

Université de Montréal

Débruitage d'image par fusion de filtrage spatio-fréquentielle

par
Djenabou Barry

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Mars, 2018

© Djenabou Barry, 2018.

RÉSUMÉ

Dans ce projet de mémoire, nous nous intéressons à la fusion de deux algorithmes de débruitage d'images très différents et opérant respectivement dans le domaine spatial pour l'un des deux et dans le domaine fréquentiel pour l'autre.

La technique proposée permettra de fusionner les caractéristiques essentielles de ces deux algorithmes. L'un est basé sur le fait qu'une image naturelle possède de façon inhérente, des auto-similarités (Non-Local means algorithm) et s'avère particulièrement efficace pour débruiter les contours, détails et les hautes fréquences spatiales d'une image. L'autre algorithme est basé sur le fait que l'on peut décomposer chaque partie de l'image en un nombre restreint ("sparse") de fonctions élémentaires périodiques (*via* la transformé en cosinus discrète) et est particulièrement pertinent pour débruiter les zones homogènes présentes dans une image.

L'image dégradée par un bruit additif gaussien est premièrement débruitée dans les deux domaines et segmentée pour la détection de contours. Ces trois estimations sont ensuite fusionnées pour obtenir l'image finale débruitée. L'article [15] et l'article [73] permettront de détecter les contours.

Nos expérimentations donnent de bons résultats de débruitage d'images. L'évaluation de ces résultats appuyés sur l'aspect visuel, l'erreur quadratique moyenne (MSE) et le rapport signal sur bruit (PSNR), démontre la pertinence et l'efficacité de l'approche. Cette technique donne de meilleurs résultats que celles de [11] et [55, 78].

Mots clés: traitement d'image, débruitage, filtrage, restauration, réduction de bruit.

ABSTRACT

In this work, we are interested in the fusion of two different algorithms of denoising and operating in the space domain for one of the two and in the frequency domain for the other one.

The proposed technique will merge (fuse) the essential characteristics of these two algorithms. One of them is based on the fact that any natural image inherently exhibits spatial self-similarities (Non-Local means algorithm) and this property turns out to be especially efficient for denoising the contours, details and the spatial high frequencies of an image.

The other algorithm is based on the fact that the image/signal is sparse and by this fact, each part of the image can be broken into a small number of periodic elementary functions (*via* the Discrete Cosine Transform). This latter denoising algorithm has proven to be especially effective for denoising homogeneous regions existing in a natural image.

The image, degraded by a Gaussian additive noise is first denoised in both domains and segmented for contour detection. The three estimates are then fused to obtain the final denoised image. The article [15] and the article [73] will allow us to detect contours.

Our experiments give good results of denoising. The evaluation of these results, based on the visual aspect, the mean squared error (MSE) and the signal-to-noise ratio (PSNR), demonstrates the relevance and effectiveness of the approach. This technique gives better results than those of [11] and [55, 78].

Keywords: image processing, denoising, filtering, restoration, noise reduction.

TABLE DES MATIÈRES

RÉSUMÉ	iii
ABSTRACT	iv
TABLE DES MATIÈRES	v
LISTE DES TABLEAUX	vii
LISTE DES FIGURES	ix
LISTE DES SIGLES	xvi
CHAPITRE 1 : INTRODUCTION	1
CHAPITRE 2 : GÉNÉRALITÉ	4
2.1 Caractéristiques d'une image	4
2.2 Dégradation de l'image	5
2.2.1 Bruit Gaussien	6
2.2.2 Bruit Uniforme	6
2.2.3 Bruit "Poivre et Sel"	6
2.2.4 Bruit Périodique	7
2.3 Débruitage des images	9
2.4 Fusion d'images	9
2.5 Critère d'analyse du débruitage d'images	10
2.5.1 Critère subjectif (aspect visuel)	10
2.5.2 Critère objectif	10
CHAPITRE 3 : DÉBRUITAGE D'IMAGE : ÉTAT DE L'ART	12
3.1 Méthodes de débruitage par moyennes	12
3.2 Méthodes utilisant des dictionnaires de patches	13

3.3	Méthodes de débruitage par les moyennes Non-Local (NL-Means)	14
3.4	Méthodes de débruitage par la transformée en ondelettes	14
3.5	Méthodes hybrides de débruitage	15
3.6	Non-Local Means (NL-Means)	16
3.6.1	Principe	16
3.6.2	Algorithme NL-Means	17
3.6.3	Évolution de l'algorithme NL-means	17
3.7	Transformée en cosinus discrète (DCT)	19
3.7.1	Principe	19
3.7.2	Les variantes de la DCT	20
3.7.3	Algorithme de la DCT	21
CHAPITRE 4 : DÉBRUITAGE D'IMAGE PAR FUSION DE FILTRAGE SPATIO-FRÉQUENTIELLE		23
4.1	Introduction	23
4.2	Principe	23
4.3	Structure de la méthode	23
4.4	Implementation de la méthode	26
4.5	Conclusion	27
CHAPITRE 5 : EXPÉRIMENTATION ET RÉSULTATS		28
5.1	Implémentation	28
5.2	Résultats	30
5.2.1	Évaluation des résultats de simulation de la méthode proposée	30
5.2.2	Comparaison	52
CHAPITRE 6 : CONCLUSION		79
BIBLIOGRAPHIE		80

LISTE DES TABLEAUX

5.I	Résultats en PSNR, du débruitage des images de la figure 5.1 par la méthode proposée. $\sigma = 10$	31
5.II	Résultats en MSE, du débruitage des images de la figure 5.1 avec la méthode proposée, $\sigma = 10$	31
5.III	Résultats en PSNR, du débruitage des images de la figure 5.1 avec la méthode proposée, $\sigma = 20$	32
5.IV	Résultats en MSE, du débruitage des images de la figure 5.1 avec la méthode proposée, $\sigma = 20$	32
5.V	Résultats en PSNR, du débruitage des images de la figure 5.1 avec la méthode proposée, $\sigma = 30$	33
5.VI	Résultats en MSE du débruitage des images de la figure 5.1 avec la méthode proposée, $\sigma = 30$	33
5.VII	Résultats de débruitage en PSNR, des images (Lena, Boat, Barbara, Cameraman, Peppers et Baboon), avec la méthode proposée et les algorithmes (NLM [13] et DCT [78]). $\sigma = 10$	71
5.VIII	Résultats de débruitage en PSNR, des images (Lena, Boat, Barbara, Cameraman, Peppers et Baboon), avec la méthode proposée et les algorithmes (NLM [13] et DCT [78]). $\sigma = 20$	72
5.IX	Résultats de débruitage en PSNR, des images (Lena, Boat, Barbara, Cameraman, Peppers et Baboon), avec la méthode proposée et les algorithmes (NLM [13] et DCT [78]). $\sigma = 30$	73
5.X	Affichage de l'erreur quadratique moyenne (MSE) pour les expériences de débruitage des images (Lena, Boat, Barbara, Cameraman, Peppers et Baboon), avec la méthode proposée et les algorithmes (NLM [13] et DCT [78]). $\sigma = 10$	74

5.XI	Affichage de l'erreur quadratique moyenne (MSE) pour les expériences de débruitage des images (Lena, Boat, Barbara, Camera-man, Peppers et Baboon), avec la méthode proposée et les algorithmes (NLM [13] et DCT [78]). $\sigma = 20$	75
5.XII	Affichage de l'erreur quadratique moyenne (MSE) pour les expériences de débruitage des images (Lena, Boat, Barbara, Camera-man, Peppers et Baboon), avec la méthode proposée et les algorithmes (NLM [13] et DCT [78]). $\sigma = 30$	76

LISTE DES FIGURES

1.1	Exemple de débruitage d'image	1
2.1	Exemple d'une image de contours par application du filtre de Canny [15].	5
2.2	Exemple d'image dégradée par un bruit : (a) gaussien, (b) uniforme, (c) poivre et sel, (d) périodique.	8
2.3	Niveaux de traitements de la fusion d'images [62]	10
3.1	Principe de l'algorithme NL-Means [11].	16
3.2	Une base DCT 8 x 8 [78]	19
4.1	Première option de la méthode proposée où la détection de contours se fait sur l'image bruitée.	24
4.2	Deuxième option de la méthode proposée où la détection de contours se fait sur l'image filtrée par l'algorithme NLM.	25
5.1	Collection des images utilisées.	29
5.2	Simulation de débruitage sur l'image Lena par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 10$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	34
5.3	Simulation de débruitage sur l'image Boat par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 10$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	35

5.4	Simulation de débruitage sur l'image Barbara par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 10$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	36
5.5	Simulation de débruitage sur l'image Cameraman par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 10$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	37
5.6	Simulation de débruitage sur l'image Peppers par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 10$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	38
5.7	Simulation de débruitage sur l'image Baboon par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 10$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	39
5.8	Simulation de débruitage sur l'image Lena par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 20$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	40

5.9	Simulation de débruitage sur l'image Boat par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 20$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	41
5.10	Simulation de débruitage sur l'image Barbara par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 20$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	42
5.11	Simulation de débruitage sur l'image Cameraman par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 20$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	43
5.12	Simulation de débruitage sur l'image Peppers par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 20$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	44
5.13	Simulation de débruitage sur l'image Baboon par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 20$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	45

5.14	Simulation de débruitage sur l'image Lena par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 30$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	46
5.15	Simulation de débruitage sur l'image Boat par la méthode proposée :(a) Image Originale, (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 30$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée. . .	47
5.16	Simulation de débruitage sur l'image Barbara par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 30$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	48
5.17	Simulation de débruitage sur l'image Cameraman par la méthode proposée :(a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 30$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	49
5.18	Simulation de débruitage sur l'image Peppers par la méthode proposée :(a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 30$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	50

5.19	Simulation de débruitage sur l'image Baboon par la méthode proposée :(a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 30$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.	51
5.20	Résultat de l'expérience de débruitage sur l'image Lena par les filtres DCT, NLM et la méthode proposée. $\sigma = 10$	53
5.21	Résultat de l'expérience de débruitage sur l'image Boat par les filtres DCT, NLM et la méthode proposée. $\sigma = 10$	54
5.22	Résultat de l'expérience de débruitage sur l'image Barbara par les filtres DCT, NLM et la méthode proposée. $\sigma = 10$	55
5.23	Résultat de l'expérience de débruitage sur l'image Cameraman par les filtres DCT, NLM et la méthode proposée. $\sigma = 10$	56
5.24	Résultat de l'expérience de débruitage sur l'image Peppers par les filtres DCT, NLM et la méthode proposée. $\sigma = 10$	57
5.25	Résultat de l'expérience de débruitage sur l'image Baboon par les filtres DCT, NLM et la méthode proposée. $\sigma = 10$	58
5.26	Résultat de l'expérience de débruitage sur l'image Lena par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 20$	59
5.27	Résultat de l'expérience de débruitage sur l'image Boat par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 20$	60
5.28	Résultat de l'expérience de débruitage sur l'image Barbara par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 20$	61
5.29	Résultat de l'expérience de débruitage sur l'image Cameraman par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 20$	62
5.30	Résultat de l'expérience de débruitage sur l'image Peppers par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 20$	63
5.31	Résultat de l'expérience de débruitage sur l'image Baboon par les filtres DCT, NLM et la méthode proposée. $\sigma = 20$	64

5.32	Résultat de l'expérience de débruitage sur l'image Lena par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 30$	65
5.33	Résultat de l'expérience de débruitage sur l'image Boat par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 30$	66
5.34	Résultat de l'expérience de débruitage sur l'image Barbara par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 30$	67
5.35	Résultat de l'expérience de débruitage sur l'image Cameraman par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 30$	68
5.36	Résultat de l'expérience de débruitage sur l'image Peppers par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 30$	69
5.37	Résultat de l'expérience de débruitage sur l'image Baboon par les filtres DCT, NLM et la méthode proposée. $\sigma = 30$	70
5.38	Graphique de comparaison des résultats présentés dans la table 5.VII	71
5.39	Graphique de comparaison des résultats présentés dans la table 5.VIII	72
5.40	Graphique de comparaison des résultats présentés dans la table 5.IX	73
5.41	Graphique de comparaison des résultats présentés dans la table 5.X	74
5.42	Graphique de comparaison des résultats présentés dans la table 5.XI	75
5.43	Graphique de comparaison des résultats présentés dans la table 5.XII	76
5.44	Différence entre l'image Peppers originale et l'image Peppers filtrée ($\sigma = 30$) par : DCT à gauche, la méthode proposée au milieu et NLM à droite. Pour la méthode proposée, nous constatons qu'il y a moins de détails de l'image Peppers dans le bruit enlevé. . . .	77
5.45	Différence pour l'image Baboon entre l'image originale et l'image débruitée ($\sigma = 30$) par : DCT à gauche, la méthode proposée au milieu et NLM à droite. Pour la méthode proposée, nous constatons qu'il y a moins de détails de l'image Baboon dans le bruit enlevé.	77

5.46	Différence pour l'image Barbara entre l'image originale et l'image débruitée ($\sigma = 30$) par : DCT à gauche, la méthode proposée au milieu et NLM à droite. Pour cet exemple aussi, nous constatons qu'avec la méthode proposée, il y a moins de détails de l'image Barbara dans le bruit enlevé.	78
5.47	Différence pour l'image Lena entre l'originale et l'image débruitée ($\sigma = 30$) par : DCT à gauche, la méthode proposée au milieu et NLM à droite. Les détails de l'image Lena sont moins visibles dans le bruit enlevé avec la méthode proposée.	78

LISTE DES SIGLES

DCT	Discrete Cosine Transform
LSD	Line Segment Detector
MSE	Mean Square Error
NLM	Non Local Means
PSNR	Peak Signal to Noise Ratio

CHAPITRE 1

INTRODUCTION

De nos jours, les images numériques sont de plus en plus utilisées car ce type de média est devenu facile à transmettre, traiter et stocker sur des supports informatiques.

En traitement d'images, il est souvent indispensable d'enlever le bruit de l'image vu que le bruit dans les images numériques est une anomalie parasite qui dégrade la qualité de l'image, tout en rendant difficile l'interprétation visuelle de l'image.

La réduction du bruit dans les images est un problème délicat et largement traité par les scientifiques. La problématique primordiale, est de réussir à éliminer un maximum de bruit, tout en préservant les structures et les détails de l'image. En général, l'image observée est contaminée par un bruit qui est soit additif ou multiplicatif. Les traitements de restauration sont généralement nécessaires pour améliorer la qualité des images observées [5, 76].

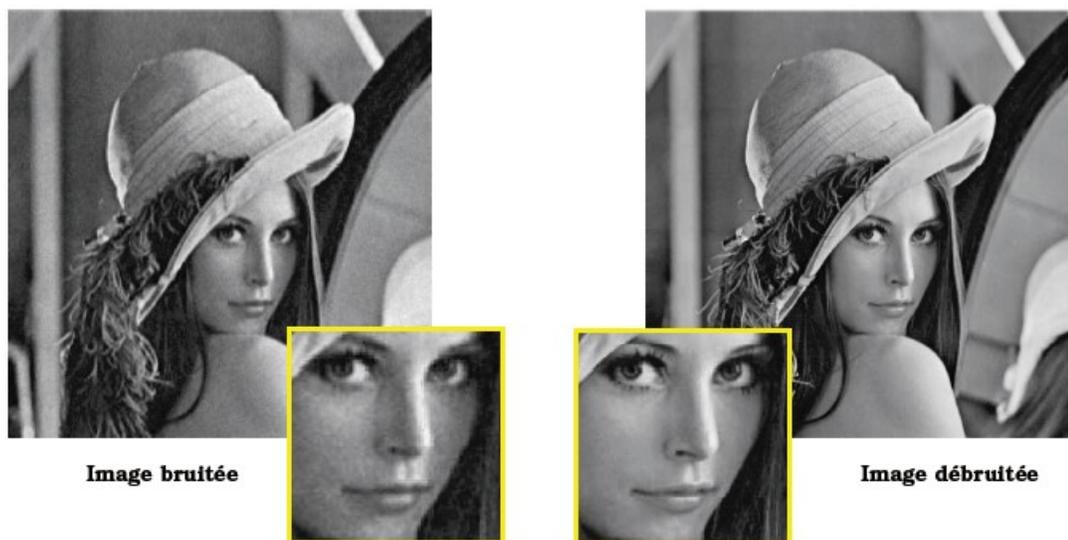


Figure 1.1 : Exemple de débruitage d'image

Dans la littérature du traitement d'image, plusieurs méthodes de débruitage d'image ont été proposées et développées. Chaque approche ayant ses propres hypothèses, ses

avantages, et ses limites. Toutefois, beaucoup de ces méthodes enlèvent en plus du bruit des détails de l'image. De ce fait, des algorithmes de débruitage continuent à être développés pour remédier à ce problème. La figure 1.1 représente un exemple de débruitage d'image.

Dans un premier temps, des méthodes basées sur un filtrage fréquentiel passe-bas ont été proposées pour réduire le bruit additif gaussien qui entache une image numérique. Ces techniques de filtrage réduisent très bien le bruit dans les zones homogènes de l'image mais ont aussi le désavantage majeur d'atténuer aussi fortement les hautes fréquences de l'image, *i.e.*, les contours, les détails et la texture de l'image [5].

Des améliorations majeures ont ensuite été apportées par des techniques locales de filtrage (fréquentiel) par ondelette ou transformé en cosinus discret (TCD) par seuillage doux (*soft thresholding*) et effectué localement pour chaque imagette ou petite partie de l'image à débruiter [55, 78]. Ces améliorations sont basées sur le fait que le signal image est intrinsèquement clairsemé (*sparse*) *i.e.*, que l'on peut décomposer et reconstruire l'image ou chaque partie de l'image en un nombre restreint de fonctions ou d'imagettes élémentaires. Mais dans ce cas encore, les erreurs de débruitage restent majoritairement localisées dans les zones de variations rapides de l'image.

Une deuxième famille de méthode de débruitage, par filtrage uniquement spatial est apparue en 2005 grâce au travail séminal de Buades [11] qui fût le premier à exploiter dans son algorithme de débruitage le fait qu'une image naturelle possède intrinsèquement des autosimilarités multi-échelles. Cette hypothèse qui modélise bien les propriétés d'une image naturelle fut aussi exploitée auparavant, avec succès, dans la modélisation fractale de Mandelbrot avec la petite différence que cette autosimilarité était exprimée à toutes les échelles du signal ou de l'image et ensuite pour la compression fractale de Barnsley à deux niveaux de résolution (ou échelle) successif. Cependant dans ce cas, les erreurs de débruitage sont peut être beaucoup moins présentes dans les zones hautes fréquences de l'image mais sont plutôt localisées dans les zones spatialement homogènes de l'image.

Le but principal de ce mémoire est de proposer une approche simple, reproductible,

permettant de combiner efficacement et rapidement les deux approches pour débruiter à la fois les zones homogènes et les zones hautes fréquences (contour, détail, micro-texture) composant une image naturelle, lorsque celle-ci est dégradée par un bruit additif gaussien de différentes variances. La méthode de débruitage d'image proposée dans ce mémoire sera donc obtenue par fusion de ces deux techniques différentes de filtrage et permettra, *in fine*, d'obtenir une image résultante débruitée qui conservera le meilleur des performances de chacune de ces deux techniques de filtrage opérant dans deux espaces différents (spatial et fréquentiel) complémentaires.

Dans le chapitre suivant, nous allons parler des caractéristiques et de la dégradation de l'image numérique, du débruitage des images, de la fusion d'images et de ses critères d'analyse. Ensuite le troisième chapitre introduit l'étude bibliographique sur le débruitage d'images et un aperçu des algorithmes NL-means et DCT. Le quatrième chapitre donne une description de la méthode proposée. Le cinquième chapitre présente l'expérimentation et les résultats et enfin une conclusion pour clôturer ce travail.

