الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DR. YAHIA FARES DE MEDEA
FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

Département du Génie Électrique
et de l'Informatique



Mémoire de fin d'étude

Présenté pour l'obtention du Diplôme

de Master en Génie Électrique et informatique

Option **:** Systèmes de télécommunications

THEME

Reconstruction des signaux bidimensionnels à partir de la phase spectrale 2D sans et avec présence du bruit

Proposé et dirigé par :

Mr : MIMEN Bakhti

Préparé et présenté par :
M^{me} : BOUBACHICHE Linda
M^{elle} : YAHIAOUI Fella



Remerciement





DEDICACES





INTRODUCTION





CHAPITER 1 :

Notions de base sur le traitement des images



CHAPITER2 :

L'Algorithme itératif standard de la reconstruction à partir de la phase spectrale

CHAPITER 3 :

Etude d'algorithme améliorée en 2D par segmentation et insertion

CHAPITER 4 :

Etude de reconstruction dans un contexte bruité



Bibliographie



Remerciement

En préambule à ce mémoire, nous souhaitons adresser notre remerciements les plus sincères aux personnes qui nous somme apporté leur aide et qui ont contribué à élaboration de ce mémoire ainsi q s'adressent d'abord à ***ALLAH le tout puissant et à son Prophète MOHAMED*** *(paix et salut sur lui) pour les chances qui me sont offertes pour réaliser ce travail.*

Avant de vous présenter ce rapport, nous tiens à remercier tous ceux qui Nous somment aidé pendant mon stage et mes études à ***Université Dr : Yahia Farés de Médéa****.*

Nous voudrons en particulier remercier au "Mr ***MIMEN Bakhti "*** *Responsable de mémoire, Vous nous avons accordé votre confiance en acceptant de diriger ce mémoire, et de nos sincères remerciements.*

Nous voudrons aussi adresser nos sincères remerciements à tous les Professeurs de ***Département du Génie électrique et informatique p****our leurs enseignements et les cours intéressants qu'ils nous donnent pendant notre scolarité a Université.*

Nous n'oublie pas de remercier aussi tous les personnels de ***2eme Année Master (LMD) (ST)*** *qui nous apporte leur aide. Depuis nos premiers jours dans cet institut, nous allons reçu beaucoup d'aides, de conseils et d'encouragements de mes amis, en particulier ceux de la* ***promotion 2 éme année Master Systèmes de télécommunication*** *(2014-2015).*

Tout cela nos permis de murir chaque jour. Nous les remercie et nous ne pourrais jamais oublier les souvenirs gais et tristes que nous allons passer avec eux durant ces Cinque ans à l'université.

A tous ceux qui de près ou de loin ont apporté leur contribution à la réalisation de travail, nous vous prie de trouver l'expression de nos profondes reconnaissances.

Dédicaces

Que ce travail témoigne de mes respects :

***A mes parents** :*

Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de mes études.

Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers eux.

Je prie le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de moi.

Très particulièrement à Mon Marie "Saidani Lakhdar"

A la famille BOUBACHICHE, AOUGBI et SAIDANI.

Ils vont trouver ici l'expression de mes sentiments de respect et de reconnaissance pour le soutien qu'ils n'ont cessé de me porter.

A tous mes professeurs :

Leur générosité et leur soutien m'oblige de leurs témoigner mon profond respect et ma loyale considération.

*A ma chère binôme **"YAHIAOUI FELLA"***

A tous mes amis surtout :

"Halima, Mazoura, Latifa, Ratiba, Ahleme, Donia Zed"

Et mes collègues :

Ils vont trouver ici le témoignage d'une fidélité et d'une amitié infinie.

BOUBACHICH Linda

Dédicaces

Merci Allah (mon dieu) de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et le bonheur de lever mes mains vers le ciel et de dire "Ya Kayoum".

Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à ma mère …

A mon père, école de mon enfance, qui a été mon ombre durant toutes mes années d'études, et qui a veillé tout au long de ma vie, à m'encourager, à me donner l'aide et à me protéger.

Que dieu les gardes et les protège.

A ma très chère sœur Hanane et son mari "....."

A mon frère Oussama.

A mes amies " Mazoura et DoniaZ"

Et en particulier à ma Cher binôme Linda.

A tous ceux qui me sont chères.
A tous ceux qui m'aiment.
A tous ceux que j'aime.
Je dédie ce travail.

YAHIAOUI Fella

ملخص:

هايس وابنهايم اثبتا انه يمكن إعادة تشكيل الإشارات من الطور الطيفي باستعمال طريقة التكرار وفق شروط معينة.

في عملنا هذا ركزنا على معايير "هايس و ابنها يم" لتطبيق هذه الخوارزمية في حالات إشارات أحادية و ثنائية الأبعاد باستغلال TFD2D وTFD1D, حيث أجريت دراسة مقارنة التي أظهرت بان Algo-Iter_2D هو أكثر كفاءة من Algo-Iter_1D من منظور المقاربة للحل و وقت التنفيذ.

كما أجريت عملية تحسين الخوارزمية لإعادة تشكيل صور ثنائية الأبعاد ذات الحجم الكبير باستعمال تقنية التجزئة إلى قطع فرعية صغيرة وإدراج عنصر ذو قيمة كبيرة, حيث كانت النتائج جيدة باستعمال خوارزميات الثنائية الأبعاد.

من جهة أخرى أجرينا دراسة للخوارزميات الأحادية و الثنائية الأبعاد في وسط مشوش, والتي بينت أن الخوارزميات الثنائية الأبعاد هي الأكثر فعالية من الخوارزميات الأحادية الأبعاد في وجود التشويش.

الكلمات المفتاحية : الطور الطيفي ,السعة الطيفية, تحويل فوريه المنفصل الأحادي البعد TFD1D, تحويل فوريه المنفصل الثنائي البعد TFD2D ,التقارب, خوارزمية التكرار , صورة ثنائية الأبعاد , إشارة أحادية البعد ,اقتطاع, تسلسل, تشويش ابيض مضاف , مربع متوسط الخطأ الموحد, مربع متوسط الخطأ الموحد الإحصائي.

Résumé :

Hayes et Openheim *ont montré que sous certaines conditions, le signal peut être reconstruit à partir de la phase spectrale en utilisant une méthode itérative.*

Dans notre travail, nous avons basé sur les critères de ***« Hayes et Openheim »*** *pour mettre en œuvre cet algorithme dans les cas 1D et 2D en exploitant la TFD1D et la TFD2D, où une étude comparative a été effectuée et qui a montré que Algo-Iter_2D est plus performant que Algo-Iter_1D de point de vue convergence et temps d'exécution.*

Une amélioration de reconstruction 2D a été effectuée par segmentation en sous blocs de taille petite et par insertion d'un élément de valeur très grande. L'implémentation a montré de bons résultats pour le cas 2D.

Une autre étude en présence du bruit a été effectuée, pour les algorithmes 2D et 1D, et qui a montré que, les algorithmes 2D sont plus performants que les algorithmes 1D en présence du bruit.

Mots clés : *Phase spectrale ; Amplitude Spectrale ; TFD1D ; TFD2D ; Convergence ; algorithme itératif ; Image (2D) ; Signal unidimensionnel ; Bruit blanc additif ; EQMN et* $EQMN_{stat}$

Summary:

Hayes and Openheim have shown that under some conditions, the signal can be reconstructed from the spectral phase using an iterative method.

In our work, we based on the criteria of "Hayes and Openheim" to implement this algorithm in cases 1D and 2D by exploiting DFT1D and DFT2D where a comparative study was done and which showed that Algo_Iter_2D performs were betters than Algo-Iter_1D in perspective of the convergence and execution time.

A improvement of 2D reconstruction was done by segmentation into sub-blocks of small size and by inserting an element of great value. The implementation has shown good results for the 2D case.

Another study in the presence of noise was made, for 2D and 1D algorithms, and showed that, 2D algorithms performs were betters than 1D algorithms in presence of noise.

Keywords *: Spectral Phase; Spectral amplitude; DFT1D; DFT2D; Convergence; iterative algorithm;2D Image; One dimensional signal; Additive white noise; NMSE and* $NMSE_{stat}$.

Table de matière

Remerciements

Dédicaces

Résumé

Liste des figures I

Liste des tableaux III

Liste des acronymes IV

Liste des symboles.....V

Table de matière……

Introduction générale.....……1

Chapitre I : Notion de bases sur le traitement des images

1. Introduction ……...3

2. Définition du signal…….....3

3. Différents type des signaux…….4

3.1 Les signaux monodimensionnels……...4

3.2 Les signaux bidimensionnels……4

4. Définition d'une image……...4

4.1 Le pixel (Picture Elément)……4

4.2 L'affichage d'une image……4

4.3 Les différents types d'image……..4

4.4 Les différents formats d'image……..5

4.5 La Luminance……7

4.6 Le Contraste……...7

5. Le bruit……8

5.1 Définition……...8

5.2 Classification de bruit…..8

6. La transformation de Fourier……..9

6.1 Transformée de Fourier en cas des signaux 1D…….9

6.2Transformée de Fourier en cas des signaux 2D……10

7. Conclusion.....11

Chapitre II : L'Algorithme itératif standard de la reconstruction à partir de la phase spectrale

1. Introduction …….....12

2. Les conditions nécessaires et suffisantes pour la reconstruction des signaux…….....12

2.1 Théorème (1)…….....12

2.2 Théorème (2)…….....13

3. Présentation de l'algorithme de Hayes…….....13

3.1 Algoritheme standard 1D de Hayes…….....13

3.2 Algorithme itératif 2D…….....15

4. Mise en œuvre des algorithmes itératifs standards 1D et 2D…….....18

4.1 Algorithme (1D)…….....18

4.2 Algorithme (2D)…….....19

5. Représentation et discussion des résultats de « Algo-Iter_1D» et « Algo-Iter_2D».....22

5.1 Résultats de « Algo-Iter 1D »…….....22

5.1.1Variationdel'$EQMN_{stat}$:$EQMN_{stat}$=f(N,N_{iter}).....22

5.1.2 Variation de temps d'exécution : T_{exe}=f (N, N_{iter})……..... 24

5.2 Résultats de « Algo-Iter_2D »…….....25

5.2.1 Variation de l'EQM_{stat}: $EQMN_{stat}$= f (N, N_{iter})…….....26

5.2.2 Variation de temps d'exécution : T_{exe}=f (N, N_{iter})…….....27

6. Etude comparative de « Algo-Iter_1D »et « Algo-Iter_2D »…….....28

6.1Exemple illustratif……...30

7. Conclusion…….....31

Chapitre III : Etude d'algorithme amélioré en 2D par segmentation et insertion

1. Introduction …….....32

2. Principe de la technique de segmentation et insertion …….....32

3. Mise en œuvre de l'algorithme itératif par segmentation et insertion …….....33

4. Implémentation et discussion des résultats de« Algo-Iter-S_2D»…….....35

4.1 Résultats de « Algo-Iter-S_2D»….....35

4.1.1 Variation de l'$EQMN_{stat}$:$EQMN_{stat}$= f ([NSxNS], N_{iter}) pour [NxN]=[256x256]….....36

4.1.2 Variation temps d'exécution: T_{exe}=f ([NSxNS], N_{iter}) pour [NxN]=[256x256]…….....38

4.1.3 Variation de l'$EQMN_{stat}$:$EQMN_{stat}$= f ([NSxNS], N_{iter}) pour [NxN]=[512x512]…….....41

4.1.4 Variation de temps d'exécution : T_{exe}=f ([NSx NS], N_{iter}) pour [NxN]=[512x512].....42

5. Etude de l'influence de l'élément inséré α sur la convergence de Algo-Iter-S_2D……...45

6. Etude comparative de « Algo-Iter_2D» et «A-Iter-S_2D »…….....48

7. Conclusion…….….. 50

Chapitre IV : Etude de reconstruction en présence du bruit

1. Introduction……..51

2. La reconstruction en présence du bruit……...51

2.1 Reconstruction de signaux 1D ….....51

2.2 Reconstruction de signaux 2D….....52

3. Implémentation des algorithmes itératifs dans un contexte bruité…….....53

3.1Résultats de « Algo-Iter_1D » dans un contexte bruite…….....53

3.2 Résultat d' « Algo-Iter_2D» dans un contexte bruite…….....56

3.3 Résultat de « Algo-Iter-S_2D» dans un contexte bruite…….....60

4. Etude comparative entre les algorithmes standards dans un contexte bruité …….....64

4.1 Etude comparative entre « Algo-Iter_1D » et « Algo-Iter_2D » dans un contexte bruité …….....64

4.2 Etude comparative entre « Algo-Iter_2D » et « Algo-Iter-S_2D » dans un contexte bruité …….....65

5. Conclusion…….....……66

Conclusion générale…….....67

Bibliographie

Annexe(A)

Annexe(B)

Liste des figures

Figure I. 1: *Exemple d'une image en couleur.* 5

Figure I. 2: *Spectre d'un bruit blanc.* 8

Figure II. 1: *Courbe de variation de l'erreur $EQMN_{stat}$ en fonction de N et N_{iter}.* 23

Figure II. 2: *Courbe de variation du temps d'exécution T_{exe} en fonction de N_{iter}.* 25

Figure II. 3: *Courbe de variation de l'erreur $EQMN_{stat}$ en fonction de N et N_{iter} (AI2D).* 27

Figure II. 4: *Courbe de variation du temps d'exécution T_{exe} en fonction de N et N_{iter} (AI2D).* 28

Figure II. 5: *Reconstruction de l'image « légume » à l'aide des algorithmes standard (AI1D et AI2D) pour $N_{iter}=(10^3$ et $10^5)$.* 30

Figure III. 1: *Courbe de variation de l'erreur $EQMN_{stat}$ en fonction de* [NSxNS] *(AI2D).* 37

Figure III. 2: *Courbe de variation de l'erreur $EQMN_{stat}$ en fonction de* N_{iter} *(AI2D).* 37

Figure III. 3: *Courbe de variation du temps d'exécution T_{exe} en fonction de* [NSx NS] *(AI2D).* 39

Figure III. 4: C*ourbe de variation du temps d'exécution T_{exe} en fonction de N_{iter} (AI2D).* 39

Figure III. 5: *Reconstruction de l' image « chat » à l'aide de l'algorithme standard par segmentation (AIS2D) pour Niter= [10^2, 10^4] et (NSxNS)=[32x32,128x128].* 40

Figure III.6: *Courbe de variation de l'erreur quadratique moyenne $EQMN_{stat}$ en fonction de* [NSxNS] *(AI2D).* 41

Figure III.7*: Courbe de variation de l'erreur quadratique moyenne EQMNstat en fonction de N_{iter} (AI2D).* 42

Figure III.8: *Courbe de variation du temps d'exécution T_{exe} en fonction de* [NSx NS] et de N_{iter}. 43

Figure III.9: *Courbe de variation du temps d'exécution T_{exe} en fonction de N_{iter} (AI2D).* 44

FigureIII.10: *Reconstruction de l'image «oiseau » à l'aide de l'algorithme standard par segmentation (AIS2D) pour N_{iter}= [10^2, 10^4] et (NSxNS)=[32x32,128x128].* 45

Figure III.11: *Reconstruction de l'image « animaux » à l'aide des algorithmes standard (AIS2D) pour $N_{iter}=10^2$ et (NSxNS)=[8x8, 128x128].* 48

Figure III.12: *Reconstruction de l'image « rose » à l'aide des algorithmes standard (AI2D et AIS2D) pour $N_{iter}=10^3$ et [NSxNS]=32x32.* 50

Figure IV.1: *Schéma bloc de la procédure de reconstruction 1D dans un contexte bruité.* 52

Figure IV.2: *Schéma bloc de la procédure de reconstruction 2D dans un contexte bruité.* 53

Figure IV. 3: *La variation de l'$EQMN_{stat}$ en fonction de niveau de bruit N_{br} et de N (AI1D).* 55

Figure IV. 4 : *La variation de l'$EQMN_{stat}$ en fonction de la taille de l'image N et de N_{br} (AI1D).* 56

Figure IV. 5: *Reconstruction d'image « femme » à l'aide d'algorithme standard (AI1D) pour $N_{iter}=[10^2, 10^4]$ et $N_{br} = [10^{-5},10^{-1}]$.* 57

Figure IV. 6: *La variation de l'$EQMN_{stat}$ en fonction de niveau de bruit N_{br} et de N (AI2D).* 58

Figure IV. 7: *La variation de l'$EQMN_{stat}$ en fonction de la taille de l'image N et de N_{br} (AI2D).* 59

Figure IV. 8: *Reconstruction d'image «maison » à l'aide de l'algorithme standard (AI2D) pour $N_{iter}=[10^2, 10^4]$ et $N_{br}=[10^{-5},10^{-1}]$.* 60

Figure IV. 9: *La variation de l'$EQMN_{stat}$ en fonction de niveau de bruit N_{br} et de (NSxMS) (AIS2D).* 62

Figure IV. 10: *La variation de l'$EQMN_{stat}$ en fonction de la taille de l'image (NSxNS) et de N_{br} (AIS2D).* 62

Figure IV. 11: *Reconstruction d'image «burj khalifa » AIS2D))* 64

Liste des tableaux

Tableau II.1: *La variation de l'erreur $EQMN_{stat}$ en fonction de N_{iter} et N.* 23

Tableau II.2 : *La variation de temps d'exécution T_{exe} en fonction de N_{iter} et N.* 24

Tableau II.3 : *La variation de l'erreur $EQMN_{stat}$ en fonction de N_{iter} et N (AI2D).* 26

Tableau II.4 : *La variation de temps d'exécution T_{exe} en fonction de N_{iter} et N(AI2D).* 28

Tableau II.5 : *Résultat de comparaison entre AI1D et AI2D pour ($N_{iter}=10^3$;10^5).* 29

Tableau III.1 : *Variation de l'erreur $EQMN_{stat}$ en fonction de [NSxNS] et N_{iter} (AIS2D).* 36

Tableau III.2 : *La variation de temps d'exécution T_{exe} en fonction de N_{iter} et [NSxNS] AI2D).* 38

Tableau III.3 : *La variation de l'erreur quadratique moyenne normalisé $EQMN_{stat}$ en fonction de N_{iter} et* (NSxNS) *(AI2D).* 41

Tableau III.4 : *La variation de temps d'exécution T_{exe} en fonction de N_{iter} et de* (NSx NS) 43

Tableau III.5 : *Variation de l'$EQMN_{stat}$ en fonction de l'élément inséré α.* 46

Tableau III.6 : *Résultat de comparaison entre AI2D et AIS2D pour[NSxNS]=[32x32, 64x64, 128x128] et (NxN)=[256x256]* 48

Tableau III.7 : *Résultat de comparaison entre AI2D et AIS2D pour [NSxNS]=[32x32, 64x64, 128x128] et (NxN)=[512x512].* 48

Tableau IV.1 : *Résultats de la variation de l'$EQMN_{stat}$ en fonction de N et de niveau du bruit blanc N_{br}.* 55

Tableau IV.2 : *Résultats de la variation de l'$EQMN_{stat}$ en fonction de N et de niveau du bruit blanc N_{br}.* 58

Tableau IV.3 : *Résultats de la variation de l'$EQMN_{stat}$ en fonction de (NSxNS) et de niveau du bruit blanc (N_{br}).* 61

Tableau IV.4 : *Résultat de comparaison entre AI1D et AI2D en présence du bruit pour [NxN]=[512x512] et [NSxNS]=[32x32].* 65

Tableau IV.5 : *Résultat de comparaison entre AI2D et AIS2D en présence du bruit pour [NxN]=[512x512] et [NSxNS]=[32x32].* 66

Liste des acronyms :

AI : *Algorithme Itérative (Iterative Algorithm).*

AD : *Algorithme Direct (Direct Algorithm).*

. ***AI1D:*** *Algorithme Itérative monodimensionnel (One-dimensional Iterative Algorithm).*

AI2D: *Algorithme Itérative bidimensionnel (Two-dimensional Iterative Algorithm).*

AIS 2D : *Algorithme Itérative bidimensionnel de Segmentation (Two dimensional iterative Segmentation algorithm).*

AIS1D : *Algorithme Itérative unidimentionnel de Segmentation (one-dimensional iterative Segmentation algorithm).*

BBA: Bruit Blanc Additif (**AWN :** Aditif White Noise).

EQM: *Erreur Quadratique Moyenne. (**MSE :** Mean Square Error).*

EQMN: *Erreur Quadratique Moyenne Normalisé. (**NMSE :** Normalized Mean Square Error).*

$EQMN_{stat}$: *Erreur Quadratique Moyenne Normalisé Statistique. (**$NMSE_{stat}$:** Normalized Mean Square Error statistics).*

IEEE: *Institut of Electrical and Electronics Engineers.*

N_{iter}: *Nombre d'itérations (Number of Iteration).*

NS : *Nombre de ligne et colones des sous blocs de segmentation. (Number of online and colones sub segmentation blocks).*

N_{br} *: niveau du bruit. (Noise level).*

TF : *Transformation de Fourier. (**FT:** Fourier Transform).*

TFI : *Transformation de Fourier Inverse. (**IFT :** Inverse Fourier Transform).*

TFD : *Transformation de Fourier Discrete.(**DFT :** Discrete Fourier Transform).*

TFDI : *Transformation de Fourier Discrete Inverse.(**IDFT :** Inverse Discrete Fourier Transform).*

TFR : *Transformée de Fourier Rapide. (**FFT:** Fast Fourier Transform).*

TFRI : *Transformée de Fourier Rapide Inverse. (**FFTI:** Inverse fast Fourier Transform).*

T_{exe}: *Temps d'execution (Execution Time).*

1D : *Unidimensionnel. (OD : One-Dimensional).*

2D : *Bidimensionnel. (TD : Two-Dimensional).*

Introduction générale

Dans le domaine de traitement du signal, l'analyse de Fourier a joué un rôle très important pour caractériser les différents phénomènes physiques tel que l'analyse des signaux cardiolographique en médecine, la parole et le domaine de traitement d'image.

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information utile.

Plusieurs travaux ont été effectués pour prouver l'importance des informations partielles tel que la phase et l'amplitude spectrales ou d'autre pour la reconstruction du signal original.

HAYES et OPPENHEIM on étudié le problème de la reconstruction de signaux à partir dela phase spectrale sous certaines conditions bien définies, en utilisant deux algorithmes de reconstruction standards ; le premier est de type itératif (AI) et le deuxième est de type direct (AD). [1.2]

Notre projet est consacré à l'étude de la reconstruction des signaux 2D à partir des informations partielles (phase spectrale 2D) en basant sur l'algorithme itératif standard de HAYES [2.5] et la version améliorée 1D par segmentation et insertion. D'où on aura une version amélioré 2D en basant sur la TFD2D standard et avec segmentation (AI2D et AIS2D)

Une étude comparative entre les différents algorithmes sera effectuée en fonction des différents paramètres (taille de l'image originale ; la taille des segments et le niveau de l'élément inséré).

D'autre part, nous allons étudier ces algorithmes en présence du bruit blanc additif (BBA) pour confirmer et étendre les résultats obtenus par ESPY et LIM [7]. L'étude dans un contexte bruité sera effectuée pour montrer l'effet du niveau de bruit qui influe sur la reconstruction. C'est-à-dire, quel est le niveau acceptable du bruit qui ne peut pas dégrader l'image reconstruite.

Le présent document est subdivisé en quatre chapitres :

- ✓ Dans le premier chapitre, on aborde quelques notions de base sur le traitement d'image.

- ✓ Le deuxième chapitre comprend la présentation et la mise en œuvre de l'algorithme itératif standard de reconstruction des signaux 1D et 2D en basant sur la TFD1D et la TFD2D à l'aide d'AI1D et AI2D respectivement.

- ✓ Le troisième chapitre est consacré à la présentation et la mise en œuvre de la technique d'amélioration par segmentation et insertion pour les signaux 2D (AIS2D).

- ✓ Le quatrième chapitre représentera l'étude de reconstruction en présence du bruit (bruit blanc), afin de voir l'effet du bruit et la possibilité d'améliorer les résultats de reconstruction en faisant une étude comparative entre les différents algorithmes dans un contexte bruité.

- ✓ En fin nous terminons ce travaille par une conclusion générale.

CHAPITRE I :

Notions de bases sur le traitement des images

1. Introduction :

Avec la parole, l'image constitue l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec autrui. C'est un moyen de communication universel dont la richesse du contenu permet aux êtres humains de tout âge et de toute culture de se comprendre.

C'est aussi le moyen le plus efficace pour communiquer, chacun peut analyser l'image à sa manière, pour en dégager une impression et d'en extraire des informations précises.

De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinente.

2. Définition du signal :

Un signal est la représentation d'une information (souvent dépendant d'un paramètre, par exemple le temps, mais pas forcément). Cette représentation a un support physique. Ce support sert à transmettre l'information.

Exemples :

- Signal 1D : tel que le signal électrique sortant d'un microphone et représentant d'une information (parole, musique,...)
- Signal 2D : telles que les Images bi-dimensions (penser à l'astronomie)
- signal 3D : tels que le débit d'un fleuve en fonction du temps, les images à trois dimensions I(x,y,z) et les séquences vidéos I(x,y,t)

3. Différents type des signaux :

3.1 Les signaux monodimensionnels (1D) :

Le signal monodimensionnel f(x) est une fonction à une seule variable qui n'est pas forcément le temps. (Exemple : une concentration, une abscisse, ligne d'image, etc.)

3.2 Les signaux bidimensionnels (2D) :

Le signal bidimensionnel f(x, y) est une fonction réelle ou complexe à deux variables x et y qui sont indépendantes. Où l'image est l'exemple concret de représentation des signaux 2D.

4 Définition d'une image :

Une image est une représentation visuelle que l'on fait grâce à la vision. Elle répond à plusieurs critères, tels que : le relief (la profondeur, les plans), la lumière ainsi que l'angle de la prise de vue. Qui influent sur la perception d'une image.

