation ne peut être que subjectivepuisque'il n'existe aucune mesure correcte pouvant traduire fidèlement la perception de l'œilhumain [11].

#### I.4.2. Critère objectif :

##### A. Erreur quadratique moyenne (MSE)

Bien qu'elle ne corresponde pas toujours avec la perception humaine, la MSE est souvent considéré comme une bonne mesure de la fidélité d'une évaluation d'image. Ce critère est défini comme suit [11] :

$$\text{MSE} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (X(i,j) - X'(i,j))^2 \quad (\text{II.3})$$

Avec :

X : Image originale.

X' : Image d'ébruitée.

M: Nombre de lignes de l'image.

N: Nombre de colonnes de l'image.

(i, j): positionnement des pixels.

**B. Rapport signal sur bruit en pic (PSNR) :**

Une autre mesure de la qualité de l'image approchée est le rapport signal sur bruit en pic (PSNR) qui est inversement proportionnel à la MSE, son unité est le décibel (dB) et est défini par la relation suivante [11]:

$$\text{PSNR} = 10 \log_{10} \left[ \frac{255}{\text{MSE}} \right] \quad \text{dB} \quad (\text{II.4})$$

Où 255 est la valeur maximale d'un pixel pour une image codée par 8 bits/pixel en niveaux de gris. Ces deux critères (mesures) en plus de la qualité visuelle sont utilisés dans toutes les parties de simulations de ce mémoire comme mesures déterminant la qualité de l'image bruité et débruité.

**c. Rapport signal sur bruit (SNR) :**

Le SNR compare le niveau du signal désiré au niveau du bruit de fond. Plus le SNR est élevé, et plus le bruit est faible dans l'image, et vice versa. L'expression de SNR est donnée dans la formule:

$$\text{SNR} = 10(\sigma_g^2/\sigma_e^2) \quad (\text{II.5})$$

$\sigma_g$  est la variance de l'image originale et  $\sigma_e$  est la variance de l'erreur entre l'image originale et l'image bruitée [14].

**II.5 Les Méthodes de filtrage Non-linéaire :**

Les images numériques sont souvent corrompues par le bruit lors de leur acquisition et de leur transmission. Un défi fondamental en matière d'amélioration de l'image consiste à réduire le bruit tout en conservant les caractéristiques d'image souhaitées telles que les bords, les textures et les détails fins. En particulier, il existe deux types de bruit courants, à savoir le bruit gaussien et le bruit impulsif, qui sont introduits au cours des processus d'acquisition et de transmission. De nombreuses méthodes ont été introduites dans la littérature pour éliminer le bruit gaussien ou impulsif

**II.5.1 Méthode d'EFPGF :**

Pour améliorer la qualité de l'image, V. Murugan, R. Balasubramanian en 2015 proposé un filtre de groupe de pairs flous amélioré (EFPGF) [15]. Il fonctionne mieux que ces méthodes pour le bruit gaussien et mixte dans les images en niveaux de gris. Le point clé de cette technique consiste à utiliser le concept de probabilité dans les images. Le pixel le moins probable de l'image peut être identifié comme un pixel bruyant et il est remplacé par la valeur de niveau de gris la plus probable. Il utilise le concept d'histogramme pour vérifier les valeurs de niveau de gris les plus faibles et les plus probables. Cette technique utilise le filtre de Wiener comme étape de prétraitement pour éliminer dans une certaine mesure le bruit gaussien. L'architecture globale du système est illustrée à la Figure II.1 :

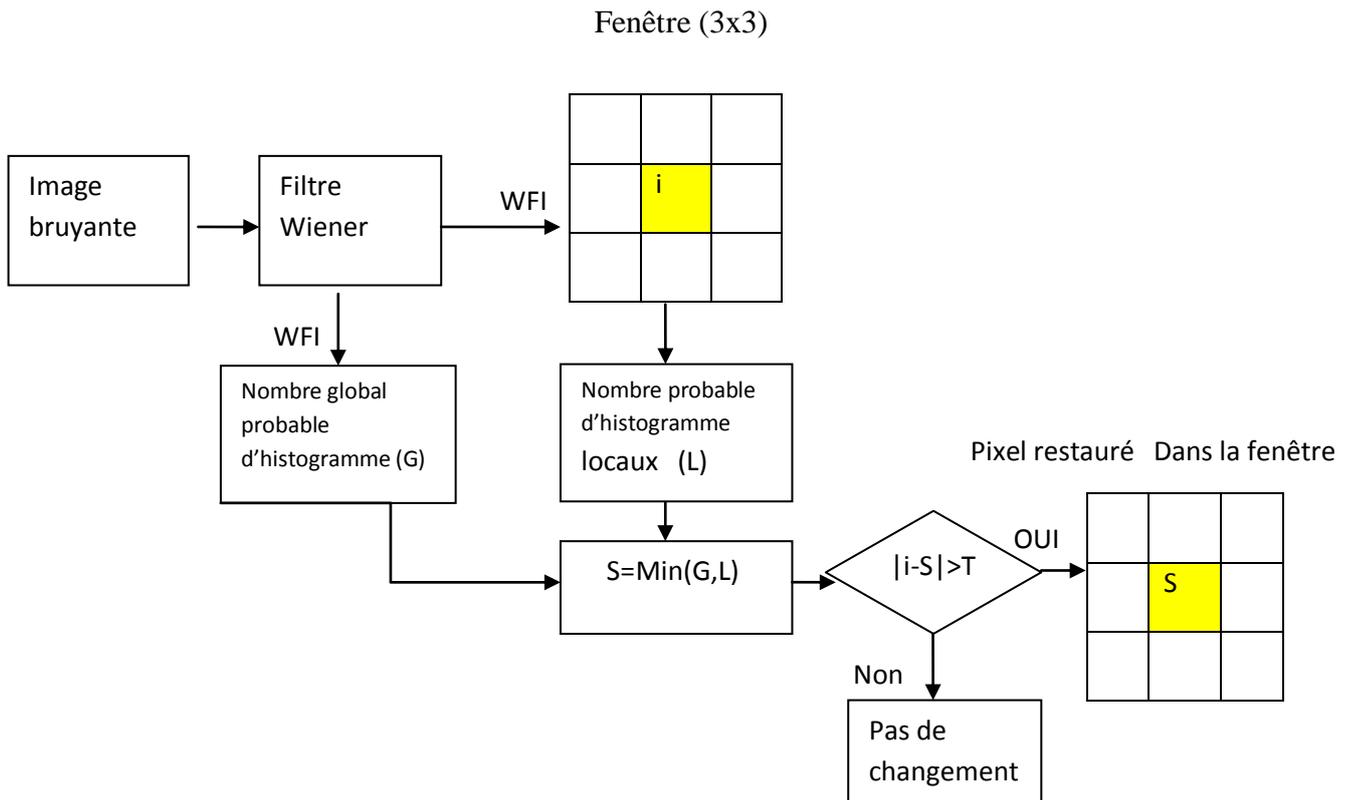


Figure II-1: « Architecture du système » [15].

L'image bruyante (I) est initialement filtrée à l'aide du filtre de Wiener. Le Wiener Filtrée Image (WFI) obtenue à cette étape.

Le niveau de gris le plus probable de l'ensemble de l'image filtrée est calculé et défini en tant que Compte global probable de l'histogramme (G), Chaque pixel (i) du WFI est restauré en utilisant des pixels voisins qui sont formés comme une fenêtre de taille 3x3, la valeur de niveau de gris la plus probable dans la fenêtre est calculée et est appelée compte d'histogramme probable local (L).

Chaque pixel peut être remplacé par la valeur de gris la plus probable (S) dépend d'un seuil (T). S est calculé en tant que minimum de L et G.

$$i = \begin{cases} i & \text{abs}(i-s) < T \\ \text{Min}\{G, L\} \text{ ou } S & \text{abs}(i-s) > T \end{cases}$$

Si seul L est utilisé, alors chaque pixel sera remplacé par la valeur de comptage d'histogramme la plus probable locale. Par conséquent, G est également utilisé pour normaliser l'image.

Si chaque pixel de l'image est remplacé, l'image restaurée aura la même valeur dans chaque coordonnée. Afin d'éviter cela, T est calculé pour ne connaître que le pixel bruyant. La différence absolue entre le pixel actuel et S est calculée pour savoir si le pixel a plus de variance que les pixels voisins.

Si la différence absolue est supérieure au seuil, le pixel est considéré comme une image bruyante et il est remplacé par S.

### II.5.2 Méthode de MAUTMPF :

Les images numériques sont souvent corrompues par l'acquisition, le stockage et la transmission par bruit impulsif. Le bruit impulsif peut être classé sous: Bruit sel et poivre et bruit à valeur aléatoire. Le pixel qui prend une valeur d'intensité maximale ou minimale (0 ou 255) est classé comme bruit de sel et de poivre.

Si le pixel prend une valeur comprise entre 0 et 255, il est classé comme bruit d'impulsion évalué de manière aléatoire. Bien qu'il existe différents algorithmes pour éliminer le bruit du sel et du poivre.

Le filtre réduit asymétrique remplace les bruyants dans une fenêtre sélectionnée en prenant soit la moyenne (filtre moyen), soit la médiane (filtre médian) des pixels restants après la réduction de 0 à 255. Cependant, le filtre moyen rend l'image floue à la fois avec des densités de bruit faibles et élevées. Et le filtre médian ne préserve pas les bords pour une densité de bruit plus élevée. En 2014 Saravankumar, s ; Ebenezerjey Kumar et K. N. Vijeyakumar proposé une technique de MAUTMPF (Morphological Based Adaptive Unsymmetrical Trimmed mid\_Point Filter [16] :

Dans la fenêtre 3×3 proposée est sélectionnée et les pixels corrompus sont supprimés de manière non symétrique. Si les pixels restants dans la fenêtre sélectionnée sont supérieurs ou égaux à 2, le point médian de la valeur maximale et minimale est pris et remplacé par le pixel central. Par contre, si le nombre de pixels restants dans la fenêtre sélectionnée est inférieur à 2, la taille de la fenêtre est augmentée de 2 et la même procédure est répétée. Les itérations de l'algorithme sont poursuivies jusqu'à ce que la taille de la fenêtre atteigne 7×7. Si une estimation de pixel bruyant ne peut pas non plus être atteinte dans une fenêtre 7×7, le pixel central est remplacé par le point milieu des valeurs d'intensité minimale et maximale des pixels déjà traités dans la fenêtre 3×3 sélectionnée initialement.

Image bruitée

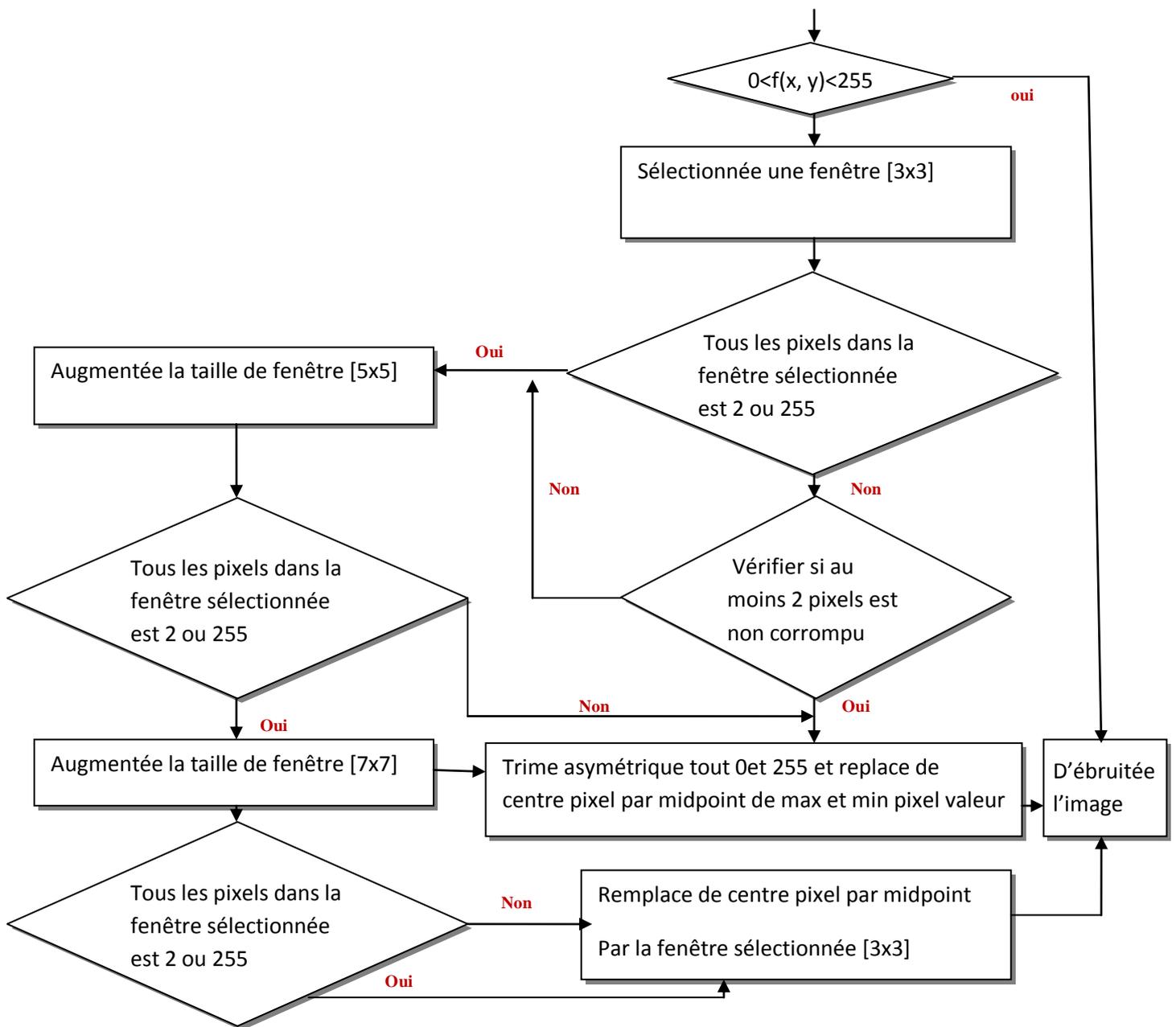


Figure II 2: « Organigramme de algorithme MAUTMPF » [16]

### Cas 1:

Si le pixel de traitement (représenté par  $P_{ij}$ ) est bruyant et que le nombre de pixels non endommagés dans la fenêtre  $3 \times 3$  sélectionnée est supérieur à 2

### Cas 2 :

Si le pixel de traitement est bruyant et que le nombre de pixels non endommagés dans la fenêtre  $3 \times 3$  sélectionnée est inférieur à 2, la taille de la fenêtre est augmentée de 2.

**Cas3:**

Si le pixel de traitement est bruyant et que le nombre de pixels non endommagés dans la fenêtre 5x5 traitée dans le second cas est inférieur à 2.

**Cas 4:**

Si le pixel de traitement est bruyant et que le nombre de pixels non endommagés dans la fenêtre 7x7 traitée dans le cas iii est inférieur à 2.

**II .5.3 Filtre NAGAO Modifiée :**

Le filtre de Nagao est un filtre non linéaire utilisé fréquemment dans les images aériennes des zones urbaines, vision industrielle,...etc.

Le filtre de Nagao examine un masque de **5x5** entourant le pixel central. Neuf fenêtres sont définies dans ce masque. Pour chaque fenêtre  $F_i$  la médiane  $M_i$  et la variance  $V_i$  sont calculées. Le résultat de l'opérateur prélevé est la médiane de la fenêtre qui présente la plus faible variance.

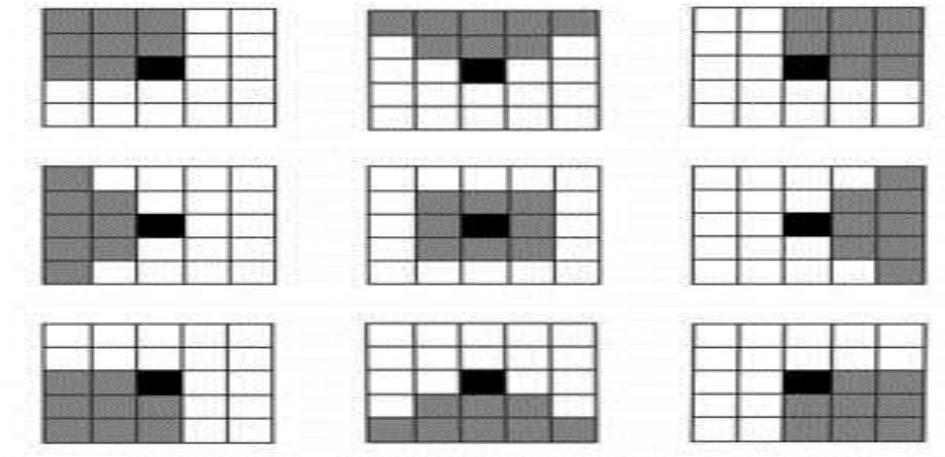


Figure II 3: « les 9 fenêtres du filtre NAGAO »

- Les contours sont remarquablement conservés par ce filtre.
- On peut d'ailleurs montrer que le NAGAO renforce donc le contraste.
- Les détails fins sont nettement plus apparents qu'avec le médian.

