



Université ABBES LAGHROUR Khenchela
Faculté des Sciences et de la Technologie
Département de Génie Industriel
جامعة عباس لغفور خنشلة
كلية العلوم والتكنولوجيا
قسم الهندسة الصناعية



N° Série :.....

Mémoire de fin d'étude

Présenté pour l'obtention du diplôme de Master

Filière : Télécommunications

Spécialité : Systèmes des Télécommunications

THEME

Élaboration d'un Logiciel de Traitement d'Images

Réalisé par :

- *Merad Tahar*
- *Merdassi Safouane*

Devant Le Jury :

Président:

Dr. Douak Fouzi

Rapporteur:

Dr. Lanani Abderrahim

Examineur:

Dr. Friha Souad

Promotion 2020/2021

اهداء

.....

- الى روح امي الميته الحية في قلبي دائما.....
- الى ام اولادي التي قاسمتني الحلو و المر.....
- الى فلذاتي كبدي املي ان يكون افضل مني.....

..... مراد الطاهر.

Remerciements



Après un sincère remerciement à notre dieu le tout puissant qui nous a aidé pour accomplir notre parcours universitaire après une longue rupture, plus de 20 ans.

Nous tenons à exprimer toute notre gratitude et notre entière reconnaissance à notre encadreur Monsieur LANANI Abderrahim, Maitre de conférences à l'université de Khenchela, pour tous les conseils précieux et l'aide qu'il nous a apporté durant la préparation de ce mémoire.

Aussi nous remerciant profondément nos enseignants qui nous ont conférés de leurs savoirs pendant toutes ces trois années sans oublier Monsieur Le vice-recteur de l'université ABBES LAGHROUR Khenchela.

Nos collègues respectivement Madame: CHERRABEN Nadia & Messieurs : ZAKI Siam, DJAMEL Abed MEHLOU Noredine qu'ils nous ont encouragé pour terminer ce parcours.

Pour tous nos collègues d'Algérie télécom Khenchela ainsi que les étudiants de notre classe Master Systèmes des télécommunications pour leurs sympathies malgré la déférence d'âge.

Mes derniers remerciements les plus distingués et sincères à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail.

- MERDADI SAFOUANE-

- MERAD TAHAR-

Table des matières

Introduction Générale	1
Chapitre I : Généralités sur le traitement d'images	3
1 Introduction	3
2 Images numériques	4
2.1 Définition.....	4
2.2 Types d'images.....	5
2.2.1 Images vectorielle	5
2.2.2 Images matricielles	5
2.3 Formats d'images numériques	5
2.4 Codage des images	6
3 Caractéristiques des images numériques	6
3.1 Résolution et définition	6
3.1.1 La résolution.....	6
3.1.2 La définition d'une image	6
3.2 Contraste	6
3.3 Luminance	7
3.4 Histogramme	7
3.4.1 Histogramme normalisé	7
3.4.2 Histogramme cumulé	7
4 Chaîne de traitement d'images	8
4.1 Dispositifs d'acquisition des images	8
4.1.1 Capteurs CCD	8
4.1.2 Capteurs CMOS	10
4.2 Prétraitement des images numériques	11
4.2.1 Correction de la dynamique et transformations d'histogramme.....	11
4.2.1.1 Décalage d'histogramme	11
4.2.1.2 Inversion de la dynamique	12
4.2.1.3 Transformation logarithmique	12

4.2.1.4 Egalisation d'histogramme.....	12
4.1.2 Filtrage des images.....	14
4.3 Traitement numérique d'images.....	14
4.3.1 Segmentation et classification.....	14
4.3.2 Compression des images.....	14
4.3.3 Détection de contours.....	15
4.4 Post-traitement des résultats.....	15
4.5 Analyse des résultats.....	16
5 Conclusion.....	16

Chapitre II : Quelques méthodes en restauration d'images17

1 Introduction.....	17
2 Les dégradations d'images.....	19
2.1 Bruits.....	19
2.1.1 Bruit impulsionnel ou binaire (Salt & Pepper noise).....	19
2.1.2 Bruit Gaussien.....	20
2.1.3 Bruit multiplicatif.....	20
2.1.4 Bruit périodique.....	21
2.2 Image floue.....	21
2.3 Problème d'aliasing (Repliement de spectre).....	21
2.4 Phénomène de Gibbs.....	22
3 Paramètres évaluatifs de la qualité d'image.....	22
4 Quelques méthodes en restauration d'image.....	23
4.1 Filtrage à base de la convolution numérique.....	23
4.1.1 Convolution numérique bidimensionnelle.....	23
4.1.2 Filtre moyenneur (flou uniforme).....	24
4.1.3 Filtre gaussien.....	25
4.2 Filtre médian.....	25
4.3 Filtre SNN (Symmetric Nearest Neighbor).....	26
4.4 Les filtres adaptatifs.....	27
4.4.1 filtre de Lee.....	27
4.4.2 Filtre de Nagao.....	29
4.4.3 Filtre de Kuan.....	30
4.4.4 Filtre de Kuwahara.....	30

4.5	<i>Filtre de voisinage non linéaire</i>	31
4.4.1	<i>Filtre Bilatéral</i>	31
4.4.2	<i>Moyennes non-locales (NL-means algorithm)</i>	32
4.6	<i>Filtre de Wiener</i>	33
4.7	<i>Les Ondelettes (Débruitage par seuillage de coefficients d'ondelettes)</i>	34
5	Conclusion	35
Chapitre III : Description du logiciel développé		36
1	Introduction	36
2	Outil de programmation utilisés « MATLAB »	37
3	Description du Logiciels réalisé	37
3.1	<i>Menu principal</i>	37
3.2	<i>Fenêtre « Filtre moyennneur »</i>	38
3.3	<i>Fenêtre « Filtre Guaussien»</i>	39
3.4	<i>Fenêtre « Filtre Médian»</i>	40
3.5	<i>Fenêtre « Filtre SNN»</i>	41
3.6	<i>Fenêtre « Filtre de Lee»</i>	41
3.7	<i>Fenêtre « Filtre Nagao»</i>	42
3.8	<i>Fenêtre « Filtre de Kuan»</i>	43
3.9	<i>Fenêtre « Filtre Kuwahara»</i>	44
3.10	<i>Fenêtre « Filtre Bilatéral»</i>	45
3.11	<i>Fenêtre « Non Local Means»</i>	46
3.12	<i>Fenêtre « Filtre de Wiener»</i>	47
3.13	<i>Fenêtre « Débruitage par seuillage de coefficients d'ondelettes»</i>	48
4	Conclusion	49
Conclusion et perspectives		50
Bibliographies		51

Introduction générale

Le traitement d'images est une phase préliminaire dans le but de préparer des images numériques à une analyse ultérieure de plus haut niveau comme l'interprétation, la visualisation, le stockage ou la communication.

Tous les secteurs de la recherche scientifique (physique, biologie, médecine, astronomie, automatique...) jouent un rôle de plus en plus important de développer cette discipline. En effet, l'utilisation des différentes techniques du traitement d'images, dans les différents domaines permet d'améliorer la détection précoce et le traitement judicieux de nombreux problèmes.

Cependant, la plupart des techniques d'imagerie subissent un problème relatif à un sujet classique : la restauration d'images.

L'opération de restauration consiste à corriger des images dégradées et à reconstruire un signal de bonne qualité à partir d'un signal direct ou indirect et de médiocre qualité, suite au processus d'acquisition.

Au cours de ces dernières années, le problème de la restauration d'image s'est posé en ces termes : comment débruiter l'image tout en respectant les caractéristiques géométriques dans l'image originale ? La difficulté est due au fait que les parties bruitées et les parties discontinues ont généralement des caractéristiques communes.

Dans la littérature du traitement d'images, plusieurs méthodes de débruitage ont été proposées et développées. Tout d'abord, très intuitifs, mais progressivement de plus en plus complexes. Commencent par les méthodes de filtrage spatial [1], celles-ci consistent à réduire le bruit dans les zones qui ne présentent pas d'objets intéressants et à accentuer la perception des structures d'intérêt. Ces techniques de filtrage utilisent un filtre passe-bas pour supprimer les hautes fréquences, ce qui a pour inconvénient d'atténuer les contours de l'image. Pour pallier à ces problèmes, de nouvelles techniques, plus performantes, ont vu le jour aux cours des années 80 et 90 ; citons notamment les approches variation-elles basées sur les EDPs [2], les approches utilisant les champs de Markov [3], et les approches basées sur les transformées multi-échelles, notamment la transformée en ondelettes [4].

En revanche, le filtrage des images nécessite un calcul judicieux et parfois compliqué pour émettre ou extraire une information avec une grande fidélité. Refaire à chaque fois les calculs est un travail non rentable sur le plan temps et effort. C'est pourquoi les logiciels forment un outil de calcul très utile et bénéfique.

Notre principal objectif consistera à développer un logiciel de traitement d'images dans le but de récupérer des images de bonne qualité à partir des images bruitées de mauvaise qualité, on se basant sur les méthodes et les algorithmes de restauration les plus connues et les plus performants.

Ce mémoire s'articule autour de trois chapitres :

Dans le premier chapitre, de ce mémoire, nous présenterons un aperçu théorique sur le traitement d'images, commencerons par l'exposition de quelques notions de bases sur les images numériques (définition, type, caractéristiques,..). Ensuite nous aborderons les différentes étapes nécessaires dans une chaîne de traitement d'images (à partir du capteur jusqu'à l'analyse des résultats).

Dans le deuxième chapitre nous présentons les différentes méthodes pour la restauration des images, à travers les filtres linéaire et non-linéaires, les méthodes à base des filtres adaptatifs, les méthodes issues de l'analyse harmonique et bien évidemment, la transformée en ondelettes.

Dans le troisième chapitre, après l'exposition de l'outil de programmation exploité, nous décrirons le logiciel que nous avons développé.

Enfin, ce travail sera conclu par une conclusion générale résumant les idées fondamentales que nous avons apportées dans ce mémoire, et permet de dégager quelques perspectives.

Chapitre 1

Généralités sur le traitement d'images

1. Introduction

Le traitement d'images est un domaine très vaste qui a connu, et qui connaît encore, un développement important depuis quelques dizaines d'années.

On désigne par traitement d'images numériques l'ensemble des techniques permettant de modifier une image numérique afin d'améliorer ou d'en extraire des informations.

Dans ce chapitre nous présenterons un aperçu théorique sur le traitement d'images, commencerons par l'exposition de quelques notions de bases sur les images numériques (définition, type, caractéristiques,..). Ensuite nous aborderons les différentes étapes nécessaires dans une chaîne de traitement d'images (à partir du capteur jusqu'à l'analyse des résultats).

2. Images numériques

2.1 Définition

Une image numérique est représentée par un tableau ou une matrice I de N lignes et M colonnes. Chaque élément de la matrice est un pixel. Ce dernier (Picture element) représente l'élément de base qui constitue l'image, il indique également l'intensité lumineuse (noté dans la figure 1.1 ci-dessous N) [5].

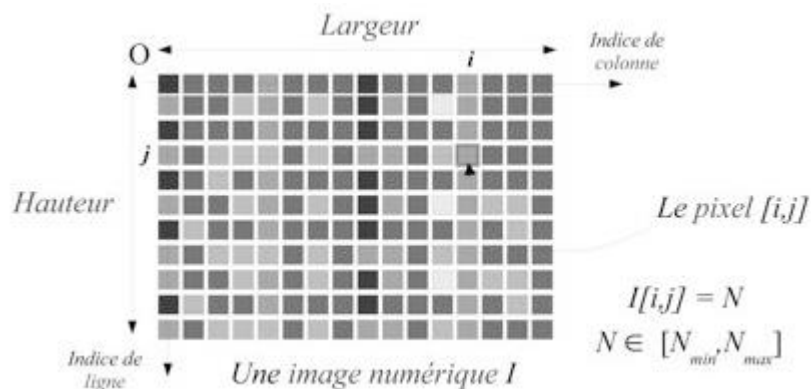


Fig. 1.1 Image numérique

Le pixel est désigné par un couple de coordonnées (i, j) où i est l'indice de ligne et j l'indice de colonne. Le nombre $I(i, j)$ est la valeur du pixel (i, j) qui représente l'intensité lumineuse ou la couleur.

N_{min}, N_{ma} : Valeur minimale est maximale de l'intensité lumineuse N .

$[N_{min}, N_{max}]$: Dynamique de l'image.

D'autre part, on distingue trois catégories d'images :

- **Image binaire**

Chaque pixel est considéré comme Noir ou Blanc, deux valeurs possibles sont alors attribuées à chaque pixel : 0 (Noir) ou 1 (Blanc).

- **Image en niveaux de gris**

En général, les images en niveaux de gris renferment 256 teintes de gris de 0 (Noir) à 255 (Blanc). Chaque pixel peut être représenté par 8 bits.

- **Image couleur**

Les images couleurs sont composées d'un empilement de 3 matrices représentant les valeurs du rouge, vert et bleu, ce qui signifie qu'à chaque pixel correspond 3 valeurs comprises entre 0 et 255, ceci donne en totalité : $256^3 = 16777216$ couleurs.

2.2 Types d'images

Pour représenter une image, on peut la décrire à l'aide de fonctions mathématiques (représentation vectorielle) ou par l'ensemble des points qui la composent (représentation matricielle).

2.2.1 Images vectorielle

Une image vectorielle est définie par un ensemble de données mathématiques : coordonnées, fonctions, attributs, etc. Une image vectorielle peut être agrandie ou rétrécie sans dégradation car l'image sera recalculée précisément en fonction de la taille souhaitée. Il existe de nombreux formats de fichiers vectoriels. On peut citer Postscript, PDF, Adobe Flash, SVG, DXF et DWG.

2.2.2 Images matricielles

Une image matricielle est définie par une grille de points ou pixels auxquels est associée une couleur. Une image matricielle se dégrade si on l'agrandit : la pixellisation devient visible.

2.3 Formats d'images numériques

Un format d'image est une représentation informatique de l'image, associée à des informations sur la façon dont l'image est codée et fournissant éventuellement des indications sur la manière de la décoder et de la manipuler. Il existe une multitude de formats d'images :

Types d'images	Nom du format	Extension
Images Vectorielles	Illustrator	.ai
	Flash	.swf
	Encapsulated postscript	.eps
	SVG	.svg
	Autocad	.dwg
Images Matricielles	JPEG	.jpg
	GIF	.gif
	PNG	.png
	TIFF	.tif
	Bitmap	.bmp

Tab.1.1 Récapitulatif sur les formats d'images [5]

2.4 Codage des images

Pour pouvoir stocker et transmettre une image comme n'importe quelle autre donnée informatique, chaque pixel de l'image doit être codé en binaire (suite de 0 et de 1) en utilisant un nombre de bits B fixe. Le nombre de bits B utilisé influe directement sur la qualité de l'image. Si le nombre de bits est B , alors on obtient 2^B couleurs possibles dans l'image, par exemple : si $B = 8$ on obtient 256 couleurs.