4.1 Le pixel (*Picture Elément)* :

Le pixel est le plus petit élément constitutif d'une image numérique. Le pixel est une unité de surface permettant de mesurer une image numérique. De ce fait, une image de la même taille qu'une seconde image, mais ayant une définition moindre (moins de pixel) sera alors moins net que la seconde.

4.2 L'affichage d'une image :

On peut afficher une image de plusieurs façons différentes : grâce à un tube cathodique, grâce à des cristaux liquide (principe du LCD), grâce au gaz (principe du PLASMA), grâce à une superposition de fines couches organiques (principe de l'OLED), ou bien grâce à un projecteur (principe du 3D).

4.3 Les différents types d'image :

a- ***Image en mode bitmap (noir et blanc) :*** avec ce mode, il est possible d'afficher uniquement des images en deux couleurs : noir et blanc.

b- ***Image en mode niveau de gris*** : il permet d'obtenir différentes valeurs de gris, afin d'afficher des images nuancées.

c- ***Image en couleur :*** Ces images sont en général codées en utilisant le codage des trois couleurs fondamentales (rouge, vert, bleu), ce sont donc des images tri-modales.il existent en 3 modes comme suite :

- ✓ Mode couleurs indexées.
- ✓ Mode couleur RVB.
- ✓ Mode couleur CMJN.

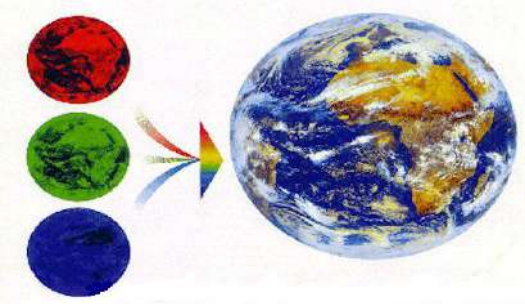


Figure I. 1: *Exemple d'une image en couleur*.

4.4 Les différents formats d'image :

Un format d'image comprend en général un en-tête qui contient des données sur l'image (taille de l'image en pixels par exemple) suivie des données de l'image. La structuration des données est différente pour chaque format d'image.

a- Le format d'image de base BMP :

Le BMP est un format natif de Windows, il permet d'enregistrer des images bitmap en 1, 4, 8 ou 24 bit en mode RVB. Il gère également les palettes pour les couleurs en mode indexées.

b- Le format d'image en astronomie FITS :

Le format d'images FITS ou Flexible Image Transport System est le format d'image le plus utilisé en astronomie. Il est conçu spécifiquement pour des données scientifiques. Il peut porter les extensions de fichier FIT ou FTS entre autres. Une caractéristique importante de ce format est qu'il dispose d'un en-tête de données

textes, au format ASCII, pour inscrire les données sur la prise de vue. Il est disponible en 16 bits et 32 bits, mais le plus utilisé en imagerie CCD est le 16 bits.

c- Le format TIFF :

Il permet de stocker des images de haute qualité en noir et blanc, couleurs RVB, CMJN jusqu'à 32 bits par pixels. Il supporte aussi les images indexées faisant usage d'une palette de couleurs, les calques et les couches alpha (transparence).

d- Le mode RAW :

C'est un format brut qui « code» les images avec un maximum d'information suivant le capteur de l'appareil qui l'a créé. Il permet ensuite de développer numériquement ses photos en les enregistrant en ‘’.tiff’’ avec les réglages souhaitées (températures de couleurs, contrastes...).

e- Le format d'image JPEG pour le Web :

Le format d'image JPEG (avec extension .jpg) est un format développé pour la compression d'images. Il est surtout utilisé pour publier les images sur le Web.

f- Le format PNG :

Il permet de stocker des images en noir et blanc (jusqu'à 16 bits par pixels), en couleurs réelles (True color, jusqu'à 48 bits par pixels) ainsi que des images indexées, faisant usage d'une palette de 256 couleurs. Il offre enfin une couche alpha de 256 niveaux pour la transparence.

g- Le format PSD :

Format natif de Photoshop, c'est un méta-fichier qui peut contenir du bitmap et du vectoriel. La couleur peut être codée sur 8, 16, 24 ou 32 bits, en Noir et Blanc, RVB et CMJN. Il gère la transparence, les couches alpha et peut prendre énormément de poids suivant le nombre de calques utilisés (chaque calque ajouté pèse !).

h- Le format GIF :

C'est un format léger qui peut également contenir des animations. Une image GIF ne peut contenir que 2, 4, 8, 16, 32, 64, 128 ou 256 couleurs parmi 16.8 millions dans sa palette en mode RVB. Elle supporte également une couleur de transparence.

4.5 La Luminance :

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par:

- ✓ Des images lumineuses (brillantes) ;
- ✓ Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir ; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- ✓ Absence de parasites.

4.6 Le Contraste :

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

Si L1 et L2 sont les degrés de luminosité respectivement de deux zones voisines A1 et A2 d'une image, le contraste C est défini par le rapport :

$$C = (L1-L2)/ (L1+L2) \qquad (I.1)$$

Le **contraste** est une propriété intrinsèque d'une image qui quantifie la différence de luminosité entre les parties claires et sombres d'une image. Visuellement il est possible de l'interpréter comme un étalement de l'histogramme de luminosité de l'image.

Pour une image en noir et blanc :

- ✓ Pour un contraste nul : l'image observée est intégralement grise.
- ✓ Pour un contraste maximum : chaque pixel de l'image est soit noir soit blanc.

5 Le bruit :

5.1 Définition :

Selon le dictionnaire de l'IEEE (Institue of Eléctrical and Electronics Engineers), le bruit est définit comme étant « l'ensemble de toutes les perturbations indésirables qui se superposent au signal utile et ont tendance à masquer son contenue ».

5.2 Classification des bruits :

Selon l'analyse mathématique, le bruit est caractérisé par sa densité spectrale et il est généralement classé en deux type suivant sa fréquence : le bruit blanc et le bruit rose.

- ✓ ***bruit rose* :** est un signal aléatoire dont l'intensité de chaque bande de fréquence de taille identique est égale, si on l'analyse dans une échelle logarithmique ; sa densité spectrale est dépendante de la fréquence.

- ✓ ***bruit blanc* :** Le bruit blanc est un signal (ou processus) avec une énergie équivalente par cycle (en **hertz**). Cela se traduit par un spectre « plat » lorsqu'on en trace le diagramme. Il est le plus simple de tous les modèles statistiques de signaux aléatoires ; sa densité spectrale est indépendante de la fréquence.

Dans notre étude, on se base sur l'influence du bruit blanc sur le signal reconstruit à partir de sa phase spectrale.

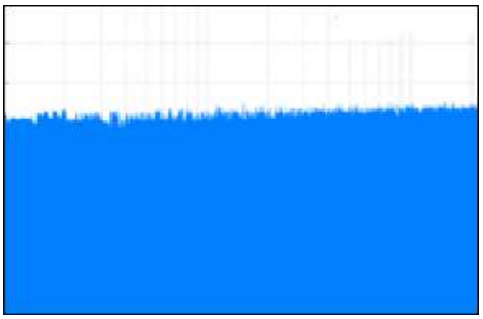


Figure I. 2: *Spectre d'un bruit blanc.*

6 La transformation de Fourier :

La transformation de Fourier est une extension, pour les fonctions non périodiques, du développement en série des fonctions périodiques de Fourier. La transformation de Fourier associe à une fonction intégrable, définie sur l'ensemble des nombres réels ou celui des nombres complexes, une fonction appelée *transformée de Fourier* dont la variable indépendante peut s'interpréter en physique comme la fréquence ou la pulsation.

6.1 Transformée de Fourier 1D : [8]

Il y a deux méthodes de représentation des signaux selon la nature de la variable indépendante choisie. Une représentation temporelle de la forme x(t), et la représentation fréquentielle de la forme X(f).

La relation entre les deux représentations joue un rôle très important en traitement du signal, le signal peut être caractérisé par sa durée, sa période, ou son amplitude. L'utilisation de tels paramètres conduits à sa représentation temporelle, le signal peut être aussi caractérisé par sa bande passante, sa fréquence ou sa phase. L'utilisation de tels paramètres conduit à sa représentation fréquentielle. Où cette représentation fréquentielle dépend des paramètres de la représentation temporelle et qui sont reliées par la transformation de Fourier.

a- La transformée de Fourier (TF) :

La transformée de Fourier peut être exprimée en fonction de $\boldsymbol{f}$ **,** telle que

$$\boldsymbol{X(w) = \int_{-\infty}^{+\infty} x(t) e^{-jwt} dt} \qquad \text{(I.2)}$$

avec $\boldsymbol{W=2\pi f}$*, donc* $\boldsymbol{X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi ft}\, dt}$ (I.3)

b- Transformée de Fourier inverse (TFI):

Le passage de la représentation fréquentiel à la représentation temporelle se fait pas la transformée de Fourier inverse (TF^{-1}) est donnée comme suit :

$$\boldsymbol{x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(w) e^{+jwt}\ dw} \qquad \text{(I.4)}$$

$$\boldsymbol{x(t) = \int_{-\infty}^{+\infty} X(f) e^{+j2\pi t}\ df} \qquad \text{(I.5)}$$

c- La transformée de Fourier discrète (TFD) :

On appelle transformée de Fourier discrète d'une suite de *N* termes *x*(0),*x*(1), . . . ,*x*(*N* −1), la suite de *N* termes *X*(0), *X*(1), . . ., *X* (*N* −1), définis par :

$$\boldsymbol{X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{nk}{N}}} \qquad \text{(I.6)}$$

En pratique, les *N* termes *x*(*n*) peuvent être N échantillons d'un signal analogique échantillonné :

d- la transformée de Fourier discrète inverse (TFDI) :

Elle est définie comme suit :

$$\boldsymbol{x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi \frac{kn}{N}}} \qquad \text{(I.7)}$$

6.2 Transformée de Fourier 2D :[11], [12], [13]

Avant d'envisager l'échantillonnage et le traitement numérique des signaux il est nécessaire de donner l'interprétation des signaux bidimensionnels dans le domaine des fréquences.

a-Transformée de Fourier 2D (TF2D):

La représentation fréquentielle des signaux 2D est l'extension directe de celle des signaux monodimensionnels. La transformée de Fourier F (u, v) d'un signal f (x, y) est :

$$\boldsymbol{F(u,v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x,y) e^{-j(ux+vy)}\, dx\, dy} \qquad \text{(I.8)}$$

Dans certaines présentations la quantité scalaire (u x + v y) est donnée sous la forme d'un produit scalaire $\vec{u}.\vec{v}$; on obtient alors une écriture similaire dans les cas mono et multidimensionnels). Cette formule permet de calculer l'amplitude de la composante du signal f (x, y) à la fréquence spatiale (u, v).

b-Transformée de Fourier 2D Inverse (TF2DI) :

La reconstitution de signal spatial se fait par addition des différentes fréquences spatiales pondérées par les amplitudes complexes ainsi calculées.

$$f(x,y) = \frac{1}{4\pi^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(u,v) e^{j(ux+vy)}\, du\, dv \qquad \text{(I.9)}$$

c-Transformée de Fourier discrète 2D (TFD2D) :

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

$$F(u,v) = \frac{1}{\sqrt{NM}} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) e^{-j2\pi\frac{kn}{N}} e^{-j2\pi\frac{kn}{M}} \qquad \text{(I.10)}$$

d-Transformée de Fourier Discrète 2D Inverse (TFD2DI) :

Elle est définie comme suit :

$$f(x,y) = \frac{1}{\sqrt{NM}} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u,v) e^{-j2\pi\frac{kn}{N}} e^{-j2\pi\frac{kn}{M}} \qquad \text{(I.11)}$$

7 Conclusion :

Nous avons présenté quelques notions de base, qu'on aura besoin au cours des chapitres suivants.

La représentation d'un signal dans les domaines temporal et fréquentiel, nous permet de trouver la relation et l'importance des informations partielles pour la reconstruction du signal original.

Le signal et caractérisé par des paramètres nécessaires dans les domaines temporel et fréquentiel. Dans le domaine fréquentiel **l'amplitude** et la **phase** constituent des paramètres pertinents qui contiennent beaucoup d'informations sur le signal original. La transformée de Fourier est une fonction complexe, qui a pour chaque composante un module et une phase. Le module et la phase sont fonction des éléments élémentaires du signal temporel.

Dans les chapitres qui suivent, nous allons présenter et étudier un algorithme itératif de reconstruction à partir de la phase spectrale (Algorithme de Hayes); car l'information utile se trouve essentiellement dans la phase de la transformée de Fourier.

Chapitre II :

L'Algorithme itératif standard de la reconstruction à partir de la phase spectrale

1. Introduction :

Hayes et Oppenheim ont montré qu'il est possible de reconstruire des signaux à partir de la phase spectrale d'un signal monodimensionnel, en utilisant deux algorithmes différents. L'un est de type direct en basant sur la résolution d'un système d'équations ; et l'autre est de type itératif en basant sur des allers-retours entre les domaines temporel et fréquentiel.

Dans ce chapitre nous allons baser sur l'algorithme de Hayes et Oppenheim, pour effectuer une étude de reconstruction à partir de la phase spectrale en basant sur la transformée de Fourier 1D pour les signaux 1D d'une part et d'autre part à la base de la transformée de Fourier 2D pour les signaux 2D.

2. *Conditions de reconstruction d'un signal à partir de la phase spectrale* :

Le signal est caractérisé par deux fonctions nécessaires, l'amplitude et la phase; ces deux fonctions sont indépendantes.

Les conditions nécessaires et suffisantes de reconstruction à partir de la phase spectrale sont données par deux théorèmes, où la démonstration de ces derniers se trouve en Annexe (A). Le premier théorème aborde le cas où la phase spectrale est connue sur tout le spectre fréquentiel continu, tandis que la deuxième permet la reconstruction des signaux à partir d'échantillons de la phase spectrale du spectre fréquentiel discret. [1].[2].[3].[9]

2.1 Théorème (1) :

Soient x(i) et y(i) deux séquences de même longueur N (finie) dont les transformations en Z n'ont pas de zéros sur le cercle unité ou par paires réciproques conjuguées.

- Si $\varphi_x(\omega) = \varphi_y(\omega), \forall\omega$ alors x(i) = βy(i), avec β une constante positive :

- Si $\tan(\varphi_x(\omega)) = \tan(\varphi_y), \forall\omega$ alors x(i) = βy(i) avec β une constante réelle, avec $\varphi_x(\omega) et\ \varphi_y(\omega)$, représentant respectivement les phases spectrales de X(w) Y(w).

2.2 <u>Théorème (2) :</u>

Soient x(i) et y(i) deux séquences de même longueur N finie et qui sont nulles en dehors de l'intervalle [0, N-1] dont les transformations en Z n'ont pas de zéros sur le cercle unité ou par paire réciproques conjuguées.

- $\varphi_x(\omega) = \varphi_y(\omega)$, à (N-1) fréquences distinctes dans l'intervalle fréquentiel [0, π] alors :

x(i) = β y(i), avec β une constante positive.

- Si $\tan(\varphi_x(\omega)) = \tan(\varphi_y(\omega))$, à (N-1) fréquences distinctes dans l'intervalle fréquentiel [0, π], alors :

x(i) = β y(i), avec β une constante réelle, avec $\varphi_x(\omega)\ et\ \varphi_y(\omega)$, représentent respectivement les phases spectrales de X(z) et Y(z).

Remarque :

Le deuxième théorème est plus intéressant de point de vue pratique, car la reconstruction s'effectue à partir d'un nombre fini d'échantillons de la phase spectrale.

3. Présentation des algorithmes itératifs standards (1D et 2D) :

L'algorithme itératif est un type d'algorithme nécessite un ensemble d'itération pour atteindre la solution désirée. Ce dernier permet la reconstruction des signaux réels **x(i)** satisfaisant les conditions de HAYES à partir de la phase spectrale on faisant des allers et retours entre les domaines temporels et fréquentiels.

3.1 Algorithme standard 1D de Hayes : [1].[2].[3].[4].[9]

Soit **x(i)** un signal réel à support temporel limité, $\boldsymbol{i} \in [\mathbf{0}, \boldsymbol{N-1}]$, et $\boldsymbol{\varphi_x}(\mathbf{k})$ sa phase spectrale calculée sur **2N** points de l'intervalle fréquentiel, $\boldsymbol{k} \in [\mathbf{0}.\mathbf{2}\boldsymbol{N-1}]$**.**

Le signal monodimensionnel $\mathbf{x_I(i)}$ reconstruit à l'I^{ème} itération est donné par l'expression suivante :

$$\mathbf{x_I(i)=R(x_{I-1}(i))=T(F'\ (x_{I-1}(i))} \qquad \text{(II.1)}$$

où :

R : l'application de reconstruction à partir de la phase spectrale.

F' : l'application de détermination de la transformation de Fourier inverse après incorporation de la phase spectrale $\boldsymbol{\varphi_x(k)}$.

T : l'application de troncation.

T et **F'** sont définis comme suit :

$$T(x_{I-1}) = \begin{cases} x_{I+1}(i) & \text{pour } 0 \leq i \leq N-1 \\ 0 & \text{ailleurs.} \end{cases} \quad \text{(II.2)}$$

et $\mathbf{F'} = (F^{-1})_{2N}\,\Phi F_2$ (II.3)

avec :

- $(F^{-1})_{2N}$ et F_{2N} sont respectivement les transformations de Fourier discrète inverse et directe.
- Φ : est l'application d'incorporation de la phase spectrale :

Si $\boldsymbol{x_{I+1}(k) = |x_{I+1}(k)|.exp(j\varphi_x(k))}$; **pour k**$\in [\mathbf{0}, \mathbf{2N-1}]$.

Sous une forme explicite, l'algorithme est décrit comme suit :

✓ ***1ére étape*** : On pose :

$$\boldsymbol{Y_0(k) = \exp(j\varphi_x(k))} \quad \text{(II.4)}$$

i.e: on a supposé que l'amplitude spectrale égale à l'unité.

✓ ***2éme étape : On*** calcule la transformation de Fourier inverse $\mathbf{y_0(i)}$ comme suit :

$$\boldsymbol{y_0(i) = F_{2N}^{-1}.Y_0(k)} \quad \text{(II.5)}$$

Avec : i$\in [0,2N-1]$ $et\ k \in [0,2N-1]$.

✓ ***3éme étape :***

$$\boldsymbol{y_I(i) = y_0(i)} \quad \text{(II.6)}$$

Donc, le chemin de notre algorithme est comme suit :

a. Déduction de la séquence d'estimation $h_i(n)$ à partir de la séquence $l_i(n)$ par troncation ; avec :

$$x_I(i) = T(y_i) = \begin{cases} y_I(i) & pour: 0 \leq i \leq N-1 \\ 0 & ailleurs. \end{cases} \quad \text{(II.7)}$$

b. Le test d'arrêt de l'algorithme :

- Le calcul de l'erreur quadratique moyenne normalisée EQMN si le nombre d'itération fixé est atteint ; et l'arrêt de la procédure.

Alors $\tilde{x}(i)$ est le signal reconstruit du signal original x(i).

L'erreur de reconstruction est : $\boldsymbol{EQMN} = \frac{EQM}{max\,((x_i - \tilde{x}_i)^2)}$

où $\boldsymbol{EQM} = \frac{1}{N}\sum_i^N (x_i - \tilde{x}_i)^2$

- Si non, on procède comme suit :

-détermination de la transformée de Fourier de la séquence $x_I(i)$ comme suit :

$\mathbf{X_I\,(k)} = F_{2N}\,[x_I(i)]$ (II.8)

-Détermination de $Y_I(k)$, pour incorporation de la phase spectrale connu $\varphi_x(k$ dans la séquence $X_I(k)$, avec :

$\mathbf{Y_{I+1}(k)} = |XI(k)\exp(j\varphi_x(k))|$ (II.9)

-Détermination de la transformée de Fourier inverse $l_{i+1}(n)$ comme suit :

$\mathbf{y_{I+1}(i)} = (F^{-1})_{2N}\, Y_{I+1}(k)$ (II.10)

- Posons : $y_I(i)=y_{I+1}(i)$ et retour à la première étape.

3.2 Algorithme itératif (2D) :

Soit $\boldsymbol{x(i,j)}$ un signal bidimensionnelle à support temporelle limité, $i \in [0, N-1] et\ j \in [0, M-1]$, et φ_x (l,c) une matrice qui represente sa phase spectrale calculée sur (2Nx2M) points du domaine fréquentiel, $l \in [0,2N-1] et\ c \in [0,2M-1]$.

Le signal bidimensionnelle reconstruit $x_I(i,j)$ à l'I^{ème} itérations est donné par l'expression suivante :

$\mathbf{x_I(i,j) = R\,(x_{I+1}(i,j)) = T\,[TF2'\,(x_{I-1}(i,j)]}$ (II.11)

Avec :T et TF2' sont deux fonction défini comme suit :

$$T(x_{i-1}) = \begin{cases} \mathbf{x_{I+1}(i,j)} & \mathbf{pour\ 0 \le i \le N-1} \\ & \mathbf{et\ \ 0 \le j \le M-1} \\ \mathbf{0} & \mathbf{ailleurs}. \end{cases}$$ (II.12)

et $\mathbf{TF2' = (TF2^{-1})_{2N,2M}\,\Phi\,(TF2)_{2N,2M}}$ (II.13)

où :

- **R** : l'application de reconstruction à partir de la phase spectrale.
- **TF2'** : l'application de détermination de la transformation de Fourier bidimensionnelle inverse après incorporation de la phase spectrale φ_x(l,c) ; Avec :
  - ✓ $(TF2^{-1})_{2N,2M}$ et $(TF2)_{2N,2M}$ sont respectivement les transformations de Fourier bidimensionnelle rapide inverse et direct.
  - ✓ **Φ** : est l'application d'incorporation de la phase spectrale.

Si $x_{i+1}(l,c) = |x_{i+1}(l,c)|.exp(j\varphi_x(l,c))$ (II.14)

Pour $l \in [0, 2N-1]$ *et* $c \in [0, 2M-1]$.

- **T** : l'application de troncation.

Sous une forme explicite, l'algorithme itératif en 2D est décrit comme suit :

1ière étape :

On pose : $Y_0(l,c) = A * exp(j\varphi_x(l,c))$ (II.15)

i.e: on a supposé que l'amplitude spectrale est une matrice unité ; avec : A[0 :N-1, 0 :M-1]=1

2ième étape :

On calcule la transformation de Fourier rapide bidimensionnelle inverse $(FFT2)^{-1}$ $y_0(i,j)$ comme suit :

$$y_0(i,j) = (FFT2)^{-1}_{2N,2M} \cdot Y_0(l,c) \quad \text{(II.16)}$$

avec : i∈ $[0, N-1]$; $j \in [0, M-1]$ ***et*** $l \in [0, N-1]$; $c \in [0, M-1]$.

3ième étape :

$$y_I(i,j) = y_0(i,j) \quad \text{(II.17)}$$

Donc, le chemin de notre algorithme est comme suit :

a. Déduction de la matrice d'estimation $\mathbf{x_I(i,j)}$ à partir de la matrice $y_I(i,j)$ par troncation ; avec :

$$\mathbf{x_I(i,j)=T\ (y_I)}=\begin{cases} y_I(i,j) & pour: 0 \le i \le N-1 \\ & et \quad 0 \le j \le M-1 \\ 0 & ailleurs. \end{cases} \qquad \text{(II.18)}$$

b. Le test d'arrêt de l'algorithme :

- Le calcul de l'erreur quadratique moyenne normalisée EQMN si le nombre d'itération fixé est atteint ; donc l'arrêt de la procédure. Où $\tilde{x}(i,j)$ est l'image 2D reconstruite de celle d'origine x (i,j).

$$\boldsymbol{EQMN} = \frac{EQM}{max\ ((x(i,j)-\tilde{x}(i,j))^2)} \qquad \text{(II.19)}$$

où : $$\boldsymbol{EQM} = \frac{1}{N*M}\sum_{i=1}^{N}\sum_{j=1}^{M}[(x(i,j)-\hat{x}\ (i,j)]^2 \qquad \text{(II.20)}$$

- **Si non**, on procède comme suit :

  - ✓ détermination de la transformée de Fourier rapide bidimensionnelle de la matrice $\mathbf{x_I(i,j)}$ comme suit :

$$\mathbf{X_I\ (l,c)}=(FFT2)_{2N,2M}\ [x_I(i,j)] \qquad \text{(II.21)}$$

  - ✓ Détermination de $Y_I(l,c)$, pour incorporation de la phase spectrale connu $\varphi_x(l,c)$ dans la matrice $x_I(l,c)$, avec :

$$\boldsymbol{Y_{I+1}(l,c)} = |x_I(L,C)|.\exp(j\varphi_x(l,)) \qquad \text{(II.22)}$$

  - ✓ Détermination de la transformée de Fourier rapide bidimensionnelle inverse $y_{I+1}(i,j)$ comme suit :

$$\mathbf{y_{I+1}(i,j)}= (FFT2^{-1})_{\ 2N,2M}\ Y_{I+1}(l,c) \qquad \text{(II.23)}$$

  - ✓ Posons **: $\mathbf{y_I(i,j)=y_{I+1}(i,j)}$** et retour à « 1 » (première étape).

