



DÉPARTEMENT D'ÉLECTRONIQUE
Laboratoire Signal & Communications



Mémoire de magister en Électronique

Option : Traitement du Signal et Communications

Présenté par :

CHARIFI Tarek

Ingénieur d'État en Électronique (ENP)

DEVELOPPEMENT D'ALGORITHMES DE SUPERRESOLUTION PAR DES METHODES D'INTERPOLATION ET TRIANGULATION DE DELAUNAY

Le jury d'examen est composé de :

Président :	Melle M.GUERTI	Professeur (ENP)
Rapporteur :	Mme L.HAMAMI	Professeur (ENP)
Examineur :	Mr D.BERKANI	Professeur (ENP)
	Mr H.BOUSBIA	Maître de conférences (ENP)
	Mr M.S.AIT CHEIKH	Maître de conférences (ENP)

REMERCIEMENTS

Mes plus intenses remerciements vont, en premier lieu, à DIEU le Miséricordieux le Tout-Puissant pour m' avoir donné le courage de mener à terme ce travail.

Je tiens, en deuxième lieu, à remercier ma promotrice, madame HAMAMI, pour son aide, son soutien et sa patience tout au long de mon projet de magister.

Ainsi que Mr OUARED, l'UT7 Genève de nous avoir initié dans ce domaine.

Je tiens également à remercier madame GUERTI d' avoir accepté de présider le jury auprès duquel est exposé le résultat du travail effectué dans ce mémoire.

Merci également aux examinateurs d' avoir accepté de juger ce modeste travail

J' adresse un spécial remerciement aux membres du laboratoire «SIGNAL ET COMMUNICATIONS » qui m' ont beaucoup aidé et soutenu durant ce projet et avec qui j' ai appris beaucoup de choses intéressantes.

Enfin, je suis reconnaissant à toutes celles et tous ceux qui ont contribué de près ou de loin à l' accomplissement de ce travail.



Dédicaces

**La joie de réussir n'est jamais complète
que si elle est partagée avec ceux qu'on aime.**

**A ma mère, à mon père, ceux à qui je dois tout.
A mes grands parents que j'adore.**

A mes frères et sœurs.

A mes oncles, à mes tantes.

A toute ma famille.

A tous ceux que j'aime et qui m'aiment.

Je dédie ce travail

Tarek



DÉPARTEMENT D'ÉLECTRONIQUE
Laboratoire Signal & Communications



Mémoire de magister en Électronique

Option : Traitement du Signal et Communications

Présenté par :

CHARIFI Tarek

Ingénieur d'État en Électronique (ENP)

DEVELOPPEMENT D'ALGORITHMES DE SUPERRESOLUTION PAR DES METHODES D'INTERPOLATION ET TRIANGULATION DE DELAUNAY

Le jury d'examen est composé de :

Président :	Melle M.GUERTI	Professeur (ENP)
Rapporteur :	Mme L.HAMAMI	Professeur (ENP)
Examineur :	Mr D.BERKANI	Professeur (ENP)
	Mr H.BOUSBIA	Maître de conférences (ENP)
	Mr M.S.AIT CHEIKH	Maître de conférences (ENP)

ملخص:

تعد عملية زيادة التوضيح (superrésolution) إحدى أهم عمليات المعالجة الأولية للصور حيث تسمح بتحسين نوعية الصورة من خلال زيادة توضحها (résolution).
يهتم هذا العمل بطرح طرق زيادة التوضيح باستعمال تقنيات التوليد المعتمدة على كثيرات الحدود و الترشيح، حيث يعرض و يقارن مختلف خوارزميات التوليد المستغلة في عمليات زيادة التوضيح الأحادي الصورة مع اقتراح خوارزميات أخرى لتحسينها و تكييفها مع عملية زيادة التوضيح المتعدد الصور باستعمال مخطط فيرانوي و تثليث ديلوناي.

المفتاح:

زيادة التوضيح ، نظرية أخذ العينات، التوليد ، مرشح ويتيكر ، اللسين ، مخطط فيرانوي، تثليث ديلوناي.

Abstract

The superresolution is one of the most significant and the most promising operations of preprocessing. It makes it possible to improve quality of the image by increasing its resolution.

This work is interested to superresolution techniques by the polynomial interpolation and the interpolation by filtering: it exposes and compares the various exploited algorithms of interpolation used in the monoframe superresolution. It proposes also other algorithms to optimize them and adapt them to the multiframe superresolution by using the Voronoï diagram and the Delaunay triangulation.

Key words:

superresolution, sampling theorem, interpolation, the Whittaker filter, bicubic spline, Voronoï Diagram, Delaunay Triangulation.

Résumé

La superrésolution est l'une des opérations de prétraitement des plus importantes et des plus prometteuses. Elle permet d'améliorer la qualité de l'image en augmentant sa résolution.

Ce travail s'intéresse aux méthodes de superrésolution par interpolation polynomiale et par interpolation par filtrage: il expose et compare les différents algorithmes d'interpolation exploités dans la superrésolution monoframe et propose des algorithmes pour les optimiser et les adapter à la superrésolution multiframe en utilisant le diagramme de Voronoï et la triangulation de Delaunay.

Mots clefs :

Superrésolution, théorème d'échantillonnage, interpolation, filtre de Whittaker, spline bicubique, diagramme de Voronoï, triangulation de Delaunay.

Sommaire

INTRODUCTION GENERALE	01
Chapitre I GENERALITES ET ETAT DE L'ART	03
I.1 Introduction	03
I.2 Définition et domaines d'application	03
I.3 Etat de l'art	04
I.4 Conclusion	09
Chapitre II Superrésolution monoframe par techniques d'interpolation	10
II.1 Introduction	10
II.2 Interpolation polynomiale	11
II.2.1 Interpolation par le plus proche voisin	11
II.2.2 Interpolation linéaire	13
II.2.3 Interpolation bilinéaire	17
II.2.4 Interpolation bicubique	20
II.2.5 Interpolation par spline bicubique	26
II.2.6 Comparaison entre les différentes interpolations polynomiale	31
II.2.7 Interpolation bicarrée	32
II.3 Interpolation par filtrage	33
II.3.1 Fenêtrage et Effet de Gibbs	34
II.3.2 Fenêtrage du filtre de Whittaker	36
II.3.3 Mise en œuvre de l'interpolation par filtrage	39
II.4 Conclusion	41
Chapitre III Superrésolution multiframe par techniques d'interpolation	42
III.1 Introduction	42
III.2 Superrésolution multiframe idéale	42
III.3 Superrésolution multiframe par interpolation	43
III.3.1 Intérêt apporté par l'utilisation de plusieurs imquettes	43
III.3.2 Algorithme général de la superrésolution multiframe par interpolation	45
III.4 SRM par interpolation par filtrage	47
III.5 SRM par interpolation polynomiale	48
III.5.1 Diagramme de Voronoï	49
III.5.2 Triangulation de Delaunay	52
III.5.3 L'interpolation polynomiale dans la superrésolution multiframe	56
III.6 Conclusion	65

Chapitre IV Implémentation et applications	66
IV.1 Introduction	66
IV.2 Présentation générale de l'application	66
IV.3 Environnement Superrésolution Monoframe	67
IV.3.1 La barre de Menu	68
IV.3.2 Zone de texte Historique	70
IV.3.3 Images indicatrices	71
IV.3.4 Zone de traitement	73
IV.3.5 Bouton Menu principal et bouton Reset	77
IV.4 Environnement Superrésolution Multiframe	78
IV.4.1 La barre de Menu	78
IV.4.2 Zone de texte Historique	79
IV.4.3 Bouton Ajouter une imagette	80
IV.4.4 Images indicatrices	81
IV.4.5 Bouton Reconstruction	83
IV.4.6 Panels de paramétrage	84
IV.4.7 Bouton Menu principal	88
IV.5 Environnement Développement et Etudes	88
IV.5.1 Fenêtre Optimisation superrésolution Monoframe	88
IV.5.2 Fenêtre Optimisation superrésolution Multiframe	93
IV.5.3 Fenêtre Calculs comparatifs statistiques	97
IV.5.4 Fenêtre Traitement basique des images	102
IV.6 Exemples d'études	111
IV.6.1 Images Dicom du modèle 11 et modèle 12	111
IV.6.2 Images de référence de Matlab	113
IV.6.3 Autres images	114
IV.7 Conclusion	114
CONCLUSION GENERALE	115
Bibliographie	117

Liste des figures

Figure 2.1	Surface d'interpolation	11
Figure 2.2	Interpolation par le plus proche voisin	12
Figure 2.3	Effet de l'interpolation par le plus proche voisin sur l'image initiale (facteur 5X5)	12
Figure 2.4	Interpolation linéaire d'un signal monodimensionnel	13
Figure 2.5	Triangulation du plan [X Y]	14
Figure 2.6	Comparaison entre l'interpolation par le plus proche voisin et l'interpolation linéaire avec un facteur de (4X4)	16
Figure 2.7	Principe de l'interpolation bilinéaire	17
Figure 2.8	Comparaison entre l'interpolation linéaire et l'interpolation bilinéaire	19
Figure 2.9	Comparaison entre le suréchantillonnage par l'interpolation linéaire et l'interpolation bilinéaire	20
Figure 2.10	Emplacement des 16 pixels utilisés pour l'extrapolation	21
Figure 2.11	Comparaison entre les valeurs de l'interpolation bilinéaire et de l'interpolation bicubique	24
Figure 2.12	Représentation tridimensionnelle des images suréchantillonnées de la Figure 3.11	25
Figure 2.13	Comparaison entre une interpolation bilinéaire et une interpolation bicubique	25
Figure 2.14	Zoom des zones encadrées de la Figure 3.13	26
Figure 2.15	Comparaison numérique entre l'interpolation bicubique et l'interpolation (Facteur 4X4)	30
Figure 2.16	Exemple 1 : Comparaison entre l'interpolation bicubique et par spline bicubique (Facteur 4X4)	30
Figure 2.17	Exemple 2 : Comparaison entre l'interpolation bicubique et par spline bicubique (Facteur 4X4)	31
Figure 2.18	Comparaison entre les discontinuités des surfaces d'interpolation	33
Figure 2.19	Nature oscillatoire du spectre de la fenêtre rectangulaire	35
Figure 2.20	L'effet Gibbs sur les images	36
Figure 2.21	la bande passante et le lobe secondaire d'une fenêtre de Hamming	36
Figure 2.22	Comparaison entre des spectres normalisés des fenêtres de 20 points	37
Figure 2.23	Réponses impulsionnelles des fenêtres rectangulaire, de Hann, de Hamming et de Blackman	38
Figure 2.24	Réponses impulsionnelles des filtres construits par le fenêtrage de Whittaker	38
Figure 2.25	Résultats du suréchantillonnage par filtrage	39
Figure 2.26	Influence de la taille de filtre sur l'image suréchantillonnée (K=4X4)	40
Figure 3.1	Construction d'une image à partir de 4 imagettes	43
Figure 3.2	Rôle de la superrésolution multiframe dans la diminution du recouvrement spectral	45
Figure 3.3	Principales étapes de la superrésolution multiframe	46
Figure 3.4	Organigramme des algorithmes de l'application de l'interpolation par filtrage dans la SRM	47
Figure 3.5	Diagramme de Voronoï d'une constellation de points de \mathcal{X}^2	49
Figure 3.6	Construction de diagramme de Voronoï par insertion de demi- plans	50
Figure 3.7	Construction incrémentale du diagramme de Voronoï	52
Figure 3.8	Dualité entre le diagramme de Voronoï et sa triangulation de Delaunay	53
Figure 3.9	Comparaison entre les circonscrits de la triangulation de Delaunay et des autres triangulations	53
Figure 3.10	Maximisation des angles les plus faibles de la triangulation par la triangulation de Delaunay	54
Figure 3.11	Triangulation par incrémentation	56
Figure 3.12	Exemple de la superrésolution multiframe par interpolation par le plus proche voisin	57
Figure 3.13	Principe de la superrésolution par interpolation bilinéaire	59
Figure 3.14	Exemple de la détermination des pixels utilisés dans la définition de la fonction d'interpolation	60
Figure 4.1	Fenêtre principale de l'application ENSP Superrésolution	61
Figure 4.2	Environnement Superrésolution Monoframe	67
Figure 4.3	Bouton Ouvrir	68
Figure 4.4	Bouton Zoom	68
Figure 4.5	Bouton Curseur	69
Figure 4.6	Bouton Historique / Save image actuelle	69
Figure 4.7	Fenêtres historique et image traitée	70
Figure 4.8	Zone de texte Historique	70

Figure 4.9 Décomposition de l'image originale en composantes RGB	71
Figure 4.10 Organigramme du fonctionnement de l'environnement superrésolution monoframe	73
Figure 4.11 Environnement Superrésolution Multiframe	78
Figure 4.12 Barre de menu	78
Figure 4.13 Zone de texte Historique	79
Figure 4.14 Bouton ajouter une image et son rôle	80
Figure 4.15 Espace d'affichage des imasettes	82
Figure 4.16 Espace d'affichage de l'image HRp	82
Figure 4.17 Organigramme fonctionnel de la fenêtre superrésolution multiframe	83
Figure 4.18 Panel Emplacement sur la composée	84
Figure 4.19 Panel Superrésolution	85
Figure 4.20 Fenêtre principale Développement & Etudes	88
Figure 4.21 Fenêtre Optimisation superrésolution Monoframe	89
Figure 4.22 Organigramme détaillé de l'algorithme d'optimisation monoframe	92
Figure 4.23 Panel historique	92
Figure 4.24 Fenêtre Optimisation Superrésolution Multiframe	93
Figure 4.25 Panel définition de l'imasette	94
Figure 4.26 Organigramme détaillé de l'algorithme d'optimisation multiframe	96
Figure 4.27 Exemple du panel historique et figure Résultat	97
Figure 4.28 Fenêtre calculs comparatifs statistiques	98
Figure 4.29 Organigramme détaillé de l'algorithme de comparaison	100
Figure 4.30 Les trois fenêtres créées par l'algorithme de comparaison pour afficher les résultats	101
Figure 4.31 Fenêtre Traitements basiques des images	102
Figure 4.32 Histogrammes de l'image traitée et de l'image validée	104
Figure 4.33 Comparaison entre une image avant et après l'égalisation de l'histogramme	105
Figure 4.34 Comparaison entre les gradients calculés par différents opérateurs	107
Figure 4.35 L'utilité de la localisation des extremums sur l'image contour	107
Figure 4.36 Comparaison entre des images binarisées avec différents seuils	108
Figure 4.37 L'effet des différents algorithmes de transformations morphologiques implémentées	109
Figure 4.38 Fenêtres produites par l'algorithme d'étiquetage et études des régions	110
Figure 4.39 Image contenant les deux modèles d'anévrismes :11 et 12	111
Figure 4.40 Echantillons de la base de données des images Dicom	111
Figure 4.41 Distributions des tailles optimales des fenêtres de Hamming appliquées et des erreurs moyennes des intensités sur pixels produites en fonction des facteurs de suréchantillonnage	112
Figure 4.42 Taux d'erreurs obtenues en appliquant les algorithmes de superrésolution sur des images de référence de Matlab	113
Figure 4.43 Exemple sur les images étudiées	114

Introduction Générale

INTRODUCTION GENERALE

Une image numérique n'est qu'une projection imparfaite et incomplète d'une scène donnée sur un plan. En effet, l'espace réel qui nous entoure est un espace continu construit d'une infinité de points alors qu'une image numérique est considérée comme un signal numérisé ayant des valeurs quantifiées affectées à un ensemble fini de points.

En principe, un système d'acquisition doit être élaboré pour que la qualité de l'image obtenue réponde aux besoins spécifiques de l'application, mais malheureusement, nous repérons plusieurs cas où la dégradation produite lors de l'acquisition est assez importante pour ne pas répondre à ces besoins. Quand l'amélioration de la résolution n'est plus possible par le remplacement physique de l'appareil d'acquisition pour des raisons de coût ou de limites technologiques, il est conseillé d'utiliser des techniques et des méthodes numériques de superrésolution. En effet, ces dernières peuvent représenter un alternatif à faible coût qui assure l'amélioration de la qualité de l'image et augmente la performance des autres traitements.

Une opération de superrésolution est donc, une opération de prétraitement qui nous permet d'augmenter la définition, le contraste et aussi de mieux faire apparaître les détails et les petits objets d'une image reconstruite à partir d'une ou plusieurs images faible résolution. Cela ne veut pas dire la réapparition des informations perdues lors de l'acquisition mais l'utilisation des informations atténuées ou cachées disponibles sur la scène pour compenser la dégradation produite.

Dans ce contexte, nous nous intéresserons dans ce mémoire à présenter, comparer et implémenter les différentes méthodes et techniques de superrésolution basées sur l'interpolation bidimensionnelle polynomiale et par filtrage. Pour ce faire, nous avons divisé ce travail en quatre chapitres, le premier contient des généralités sur la superrésolution et dresse un état de l'art des importants travaux et théories exploités dans ce domaine.

Dans le deuxième chapitre, nous avons détaillé et comparé les techniques de superrésolution monoframe qui exploite une seule image dans la reconstruction d'une image haute résolution en utilisant des polynômes d'interpolation ou des filtres de Whittaker.

Le troisième chapitre exprime l'intérêt général de l'utilisation de plusieurs imageries dans la reconstruction par superrésolution multiframe et propose des algorithmes d'adaptation des techniques de superrésolution monoframe par interpolation en utilisant le diagramme de Voronoï et la triangulation de Delaunay.

Enfin, nous avons implémenté, dans le dernier chapitre, l'ensemble des techniques présentées dans les chapitres précédents en réalisant une interface graphique sous Matlab dotée de programmes d'optimisation, d'exploitation et de comparaison des algorithmes de superrésolution.

Nous avons terminé notre travail par une conclusion générale où nous avons présenté les importantes remarques et conclusions extraites de notre étude.

Chapitre I

GENERALITES ET ETAT DE L'ART

Chapitre I

GENERALITES ET ETAT DE L'ART

I.1. Introduction

La résolution d'une image numérique est définie comme étant le nombre de pixels (points) utilisés dans une longueur de l'image. Cette définition nous mène à détecter la relation qui existe entre la résolution de l'image et sa qualité : Alors que la résolution détermine le nombre d'échantillons (éléments d'information) composant une image, la qualité d'une image est liée à la quantité d'informations qu'elle contient, plus cette quantité est importante plus la qualité augmente et vice versa.

La résolution peut être donc considérée comme l'une des plus importantes caractéristiques de l'image numérique ; afin de l'améliorer, deux axes de recherche sont apparus :

- Le premier axe s'intéresse à l'amélioration physique des appareils d'acquisition en développant les caractéristiques techniques des capteurs utilisés. Malgré la grande révolution dans ce domaine, cet axe souffre de plusieurs obstacles dont la difficulté d'augmenter le taux d'intégration des capteurs utilisés et d'améliorer leurs fonctions de transfert suite aux différentes limites physiques et technologiques sans oublier le coût très élevé de ces technologies de pointe.

- Le second axe utilise des algorithmes numériques de SUPERRESOLUTION qui représente une solution moins coûteuse et en cours de développement très rapide.

I.2. Définition et domaines d'application

Nous pouvons définir la *SUPERRESOLUTION* comme un ensemble de techniques et de méthodes software permettant d'obtenir une image *HR* haute résolution à partir d'une image ou de plusieurs images de résolution inférieure obtenues de la même scène. En plus de construire une image de plus grande dimension, une méthode de superrésolution sert à augmenter son contraste et à faire apparaître les petits objets plus clairs afin de pouvoir les détecter facilement.

Tous ces rôles dédiés à la superrésolution lui ont donné une grande importance dans plusieurs domaines d'applications ; citons parmi elles à titre d'exemples :

- Affichage et amélioration visuelle des images ;
- Amélioration et conversions des standards vidéo (vidéos HD) ;
- L'imagerie médicale (cellules, anévrismes...);
- L'imagerie satellitaire (bâtiments, routes, véhicules, animaux ...);
- L'imagerie astronomique (planètes, étoiles, astres) ;
- La reconnaissance d'écriture (Immatriculation...);
- La reconnaissance des personnes et des formes (visages, empreintes...);
- Détection, reconnaissance et poursuite de cibles...

I.3. Etat de l'art

La superrésolution est une problématique de racine très profonde, voir plus ancienne que l'informatique actuelle. En effet, en la définissant comme une opération de suréchantillonnage, nous constatons que nous pouvons la traiter comme étant une problématique du traitement de signal qui utilise plusieurs théories d'origine mathématiques telle que les statistiques, les probabilités, l'analyse de Fourier, les transformées et les théories d'informations.

En 1928, le suédois Harry Nyquist a montré dans son article « *Certain topics in telegraph transmission theory* » [1] la possibilité d'émission des messages sous forme d'impulsions échantillonnées avec une fréquence d'échantillonnage qui peut atteindre le double de la bande passante du système, sans considérer explicitement l'étude de la théorie d'échantillonnage et de reconstruction des signaux. En même temps, les recherches du mathématicien allemand Küpfmüller ont abouti aux mêmes résultats obtenus par Nyquist et il a pu introduire le rôle de la fonction *sinc* dans la reconstruction des signaux dans son article « *Über die Dynamik der selbsttätigen Verstärkungsregler* » [2] publié en 1928.

En 1949, Claude E Shannon a publié son fameux article « *Communication in the presence of noise* »[3] dont il a démontré le théorème d'échantillonnage en se basant sur les travaux de Nyquist et des théories d'interpolation de J.M Whittaker (1915) [4]et de E.T Whittaker (1935) [5]. En réalité, plusieurs autres publications en différentes langues ont abouti aux mêmes résultats présentés par C E Shannon avant 1949, en effet le mathématicien russe V. A Kotelnikov a publié des résultats similaires en 1933[6], de même pour l'assistant allemand de Küpfmüller H. Raabe qui a démontré le théorème d'échantillonnage dans sa thèse de Ph.D en 1939, et en 1949 Someya a établi le théorème en japonais [7].

Malgré que le théorème d'échantillonnage a démontré la possibilité de la reconstruction des signaux échantillonnés en respectant certaines conditions, cette reconstruction reste impossible puisque le filtre de reconstruction est un filtre théorique, l'utilisation des filtres tronqués produisent des perturbations au niveau du signal dues au phénomène de Gibbs, ce phénomène a été mis en évidence pour la première fois en 1848 par l'article du mathématicien anglais Wilbraham « *On a certain periodic function* » [8] où il a considéré ces perturbations comme des réflexions et des échos dus au signal principal. Ce phénomène a été remarqué aussi en 1898 sur la machine graphique du physicien américain d'origine allemande Albert Abraham Michelson, destinée au dessin des termes d'une série de Fourier, en observant la présence des amplifications au niveau des discontinuités.

Contrairement à A. Michelson qui a pensé à un problème mécanique dans la conception de sa machine, en 1899 le physico-chimiste et le mathématicien américain Josiah Willard Gibbs a montré dans son livre « *Fourier Series* » [9], pour des cas simples, que le phénomène est naturel d'origine mathématique, puis en 1906, le mathématicien américain Maxime Bôcher a présenté une analyse mathématique détaillée du phénomène dans son travail « *Introduction to the theory of Fourier's series* » [10] en le nommant « phénomène de Gibbs ».

Après la deuxième guerre mondiale, plusieurs laboratoires se sont intéressés à l'utilisation et le développement de nouvelles fonctions de fenêtrage afin de minimiser l'effet du phénomène de Gibbs. Parmi eux nous trouvons les laboratoires Bell (les leaders en matière du traitement de signal et de la transmission) qui ont développé plusieurs fenêtres dont on trouve : La fenêtre de Blackman, La fenêtre de Tukey, la fenêtre de Hann (en hommage du mathématicien autrichien Julius Ferdinand Von Hann) développées par les mathématiciens américains John Wilder Tukey et Ralph Beebe Blackman et publiées en 1959 dans leurs articles « *Particular Pairs of Windows* » [11], et la fenêtre de Hamming proposée en 1977 par le mathématicien Richard Wesley Hamming dans son livre « *Digital Filters* » [12]. Ces fenêtres représentent généralement les fenêtres les plus connues et les plus utilisées sans oublier les autres fenêtres telles que la fenêtre triangulaire, la fenêtre de Lanczos, la fenêtre de Gauss et la fenêtre de Kaiser.

L'ensemble de ces théories ont construit le noyau fondamental des algorithmes de superrésolution. Aujourd'hui, nous pouvons diviser les méthodes de superrésolution en deux classes : les méthodes de superrésolution monoframe qui utilisent une seule image pour augmenter sa résolution et les méthodes de superrésolution multiframe qui utilisent plusieurs images de la même scène dans la reconstruction d'une image de résolution supérieure. Nous pouvons aussi les classer selon le domaine de fonctionnement, en méthodes de domaine fréquentiel et méthodes du domaine spatial.

Les algorithmes de superrésolution fréquentiels ont débuté avec les premiers travaux sur la superrésolution multiframes publiés en 1984 par Tsai et Huang dans « *Advances in Computer Vision and Image Processing* » [13]. L'algorithme de Tsai et Huang se base sur l'établissement d'un système d'équations linéaires reliant les coefficients des transformées de Fourier discrètes des images basse résolution aux échantillons de la transformée de Fourier continue de la scène originale inconnue. La superrésolution est obtenue par la résolution de ce système et l'application de la transformée de Fourier discrète inverse. Cet algorithme a présenté plusieurs inconvénients tels que la supposition que l'échantillonnage des images est idéal ce qui conduit à négliger la fonction de transfert des capteurs et le bruit. En 1990, P.S. Kim et son équipe ont proposé un algorithme basé sur les calculs des moindres carrés capable d'introduire l'effet du bruit, de l'échantillonnage non idéal et la fonction de transfert des capteurs dans le système d'équations [14]. Puis, en 1993, ils ont développé un autre algorithme en se basant sur le premier et la méthode de Tikhonov[15]. En 1999 Rhee et Kang ont choisi d'utiliser la transformation en cosinus discrète (DCT) au lieu de la transformation de Fourier afin d'économiser le temps et les ressources utilisés dans les calculs [16]. Mais malgré ces tentatives d'amélioration, les algorithmes fréquentiels de superrésolution restent limités.

Concernant les algorithmes de superrésolution du domaine spatial, nous pouvons les diviser en deux principaux types : les méthodes d'interpolation, et les méthodes basées sur le modèle inverse de dégradation [17].

Les méthodes de superrésolution par interpolation sont les méthodes les plus utilisées vue leurs simplicité, facilité d'implémentation, économie en temps et ressources de calcul et efficacité. En effet, les erreurs produites par ces méthodes sont acceptables pour la majorité des applications grand public voire même pour certaines applications de précision (selon le facteur de suréchantillonnage et la méthode utilisée). Nous pouvons diviser ces méthodes d'interpolation en deux groupes : les algorithmes d'interpolation par échantillonnage multicanaux et les algorithmes d'interpolation à partir de la grille d'échantillonnage.

Les algorithmes d'interpolation par échantillonnage multicanaux utilisent le théorème d'échantillonnage généralisé : en supposant que les images basse résolution sont obtenues par un filtrage linéaire de l'image continue, la reconstruction est donc possible en utilisant un filtre d'effet inverse. En 1992, Hanoch Ur et Daniel Gauss ont proposé dans leur article « *Improved resolution from sub-pixel shifted pictures* »[18] d'utiliser un filtre de Whittaker pour effectuer la superrésolution multiframes en fusionnant les images de faibles résolutions sur la même grille puis en filtrant l'image composée par un filtre de Whittaker afin d'obtenir une image de

résolution supérieure. Suite au modèle utilisé par Shekarforoush et Chellappa en 1999 dans leur article « *Data-driven multichannel superresolution with application to video sequences* »[19], ils ont proposé d'appliquer à chaque image basse résolution un filtre inverse avant la fusion des images résultantes. En effet, chacune de ces images basse résolution peut avoir une fonction de transfert propre à elle d'où l'utilisation d'un filtre d'interpolation propre à chaque image avant la fusion pour minimiser le taux d'erreur produite.

Alors que l'interpolation par échantillonnage multicanaux utilise les filtres linéaires, l'interpolation à partir de la grille d'échantillonnage utilise des fonctions d'interpolation pour extraire les valeurs des pixels inconnus de l'image superrésolue. Les fonctions d'interpolation polynomiales ont toujours représenté les fonctions d'interpolation les plus utilisées. En effet elles sont plus simples à calculer et elles peuvent exprimer toutes les autres fonctions en développant ces dernières sous forme polynomiale par l'utilisation du théorème du développement limité « développement de Taylor ».

L'utilisation des algorithmes de superrésolution monoframe par interpolation polynomiale a débuté avec les premiers travaux sur la transmission et la représentation des images numériques. La superrésolution par le plus proche voisin et par interpolation linéaire sur une grille d'échantillonnage uniforme représentaient les premiers algorithmes exploités puisque ils nécessitent un minimum de ressources par rapport aux autres.

Seulement au début des années 70 du vingtième siècle que le besoin d'introduire de nouveaux algorithmes de superrésolution plus fins est apparu afin de minimiser les discontinuités produites lors de l'agrandissement des images. L'une des premières applications dans ce domaine était la rectification géométrique des images numériques obtenues par le premier satellite des Technologies des Ressources terrestres « Earth Resources Technology Satellite » lancé en 1972 par le US.NASA « United States National Aeronautics and Space Administration » où K.W. Simon a été mené, en 1975, à utiliser, pour la première fois, des polynômes d'interpolation cubiques et bicubiques pour augmenter la résolution des images obtenues [20]. Ces algorithmes ont construit l'intérêt de plusieurs travaux de recherche afin de les optimiser et les améliorer avant de les représenter sous leur forme actuelle [21].

Peu après, en 1978, Hou et Andrews ont introduit l'utilisation de l'interpolation par spline bicubique pour la superrésolution « *Cubic splines for image interpolation and digital filtering* »[22] en exploitant la notion des fonctions polynomiales spline « cannelure » et leurs propriétés définies par le mathématicien I. J. Schoenberg en 1946 dans « *Contributions to the problem of approximation of equidistant data by analytic functions.* »[23]. L'étude comparative

sur la superrésolution polynomiale monoframe « *Comparison of interpolating methods for image resampling* » [24] publiée en 1983 par J. A. Parker, R. V. Kenyon et D. E. Troxel a montré que l'interpolation par spline bicubique est plus lisse et produit généralement moins d'erreurs comparée à l'interpolation par le plus proche voisin, l'interpolation linéaire et l'interpolation bicubique. Cette étude a représenté le début d'une série d'autres comparaisons dont nous trouvons l'étude de J.L Ostuni et son équipe « *Analysis of interpolation effects in the reslicing of functional MR images* » [25] publiée en 1997 dans laquelle il ont comparé la superrésolution par l'interpolation par le plus proche voisin, par interpolation bilinéaire, par spline bicubique et par filtrage de Whittaker fenêtré avec une fenêtre de Hann sur des images IRM où ils ont conclu que l'interpolation par spline bicubique représente le plus faible taux d'erreurs pour les images utilisées alors qu'elle consomme de quatre à six fois de ressources et de temps de calcul par rapport aux autres. En 1999, T. M. Lehmann, C. Gönner, et K. Spitzer ont publié « *Survey: Interpolation methods in medical image processing* » [26] où ils ont comparé tous les algorithmes d'interpolation connues sur des images acquises par les différents appareils : CT, MRI, PET, SPECT, 3-D rotational et par rayon X, et ils ont remarqué que le taux d'erreurs produit par l'interpolation par spline bicubique est généralement le plus faible.

Alors que l'interpolation monoframe utilise généralement une grille d'interpolation uniforme, la superrésolution multiframe utilise une grille non uniforme d'interpolation, cela a posé un grand problème devant son développement. En effet, l'optimisation des fonctions d'interpolation est reliée au choix de la disposition des pixels sources (pixels connus) utilisés. Les premiers algorithmes de choix des pixels étaient trop lents et demandent trop de ressources dont nous citons l'algorithme itératif de K.D.Sauer et J.P.Allenbah proposé en 1987 [27]. Cet algorithme a été amélioré en introduisant la projection de la grille sur des ensembles convexes par H.Stark et S.Yeh en 1990 [28] puis en développant un algorithme plus rapide basé sur l'idée précédente et la transformée en FFT par D.Granrath et J.Lersch en 1998. [29]

Au début du 21^{ème} siècle de nouvelles méthodes d'interpolation sur une grille non uniforme sont développées : En 2000, T.E.Boult et M.C.Chiang ont exposé, dans leur article « *Efficient Superresolution via Image Warping* » [30], l'utilisation des techniques basées sur les fonctions de transfert optiques dans l'interpolation, alors que N.Nguyen et P.Milanfar ont développé des algorithmes de superrésolution en utilisant les ondelettes.[31]. En 2002, S. Lertrattanapanich et N. K. Bose ont utilisé la triangulation de Delaunay dans la définition des pixels sources intervenant pour l'extrapolation de chaque pixel de l'image superrésolue [32], et en 2006 T.Saito et son équipe ont amélioré les résultats d'interpolation par l'introduction d'un traitement spécifique au niveau des hautes fréquences [33].

La superrésolution par des algorithmes basés sur le modèle inverse utilise la modélisation de l'opération d'acquisition des images de basse résolution à partir d'une image haute résolution (ou d'une image analogique) en prenant en considération toutes les erreurs dues aux différents bruits et déformations produites lors de l'acquisition. Une fois le modèle connu, on calcule les images de haute résolution en estimant le modèle inverse. Parmi les techniques statistiques développées pour la modélisation nous citons celle de « simulation puis correction » [34], [35], [36] basée sur la correction du modèle suite à une procédure d'apprentissage jusqu'à la définition d'un modèle optimisé, la technique « Maximum de vraisemblance » [37] et la technique « Estimation bayésienne » qui permet d'introduire des conditions a priori [38],[39].

I.4. Conclusion

Ce chapitre présente des informations générales sur la superrésolution ainsi qu'un état de l'art interprétant les importants travaux élaborés dans ce domaine. À travers le nombre de ces travaux nous pouvons sentir l'importance dédiée à l'opération de superrésolution comme une opération de prétraitement qualifiée d'augmenter la qualité des images en augmentant leur résolution.

Nous pouvons remarquer que les techniques de superrésolution par interpolation sont les techniques les plus favorisées et les plus utilisées vue leurs simplicité, facilité d'implémentation, économie en temps et ressources de calcul et efficacité. Tous ces avantages nous ont motivé pour consacrer ce travail à l'étude et l'implémentation de ces techniques.

Pour cela, nous allons consacrer le chapitre suivant à la présentation et la comparaison des différentes techniques pratiques de superrésolution monoframe par interpolation permettant d'obtenir des images haute résolution avec un minimum de distorsion.

Chapitre II

**SUPERRESOLUTION MONOFRAME PAR
TECHNIQUES D'INTERPOLATION**

Chapitre II

SUPERRESOLUTION MONOFRAME PAR TECHNIQUES D'INTERPOLATION

II.1 Introduction

La résolution des images numériques a toujours représenté une grande importance dans l'analyse et l'exploitation de ses informations, en effet, plus la résolution est importante, plus l'image est riche en détails, donc d'informations. Malheureusement, malgré la grande révolution connue dernièrement dans l'amélioration des appareils d'acquisition, ces derniers restent incapables de répondre à nos besoins suite à leurs limites physiques et technologiques.

Pour cela, on a opté pour les algorithmes de superrésolution en procédant au développement et à l'exploitation des théorèmes et des algorithmes de la reconstruction et du suréchantillonnage (voir Annexe I).

Les opérations de reconstruction de l'image analogique et du suréchantillonnage de l'image numérique ne peuvent être pratiquement réalisées sans distorsions et sans erreurs de manière idéale. La solution était de développer des algorithmes d'interpolation qui augmentent la résolution de l'image avec un niveau de distorsion et un taux d'erreurs acceptables.

L'interpolation consiste à trouver un modèle mathématique défini par la courbe d'une fonction d'interpolation construite à partir des échantillons déterminés dits échantillons d'interpolation ou points sources. Dans le cas des images, cette courbe est appelée surface d'interpolation. La Figure 2.1 montre un modèle de surface d'interpolation construite à partir des échantillons d'interpolation afin d'estimer les valeurs des échantillons non connus dits échantillons extrapolés par une opération d'extrapolation.

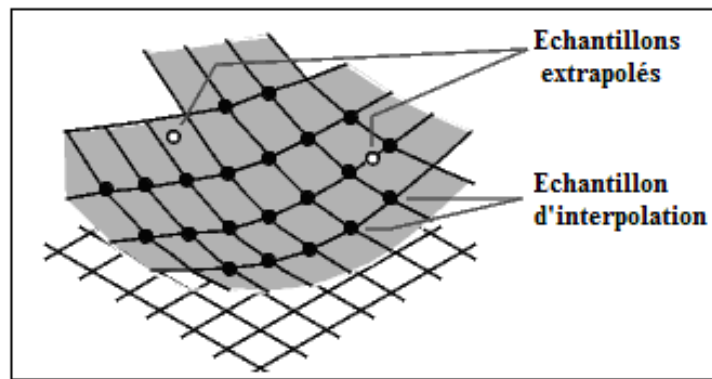


Figure 2. 1 Surface d'interpolation.

En général, chaque opération d'interpolation est précédée d'une opération d'ajout de zéro ou « zero padding », qui consiste à ajouter $(Kx-1) \times (Ky-1)$ zéros entre chaque quatre pixels sources de l'image initiale $i(x,y)$ à la place des échantillons extrapolés dans l'image suréchantillonnée d'un facteur de $Kx \times Ky$.

Les opérations de reconstruction ou de suréchantillonnage peuvent être faites à partir d'une ou de plusieurs images de faibles résolutions. Ce chapitre sera consacré au suréchantillonnage exploitant une seule image (superrésolution monoframe).

Nous nous intéresserons, tout au long de ce travail, au suréchantillonnage par des algorithmes d'interpolation polynomiale et d'interpolation par filtrages linéaires :

II.2 Interpolation polynomiale

L'interpolation polynomiale est la méthode d'interpolation la plus utilisée, elle se base sur la détermination des fonctions « surfaces » d'interpolation polynomiale sous la forme suivante :

$$f_{\text{int}}(x, y) = \sum_n \sum_m a_{nm} x^n y^m \quad (2.1)$$

Où n, m sont des nombres entiers

Dans ce qui suit, nous allons exposer les méthodes d'interpolation polynomiale les plus utilisées en superrésolution des images.

II.2.1 Interpolation par le plus proche voisin

Connue, aussi, sous le nom de **Interpolateur Polynomiale d'Ordre Zéro**, ce type d'interpolateur est très utilisé dans l'affichage des images numériques sur les écrans à forte résolution et les zooms des images. Son algorithme est basé sur le même principe que l'échantillonneur bloqueur.

Soit $i(x,y)$ l'image initiale de dimension $N \times M$ que l'on veut suréchantillonner avec un facteur $Kx \times Ky$ en utilisant l'algorithme d'interpolation par le plus proche voisin : Après une

opération de zero padding, on remplace les zéros ajoutés par la valeur de l'échantillon de l'image originale le plus proche.

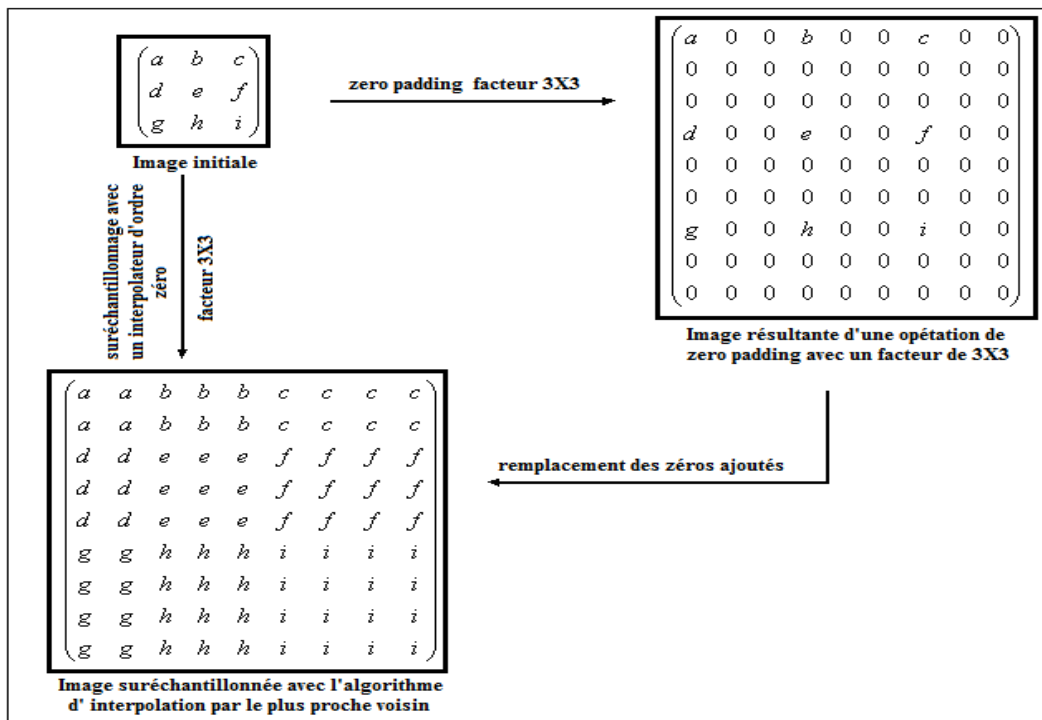


Figure2. 2 Interpolation par le plus proche voisin.

Cette méthode de suréchantillonnage a le mérite d'être la méthode la plus rapide et la plus facile à implémenter puisqu'aucun calcul de valeurs intermédiaires n'est effectué. En revanche, elle détermine une zone d'influence exclusive pour chaque point source, ce qui va accentuer la sensation de discontinuité des formes en formant de larges paliers en escalier.

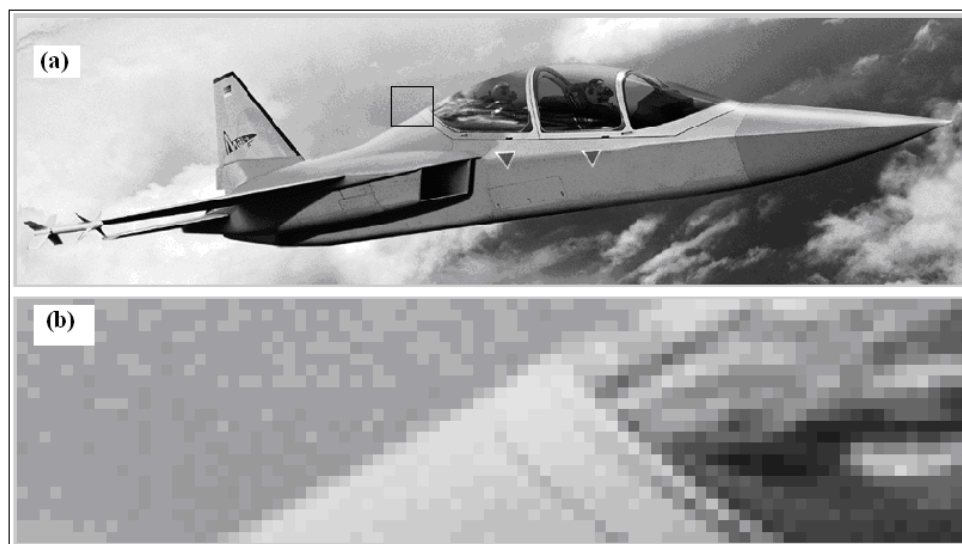


Figure 2. 3 Effet de l'interpolation par le plus proche voisin sur l'image initiale (facteur 5X5).

(a) Image initiale (b) Image suréchantillonnée avec l'interpolateur d'ordre zéro « apparition de palier en escalier »

II.2.2 Interpolation linéaire

Une interpolation linéaire, dite aussi interpolation d'ordre 1, d'un signal monodimensionnel consiste à calculer les valeurs intermédiaires ajoutées $y(x)$ entre deux échantillons successifs connus du signal initial $y(x_0)$ et $y(x_1)$ tel que $(x,y(x))$ soit un point du segment de droite $[(x_0,y(x_0)),(x_1,y(x_1))]$.

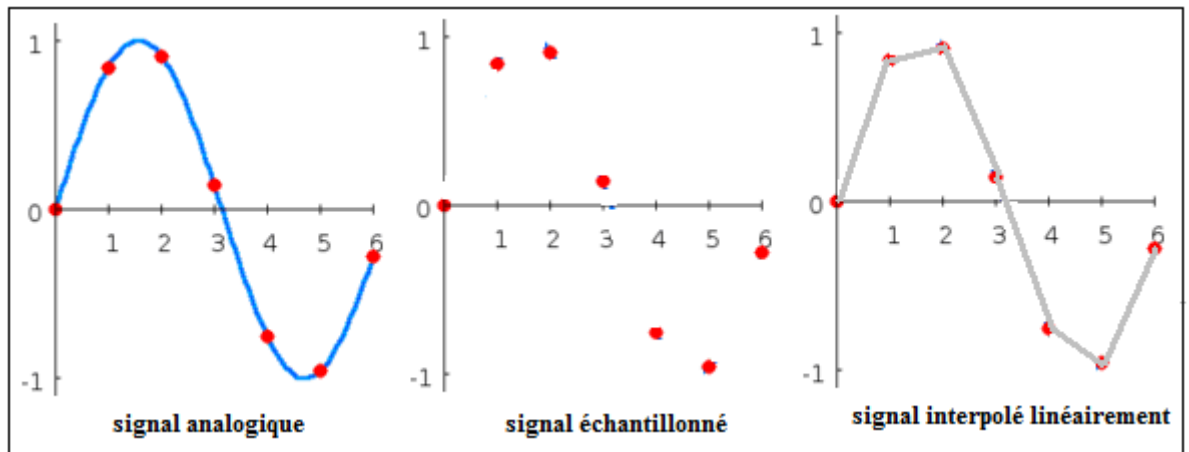


Figure 2. 4 Interpolation linéaire d'un signal monodimensionnel.

Cela se traduit mathématiquement par

$$\forall x \in [x_0, x_1]$$

$$f(x) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + f(x_0) \tag{2.2}$$

Pour un signal bidimensionnel, tel que les images, l'interpolation linéaire utilise des plans d'interpolation au lieu d'utiliser des droites afin de pouvoir déterminer les valeurs intermédiaires à extrapoler.

Or l'équation d'un plan dans l'espace s'écrit sous la forme :

$$f(x, y) = ax + by + c \tag{2.3}$$

où a,b,c sont trois variables réelles.

Donc, la détermination d'un plan d'interpolation nécessite la connaissance de trois points appartenant à ce plan. Dans le cas de l'interpolation linéaire, ces points sont des points sources de l'image initiale délimitant les plans d'interpolation ; nous devons donc procéder à une opération de triangulation.

La triangulation du plan $[X \ Y]$ signifie le maillage et la décomposition de ce plan en plusieurs triangles élémentaires adjacents tel que l'union de tous ces triangles construit le plan ; nous pouvons dire aussi que la triangulation signifie que quelque soit le point (x,y) appartenant au plan (P) triangulé, il existe un et un seul triangle de cette triangulation qui peut contenir ce point.

Soit une image $i(x,y)$ qu'on veut suréchantillonner par interpolation linéaire. Puisque la distribution des pixels d'une image dans le plan $[X Y]$ est une distribution uniforme, nous pouvons donc la trianguler comme le montre la Figure 2.5.

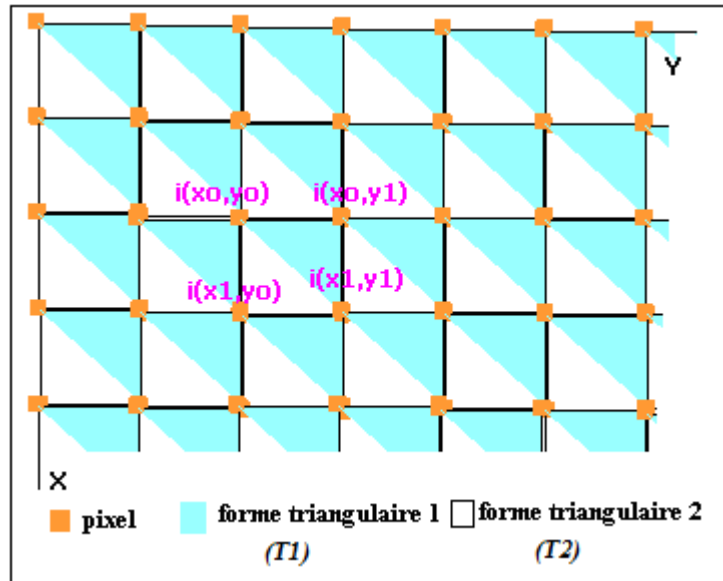


Figure 2. 5 Triangulation du plan $[X Y]$.

La triangulation de cette distribution nous permet d'avoir deux formes triangulaires élémentaires entre chaque quatre pixels adjacents, chaque forme triangulaire est définie par l'équation de son plan.

Soit $p(x_0,y_0)$, $p(x_0,y_1)$, $p(x_1,y_1)$ et $p(x_1,y_0)$, les positions des quatre pixels adjacents définissant deux triangles (T1) et (T2) et soit $i(x_0,y_0)$, $i(x_0,y_1)$, $i(x_1,y_1)$ et $i(x_1,y_0)$ leurs valeurs, comme le montre la Figure 2.5.

Afin d'extrapoler la valeur d'un pixel $i(x,y)$ existant entre les quatre pixels $i(x_0,y_0)$, $i(x_0,y_1)$, $i(x_1,y_1)$ et $i(x_1,y_0)$ à partir d'une interpolation linéaire, il est nécessaire de déterminer à quel triangle il appartient. En effet, chaque pixel $p(x,y)$ doit faire partie de l'un des triangles (T1) ou (T2), donc sa valeur est extrapolée à partir de l'équation du plan auquel il appartient.