CHAPITRE 2

GÉNÉRALITÉ

Nous allons présenter dans ce chapitre, quelques concepts attachés au domaine du débruitage d'images, tout en donnant un aperçu sur : les caractéristiques d'une image, quelques types de bruits dégradant les images et le débruitage des images. Nous terminons ce chapitre par la présentation non exhaustive des critères d'analyse généralement utilisés en débruitage d'images.

2.1 Caractéristiques d'une image

Une image numérique est constituée d'un ensemble de points appelés pixels (abréviation de Picture Element). Le pixel représente ainsi le plus petit élément constitutif d'une image numérique. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image [57].

Lorsque l'image est en niveaux de gris, la représentation la plus classique se fait en associant à chaque pixel un niveau de gris codé sous 8 bits : les valeurs possibles pour l'intensité des pixels de l'image sont ainsi les entiers compris entre 0 et 255 [66].

L'image numérique contient des détails de hautes fréquences qui se traduisent par une ou des variations spatiales rapides des niveaux de gris et qui correspondent en fait aux détails et contours existant dans l'image lesquels séparent les variations de plus basses fréquences (variations spatiales lente des niveaux de gris) qui constituent les zones homogènes de l'image.

Une façon de détecter les détails hautes fréquences d'une image, et les localiser précisément dans l'image, est par application directe du filtre de Canny (figure 2.1).



Figure 2.1 : Exemple d'une image de contours par application du filtre de Canny [15].

2.2 Dégradation de l'image

Une variété de facteurs affecte les performances des capteurs en imagerie numérique et se traduit par du bruit (que l'on qualifie d'électronique) sur l'image acquise numériquement.

La première peut être produite par les composantes électroniques du capteur qui génèrent un signal parasite additif gaussien (généralement non corrélé) qui vient se superposer au signal utile.

La seconde peut être due aux conditions environnementales durant l'acquisition comme un éclairage insuffisant lors de la prise de vues et la limite en faible lumière du capteur. Une seconde source de bruit peut être dû à une possible température excessive de l'environnement et/ou du capteur; le bruit électronique augmentant linéairement avec la température (dans les studios professionnels, ce bruit peut être réduit efficacement par refroidissement du capteur). [34].

Les images peuvent être 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 [34].

La qualité de l'image numérique peut être dégradée par plusieurs types de bruits survenus lors de l'acquisition et de la transmission. Ainsi il est important de connaître la nature du bruit qui contamine l'image.

2.2.1 Bruit Gaussien

C'est le type de dégradation produite par les composantes électroniques du capteur (relation linéaire avec la température du capteur) et liée à la limite en faible lumière de celui-ci. C'est donc le bruit qui entache majoritairement les images numériques et que nous considérerons plus tard.

Pour créer synthétiquement ce bruit additif gaussien non corrélé (blanc), une variable aléatoire gaussienne a été ajoutée par le système à l'image "idéale". Dans le cas précédent des photos numériques, la source est la précision du capteur CCD ou CMOS mise en évidence par un gain élevé (figure 2.2 (a)).

2.2.2 Bruit Uniforme

Une variable aléatoire uniforme (figure 2.2 (b)) a été ajoutée par le système à l'image "idéale". C'est généralement un cas d'école non réaliste mais simple. Visuellement, il est proche du bruit gaussien (seule la distribution diffère).

2.2.3 Bruit "Poivre et Sel"

Ce bruit peut être créé par la saturation de plusieurs capteurs. Certains pixels sont saturés. Visuellement, cela se traduit par des points blancs et noirs répartis avec une certaine densité dans l'image (figure 2.2 (c)). La source la plus fréquente est un pixel mort dans le capteur dû au processus de fabrication des CCD (que l'on retrouve dans celle des moniteurs LCD). Les appareils photos les "cachent" en les détectant pour chaque

appareil avant commercialisation et en utilisant un mécanisme de remplacement par la valeur d'un pixel voisin lors de la prise de photos.

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 [38].

2.2.4 Bruit Périodique

Un bruit de période (figure 2.2 (d)) fixe a été ajouté par le système à l'image "idéale". La source est par exemple le tramage dû à un scannage papier, ou à une compression JPEG (blocs de 8 pixels) ou une interférence haute fréquence sur une image produite par un moniteur utilisant un canon à électron.



(a)



(b)



(c)



(d)

Figure 2.2 : Exemple d'image dégradée par un bruit : (a) gaussien, (b) uniforme, (c) poivre et sel, (d) périodique.

2.3 Débruitage des images

Le débruitage est un vaste et délicat sujet dans le traitement des images. Il existe de nombreuses approches de débruitage d'images provenant de différentes théories mathématiques. Dans [66], le but d'une procédure de débruitage est défini comme le fait de créer une image \hat{I} , qui pour chaque pixel $x \in \Omega$ donne un estimateur $\hat{I}(x)$ de l'intensité de l'image originale $I(x)$.

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 essaye 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 [6].

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. Dans le chapitre suivant, nous allons décrire quelques types de méthodes standard de débruitage d'images qui sont appliquées dans le domaine uniquement spatial ou dans le domaine uniquement fréquentiel.

2.4 Fusion d'images

Dans l'article [62], la fusion d'images est défini comme étant la combinaison de deux ou de plusieurs images différentes pour former à l'aide d'un algorithme une nouvelle image. La combinaison peut se faire à 3 niveaux : soit au niveau des pixels (fusion des pixels), soit au niveau des caractéristiques après une segmentation (fusion des objets extraits de l'image), soit au niveau décisionnel lors de la phase finale de la segmentation (fusion des objets extraits et identifiés) [62].

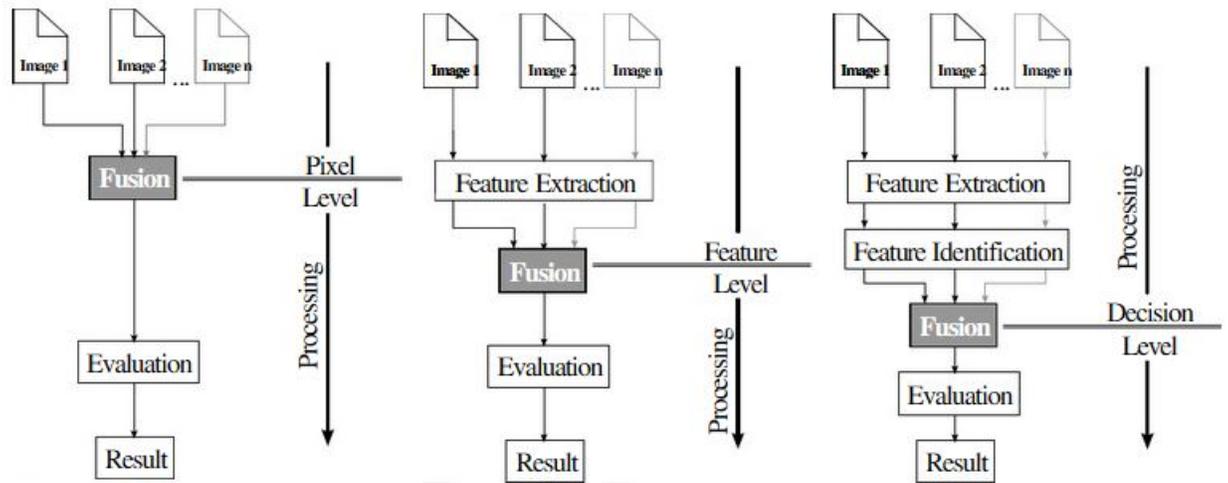


Figure 2.3 : Niveaux de traitements de la fusion d'images [62]

2.5 Critère d'analyse du débruitage d'images

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

2.5.1 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 subjective puisqu'il n'existe aucune mesure correcte pouvant traduire fidèlement la perception de l'œil humain [34].

2.5.2 Critère objectif

Dans la littérature, l'erreur quadratique moyenne (MSE) et le rapport signal sur bruit (PSNR) sont les plus utilisés.

2.5.2.1 Erreur quadratique moyenne (MSE)

Bien qu'elle ne corresponde pas toujours avec la perception humaine, la MSE est souvent considérée comme une bonne mesure de l'exactitude d'une évaluation de l'image. Ce critère est défini comme suit [34] :

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (X(i, j) - Y(i, j))^2 \quad (2.1)$$

avec X : Image originale.
Y : Image débruitée.
M : Nombre de lignes de l'image.
N : Nombre de colonnes de l'image.
(i, j) : positionnement des pixels.

2.5.2.2 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

$$PSNR = 10 \log_{10} \left[\frac{255^2}{MSE} \right] \text{ dB} \quad (2.2)$$

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

CHAPITRE 3

DÉBRUITAGE D'IMAGE : ÉTAT DE L'ART

Dans ce chapitre, nous allons donner un bref aperçu des approches de débruitage d'image, tout en nous concentrant sur celles de NL-means et DCT.

3.1 Méthodes de débruitage par moyennes

Ces méthodes concernent l'ensemble des méthodes qui utilisent une moyenne de pixels pour traiter chaque pixel d'intérêt. Le traitement se fait dans ce type d'approches dans le domaine des pixels voisins.

Rosenblatt [64] en 1956 et Parzen [60] en 1962, ont introduit les méthodes à noyau pour l'estimation de densité. En 1964, Nadaraya [56] et Watson [75] ont étendu la méthode à noyau pour la régression à effet aléatoire. Ensuite Priestley et Chao [63] en 1972 puis par Benedetti [4] 1977 pour la régression à effet déterministe qui est fondé sur des moyennes locales, dans le sens où les pixels utilisés sont limités à ceux proches spatialement du pixel d'intérêt. Le filtre sigma a été proposé par Lee en 1983 [46], puis par Yaroslavsky en 1985 [77]. Ce Filtre consiste à faire une moyenne tronquée des pixels proches, il repose sur la ressemblance entre la valeur des pixels. En 1998, Tomasi et Manduchi [71], ont introduit le filtre Bilatère pour associer les méthodes à noyau avec l'approche de Yaroslavsky [77]. Cette méthode est fondée sur la moyenne locale des pixels, où les poids sont le produit de deux termes (l'un correspond à l'écart spatial entre deux pixels, l'autre correspond à l'écart photométrique). Perona et Malik [61] ont proposé en 1990, une méthode de diffusion anisotrope basée sur les équations aux dérivées partielles (EDP), dont l'idée est de débruiter l'image de manière à préserver les contours. La diffusion anisotrope consiste à régulariser l'image par convolution, selon des directions opposées au gradient. Dans l'article [16], F. Catté, P.L. Lions, J.M. Morel et T. Coll démontrent le caractère mal posé de la diffusion anisotrope de Perona et Malik [61] et proposent une version régularisée tout en donnant des résultats de stabilité en pré-

sence de bruit ainsi que des preuves d'existence et d'unicité de la solution. Le débruitage d'images par approche variationnelle TV (Total Variation), a été introduit par Rudin *et al.* [65] en 1992, l'idée est de minimiser un terme d'attache aux données de type quadratique avec une contrainte sur le gradient de la fonction. En 2005, Tibshirani, *et al.* [70] ont introduit une méthode qui ressemble à l'approche variationnelle noté fused LASSO ou F-LASSO.

3.2 Méthodes utilisant des dictionnaires de patches

L'utilisation des dictionnaires de patches consiste à supposer qu'une image est tout simplement ce qui ressemble le plus à une banque d'images que l'on s'est fixée initialement. Cette approche tire parti de deux avancées récentes, l'une en statistique en grande dimension et l'autre en traitement des images. L'idée est de combiner les gains des méthodes à patches telles que les NL-Means, avec des méthodes performantes d'apprentissage dans des dictionnaires redondants [66]. L'apprentissage de dictionnaire a été introduit par B. A. Olshausen et D. J. Field dans les articles [58] et [59]. La méthode BM3D (Block Matching 3D) introduite par Dabov, Foi, Katkovnik, et Egiazarian [21] en 2007, puis étendue par les mêmes auteurs dans [22] en 2008 et dans l'article [23] en 2009, cette méthode qui mélange diverses techniques, est basée sur un traitement par blocs et donne dans l'état de l'art du débruitage d'image, les meilleurs résultats numériques. En 2010, Yu et al. proposent dans [79], une méthode d'apprentissage de dictionnaire basée sur une procédure de type [sélection de modèle] pour trouver quelle base s'adapte le mieux à chaque patch. Ils proposent de construire le dictionnaire de manière synthétique. En 2011, M. Lebrun, A. Buades et J.M. Morel présentent dans l'article [45] une évolution de l'algorithme BM3D, qui consiste à appliquer une Analyse en Composantes Principales au bloc 3D de façon à obtenir une base mieux adaptée que celle des cosinus discrets. La même année, C.V. Angelino, E. Debreuve et M. Barlaud observent qu'il y a, dans l'estimation obtenue par l'algorithme BM3D, des effets de marche d'escalier dans les zones où le niveau de gris est homogène [2].

3.3 Méthodes de débruitage par les moyennes Non-Local (NL-Means)

Cette méthode est proche du débruitage par apprentissage de dictionnaire. La différence majeure étant que l'image elle-même fournit en quelque sorte le dictionnaire. NL-Means est une méthode à patches qui est basée sur l'auto-similarité de l'image. Elle a été introduite par Buades, Coll, et Morel en 2005 [12]. Kervrann et Boulanger, Awate et Whitaker ont proposé des méthodes similaires en 2006 [41]. Cette méthode est plus fréquemment désignée par NL-Means. L'approche itérative des NL-Means a été proposée par Brox et Cremers en 2007 [8], puis détaillée par Brox, Kleinschmidt, et Cremers en 2008 [9]. En 2009, Tschumperlé et Brun ont essayé de lier les méthodes de débruitage par patches et les méthodes de diffusion anisotrope [72]. Louchet et Moisan ont fait l'étude sur l'association entre la variation totale, les modèles bayésiens et les approches par patches en 2011 [49].

3.4 Méthodes de débruitage par la transformée en ondelettes

En 1909, Alfred Haar définit une fonction composée d'une courte impulsion négative suivie d'une courte d'impulsion positive, connue pour être la première ondelette (Ondelette de Haar) [37]. Le terme d'ondelette fut introduit dans le langage mathématique en 1984 par Jean Morlet et Alex Grossmann [36]. Ingrid Daubechies a mis au point en 1987 [25] des ondelettes orthogonales appelées ondelettes de Daubechies, facilement implémentables, et utilisées dans le standard JPEG 2000, et construit les ondelettes à support compact en 1988 [26]. En 1989, Stéphane Mallat a fait le lien entre les ondelettes et l'analyse multi résolution [53]. Yves Meyer présente la théorie des ondelettes simplifiée par l'utilisation systématique des bases d'ondelettes en 1990 [54]. La méthode *wavelet shrinkage* a été introduite par David Donoho en 1992 [32]. D. L. Donoho et I. M. Johnstone ont introduit le seuillage des coefficients de la transformée en ondelettes en 1994 [33]. E. Candès et D. Donoho ont menés des travaux pour construire des bases qui s'adaptent à la régularité existant le long des contours de l'image en 1999 [14]. Les bandelettes, ont été introduites par E. Le Pennec et S. Mallat en 2005 [44]. Ensuite, E. Le Pennec *et al.* parcourent l'application des bandelettes au débruitage en 2007 [43].

3.5 Méthodes hybrides de débruitage

G. Gilboa et S. Osher dans l'article [35], proposent une méthode de débruitage par minimisation de la variation totale non-locale en 2008, ensuite reprise par Lou *et al.* [48], pour des applications de déconvolution et de reconstruction en tomographie, est une approche mixte entre la minimisation de la variation totale et l'algorithme N-L means. Cet algorithme consiste à minimiser une fonctionnelle d'énergie. En 2009, J. Mairal *et al.* proposent dans l'article [51] l'algorithme NLSM (Non local Sparse Model), qui est une méthode mixte entre les algorithmes BM3D et K-SVD. Son principe est de fixer un patch et on cherche les patchs qui lui sont similaires, ensuite chercher dans le dictionnaire courant l'écriture la plus parcimonieuse. En 2011, C. Louchet et L. Moisan dans l'article [49], proposent de combiner l'approche par minimisation de la variation totale et l'algorithme N-L means, le but de cette méthode est de corriger l'effet des patchs rares dans l'algorithme N-L means. P. Chatterjee et P. Milanfar présentent dans les articles [18, 19] respectivement en 2011 et 2012, l'algorithme PLOW (Patch-based Locally Optimal Wiener filtering), basé sur une segmentation de l'image. Le principe est de former dans l'image un certain nombre de clusters à l'aide de l'algorithme K-means et d'appliquer l'estimateur LMMSE (en anglais Linear Minimum Mean Square Error Estimator, décrit dans le livre de S. M. Kay [67]) à chaque patch après l'avoir associé à l'un des clusters. Dans l'article [78], G. Yu et G. Sapiro présentent l'algorithme de débruitage DCT means, basé sur deux techniques : le seuillage dans une base orthonormée et le débruitage invariant par translation introduit par R. Coifman et D. Donoho dans l'article [20]. Le débruitage invariant par translation est d'effectuer la translation de l'image, pour une gamme de translations, puis de débruiter par seuillage dans une base d'ondelettes toute la gamme tradatée de l'image et enfin faire la moyenne des différentes estimations obtenues en un pixel donné. A. Buades, M. Lebrun et J.M. Morel ont présenté dans [10] une méthode de débruitage sous le nom de NL-Bayes, basée sur une approche bayésienne en deux étapes. Ces deux étapes de filtrage ont la même structure que celles de l'algorithme BM3D mais le débruitage se fait par une estimation MAP (Maximum A Posteriori).

3.6 Non-Local Means (NL-Means)

3.6.1 Principe

Comme montré par Buades et al. dans [11], le principe des moyennes non-locales (NL Means), consiste à tirer profit d'une redondance de l'information à longue distance que l'on peut trouver dans les images. Cette approche est basée sur l'auto-similarité existant dans l'image elle même, alors il s'agit de trouver les pixels (patches) similaires dans l'image, ensuite calculer leur moyenne pondérée en fonction de leur similarité avec le pixel à débruiter.(figure 3.1).

Soit $S(x)$ l'ensemble des pixels semblables à x ($x \in \Omega$). Ainsi :

$$u(x) = \sum_{y \in \Omega_x} w(x,y)v(y) \quad (3.1)$$

où $u(x)$ est la valeur débruitée d'un pixel x .

$v(y)$ la valeur à débruiter au point y ($y \in \Omega_x$).

$w(x,y)$ les poids.

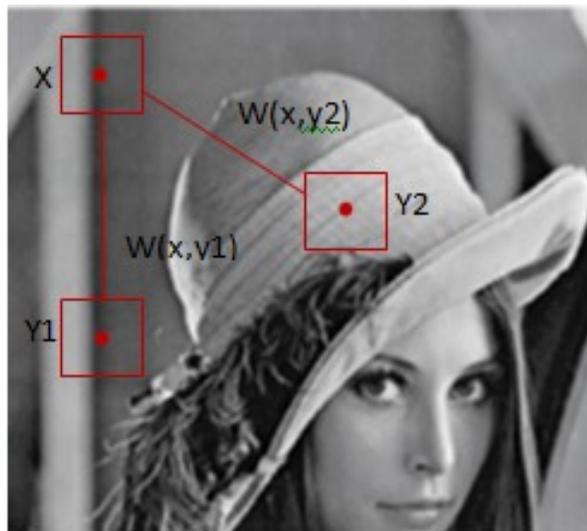


Figure 3.1 : Principe de l'algorithme NL-Means [11].

3.6.2 Algorithme NL-Means

L'algorithme NLM présenté dans [11], est un algorithme de débruitage par moyenne pondérée, dont le noyau qui forme les poids est de type Nadaraya [56] - Watson [75] évalué sur la distance entre des voisinages de ces pixels (patches). Les poids $w(x,y)$ étant normalisés, nous obtenons :

$$\sum_{y \in \Omega_x} w(x,y) = 1$$

Dans l'algorithme N-L means, l'image débruitée est estimée selon :

$$NLM(x) = \sum_{y \in \Omega_x} \frac{1}{Z(x)} e^{-\frac{\|P(x)-P(y)\|_{2,\alpha}^2}{2h^2}} v(y) \quad (3.2)$$

avec Z : le facteur de normalisation,

$P(x)$: le path au pixel x fixé,

$P(y)$: les patches similaires à $P(x)$,

h : le paramètre de contrôle du degré de lissage.