3 Caractéristiques des images numériques

3.1 Résolution et définition

3.1.1 La résolution

Représente le nombre de pixels par unité de longueur, c'est en fait la densité de pixels à travers l'image. La résolution s'exprime en dpi (dot per inch = point par pouce) pour une imprimante, ou en ppp (pixel par pouce) pour les moniteurs

3.1.1 La définition d'une image

Représente le nombre total de pixels dans l'image. Elle est définie par le nombre de pixels sur la hauteur de l'image multiplié par le nombre de pixels sur la largeur de l'image, par exemple : 2000 pixels par 3000 pixels abrégés en « 2000 × 3000 = 6000000 ou 6 Mpx ».

3.2 Contraste

Le contraste est une propriété intrinsèque d'une image qui quantifie la différence de luminosité entre les parties claires et sombres d'une image. Le contraste C est défini par le rapport :

$$C = \frac{L_{\max} - L_{\min}}{L_{\max} + L_{\min}} \quad (1.1)$$

3.3 Luminance

La luminosité de l'image ou d'une zone d'image = moyenne des niveaux de gris, c'est-à-dire :

$$L = \frac{1}{N} \sum_{i=0}^x h(i) \cdot i \quad (1.2)$$

3.4 Histogramme

L'histogramme $h(x)$ d'une image représente la distribution des intensités des pixels. Un histogramme est une fonction qui donne, pour chaque intensité lumineuse (niveau de gris), le nombre de pixels ayant cette valeur.

L'abscisse d'un histogramme $h(x)$ représente les niveaux d'intensité allant du plus foncé à gauche au plus clair à droite.

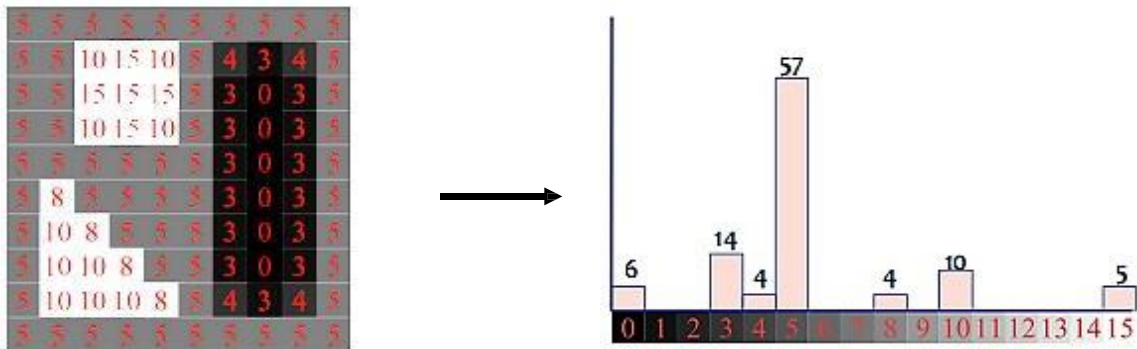


Fig. 1.2 Histogramme d'une image

3.4.1 Histogramme normalisé

La probabilité d'apparition d'un niveau de gris dans l'image est :

$$h_n(x) = h(x)/N, \quad h_n(x) \in [0,1] \quad (1.3)$$

La fonction correspondante $h_n(x)$ est l'histogramme normalisé, avec N représente le nombre de pixels dans l'image.

3.4.2 Histogramme cumulé

L'histogramme cumulé dénombre les occurrences cumulées de chacun des niveaux :

$$h_c(x) = \sum_{i=0}^x h(i) \quad (1.4)$$

4 Chaîne de traitement d'images

L'acquisition d'images est une mesure spatiale d'une interaction entre une onde et de la matière. L'onde est émise par une source et reçue par un capteur. Par exemple dans le cas de l'échographie, l'ultrason, une onde acoustique, est émis et reçue par la sonde. L'interaction est la réflexion de l'ultrason sur la structure du corps. Un système de traitement d'images est composé de :

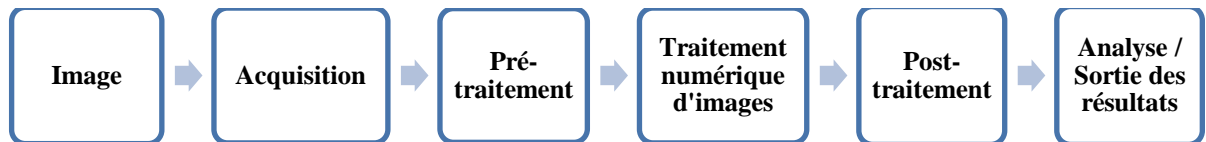


Fig.1.3 Schéma d'un système de traitement d'images [5].

4.1 Dispositifs d'acquisition des images

Pour l'acquisition des images numériques, le dispositif d'acquisition doit posséder un éclairage, un élément optique, unité de traitement ou processeur et une mémoire de sauvegarde ou écran d'affichage.

Le dispositif d'acquisition capture l'information lumineuse et la convertit en signal électrique analogique grâce aux photo-détecteurs. Parmi les capteurs les plus connus, on trouve les capteurs CCD (Charge-Coupled Device) et CMOS (Complementary Metal-Oxide- Semiconductor).

4.1.1 Capteurs CCD

Les capteurs appelés CCD (Charged Coupled Device ou en français Dispositif de Transfert de Charges), est un composant électronique servant à convertir la lumière visible en un signal analogique. Ces capteurs fonctionnent grâce à l'effet photoélectrique qui consiste à convertir un photon (constituant de la lumière) en électrons (signal analogique). Le lieu où se fait la conversion du photon en électron est un photo-site.

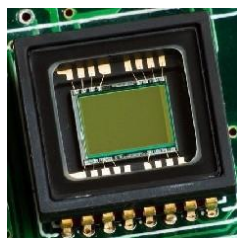


Fig. 1.4 Capteur CCD[5]

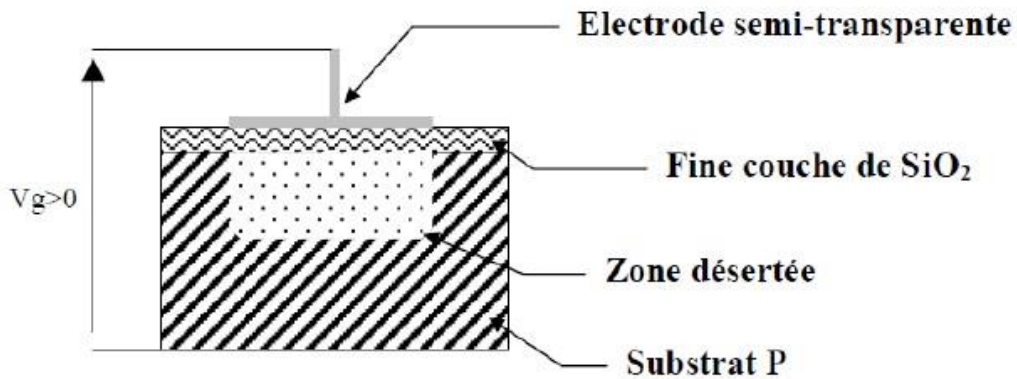


Fig. 1.5 Photo-site d'un capteur CCD

Un CCD transforme les photons lumineux qu'il reçoit sur l'électrode semi-transparente en électrons par effet photoélectrique, puis collecte les électrons dans la zone désertée. Le nombre d'électrons collectés est proportionnel à la quantité de lumière reçue.

À la fin de l'exposition, les charges sont transférées de photosite en photosite par le jeu de variations de potentiel cycliques provenant d'une horloge externe. Puis elles sont envoyées dans un registre de sortie [5].

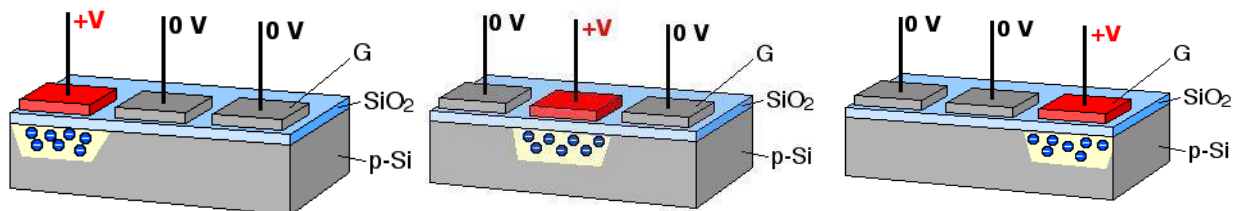


Fig. 1.6 Transfert de charge

Enfin les charges sont transformées en tension proportionnelle au nombre d'électrons. Ce signal sera, à l'extérieur du CCD, amplifié et numérisé.

Actuellement il existe 3 types de ces capteurs :

- *Le CCD plein cadre (full frame)*

En technologie plein cadre la construction est assez simple puisque les lignes CCD verticales jouent à la fois le rôle de capteur de lumière et de moyen de transmission des charges.

- *Le CCD a transfert de trame*

IL utilise deux matrices de photo-sites. Une de ces deux matrices est exposée à la lumière, elle sert à la détection des photons. L'autre est masquée, elle est utilisée pour assurer le transfert des charges.

- *Le CCD interligne*

Il associe un photo-site à chaque cellule CCD. Il permet de mieux reproduire la lumière visible sur tout son spectre.

Le fonctionnement général d'une caméra à capteur CCD (CCD de tout type) est représenté dans la figure suivante :

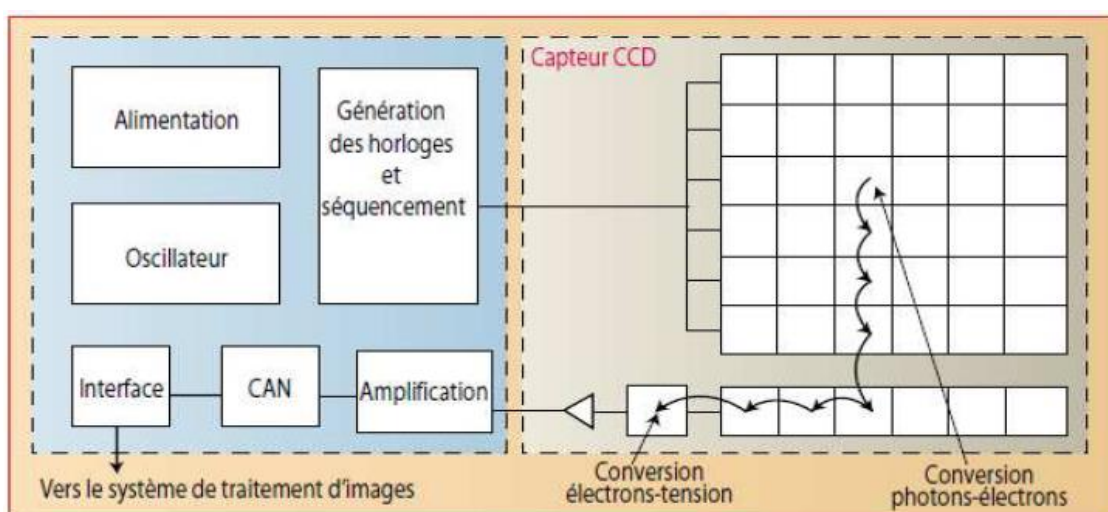


Fig. 1.7 Camera à capteur CCD

4.1.1 Capteurs CMOS

Les capteurs CMOS (Complementary Metal Oxide Semiconductor ou en français Semi-conducteur à Oxyde de Métal Complémentaire) fonctionnent différemment, même si le principe de base reste le même. Les photo-sites, comme sur un CCD sont sensibles à la lumière et acquièrent une certaine charge électrique en fonction de la quantité de lumière reçue. Mais ensuite, à l'intérieur de chaque photo-site, la charge générée est convertie directement en tension utilisable [5].

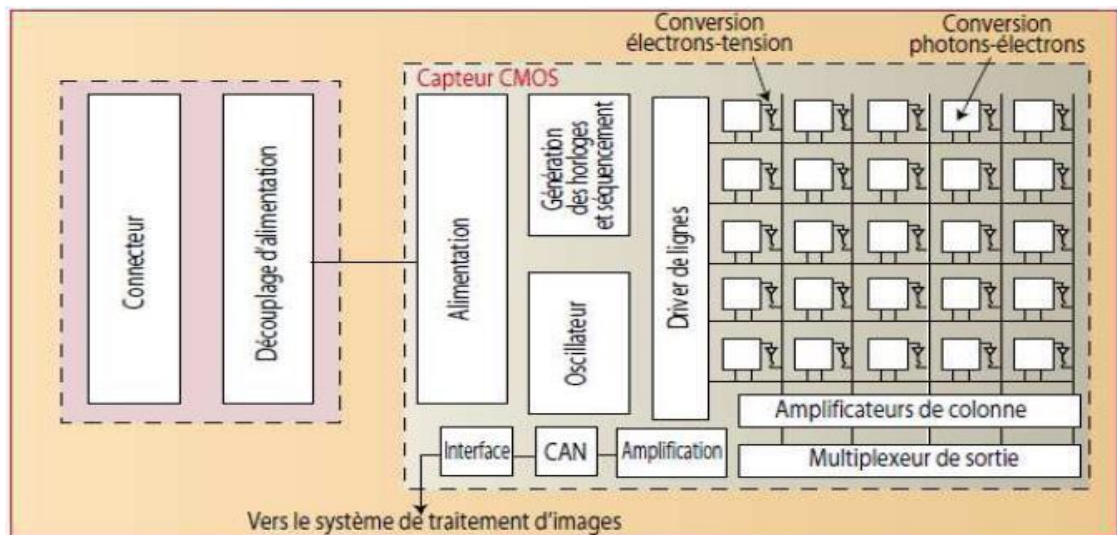


Fig. 1.8 Camera à capteur CMOS

4.2 Prétraitement des images numériques

4.2.1 Correction de la dynamique et transformations d'histogramme

4.2.1.1 Décalage d'histogramme

Pour augmenter la luminosité de l'image, il suffit de décaler l'histogramme vers la droite.