- **Cas où le point $p(x,y)$ appartient au triangle (T1)**

Le point $p(x,y)$ appartient au triangle (T1) limité par les positions $p(x_0,y_0)$, $p(x_0,y_1)$, $p(x_1,y_1)$ signifie que :

$$\left\{ \begin{array}{l} (x, y) \in [x_0, x_1] \times [y_0, y_1] \\ y \leq \frac{y_1 - y_0}{x_1 - x_0} (x - x_0) + y_0 \end{array} \right.$$

Pour déterminer l'équation du premier plan d'interpolation (P1) relatif au triangle (T1), on remplace les valeurs des pixels $i(x_0,y_0)$, $i(x_0,y_1)$ et $i(x_1,y_1)$ dans la relation (2.3):

$$i(x, y) = ax + by + c$$

On aura le système d'équations suivant :

$$\begin{cases} i(x_0, y_0) = ax_0 + by_0 + c \\ i(x_0, y_1) = ax_0 + by_1 + c \\ i(x_1, y_1) = ax_1 + by_1 + c \end{cases} \quad (2.4)$$

Par le changement de variables

$$\begin{cases} x' = \frac{(x - x_0)}{(x_1 - x_0)} \\ y' = \frac{(y - y_0)}{(y_1 - y_0)} \\ f(x', y') = i(x, y) \end{cases} \quad (2.5)$$

Le système (2.4) s'écrira :

$$\begin{cases} f(0,0) = a.0 + b.0 + c \\ f(0,1) = a.0 + b.1 + c \\ f(1,1) = a.1 + b.1 + c \end{cases}$$

$$\Rightarrow \begin{cases} c = f(0,0) \\ b = f(0,1) - c \\ a = f(1,1) - b - c \end{cases}$$

$$\Rightarrow \begin{cases} a = f(1,1) - f(0,1) \\ b = f(0,1) - f(0,0) \\ c = f(0,0) \end{cases}$$

Donc on aura en conclusion:

$$p(x,y) \in (T1) \Leftrightarrow i(x,y) = \frac{i(x_1,y_1) - i(x_0,y_1)}{(x_1 - x_0)}(x - x_0) + \frac{i(x_0,y_1) - i(x_0,y_0)}{(y_1 - y_0)}(y - y_0) + i(x_0,y_0) \quad (2.6)$$

- Cas où le point $p(x,y)$ appartient au triangle (T2)

Le point $p(x,y)$ appartient au triangle (T2) limité par les positions $p(x_0,y_0)$, $p(x_1,y_0)$, $p(x_1,y_1)$ signifie que :

$$\begin{cases} (x, y) \in [x_0, x_1] \times [y_0, y_1] \\ \wedge \\ y \geq \frac{y_1 - y_0}{x_1 - x_0} (x - x_0) + y_0 \end{cases}$$

En remplaçant les coordonnées des points limitant le triangle (T2) dans l'équation du plan (P2), on aura :

$$\begin{cases} i(x_0, y_0) = ax_0 + by_0 + c \\ i(x_1, y_0) = ax_1 + by_0 + c \\ i(x_1, y_1) = ax_1 + by_1 + c \end{cases}$$

En suivant le même développement fait pour l'extrapolation de la valeur d'un pixel du triangle (T1), la valeur d'un point $p(x,y)$ appartient à (T2) est donnée par :

$$p(x,y) \in (T2) \Leftrightarrow i(x,y) = \frac{i(x_1, y_0) - i(x_0, y_0)}{(x_1 - x_0)}(x - x_0) + \frac{i(x_1, y_1) - i(x_1, y_0)}{(y_1 - y_0)}(y - y_0) + i(x_0, y_0) \quad (2.7)$$

La Figure 2.6 représente une comparaison, avec des représentations mathématiques, entre un suréchantillonnage avec une interpolation par le plus proche voisin et un suréchantillonnage avec une interpolation linéaire avec un facteur de 4×4

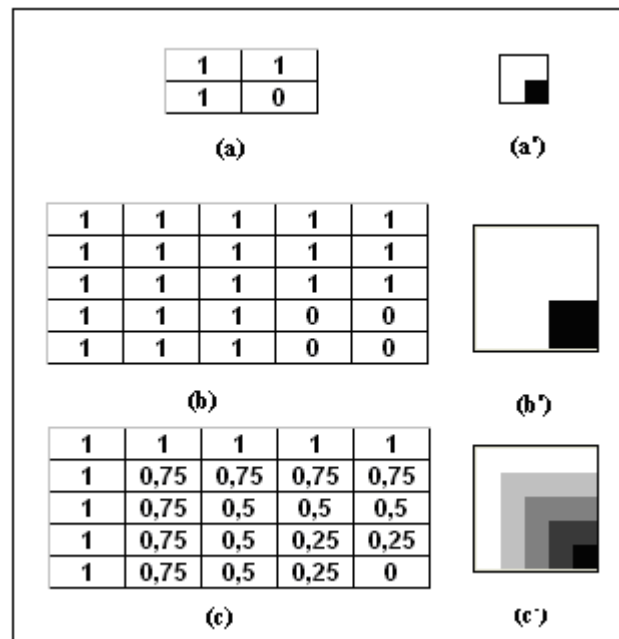


Figure 2.6 Comparaison entre l'interpolation par le plus proche voisin et l'interpolation linéaire avec un facteur de (4X4).

(a) Représentation matricielle de l'image initiale. (a') L'image initiale.

(b) Représentation matricielle de l'image interpolée par le plus proche voisin.

(b') Image interpolée par le plus proche voisin.

(c) Représentation matricielle de l'image interpolée de (a) linéairement. (c') Image interpolée linéairement.

En comparant le suréchantillonnage par l'interpolation par le plus proche voisin avec le suréchantillonnage par l'interpolation linéaire, nous remarquons que l'utilisation de l'interpolation linéaire diminue la largeur des paliers sans être suffisante pour les éliminer, elle présente aussi une discontinuité moins apparente que celle de l'interpolation par le plus proche voisin sauf au niveau de l'intersection entre les deux triangles où on risque d'avoir une forte discontinuité.

II.2.3 Interpolation bilinéaire

L'interpolation bilinéaire est une autre méthode dérivée de l'interpolation linéaire monodimensionnelle, elle est utilisée pour le calcul des valeurs intermédiaires extrapolées lors d'une opération de suréchantillonnage des fonctions de deux variables x et y uniformément distribuées sur le plan $[X Y]$.

Malgré qu'elles soient de même source, le principe de l'interpolation bilinéaire est différent de celui de l'interpolation linéaire. En effet, l'interpolation bilinéaire se base sur la double interpolation monodimensionnelle suivant l'un des axes puis sur le deuxième axe. Donc la surface élémentaire d'interpolation est délimitée par quatre points sources adjacents qui l'entourent

Soit $p(x_0, y_0)$, $p(x_0, y_1)$, $p(x_1, y_1)$ et $p(x_1, y_0)$ quatre points sources adjacents à la surface d'interpolation ayant les valeurs $i(x_0, y_0)$, $i(x_0, y_1)$, $i(x_1, y_1)$ et $i(x_1, y_0)$, et soit $i(x, y)$ la valeur du pixel de position $p(x, y)$ qu'on veut extrapoler à partir d'un suréchantillonnage par interpolation bilinéaire.

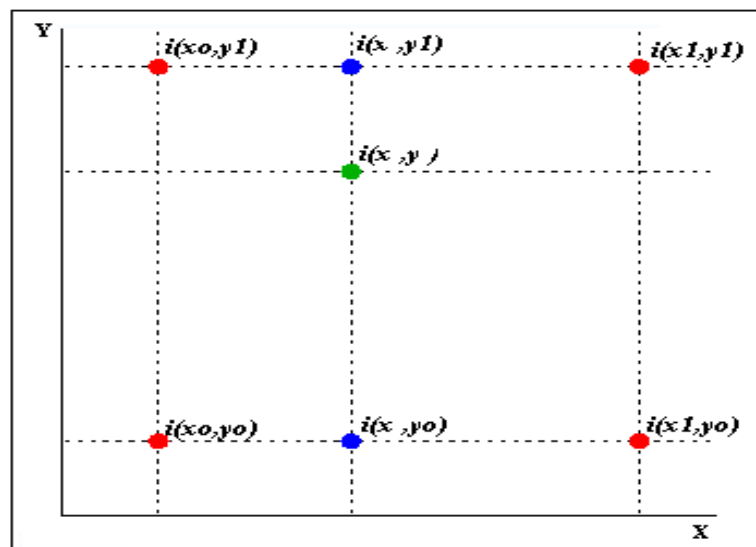


Figure 2. 7 Principe de l'interpolation bilinéaire.

Pour extrapoler la valeur du pixel $i(x, y)$, nous devons faire deux interpolations linéaires monodimensionnelles suivant les deux axes séparément : choisissons de commencer par une interpolation linéaire suivant l'axe X puis une autre suivant l'axe Y.

Afin de déterminer la valeur de $i(x, y)$, nous devons extrapoler d'abord la valeur du point $i(x, y_0)$ à partir des deux points $i(x_0, y_0)$ et $i(x_1, y_0)$ par une interpolation linéaire. En exploitant la relation (2.2), on aura :

$$i(x, y_0) = \frac{i(x_1, y_0) - i(x_0, y_0)}{x_1 - x_0} (x - x_0) + i(x_0, y_0) \quad (2.8)$$

De même, nous extrapolons le point $i(x, y_1)$ à partir des deux points sources $i(x_0, y_1)$ et $i(x_1, y_1)$:

$$i(x, y_1) = \frac{i(x_1, y_1) - i(x_0, y_1)}{x_1 - x_0} (x - x_0) + i(x_0, y_1) \quad (2.9)$$

Enfin, l'extrapolation de la valeur de $i(x, y)$, à partir d'une interpolation linéaire suivant l'axe Y en utilisant les deux points déjà extrapolés $i(x, y_0)$ et $i(x, y_1)$, donne :

$$i(x, y) = \frac{i(x, y_1) - i(x, y_0)}{y_1 - y_0} (y - y_0) + i(x, y_0) \quad (2.10)$$

En remplaçant les valeurs de $i(x, y_0)$ et $i(x, y_1)$ dans (2.10), nous trouvons :

$$\begin{aligned} i(x, y) = & \frac{i(x_0, y_0)}{(x_1 - x_0)(y_1 - y_0)} (x - x_0)(y - y_0) + \frac{i(x_0, y_1)}{(x_1 - x_0)(y_1 - y_0)} (x - x_0)(y_1 - y) \\ & + \frac{i(x_1, y_0)}{(x_1 - x_0)(y_1 - y_0)} (x_1 - x)(y - y_0) + \frac{i(x_1, y_1)}{(x_1 - x_0)(y_1 - y_0)} (x_1 - x)(y_1 - y) \end{aligned} \quad (2.11)$$

En normalisant la grille par le changement de variables de la relation (2.5), (2.11) s'écrira :

$$f(x', y') = f(0,0)(1 - x')(1 - y') + f(1,0)x'(1 - y') + f(0,1)(1 - x')y' + f(1,1)x'y' \quad (2.12)$$

Ou sous la forme matricielle :

$$f(x', y') = \begin{bmatrix} 1 - x' & x' \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1 - y' \\ y' \end{bmatrix} \quad (2.13)$$

Cette écriture peut être simplifiée en :

$$f(x', y') = b_1 + b_2x' + b_3y' + b_4x'y' \quad (2.14)$$

avec :

$$\begin{cases} b_1 = f(0,0) \\ b_2 = f(1,0) - f(0,0) \\ b_3 = f(0,1) - f(0,0) \\ b_4 = f(0,0) + f(1,1) - f(1,0) - f(0,1) \end{cases}$$

Nous obtenons le même résultat en commençant l'interpolation par l'axe Y suivie d'une interpolation par l'axe X.

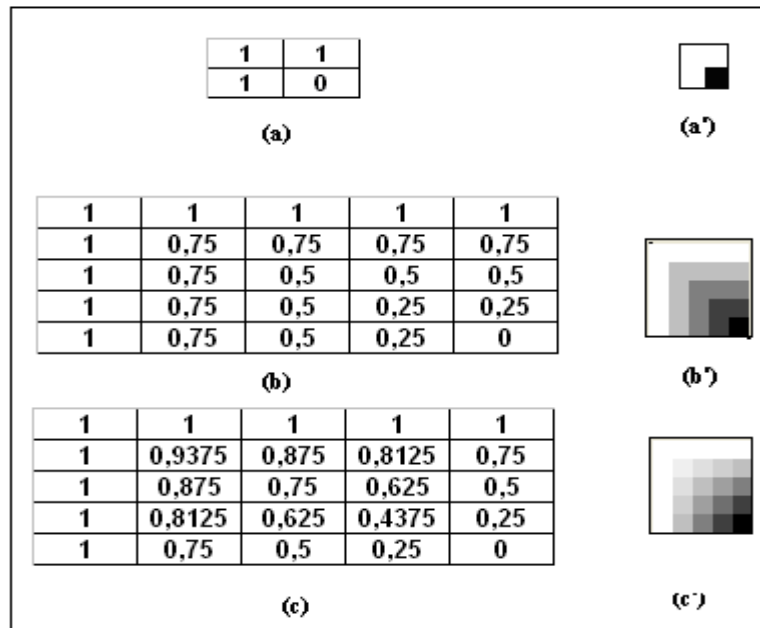


Figure 2. 8 Comparaison entre l'interpolation linéaire et l'interpolation bilinéaire.

- (a) Représentation matricielle de l'image initiale. (a') L'image initiale.
 (b) Représentation matricielle de l'image interpolée de (a) linéairement. (b') Image interpolée linéairement.
 (c) Représentation matricielle de l'image interpolée de (a) bilinéairement. (c') Image interpolée bilinéairement.

La Figure 2.8 représente une comparaison entre une image interpolée linéairement et une autre interpolée bilinéairement avec un facteur de 4×4 . Nous remarquons que l'effet d'escalier a pratiquement disparu dans l'image interpolée bilinéairement.

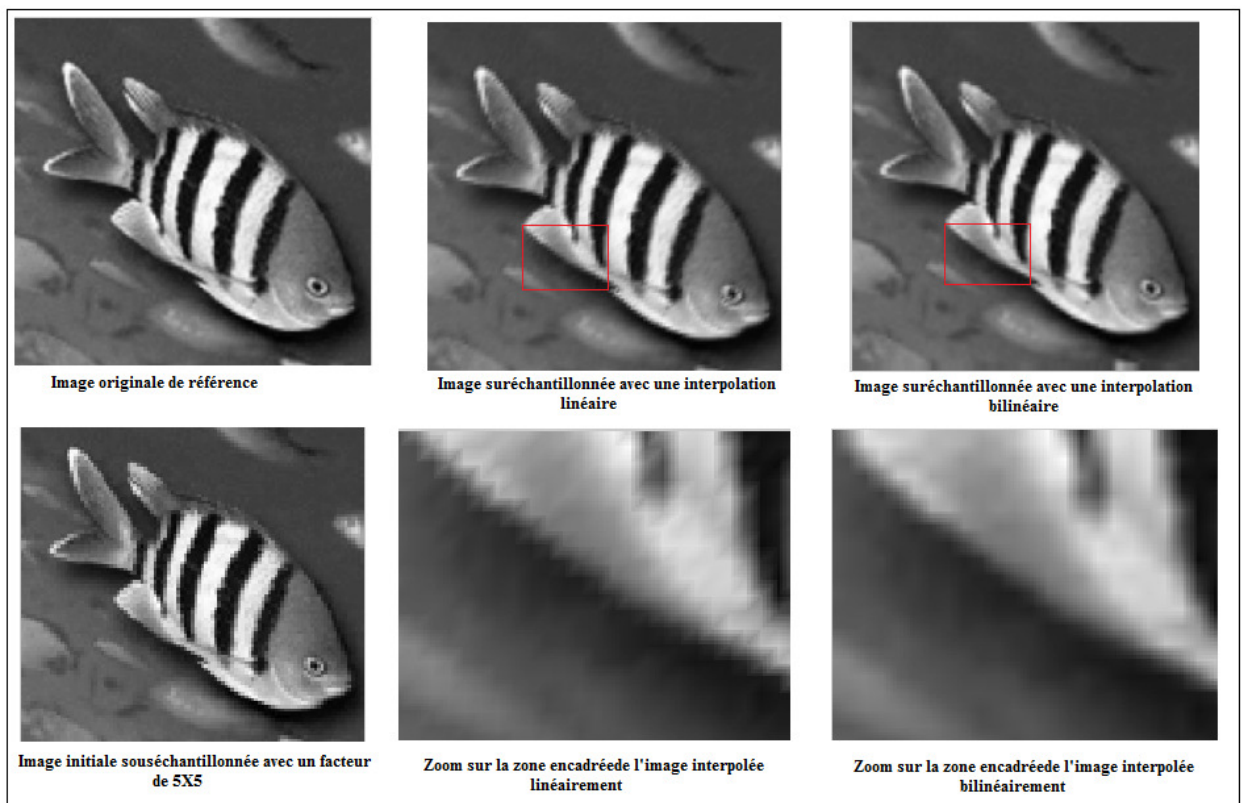


Figure 2. 9 Comparaison entre le suréchantillonnage par l'interpolation linéaire et l'interpolation bilinéaire.

La Figure 2.9 représente une comparaison entre une image de référence et des images suréchantillonnées après une opération de souséchantillonnage d'un facteur de 5X5, une fois par interpolation linéaire et une autre fois par interpolation bilinéaire.

En analysant les deux zones zoomées de la Figure 2.9, nous remarquons que l'interpolation bilinéaire présente moins d'effet d'escalier qui est presque éliminé et moins de discontinuité par rapport à l'interpolation linéaire.

II.2.4 Interpolation bicubique

Une autre méthode d'interpolation utilisant les interpolations monodimensionnelles, l'interpolation bicubique se réalise par la double interpolation cubique monodimensionnelle sur les deux axes. Le polynôme interpolateur cubique est un polynôme de degré trois de la forme :[40]

$$f(x) = ax^3 + bx^2 + cx + d \quad (2.15)$$

avec a, b, c, d quatre constantes réelles.

Donc, la détermination du polynôme d'interpolation nécessite l'usage de quatre points sources : pour chaque $x \in [x_0, x_1]$, $f(x)$ est calculée à partir des points $((x_{-1}, f(x_{-1})), (x_0, f(x_0)), (x_1, f(x_1))$ et $(x_2, f(x_2))$.

De même que l'interpolation bilinéaire, le développement de l'algorithme d'interpolation bicubique montre que l'équation de la surface d'interpolation est une équation quadratique de la forme :

$$i(x, y) = \sum_{n=0}^3 \sum_{m=0}^3 a_{nm} x^n y^m \quad (2.16)$$

où a_{ij} représente 16 constantes réelles.

La détermination de ce polynôme revient à la détermination des coefficients a_{ij} par la résolution d'un système d'équations composé, à partir des points sources, de 16 équations linéaires de la forme (2.16).

Les 16 points sources utilisés pour le calcul des coefficients de polynôme d'interpolation sont les points qui entourent la surface d'interpolation (Voir Figure 2.10)

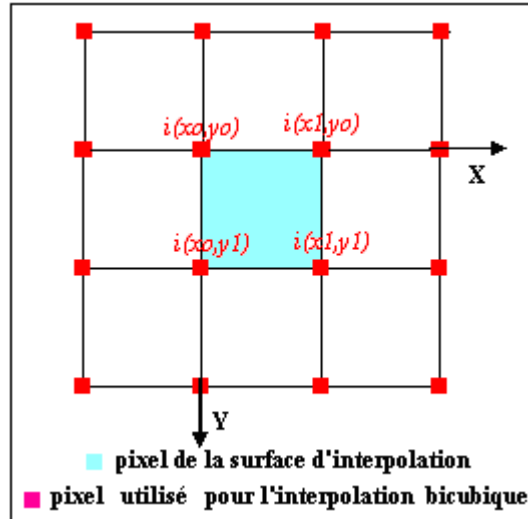


Figure 2. 10 Emplacement des 16 pixels utilisés pour l'extrapolation.

Le système d'équations sera construit par les 16 équations :

$$\begin{cases} i(x_i, y_j) = \sum_{n=0}^3 \sum_{m=0}^3 a_{nm} x_i^n y_j^m \end{cases} \quad (2.17)$$

Avec i, j des variables entières qui varient de -1 à 2.

En utilisant le changement de variables de la relation (2.5), le polynôme d'interpolation devient :

$$i(x, y) = \sum_{n=0}^3 \sum_{m=0}^3 a_{nm} x^n y^m \xrightarrow{(3.5)} f(x', y') = \sum_{n=0}^3 \sum_{m=0}^3 a_{nm} x'^n y'^m \quad (2.18)$$

Et le système (2.17) s'écrira :

$$\begin{cases} f(i, j) = \sum_{n=0}^3 \sum_{m=0}^3 a_{nm} i^n j^m \end{cases} \quad (2.19)$$

avec i, j des variables entières qui varient de (-1) à (2).

En résolvant (2.19) et en faisant le changement de variables suivant :

$$\begin{cases} \frac{\partial f(x_i, y_j)}{\partial x} = \frac{f(x_i, y_j) - f(x_{i-1}, y_j)}{x_i - x_{i-1}} \\ \frac{\partial f(x_i, y_j)}{\partial y} = \frac{f(x_i, y_i) - f(x_i, y_{i-1})}{y_i - y_{i-1}} \\ \frac{\partial^2 f(x_i, y_j)}{\partial x \partial y} = \frac{2f(x_i, y_j) - f(x_{i+1}, y_{i+1}) - f(x_{i-1}, y_{i-1})}{(x_i - x_{i-1})(y_i - y_{i-1})} \end{cases} \quad (2.20)$$

Nous trouvons : [17]

$$\begin{aligned}
 f(x', y') = & (1 - y'^2(3 - 2y'))(1 - x'^2(3 - 2x'))f(0,0) + (1 - y'^2(3 - 2y'))x'^2(3 - 2x')f(1,0) \\
 & + y'^2(3 - 2y')x'^2(3 - 2x')f(1,1) + y'^2(3 - 2y')(1 - x'^2(3 - 2x'))f(0,1) \\
 & + (1 - y'^2(3 - 2y'))x'(x'-1)^2 \frac{\partial f(0,0)}{\partial x'} + (1 - y'^2(3 - 2y'))x'^2(x-1) \frac{\partial f(1,0)}{\partial x'} \\
 & + y'^2(3 - 2y')x'^2(x'-1) \frac{\partial f(1,1)}{\partial x'} + y'^2(3 - 2y')x'(x'-1)^2 \frac{\partial f(0,1)}{\partial x'} \\
 & + y'(y'-1)^2(1 - x'^2(3 - 2x')) \frac{\partial f(0,0)}{\partial y'} + y'(y'-1)^2x'^2(3 - 2x') \frac{\partial f(1,0)}{\partial y'} \\
 & + y'^2(y'-1)x'^2(3 - 2x') \frac{\partial f(1,1)}{\partial y'} + y'^2(y'-1)(1 - x'^2(3 - 2x'))^2 \frac{\partial f(0,1)}{\partial y'} \\
 & + x'(x'-1)^2y'(y'-1)^2 \frac{\partial^2 f(0,0)}{\partial x' \partial y'} + x'^2(x'-1)y'(y'-1)^2 \frac{\partial^2 f(1,0)}{\partial x' \partial y'} \\
 & + x'^2(x'-1)y'^2(y'-1) \frac{\partial^2 f(1,1)}{\partial x' \partial y'} + x'(x'-1)^2y'^2(y'-1) \frac{\partial^2 f(0,1)}{\partial x' \partial y'}
 \end{aligned} \tag{2.21}$$

En faisant le changement de variables suivant :

$$\begin{cases}
 \frac{\partial f(x', y')}{\partial x'} = \frac{\partial i(x, y)}{\partial x} \\
 \frac{\partial f(x', y')}{\partial y'} = \frac{\partial i(x, y)}{\partial y} \\
 \frac{\partial^2 f(x', y')}{\partial x' \partial y'} = \frac{\partial^2 i(x, y)}{\partial x \partial y}
 \end{cases} \tag{2.22}$$

tel que :

$$\begin{cases}
 \frac{\partial i(x_i, y_j)}{\partial x} = \frac{i(x_i, y_j) - i(x_{i-1}, y_j)}{x_i - x_{i-1}} \\
 \frac{\partial i(x_i, y_j)}{\partial y} = \frac{i(x_i, y_j) - i(x_i, y_{j-1})}{y_j - y_{j-1}} \\
 \frac{\partial^2 i(x_i, y_j)}{\partial x \partial y} = \frac{2i(x_i, y_j) - i(x_{i+1}, y_{j+1}) - i(x_{i-1}, y_{j-1})}{(x_i - x_{i-1})(y_j - y_{j-1})}
 \end{cases}$$

Nous aurons :

$$\begin{aligned}
 (3.2) \Rightarrow i(x, y) = & \left(1 - \left(\frac{y-y_0}{y_1-y_0} \right)^2 \left(3 - 2 \left(\frac{y-y_0}{y_1-y_0} \right) \right) \right) \left(1 - \left(\frac{x-x_0}{x_1-x_0} \right)^2 \left(3 - 2 \left(\frac{x-x_0}{x_1-x_0} \right) \right) \right) i(x_0, y_0) \\
 & + \left(1 - \left(\frac{y-y_0}{y_1-y_0} \right)^2 \left(3 - 2 \left(\frac{y-y_0}{y_1-y_0} \right) \right) \right) \left(\frac{x-x_0}{x_1-x_0} \right)^2 \left(3 - 2 \left(\frac{x-x_0}{x_1-x_0} \right) \right) i(x_1, y_0) \\
 & + \left(\frac{y-y_0}{y_1-y_0} \right)^2 \left(3 - 2 \left(\frac{y-y_0}{y_1-y_0} \right) \right) \left(\frac{x-x_0}{x_1-x_0} \right)^2 \left(3 - 2 \left(\frac{x-x_0}{x_1-x_0} \right) \right) i(x_1, y_1) \\
 & + \left(\frac{y-y_0}{y_1-y_0} \right)^2 \left(3 - 2 \left(\frac{y-y_0}{y_1-y_0} \right) \right) \left(1 - \left(\frac{x-x_0}{x_1-x_0} \right)^2 \left(3 - 2 \left(\frac{x-x_0}{x_1-x_0} \right) \right) \right) i(x_0, y_1) \\
 & + \left(1 - \left(\frac{y-y_0}{y_1-y_0} \right)^2 \left(3 - 2 \left(\frac{y-y_0}{y_1-y_0} \right) \right) \right) \left(\frac{x-x_0}{x_1-x_0} \right) \left(\frac{x-x_0}{x_1-x_0} - 1 \right) \frac{\partial i(x_0, y_0)}{\partial x} \\
 & + \left(1 - \left(\frac{y-y_0}{y_1-y_0} \right)^2 \left(3 - 2 \left(\frac{y-y_0}{y_1-y_0} \right) \right) \right) \left(\frac{x-x_0}{x_1-x_0} \right)^2 \left(\left(\frac{x-x_0}{x_1-x_0} \right) - 1 \right) \frac{\partial i(x_1, y_0)}{\partial x} \\
 & + \left(\frac{y-y_0}{y_1-y_0} \right)^2 \left(3 - 2 \left(\frac{y-y_0}{y_1-y_0} \right) \right) \left(\frac{x-x_0}{x_1-x_0} \right)^2 \left(\left(\frac{x-x_0}{x_1-x_0} \right) - 1 \right) \frac{\partial i(x_1, y_1)}{\partial x} \\
 & + \left(\frac{y-y_0}{y_1-y_0} \right)^2 \left(3 - 2 \left(\frac{y-y_0}{y_1-y_0} \right) \right) \left(\frac{x-x_0}{x_1-x_0} \right) \left(\left(\frac{x-x_0}{x_1-x_0} \right) - 1 \right) \frac{\partial i(x_0, y_1)}{\partial x} \\
 & + \left(\frac{y-y_0}{y_1-y_0} \right) \left(\left(\frac{y-y_0}{y_1-y_0} \right) - 1 \right)^2 \left(1 - \left(\frac{x-x_0}{x_1-x_0} \right)^2 \left(3 - 2 \left(\frac{x-x_0}{x_1-x_0} \right) \right) \right) \frac{\partial i(x_0, y_0)}{\partial y} \\
 & + \left(\frac{y-y_0}{y_1-y_0} \right) \left(\left(\frac{y-y_0}{y_1-y_0} \right) - 1 \right)^2 \left(\frac{x-x_0}{x_1-x_0} \right)^2 \left(3 - 2 \left(\frac{x-x_0}{x_1-x_0} \right) \right) \frac{\partial i(x_1, y_0)}{\partial y} \\
 & + \left(\frac{y-y_0}{y_1-y_0} \right)^2 \left(\left(\frac{y-y_0}{y_1-y_0} \right) - 1 \right) \left(\frac{x-x_0}{x_1-x_0} \right)^2 \left(3 - 2 \left(\frac{x-x_0}{x_1-x_0} \right) \right) \frac{\partial i(x_1, y_1)}{\partial y} \\
 & + \left(\frac{y-y_0}{y_1-y_0} \right)^2 \left(\left(\frac{y-y_0}{y_1-y_0} \right) - 1 \right) \left(1 - \left(\frac{x-x_0}{x_1-x_0} \right)^2 \left(3 - 2 \left(\frac{x-x_0}{x_1-x_0} \right) \right) \right) \frac{\partial i(x_0, y_1)}{\partial y} \\
 & + \left(\frac{x-x_0}{x_1-x_0} \right) \left(\left(\frac{x-x_0}{x_1-x_0} \right) - 1 \right)^2 \left(\frac{y-y_0}{y_1-y_0} \right) \left(\left(\frac{y-y_0}{y_1-y_0} \right) - 1 \right) \frac{\partial^2 i(x_0, y_0)}{\partial x \partial y} \\
 & + \left(\frac{x-x_0}{x_1-x_0} \right)^2 \left(\left(\frac{x-x_0}{x_1-x_0} \right) - 1 \right) \left(\frac{y-y_0}{y_1-y_0} \right) \left(\left(\frac{y-y_0}{y_1-y_0} \right) - 1 \right) \frac{\partial^2 i(x_1, y_0)}{\partial x \partial y} \\
 & + \left(\frac{x-x_0}{x_1-x_0} \right)^2 \left(\left(\frac{x-x_0}{x_1-x_0} \right) - 1 \right) \left(\frac{y-y_0}{y_1-y_0} \right)^2 \left(\left(\frac{y-y_0}{y_1-y_0} \right) - 1 \right) \frac{\partial^2 i(x_1, y_1)}{\partial x \partial y} \\
 & + \left(\frac{x-x_0}{x_1-x_0} \right) \left(\left(\frac{x-x_0}{x_1-x_0} \right) - 1 \right)^2 \left(\frac{y-y_0}{y_1-y_0} \right)^2 \left(\left(\frac{y-y_0}{y_1-y_0} \right) - 1 \right) \frac{\partial^2 i(x_0, y_1)}{\partial x \partial y}
 \end{aligned} \tag{2.23}$$

La Figure 2.11 représente une comparaison entre une interpolation bilinéaire et une interpolation bicubique de 16 pixels avec un facteur de 4X4.

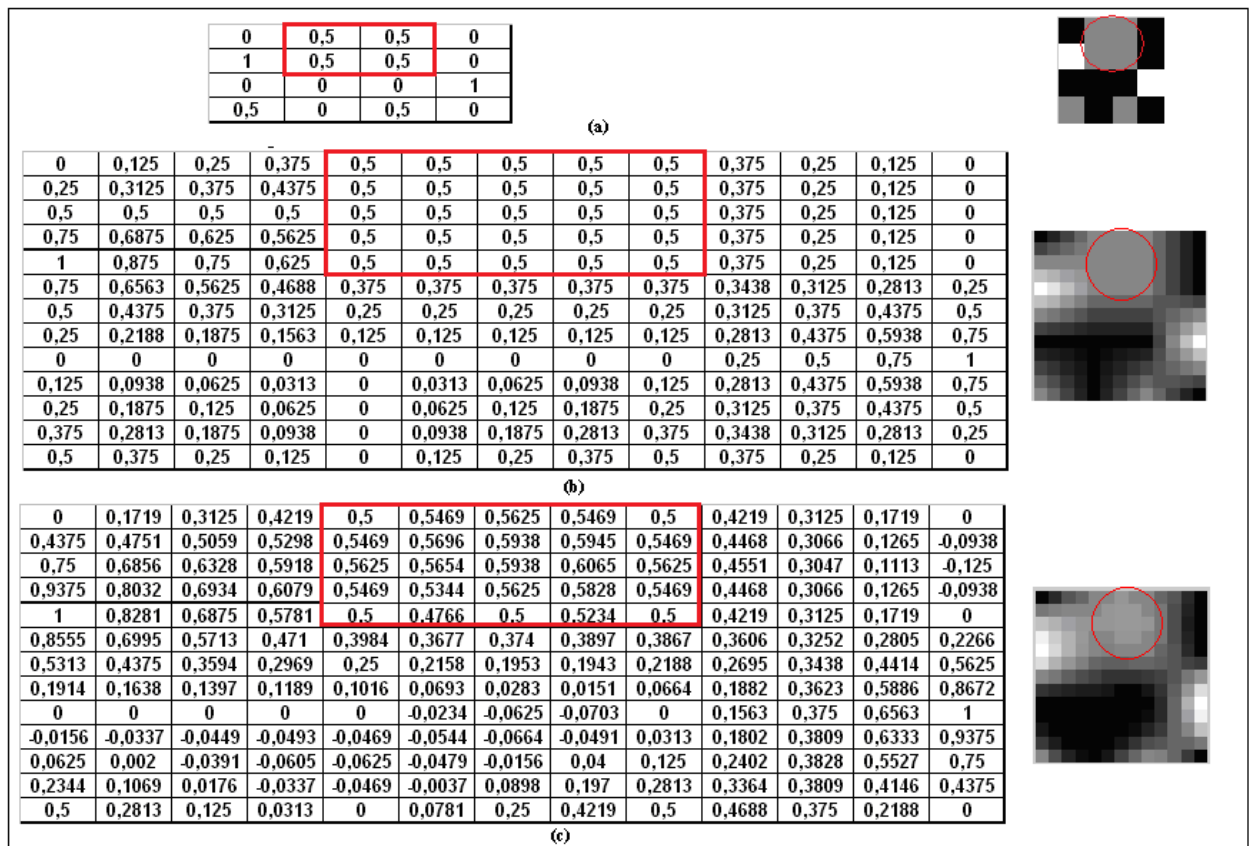


Figure 2. 11 Comparaison entre les valeurs de l'interpolation bilinéaire et de l'interpolation bicubique.

(a) l'image à suréchantillonner

(b) image interpolée bilinéairement

(c) image interpolée bicubiquement

Nous pouvons remarquer, en comparant les zones encadrées des trois images de la Figure 2.11, que la zone encadrée de l'image interpolée bilinéairement reste plate avec des valeurs constantes, tandis que celle de l'image interpolée bicubiquement présente des variations ce qui produit un dynamisme dans l'image.

La Figure 2.12 est une représentation en 3 dimensions des images de la Figure 2.11. Cette représentation nous montre une autre qualité de l'interpolation bicubique : l'interpolation bicubique diminue la discontinuité et les variations rapides de l'image interpolée par ses caractéristiques mathématiques et donne la possibilité d'introduire des points d'inflexions.

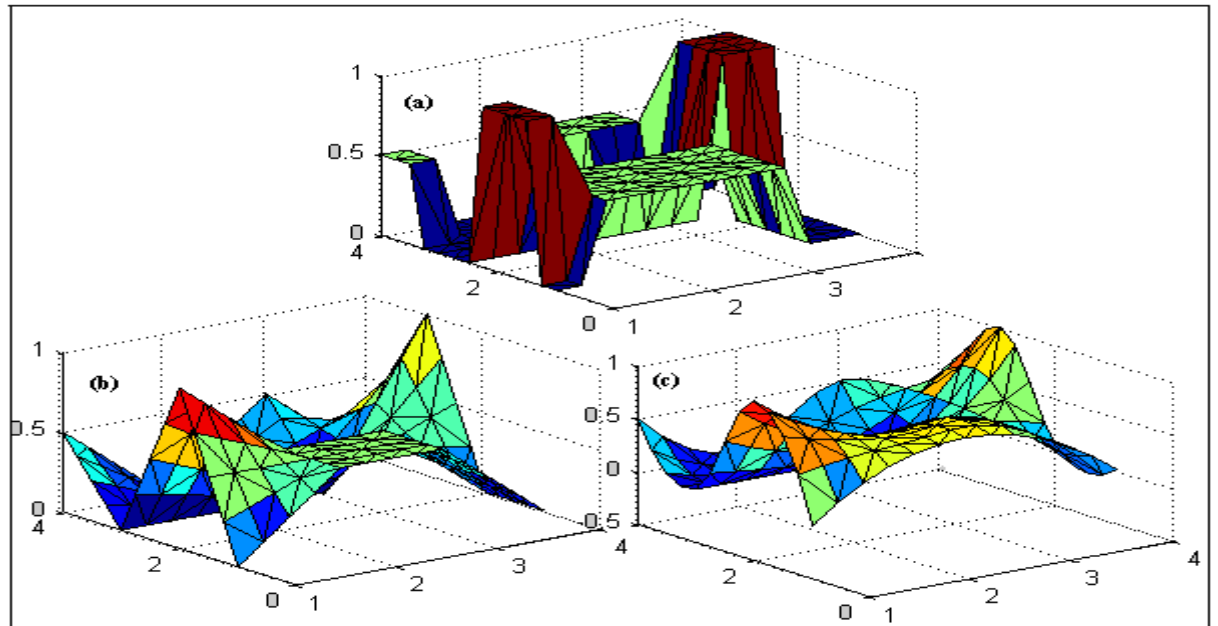


Figure 2. 12 Représentation tridimensionnelle des images suréchantillonnées de la Figure 2.11.

- (a) Image suréchantillonnée par le plus proche voisin
- (b) Image interpolée bilinéairement
- (c) Image interpolée bicubiquement

La Figure 2.13 est une composition de quatre images : l'image originale, son image souséchantillonnée avec un facteur de 4X4, l'image suréchantillonnée par interpolation bilinéaire ainsi que l'image suréchantillonnée par interpolation bicubique.

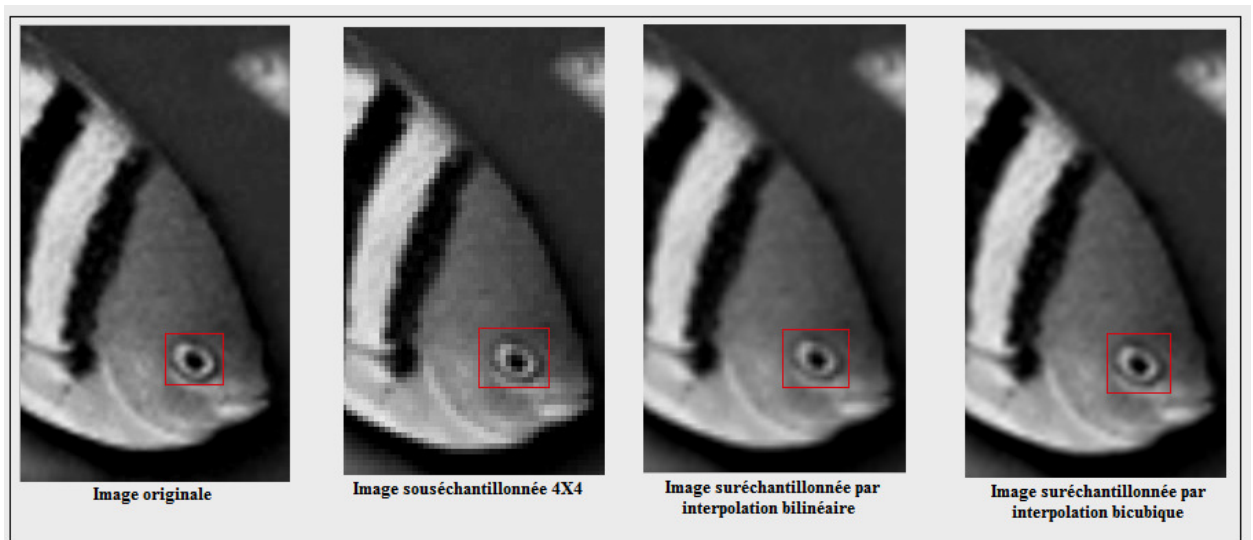


Figure 2. 13 Comparaison entre une interpolation bilinéaire et une interpolation bicubique.

La Figure 2.14 représente des agrandissements des zones encadrées de la Figure 2.13

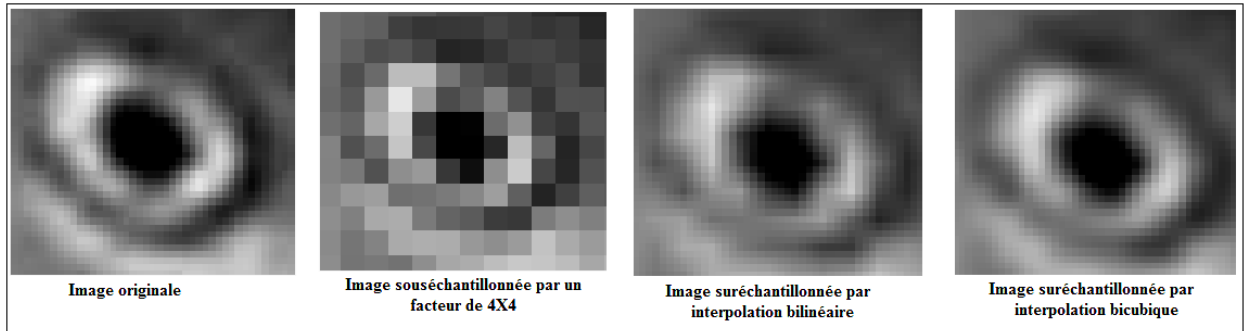


Figure 2. 14 Zoom des zones encadrées de la Figure 2.13.

En observant les résultats obtenus par l'interpolation bicubique et ceux par l'interpolation bilinéaire, nous constatons que la forme de l'œil est mieux retrouvée avec l'interpolation bicubique.

L'interpolation bicubique est une interpolation qui assure la présence du dynamisme et la continuité globale de l'image à cause des caractéristiques de polynôme tels que la possibilité d'introduire des points d'inflexions et l'utilisation des 16 points sources voisins de la zone d'interpolation. Malheureusement, l'utilisation de tous ces points dans l'extrapolation représente un inconvénient dans les calculs.

II.2.5 Interpolation par spline bicubique

La notion d'interpolation par spline signifie mathématiquement l'utilisation d'une fonction continue par morceaux par des polynômes pour l'interpolation. [41]

D'où, l'interpolation par spline bicubique est une méthode d'interpolation locale, qui se calcule en se basant sur les quatre pixels sources entourant la surface d'interpolation. Cette interpolation assure le dynamisme de l'image par l'utilisation d'une fonction d'interpolation semblable à celle de l'interpolation bicubique en utilisant moins de points sources.

Donc la fonction d'interpolation est :

$$i(x, y) = \sum_{n=0}^3 \sum_{m=0}^3 a_{nm} x^n y^m \quad (2.24)$$

Le calcul de cette fonction nécessite la formation et la résolution d'un système de 16 équations linéaires :

Quatre équations de ce système sont déterminées directement des valeurs des quatre pixels sources qui entourent la surface d'interpolation, ou :

$$i(x_i, y_j) = \sum_{n=0}^3 \sum_{m=0}^3 a_{nm} x_i^n y_j^m \quad (2.25)$$

Avec i et j deux entiers qui peuvent être soit 0 ou 1

Les 12 autres équations sont obtenues en exploitant la notion de spline naturel. On dit qu'une fonction d'interpolation est un spline naturel si et seulement si chacun de ses points sources est un point d'inflexion ou, autrement, si ses deuxièmes dérivées au niveau des points sources utilisés sont nulles. [22]

Donc, on peut déterminer quatre équations en annulant les secondes dérivées par rapport à x :

$$\begin{aligned} \frac{\partial^2 i(x_i, y_j)}{\partial x^2} &= 0 \\ \Rightarrow \sum_{m=0}^3 2a_{2m} y_j^m + \sum_{m=0}^3 6a_{3m} x_i y_j^m &= 0 \end{aligned} \quad (2.26)$$

De même pour les dérivées secondes par rapport à y on trouve :

$$\sum_{n=0}^3 2a_{n2} x_i^n + \sum_{n=0}^3 6a_{n3} y_j x_i^n = 0 \quad (2.27)$$

Et pour les quatre dernières équations, on annule la dérivée par rapport à x et y

$$\begin{aligned} \frac{\partial^2 i(x_i, y_j)}{\partial x \partial y} &= 0 \\ \Rightarrow \sum_{n=1}^3 \sum_{m=1}^3 nm.a_{nm} x_i^{n-1} y_j^{m-1} &= 0 \end{aligned} \quad (2.28)$$

En utilisant le changement de variables de la relation 2.5 le système s'écrit :

$$\left\{ \begin{aligned} f(x'_i, y'_j) &= \sum_{n=0}^3 \sum_{m=0}^3 a_{nm} x_i'^n y_j'^m \\ \sum_{m=0}^3 2a_{2m} y_j'^m + \sum_{m=0}^3 6a_{3m} x'_i y_j'^m &= 0 \\ \sum_{n=0}^3 2a_{n2} x_i'^n + \sum_{n=0}^3 6a_{n3} y'_j x_i'^n &= 0 \\ \sum_{n=1}^3 \sum_{m=1}^3 nm.a_{nm} x_i'^{n-1} y_j'^{m-1} &= 0 \end{aligned} \right. \quad (2.29)$$

La forme matricielle de ce système est

$$\begin{bmatrix}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 2 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 2 & 6 & 0 & 0 & 2 & 6 & 0 & 0 & 2 & 6 & 0 & 0 & 2 & 6 \\
 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 6 & 6 & 6 & 6 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 2 & 4 & 6 & 0 & 3 & 6 & 9 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0
 \end{bmatrix}
 \begin{bmatrix}
 a_{00} \\
 a_{10} \\
 a_{20} \\
 a_{30} \\
 a_{01} \\
 a_{11} \\
 a_{21} \\
 a_{31} \\
 a_{02} \\
 a_{12} \\
 a_{22} \\
 a_{32} \\
 a_{03} \\
 a_{13} \\
 a_{23} \\
 a_{33}
 \end{bmatrix}
 =
 \begin{bmatrix}
 f(0,0) \\
 f(1,0) \\
 f(1,1) \\
 f(0,1) \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0
 \end{bmatrix}$$

La résolution de ce système, en calculant la matrice inverse, donne:

$$\begin{bmatrix}
 a_{00} \\
 a_{10} \\
 a_{20} \\
 a_{30} \\
 a_{01} \\
 a_{11} \\
 a_{21} \\
 a_{31} \\
 a_{02} \\
 a_{12} \\
 a_{22} \\
 a_{32} \\
 a_{03} \\
 a_{13} \\
 a_{23} \\
 a_{33}
 \end{bmatrix}
 =
 \begin{bmatrix}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -1 & 1 & 0 & 0 & -1/3 & -1/6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & -1/6 & 1/6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1/3 & 0 & -1/6 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 3 & -3 & -3 & 3 & 0 & 0 & 0 & 0 & 1 & -1 & 0.5 & -0.5 & -2 & -1 & 0 & 0 \\
 -2 & 2 & 2 & -2 & 0 & 0 & 0 & 0 & -2/3 & 2/3 & -1/3 & 1/3 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 3 & -3 & -3 & 3 & 1 & 0.5 & -1 & -0.5 & 0 & 0 & 0 & 0 & -2 & 0 & -1 & 0 \\
 -9 & 9 & 9 & -9 & -1.5 & 0 & 1.5 & 0 & -1.5 & 1.5 & 0 & 0 & 4 & 2 & 2 & 1 \\
 6 & -6 & -6 & 6 & 0.5 & -0.5 & -0.5 & 0.5 & 1 & -1 & 0 & 0 & -2 & -2 & -1 & -1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/6 & 0 & 1/6 & 0 & 0 & 0 & 0 & 0 \\
 -2 & 2 & 2 & -2 & -2/3 & -1/3 & 2/3 & 1/3 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
 6 & -6 & -6 & 6 & 1 & 0 & -1 & 0 & 0.5 & -0.5 & -0.5 & 0.5 & -2 & -1 & -2 & -1 \\
 -4 & 4 & 4 & -4 & -1/3 & 1/3 & 1/3 & -1/3 & -1/3 & 1/3 & 1/3 & -1/3 & 1 & 1 & 1 & 1
 \end{bmatrix}
 \begin{bmatrix}
 f(0,0) \\
 f(1,0) \\
 f(0,1) \\
 f(1,1) \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0
 \end{bmatrix}$$

Ce système peut être simplifié en

$$\begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 3 & -3 & -3 & 3 \\ -2 & 2 & 2 & -2 \\ 0 & 0 & 0 & 0 \\ 3 & -3 & -3 & 3 \\ -9 & 9 & 9 & -9 \\ 6 & -6 & -6 & 6 \\ 0 & 0 & 0 & 0 \\ -2 & 2 & 2 & -2 \\ 6 & -6 & -6 & 6 \\ -4 & 4 & 4 & -4 \end{bmatrix} \begin{bmatrix} f(0,0) \\ f(1,0) \\ f(0,1) \\ f(1,1) \end{bmatrix}$$

D'où, si on pose $\alpha = (f(0,0) - f(1,0) - f(0,1) + f(1,1))$, on aura:

$$a_{00} = f(0,0),$$

$$a_{10} = f(1,0) - f(0,0),$$

$$a_{01} = f(0,1) - f(0,0),$$

$$a_{20} = a_{30} = a_{02} = a_{11} = a_{03} = 0,$$

$$a_{21} = a_{12} = 3\alpha,$$

$$a_{31} = a_{13} = -2\alpha,$$

$$a_{32} = a_{23} = 6\alpha,$$

$$a_{22} = -9\alpha,$$

$$a_{33} = -4\alpha.$$

Nous remarquons que l'implémentation de cet interpolateur est simple puisqu'il est composé de 11 coefficients simples à calculer dont 8 sont des multiples de α .

La Figure 2.15 représente une comparaison numérique entre une interpolation bicubique et une interpolation par spline bicubique.

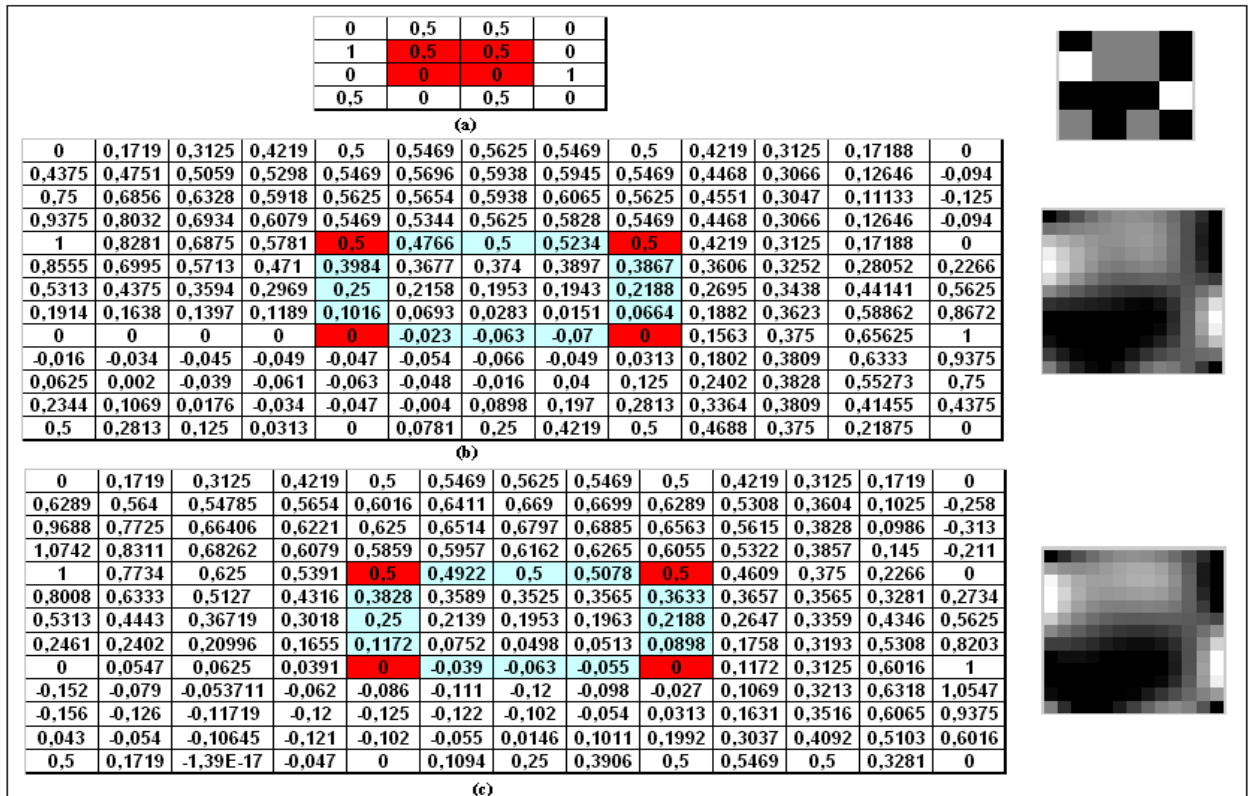


Figure 2. 15 Comparaison numérique entre l'interpolation bicubique et l'interpolation (Facteur 4X4).

(a) l'image initiale (b) image suréchantillonnée par interpolation bicubique

(c) image suréchantillonnée par interpolation par spline bicubique

En analysant cette Figure, on trouve que l'image interpolée par spline bicubique présente légèrement moins de dynamisme par rapport à celle interpolée bicubiquement.

Les Figures 2.16 et 2.17 montrent une comparaison entre les dynamismes des images interpolées bicubiquement et ceux des images interpolées par spline bicubique.