**II .5.4 Filtre Médian Standard :**

Le filtre médian (*median filter*) est un filtre non linéaire. Son principe consiste à

Remplacer un pixel par la médiane de ses voisins. Les pixels voisins sont triés suivant un ordre croissant de **11** vers **150**, puis la valeur médiane de la série numérique obtenue (**25**) devra remplacer le point central.

Le filtre médian sert à diminuer plusieurs types de bruits. Les chercheurs ont remarqués que ce filtre est plus efficace dans le cas d'un bruit de type « sel et poivre ». En outre, ce filtre est connu par sa préservation de contour, mais affecte les angles et les détails fins.[17].

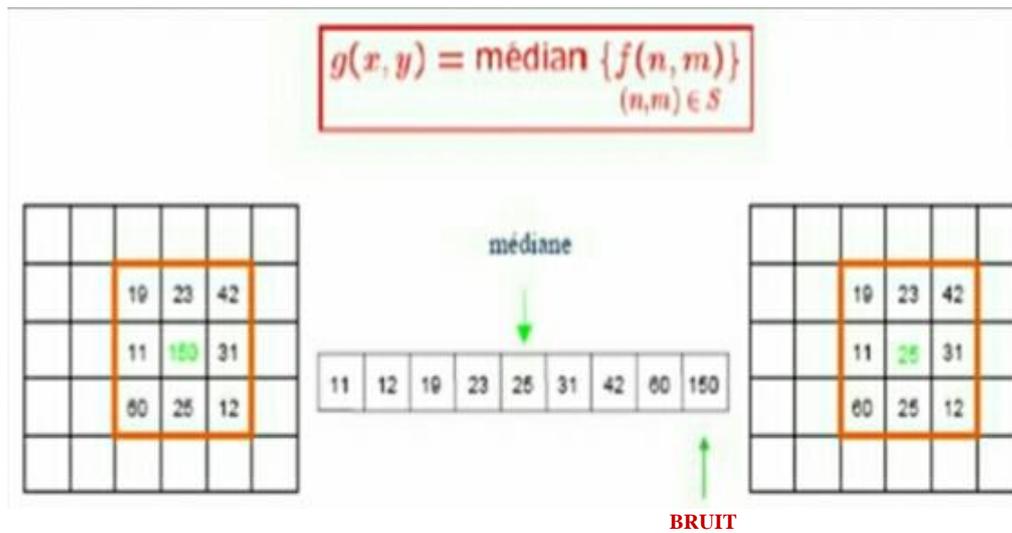
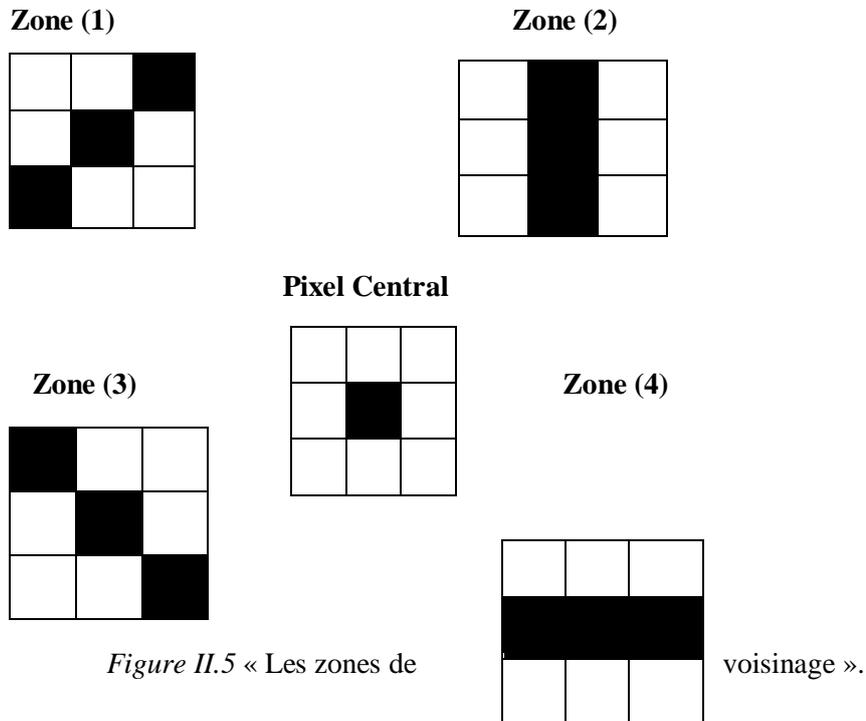


Figure II.4 « Mise en œuvre du filtre Médian ».

## II .5.5 Nouvelle Approche Proposée :

Dans ce travail nous proposons un filtre non-linéaire appliqué sur des images numériques qui permet de réduire le bruit, et supprimer des points aberrants, en respectant les contours. Dans une fenêtre de 3x3 entourant le pixel à traiter, la médiane de 4 différentes régions est calculée. La valeur du pixel central est remplacée par la valeur médiane de la région qui a la plus faible médiane.



Le principe du filtre est résumé dans l'algorithme suivant :

### Algorithme

Pour chaque pixel de l'image :

**Etape1** : Centrer le pixel dans une fenêtre de 3x3

**Etape2** : Calculer la Médiane des 4 zones

**Etape3** : Rechercher la zone qui a la plus faible Médiane.

**Etape4** : Remplacer la valeur du pixel par médiane de la zone sélectionnée.

## II.6 Résultats de débruitage:

Le tableau II.1, le tableau II.2, Le tableau II.3 monstons les résultats de débruitage (PSNR, SNR, MSE) par les différentes méthodes sur les images Numérique avec différentes bruits gaussien et impulsif

Type des bruits	Images	$\sigma$	Bruitée	EFPGF	MAUTMPF	Filtre Médiane Standard	Filtre NAGAO modifiée	Méthode proposée
Impulsif	Luna (256x256)	0.01	47.2673	47.0356	44.32315	47.1866	41.1790	<b>47.4963</b>
		0.02	44.2350	44.1463	42.45350	44.2964	40.4720	<b>44.2133</b>
		0.03	42.4084	43.4118	42.11590	42.5310	41.6234	<b>42.1693</b>
		0.04	41.2469	42.4112	41.29395	41.2973	41.1410	<b>42.2975</b>
		0.05	40.4549	40.6163	40.47125	40.3937	40.2876	<b>40.2349</b>
		0.06	39.4035	39.6190	39.54445	39.4545	39.4854	<b>39.5268</b>
		0.07	38.9907	38.8412	38.99785	38.8654	38.8050	<b>39.0162</b>
		0.08	38.2483	38.2347	38.42625	38.4541	38.4042	<b>38.3124</b>
		0.09	37.6601	37.9324	37.84305	37.8102	37.8260	<b>37.7544</b>
		0.10	37.3339	37.8405	37.43060	37.3251	37.3273	<b>37.2849</b>
	Cameraman (256x256)	0.01	47.7853	47.9653	47.89675	47.6023	47.8082	<b>47.8954</b>
		0.02	44.4243	44.6043	44.53065	44.5374	44.4370	<b>44.2873</b>
		0.03	42.5413	42.7213	42.79180	42.3917	42.8423	<b>42.5243</b>
		0.04	41.4282	41.6082	41.49727	41.3210	41.3662	<b>41.2813</b>
		0.05	40.1662	40.3462	40.40195	40.4023	40.4377	<b>40.3318</b>
		0.06	39.6839	39.8639	39.72892	39.7314	39.5739	<b>39.6098</b>
		0.07	38.9591	39.1391	39.08055	38.8789	39.0020	<b>38.9007</b>
		0.08	38.2959	38.4759	38.46432	38.3645	38.4325	<b>38.4001</b>
		0.09	37.8628	38.0428	37.99355	37.8978	37.9243	<b>37.3847</b>
		0.10	37.2396	37.4196	37.39005	37.4424	37.3405	<b>37.3251</b>

<b>Gaussien</b>  <b>M=0</b>	<b>Luna</b>  <b>(256x256)</b>	<b>0.01</b>	29.0172	29.13650	29.0383	29.0089	29.0347	<b>29.0188</b>
		<b>0.02</b>	28.4912	29.07727	29.4784	28.4969	28.4760	<b>28.5348</b>
		<b>0.03</b>	28.2604	28.30145	28.2570	28.2913	28.1459	<b>28.2675</b>
		<b>0.04</b>	28.1341	28.25826	28.1444	28.0904	28.1728	<b>28.1086</b>
		<b>0.05</b>	28.0323	28.13435	28.0436	28.0522	28.0251	<b>28.0543</b>
		<b>0.06</b>	27.9818	28.01747	27.9832	27.9840	27.9662	<b>27.9963</b>
		<b>0.07</b>	27.9318	28.02815	27.9133	27.9365	27.9430	<b>27.9396</b>
		<b>0.08</b>	27.9052	27.93629	27.8859	27.8990	27.8399	<b>27.9182</b>
		<b>0.09</b>	27.8515	27.95539	27.8551	27.8622	27.8527	<b>27.8818</b>
		<b>0.10</b>	27.8062	27.92664	27.8170	27.8621	27.8358	<b>27.8136</b>
	<b>Cameraman</b>  <b>(256x256)</b>	<b>0.01</b>	28.9943	29.11575	29.0334	28.9696	28.9981	<b>29.0598</b>
		<b>0.02</b>	28.4777	28.59865	28.4809	28.4789	28.5164	<b>28.4856</b>
		<b>0.03</b>	28.2636	28.39126	28.3032	28.2773	28.2820	<b>28.2211</b>
		<b>0.04</b>	28.1193	28.23085	28.1380	28.1798	28.1390	<b>28.1697</b>
		<b>0.05</b>	28.0826	28.13933	28.0412	28.0998	28.0254	<b>28.0491</b>
		<b>0.06</b>	27.9818	28.05275	27.9849	28.0323	27.9206	<b>27.9805</b>
		<b>0.07</b>	27.9446	27.96133	27.9046	28.0045	27.7620	<b>27.3438</b>
		<b>0.08</b>	27.9210	27.94463	27.8935	27.9040	27.9151	<b>27.8873</b>
		<b>0.09</b>	27.8876	27.93017	27.8446	27.8709	27.8988	<b>27.8501</b>
		<b>0.10</b>	27.8684	27.91985	27.8571	27.8307	27.8426	<b>27.8385</b>

Tableau II.1 : « Résultats de débruitage (PSNR [dB]) sur différentes images de tests par toutes les méthodes de débruitage étudiées. »

Type des bruits	Images	$\sigma$	Bruitée	EFPGF	MAUTMPF	Filtre Médiane Standard	Filtre NAGAO modifiée	Méthod proposé
-----------------	--------	----------	---------	-------	---------	-------------------------	-----------------------	----------------

<b>Impulsif</b>	<b>Luna (256x256)</b>	<b>0.01</b>	22.9244	22.9906	22.91720	22.8438	22.8362	<b>23.153</b>
		<b>0.02</b>	19.8922	19.8039	19.87845	19.9539	20.4292	<b>19.870</b>
		<b>0.03</b>	18.0655	18.0720	18.13010	18.1882	18.3004	<b>18.326</b>
		<b>0.04</b>	16.9041	17.0983	17.02885	16.9594	16.7982	<b>16.326</b>
		<b>0.05</b>	16.1121	16.2765	16.16365	16.0508	15.9446	<b>16.954</b>
		<b>0.06</b>	15.0607	15.2761	15.19390	15.1117	15.1426	<b>15.892</b>
		<b>0.07</b>	14.6479	14.4984	14.51050	14.5226	14.4622	<b>14.673</b>
		<b>0.08</b>	13.9054	13.8919	14.00160	14.1113	14.0614	<b>13.969</b>
		<b>0.09</b>	13.3173	13.5896	13.52851	13.4674	13.4232	<b>13.411</b>
		<b>0.10</b>	12.9911	13.1364	13.05942	12.9824	12.9845	<b>12.941</b>
	<b>Cameraman (256x256)</b>	<b>0.01</b>	23.4176	22.9176	23.07612	23.2346	23.4404	<b>22.827</b>
		<b>0.02</b>	20.0556	19.5556	19.86265	20.1697	20.0695	<b>19.919</b>
		<b>0.03</b>	18.1735	17.6735	17.83770	18.0019	18.4745	<b>18.156</b>
		<b>0.04</b>	17.0609	16.5609	16.75213	16.9433	16.9984	<b>16.913</b>
		<b>0.05</b>	15.8984	15.3984	15.73645	16.0745	16.0700	<b>15.964</b>
		<b>0.06</b>	15.3162	14.8162	15.08995	15.3637	15.2662	<b>15.242</b>
		<b>0.07</b>	14.5913	14.0913	14.30125	14.5112	14.6342	<b>14.533</b>
		<b>0.08</b>	13.9282	13.4282	13.71250	13.9968	14.0648	<b>14.032</b>
		<b>0.09</b>	13.4951	12.9951	13.26266	13.5301	13.5565	<b>13.527</b>
		<b>0.10</b>	12.8719	12.3719	12.72347	13.0749	12.9728	<b>12.957</b>
<b>Luna (256x256)</b>	<b>0.01</b>	4.6744	4.6755	4.67081	4.6661	4.6919	<b>4.7170</b>	
	<b>0.02</b>	4.1484	4.1456	4.14985	4.1541	4.1332	<b>4.1828</b>	
	<b>0.03</b>	3.9176	3.9142	3.93135	3.9485	3.9383	<b>3.8783</b>	
	<b>0.04</b>	3.7913	3.8006	3.77411	3.7476	3.8300	<b>3.8269</b>	
	<b>0.05</b>	3.6895	3.7007	3.70505	3.7094	3.6822	<b>3.7063</b>	
	<b>0.06</b>	3.6390	3.6404	3.66575	3.6911	3.6244	<b>3.6377</b>	
	<b>0.07</b>	3.5859	3.5705	3.58210	3.5937	3.6001	<b>3.6010</b>	

<b>Gaussien</b> <b>M=0</b>		<b>0.08</b>	3.5624	3.5431	3.54915	3.5552	3.5471	<b>3.5445</b>
		<b>0.09</b>	3.5087	3.5066	3.51313	3.5194	3.5099	<b>3.5073</b>
		<b>0.10</b>	3.4633	3.4907	3.50533	3.5193	3.4930	<b>3.4957</b>
	<b>Cameraman</b> <b>(256x256)</b>	<b>0.01</b>	4.6265	4.6656	4.60373	4.6018	4.6304	<b>4.6383</b>
		<b>0.02</b>	4.1100	4.1131	4.08265	4.1122	4.1487	<b>4.1641</b>
		<b>0.03</b>	3.8959	3.9355	3.74133	3.6065	3.9143	<b>3.8997</b>
		<b>0.04</b>	3.7516	3.7702	3.74615	3.7821	3.7713	<b>3.7409</b>
		<b>0.05</b>	3.7148	3.6734	3.65933	3.7046	3.6576	<b>3.6866</b>
		<b>0.06</b>	3.6140	3.6171	3.59745	3.6377	3.6129	<b>3.6286</b>
		<b>0.07</b>	3.5768	3.5968	3.54155	3.5463	3.5962	<b>3.5719</b>
<b>0.08</b>	3.5532	3.5657	3.51218	3.5185	3.5474	<b>3.5504</b>		
<b>0.09</b>	3.5198	3.4768	3.45995	3.5031	3.5263	<b>3.5141</b>		
<b>0.10</b>	3.5016	3.4893	3.44615	3.4630	3.4748	<b>3.4458</b>		

Tableau II.2 : « Résultats de débruitage (SNR) sur différentes images de tests par toutes les méthodes de d'Ébruitage étudiées. »