3.6.3 Évolution de l'algorithme NL-means

Depuis l'introduction de l'algorithme NL-Means, plusieurs travaux ont été réalisés pour l'améliorer. Nous répertorions dans cette section, quelques approches. En 2005 M. Mahmoudi et G. Sapiro ont proposé dans [50], de faire une sélection primitive des patches en les attribuant un poids non nul. Ce qui apporte une amélioration en termes de qualité de débruitage. Pour réduire l'effet du bruit sur le calcul de la distance entre les patches, Azzabou, *et al.* proposent dans [3], un nouveau critère de similarité plus robuste au bruit, calculée à partir d'un dictionnaire adapté au contenu de l'image. Une projection de l'ensemble des patches de l'image à ce sous-espace est utilisée pour définir une métrique entre un pixel et les voisins. Dans l'article [27], A. Dauwe *et al.* proposent une classification préliminaire des patches selon leurs moments d'ordre pour accélérer l'algorithme. Cette méthode donne un meilleur débruitage des textures. Les SSI introduit par Wang *et al.*

[74] puis par Darbon *et al.* [24] en 2008, qui permettent de calculer la fonction sur tous les sous carrés d'une certaine taille, qui donne la distance au carré entre deux patches. Dans les articles [41] et [42], Kervrann et Boulanger proposent une approche basée sur la sélection ponctuelle de petits patches d'image de taille fixe dans un voisinage adaptatif. S. Zimmer, S. Didas, et J. Weickert [80] en 2008, [151] ont mené des travaux pour ajouter des rotations de patches à fin d'augmenter la similarité. Dans le même contexte, en 2009 les auteurs Sun *et al.* [68], Ji *et al.* [39], Alexander et Vrscay [1], Lou *et al.* [47] ont montré que les techniques d'invariance peuvent augmenter le nombre de patches similaires. T. Tasdizen en 2009, poursuit le même objectif que [3], en proposant dans son article [69], de calculer la distance entre les patches après projection des patches sur les premières composantes principales de l'ensemble des patches de la zone de recherche. Dans les articles [28] et [29] publiés en 2009, Deledalle *et al.* proposent une méthode pour généraliser l'algorithme NLM de Buades *et al.* [11] aux bruits non gaussiens et/ou non-additifs. Ils définissent la similarité entre patches à partir du modèle paramétrique du bruit considéré. Cette approche débruite l'image par maximisation de la vraisemblance pondérée. Les poids sont affinés à chaque itération à partir des patches débruités à l'itération précédente. En 2011, Maleki, Narayan, et Baraniuk dans l'article [52], ont effectué une analyse asymptotique du risque de l'algorithme NL-Means pour des images constantes de classe Horizon par morceaux avec une discontinuité du contour. Ils prouvent que le risque des NLM pour cette classe d'images, est plus élevé que le risque de seuillage d'ondelettes. La même année, Jin, Grama et Liu proposent dans [40] un algorithme de débruitage d'image pour traiter le bruit additif blanc gaussien, qui consiste à choisir les poids en minimisant une borne supérieure serrée de l'erreur quadratique moyenne. L'algorithme proposé est sans paramètre dans le sens où il calcule automatiquement la bande passante du noyau de lissage. En 2012 Deledalle, Duval et Salmon ont proposé dans l'article [31], une approche dont le concept est de remplacer les patches carrés habituels par des patches adaptatifs, de façon à profiter de la géométrie locale de l'image. Un nouveau critère basé sur le rapport de vraisemblance généralisé, est proposé par Deledalle, Denis et Tupin dans l'article [30]. Ce critère donne de bons résultats pour le débruitage par NL-Means des images contaminées par un bruit Gamma ou Poisson.

3.7 Transformée en cosinus discrète (DCT)

3.7.1 Principe

La DCT étant la projection de l'image sur une base de fonctions cosinus, elle a pour principe, de décomposer l'image d'entrée en blocs de taille $N \times N$, ensuite appliquer la DCT sur les pixels de chaque bloc.

La figure 3.2 représente un exemple de base DCT 8×8 .

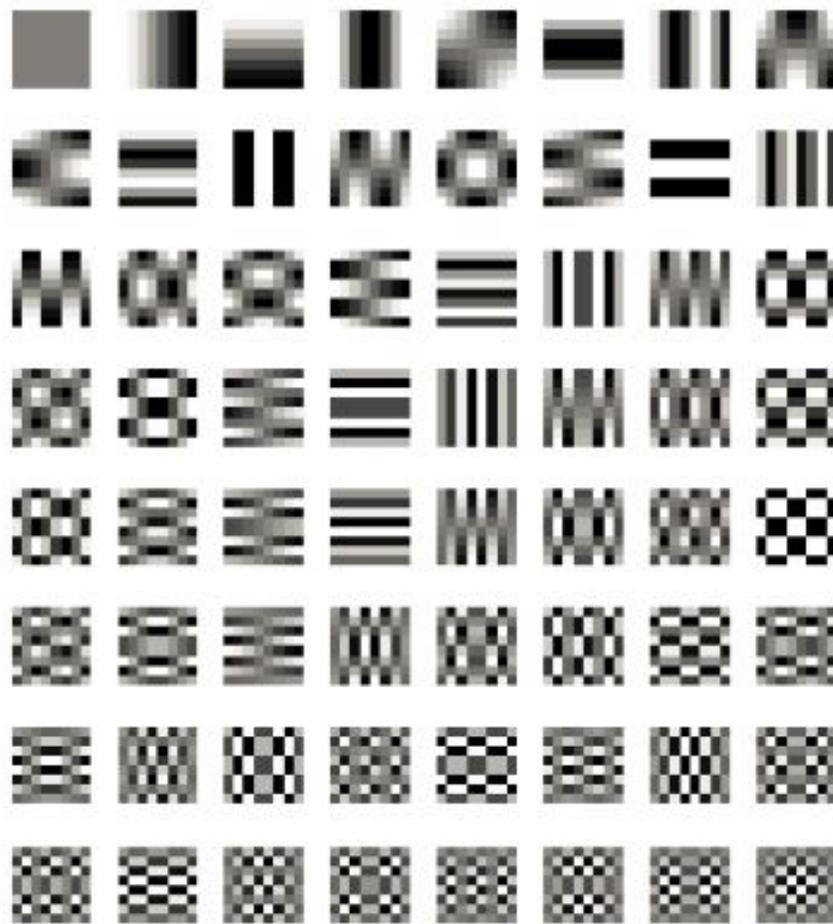


Figure 3.2 : Une base DCT 8×8 [78]

3.7.2 Les variantes de la DCT

Nous allons présenter dans cette section, les quatre variantes de la DCT les plus utilisées dont : DCT-I, DCT-II, DCT-III et DCT-IV.

La plus courante entre ces quatre est la DCT-II et son inverse DCT-III aussi appelée respectivement DCT et IDCT [7].

3.7.2.1 DCT-I

$$X_k = \frac{1}{2}(x_0 + (-1)^k x_{N-1}) + \sum_{n=1}^{N-2} x_n \cos\left[\frac{\pi}{N-1} nk\right] \quad (3.3)$$

La multiplication de x_0 et x_{N-1} par $\sqrt{2}$, ensuite X_0 et X_{N-1} par $\frac{1}{\sqrt{2}}$, la transformée devient orthonormée.

La DCT-I est définie pour $N > 2$, contrairement aux autres variantes qui sont définie pour toutes les valeurs N positives.

3.7.2.2 DCT-II

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \quad (3.4)$$

Cette variante de DCT est la plus utilisée et pour la rendre orthonormée, il faut multiplier X_0 par $\frac{1}{\sqrt{2}}$.

3.7.2.3 DCT-III

$$X_k = \frac{1}{2}\left(x_0 + \sum_{n=1}^{N-1} x_n \cos\left[\frac{\pi}{N}n\left(k + \frac{1}{2}\right)\right]\right) \quad (3.5)$$

La DCT-III est la transformée inverse de la DCT-II. La multiplication de x_0 par $\sqrt{2}$, rend la transformée orthonormée.

3.7.2.4 DCT-IV

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)\left(k + \frac{1}{2}\right)\right] \quad (3.6)$$

La DCT-IV est une matrice orthonormée.

3.7.3 Algorithme de la DCT

L'avantage de La DCT est sa capacité de concentrer l'énergie du signal après le changement de la base, dans les premiers coefficients transformés donc dans les basses fréquences. Ce qui permet l'élimination du reste des coefficients de haute fréquence.

3.7.3.1 Le seuillage

Soit un signal f contaminé par un bruit w , souvent modélisé comme un gaussien indépendant de moyenne nulle :

$$y = f + w \quad (3.7)$$

où y est le signal bruité observé.

Le débruitage du signal vise à estimer f à partir de y . Les vecteurs ϕ_n d'une base orthonormée $B = \{\phi_n\}_{1 \leq n \leq N}$, doivent satisfaire à la condition suivante :

$$\langle \phi_m, \phi_n \rangle = \begin{cases} 1, & \text{si } n=m \\ 0, & \text{autrement.} \end{cases} \quad (3.8)$$

Un estimateur de seuillage projette le signal bruité à la base, et reconstitue le signal débruité avec des coefficients de transformation plus grandes que le seuil T :

$$T(y) = \tilde{f} = \sum_{n=1}^N \rho \tau(\langle y, \phi_n \rangle) \phi_n, \quad (3.9)$$

où

$$\rho \tau(x) = \begin{cases} x, & \text{si } |x| > T \\ 0, & \text{sinon} \end{cases} \quad (3.10)$$

est un opérateur de seuillage.

3.7.3.2 Le débruitage

La base de la transformée en cosinus discrète locale (DCT) la plus utilisée est la base DCT 8 x 8 (figure 3.2).

L'algorithme de débruitage proposé dans [55, 78], décompose l'image en patches locaux de taille par exemple $\sqrt{N} = 16 \times 16$, et débruite les patches avec estimation par seuillage dans le domaine DCT. La fenêtre DCT 16 x 16 conduit en moyenne, aux meilleurs résultats de débruitage.

CHAPITRE 4

DÉBRUITAGE D'IMAGE PAR FUSION DE FILTRAGE SPATIO-FRÉQUENTIELLE

4.1 Introduction

Dans la littérature, nous avons constaté que les moyennes non-locales (NL Means) [11] débruite bien les zones de contours (fréquences spatiales élevées de l'image) et la transformée en cosinus discrète (DCT) [55, 78], quant à elle, débruite mieux les zones homogènes (fréquences spatiales basses), d'où il nous a paru intéressant de combiner ces deux filtres différents pour générer une image offrant à la fois une bonne restauration dans ces deux types de bande fréquentielle ; l'image résultante gardera ainsi les zones homogènes idéalement filtrée par l'algorithme de DCT tout en intégrant les détails spatiaux de l'image originale qui sont idéalement filtrés par l'algorithme de Buades. Cette fusion vise à obtenir une image qui comporte moins de bruit et proche de l'image originale.

4.2 Principe

Le principe de notre technique, est de combiner les niveaux de gris de l'image débruitée dans le domaine fréquentiel (DCT) avec les niveaux de gris de l'image débruitée dans le domaine spatial (NL-Means) et classés comme contours par un filtre de détection de contours.

4.3 Structure de la méthode

Les schemas 4.1, 4.2 illustrent les deux options pour la détection des contours de la structure générale de la méthode proposée.

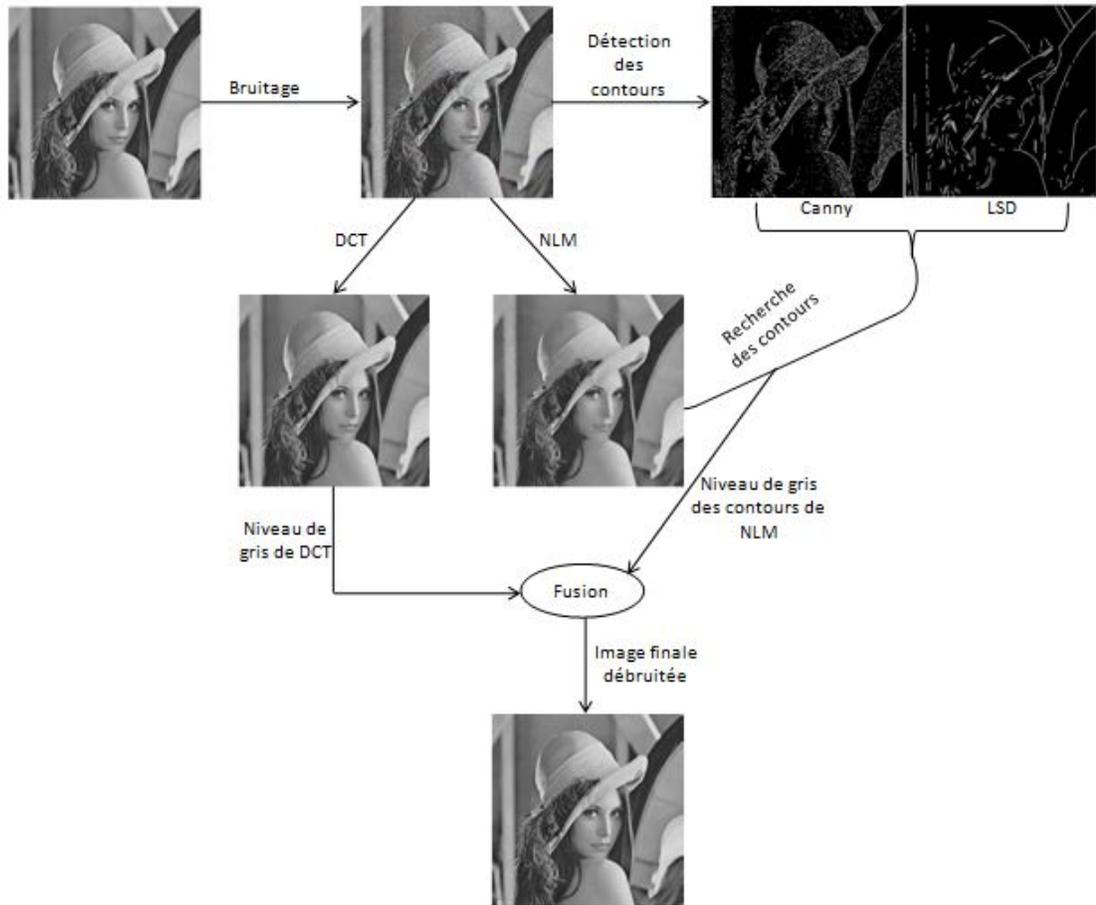


Figure 4.1 : Première option de la méthode proposée où la détection de contours se fait sur l'image bruitée.

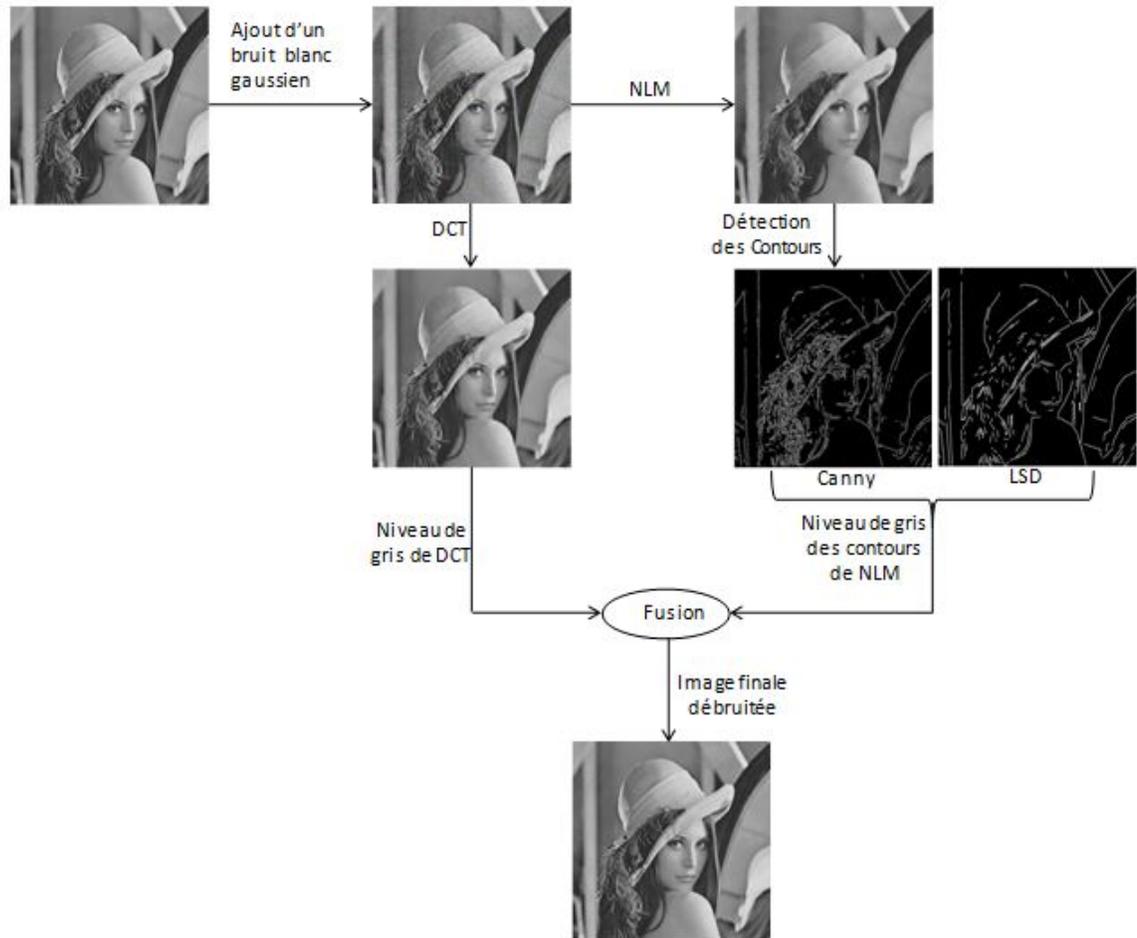


Figure 4.2 : Deuxième option de la méthode proposée où la détection de contours se fait sur l'image filtrée par l'algorithme NLM.

4.4 Implementation de la méthode

Les étapes suivantes décrivent l'algorithme de notre méthode pour les deux options :

1. Chargement d'une image de bonne qualité : *Image* ;
2. Simulation de la dégradation de cette image par ajout d'un bruit blanc gaussien : *ImgDegrad* ;
3. Une première estimation par débruitage de l'image dégradée dans le domaine spatial avec NLM : *ImgNLM* ;
4. Une deuxième estimation par débruitage de l'image dégradée dans le domaine fréquentiel avec DCT : *ImgDCT* ;
5. Détection de contours par le filtre de Canny [15] ou LSD [73] :
 - de *ImgDegrad* : *DegradSeg1* pour Canny et *DegradSeg2* pour LSD ; "Option 1"
 - de *ImgNLM* : *NLMSeg1* pour Canny et *NLMSeg2* pour LSD ; "Option 2"
6. Recherche des niveaux de gris spatiaux pour la fusion : Pour tout pixel P de *ImgNLM* non affecté à un contour C,
 - vérifier qu'il correspond à un contour C de (*DegradSeg1* ou *DegradSeg2* ; "Option 1"
 - vérifier qu'il correspond à un contour C de (*NLMSeg1* ou *NLMSeg2* ; "Option 2"
7. Si P correspond à C, mettre P dans *NivgrisNLM* pour la fusion ;
8. Sinon repartir sur 6 ;
9. Fusion de *NivgrisNLM* avec les niveaux de gris de *ImgDCT* : *ImgFusionee* ;
10. Sauvegarde de l'image finale débruitée *ImgFusionee* ;

11. Affichage de l'image finale débruitée *ImgFusionnee* ;

12. Fin.

4.5 Conclusion

Dans ce chapitre, nous avons décrit le processus de notre méthode de débruitage d'image, basé sur la fusion de filtrage spatio-fréquentielle. Au cours de ce processus, nous avons eu recours à un prétraitement dont le débruitage de l'image dégradée par les deux algorithmes de bases (NLM et DCT) et la détection de contours en utilisant un masque binaire de 0 et 1 des pixels pré-sélectionnés comme étant des contours (le 1 indique que le pixel est un contour et le 0 indique qu'il ne l'est pas). Ensuite, nous avons utilisé une étape de fusion de ces deux estimations [**fuse*].*MethodFusion()*] pour reconstruire l'image finale débruitée *ImgFusionnee*.