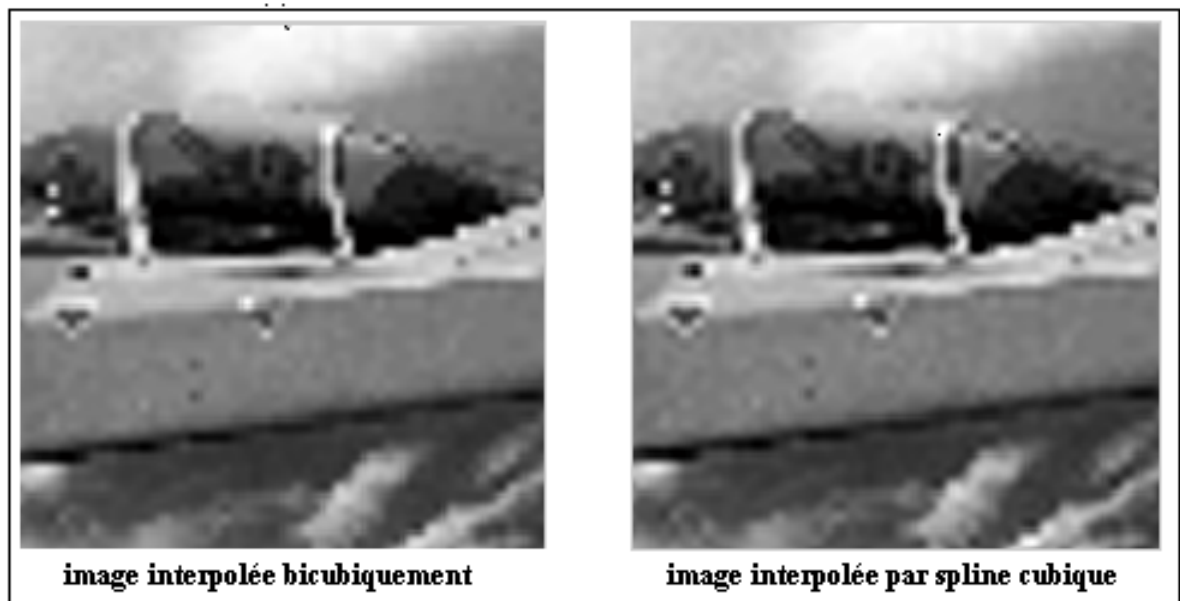


Figure 2. 16 Exemple 1 : Comparaison entre l'interpolation bicubique et par spline bicubique (facteur 4X4).

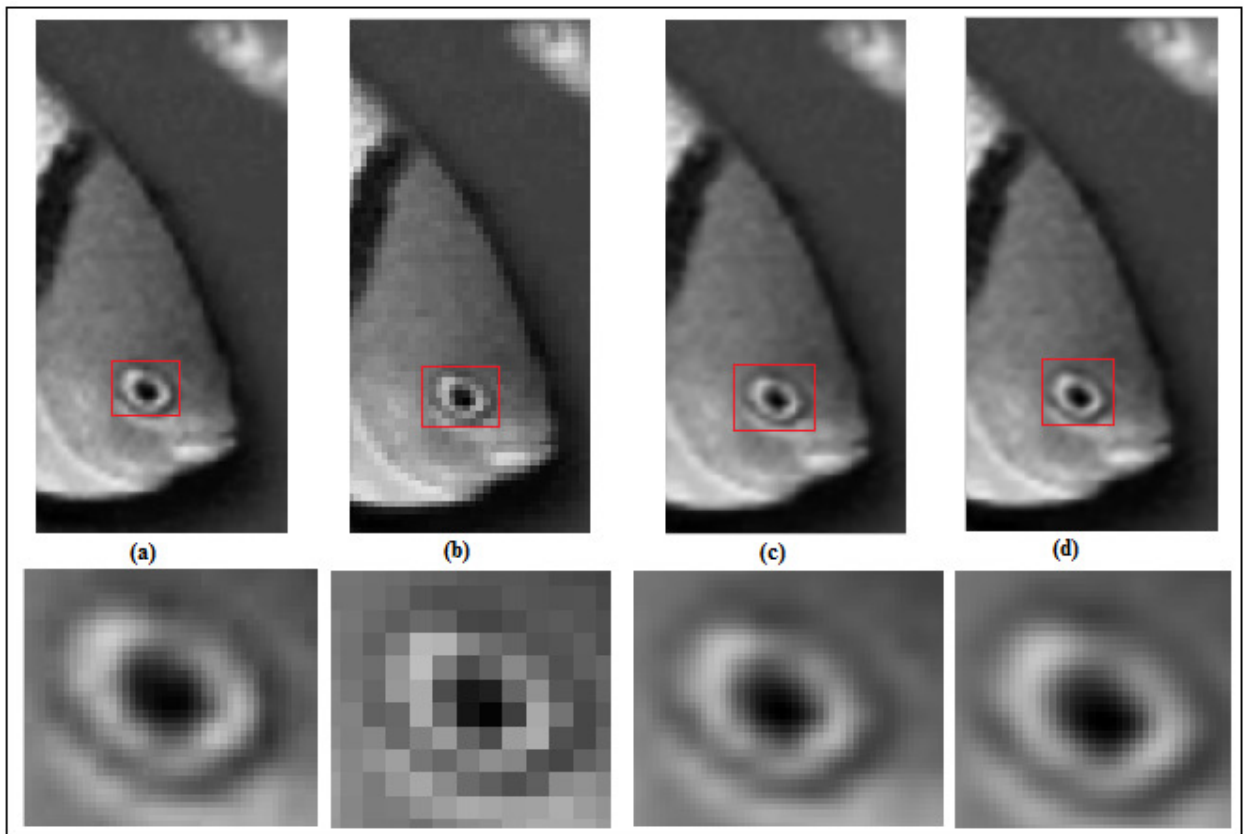


Figure 2. 17 Exemple 2 :Comparaison entre l'interpolation bicubique et par spline bicubique (facteur 4X4).

(a) image initiale (b) image souséchantillonnée avec un facteur 4X4
(c) image interpolée bicubiquement (d) image interpolée par spline bicubique

Nous constatons de l'analyse de la Figure 2.17 que l'avantage major présent dans l'interpolation par spline bicubique est la présence d'un dynamisme important, qui reste légèrement inférieur à celui de l'interpolation bicubique, avec moins de calcul.

II.2.6 Comparaison entre les différentes interpolations polynomiales

En comparant toutes les méthodes d'interpolation polynomiale déjà vues, nous pouvons conclure que:

Nous optons pour l'utilisation de l'interpolation par le plus proche voisin pour un agrandissement d'une image en conservant la même matrice image de faible résolution, sans modification, ce qui ne nécessite pas une grande capacité de calcul. Cette interpolation présente de grands inconvénients représentés dans la discontinuité totale et l'absence de dynamisme dans les images interpolées.

L'interpolation bilinéaire est idéale pour interpoler les images qui représentent un très faible taux de dynamisme. Elle assure la continuité dans toute l'image mais elle est incapable d'introduire le dynamisme dans les images.

Pour assurer la continuité de l'image avec un bon taux de dynamisme, on doit passer à l'interpolation avec des polynômes de degré supérieur, et c'est là que l'interpolation bicubique intervient, elle assure à l'image un dynamisme en plus de la continuité globale. Mais malheureusement, cette méthode pose le problème de grand nombre des ressources utilisées (16 pixels pour chaque surface d'interpolation) ainsi que la complexité des calculs des coefficients.

Pour compenser les points faibles de l'interpolation bicubique, l'interpolation par spline bicubique utilise moins de ressources (quatre pixels) et les coefficients calculés sont très simples à déterminer puisqu'ils sont soit nuls ou des multiples de même facteur. Cette dernière technique est idéale pour les images qui ont un dynamisme élevé malgré qu'elle introduit un dynamisme légèrement inférieur à l'interpolation bicubique [26].

II.2.7 Interpolation bicarrée

Nous avons étudié, jusqu'à présent, les méthodes d'interpolation en utilisant des polynômes de degré inférieur à 3. L'interpolation bicarrée est une autre méthode d'interpolation polynomiale. Elle utilise une double interpolation monodimensionnelle sur les deux axes avec une fonction d'interpolation de la forme :

$$f(x) = ax^2 + bx + c \quad (2.30)$$

avec a , b , c , des constantes réelles.

La forme générale de sa fonction d'interpolation est :

$$i(x, y) = \sum_{n=0}^2 \sum_{m=0}^2 a_{nm} x^n y^m. \quad (2.31)$$

Le calcul des coefficients de la fonction d'interpolation nécessite l'utilisation de 9 points sources, qui ne peuvent pas être choisis symétriquement. Cela produit une grande distorsion et des erreurs importantes dans l'image interpolée.

La discontinuité est un autre inconvénient de cette méthode, en effet l'utilisation des fonctions d'interpolation concave (qui n'acceptent pas des points d'inflexions tels que les polynômes de degré 2) produit une variation rapide du gradient à la limite de chaque surface d'interpolation ce qui produit une très grande discontinuité.

La Figure 2.18 représente une comparaison entre les discontinuités produites suivant une direction choisie. Nous remarquons que la discontinuité est plus importante pour l'interpolation bicarrée.

Tous ces inconvénients ont fait de l'interpolation bicarrée une interpolation déconseillée pour le suréchantillonnage.

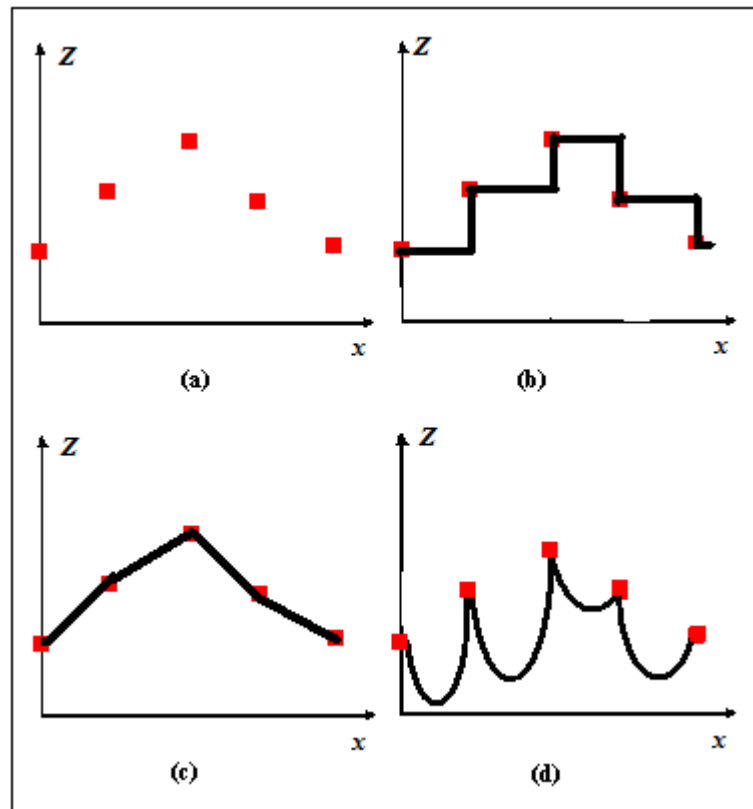


Figure 2. 18 Comparaison entre les discontinuités des surfaces d'interpolation.

(a) projection des pixels (b) projection de la surface d'interpolation par le plus proche voisin
 (c) projection de la surface d'interpolation bilinéaire (d) projection de la surface d'interpolation bicarrée

Après l'étude des algorithmes d'interpolation basés sur les polynômes d'interpolation de degré 3, nous avons constaté qu'avec un polynôme de degré 3 nous pouvons avoir des extremums et des points d'inflexions, ce qui suffit pour introduire un dynamisme et annuler la discontinuité de l'image. Pour cela, nous pouvons arrêter notre étude sur l'interpolation polynomiale à ce niveau sans introduire d'autres algorithmes basés sur des polynômes de degré supérieur.

II.3 Interpolation par filtrage

La théorie d'échantillonnage mentionne qu'il est possible théoriquement de reconstruire un signal analogique à partir de ses échantillons échantillonnés sans repliement, en respectant la *condition de Shannon*, par une interpolation de Whittaker (voir Annexe 1). Cette reconstruction idéale est donc possible en utilisant un filtre de réponse impulsionnelle non causale h_c définie sur le domaine $]-\infty, +\infty[$ tel que :

$$h_c(x, y) = x.e.y.e. \frac{\sin\left(\frac{2\pi f_e x}{2} x\right)}{\pi x} \cdot \frac{\sin\left(\frac{2\pi f_e y}{2} y\right)}{\pi y}$$

Cette théorie est exploitée en superrésolution. En effet, une opération de superrésolution, ou plus simplement le suréchantillonnage, est une opération qui admet l'apparition des erreurs, puisque l'opération de souséchantillonnage est pratiquement irréversible dans la majorité des cas d'où un suréchantillonnage idéal n'est pas toujours possible.

L'opération d'interpolation par filtrage se base essentiellement sur la théorie de *Shannon-Nyquist*, il suffit donc de filtrer l'image initiale par un filtre dérivé du filtre idéal de Whittaker.

Le filtre de Whittaker a une réponse impulsionnelle définie sur l'intervalle $]-\infty, +\infty[$. L'implémentation pratique d'un tel filtre est impossible, pour cela nous procédons à une opération de fenêtrage.

II.3.1 Fenêtrage et Effet de Gibbs

L'opération de fenêtrage est une opération très répandue en traitement du signal, elle consiste à multiplier le signal par une fonction limitée dans l'espace dite fonction fenêtre définie comme suit :

$$f(x, y) = \begin{cases} g(x, y) & x_0 \leq x \leq x_1 \wedge y_0 \leq y \leq y_1 \\ 0 & \text{ailleurs} \end{cases} \quad (2.32)$$

où $f(x, y)$ est la fonction fenêtre,

et $g(x, y)$ est une fonction définie non nulle dans $[x_0, x_1] \times [y_0, y_1]$.

La relation 2.32 peut s'écrire sous la forme :

$$f(x, y) = \begin{cases} g(x, y) & [x_0, x_1] \times [y_0, y_1] \\ 0.g(x, y) & \text{ailleurs} \end{cases} \quad (2.33)$$

$$\Rightarrow f(x, y) = g(x, y).rect_{[x_0, x_1][y_0, y_1]}$$

où la fonction $rect_{[x_0, x_1][y_0, y_1]}$ est une fonction fenêtre dite fenêtre rectangulaire définie pour $g(x, y) = 1, \forall (x, y) \in [x_0, x_1] \times [y_0, y_1]$

Dans le domaine fréquentiel, la relation 2.33 implique

$$F(fx, fy) = G(fx, fy) \otimes TF(rect_{[x_0, x_1][y_0, y_1]}) \quad (2.34)$$

avec $F(fx, fy) = TF(f(x, y))$ et $G(fx, fy) = TF(g(x, y))$

Or on a :

$$TF(rect_{[x_0, x_1][y_0, y_1]}) = e^{-j\pi.f.(x_1-x_0)} e^{-j\pi.f.(y_1-y_0)} \cdot \frac{\sin(\pi.f.(x_1-x_0))}{\pi.f.(x_1-x_0)} \frac{\sin(\pi.f.(y_1-y_0))}{\pi.f.(y_1-y_0)} \quad (2.35)$$

En remplaçant 2.35 dans 2.34 on aura :

$$F(fx, fy) = G(fx, fy) \otimes \left[e^{-j\pi.f.((x_1-x_0)+(y_1-y_0))} \cdot \frac{\sin(\pi.f.(x_1-x_0))}{\pi.f.(x_1-x_0)} \frac{\sin(\pi.f.(y_1-y_0))}{\pi.f.(y_1-y_0)} \right] \quad (2.36)$$

$$F(fx, fy) = G(fx, fy) \otimes \left[e^{-j\pi f \cdot ((x_1-x_0)+(y_1-y_0))} \cdot \text{sinc}(\pi f \cdot (x_1-x_0)) \text{sinc}(\pi f \cdot (y_1-y_0)) \right] \quad (2.37)$$

En analysant la relation 2.37, nous constatons que les fonctions de transfert des fenêtres ont une nature oscillatoire pour tout $(fx, fy) \in]-\infty, +\infty[^2$ à cause de la convolution de $G(fx, fy)$ avec $TF(\text{rect}_{[x_0, x_1]} \times \text{rect}_{[y_0, y_1]})$ qui a une nature oscillatoire. La Figure 2.19 montre la nature oscillatoire du module de la fonction de transfert de la fonction $\text{rect}_{[x_0, x_1]}$

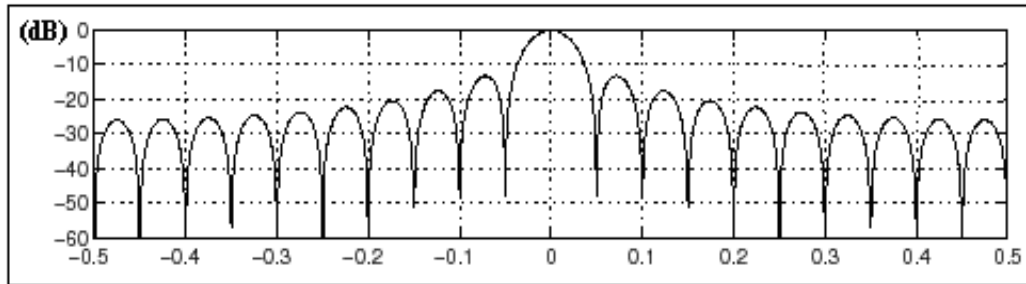


Figure 2. 19 Nature oscillatoire du spectre de la fenêtre rectangulaire.

Dans le cas de la superrésolution par filtrage, la réponse impulsionnelle du filtre de Whittaker fenêtré utilisée pour l'interpolation est de la forme:

$$h(x, y) = h_c(x, y) \cdot f(x, y) \quad (2.38)$$

où h_c est le filtre idéal de Whittaker :

$$h_c(x, y) = x \cdot y \cdot \frac{\sin\left(\frac{2\pi f e_x}{2} x\right)}{\pi x} \cdot \frac{\sin\left(\frac{2\pi f e_y}{2} y\right)}{\pi y} \quad (2.39)$$

Dans le domaine fréquentiel, la relation 2.38 implique :

$$H(fx, fy) = H_c(fx, fy) \otimes F(fx, fy) \quad (2.40)$$

En pratique, un système numérique ne peut pas représenter ou utiliser un signal ayant une fonction de transfert variable pour tout nombre réel. En effet, il est considéré comme un filtre passe bas pour cela il tronque systématiquement les hautes fréquences ce qui produit des parasites et des ondulations au niveau des zones de discontinuité « les contours » à cause de la troncature des hautes fréquences. Ce phénomène est appelé, phénomène de *Gibbs*.

La Figure 2.20 montre l'effet de Gibbs sur une image reconstruite avec la troncature des hautes fréquences, des oscillations et du flou présents au niveau des zones de discontinuité dues au phénomène de Gibbs :



Figure 2. 20 L'effet Gibbs sur les images.

La relation 2.37 montre que le choix de la fonction $g(x,y)$ a une influence directe sur l'effet Gibbs, pour cela nous allons choisir des fenêtres pour les utiliser en fenêtrage du filtre du Whittaker en étudiant leur effet sur la superrésolution des images.

II.3.2 Fenêtrage du filtre de Whittaker

L'effet oscillatoire d'une fenêtre sur un signal a une relation directe avec la largeur de la bande passante du lobe principal à -3dB de sa fonction de transfert ainsi que du niveau de son lobe secondaire. La Figure 2.21 présente la bande passante et le niveau de second lobe.

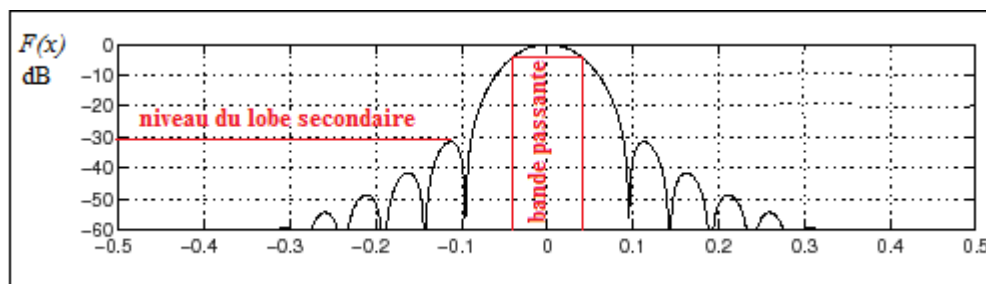


Figure 2. 21 La bande passante et le lobe secondaire d'une fenêtre de Hamming.

La fenêtre la plus connue et la plus simple à implémenter est la fenêtre rectangulaire définie pour $g(x,y)=1$, ou plus simplement :

$$f(x,y) = \begin{cases} 1 & [x_0, x_1] \times [y_0, y_1] \\ 0 & \text{ailleurs} \end{cases}$$

Cette fenêtre présente une largeur de bande passante du lobe principal à -3dB très faible tel que $\Omega = \frac{0.89}{N}$, N représente le nombre de points utilisés, et le niveau du second lobe très élevé (-13 dB) ce qui présente de grands perturbations dans le domaine fréquentiel.[46]

La fenêtre de Hamming est un autre type de fenêtre de troncature définie par la relation

$$2.32 \text{ avec : } g(x, y) = \left(0.54 - 0.46 \cos\left(\frac{2\pi x}{N_x}\right) \right) \left(0.54 - 0.46 \cos\left(\frac{2\pi y}{N_y}\right) \right),$$

sa fonction de transfert présente de bonnes caractéristiques : une largeur de la bande passante du lobe principal à -3dB est de $\frac{1.3}{N}$, et le niveau du second lobe est de -43 dB.[47]

Une autre fenêtre qui est de même forme que la forme de Hamming, la fenêtre de Hann, appelée aussi fenêtre de Hanning, définie par la relation 2.32 avec

$$g(x, y) = \left(0.5 - 0.5 \cos\left(\frac{2\pi x}{N_x}\right) \right) \left(0.5 - 0.5 \cos\left(\frac{2\pi y}{N_y}\right) \right)$$

présente une largeur de la bande passante du lobe principal à -3dB égale à $\frac{1.44}{N}$ et un niveau de second lobe de (-32). [47]

La Figure 2.22 représente une comparaison entre les spectres de la fenêtre rectangulaire, celle de Hamming et la fenêtre de Hanning.

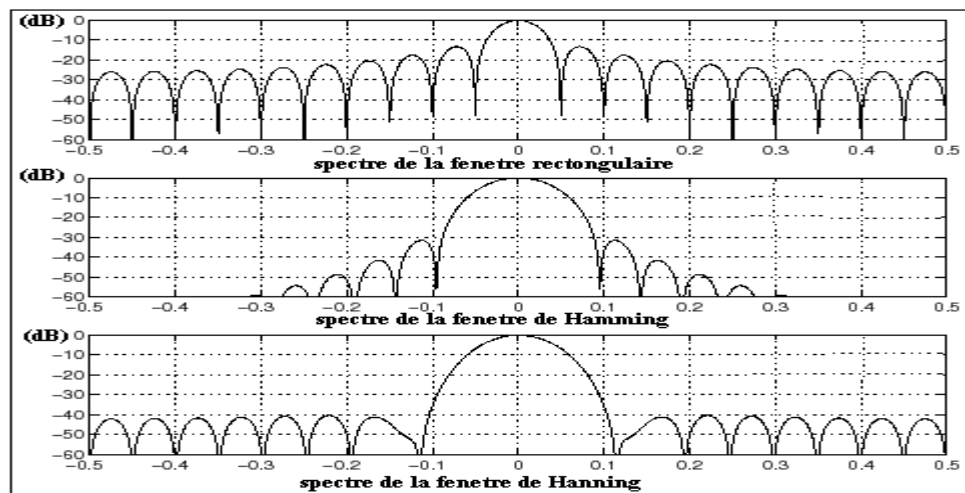


Figure 2. 22 Comparaison entre des spectres normalisés des fenêtres de 20 points.

La comparaison entre ces trois fenêtres montre que chacune des fenêtres de Hamming et de Hann améliorent la fenêtre rectangulaire et on remarque que la fenêtre de Hamming se caractérise par le plus faible niveau du lobe secondaire tandis que la fenêtre de Hann a la plus large bande passante.

La fenêtre de Blackman est une autre fenêtre définie par la relation 2.32 avec $g(x, y) = 0.42 - 0.5 \cos\left(2\pi \frac{x}{N_x}\right) + 0.08 \cos\left(4\pi \frac{x}{N_x}\right)$, cette fenêtre est une fenêtre optimisée ; en effet, elle présente une bande passante de $\frac{1.44}{N}$ avec un niveau du second lobe de -67dB. [47]

La Figure 2.23 représente les réponses impulsionnelles de ces fenêtres étudiées :

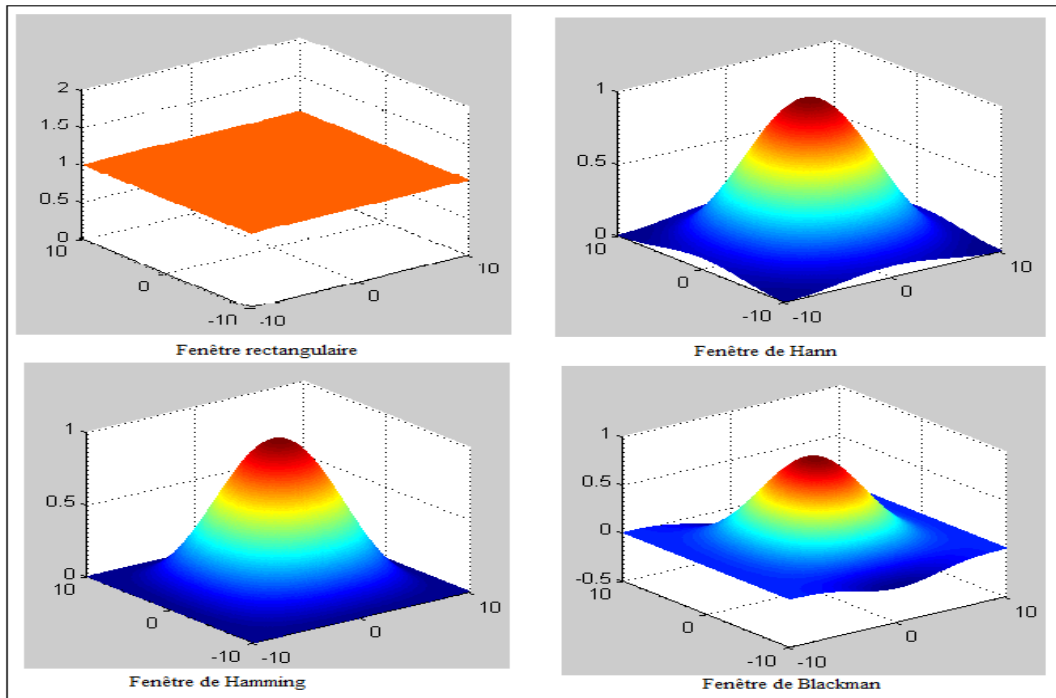


Figure 2. 23 Réponses impulsionnelles des fenêtres rectangulaire, de Hann, de Hamming et de Blackman.

La Figure 2.24 représente le fenêtrage du filtre de Whittaker avec la fenêtre rectangulaire, les fenêtres de Hann, de Hamming et de Blackman.

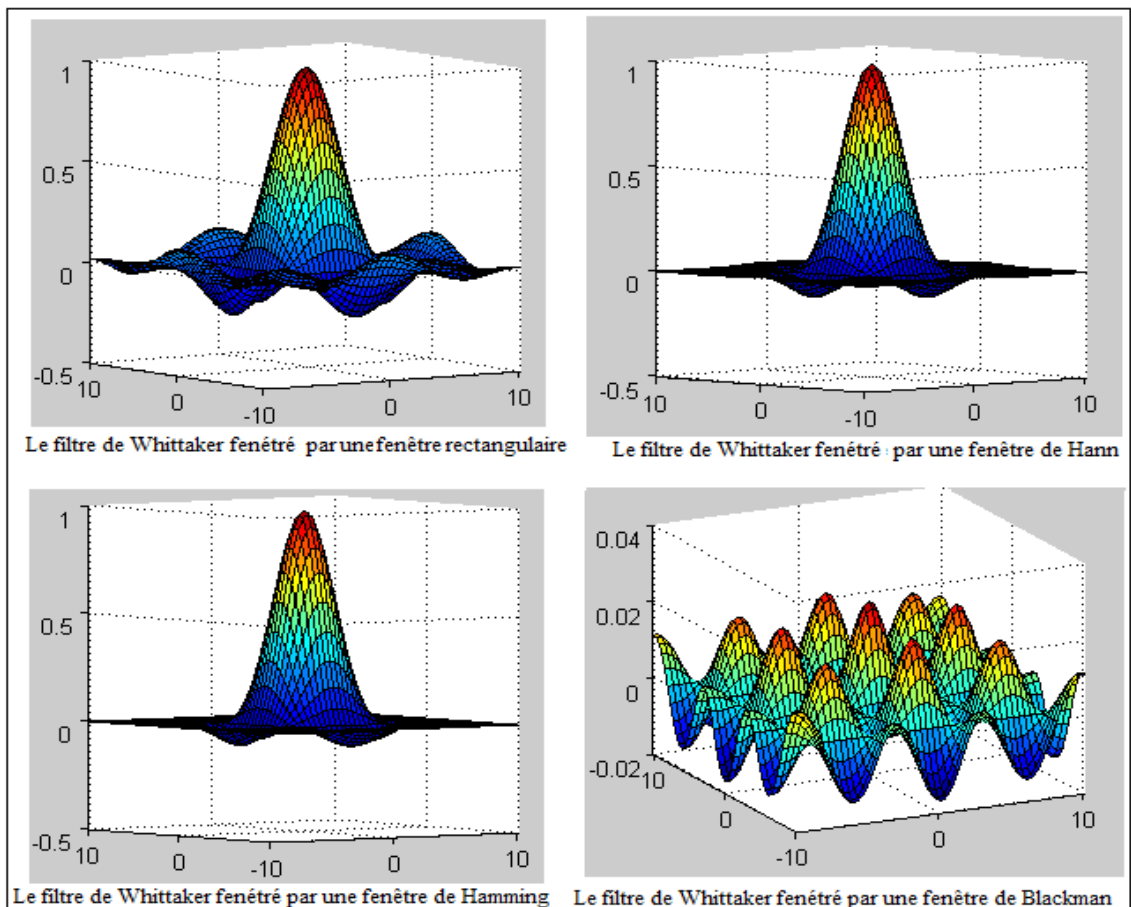


Figure 2. 24 Réponses impulsionnelles des filtres construits par le fenêtrage de Whittaker.

L'effet de ces différents filtres sur des images de faible résolution est illustré par la Figure 2.24 : cette dernière expose un exemple où on a suréchantillonné une image par le filtrage avec les filtres de la Figure 2.24 en améliorant sa résolution par un facteur de 4X4.

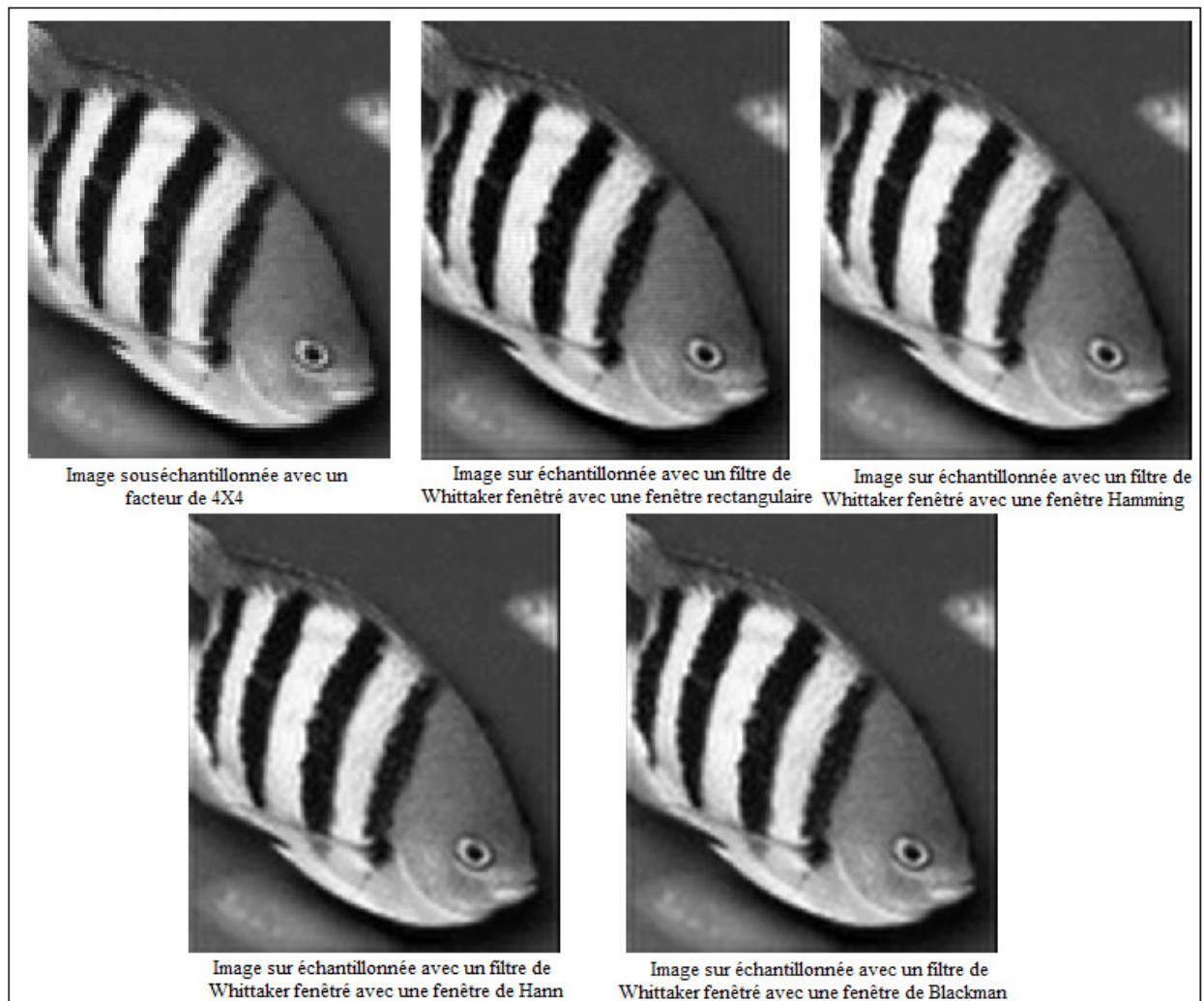


Figure 2. 25 Résultats du suréchantillonnage par filtrage.

En comparant l'effet de Gibbs produit par chacun de ces filtres, nous remarquons que les ondulations sont apparues au niveau des discontinuités surtout à la bordure des images et le taux de ces ondulations est maximum pour le filtre fenêtré par la fenêtre rectangulaire.

II.3.3 Mise en œuvre de l'interpolation par filtrage

L'interpolation par filtrage doit passer par deux étapes :

- **Construction de la matrice générale de l'image :** Dans cette étape, nous allons construire la matrice générale de l'image introduisant $(k_x - 1) \times (k_y - 1)$ zéros entre chaque 4 pixels $p(x,y)$, $p(x,y+1)$, $p(x+1,y+1)$ et $p(x+1,y)$.

- **Construction du filtre d'interpolation :** Le filtre interpolateur est construit à partir du fenêtrage du filtre de Whittaker, le filtre interpolateur est caractérisé par sa fenêtre et le nombre

de points utilisés N qui doit répondre au facteur de suréchantillonnage K , au temps et capacité de calcul disponible. En effet pour un nombre N faible, certains points extrapolés ne tiennent pas en compte leurs points sources voisins lors de la convolution d'où l'apparition de lignes noires sur l'image suréchantillonnée. De même pour N important, la largeur de la bande passante du filtre sera faible d'où la naissance d'ondulations au niveau des discontinuités.

La Figure 2.26 montre l'influence du choix de N sur l'image suréchantillonnée par une interpolation par filtrage en utilisant un filtre de Whittaker fenêtré par une fenêtre de Hamming.

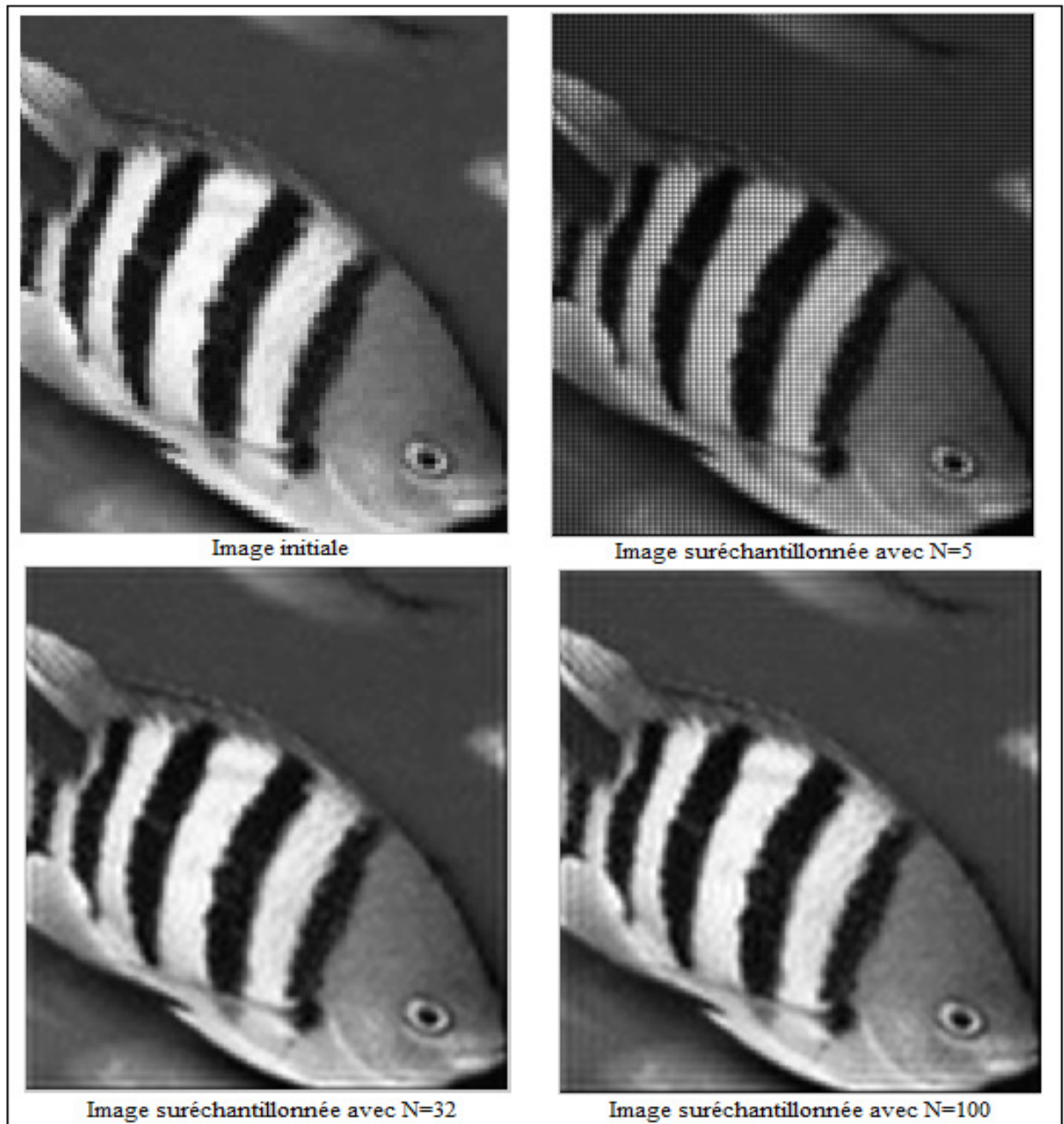


Figure 2. 26 Influence de la taille de filtre sur l'image suréchantillonnée ($K=4X4$).

Une fois la matrice générale et le filtre construits, le suréchantillonnage sera réalisé par la convolution de l'image avec la réponse impulsionnelle du filtre.

II.4 Conclusion

Ce chapitre a présenté plusieurs techniques de superrésolution monoframe (qui utilisent une seule image faible résolution dans la reconstruction de l'image haute résolution) basées sur des méthodes d'interpolation par filtrage et d'interpolation polynomiale. La comparaison entre ces différentes techniques, nous a montré que le choix entre eux doit se baser sur plusieurs facteurs : les ressources et la capacité de calculs disponibles, la continuité et le dynamisme de l'image ainsi que de taux d'erreurs et de distorsion tolérables.

Dans le chapitre suivant, nous allons proposer des algorithmes dans le but d'adapter et d'exploiter ces techniques de superrésolution dans la superrésolution multiframe en utilisant plusieurs images de faible résolution issues de la même scène dans la reconstruction d'une image haute résolution afin d'optimiser la distorsion produite.

Chapitre III

**SUPERRESOLUTION MULTIFRAME PAR
TECHNIQUES D'INTERPOLATION**

Chapitre III

SUPERRESOLUTION MULTIFRAME PAR TECHNIQUES D'INTERPOLATION

III.1 Introduction

La précision est devenue l'un des besoins les plus importants notamment pour les applications basées sur le traitement des images. Pour des applications de haute précision, telles que certaines applications militaires et applications médicales, la résolution des images utilisées joue un grand rôle sur la valeur des résultats obtenus. Malheureusement, la technologie actuelle reste toujours limitée et ne peut pas satisfaire ces besoins qui ne cessent d'augmenter.

III.2 Superrésolution multiframe idéale

Nous avons vu dans les chapitres précédents que le suréchantillonnage idéal d'une image est purement théorique et que chacune des opérations d'interpolation déjà étudiées peut produire des erreurs. Le taux de ces erreurs peut être inacceptable pour des applications de haute précision.

Le taux de ces erreurs peut s'annuler en utilisant plusieurs images de faible résolution issues de la même scène dans la construction d'une image de résolution supérieure. En effet, chaque image HR numérique peut être décomposée en n images de résolution inférieure LR_i dites imagettes, la réunion de toutes ces imagettes LR_i compose l'image initiale HR : Donc pour construire une image de résolution $N_x \times N_y$ avec des imagettes de résolution $\frac{N_x}{k_x} \times \frac{N_y}{k_y}$, nous devons avoir les $k_x \times k_y$ imagettes qui la composent.

La Figure 3.1 représente la construction d'une image avec 4 imagettes issues de la même scène, l'image résultante à une résolution 2×2 la résolution des imagettes.

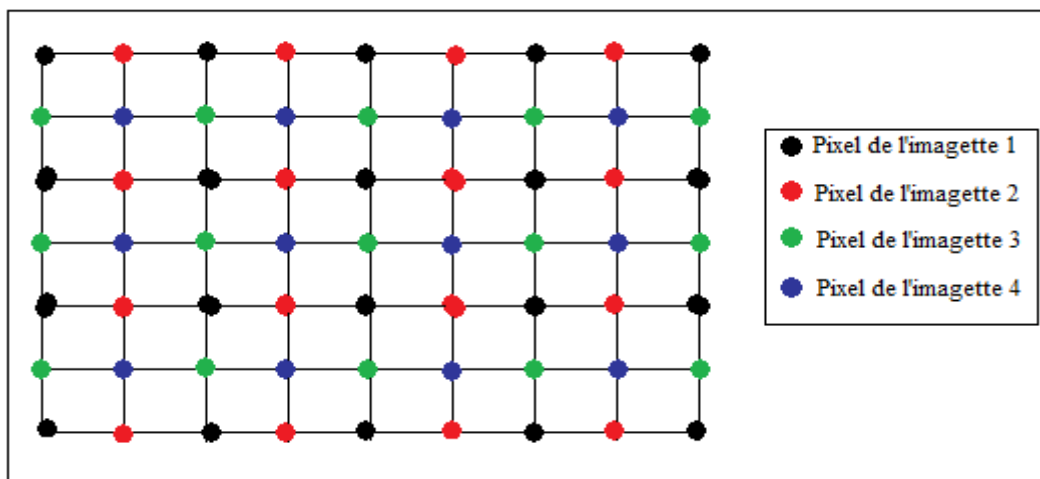


Figure 3.1 Construction d'une image à partir de 4 imassettes

Donc il est possible de construire sans distorsions et sans erreurs une image HR de résolution multiple de la résolution de l'appareil d'acquisition utilisé en exploitant des images prises par ce dernier : pour avoir une image haute résolution d'une résolution $k_x \times k_y$, plus grande que la résolution de l'appareil d'acquisition, on doit utiliser les $k_x k_y$ imassettes prises en déplaçant, sur le même plan et pour chaque imassette, l'appareil d'un pas égal à $1/k_x$ la distance entre 2 pixels de l'axe X et celles prises en la déplaçant d'un pas de $1/k_y$ sur l'axe Y comme montré sur la Figure 3.1. Cette opération de construction est appelée « Superrésolution Multiframe Idéale » ou « Suréchantillonnage Multiframe Idéal ».

III.3 Superrésolution multiframe par interpolation

Dans certains cas, la superrésolution idéale n'est pas possible, cela peut être dû à l'indisponibilité de la totalité des imassettes ou à un manque d'information sur l'emplacement de ces imassettes. Dans ce cas, nous serons obligé d'exploiter les imassettes disponibles pour extrapoler les valeurs des pixels manquants.

III.3.1 Intérêt apporté par l'utilisation de plusieurs imassettes

Nous avons vu précédemment que l'utilisation de $k_x k_y$ images de résolution $n_x \times n_y$ acquises de la manière présentée dans III.2 produit une image suréchantillonnée avec un facteur de $k_x \times k_y$ sans erreurs. Par ailleurs, nous pouvons démontrer que le taux des erreurs produites lors de l'extrapolation diminue en fonction des imassettes utilisées.

Soit HR l'image suréchantillonnée de résolution $N_x \times N_y$ avec $N = N_x \cdot N_y$ pixels construite à partir de m imassettes différentes LR_i de résolution $n_x \times n_y = (N_x/k_x) \times (N_y/k_y)$ avec $n = n_x \cdot n_y$ pixels pour chaque imassette. Par conséquent, l'image HR contient $m \cdot n_x \cdot n_y$ pixels

sources d'une probabilité d'erreur nulle issus directement de l'ensemble des imgettes LR_i et sert à extrapoler les autres pixels de l'image. D'où, le nombre N_{Err} des pixels présentant une probabilité d'erreur est égal à :

$$\begin{aligned}
 N_{Err} &= N - m.n \\
 \Rightarrow N_{Err} &= N_x N_y - m.n_x.n_y \\
 \Rightarrow N_{Err} &= N_x.N_y - m.\frac{N_x N_y}{k_x k_y} \\
 \Rightarrow N_{Err} &= N_x.N_y \left(1 - \frac{m}{k_x k_y}\right) \tag{3.1}
 \end{aligned}$$

Définissons R_{Err} la probabilité d'avoir des pixels erronés dans une image comme le rapport entre le nombre de pixels qui présentent une probabilité d'erreur et le nombre total des pixels de cette image, d'où :

$$\begin{aligned}
 R_{Err} &= \frac{N_{Err}}{N} \\
 \Rightarrow R_{Err} &= \frac{N_x.N_y \left(1 - \frac{m}{k_x k_y}\right)}{N_x.N_y} \\
 \Rightarrow R_{Err} &= 1 - \frac{m}{k_x k_y} \tag{3.2}
 \end{aligned}$$

La relation 3.2 montre que R_{Err} diminue en augmentant le nombre des imgettes utilisées et il s'annule pour le cas de la superrésolution multiframe idéale, c à d, le cas où le nombre d'imgette est égal à $k_x.k_y$.

En plus, la probabilité d'erreur d'un pixel extrapolé est une fonction de la distance entre ce pixel et les pixels sources utilisés pour son extrapolation : plus cette distance est grande, plus la probabilité augmente et vice versa. Or l'augmentation du nombre de pixels sources par l'augmentation du nombre d'imgettes utilisées diminue automatiquement les distances entre les pixels sources et les pixels extrapolés, par conséquent les probabilités d'erreurs des pixels diminuent.

D'une autre part, la superrésolution multiframe possède le pouvoir de diminuer le phénomène de recouvrement spectral produit lors des opérations d'échantillonnage et de souséchantillonnage (voire de l'annuler) ce qui présente un grand avantage. En effet, la diminution de la distance entre les pixels sources, en introduisant des pixels issus d'autres imgettes, est équivalent à une opération de souséchantillonnage avec un pas plus petit ou avec

une fréquence d'échantillonnage f_e plus grande, cette augmentation de f_e se traduit par la diminution de la zone de recouvrement qui sera nulle pour f_e supérieure ou égale au double de la fréquence fondamentale f_0 de l'image.

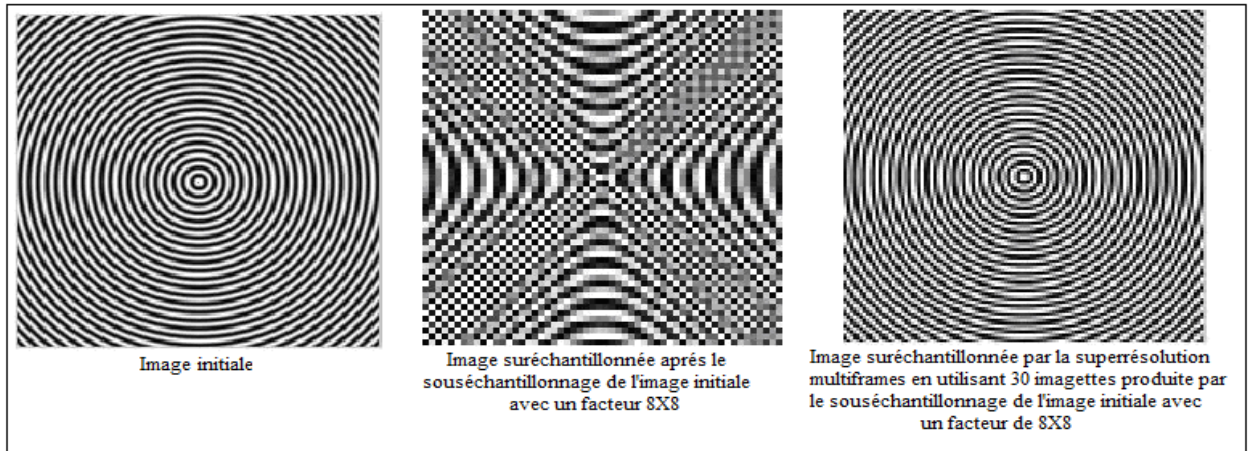


Figure 3. 2 Rôle de la superrésolution multiframe dans la diminution du recouvrement spectral.

La Figure 3.2 représente une comparaison entre une image suréchantillonnée avec un algorithme de superrésolution par interpolation par filtrage en utilisant une seule imagerie souséchantillonnée avec un facteur de 8X8, et une autre image construite par une opération de suréchantillonnage multiframe qui se base sur l'interpolation par filtrage en utilisant 30 imagerie de même résolution que la première imagerie utilisée. Cette comparaison montre comment la superrésolution multiframe a pu diminuer les taux d'erreurs et de distorsion causés par le recouvrement.

Nous pouvons conclure, en se basant sur l'analyse précédente, que plus le nombre d'images utilisées dans la construction de l'image suréchantillonnée est important, plus le taux d'erreur et de distorsion de l'image résultante sont faibles.

III.3.2 Algorithme général de la superrésolution multiframe par interpolation

Afin de pouvoir réaliser la superrésolution multiframe, nous proposons cet algorithme général en 3 étapes de reconstruction :

- **Génération de la matrice zéros HRz :**

Dans cette première étape, nous devons générer une matrice nulle HRz qui sert comme structure initiale pour l'image haute résolution HR . La taille de HRz doit être donc égale à celle de la taille de la matrice HR .

- **Placement des images $\{LR_i\}$ sur la matrice HRz :**

Une fois la matrice HRz générée, nous procédons au placement des pixels des images $\{LR\}$ sur la matrice HRz en affectant la valeur de chaque pixel issu d'une image au zéro

correspondant à son emplacement sur l'image HRz . Ces pixels vont jouer le rôle des pixels source de la matrice HR , notons la matrice résultante par HRp .

Afin de pouvoir exploiter ces différentes imagettes mises à notre disposition pour la reconstruction d'une image de résolution supérieure, nous devons, impérativement, connaître les coordonnées exactes de l'appareil lors de l'acquisition de chaque imagette, autrement dit, connaître la relation qui existe entre chacun des pixels des imagettes et les pixels de la matrice HRz pour déterminer les pixels sources et les pixels à extrapoler.