Type des bruits	Images	$\sigma$	Bruitée	EFGF	MAUTMPF	Filtre Médiane Standard	Filtre NAGAO modifiée	Méthode proposée
-----------------	--------	----------	---------	------	---------	-------------------------	-----------------------	------------------

<b>Impulsif</b>	<b>Luna (256x256)</b>	<b>0.01</b>	1.2200	1.2010	1.12310	1.2428	1.2450	<b>1.1573</b>
		<b>0.02</b>	2.4523	2.5030	2.31255	2.4179	2.3221	<b>2.4647</b>
		<b>0.03</b>	3.7346	3.7291	3.53355	3.5304	3.5380	<b>3.5169</b>
		<b>0.04</b>	4.8796	4.6662	4.73315	4.8233	5.0001	<b>4.8232</b>
		<b>0.05</b>	5.8559	5.6384	5.76225	5.9390	6.0861	<b>6.1602</b>
		<b>0.06</b>	7.4598	7.5198	7.32015	7.3727	7.3205	<b>7.2510</b>
		<b>0.07</b>	8.2037	8.4910	8.42660	8.4439	8.5622	<b>8.1556</b>
		<b>0.08</b>	9.7332	9.9635	9.57673	9.2826	9.3899	<b>9.5904</b>
		<b>0.09</b>	11.1448	10.4674	10.49702	10.7662	10.7270	<b>10.9053</b>
		<b>0.10</b>	12.0140	12.6154	12.22395	12.0382	12.0325	<b>12.1513</b>
	<b>Cameraman (256x256)</b>	<b>0.01</b>	1.0828	1.1971	1.05117	1.2194	1.0771	<b>1.2403</b>
		<b>0.02</b>	2.3477	2.4608	2.21755	2.2874	2.3408	<b>2.4230</b>
		<b>0.03</b>	3.6221	3.4915	3.59505	3.7680	3.3715	<b>3.6362</b>
		<b>0.04</b>	4.6797	4.5975	4.64395	4.8082	4.4775	<b>4.8412</b>
		<b>0.05</b>	6.1160	5.1991	5.89445	5.8729	5.0791	<b>6.0242</b>
		<b>0.06</b>	6.9934	7.2928	6.85535	6.9173	7.1728	<b>7.1138</b>
		<b>0.07</b>	8.2636	8.3024	8.24060	8.4176	8.1824	<b>8.3755</b>
		<b>0.08</b>	9.6269	9.4488	9.45157	9.4761	9.3288	<b>9.3988</b>
		<b>0.09</b>	10.6365	10.6071	10.4938	10.5511	10.4871	<b>10.5586</b>
		<b>0.10</b>	12.2777	12.1158	11.8977	11.7177	11.9958	<b>12.0385</b>
<b>Luna (256x256)</b>	<b>0.01</b>	81.5387	81.1435	81.32710	81.6934	81.7107	<b>80.7425</b>	
	<b>0.02</b>	92.0372	92.0767	92.11825	91.9154	92.3598	<b>91.1555</b>	
	<b>0.03</b>	97.0991	97.1350	96.76645	96.3710	96.5979	<b>97.8400</b>	
	<b>0.04</b>	99.9232	99.6866	99.26065	100.9342	99.0347	<b>99.1075</b>	
	<b>0.05</b>	102.2933	102.0292	102.1474	101.8255	102.4648	<b>101.8986</b>	
	<b>0.06</b>	103.4009	103.4573	103.5596	103.4392	103.8619	<b>103.5203</b>	

<b>Gaussien</b> <b>M=0</b>		<b>0.07</b>	104.6932	105.1353	104.6777	104.5753	104.4202	<b>104.3998</b>	
		<b>0.08</b>	105.3316	105.8013	105.6522	105.5057	105.7032	<b>105.7667</b>	
		<b>0.09</b>	106.6418	106.5544	106.4836	106.3799	106.6128	<b>106.7776</b>	
		<b>0.10</b>	107.7617	107.4921	107.1605	106.3829	107.0290	<b>106.2621</b>	
			<b>0.01</b>	81.9698	81.2354	81.73620	82.4370	81.8970	<b>81.7484</b>
			<b>0.02</b>	92.3226	92.2551	91.88957	91.7239	91.5033	<b>91.1170</b>
			<b>0.03</b>	96.9881	96.1077	96.29572	96.6837	96.5777	<b>96.9024</b>
			<b>0.04</b>	100.2645	99.8352	99.59945	99.5637	99.8911	<b>100.5197</b>
		<b>Cameraman</b>	<b>0.05</b>	101.1165	102.0853	101.6192	101.3532	102.5468	<b>101.7763</b>
			<b>0.06</b>	103.4907	103.4166	103.0727	102.9288	103.8441	<b>103.1447</b>
			<b>0.07</b>	104.3808	103.9020	104.4104	105.1188	105.0911	<b>104.5009</b>
			<b>0.08</b>	104.9497	104.6484	105.1207	105.7930	105.4863	<b>105.0182</b>
			<b>0.09</b>	105.7598	106.8130	107.3906	108.1682	105.6033	<b>105.9000</b>
			<b>0.10</b>	106.2040	106.9061	108.9301	107.1542	106.8615	<b>107.5778</b>

Tableau II.3: « Résultats de débruitage (MSE) sur différentes images de tests par toutes les méthodes de débruitage étudiées.

## II.7 Représentation des résultats:

La figure II.6 et II.7 montrent une comparaison graphique entre différentes techniques pour l'image cameraman.

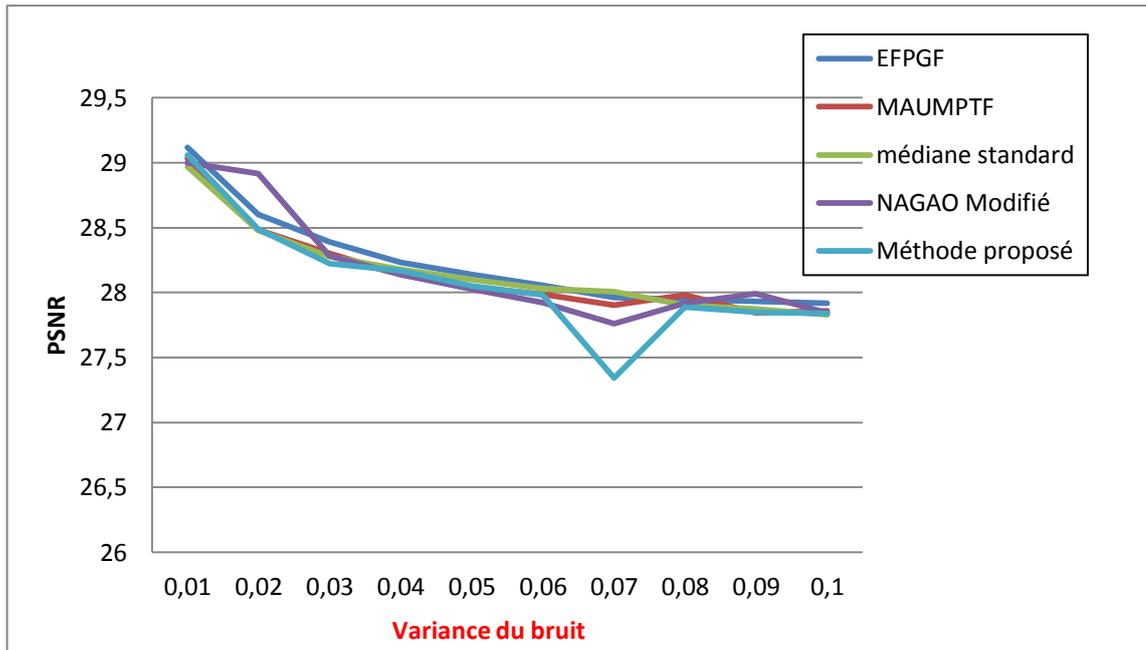


Figure II.6 : « Représentation du PSNR des images cameraman débruitées en fonction de la variance de bruit gaussien, par différentes techniques de débruitage».

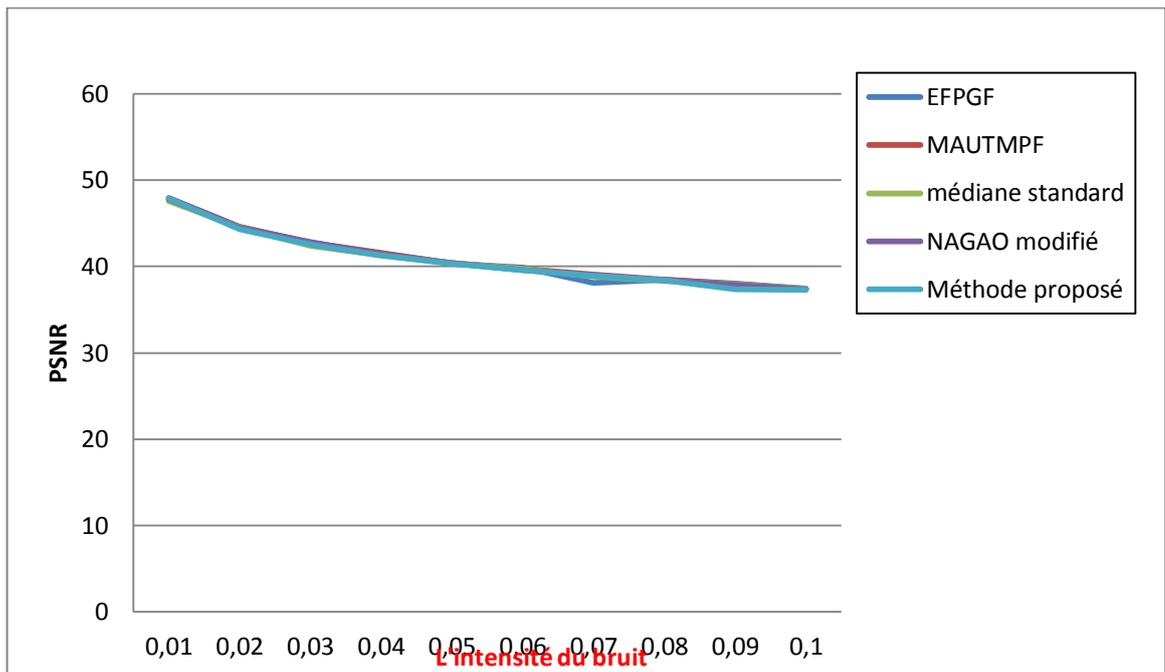


Figure II.7 : « Représentation du PSNR des images cameraman débruitées en fonction de la pourcentage de bruit impulsif, par différentes techniques de débruitage».

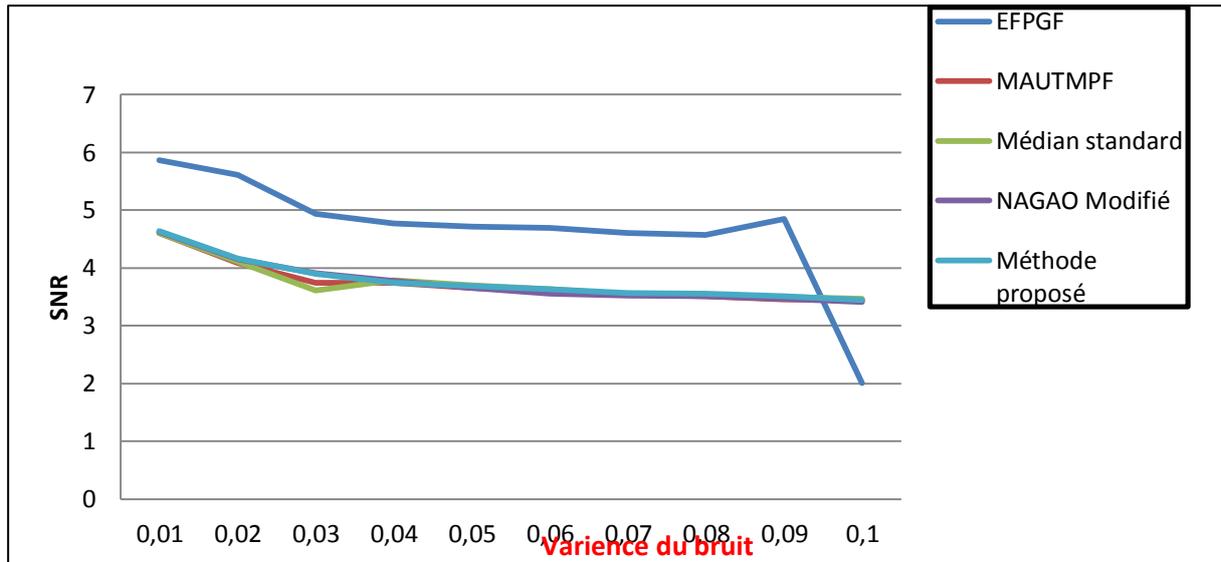


Figure II.8 : « Représentation du SNR des images cameraman débruitées en fonction de la pourcentage de bruit Gaussien, par différentes techniques de débruitage».

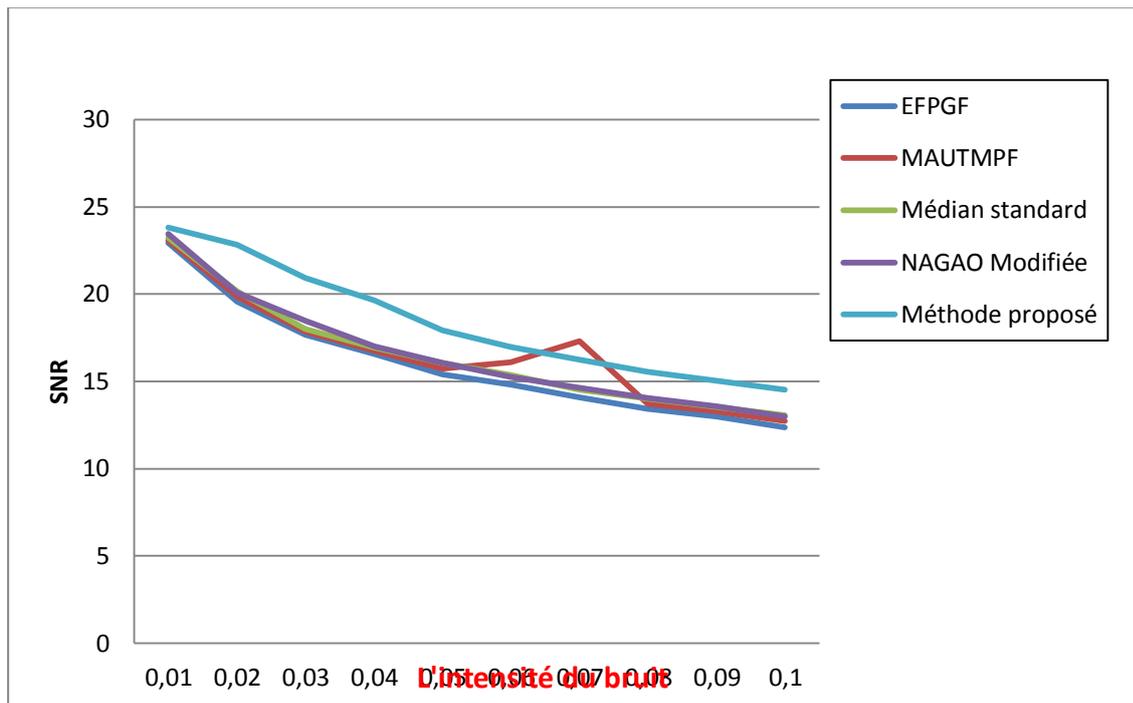


Figure II.9 : « Représentation du SNR des images cameraman débruitées en fonction de la pourcentage de bruit Impulsif, par différentes techniques de débruitage».