Dans le chapitre suivant, nous allons présenter les expérimentations et les résultats.

CHAPITRE 5

EXPÉRIMENTATION ET RÉSULTATS

Dans ce chapitre, nous montrons des expériences de simulation de notre méthode, tout en présentant les résultats en images, en PSNR et en MSE.

Pour évaluer la performance de l'approche, nous la comparons avec les méthodes des moyennes non-locales (NL-Means) [13] et de la transformée en cosinus discrète (DCT) [78]. Dans ce rapport de mémoire, nous nous intéressons aux images en noir et blanc.

5.1 Implémentation

Nous avons implémenté le projet en C++ sous ubuntu 13.10. Pour la simulation, nous utilisons des images en niveau de gris de dimension 512 x 512 (voir figure 5.1) et nous procédons comme suit :

1. Charger une image de la collection d'images (figure 5.1) ;
2. Dégrader l'image en simulant numériquement un bruit additif gaussien de variance au choix ;
3. Débruiter l'image dégradée avec la méthode proposée ;
4. Mesurer la performance de la méthode, en comparant l'image résultante avec l'image d'entrée de bonne qualité ;
5. Enregistrer l'image filtrée et celle bruitée ;
6. Création de trois fenêtres d'affichage ;
7. Création des images (filtrée, dégradée et originale) ;
8. Affichage.



Lena



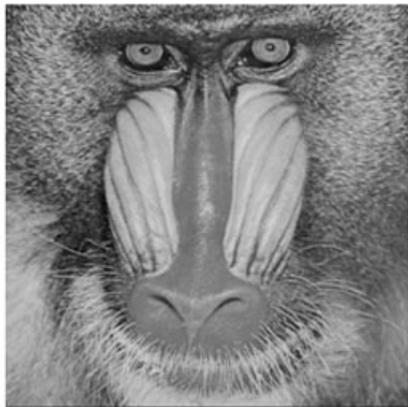
Boat



Barbara



Cameraman



Baboon



Peppers

Figure 5.1 : Collection des images utilisées.

5.2 Résultats

5.2.1 Évaluation des résultats de simulation de la méthode proposée

Dans les tables suivantes :

- Opt1 : représentent l'Option 1 de l'approche où la détection de contours est faite sur l'image bruitée. Avec les paramètres :

$$i = 7, e = 7, h = 100, p = 10.0, c = 1, \sigma = 10.$$

$$i = 7, e = 7, h = 180, p = 10.0, c = 1, \sigma = 20.$$

$$i = 9, e = 9, h = 300, p = 10.0, c = 1, \sigma = 30.$$

- Opt2 : représentent l'Option 2 de l'approche où la détection de contours est faite sur l'image débruitée par NLM. Avec les mêmes paramètres que l'option 1, seulement c change ($c = 2$).

- S : est le seuil utilisé par Canny choisi empiriquement.

Image	Opt1 ($S = 0.8$)	Opt1 ($S = 0.9$)	Opt2 ($S = 0.8$)	Opt2 ($S = 0.9$)
Lena	36.17	36.44	36.86	37.10
Boat	36.08	36.63	36.90	37.32
Barbara	37.06	37.18	37.34	37.64
Cameraman	37.92	38.74	39.11	40.10
Peppers	36.67	37.00	37.31	37.76
Baboon	34.70	35.02	35.45	35.83

Tableau 5.I : Résultats en PSNR, du débruitage des images de la figure 5.1 par la méthode proposée. $\sigma = 10$.

Image	Opt1 ($S = 0.8$)	Opt1 ($S = 0.9$)	Opt2 ($S = 0.8$)	Opt2 ($S = 0.9$)
Lena	15.69	14.75	13.41	12.67
Boat	16.04	14.13	13.28	12.04
Barbara	12.81	12.46	12.01	11.19
Cameraman	10.50	8.70	7.99	6.35
Peppers	14.01	12.98	12.08	10.89
Baboon	22.01	20.49	18.56	16.97

Tableau 5.II : Résultats en MSE, du débruitage des images de la figure 5.1 avec la méthode proposée, $\sigma = 10$.

Image	Opt1 ($S = 0.8$)	Opt1 ($S = 0.9$)	Opt2 ($S = 0.8$)	Opt2 ($S = 0.9$)
Lena	33.77	33.98	34.22	34.60
Boat	32.87	33.14	35.26	33.65
Barbara	33.20	33.47	33.75	34.03
Cameraman	34.31	34.75	35.26	36.05
Peppers	34.31	34.70	34.79	35.05
Baboon	29.38	29.65	30.15	30.69

Tableau 5.III : Résultats en PSNR, du débruitage des images de la figure 5.1 avec la méthode proposée, $\sigma = 20$.

Image	Opt1 ($S = 0.8$)	Opt1 ($S = 0.9$)	Opt2 ($S = 0.8$)	Opt2 ($S = 0.9$)
Lena	27.31	25.98	24.61	22.56
Boat	33.58	31.56	30.12	28.03
Barbara	31.13	29.26	27.39	25.70
Cameraman	24.11	21.80	19.36	16.16
Peppers	24.13	22.01	21.57	20.34
Baboon	75.01	70.53	62.88	55.50

Tableau 5.IV : Résultats en MSE, du débruitage des images de la figure 5.1 avec la méthode proposée, $\sigma = 20$.

Image	Opt1 ($S = 0.8$)	Opt1 ($S = 0.9$)	Opt2 ($S = 0.8$)	Opt2 ($S = 0.9$)
Lena	31.74	31.82	32.00	32.23
Boat	31.25	31.48	31.62	31.97
Barbara	31.07	31.11	31.58	32.27
Cameraman	31.99	32.03	32.50	33.73
Peppers	32.13	32.28	36.13	36.88
Baboon	27.06	27.75	28.01	28.15

Tableau 5.V : Résultats en PSNR, du débruitage des images de la figure 5.1 avec la méthode proposée, $\sigma = 30$.

Image	Opt1 ($S = 0.8$)	Opt1 ($S = 0.9$)	Opt2 ($S = 0.8$)	Opt2 ($S = 0.9$)
Lena	43.56	42.77	40.99	38.94
Boat	48.72	46.21	44.79	41.34
Barbara	50.83	50.41	45.19	38.56
Cameraman	41.08	40.70	36.59	27.56
Peppers	39.81	38.44	15.86	13.32
Baboon	127.99	109.09	102.9	99.60

Tableau 5.VI : Résultats en MSE du débruitage des images de la figure 5.1 avec la méthode proposée, $\sigma = 30$.

les résultats dans Les tables 5.I , 5.III , 5.V , 5.II , 5.IV , 5.VI nous montrent que l'option 2 avec le seuil $S = 0.9$ donnent le meilleur résultat.

Les figures 5.2 ,5.3 , 5.4 , 5.5 , 5.6 , 5.7 , 5.8 , 5.9 , 5.10 , 5.11 , 5.12 , 5.13 , 5.14 , 5.15 , 5.16 , 5.17 , 5.18 , 5.19 , représentent quelques résultats en images obtenues au cours de la simulation de l'approche, effectuée sur la collection d'images 5.1. Avec le seuil $S = 0.9$ pour la détection des contours dans les deux options de notre approche.



Figure 5.2 : Simulation de débruitage sur l'image Lena par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 10$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.

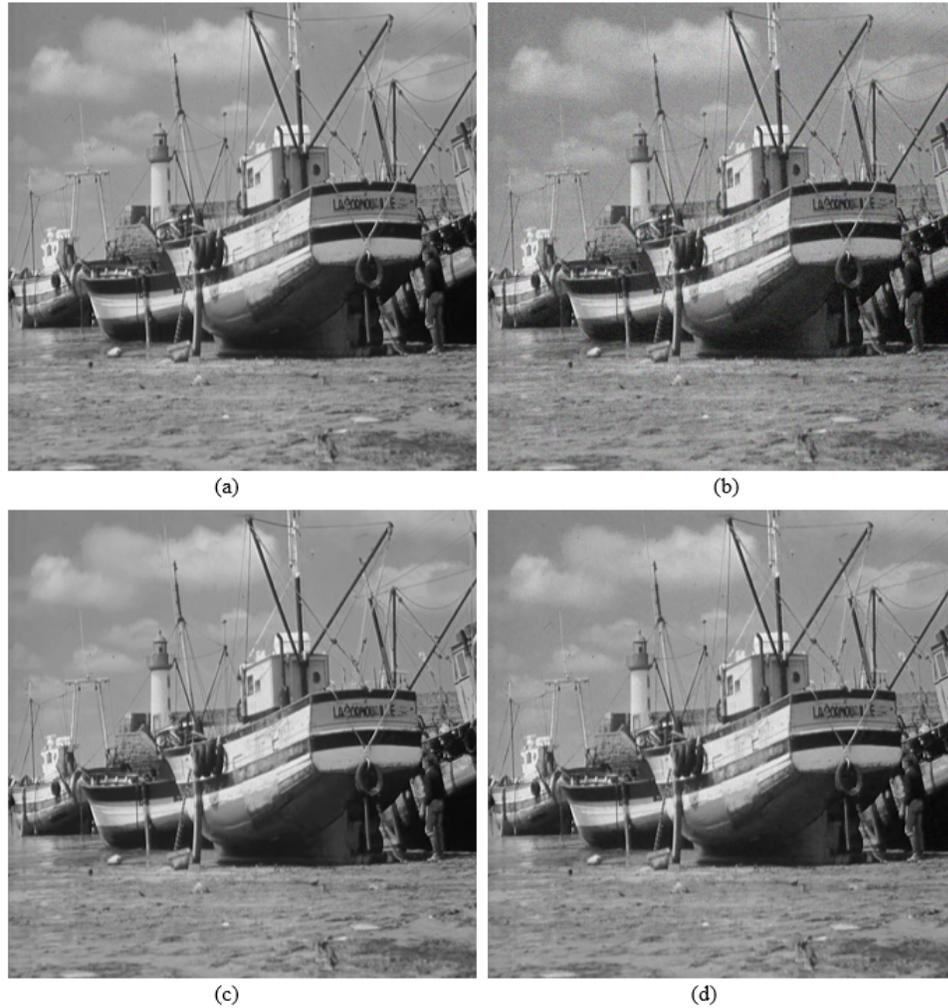


Figure 5.3 : Simulation de débruitage sur l'image Boat par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 10$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.



Figure 5.4 : Simulation de débruitage sur l'image Barbara par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 10$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.



Figure 5.5 : Simulation de débruitage sur l'image Cameraman par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 10$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.

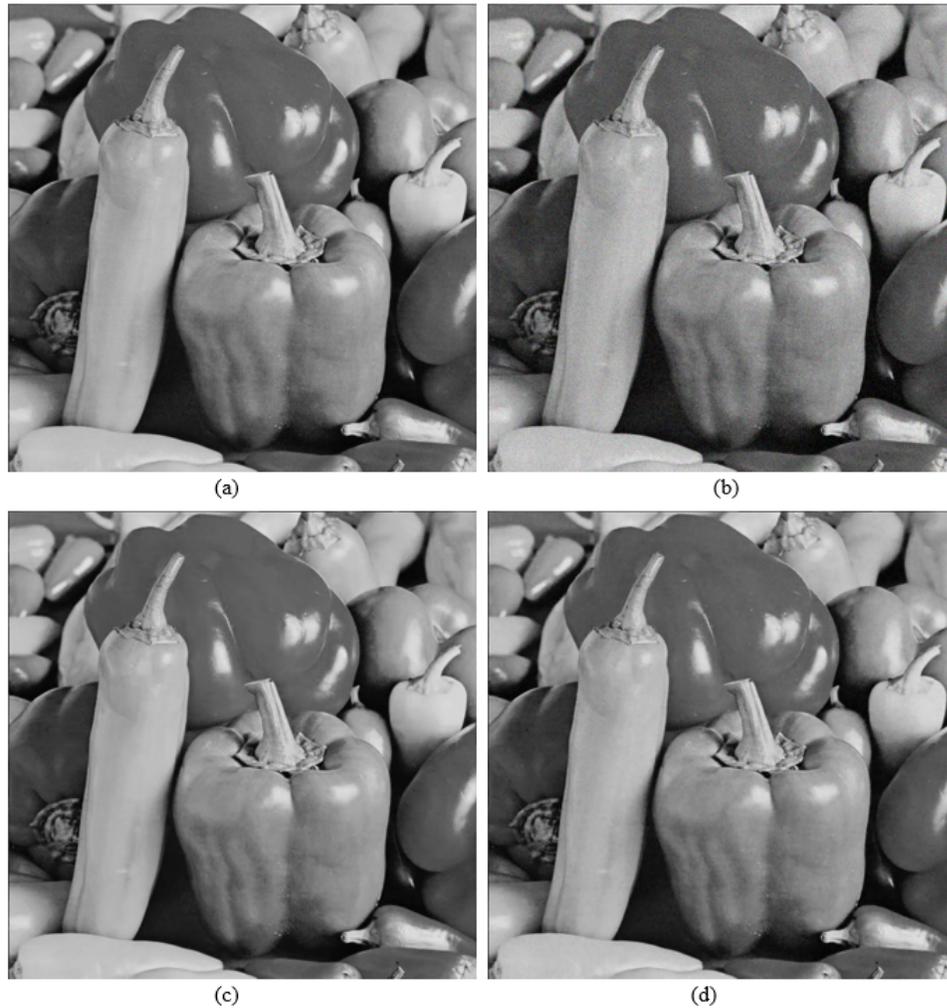


Figure 5.6 : Simulation de débruitage sur l'image Peppers par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 10$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.

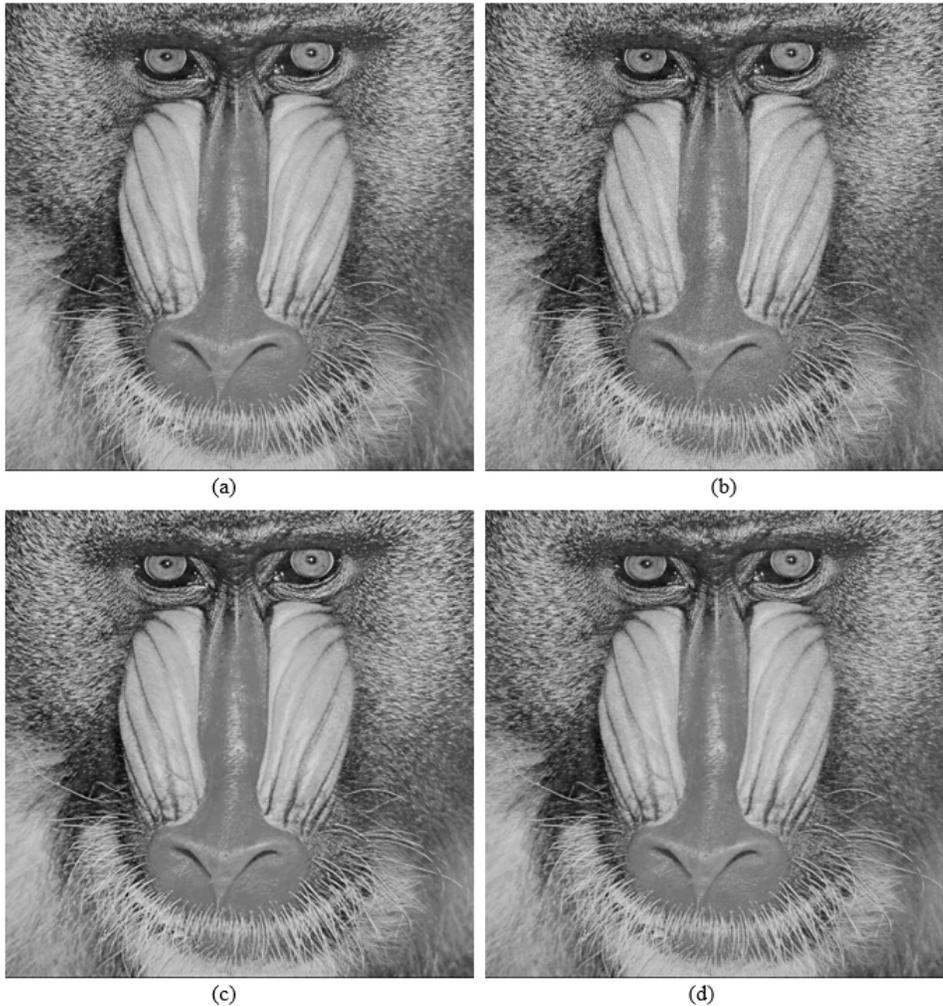


Figure 5.7 : Simulation de débruitage sur l'image Baboon par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 10$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.



Figure 5.8 : Simulation de débruitage sur l'image Lena par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 20$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.

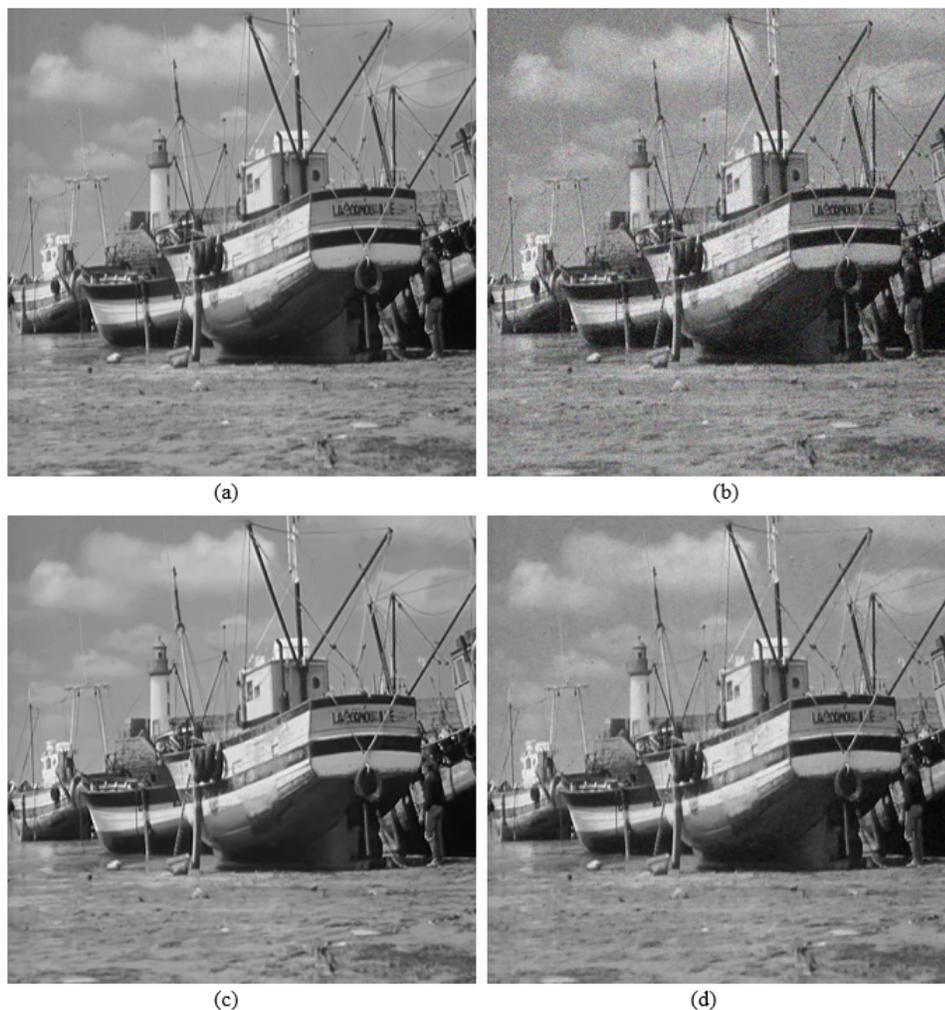


Figure 5.9 : Simulation de débruitage sur l'image Boat par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 20$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.