Il pourrait être intéressant, pour l'implémentation de certains algorithmes d'interpolation, de construire une matrice binaire $HRmap$ de la même dimension que HRz tel que :

$$\begin{cases} HRmap(i, j) = 1 & \text{si } HR(i, j) \text{ est un point source} \\ HRmap(i, j) = 0 & \text{si } HR(i, j) \text{ est un point extrapolé} \end{cases}$$

- Interpolation:

Dans cette étape, nous nous intéresserons à l'extrapolation des valeurs des pixels non déterminés lors de l'étape précédente après une opération d'interpolation basée sur la matrice HRp .

La Figure 3.3 représente un résumé des principales étapes de la superrésolution multiframe :

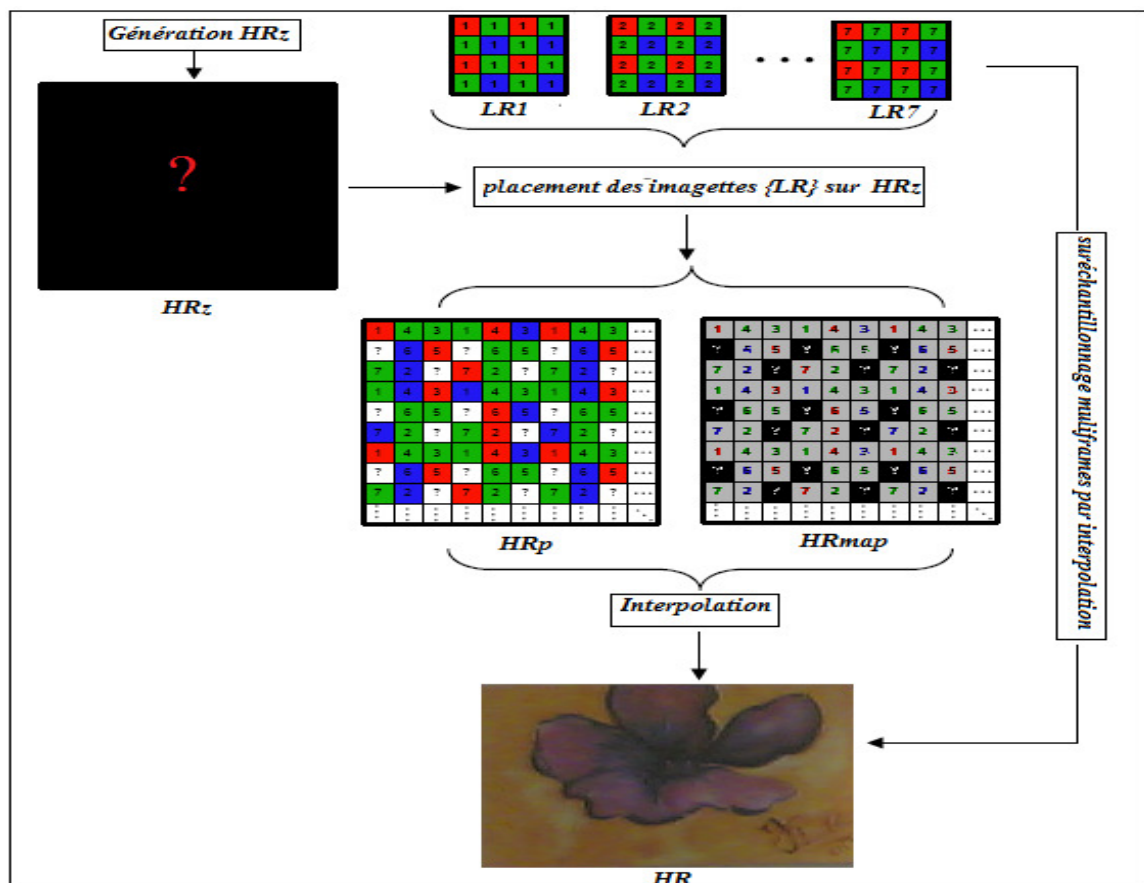


Figure 3. 3 Principales étapes de la superrésolution multiframe

Nous allons essayer de montrer comment nous pouvons exploiter les techniques de superrésolution monoframe par interpolations, vus dans le chapitre précédent, dans la superrésolution multiframe (SRM).

III.4 SRM par interpolation par filtrage

L'interpolation par filtrage de la superrésolution multiframe se base toujours sur le principe de l'utilisation des filtres de Whittaker fenêtrés. L'implémentation de cette interpolation peut être réalisée par l'un des deux algorithmes suivants :

- **L'interpolation par filtrage de la matrice HRp :**

Dans cet algorithme, nous allons filtrer, directement, l'image HRp , construite par le placement de chacune des imagerie LR_i sur la matrice zéros HRz , par un filtre de Whittaker fenêtré. Cet algorithme représente une application directe de l'algorithme développé pour l'interpolation par filtrage d'une seule imagerie. [18]

- **L'interpolation par filtres indépendants des imagerie :**

Cette interpolation consiste à superposer les images HR_i interpolée par filtrage à partir des imagerie LR_i pour construire l'image HR . Donc l'avantage rapporté par cette méthode est représenté par le pouvoir d'utiliser des filtres différents compatibles avec chaque imagerie afin de minimiser le taux d'erreur. [19]

La Figure 3.4 représente l'organigramme général des algorithmes utilisés pour la superrésolution multiframe en utilisant l'interpolation par filtrage.

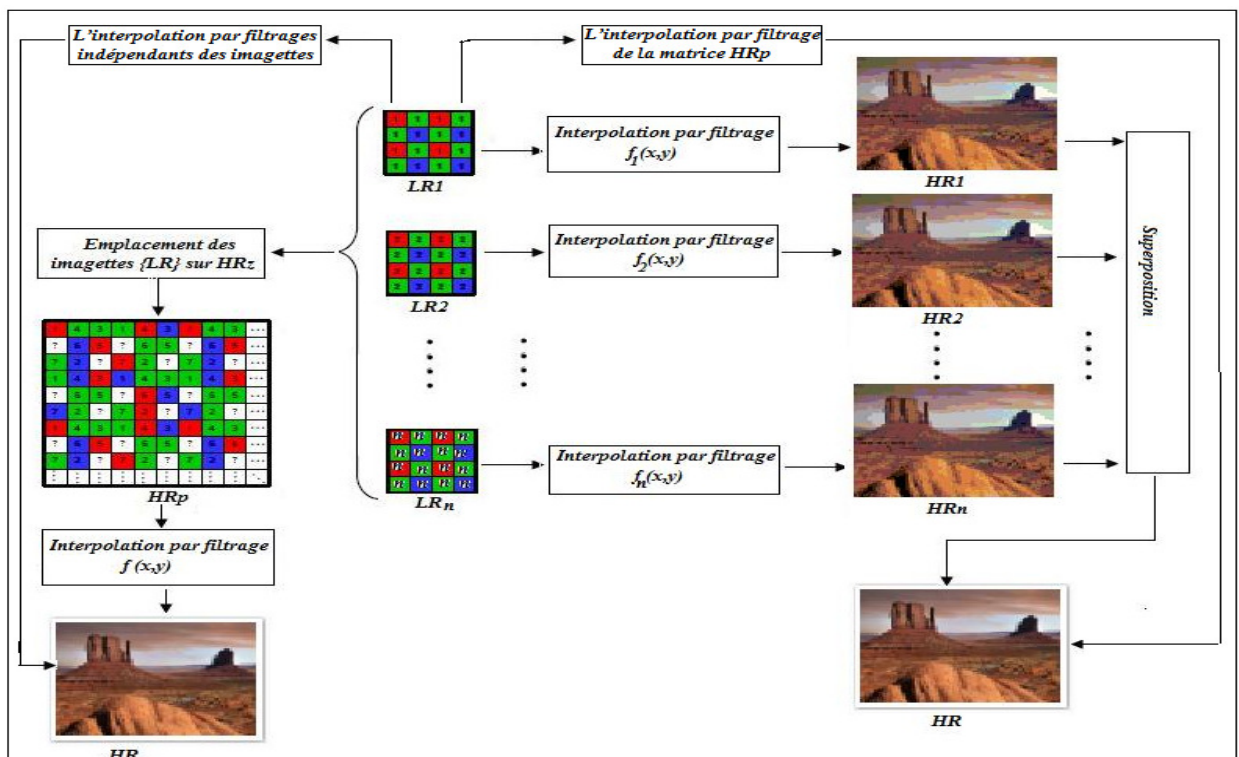


Figure 3. 4 Organigramme des algorithmes de l'application de l'interpolation par filtrage dans la SRM

Nous pouvons considérer L'interpolation par filtrage de la matrice HRp comme un cas particulier de l'interpolation par filtrages indépendants des imagerie. En effet, si on filtre toutes les imagerie avec le même filtre puis on superpose les images résultantes, ceci équivaut exactement au filtrage de la superposition de toutes les imagerie avec le même filtre choisi précédemment.

L'expérience a montré que le taux d'erreur peut être optimisé en utilisant une combinaison de filtres différents pour chaque imagerie ce qui favorise l'interpolation par filtrage indépendant.

III.5 SRM par interpolation polynomiale

Nous avons étudié dans le chapitre précédent différents algorithmes et techniques d'interpolation polynomiale appliquée à la superrésolution monoframe qui exploite les informations disponibles dans une seule imagerie pour construire une image de résolution plus importante. Dans un tel cas, les positions des pixels sources sont uniformément distribuées sur les deux axes X et Y ; pour cela, l'identification des pixels sources intervenant dans l'extrapolation de chaque pixel de l'image suréchantillonnée ne pose pas un grand problème.

Contrairement à la superrésolution monoframe, les pixels sources utilisés dans la superrésolution multiframe ne sont pas toujours uniformément positionnés. Pour cela, la détermination des pixels sources utilisés pour l'extrapolation de chaque pixel de l'image suréchantillonnée n'est pas évidente.

L'algorithme général de La superrésolution par interpolation polynomiale est composé des deux étapes suivantes

- **La détermination des pixels sources exploités pour l'extrapolation de chaque pixel:**

Dans cette étape, nous allons décomposer l'image suréchantillonnée en plusieurs champs, en utilisant la matrice $HRmap$ pour que les pixels de chaque champ utilisent les mêmes pixels sources pour leurs extrapolations.

Dans ce travail, nous allons nous intéresser aux deux méthodes de décomposition les plus utilisées dans ce genre de problèmes : le diagramme de Voronoï et la triangulation de Delaunay.

- **L'extrapolation des pixels de l'image suréchantillonnée par interpolation polynomiale:**

Pour cela, nous allons calculer les pixels de valeurs inconnues à partir de la matrice HRp en adaptant les algorithmes de l'interpolation polynômiale appliqués pour une seule image « superrésolution monoframe » à la superrésolution multiframe.

Nous définirons, dans ce qui suit, les deux structures de la théorie des graphes utilisées dans la première étape : le diagramme de Voronoï et la triangulation de Delaunay.

III.5.1 Diagramme de Voronoï

Le diagramme de Voronoï est l'une des plus importantes structures utilisées dans la théorie des graphes ; introduit dans les espaces vectoriels \mathfrak{R}^2 et \mathfrak{R}^3 par le mathématicien allemand Johann Peter Gustav Lejeune **Dirichlet** dans l'étude des formes quadratique en 1850, puis utilisé par le médecin britannique John **Snow** dans sa recherche sur la propagation du choléra en 1854, pour être formalisé et généralisé par le mathématicien russe Georgy Fedoseevich **Voronoï** en 1908.

Soit (E) un espace vectoriel et (P) un ensemble fini de n points appartenant à cet espace tel que : $(P) = \{p_i, 1 \leq i \leq n\}$. On appelle une cellule C_i , cellule de Voronoï de centroïde p_i l'ensemble des points de (E) les plus proches de p_i que de tout autre point de (P) :

$$C_i = \{q \in (E), \forall j \neq i, \|qp_i\| \leq \|qp_j\|\} \quad (3.3)$$

L'union de toutes les cellules de Voronoï de (P) forme le diagramme de Voronoï de (P) et on note :

$$Vor(P) = \bigcup_{i=1}^n C_i \quad (3.4)$$

Soit d la dimension de (E) . L'intersection de $d+1$ cellules de Voronoï, si elle existe, forme un sommet de Voronoï. De même, on appelle l'intersection de d cellules de Voronoï, si elle n'est pas nulle, une arête de Voronoï. Enfin, la face de Voronoï est l'intersection de i cellules de Voronoï avec $2 \leq i \leq d-1$.

La Figure 3.5 représente le résultat de la division de l'espace \mathfrak{R}^2 en cellules de Voronoï relativement à une constellation de points (P) et la formation du diagramme de Voronoï.

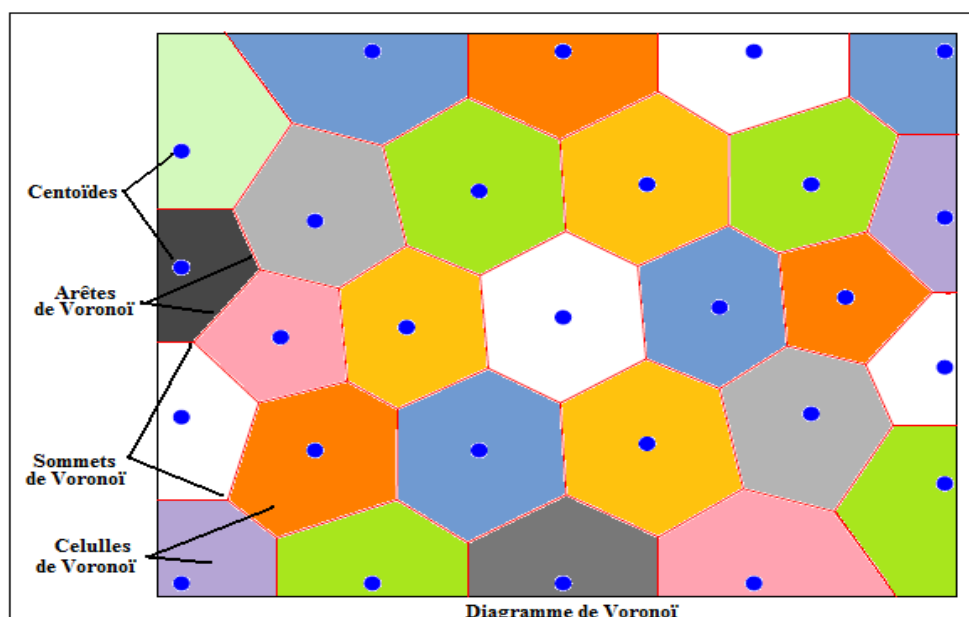


Figure 3. 5 Diagramme de Voronoï d'une constellation de points de \mathfrak{R}^2

Nous pouvons remarquer que chaque cellule de Voronoï est un polygone convexe tel que toute arête de Voronoï séparant deux cellules C_i et C_j est portée par la médiatrice du segment $[p_i, p_j]$, et tout sommet s séparant trois cellules C_i , C_j et C_k représente le centre de gravité du triangle $p_i p_j p_k$.

En se basant sur ces caractéristiques et vue l'importance de ce diagramme, plusieurs algorithmes ont été mis en œuvre afin de minimiser le temps de calcul, citons à titre d'exemple ces deux algorithmes basiques :

- **Construction par insertion des demi-plans :**

Dans ce cas, chaque cellule de Voronoï C_i est construite de manière indépendante par l'intersection des $n-1$ demi-plans limités par les médiatrices des segments $[p_i, p_j]$.

La Figure 3.6 est un exemple qui représente l'évolution de l'algorithme de construction du diagramme de Voronoï par insertion de demi-plans : L'image (1) représente la constellation (P) définissant le diagramme de Voronoï. Afin de pouvoir réaliser la construction du diagramme, nous devons calculer chaque cellule de Voronoï séparément.

On veut délimiter la cellule C_i correspondant au point p_i , nous devons donc calculer, de manière récurrente et pour chaque $j \neq i$, la médiatrice du segment $[p_i, p_j]$. L'intersection de tous les plans contenant p_i et délimitée par ces médiatrices construit la cellule C_i comme le montre les images : (2), (3), (4), (5), (6), (7) et (8) de la Figure, cette opération doit être répétée pour chaque point de la constellation.

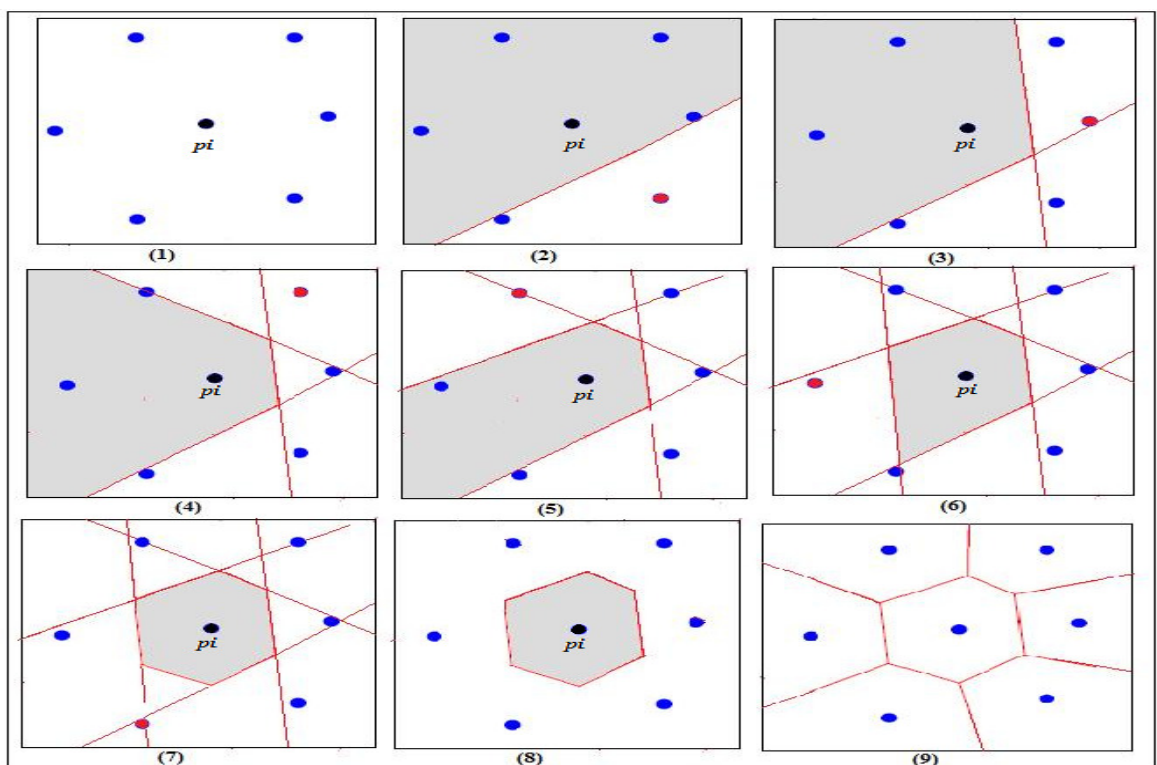


Figure 3. 6 Construction de diagramme de Voronoï par insertion de demi-plans

- **Construction incrémentale :**

Dans cet algorithme proposé en 1978 par Green et Sibson [42], la construction du diagramme de Voronoï se calcule d'une manière récurrente. En effet, en supposant connaître le diagramme de Voronoï des $n-1$ centroïdes $(P') = \{p_1, p_2 \dots p_{n-1}\}$, nous calculons le diagramme de Voronoï de la constellation (P) construite par l'ajout d'un nouveau centroïde p_n à (P') :

$$(P) = (P') \cup \{p_n\}.$$

Nous pouvons remarquer que l'insertion d'un nouveau centroïde p_n ne modifie pas la totalité du diagramme de Voronoï, mais son effet est localisé au niveau de la cellule de Voronoï C_i à qui il appartient : $p_n \in C_i$ et les cellules qui l'entourent.

L'algorithme général de cette construction est basé sur les points suivants :

1. Identifier la cellule C_i , de centroïde p_i , tel que $p_n \in C_i$;
2. Déterminer les sommets qui vont disparaître, ces sommets sont les sommets s_{ik} de C_i tel que p_n appartient au cercle de centre s_{ik} et de rayon $\|p_i s_{ik}\|$, autrement dit : $\|p_n s_{ik}\| \leq \|p_i s_{ik}\|$.
3. Tracer la médiatrice entre $[p_i, p_n]$ en calculant les points a et b de son intersection avec les frontières de la cellule C_i , $[p_i, p_n]$ représente une arête de la cellule C_n ;
4. Pour chaque sommet s_{ik} , nous devons calculer les médiatrices entre p_n et les centroïdes p_j ($j \neq i$) auxquelles s_{ik} appartient,
5. Ainsi, nous construisons les frontières de C_n par l'intersection de ces médiatrices ;
6. Enfin, nous devons mettre à jour les arêtes de Voronoï de la cellule C_n en supprimant les morceaux à l'intérieur de cette dernière.

La Figure 3.7 représente un exemple de la construction incrémentale du diagramme de Voronoï en exprimant les différentes étapes de l'algorithme.

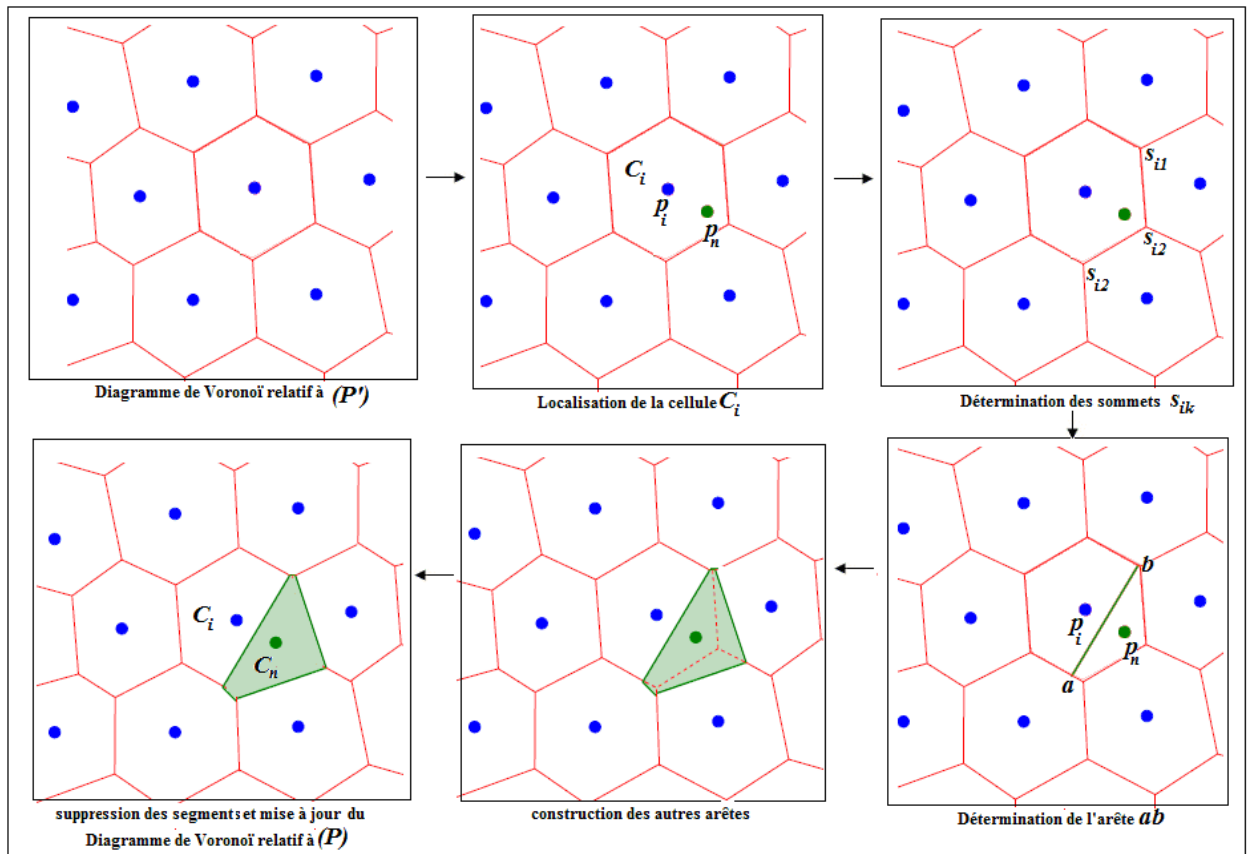


Figure 3. 7 Construction incrémentale du diagramme de Voronoï

III.5.2 Triangulation de Delaunay

On appelle triangulation, l'opération de la division d'un espace vectoriel en un ensemble de triangles $(T) = \{t_i, i \in \mathbb{N}^*\}$.

En 1934, Boris Delaunay, le mathématicien russe et l'élève de Voronoï, a proposé sa triangulation comme une triangulation duale au diagramme de Voronoï d'un ensemble de n points $(P) = \{p_i, i \in \mathbb{N}\}$ tel que les sommets de ses triangles sont les points $p_i \in (P)$ et ses arêtes sont déterminées par la liaison des centroïdes des cellules de Voronoï adjacentes deux à deux.

Cette triangulation divise l'espace en triangles de sorte que le disque circonscrit à chaque triangle ne peut contenir un centroïde différent des sommets de ce triangle. Elle se caractérise d'être la triangulation qui maximise les angles des triangles, autrement dit : La triangulation de Delaunay d'un ensemble de points est celle qui évite le mieux la création de triangles aplatis et qui tendent, autant que possible, à une forme proche d'un triangle équilatéral.

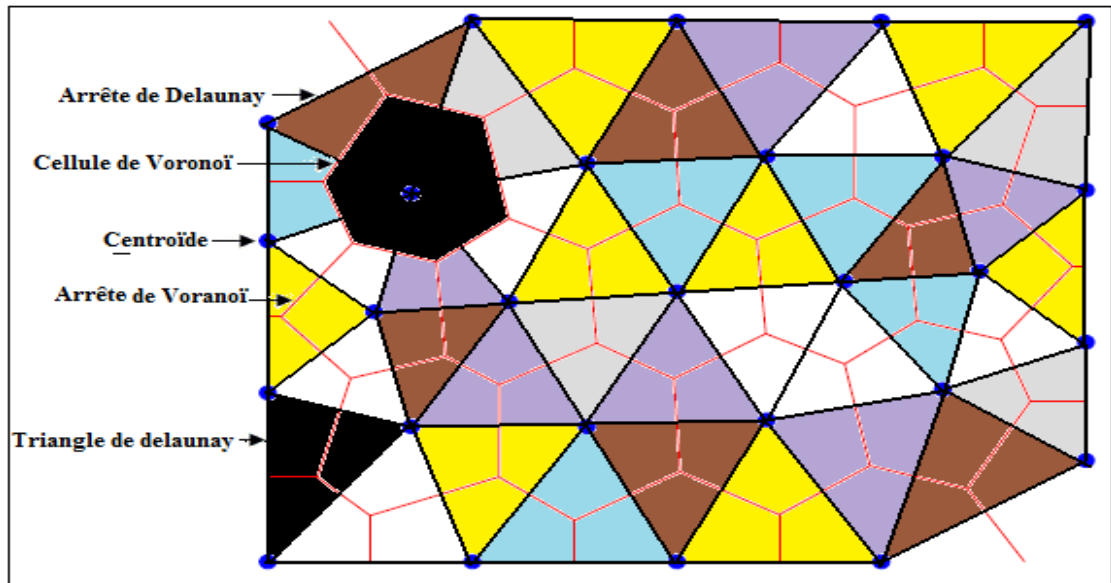


Figure 3. 8 Dualité entre le diagramme de Voronoï et sa triangulation de Delaunay

La Figure 3.9 représente une comparaison entre deux triangulations de la même constellation composée de quatre points A, B, C et D : L'image (2) représente une triangulation différente de la triangulation de Delaunay puisqu'il existe un point de la constellation, B par exemple, différent du triangle de Delaunay, ACD , et qui appartient au disque limité par le cercle circonscrit de ce triangle. Les images (2) et (3) représentent deux triangulations de Delaunay où nous pouvons remarquer que chaque disque limité par le cercle circonscrit ne contient aucun point de la constellation différent des sommets de ce triangle.

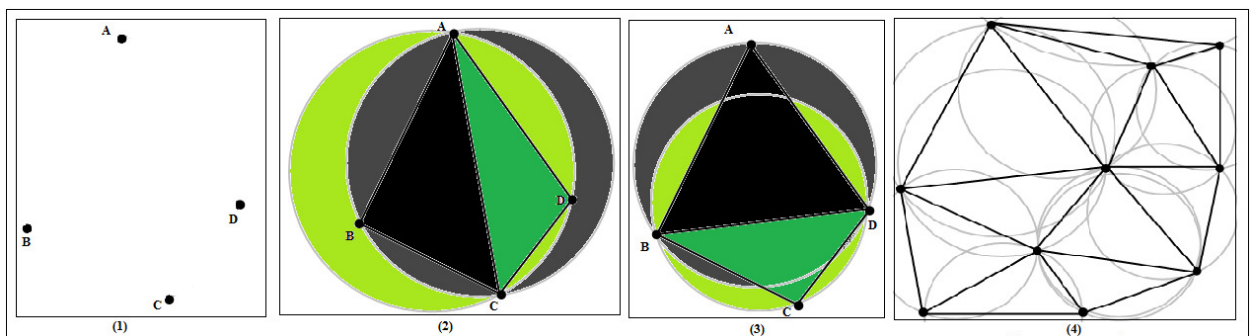


Figure 3. 9 Comparaison entre les circonscrits de la triangulation de Delaunay et des autres triangulations.

- (1) points de la constellation
- (2) triangulation différente de la triangulation de Delaunay
- (3) Triangulation de Delaunay
- (4) La triangulation de Delaunay sur une autre constellation

En comparant les deux triangulations présentées par la Figure 3.10, nous remarquons que la triangulation de Delaunay, contrairement aux autres triangulations, tend vers la maximisation de l'angle le plus faible de chaque triangle d'où une meilleure répartition de la surface sur les triangles de Delaunay.

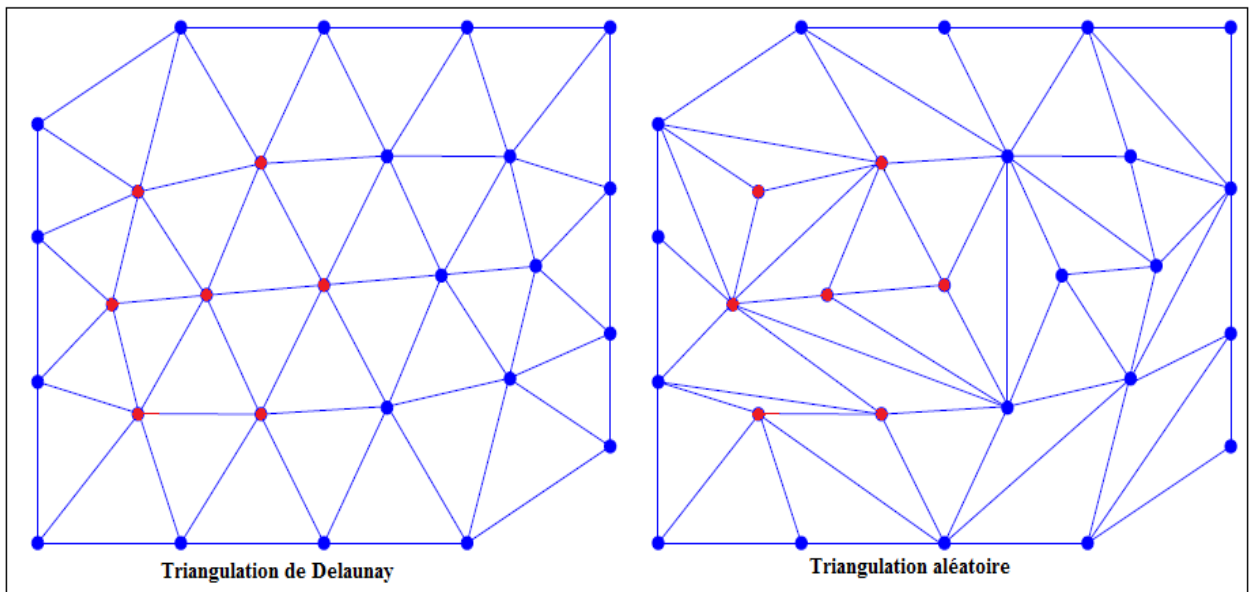


Figure 3. 10 Maximisation des angles les plus faibles de la triangulation par la triangulation de Delaunay

La triangulation de Delaunay est considérée comme l'une des plus importantes structures de la théorie des graphes : Elle est très ré pondante pour les problèmes d'optimisations des trajectoires, de décompositions des plans et espaces, de la reconstruction cristallographique et des surfaces, ainsi que pour le codage et le décodage.

L'importance et les caractéristiques de la triangulation de Delaunay ont motivé les chercheurs pour développer et optimiser plusieurs algorithmes ; citons à titre d'exemple ces deux algorithmes :

- **Triangulation par l'exploitation du diagramme de Voronoï :**

Cet algorithme de triangulation exploite la dualité existante entre le diagramme de Voronoï et la triangulation de Delaunay. Supposant la connaissance du diagramme de Voronoï d'une constellation donnée, la triangulation de Delaunay de cette constellation revient à déterminer les arêtes de Delaunay en reliant les différents centroïdes des cellules de Voronoï adjacentes comme le montre la Figure 3.8.

- **Triangulation par incrémentation :**

Proposé par Guibas, Knuth et Sharir, cet algorithme consiste à englober l'espace de travail par un triangle $p_1 p_2 p_3$, calculer la triangulation de l'espace en introduisant les points de la constellation un par un, en terminant par la suppression des points $p_1 p_2 p_3$ et les arêtes liées à eux. [43]

L'approche incrémentale est une approche récurrente : supposant connaître la triangulation de Delaunay T' de $n-1$ points $(P') = \{p_1, p_2, \dots, p_{n-1}\}$ et nous voulons déterminer la triangulation de

Delaunay T de la constellation $(P) = (P') \cup \{p_n\}$. Pour cela, on doit identifier le triangle de Delaunay $p_i p_j p_k$ de la triangulation T' à laquelle le point p_n appartient ainsi que le point p_l le point le plus proche du point introduit, avec $l \notin \{i, j, k\}$, deux cas sont possibles : Soit p_n appartient à l'arête $[p_i, p_j]$, dans ce cas on aura la construction des quatre triangles de Delaunay $p_i p_k p_n$, $p_j p_k p_n$, $p_j p_l p_n$ et $p_i p_n p_l$. Soit il appartient à l'intérieur du triangle $p_i p_j p_k$ tel que les distances entre $(p_n$ et $p_i)$ et entre $(p_j$ et $p_n)$ sont inférieures à la distance entre $(p_k$ et $p_n)$, dans ce cas, les deux triangulations possibles sont $p_k p_i p_n$, $p_k p_j p_n$, $p_n p_i p_j$, $p_j p_l p_i$ et $p_k p_i p_n$, $p_k p_j p_n$, $p_n p_l p_j$, $p_n p_l p_i$ et l'une de ces triangulations est une triangulation de Delaunay. En effet, l'une des propriétés les plus importantes de la triangulation de Delaunay mentionne que parmi les deux triangulations possibles d'un quadrilatère quelconque, une est de Delaunay. Ainsi dans ce cas, si un ou des triangles ne vérifient pas la condition de Delaunay, il suffit de les supprimer en échangeant une de leurs arêtes dans le quadrilatère $p_k p_i p_j p_l$.

La Figure 3.11 représente un exemple de triangulation de Delaunay par incrémentation exprimant les deux cas possibles, où $p_i p_j p_k$ représente le triangle de Delaunay qui va contenir le point incrémenté p_n et $p_l, l \notin \{i, j, k\}$, est le point le plus proche de p_n : Si le nouveau point p_n appartient à l'arête $p_i p_j$, la nouvelle triangulation de Delaunay est réalisée par la création des arêtes de Delaunay $p_n p_k$ et $p_l p_n$. Et si p_n est à l'intérieur du triangle de Delaunay $p_i p_j p_k$, nous vérifions son appartenance au triangle de Delaunay $p_i p_j p_l$ par le calcul du déterminant Δ tel que :

$$\Delta = \begin{vmatrix} x_i & y_i & x_i^2 + y_i^2 & 1 \\ x_l & y_l & x_l^2 + y_l^2 & 1 \\ x_j & y_j & x_j^2 + y_j^2 & 1 \\ x_n & y_n & x_n^2 + y_n^2 & 1 \end{vmatrix}$$

Si $\Delta < 0 \Rightarrow$ le point p_n n'appartient pas à l'intérieur du cercle circonscrit du triangle $p_i p_j p_l$, et la nouvelle triangulation de Delaunay sera construite à partir des triangles de Delaunay $p_k p_i p_n$, $p_k p_j p_n$, $p_n p_i p_j$, $p_j p_l p_i$, sinon elle sera construite à partir des triangles suivant $p_k p_i p_n$, $p_k p_j p_n$, $p_n p_l p_j$, $p_n p_l p_i$.

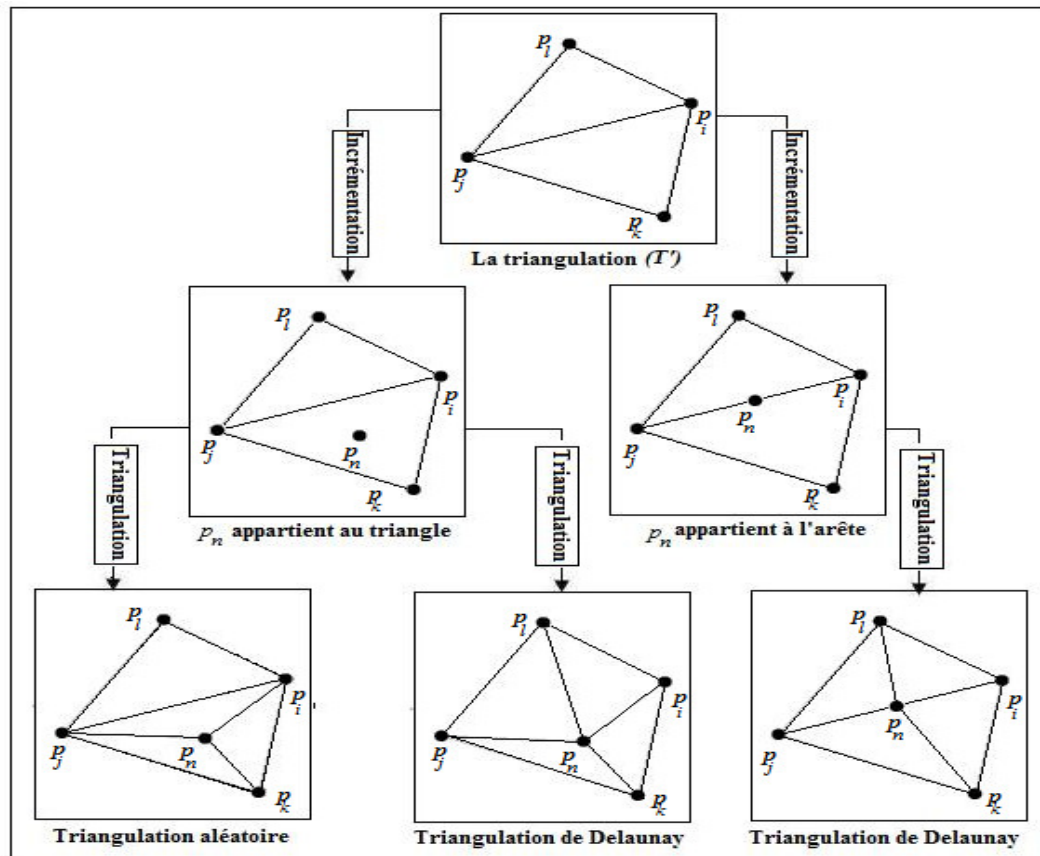


Figure 3. 11 Triangulation par incrémentation

Nous allons voir dans ce qui suit comment utiliser le diagramme de Voronoï et la triangulation de Delaunay dans l'adaptation des algorithmes de superrésolution monoframe par les interpolations polynomiales à la superrésolution multiframe.

III.5.3 L'interpolation polynomiale dans la superrésolution multiframe

Nous avons présenté, dans le chapitre précédent, des algorithmes de superrésolution monoframe basés sur des polynômes interpolateurs de degré zéro « interpolation par le plus proche voisin », de premier degré « interpolation linéaire et interpolation bilinéaire » ainsi que des polynômes de degré trois « interpolation bicubique et interpolation par spline bicubique ». Ces algorithmes peuvent être généralisés et utilisés dans la superrésolution multiframe :

- L'interpolation par le plus proche voisin :

Afin de pouvoir réaliser et appliquer cette interpolation pour la superrésolution multiframe, nous devons déterminer le diagramme de Voronoï de l'image suréchantillonnée en prenant les pixels sources comme des centroïdes. Chaque autre pixel de cette image aura une valeur égale à la valeur du centroïde de la cellule à laquelle il appartient.

La Figure 3.12 représente un exemple de la superrésolution multiframe en appliquant l'algorithme d'interpolation par le plus proche voisin : nous pouvons remarquer que chaque pixel de l'image interpolée a la valeur du pixel source qui joue le rôle de centroïde de la zone à laquelle il appartient.

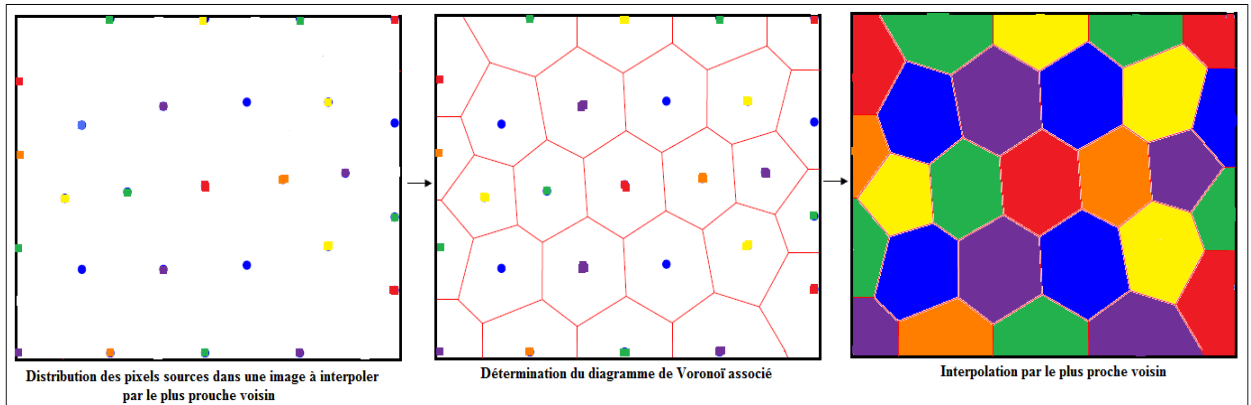


Figure 3. 12 Exemple de la superrésolution multiframe par interpolation par le plus proche voisin

- La superrésolution par interpolation linéaire :

Le principe de l'interpolation linéaire est basé sur la triangulation de l'image source HR_p pour affecter à chaque triangle un plan d'interpolation calculé par rapport aux sommets du triangle.

Dans le cas de la superrésolution multiframe, la triangulation utilisée pour le calcul des plan d'interpolation est la triangulation de Delaunay vu ses caractéristiques géométriques et sa dualité avec le diagramme de Voronoï : En considérant chaque pixel source de la matrice image source HR_p comme un sommet de Delaunay, nous extrapolons les valeurs des autres pixels à partir des plans d'interpolation associés aux triangles de Delaunay auxquels ils appartiennent.[32]

Soit i la valeur du pixel associé au point $p(x, y)$ de l'image HR à extrapoler par interpolation linéaire, soit i_1, i_2 et i_3 les valeurs des trois pixels associés aux points $p(x_1, y_1)$, $p(x_2, y_2)$ et $p(x_3, y_3)$ formant le triangle de Delaunay auquel $p(x, y)$ appartient : nous définissons le plan d'interpolation associé au triangle de Delaunay $p(x_1, y_1)p(x_2, y_2)p(x_3, y_3)$ par l'équation :

$$p(x, y) \in p(x_1, y_1)p(x_2, y_2)p(x_3, y_3) : i = ax + by + c \tag{3.3}$$

Ou :

$$i = [x \quad y \quad 1] \begin{pmatrix} a \\ b \\ c \end{pmatrix} \tag{3.4}$$

Tel que

$$\begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}^{-1} \begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} \quad (3.5)$$

Finalement de 3.3, 3.4 et 3.5, nous aurons :

$$p(x, y) \in p(x_1, y_1)p(x_2, y_2)p(x_3, y_3) :$$

$$i = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}^{-1} \begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} \quad (3.6)$$

- La superrésolution par interpolation bilinéaire :

Cette interpolation exploite exactement le même algorithme présenté dans la superrésolution monoframe pour la superrésolution multiframe : l'extrapolation des pixels est faite par une double interpolation linéaire suivant les deux axes de l'image.

Supposons que la première interpolation linéaire est faite suivant l'axe des X, dans ce cas, nous déterminons toutes les valeurs des pixels situés entre deux points sources successifs de mêmes abscisses : Soit i le pixel associé au point $p(x, y)$ situé entre les deux pixels sources i_1 et i_2 associés aux point $p(x, y_1)$ et $p(x, y_2)$, sa valeur est exprimée par l'équation : $i = \frac{i_2 - i_1}{y_2 - y_1}(y - y_1) + i_1$.

Pour le calcul des valeurs des autres pixels, une deuxième interpolation linéaire est effectuée suivant l'axe des Y en calculant leurs valeurs linéairement par rapport aux deux pixels de mêmes ordonnées qui les entourent « ces deux pixels peuvent être des pixels sources ou des pixels extrapolés » : Soit i le pixel associé au point $p(x, y)$ situé entre deux pixels connus successifs i_1 et i_2 associés aux point $p(x_1, y)$ et $p(x_2, y)$, la valeur de i est exprimée cette fois-ci par l'équation : $i = \frac{i_2 - i_1}{x_2 - x_1}(x - x_1) + i_1$.

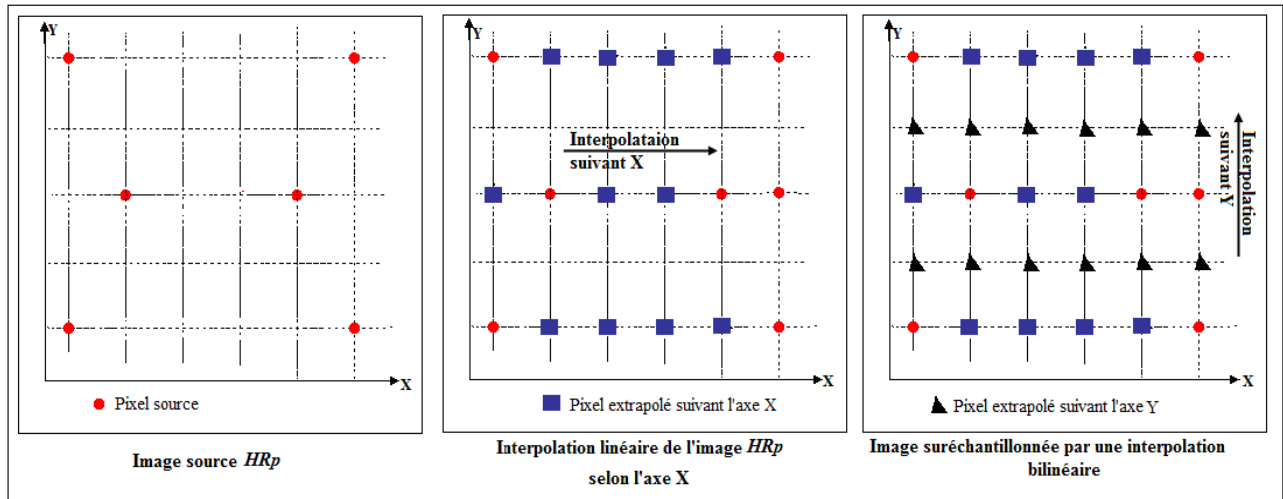


Figure 3.13 Principe de la superrésolution par interpolation bilinéaire

La Figure 3.13 représente la procédure générale de la superrésolution par interpolation bilinéaire, nous pouvons remarquer que cette interpolation permet d'extrapoler tous les pixels de l'image par une double interpolation linéaire. Dans l'exemple de la Figure 3.13, l'extrapolation des pixels selon la direction Y est faite par l'utilisation des couples de pixels composés de pixels sources, de pixels extrapolés suivant la direction de l'axe X ou d'un pixel source et un pixel extrapolé.

- **La superrésolution par interpolation bicubique:**

Nous avons vu précédemment que la superrésolution par interpolation bicubique assure le dynamisme de l'image en utilisant une double interpolation cubique suivant les deux directions X et Y, la fonction d'interpolation résultante est donc de la forme :

$$i(x, y) = \sum_{n=0}^3 \sum_{m=0}^3 a_{nm} x^n y^m \tag{3.7}$$

où $i(x, y)$ représente la valeur du pixel associé au point $p(x, y)$ appartenant à la surface d'interpolation.

Le calcul des coefficients de la fonction d'interpolation nécessite la connaissance de 16 valeurs de points sources entourant la surface d'interpolation. Afin d'optimiser les erreurs d'interpolation, nous avons opté, une autre fois, pour l'utilisation de la triangulation de Delaunay en déterminant les 16 points sources les plus proches du pixel à extrapoler.

Notre algorithme proposé pour la détermination de ces points est un algorithme récurent par niveaux de sélection, chaque niveau détermine un nombre de points utilisés pour le calcul de ces coefficients :

- Niveau 1 : Dans ce niveau, les trois premiers points du calcul de la fonction d'interpolation sont déterminés par les sommets du triangle de Delaunay contenant le point $p(x, y)$,

- Niveau 2 : Les points sources ayant des relations directes par triangulation de Delaunay avec les points de calcul déterminés dans le niveau 1 peuvent être des points du calcul de la fonction d'interpolation. Si le nombre des points déterminés par ce niveau et par le niveau 1 est supérieur ou égal à 16, les points de calcul utilisés seront les 16 points les plus proches du point extrapolé. Sinon, tous les points déterminés à ce niveau sont utilisés pour le calcul de la fonction d'interpolation et on doit passer à un niveau supérieur de sélection.

- Pour chaque niveau N+1, on détermine les points sources ayant une relation par triangulation de Delaunay avec les points de calcul déterminés dans le niveau N. Si la somme du nombre de points déterminés à ce niveau et le nombre des points de calcul déterminés dans les niveaux ultérieurs est supérieur ou égale à 16, on doit sélectionner les 16 points sources les plus proches du point à extrapoler. Sinon, on ajoute les points déterminés par ce niveau comme des points de calcul, puis on passe à un niveau de sélection supérieur.