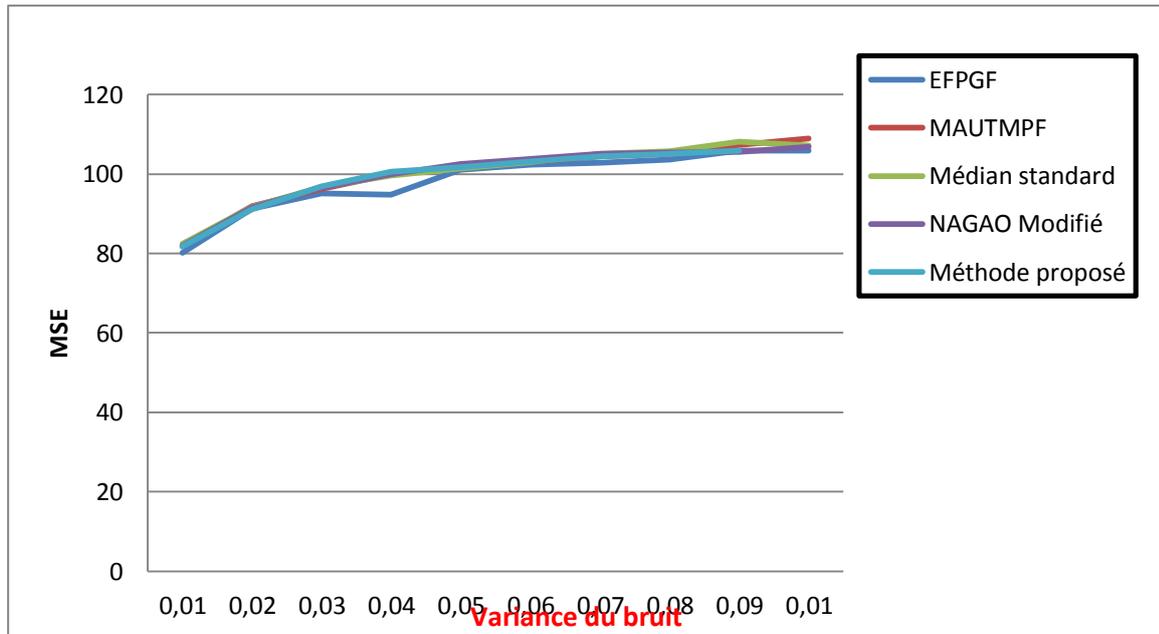


Figure II.10 : « Représentation du MSE des images cameraman débruitées en fonction de la pourcentage de bruit Gaussien, par différentes techniques de débruitage».

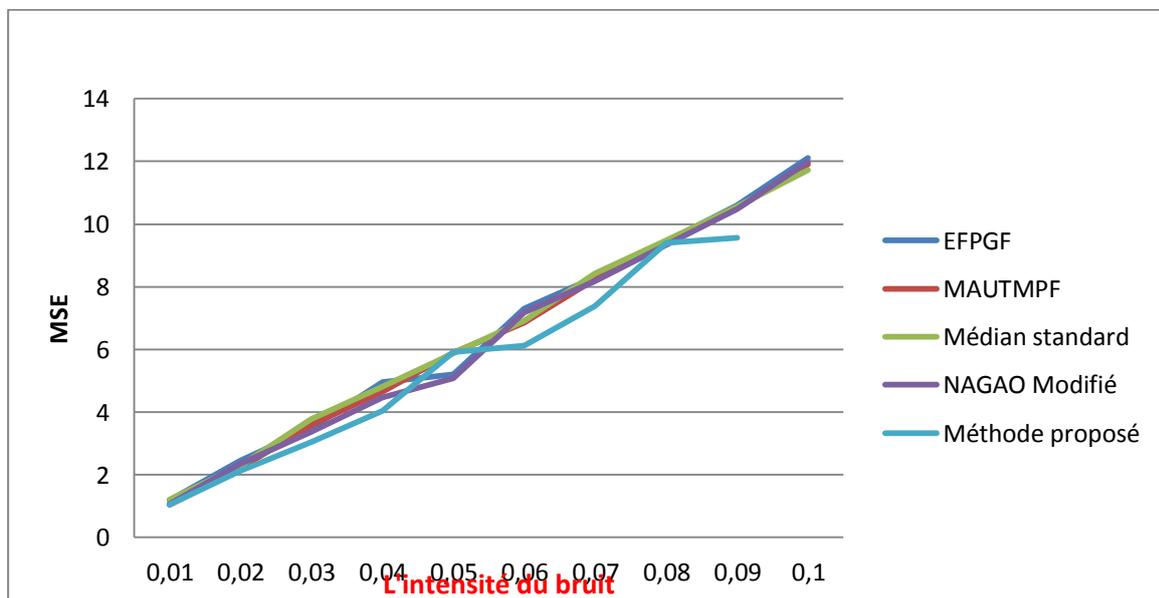


Figure II.11 : « Représentation du MSE des images cameraman débruitées en fonction de la pourcentage de bruit Impulsif, par différentes techniques de débruitage».

## II.8 Etude Subjective :

Nous présentons dans cette partie les résultats d'interprétation des méthodes de débruitage sur des images de différentes natures. Les programmes ont été exécutés sous l'environnement MATLAB.

### II.8.1 Bruit Gaussien :

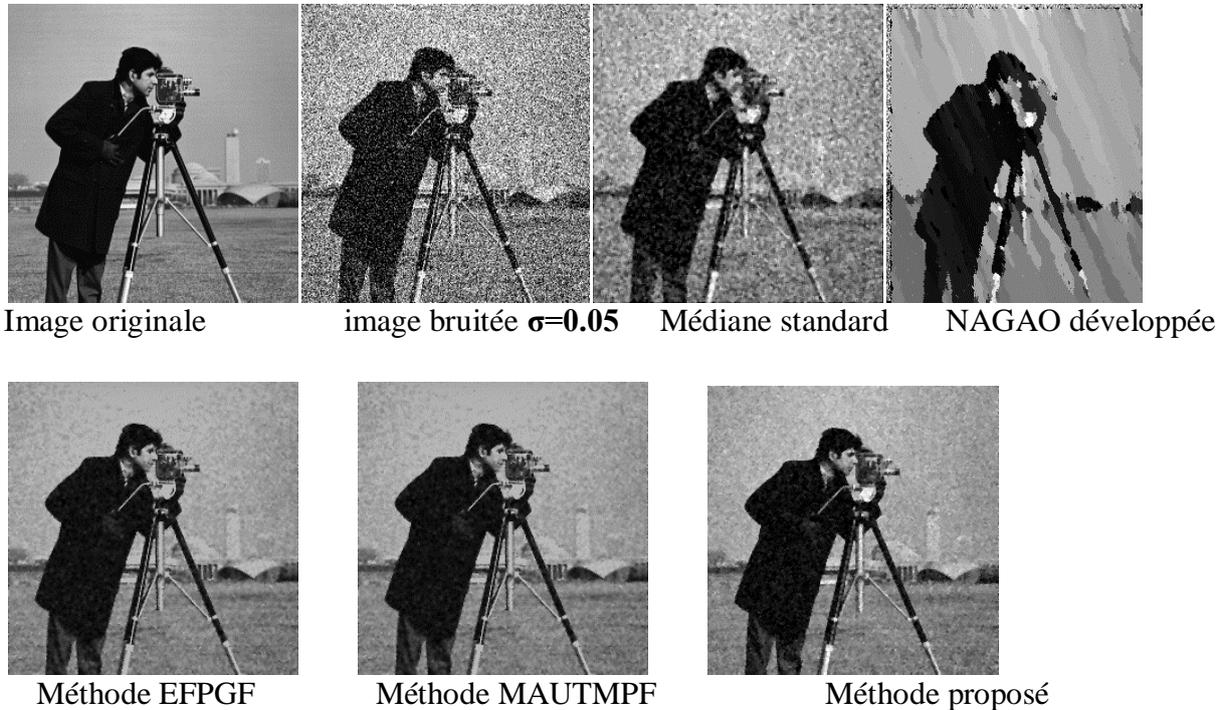


Figure II.12: « Images cameraman bruitée par bruit gaussien avec une variance  $\sigma=0.05$  d'ébruitée par différent techniques. »

## II.8.2 Bruit impulsif :

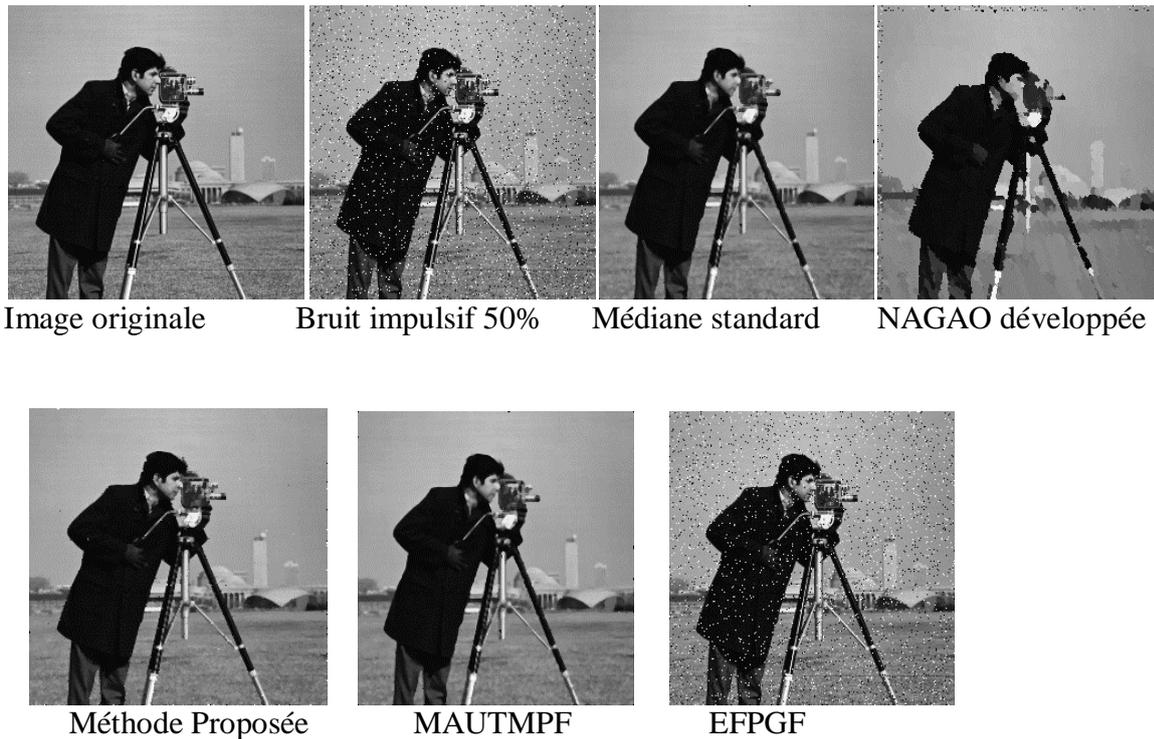


Figure II.13: « Images cameraman bruitée par bruit impulsif avec une pourcentage 50% d'ébruitée par différent techniques. »

- ✚ Nous avons appliqué masque (3x3) de filtre médian sur l'image (image Cameraman) bruitée par bruit sel et poivre, bruit gaussien. Nous avons calculé des mesures statistiques souvent utilisés en traitement d'images (PSNR, SNR, et MSE). Les résultats sont résumés dans les tableaux [II.1, II.2, II.3]
- ✚ Nous remarquons que le filtre Médian est plus efficace sur le bruit sel et poivre, avec une valeur maximale de PSNR et SNR, et avec une valeur minimale de MSE pour le masque 3x3.
- ✚ Nous avons appliqué le masque 5x5 du filtre NAGAO sur l'image cameraman bruitée, ensuite les SNR, PSNR, et MSE sont calculés.
- ✚ Nous voyons que malgré les améliorations obtenues sur le bruit sel et poivre et le bruit Gaussien, les meilleurs résultats sont ceux du bruit impulsif.
- ✚ Nous voyons que même si le bruit sel et poivre est éliminé, ce filtre influe sur l'image par Une amélioration de contraste mais avec une apparition de l'effet de bloc et aussi de l'effet de Bord.
- ✚ D'après les mesures statistiques obtenues nous avons remarqués que le EFPGF est bien adapté sur le bruit Gaussien
- ✚ D'après les mesures statistiques obtenues nous avons remarqués que le MAUTMPF est bien adapté sur le bruit Impulsif.
- ✚ D'après les mesures statistiques de la méthode proposée :
  - nous remarquons que l'image filtrée la plus proche à l'image originale est celle du bruit impulsif mais un effet de flou apparait et le bruit gaussien est diminué d'une façon remarquable.

## II.9 Conclusion :

Le choix du filtre pour le débruitage d'images numériques reste encore un défi scientifique pour les chercheurs. Il y a un bon nombre de techniques en traitement d'images disponibles à appliquer et supprimer le bruit dans une image. Dans notre étude nous nous intéressons aux cinq types de filtrage et aux deux types de bruit.

L'exécution des filtres est évaluée en utilisant les paramètres statistiques (qualité objective) et la qualité visuelle (subjective). Les résultats de comparaison prouvent que : les filtres qui donnent les meilleurs résultats sont la méthode proposée. Précisément Ce dernier est le filtre le plus performant, avec une meilleure préservation des détails et des Bords.

# CHAPITRE III :

**Etude générale de la Plate-formeTMS320C6713DSK**

### III.1 Introduction

L'objet de cette partie est d'étudier le DSP TMS320C6713. Néanmoins les informations fournies ne sont pas exhaustives et, pour une utilisation précise de telle ou telle partie, on pourra avantageusement utiliser les documentations fournies par le constructeur pour mieux détailler ce chapitre.

La carte DSK est supportée par le logiciel Code Composer Studio qui contient des outils de développement, y compris un compilateur de haute optimisation de C/C++, émulation basée sur le JTAG, débogueur à temps réel (real-time debugging).

### III.2 Présentation du TMS320C6713DSK

Des outils spécifiques de mise en œuvre matérielle sont nécessaires pour tester et intégrer un algorithme de traitement de la parole sur une plate-forme DSP dédiée. Le traitement des signaux numériques en temps réel a considérablement progressé après l'introduction de processeurs DSP spécialisés. Des kits de démarrage appropriés avec un processeur DSP spécifique et des outils logiciels associés tels que compilateurs, assembleurs, simulateurs, débogueurs, etc., sont fournis afin de faciliter la conception du système et le développement d'applications.

#### III.2.1 La famille TMS320C:

Cette famille de DSP englobe des processeurs 16-32 bits, à points fixes ou flottants. Aujourd'hui, la famille TMS320 est divisée en trois Plates-formes qui sont les TMS320c2000, TMS320c5000, TMS320c6000. cette dernière englobe trois type de processeur, les TMS320c62x, TMS320c64x, TMS320c67x. On peut voir sur la figure 2.1 les différents types de processeur de cette famille (TMS320c). [21]

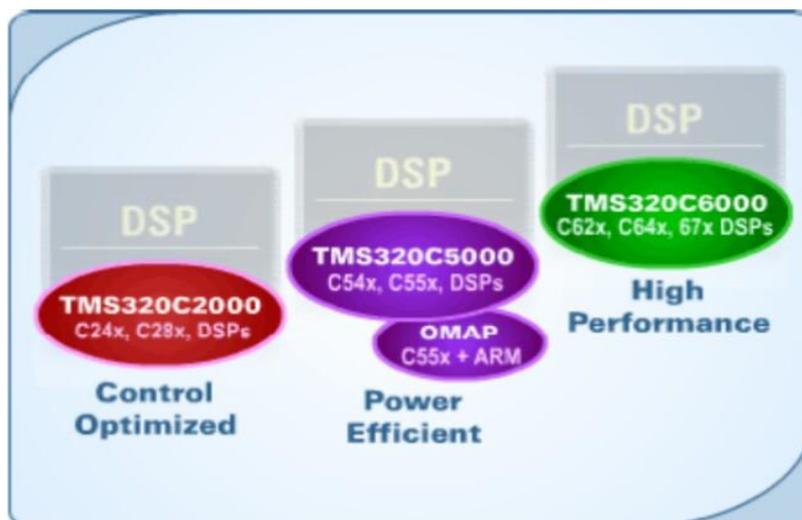


Figure III.1 : « La famille TMS320C »

### III.2.2 Les différentes familles de C6x :

#### III.2.2.1 TMS320C62X :

Ce sont les premiers C6x sortis par Texas Instruments en 1998. Avec des fréquences allant de 150 à 300 MHz, ils ont été les premiers DSP à architecture VLIW (very-long-instruction word).