4. Mise en œuvre des algorithmes itératifs standards 1D et 2D :

4.1 Algorithme (1D) :

Pour la mise en œuvre d'algorithme itératif ce dernier nécessite le traitement de deux parties :

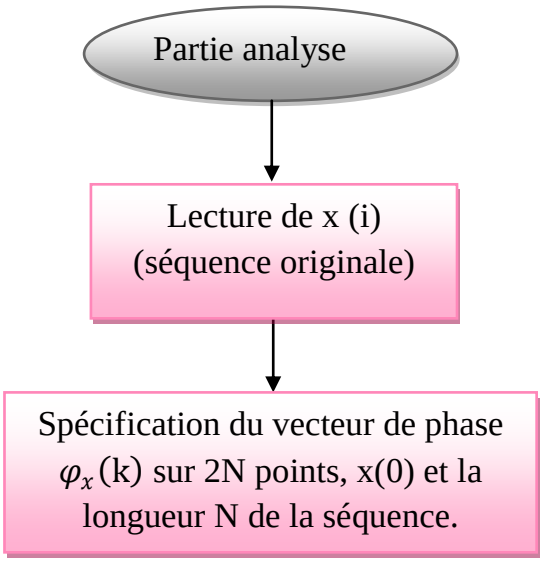
- ***Partie analyse :*** elle consiste à extraire du signal d'origine les informations nécessaires pour la reconstruction. Cette partie contient deux étapes :

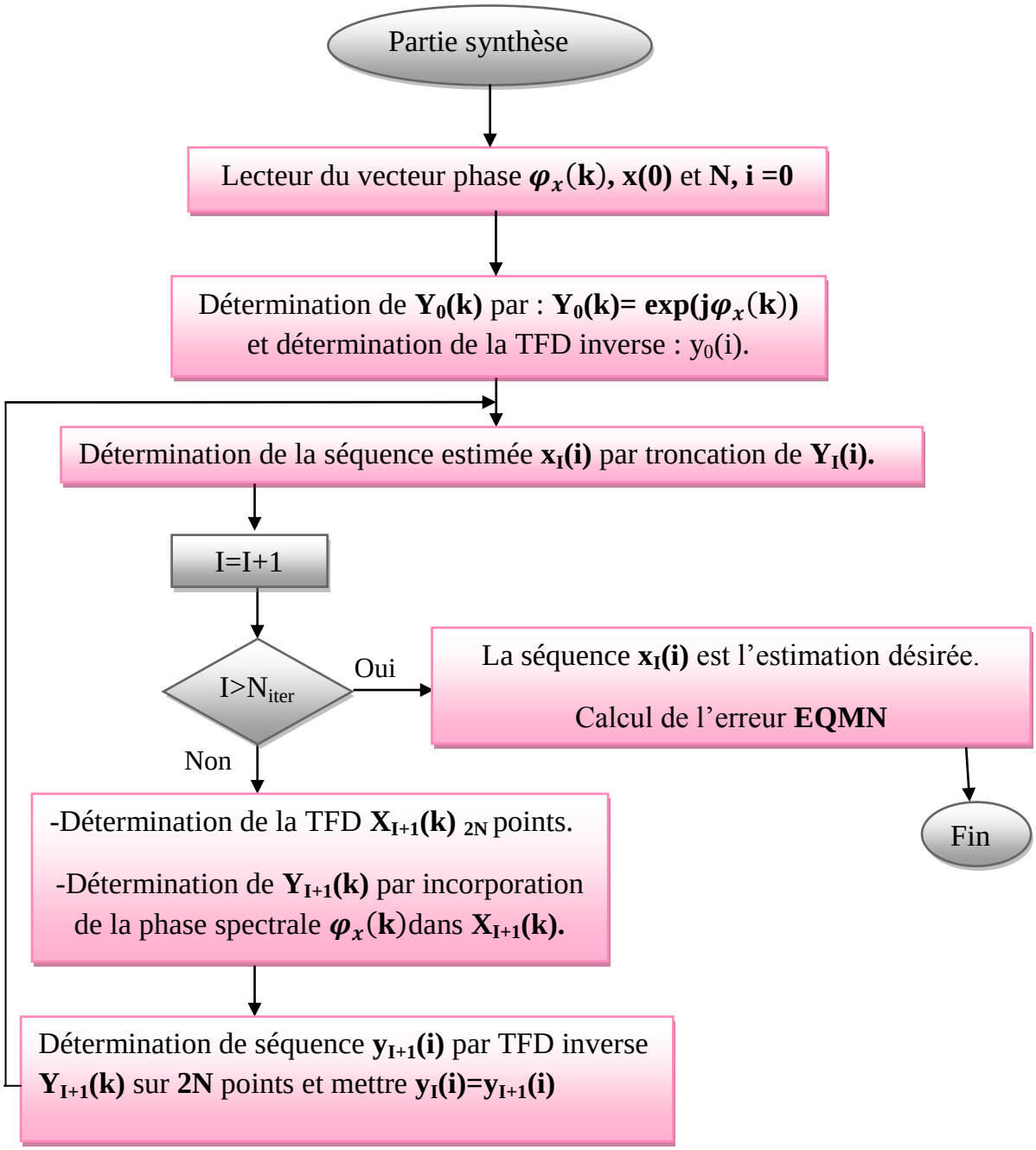
1. lecteur de séquence d'origine x(i) de support [0, N-1]
2. calcul de la transformée de Fourier discrète (TFD) X(k) et déduction du vecteur phase $\varphi_x(k)$ avec k∈ $[0, N-1]$.

- ***Partie synthèse :*** c'est la partie de la reconstruction du signal.

Cette partie consiste à effectuer des itérations pour reconstruire la séquence désirée à partir de la phase spectrale obtenue du signal original en faisant des aller et retours entre les domaines temporels et fréquentiels à l'aide de la TFD et de la TFDI en suivant les mêmes étapes citées dans la section (II.2.1).

Organigramme :





4.2 Algorithme (2D) :

Pour la mise en œuvre d'algorithme itératif 2D ce dernier nécessite le traitement de deux parties :

- ***Partie analyse :*** elle consiste à extraire du signal d'origine les informations nécessaires pour la reconstruction. cette partie contient deux étapes :

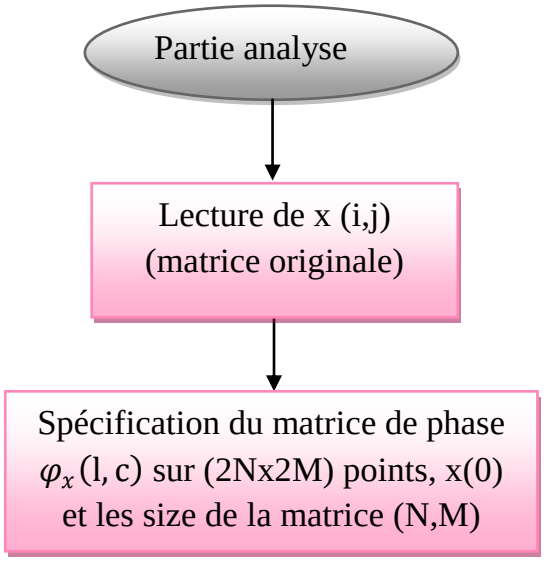
1. lecteur de matrice d'origine x(i,j) de support : [0, N-1]et [0,M-1].

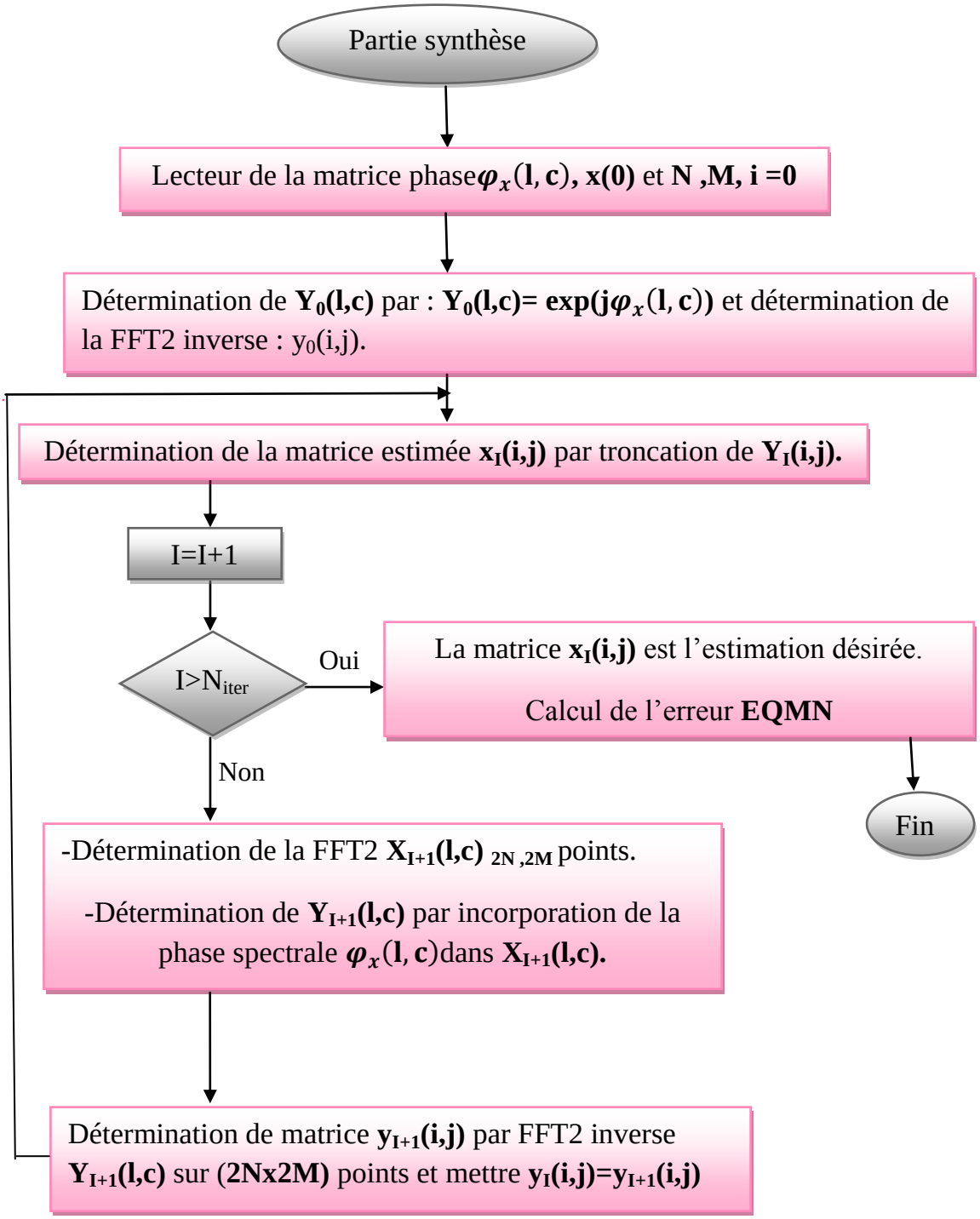
2. Calcul de la transformée de Fourier rapide 2D, (FFT2) X(l ,c) et déduction du matrice phase $\varphi_x(l, c)$ avec l∈ $[0, N-1]$ et c ∈ $[0, M-1]$.

❖ ***Partie synthèse :*** c'est la partie de la reconstruction du signal.

Cette partie consiste à effectuer des itérations pour reconstruire la matrice désirée à partir de la phase spectral obtenue du signal original en faisant des aller et retours entre les domaines temporels et fréquentiels à l'aide de la FFT2 et de la FFT2 inverse en suivant les même étape citées dans la section (II.2.2).

Organigramme :




Partie synthèse
Lecteur de la matrice phase $\varphi_x(l,c)$, x(0) et N ,M, i =0
Détermination de $Y_0(l,c)$ par : $Y_0(l,c)= \exp(j\varphi_x(l,c))$ et détermination de la FFT2 inverse : $y_0(i,j)$.
Détermination de la matrice estimée $x_I(i,j)$ par troncation de $Y_I(i,j)$.
I=I+1
I>N_iter
Oui
La matrice $x_I(i,j)$ est l'estimation désirée.
Calcul de l'erreur EQMN
Fin
Non
-Détermination de la FFT2 $X_{I+1}(l,c)$ $_{2N\ ,2M}$ points.
-Détermination de $Y_{I+1}(l,c)$ par incorporation de la phase spectrale $\varphi_x(l,c)$dans $X_{I+1}(l,c)$.
Détermination de matrice $y_{I+1}(i,j)$ par FFT2 inverse $Y_{I+1}(l,c)$ sur (2Nx2M) points et mettre $y_I(i,j)=y_{I+1}(i,j)$

5. Présentation et discussion des résultats de « AI1D » et « AI2D » :

Dans cette implémentation on s'intéresse à l'étude de l'effet du nombre d'itérations ainsi que la taille du signal original (1D et 2D) sur la variation de l'erreur quadratique moyenne normalisée ainsi que le temps nécessaire à la reconstruction.

5.1 Résultats de « AI1D » :

Pour cette implémentation, nous avons étudié la variation de l'erreur quadratique moyenne normalisée ainsi que le temps d'exécution en fonction du nombre d'itérations pour des images de taille [NxN] variable. Pour chaque taille (NxN), on utilise 10 images, où $EQMN_{stat}$ est obtenue par moyenne des EQMN obtenues pour chaque image, telle que :

$$\boldsymbol{EQMN} = \frac{EQM}{max\,((x_i - \tilde{x}_i)^2)}$$

$$\text{où} \qquad \boldsymbol{EQM} = \frac{1}{N}\sum_i^N (x_i - \tilde{x}_i)^2$$

L'erreur quadratique moyenne normalisée statistique, $EQMN_{stat}$, est la moyenne des EQMN pour chaque signal des $\mathbf{N_{signaux}}$ reconstruits.

$$\boldsymbol{EQMN_{stat}} = \frac{1}{N_{Signaux}}\sum_k^{N_{Signaux}} EQMN(k) \qquad \text{(II.24)}$$

Pour Algo-Iter(1D), on transforme la matrice d'image en un vecteur soit ligne par ligne ou colonne par colonne. Et à la fin de reconstruction on procède à la concaténation des lignes (ou des colonnes) reconstruites pour obtenir l'image résultats.

Le nombre des itérations est : $N_{iter}=10^1$, $N_{iter}=10^2$, $N_{iter}=10^3$, $N_{iter}=10^4$, $N_{iter}=10^5$.

Les résultats de reconstruction sont représentés dans les tableaux suivant, où **N** est la longueur de séquence.

5.1.1 Variation de l'$EQMN_{stat}$: $\mathbf{EQMN_{stat}=f(N,N_{iter})}$

Le tableau suivant représente le résultat de la variation de l'erreur quadratique moyenne normalisée $\mathbf{EQMN_{stat}}$ en fonction de la longueur de séquence à reconstruire **N** et du nombre d'itération $\mathbf{N_{iter}}$.

Tableau II. 1: *La variation de l'erreur $EQMN_{stat}$ en fonction de N_{iter} et N.*

N \ N_{iter}	10^1	10^2	10^3	10^4	10^5
16	0.83497	0.35378	0.14718	0.11911	0.01094
32	0.86660	0.50170	0.18936	0.12966	0.02876
64	0.89832	0.55566	0.22048	0.13628	0.03296
128	0.96094	0.65737	0.42186	0.18936	0.04448
256	0.97356	0.79026	0.42596	0.26387	0.06143
512	0.98849	0.82723	0.55871	0.44321	0.26051
1024	0.99574	0.86454	0.59327	0.56671	0.36745
2048	1	0.87923	0.74252	0.61596	0.40097
4096	1	0.94408	0.82290	0.70087	0.48045

La figure suivante présente la variation de l'erreur quadratique moyenne normalisée ***EQMN$_{stat}$*** en fonction du nombre d'itération pour différentes taille d'image.

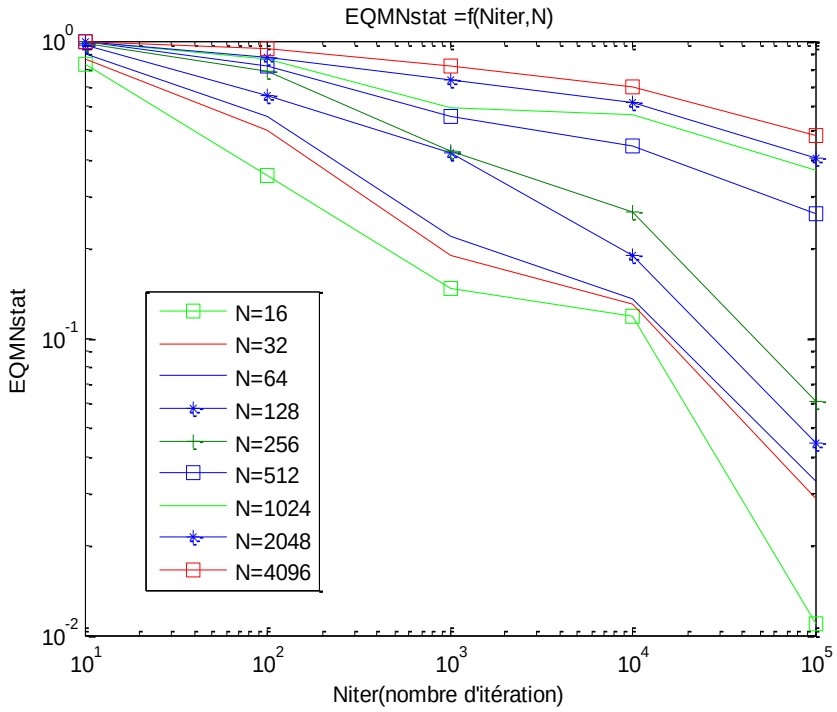


Figure II. 1: *Courbe de variation de l'erreur $EQMN_{stat}$ en fonction de N et N_{iter}.*

D'après les courbes de la figure précédente, on constate que :

- l'erreur **$EQMN_{stat}$** diminue en fonction d'augmentation du nombre d'itérations, et donc il y a une bonne convergence.
- pour un nombre d'itérations fixé, l'erreur **$EQMN_{stat}$** augmente avec la taille du signal 1D.

- nous tirons une importante remarque, c'est que la reconstruction de l'image de grande taille nécessite un très grand nombre d'itérations pour assurer une bonne convergence.

5.1.2 Variation de temps d'exécution : $T_{exe}=f(N,N_{iter})$

Le tableau suivant représente les résultats de la variation de temps d'exécution $\mathbf{T_{exe}}$ en fonction du nombre d'itération $\mathbf{N_{iter}}$ et en fonction de la taille de l'image à reconstruire **N**.

Tableau II. 2:*Lla variation de temps d'exécution T_{exe} en fonction de N_{iter} et N.*

N \ N_{iter}	10^1	10^2	10^3	10^4	10^5
16	0.01424	0.02624	0.14747	1.10197	8.46784
32	0.01782	0.03132	0.19484	1.36433	8.48540
64	0.01868	0.03972	0.24281	2.08532	18.4842
128	0.02268	0.06669	0.42375	3.74038	26.8881
256	0.03162	0.11119	0.82929	6.60085	52.4989
512	0.04008	0.19720	1.53947	14.4006	139.578
1024	0.04744	0.24355	2.30891	19.1036	147.950
2048	0.06498	0.46503	4.03546	28.9801	180.442
4096	0.21679	0.82386	5.94847	37.1227	329.581

La figure suivante présente la variation de temps d'exécution $\mathbf{T_{exe}}$ en fonction du nombre d'itération pour différentes taille d'image.

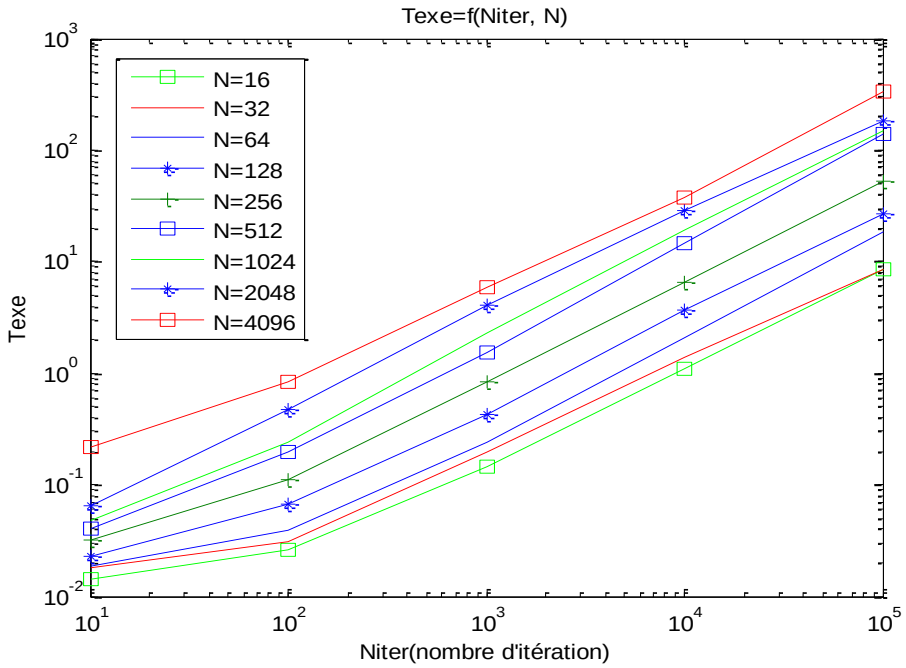


Figure II. 2: *Courbe de variation du temps d'exécution T_{exe} en fonction de N_{iter}.*

D'après la figure(II.2), on remarque que le temps d'exécution T_{exe} augmente avec la taille N du signal 1D et le nombre d'itération N_{iter}.

Après dépassement d'une certaine longueur de la séquence à reconstruire, il y'aura un temps très prohibitif à cause de la complexité des opérations de la transformation de Fourier. C'est pour cela, une opération de segmentation est nécessaire pour réduire la complexité des transformations de Fourier directe et inverse ; et donc, on procède à la reconstruction par partie.

5.2 Résultats de « AI2D » :

Pour cette implémentation, nous avons étudié la variation de l'erreur quadratique moyenne normalisée ainsi que le temps d'exécution en fonction du nombre d'itérations pour des images de taille N variable. Pour chaque taille N, on utilise 10 images, où $EQMN_{stat}$ est obtenue par moyenne des EQMN obtenues pour chaque image ; avec :

$$\boldsymbol{EQMN_{stat}} = \frac{1}{N_{Images}} \sum_{k}^{N_{Images}} EQMN(k) \qquad \text{(II.25)}$$

$$\text{où : } \boldsymbol{EQMN} = \frac{EQM}{max\,((x(i,j)-\tilde{x}(i,j))^2)}$$

$$\text{et } \boldsymbol{EQM} = \frac{1}{N*M} \sum_{i=1}^{N} \sum_{j=1}^{M} [(x(i,j) - \hat{x}\,(i,j)]^2$$

Pour AI2D, on a fait la reconstruction du matrice d'image complètement son transformation de celle par un vecteur (ligne par ligne ou colonne par colonne) . Et à la fin de reconstruction on recouvrer l'image résultats.

Le nombre des itérations est : $N_{iter}=10^1$, $N_{iter}=10^2$, $N_{iter}=10^3$, $N_{iter}=10^4$, $N_{iter}=10^5$.

Les résultats de reconstruction sont représentés dans les tableaux suivant, où **N** est la taille de l'image.

5.2.1 Variation de l'$EQMN_{stat}$: EQMN = f(N,N_{iter})

Le tableau suivant représente le résultat de la variation de l'erreur quadratique moyenne normalisé **$EQMN_{stat}$** en fonction de **N** des **5** images reconstruites et du nombre d'itération **N_{iter}.**

Tableau II. 3 : *La variation de l'erreur $EQMN_{stat}$ en fonction de N_{iter} et N (AI2D).*

N_{iter} / NxN	10^1	10^2	10^3	10^4	10^5
16x16	0.960761	0.337488	0.007686	0.0004641	0.000462
32x32	0.961545	0.517465	0.091804	0.003968	0.003963
64x64	1	0.591484	0.1949691	0.0143645	0.006046
128x128	1	0.636129	0.245424	0.030798	0.00874
256x256	1	0.6790	0.3596	0.0965	0.0190276

La figure ci-dessus présente la variation de l'erreur quadratique moyenne normalisée statistique **$EQMN_{stat}$** en fonction du nombre d'itération pour différentes taille des images.

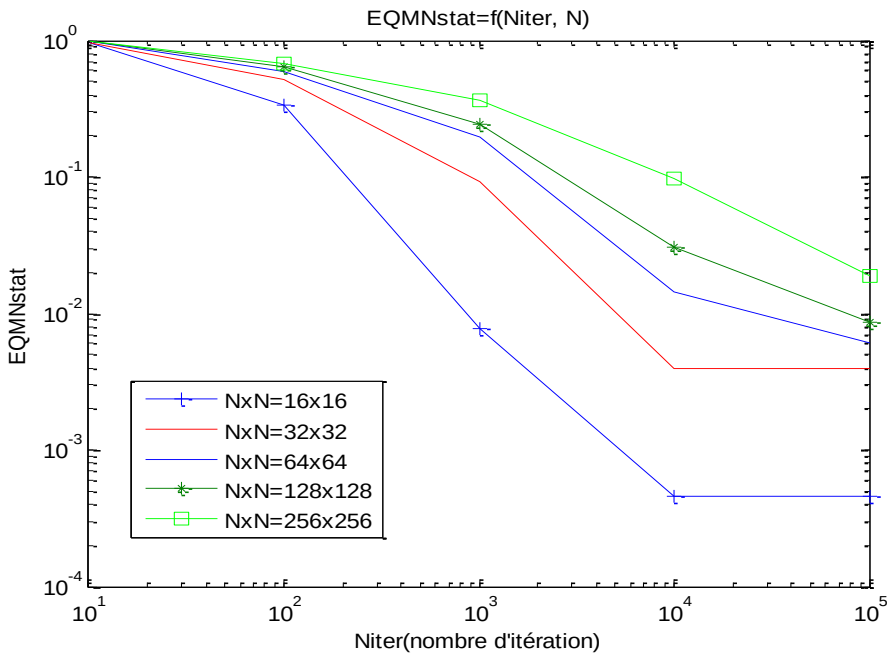


Figure II. 3: *Courbe de variation de l'erreur $EQMN_{stat}$ en fonction de N et N_{iter} (AI2D).*

D'après les courbes de la figure ci-dessus, on constate que :

- l'erreur $\boldsymbol{EQMN_{stat}}$ diminue en fonction d'augmentation du nombre d'itérations, et donc il y a une bonne convergence.
- pour un nombre d'itérations fixé, l'erreur $\mathbf{EQMN_{stat}}$ augmente avec la taille de l'image 2D.
- nous tirons une importante remarque, c'est que la reconstruction de l'image de grande taille nécessite un très grand nombre d'itérations pour assurer une bonne convergence.