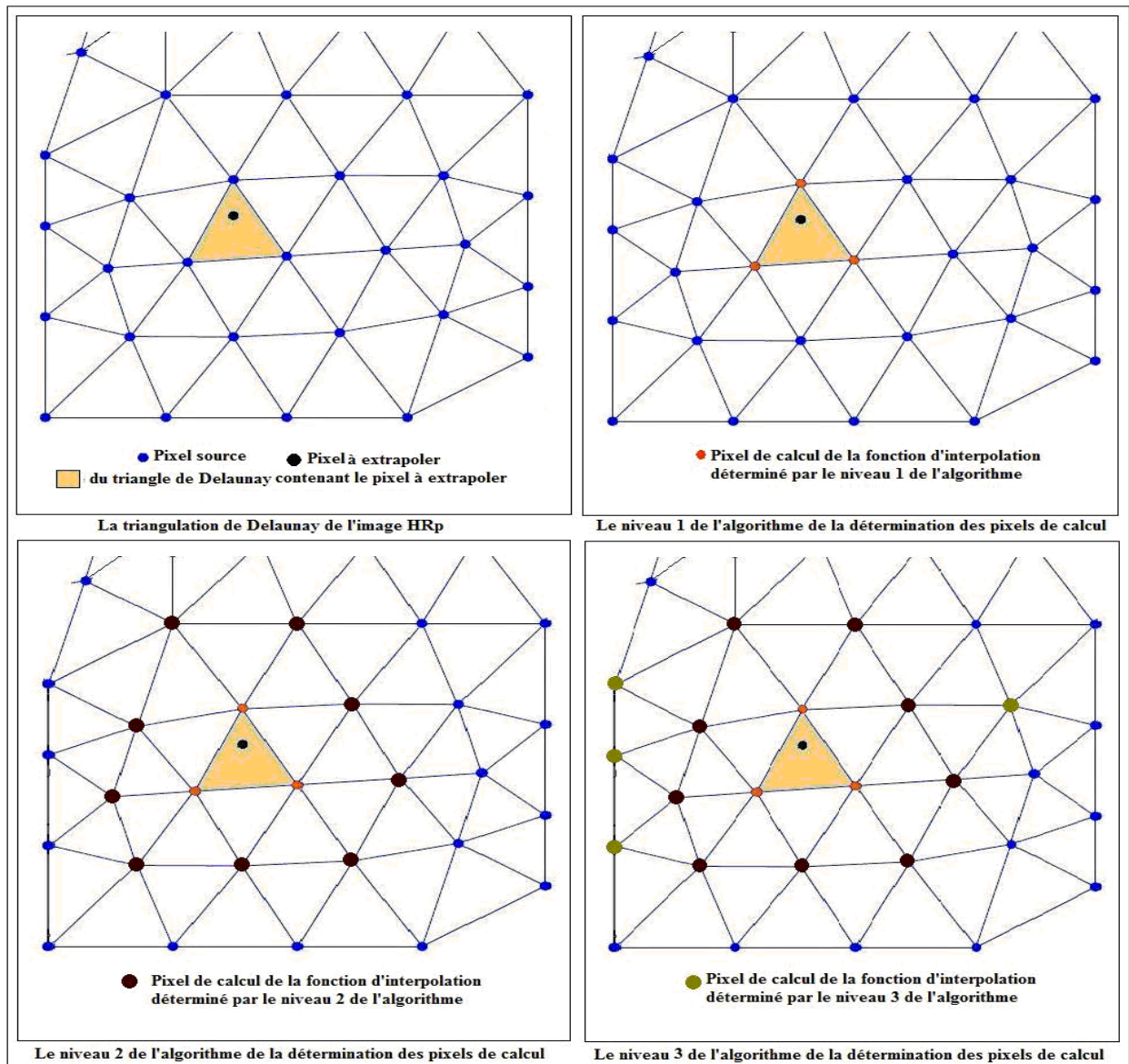


Figure 3. 14 Exemple de la détermination des pixels utilisés dans la définition de la fonction d'interpolation.

La Figure 3.14 représente un exemple qui exprime l'algorithme de sélection des points utilisés pour le calcul de la fonction d'interpolation : nous remarquons que cet exemple nécessite 3 niveaux de sélection pour déterminer les 16 points utilisés pour le calcul de la fonction d'interpolation.

Soit $I_c = [i_1(x_1, y_1), i_2(x_2, y_2), i_3(x_3, y_3) \dots i_{16}(x_{16}, y_{16})]^T$ le vecteur composé des valeurs des 16 points déterminés précédemment pour le calcul de la fonction d'interpolation, avec $X_c = [x_1, x_2, \dots, x_{16}]^T$ et $Y_c = [y_1, y_2, \dots, y_{16}]^T$ les vecteurs associés à leurs abscisses et leurs ordonnées.

Nous avons pour chaque point $i(x, y)$ de la surface d'interpolation :

$$i(x, y) = \sum_{n=0}^3 \sum_{m=0}^3 a_{nm} x^n y^m$$

$$\Rightarrow i(x, y) = \begin{bmatrix} x^0 & x^1 & x^2 & x^3 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} y^0 \\ y^1 \\ y^2 \\ y^3 \end{bmatrix} \quad (3.8)$$

La relation (3.8) peut être traduite par la relation (3.9) :

$$\Rightarrow i(x, y) = \begin{bmatrix} x^0 y^0 \\ x^0 y^1 \\ x^0 y^2 \\ x^0 y^3 \\ x^1 y^0 \\ x^1 y^1 \\ x^1 y^2 \\ x^1 y^3 \\ x^2 y^0 \\ x^2 y^1 \\ x^2 y^2 \\ x^2 y^3 \\ x^3 y^0 \\ x^3 y^1 \\ x^3 y^2 \\ x^3 y^3 \end{bmatrix}^T * \begin{bmatrix} a_{00} \\ a_{01} \\ a_{02} \\ a_{03} \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{20} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{30} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} \quad (3.9)$$

Soit A et G deux vecteurs définis comme suit :

$$A = [a_{00} \ a_{01} \ a_{02} \ a_{03} \ a_{10} \ a_{11} \ a_{12} \ a_{13} \ a_{20} \ a_{21} \ a_{22} \ a_{23} \ a_{30} \ a_{31} \ a_{32} \ a_{33}]^T$$

$$G(x, y) = \begin{bmatrix} x^0 y^0 & x^0 y^1 & x^0 y^2 & x^0 y^3 & x^1 y^0 & x^1 y^1 & x^1 y^2 & x^1 y^3 & x^2 y^0 & x^2 y^1 & x^2 y^2 & x^2 y^3 & x^3 y^0 & x^3 y^1 & x^3 y^2 & x^3 y^3 \end{bmatrix}$$

D'où : (4.9) $\Rightarrow i(x, y) = G(x, y)A$

Afin de déterminer le vecteur A , nous allons utiliser les 16 pixels sources les plus proches de la surface d'interpolation, composant le vecteur I_c , en supposant qu'ils vérifient la relation (3.7), c'est-à-dire :

$$\forall j \in \{1,2,\dots,16\}: I_c[j] = \sum_{n=0}^3 \sum_{m=0}^3 a_{nm} (X_c[j])^n (Y_c[j])^m$$

$$I_c = \left\{ \begin{array}{c} \left[\begin{array}{c} X_c^0 \\ X_c^0 \\ X_c^0 \\ X_c^0 \\ X_c^1 \\ X_c^1 \\ X_c^1 \\ X_c^1 \\ X_c^2 \\ X_c^2 \\ X_c^2 \\ X_c^2 \\ X_c^3 \\ X_c^3 \\ X_c^3 \\ X_c^3 \end{array} \right]^T \\ \cdot * \\ \left[\begin{array}{c} Y_c^0 \\ Y_c^1 \\ Y_c^2 \\ Y_c^3 \\ Y_c^0 \\ Y_c^1 \\ Y_c^2 \\ Y_c^3 \\ Y_c^0 \\ Y_c^1 \\ Y_c^2 \\ Y_c^3 \\ Y_c^0 \\ Y_c^1 \\ Y_c^2 \\ Y_c^3 \end{array} \right]^T \end{array} \right\} A \quad (3.10)$$

avec : $X_c^n = [x_1^n, x_2^n, \dots, x_{16}^n]^T$ et $Y_c^n = [y_1^n, y_2^n, \dots, y_{16}^n]^T$ et $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot * \begin{bmatrix} x & y \\ w & z \end{bmatrix} = \begin{bmatrix} ax & by \\ cw & dz \end{bmatrix}$

Posons :

$$I_c = W A \quad (3.11)$$

avec :

$$W = \left\{ \begin{array}{c} \left[\begin{array}{c} X_c^0 \\ X_c^0 \\ X_c^0 \\ X_c^0 \\ X_c^1 \\ X_c^1 \\ X_c^1 \\ X_c^1 \\ X_c^2 \\ X_c^2 \\ X_c^2 \\ X_c^2 \\ X_c^3 \\ X_c^3 \\ X_c^3 \\ X_c^3 \end{array} \right]^T \\ \cdot * \\ \left[\begin{array}{c} Y_c^0 \\ Y_c^1 \\ Y_c^2 \\ Y_c^3 \\ Y_c^0 \\ Y_c^1 \\ Y_c^2 \\ Y_c^3 \\ Y_c^0 \\ Y_c^1 \\ Y_c^2 \\ Y_c^3 \\ Y_c^0 \\ Y_c^1 \\ Y_c^2 \\ Y_c^3 \end{array} \right]^T \end{array} \right\} \quad (3.12)$$

Des relations (3.10) et (3.12), nous constatons que la matrice W est une matrice carrée de dimension 16×16 , d'où :

$$(3.11) \Rightarrow A = W^{-1} I_c \tag{3.13}$$

En remplaçant 3.13 dans 3.9, nous pouvons extrapoler le point $i(x, y)$ appartenant à la surface d'interpolation par la relation :

$$i(x, y) = G(x, y)W^{-1}I_c \tag{3.14}$$

- La superrésolution par interpolation par spline bicubique:

L'interpolation par spline bicubique est une interpolation qui assure le dynamisme de l'image suréchantillonnée, elle est plus rapide que l'interpolation bicubique malgré qu'elles utilisent la même forme polynomiale pour chaque point $i(x, y)$ appartenant à la surface d'interpolation : $i(x, y) = \sum_{n=0}^3 \sum_{m=0}^3 a_{nm} x^n y^m$. En effet, l'interpolation par spline bicubique utilise les

quatre points sources les plus proches de la surface d'interpolation pour calculer les 16 coefficients de la fonction d'interpolation bicubique en supposant qu'elle utilise le spline naturel.

L'application de l'algorithme récurant par niveaux de sélection détermine ces quatre points sources $i_1(x_1, y_1), i_2(x_2, y_2), i_3(x_3, y_3)$ et $i_4(x_4, y_4)$ en deux niveaux : Dans le premier niveau, nous déterminons trois points tandis que le quatrième point est déterminé en passant au deuxième niveau.

Afin de déterminer les coefficients de la fonction d'interpolation, un système de 16 équations linéaires doit être établi. Quatre équations sont déterminées en supposant que les points de calcul sont des points de la surface d'interpolation et les autres équations peuvent être établies en exploitant la notion de spline naturel.

La notion de spline naturel signifie que les dérivées secondes de la fonction d'interpolation s'annulent aux quatre points de calcul. Le système s'écrit donc sous la forme :

$$\forall k \in \{1, 2, 3, 4\} : \begin{cases} i(x_k, y_k) = \sum_{n=0}^3 \sum_{m=0}^3 a_{nm} x_k^n y_k^m \\ \frac{\partial^2 i(x_k, y_k)}{\partial x^2} = 0 \\ \frac{\partial^2 i(x_k, y_k)}{\partial y^2} = 0 \\ \frac{\partial^2 i(x_k, y_k)}{\partial x \partial y} = 0 \end{cases} \tag{3.15}$$

Ce qui implique :

III.6 Conclusion

Ce chapitre nous a montré le rôle de la superrésolution multiframe dans la réduction du taux d'erreur produite lors de la reconstruction d'une image *HR* ; ainsi que les différents algorithmes d'adaptation des techniques de superrésolution monoframe par interpolation déjà étudiées à la superrésolution multiframe en utilisant le diagramme de Voronoï et la triangulation de Delaunay.

Nous avons remarqué que la superrésolution multiframe par filtrage de Whittaker est la technique de superrésolution multiframe par interpolation la plus choisie et la plus utilisée, cela peut-être dû à sa simplicité, sa rapidité et son économie en ressources. En effet, cette technique, qui donne généralement de bons résultats par un simple filtrage, ne nécessite pas une longue opération de détermination des pixels sources utilisés pour le calcul de chaque pixel extrapolé.

L'expérience nous a montré que l'interpolation par le plus proche voisin et l'interpolation bilinéaire sont des interpolations rapides et idéales pour les images à évolution linéaire. Mais malheureusement, elles n'ont pas le pouvoir d'introduire un dynamisme aux images suréchantillonnées. Tandis que l'utilisation d'une interpolation de degré 3, voire l'interpolation bicubique et l'interpolation par spline bicubique, assure la création d'un effet de dynamisme sur l'image en utilisant des algorithmes de calcul lent et nécessitant des ressources supplémentaires.

Chapitre IV

IMPLEMENTATION ET APPLICATIONS

Chapitre IV

IMPLEMENTATION ET APPLICATIONS

IV.1 Introduction

Nous avons vu dans les chapitres précédents l'utilité de la superrésolution dans l'analyse des signaux, ainsi que les différents algorithmes de la superrésolution appliqués aux signaux numériques bidimensionnels tels que les images numériques. Ces algorithmes sont classés en deux catégories : les algorithmes de la superrésolution monoframe et les algorithmes de superrésolution multiframe.

Dans ce chapitre, nous allons présenter notre application qui consiste en l'implémentation des différents algorithmes étudiés dans les chapitres précédents nous permettant de les exploiter, de les analyser, de les optimiser et de comparer leurs résultats.

IV.2 Présentation générale de l'application

Le but de ce travail est le développement d'une application capable d'améliorer et d'augmenter la résolution des images numériques. Pour cela nous avons choisi la plateforme de développement MATLAB 7 qui est un outil mathématique développé, puissant, optimisé, riche en fonctions adaptées au traitement d'image et doté du GUIDE (Graphical User Interface Development Environment) dont l'environnement de développement d'interface IDE est dédié à la création des interfaces graphiques Homme/machine.

Notre application, nommée « *ENP Superrésolution* », est composée de trois environnements : l'environnement Superrésolution monoframe, l'environnement Superrésolution multiframe et l'environnement Développement et Etudes.

La Figure 4.1 montre la fenêtre principale développée qui nous permet d'accéder à ces environnements en tapant *superresolution* sur la fenêtre principale de Matlab.

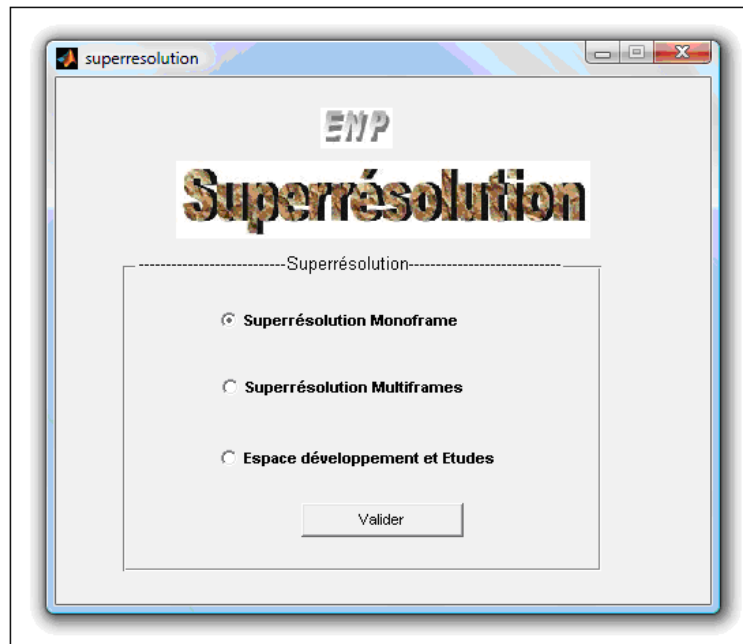


Figure 4. 1 Fenêtre principale de l'application *ENSP Superrésolution*.

Nous allons présenter, dans ce qui suit, les interfaces et les fenêtres consacrées à chaque environnement en détaillant les principaux algorithmes et fonctions utilisés:

IV.3 Environnement *Superrésolution Monoframe*

Cet environnement exploite les différents algorithmes de superrésolution monoframe développés dans le chapitre II. Il nous permet de lire l'image, de déterminer la zone de travail, de la souséchantillonner ou de la suréchantillonner avec des algorithmes de superrésolution par interpolation polynomiale ou par filtrage en enregistrant l'historique de toutes les opérations effectuées afin de faciliter l'étude et le choix des meilleurs algorithmes adaptés à celle-ci.

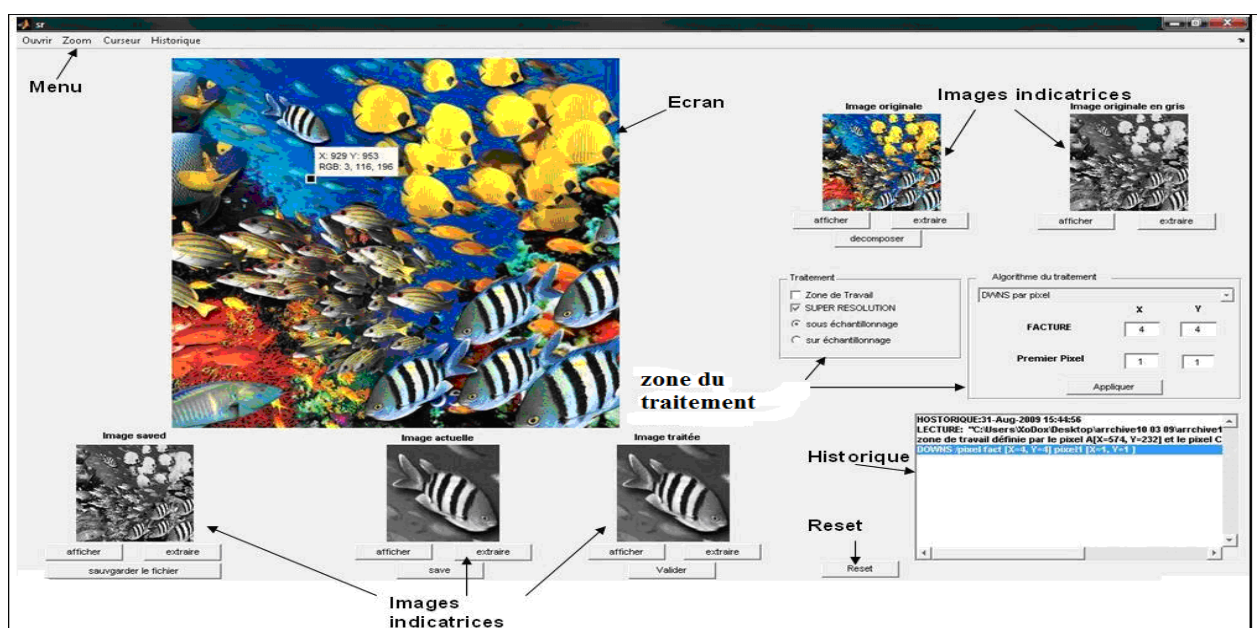


Figure 4. 2 Environnement *Superrésolution Monoframe*.

La Figure 4.2 représente la fenêtre dédiée à l’environnement Superrésolution monoframe, nous remarquons qu’elle est composée de :

IV.3.1 Barre de Menu

Située en haut de la fenêtre, la barre de menu est composée du :

- **Bouton Ouvrir**

Ce bouton permet de lire des images de type BMP, JPEG, TIFF, GIF, PNG, HDF, PCX, ICO, CUR, RAS, PBM, PGM, PPM, ainsi que des images médicales de type DICOM pour le traitement de leurs résolutions.

Le clic sur ce bouton fait apparaître une boîte de dialogue qui permet de choisir une nouvelle image à traiter. Une fois l’image choisie, elle s’affiche sur l’écran et sur les images indicatrices.

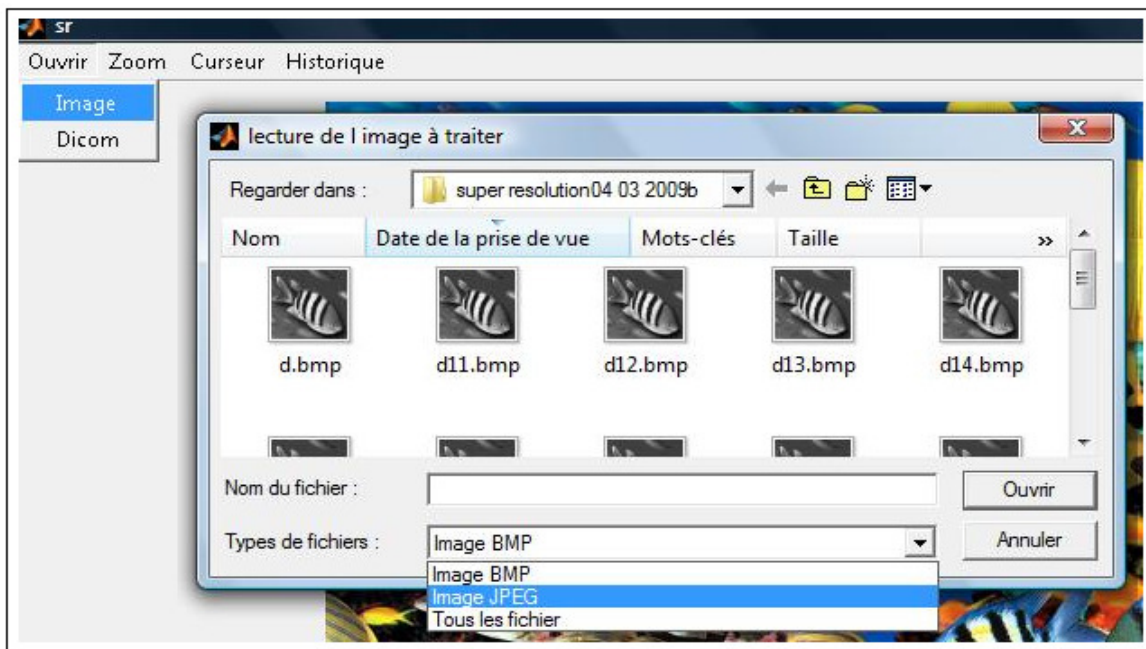


Figure 4. 3 Bouton Ouvrir.

- **Bouton Zoom**

Ce bouton permet d’activer, d’annuler ou de désactiver l’option zoom qui sert à agrandir les zones sélectionnées des images.

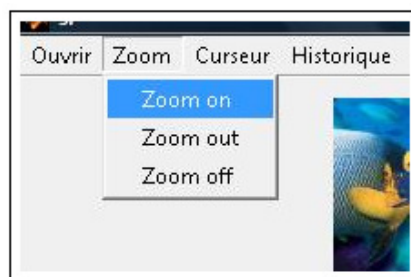


Figure 4. 4 Bouton Zoom .

- Bouton *Curseur*

L'activation de cette option permet d'afficher les informations du pixel sélectionné de l'image (cordonnées et valeur)



Figure 4. 5 Bouton *Curseur*.

- Bouton *Historique*:

Ce bouton donne la possibilité de choisir entre la sauvegarde et l'affichage d'une image traitée.

La sauvegarde de l'image se fait par son enregistrement sous forme d'un fichier BMP ou JPEG et par la création d'un fichier *HISTORIQUE* contenant l'image traitée, des informations sur l'image initiale et toutes les opérations effectuées lors du traitement. Ce fichier sera généré dans un dossier ayant un nom composé du nom de fichier (BMP ou JPEG) de l'image traitée enregistrée et de la date et l'heure de l'enregistrement. A la fin de l'enregistrement, un message de confirmation s'affichera pour confirmer la fin et l'emplacement des fichiers sauvegardés.

La figure 4.6 montre le bouton « *Save image actuelle* » qui sert à sauvegarder l'image traitée et la boîte de dialogue utilisée pour cela.

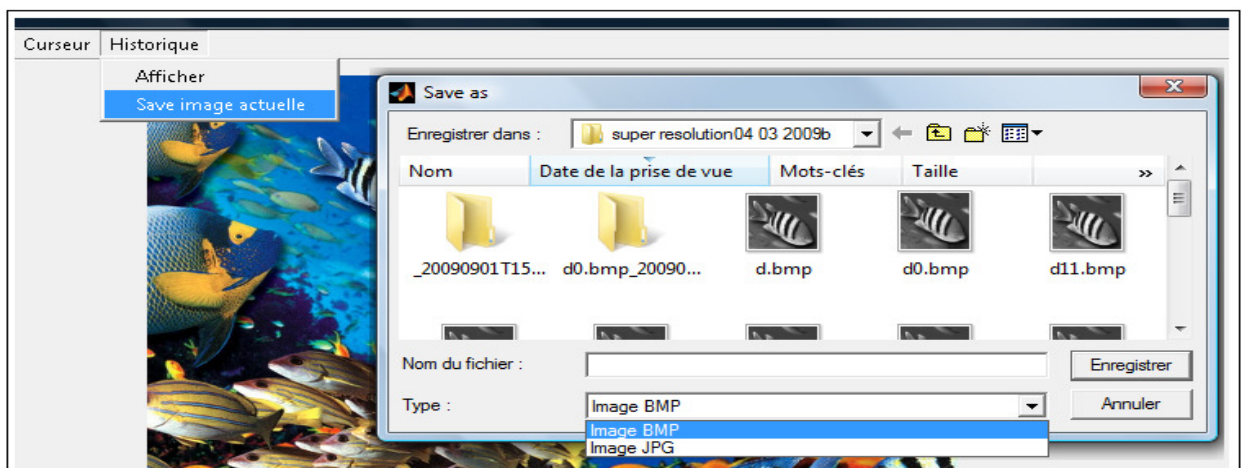


Figure 4. 6 Bouton *Historique* / *Save image actuelle*.

L'affichage de l'image enregistrée se fait par la lecture de son fichier *HISTORIQUE*, cette lecture fait apparaître deux autres fenêtres : la première « la *fenêtre historique* » affiche la source de l'image avec tous les traitements effectués, et la deuxième « *fenêtre image traitée* » affiche l'image traitée.

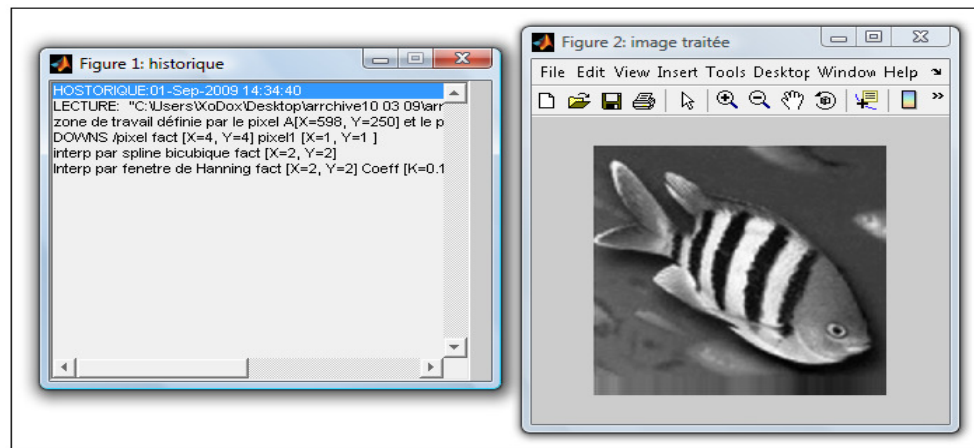


Figure 4. 7 Fenêtres historique et image traitée.

La Figure 4.7 représente les deux fenêtres d'affichage, la fenêtre *historique* et la fenêtre image traitée.

Le fait que les historiques s'affichent sur des fenêtres différentes numérotées permet d'ouvrir plusieurs historiques à la fois et d'effectuer des comparaisons entre des images déjà traitées et l'image en cours de traitement.

IV.3.2 Zone de texte *Historique*

Cette zone est générée automatiquement à l'ouverture d'une nouvelle image, elle affiche toutes les opérations de traitement effectuées afin de les enregistrer avec l'image traitée dans le fichier *HISTORIQUE*.

La Figure 4.8 représente un exemple montrant la structure générale de cette zone

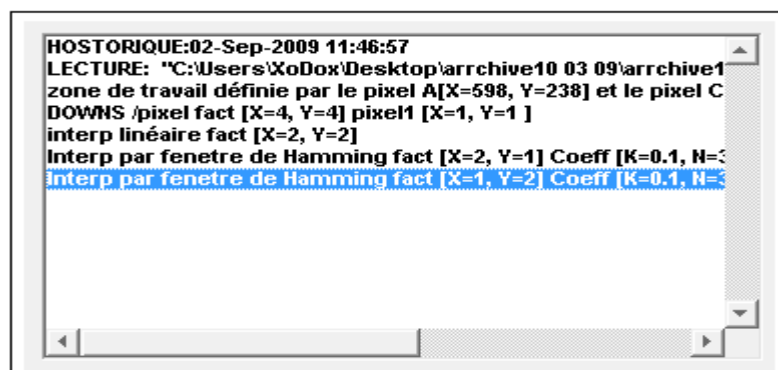


Figure 4. 8 Zone de texte Historique.

Cette structure permet d'avoir les informations suivantes :

- **Informations sur la date et l'heure:**

La première ligne de la zone de texte *Historique* contient toujours la date et l'heure du début de traitement sous la forme :

HISTORIQUE : JJ-MOIS-AAAA HH :MM :SS.

- **L'adresse de l'image source :**

Nous trouvons dans la deuxième ligne l'adresse de l'image initiale lue, cette ligne est sous la forme :

LECTURE : « @adresse de l'image ».

- **Informations sur les traitements effectués:**

Dés la troisième ligne, nous pouvons trouver des informations sur toutes les opérations du traitement et sur les paramètres choisis. Chaque opération est mentionnée avec ses paramètres sur une ligne indépendante en respectant l'ordre chronologique du traitement. La syntaxe utilisée pour exprimer ces opérations est de la forme :

NOM_DE_L'OPERATION PARAMETRES.

IV.3.3 Images indicatrices

Ces espaces d'affichage, qui occupent la majorité de la surface de la fenêtre, sont composées de :

- **Ecran:**

Occupant plus de la moitié de la surface de fenêtre, l'écran représente son plus grand espace d'affichage, il permet d'exposer les autres images indicatrices 16 fois plus grandes.

- **Image indicatrice *Image originale*:**

Cet espace qui affiche l'image initiale lue sans modification est équipé de trois boutons:

- le bouton *afficher* permet d'afficher l'image originale sur l'écran.
- le bouton *extraire* permet d'afficher l'image originale sur une autre fenêtre.
- le bouton *décomposer* permet d'afficher l'image originale avec ses trois composantes

RGB sur une autre fenêtre.

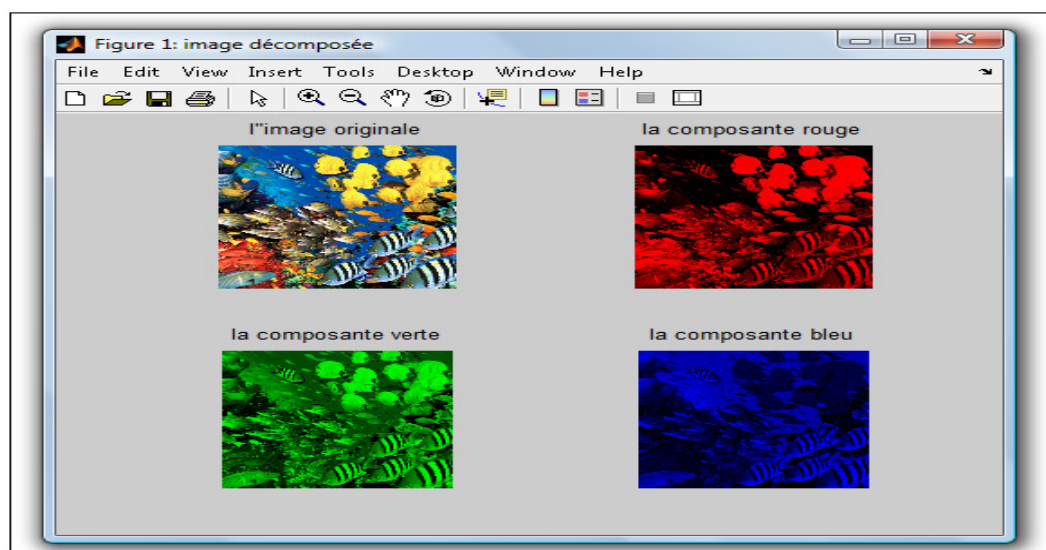


Figure 4. 9 Décomposition de l'image originale en composantes RGB.

- **Image indicatrice *Image originale en niveaux de gris*:**

Dans ce travail nous avons choisi de traiter les images bidimensionnelles monochromes, c'est-à-dire les images ayant pour chacun de leurs pixels une valeur scalaire. En effet, la majorité des images médicales sont des images monochromes, c'est pourquoi nous avons opté pour la transformation de toutes les images en images à niveaux de gris avant le traitement.

Vu que les composantes RGB de l'image sont des composantes linéairement indépendantes, le traitement de la résolution d'une de ces images revient au traitement de chaque composante indépendamment en considérant chacune comme image monochrome.

L'espace d'affichage de l'image originale en niveaux de gris est équipé de deux boutons : le bouton *afficher* pour afficher l'image en niveaux de gris sur l'écran et le bouton *extraire* pour extraire l'image en niveaux de gris et l'afficher sur une autre fenêtre.

- **Image indicatrice *Image actuelle*:**

Cet espace est dédié à l'affichage de l'image à laquelle nous voulons appliquer la prochaine opération de traitement. Il est équipé de trois boutons:

- le bouton *afficher* permet d'afficher l'*image actuelle* sur l'écran,
- le bouton *extraire* permet d'afficher l'*image actuelle* sur une autre fenêtre,
- le bouton *save* permet d'enregistrer l'*image actuelle* dans une autre image « *image saved* » avec tout l'historique des opérations effectuées.

- **Image indicatrice *Image saved*:**

Cet espace affiche l'image prête à être enregistrée dans un fichier BMP ou JPEG, il présente trois options:

- le bouton *afficher* permet d'afficher l'*image saved* sur l'écran,
- le bouton *extraire* permet d'afficher l'*image saved* sur une autre fenêtre,
- le bouton *sauvegarder le fichier* permet l'enregistrement de l'*image saved* sous forme d'un fichier BMP ou JPEG ainsi que la création du fichier *Historique* qui contient l'*image saved* et son historique.

- **Image indicatrice de l'*Image traitée*:**

Ce dernier espace affiche l'image obtenue par la réalisation d'une opération de traitement sur l'image actuelle, il est doté de trois boutons d'options :

- le bouton *afficher* permet d'afficher l'*image traitée* sur l'écran,
- le bouton *extraire* permet d'afficher l'*image traitée* sur une autre fenêtre,
- le bouton *valider* sert à valider le traitement en remplaçant le contenu de l'*image actuelle* par l'*image traitée*.

La Figure 4.10 présente l'organigramme général du fonctionnement de l'environnement de la superrésolution monoframe, de la lecture de l'image à l'enregistrement de l'image traitée.

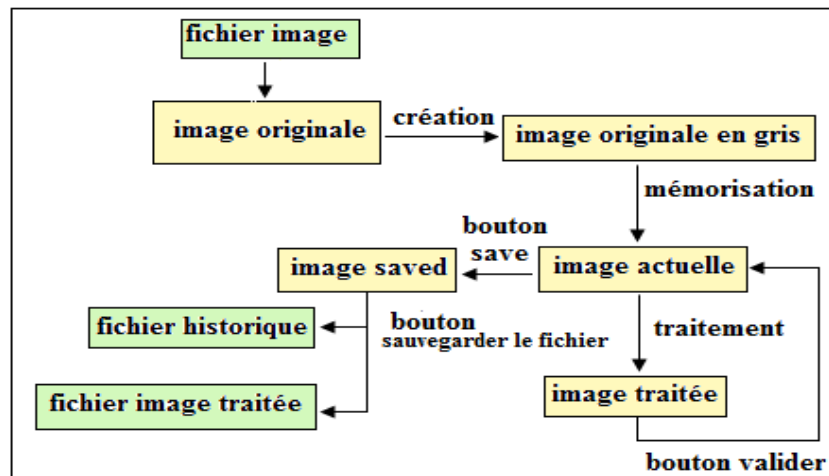


Figure 4. 10 Organigramme du fonctionnement de l'environnement superrésolution monoframe.

IV.3.4 Zone de traitement

Cette zone expose les différentes opérations et algorithmes à effectuer sur l'*image actuelle*, elle est composée de deux panels : le panel *Traitement* pour choisir le type d'opération et le panel *Algorithme du traitement* pour déterminer l'algorithme à utiliser.

Concernant les types d'opérations disponibles dans cet environnement nous trouvons :

- **Limitation de la zone de travail:**

Cette opération permet la détermination de la zone d'intérêt de l'*image actuelle* afin de limiter les calculs sur cette zone. En effet, la limitation de la zone de calcul permet d'optimiser le temps et les ressources du calcul.

La zone de travail est définie comme une partie rectangulaire de l'*image actuelle* encadrée par le rectangle $ABCD$ tel que les coordonnées de ses sommets $A(x_A, y_A), B(x_B, y_B), C(x_C, y_C)$ et $D(x_D, y_D)$ vérifient :

$$\begin{cases} x_A = \min(x_A, x_B, x_C, x_D) \wedge y_A = \min(y_A, y_B, y_C, y_D) \\ x_C = \max(x_A, x_B, x_C, x_D) \wedge y_C = \max(y_A, y_B, y_C, y_D) \\ x_B = x_A \quad \wedge \quad y_B = y_C \\ x_D = x_C \quad \wedge \quad y_D = y_A \end{cases} \quad (4.1)$$

Nous avons proposé deux algorithmes pour la limitation de la zone de travail et pour la définition du rectangle $ABCD$:

- la définition du rectangle $ABCD$ en introduisant comme paramètre les coordonnées des pixels A et C , les coordonnées des autres points seront alors déterminées directement de la relation 4.1 ;

- la définition du rectangle $ABCD$ en introduisant comme paramètre les coordonnées du pixel A et le couple (x_{rec}, y_{rec}) des dimensions du rectangle selon l'axe X et l'axe Y. les coordonnées des autres pixels seront déterminées par la relation 4.2 :

$$\begin{cases} x_C = x_A + x_{rec} & y_C = y_A + y_{rec} \\ x_D = x_C & y_D = y_A \\ x_B = x_A & y_B = y_C \end{cases} \quad (4.2)$$

Souséchantillonnage de l'image:

Cette opération est très utile pour diminuer la résolution de l'image actuelle afin de simuler l'effet des appareils d'acquisition de faible résolution sur la prise d'image et de vérifier l'efficacité des algorithmes et des paramètres de la super-résolution en comparant l'image avant le souséchantillonnage avec celle obtenue après une opération de souséchantillonnage suivie des opérations de suréchantillonnage.

Nous avons constaté qu'il existe deux types de capteurs d'image disponibles : des capteurs qui affectent aux pixels les valeurs d'intensité prises à leurs niveaux, et d'autres qui affectent les valeurs des intensités moyennes de la zone qu'ils gèrent. D'où, la proposition des deux fonctions de souséchantillonnage suivantes en simulant la position des capteurs par les positions des pixels de l'image initiale et la distance entre les capteurs par le facteur de sous échantillonnage :

- la fonction *DWNS par pixel* :

Cette fonction, ayant comme paramètres d'entrée les coordonnées du premier pixel de l'image actuelle $p_{act}(x_{act1}, y_{act1})$ de valeur $i_{act}(x_{act1}, y_{act1})$ et le facteur de souséchantillonnage (k_x, k_y) , affecte à chaque pixel de l'image traitée $p_{tr}(x_{tr}, y_{tr})$ la valeur $i_{tr}(x_{tr}, y_{tr})$ tel que :

$$i_{tr}(x_{tr}, y_{tr}) = i_{act}(x_{act1} + k_x(x_{tr} - 1), y_{act1} + k_y(y_{tr} - 1)) \quad (4.3)$$

avec $i_{act}(x_{act1} + k_x(x_{tr} - 1), y_{act1} + k_y(y_{tr} - 1))$ la valeur du pixel $p_{act}(x_{act1} + k_x(x_{tr} - 1), y_{act1} + k_y(y_{tr} - 1))$.

- la fonction *DWNS par moyenne* :

Cette fonction, qui nécessite l'introduction du facteur de souséchantillonnage (k_x, k_y) comme paramètre, affecte à chaque pixel de l'image traitée $p_{tr}(x_{tr}, y_{tr})$ la valeur $i_{tr}(x_{tr}, y_{tr})$ tel que :

$$i_{tr}(x_{tr}, y_{tr}) = \sum_{m=1}^{k_x} \sum_{n=1}^{k_y} \frac{i_{act}(m + k_x(x_{tr} - 1), n + k_y(y_{tr} - 1))}{k_x k_y} \quad (4.4)$$

et $i_{act}(x, y)$ représente la valeur du pixel $p_{act}(x, y)$ de l'*image actuelle*.

- **Suréchantillonnage de l'image:**

L'opération du suréchantillonnage exploite les algorithmes de superrésolution par interpolation polynomiale et par filtrage étudiés et développés dans le chapitre II de ce travail, ces algorithmes sont implémentés par les 9 fonctions suivantes :

- la fonction *Interpolation par le plus proche voisin* :

Cette fonction, qui implémente l'interpolation polynomiale d'ordre zéro développée dans le paragraphe II.2.1, ne nécessite que l'introduction du facteur de suréchantillonnage (Kx, Ky) comme paramètre pour calculer l'image traitée de l'image actuelle. En effet la valeur $i_{tr}(x_{tr}, y_{tr})$ de chaque pixel $p_{tr}(x_{tr}, y_{tr})$ est donnée par la relation :

$$i_{tr}(x, y) = i_{act} \left(1 + \frac{(x-1) - ((x-1) \bmod (Kx))}{Kx}, 1 + \frac{(y-1) - ((y-1) \bmod (Ky))}{Ky} \right) \quad (4.5)$$

où $i_{act}(x, y)$ représente la valeur du pixel $p_{act}(x, y)$ de l'*image actuelle*.

- la fonction *Interpolation linéaire* :

Cette fonction exploite le développement théorique du paragraphe II.2.2 en utilisant la même triangulation présentée dans la Figure 2.5 et les calculs du plan avec les relations 2.6 et 2.7 simplifiées pour le calcul des pixels de la matrice *image traitée* :

$$\begin{cases} p(x, y) \in (T1) \Leftrightarrow i(x, y) = \frac{i(x_1, y_1) - i(x_0, y_1)}{(x_1 - x_0)}(x - x_0) + \frac{i(x_0, y_1) - i(x_0, y_0)}{(y_1 - y_0)}(y - y_0) + i(x_0, y_0) \\ p(x, y) \in (T2) \Leftrightarrow i(x, y) = \frac{i(x_1, y_0) - i(x_0, y_0)}{(x_1 - x_0)}(x - x_0) + \frac{i(x_1, y_1) - i(x_1, y_0)}{(y_1 - y_0)}(y - y_0) + i(x_0, y_0) \end{cases}$$

Le seul paramètre qui doit être introduit pour les calculs dans cette fonction est le facteur de suréchantillonnage (Kx, Ky) .

- la fonction *Interpolation bilinéaire* :

Le suréchantillonnage par interpolation polynomiale bilinéaire a fait l'objet du paragraphe II.2.3, l'implémentation de cette interpolation nécessite l'introduction du facteur de suréchantillonnage (Kx, Ky) afin d'utiliser la relation 2.11 du chapitre II pour calculer la matrice *image traitée*.

- la fonction *Interpolation bicubique* :

Cette fonction représente l'implémentation de l'algorithme de superrésolution par interpolation polynomiale d'ordre 3 exprimée dans le paragraphe II.2.4 en utilisant les 16 points sources voisins et la relation 2.23 pour le calcul de chaque pixel de l'*image traitée*.

- la fonction *Interpolation par spline bicubique* :

Cette fonction implémente le dernier type d'interpolations polynomiales exposées dans le chapitre II. Il s'agit de l'interpolation polynomiale d'ordre 3 par spline bicubique, cette fonction exige la connaissance du facteur de suréchantillonnage pour la détermination des valeurs des pixels de l'*image traitée* en exploitant leurs quatre pixels sources voisins comme indiqué dans le paragraphe II.2.5.

- la fonction *Interpolation par filtrage de Whittaker avec fenêtre rectangulaire* :

Cette fonction utilise l'algorithme de superrésolution par filtrage de Whittaker fenêtré par une fenêtre rectangulaire fpb_{rec} développé dans le paragraphe II.3.2. Le filtre fpb_{rec} est défini sur une fenêtre de dimension $n \times n$ et par sa réponse impulsionnelle:

$$fpb_{rec} = sinc2d * w$$

Tel que :

$$sinc2d(x, y) = k.KxKy \frac{\sin\left(\frac{\pi.x}{Kx}\right)}{x} \cdot \frac{\sin\left(\frac{\pi.y}{Ky}\right)}{y}$$

$$w(x, y) = \begin{cases} 1 & [-n, n] \times [-n, n] \\ 0 & \text{ailleurs} \end{cases}$$

où (Kx, Ky) , n et k sont des paramètres à introduire définis par : (Kx, Ky) le couple de suréchantillonnage, avec n la demi longueur du filtre et k un facteur de correction de l'intensité de l'image.

- la fonction *Interpolation par filtrage de Whittaker avec fenêtre Hamming* :

Cette fonction utilise l'algorithme de superrésolution par filtrage de Whittaker fenêtré par une fenêtre Hamming fpb_{ham} développé dans le paragraphe II.2.2. Le filtre fpb_{ham} est défini sur une fenêtre de dimension $n \times n$ et par sa réponse impulsionnelle:

$$fpb_{ham} = sinc2d * w$$

Tel que :

$$w(x, y) = \left(0.54 - 0.46 \cos\left(\frac{2\pi x}{n}\right) \right) \left(0.54 - 0.46 \cos\left(\frac{2\pi y}{n}\right) \right)$$

où (Kx, Ky) , n et k sont des paramètres à introduire définis par : (Kx, Ky) le facteur de suréchantillonnage, avec n la demi longueur du filtre et k un facteur de correction de l'intensité de l'image.

- la fonction *Interpolation par filtrage de Whittaker avec fenêtre de Hann* :

De même que pour la fonction précédente, cette fonction exploite le filtrage de Whittaker fenêtré par une fenêtre de Hann fpb_{han} pour la superrésolution comme indiqué dans le paragraphe II.3.2. Ce filtre est défini sur une fenêtre w de dimension $n \times n$ et par la réponse impulsionnelle:

$$fpb_{han} = sinc2d * w$$

Tel que :

$$w(x, y) = \left(0.5 - 0.5 \cos \left(\frac{2\pi x}{n} \right) \right) \left(0.5 - 0.5 \cos \left(\frac{2\pi y}{n} \right) \right)$$

où (Kx, Ky) , n et k sont des paramètres à introduire définis par : (Kx, Ky) le facteur de suréchantillonnage, avec n la demi longueur du filtre et k un facteur de correction de l'intensité de l'image.

- la fonction *Interpolation par filtrage de Whittaker avec fenêtre de Blackman* :

Cette fonction représente la dernière fonction d'interpolation par filtrage développée et implémentée dans cet environnement. Le filtre $fpb_{blackman}$ utilisé est défini par :

$$fpb_{blackman} = sinc2d * w$$

Tel que :

$$w(x, y) = 0.42 - 0.5 \cos \left(2\pi \frac{x}{N_x} \right) + 0.08 \cos \left(4\pi \frac{x}{N_x} \right)$$

Où (Kx, Ky) , n et k sont des paramètres à introduire définis par : (Kx, Ky) le facteur de suréchantillonnage, avec n la demi longueur du filtre et k un facteur de correction de l'intensité de l'image.

Nous avons spécifié une fenêtre dans l'environnement Développements et Etudes pour le calcul et l'optimisation des paramètres n et k en fonction de l'image utilisée et du facteur de suréchantillonnage (Kx, Ky) en minimisant le taux d'erreurs produites pour chacun des algorithmes de superrésolution monoframe par filtrage.

IV.3.5 Bouton *Menu principal* et bouton *Reset*

Le bouton *Menu principal* permet l'affichage de la fenêtre du menu principal afin d'utiliser un autre environnement de travail. Le bouton *reset* est un bouton qui annule les traitements effectués sur l'image traitée.

IV.4 Environnement *Superrésolution Multiframe*

Cet environnement est une implémentation des algorithmes de superrésolution multiframe exposés dans le chapitre III de ce travail, il permet de lire les imagettes, de les placer sur la matrice HRz en formant la matrice image HRp , d'interpoler HRp par les algorithmes de superrésolution polynomiale et par filtrage pour construire une image de haute résolution en sauvegardant l'historique des opérations effectuées.

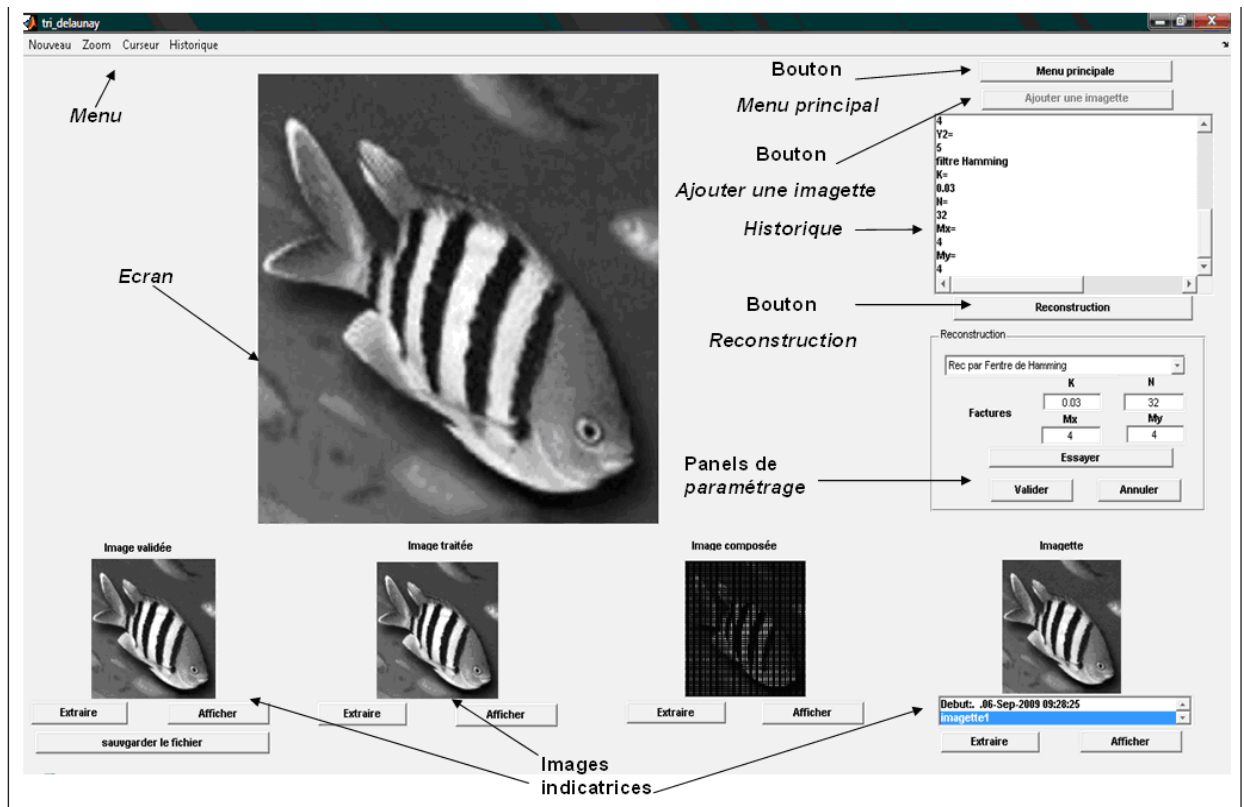


Figure 4. 11 Environnement *Superrésolution Multiframe*.

La Figure 4.11 représente la fenêtre dédiée à l'environnement *Superrésolution multiframe*. Nous avons choisi de suivre la même philosophie dans sa présentation que celle de l'environnement *Superrésolution monoframe*, afin de faciliter son exploitation. En effet nous remarquons qu'elle est composée de :

IV.4.1 Barre de *Menu*

La Figure 4.12 représente la barre de menu située en haut de la fenêtre et composée des boutons suivants :

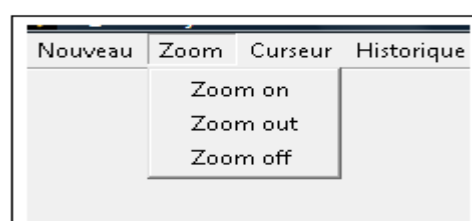


Figure4. 12 Barre de menu

- **Bouton *Nouveau***

Ce bouton permet de lancer un nouveau projet de superrésolution. Le clic sur ce bouton affiche un message de confirmation avant l'initialisation de toutes les variables intermédiaires, images et historiques.