Figure 5.10 : Simulation de débruitage sur l'image Barbara par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 20$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.

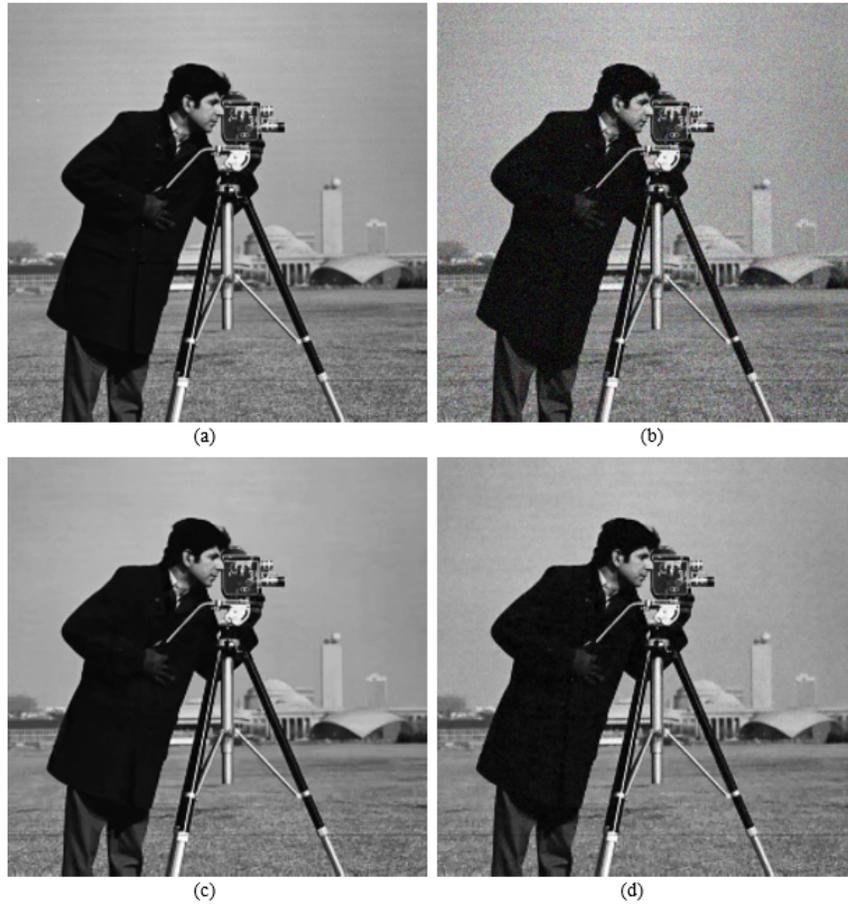


Figure 5.11 : Simulation de débruitage sur l'image Cameraman par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 20$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.

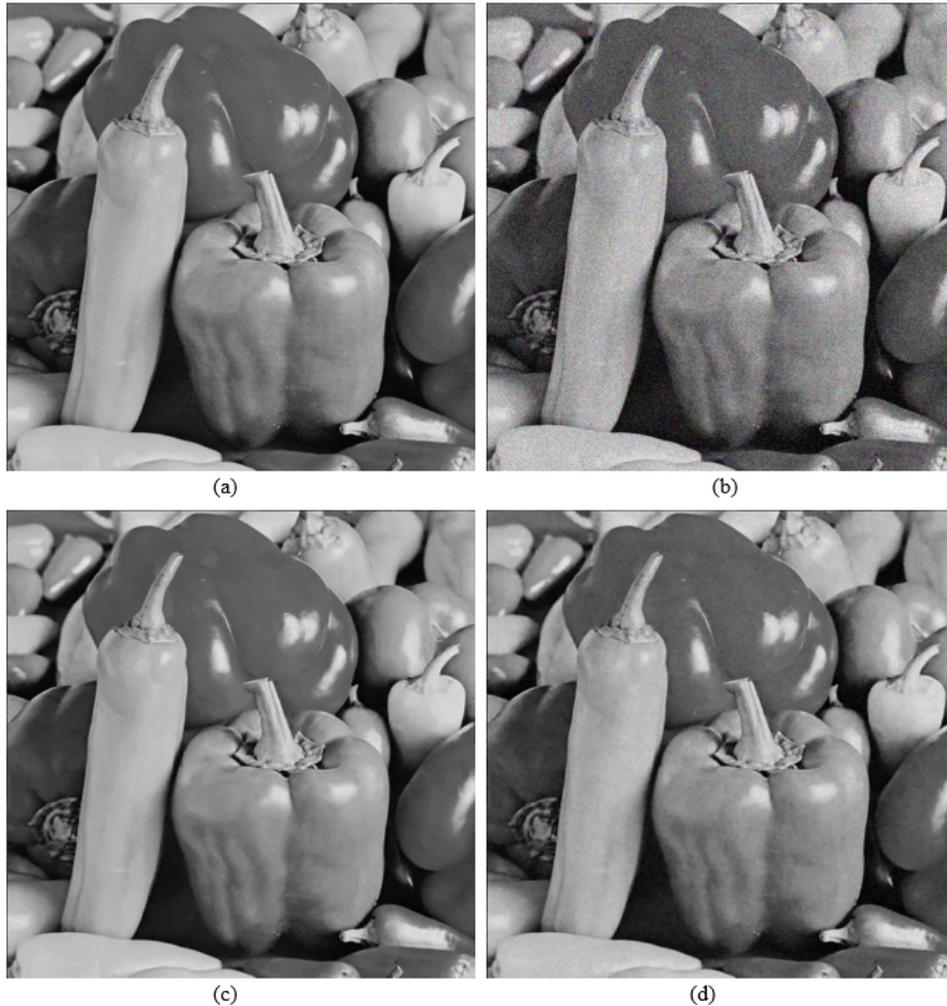


Figure 5.12 : Simulation de débruitage sur l'image Peppers par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 20$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.

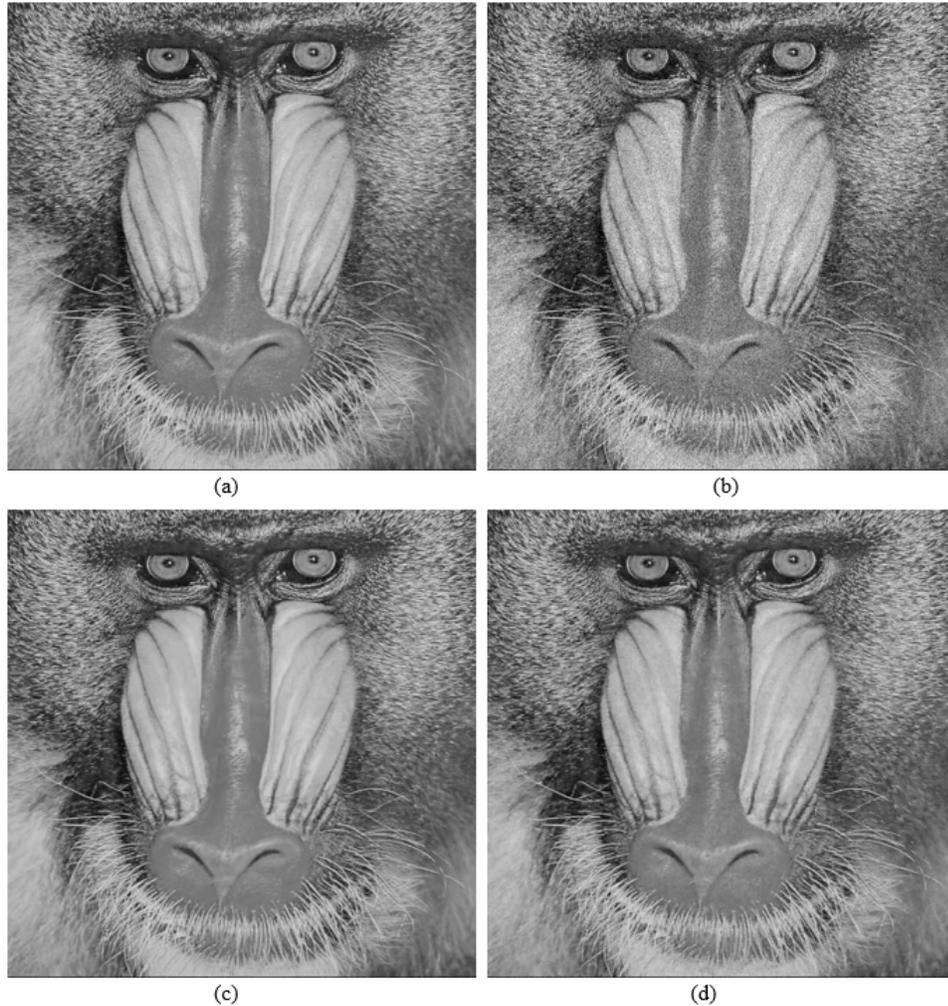


Figure 5.13 : Simulation de débruitage sur l'image Baboon par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 20$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.



Figure 5.14 : Simulation de débruitage sur l'image Lena par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 30$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.

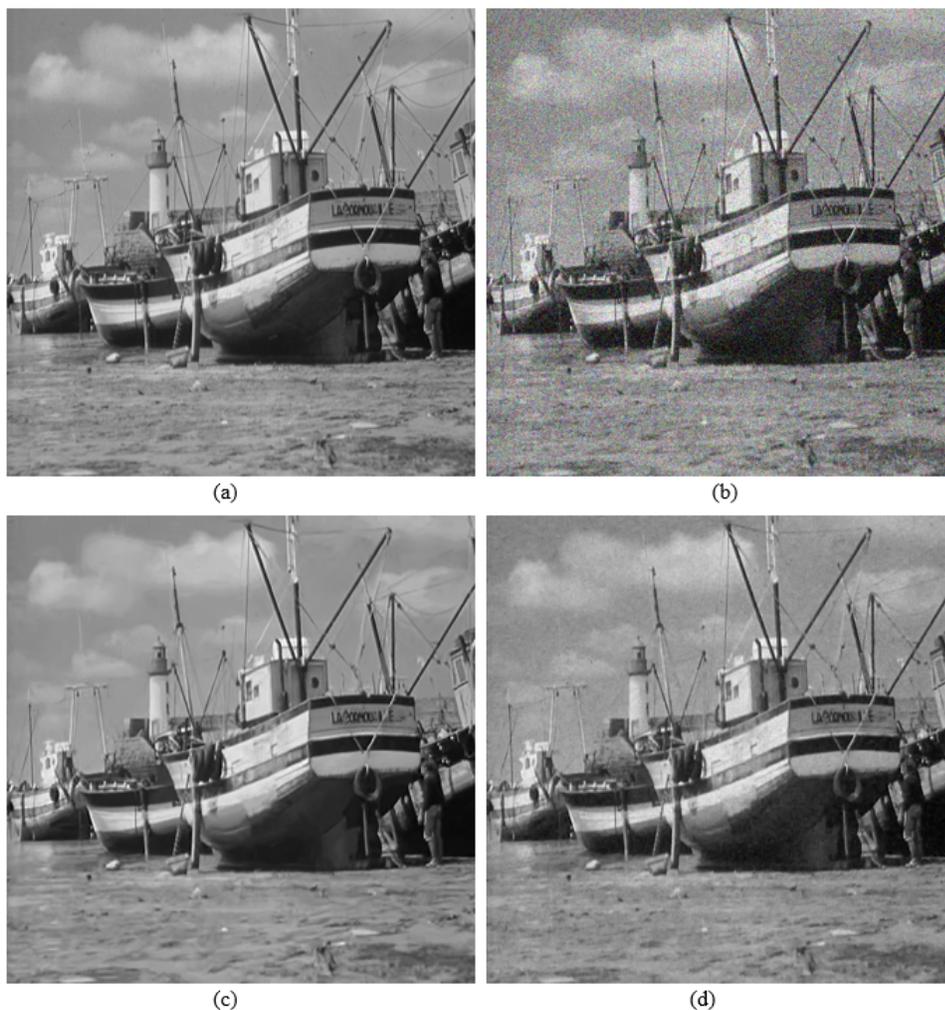


Figure 5.15 : Simulation de débruitage sur l'image Boat par la méthode proposée :(a) Image Originale, (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 30$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.



Figure 5.16 : Simulation de débruitage sur l'image Barbara par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 30$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.

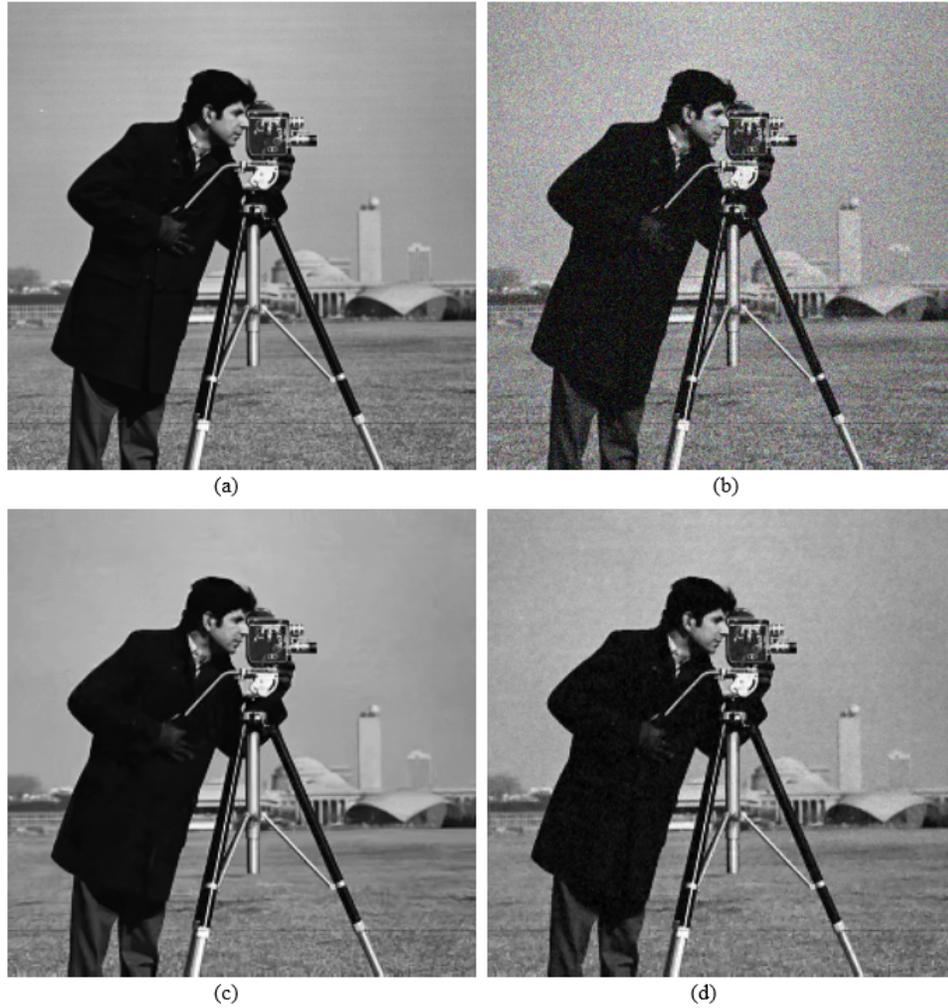


Figure 5.17 : Simulation de débruitage sur l'image Cameraman par la méthode proposée : (a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 30$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.

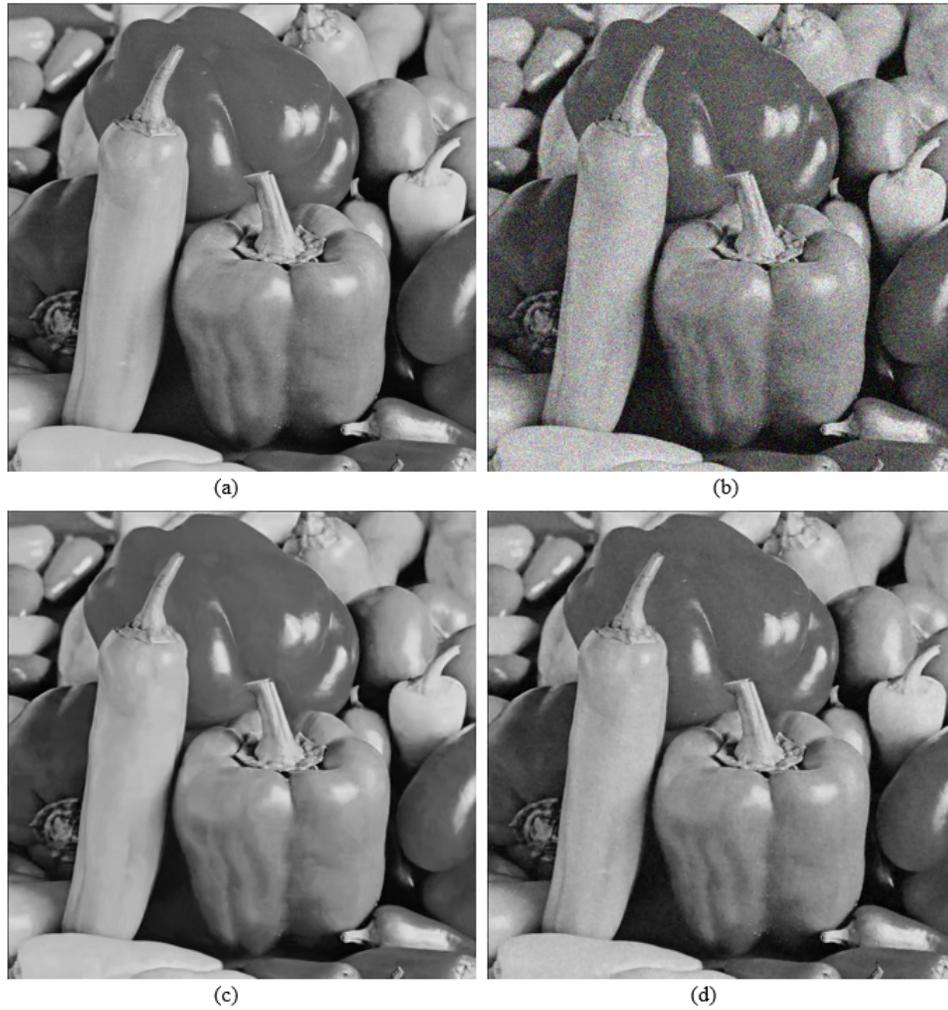


Figure 5.18 : Simulation de débruitage sur l'image Peppers par la méthode proposée :(a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 30$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.