#### III.2.2.2 TMS320C64X :

Ce sont les derniers DSP de gamme proposée par Texas Instruments. Comme les C62x, ce sont des DSP 16 bits virgule fixe. L'avantage de cette famille est qu'elle bénéficie des progrès technologiques et peut atteindre des fréquences allant de 300 à 600 MHz. Des efforts ont également été réalisés pour les transferts de données dans le processeur avec des bus 64 bits. [22]

#### III.2.2.3 TMS320C67X :

Ce sont les DSP à virgule flottante de la gamme. Les données sont de 64 bits, et les bus internes également. Les fréquences proposées vont de 100 à 225 MHz. Le jeu d'instructions est identique à celui du C62x, auquel ont été ajoutées des instructions pour les calculs spécifiques en virgules flottantes. Actuellement, aucune nouvelle version n'est annoncée. [23]

Application		Type de DSP	Caractéristiques
<b>TMS320C6000 DSP hautes performances</b>			
C62x	Applications exigeantes en vitesse : stations de base des réseaux de communications mobiles, équipement de radiodiffusion, réseaux informatiques	16 bits virgule fixe architecture VLIW	1200-2400 MIPS
C67x	Applications exigeantes en précision, dynamique et vitesse : Antennes adaptatives des stations de base, imagerie médicale, reconnaissance de parole, graphisme ...	32 bits virgule flottante architecture VLIW	600MFLOPS-1GFLOPS
<b>TMS320C5000 DSP optimisés en consommation</b>			
C54x	Applications de télécommunications exigeantes en coût, consommation, vitesse : terminaux mobile, voix sur IP, alphapages ...	16 bits virgule fixe	0,54 mW/MIPS 30-200 MIPS
<b>TMS320C2000 DSP optimisés pour les applications de contrôle</b>			
C20x	Applications de grands volume en téléphonie, électronique grand public, appareils photos numériques ou contrôleurs de disques durs ....	16 bits virgule fixe	PLL, UART, timers, mémoire flash intégrée 20-40MIPS
C24x	Applications de contrôle moteur, automatisation, robotique, contrôle d'appareils électroménagers... Bon compromis prix/performance	16 bits virgule fixe	Port série SCI, SPI et CAN, Convertisseur Analogique /numérique 20MIPS

Tableau III.1 : « Principales caractéristiques des trois plates-formes de DSP ».

### III.2.3 Caractéristiques principales du TMS320C6713 :

- DSP TMS320C6713 fonctionnant à 225 MHz.
- Codec stéréo de 16 bits (AIC23)
- Quatre (4) connecteurs permettent des entrées et des sorties analogiques:
  - MIC IN : entrée de microphone,
  - LINE IN: entrée analogique “ligne”
  - LINE OUT: sortie analogique “ligne”
  - HEADPHONE: sortie écouteurs/casque (multiplexée avec l’entrée ligne).
- Deux connecteurs de 80 broches permettent des extensions de périphériques et mémoire externes.
- Un émulateur JTAG intégré jumelé avec la connectivité USB.
- une interface USB avec l’ordinateur hôte.[20]

### III.2.4 L’architecture du TMS320c6713 :

Le TMS320C6713 est constitué de trois parties principales (figure suivante) :[20]

- l'unité centrale de traitement CPU.
- les périphériques.
- et la mémoire.

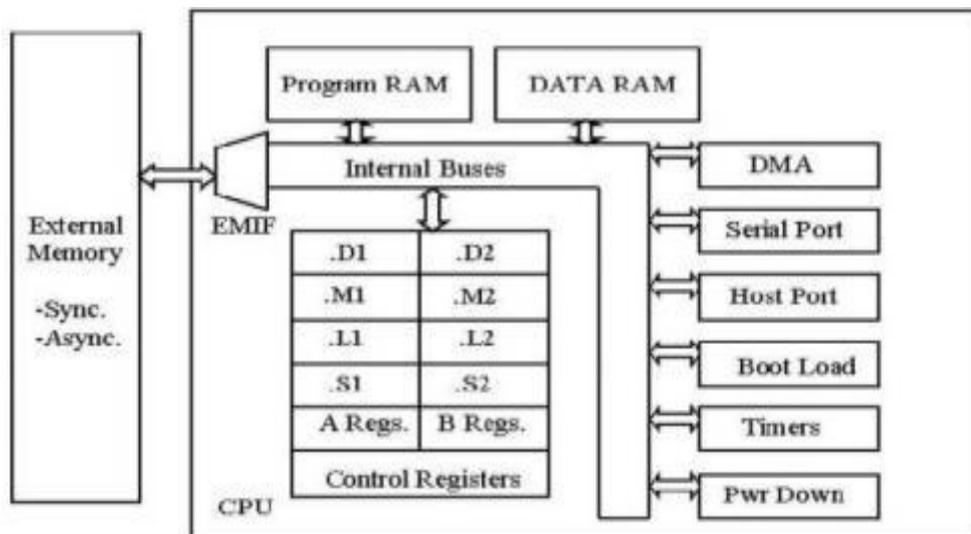


Figure III.2 : « Bloc diagramme simplifié de la famille TMS320C67xx »

#### III.2.4.1 Unité centrale de traitement (CPU) :

Le CPU est constitué d'une unité de contrôle de programme, de deux unités fonctionnelles, de deux blocs de 16 registres de 32 bits, de contrôleurs d'interruptions et d'autres éléments.

**III.2.4.1.1 Unité de contrôle de programme :**

Elle est constituée des éléments suivants,

- Unité "fetch" programme : Elle a pour rôle récupérer les programmes. Cette opération se déroule en quatre phases :

- Phase PG: l'adresse du code est générée.
- Phase PS : l'adresse est envoyée à la mémoire.
- Phase PW: l'attente de lecture du code de la mémoire.
- Phases PR : la lecture du code.

- Unité "dispatche" de l'instruction: le code récupéré de la mémoire est affecté à l'unité fonctionnelle associée.

- Unité de décodage de l'instruction: elle a pour rôle de décoder l'instruction.

**III.2.4.1.2 Unités fonctionnelles :**

Le CPU contient huit unités: fonctionnelles divisées en deux parties 1 et 2. Leurs fonctions sont les suivantes:

- Unités .M1 et .M2 : ces unités sont dédiées à la multiplication.

- Unités .L1 et .L2 : ces unités sont dédiées à l'arithmétique et la logique.

- Unités .D1 et .D2 : ces unités sont dédiées au chargement, la sauvegarde et calcul d'adresse.

-Unités .S1 et .S2: ces unités sont dédiées pour le décalage de bit, l'arithmétique, la logique et le branchement.

Les Unités fonctionnelles	Leurs rôles
.M1. M2 (multiply)	Effectue toutes les opérations de multiplication en un seul cycle d'horloge : -une multiplication 32 x 32 bit. -deux multiplications 16 x 16 bit. -deux multiplications 16 x 32 bit, 4 multiplications 8 x 8bit.
.L1 .L2 (arithmetical)	Effectue en parallèle les opérations +/- sur 32 bit ou 2 fois 16 bit
.S1 .S2	Parallélise les instructions 8 bits /16 bits duales 16 bits. Il réalise aussi en parallèle des opérations MIN et MAX.
.D1 .D2	Charge les données de la mémoire vers les registres et stocke les résultats des registres dans la mémoire.

Tableau III.2 : « Le rôle des unités fonctionnelles ».

**III.2.4.1.3 Registres :**

Le CPU contient 32 registres de 32 bits divisés en deux blocs égaux : registre fichier A (AO-A15) et registre fichier B (BO-B15), leurs fonctions sont réparties comme suit: - Les registres A1-A2 et BO-B1-B2 : ils sont utilisés comme registres conditionnels. - Les registres A4-A7 et B4-B7: ils sont utilisés pour adressage circulaire. - Les registres AO-A9, BO-B2 et B4-B9 : ils sont utilisés comme registres temporaires. - Les registres A10-A15 et B10-B15 : ils sont utilisés pour la sauvegarde et la restitution de données d'un sous-programme. À ces 32 registres s'ajoutent les registres de contrôles et d'interruptions.

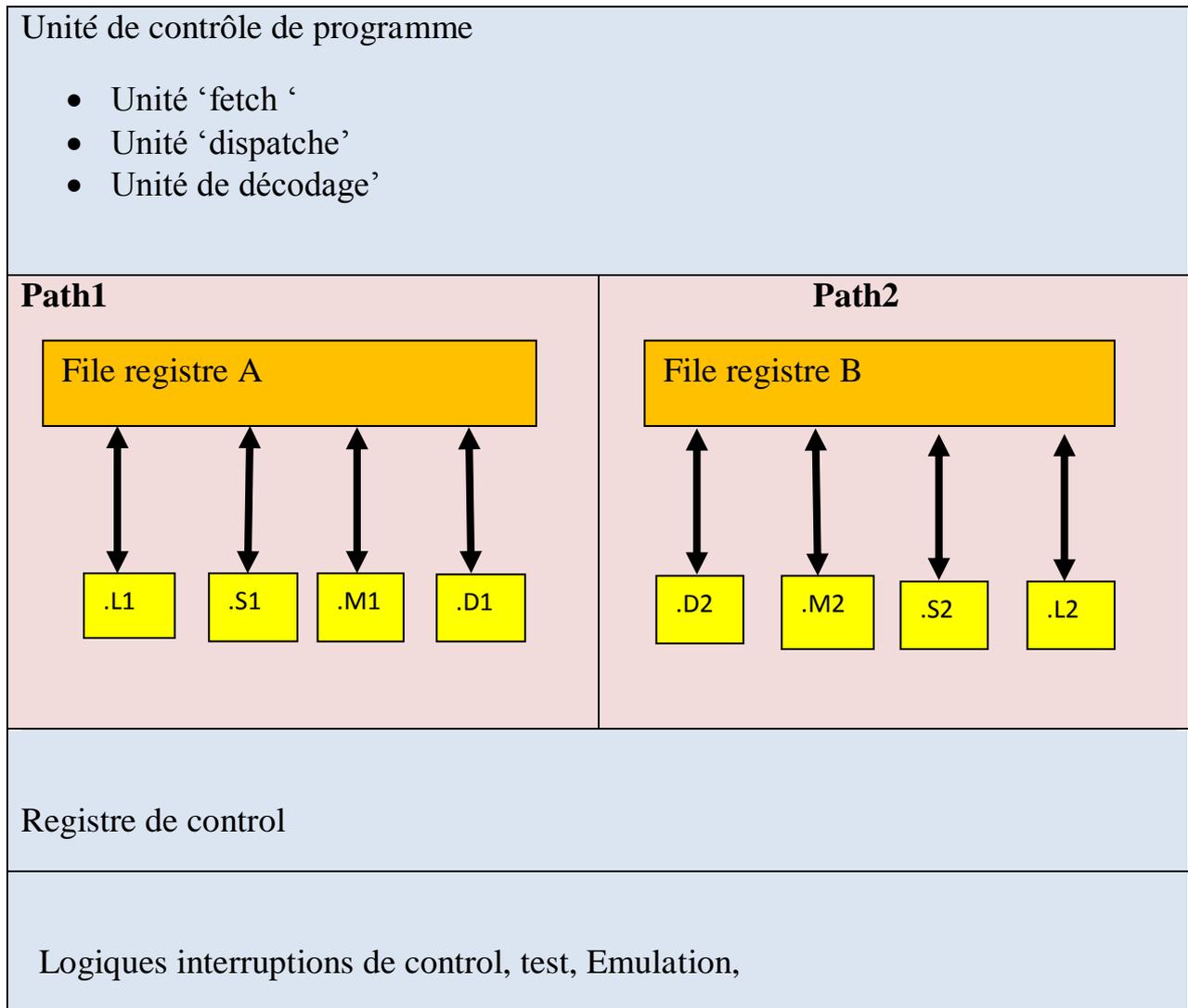


Tableau III.3 : « L'unité centrale de traitement CPU»

### III.2.4.2 Les périphériques du TMS320C6713 :

Le TMS320C6713 a plusieurs périphériques qui sont :

- Le contrôleur DMA : Il permet sans l'aide du CPU de transférer des données entre les espaces mémoire (interne, externe et des périphériques). Il a quatre canaux programmables et un autre canal auxiliaire.
- Le contrôleur EDMA : Il permet le transfert des données entre les espaces mémoire comme le DMA. Il a 16 canaux programmables.
- L'interface port hôte HPI. Il donne au processeur hôte un contrôle total pour un accès direct de l'espace mémoire du CPU et à la cartographie de la mémoire des périphériques du DSP.
- Deux McBSP qui est des ports séries multi-canaux protégés. Ils permettent la communication avec les périphériques externes. Ils ont la même structure. Ils supportent une communication full-duplex.
- L'interface de mémoire externe EMIF : Il permet l'interface avec plusieurs éléments (mémoires) externes.
- Les compteurs : Le DSP possède deux compteurs qui peuvent être synchronisés par une source interne ou externe et ils sont utilisés comme générateurs de pulsations, compteurs d'événements externes, interrupteurs du CPU après l'exécution de tâches et déclencheur du DMA/EDMA.
- Les interruptions : l'ensemble des périphériques contient jusqu'à 32 sources d'interruptions.

### III.2.4.3 La structure de la mémoire

#### a) Mémoire/périphérique :

- 64K octets L1P de mémoire cache de programme,
- 64K octets L1D de mémoire cache de données,
- 64 K octets L2 de mémoire cache RAM.
- 32 bits interface de mémoire externe (EMIF) ;
- contrôleur d'accès mémoire direct (EDMA) ;
- un port parallèle de 16 bits (HPI) ;
- 2 bus série (McBSP) ;
- 2 timers de 32 bits ;
- générateur d'horloge par PLL

#### b) Mémoire interne

La mémoire interne a une taille de 260 KB qui est décomposée en deux niveaux :

Le niveau (L1) est constitué de deux mémoires caches de 4 KB chacune, (L1P) qui est utilisée pour les programmes et (L1D) qui est utilisée pour les données.

Le Niveau (L2) est composé de 256 KB de mémoire partagée entre mémoire des données et mémoire de programmes.

### III.3 Présentation du Code Composer Studio CCS :

L'évaluation des performances des algorithmes sur le DSP est effectuée en utilisant Code Composer Studio (CCS), son fonctionnement est illustré par la figure suivante :[19]

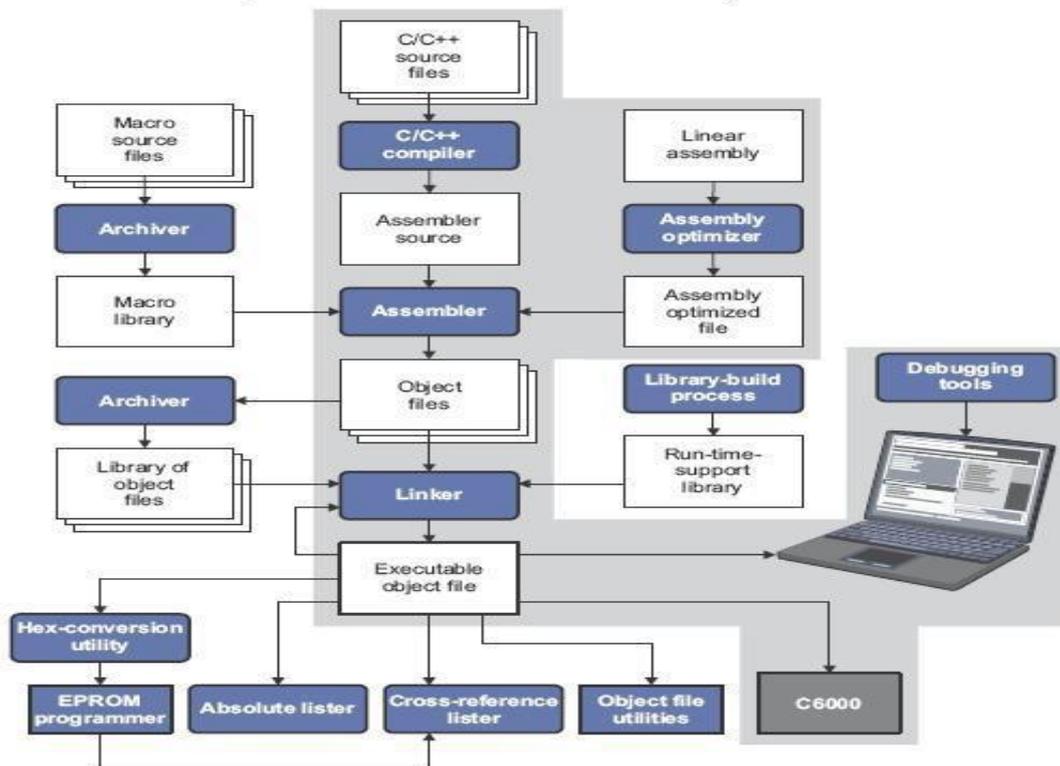


Figure III.3 : « Structure du système du développement du TMS320C6713 ».