5.2.2 Variation de temps d'exécution : $\mathbf{T_{exe}=f(N,N_{iter})}$

Le tableau suivant représente les résultats de la variation de temps d'exécution $\mathbf{T_{exe}}$ en fonction du nombre d'itération $\mathbf{N_{iter}}$ et en fonction de la taille de l'image à reconstruire **N**.

Tableau II. 4: *La variation de temps d'exécution T_{exe} en fonction de N_{iter} et N(AI2D).*

N_{iter} / NxN	10^1	10^2	10^3	10^4	10^5
16x16	0.050070	0.302134	2.46493	18.7895	170.112
32x32	0.131046	0.968294	8.12006	63.2929	542.35
64x64	0.339506	2.54256	13.9057	111.71	1097.66
128x128	1.26506	7.79076	53.1582	503.228	5015.625
256x256	4.60822	28.7663	251.522	2590.07	24461.5

La figure suivant nous permettre de la clarification des résultats du tableau précédent.

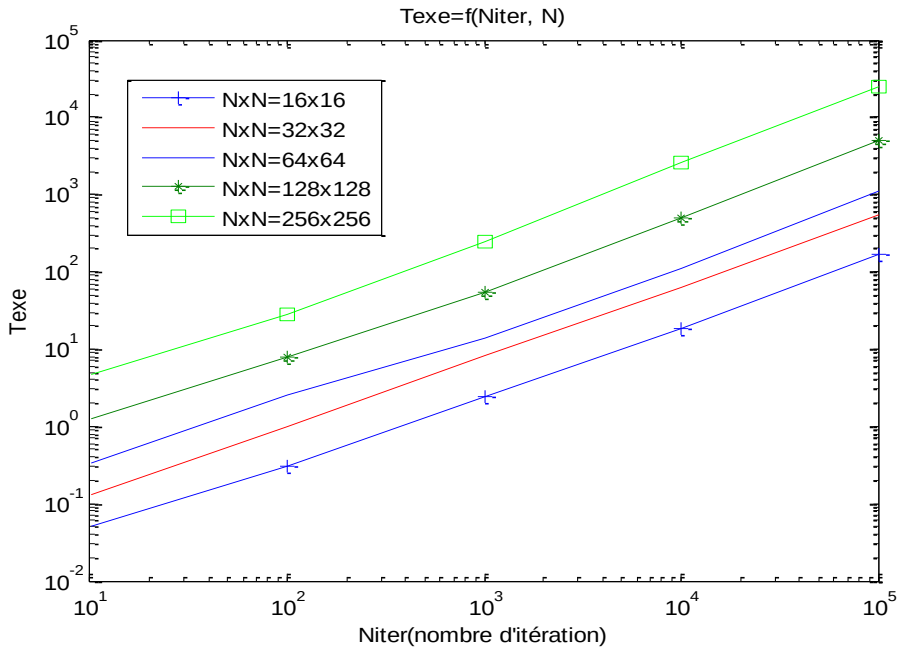


Figure II. 4: *Courbe de variation du temps d'exécution T_{exe} en fonction de N et N_{iter} (AI2D).*

On remarque dans cette figure, le temps d'exécution T_{exe} augmente avec la taille de l'image N et le nombre d'itérations N_{iter}.

6. Etude comparative de « AI1D »et « AI2D » :

Une étude comparative entre « AI1D» et « AI2D» a été effectuée pour la reconstruction des 10 images de taille NxN, où N ∈[16x16, 32x32, 64x64] et Niter = [10^3 ; 10^5] :

Tableau II. 5: *Résultat de comparaison entre A1D et AI2D pour ($N_{iter}=10^3$;10^5).*

	AI1D Niter=10^3		AI1D Niter=10^5		AI2D Niter=10^3		AI2D Niter=10^5	
NxN	**T_{exe} (sec)**	**$EQMN_{stat}$**	**T_{exe}(sec)**	**$EQMN_{stat}$**	**T_{exe}(sec)**	**$EQMN_{stat}$**	**T_{exe}(sec)**	**$EQMN_{stat}$**
16x16	0.829295	0.421968	52.4989	0.061432	2.305	0.015516	157.37	0.005707
32x32	2.308910	0.673275	147.9509	0.167456	10.183	0.0301	688.61	0.00103
64x64	5.94847	0.832904	329.581	0.580463	14.031	0.51598	1088.1	0.000216

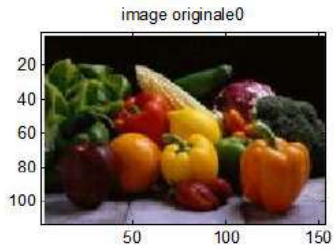
D'après ce tableau de la comparaison, nous remarquons bien que :

- ✓ le temps d'exécution est très petit pour AI1D que AI2D, à cause de la complexité des calculs effectués par la FFT2D pour le même nombre d'itérations.
- ✓ Pour le même nombre d'itérations, l'erreur quadratique moyenne normalisée $EQMN_{stat}$ obtenue par AI2D est meilleure par rapport à celle de AI1D.
- ✓ Mais, on remarque que l'erreur $EQMN_{stat}$ obtenue par AI2D pour $N_{iter}=10^3$ est très petite que celle obtenue par AI1D pour $N_{iter}=10^5$; de même le temps d'exécution devient plus réduit par AI2D que AI1D pour ces deux valeurs de nombres d'itérations.

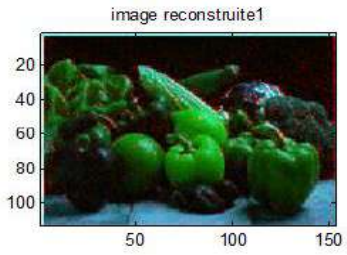
  Donc, on peut dire que AI2D est meilleur que AI1D, de point de vu temps d'exécution et erreur.

6.3 Exemple illustratif :

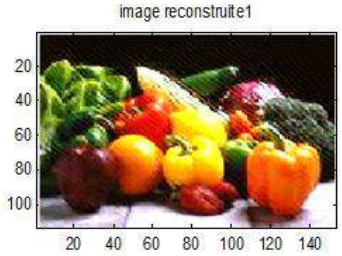
Les deux algorithmes AI1D et AI2D sont limités à la reconstruction des images de taille NxN ≤ 256x256. Un exemple a été effectué en reconstruisant une image de taille (153X113) pixels à l'aide de AI1D et AI2D, la figure suivante représentent les résultats obtenus de cette application.



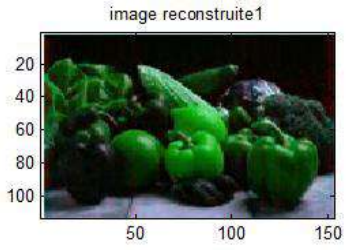
a-L'image originale « légume » (153*113) pixels.



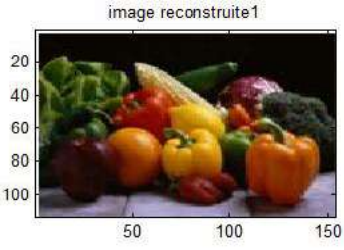
b-Image reconstruite à l'aide de«AI1D » avec : EQMN=0.587721 et T_{exe}=36.533sec ; $N_{iter}=10^3$.



c-Image reconstruite à l'aide de «AI2D » avec : EQM=0.067918 et T_{exe}=85.857sec ; $N_{iter}=10^3$.



d-Image reconstruite à l'aide de «AI1D» avec : EQMN=0.213524 et T_{exe}=3631.5sec ; $N_{iter}=10^5$



e-Image reconstruite à l'aide de «AI2D » avec : EQMN=0.0002462 et T_{exe}=8318.4sec ; $N_{iter}=10^5$.

Figure II. 5: *Reconstruction de l'image « légume » à l'aide des algorithmes standard (AI1D et AI2D) pour $N_{iter}=(10^3$ et $10^5)$.*

7. *Conclusion :*

Donc d'après les résultats obtenus, on constate que la reconstruction des signaux dépend fortement du nombre d'itérations et de la taille du signal 1D ou 2D. De plus, l'algorithme de Hayes et Openheim est décrit pour la reconstruction des signaux 1D en utilisant la TFD-1D. D'autre part, nous avons fait une extension vers l'utilisation en 2D en utilisant la TF2D. Nous pouvons conclure que AI2D est plus performant que AI1D pour la reconstruction des images avec une erreur EQMN plus petite et un temps d'exécution plus réduit.

La reconstruction des signaux 1D et 2D de tailles très élevés (NxN)≥(256x256), pause un gros problème de points de vue temps d'exécution et de point de vu espace mémoire. Pour remédier à ce problème de reconstruction, nous allons présenter dans le chapitre suivant une technique de segmentation et insertion pour la reconstruction des signaux 2D.

CHAPITRE III :

Etude d'algorithme amélioré par segmentation et insertion

1. Introduction :

Suite au chapitre précédent, on a constaté que les algorithmes ont été consacré à l'étude des performances de reconstruction des signaux à partir de la phase spectral, et après implémentation nous a affirmé que les deux algorithmes donnent de très bons résultats pour les signaux de petites taille (NxN ≤ 64x64), par contre elles sont moins efficaces pour les signaux de très grandes tailles (N xN≥ 256x256).

Pour remédier au problème de reconstruction des images de taille très grande (NxN ≥ 256x256), on va étudier dans ce chapitre l'utilisation de la segmentation en sous-blocs de tailles réduits, en exploitant l'élément inséré suivant des facteurs différents.

2. Principe de la technique de segmentation et insertion : [9]

Pour la reconstruction d'une image de très grande taille satisfaisante les conditions de HAYES [1]. On propose la technique suivante :

On effectue la reconstruction par segmentation de l'image en sous bloc de petites tailles. La reconstruction s'effectuera sur les images séparément, et la concaténation de ses sous-blocs reconstruites nous donne l'image désirée.

Le problème qui se pose lors de la reconstruction de chaque sous bloc possède un facteur multiplicatif propre. Pour remédier à cet inconvénient, nous insérons un élément identique non nul comme premier élément à toutes les sous bloc obtenues par segmentation.

Dans cette technique, nous supposons que les sous bloc d'image satisfont aux conditions de HAYES.

Soit **x (i, j)** une matrice réelle de longueur (**N**x**N)** très grande. La méthode proposée consiste à faire la reconstruction de la matrice **x (i, j)** en reconstruisant les sous matrice (bloc) $\boldsymbol{x_{si}(i,j)}$ de taille [**(NS-1)** x **(NS-1)]** à partir de leurs phases spectrales correspondantes $\boldsymbol{\varphi_{si}(i,j)}$, où **NS << N**.

3. *Mise en œuvre de l'algorithme itératif par segmentation et insertion :*[9]

La procédure de la reconstruction en utilisant la segmentation et l'insertion se composes de deux parties principales, partie analyse et partie synthèse.

A. Partie analyse :

Les étapes de cette partie sont :

1. La décomposition de la matrice **x (i, j)** de taille (**NxN)** en **h** sous matrice de tailles [**(NS-1)** x **(NS-1)]**, ou **NS << N**, d'où :

x(i, j)=[$x_{s1}, x_{s2}, \dots, x_{si}, \dots, x_{sh}$] **(III.1)**

2. L'insertion d'un élément $\alpha \neq 0$, comme premier élément de chaque sous bloc, avec :

$$x_{si} = [\alpha, x_{si}] \qquad \textbf{(III.2)}$$

3. Déduction de la phase spectrale φ_{si} des sous bloc $x_{si}(i,j)$ à l'aide de la transformée de Fourier rapide bidimensionnel de chaque sous bloc nous nommerons par φ_{si} le matrice contenant la phase de toutes les sous matrice :

$$\varphi_x = [\varphi_{s1}, \varphi_{s2}, \dots \dots \dots, \varphi_{si}, \dots, \varphi_{sh}] \qquad \textbf{(III.3)}$$

B. Partie synthèse :

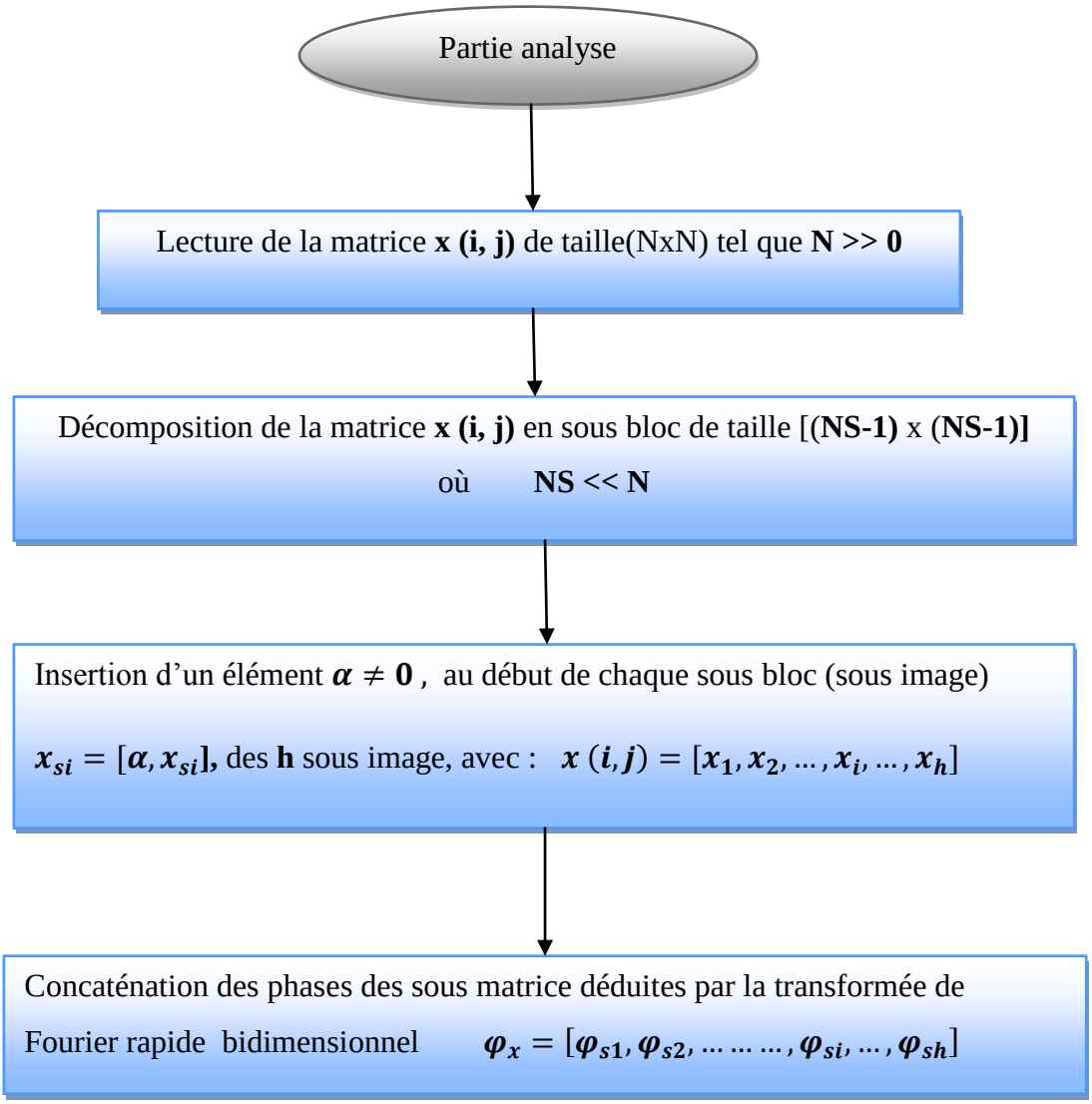
Cette partie est composée des étapes suivantes :

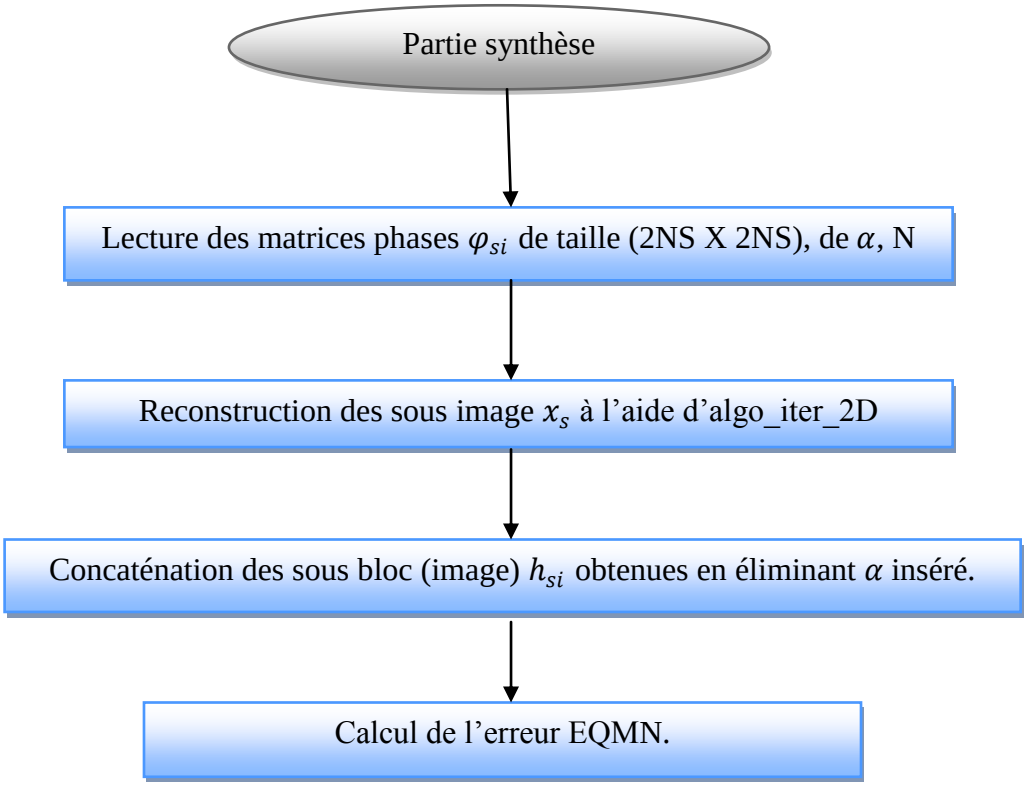
1. Connaissant la matrice phase φ_x, la taille (**NS** x**NS)** des sous bloc et l'élément inséré $\alpha \neq 0$, nous pouvons reconstruire chaque sous image séparément pour obtenir celle qui lui correspond $x_{si}(i,j)$ on utilisant un algorithme standard **(Algo_Iter_2D)** déjà étudie précédemment.
2. Nous éliminons le premier élément inséré α de chaque sous matrice pour déduire la matrice générale désirée **x (i, j)** comme suit :

$$x\,(i,j) = [x_1, x_2, \dots, x_i, \dots, x_h] \qquad \textbf{(III.4)}$$

3. Calcul de l'erreur quadratique moyenne EQMN.

Organigramme





4. Implémentation et discussion des résultats de« AIS2D» :

Dans cette implémentation on s'intéresse à l'étude de l'effet du nombre d'itérations ainsi que la taille des sous blocs sur la variation de l'erreur quadratique moyenne normalisée ainsi que le temps nécessaire à la reconstruction.

4.1 Résultats de « AIS2D» :

Pour cette implémentation, la mise en œuvre de AIS2D se faire en reconstruisant des images de taille (**NxN)** variable, tel que (NxN)= [256x256 et 512x512] pour différentes taille [**NSxNS**] des sous bloc ; Pour chaque taille N, on utilise 5 images, où $\boldsymbol{EQMN_{stat}}$ est obtenue par moyenne des ***EQMN*** de chaque image; voir la page(25)

$$\boldsymbol{EQMN_{stat}} = \frac{1}{N_{Images}} \sum_{k}^{N_{Images}} \boldsymbol{EQMN}(k)$$

Pour AIS2D, on a fait la reconstruction des sous bloc d'image l'une par l'autre et à la fin de reconstruction on procède à la concaténation des sous blocs d'image reconstruits chacune dans sa position pour obtenir l'image globale reconstruite.

La reconstruction à été effectuée pour divers nombres d'itérations :

$$N_{iter}=10^1,\ N_{iter}=10^2,\ N_{iter}=10^3,\ N_{iter}=10^4$$

Les résultats de reconstruction sont représentés dans les tableaux suivant, où **(NSxNS)** est la taille de les sous bloc d'image.

4.1.1 Variation de l'$EQMN_{stat}$: $EQMN_{stat}$ = f ([NSxNS], N_{iter}) pour [NxN]=[256x256] :

Le tableau suivant représente le résultat de la variation de l'erreur quadratique moyenne normalisée $\mathbf{EQMN_{stat}}$ en fonction de la taille des sous bloc d'image à reconstruire **(NSxNS)** et du nombre d'itération $\mathbf{N_{iter}}$.

Tableau III. 1: *Variation de l'erreur $EQMN_{stat}$ en fonction de* [NSxNS] *et N_{iter} (AIS2D).*

Résultat de la reconstruction pour 5 image de taille N=256x256				
N_{iter} / NSxNS	10^1	10^2	10^3	10^4
8x8	1	0.9111696	0.9111696	0.9111696
16x16	1	0.041263	0.041263	0.041263
32x32	1	0.017593	0.000209046	0.000209046
64x64	1	0.004234	0.000022718	0.000022718
128x128	1	0.00250374	0.000006277	0.000005265

Les figures suivant illustrent les résultats du tableau précédent :

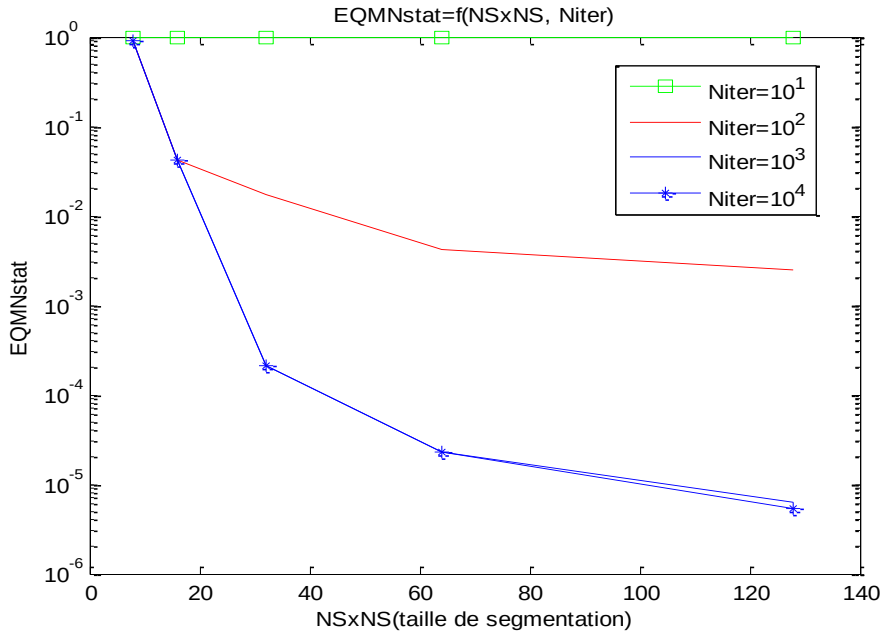


Figure III. 1: *Courbe de variation de l'erreur $EQMN_{stat}$ en fonction de* [NSxNS] *(AI2D).*

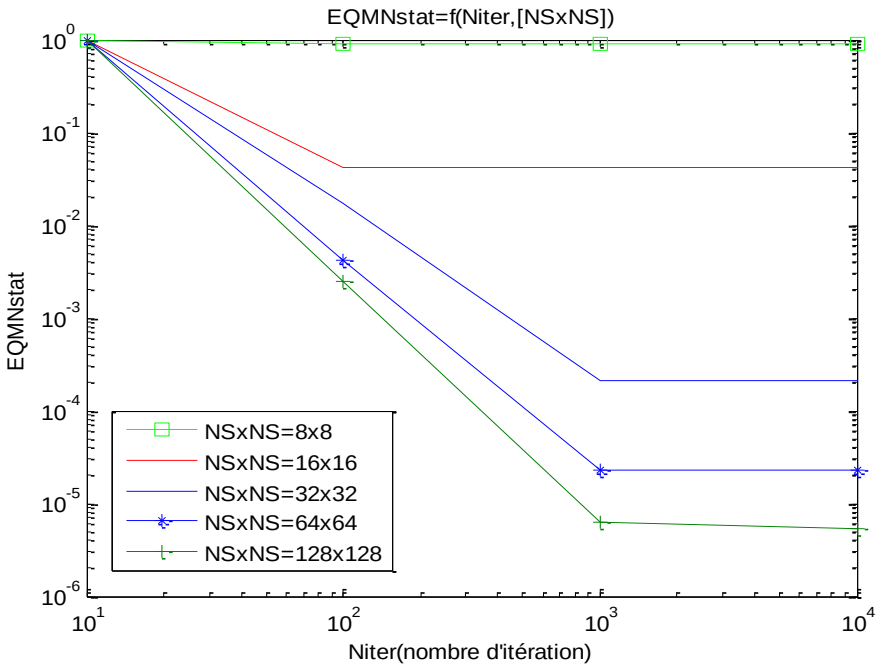


Figure III. 2: *Courbe de variation de l'erreur $EQMN_{stat}$ en fonction de* N_{iter} *(AI2D).*

D'après les résultats du tableau précédent et celle de la figure ci-dessus, nous constatons que l'erreur de la reconstruction dépend principalement de la taille [**NSxNS**] des sous bloc et du nombre d'itération, d'une autre manière :

- pour un nombre d'itération N_{iter} =10, l'erreur de la reconstruction $\mathbf{EQMN_{stat}}$ est fixe (≈1), mais pour N_{iter} >10 l'EQMN diminue quand la taille [NSxNS] des sous bloc augmente.
- pour une taille [NSxNS] fixe, l'erreur $\mathbf{EQMN_{stat}}$ diminue lorsque le nombre d'itération $\mathbf{N_{iter}}$ augmente.
- la reconstruction d'une image de grande taille segmentée en sous blocs de taille [(NSxNS) ≥ (32x32)] donne de bons résultats de reconstruction (erreur très faible pour un nombre d'itérations réduits) par rapport à celle segmentée en sous blocs de taille [(NSxNS) ≤ (32x32)].