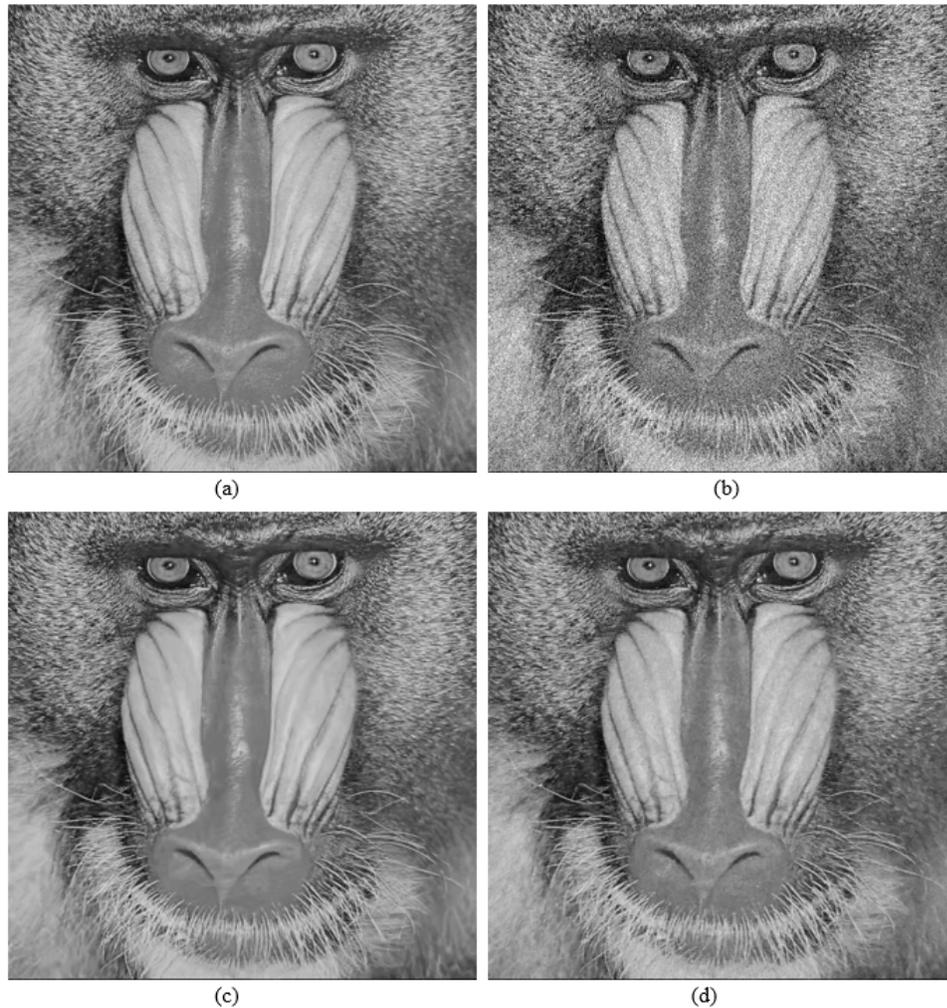


Figure 5.19 : Simulation de débruitage sur l'image Baboon par la méthode proposée :(a) Image Originale, (b) Image bruitée avec un bruit blanc gaussien $\sigma = 30$, (c) Image débruitée avec l'option 2 de la méthode où la détection de contours se fait sur l'image filtrée par NLM, (d) Image débruitée avec l'option 1 de la méthode où la détection de contours se fait sur l'image bruitée.

5.2.2 Comparaison

Pour l'évaluation de la performance de notre technique, nous la comparons avec les filtres NL-means [13] et DCT [78] selon les critères définis dans la section 2.5, dont : l'aspect visuel de la qualité de restauration de l'image, le rapport signal sur bruit en pic (PSNR) et l'erreur quadratique moyenne (MSE).

- Un PSNR plus grand indique que le filtre est performant.
- Une MSE plus petite indique que le filtre est performant (estimation plus proche de l'image originale).
- Plus la différence entre l'image originale et image débruitée, ressemble à un bruit (moins de détails de l'image originale), plus la méthode est performante.

Pour toutes les expériences dans cette section, nous utilisons les paramètres de l'option 2 5.2.1 avec le seuil fixé à 0.9 pour la méthode proposée.

Pour les algorithmes de débruitage DCT et NLM, nous avons conservé les paramètres par défaut proposés par les auteurs et qui sont adaptatifs à la variance du bruit Gaussien dégradant l'image de départ.

Les figures 5.20, 5.21, 5.22, 5.23, 5.24, 5.25, 5.26, 5.27, 5.28, 5.29, 5.30, 5.31, 5.32, 5.33, 5.34, 5.35, 5.36, 5.37, représentent les images obtenues au cours de l'expérience de débruitage par NLM, DCT et la méthode proposée, effectuée sur les six images de la collection 5.1. Ces images permettront à l'œil humain de décider de l'amélioration de la qualité de l'image filtrée et de faire une comparaison des performances de ces trois méthodes de débruitage.



(a) Image bruitée, $\sigma = 10$



(b) Image débruitée par la méthode proposée



(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.20 : Résultat de l'expérience de débruitage sur l'image Lena par les filtres DCT, NLM et la méthode proposée. $\sigma = 10$.



(a) Image bruitée, $\sigma = 10$



(b) Image débruitée par la méthode proposée

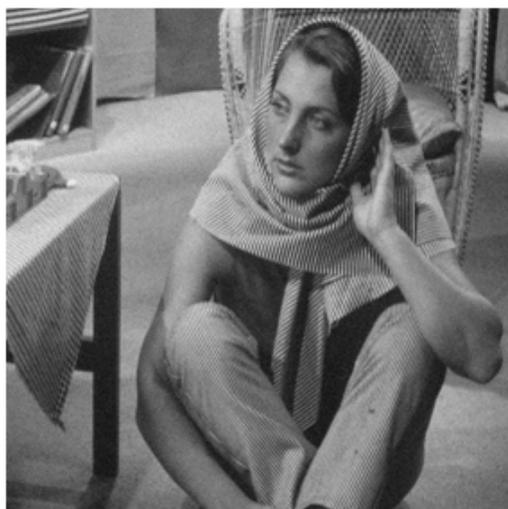


(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.21 : Résultat de l'expérience de débruitage sur l'image Boat par les filtres DCT, NLM et la méthode proposée. $\sigma = 10$.



(a) Image bruitée, $\sigma = 10$



(b) Image débruitée par la méthode proposée



(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.22 : Résultat de l'expérience de débruitage sur l'image Barbara par les filtres DCT, NLM et la méthode proposée. $\sigma = 10$.



(a) Image bruitée, $\sigma = 10$



(b) Image débruitée par la méthode proposée

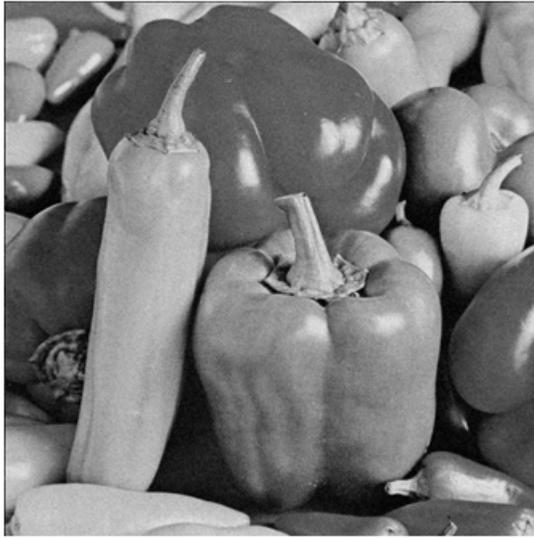


(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.23 : Résultat de l'expérience de débruitage sur l'image Cameraman par les filtres DCT, NLM et la méthode proposée. $\sigma = 10$.



(a) Image bruitée, $\sigma = 10$



(b) Image débruitée par la méthode proposée



(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.24 : Résultat de l'expérience de débruitage sur l'image Peppers par les filtres DCT, NLM et la méthode proposée. $\sigma = 10$.

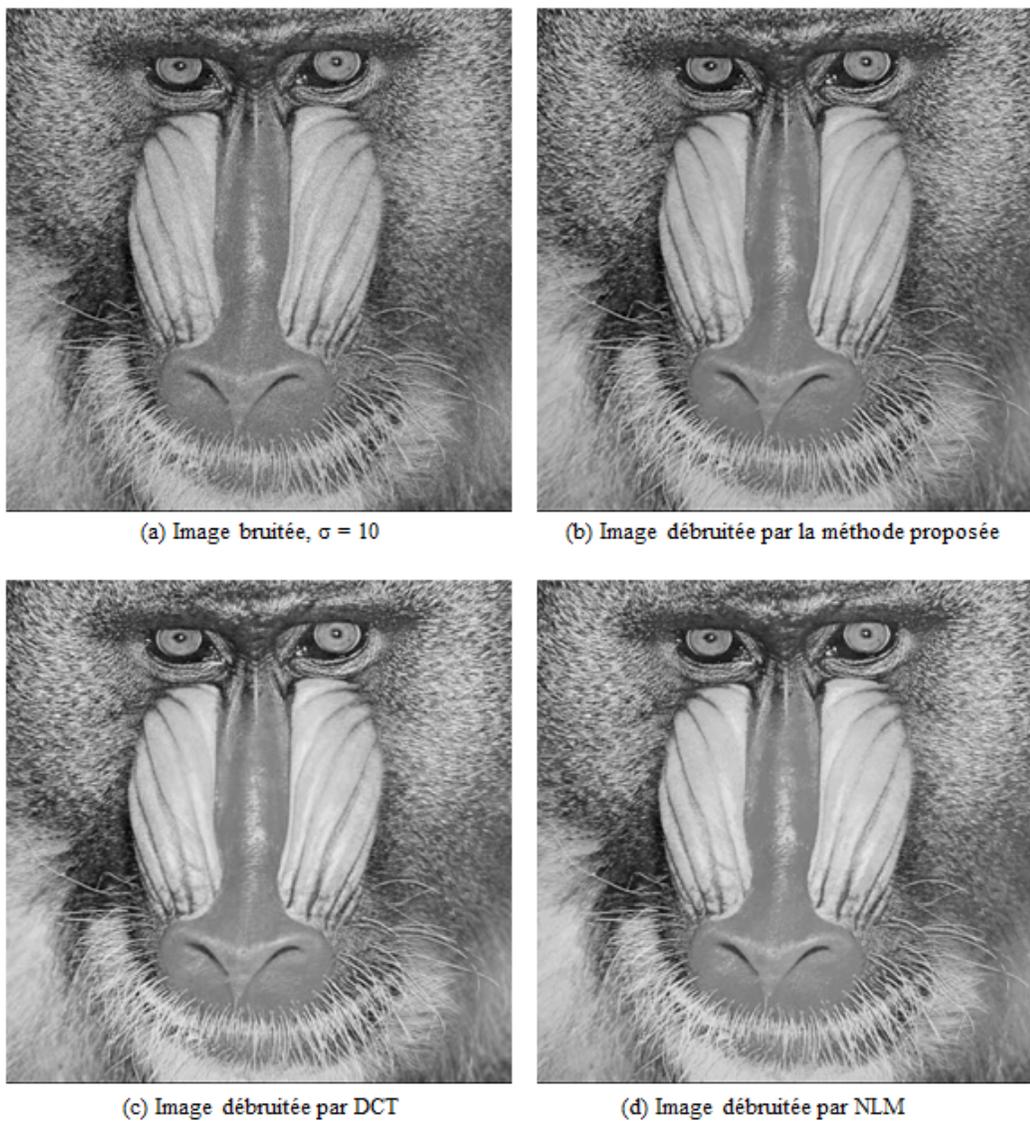


Figure 5.25 : Résultat de l'expérience de débruitage sur l'image Baboon par les filtres DCT, NLM et la méthode proposée. $\sigma = 10$.



(a) Image bruitée, $\sigma = 20$



(b) Image débruitée par la méthode proposée



(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.26 : Résultat de l'expérience de débruitage sur l'image Lena par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 20$.



(a) Image bruitée, $\sigma = 20$



(b) Image débruitée par la méthode proposée

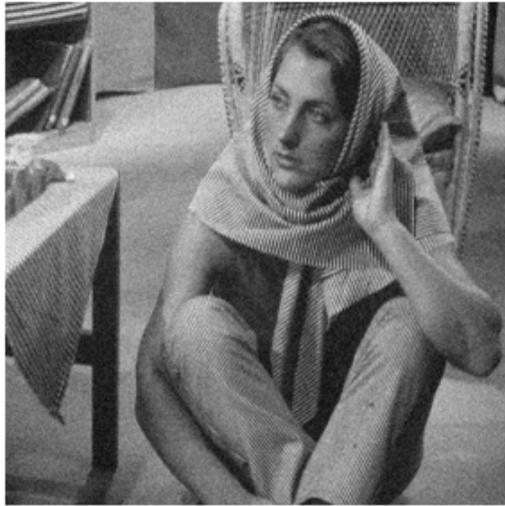


(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.27 : Résultat de l'expérience de débruitage sur l'image Boat par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 20$.



(a) Image bruitée, $\sigma = 20$



(b) Image débruitée par la méthode proposée



(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.28 : Résultat de l'expérience de débruitage sur l'image Barbara par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 20$.



(a) Image bruitée, $\sigma = 20$



(b) Image débruitée par la méthode proposée

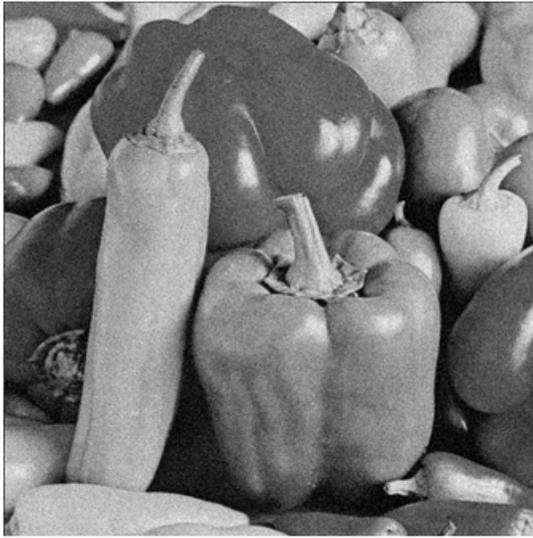


(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.29 : Résultat de l'expérience de débruitage sur l'image Cameraman par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 20$.



(a) Image bruitée, $\sigma = 20$



(b) Image débruitée par la méthode proposée

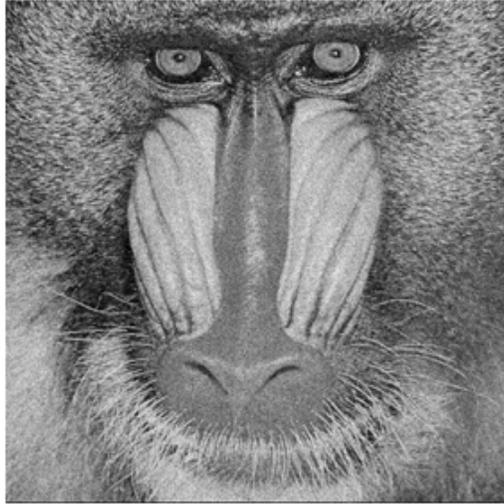


(c) Image débruitée par DCT

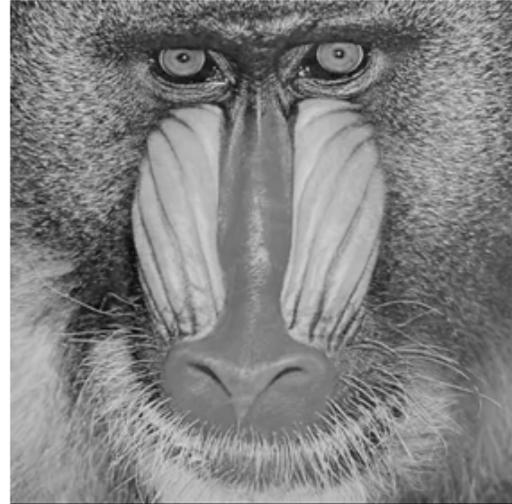


(d) Image débruitée par NLM

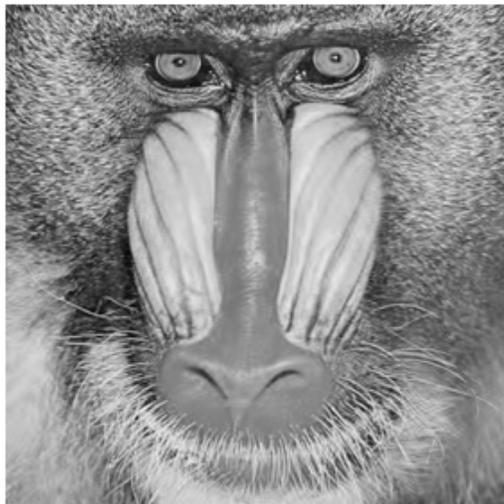
Figure 5.30 : Résultat de l'expérience de débruitage sur l'image Peppers par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 20$.



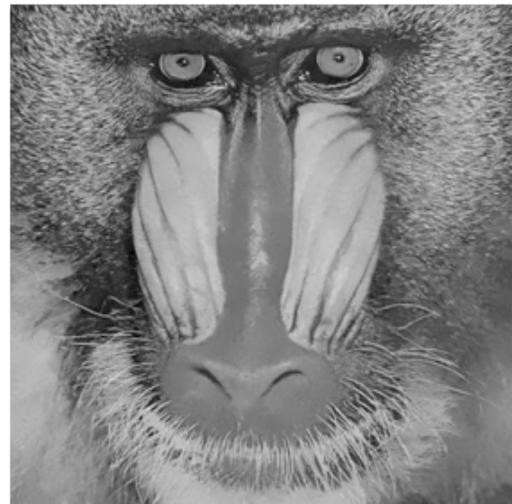
(a) Image bruitée, $\sigma = 20$



(b) Image débruitée par la méthode proposée



(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.31 : Résultat de l'expérience de débruitage sur l'image Baboon par les filtres DCT, NLM et la méthode proposée. $\sigma = 20$.



(a) Image bruitée, $\sigma = 30$



(b) Image débruitée par la méthode proposée



(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.32 : Résultat de l'expérience de débruitage sur l'image Lena par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 30$.



(a) Image bruitée, $\sigma = 30$



(b) Image débruitée par la méthode proposée



(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.33 : Résultat de l'expérience de débruitage sur l'image Boat par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 30$.



(a) Image bruitée, $\sigma = 30$



(b) Image débruitée par la méthode proposée



(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.34 : Résultat de l'expérience de débruitage sur l'image Barbara par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 30$.



(a) Image bruitée, $\sigma = 30$



(b) Image débruitée par la méthode proposée

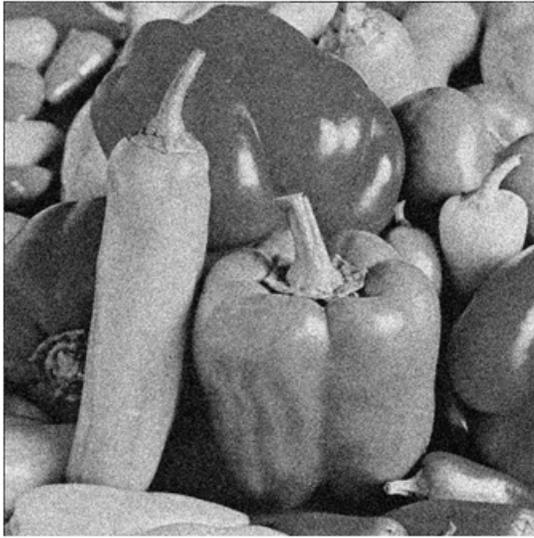


(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.35 : Résultat de l'expérience de débruitage sur l'image Cameraman par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 30$.