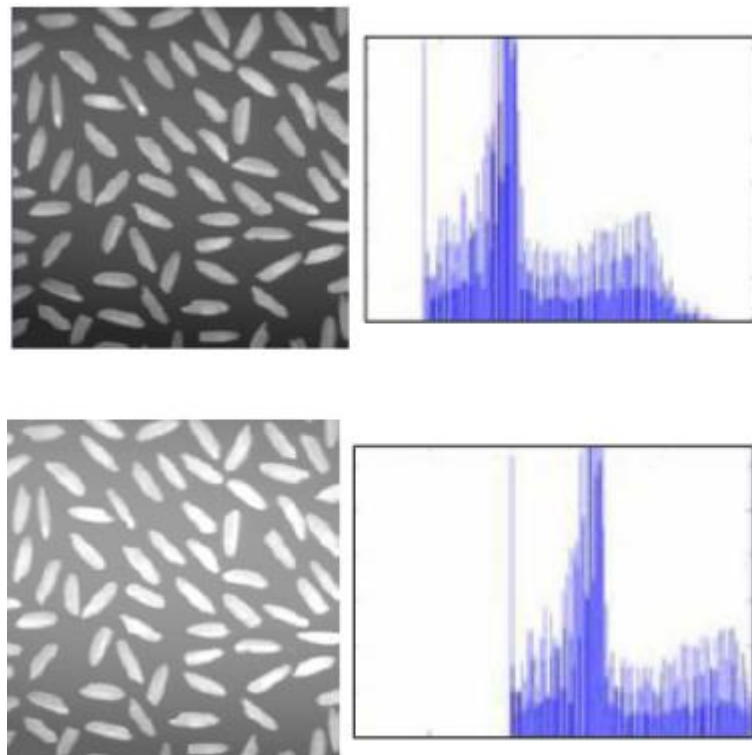


Fig. 1.9 Décalage de l'histogramme d'une image.

4.2.1.2 Inversion de la dynamique

On inverse les extrêmes noir et blanc. Parfois, on distingue mieux certains détails en blanc sur fond noir qu'en noir sur fond blanc :

$$g(x, y) = g_{max} - f(x, y) = 255 - f(x, y) \quad (1.5)$$

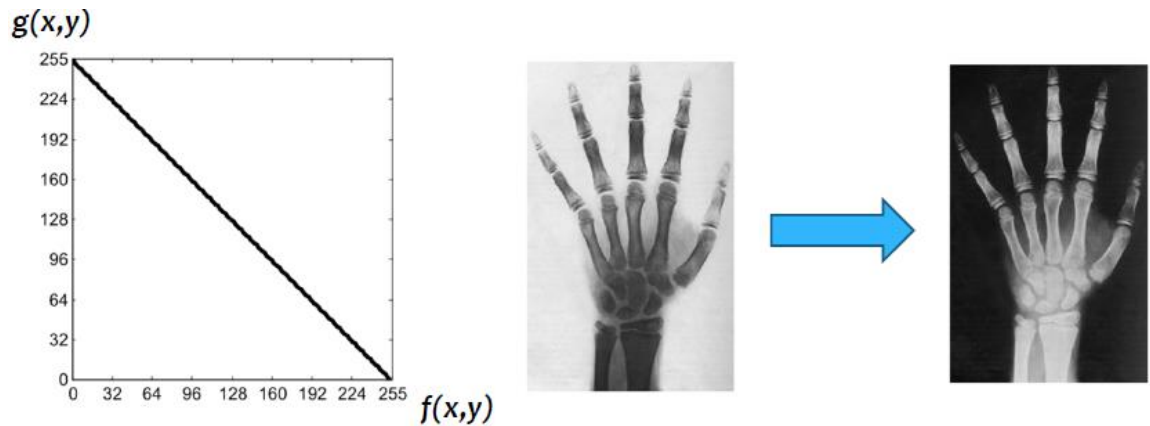


Fig. 1.10 Inversion de la dynamique

4.2.1.3 Transformation logarithmique

Cette transformation de l'intensité permet de dilater les intensités faibles et de compresser les niveaux d'intensités élevées.

$$g(x, y) = \frac{g_{max}}{\ln(1 + g_{max})} \cdot \ln(1 + f(x, y)) \quad (1.6)$$



Fig. 1.11 Transformation logarithmique

4.2.1.4 Egalisation d'histogramme

La transformation d'égalisation est construite de telle façon que l'histogramme de l'image transformée soit le plus plat possible. Cette technique améliore le contraste et permet d'augmenter artificiellement la clarté d'une image grâce à une meilleure répartition des intensités

relatives, où chaque niveau de gris dans l'image se produit par la même fréquence.

Supposons une image ayant L niveaux de gris différents $0, 1, 2, \dots, L - 1$ et que le niveau de gris i se produit n_i fois dans l'image. Supposons également que le nombre total de pixels dans l'image est n (c.à.d. $n_0 + n_1 + n_2 + \dots + n_{L-1} = n$).

Pour transformer les niveaux de gris afin d'obtenir le meilleur contraste, on remplace le niveau de gris i par :

$$\left(\frac{n_0 + n_1 + n_2 + \dots + n_i}{n} \right) \cdot (L - 1) \quad (1.7)$$

La valeur obtenue, après calcul, est arrondie au nombre entier le plus proche [5].

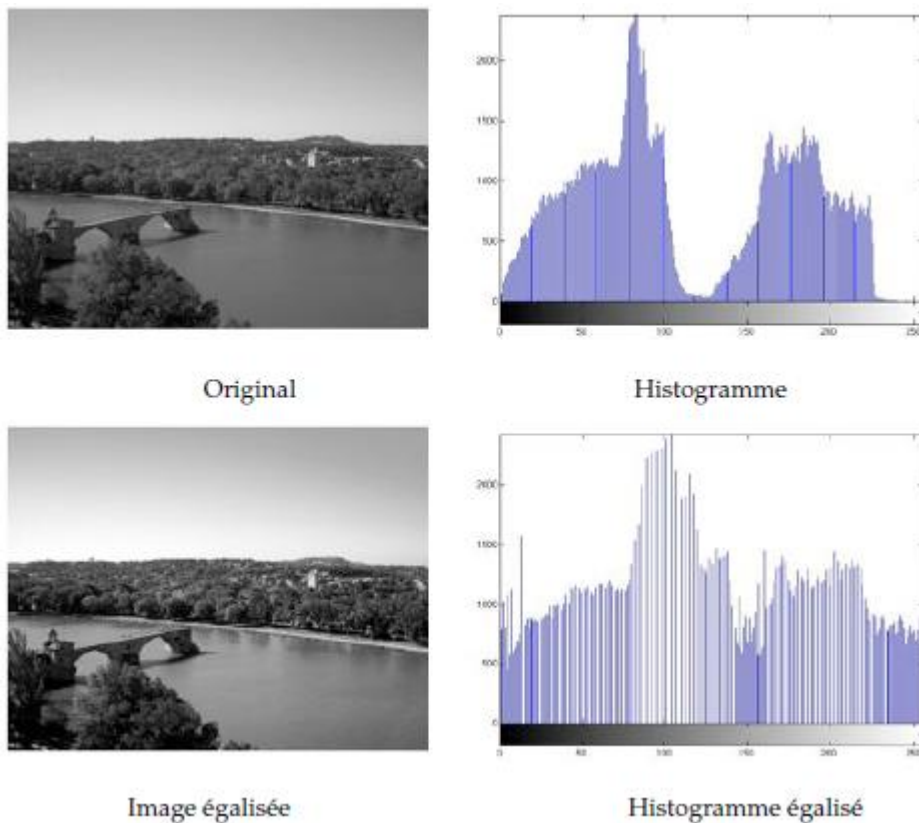


Fig. 1.12 Egalisation d'histogramme [5]

4.2.2 Filtrage des images

Le filtrage d'une image est une opération ayant pour objet de réduire ou d'éliminer ou de rehausser certains éléments présents dans une image. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine.

En revanche, de nombreux filtres peuvent s'opérer tant dans le domaine spatial que le domaine spectral. Ces filtres peuvent s'écrire sous la forme d'un produit de convolution entre une matrice (ou noyau de convolution) et d'une image.

Malheureusement, en appliquant de gros masques de convolution, on peut constater que le temps de calcul devient très important, Pour cela, on peut appliquer les masques dans l'espace des fonctions de Fourier qui rendront les calculs beaucoup plus rapides

4.3 Traitement numérique d'images

4.3.1 Segmentation et classification

La segmentation est une opération de traitement d'images qui a pour objectif de partitionner une image en ensembles de pixels homogènes (selon un critère prédéfini). En effet, la segmentation tente à associer à chaque pixel de l'image un label (i.e. Étiquette) en s'appuyant sur l'information portée sur les niveaux de gris, la couleur ou la distribution spatiale. Etant donné une image, il existe toujours plusieurs segmentations possibles. Une bonne méthode de segmentation sera donc celle qui permettra d'arriver à une bonne interprétation. Elle devra donc simplifier l'image sans pour autant en avoir trop réduit le contenu [5].

4.3.2 Compression des images

La compression d'image est une application de la compression de données sur des images numériques. Cette compression a pour utilité de réduire la redondance des données d'une image afin de pouvoir l'emmagasiner sans occuper beaucoup d'espace ou la transmettre rapidement.

La compression d'image peut être effectuée avec perte de données ou sans perte. La compression sans perte est souvent préférée là où la netteté des traits est

primordiale : schémas, dessins techniques, icônes, bandes dessinées. La compression avec perte, plus radicale, est utile pour les transmissions à bas débit, mais dégrade la qualité de l'image restituée. Les méthodes de compression sans perte sont également préférées là où la précision est vitale : balayages médicaux ou numérisations d'images pour archivage. Les méthodes avec perte restent acceptables pour des photos dans les applications où une perte mineure de fidélité (parfois imperceptible) est tolérée pour réduire les coûts de stockage ou d'envoi.

4.3.3 Détection de contours

La détection de contours est une technique de réduction d'information dans les images, qui consiste à rechercher un ensemble de points délimitant toutes les régions de l'image, d'autre part la détection de contour transforme l'image en un ensemble de courbes, pas forcément fermées, formant les frontières significatives de l'image [5].



Fig. 1.13 Exemple de détection de contour [5]

4.4 Post-traitement des résultats

L'étape de Post-traitement comprend :

- La correction des résultats,
- Éliminer/corriger les erreurs,
- Morphologie mathématique,
-

4.5 Analyse des résultats

- Visualisation du résultat,
- Reconnaissance des formes,
- Indexation et recherche d'images,
- Détection d'obstacles pour un robot,
-

5 Conclusion

Dans ce chapitre, nous avons donné un récapitulatif sur les définitions et notions élémentaires liées au traitement d'images. Il s'agissait des notions de bases sur les images numériques, caractéristiques et formats. De même, nous avons abordé les différentes étapes nécessaires dans une chaîne de traitement d'images (depuis l'acquisition des images jusqu'à l'analyse des résultats).

Chapitre 2

Quelques méthodes en restauration d'images

1. Introduction

Une image numérique est un objet très irrégulier pour de nombreuses raisons. En effet, elle est souvent confrontée à un phénomène de bruit ce qui rend le traitement des données difficile. Le phénomène de bruit peut être soit d'origine externe, perturbation du système d'acquisition de l'image par des facteurs tel que la luminosité, la pollution, soit d'origine interne au signal du fait des fluctuations spontanées des grandeurs électriques mises en œuvre dans l'acquisition de l'image. L'image peut être aussi détériorée suite à une compression ou décompression mal calibrée.

Cependant, le débruitage est l'un des sujets les plus délicats du traitement d'image. Il a vu couler beaucoup d'encre et de nombreuses méthodes lui ont été consacrées, tout d'abord très intuitives, mais progressivement de plus en plus complexes.

Au cours de ces dernières années, le problème de la restauration d'image s'est posé en ces termes : Comment débruiter l'image tout en respectant les caractéristiques géométriques dans l'image originale ? La difficulté est due au fait que les parties bruitées et les parties discontinues ont généralement des caractéristiques communes. Dans ce chapitre nous présenterons quelques méthodes de filtrages (linéaires et non linéaires) confrontés en traitement d'image. Les résultats expérimentaux sont présentés sur une image test (Image du satellite Alsat-2B prise le 16/10/2016 sur le port de Skikda [6].) de taille (500 X 700), (Figure 2.1) avec un bruit blanc gaussien ($\sigma = 10\%$). Le bruit est important mais visuellement, les informations principales de l'image originale sont conservées.

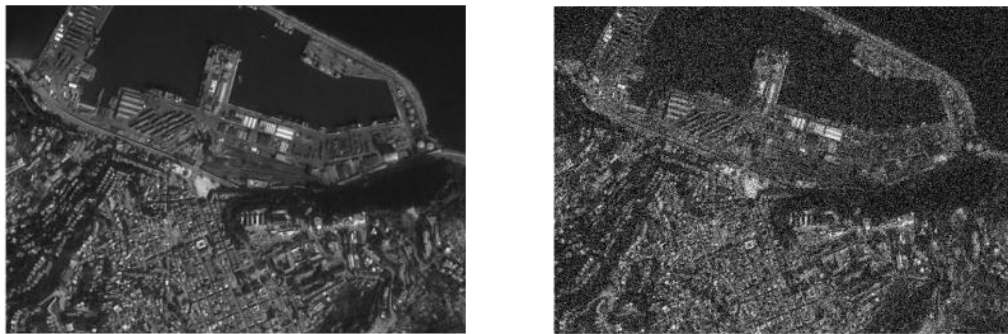


Image originale
500 X 700

Image bruitée $\sigma = 10\%$
PSNR = 20.00

Fig.2.1 *L'image originale et l'image bruitée.*
(Image du satellite Alsat-2B prise le 16/10/2016 sur le port de Skikda)[6]

2. Les dégradations d'images

Les dégradations d'images que l'on rencontre le plus souvent sont provoquées soit par le mouvement de la caméra au cours de la prise de vue, soit par une mauvaise mise au point ou soit par les turbulences atmosphériques.

Ces dégradations peuvent être un bruit (erreur dans les valeurs des pixels), des effets optiques tels que les images flous ou bien problème d'aliasing (repliement de spectre), etc.

2.1 Bruits

On définit le bruit comme étant toute dégradation dans le signal image, causée par des perturbations externes.

Lorsqu'une image est transmise d'un point à un autre (satellites, câbles de transmission, réseau de câbles), des erreurs se produisent dans ce signal.