4.1.2 Variation temps d'exécution : Txe=f ([NSxNS], Niter) pour [NxN]=[256x256] :

Le tableau suivant représente les résultats de la variation de temps d'exécution $\mathbf{T_{exe}}$ en fonction du nombre d'itération $\mathbf{N_{iter}}$ et de la taille des sous bloc d'image à reconstruire (**NSxNS)**.

Tableau III. 2: *La variation de temps d'exécution T_{exe} en fonction de N_{iter} et* [NSxNS] *(AI2D).*

Résultat de la reconstruction pour 5 image de taille N=256x256				
N_{iter} / NSxNS	10^1	10^2	10^3	10^4
8x8	9.91468	75.13074	681.639	6655.54
16x16	6.16262	49.8322	423.184	4317.92
32x32	5.54264	39.87016	410.313	3694.62
64x64	3.8874	19.6358	187.673	1857.768
128x128	3.58002	18.8938	181.633	1661.1

La figure suivant illustre les résultats du tableau précédent :

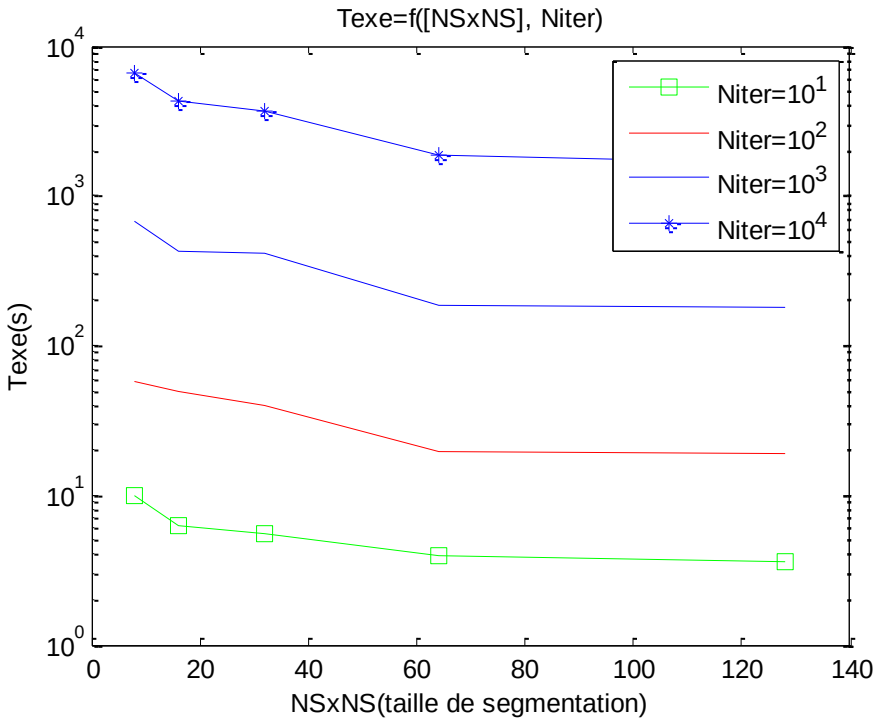


Figure III. 3: *Courbe de variation du temps d'exécution T_{exe} en fonction de* [NSx NS] *(AI2D).*

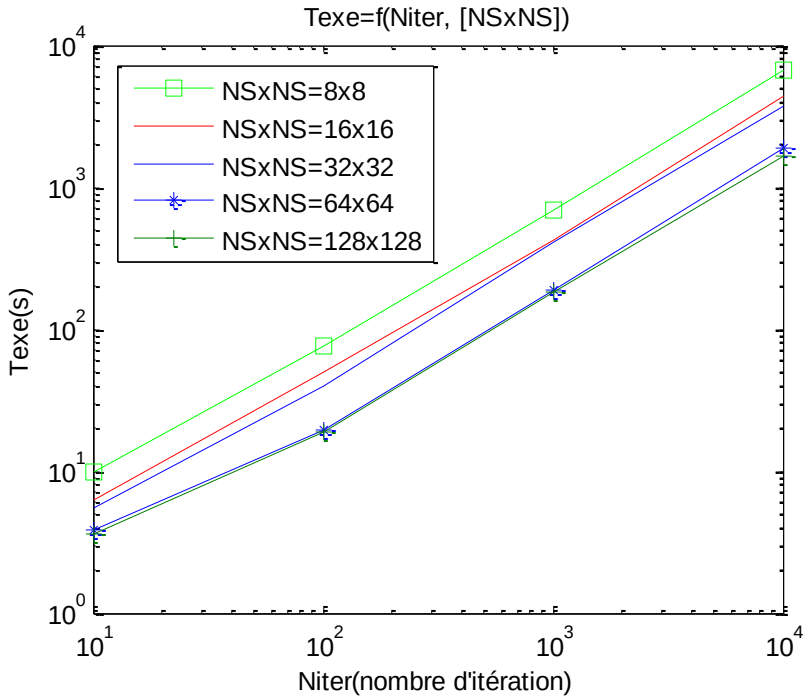
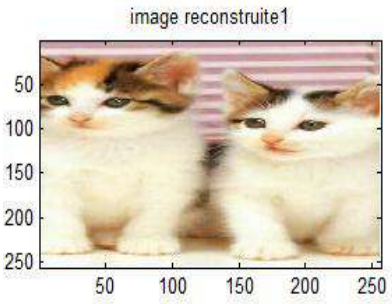


Figure III. 4: *Courbe de variation du temps d'exécution T_{exe} en fonction de N_{iter} (AI2D).*

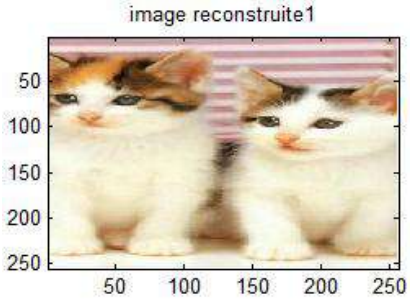
D'après les résultats obtenus dans les figures ci-dessus nous remarquons que le temps d'exécution diminue quand la taille [NSxNS] des sous blocs augmente, mais il augmente en fonction du nombre d'itération (N_{iter}).

Exemple illustratif :

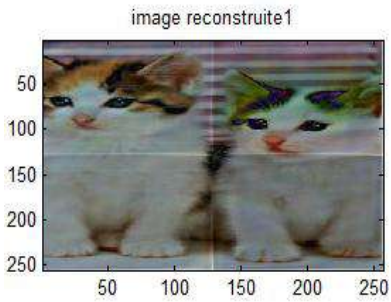
L'exemple suivant illustre les résultats obtenus dans la partie (4.1) en appliquant AIS2D sur une image de taille (NxN)=[256x256] pour des tailles des sous blocs différentes et de nombres d'itérations différents, comme les montrent la figure suivante :



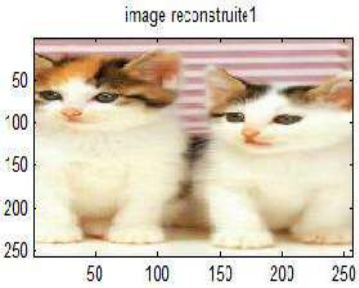
A-Image originale "chat " de taille (256x256) pixels



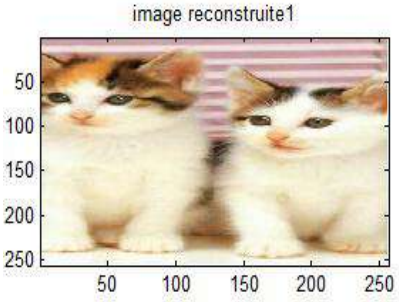
B-Image reconstruite par "AIS2D",avec :
(NSxNS)=(32x32) , N_{iter}=100 ; α=100.max(β)
EQMN= 0.0009516 et T_{exe}=39.3039s



C-Image reconstruite par "AIS2D",avec :
(NSxNS)=128x128 , N_{iter}=100 ; α=100.max(β)
EQMN= 0.0003112 et T_{exe}=20.504s



D-Image reconstruite à l'aide de"AIS2D"avec :
(NSxNS)=32x32 , N_{iter}=10000 ; α=100.max(β) ;
EQMN= 0.000047 et T_{exe}=3716.6s



E-Image reconstruite à l'aide de "AIS2D"avec :
(NSxNS)=128x128 et N_{iter}=10000 ;
α=100.max(β) ; EQMN= 0.0000024 et T_{exe}=1852.9s

Figure III. 5: *Reconstruction de l'image « chat » à l'aide de l'algorithme standard par segmentation (AIS2D) pour N_{iter}= [10^2, 10^4] et (NSxNS)=[32x32,128x128].*

4.1.3 Variation de l'$EQMN_{stat}$: EQMN= f ([NSxNS], N_{iter}) pour [NxN]=[512x512] :

Le tableau suivant représente le résultat de la variation de l'erreur quadratique moyenne normalisé **$EQMN_{stat}$** en fonction de **(NSxNS)** et du nombre d'itération **N_{iter}**.

Tableau III. 3: *La variation de l'erreur quadratique moyenne normalisé $EQMN_{stat}$ en fonction de N_{iter} et* (NSxNS) *(AI2D).*

Résultat de la reconstruction pour 5 images de taille N=512x512				
N_{iter} / NSxNS	10^1	10^2	10^3	10^4
8x8	1	0.92056	0.92056	0.92056
16x16	1	0.03100	0.03100	0.03100
32x32	1	0.00585	0.000139	0.000139
64x64	1	0.00036	0.000007	0.000007
128x128	1	0.00022	0.0000018	0.0000005

Les figures ci-dessous illustrent résultats du tableau précédent.

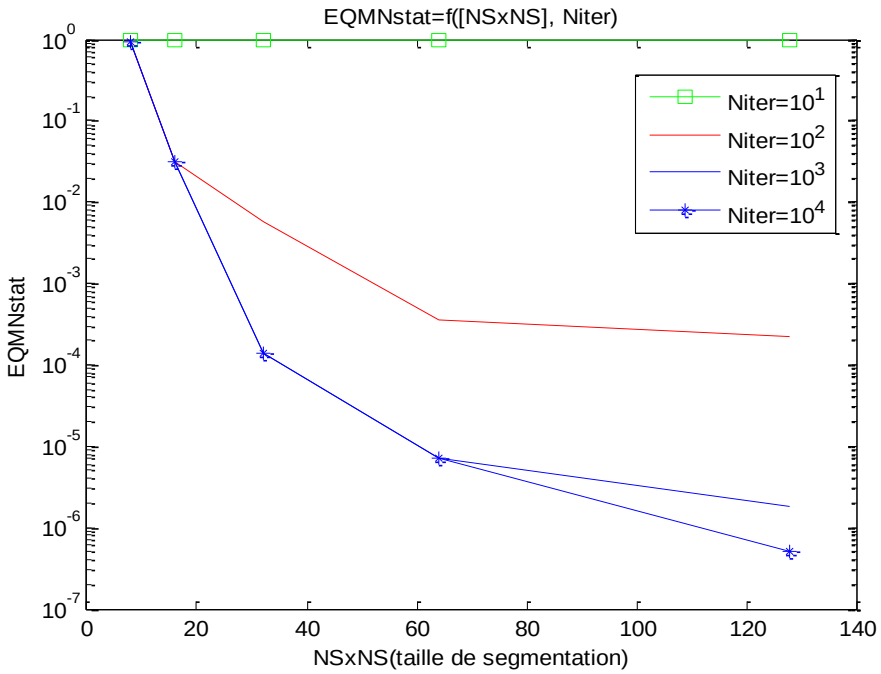


Figure III. 6: *Courbe de variation de l'erreur quadratique moyenne $EQMN_{stat}$ en fonction de* [NSxNS] *(AI2D).*

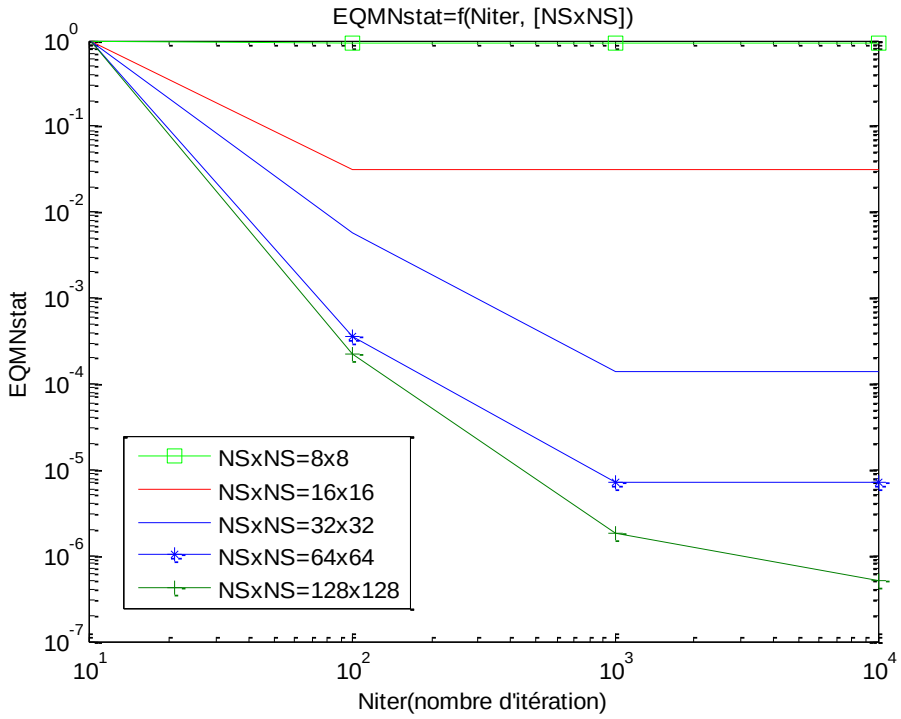


Figure III. 7: *Courbe de variation de l'erreur quadratique moyenne EQMNstat en fonction de N_{iter} (AI2D).*

D'après les résultats du tableau précédent et celle de la figure ci-dessus, nous constatons que l'erreur de la reconstruction dépend principalement de la taille [**NSxNS**] des sous bloc et du nombre d'itération, d'une autre manière :

- pour un nombre d'itération N_{iter} =10, l'erreur de la reconstruction $\mathbf{EQMN_{stat}}$ est fixe (≈1), mais pour N_{iter} >10 l'EQMN diminue quand la taille [NSxNS] des sous bloc augmente.
- pour une taille [NSxNS] fixe, l'erreur $\mathbf{EQMN_{stat}}$ diminue lorsque le nombre d'itération $\mathbf{N_{iter}}$ augmente.

la reconstruction d'une image de grande taille segmentée en sous blocs de taille [(NSxNS) ≥ (32x32)] donne de bons résultats de reconstruction (erreur très faible pour un nombre d'itérations réduits) par rapport à celle segmentée en sous blocs de taille [(NSxNS) ≤ (32x32)].

4.1.4 Variation de temps d'exécution : T_{exe}=f([NSx NS], N_{iter}) pour [NxN]=[512x512] :

Le tableau suivant représente les résultats de la variation de temps d'exécution $\mathbf{T_{exe}}$ en fonction du nombre d'itération $\mathbf{N_{iter}}$ et de la taille des sous bloc d'image à reconstruire (**NSxNS)**.

Tableau III. 4: *La variation de temps d'exécution T_{exe} en fonction de N_{iter} et de* (NSx NS)

Résultat de la reconstruction pour 5 image de taille N=512x512				
N_{iter} / NSxNS	10^1	10^2	10^3	10^4
8x8	19.10624	154.04586	1501.42	14365.8
16x16	9.37506	80.4104	791.2464	8249.472
32x32	5.79012	48.9934	494.5736	6193.28
64x64	4.2031	36.23024	359.17094	4561.8
128x128	3.9713	34.06688	342.95564	3411.16

Les figures suivantes montrent les résultats du tableau (III.4) sous forme de courbes.

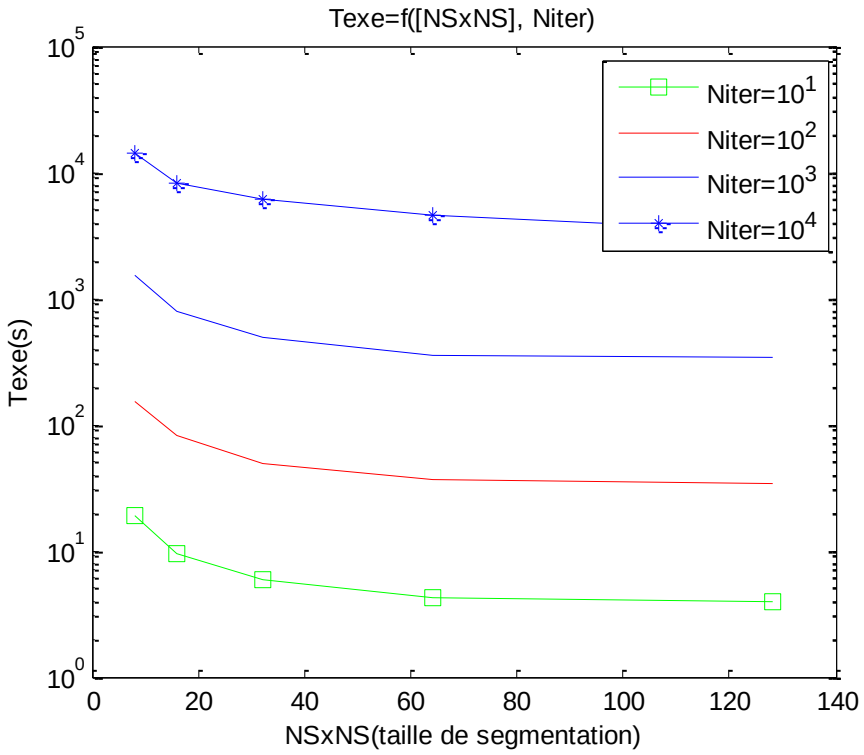


Figure III. 8: *Courbe de variation du temps d'exécution T_{exe} en fonction de* [NSx NS] et de N_{iter}.

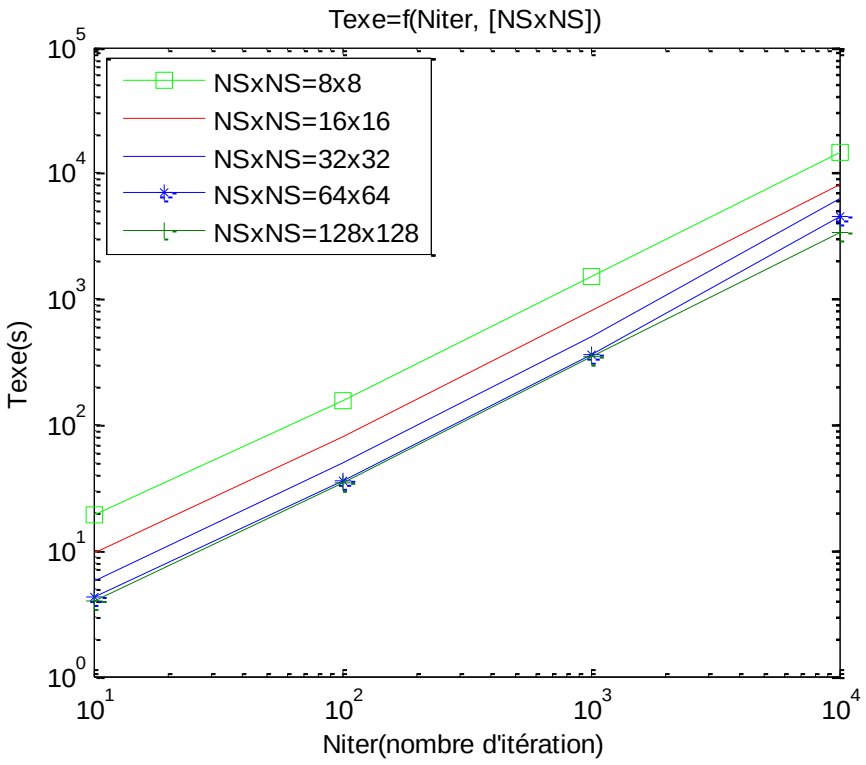
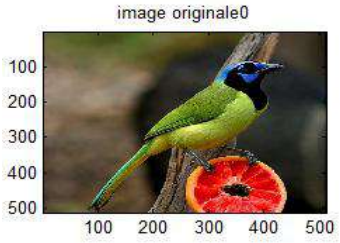


Figure III. 9: *Courbe de variation du temps d'exécution T_{exe} en fonction de N_{iter} (AI2D).*

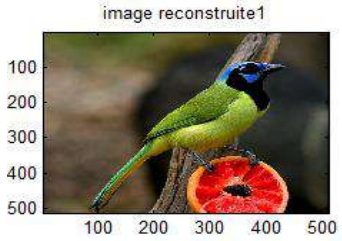
D'après les figures précédentes, nous remarquons que le temps d'exécution augmente en fonction du nombre d'itérations (N_{iter}). D'autre part, il diminue quand la taille [NSxNS] des sous blocs augmente.

Exemple illustratif :

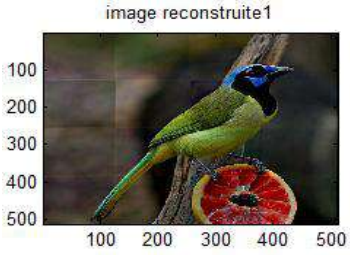
L'exemple suivant illustre les résultats obtenus dans la partie (4.1) en appliquant AIS2D sur une image de taille (NxN)=[512x512] pour des tailles des sous blocs différentes et de nombres d'itérations différents, comme les montrent la figure suivante :



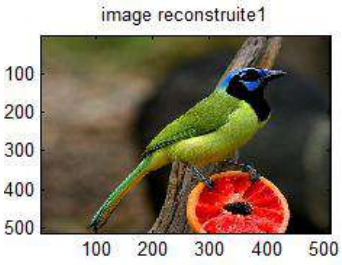
A-Image originale "oiseau " (512x512) pixels



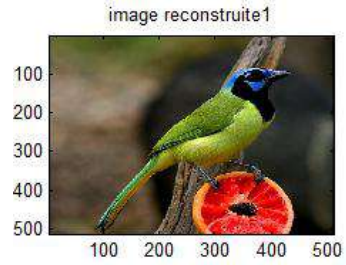
B- Image reconstruite à l'aide de"AIS2D" avec :
(NSxNS)=32x32 , N_{iter}=100 ; α=100.max(β)
EQMN=0.0008478 et T_{exe}=118.82 s



C- Image reconstruite à l'aide de "AIS2D" avec :
(NSxNS)=128x128 , N_{iter}=100 ; α=100.max(β)
EQMN= 0.000157 et T_{exe}=96.902 s



D- Image reconstruite à l'aide de"AIS2D" avec :
(NSxNS)=32x32 , N_{iter}=10000 ; α=100.max(β)
EQMN=0.00038 et T_{exe}=5049.7s



E-Image reconstruite à l'aide de "AIS2D"
avec :(NSxNS)=128x128,N_{iter}=10000 ; α=100.max(β) ;
EQMN= 0.0000004 et T_{exe}=3341.1s

Figure III. 10: *Reconstruction de l'image «oiseau » à l'aide de l'algorithme standard par segmentation (AIS2D) pour N_{iter}= [10^2, 10^4] et (NSxNS)=[32x32,128x128].*

5. Etude de l'influence de l'élément inséré α sur la convergence d'AIS2D:

Pour démontrer l'importance de l'élément α inséré dans l'algorithme itérative par segmentation (Algo-Iter-S_2D), on a fait une étude de reconstruction des images de taille N=[256x256, 512x512] à été effectuée à l'aide de Algo-iter-S_2D, pour les différentes tailles de sous bloc d'image (NSxNS)= [8x8, 16x16, 32x32, 64x64, 128x128] et pour différentes valeurs de α, avec « $\alpha = k.\beta$ » où :

k= [min, mean, max , 10*max,100*max,1000*max], et β est la matrice amplitude de chaque image et on écrit : $\beta = |signal|$.

Le tableau suivant représente la variation de l'erreur de reconstruction en fonction de la valeur de l'élément inséré, ainsi que la taille des sous bloc d'image (NS x NS).

Le nombre des itérations est : $N_{iter}=10^2$, $N_{iter}=10^4$, pour Algo-Iter-S_2D avec des tailles d'image différent ; N= [256x256, 512x512].