(a) Image bruitée, $\sigma = 30$



(b) Image débruitée par la méthode proposée

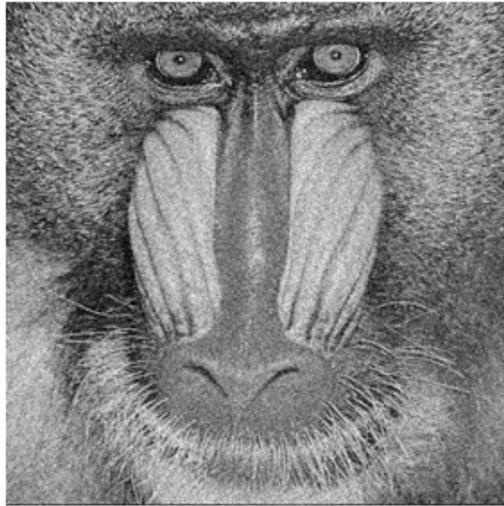


(c) Image débruitée par DCT

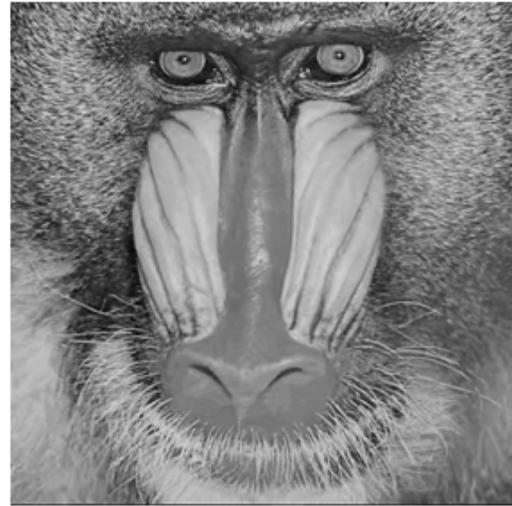


(d) Image débruitée par NLM

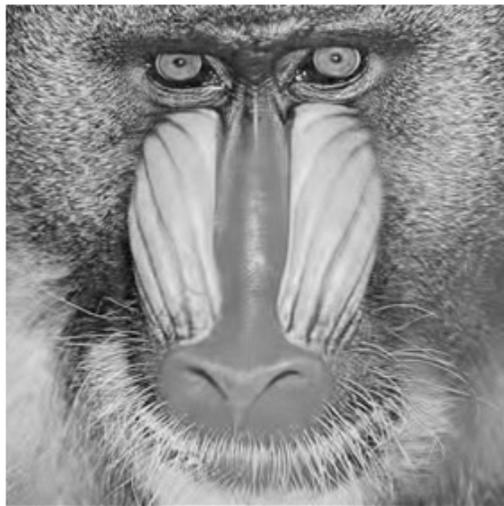
Figure 5.36 : Résultat de l'expérience de débruitage sur l'image Peppers par les algorithmes (DCT, NLM et Méthode proposée). $\sigma = 30$.



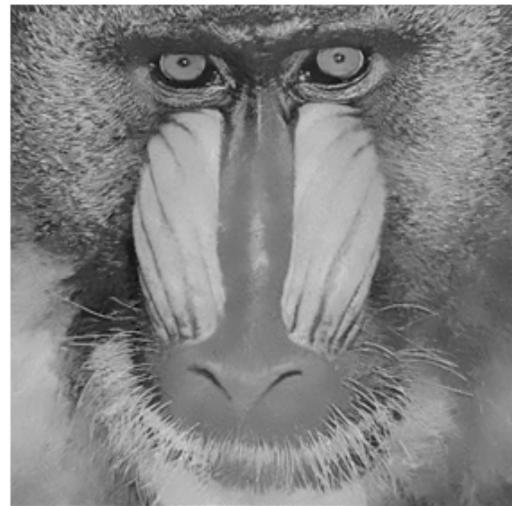
(a) Image bruitée, $\sigma = 30$



(b) Image débruitée par la méthode proposée



(c) Image débruitée par DCT



(d) Image débruitée par NLM

Figure 5.37 : Résultat de l'expérience de débruitage sur l'image Baboon par les filtres DCT, NLM et la méthode proposée. $\sigma = 30$.

Image	DCT	NLM	Proposée
Lena	35.91	33.60	37.10
Boat	35.79	32.90	37.32
Barbara	35.94	32.89	37.64
Cameraman	36.69	33.64	40.10
Peppers	36.29	34.02	37.76
Baboon	33.63	30.20	35.83

Tableau 5.VII : Résultats de débruitage en PSNR, des images (Lena, Boat, Barbara, Cameraman, Peppers et Baboon), avec la méthode proposée et les algorithmes (NLM [13] et DCT [78]) . $\sigma = 10$.

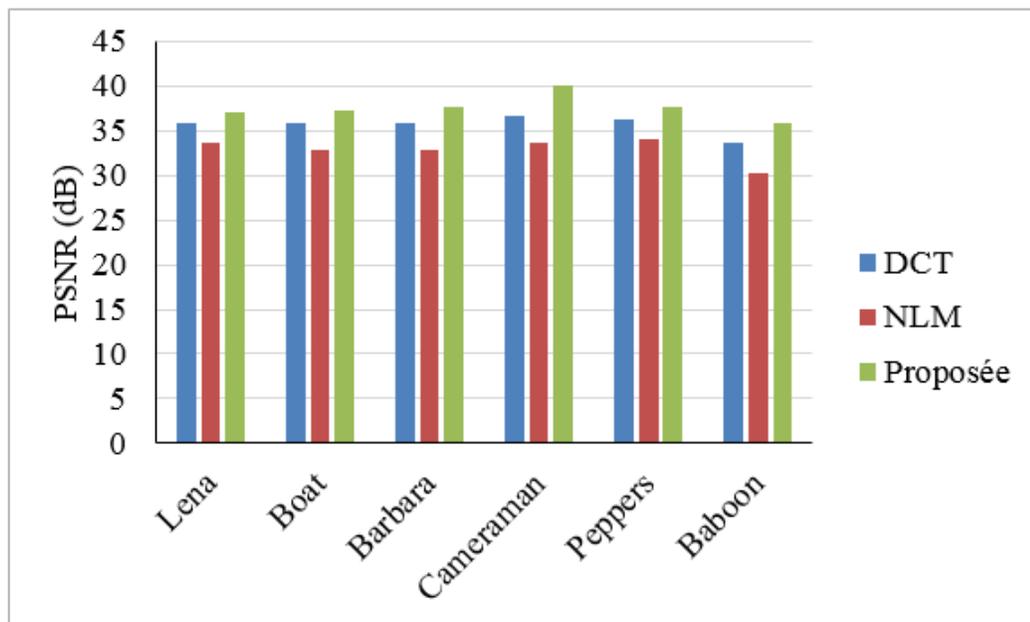


Figure 5.38 : Graphique de comparaison des résultats présentés dans la table5.VII

Image	DCT	NLM	Proposée
Lena	33.38	31.04	34.60
Boat	32.28	29.75	33.65
Barbara	32.70	30.13	34.03
Cameraman	32.32	30.14	36.05
Peppers	33.82	32.52	35.05
Baboon	28.81	26.26	30.69

Tableau 5.VIII : Résultats de débruitage en PSNR, des images (Lena, Boat, Barbara, Cameraman, Peppers et Baboon), avec la méthode proposée et les algorithmes (NLM [13] et DCT [78]). $\sigma = 20$.

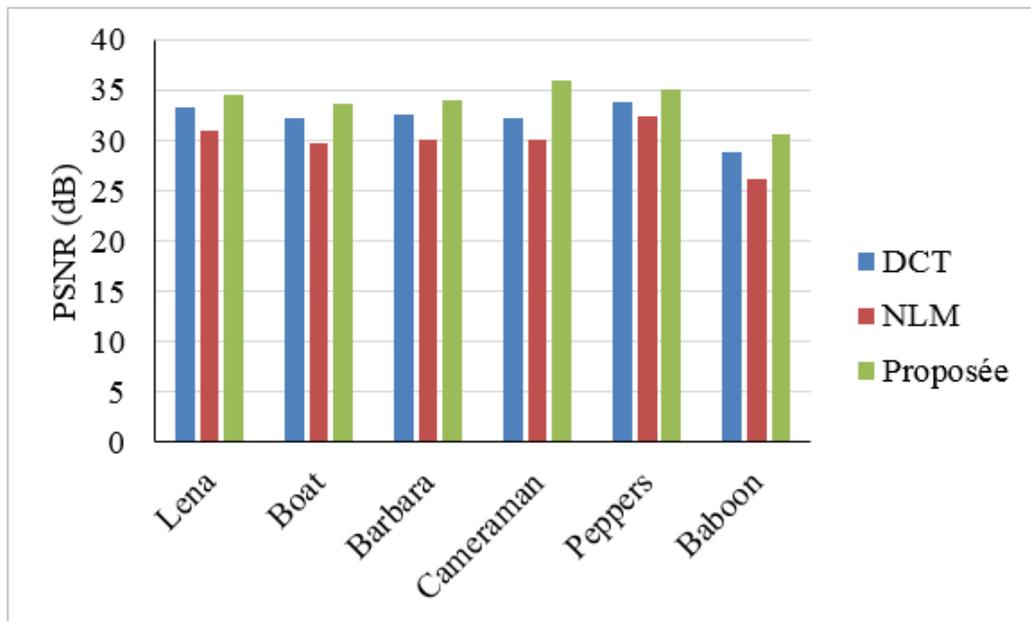


Figure 5.39 : Graphique de comparaison des résultats présentés dans la table5.VIII

Image	DCT	NLM	Proposée
Lena	31.83	29.14	32.30
Boat	30.28	27.70	31.97
Barbara	30.71	27.87	32.27
Cameraman	30.09	28.02	33.73
Peppers	32.05	29.50	36.88
Baboon	26.31	24.12	28.15

Tableau 5.IX : Résultats de débruitage en PSNR, des images (Lena, Boat, Barbara, Cameraman, Peppers et Baboon), avec la méthode proposée et les algorithmes (NLM [13] et DCT [78]). $\sigma = 30$.

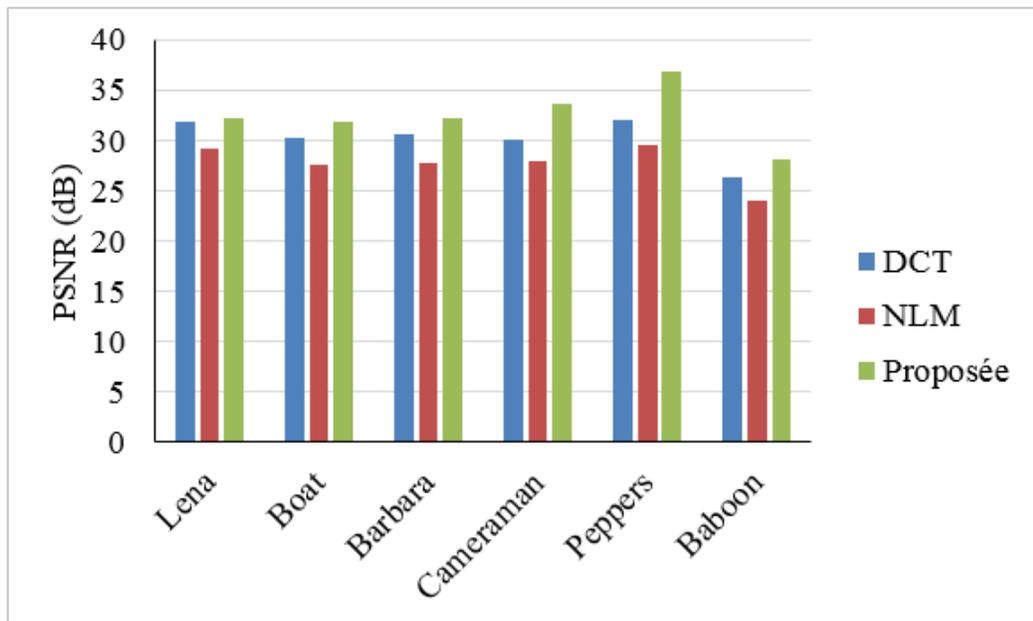


Figure 5.40 : Graphique de comparaison des résultats présentés dans la table 5.IX

Image	DCT	NLM	Proposée
Lena	16.73	28.41	12.67
Boat	17.14	33.41	12.04
Barbara	16.56	33.29	11.19
Cameraman	10.24	21.44	6.35
Peppers	15.29	25.80	10.89
Baboon	28.20	62.09	16.97

Tableau 5.X : Affichage de l'erreur quadratique moyenne (MSE) pour les expériences de débruitage des images (Lena, Boat, Barbara, Cameraman, Peppers et Baboon), avec la méthode proposée et les algorithmes (NLM [13] et DCT [78]). $\sigma = 10$.

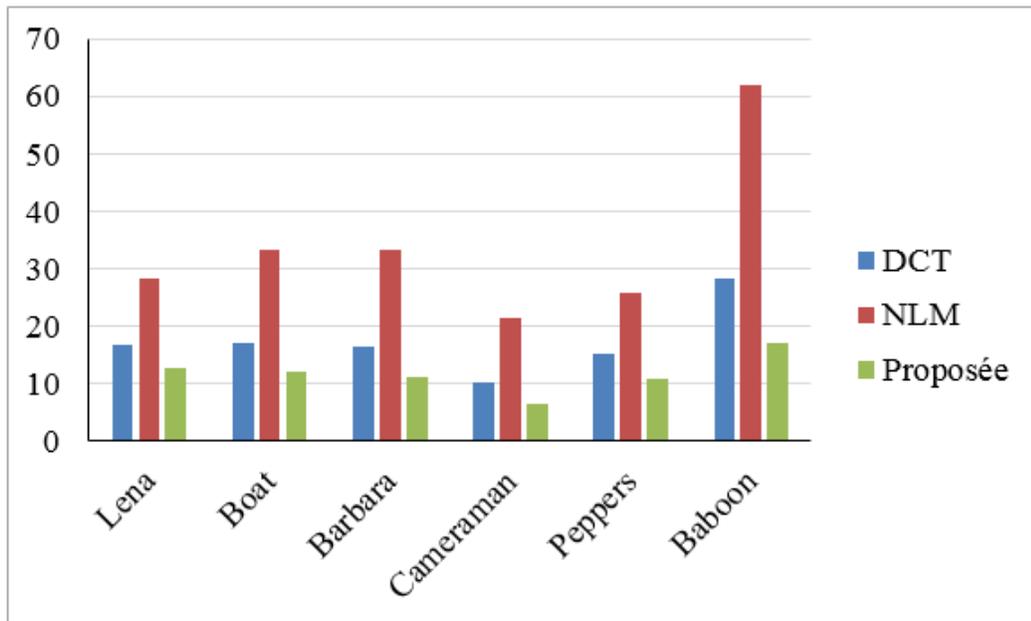


Figure 5.41 : Graphique de comparaison des résultats présentés dans la table 5.X

Image	DCT	NLM	Proposée
Lena	29.92	51.12	22.56
Boat	39.44	68.89	28.03
Barbara	34.93	63.04	25.70
Cameraman	26.94	47.75	16.16
Peppers	28.52	46.10	20.34
Baboon	85.56	154.01	55.50

Tableau 5.XI : Affichage de l'erreur quadratique moyenne (MSE) pour les expériences de débruitage des images (Lena, Boat, Barbara, Cameraman, Peppers et Baboon), avec la méthode proposée et les algorithmes (NLM [13] et DCT [78]). $\sigma = 20$.

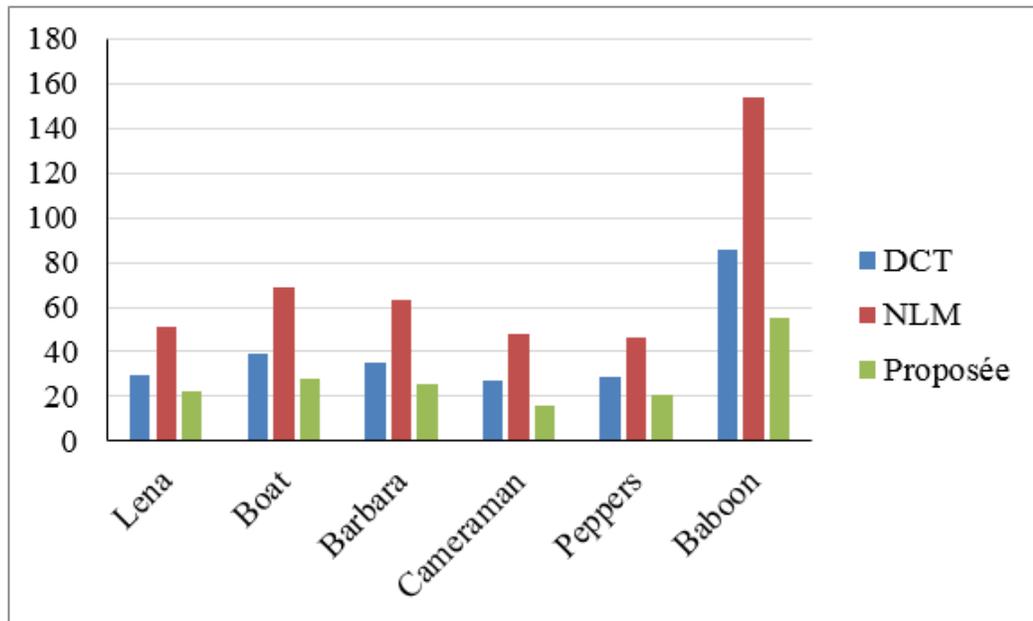


Figure 5.42 : Graphique de comparaison des résultats présentés dans la table5.XI

Image	DCT	NLM	Proposée
Lena	42.64	79.39	38.94
Boat	61.00	110.46	41.34
Barbara	55.20	106.30	38.56
Cameraman	45.97	78.15	27.56
Peppers	40.58	72.93	13.32
Baboon	152.03	251.86	99.60

Tableau 5.XII : Affichage de l'erreur quadratique moyenne (MSE) pour les expériences de débruitage des images (Lena, Boat, Barbara, Cameraman, Peppers et Baboon), avec la méthode proposée et les algorithmes (NLM [13] et DCT [78]). $\sigma = 30$.

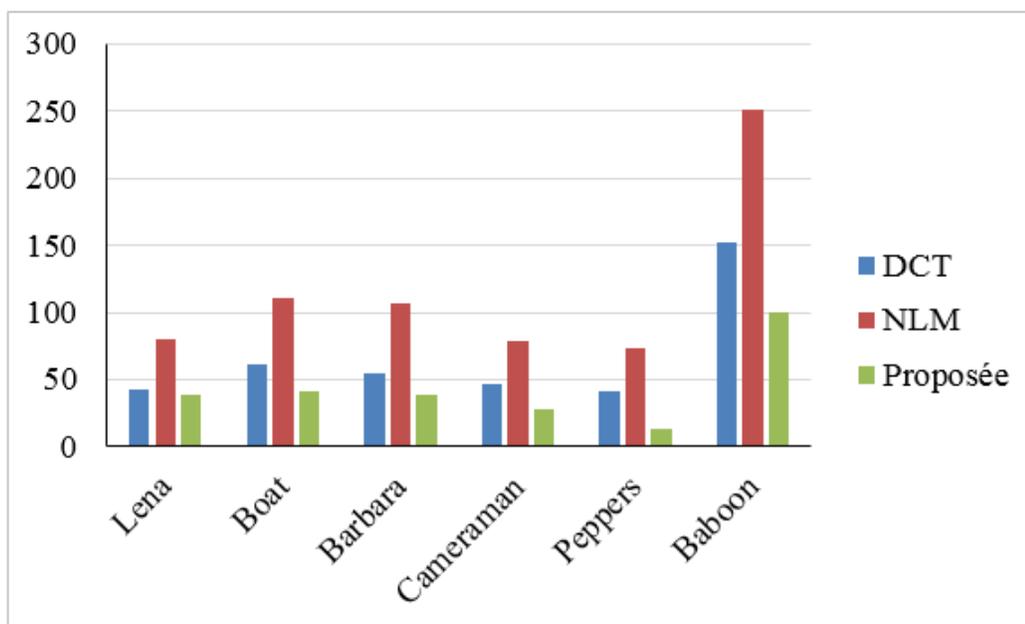


Figure 5.43 : Graphique de comparaison des résultats présentés dans la table 5.XII

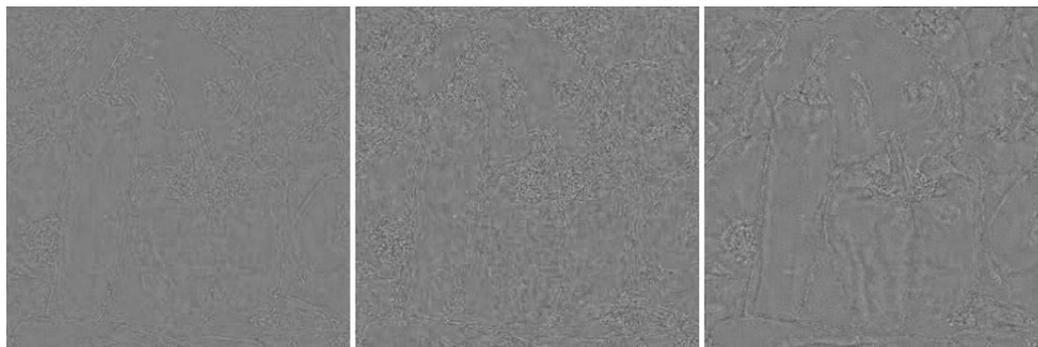


Figure 5.44 : Différence entre l'image Peppers originale et l'image Peppers filtrée ($\sigma = 30$) par : DCT à gauche, la méthode proposée au milieu et NLM à droite. Pour la méthode proposée, nous constatons qu'il y a moins de détails de l'image Peppers dans le bruit enlevé.