Le logiciel Code Composer Studio est une plate-forme de développement qui inclut les éléments suivants,

- **Environnement de développement intégré (IDE)**: il permet l'édition ('built'), et la correction ('debug') des programmes destinés au DSP.

- **Outils de génération du code pour le TMS320C6000**: ces outils sont le compilateur, l'assembleur et l'éditeur de lien.

- **Le compilateur C/C++** permet de compiler le programme source (xxx.c) pour le convertir en assembleur (xxx.asm), l'assembleur reçoit le fichier xxx.asm et le convertit en langage machine ou fichier objet (xxx.obj), enfin l'éditeur de liens (linker) qui combine les fichiers objet et les fichiers bibliothèques et le fichier xxx.cmd pour produire un fichier exécutable avec une extension .out, c'est ce fichier qui sera chargé sur le processeur C6713 pour être exécuter.

- **DSP/BIOS**: c'est un outil d'analyse en temps réel, pour s'en servir, on doit créer un fichier de configuration 'xxx.cdb', où seront définis les objets utilisés par l'outil DSP/BIOS. Ce fichier permet aussi de faciliter l'organisation de la mémoire et la gestion du vecteur des interruptions, en offrant la possibilité de les faire sur un environnement visuel via la section de gestion de la mémoire MEM (Memory Section Manager), et via HWI (Hardware Interrupt Service Routine Manager) pour les interruptions.

- **JTAG** (Joint Team Action Group) et RTDX (Real Time Data Exchange) le RTDX permet un échange de données en temps réel entre l'hôte (PC par exemple) et la destination (la carte DSK dans notre cas), il permet aussi l'analyse et la visualisation des données au cours de l'exécution du programme, alors que le lien JTAG est utilisé pour atteindre l'émulateur (qui se trouve à l'intérieure du DSP), c'est ce dernier qui permet au CCS de contrôler en temps réel l'exécution du programme.

- **Simulateur intégré** : le logiciel Code Composer Studio offre la possibilité de tester des programmes pour DSP sans utiliser la carte à l'aide d'un simulateur intégré. L'application CCS fournit tous les outils Software nécessaires pour le développement du DSP

### III.3.1 Environnement de développement DSP :

Les exécutables sur DSP sont élaborés en utilisant des environnements de développement intégrés (IDE) fournies par les fabricants des DSP, ils intègrent de nombreuses fonctions, telles que l'éditeur, le débogage, la gestion des fichiers de projet, et le profilage.

L'environnement de développement pour Texas Instrument (TI) est appelé « Code Composer Studio ».

Le constructeur TI met à disposition gratuitement le compilateur, l'assembleur et l'éditeur de liens optimiseur pour une utilisation non commerciale.

La figure III.4 montre un exemple d'un écran typique du Code Composer. Sur le côté gauche il y a la liste de tous les fichiers inclus dans le projet.

Au centre de l'écran deux fenêtres affichent le code, comme un fichier C (process.c) et le code assembleur (fenêtre Disassembly).

L'exécution s'est arrêtée sur un point d'arrêt (breakpoint) activé au préalable. A dessous des fenêtres de codes, deux fenêtres de mémoire sont également visibles, détaillant les données présentes à partir des adresses 0x80000000, et à partir des adresses 0x40000030. La donnée à l'adresse 0x80000002 est d'une couleur différente parce que sa valeur a changé récemment.

Au bas de l'écran de l'IDE, on trouve :

- la fenêtre de compilation/Link, qui donne les résultats de la dernière compilation du code.
- la fenêtre Watch, qui affiche les valeurs prise par deux variables.
- et la fenêtre Register, qui affiche le contenu de tous les registres du DSP.

Sur le côté droit il y a trois graphiques: les jaunes montrent les régions de mémoire, tandis que le vert représente la transformée de Fourier rapide de données stockées en mémoire, tel que calculé par l'IDE. [22]

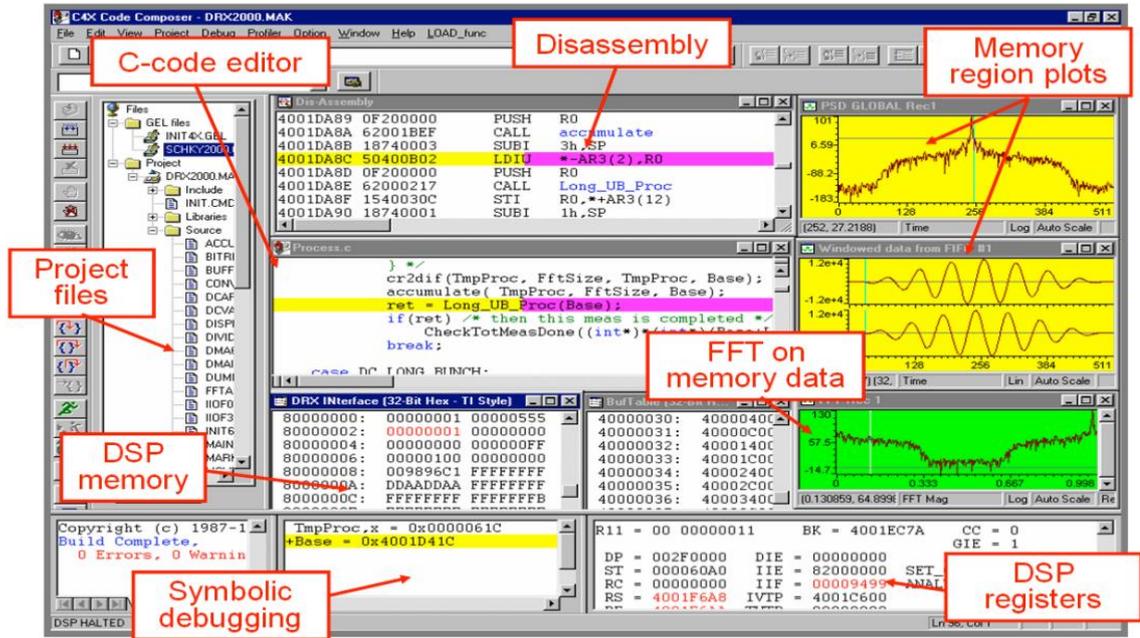


Figure III.4: « Capture d'écran du Code Composer Studio ».

### III.4 Conclusion :

Dans ce chapitre, Premièrement nous avons décrit le DSP TMS320C6713, et nous avons présenté son architecture interne, ces caractéristiques pour mieux comprendre son fonctionnement global. Dans la deuxième partie de ce chapitre nous avons présenté le logiciel « Code Composer Studio » et comment implémenter des filtres numériques sous DSP Plate-forme.

# CHAPITRE IV :

**Application de l'algorithme Proposé sur TMS320C6713DSK**

---

## IV.1 Introduction

Les systèmes embarqués utilisent plusieurs technologies pour une fiabilité et des temps d'exécution acceptables. La technologie principale utilisée est le DSP (Digital Signal Processor). En effet, ils sont des processeurs spécifiques au traitement du signal dans des images particulières. Ils se caractérisent par un pouvoir important des calculs arithmétiques et logiques en temps réel.

Ce chapitre explique comment installer et tester l'environnement de développement intégré (IDE) de Code Composer Studio (CCS) de Texas Instruments, que ce soit pour le TMS320C6713 ou le Kit de démarrage du traitement du signal numérique DSP. Le système proposé présente les fonctionnalités matérielles et logicielles de DSK et CCS Version 3.1.

Dans un premier temps, une architecture du bruit Gaussien a été développée et simulée sur Matlab. Par la suite, une deuxième architecture du bruit impulsif a été développée et implémentée sur DSP via l'environnement de développement Code Composer Studio.

## IV.2 Présentation du Model Simulink Proposé pour l'implémentation

Notre modèle simulink développé utilise des différents Blocksets de « Video and Image Processing Blockset », on présente ces différents blocks comme suit :

N°	Blocks	Library	Quantity
1	Image From File	Video and Image Processing Blockset> Sources	01
2	Median Filter	Video and Image Processing Blockset> Analysis & Enhancement	01
3	Video Viewer	Video and Image Processing Blockset> Sinks	05
4	Embedded MATLAB Function	Simulink> User-Defined Functions	06
5	Image data type conversion	Video and Image Processing Blockset> Conversions	02
6	PSNR	Video and Image Processing Blockset> Statistics	05
7	Color space conversion	Video and Image Processing Blockset> Conversions	05

Tableau IV.1 : « Différents Simulink Blocksets Utilisés »

Dans notre modèle Simulink, on a le block « original image » permet de définir le lien de l'image choisie et faire une lecture de fichier puis l'image sélectionnée sera convertie de RGB à l'intensité à l'aide du block « Color Space conversion » on ajoute un bruit « add salt and pepper with noise ratio », puis l'image sera convertie en double à l'aide du block « image data type conversion » on ajoute 5 filtres permet d'éliminer le bruit lié à l'image pour avoir une image sans bruit (filtré) puis elle sera convertie en uint8 à l'aide du block « convert image to uint8 » l'image pénètre dans le I1 de MSE « Mean Squar Error » et le I2 de « PSNR ».

L’architecture de notre Model Simulink proposé est présentée comme suit :

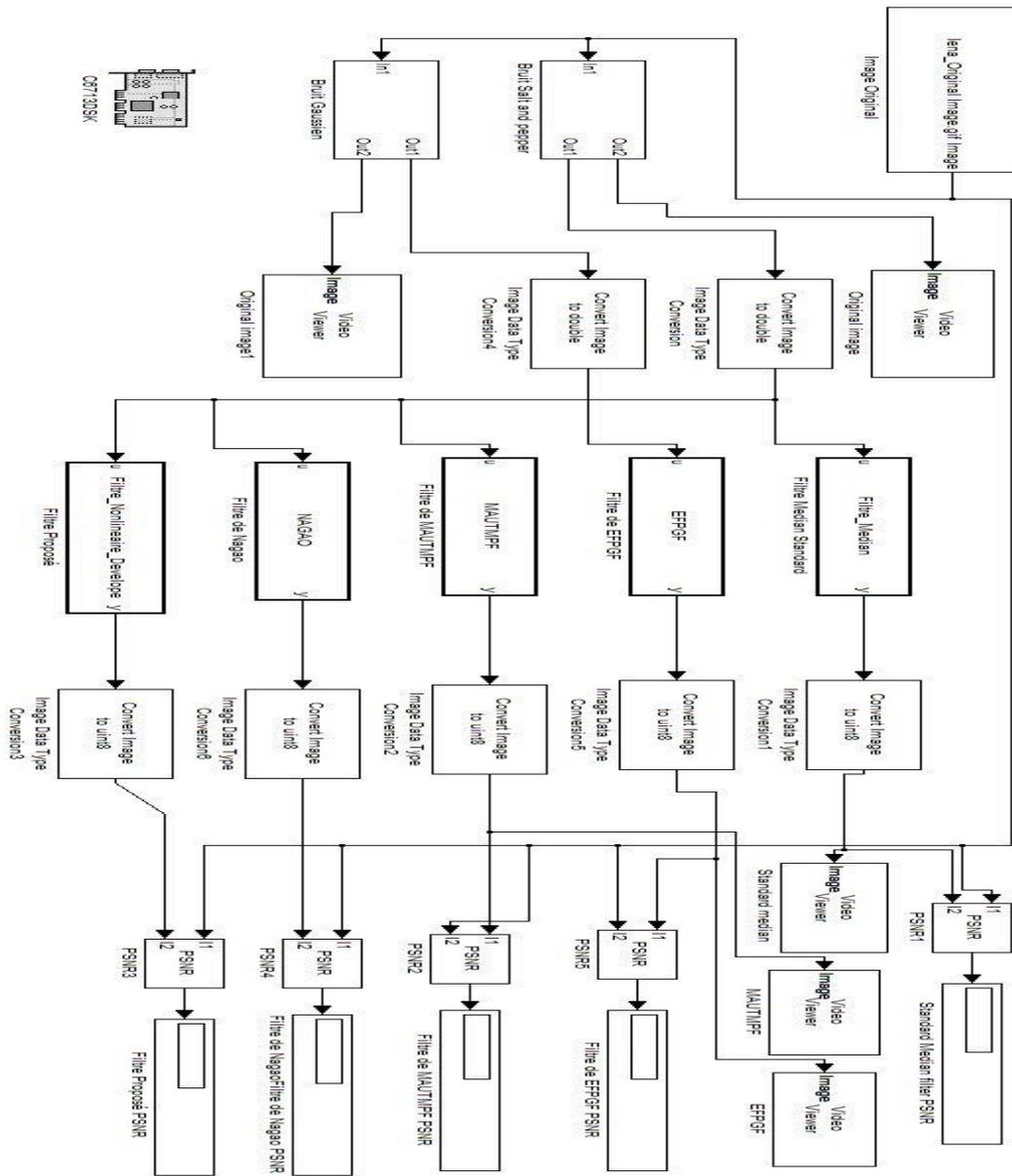


Figure IV.1 : « Architecture du Model Simulink proposé »

Concernant les paramètres de chaque block, ils sont bien détaillé sauf le bloc « Embedded MATLAB Function » qui contient le code Matlab.

### IV.3 Simulation et résultats obtenus à l’aide de Matlab Simulink

En montrant nos résultats, nous avons cherché à appliquer notre fonction de réduction du bruit d’origine à une image. La première partie de ce laboratoire nous a impliqué la saisie d’une image parfaitement normale et l’ajout de bruit de sel et de poivre à l’image. Nous avons utilisé des différentes valeurs de l’intensité du bruit. Ce processus est illustré à la Figure 1. Après

avoir entré ces images dans Matlab, nous avons cherché à traiter les données avec nos différents algorithmes proposés, les filtres de réduction du bruit utilisés sont :

- Filtre de NAGAO modifiée,
- Méthode d'EFPGF,
- Filtre Médiane Standard,
- Méthode de MAUTMPF,
- Filtre proposé,

À l'aide du filtre de réduction du bruit d'origine, nous avons pu constater des améliorations remarquables sur les images utilisées. Notre appartenance fonctionne à la fois pour le bruit positif et négatif intersecté à 255. C'est le point médian de cette échelle, et il ressort clairement des images que le bruit a été réduit de manière égale. La figure 3 montre le changement que cette fonction avait sur nos images claires et sombres.

Pour exécuter notre model Simulink proposé en mode Matlab Simulink il faut choisi le menu « simulation » puis on appuyé sur « Start » ou bien « Ctrl+T ». On obtient les résultats suivants :

#### IV.4 Implémentation des Algorithmes sur TMS320C6713 DSK

Comme décrit précédemment, MATLAB a développé un outil ‘Link for code composer studio’ qui lui permet de communiquer avec le code composer studio et on utilisant cet outil ; le model SIMULINK est traduit en projet écrit en C qui pourrai être exécuté sur CCS. Sur Simulink, il y a toute une librairie contenant des blocks spécifiques pour chaque type de carte DSP, dans notre model on ajoute le block C6713, et on choisit le type de la carte et le type de processeur et la fréquence d'horloge.

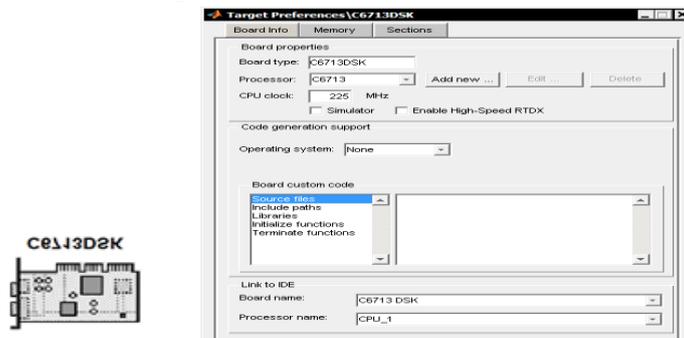


Figure IV.2 : « Le Block du C6713 »

##### IV.4.1 Instruction pour la connexion de la carte

Cette procédure est à réaliser avec attention chaque fois que pour une raison ou une autre la connexion USB ne fonctionne plus.

- Assurer que le code composer studio CCS est fermé.
- Brancher l'alimentation de la carte DSK : d'abord l'alimentation au secteur, puis l'alimentation continue à la carte.

- Attendre que les LEDs aient clignoté puis restent allumées.
- Brancher le câble USB.
- Lancer CCS (Code Composer Studio)
- Se connecter à la cible : Debug>Connect

Une icône verte doit apparaître en bas à gauche de la fenêtre de CCS pour préciser que la carte est bien connectée au PC.