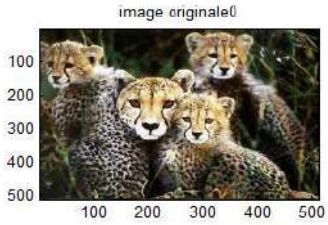
Tableau III. 5: *Variation de l'$EQMN_{stat}$ en fonction de l'élément inséré α.*

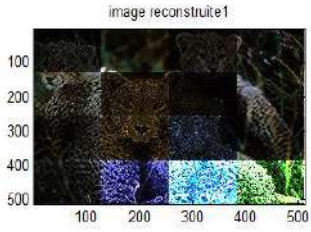
		AIS2D ($N_{iter}=10^2$)		AIS2D ($N_{iter}=10^4$)	
NS x NS	α = k(β)	$EQMN_{stat}$ (N=256x256)	$EQMN_{stat}$ (N=512x512)	$EQMN_{stat}$ (N=256x256)	$EQMN_{stat}$ (N=512x512)
8x8	Min(β)	1	0.997469	0.728924	0.358039
	Mean(β)	0.997878	0.969395	0.383800	0.368408
	Max(β)	0.952816	0.602937	0.349247	0.109709
	10*max(β)	0.006216	0.003331	0.004446	0.001189
	100*max(β)	0.001705	0.003230	0.001035	0.000225
	1000*max(β)	0.0003359	0.001136	0.000094	0.000055
16x16	Min(β)	1	0.978816	0.7289519	0.244680
	Mean(β)	0.971207	0.653693	0.210677	0.178096
	Max(β)	0.035262	0.025899	0.082655	0.009694
	10*max(β)	0.016912	0.003264	0.0004010	0.000072
	100*max(β)	0.010893	0.001658	0.0002741	0.000045
	1000*max(β)	0.003143	0.000842	0.0001969	0.000009
32x32	Min(β)	1	0.967367	0.994966	0.176021
	Mean(β)	0.077190	0.167380	0.7289519	0.094752
	Max(β)	0.049738	0.118583	0.0005094	0.000082
	10*max(β)	0.048365	0.022438	0.0004506	0.000016
	100*max(β)	0.00034	0.013700	0.0001074	0.000014
	1000*max(β)	0.000596	0.000197	0.00009157	0.00000235
64x64	Min(β)	1	0.967367	0.7289519	0.175115
	Mean(β)	0.593379	0.727428	0.0047118	0.000146
	Max(β)	0.3623295	0.168147	0.00008762	0.0000055
	10*max(β)	0.1348878	0.0984191	0.00003817	0.0000026
	100*max(β)	0.0264361	0.0133593	0.00003425	0.0000024
	1000*max(β)	0.0003208	0.0001783	0.00000368	0.0000018
128x128	Min(β)	1	0.960609	0.7289519	0.174783
	Mean(β)	0.477623	0.750122	0.00002667	0.00000119
	Max(β)	0.414875	0.531510	0.00001790	0.00000033
	10*max(β)	0.397380	0.329769	0.00001724	0.00000027
	100*max(β)	0.197837	0.119346	0.000017	0.00000023
	1000*max(β)	0.000174	0.0005656	0.00001433	0.00000022

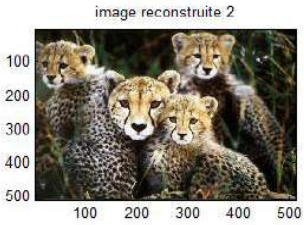
Les résultats ci-dessus nous montrent que l'élément inséré **α** a une grande importance pour améliorer les performances de reconstruction car, l'augmentation de l'élément inséré **α** permet de diminuer l'erreur de reconstruction $EQMN_{stat}$, et donc d'améliorer les performances de reconstruction d'algorithme standard.

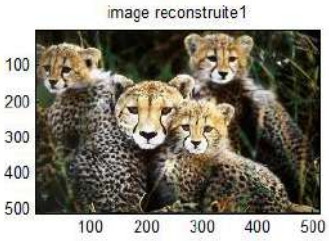
Exemple illustratif :

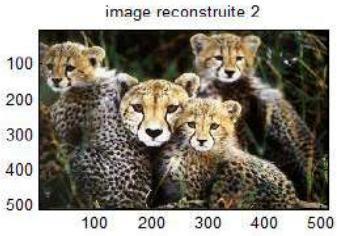
Pour confirmer l'étude de l'influence de l'élément inséré **α** sur la convergence d'algorithme standard 2D par segmentation, on va présenter dans la figure ci-dessous les résultats de la reconstruction d'une image : « animaux » à l'aide d'AIS2D pour différentes valeurs de α.

<table>
<tr><td colspan="2">

A-L'image originale "animaux " (512x512) pixels</td></tr>
<tr><td>

B-Image reconstruite à laide de "AIS2D "avec:
$\alpha = mean$; [NSxNS]=128x128 et $N_{iter}=10^2$;
EQMN=0.0433078 et T_{exe}=84.357s</td><td>

C-Image reconstruite à laide de "AIS2D "avec:
$\alpha = 1000 * \beta$; [NSxNS]=128x128 et $N_{iter}=10^2$;
EQMN= 0.00010616 et T_{exe}=81.9351 s</td></tr>
<tr><td>

D-Image reconstruite à laide de "AIS2D " avec :
$\alpha = mean$; [NSxNS]= 8x8 et $N_{iter}=10^2$
EQMN=0.0499521 et T_{exe}=232.8765s</td><td>

C-Image reconstruite à laide de "AIS2D " avec :
$\alpha = 1000 * \beta$;[NSxNS]= 8x8 et $N_{iter}=10^2$
EQMN= 0.0006221 et T_{exe}=210.5657s</td></tr>
</table>

Figure III. 11: *Reconstruction de l'image « animaux » à l'aide des algorithmes standard (AIS2D) pour $N_{iter}=10^2$ et (NSxNS)=[8x8, 128x128].*

6. Etude comparative de « AI2D» et « AIS2D» :

Une étude comparative, entre les résultats obtenus par la méthode de reconstruction itérative a été effectuée en se basant sur l'erreur de reconstruction et de temps d'exécution.

La reconstruction qui fait sur des images de taille N variable par AI2D et AIS2D pour différent nombre d'itération $N_{iter}=10^1$, $N_{iter}=10^2$, $N_{iter}=10^3$, $N_{iter}=10^4$; en tenant compte seulement de la segmentation en sous bloc de taille (NSxNS) = 32x32.

Le résultat de la comparaison a été représenté dans le tableau ci-dessous :

Tableau III. 6: *Résultat de comparaison entre AI2D et AIS2D pour* ([NSxNS]=[32x32, 64x64, 128x128] et (NxN)=[256x256].

	5 images de taille [NxN]=[256x256]							
	AIS2D		AIS2D					
			[NSxNS]=[32x32]		[NSxNS]=[64x64]		[NSxNS]=[128x128]	
N_{iter}	$EQMN_{stat}$	T_{exe}(sec)	$EQMN_{stat}$	T_{exe}(sec)	$EQMN_{stat}$	T_{exe}(sec)	$EQMN_{stat}$	T_{exe}(sec)
10^1	1	4.60822	1	4.5824	1	3.8874	1	3.58002
10^2	$679\ 10^{-3}$	28.7663	$17.59\ 10^{-3}$	24.87016	$4.234\ 10^{-3}$	19.6358	$2.50374\ 10^{-3}$	18.8938
10^3	$35960\ 10^{-5}$	251.522	$20.9046\ 10^{-5}$	210.313	$2.2718\ 10^{-5}$	187.673	$6.277\ 10^{-6}$	181.633
10^4	$26500\ 10^{-5}$	2590.07	$20.9046\ 10^{-5}$	2694.62	$2.2718\ 10^{-5}$	1857.768	$6.277\ 10^{-6}$	1661.1

Tableau III. 7: *Résultat de comparaison entre AI2D et AIS2D pour ([NSxNS]=[32x32, 64x64, 128x128] et (NxN)=[512x512]*.

	5 images de taille [NxN]=[512x512]							
	AI2D		AIS2D					
			[NSxNS]=[32x32]		[NSxNS]=[64x64]		[NSxNS]=[128x128]	
N_{iter}	$EQMN_{stat}$	T_{exe}(sec)	$EQMN_{stat}$	T_{exe}(sec)	$EQMN_{stat}$	T_{exe}(sec)	$EQMN_{stat}$	T_{exe}(sec)
10^1	1	13.695	1	5.79012	1	4.2031	1	3.9713
10^2	$976.06\ 10^{-3}$	118.75	$5.85\ 10^{-3}$	48.9934	$0.36\ 10^{-3}$	36.23024	$0.22\ 10^{-3}$	34.06688
10^3	$84556.6\ 10^{-5}$	1075.4	$13.9\ 10^{-5}$	494.5736	$0.7\ 10^{-5}$	359.17094	$0.18\ 10^{-5}$	342.95564
10^4	$58395.1\ 10^{-5}$	5879.8	$13.9\ 10^{-5}$	6193.28	$0.7\ 10^{-5}$	4561.8	$0.05\ 10^{-5}$	3411.16

D'après les résultats des tableaux ci-dessus, nous remarquons que :

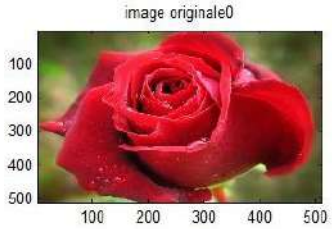
- L'erreur de reconstruction l'$EQMN_{stat}$ des deux tableaux montre que AIS2D est plus performant que AI2D. Car, plus la taille NSxNS est grande plus l'erreur de

reconstruction est meilleure et qui reste toujours en relation avec le nombre d'itération, c'est à dire plus le nombre d'itérations est élevé plus l'erreur de reconstruction est petit.

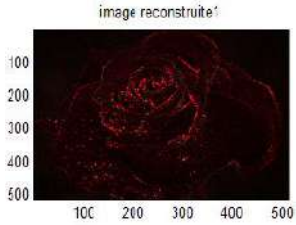
- Le temps d'exécution obtenu par AIS2D très petit que celui obtenu par AI2D pour différentes tailles de segmentation. Donc, la segmentation nous permet une amélioration de rapidité d'exécution. Et son augmentation reste liée aux nombres d'itérations utilisées.

Exemple de comparaison :

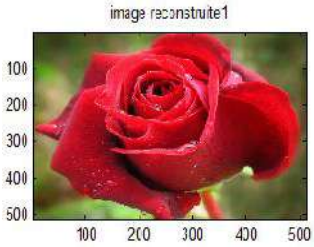
Pour affirmer les résultats obtenus avec et sans segmentation, une étude comparative entre les deux algorithmes standard **AI2D** et avec segmentation **AIS2D** a été effectuée. En appliquant ses deux derniers pour la reconstruction d'une image : « rose » de taille (512x512) pixels avec le nombre d'itérations (100 et 10000) d'une part et de taille des sous blocs (NSxNS)= (32x32), les résultats sont présentés sur la figure suivante :



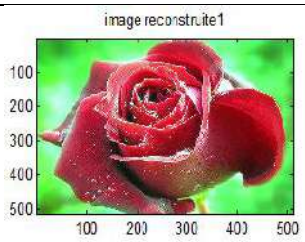
A-L'image originale "rose " (512x512) pixels



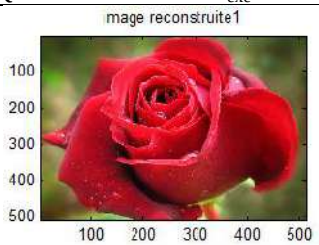
B-Image reconstruite à laide de"AI2D"avec : N_{iter}=100 ; EQMN= 0.965041 et T_{exe}=122.89s



C-Image reconstruite à laide de "AIS2D" avec : (NSxNS)=32x32 ; N_{iter}=100; α=100.max(β) EQMN= 0.008741 et T_{exe}=96.902s



D-Image reconstruite à laide de "AI2D"avec : N_{iter}=10000; EQMN= 0.556466 et T_{exe}=5469.7s



E-Image reconstruite à laide de "AIS2D"avec :(NSxNS)=32x32 , N_{iter}=10000; α=100.max(β) ;EQMN=0.000139 et T_{exe}=6256.54s

Figure III. 12: *Reconstruction de l'image « rose » à l'aide des algorithmes standard (AI2D et AIS2D) pour $N_{iter}=10^3$ et [NSxNS]=32x32.*

7. Conclusion :

D'après les résultats obtenus, on constate que la reconstruction des images par la technique de segmentation et insertion dépend fortement du nombre d'itérations et de la taille des sous bloc. Car, les performances de la reconstruction d'images de taille très grande [(NxN) ≥ (256x256)] sont améliorées en utilisant Algo-Iter-S_2D) en segmentant l'image en sous blocs de petite taille [NSxNS ≥ 32x32] et en insérant un élément ($\alpha \geq 10.\max(\beta)$).

Donc, nous pouvons dire que Algo-Iter-S_2D est plus performant que Algo-Iter_2D pour la reconstruction des images de très grand taille [(NxN) ≥256x256], où l'erreur de reconstruction (EQMN) et le temps d'exécution sont réduits.

Pour voir l'effet du bruit sur la reconstruction des signaux à partir de la phase contaminée par ce dernier ; nous allons étudier dans le chapitre suivant les différents algorithmes étudiés précédemment dans un contexte bruité.

Chapitre IV :

Etude de la reconstruction dans un contexte bruité

1. Introduction :

La transmission de l'information quelque soit sa nature (image, parole, …) peut être perturbée par des bruits de divers types, qui peuvent être additif, soustractif, multiplicatif ou le mélange entre les trois types.

Dans les chapitres précédents nous avons étudié les performances des algorithmes AI1D, AI2D et AIS2D dans un contexte non bruité.

Dans ce chapitre, nous allons étudier ses algorithmes dans un contexte bruité pour voir l'effet du niveau de ce dernier sur la reconstruction à partir de la phase spectrale bruitée par un bruit additif (bruit blanc)

2. La reconstruction en présence du bruit :

2.1 Reconstruction de signaux 1D :[9]

Soit x(i) une séquence de support fini $i \in [0, N-1]$ est satisfaisant les conditions du théorème de HAYES [1,2,3,5], $\varphi_x(k)$ sa phase spectrale exacte. Un bruit additif br(k) contamine la phase spectrale $\varphi_x(k)$ pour obtenir la phase spectrale dégradée $\varphi_{xbr}(k)$ qui est donné par l'expression suivante :

$$\varphi_{xbr} = \varphi_x(k) + br(k)\ ; \qquad \text{(III.1)}$$

avec : $k \in [0, N-1]$ et *br(k)* : le bruit contaminant la phase spectrale et qui varie entre –Ab et +Ab.

Le schéma bloc de la figure suivante, montre le principe de contamination de la phase 1D et la reconstruction des signaux à partir de cette phase bruitée.

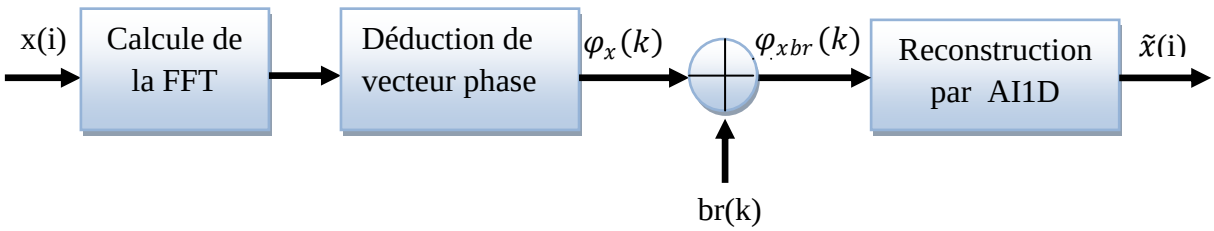


Figure IV. 1: *Schéma bloc de la procédure de reconstruction 1D dans un contexte bruité.*

Les étapes suivantes explicitent le principe de reconstruction dans un contexte bruité 1D.

1. à partir d'une séquence x(i) satisfaisant les conditions du théorème de HAYES, nous déduisant la phase spectrale $\varphi_x(k)$.
2. nous additionnons à cette phase $\varphi_x(k)$ le bruit br(k) pour obtenir la phase dégradée : $\boldsymbol{\varphi_{xbr}(k) = \varphi_x(k) + br(k)}$
3. nous utilisons l'algorithme étudié précédemment pour reconstruire la séquence $\tilde{x}(n)$.

avec :

x(i) : la séquence d'origine.

$\boldsymbol{\varphi_x(k)}$**:** la phase spectrale exacte du signal d'origine x(i).

br(k) : l'amplitude du bruit contaminant la phase spectrale $\boldsymbol{\varphi_x(k)}$ pour obtenir la phase dégradée $\boldsymbol{\varphi_{xbr}}$**(k)**.

$\tilde{\boldsymbol{x}}(\boldsymbol{i})$ **:** la séquence reconstruite à partir de la phase spectrale dégradée $\boldsymbol{\varphi_{xbr}}$.

2.2 Reconstruction de signaux (2D) :

Soit x (i, j) une image de support fini $i \in [0, N-1] et\ j \in [0, M-1]$est satisfaisons les conditions du théorème de HAYES, $\varphi_x(l,c)$ sa phase spectrale exacte. Un bruit additif br(l,c) contamine la phase spectrale $\varphi_x(l,c)$ pour obtenir la phase spectrale dégradée $\varphi_{xbr}(l,c)$ qui est donné par l'expression suivante :

$$\boldsymbol{\varphi_{xbr} = \varphi_x(l,c) + br(l,c)};\quad \textbf{avec : } \boldsymbol{l \in [0, N-1] et\ c \in [0, M-1]}$$

Le bruit ***br (l,c)*** contaminant la phase spectrale est un bruit d'amplitude variant entre –Ab et +Ab.

Le schéma bloc de la figure suivante, montre le principe de contamination de la phase 2D et la reconstruction des signaux à partir de cette phase bruitée.

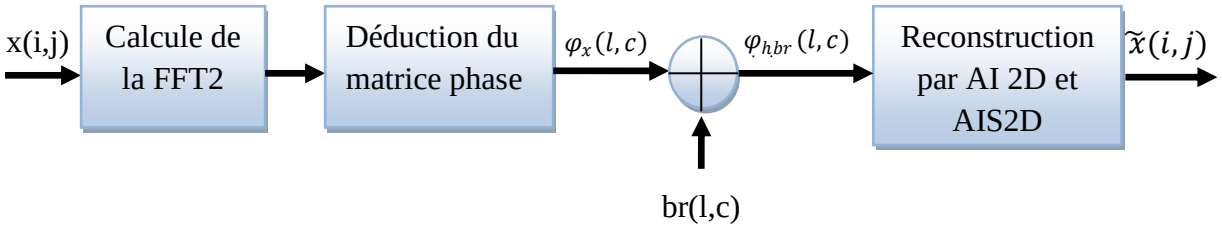


Figure IV. 2: *Schéma bloc de la procédure de reconstruction 2D dans un contexte bruité.*

Le principe de reconstruction dans un contexte bruité, s'effectue par les étapes suivantes :

1. à partir d'une matrice x (i, j) satisfaisant les conditions du théorème de HAYES, nous déduisant la matrice de la phase spectrale $\varphi_x(l,c)$.
2. nous additionnons à cette phase $\varphi_x(l,c)$ .Le bruit pour la phase dégradée $\boldsymbol{\varphi_{xbr}(l,c)}$**= $\boldsymbol{\varphi_x(l,c)}$ + br (l, c)**
3. nous utilisons l'algorithme étudie précédemment pour reconstruire l'image x (i, j).

Avec :

x **(i, j) :** La matrice d'origine.

$\boldsymbol{\varphi_x(l,c)}$ **:** La phase spectrale exacte du signal d'origine x (i, j).

br (l, c) : l'amplitude du bruit contaminant la phase spectrale exacte $\varphi_x(l,c)$ pour obtenir la phase dégradée $\boldsymbol{\varphi_{xbr}}$.

$\tilde{\boldsymbol{x}}$ $(\boldsymbol{i},\boldsymbol{j})$ **:** La matrice reconstruite à partir de la phase spectrale dégradée φ_{xbr} **.**

3. Implémentation des algorithmes itératifs dans un contexte bruité :

3.1 Résultats de « AI1D » dans un contexte bruite :

Cette implémentation a été effectuée de la manière suivante :

Pour chaque taille N∈[16, 32, 64, 128, 256, 512, 1024, 2048, 4096], les signaux ont été reconstruites à l'aide de AI1D à partir de leurs phases contaminées;

On utilise 5 signaux pour chaque taille, où $\boldsymbol{EQMN_{stat}}$ est obtenue par moyenne des ***EQMN*** (voir la page(22)) de chaque signal ; avec :

$$\boldsymbol{EQMN_{stat}} = \frac{1}{N_{Signaux}} \sum_{k}^{N_{Signaux}} EQMN(k)$$

telle que ces signaux de taille N ont été reconstruites pour un nombre maximal d'itération N_{iter} =10000 et pour chaque amplitude de bruit Ab, où :

$$\text{Ab} \in [\pi.10^{-5}, \pi.10^{-4}, \pi.10^{-3}, \pi.10^{-2}, \pi.10^{-1}, \pi.102, \pi.10^{1}]$$

Les résultats de l'$EQMN_{Stat}$ en fonction de N et de l'amplitude N_{br} du bruit blanc est représenté dans le tableau suivant :

Tableau IV. 1: *Résultats de la variation de l'EQMN$_{stat}$ en fonction de N et de niveau du bruit blanc N$_{br}$.*

N \ N_{br}	$\pi.10^{-5}$	$\pi.10^{-4}$	$\pi.10^{-3}$	$\pi.10^{-2}$	$\pi.10^{-1}$
16	0.09802	0.128070	0.218073	0.299034	0.79405
32	0.115722	0.155800	0.255808	0.340548	0.80167
64	0.136105	0.199583	0.293994	0.439948	0.83204
128	0.167325	0.227337	0.317338	0.591082	0.86156
256	0.199891	0.260548	0.350608	0.660608	0.89141
512	0.210177	0.301780	0.381834	0.710188	0.92670
1024	0.293075	0.393171	0.433246	0.793299	0.94500
2048	0.388035	0.408076	0.498117	0.804695	0.99234
4096	0.421078	0.501181	0.551083	0.868915	1

Les figures suivants nous permettre de la clarification des résultats du tableau précédent.

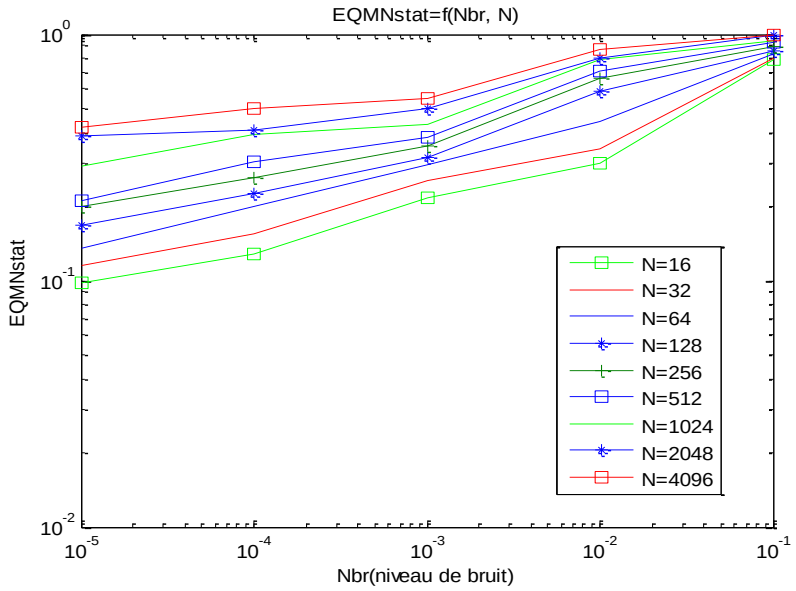


Figure IV. 3: *la variation de l'EQMN$_{stat}$ en fonction de niveau de bruit N$_{br}$ et de N (AI1D).*

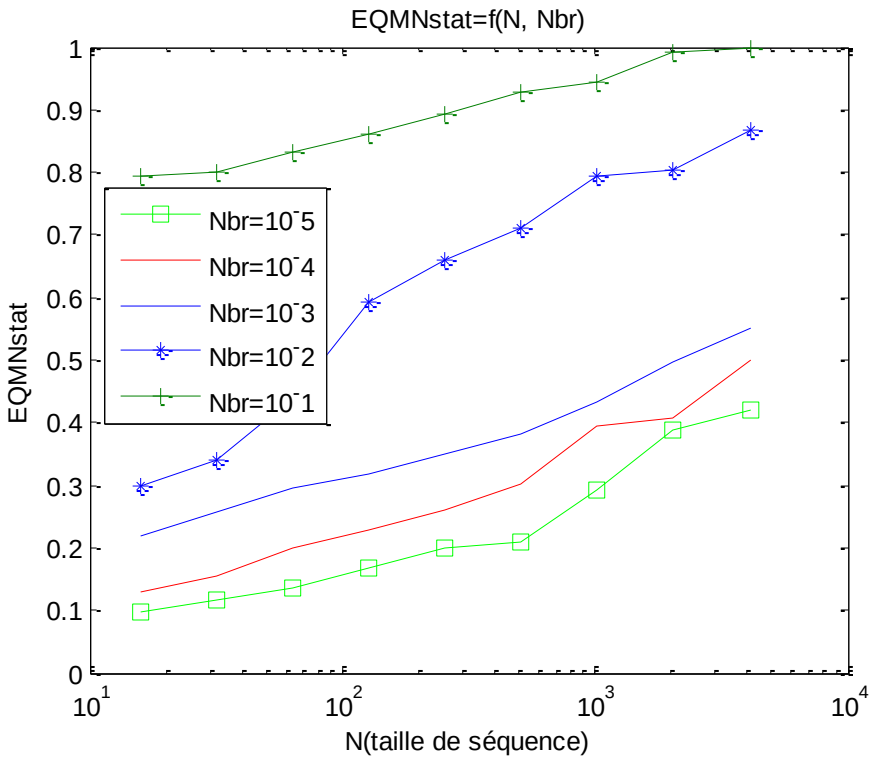


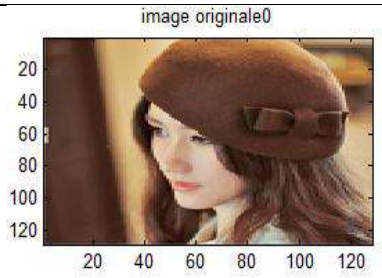
Figure IV. 4 : *la variation de l'EQMN$_{stat}$ en fonction de la taille de l'image N et de N$_{br}$ (AI1D).*

D'après les résultats des figures précédentes nous remarquons que :

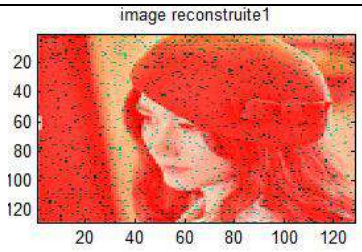
- ✓ L'erreur EQMN$_{stat}$ augmente en fonction de la longueur de séquence N et de niveau du bruit N$_{br}$.
- ✓ L'augmentation de EQMN$_{stat}$ devient importante dans le cas des deux derniers niveaux de bruit ($\pi.10^{-2}$, $\pi.10^{-1}$). Tandis que l'influence pour les autres niveaux faibles ($\pi.10^{-5}$, $\pi.10^{-4}$, $\pi.10^{-3}$) la reconstruction est meilleur.