Ces erreurs apparaissent sous plusieurs formes. Connaissant le type d'erreur, on peut choisir la méthode la plus appropriée pour réduire ces effets [5].

2.1.1 Bruit impulsionnel ou binaire (Salt & Pepper noise)

Il est causé par des perturbations soudaines et aigues (ou violents) affectant le signal image. Il apparaît sous forme de points (pixels) noirs et blancs à travers l'image [7].

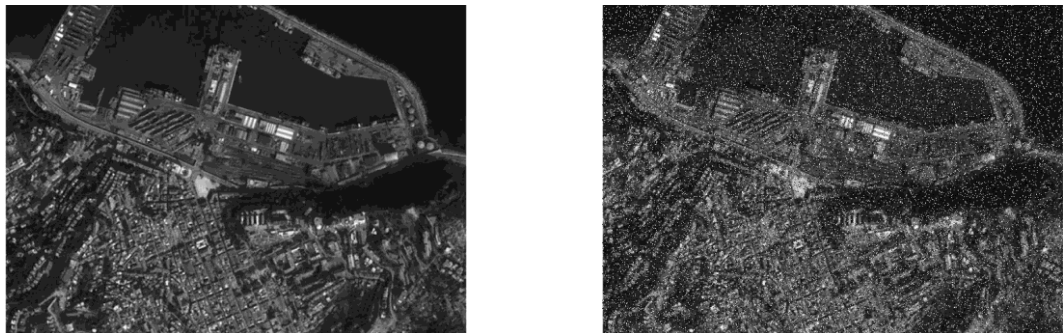


Fig.2.2 Image bruitée avec un bruit impulsionnel ou binaire (Salt & Pepper noise)

2.1.2 Bruit 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 :

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{(x-\mu)^2}{2\sigma^2} \right] \quad (2.1)$$

Où $f(x)$ est la densité de probabilité gaussienne (unidimensionnelle), μ et σ représentent respectivement la moyenne et la variance [7].

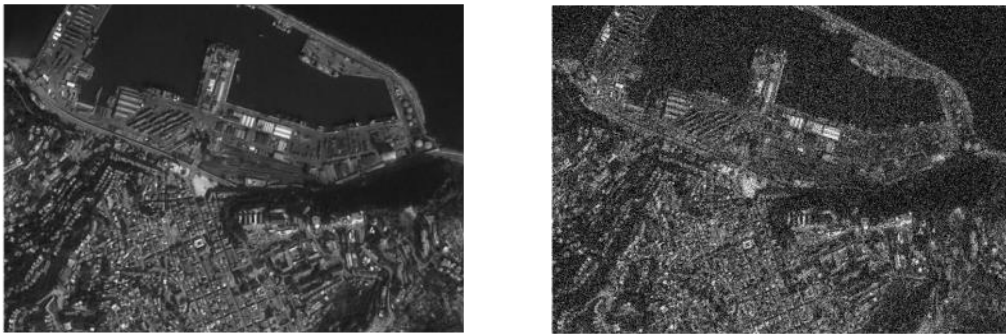


Fig.2.3 Image bruitée avec un bruit Gaussien

2.1.3 Bruit multiplicatif

Le bruit multiplicatif peut être modélisé par des valeurs aléatoires multipliées par les valeurs des pixels [7].

$$\text{Image bruitée} = (\text{Image originale}) \times (\text{bruit}) \quad (2.2)$$

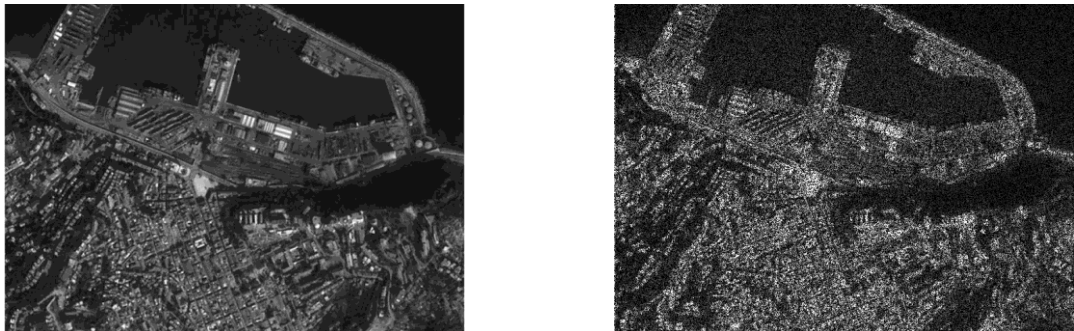


Fig.2.4 Image bruitée avec un bruit multiplicatif

2.1.4 Bruit périodique

Lorsqu'un signal image est exposé à une perturbation périodique (non aléatoire), on obtient une image affectée par un bruit périodique. Ce bruit apparaît sous forme de barres à travers l'image.

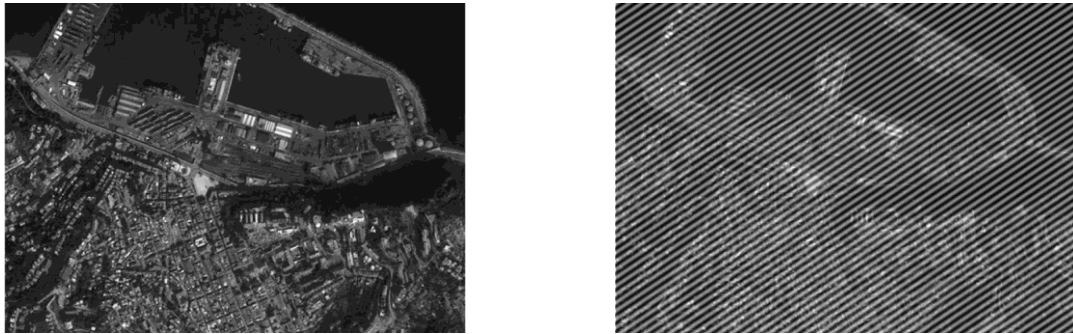


Fig.2.5 Image bruitée avec un bruit périodique

2.2 Image floue

Le flou est une sorte du bruit convolutif. L'image affectée par ce type de bruit est modélisée par le produit de convolution

$$\text{Image bruitée} = (\text{Image originale}) * (\text{bruit}) \quad (2.3)$$

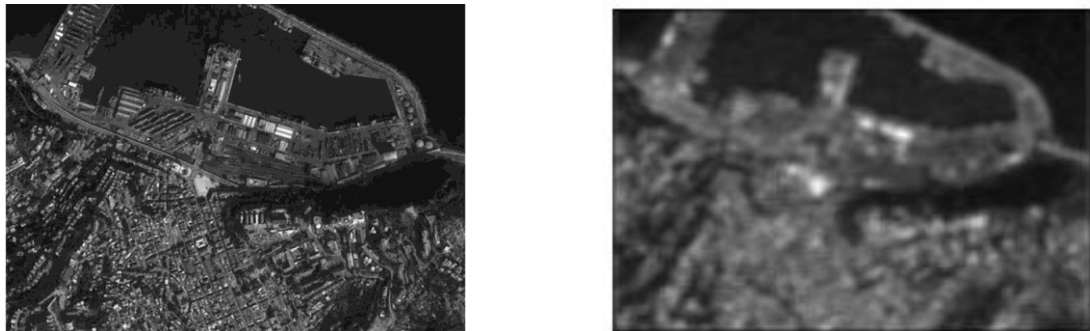


Fig.2.6 Image floue

2.3 Problème d'aliasing (Repliement de spectre)

Crènelage ou **repli de spectre**, c'est le phénomène de recouvrement du spectre (Figure 2.7), il peut se produire lors du passage du monde continu au monde discret, d'un signal ou d'une image, lorsque des fréquences qui ne sont pas représentées dans le signal original sont introduites par erreur dans le signal, par conséquence de son échantillonnage ne respectant pas le théorème d'échantillonnage de Nyquist-Shannon [8].

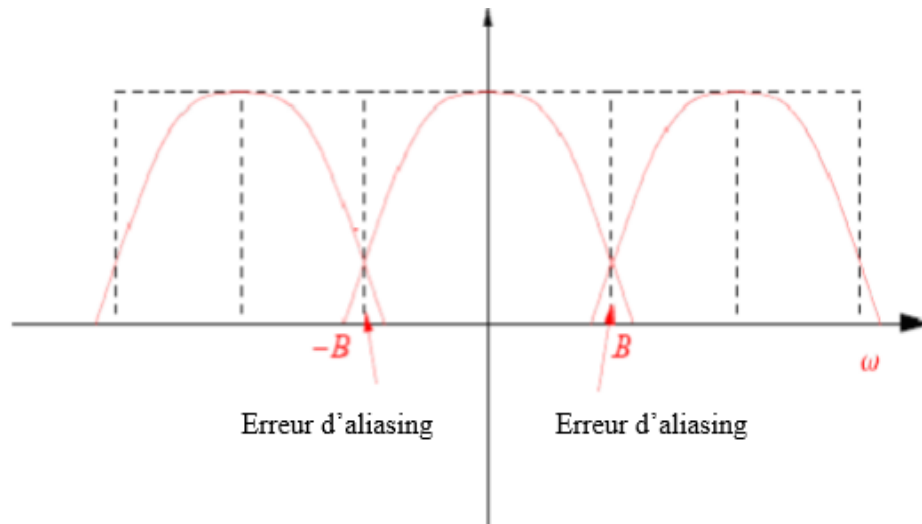


Fig.2.7 Problème d'Aliasing.

2.4 Phénomène de Gibbs

Lors de l'étude des transformées de Fourier, il apparaît parfois une déformation de l'image, connue sous le nom de phénomène de Gibbs (figure). Ce phénomène est un effet de bord qui se produit à proximité d'une discontinuité (contour) [1].

3. Paramètres évaluatifs de la qualité d'image

L'effet du bruit sur l'image est souvent décrit par Le rapport signal sur bruit (SNR, pour Signal to Noise Ratio) qui est souvent utilisé comme un paramètre quantitatif de bruit dans les applications de traitement de signaux et d'images, il est donné par la relation suivante :

$$SNR = \frac{\sigma_w}{\sigma_n} = \sqrt{\frac{\sigma_y^2}{\sigma_n^2} - 1} \quad (2.4)$$

Où σ_n^2 et σ_y^2 sont les variances respectives de l'image réelle et de l'image traitée. Un autre paramètre quantitatif est le PSNR (Peak Signal to Noise Ratio), il est souvent utilisé pour mesurer la qualité de l'image obtenue par rapport à l'image standard numérique.

Pour une image reconstruite I et l'image de référence I_0 de taille $M \times N$, le PSNR en décibels (dB) est défini comme suit :

$$PSNR = 10 \log_{10} \frac{D^2}{EQM} \quad (2.5)$$

$$EQM = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (f_0(i,j) - f(i,j))^2 \quad (2.6)$$

Où D est l'écart entre la plus grande et la plus petite valeur d'intensité de l'image.

Dans le cas standard d'une image à niveaux de gris entre 0 et 255, $D = 255$. EQM est l'erreur quadratique moyenne de cette image.

Une image parfaitement reconstruite a un $PSNR$ infini et une erreur quadratique moyenne (EQM) nulle [7].

4. Quelques méthodes en restauration d'image

Le filtrage d'une image est une opération ayant pour objet de réduire ou d'éliminer ou de rehausser certains éléments présents dans une image. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine.

En revanche, de nombreux filtres peuvent s'opérer tant dans le domaine spatial que le domaine spectral. Ces filtres peuvent s'écrire sous la forme d'un produit de convolution entre une matrice (ou noyau de convolution) et d'une image.

Malheureusement, en appliquant de gros masques de convolution, on peut constater que le temps de calcul devient très important, Pour cela, on peut appliquer les masques dans l'espace des fonctions de Fourier qui rendront les calculs beaucoup plus rapides.

4.1 Filtrage à base de la convolution numérique

4.1.1 Convolution numérique bidimensionnelle

4.1.1.1 Définition

Considérons une image monochrome dans laquelle la fonction $I(i, j)$ représente l'intensité lumineuse du pixel de coordonnées (i, j) . La convolution numérique de cette fonction avec une réponse impulsionnelle bidimensionnelle $h(m, n)$ conduit à une nouvelle image de fonction $g(i, j)$. Cette convolution s'écrit :

$$g(i, j) = \sum_{m=-M}^M \sum_{n=-N}^N h(m, n) I(i - m, j - n) \quad (2.7)$$

$g(i, j)$ Est la somme, pondérée par les coefficients $h(m, n)$, des intensités des pixels appartenant à un voisinage du pixel de coordonnées (i, j) . Le traitement est dit localisé. La réponse impulsionnelle $h(m, n)$ est appelé masque de convolution, noyau de convolution, filtre, fenêtre, kernel, ...

4.1.1.2 Illustration de l'opération de convolution appliquée à une image numérique

La figure (2.8) illustre le passage d'un masque de convolution sur une image numérique monochrome. Le masque est déplacé dans toute l'image initiale pour obtenir une image traitée complète.

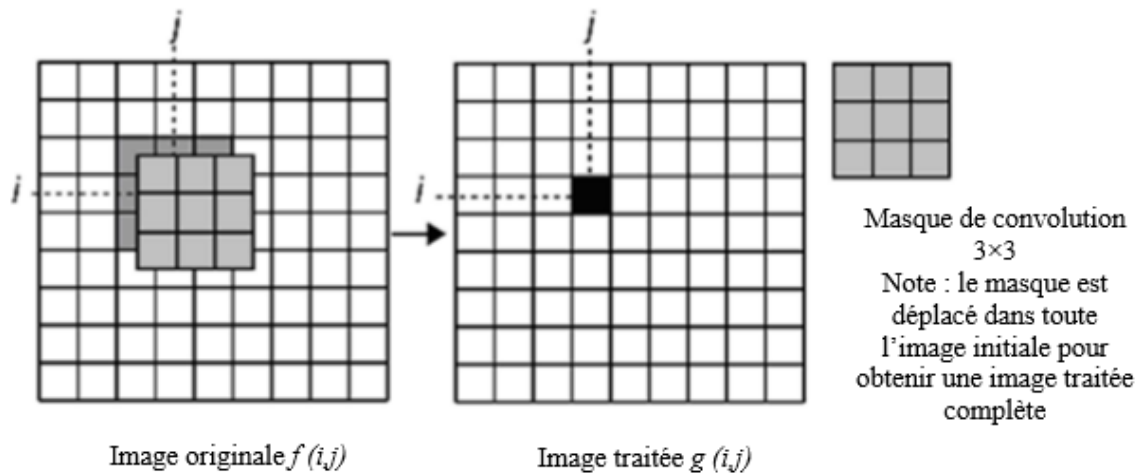


Fig.2.8 Application d'un masque de convolution sur une image [8].