Puis il faut faire une vérification du bon fonctionnement de la carte, si elle est correctement connectée au PC, Pour cela les étapes suivantes sont à suivre :

Vérification du bon fonctionnement de tous les périphériques de la carte DSK :

Double clique sur l'icône « 6713DSK Diagnostics Utility » puis sur le bouton « Start »

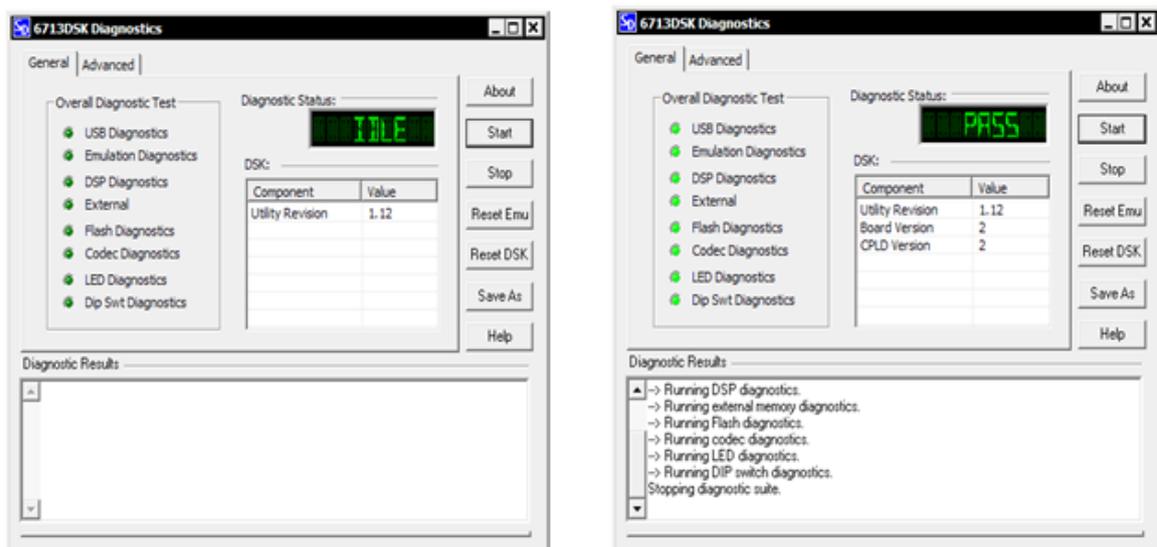


Figure IV.3 : « Vérification de la bonne marche de la carte DSK. »

#### IV.4.2 Les Paramètres de configuration

Pour régler les paramètres de la simulation, on clique sur « simulation » puis « Configuration Paramètres » ou bien sur Ctrl+E. La fenêtre suivante apparaît :

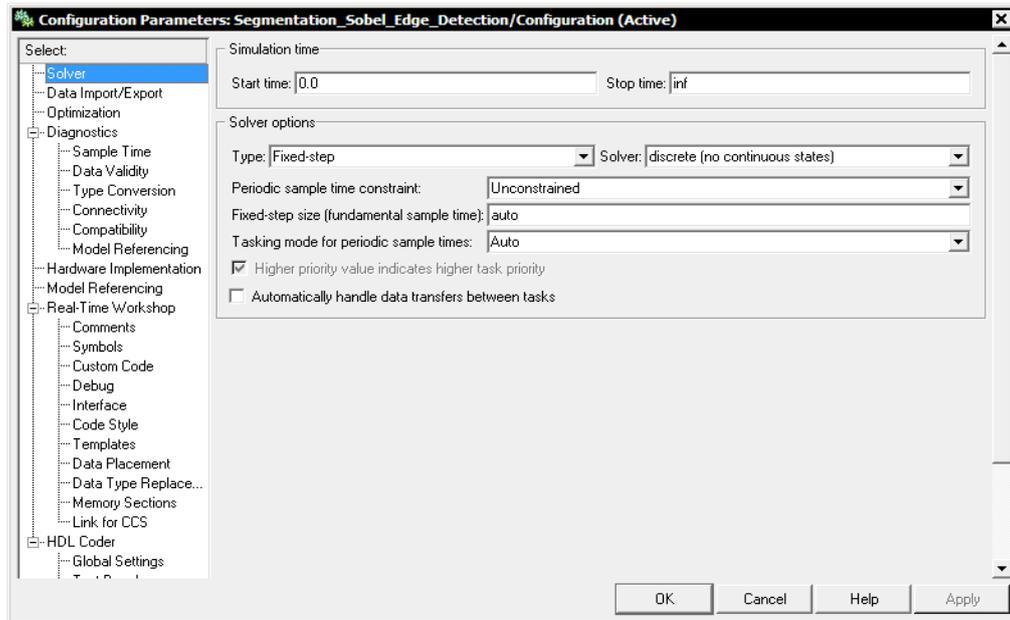


Figure IV.4 : « Fenêtre de la configuration des Paramètres. »

Sélectionner le champ « Real-Time Workshop » pour modifier les paramètres de l'application, dans le champ « Build action » sélectionner l'option « Creat\_projet ».

N.B : il existe d'autres options sur ce champ qui permettent la création et le chargement du programme sur la carte DSK.

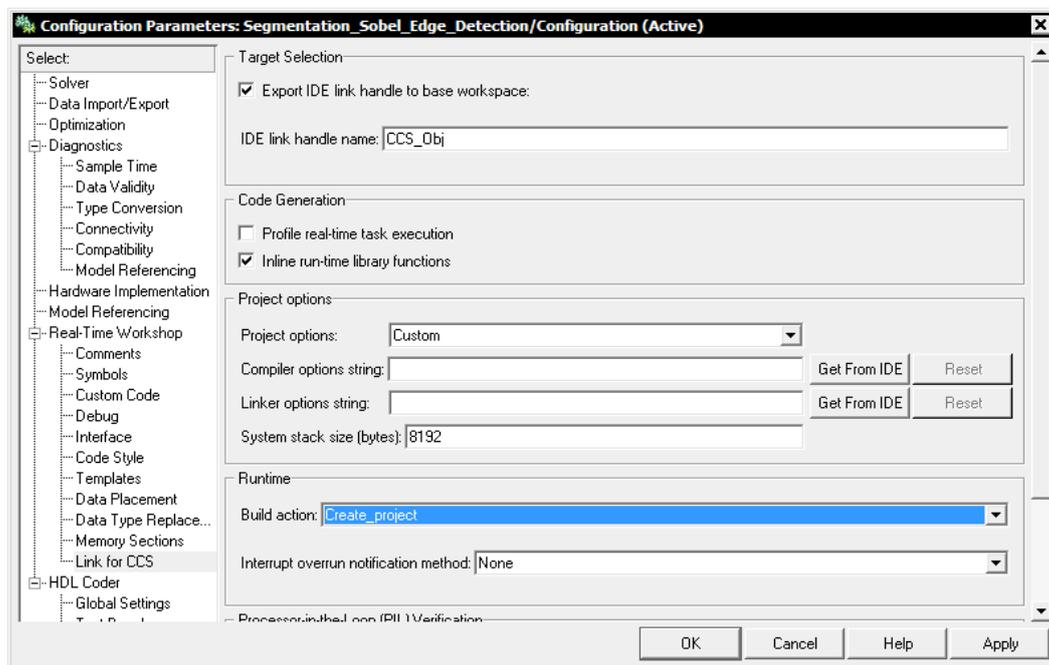
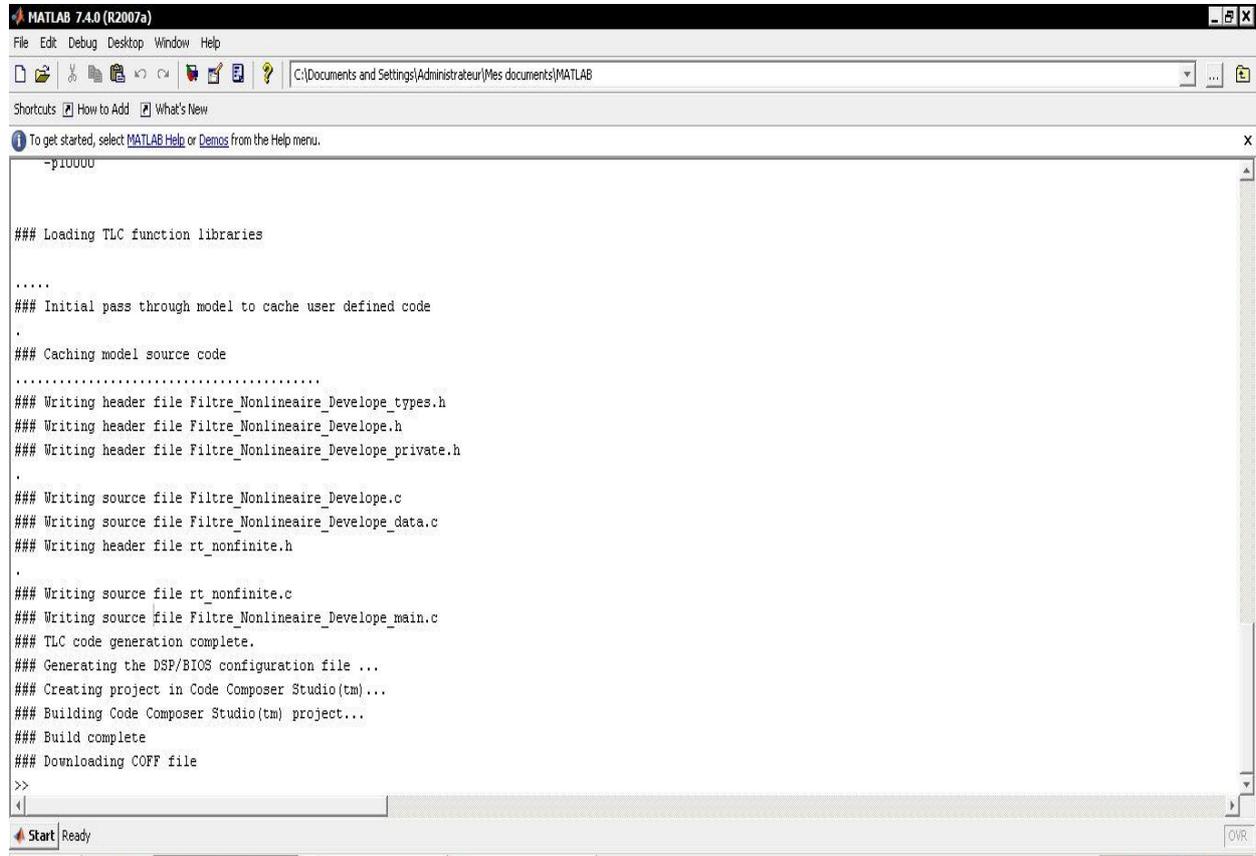


Figure IV.5 : « Paramètre du Link for CCS. »

#### IV.4.3 Création d'un projet en « C/C++ » à partir d'un model Simulink

Concernant le passage de Matlab au code composer studio CCS, MATLAB a développé un outil "Link for code composer studio" qui lui permet de communiquer avec le code composer

studio et on utilisant cet outil ; le model SIMULINK est traduit en projet écrit en C qui pourra être exécuté sur CCS, pour des applications en temps réel.  
 Pour démarrer la création du projet en cliquant sur « Incremental build » ou bien sur Ctrl+B, Matlab crée un lien avec le code composer studio et génère le projet écrit en « C ».



```

MATLAB 7.4.0 (R2007a)
File Edit Debug Desktop Window Help
C:\Documents and Settings\Administrateur\Mes documents\MATLAB
Shortcuts How to Add What's New
To get started, select MATLAB Help or Demos from the Help menu.
-#100000
### Loading TLC function libraries
.....
### Initial pass through model to cache user defined code
.
### Caching model source code
.....
### Writing header file Filtre_Nonlineaire_Develope_types.h
### Writing header file Filtre_Nonlineaire_Develope.h
### Writing header file Filtre_Nonlineaire_Develope_private.h
.
### Writing source file Filtre_Nonlineaire_Develope.c
### Writing source file Filtre_Nonlineaire_Develope_data.c
### Writing header file rt_nonfinite.h
.
### Writing source file rt_nonfinite.c
### Writing source file Filtre_Nonlineaire_Develope_main.c
### TLC code generation complete.
### Generating the DSP/BIOS configuration file ...
### Creating project in Code Composer Studio(tm)...
### Building Code Composer Studio(tm) project...
### Build complete
### Downloading COFF file
>>
  
```

Figure IV.6 : « Le passage du Simulink au « C/C++ ». »

#### IV.4.4 Différentes opérations pour l'implémentation sur TMS320C6713 DSK

Dans la liste du projet à gauche de la fenêtre du CCStudio, cliquez sur le programme (C/C++) ajouté. Ensuite dans le menu principal cliquez sur « **Project** » et choisissez « **Compile File** ».

Après la compilation du programme, construisez le projet en cliquant dans le menu de « **Project** » puis sur « **Rebuild all** ».

Après la construction du projet, le programme doit être chargé sur la carte. Dans le menu principal choisissez « **File** » et ensuite « **Load Program** ». Une nouvelle fenêtre s'ouvrira dans laquelle vous choisissez le fichier « **Noise\_Reducer.out** » et cliquez sur « **Open** ».

Pour exécuter ce programme, cliquez dans le menu « **Debug** » sur « **Go Main** » et ensuite dans le même menu « **Debug** » cliquez sur « **Run** ».

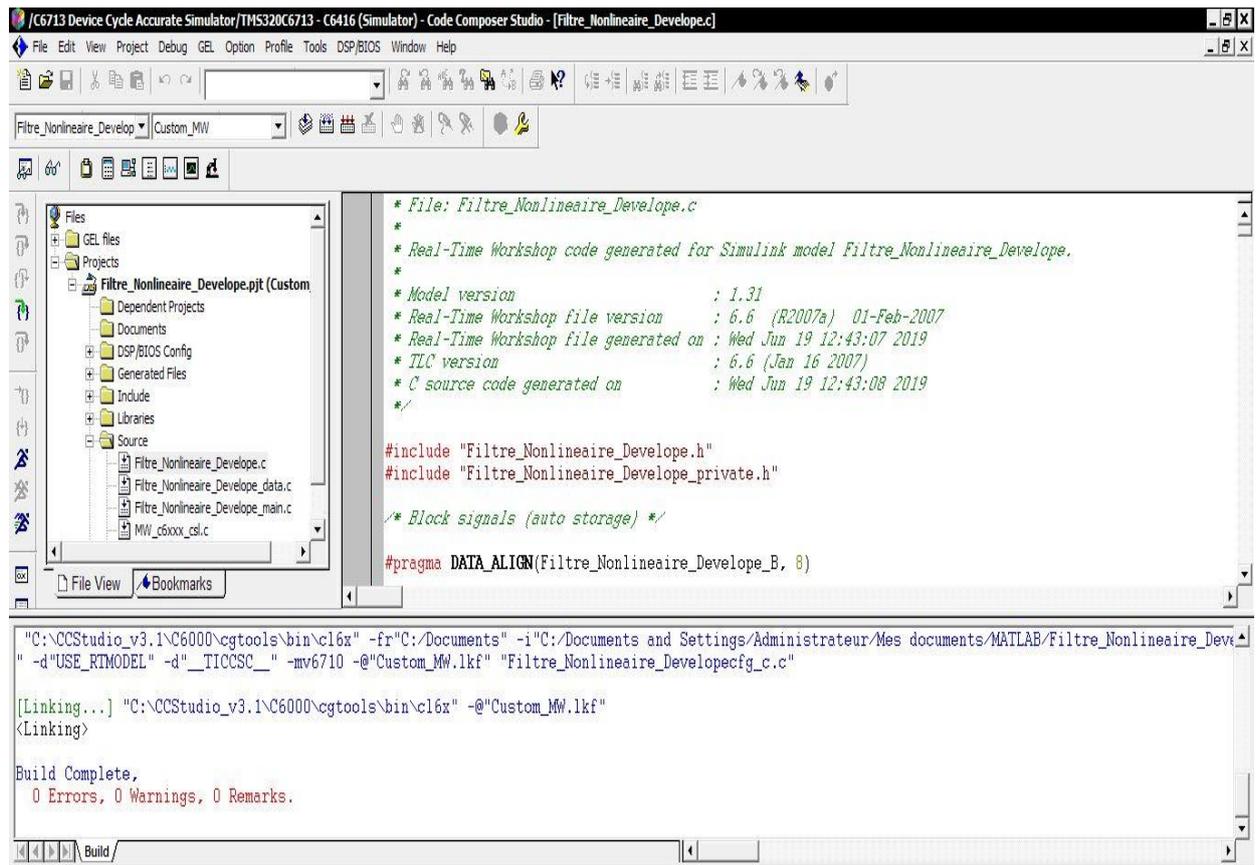


Figure IV.7 : « Création du projet Filtre Non-Linéaire développé sur CCS. »