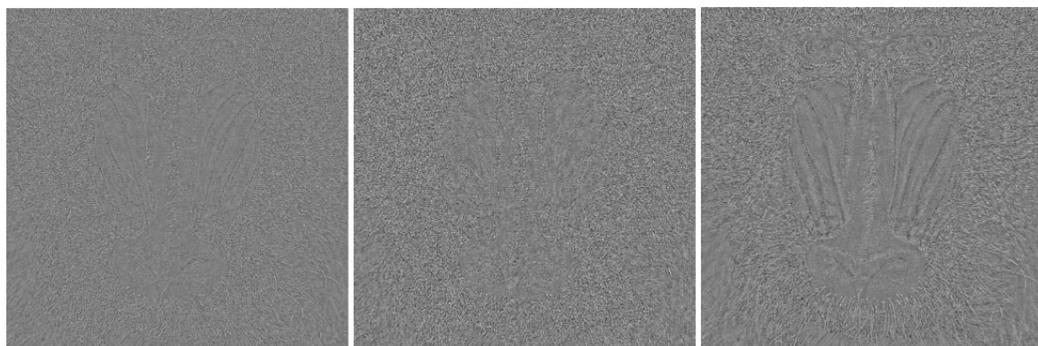


Figure 5.45 : Différence pour l'image Baboon entre l'image originale et l'image débruitée ($\sigma = 30$) par : DCT à gauche, la méthode proposée au milieu et NLM à droite. Pour la méthode proposée, nous constatons qu'il y a moins de détails de l'image Baboon dans le bruit enlevé.

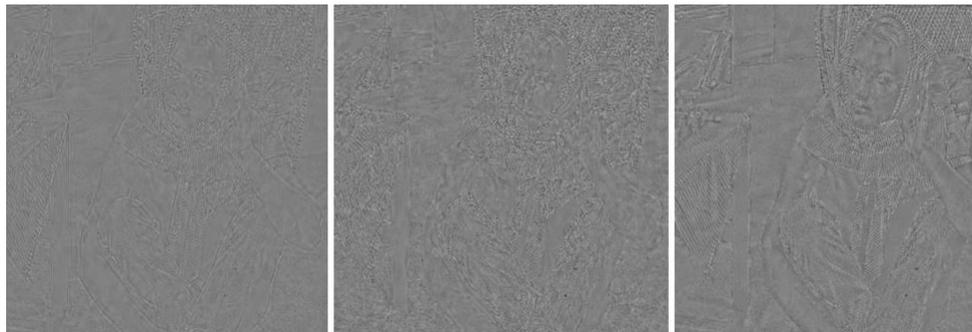


Figure 5.46 : Différence pour l'image Barbara entre l'image originale et l'image débruitée ($\sigma = 30$) par : DCT à gauche, la méthode proposée au milieu et NLM à droite. Pour cet exemple aussi, nous constatons qu'avec la méthode proposée, il y a moins de détails de l'image Barbara dans le bruit enlevé.

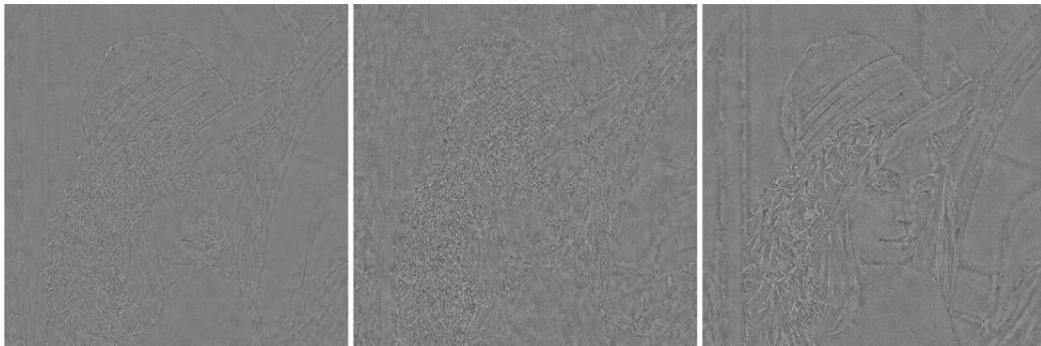


Figure 5.47 : Différence pour l'image Lena entre l'originale et l'image débruitée ($\sigma = 30$) par : DCT à gauche, la méthode proposée au milieu et NLM à droite. Les détails de l'image Lena sont moins visibles dans le bruit enlevé avec la méthode proposée.

CHAPITRE 6

CONCLUSION

Nous avons proposé dans ce rapport de mémoire, une approche pour améliorer le débruitage des images dégradées par un bruit blanc additif gaussien. Nos intérêts étant axés sur les limites de la méthode de débruitage des NLM et celle de la DCT, nous avons proposé une méthode de débruitage combinant ces deux filtres dont le but principal est de bien restaurer l'image dégradée par un bruit blanc gaussien à la fois dans les zones de régions homogènes de l'image mais aussi dans les zones de variation rapide de l'image.

D'après les résultats, nous constatons que cette approche de fusion spatio-fréquentielle a permis à l'image finale débruitée de combiner facilement et astucieusement l'efficacité des méthodes de débruitage NLM et DCT, opérant dans deux espaces (spatial et fréquentiel) complémentaires.

Les résultats expérimentaux montrent la performance de l'approche surtout dans le cas où la détection de contours se fait sur l'image débruitée par le filtre NLM et montrent aussi la supériorité de la méthode proposée aux filtres NLM et DCT. Cette supériorité a été démontrée visuellement par la qualité des images obtenues 5.2.2 et quantitativement en terme PSNR (voir les tables :5.VII, 5.VIII, 5.IX et les graphiques 5.38, 5.39, 5.40) et en terme MSE (voir les tables 5.X, 5.XI, 5.XII et les graphiques 5.41, 5.42, 5.43). En moyenne la technique de fusion de ces deux algorithmes de débruitage permet un gain, comparativement à la meilleure des deux méthodes utilisée individuellement, compris approximativement) entre 20% et 30%, en termes de MSE, dépendamment de la variance du bruit blanc Gaussien entachant l'image. Le gain obtenu est presque comparable aux meilleurs techniques de débruitage existant à ce jour pour une complexité d'implémentation beaucoup moins grande. On peut citer BM3D [23] dans le domaine spatial et BayesShrink [17] dans le domaine des ondelettes.

BIBLIOGRAPHIE

- [1] SK Alexander et ER Vrscaj. IFS imaging beyond compression. *Nonlinear Analysis : Theory, Methods & Applications*, 71(12):e1215–e1226, 2009.
- [2] Cesario Vincenzo Angelino, Eric Debreuve et Michel Barlaud. Confidence-based denoising relying on a transformation-invariant, robust patch similarity exploring ways to improve patch synchronous summation. Dans *International Conference on Imaging Theory and Applications*, 2011.
- [3] Noura Azzabou, Nikos Paragios et Frederic Guichard. Image denoising based on adapted dictionary computation. Dans *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 3, pages III–109. IEEE, 2007.
- [4] Jacqueline K Benedetti. On the nonparametric estimation of regression functions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 248–253, 1977.
- [5] Larbi Boubchir. *Approches bayésiennes pour le débruitage des images dans le domaine des transformées multi-échelles parcimonieuses orientées et non orientées*. Thèse de doctorat, Université de Caen, 2007.
- [6] Stéphane Bres, Jean-Michel Jolion et Frank Lebourgeois. *Traitement et analyse des images numériques*. Hermes Science Publications, 2003.
- [7] V Bristanak, P Yip et KR Rao. “discrete cosine and sine transforms“. *Elsevier, Amsterdam*, 2007.
- [8] Thomas Brox et Daniel Cremers. Iterated nonlocal means for texture restoration. *Scale Space and Variational Methods in Computer Vision*, pages 13–24, 2007.
- [9] Thomas Brox, Oliver Kleinschmidt et Daniel Cremers. Efficient nonlocal means for denoising of textural patterns. *IEEE Transactions on Image Processing*, 17(7): 1083–1092, 2008.

- [10] A Buades, M Lebrun et JM Morel. Implementation of the non-local bayes image denoising algorithm. *Image Processing On Line*, 2012.
- [11] Antoni Buades, Bartomeu Coll et J-M Morel. A non-local algorithm for image denoising. Dans *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.
- [12] Antoni Buades, Bartomeu Coll et Jean-Michel Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.
- [13] Antoni Buades, Bartomeu Coll et Jean-Michel Morel. Non-Local Means Denoising. *Image Processing On Line*, 1:208–212, 2011.
- [14] Emmanuel J Candes et David L Donoho. Curvelets : A surprisingly effective non-adaptive representation for objects with edges. *FPO AE 09499-0039*, 1999.
- [15] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.
- [16] Francine Catté, Pierre-Louis Lions, Jean-Michel Morel et Tomeu Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical analysis*, 29(1):182–193, 1992.
- [17] S Grace Chang, Bin Yu et Martin Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE transactions on image processing*, 9(9):1532–1546, 2000.
- [18] Priyam Chatterjee et Peyman Milanfar. Patch-based locally optimal denoising. Dans *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2553–2556. IEEE, 2011.
- [19] Priyam Chatterjee et Peyman Milanfar. Patch-based near-optimal image denoising. *IEEE Transactions on Image Processing*, 21(4):1635–1649, 2012.

- [20] Ronald R Coifman et David L Donoho. *Translation-invariant de-noising*. Springer, 1995.
- [21] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik et Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.
- [22] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik et Karen Egiazarian. A non-local and shape-adaptive transform-domain collaborative filtering. Dans *Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA*, 2008.
- [23] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik et Karen Egiazarian. Bm3d image denoising with shape-adaptive principal component analysis. Dans *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.
- [24] Jérôme Darbon, Alexandre Cunha, Tony F Chan, Stanley Osher et Grant J Jensen. Fast nonlocal filtering applied to electron cryomicroscopy. Dans *Biomedical Imaging : From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 1331–1334. IEEE, 2008.
- [25] Ingrid Daubechies. Discrete sets of coherent states and their use in signal analysis. Dans *Differential Equations and Mathematical Physics*, pages 73–82. Springer, 1987.
- [26] Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 41(7):909–996, 1988.
- [27] Antoon Dauwe, Bart Goossens, Hiêp Quang Luong et Wilfried Philips. A fast non-local image denoising algorithm. Dans *Electronic Imaging 2008*, pages 681210–681210. International Society for Optics and Photonics, 2008.
- [28] Charles Deledalle, Loïc Denis et Florence Tupin. Débruitage non-local itératif fondé sur un critere de similarité probabiliste. *the proceedings of GRETSI, Dijon, France*, 2009.

- [29] Charles-Alban Deledalle, Loïc Denis et Florence Tupin. Iterative weighted maximum likelihood denoising with probabilistic patch-based weights. *IEEE Transactions on Image Processing*, 18(12):2661–2672, 2009.
- [30] Charles-Alban Deledalle, Loïc Denis et Florence Tupin. How to compare noisy patches ? patch similarity beyond gaussian noise. *International journal of computer vision*, 99(1):86–102, 2012.
- [31] Charles-Alban Deledalle, Vincent Duval et Joseph Salmon. Non-local methods with shape-adaptive patches (nlm-sap). *Journal of Mathematical Imaging and Vision*, 43(2):103–120, 2012.
- [32] David L Donoho. Wavelet shrinkage and wvd : A 10-minute tour. Dans *Presented on the International Conference on Wavelets and Applications, Toulouse, France*, 1992.
- [33] David L Donoho et Iain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, pages 425–455, 1994.
- [34] Mohsen Ghazel. *Adaptive fractal and wavelet image denoising*. Thèse de doctorat, Citeseer, 2004.
- [35] Guy Gilboa et Stanley Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028, 2008.
- [36] Alexander Grossmann et Jean Morlet. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM journal on mathematical analysis*, 15(4):723–736, 1984.
- [37] Alfred Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.
- [38] Yousef M Hawwar. *Nonlinear image filtering in the wavelet transform domain*. Hermes Science Publications, 2001.

- [39] Zexuan Ji, Qiang Chen, Quan-Sen Sun et De-Shen Xia. A moment-based nonlocal-means algorithm for image denoising. *Information Processing Letters*, 109(23-24): 1238–1244, 2009.
- [40] Qiyu Jin, Ion Grama et Quansheng Liu. Removing gaussian noise by optimization of weights in non-local means. *arXiv preprint arXiv :1109.5640*, 2011.
- [41] Charles Kervrann et Jérôme Boulanger. Optimal spatial adaptation for patch-based image denoising. *IEEE Transactions on Image Processing*, 15(10):2866–2878, 2006.
- [42] Charles Kervrann et Jérôme Boulanger. Local adaptivity to variable smoothness for exemplar-based image regularization and representation. *International Journal of Computer Vision*, 79(1):45–69, 2008.
- [43] Erwan Le Pennec, Charles Dossal, Gabriel Peyré et Stéphane Mallat. Débruitage géométrique d’images dans des bases orthonormées de bandelettes. Dans *GRET-SI’07*, pages 969–972. GRETSI, Groupe d’Etudes du Traitement du Signal et des Images, 2007.
- [44] Erwan Le Pennec et Stéphane Mallat. Sparse geometric image representations with bandelets. *IEEE transactions on image processing*, 14(4):423–438, 2005.
- [45] M Lebrun, A Buades et JM Morel. Study and analysis of nl-pca. Dans *Image Processing on Line. Ipol. im. Workshop*, 2011.
- [46] Jong-Sen Lee. Digital image smoothing and the sigma filter. *Computer vision, graphics, and image processing*, 24(2):255–269, 1983.
- [47] Yifei Lou, Paolo Favaro, Stefano Soatto et Andrea Bertozzi. Nonlocal similarity image filtering. Dans *International Conference on Image Analysis and Processing*, pages 62–71. Springer, 2009.
- [48] Yifei Lou, Xiaoqun Zhang, Stanley Osher et Andrea Bertozzi. Image recovery via nonlocal operators. *Journal of Scientific Computing*, 42(2):185–197, 2010.

- [49] Cécile Louchet et Lionel Moisan. Total variation as a local filter. *SIAM Journal on Imaging Sciences*, 4(2):651–694, 2011.
- [50] Mona Mahmoudi et Guillermo Sapiro. Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE signal processing letters*, 12(12):839–842, 2005.
- [51] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro et Andrew Zisserman. Non-local sparse models for image restoration. Dans *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2272–2279. IEEE, 2009.
- [52] Arian Maleki, Manjari Narayan et Richard Baraniuk. Suboptimality of nonlocal means on images with sharp edges. Dans *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pages 299–305. IEEE, 2011.
- [53] Stephane G Mallat. A theory for multiresolution signal decomposition : the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.
- [54] Y Meyer. Ondelettes et operateurs. i, hermann. *Paris, France*, 1990.
- [55] M. Mignotte, J. Meunier et J.-P. Soucy. Dct-based complexity regularization for em tomographic reconstruction. *IEEE trans. on Biomedical Engineering*, 55(2): 801–805, 2008. tomographic reconstruction.
- [56] Elizbar A Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.
- [57] Amir Nakib. *Conception de métaheuristiques d’optimisation pour la segmentation d’images. Application à des images biomédicales*. Thèse de doctorat, Université de Paris, 2008.
- [58] Bruno A Olshausen et David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996.

- [59] Bruno A Olshausen et David J Field. Sparse coding with an overcomplete basis set : A strategy employed by v1 ? *Vision research*, 37(23):3311–3325, 1997.
- [60] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.
- [61] Pietro Perona et Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7): 629–639, 1990.
- [62] Cle Pohl et John L Van Genderen. Review article multisensor image fusion in remote sensing : concepts, methods and applications. *International journal of remote sensing*, 19(5):823–854, 1998.
- [63] MB Priestley et MT Chao. Non-parametric function fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 385–392, 1972.
- [64] Murray Rosenblatt et al. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.
- [65] Leonid I Rudin, Stanley Osher et Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D : Nonlinear Phenomena*, 60(1-4):259–268, 1992.
- [66] Joseph Salmon. *Agrégation d’estimateurs et méthodes à patch pour le débruitage d’images numériques*. Thèse de doctorat, Université Paris-Diderot-Paris VII, 2010.
- [67] Sailes K Sengijpta. *Fundamentals of statistical signal processing : Estimation theory*, 1995.
- [68] WF Sun, YH Peng et WL Hwang. Modified similarity metric for non-local means algorithm. *Electronics letters*, 45(25):1307–1309, 2009.
- [69] Tolga Tasdizen. Principal neighborhood dictionaries for nonlocal means image denoising. *IEEE Transactions on Image Processing*, 18(12):2649–2660, 2009.

- [70] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu et Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(1):91–108, 2005.
- [71] Carlo Tomasi et Roberto Manduchi. Bilateral filtering for gray and color images. Dans *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.
- [72] David Tschumperlé et Luc Brun. Non-local image smoothing by applying anisotropic diffusion pde's in the space of patches. Dans *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 2957–2960. IEEE, 2009.
- [73] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel et Gregory Randall. Lsd : A fast line segment detector with a false detection control. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):722–732, 2010.
- [74] Jin Wang, Yanwen Guo, Yiting Ying, Yanli Liu et Qunsheng Peng. Fast non-local algorithm for image denoising. Dans *Image Processing, 2006 IEEE International Conference on*, pages 1429–1432. IEEE, 2006.
- [75] Geoffrey S Watson. Smooth regression analysis. *Sankhyā : The Indian Journal of Statistics, Series A*, pages 359–372, 1964.
- [76] Fengxia Yan, Lizhi Cheng et Silong Peng. A new interscale and intrascale orthonormal wavelet thresholding for sure-based image denoising. *IEEE Signal Processing Letters*, 15:139–142, 2008.
- [77] LP Yaroslavsky. Digital picture processing, volume 9 of springer series in information sciences. *Springer-Verlag, Berlin*, 41:42, 1985.
- [78] Guoshen Yu et Guillermo Sapiro. Dct image denoising : a simple and effective image denoising algorithm. *Image Processing On Line*, 1:292–296, 2011.

- [79] Guoshen Yu, Guillermo Sapiro et Stéphane Mallat. Image modeling and enhancement via structured sparse model selection. Dans *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1641–1644. IEEE, 2010.
- [80] Sebastian Zimmer, Stephan Didas et Joachim Weickert. A rotationally invariant block matching strategy improving image denoising with non-local means. Dans *Proc. 2008 International Workshop on Local and Non-Local Approximation in Image Processing*, pages 135–142, 2008.