4.1.2 Filtre moyenneur (flou uniforme)

Son principe consiste à remplacer chaque pixel par la valeur moyenne de ses voisins. Le filtre moyenneur permet de lisser l'image, réduit le bruit, Réduit les détails non importants et Brouille ou rend l'image floue.

Dans la figure (2.9), nous présentons les résultats d'image débruitée par un filtre moyenneur.

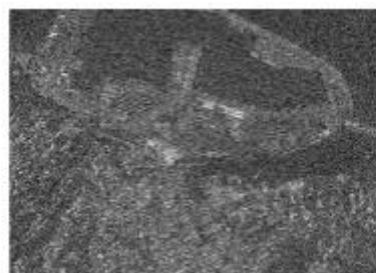


Image originale bruitée
PSNR = 20.00



Image débruitée
PSNR = 25.39

Fig.2.9 Application du filtre moyenneur avec un masque [3x3].

4.1.3 Filtre gaussien

Dans l'approche gaussien On utilise des masques réalisant la moyenne pondérée dans le voisinage du pixel considéré. Cependant les coefficients du filtre gaussien est calculer par :

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.8)$$

On outre Les pixels du voisinage qui sont proches du pixel central ont un poids plus fort (plus d'influence) que ceux qui sont plus éloignés [9].

Dans la Figure (2.10), nous présentons les résultats d'image débruitée par un filtre gaussien.

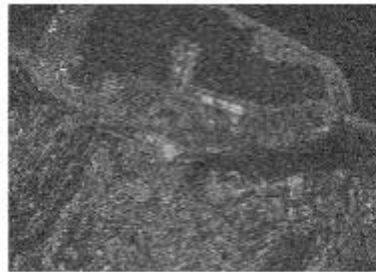


Image originale bruitée

PSNR = 20.00



Image débruitée

PSNR = 26.05

Fig.2.10 Application du filtre gaussien avec un masque [3x3].

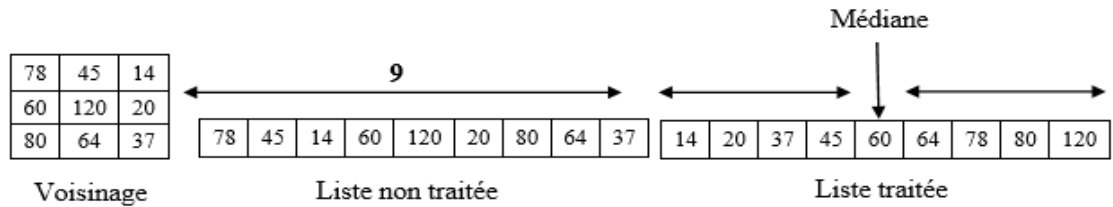
4.2 Filtre médian

Pour nettoyer le bruit dans une image, il existe mieux que le filtre moyenneur ou le filtre gaussien Il s'agit du filtre médian. C'est un filtre non-linéaire, qui ne peut pas s'implémenter comme un produit de convolution. Le filtrage médian procède tout d'abord par un tri des valeurs de niveau de gris du voisinage suivi d'une sélection de l'élément milieu du tri. Le tri se fait par ordre croissant généralement. Il conduit à former l'ensemble ordonné des valeurs de gris du voisinage de $I(x_0, y_0)$. Les éléments ordonnés étant notés $I(i)$, le tri croissant est caractérisé par:

$$I_1 < I_2 < \dots < \frac{I_{N+1}}{2} < I_{N-1} < I_N \quad (2.9)$$

L'élément médian du voisinage est $I_{\frac{N+1}{2}}$. Sa propriété est d'être précédée par $\frac{N-1}{2}$ valeurs inférieures et suivi par autant de valeurs supérieures. Le filtrage consiste à remplacer $I(x_0, y_0)$ par la valeur médiane du voisinage $I_{\frac{N+1}{2}}$.

Exemple de filtre médian



Le filtre médian est pratiquement toujours préféré au filtre moyenneur ou au filtre gaussien pour améliorer les images bruitées avec un bruit Impulsionnel. Après l'application de ce type de filtrage nous avons obtenu les résultats suivants Figure (2.11) :

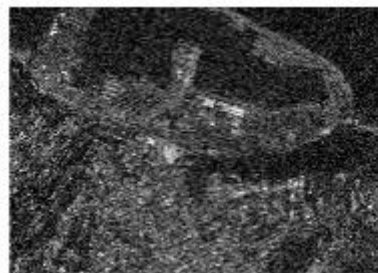


Image originale bruitée

PSNR=20.00



Image débruitée

PSNR =26.63

Fig.2.11 Application d'un filtre médian

4.3 Filtre SNN (Symmetric Nearest Neighbor)

Ce filtre est utilisé comme atténuateur du bruit avec la conservation des contours. Son principe repose sur la symétrie, en effet, pour chaque paire de pixels symétriques par rapport au pixel central, on sélectionne celui dont la valeur est la plus proche de celle du pixel central [9].



On fait ensuite la moyenne des 4 valeurs et on la remplace dans le pixel central.

Exemple

10	12	40	16	19	10
14	22	52	10	55	41
10	14	51	21	14	10
32	22	9	9	19	14
41	18	9	22	27	11
10	7	8	8	4	5

$\{9, 19\} \rightarrow 9$
 $\{51, 27\} \rightarrow 27$
 $\{21, 22\} \rightarrow 21$

Moyenne = 16

Après l'application de ce type de filtrage nous avons obtenu les résultats suivants Figure (2.12) :

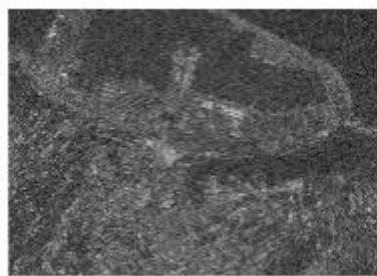


Image originale bruitée

PSNR = 20.00



Image débruitée

PSNR = 25.27

Fig.2.12 Application du filtre SNN.

4.4 Les filtres adaptatifs

4.4.1 Filtre de Lee

Le filtre de Lee [10] propose d'estimer la moyenne et la variance de R (R est l'information que l'on veut mesurer) par la moyenne locale observée, I et la variance locale observée, $Var(I)$. I et $Var(I)$ sont calculés sur une fenêtre. Cette méthode est dite LS (Local Statistic) [8].

R et $Var(R)$ sont exprimés en fonction de I et $Var(I)$ de la façon suivante:

$$\bar{R} = \bar{I} \tag{2.10}$$

$$var(R) = \frac{var(I) - \bar{I}^2 \sigma_u^2}{\sigma_u^2 + 1} \tag{2.11}$$

σ_u Est l'écart type du signal bruit U . $\sigma_u = C_u$ étant le coefficient de variation de U . Lee propose d'approximer le modèle multiplicatif du signal image I par un modèle linéaire de la forme $\hat{R} = aI + b\bar{R}$ où \hat{R} est

l'estimateur de R donnant la plus petite erreur quadratique par rapport à la moyenne (Minimum Mean Square Error MMSE). Les variables a et b sont choisies de façon à minimiser $E((\hat{R} - R)^2)$, E étant est l'espérance mathématique. Nous obtenons:

$$\hat{R} = \alpha I + (1 - \alpha)\bar{R} \quad (2.12)$$

Avec

$$\alpha = \frac{\text{var}(R)}{\text{var}(I)} \quad (2.13)$$

En remplaçant $\text{Var}(R)$ par l'expression de l'équation (2.11). On aura:

$$\alpha = \frac{\text{var}(I) - \bar{I}^2 \sigma_u^2}{(1 + \sigma_u^2) \text{var}(I)} \quad (2.14)$$

$1 + \sigma_u^2$ Peut être remplacée par 1 car $\sigma_u^2 \approx 0$ on aura donc:

$$\alpha = \frac{\text{var}(I) - \bar{I}^2 \sigma_u^2}{\text{var}(I)} = 1 - \frac{C_u^2}{C_I^2} \quad (2.15)$$

Finalement le filtre de Lee est défini de la façon suivante:

$$\hat{R}(t) = w(t)I(t) + (1 - w(t))\bar{I}(t) \quad (2.16)$$

Avec

$$w(t) = 1 - \frac{C_u^2}{C_I^2(t)}, \quad C_I^2(t) = \frac{\text{var}(I)}{\bar{I}(t)^2} \quad (2.17)$$

Notons que le facteur de pondération est une valeur entre 0 et 1. Si l'équation (2.17) donne une valeur négative pour $w(t)$, cette valeur sera remplacée par zéro. Le filtre de Lee peut être interprété de la façon suivante:

Le coefficient de variation au carré d'une région, C_I^2 est comparé au coefficient de variation au carré du bruit. Si $C_I^2 \leq C_u^2$ le facteur de pondération aura la valeur zéro. Ceci correspond à estimer $R(t)$ par la moyenne calculée sur la fenêtre $\bar{I}(t)$ (*région homogène*). Par contre, si $C_I^2 > C_u^2$ le facteur de pondération aura une valeur qui dépend du rapport $\frac{C_u^2}{C_I^2}$.

Le filtre a tendance à garder la valeur observée du pixel, $I(t)$, tant que ce rapport est plus petit (*région hétérogène*).

Nous avons appliqué Le filtre de Lee, La figure (2.13), montrent les résultats du filtrage

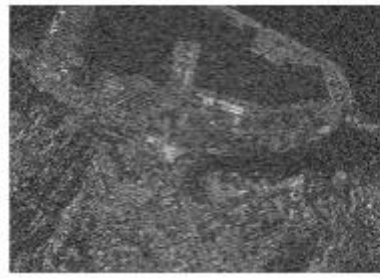


Image originale bruitée

PSNR = 20.00



Image débruitée

PSNR = 23.54

Fig.2.13 Application du filtre de Lee.

4.4.2 Filtre de Nagao

Dans l'approche de Nagao [1], on travaille sur une fenêtre de taille 5×5 entourant le pixel central (qui appartient à toutes ces fenêtres). On définit 9 fenêtres possibles (Figure 2.14), toutes de 9 pixels, identifiées par un indice k . Sur chaque fenêtre on mesure la moyenne m_K et la variance σ_K^2 des niveaux de gris. On choisit alors de remplacer le pixel central par la valeur moyenne de la fenêtre dont la variance est la plus faible.

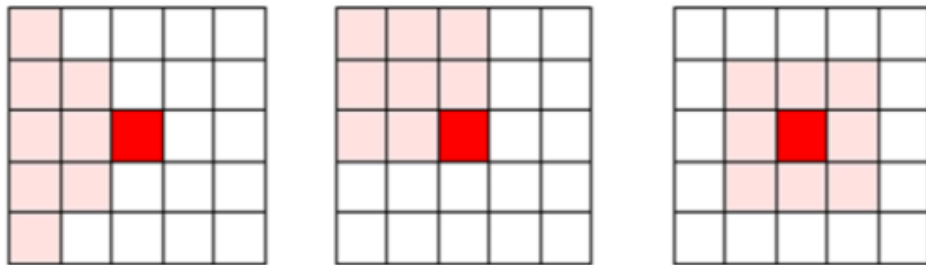
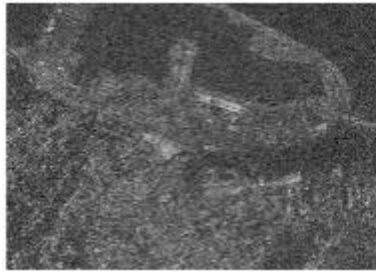


Fig.2.14 Le pixel central d'une fenêtre 5×5 appartient à 9 fenêtres de 9 pixels chacune : 4 se déduisent de la fenêtre représentée à gauche par rotations de $n \times 90^\circ$ de la fenêtre au centre et la fenêtre de droite [11].

Après l'application de ce type de filtrage, nous avons obtenus les résultats qui sont illustrés dans la figure (2.15).

*Image originale bruitée*

PSNR = 20.00

*Image débruitée*

PSNR = 23.07

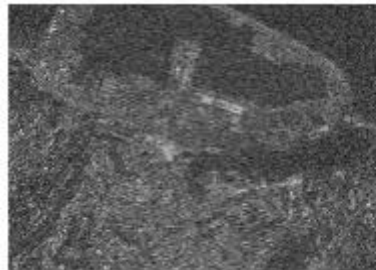
Fig.2.15 Application du filtre de Nagao.

4.4.3 Filtre de Kuan

Le filtre de Kuan [12] transforme le modèle multiplicatif du bruit en un modèle additif sur lequel il applique le critère de MMSE (Minimum Mean Square Error). Le filtre obtenu est un filtre qui a la même forme que le filtre proposé par Lee, mais avec une fonction de pondération différente:

$$w(t) = \frac{1 - \frac{c_u^2}{c_f^2(t)}}{1 + c_u^2} \quad (2.18)$$

Ce filtre se comporte relativement de la même manière que le filtre de Lee. La figure (2.16), montrent les résultats du filtrage.

*Image originale bruitée*

PSNR = 20.00

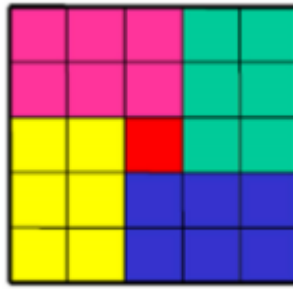
*Image débruitée*

PSNR = 23.54

Fig.2.16 Application du filtre de Kuan.

4.4.4 Filtre de Kuwahara

Ce filtre permet d'atténuer le bruit avec une conservation pour les contours, son principe repose sur le calcul de la valeur moyenne des sous-voisinages et conserver celle qui a la plus petite variance [9].



On a 4 régions donc 4 moyennes et 4 variances, le pixel central sera remplacé par la moyenne de la région de variance minimale.

Les résultats de l'application du filtre de *Kuwahara* sont illustrés dans la figure (2.17).

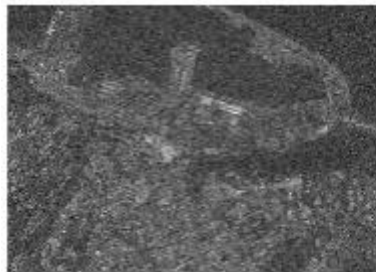


Image originale bruitée

PSNR = 20.00



Image débruitée

PSNR = 23.54

Fig.2.17 Application du filtre de Kuwahara.