- **Bouton *Zoom*:**

Ce bouton permet d'activer, d'annuler ou de désactiver l'option zoom afin de pouvoir agrandir une zone d'intérêt.

- **Bouton *Curseur*:**

L'activation de cette option permet l'affichage des informations du pixel sélectionné de l'image (coordonnées et valeur).

- **Le bouton *Historique*:**

Exactement comme pour la fenêtre de l'environnement Superrésolution Monoframe, Ce bouton donne la possibilité de choisir entre la sauvegarde et l'affichage d'une image traitée.

IV.4.2 Zone de texte *Historique*

Cette zone s'initialise automatiquement pour chaque nouveau projet, elle contient l'historique de toutes les opérations et les paramètres des traitements effectués pour être enregistrés avec l'image traitée dans le fichier *HISTORIQUE*, la Figure 4.13 représente un exemple du contenu de cette zone

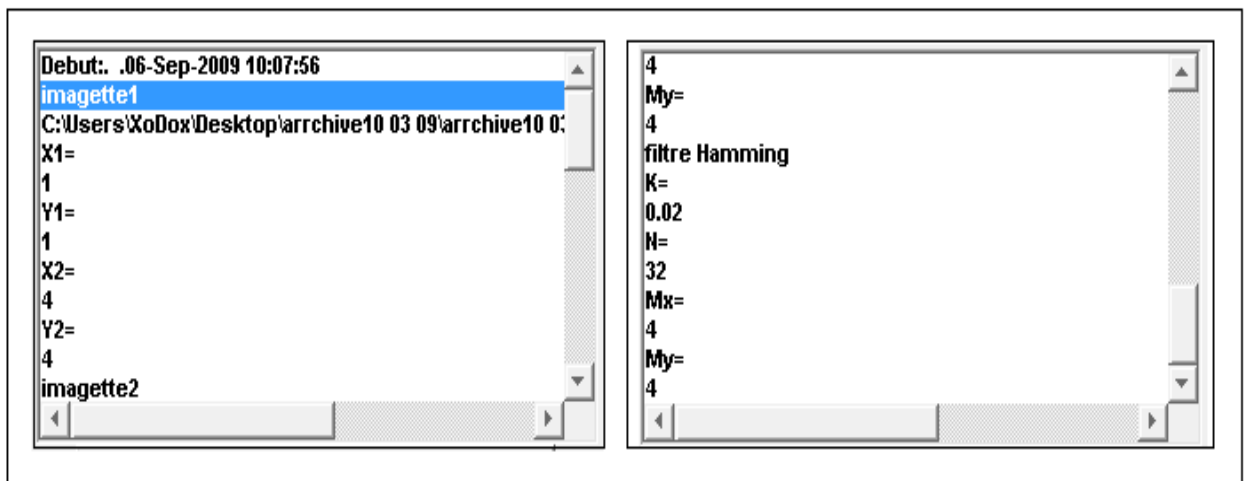


Figure 4. 13 Zone de texte *Historique*.

Cette zone est structurée pour avoir chaque information sur une ligne séparée afin de nous permettre d'avoir toutes les informations concernant :

- **Date et heure de début de traitement:**

Nous trouvons toujours, dans la première ligne de la zone de texte Historique, la date et l'heure du début de traitement sous la forme :

Debut: . . JJ-MOIS-AAAA HH :MM :SS.

- **Informations sur les imagettes utilisées:**

Cette zone contient toutes les informations nécessaires sur les imagettes utilisées telles que le numéro de l'imagette utilisée, l'adresse de son fichier source et l'emplacement de ses pixels sur la matrice image composée *HRp*. La syntaxe proposée pour cela est de la forme :

Imagette_N°imagette ↵
@dresse_imagette ↵
Emplacement sur HRp ↵

- **Informations sur les opérations effectuées:**

Toutes les informations concernant les opérations, les algorithmes, et les paramètres de la superrésolution multiframe sont enregistrées et affichées en ordre dans la zone de texte *Historique* sous la syntaxe suivante :

Nom_opération ↵
paramètres ↵

IV.4.3 Bouton *Ajouter une imagette*

Ce bouton est utilisé pour la création de la matrice *image composée HRp* par l'insertion et le placement des imagettes sur elle.

En cliquant sur ce bouton, une boîte de dialogue «*Lecture d'une imagette pour le traitement*» s'ouvre pour permettre de choisir et lire une imagette de type BMP, JPEG ou DICOM pour l'insérer dans l'*image composée HRp*. L'emplacement des pixels de l'imagette sur *HRp* se fait automatiquement en définissant l'emplacement des deux pixels $p(1,1)$ et $p(2,2)$ de la matrice imagette LR_i sur l'*image composée* : en effet la connaissance de ces deux pixels permet le calcul de la résolution et du pas de déplacement de l'appareil d'acquisition.

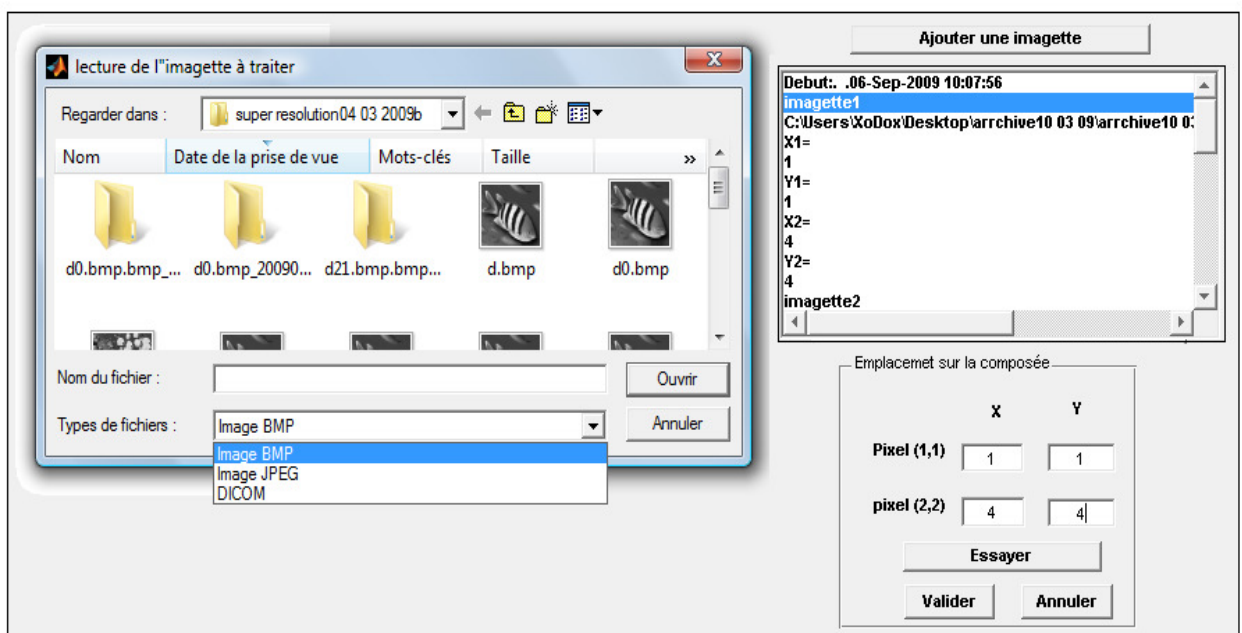


Figure 4. 14 Bouton ajouter une image et son rôle.

La Figure 4.14 est une figure de synthèse qui fait apparaître le bouton « *Ajouter une imagerie* », la boîte de dialogue « *Lecture d'une imagerie pour le traitement* » et le panel de paramétrage « *Emplacement sur la composée* » utilisée à la définition de l'emplacement de $p(1,1)$ et de $p(2,2)$.

En parallèle à l'ajout de l'imagerie LR_i à l'image composée HRp , la matrice binaire $HRmap$ qui indique l'emplacement des pixels sources sur la matrice HRp se modifie. Cette matrice est très importante, comme nous l'avons vu pour l'application des algorithmes de super-résolution multiframe par interpolation polynomiale.

IV.4.4 Images indicatrices

Comme pour l'environnement Super-résolution monoframe, ces espaces servent à l'affichage des différentes variables images utilisées, elles sont composées de :

- **Ecran:**

En occupant les deux tiers de la surface de la fenêtre, l'écran permet d'afficher les différentes variables images disponibles en grand format afin de faciliter l'analyse des résultats obtenus. En effet, il affiche automatiquement le résultat obtenu après chaque opération effectuée comme il peut afficher les images utilisées LR , l'image composée HRp , l'image traitée HR ou l'image validée suivant la demande et le besoin de l'utilisateur.

- **Image indicatrice Imagerie:**

Cet espace permet d'afficher une des images utilisées pour la reconstruction et le traitement selon le besoin, pour cela, nous l'avons doté comme le montre la Figure 4.15, de :

- une zone d'affichage pour afficher l'image sélectionnée
- une *Listbox* qui contient la liste de toutes les images utilisées afin de permettre la sélection de l'image à afficher.
- le bouton *afficher* qui permet d'afficher l'image sélectionnée sur l'écran.
- le bouton *extraire* qui permet d'afficher l'image sélectionnée sur une autre fenêtre.

Afin de minimiser les ressources mémoires utilisées, nous avons opté pour la définition d'une seule variable qu'on utilise pour l'affichage des images ; cette variable détermine et affiche son contenu automatiquement à partir de l'historique et de l'image composée HRp à chaque fois qu'on sélectionne une autre image de la *Listbox*.



Figure 4. 15 Espace d'affichage des imagerie.

- **Image indicatrice *Image composée*:**

Cet espace est utilisé pour l'affichage de l'image *HRp* composée des imagerie sélectionnées. Afin d'améliorer l'affichage de cette image, l'espace d'affichage de l'image composée est doté des deux options suivantes :

- le bouton *afficher* pour afficher l'image composée sur l'écran,
- le bouton *extraire* pour afficher l'image composée sur une autre fenêtre.

La Figure 4.16 montre les objets qui composent l'image indicatrice *image composée*.

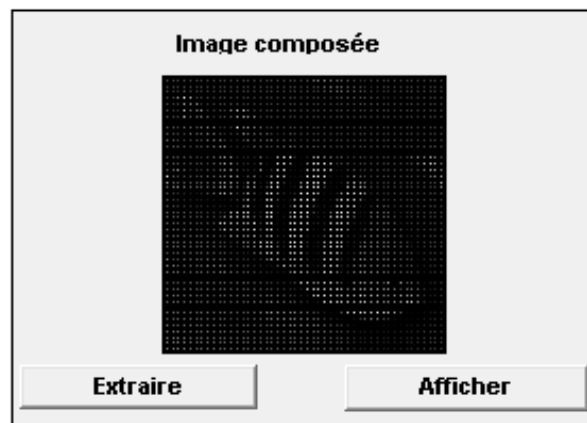


Figure 4. 16 Espace d'affichage de l'image *HRp*.

- **Image indicatrice *Image traitée*:**

L'image affichée dans cet espace est l'image traitée. L'image traitée est une variable temporelle utilisée pour afficher les résultats de l'application des algorithmes de superrésolution multiframe testés sur l'image composée avant de les valider. Les options disponibles à ce niveau sont :

- le bouton *afficher* pour afficher l'image traitée sur l'écran.
- le bouton *extraire* pour afficher l'image traitée sur une autre fenêtre.

- **Image indicatrice *image validée*:**

L'*image validée* est une image traitée prête à être enregistrée d'une façon identique à l'enregistrement de l'*image saved* de l'environnement Superrésolution Monoframe, l'espace d'affichage de cette image comprend les options suivantes:

- le bouton *afficher* pour afficher l'*image validée* sur l'écran,
- le bouton *extraire* pour afficher l'*image validée* sur une autre fenêtre,
- le bouton *sauvegarder le fichier* pour enregistrer l'image validée sous forme d'un fichier BMP ou JPEG ainsi de la création du fichier *HISTORIQUE* qui contient l'*image validée* et son historique.

Nous avons donc résumé le principe de fonctionnement de la fenêtre dédiée à l'environnement Superrésolution Multiframe dans l'organigramme de la Figure 4.17, où nous présentons les variables y'intervenant et les différentes étapes à suivre dans un nouveau projet de superrésolution multiframe.

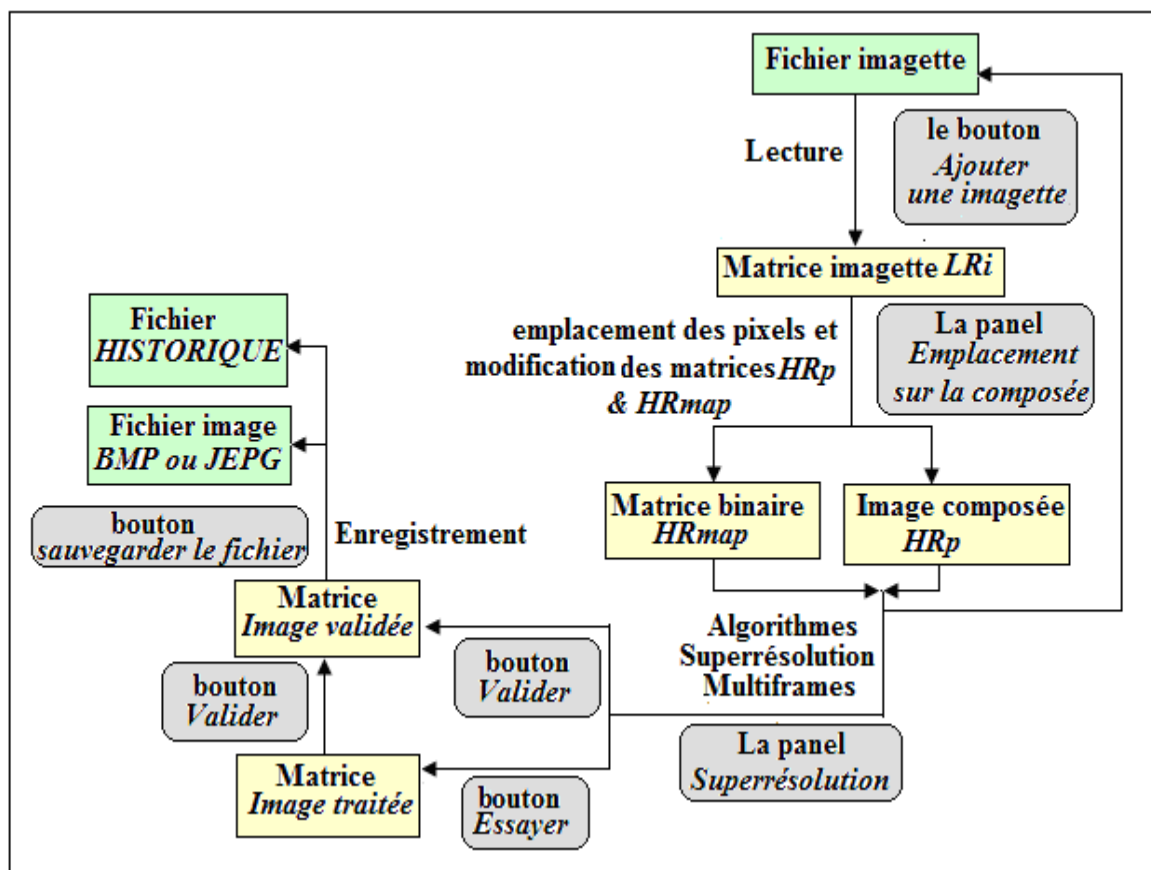


Figure 4. 17 Organigramme fonctionnel de la fenêtre superrésolution multiframe.

IV.4.5 Bouton *Reconstruction*

Ce bouton sert à afficher le panel *Superrésolution* pour choisir l'algorithme de superrésolution multiframe à appliquer.

IV.4.6 Panels de paramétrage

Deux panels sont disponibles sur la fenêtre de l'environnement pour l'insertion des paramètres et le choix de l'algorithme de superrésolution à appliquer :

- **Panel *Emplacement sur la composée*:**

Ce panel s'affiche lors du clic sur le bouton *Ajouter une imagerie* pour déterminer l'emplacement de l'imagerie choisie LRi sur l'image composée HRp en introduisant l'emplacement des deux pixels $p(1,1)$ et $p(2,2)$ de LRi sur la matrice HRp . En effet, la connaissance de l'emplacement de ces deux pixels permet le calcul des emplacements des autres: Soient $P(x_1, y_1)$ et $P(x_2, y_2)$ les pixels de HRp correspondant aux pixels $p(1,1)$ et $p(2,2)$ de la matrice LRi , et soit $P(x_n, y_m)$ le pixel de HRp qui correspond au pixel $p(n,m)$. Les coordonnées de $P(x_n, y_m)$ sont déterminées par la relation:

$$\begin{cases} x_n = (n-1)(x_2 - x_1) + x_1 \\ y_m = (m-1)(y_2 - y_1) + y_1 \end{cases} \Rightarrow \begin{cases} x_n = (n-1)x_2 + (2-n)x_1 \\ y_m = (m-1)y_2 + (2-m)y_1 \end{cases} \quad (4.6)$$

The image shows a dialog box titled "Emplacement sur la composée". It has a table-like structure with two columns labeled "X" and "Y".

	X	Y
Pixel (1,1)	1	1
pixel (2,2)	4	4

Below the table are three buttons: "Essayer", "Valider", and "Annuler".

Figure 4. 18 Panel *Emplacement sur la composée*.

Nous remarquons dans la Figure 4.18 que le panel *Emplacement sur la composée* est composé de:

- Quatre champs de texte pour introduire les coordonnées des pixels de HRp correspondant aux pixels $p(1,1)$ et $p(2,2)$ de LRi
- un bouton *Essayer* pour essayer l'emplacement de l'imagerie LRi sur l'image composée, le résultat est affecté à la matrice *image traitée* et il ne peut pas être exploité pour le reste du traitement sans sa validation.
- un bouton *Valider* qui sert à la validation de l'emplacement de l'imagerie LRi sur la grille de l'image composée HRp afin d'exploiter cette dernière dans le reste du traitement.

- **Panel *Superrésolution*:**

Ce panel s'affiche en cliquant sur le bouton *Reconstruction* pour proposer l'application d'un algorithme de superrésolution multiframe parmi les algorithmes par interpolation polynomiale et les algorithmes par interpolation par filtres de *Whittaker* fenêtrés développés dans le chapitre III.

La Figure 4.19 présente les objets et les fonctions proposés par le panel *Superrésolution*.

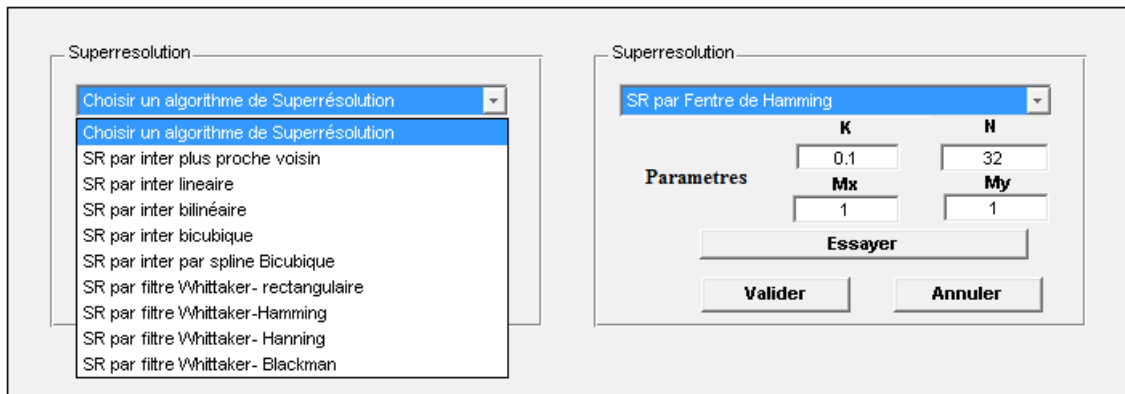


Figure 4. 19 Panel *Superrésolution*.

Les fonctions présentes sur cet environnement pour l'implémentation des algorithmes de la superrésolution par interpolation polynomiale ne nécessitent l'introduction d'aucun paramètre externe. En effet, l'utilisation de la matrice composée HRp et la matrice binaire $HRmap$ est suffisante pour déterminer le diagramme de Voronoï et la triangulation de Delaunay correspondants et pour extrapoler tous les pixels de l'image validée. Ces fonctions sont représentées par :

- La fonction *SR par inter plus proche voisin* :

Qui implémente l'algorithme de la superrésolution par interpolation par le plus proche voisin, cet algorithme est basé sur la division de l'image en cellules de Voronoï et l'extrapolation de la valeur de chaque pixel par rapport au centroïde de la cellule à laquelle il appartient.

Pour la construction du diagramme de Voronoï, nous avons choisi d'utiliser la fonction *Voronoi* proposée par la version 7.0 de *Matlab*. En effet cette fonction utilise un algorithme plus rapide et plus optimisé que les algorithmes basiques proposés dans le chapitre précédent.

- La fonction *SR par inter lineaire* :

Cette fonction est une implémentation de l'algorithme de la superrésolution linéaire qui exploite la triangulation de Delaunay pour la détermination des plans d'interpolation. Nous avons choisi d'utiliser l'algorithme de triangulation de Delaunay par l'exploitation du diagramme de Voronoï : pour cela nous avons calculé le diagramme de Voronoï avec la fonction *Voronoi* de *Matlab* et déterminé la triangulation de Delaunay correspondante comme il est

indiqué dans le chapitre précédent. L'extrapolation des pixels de l'image interpolée est obtenue par l'application de la relation 3.6.

- La fonction *SR par inter bilineaire* :

La fonction *SR par inter bilineaire* utilise l'interpolation bilinéaire pour l'extrapolation des pixels de l'image *HR*. Nous avons choisi de commencer l'extrapolation suivant la direction *X* puis suivant la direction *Y* comme il est indiqué dans le paragraphe III.5.3.

- La fonction *SR par inter bicubique* :

La fonction *SR par inter bicubique* est la fonction qui implémente l'algorithme de la superrésolution multiframe par l'interpolation bicubique développée dans III.5.3, cette interpolation nécessite la connaissance des 16 pixels sources voisins à chaque pixel à extrapoler. C'est pour cela que nous avons choisi d'utiliser la fonction de triangulation de Delaunay *Delaunay* de *Matlab*, qui est une fonction plus optimale et plus rapide que la fonction de triangulation implémentée dans la fonction *SR par inter lineaire* et d'appliquer l'algorithme récurant par niveaux de sélection pour la sélection des 16 voisins nécessaires pour l'extrapolation des pixels en utilisant la relation 3.14.

- La fonction *SR par inter par spline bicubique* :

La fonction *SR par inter par spline bicubique*, comme son nom l'indique, implémente l'algorithme de la superrésolution multiframe qui utilise l'interpolation par spline bicubique. Les quatre pixels sources voisins nécessaires pour l'extrapolation de chaque pixel sont déterminés par l'application des deux premiers niveaux de l'algorithme récurant par niveaux de sélection.

Comme il est indiqué au chapitre III, une fois les quatre pixels voisins déterminés, l'extrapolation est faite en appliquant la relation 3.16. Pour la fonction de triangulation de Delaunay de l'image, nous avons choisi d'utiliser la fonction optimisée *Delaunay*.

Pour les fonctions de superrésolution multiframe par filtrages disponibles, nous devons introduire trois paramètres pour l'extrapolation des pixels : le facteur (M_x, M_y) , la taille du filtre N et le facteur de correction de l'intensité K . Le facteur (M_x, M_y) est équivalent au facteur de suréchantillonnage dans la superrésolution monoframe. Donc par définition, les valeurs de M_x et M_y doivent vérifier :

$$\begin{cases} 1 \leq M_x \leq \max \left(\frac{X_{HRp}}{X_{LRi}} \right)_{i \in \{1, 2, \dots, n\}} \\ 1 \leq M_y \leq \max \left(\frac{Y_{HRp}}{Y_{LRi}} \right)_{i \in \{1, 2, \dots, n\}} \end{cases}$$

Où (X_{HRp}, Y_{HRp}) et (X_{LRi}, Y_{LRi}) représentent les dimensions de l'image HRp et de la $i^{\text{ème}}$ imagerie LRi , et n est le nombre d'images utilisées.

L'environnement Développement et Etudes a spécifié une fenêtre pour le calcul et l'optimisation des paramètres en fonction des images utilisées « pixels sources » et de leur distribution sur l'image HRp .

L'environnement Superrésolution Multiframe utilise quatre fonctions de superrésolution multiframe par filtrage :

- La fonction *inter par filtre Whittaker - rectangulaire* :

Cette fonction utilise l'algorithme de superrésolution par filtrage de Whittaker fenêtré par une fenêtre rectangulaire fpb_{rec} :

$$fpb_{rec} = sinc2d * w \text{ Tel que}$$

$$sinc2d(x, y) = k.MxMy \frac{\sin\left(\frac{\pi.x}{Mx}\right)}{x} \cdot \frac{\sin\left(\frac{\pi.y}{My}\right)}{y}$$

$$\text{et } w(x, y) = \begin{cases} 1 & [-N, N] \times [-N, N] \\ 0 & \text{ailleurs} \end{cases}$$

- La fonction *inter Whittaker - Hamming* :

Cette fonction utilise l'algorithme de superrésolution par filtrage de Whittaker fenêtré par une fenêtre Hamming en filtrant l'image avec le filtre fpb_{ham} défini par :

$$fpb_{ham} = sinc2d * w$$

$$\text{Tel que : } w(x, y) = \left(0.54 - 0.46 \cos\left(\frac{2\pi x}{N}\right)\right) \left(0.54 - 0.46 \cos\left(\frac{2\pi y}{N}\right)\right)$$

- La fonction *inter Whittaker - Hanning* :

De même que pour la fonction précédente, cette fonction exploite le filtrage de Whittaker fenêtré par une fenêtre Hann fpb_{han} . Ce filtre est défini par la réponse impulsionnelle:

$$fpb_{han} = sinc2d * w$$

$$\text{Tel que : } w(x, y) = \left(0.5 - 0.5 \cos\left(\frac{2\pi x}{N}\right)\right) \left(0.5 - 0.5 \cos\left(\frac{2\pi y}{N}\right)\right)$$

- La fonction *interpolation par filtrage de Whittaker avec fenêtre Blackman* :

Cette fonction utilise le filtre $fpb_{blackman}$ de Whittaker fenêtré par une fenêtre de Blackman défini par :

$$fpb_{blackman} = sinc2d * w$$

$$\text{Tel que : } w(x, y) = 0.42 - 0.5 \cos\left(2\pi \frac{x}{N}\right) + 0.08 \cos\left(4\pi \frac{x}{N}\right)$$

IV.4.7 Bouton *Menu principal*

Ce bouton permet l'affichage de la fenêtre *Menu principal* afin d'accéder à un autre environnement.

IV.5 Environnement *Développement et Etudes*

Cet environnement sert à calculer et optimiser les différents paramètres utilisés par les fonctions de superrésolution précédentes, de comparer les images statistiquement comme il permet d'effectuer des traitements basiques sur les images.

La Figure 4.20 représente la fenêtre principale *Développement & Etudes* qui permet l'accès aux autres fenêtres de l'environnement

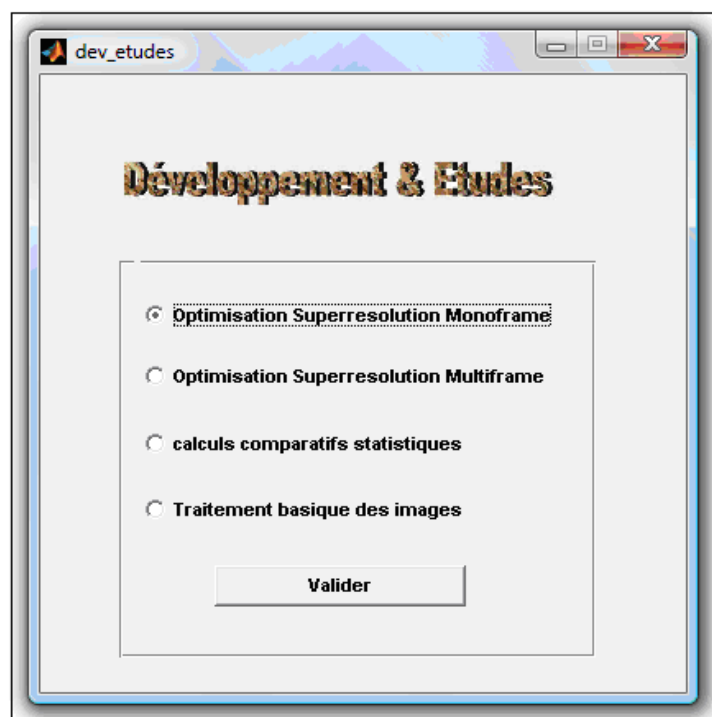


Figure 4. 20 Fenêtre principale *Développement & Etudes*.

L'environnement est doté de quatre autres fenêtres pour assurer son fonctionnement : la fenêtre *Optimisation Superrésolution Monoframe*, la fenêtre *Optimisation Superrésolution Multiframe*, la fenêtre *Calculs comparatifs statistiques* et la fenêtre *Traitement basique des images*.

IV.5.1 Fenêtre *Optimisation superrésolution Monoframe*

Cette fenêtre sert à calculer et à optimiser les différents paramètres de chaque filtre de Whittaker fenêtré utilisé dans les fonctions de la superrésolution monoframe en fonction du facteur de suréchantillonnage et de la nature des images utilisées.

L'algorithme général de fonctionnement de cette fenêtre est basé sur la construction d'une base de données à partir de la lecture d'une ou plusieurs images, de leur souséchantillonnage,

puis du suréchantillonnage des images souséchantillonnées auparavant en utilisant un filtre de Whittaker fenêtré et en faisant varier les valeurs des paramètres. Nous calculons ainsi les taux d'erreurs produites entre les images initiales de la base de données et les images construites par filtrage afin de déterminer les valeurs optimisées des paramètres.

La Figure 4.21 représente la fenêtre *Optimisation Superrésolution Monoframe* et les différents objets qui la constituent.

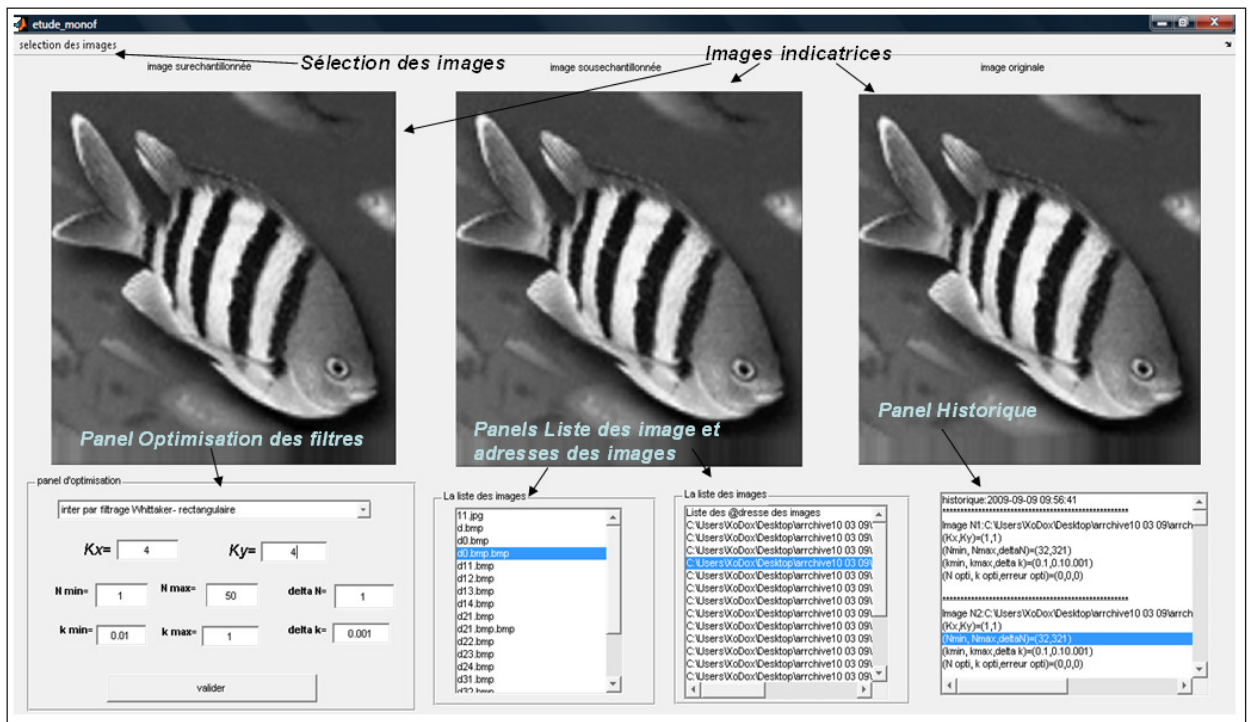


Figure 4. 21 Fenêtre Optimisation superrésolution Monoframe.

- **Bouton *Sélection des images***:

L'action sur ce bouton fait apparaître une boîte de dialogue pour sélectionner une ou plusieurs images de type BMP, JPEG, TIFF ou DICOM pour la construction de la base de données.

Afin d'optimiser les ressources utilisées, la base de données est construite par les adresses des images sélectionnées. La lecture de chaque image se fait lors de son traitement à partir de son adresse.

- **Panel *La liste des images*** :

Ce panel contient la liste des noms de fichiers images sélectionnées pour la construction de la base de données. Le titre de ce panel peut prendre deux valeurs : « *La liste des images* » lorsque les images sélectionnées sont des images BMP, JPEG, TIFF ou « *La liste des images Dicom* » lorsque les images sélectionnées sont des images de type Dicom. Ces valeurs sont utilisées pour déterminer la fonction de lecture des fichiers images.

- **Panel *La liste des @dresses des images*** :

Ce panel contient la liste des adresses des images utilisées pour le calcul des paramètres optimisés des fonctions de superrésolution monoframe par filtrage.

Lors des calculs, l'algorithme d'optimisation lit chaque image de la base de données en utilisant son adresse disponible sur *La liste des @dresses des images* et sélectionne son nom et son adresse automatiquement sur les deux panels pour indiquer l'état d'avancement des opérations de calculs.

- **Images indicatrices** :

Ces images servent à afficher les contenus des différentes variables images utilisées pour chaque image de la base de données :

- L'image indicatrice *Image originale* : utilisée comme image de référence, elle affiche directement l'image lue de la base de données.

- L'image indicatrice *Image souséchantillonnée* : Elle affiche l'image produite par le souséchantillonnage de l'image originale par l'algorithme avec le facteur de souséchantillonnage introduit (K_x, K_y) .

- L'image indicatrice *Image suréchantillonnée optimisée* : Elle affiche l'image suréchantillonnée par filtrage à partir de l'image souséchantillonnée en utilisant le même facteur (K_x, K_y) et les paramètres optimisés calculés du filtre choisi.

- **Panel *Optimisation des filtres*** :

Ce panel est utilisé pour le choix du filtre à optimiser, l'introduction de différents éléments de calcul utilisés par l'algorithme d'optimisation et le lancement de l'algorithme. Ce panel est composé essentiellement de :

- *La liste défilante des filtres de Whittaker fenêtré* :

Cette liste propose de sélectionner un algorithme parmi les quatre algorithmes de superrésolution monoframe par filtrage de Whittaker fenêtré : soit par une fenêtre rectangulaire, soit par une fenêtre de Hamming, une fenêtre de Han ou une fenêtre de Blackman, pour déterminer ses paramètres optimisés.

- *les Zones de texte K_x et K_y* :

Elles sont utilisées pour introduire le facteur (K_x, K_y) du souséchantillonnage et du suréchantillonnage selon les deux directions X et Y.

- *les zones de texte N_{min} , N_{max} et δN* :

Elles servent à l'introduction des valeurs de la taille minimale N_{min} , de la taille maximale N_{max} de la fenêtre utilisée, ainsi que le pas de la variation de cette taille δN .

- les zones de texte $kmin$, $kmax$ et $deltak$:

Utilisées pour déterminer l'intervalle de la variation de k , k étant le facteur de correction de l'intensité limité par la valeur minimale de k $kmin$ et sa valeur maximale $kmax$, le pas de cette variation est donné par $deltak$.

- le bouton *Valider* :

Il sert à lancer l'exécution de l'algorithme d'optimisation.

En premier lieu, pour chaque adresse(i) de la base de données, l'algorithme d'optimisation lit l'image *originale* $\{i\}$ correspondant à cette adresse, il la souséchantillonne avec le facteur (Kx , Ky) en créant l'image *souséchantillonnée* $\{i\}$ et initialise la variable *Erreur_opti* $\{i\}$ en maximisant sa valeur. A la fin de l'algorithme, la variable *Erreur_opti* $\{i\}$ doit correspondre au taux d'erreurs calculé par la comparaison entre l'image *originale* $\{i\}$ et l'image suréchantillonnée avec les paramètres optimaux à partir de l'image *souséchantillonnée* $\{i\}$.

En utilisant le même facteur de souséchantillonnage (Kx , Ky), on suréchantillonne l'image *souséchantillonnée* $\{i\}$ par l'interpolation par filtrage choisie en variant la *taille* du filtre N entre $Nmin$ et $Nmax$ avec un pas de $deltaN$ et le facteur de correction de l'intensité k entre $kmin$ et $kmax$ avec un pas de $deltak$. Nous aurons alors une image *suréchantillonnée* $\{i,N,k\}$ pour chaque couple (N,k). En comparant l'image *originale* $\{i\}$ avec l'image *suréchantillonnée* $\{i,N,k\}$, on calcule le taux d'erreur $e\{N,K\}$, et par la comparaison de ce dernier avec la valeur de *Erreur_opti*, les valeurs de la taille optimisée du filtre N_opti , du facteur de correction optimisé k_opti et de *Erreur_opti* changent :

$$Erreur_opti > e\{N,k\} \Rightarrow \begin{cases} Erreur_opti = e\{N,k\} \\ N_opti = N \\ k_opti = k \end{cases}$$

A la sortie de cette boucle, les valeurs de N_opti et k_opti représentent les paramètres optimisés du filtre calculé pour le facteur de suréchantillonnage (Kx , Ky) en utilisant la $i^{ème}$ image de la base de données.

Pour terminer, l'algorithme affiche l'image *originale* $\{i\}$, l'image *souséchantillonnée* $\{i\}$ et l'image *suréchantillonnée* $\{i,N_opti,k_opti\}$ ainsi que les valeurs de i , N_opti , k_opti , *Erreur_opti*.

La Figure 4.22 représente l'organigramme détaillé de l'algorithme d'optimisation

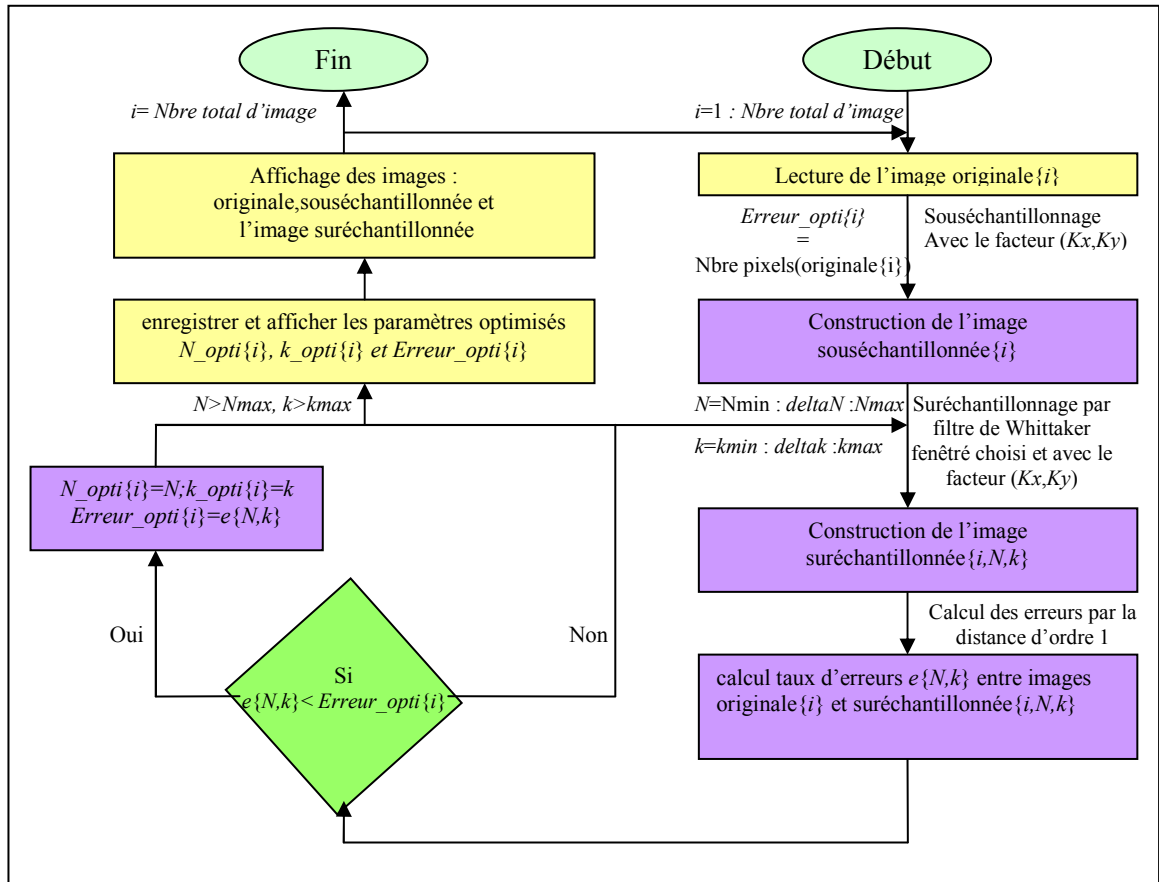


Figure 4. 22 Organigramme détaillé de l'algorithme d'optimisation monoframe.

- **Panel Historique:**

Ce panel affiche l'historique des résultats obtenus par l'algorithme d'optimisation des paramètres du filtre choisi suivant la syntaxe :

- Historique : date et heure de début
- Image Ni : l'adresse de la i^{eme} image
- La valeur de (Kx,Ky)
- La valeur de (Nmin,Nmax,deltaN) de la i^{eme} image
- La valeur de (kmin,kmax,deltak) de la i^{eme} image
- La valeur de (N opti,k opti,erreur opti) de la i^{eme} image

La Figure 4.23 représente un exemple d'affichage sur le panel historique

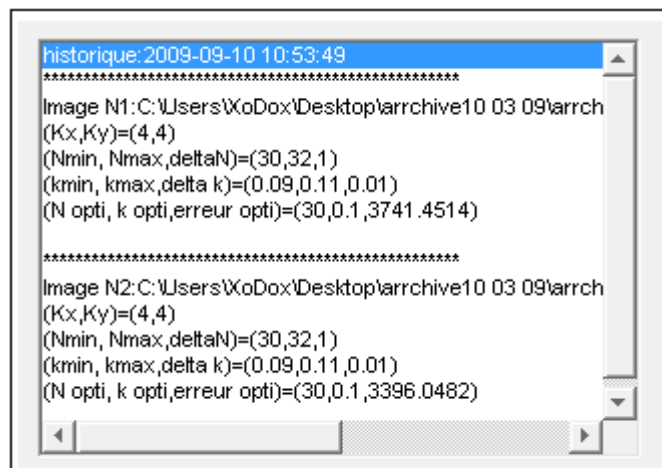


Figure 4. 23 Panel historique

- **Bouton *Menu principale*:**

Ce bouton permet l’affichage de la fenêtre *Menu principale* pour la sélection d’un autre environnement de travail.

IV.5.2 Fenêtre *Optimisation superrésolution Multiframe*

Comme pour la fenêtre précédente, cette fenêtre est désignée pour calculer et optimiser les différents paramètres de chaque filtre de Whittaker fenêtré utilisé dans les fonctions de la superrésolution multiframe en fonction de la distribution et de la nature des imageries utilisées.

L’algorithme général de fonctionnement de cette fenêtre se base sur le choix d’une image de référence dite *image originale*, la création des imageries à partir de cette image, et la construction de l’*image composée HRp* en plaçant chaque imagerie sur une matrice nulle *HRz* ayant les mêmes dimensions que l’*image originale*. En appliquant un algorithme de superrésolution multiframe par filtre de Whittaker fenêtré, nous varions ses paramètres afin d’optimiser le taux d’erreurs existant entre l’*image originale* et les images construites de haute résolution *HR*.

La Figure 4.24 représente la fenêtre *Optimisation superrésolution Multiframe* ainsi que les différents objets qui la composent :

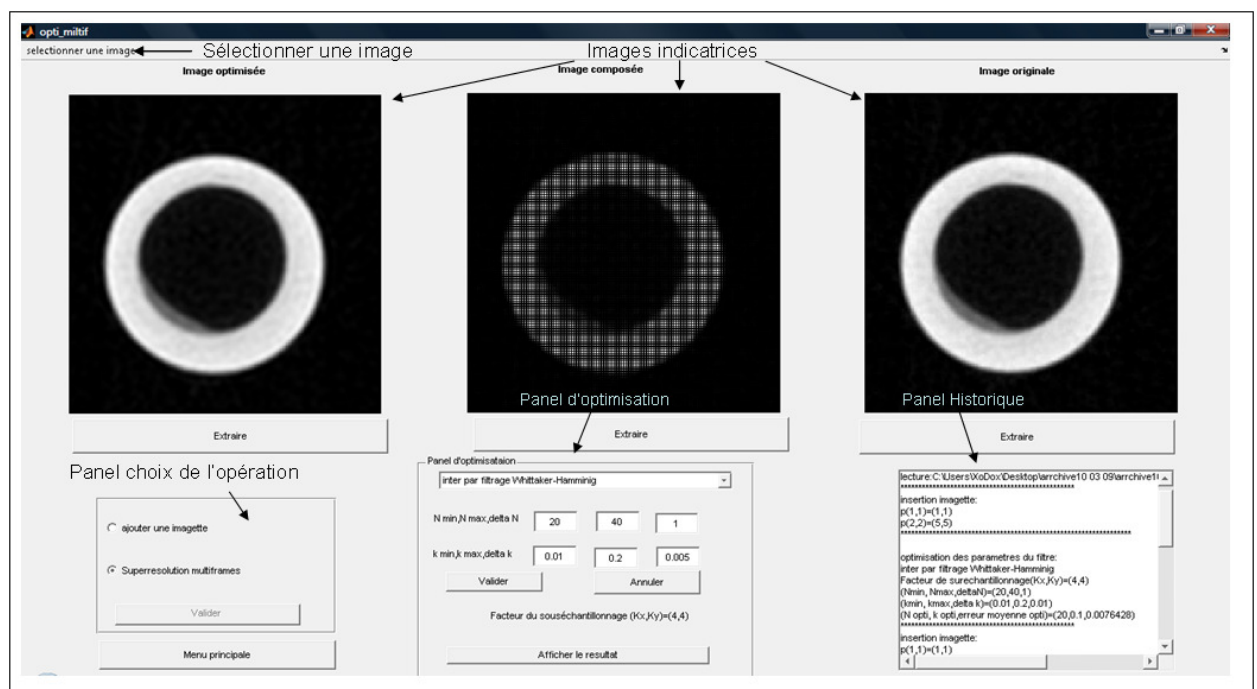


Figure 4. 24 Fenêtre *Optimisation Superrésolution Multiframe*.

- **Bouton *Sélection des images*:**

L’action sur ce bouton fait apparaître une boîte de dialogue pour la sélection et la lecture d’une image de référence « *image originale* » de type BMP, JPEG, TIFF ou DICOM.

En conséquence, nous initialisons l'*image composée HRp* comme une matrice nulle de la même dimension que l'image originale.

- **Images indicatrices :**

Ces images servent à l'affichage des différentes images utilisées, chaque image indicatrice est dotée d'un bouton *Extraire* pour afficher son image sur une autre figure afin de faciliter la comparaison entre elles. Trois images indicatrices sont disponibles sur la fenêtre :

- L'image indicatrice *Image originale* : qui affiche l'image de référence *image originale* lue,
- L'image indicatrice *Image composée* : qui affiche l'image composée des différentes imagettes construites à partir du souséchantillonnage de l'*image originale*,
- L'image indicatrice *Image optimisée* : qui affiche le résultat obtenu par l'application de l'algorithme de superrésolution multiframe par filtrage en utilisant les paramètres optimisés.

- **Panel Choix de l'opération:**

Ce panel permet le choix de l'opération à effectuer, deux opérations principales sont proposées :

- L'opération *Ajouter une imagette* : permet l'apparition du panel *définition de l'imagette*.
- L'opération *Optimiser l'algorithme de superrésolution multiframe* : permet l'affichage du *Panel d'optimisation*.

- **Panel Définition de l'imagette:**

La Figure 4.25 représente les composants de ce panel utilisé pour la création d'une imagette *LR* à partir de l'*image originale* et de l'*image composée HRp* en plaçant chaque pixel de *LR* sur sa position sur *HRp* comme il est indiqué dans le paragraphe III.3.2.

The image shows a software panel titled "définition de l'imagette". It has a light gray background and a thin border. Inside, there are two columns of input fields. The first row is labeled "Facteur (Kx,Ky)" and has two text boxes, both containing the number "3". The second row is labeled "Position p(1,1)" and has two text boxes, both containing the number "1". The third row is labeled "Position p(2,2)" and has two text boxes, both containing the number "4". At the bottom of the panel, there are two buttons: "Valider" on the left and "Annuler" on the right.

Figure 4. 25 Panel définition de l'imagette.

En introduisant les positions des deux pixels $p(1,1)$ et $p(2,2)$ de l'imagette *LR* sur l'*image composée HRp*, l'algorithme calcule le facteur de souséchantillonnage (Kx, Ky) pour l'imagette,

souséchantillonne l'*image originale* avec ce facteur en formant LR puis place LR sur l'image composée HRp et affiche le résultat sur l'image indicatrice *Image composée*.

- **Panel d'optimisation:**

Ce panel permet la définition du filtre à optimiser, l'introduction des différents éléments de calcul utilisés par l'algorithme d'optimisation et le lancement de l'algorithme. Les objets disponibles sur le panel sont:

- La liste défilante des filtres Whittaker fenêtrés :

Cette liste sert à choisir un algorithme parmi les quatre algorithmes de superrésolution multiframe par filtrage de Whittaker fenêtré : soit par une fenêtre rectangulaire, soit par une fenêtre de Hamming, une fenêtre de Han ou une fenêtre de Blackman, pour déterminer ses paramètres optimisés.

- Les zones de texte $Nmin$, $Nmax$ et $deltaN$

Elles servent à l'introduction des valeurs de la taille minimale $Nmin$ et la taille maximale $Nmax$ de la fenêtre utilisée, ainsi que le pas de la variation de cette taille $deltaN$.

- Les zones de texte $kmin$, $kmax$ et $deltak$

Elles servent à déterminer l'intervalle de la variation de k , k étant le facteur de correction de l'intensité limité par la valeur minimale de k $kmin$ et sa valeur maximale $kmax$. Le pas de cette variation est déterminé par la variable $deltak$.

- le bouton *Valider*

Le bouton *Valider* sert à lancer l'exécution de l'algorithme d'optimisation. Ce dernier commence par la maximisation de la valeur de la variable $Erreur_opti$ correspondant au taux d'erreur calculé par la comparaison entre l'image originale $\{i\}$ et l'image composée filtrée avec le filtre de Whittaker fenêtré optimisé obtenu par l'utilisation des paramètres optimaux calculés (le facteur de suréchantillonnage optimisé (Kx_opti, Ky_opti) , la taille optimisé du filtre $Nopti$ et le coefficient de correction de l'intensité $kopti$).