Exemple illustratif :

Pour voir l'effet du niveau de bruit sur la reconstruction à partir de la phase spectrale bruitée, nous allons présenter les résultats de la reconstruction d'une image : « femme » à l'aide d'**AI1D** pour un niveau de bruit $N_{br} \in [\pi.10^{-5}, \pi.10^{-1}]$ et pour des nombres d'itérations différents N_{iter} = [100 et 10000] :



A-Image originale “femme“ (128x128) pixels



B-Image reconstruite par « AI1D» pour le bruit blanc avec : $N_{br}=10^{-1}$; $N_{iter}=100$; EQMN=1 et T_{exe}= 4.9241s



C-Image reconstruite par « AI1D » pour le bruit blanc avec : $N_{br}=10^{-5}$; $N_{iter}=100$ EQMN=0.27257 ; T_{exe}=5.3517s



D-Image reconstruite par « AI1D» pour le bruit blanc avec :$N_{br}=10^{-1}$; $N_{iter}=10000$; EQMN=1,; T_{exe}= 264.94s



E-Image reconstruite par « AI1D » pour le bruit blanc avec :$N_{br}=10^{-5}$; $N_{iter}=10000$; EQMN=0.002066 ; T_{exe}=189.40s

Figure IV. 5: *Reconstruction d’image « femme » à l’aide d’algorithme standard (AI1D) pour $N_{iter}= [10^2, 10^4]$ et $N_{br} = [10^{-5},10^{-1}]$.*

3.2)- Résultat d’ « AI2D» dans un contexte bruite :

Après l’obtention des résultats logique pour AI1D, nous allons étendre cette étude pour la méthode standard itérative bidimensionnelle (AI2D).

La mise en œuvre d’AI2D en présence de bruit se fera d’une manière identique que celle pour AI1D.

Pour chaque taille [NxN] ∈ [16x16, 32x32, 64x64, 128x128, 256x256], 5 image ont été reconstruites à l’aide de AI2D à partir de leurs phases contaminées ; où $\boldsymbol{EQMN_{stat}}$ est obtenue par moyenne des ***EQMN*** (voir la page (25)) de chaque image ; avec :

$$\boldsymbol{EQMN_{stat}} = \frac{1}{N_{Images}} \sum_{k}^{N_{Images}} \boldsymbol{EQMN}(k)$$

Telle que ces images de taille (NxN) ont été reconstruites pour chaque amplitude de bruit Ab, où : Ab∈ $[\pi.10^{-5}, \pi.10^{-4}, \pi.10^{-3}, \pi.10^{-2}, \pi.10^{-1}, \pi.10^{1}]$. ainsi que le nombre maximal d'itération est fixé à 10000.

Le résultat de l'EQMN$_{stat}$ en fonction de N et de l'amplitude Ab pour le bruit blanc est représenté dans le tableau suivant :

Tableau IV. 2: *Résultats de la variation de l'EQMN$_{stat}$ en fonction de N et de niveau du bruit blanc N$_{br}$.*

N$_{br}$ / **N**	**$\pi.10^{-5}$**	**$\pi.10^{-4}$**	**$\pi.10^{-3}$**	**$\pi.10^{-2}$**	**$\pi.10^{-1}$**
16x16	0.000003956	0.00000592	0.00001850	0.00081323	0.984196
32x32	0.00001128	0.00001243	0.00003789	0.00126012	1
64x64	0.0002954	0.0003677	0.0005404	0.0075544	1
128x128	0.003684	0.004262	0.0049392	0.0172144	1
256x256	0.02576	0.031721	0.0369731	0.2467571	1

Les figures suivantes représentent le résultat de tableau précédent sous forme des courbes :

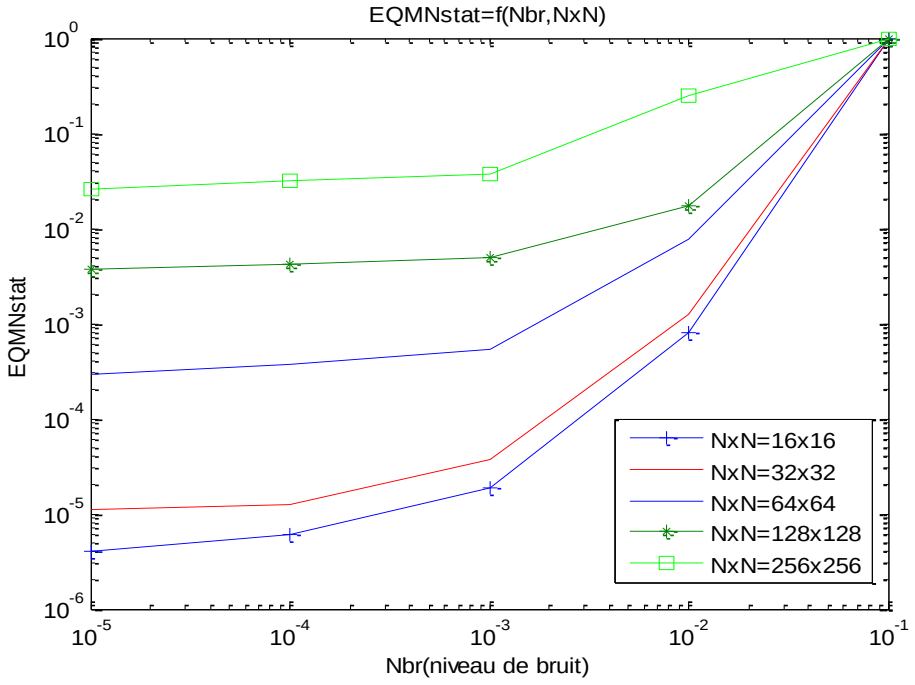


Figure IV. 6: *la variation de l'EQMN$_{stat}$ en fonction de niveau de bruit N$_{br}$ et de N (AI2D).*

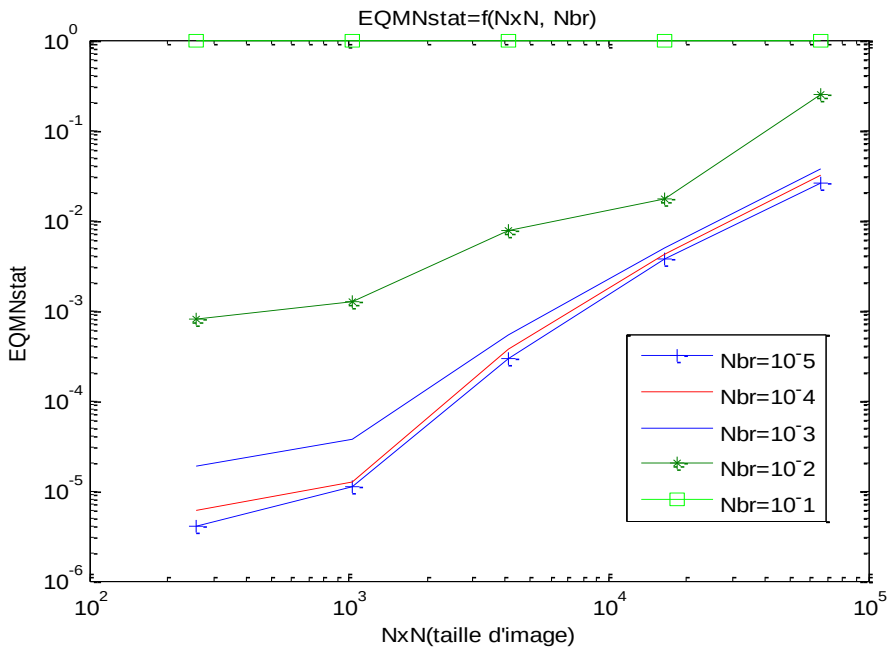


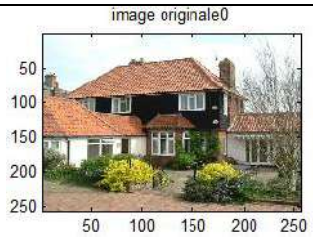
Figure IV. 7: *la variation de l'$EQMN_{stat}$ en fonction de la taille de l'image N et de N_{br} (AI2D).*

D'après les résultats de les figures précédent nous remarquons que :

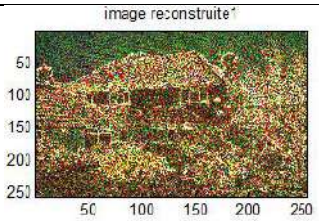
- ✓ L'erreur $EQMN_{stat}$ augmente en fonction de la taille de l'image N et de niveau du bruit N_{br}.
- ✓ L'augmentation de l'$EQMN_{stat}$ devient importante dans le cas des deux derniers niveaux de bruit ($\pi.10^{-2}$, $\pi.10^{-1}$). Tan disque l'influence pour les autres niveaux faibles ($\pi.10^{-5}$, $\pi.10^{-4}$, $\pi.10^{-3}$) la reconstruction est meilleur.

Exemple illustratif :

Pour affirmer les résultats obtenus par l'algorithme standard **AI2D** dans un context bruitée. En appliquant se dernier pour la reconstruction d'une image : « maison » de taille (256x256) pixels avec le nombre d'itérations (100 et 10000) d'une part et un niveau de bruit $N_{br} \in [\pi.10^{-5}, \pi.10^{-1}]$ d'autre part, les résultats sont présentés sur la figure suivante :



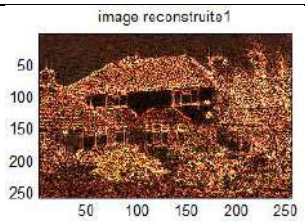
A-Image originale "maison" (256x256) pixels



B-Image reconstruite par « AI2D »pour le bruit blanc

avec : $N_{br}=10^{-1}$, $N_{iter}=100$

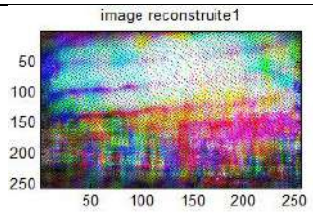
EQMN=0.91271, T_{exe}= 14.834s



C-Image reconstruite par « AI2D»pour le bruit blanc

avec :$N_{br}=10^{-5}$, $N_{iter}=100$

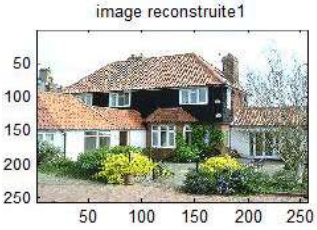
EQMN=0.098143 , T_{exe}=17.469s



D-Image reconstruite par « Ait2D» pour le bruit blanc

avec : $N_{br}=10^{-1}$, $N_{iter}=10000$

EQMN=0.920982, T_{exe}= 1156.4s



E-Image reconstruite par « AI2D» pour le bruit blanc

avec : $N_{br}=10^{-5}$, $N_{iter}=10000$

EQMN=0.027853 , T_{exe}=1410.5s

Figure IV. 8: *Reconstruction d'image «maison » à l'aide de l'algorithme standard (AI2D) pour Niter= [10^2, 10^4] et N_{br}=[10^{-5},10^{-1}].*

3.3 Résultat de « AIS2D» dans un contexte bruite :

Après l'étude détaillée de AI2D dans un contexte bruité, nous allons étudie cet algorithme standard dans le cas de segmentation et insertion en présence de bruit, sachant que l'AI2D sont validé pour des images de petite taille (N≤256x256).

Alors Pour cette implémentation, la mise en œuvre de AIS2D se faire en reconstruisant des images de taille (**NxN)**=[512x512] pour différentes taille [**NSxNS**] des sous bloc ; Pour cette taille (NxN), on utilise 5 images, où $\boldsymbol{EQMN_{stat}}$ est obtenue par moyenne des $\boldsymbol{EQMN}$ (voir la page (25)) de chaque image; avec :

$$\boldsymbol{EQMN_{stat}} = \frac{1}{N_{Images}} \sum_{k}^{N_{Images}} \boldsymbol{EQMN}(k)$$

La reconstruction de ces images de taille [NxN]= [512x512] a été effectuée pour chaque amplitude de bruit Ab, où : Ab∈ $[\pi.10^{-5}, \pi.10^{-4}, \pi.10^{-3}, \pi.10^{-2}, \pi.10^{-1}, \pi.10^{1}]$.

Le résultat de l'EQMN$_{stat}$ en fonction de (NSxNS) et de l'amplitude Ab pour le bruit blanc est représenté dans le tableau suivant :

Tableau IV. 3: *Résultats de la variation de l'EQMN$_{stat}$ en fonction de (NSxNS) et de niveau du bruit blanc (N$_{br}$).*

N$_{br}$ / NSxNS	**$\pi.10^{-5}$**	**$\pi.10^{-4}$**	**$\pi.10^{-3}$**	**$\pi.10^{-2}$**	**$\pi.10^{-1}$**
8x8	0.00000832	0.0000101	0.00005186	0.001113	1
16x16	0.00001211	0.00001479	0.0001252	0.001530	1
32x32	0.00001348	0.00001654	0.0001444	0.01223	1
64x64	0.00001419	0.00002931	0.0002693	0.018144	1
128x128	0.0002478	0.0004108	0.0079125	0.144216	1

Les courbes des figures suivantes illustrent les résultats du tableau précédent :

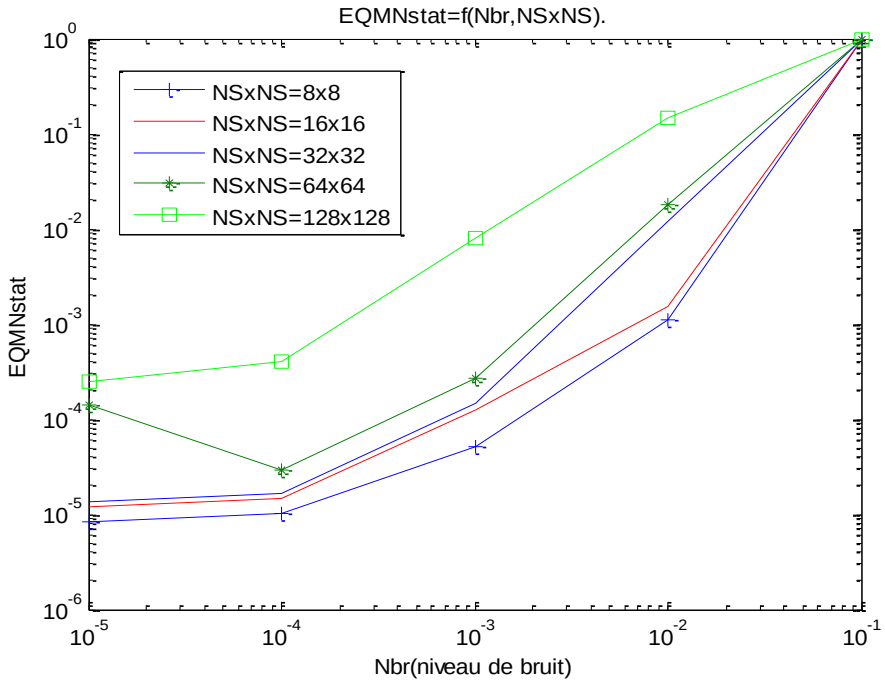


Figure IV. 9: *la variation de l'$EQMN_{stat}$ en fonction de niveau de bruit N_{br} et de (NSxMS) (AIS2D).*

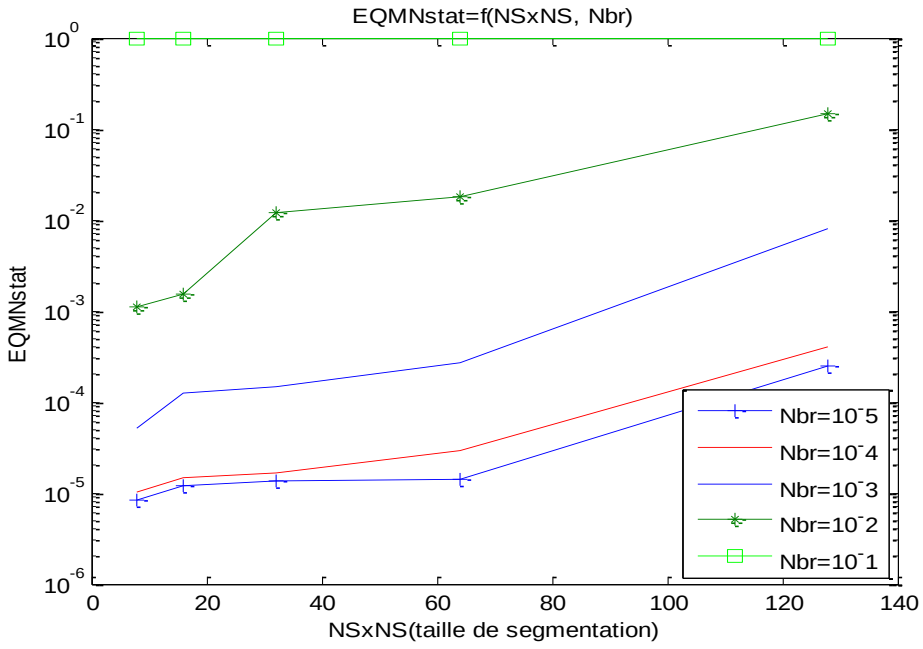


Figure IV. 10: *la variation de l'$EQMN_{stat}$ en fonction de la taille de l'image (NSxNS) et de N_{br} (AIS2D).*

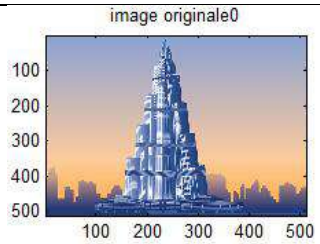
D'après les résultats des figures précédentes nous remarquons que :

- ✓ L'erreur $EQMN_{stat}$ augmente en fonction de la taille des sous blocs d'images (NSxNS) et du niveau de bruit N_{br}.

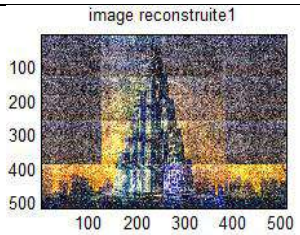
- ✓ L'augmentation de l'EQMN$_{stat}$ devient aussi importante dans le cas des deux derniers niveaux de bruit ($\pi.10^{-2}$, $\pi.10^{-1}$). Tan disque la reconstruction est meilleure pour les autres niveaux faibles ($\pi.10^{-5}$, $\pi.10^{-4}$, $\pi.10^{-3}$).

Exemple illustratif :

Voici maintenant un exemple de reconstruction dans un contexte bruité appliqué à une image «**Burj Khalifa »** en utilisant « AI2D)», où le niveau de bruit$N_{br} \in [\pi.10^{-5}, \pi.10^{-1}]$, la taille de segmentation (NSxMS) = [32x32 et128x128] et le nombre d'itérations N_{iter} = [100 et 10000].

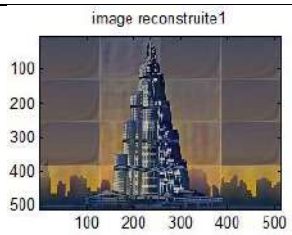


A-Image originale "burj khalifa" (256x256) pixels



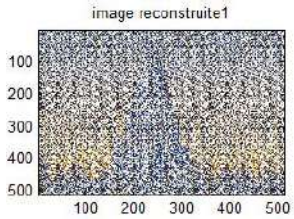
B-Image reconstruite par « AIS2D » pour le bruit blanc avec :$N_{br}=10^{-1}$, $N_{iter}=100$

EQMN=1 , T_{exe}=33.1487s (NSxNS)=128x128



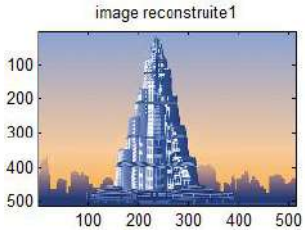
C-Image reconstruite par « AIS2D» pour le bruit blanc avec : $N_{br}=10^{-5}$, $N_{iter}=100$

EQMN=0.0000901, T_{exe}= 33.123s,(NSxNS)=128x128



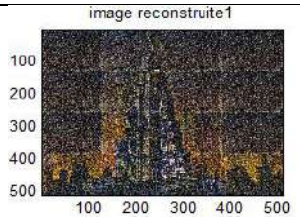
D-Image reconstruite par « AIS2D » pour le bruit blanc avec : $N_{br}=10^{-1}$, $N_{iter}=100$

EQMN=1, T_{exe}= 49.314s; (NSxNS)=32x32



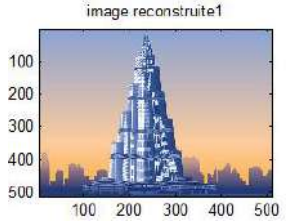
E-Image reconstruite par « AIS2D » pour le bruit blanc avec : $N_{br}=10^{-5}$, $N_{iter}=100$

EQMN=0.0001499 , T_{exe}=53.830s; (NSxNS)=32x32



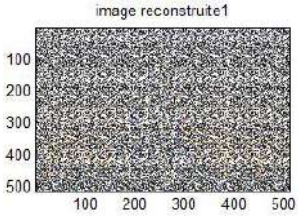
F-Image reconstruite par « AIS2D» pour le bruit blanc avec : $N_{br}=10^{-1}$, $N_{iter}=10000$

EQMN=1, T_{exe}=3036.5s; (NSxNS)=128x128



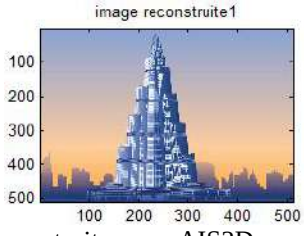
G-Image reconstruite par « AIS2D» pour le bruit blanc avec :$N_{br}=10^{-5}$, $N_{iter}=10000$

EQMN=0.00031, T_{exe}= 2955.2s; (NSxNS)=128x128



H-Image reconstruite par « AIS2D» pour le bruit blanc avec : $N_{br}=10^{-1}$,$N_{iter}=10000$

EQMN=1, T_{exe}= 5478.9s ;(NSxNS)=32x32



I-Image reconstruite par « AIS2D » pour le bruit blancavec : $N_{br}=10^{-5}$, $N_{iter}=10000$ EQMN=0.00032, T_{exe}=4831.7s (NSxNS)=32x32

Figure IV. 11: *Reconstruction d'image «burj khalifa » AIS2D))*
pour $N_{iter}= [10^2, 10^4]$; $N_{br}=[\pi 10^{-5},\pi 10^{-1}]$ *et (NSxNS)=[32x32, 128x128].*

5. Conclusion :

Dans ce chapitre, nous avons mis en œuvre des algorithmes (AI1D, AI2D et AIS2D) dans un contexte bruité.

Vue les résultats obtenus, on constate que le niveau du bruit à une grande influence sur la reconstruction, surtout pour les niveaux du bruit $N_{br}=[\pi 10^{-2}, \pi 10^{-1}]$, qui causent une dégradation importante. Par contre, les niveaux du bruit $N_{br}=[\pi 10^{-5}, \pi 10^{-4}, \pi 10^{-3}]$ n'ont pas une grande influence sur la reconstruction.

D'après l'étude comparative effectuée, on remarque que AIS2D présente de meilleurs résultats que les autres algorithmes en présence du bruit, et donc il y'aura une amélioration des résultats par réduction de l'effet du bruit.

4. Etude comparative entre les algorithmes standards dans un contexte bruité :

4.1 Etude comparative entre « AI1D » et « AI2D » dans un contexte bruité :

Pour mieux faire la comparaison on va vous présenter les résultats des deux algorithmes (AI1D et AI2D) en présence de bruit dans le tableau suivant :

Tableau IV.4: *Résultat de comparaison entre AI1D) et AI2D en présence du bruit pour [*NxN]=[512x512] et [NSxNS]=[32x32].

	Erreur de reconstruction en présence du bruit(EQMN)									
	AI1D)					AI2D)				
	N_{br}					N_{br}				
NxN	$\pi.10^{-5}$	$\pi.10^{-4}$	$\pi.10^{-3}$	$\pi.10^{-2}$	$\pi.10^{-1}$	$\pi.10^{-5}$	$\pi.10^{-4}$	$\pi.10^{-3}$	$\pi.10^{-2}$	$\pi.10^{-1}$
16x16	0.199891	0.260548	0.350608	0.660608	0.89141	0.000003956	0.00000592	0.00001850	0.00081323	0.984196
32x32	0.293075	0.393171	0.433246	0.793299	0.94500	0.00001128	0.00001243	0.00003789	0.00126012	1
64x64	0.421078	0.501181	0.551083	0.868915	1	0.0002954	0.0003677	0.0005404	0.0075544	1

D’après les résultats obtenus, nous remarquons que : AI2D est moins sensible au bruit que l’AI1D pour les différents niveaux du bruit N_{br}.

4.2 Etude comparative entre « AI2D » et « AIS2D » dans un contexte bruité :

Le tableau ci-dessous, montre les résultats de comparaison entre AI2D et AIS2D en présence de bruit :

Tableau IV.5: *Résultat de comparaison entre AI2D et AIS2D en présence du bruit pour [*NxN]=[512x512] et [NSxNS]=[32x32].