4.5 Filtre de voisinage non linéaire

4.5.1 Filtre Bilatéral

Pour ce filtre, en tenant compte des poids de distances spatiales, l'image débruitée est définie par [13]:

$$\tilde{I}_{SUSAN}(x) = \frac{1}{C(x)} \int I(y) e^{-\frac{|y-x|^2}{\rho^2}} e^{-\frac{|I(y)-I(x)|^2}{h^2}} dy \quad (2.19)$$

Avec

$$C(x) = \int e^{-\frac{|y-x|^2}{\rho^2}} e^{-\frac{|I(y)-I(x)|^2}{h^2}} dy \quad (2.20)$$

Les paramètres ρ et h contrôlent séparément la moyenne du voisinage spatial et celle de l'intensité.

Nous avons appliqué le filtre bilatéral sur l'image satellite bruitée. La figure montre le résultat du filtrage.

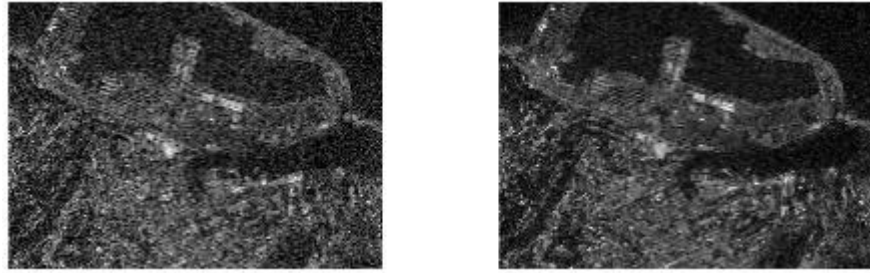


Image originale bruitée

PSNR = 20.00

Image débruitée

PSNR = 23.88

Fig.2.18 Application du filtre Bilatéral.

La méthode du filtre bilatéral [13] (SUSAN) (Figure 2.18) améliore légèrement l'image en intégrant le lissage spatial. Mais nous pouvons observer toutefois des effets de pixellisation dans les images restaurées, car ces filtres visent implicitement à reconstruire une image constante par morceaux. Cependant, ce type de filtrage est utilisé beaucoup plus pour les images couleurs.

4.5.2 Moyennes non-locales (NL-means algorithm)

L'algorithme de moyennes non-locales (NL-means) est proposé par A. Buades, B. Coll. et J. M. Morel dans [14]. L'idée est d'utiliser les informations redondantes dans l'image, et plus particulièrement le fait qu'une vignette dans l'image ressemble à d'autres vignettes de cette image. Un voisinage d'un pixel x est défini comme les points y de l'image pour lesquels on peut associer des vignettes similaires. L'image débruitée est donnée par :

$$\tilde{I}_{NL\text{-}mean_{s_n,a}}(x) = \frac{1}{c(x)} \int I(y) \cdot e^{-\frac{G_a * |I(x+\cdot) - I(y+\cdot)|^2}{h^2}} dy \quad (2.21)$$

Où G_a est le noyau gaussien d'écart-type a , h est un paramètre de filtrage et la fonction

$$c(x) = \int e^{-\frac{G_a * |I(x+\cdot) - I(z+\cdot)|^2}{h^2}} dz \quad (2.22)$$

est un facteur de normalisation.

Cet algorithme est optimal asymptotiquement sous un modèle statistique d'image. Récemment, dans [15], J. Boulanger et C. Kervrann proposent de sélectionner le voisinage spatial par une méthode adaptative pour chaque pixel. L'approche est basée sur une analyse statistique de l'équivalence entre le biais et la variance de l'estimateur de moyennes non-locales.

Une autre amélioration du filtre de moyennes non-locales est proposée dans [16], où A. Buades, B. Coll. et J. M. Morel montrent qu'une simple régression linéaire permet d'éviter les effets d'escaliers.

L'application de l'algorithme « *non local means* » a abouti au résultat suivant (Figure 2.19) :

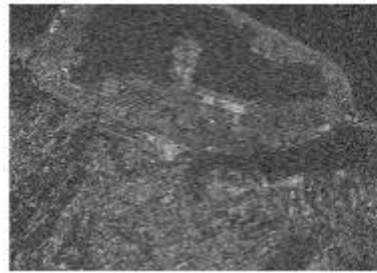


Image originale bruitée

PSNR = 20.00



Image débruitée

PSNR = 27.66

Fig.2.19 Application de l'algorithme « *non local means* »

4.6 Le filtre de Wiener

Le modèle de dégradation d'image peut être défini par :

$$g = HI + n \quad (2.23)$$

g : L'image dégradée,

I : l'image à restaurer,

H : la dégradation,

n : le bruit additif.

L'objectif de la restauration d'image est d'obtenir une estimée \hat{I} de l'image originale I à partir de l'image dégradée g .

La fonction de coût du filtre de Wiener est une estimation de la moyenne statistique de l'erreur entre I et \hat{I} :

$$J(\hat{I}) = E \{ \|I - \hat{I}\|^2 \} \quad (2.24)$$

L'image \hat{I} qui minimise (II.14) est donnée dans l'espace de Fourier par [17]:

$$\hat{I} = \frac{1}{H} \frac{|H|^2}{|H|^2 - \frac{S_{nn}}{S_{II}}} g \quad (2.25)$$

Où S_{II} et S_{nn} représentent respectivement les densités spectrales de puissance de l'image originale I et du bruit n . Le filtre de *Wiener* exige des connaissances à priori sur ces deux grandeurs.

Les résultats de l'application du filtre de *Wiener* sont illustrés sur la figure (2.20).

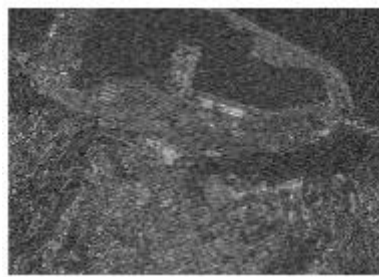


Image originale bruitée

PSNR = 20.00



Image débruitée

PSNR = 26.59

Fig.2.20 Application du filtre de Wiener

4.7 Les Ondelettes (Débruitage par seuillage de coefficients d'ondelettes)

Les ondelettes ont été introduites à la fin des années 80 comme outil d'analyse multi échelle du signal et des images. La théorie permet de sélectionner des bases intrinsèquement adaptées à la représentation d'une classe de signaux et ainsi d'approcher la fonction précisément en respectant le plus possible l'information. Ainsi, les ondelettes sont appliquées à la restauration d'images bruitées et à la compression des images.

Le débruitage par ondelettes consiste à décomposer l'image bruitée dans une base orthogonale en ondelettes, à supprimer les coefficients inférieurs à un seuil donné en utilisant le seuillage dur, doux, NNG, SCAD,...et à reconstruire une image par la formule de synthèse.

Le choix de seuil $T = \sigma\sqrt{2\log N}$ où N est la taille d'échantillonnage, σ^2 est la variance du bruit donne un estimateur optimal au sens minimax quand N tend vers l'infini [17]. Ces algorithmes sont formulés par Donoho et Johnstone

dans [17], [18]. L'image débruitée est représentée par des coefficients larges et la performance de l'estimateur dépend donc de la vitesse de décroissance des coefficients. Grâce à sa localisation en temps et en échelle, la base d'ondelettes permet de représenter des fonctions régulières par morceaux plus efficacement qu'avec des bases orthogonales classiques comme la base de Fourier ou en cosinus car la décroissance des coefficients en ondelettes dans un voisinage d'une discontinuité est plus rapide que celle des coefficients de Fourier.

Nous représentons sur la figure (2.21), les résultats obtenus par l'application de cet algorithme.



Image originale bruitée

PSNR = 20.00



Image débruitée

PSNR = 26.66

Fig.2.21 Débruitage par seuillage de coefficient d'ondelettes.

5. Conclusion

Nous avons présenté, dans ce chapitre, un large panorama des méthodes permettant d'obtenir une restauration ou une amélioration des images. Chacune des approches présente des caractéristiques qui les rendent opérationnelle pour une certaine classe d'images. Nous avons énoncé ces caractéristiques et présenté, pour chaque méthode, son comportement général.

Chapitre 3

Description du logiciel développé

1. Introduction

Le filtrage des images nécessite un calcul judicieux et parfois compliqué pour émettre ou extraire une information avec une grande fidélité. Refaire à chaque fois les calculs est un travail non rentable sur le plan temps et effort. C'est pourquoi les logiciels forment un outil de calcul très utile et bénéfique. Le logiciel que nous avons réalisé permet de filtrer les images en utilisant un large panorama de méthodes et d'algorithmes.

Après avoir présenté, dans le chapitre précédent, l'aspect mathématique de plusieurs algorithmes de débruitage et modèles de filtres, nous allons passer dans ce chapitre à la description du logiciel développé sous *Matlab*. Dans ce qui suit, nous allons tout d'abord donner un aperçu sur l'outil de programmation exploité avant de passer à la description du logiciel.

2. Outil de programmation utilisés « *MATLAB* »

L'outil de programmation principal utilisé pour le développement du logiciel est le *MATLAB*.

MATLAB dont le nom provient de **Matrix Laboratory** est un langage de calcul scientifique basé sur le type de variable matricielle. Outre ses grandes capacités de calcul numérique appliqué à différents domaines scientifiques tels que le traitement de signal, l'automatique,... etc., *MATLAB* intègre des fonctionnalités graphiques de grande qualité en 2D ou 3D (2 ou 3 dimensions).

L'utilisation de *MATLAB* peut se faire de façon interactive en exécutant des commandes directement après le prompt ou par des scripts. Ces derniers peuvent être des simples programmes ou des fonctions avec des paramètres d'appel et de retour. Les fonctions sont très utiles dans la mesure où chaque utilisateur peut étendre les possibilités de *MATLAB* à son domaine d'application [19].

3. Description du Logiciels réalisé

3.1 Menu principal

Le menu principal apparaît sous forme d'une page d'accueil ou autrement dit, la page du *Bienvenu*. La figure ci-après (figure 3.1), illustre une vue générale de l'environnement comprenant un assemblage de boutons liés directement aux différents modèles de filtres réalisés : filtre moyennneur, filtre Gaussien, filtre médian, filtre SNN, filtre de Lee, filtre Nagao, filtre de Kuan, filtre de Kuwahara, filtre bilatéral, filtage à base de l'algorithme non local means, filtre de wiener et enfin les ondelettes (seuillage par coefficients d'ondelettes). En plus un bouton « quitter » afin de quitter l'application.

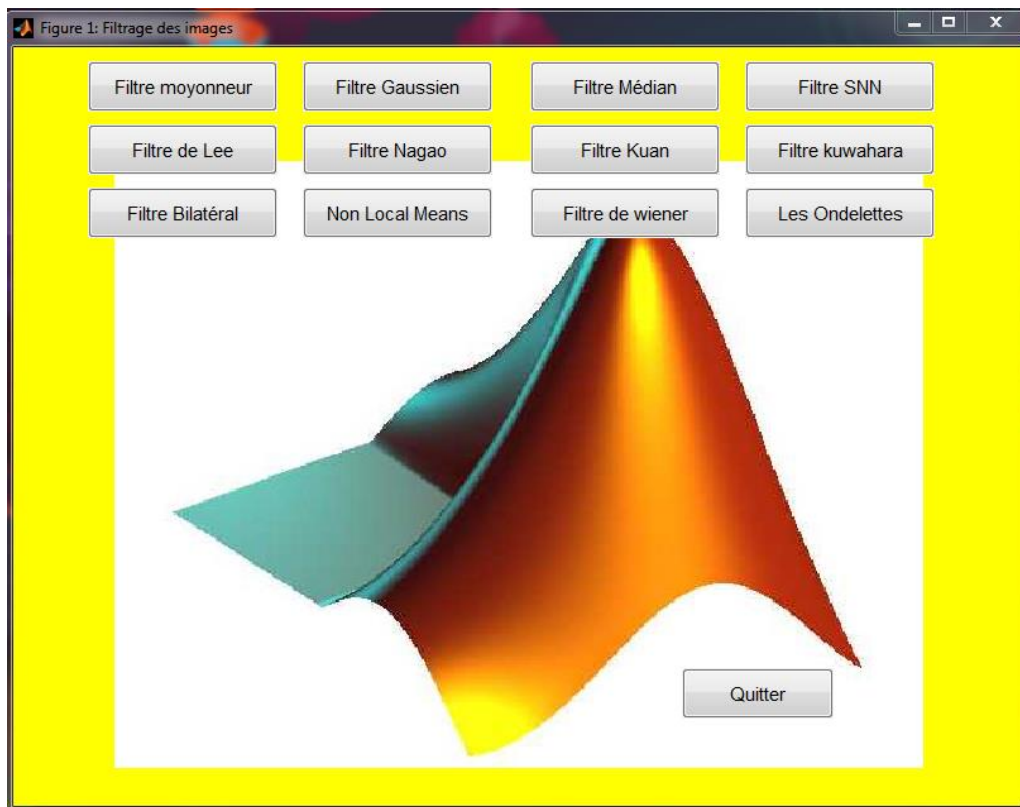


Fig. 3. 1 : Fenêtre du menu principal.

3. 2 Fenêtre « Filtre moyonneur »

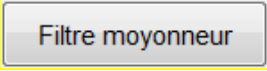
En appuyant sur le bouton correspondant «  », la fenêtre suivante s'ouvre :



Fig. 3. 2 : Contenu de la fenêtre « Filtre Moyonneur »

Cette fenêtre (figure 3.2) contient plusieurs rubriques et boutons, commençant par la rubrique **choix d'une image test**, où l'utilisateur choisira une image test, en revanche dans la rubrique **Bruit** l'utilisateur peut ajouter un bruit de type (impulsionnel, Gaussien ou multiplicatif) avec une quantité contrôlée.

Après avoir choisir l'image test et le type de bruit, l'utilisateur doit choisir **la taille du filtre**, ensuite en cliquant sur le bouton « *Application du filtre* » un algorithme qui réalise le filtre moyenneur sera exécuté ce qui permet un filtrage de l'image dégradée (l'image dégradée ainsi que l'image débruitee seront afficher dans une nouvelle fenêtre).

Afin de quantifier la qualité de filtrage, il suffit de cliquer sur le bouton « *calcul du PSNR* » afin de calculer le PSNR initial et le PSNR finale.