#### IV.4.5 Résultats obtenu à l'aide du Code Composer Studio CCS

Les figures ci-dessous montrent les résultats de l'application du filtre proposé sur des images numériques :

IV.4.5.1 l'image original à l'aide du CCS

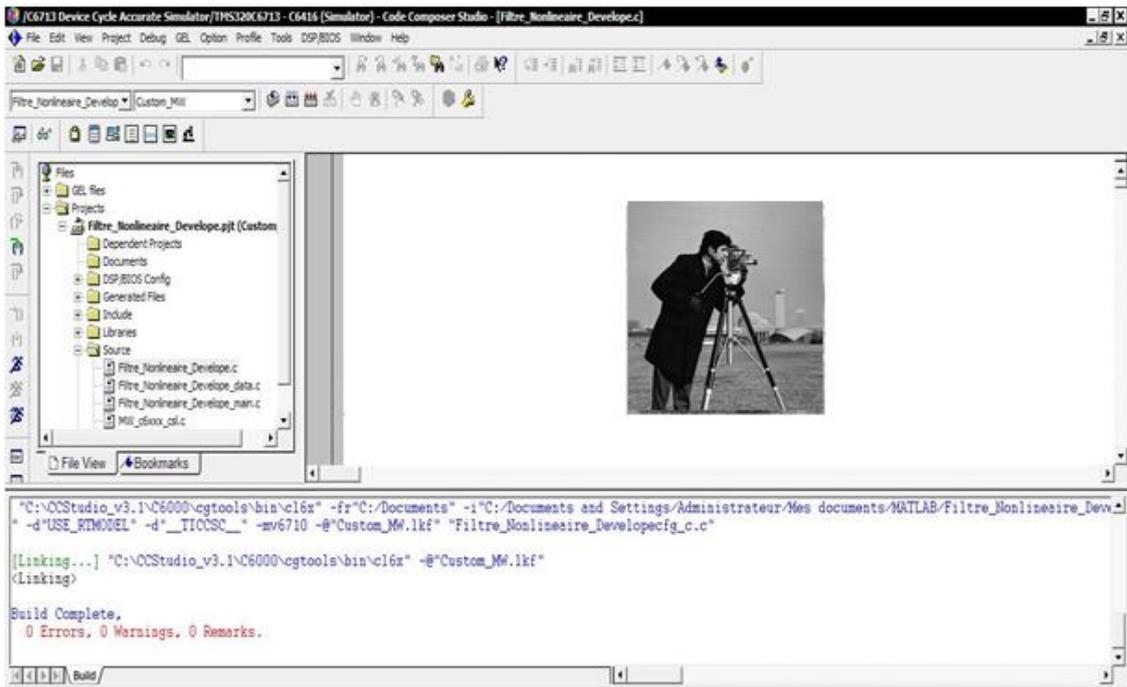


Figure IV.8: « Capture d'écran de l'image original à l'aide du CCS. »

IV.4.5.2 l'image bruitée à l'aide du CCS

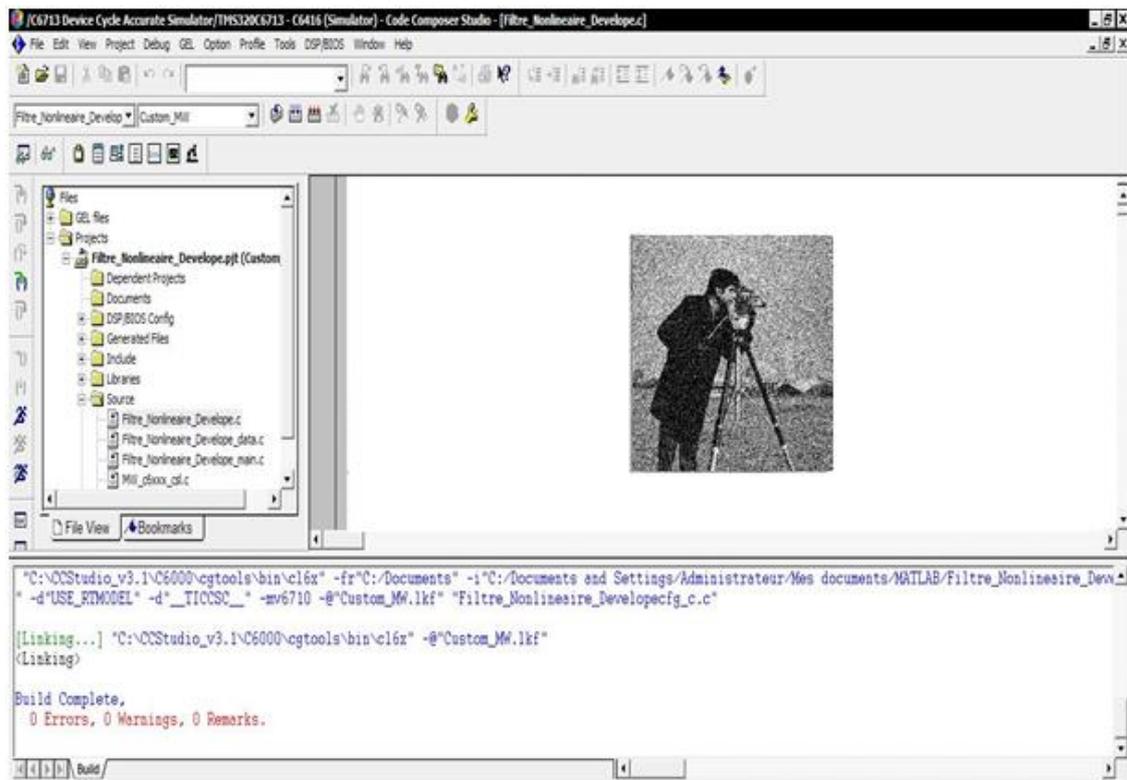


Figure IV.9 : « Capture d'écran de l'image bruitée à l'aide du CCS. »

#### IV.4.5.3 l'image résultante après l'utilisation du filtre proposé CCS

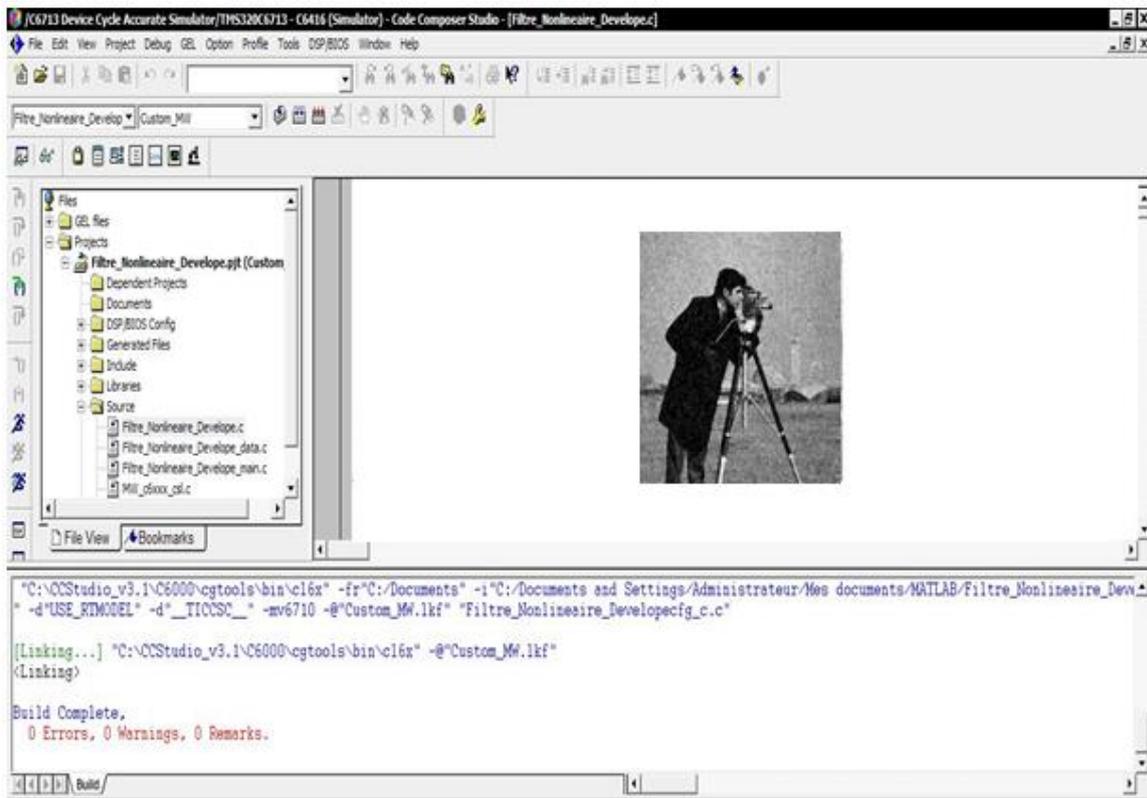


Figure IV.10 : « Capture d'écran CCS de l'image résultante (après l'utilisation du filtre proposé). »

D'après les résultats de cette méthode développée, on peut dire que cette méthode donne une bonne réduction du bruit des images. Cette méthode donne des meilleurs résultats (PSNR, SNR, MSE) avec une meilleure préservation des détails et des bords.

#### IV.4.6 conclusion

Ce chapitre explique également la mise en œuvre logicielle et matérielle à l'aide de la carte DSK TMS320C6713. L'analyse statistique est effectuée à l'aide des statistiques de profil disponibles dans l'environnement CCS (Code Composer Studio). Les résultats de la simulation MATLAB montrent que les valeurs de PSNR sont assez améliorées.

La plate-forme utilisée, malgré des ressources en mémoire limitées, reste adéquate pour ce type de traitement et offre de bonnes performances par rapport à un ordinateur ordinaire. Nous avons conclu que le TMS320C6713 est un meilleur choix pour le traitement numérique d'image. Pour pouvoir résoudre le problème de l'accès à la mémoire tout en améliorant les avantages du traitement DSP, nos travaux futurs seront axés sur l'étude de plates-formes mixtes intégrant plusieurs types de processeurs. Nous pouvons donc implémenter différemment les parties des algorithmes (une partie sur FPGA et une autre partie sur DSP).

# Conclusion Générale

---

## Conclusion Générale

Notre travail consistait à effectuer plusieurs expériences afin d'évaluer les performances de quelques techniques, à savoir : Médian standard, NAGAO modifiée, EFPGF, MAUTMPF, et une méthode proposée dans le but de proposer le meilleur technique d'après des résultats statistiques et visuels.

Il est difficile de choisir une méthode de filtrage pour un type de bruit. Certains filtres sont adaptés pour un type de bruit et d'autres pour d'autres types de bruits.

De nombreuses expérimentations ont été effectuées. Les résultats des différents filtres ont été comparés. Des critères quantitatifs ont été utilisés pour mesurer les performances tel que : PSNR, SNR et MSE et la qualité visuelle.

Pour cela il existe des nombreuses applications pour lesquelles le DSP devient un choix idéal car ils fournissent la meilleure combinaison possible de performances, la puissance et le coût. Les questions de pouvoir prennent de l'importance que les processeurs DSP sont incorporés dans les appareils portatifs, mobiles et portables. Cela conduit au développement d'une classe importante de DSP, Avec les nouvelles technologies de fabrication de circuits intégrés disponibles nous pouvons nous attendre à voir plus sur puce périphériques et la mémoire.

Ous avons pu constater que les DSP ne représentent qu'une partie du traitement numérique du signal, mais il reste essentiel par son rôle important dans la chaîne de traitement. Nous avons choisi ce DSP car il est adapté à des applications de traitement d'images. Puisque notre application concerne Le filtrage Non-Linéaire d'image numérique, nous avons trouvés que le choix du DSP TMS320C6713 est très performant, au niveau de calcule et mémoire, est un choix judicieux.

Nous souhaitons que ce travail serve de base pour les prochains projets et qu'ils aient la chance d'étudier le TMS 320C6713, et nous invitons les prochains étudiants à essayer de poursuivre dans ce vaste sujet

# Bibliographie

## I) Ouvrage :

- [1] G.Wyszecki and W.S. Stiles. Color Science: Concepts and Methods, Quantitative Data and Formulae by Wyszecki, Günther, Stiles, W. S. (2000) Paperback.
- [2] S. Bres, J. Jolion, F. Lebourgeois. Traitement et analyse des images numérique. Paris, Lavoisier, (Oct 2003, 411 pages)
- [6] N MERABET, M MAHLIA, recherche d'images par le contenu, université abou bakrbelkaid- tlemcen.2011
- [7] A. BOUKHECHEBA et F. K BOUKHECHEBA, « Contribution à la segmentation en régions par une approche neuronale, dans le cadre de la stéréovision par région », mémoire de fin d'étude d'ingénieur, Editions 1996.
- [8] M .T.Chikh , amélioration des images par un modèle de réseau de neurones cccc (comparaison avec les filtres de base), université abou-bakr belkaid tlemcen.2011
- [9] K. Aounallah, les approches de segmentation d'image par coopération régions- fffffffcontours, 2010.
- [10] C.Houassine, segmentation d'images par une approche biomimétique hybride. cccc université universite m'hamed bougara- boumerdes. 2012.
- [11] M. Ghazel « Adaptive Fractal and Wavelet Image Denoising », Thèse de doctorat Electrical and Computer Engineering, University of Waterloo, 2004.
- [12] Y. M. Hawwar « Nonlinear image filtering in the wavelet transform domain », Thèse de doctorat, The university of Wisconsin-Milwaukee, 2001.
- [13] S. Bres, J. M. Jolion et F. le bourgeois «Traitement et analyse des images numériques» , LAVOISIER, 2003.
- [14] S.ESAKKIRAJAN, T.VEERAKUMAR et S.JAYARAMAN, «Digital Image Processing», Tata McGraw Hill Publication, New Delhi, 2009.
- [15] V.Murugan, T. Avudaiappan and R. Balasubramanian, "Implementation of MRI Brain Image Enhancement Techniques Using Parallel Processing On Clustering Environment", Australian Journal of Basic and Applied Sciences, 8(13), Pages: 390-402, August 2014.

[16] Esakkirajan, S., T. Veerakumar, A.N. Subramanyam and C.H.P. Chand, 2011. Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. IEEE Signal Proc. Lett., 5: 287-290. DOI: 10.1109/LSP.2011.2122333

[17] A BAROUDI et N.LARABI, « Comparaison entre les différents filtres d'images », Université Abou Bakr Belkaid– Tlemcen 2011-2012.

[21] texas instruments inc TMS320C6000 CPU and Instruction Set Reference Guide Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2010, Texas Instruments Incorporated

[22] Rulph Chassaing Digital Signal Processing and Applications with the C6713 and C6416 DSK Worcester Polytechnic Institute

[23] Bernd Jähne :Digital Image Processing,(2 em edition) 2002

## II) Web graphie :



[3] Le site : <http://www.crdp.ac-grenoble.fr/image/general/general.htm>, visitez le 10 avril 2013.



[4] Le site : [http://www.ac-nice.fr/ia06/iencagnes/file/tice/tuto/Image\\_numerique.pdf](http://www.ac-nice.fr/ia06/iencagnes/file/tice/tuto/Image_numerique.pdf), visitez le 15 avril 2013.



[5] Le site : [http://foad.refer.org/IMG/pdf/D226\\_Chapitre-5.pdf](http://foad.refer.org/IMG/pdf/D226_Chapitre-5.pdf), visitez le 21 avril 2013.

## III) Documents PDF :



[18] Master 2 : K. Aziz, ‘*Débruitage des images fixes par les techniques de L'intelligence artificielle*’, 2011-2012, BATNA.



[19] Master 2 : B. Aammar et B. Abdellatif. ‘Hybridation de deux algorithmes pour une détection de la tumeur cérébrale sur TMS320C6416 DSK’, 2016-2017, Ain T'émouchent.



[20] Master 2 : T. Oussama ‘Implémentation sur un DSP d'un filtre numérique passe-bas en utilisant la carte « TMS320C6713 platform ».filtrage des signaux ECG’, 2018, ANNAB