	Erreur de reconstruction en présence du bruit(EQMN)									
	AI2D					AIS2D				
	N_{br}					Nbr				
NxN	$\pi.10^{-5}$	$\pi.10^{-4}$	$\pi.10^{-3}$	$\pi.10^{-2}$	$\pi.10^{-1}$	$\pi.10^{-5}$	$\pi.10^{-4}$	$\pi.10^{-3}$	$\pi.10^{-2}$	$\pi.10^{-1}$
16x16	0.000003956	0.00000592	0.00001850	0.00081323	0.984196	0.00001211	0.00001479	0.0001252	0.001530	1
32x32	0.00001128	0.00001243	0.00003789	0.00126012	1	0.00001348	0.00001654	0.0001444	0.01223	1
64x64	0.0002954	0.0003677	0.0005404	0.0075544	1	0.0001419	0.00001931	0.0002693	0.018144	1
128x128	0.003684	0.004262	0.0049392	0.0172144	1	0.0002478	0.0004108	0.0079125	0.144216	1

D'après les résultats obtenus dans le tableau de comparaison ci-dessus, nous remarquons :

- ✓ pour les deux algorithmes aux niveaux de bruit $N_{br} \in [\pi.10^{-5}, \pi.10^{-4}, \pi.10^{-3}]$ l'erreur de reconstruction est faible, mais dans le reste il est très important $N_{br} \in [\pi.10^{-2}, \pi.10^{-1}]$.
- ✓ l'AIS2D est moins sensible au bruit que l'AI2D pour les différents niveaux du bruit N_{br}, donc la segmentation et l'insertion peuvent réduire l'effet du bruit.

Conclusion générale

Les résultats que nous avant trouvé dans ce travail nous ont permis de citer :

L'importance de la phase spectrale dans la reconstruction du signale dans certaines conditions que nous avons exposé au cours de notre travail : on peut reconstruire le signal à partir de sa phase spectrale en utilisant l'algorithme itératif de HAYES qui se base sur des allers-retours entre les domaines temporel et fréquentiel. ce dernier nécessite le traitement de deux parties :

- *Partie analyse :* elle consiste à extraire du signal d'origine les informations nécessaires pour la reconstruction, tel que la phase spectrale obtenue par la TFD1D ou TFD2D selon le cas.

- *Partie synthèse :* c'est la partie de la reconstruction du signal.

La mise en œuvre et les résultats de « AI1D et AI2D », ont montré que AI2D et plus performant que AI1D, car il permet la convergence vers le signal original en un nombre d'itérations plus réduit et donc c'est une réduction du temps d'exécution. De plus, les deux algorithmes 1D er 2D sont limités seulement à la reconstruction des images de taille [NxN] ≤ [256x256].

Pour des images de taille très grande, nous avons exploité une technique de segmentation et insertion par AIS1D, pour améliorer cette technique en 2D (AIS2D). L'implémentation et les résultats des algorithmes « AI2D et AIS2D» ont montré que AIS2D est plus performant que AI2D pour la reconstruction des images de taille [NxN] > [256x256] en utilisant une taille de segmentation [NsxNs] ≥ [32x32], pour un nombre d'itérations réduit.

La présence du bruit blanc additif, a conduit à la dégradation de la phase se qui conduit à une mauvaise reconstruction suivant le niveau du bruit. Car, on a trouvé que :

- Pour des niveaux du bruit $N_{br} \leq \pi.10^{-3}$, la dégradation est presque négligeable en augmentant le nombre d'itérations.
- Pour des niveaux du bruit $N_{br} \geq 10^{-2}$, la dégradation est très forte malgré l'augmentation du nombre d'itérations.

De plus, l'étude comparative des algorithmes itératif 1D et 2D en présence du bruit, a montré que (AI2D et AIS2D) ont présenté de meilleurs résultats que (AI1D).

Bibliographies:

[1] :M.H.HAYES.j.s.LIM and A.v OPPENHEIMAND « signal reconstruction from phase or magnitude » IEEE Trans. Acoust. Speech Signal Processing. Vol: ASSP-28, PP. 672-680,Dec1980.

[2] :M.H.HAYES, «Signal Reconstruction from Phase or Magnitude » SC.D. dissertation, Dep. Elec, ENG.Comput.SCI. MASSHUSETT.Technol ,Combridge ,JUNE 1981

[3] : M.H.HAYES et J.S.LIM and AV.OPPENHEIM « phase only signal reconstructions» in Proc. IEEE INT ConfAcoust. Speech Signal processing.pp.437-440 Apr 1980

[4]:AV.OPPENHEIMAND. J.S.LIM « the importance of phase in signals » invited. Paper; proc. IEEE ,vol, 69,pp.529-541.May 1981.

[5] :AV. OPPENHEIM MH HAYES?AND J.S LIM « itérativeprocédures for signal reconstruction from phase»presented at the SPIE.Conf. washinngton,DC. Apr 1980

[6] :Bernard.Picinbono « Signaux et systèmeslinéaire » ;Edition.Marketing., /ELLIPSES ,1980.

[7] : C.Y.ESPY LIM. « Effects of Noise in signal reconstruction from its Fourier transform phase » S.M Thesis. Dep . Elec. Eng.Comput. Sci,MASSHUSETTS.AC.inst.Technol. Cambridge, May 1981.

[8] : Rais Elhadi BAKKA « introduction au traitement du signal », Alger.OP4.1993 146p

[9] : FekharImane.Talhiselma, « Reconstruction de signaux à partir de la phase spectrale », Mémoire d'Ingéniorat, université Dr.yahiafares de medea,2002

[10] : C. Brielmann, « Transformée de Fourier pour les signaux échantillonnés »,The Scientist and Engineer's Guide to Digital Signal Processing (S. Smith, www.dspguide.com),HE-Arc, ingénierie.

[11] : Joël Le Roux. « Traitement Numérique des Signaux Bidimensionnels »,thèse,vol.66, PP. 5-14. 19 décembre 2000.

[12] : Denis Prêtre. Murat Kunt, Ce document s'inspire du livre « Traitement de l'information », Presses Polytechniques Romandes. HE-Arc, ingénierie., dernière révision : 19/01/2004.

Annexe (A)

1. Démonstration de théorème (2.1) :

Si le transformée en Z des suites x(i) et y(i) n'ont pas de zéros sur le cercle unité ou par paires réciproque, on peut dire que.

- Si $X(z_0)=0 \Longrightarrow X(1/z_0)\neq 0$; et
- Si $Y(z_0)=0 \Longrightarrow Y(1/z_0)\neq 0$.

Soit h(i) une suite de longueur finie et définie comme suite.

$$h(i) = x(i) \otimes y(-i) \qquad (1)$$

$\otimes$: désigne le produit de convolution.

Sa transformation en z est H(z). Telle que :

$$H(z)=X(z) \,.\, Y(1/Z) \qquad (2)$$

Que l'on peut écrire comme suit :

$$H(z) = |H(z)|.\exp(j\varphi_h(\omega)) \qquad (3)$$

En utilisant (2), on aura

$$H(z) = |X(z)|.|Y(1/z)|\exp(j(\varphi_x(\omega) - \varphi_y(\omega)) \qquad (4)$$

Avec

$$\varphi_h(\omega) = \varphi_x(\omega) - \varphi_y(\omega) \qquad (5)$$

Avec $\varphi_x(\omega)\ et\ \varphi_y(\omega)$ représentent respectivement les phases spectrale de X(z) et Y(z).

- Si $\varphi_x(\omega) = \varphi_y(\omega), alors\ \varphi_h(\omega) = 0$
- Si $\tan(\varphi_x(\omega)) = \tan\left(\varphi_y(\omega)\right), alors\ \varphi_h(\omega) = \{0\ ou\ \pi\}$.

En replaçant $\varphi_h(\omega)$ dans (3), on aboutit à

$$H(z) = |H(z)|.\exp(j\varphi_h(\omega)) = |X(z)|\cos(\varphi_h(\omega)) \qquad (6)$$

D'après l'expression (6), on déduit que la partie imaginaire est nulle, d'où

$$H(z) \text{ est réelle et paire} \Longrightarrow H(z)=H(1/z). \qquad (7)$$

L'expression (7) nous montre que H(z) a des zéros par paire réciproque. C'est-à-dire, si on pose que z_0 est un zéro pour H(z), alors on aura

$H(z_0) = 0$ et $H(1/z_0) = 0$

Partir des expressions (2) et (7), on trouve

$H(z_0) = X(z_0)\,.\,Y(1/z_0)=0$

$H(z_0) = X(1/z_0)\,.\,Y(z_0)=0$

Si z_0 est une zéro pour X(z) et H(z), ceci implique que $X(z_0)=0$, or par hypothèse X(z) n'admet pas les zéros par paires réciproques ou sur le cercle unité, donc $X(1/z_0) \neq 0$, en comparant ses résultats avec les deux expressions (8) et (9), on trouve que :

$Y(z_0) = 0$ et $Y(1/z0) \neq 0$ (par hypothèse).

En conclusion, les zéros de X(z) et Y(z) sont identique.

A partir de (2) et (7), on a :

$$H(z) = X(z)\,.\,Y(1/z) \qquad (8)$$

et $$H(z_0) = X(1/z)\,.\,Y(z) \qquad (9)$$

$$\Longrightarrow \quad X(z) = \frac{H(z)}{Y(z)} = \frac{X\left(\frac{1}{z}\right)}{Y\left(\frac{1}{z}\right)} Y(z)$$

$$\Longrightarrow \quad X(z) = \beta.Y(z) \qquad (10)$$

$$x(i) = \beta.y(i) \qquad (11)$$

En remplaçant (11) dans (1), on trouve que :

$$h(i) = \beta.y(i) \bigotimes y(-i)$$

$$H(z) = \beta.Y(z)\,.\,Y(1/z) \qquad (12)$$

A partir de (3) et (12) on aura :

$$H(z) = |H(z)|.\exp\left(j\varphi_h(\omega)\right) = H(z) = |\beta|.|Y(z)|.|Y(1/z)|\exp\left(j\varphi_x(\omega) - \varphi_y(\omega)\right)$$

D'après cette dernière expression on remarque que :

- Si $\varphi_x(\omega) = \varphi_y(\omega)$, c'est-à-dire $\varphi_h(\omega) = \varphi_x(\omega) - \varphi_y(\omega) = 0$, alors β est une constante positive ;
- Si $\tan\left(\varphi_x(\omega)\right) = \tan\left(\varphi_y(\omega)\right)$, c'est-à-dire : $\varphi_h(\omega) = \{0\ ou\ \pi\}$, alors β est une constante réelle.

2. Démonstration de théorème (2.2) :

La validité de ce théorème suit d'une manière presque identique celle du théorème (2.2) précédent.

Soit h(i) une suite de longueur finie et qui est nulles en dehors de l'intervalle [-N-1,N-1], et sa transformation de Fourier est donnée par l'expression suivante:

$H(\omega) = |H(\omega)|.\exp(j\varphi_x(\omega))$

La phase $\varphi_h(\omega)$ égale a 0 ou π à N-1 fréquences distinctes w_1, w_2 ……., w_{N-1} dans l'intervalle fréquentiel [0, π], c'est-à-dire

$$\varphi_h(\omega_k) = 0 \; ou \; \pi \; pour \; K = 1,2, \dots\dots\dots\dots\dots\dots, N-1,$$

$$avec \; \omega_k \neq \omega_1 \; et \; 0 < \omega_k < \pi \; pour \; k = 1,2, \dots\dots\dots\dots, N-1,$$

Donc, $(H(\omega_k)$ prend des valeurs réelles, c'est-à-dire la parie imaginaire est nulle.

$$Im[H(\omega_k)] = \sum_{n=N-1}^{N-1} h(i)\sin(i\omega_k) = 0$$

Donc

$$\sum_{-N-1}^{-1} h(-i)\sin(-i\omega_k) + \sum_{1}^{N+1} h(i)\sin(i\omega_k) = 0$$

Mais

$$-\sum_{1}^{N-1} h(-i)\sin(i\omega_k) + \sum_{1}^{N-1} h(i)\sin(i\omega_k) = 0$$

Ceci est équivalent à :

$$\sum_{1}^{N-1} (h(i) - h(-i))\sin(i\omega_k) = 0 \; pour \; i = 1, \dots\dots\dots N-1$$

D'après cette dernière expression on constante que la séquence h(i) est paire.

Considérons à présent deux séquence x(i) et y(i) satisfont les conditions du théorème (2) dont la même phase pour les (N-1) fréquences distinctes dans l'intervalle fréquentiel [0, π], définissons une séquence h(i) comme suit

$$h(i) = x(i) \otimes y(-i)$$

Avec ⊗ : désigne le produit de convolution

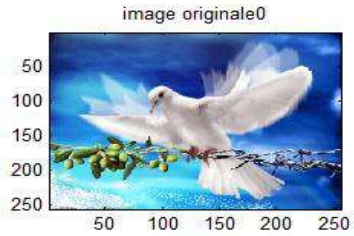
Puisque h(0) en dehors de l'intervalle temporel [N-1, N-1], et que sa phase spectrale $\varphi_h(\omega)$ prend (0 ou π) pour les (N-1) fréquences distinctes dans l'intervalle fréquentiel [0, π], avec $w_k \neq w_1$pour $k \neq 1$, c'est-à-dire $0 < w_k < \pi$ pour k=1,2,……, N-1, alors la séquence h(i) est paire.

De même que le théorème (1), on conclut que : x(i)=β y(i)

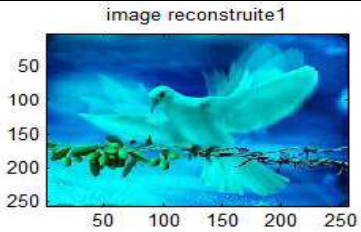
Ou β sera une constante positive, si les échantillons de la phase de x(i) et de y(i) sont égaux, et aura une constante réelle si les échantillons de la tangente de la phase spectrale sont égaux.

Annexe(B)

Les résultats de la reconstruction des différents algorithmes :

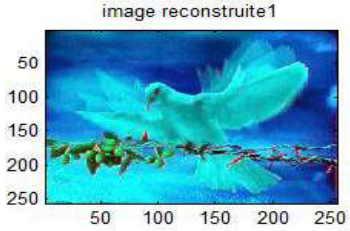


A-L'image originale « Pigeon » (256x256) pixels.



B-Image reconstruite à l'aide de«AI1D » avec :

EQMN=0.98809 et T_{exe}=5.8634sec ; $N_{iter}=10^2$.



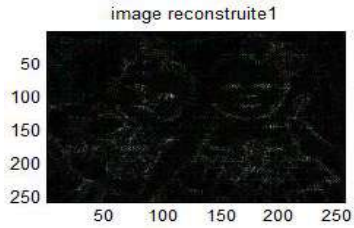
C-Image reconstruite à l'aide de «AI1D » avec :

EQM=0.82635 et T_{exe}=582.46sec ; $N_{iter}=10^4$

Figure II. 1: *Reconstruction de l'image « Pigeon» à l'aide des algorithmes standard (AI1D) pour $N_{iter}=(10^2$ et $10^4)$.*



A-L'image originale «Enfants » (256x256) pixels.



B-Image reconstruite à l'aide de «AI2D » avec :

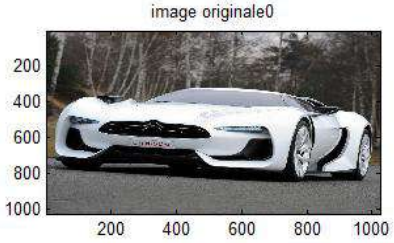
EQM=0.97504 et T_{exe}=12.955sec ; $N_{iter}=10^2$



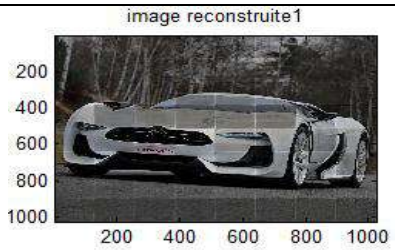
C-Image reconstruite à l'aide de «AI2D » avec :

EQM=0.0000521 et T_{exe}=1242.2sec ; $N_{iter}=10^4$

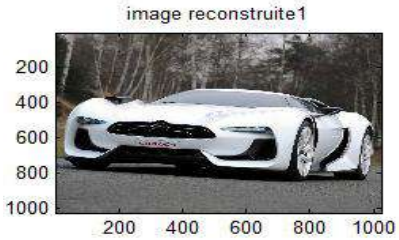
Figure II. 2: *Reconstruction de l'image « Enfants» à l'aide des algorithmes standard (AI2D) pour $N_{iter}=(10^2$ et $10^4)$.*



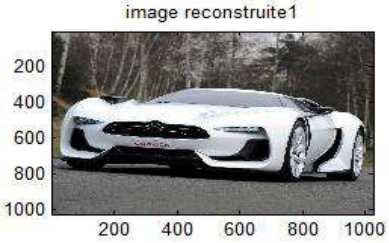
A-L'image originale "Automobile " (1024x1024) pixels



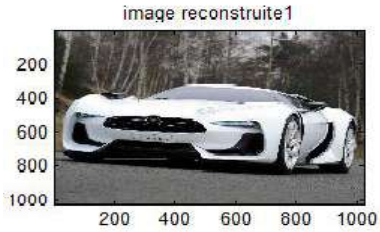
B-Image reconstruite à laide de "AIS2D" avec :(NSxNS)=(128x128),N_{iter}=100; α=100.max(β) EQMN= 0.03406 et T_{exe}=142.0798s



C-Image reconstruite à laide de "AIS2D" avec :(NSxNS)=(128x128);N_{iter}=10000; α=100.max(β) EQMN= $6.469x10^{-7}$ et T_{exe}=14062s

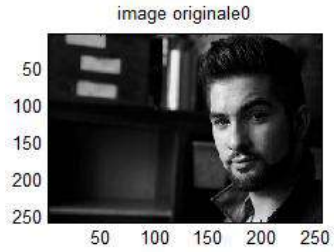


D-Image reconstruite à laide de "AIS2D"avec :(NSxNS)=(32x32);N_{iter}=100; α=100.max(β) EQMN= 0.09954 et T_{exe}=327.3278s

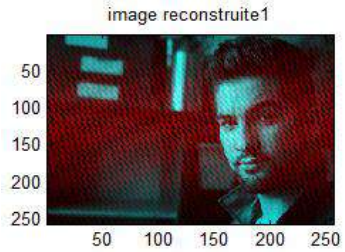


E-Image reconstruite à laide de "AIS2D" avec :(NSxNS)=32x32;N_{iter}=10000; α=100.max(β) EQMN= 0.07837 et T_{exe}=22592s

Figure III.1: *Reconstruction de l'image « Automobile » à l'aide d'algorithme standard (AIS2D) pour N_{iter}=(10^2 et 10^4) et^4 [NSxNS]=[32x32 ,128x128].*



A-L'image originale "Homme " (256x256) pixels



B-Image reconstruite par « AI1D» pour le bruit blanc avec : $N_{br}=10^{-1}$; $N_{iter}=100$;
EQMN=1 et T_{exe}= 15.850s



C-Image reconstruite par « AI1D» pour le bruit blanc avec : $N_{br}=10^{-5}$; $N_{iter}=100$;
EQMN=0.98102 et T_{exe}= 15.436s



D-Image reconstruite par « AI1D» pour le bruit blanc avec : $N_{br}=10^{-1}$; $N_{iter}=10000$;
EQMN=1 et T_{exe}= 1293.3s

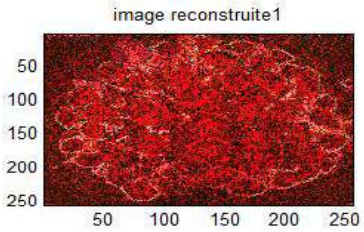


E-Image reconstruite par « AI1D» pour le bruit blanc avec : $N_{br}=10^{-5}$; $N_{iter}=10000$;
EQMN=0.15538 et T_{exe}= 1418.5s

Figure IV. 1: *Reconstruction d'image « Homme » à l'aide d'algorithme standard (AI1D) pour* N_{iter}= $[10^2, 10^4]$ *et* N_{br} = $[10^{-5},10^{-1}]$.

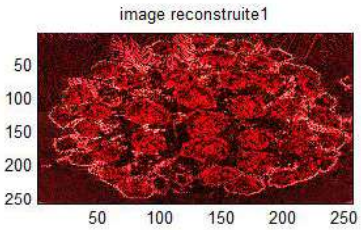


A-Image originale « Bouquet » (256x256) pixels



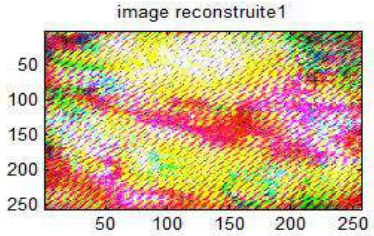
B-Image reconstruite par « AI2D »pour le bruit blanc avec : $N_{br}=10^{-1}$, $N_{iter}=100$

EQMN=1, T_{exe}= 15.127s



C-Image reconstruite par « AI2D »pour le bruit blanc avec : $N_{br}=10^{-5}$, $N_{iter}=100$

EQMN=0.81364, T_{exe}= 14.610s



D-Image reconstruite par « AI2D »pour le bruit blanc avec : $N_{br}=10^{-1}$, $N_{iter}=10000$

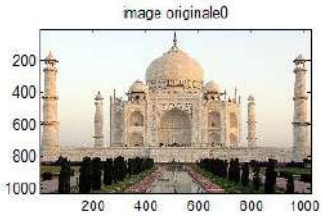
EQMN=1, T_{exe}= 1442.6s



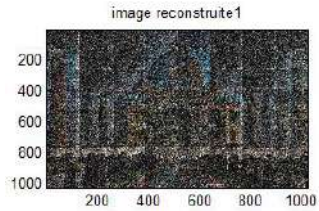
E-Image reconstruite par « AI2D »pour le bruit blanc avec : $N_{br}=10^{-5}$, $N_{iter}=10000$

EQMN=0.0009227, T_{exe}= 1332.5s

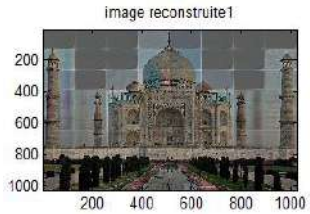
Figure IV. 2: *Reconstruction d'image «Bouquet » à l'aide de l'algorithme standard (AI2D) pour Niter= [10^2, 10^4] et N_{br}=[10^{-5},10^{-1}].*



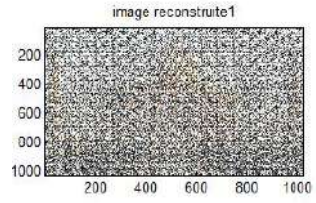
A- L'image originale « Taj Mahal » (1024x1024) pixels.



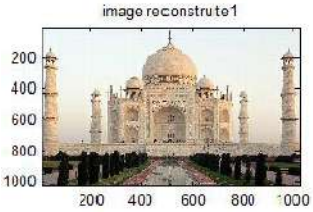
B- Image reconstruite à laide de « AIS2D »pour le bruit blanc avec :Nbr=10^{-1},Niter=100 EQMN=1, Texe=125.4709 s,(NSxNS)=128x128



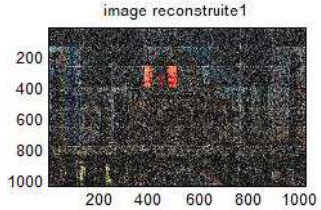
C- Image reconstruite à laide de « AIS2D »pour le bruit blanc avec :Nbr=10^{-5},Niter=100 EQMN=0.0004389, Texe=130.3319s ,(NSxNS)=128x128



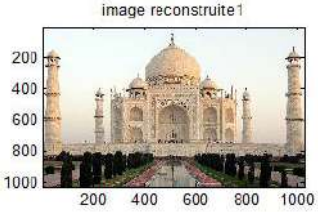
D- Image reconstruite à laide de « AIS2D »pour le bruit blanc avec :Nbr=10^{-1},Niter=100, EQMN=1, Texe=337.3541s,(NSxNS)=32x32



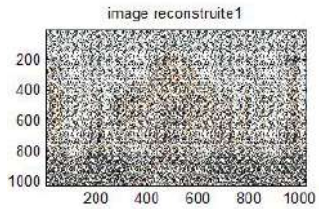
E- Image reconstruite à laide de « AIS2D »pour le bruit blanc avec :Nbr=10^{-5},Niter=100 ,EQMN=0.49919, Texe=322.8367s ,(NSxNS)=32x32



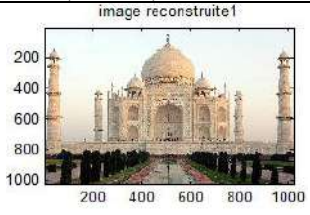
F- Image reconstruite à laide de « AIS2D »pour le bruit blanc avec :Nbr=10^{-1},Niter=10000, EQMN=1, Texe=13942 s,(NSxNS)=128x128



G- Image reconstruite à laide de « AIS2D »pour le bruit blanc avec :Nbr=10^{-5},Niter=10000, EQMN=0.000000636, Texe=12778 s,(NSxNS)=128x128



H- Image reconstruite à laide de « AIS2D »pour le bruit blanc avec :Nbr=10^{-1},Niter=10000, EQMN=1, Texe=21897 s,(NSxNS)=32x32



I- Image reconstruite à laide de «AIS2D »pour le bruit blanc avec :Nbr=10^{-5},Niter=10000, EQMN=0.00001042, Texe=28663s,(NSxNS)=32x32

Figure IV.3: *Reconstruction d'image «Taj Mahal » (AIS2D) pour* $N_{iter}= [10^2, 10^4]$; $N_{br}=[\pi 10^{-5}, \pi 10^{-1}]$ *et (NSxNS)=[32x32, 128x128].*