Le bouton « *Quitter* » permet de quitter l'application, cependant le bouton « *Back* » permet le retour vers la fenêtre du menu principale.

3.3 Fenêtre « Filtre Gaussien »

En appuyant sur le bouton correspondant « **Filtre Gaussien** », la fenêtre suivante s'ouvre :



Fig. 3.3 : Contenu de la fenêtre « Filtre Gaussien »

Cette fenêtre (figure 3.3) est similaire à la fenêtre précédente (filtre moyenneur) la seule différence réside dans le bouton « *Application du filtre* », lors du clic sur ce bouton un algorithme qui réalise le filtre Gaussien sera exécuté.

3.4 Fenêtre « Filtre Médian »

En appuyant sur le bouton correspondant «  », la fenêtre suivante s'ouvre :

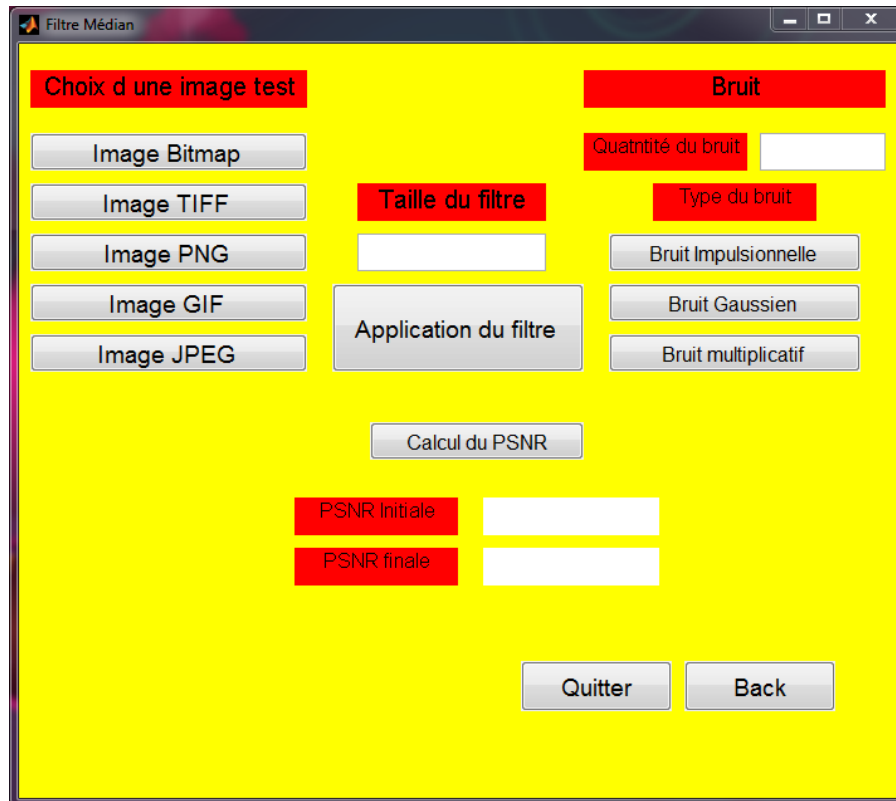


Fig. 3. 4 : Contenu de la fenêtre « Filtre Médian »

Cette fenêtre (3.4) est similaire aux fenêtres précédentes, lors du clic sur le bouton « *Application du filtre* » un algorithme qui réalise le filtre médian sera exécuté.

3.5 Fenêtre « Filtre SNN»

En appuyant sur le bouton correspondant «  », la fenêtre suivante s'ouvre (figure 3.5):



Fig. 3.5 : Contenu de la fenêtre « Filtre SNN »

Cette fenêtre est similaire aux précédentes, l'ordre du clic sur le bouton « *Application du filtre* » un algorithme qui réalise le filtre SNN sera exécuté.

3.6 Fenêtre « Filtre de Lee»

En appuyant sur le bouton correspondant «  », la fenêtre suivante s'ouvre (figure 3.6):

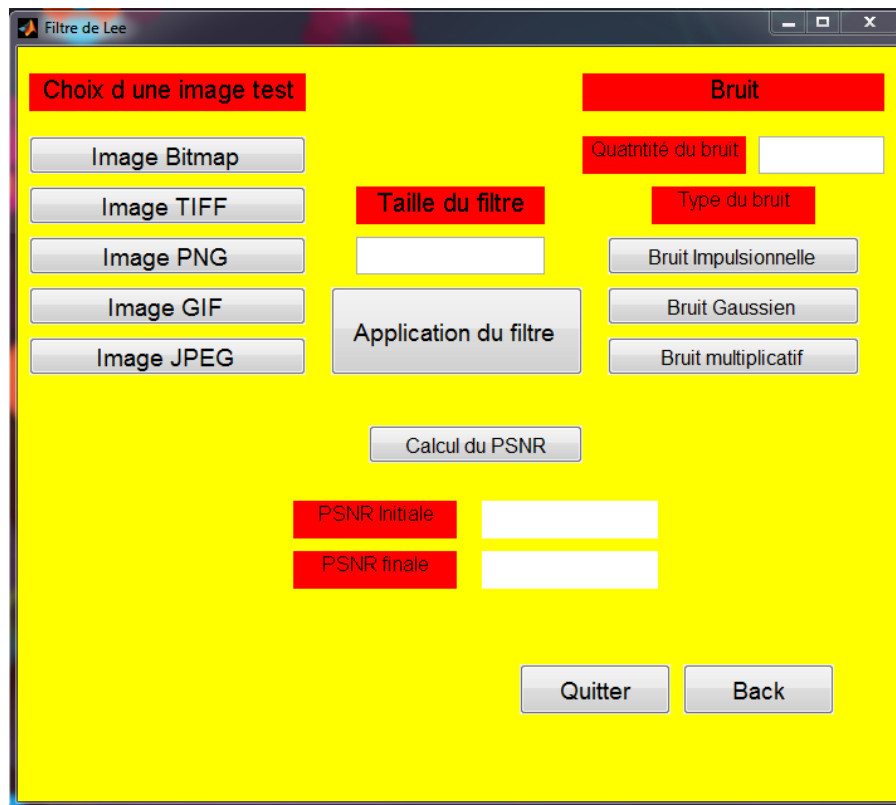


Fig. 3. 6 : Contenu de la fenêtre « Filtre de Lee »

Cette fenêtre est similaire aux précédentes, lors du clic sur le bouton « *Application du filtre* » un algorithme qui réalise le filtre de Lee sera exécuté.

3.7 Fenêtre « Filtre Nagao »

En appuyant sur le bouton correspondant «  », la fenêtre suivante s'ouvre (figure 3.7):



Fig. 3. 7 : Contenu de la fenêtre « Filtre Nagao »

Cette fenêtre est similaire aux précédentes, lors du clic sur le bouton « *Application du filtre* » un algorithme qui réalise le filtre de Nagao sera exécuté.

3. 8 Fenêtre « Filtre de Kuan »

En appuyant sur le bouton correspondant «  », la fenêtre suivante s'ouvre (figure 3.8):

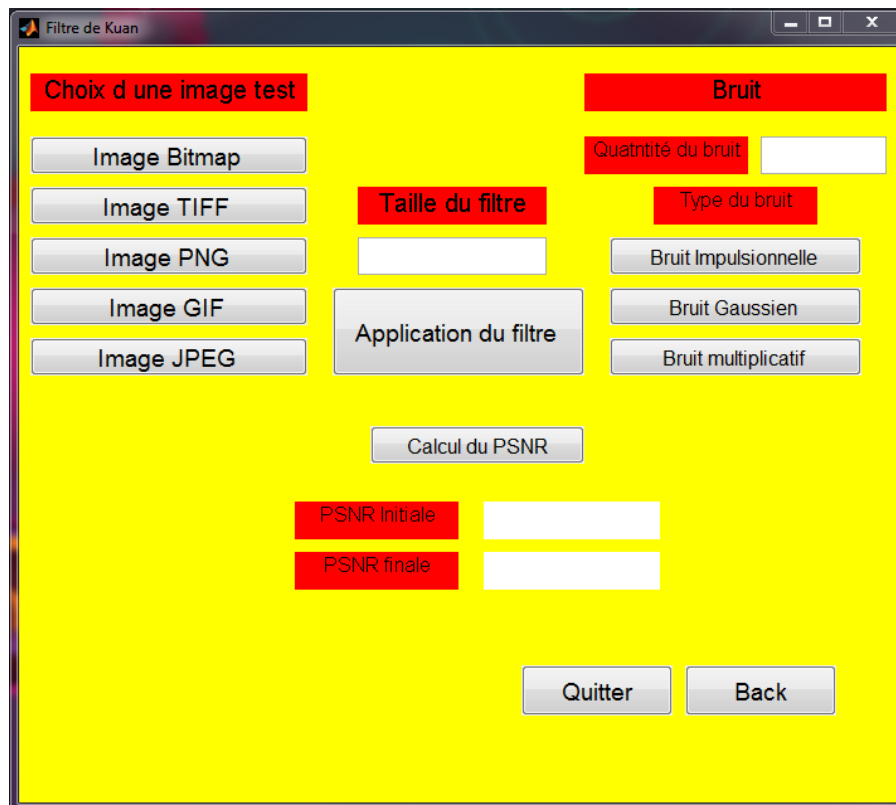


Fig. 3. 8 : Contenu de la fenêtre « Filtre de Kuan »

Lors du clic sur le bouton « *Application du filtre* » un algorithme qui réalise le filtre de Kuan sera exécuté.

3.9 Fenêtre « Filtre Kuwahara »

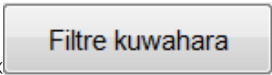
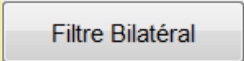
En appuyant sur le bouton correspondant «  », la fenêtre suivante s'ouvre (figure 3.9):



Fig. 3. 9 : Contenu de la fenêtre « Filtre Kuwahara »

Lors du clic sur le bouton « *Application du filtre* » un algorithme qui réalise le filtre de Kuwahara sera exécuté.

3. 10 Fenêtre « Filtre Bilatéral »

En appuyant sur le bouton correspondant «  », la fenêtre suivante s'ouvre (figure 3.10):

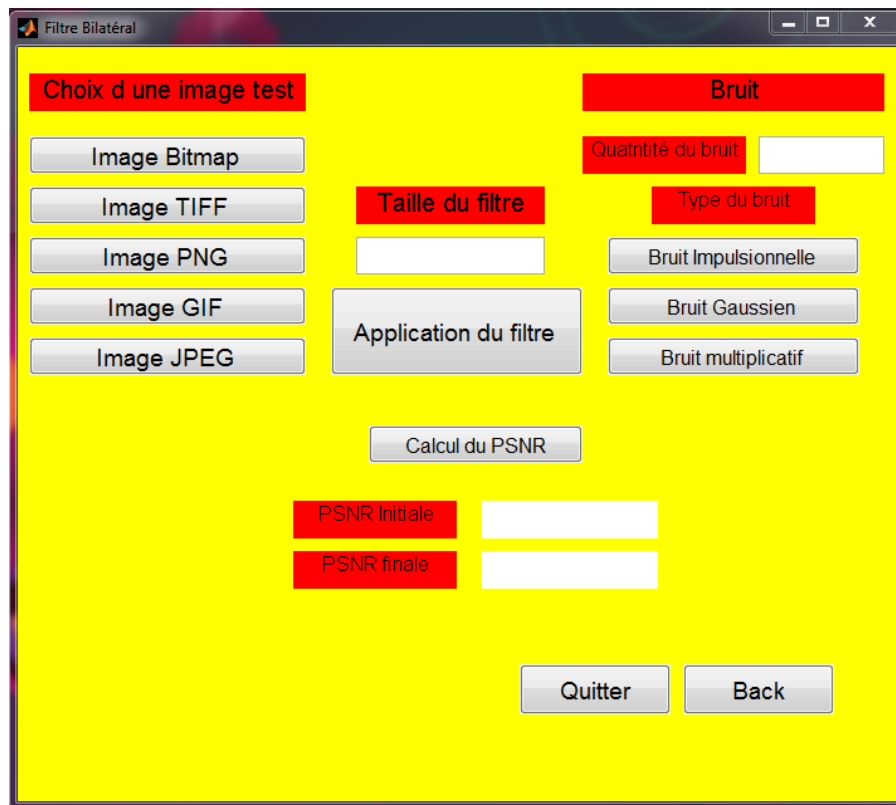



Fig. 3. 10 : Contenu de la fenêtre « Filtre Bilatéral »

Lors du clic sur le bouton « *Application du filtre* » un algorithme qui réalise le filtre bilatéral sera exécuté.

3. 11 Fenêtre « Non Local Means »

En appuyant sur le bouton correspondant «  », la fenêtre suivante s'ouvre (figure 3.11):

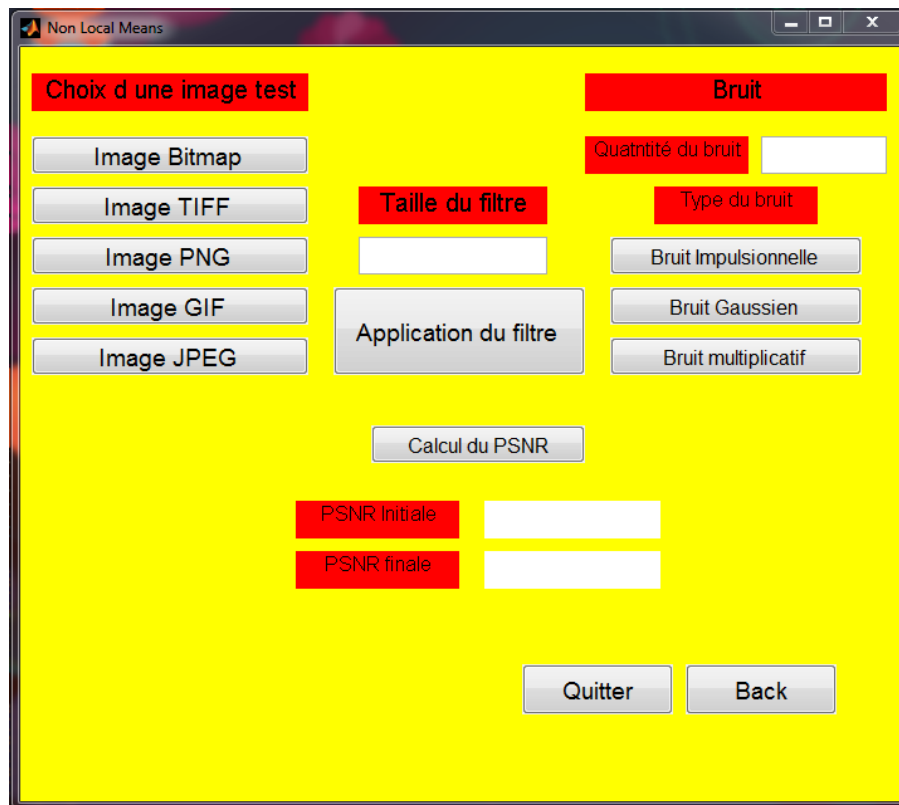


Fig. 3. 11 : Contenu de la fenêtre «Non Local Means»

Lors du clic sur le bouton « *Application du filtre* » l'algorithme Non Local Means sera exécuté.