En variant la valeur du facteur de suréchantillonnage (Kx, Ky) entre 1 et le maximum des facteurs de souséchantillonnage des imgettes LR avec un pas de 0.1, la taille du filtre N entre $Nmin$ et $Nmax$ avec un pas de $deltaN$ et le facteur de correction de l'intensité k entre $kNmin$ et $kNmax$ avec un pas de $deltak$, on filtre l'*image composée* par le filtre de Whittaker choisi. Puis, en comparant l'image résultante de chaque filtrage $HR\{(Kx,Ky),N,k\}$ avec l'*image originale*, nous calculons le taux d'erreurs produit $e\{(Kx,Ky),N,k\}$. Si ce taux d'erreur est supérieur au contenu de la variable $Erreur_opti$ on aura :

$$Erreur_opti > e\{N,k\} \Rightarrow \begin{cases} Erreur_opti = e\{N,k\} \\ N_opti = N \\ k_opti = k \\ (Kx_opti, Ky_opti) = (Kx, Ky) \end{cases}$$

A la sortie de cette boucle, les contenus des variables (Kx_opti, Ky_opti) , N_opti et k_opti représentent les valeurs optimisées du filtre choisi en utilisant les imagerettes LR_i .

A la fin de l'algorithme, on affiche les paramètres utilisés et les paramètres optimisés dans le panel *historique* et l'image calculée par le filtrage de l'*image composée* par le filtre optimal dans l'image indicatrice *Image optimisée*.

La Figure 4.22 représente l'organigramme détaillé de l'algorithme d'optimisation multiframe suivi.

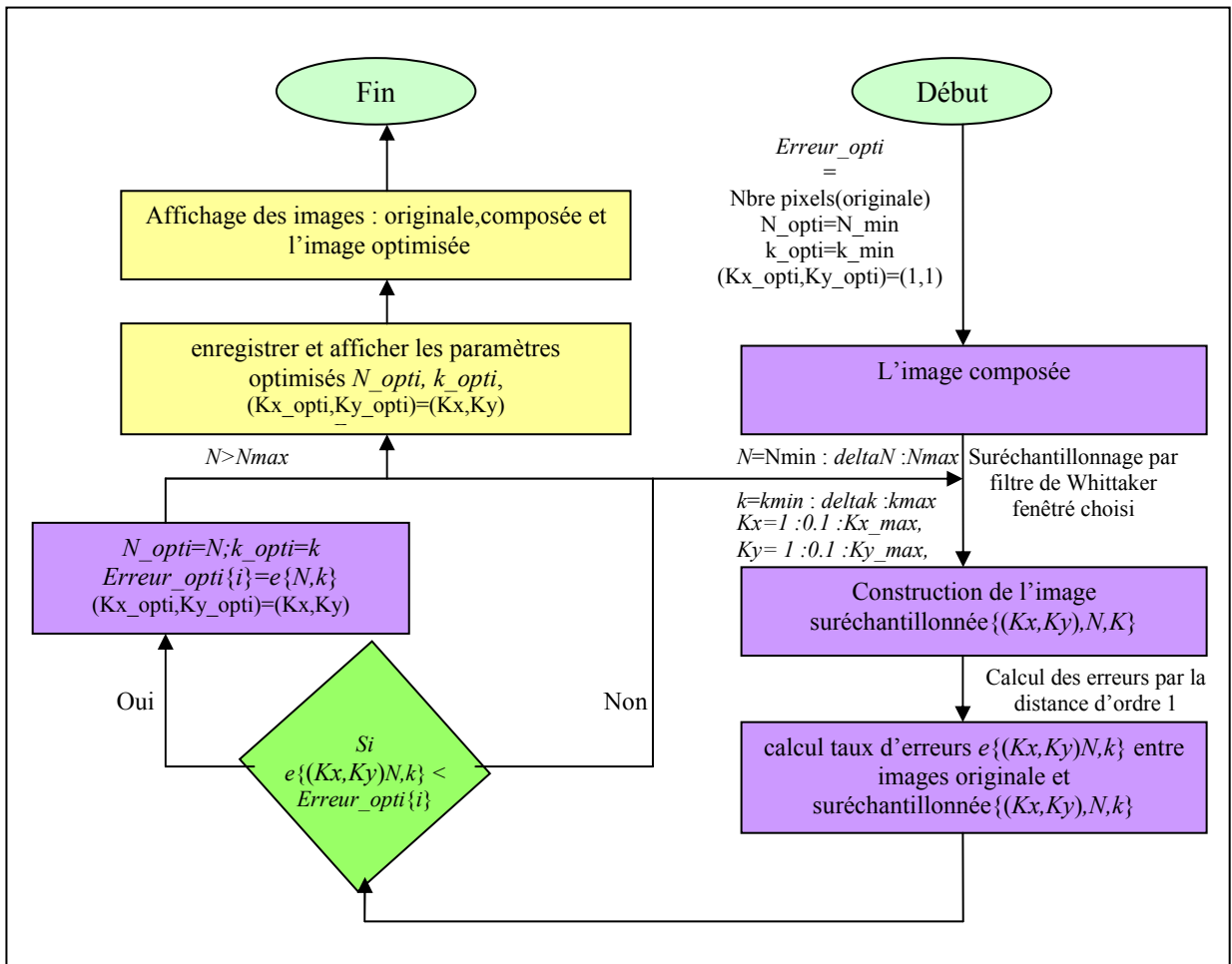


Figure 4. 26 Organigramme détaillé de l'algorithme d'optimisation multiframe.

- **Panel Historique:**

Il contient l'historique des paramètres insérés et les résultats obtenus des différentes opérations effectuées suivant la syntaxe :


```

Historique : date et heure de début
lecture: l'adresse de la l'image originale
*****
Insertion imagette
P(1,1)= position du pixel p(1,1)de l'imagette sur HRp
P(2,2)= position du pixel de l'imagette sur HRp
*****
La valeur de (Nmin,Nmax,deltaN)
La valeur de (kmin,kmax,deltak)
La valeur de ((Kx_opti,Ky_opti),N_opti,k_opti,erreur_opti)
    
```

Le résultat de l'optimisation peut être représenté sur une autre Figure *Résultat*. La Figure 4.27 représente un exemple d'affichage des paramètres initiaux et des résultats de l'optimisation sur le panel *historique* et la fenêtre *Résultat*.

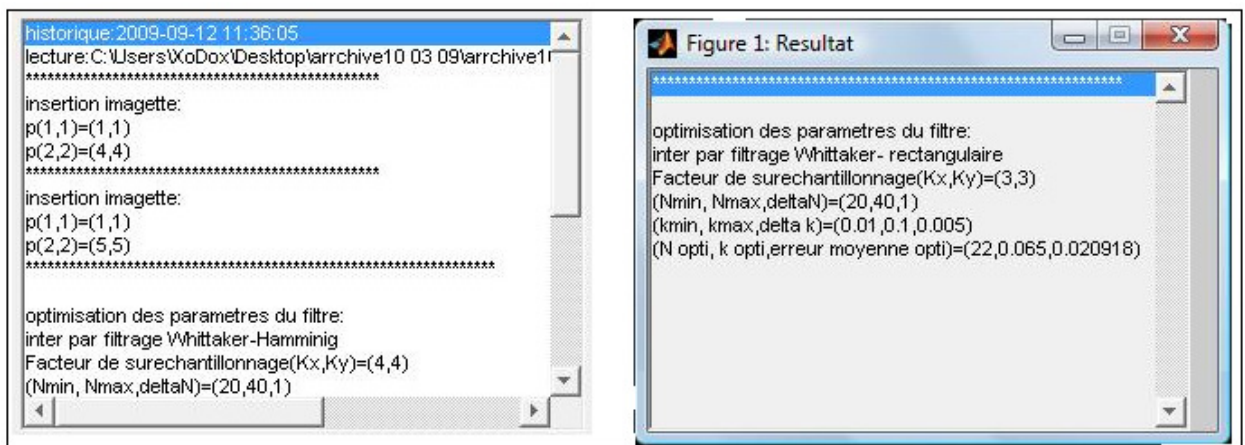


Figure 4. 27 Exemple du panel historique et figure Résultat.

- **Bouton Menu principal:**

Ce bouton permet l'affichage de la fenêtre *Menu principal* pour la sélection d'un autre environnement de travail.

IV.5.3 Fenêtre *Calculs comparatifs statistiques*

Cette fenêtre, utilisée pour la comparaison entre deux images, représente un outil très important pour la détection des différences visuelles et statistiques entre les images. Elle peut être très utile pour déterminer l'algorithme de superrésolution le plus fiable en comparant une image choisie comme image de référence avec l'image obtenue par son souséchantillonnage suivi de son suréchantillonnage.

L'algorithme général du fonctionnement de cette fenêtre est basé sur le choix des deux images, la sélection de la partie à comparer de chaque image puis le lancement de la comparaison entre les deux parties sélectionnées.

La Figure 4.28 représente les différents objets composant cette fenêtre, nous remarquons qu'elle se compose de :

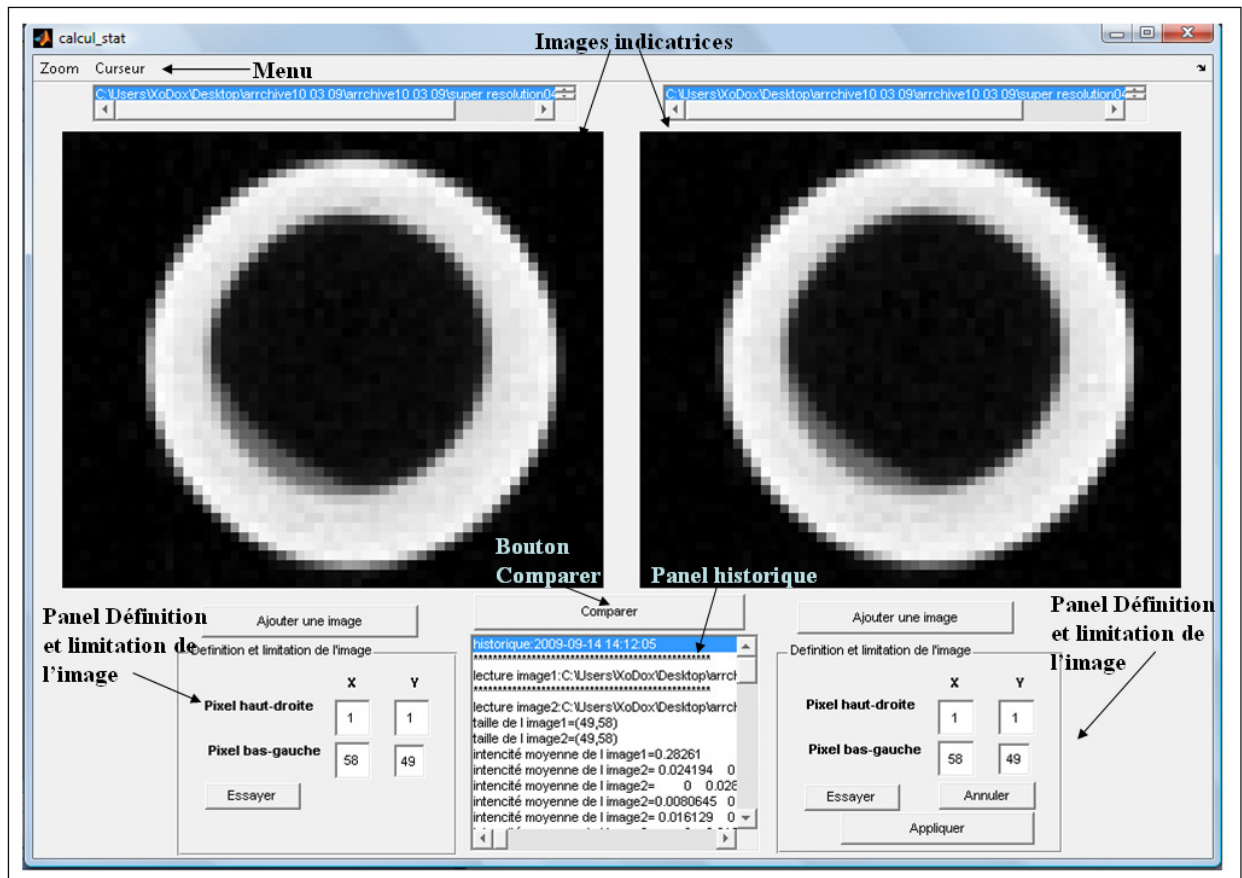


Figure 4. 28 Fenêtre calculs comparatifs statistiques.

- **Barre de Menu:**

La barre de *Menu* est Composée de trois boutons :

- Bouton *Zoom* : Il est utilisé pour l'agrandissement par une interpolation polynomiale d'ordre zéro de la zone sélectionnée, il présente trois options : *Zoom on* pour activer la sélection de la zone à zoomer, *Zoom off* pour désactiver l'option précédente, et l'option *Zoom out* pour afficher l'image sans agrandissement.

- Bouton *Curseur* : il permet d'afficher les coordonnées et la valeur du pixel sélectionné.

- Bouton *Menu* principal : il permet d'afficher la fenêtre principale.

- **Les deux boutons *Ajouter une image*:**

Ces deux boutons servent à sélectionner les deux images à comparer. L'action sur l'un de ces boutons affiche une boîte de dialogue pour la lecture d'un fichier image de type BMP, JPEG, TIFF ou DICOM. L'image sélectionnée sera affichée sur l'image indicatrice située au dessus du bouton utilisé.

- **Images indicatrices:**

La fenêtre est dotée de deux images indicatrices pour afficher les images choisies pour la comparaison. L'adresse du fichier image affichée est mentionnée au dessus de l'image indicatrice.

- **Les panels *limitation et définition de l'image* :**

La fenêtre contient deux panels *limitation et définition de l'image* : un panel pour chacune des images sélectionnées pour la comparaison. Le rôle de chaque panel est donc de définir la zone d'intérêt de l'image appropriée.

Un panel *limitation et définition de l'image* est composé de :

- Des zones textes pour la définition de la zone d'intérêt : la zone d'intérêt est définie comme étant une partie rectangulaire de l'image, la définition de cette zone se fait par l'introduction des coordonnées des pixels situés aux extrémités du rectangle dans les zones de texte appropriées.

- Bouton *Essayer* utilisé pour le calcul et l'affichage de la partie d'intérêt sur l'image indicatrice. Dans ce cas, l'image calculée en actionnant ce bouton est une image d'essai qui ne peut être comparée avec l'image affichée sur l'autre image indicatrice.

- Bouton *Valider* utilisé pour valider l'image obtenue par l'action sur le bouton *Essayer*, une fois que l'image sélectionnée est validée, elle peut être utilisée pour la comparaison.

- Bouton *Annuler* pour annuler la sélection.

- **Bouton *Comparer* :**

Ce bouton sert à lancer l'algorithme de comparaison détaillé sur l'organigramme de la Figure 4.29.

Au début, l'algorithme commence par la comparaison entre les dimensions des deux images : en fait, on ne peut pas comparer deux matrices de dimensions différentes. Si c'est le cas, l'algorithme affiche une boîte de dialogue pour attirer l'attention de l'utilisateur sur le problème en lui proposant de continuer ou d'annuler la comparaison. Si l'utilisateur choisit de continuer, l'algorithme remplace les deux images par les images réduites sur une dimension égale au minimum de leurs dimensions initiales.

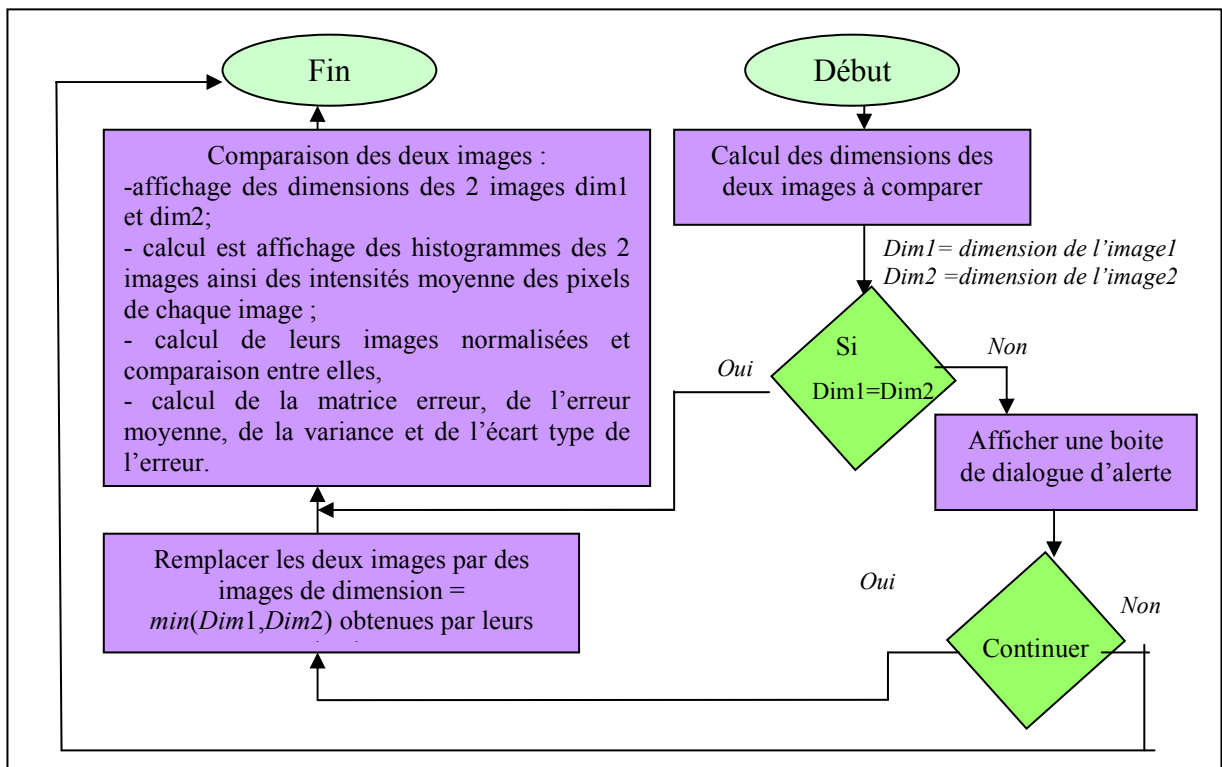


Figure 4. 29 Organigramme détaillé de l'algorithme de comparaison.

Les résultats de la comparaison sont affichés sur trois autres fenêtres créées automatiquement par l'algorithme de comparaison qui sont :

- Fenêtre *Histogrammes* qui affiche les histogrammes appropriés à chaque image ;
- Fenêtre *Images* qui affiche quatre images, les deux premières correspondant aux images normalisées des images comparées, la troisième image correspondant à l'image erreur entre les deux images comparées qui est égale à la valeur absolue de la différence entre les deux images. La dernière image représente l'image erreur entre les deux images normalisées affichées dans les deux premières images de la fenêtre, elle est calculée par la valeur absolue de la différence entre les deux images normalisées.

La normalisation des images se fait par la multiplication de l'image par l'inverse de la valeur de l'intensité maximale de cette image. Le but de cette opération est d'éliminer la différence linéaire qui peut exister entre les intensités de deux images différentes : en effet, si $image1 = \alpha \cdot image2$ avec α une constante réelle, la normalisation de ces deux images donne la même image.

- Fenêtre *Résultats* affiche les résultats des calculs statistiques : elle affiche la dimension initiale de chaque image, l'intensité moyenne par pixel calculée pour chacune des images,

l'erreur moyenne par pixel qui existe entre les deux images ainsi que la variance et l'écart type calculés pour la matrice erreur.

La Figure 4.30 représente les trois fenêtres créées automatiquement par l'algorithme de comparaison pour afficher les résultats de la comparaison.

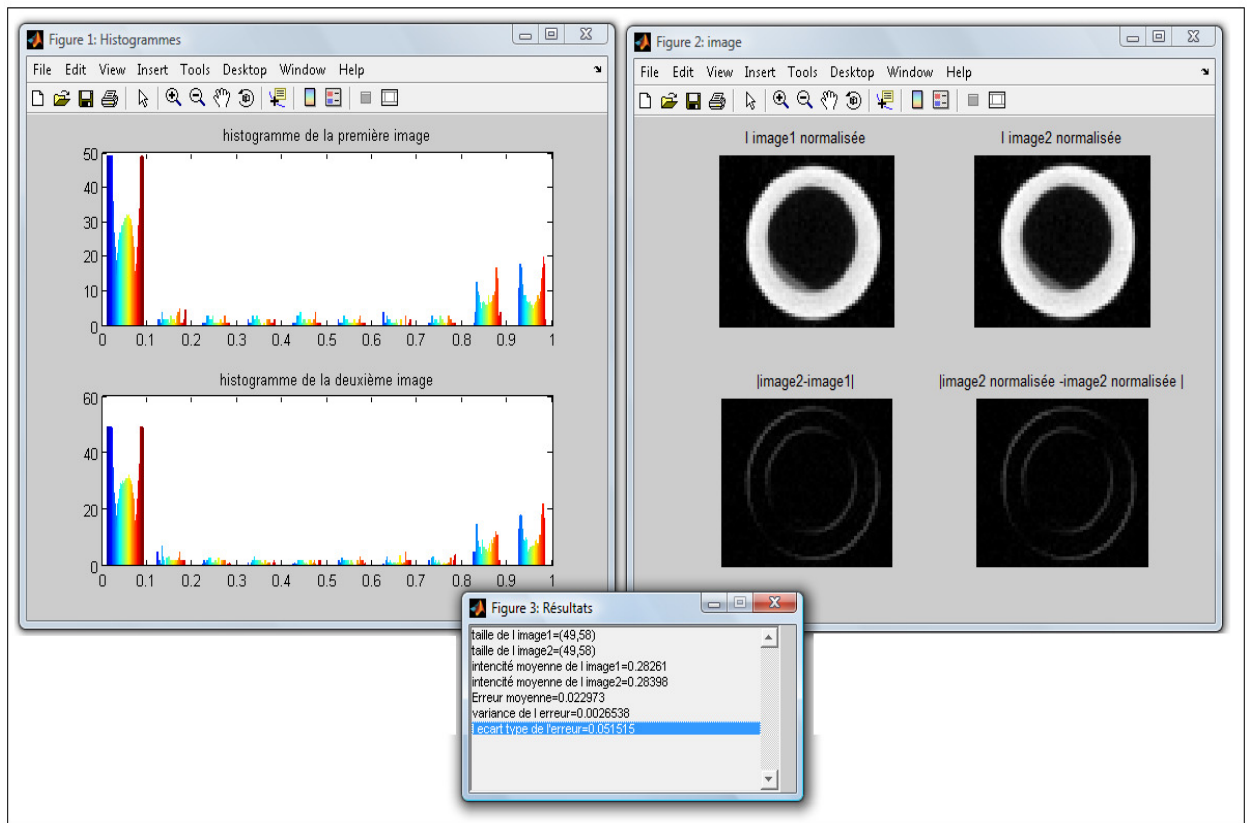


Figure 4. 30 Les trois fenêtres créées par l'algorithme de comparaison pour afficher les résultats.

- **Panel Historique :**

Ce panel affiche l'historique des opérations de lecture et de limitation des images comparées ainsi que les résultats de la comparaison, la syntaxe utilisée est de la forme :

```

Historique : date et heure de début
*****
Image N : l'adresse et le numéro de l'image comparée
*****
Limitation de l'image N° : (x1,y1)=(cordonnée de p(1,1)) ; (x1,y1)=(cordonnée de p(2,2))
*****
Taille de l'image 1 :(sur l'axe X, sur l'axe Y)
Taille de l'image 2 :(sur l'axe X, sur l'axe Y)
Intensité moyenne de l'image 1
Intensité moyenne de l'image 2
Erreur moyenne
Variance de l'erreur
Ecart type
*****
    
```

IV.5.4 Fenêtre *Traitement basique des images*

La fenêtre Traitement basique des images représente la dernière fenêtre développée pour cet environnement, elle représente un outil complémentaire important pour notre application en présentant les principales opérations basiques du traitement des images qui nous permettent le traitement, la modification, l'amélioration et l'analyse des images avant et après l'application des algorithmes de superrésolution.

La Figure 4.31 représente les différents objets qui composent cette fenêtre.

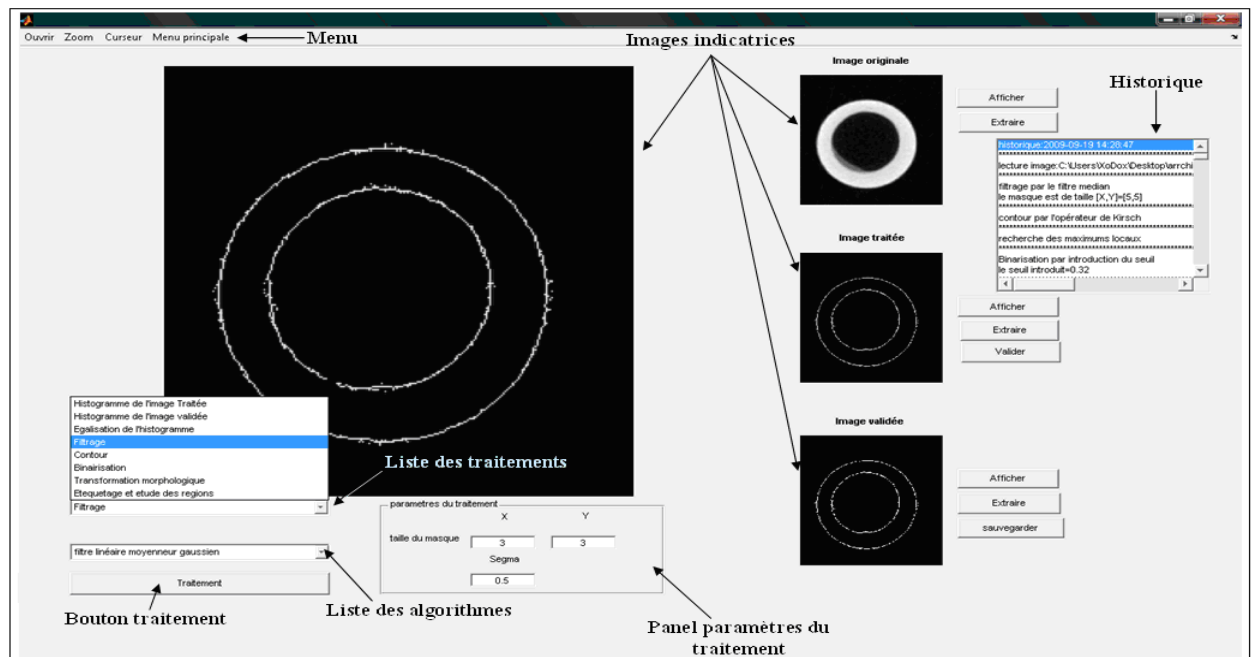


Figure 4. 31 Fenêtre Traitements basiques des images.

- **Barre de Menu:**

La barre de Menu présente quatre boutons d'options :

- Bouton *Ouvrir* qui affiche une boîte de dialogue permettant le choix d'une image de type BMP, JPEG, TIFF ou Dicom pour le traitement, cette image est conservée dans la variable *image originale*.
- Bouton *Zoom* avec ses trois options : *Zoom on*, *Zoom off* et *Zoom out* permet l'activation, la désactivation ou l'annulation de l'agrandissement des images.
- Bouton *Curseur* sert à afficher les coordonnées et la valeur du pixel sélectionné.
- Bouton *Menu principale* est utilisé pour afficher la fenêtre *Menu principale* pour la sélection d'un autre environnement de travail.

- **Images indicatrices:**

Cette fenêtre contient quatre images indicatrices :

- l'*Ecran* : qui occupe la moitié de la surface de la fenêtre, est utilisé pour afficher les images en grand format.

- L'image indicatrice *image originale* :

Elle sert à afficher l'image lue par le bouton *Ouvrir* sans modification, cette image est équipée de deux boutons :

- ♦ Bouton *Afficher* pour afficher l'image *originale* sur l'écran,
- ♦ Bouton *Extraire* pour afficher l'image sur une autre fenêtre.

- L'image indicatrice *image traitée* :

Cette image indicatrice est utilisée pour afficher l'image résultante du traitement de l'image *Validée* par les algorithmes présentés dans cette fenêtre. Elle se compose de trois boutons :

- ♦ Bouton *Afficher* pour afficher l'*image traitée* sur l'écran,
- ♦ Bouton *Extraire* pour afficher l'*image traitée* sur une autre fenêtre,
- ♦ Bouton *Valider* qui sert à remplacer le contenu de l'*image Validée* par le contenu de l'*image traitée* afin d'utiliser le résultat obtenu dans une autre opération de traitement.

- L'image indicatrice *image Validée* :

Qui contient l'image utilisée dans le traitement. Elle est dotée de trois boutons :

- ♦ Bouton *Afficher* utilisé pour afficher l'image *Validée* sur l'écran,
- ♦ Boutons *Extraire* utilisé pour afficher l'image *Validée* sur une autre fenêtre,
- ♦ Bouton *Sauvegarder* pour l'enregistrement de l'image *Validée* sur un fichier BMP ou JPEG ainsi que de son historique dans un fichier *Historique.mat*.

- **Zone de traitement:**

Cette zone est composée de la *liste des traitements*, de la *liste des algorithmes*, du *panel des paramètres* et du bouton *traitement* :

- La *liste des traitements* contient une liste des opérations de traitement offertes par cette fenêtre,
- La *liste des algorithmes* affiche les algorithmes disponibles pour chaque opération choisie,
- Le *panel des paramètres* permet l'introduction des différents paramètres nécessaires pour chaque algorithme
- Le bouton *Traitement* est utilisé pour lancer l'algorithme de traitement choisi.

Lors du choix d'une nouvelle image pour le traitement, l'image choisie est enregistrée dans la variable *image originale* et les autres variables images (*image traitée* et *image validée*) s'initialisent à une valeur égale au contenu de l'*image originale*.

En cliquant sur le bouton *Traitement*, l'algorithme de traitement choisi sera appliqué sur l'image *validée* et le résultat sera affecté à l'image indicatrice de l'*image traitée*, et en cliquant

sur le bouton *Valider* le contenu de l'*image traitée* remplace le contenu de l'*image validée* pour être utilisé dans la prochaine opération du traitement.

Les opérations et les algorithmes de traitement disponibles dans cette fenêtre sont :

- *Affichage de l'histogramme de l'image traitée*

Cette opération permet l'affichage de l'histogramme de l'image traitée sur une autre fenêtre.

- *Affichage de l'histogramme de l'image Validée*

Cette opération permet l'affichage de l'histogramme de l'image validée sur une autre fenêtre, l'affichage des histogrammes est très utile pour la comparaison, le jugement de l'utilité de l'égalisation de l'histogramme et pour le choix du seuil de binarisation de l'image.

La Figure 4.31 représente un exemple sur les fenêtres d'affichage des histogrammes de l'*image traitée* et de l'*image validée*.

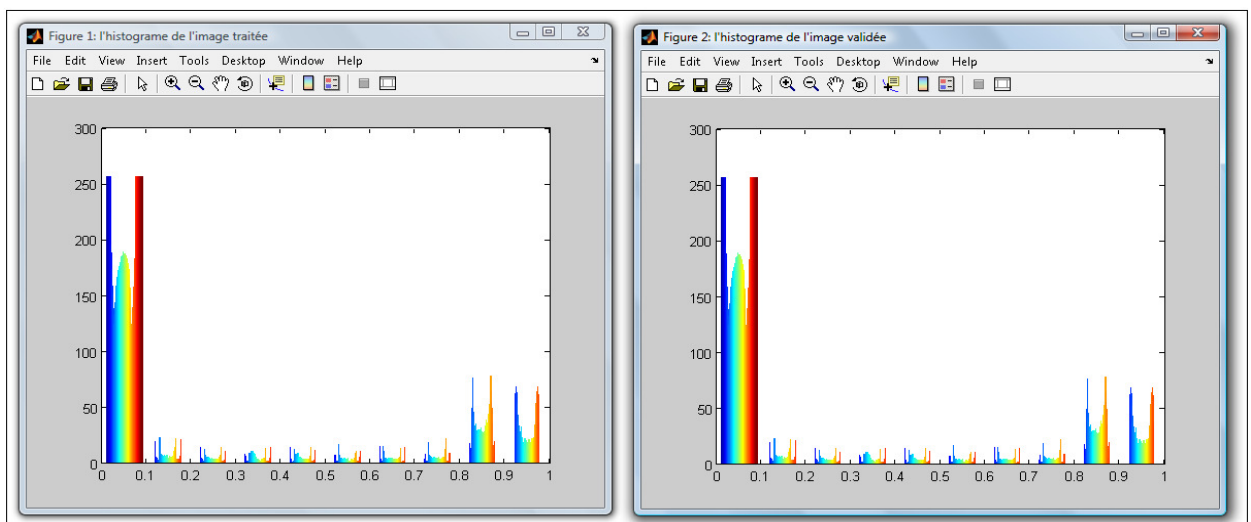


Figure 4. 32 Histogrammes de l'*image traitée* et de l'*image validée*.

- *Egalisation de l'histogramme*

L'égalisation de l'histogramme revient à harmoniser la répartition des niveaux de luminosité de l'image en essayant d'avoir le même nombre de pixels pour chaque niveau de gris. Pour cela, Nous avons utilisé la fonction prédéfinie de Matlab *histeq*.

La Figure 4.33 représente une comparaison entre une image avant et après l'égalisation de l'histogramme par la fenêtre *Traitement basique des images*.

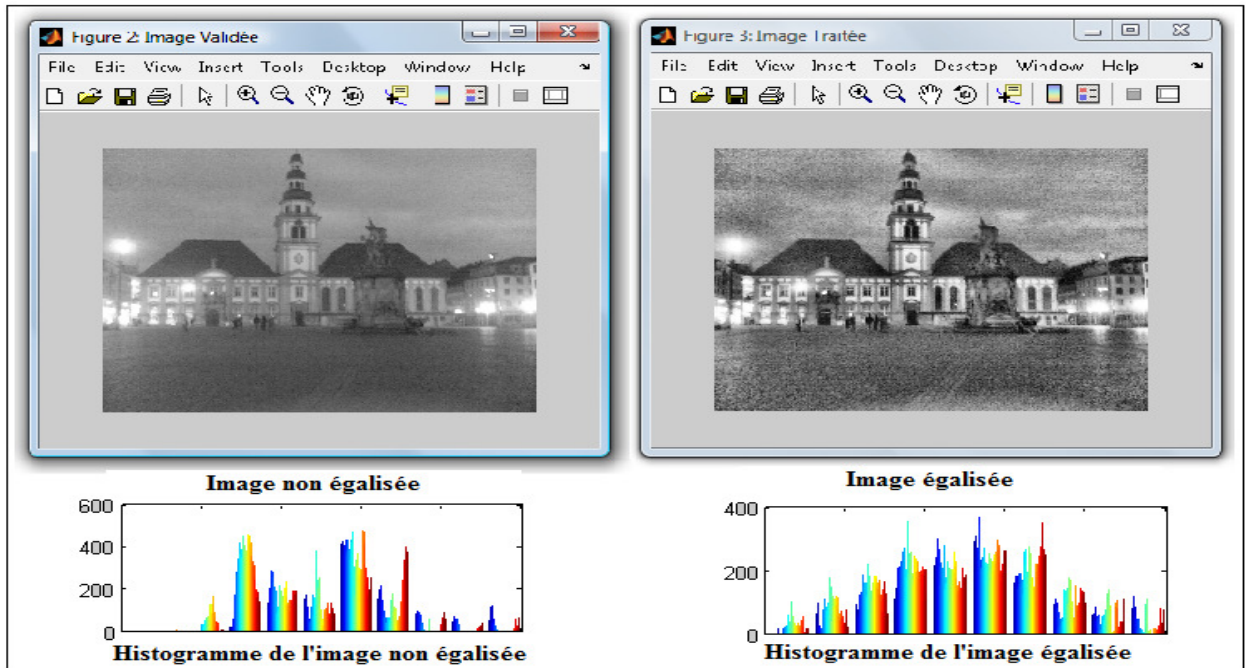


Figure 4. 33 Comparaison entre une image avant et après l'égalisation de l'histogramme.

- Filtrage de l'image traitée

Cette fenêtre offre la possibilité de filtrer l'image avec des filtres de bruit et de lissage: le filtre médian, le filtre rectangulaire, le filtre conique et le filtre moyenneur gaussien.

- ♦ Le *filtre médian* est un filtre non linéaire qui affecte au pixel la valeur médiane existant entre sa valeur et les valeurs de ses voisins définies par le masque utilisé. Cet algorithme nécessite l'introduction de la taille du masque à utiliser pour le filtrage.
- ♦ Le *filtre rectangulaire* est un filtre linéaire de lissage. Toutes les composantes de son masque sont égales à 1. Donc, pour utiliser ce filtre nous devons définir les dimensions du masque.
- ♦ Le *filtre conique* est aussi un filtre linéaire de lissage ayant un masque h défini comme suit :

$$h = \frac{1}{25} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 1 & 2 & 5 & 2 & 1 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

- ♦ Le *filtre de lissage moyenneur gaussien* est un autre filtre linéaire défini par son masque h de dimension $[N,K]$ tel que:

$$\forall x \in \mathfrak{R}^*, \forall y \in \mathfrak{R}^*, \sigma \in \mathfrak{R}^* : (x, y) \leq (N, K) \Rightarrow h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right).$$

L'utilisation de ce filtre nécessite l'introduction des dimensions du filtre et la valeur du σ .

- Détection du contour de l'image traitée :

Pour cela, nous avons proposé 5 opérateurs de calcul du gradient et un algorithme de localisation des crêtes (extremums locaux) dans une image. Le contour d'une image peut être déterminé en localisant les extremums d'une image *traitée* avec un opérateur gradient.

Le module du gradient de l'image traitée $I_{traitée}$ est défini par la relation

$$\left\| \overrightarrow{Grad}(I_{traitée}) \right\| = \sqrt{\left(\frac{\delta I_{traitée}}{\delta x} \right)^2 + \left(\frac{\delta I_{traitée}}{\delta y} \right)^2}$$

Où $\left(\frac{\delta I_{traitée}}{\delta x} \right)$ et $\left(\frac{\delta I_{traitée}}{\delta y} \right)$ représentent les dérivées de l'image traitée $I_{traitée}$ par rapport aux axes X et Y calculées par le filtrage de l'image,

Quatre opérateurs de calcul des dérivées et un opérateur de calcul direct du gradient sont proposés dans cette application :

♦ L'opérateur de dérivation simple :

Cet opérateur définit $\frac{\delta I_{traitée}}{\delta x}$ et $\frac{\delta I_{traitée}}{\delta y}$ par les produits de convolution suivants:

$$\frac{\delta I_{traitée}}{\delta x} = I_{traitée} \otimes [1 \ 0 \ -1] \text{ et } \frac{\delta I_{traitée}}{\delta y} = I_{traitée} \otimes [1 \ 0 \ -1]^t$$

♦ L'opérateur de Prewitt :

L'opérateur de Prewitt définit $\frac{\delta I_{traitée}}{\delta x}$ et $\frac{\delta I_{traitée}}{\delta y}$ par la relation:

$$\frac{\delta I_{traitée}}{\delta x} = I_{traitée} \otimes \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \text{ et } \frac{\delta I_{traitée}}{\delta y} = I_{traitée} \otimes \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

♦ L'opérateur de Sobel :

qui définit $\frac{\delta I_{traitée}}{\delta x}$ et $\frac{\delta I_{traitée}}{\delta y}$ par :

$$\frac{\delta I_{traitée}}{\delta x} = I_{traitée} \otimes \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \text{ et } \frac{\delta I_{traitée}}{\delta y} = I_{traitée} \otimes \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

♦ L'opérateur MDIF :

Cet opérateur définit $\frac{\delta I_{traitée}}{\delta x}$ et $\frac{\delta I_{traitée}}{\delta y}$ par la relation :

$$\frac{\delta I_{traitée}}{\delta x} = I_{traitée} \otimes \begin{bmatrix} 0 & 1 & 0 & -1 & 0 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 3 & 0 & -3 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 0 & 1 & 0 & -1 & 0 \end{bmatrix} \text{ et } \frac{\delta I_{traitée}}{\delta y} = I_{traitée} \otimes \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -3 & -2 & -1 \\ 0 & -1 & -1 & -1 & 0 \end{bmatrix}$$

♦ L'opérateur de gradients directionnels de Kirsch :

Cet opérateur définit le gradient par la relation $\|\overrightarrow{Grad}(I_{traitee})\| = \max_{\alpha=0,1,\dots,7} \{ \|h_{\alpha} \otimes I_{traitee}\| \}$, avec :

$$h_0 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}, h_1 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}, h_2 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}, h_3 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix},$$

$$h_4 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}, h_5 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}, h_6 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}, h_7 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}.$$

La Figure 4.34 représente une comparaison entre les images gradients obtenues par l'application des différents opérateurs : dérivation directe, opérateur de Prewitt, opérateur de Sobel, opérateur MDIF et par l'opérateur de Kirsch.

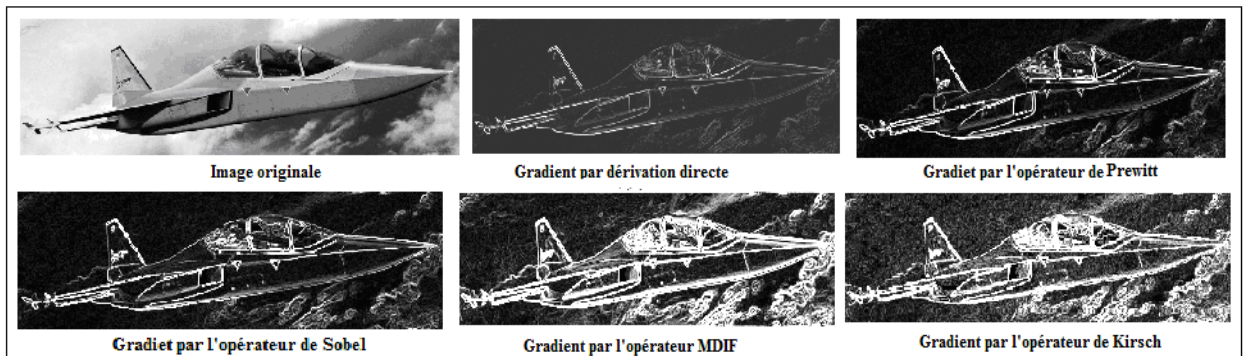


Figure 4. 34 Comparaison entre les gradients calculés par différents opérateurs.

♦ La localisation des extremums :

Cet algorithme crée une image de la même dimension que l'image validée en n'affichant que les maximums et les minimums locaux de celle-ci, le principe de son fonctionnement est de comparer la valeur de chaque pixel de l'image validée avec la valeur des pixels voisins qui le précèdent suivant la direction de développement du gradient, ce pixel ne sera affiché sur l'image traitée que s'il est supérieur à ses voisins.

La Figure 4.35 montre l'utilité de la localisation des extremums sur l'image gradient.

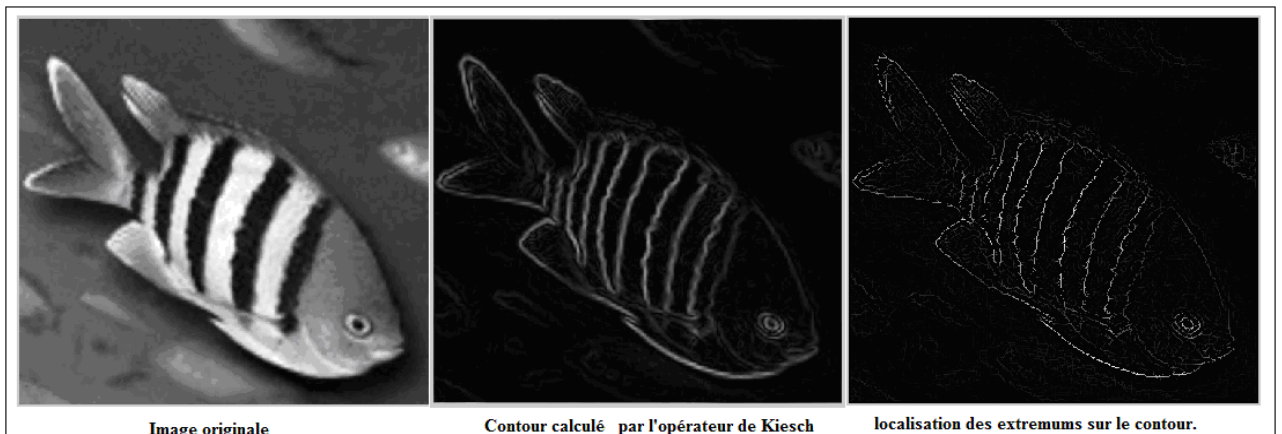


Figure 4. 35 L'utilité de la localisation des extremums sur l'image contour.

- *Binarisation de l'image :*

La binarisation consiste à transformer une image donnée I en une image binaire Ib en comparant l'intensité de chaque pixel avec un seuil s dit seuil de binarisation :

$$Ib(x,y) = \begin{cases} 1 & \text{si } I(x,y) \geq s \\ 0 & \text{si } I(x,y) < s \end{cases}$$

Deux algorithmes de binarisation sont proposés dans cette fenêtre : l'algorithme de binarisation par l'introduction manuelle du seuil qui nécessite l'introduction de la valeur du seuil, et l'algorithme de binarisation automatique qui utilise le seuil calculé par la méthode Otsu.

Le seuil d'Otsu est le seuil qui minimise l'écart quadratique entre l'image binaire (*image traitée*) et l'image en niveaux de gris (*image validée*). Le développement mathématique de cette méthode est présenté dans l'annexe 2.

La Figure 4.36 représente une comparaison entre trois images binarisées de la même image : deux sont binarisées en choisissant deux seuils introduits manuellement ($s_1=0.35$ et $s_2=0.6$) et une troisième avec un seuil calculé automatiquement par la méthode d'Otsu.

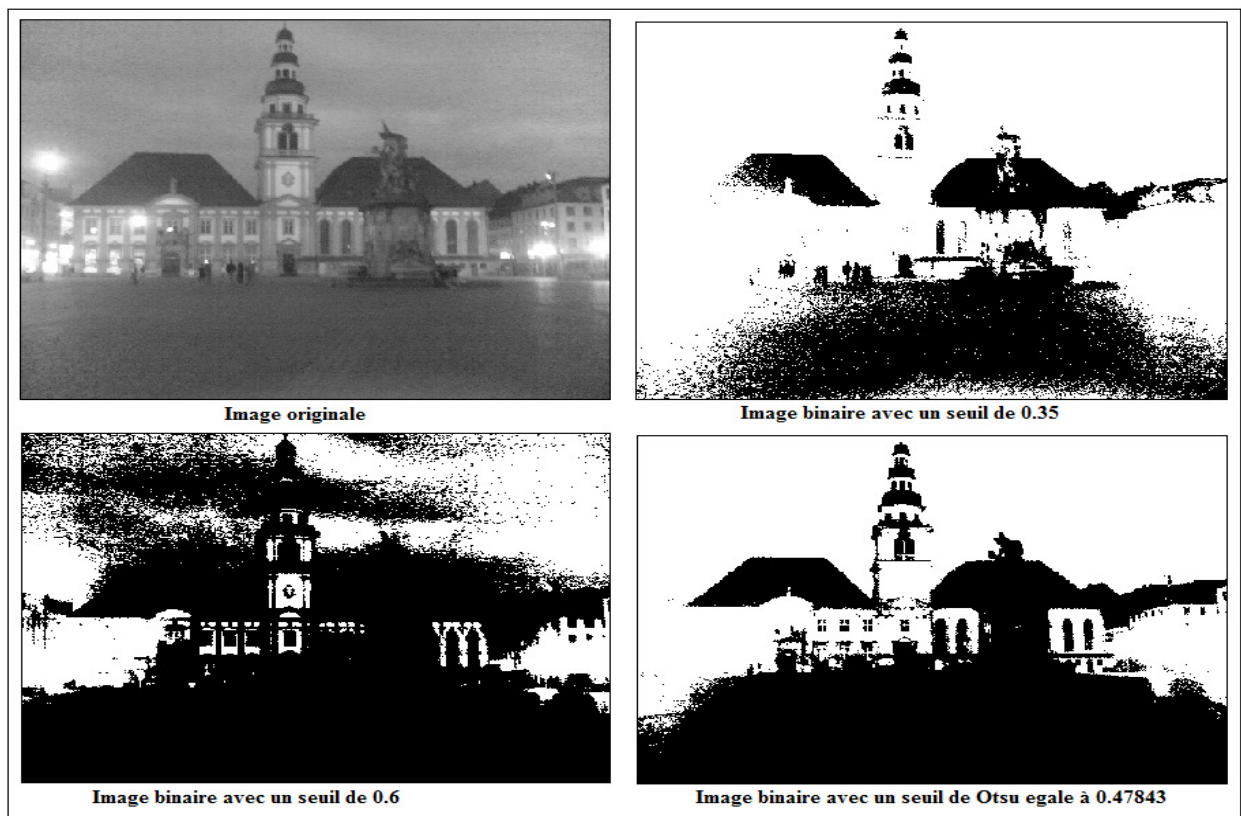


Figure 4. 36 Comparaison entre des images binarisées avec différents seuils.

- *Transformations morphologiques :*

Nous avons proposé quatre algorithmes de base de transformations morphologiques applicables sur les images binaires. Chacun de ces algorithmes filtre l'image avec un filtre non linéaire avec un masque rectangulaire dont on doit introduire la dimension :

♦ *La Dilatation :*

Cette transformation morphologique associe à chaque pixel de l'*image traitée* le maximum de sa valeur et des valeurs de ses voisins définis par le masque rectangulaire utilisé. Cette transformation favorise les zones des pixels blancs « ayant une valeur égale à 1 » en augmentant leur surface.

♦ *L'Erosion :*

Contrairement à la dilatation, l'érosion associe à chaque pixel de l'image traitée le minimum de sa valeur et des valeurs de ses voisins définis par le masque rectangulaire utilisé. Cette transformation favorise les zones des pixels noirs « ayant une valeur égale à 0 » en augmentant leur surface.

♦ *L'Ouverture:*

Cette opération est une composition des deux transformations précédentes, elle est obtenue par l'application d'une transformation de dilatation suivie d'une transformation d'érosion.

♦ *La Fermeture:*

La fermeture est l'opération inverse à l'ouverture, elle est obtenue par l'application d'une transformation d'érosion suivie d'une transformation de dilatation.

La Figure 4.37 montre l'effet de l'application des différents algorithmes de transformation morphologique présentés sur une image binaire.

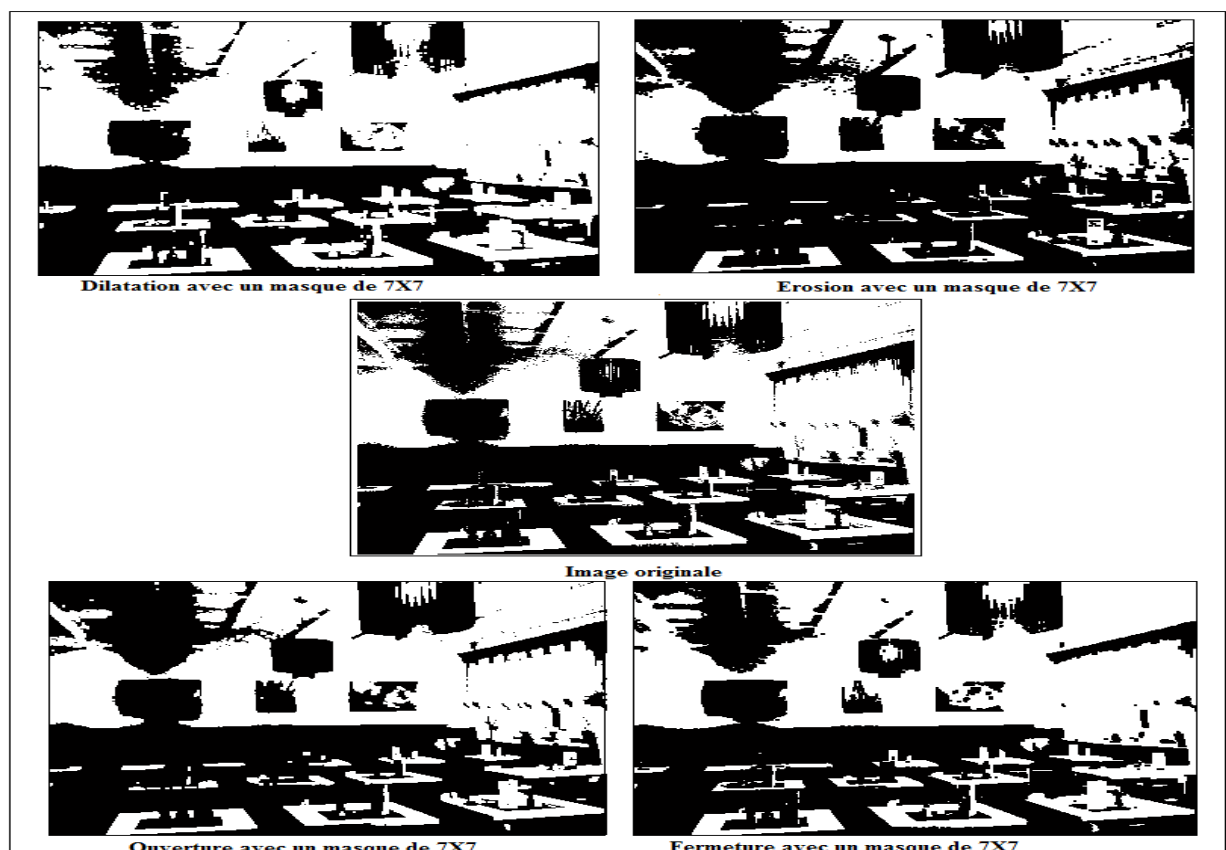


Figure 4. 37 L'effet des différents algorithmes de transformations morphologiques implémentées.