3. 12 Fenêtre « Filtre de Wiener »

En appuyant sur le bouton correspondant «  », la fenêtre suivante s'ouvre (figure 3.12):

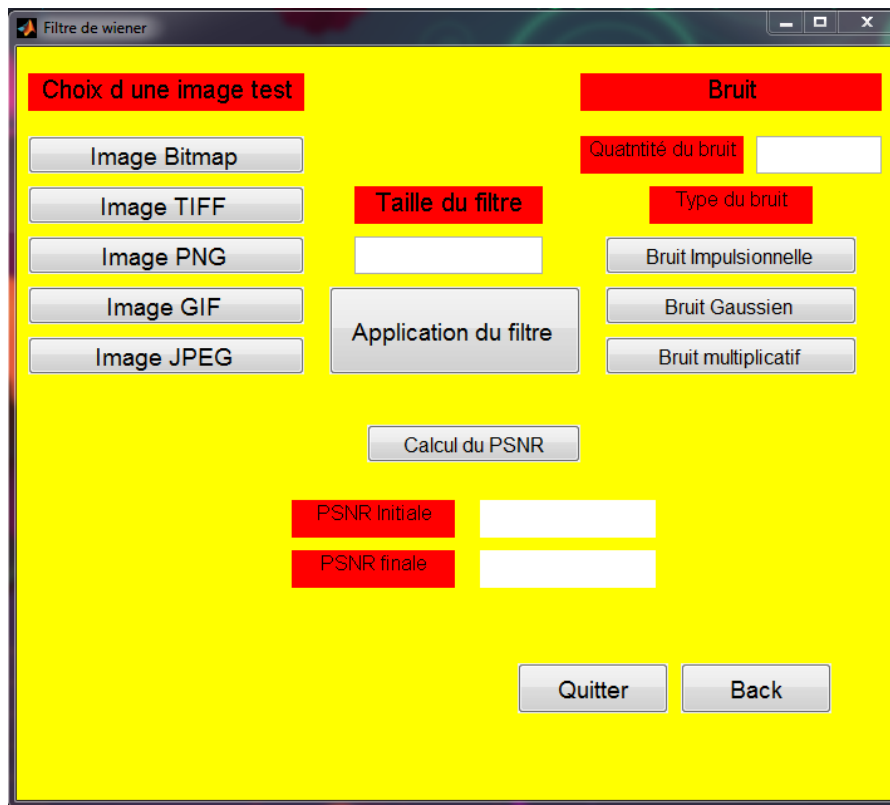


Fig. 3. 12 : Contenu de la fenêtre «Filtre de Wiener»

Lors du clic sur le bouton « *Application du filtre* » un algorithme qui réalise le filtre de Wiener sera exécuté.

3. 13 Fenêtre « Débruitage par seuillage de coefficients d'ondelettes »

En appuyant sur le bouton correspondant «  », la fenêtre suivante s'ouvre (figure 3.12):

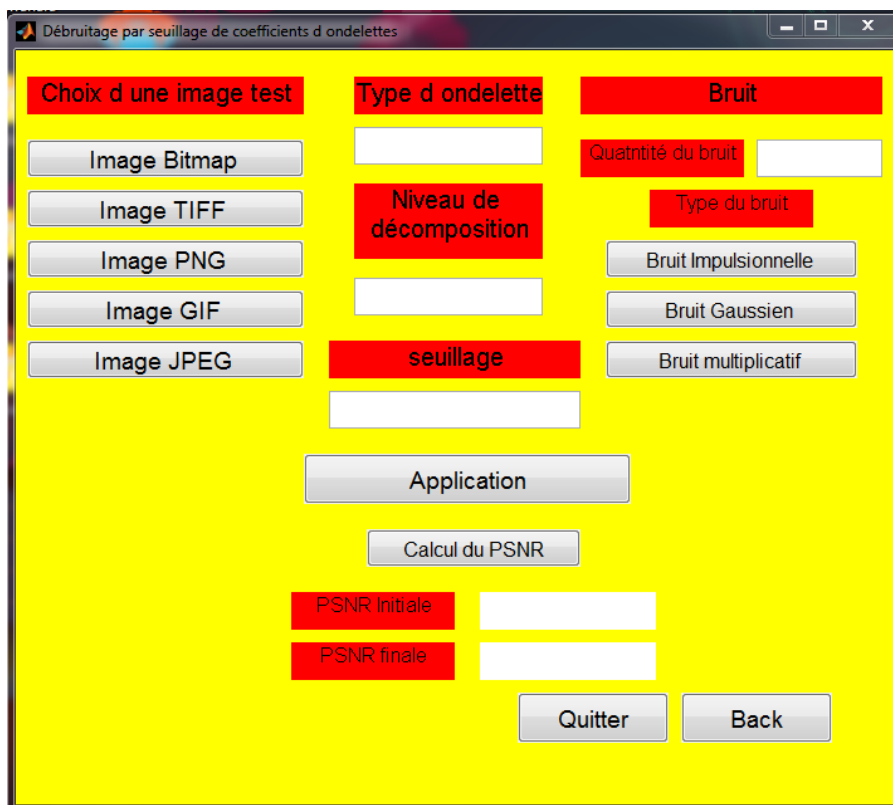


Fig. 3. 12 : Contenu de la fenêtre «Débruitage par seuillage de coefficients d'ondelettes»

Cette fenêtre est un peu différente par rapport aux précédentes, elle contient quelques nouvelles rubriques telle que : le type d'ondelettes où l'utilisateur doit insérer le nom d'ondelettes qui veut l'utiliser (haar, db2, db3,..., db45, sym2, sym3,.....sym7, coif1, coif2,..... coif5,.....).

D'autres paramètres doivent être insérer par l'utilisateur, il s'agit du niveau de décomposition (1, 2, 3,...) ainsi que le type de seuillage : soft (s), hard(h), NNG, SCAD.

Lord du clic sur le bouton « Application du filtre » un algorithme de débruitage par seuillage de coefficients d'ondelettes sera exécuté.

4. Conclusion

Ce chapitre a fait l'objet d'une description brève du logiciel de calcul que nous avons élaboré. Il offre la possibilité de filtrer des images dégradé en utilisant plusieurs algorithme et méthodes, il a l'avantage de traiter plusieurs type d'image (Bitmap, Gif, Png...) ainsi qu'il présente un outil très intéressant aux étudiants afin qu'ils se familiarisent avec la restauration des images.

Conclusion et perspectives

Dans ce mémoire, nous avons développé un logiciel de traitement d'images dans le but de récupérer des images de bonne qualité à partir des images bruitées de mauvaise qualité, on se basant sur les méthodes et les algorithmes de restauration les plus connues et les plus performants.

Nous avons tout d'abord présenté un récapitulatif sur les définitions et notions élémentaires liées au traitement d'images. Commencant par l'exposition de quelques notions de bases sur les images numériques (définition, type, caractéristiques,...). Ensuite nous avons abordé les différentes étapes nécessaires dans une chaîne de traitement d'images (à partir du capteur jusqu'à l'analyse des résultats).

Dans la deuxième partie nous avons présentés un large panorama des méthodes permettant d'obtenir une restauration ou une amélioration des images. Chacune des approches présente des caractéristiques qui les rendent opérationnelle pour une certaine classe d'images. Nous avons énoncé ces caractéristiques et présenté, pour chaque méthode, son comportement général.

La troisième partie a fait l'objet d'une description brève du logiciel de calcul que nous avons élaboré.

Le logiciel développé au sein de ce mémoire présente un outil de simulation largement suffisant pour satisfaire, principalement, les besoins pédagogiques des étudiants, ainsi qu'il présente un outil très intéressant aux étudiants afin qu'ils se familiarisent avec la restauration des images.

Comme perspectives, nous proposons d'enrichir ce travail par élargir le logiciel que nous avons développé par l'intégration d'autre opération de traitement d'images telle que la compression et la segmentation des images.

Bibliographies

- [1] Zhang, X. (2006). *Reconstruction et Régularisation en Tomographie par une Méthode de Fourier Basée sur la Variation Totale*, Thèse de doctorat : Université de Bretagne-Sud.
- [2] T. F. Chan, J. Shen, and L. Vese. (2003). *Variational pde models in image processing*. Notices of AMS, 50(1):14–26.
- [3] Li, S. Z. (1995). *Markov Random Field Modeling in Computer Vision*. New York: Springer-Verlag.
- [4] Boubchir, L. (2007). *Approches bayésiennes pour le débruitage des images dans le domaine des transformées multi-échelles parcimonieuses orientées et non orientées*. Thèse doctorat. Université de Caen/ Basse Normandie.
- [5] Lanani, A. (2021). *Polycopie Traitement d'images*. Université Abess Laghrour Khenchela.
- [6] Site officiel de l'agence spatial algérienne. <http://www.asal.dz>.
- [7] M. Ghazel « *Adaptive Fractal and Wavelet Image Denoising* », *Thèse de doctorat Electrical and Computer Engineering*, University of Waterloo, 2004.
- [8] E. Tisserand, J.F. Pautex, P. Schweitzer (2008). *Analyse et traitement des signaux* 2^{ème} édition. DUNOD, Paris.
- [9] Lanani abderrahim. *Analyse et Optimisation d'Artefact dans une Image*, UNIVERSITE EL HADJ LAKHDAR- BATNA, mémoire de magister 2012.
- [10] Lee Jong-Sen. (1980). *Digital image enhancement and noise filtering by use of local statistics*. IEEE Trans. Pattern Analysis and Machine Intelligence, 2(2): 165-168.

- [11] I. Bloch, Y. Gousseau, H. Maître, D. Matignon, B. Pesquet-Popescu, F. Schmitt, M. Sigelle, F. Tupin. (2004). *Le traitement des images, Tome 2*, Polycopié du cours ANIM, version 5.0. Département TSI - Télécom-Paris.
- [12] E. Tisserand, J.F. Pautex, P. Schweitzer (2008). *Analyse et traitement des signaux 2ème édition*. DUNOD, Paris.
- [13] S. M. Smith and M. Brady. (1995). *SUSAN. A new approach to low-level image processing*. Int. J. Com. Vis., 23(1).
- [14] A. Buades, B. Coll, and J. M. Morel. (2005). *A review of image denoising algorithms, with a new one*. Multiscale Model. Simul.4(2).
- [15] J. Boulanger & C. Kervrann. (2005). *Local adaptivity to variable smoothness for exemplar based image denoising and representation*. Rapport de Recherche RR-5624, INRIA.
- [16] *The staircasing effect in neighborhood filters and its solution*. IEEE Transactions on Image Processing, 15(6), 2006.
- [17] Donoho, D. L., Johnstone, I. M., Kerkyacharian, G., & picard, D. (1995). *Wavelet shrinkage: Asymptopia?* J. R. Statist. Soc. B., 57(2): 301–337.
- [18] Donoho D.L. & Johnstone I.M. (1994). *Ideal Denoising in an orthonormal basis chosen from a library of bases*. Department of Statistics. Stanford University, 1994
- [19] M.MOKHTARI. *matlab5.2 & 5.3 et simulink2 & 3 pour étudiants et ingénieurs* », éd. springer. 2000.

Résumé

Les images issues des différents systèmes d'imagerie sont généralement dégradées par un bruit. La tâche qui consiste à restaurer une image de bonne qualité à partir de sa version dégradée est communément appelée restauration. Celui-ci a engendré une importante littérature en prétraitement des images. Cependant, il faut recourir à des méthodes de filtrage appropriées pour réduire le bruit dans les images. Ces dernières années, plusieurs approches de filtrage linéaire et non linéaire ont été élaborées tout d'abord très intuitives, mais progressivement de plus en plus complexes. Dans notre travail, nous analysons ces approches commençons par les filtres de voisinage linéaire et non linéaire, les filtres adaptatives et le filtrage fréquentielle et bien évidemment les ondelettes. En revanche, le filtrage des images nécessite un calcul judicieux et parfois compliqué pour émettre ou extraire une information avec une grande fidélité où refaire à chaque fois les calculs est un travail non rentable sur le plan temps et effort, c'est pourquoi les logiciels forment un outil de calcul très utile et bénéfique. Dans notre travail nous avons développé un logiciel de traitement d'images dans le but de récupérer des images de bonne qualité à partir des images bruitées de mauvaise qualité, on se basant sur les méthodes et les algorithmes de restauration les plus connues et les plus performants. Ce logiciel présente un outil de simulation largement suffisant pour satisfaire, principalement, les besoins pédagogiques des étudiants, ainsi qu'il présente un outil très intéressant aux étudiants afin qu'ils se familiarisent avec la restauration des images.

Mots-clés

Traitement d'images, débruitage, filtres de voisinage, filtres adaptatives, filtrage fréquentielle, ondelettes, Logiciels

Abstract

Images from different imaging systems are generally degraded by noise. The task of restoring a good quality image from its degraded version is commonly referred to as restoration. This has generated a large literature in image preprocessing. However, appropriate filtering methods must be used to reduce noise in the images. In recent years, several linear and non-linear filtering approaches have been developed, initially very intuitive, but gradually more and more complex. In our work, we analyze these approaches start with linear and nonlinear neighborhood filters, adaptive filters and frequency filtering and of course wavelets. On the other hand, the filtering of the images requires a judicious and sometimes complicated calculation in order to transmit or extract information with great fidelity. Redoing the calculations every time is a waste of time and effort, which is why software is a very useful and beneficial calculation tool. In our work, we have developed image processing software with the aim of recovering good quality images from poor quality noisy images, based on the best known and most popular restoration methods and algorithms efficient. This software presents a simulation tool largely sufficient to satisfy, mainly, the educational needs of the students, as well as it presents a very interesting tool to the students so that they become familiar with the restoration of the images.

Keywords

Image processing, denoising, neighborhood filters, adaptive filters, frequency filtering, wavelets, Software