- *Etiquetage et étude des régions* :

Cette opération détecte, sépare et affiche les caractéristiques métriques des différents objets composant l'image traitée en exploitant une combinaison de fonctions optimisées prédéfinies dans la plateforme Matlab.

L'algorithme de cette opération commence par la binarisation de l'image traitée avec un seuil de 0.5. Pour utiliser un autre seuil, il suffit que l'image traitée soit déjà binarisée, il faut ensuite effectuer l'opération d'étiquetage en utilisant la fonction *bwlabel*. Il termine par le calcul et l'affichage des différents objets et leurs caractéristiques métriques.

A la fin, l'algorithme produit 3 autres fenêtres : la première affiche l'image étiquetée en affectant à chaque pixel de cette image l'index de la région à laquelle il appartient, la deuxième affiche les objets de l'image traitée avec des couleurs différentes, et la dernière fenêtre affiche le nombre des objets disponibles et les caractéristiques métriques de chaque objet (surface, centre de gravité, diamètre maximal, et diamètre minimal) suivant la syntaxe suivante :

```

Le nombre des objets est égal à [nombre d'objet]
*****
Information sur l'objet N [numéro de l'objet]
La surface de l'objet= [S] pixel²
Le centre du gravité de l'objet=(xg,yg)
Le diamètre maximal de l'objet= D_max pixel
Le diamètre minimal de l'objet= d_min pixel
*****
    
```

La Figure 4.38 représente les trois fenêtres produites par l'application de cet algorithme

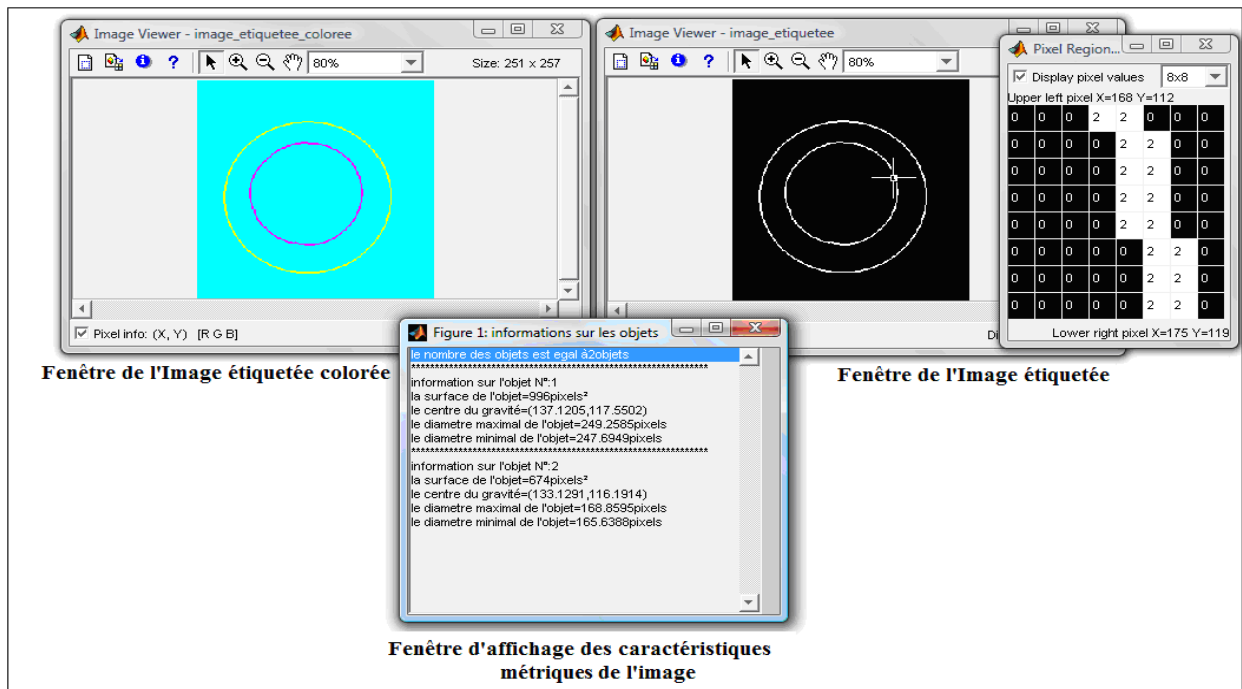


Figure 4. 38 Fenêtres produites par l'algorithme d'étiquetage et études des régions.

- **Panel Historique :**

Ce panel affiche l'adresse et tout l'historique des opérations, des algorithmes et des paramètres utilisés pour le traitement effectué.

IV.6 Exemples d'études

Afin d'étudier, de comparer et d'optimiser les algorithmes de super-résolution vus précédemment, nous avons utilisé plusieurs types d'images dont :

IV.6.1 Images Dicom du modèle 11 et modèle 12

Ces images construisent une base de données de 1024 images Dicom obtenues en scannant deux modèles en silicone d'anévrismes avec un scanner flat panel de Philips (Alura FD20) : le modèle 11 simule un anévrisme d'un diamètre normal (2.5 fois le diamètre de l'artère), et le modèle 12 simule un anévrisme géant (d'un diamètre 5 fois plus grand que le diamètre de l'artère). Ces images provenant du CHU Genève, nous ont été remises par Mr R.OUARED dans le cadre du projet « MEDPROJECTS »

La Figure 4.39 représente une image des deux modèles utilisés: le modèle 11 et le modèle 12.

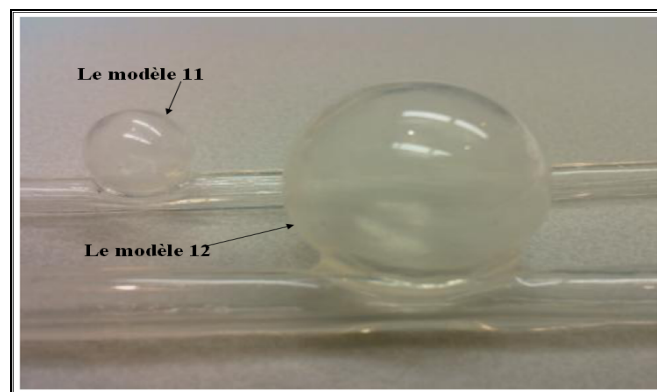


Figure 4. 39 Image contenant les deux modèles d'anévrismes :11 et 12.

Les images Dicom de la base de données sont des coupes en niveaux de gris de dimension 556×556 pixels dont la surface de la zone d'intérêt ne peut occuper plus de la moitié de la surface de l'image, la figure 4.40 représente des échantillons de la base de données:

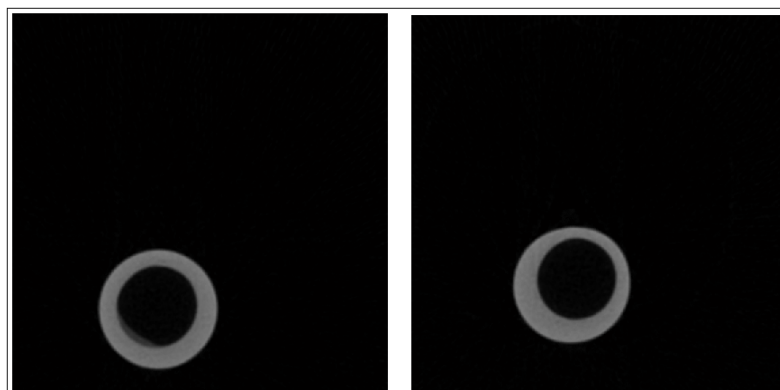


Figure 4. 40 Echantillons de la base de données des images Dicom.

Cette base de données nous a servi comme exemple du test, du calcul et d'optimisation des filtres de *Whittaker* en utilisant la fenêtre *Optimisation superrésolution monoframe* : En étudiant les taux d'erreurs dues au suréchantillonnage des images souséchantillonnées à partir des images de la base de données, nous avons constaté que le facteur de correction de l'intensité optimisé k est pratiquement d'une valeur égale à 0.10000 ± 0.0075 pour tous les filtres (k étant calculé entre $k_{min} = 0.01$ et $k_{max} = 0.2$ avec un pas de 0.0025). Et que la taille moyenne du filtre utilisé est liée au facteur de suréchantillonnage (K_x, K_y) : En augmentant ce facteur, la taille du filtre augmente et vice versa. En effet, la taille optimisée du filtre fenêtré avec une fenêtre de *Hamming* pour cette base de données est d'une valeur moyenne de 10 pour le facteur 2×2 , de 11 pour le facteur 4×4 et de 14 pour le facteur 8×8 .

L'erreur moyenne sur les intensités des pixels est maximisée par 0.0029 pour le facteur de suréchantillonnage 2×2 , de 0.0054 pour le facteur 4×4 et de 0.0081 pour le facteur 8×8 . L'influence de ces erreurs sur les mesures métriques est négligeable. En effet, nous avons remarqué que l'erreur moyenne calculée sur les diamètres maximaux de la paroi extérieure de l'altère (voir l'altère et l'anévrisme sur certaines images de la base de données) était de 0.2043 pixel par diamètre d'image avec une erreur maximale de 3 pixels par diamètre. Ces erreurs restent négligeables devant la valeur minimale des diamètres maximaux de la paroi extérieure (244 pixels).

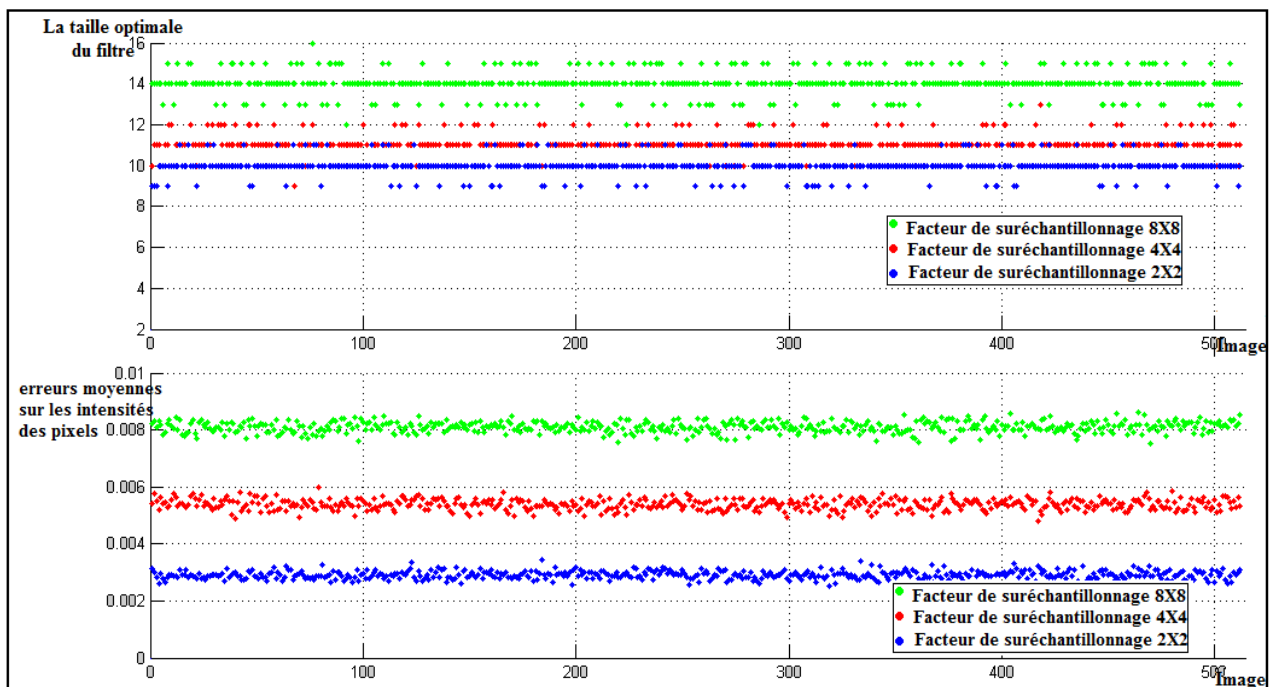


Figure 4.41 Distributions des tailles optimales des fenêtres de Hamming appliquées et des erreurs moyennes des intensités sur pixels produites en fonction des facteurs de suréchantillonnage.

La Figure 4.41 représente la distribution des tailles optimales des fenêtres de Hamming appliquées sur le filtre de Whittaker, ainsi que la distribution des erreurs moyennes sur les

intensités des pixels en fonction des facteurs de suréchantillonnages et des images de la base de données.

IV.6.2 Images de référence de Matlab

Le logiciel Matlab utilise une variété d'images dans ses exemples et ses démonstrations. Ces images sont devenues les images de référence les plus exploitées par les différents utilisateurs de la plateforme à travers le monde pour tester et comparer leurs programmes.

Nous avons essayé de tester et de comparer les algorithmes de la superrésolution développés dans ce travail en les appliquant sur un groupe de 15 images sélectionnées des images de référence présentées par Matlab.

Le tableau de la Figure 4.42 expose et compare les taux d'erreurs obtenues en appliquant les différents algorithmes de superrésolution monoframe sur les images de référence de Matlab pour un facteur de suréchantillonnage 4×4 . Les tailles des fenêtres N et les facteurs de correction des intensités K utilisés sont des paramètres optimisés calculés en variant N entre 10 et 15 avec un pas de 1 et K entre 0.085 0.2 avec un pas de 0.005. En comparant les résultats présentés dans cette Figure, nous remarquons que le meilleur algorithme de superrésolution adapté à l'image dépend de la nature des images.

Nom de l'image	Erreur d'interpolation polynomiale			Erreur d'interpolation par filtre de Whittaker fenêtré par une fenêtre de								
	bilinéaire	bicubique	Spline bicubique	Hamming			Hann			Blackman		
				N	K	Erreur	N	K	Erreur	N	K	Erreur
board.tif	0.19125	0.19925	0.19872	10	0.09	0.20015	10	0.09	0.19938	10	0.125	0.20317
Trees.png	0.055647	0.054627	0.056588	10	0.1	0.058745	10	0.1	0.05851	10	0.14	0.06178
cameraman.tif	0.040593	0.041643	0.043775	10	0.1	0.04771	10	0.1	0.047294	10	0.14	0.050524
pout.tif	0.028185	0.027988	0.028118	10	0.1	0.018109	10	0.1	0.017994	15	0.14	0.019762
rice.png	0.050504	0.049804	0.0052714	10	0.1	0.055526	10	0.1	0.55206	10	0.14	0.58643
CT-MONO2-16-ankle.dcm	0.026193	0.027094	0.028305	10	0.1	0.036877	11	0.1	0.036453	10	0.14	0.038363
moon.tif	0.011833	0.01184	0.012261	10	0.1	0.012515	10	0.1	0.012465	15	0.14	0.013495
testpat1.png	0.10267	0.089495	0.089643	10	0.1	0.12023	10	0.1	0.11964	10	0.14	0.12853
circles.png	0.020022	0.018417	0.020145	10	0.1	0.0248	10	0.1	0.024363	10	0.14	0.027567
peppers.png	0.012778	0.012551	0.013232	10	0.1	0.015027	10	0.1	0.014945	15	0.14	0.01651
saturn.png	0.357532	0.037401	0.039489	11	0.1	0.0038268	10	0.1	0.0039503	15	0.14	0.0042695
circuit.tif	0.0036721	0.003319	0.0033227	10	0.1	0.040788	10	0.1	0.040548	10	0.14	0.042383
Westconcordort hophoto.png	0.067842	0.68744	0.071768	10	0.1	0.073145	10	0.1	0.072836	10	0.14	0.075282

Figure 4. 42 Taux d'erreurs obtenues en appliquant les algorithmes de superrésolution sur des images de référence de Matlab.

Ces images nous ont aussi servi pour vérifier la relation existante entre le taux d'erreur produite lors de la reconstruction par superrésolution multiframe et le nombre des imagerie utilisées ainsi que leur distribution : nous avons constaté que le taux d'erreur diminue en augmentant le nombre d'imagerie utilisées ou en choisissant une distribution qui augmente le nombre de pixels sources au niveau des contours.

IV.6.3 Autres images

En testant ces algorithmes sur d'autres images, nous pouvons confirmer qu'on ne peut pas généraliser l'existence d'un meilleur algorithme parmi les algorithmes étudiés, et que l'on pourrait adapter à tous les types d'images. En effet, chaque type possède son propre algorithme de superrésolution qui lui est adapté.

A titre d'exemple, nous avons constaté que l'algorithme de superrésolution le plus adapté à l'image gauche de la Figure 4.43 est l'algorithme d'interpolation par le plus proche voisin, et que l'algorithme le plus adapté à l'image droite de la même Figure est l'algorithme d'interpolation par le filtre de *Whittaker* fenêtré avec un filtre de *Blackman*.

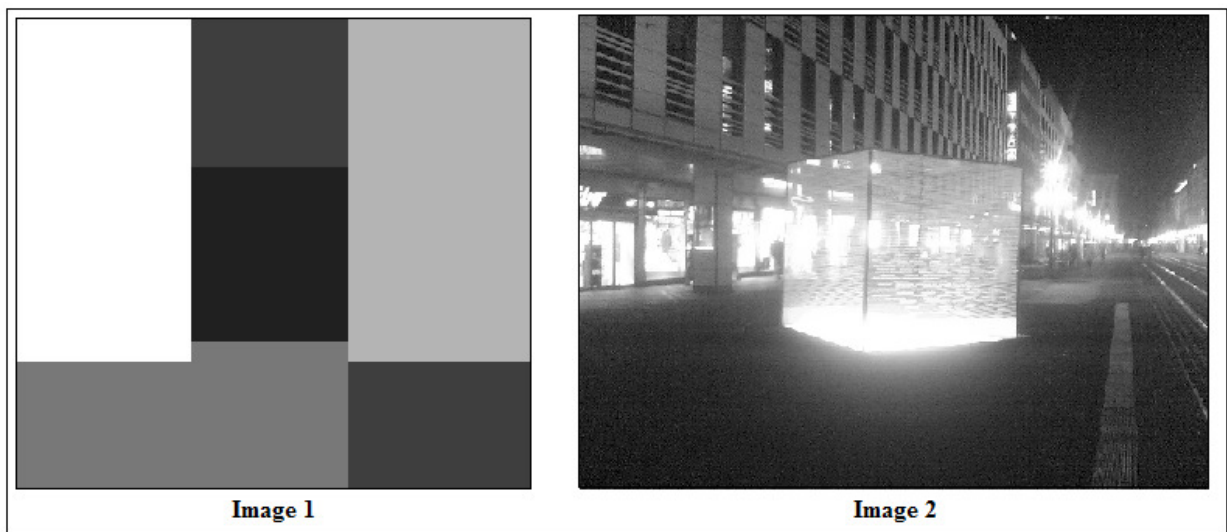


Figure 4. 43 Exemple sur les images étudiées.

- L'algorithme de superrésolution le plus adapté à l'image1 est l'algorithme d'interpolation par le plus proche voisin.
- L'algorithme de superrésolution le plus adapté à l'image2 est l'algorithme d'interpolation par le filtre de *Whittaker* fenêtré avec un filtre de *Blackman*.

IV.7 Conclusion

Nous avons présenté, dans ce chapitre, l'interface graphique développée à cet effet afin de pouvoir exploiter les techniques de superrésolution présentées dans les chapitres précédents, ainsi que les différents algorithmes d'optimisation, de comparaison que nous avons élaboré.

En comparant les différents résultats de l'application des techniques de superrésolution, nous avons remarqué que chaque type d'image a son propre algorithme de superrésolution qui lui est adapté. Nous avons aussi constaté que le taux d'erreur diminue en augmentant le nombre d'images utilisées ou le nombre des pixels sources situé au niveau des contours.

Concernant les coefficients optimisés des filtres de *Whittaker* fenêtrés, nous avons constaté que le choix optimal de la taille du filtre dépend du facteur de suréchantillonnage, de la nature de l'image et de la fenêtre utilisée, alors que le choix optimal du coefficient de l'intensité ne dépend que de la nature de l'image superrésolue.

Conclusion Générale

CONCLUSION GENERALE

Le présent travail met en évidence la possibilité de reconstruire une image à haute résolution à partir d'une ou plusieurs images en exploitant différents algorithmes de superrésolution monoframe basés sur les polynômes et les filtres d'interpolation et en les adaptant à la superrésolution multiframe par utilisation du diagramme de Voronoï et de la triangulation de Delaunay.

Dans une première étape, nous avons montré l'utilité de la superrésolution comme une opération de prétraitement dans l'amélioration de la qualité de l'image en augmentant sa résolution. Nous avons aussi exposé les différents travaux et théories employés dans ce domaine en se basant sur la théorie d'échantillonnage et de la reconstruction des signaux.

Dans une seconde étape, nous avons détaillé et implémenté des différents algorithmes de superrésolution basés sur l'interpolation. Comme nous avons élaboré notre propre algorithme « *Algorithme récurrent par niveaux de sélection* » qui permet de choisir les points sources utilisés dans la superrésolution multiframe par interpolation bicubique et par spline bicubique

Une étude comparative entre ces derniers nous a permis de constater que la reconstruction par la superrésolution multiframe présente des résultats meilleurs que celle par la superrésolution monoframe et que nous ne pouvons pas généraliser l'existence d'une technique d'interpolation optimale pour toute variante d'images, mais cela dépend principalement de l'image choisie.

A travers cette étude, nous avons aussi constaté que le choix optimal de la fenêtre appliquée sur le filtre de Whittaker et du coefficient de l'intensité k ne dépendaient que de la nature de l'image superrésolue, alors que le choix optimal de la taille du filtre dépendait du facteur de suréchantillonnage, de la nature de l'image et de la fenêtre utilisée.

Concernant la superrésolution multiframe, nous avons déduit que l'emplacement optimal des imageries est celui qui permet une forte représentation (un grand nombre de points sources) au niveau des contours. En effet, cet emplacement permet d'augmenter la fréquence

d'échantillonnage au niveau des contours (les hautes fréquences) d'où la convergence vers la vérification du théorème de Shannon.

Ce travail nous a permis d'acquérir les notions de base et les principes fondamentaux nécessaires pour l'approfondissement de nos recherches dans le domaine de la superrésolution en espérant pouvoir développer d'autres algorithmes plus fiables et plus efficaces et de les exploiter sur des images colorées.

Bibliographie

Bibliographie

- [1] H. Nyquist, *Certain topics in telegraph transmission theory*, Trans. AIEE, vol. 47, pp. 617-644, Apr. 1928, repris dans: *Proc. IEEE, Vol. 90, No. 2, Feb 2002*
- [2] K. Küpfmüller, *Über die Dynamik der selbsttätigen Verstärkungsregler*, Elektrische Nachrichtentechnik, vol. 5, no. 11, pp. 459-467, 1928. (Version allemande)
K. Küpfmüller, *On the dynamics of automatic gain controllers*, Elektrische Nachrichtentechnik, vol. 5, no. 11, pp. 459-467. (Traduction en anglais)
- [3] C. E. Shannon, *Communication in the presence of noise*, Proc. Institute of Radio Engineers, vol. 37, no.1, pp. 10-21, Jan. 1949. repris dans: *Proc. IEEE, Vol. 86, No. 2, 1998*
- [4] E. T. Whittaker, *On the Functions Which are Represented by the Expansions of the Interpolation Theory*, Proc. Royal Soc. Edinburgh, Sec. A, vol.35, pp. 181-194, 1915
- [5] J. M. Whittaker, *Interpolatory Function Theory*, Cambridge Univ. Press, Cambridge, England, 1935
- [6] V. A. Kotelnikov, *On the carrying capacity of the ether and wire in telecommunications*, Material for the First All-Union Conference on Questions of Communication, Izd. Red. Upr. Svyazi RKKA, Moscow, 1933
- [7] Hans Dieter Lüke, *The Origins of the Sampling Theorem*, IEEE Communications Magazine, pp.106–108, April 1999.
- [8] Henry Wilbraham, *On a certain periodic function*, Cambridge Dublin Math. J. 3, 1848, p. 198-201
- [9] Josiah Willard Gibbs, *Fourier Series*, Nature 59, 200 (1898) and 606 (1899).
- [10] Maxime Bôcher, *Introduction to the theory of Fourier's series*, Annals of Math. 7 (1906), p. 81-152
- [11] Ralph Beebe Blackman et John Tukey, *Particular Pairs of Windows*, The Measurement of Power Spectra from The Point of View of Communications Engineering, New York: Dover, 1959,
- [12] Richard Wesley Hamming, *Digital Filters*, Prentice Hall, 1977; second edition 1983; third edition 1989.
- [13] R.Y.Tsai et T.S.Huang, *Advances in Computer Vision and Image Processing*, Volume 1, chapitre Multiframe Image Restoration and Registration, JAI Press Inc, 1984.
- [14] S.P.Kim et N.K.Bose, *Recursive reconstruction of High-Resolution Image From Noisy Undersampled Multiframe*, IEEE Transaction on Acoustics, Speech and Signal Processing, 38(6), Juin 1990.
- [15] S.P.Kim et W.Su., *Recursive High-Resolution reconstruction of Blurred Multiframe Image*, IEEE Transaction on Image Processing, 2(4), Octobre 1993
- [16] S.Rhee et M.G.Kang, *Discrete Cosine Transform Based Regularized High-Resolution Image Reconstruction Algorithm*. Optical Engineering, 38 (8), août 1999.

- [17] Fabrice Humblot, *Détection de petits objets dans une image en utilisant les techniques de Superrésolution*, thèse de Doctorat en sciences de l'université de Paris sud XI, Décembre 2005.
- [18] H. Ur et D. Gross, , *Graphical Model and Image Processing*, 54(2), Mars 1992.
- [19] H. Shekarforoush et R. Chellapa ; *Data-Driven Multichannel Superresolution with Application to Video Sequences*, *Journal of Optical Society of America A*, 16(3), Mars 1999.
- [20] K. W. Simon, *Digital image reconstruction and resampling for geometric manipulation*, in *Symposium on Machine Processing of Remotely Sensed Data*, C. D. McGillem and D. B. Morrison, Eds. Piscataway, NJ: IEEE Press, 1975.
- [21] ERIK MEIJERING, *A Chronology of Interpolation: From Ancient Astronomy to Modern Signal and Image Processing*, *PROCEEDINGS OF THE IEEE*, VOL. 90, NO. 3, Mars 2002
- [22] H. S. Hou and H. C. Andrews, *Cubic splines for image interpolation and digital filtering*, *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, Dec. 1978.
- [23] I. J. Schoenberg, *Contributions to the problem of approximation of equidistant data by analytic functions. Part B—on the problem of osculatory interpolation. A second class of analytic approximation formulae*, *Quart. Appl. Math.*, vol. IV, no. 2, pp. 112–141, 1946.
- [24] J. A. Parker, R. V. Kenyon, and D. E. Troxel, *Comparison of interpolating methods for image resampling*, *IEEE Trans. Med. Imag.*, vol. MI-2, pp. 31–39, Mar. 1983.
- [25] J. L. Ostuni, A. K. S. Santha, V. S. Mattay, D. R. Weinberger, R. L. Levin, and J. A. Frank, *Analysis of interpolation effects in the reslicing of functional MR images ; J. Comput.-Assisted Tomography*, vol. 21, no. 5, pp. 803–810, 1997.
- [26] T. M. Lehmann, C. Gönner, and K. Spitzer, *Survey: Interpolation methods in medical image processing*, *IEEE Trans. Med. Imag.*, vol. 18, pp. 1049–1075, Nov. 1999.
- [27] K.D Sauer et J.P. Allebach. ; *Iterative Reconstruction of Band-Limited Images from Nonuniformly Spaced Samples*, *IEEE Transactions on Circuits and Systems*, 34 (12), décembre 1987
- [28] S.Yeh et H.Stark. ; *Iterative and One Step Reconstruction from Nonuniform Samples by Convex Projections*, *Jornal of Optical Society of America*, 7(3), mars 1990
- [29] D.Granrath et J.Lersch, *Fusion of Images on Affine Sampling Grids* .*Journal of the Optical Society of America*, Avril 1998.
- [30] .E.Boult et M.C.Chiang, *Efficient Superresolution via Image Warping ;. Image and Vision Computing*, 18 : 761-771, 2000
- [31] N.Nguyen et P.Milanfar, *An Efficient Wavelet-Based Algorithm for Image Superresolution*, In *Proceeding of the IEEE International Conference on Image Processing*, Vancouver,2000.
- [32] S. Lertrattanapanich et N. K. Bose,*High Resolution Image Formation From Low Resolution Frames Using Delaunay Triangulation*, *IEEE Transactions on Image Processing*, vol. 11, no. 12, Décembre 2002
- [33] T. Saito, Y. Ishii, Y. Nakagawa et T. Komatsu, *Adaptable image interpolation with skeleton-texture separation*, In *Proceedings of the 13th IEEE International Conference on Image Processing*, 2006.

- [34] S.Peleg, D.Keren et L.Shweitzer ; *Improving Image Resolution Using Subpixel Motion* ; Pattern Recognition Letters, 5(3), mars 1987.
- [35] A.Zomet et S.Peleg, *Efficient Superresolution and applications to Mosaics*, in Proceeding of the international conference on Pattern Recognition(ICPR 00), Barcelone, September 2000.
- [36] N.Nguyen, P.Milanfar et G.H.Golub ; *A computationally Efficient Image Superresolution Algorithm* ; IEEE Transactions on image Processing, Volume 10(4), Vancouver, mars 2001.
- [37] B.C.Tom, A.K.Katsagellos et N.P.Galatsanos ; *Reconstruction of a high resolution Image from registration and restoration of low resolution images* ; In proceeding of the IEEE International Conference on Image Processing, Austin, 1994.
- [38] R.R.Schultz et R.L.Stevenson ; *Extraction of High- resolution Frames from Video Sequences* ;. IEEE Transactions on Image Processing, 5(6), juin 1996.
- [39] Simon Baker et Takeo Kanade. *Limits on Super-Resolution and How to Break Them*. Computer Vision and Pattern Recognition (CVPR 2000), 2000.
- [40] B.Demidovitch et I.Maron, *Eléments de calcul numérique*, édition Mir, Moscou, 1979.
- [41] M.Bourdeau et J Gelinias, *Analyse numérique élémentaire*, Gaëtan Morin, Québec, Canada, 1998.
- [42] P. J. Green, R. R. Sibson. *Computing Dirichlet Tessellations in the Plane*. Computer Journal, vol. 21 nr 2, p. 168-173, 1978.
- [43] L. Guibas, D. Knuth, M. Sharir. *Randomized Incremental Construction of Delaunay and Voronoi Diagrams*. Algorithmica, vol. 7, p. 381-413, 1992.
- [44] Frederic De Coulon, *Théorie et traitement des signaux*, Presse polytechnique et universitaires ROMANDES, Lausanne, 1998
- [45] Alan V Oppenheim, Alan S Wilsky et S.Hamid Nawab; *Signals and systems*, (2^{ième} édition), Prentice-hall, 1997
- [46] Simon Haykin et Barry Van Veen, *Signals and systems*, (2^{ième} édition, édition internationale), John Wiley & sons, 2003.
- [47] M.J Roberts, *Signals and systems: Analysis using transform methods and Matlab*, (édition internationale), MC Graw Hill, 2003.
- [48] M.Kunt, G.Granlund, M.Kocher, *Traitement numérique des image*, Presse polytechnique et universitaires ROMANDES et CENT-ENST, 1993.

ANNEXE

Annexe I

THEORIE D'ECHANTILLONNAGE ET RECONSTRUCTION DES SIGNAUX

A1.1 Introduction

La théorie d'échantillonnage est basée sur le théorème de Shannon-Nyquist considéré comme l'un des plus importants théorèmes de traitement des signaux numériques. En effet, ce dernier exprime les conditions nécessaires et suffisantes pour effectuer un échantillonnage idéal et reconstruire les signaux discrets sans erreurs.

Dans cet annexe, nous allons, en premier lieu, définir et appliquer ce théorème sur l'espace des signaux monodimensionnels, puis généraliser son application sur l'espace des signaux bidimensionnels.

A1.2 L'échantillonnage

L'opération d'échantillonnage est une opération de discrétisation du signal analogique dont son évolution est continue en prélevant des valeurs numériques ponctuelles représentant ses amplitudes (**échantillons**) à des points bien déterminés (**instants d'échantillonnage**).

Nous qualifions un échantillonnage comme idéal dès que l'on peut supposer ou approcher une prise instantanée des échantillons. Cela peut être modélisé, mathématiquement, par le produit du signal analogique $z(x)$ et une suite périodique d'impulsions idéales appelée peigne de Dirac $P(\delta)$ définie comme suit:

$$P(\delta) = \sum_{k=-\infty}^{k=+\infty} \delta(x - x_k)$$

Où x_k représente l'instant k d'échantillonnage et $\delta(x)$ une impulsion de Dirac définie comme suit :

$$\delta(x) = \begin{cases} 1 & x = 0 \\ 0 & \text{si non} \end{cases}$$

D'où le signal échantillonné $z^*(x)$ est défini par :

$$z^*(x) = z(x) \cdot \sum_{k=-\infty}^{k=+\infty} \delta(x - x_k) \quad (\text{A1.1})$$

Pour un échantillonnage périodique d'une période x_e , on aura :

$$z^*(x) = z(x) \cdot \sum_{k=-\infty}^{k=+\infty} \delta(x - kx_e) \quad (\text{A1.2})$$

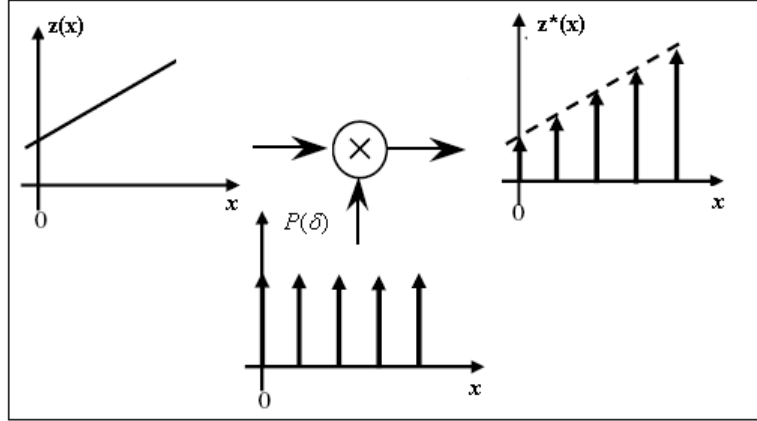


Figure A1. 1 Modèle mathématique de l'échantillonnage.

$z(x)$: signal analogique d'entrée. $z^*(x)$: signal échantillonné de sortie.

$P(\delta)$: peigne de Dirac.

Soit $Z(f)$ et $Z^*(f)$ les représentations fréquentielles des signaux $z(x)$ et $z^*(x)$ et $f_e = 1/x_e$ la fréquence d'échantillonnage :

Nous avons :

$$\begin{aligned} z^*(x) &= z(x) \sum_{k=-\infty}^{k=+\infty} \delta(x - kx_e) \\ \Rightarrow TF\{z^*(x)\} &= TF\left\{z(x) \sum_{k=-\infty}^{k=+\infty} \delta(x - kx_e)\right\} \\ \Rightarrow Z^*(f) &= Z(f) * TF\left\{\sum_{k=-\infty}^{k=+\infty} \delta(x - kx_e)\right\} \end{aligned}$$

Or on a :

$$TF\left\{\sum_{k=-\infty}^{k=+\infty} \delta(x - kx_e)\right\} = \sum_{m=-\infty}^{m=+\infty} \delta(f - mf_e)$$

D'où nous trouverons:

$$\begin{aligned} Z^*(f) &= Z(f) * \sum_{m=-\infty}^{m=+\infty} \delta(f - mf_e) \\ \Rightarrow Z^*(f) &= \sum_{m=-\infty}^{m=+\infty} Z(f - mf_e) \end{aligned} \quad (\text{A1.3})$$

En effet, la relation (A1.3) montre que l'opération d'échantillonnage produit une périodicité du spectre à la sortie du système: le spectre du signal de sortie $Z^*(f)$ est constitué d'une suite périodique de réplique du spectre du signal d'entrée $Z(f)$ avec une périodicité $f_e = \frac{1}{x_e}$.

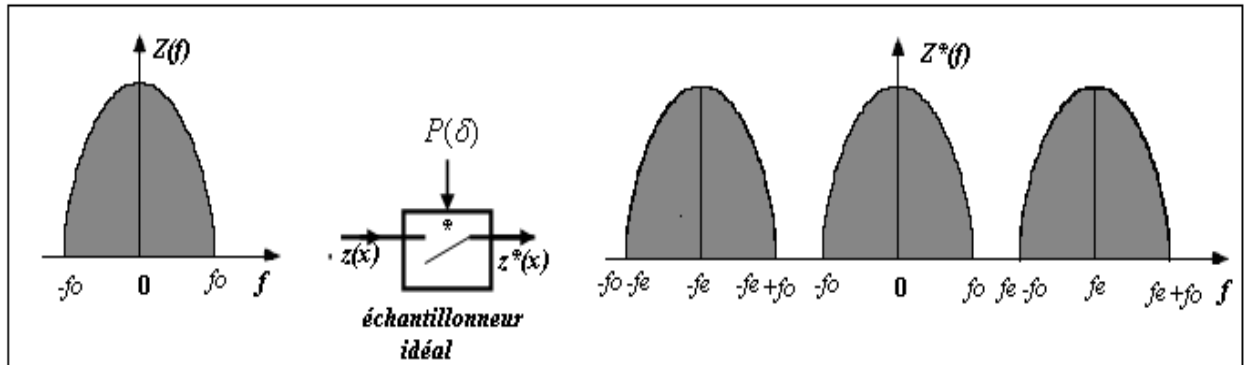


Figure A1.2 Effet de l'échantillonnage au domaine spectral.

$Z(f)$: spectre du signal analogique d'entrée $z(x)$. $Z^*(f)$: spectre du signal échantillonné $z^*(x)$.

• **Notation**

- Soit $z[k]$ une autre représentation du signal échantillonné $z^*(x)$ tel que

$$z[k_i] = z^*(k_i \cdot x_e).$$

- nous appelons le signal $z(x)$ un signal original du signal échantillonné $z^*(x)$.

A1.3 Souséchantillonnage, Suréchantillonnage et Reconstruction du signal

Nous appelons un **Souséchantillonnage** ou **Down sampling** l'opération d'extraction des échantillons d'un signal déjà échantillonné en lui diminuant sa fréquence d'échantillonnage. Le résultat d'un souséchantillonnage uniforme, avec le facteur K_{ds} , d'un signal $z^*(x)$ échantillonné, avec la fréquence f_e , est équivalent à un échantillonnage avec la fréquence $f_{ds} = \frac{f_e}{K_{ds}}$.

Contrairement au souséchantillonnage, le **Suréchantillonnage** ou le **Up sampling** est défini comme l'introduction d'autres échantillons à un signal échantillonné afin d'augmenter sa fréquence d'échantillonnage. Le résultat d'un suréchantillonnage uniforme, avec un facteur K_{us} , d'un signal $z^*(x)$ échantillonné, avec la fréquence f_e , est équivalent à un échantillonnage avec la fréquence $f_{us} = K_{us} \cdot f_e$.

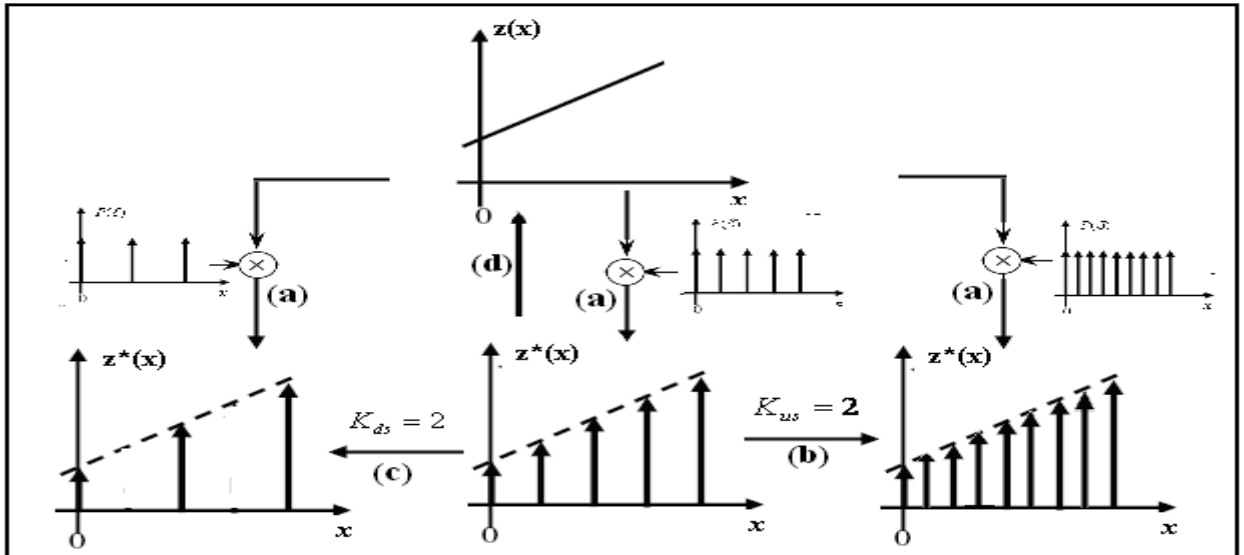


Figure A1.3 Souséchantillonnage, Suréchantillonnage et Reconstruction.

- (a) opérations d'échantillonnage. (b) opération de suréchantillonnage.
- (c) opération de souséchantillonnage. (d) opération de reconstruction.

Nous définissons une opération de **reconstruction** comme l'opération inverse à l'opération d'échantillonnage. Autrement dit, la reconstruction est l'opération de la production d'un signal analogique $z(x)$ à partir du signal échantillonné $z^*(x)$.

Cette reconstruction n'est pas toujours évidente. En effet l'échantillonnage de plusieurs fonctions peut aboutir à la même fonction discrète comme le montre la relation A1.3 : A titre d'exemple, la Figure A1.4 montre que l'échantillonnage de deux fonctions différentes, avec deux fréquences bien choisies peut produire la même fonction échantillonnée : dans ce cas, si on échantillonne Z1 avec une fréquence $f_{e1} = 2f_1$ et Z2 avec une fréquence $f_{e2} = 2f_2$ on trouvera le même résultat, c'est-à-dire, $Z1^*(f) = Z2^*(f) = Z^*(f)$.

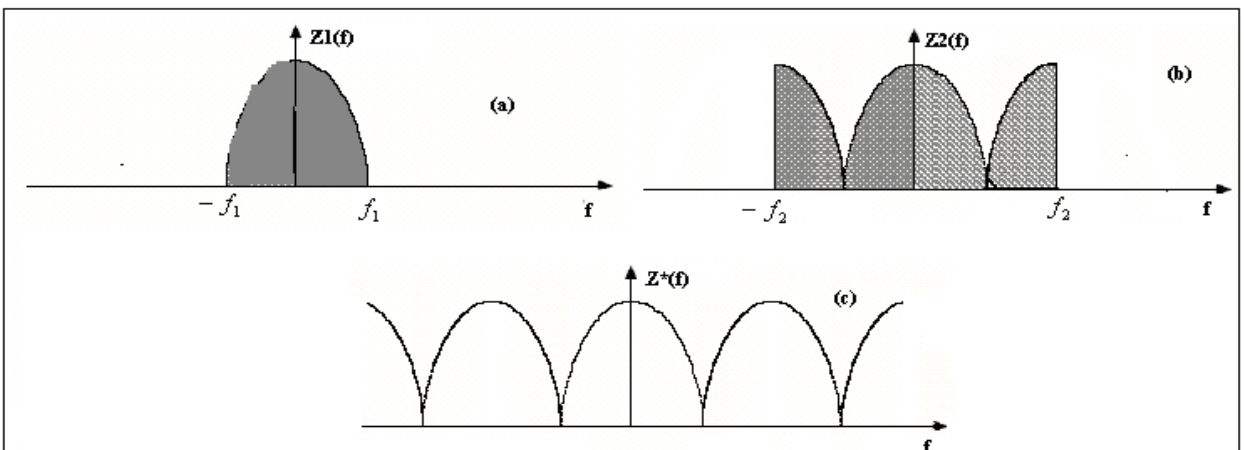


Figure A1.4 Même fonction échantillonnée pour deux fonctions différentes.

- (a) Z1 est le spectre de la première fonction. (b) Z2 est le spectre de la deuxième fonction.
- (c) Z* est le spectre de la fonction échantillonnée de Z1 et Z2.

Dans l'exemple de la Figure A1.5, le résultat d'échantillonnage des fonctions z_1 et z_2 définies par :

$$\begin{cases} z_1 = \sin\left(\frac{2\pi \cdot f_e \cdot x}{9}\right) \\ z_2 = \sin\left(\frac{16\pi \cdot f_e \cdot x}{9}\right) \end{cases}$$

est le même en utilisant la même fréquence d'échantillonnage f_e , autrement dit : $z[k] = z_1[k] = z_2[k]$.

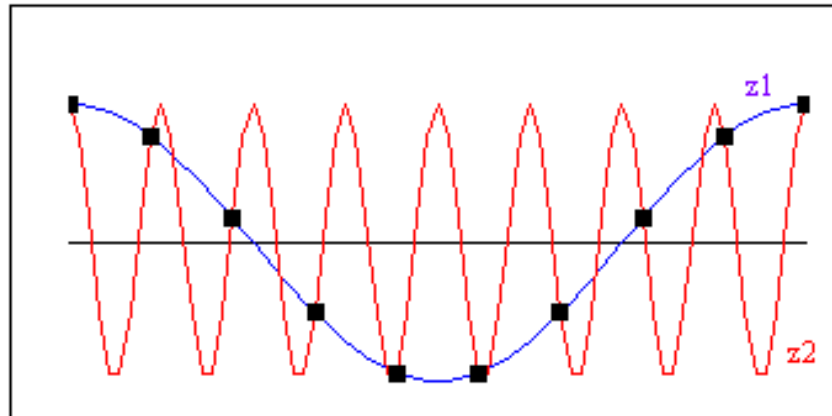


Figure A1. 5 Même fonction échantillonnée obtenue par l'échantillonnage de deux fonctions analogiques avec la même fréquence d'échantillonnage.

$$z_1 = \sin\left(\frac{2\pi \cdot f_e \cdot x}{9}\right) \qquad z_2 = \sin\left(\frac{16\pi \cdot f_e \cdot x}{9}\right)$$

A partir des deux exemples précédents, nous pouvons conclure qu'une bonne reconstruction du signal doit être faite en connaissance de la fréquence de coupure et de la fréquence d'échantillonnage.

A1.4 Recouvrement du signal « aliasing »

Malgré la reconnaissance de la fréquence de coupure du signal analogique et la fréquence d'échantillonnage, la reconstruction peut ne pas être possible. En effet, un mauvais choix de la fréquence d'échantillonnage peut produire un phénomène de recouvrement ou repliement de spectre « Aliasing en anglais », ce qui rend l'opération d'échantillonnage irréversible.

La relation A1.3 nous a montré que le spectre de la fonction échantillonnée est constitué de la somme des répétitions périodiques du spectre du signal analogique. Si la fréquence d'échantillonnage est petite, ces répétitions peuvent se chevaucher en les sommant comme le montre l'exemple de la Figure A1.6.

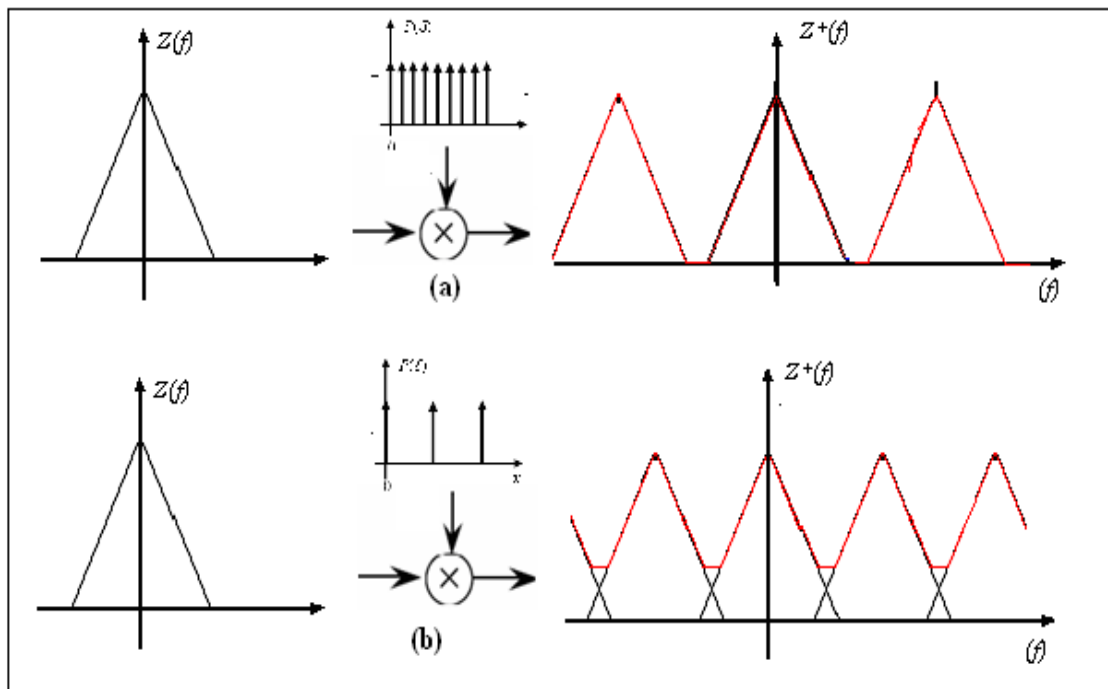


Figure A1.6 Phénomène de recouvrement
 (a) Echantillonnage sans recouvrement (b) Echantillonnage avec recouvrement

- **Notation**

- nous appelons une fonction élémentaire Z_{el} la plus petite fonction composante de la fonction Z^* tel que :

$$Z^*(f) = \sum_{m=-\infty}^{m=+\infty} Z_{el}(f - mf)$$

Théorème d'échantillonnage

La reconstruction du signal à partir de son signal échantillonné peut être possible en exploitant la périodicité du spectre exprimée par la relation A1.3. Vue la dualité existante entre le domaine temporel (spatial) et le domaine fréquentiel (spectral), chaque signal réel ayant une évolution spatial $z(x)$ a une seule représentation fréquentielle $Z(f)$.

Par ailleurs, si $z(x)$ est un signal de fréquence de coupure f_0 et échantillonné avec une petite fréquence f_e , un phénomène de repliement se produira. Dans ce cas, le spectre de $Z^*(f)$ ne sera pas construit des répétitions périodiques de $Z(f)$ mais des répétitions d'une autre fonction élémentaire comme le montre l'exemple de la figure A1.7.

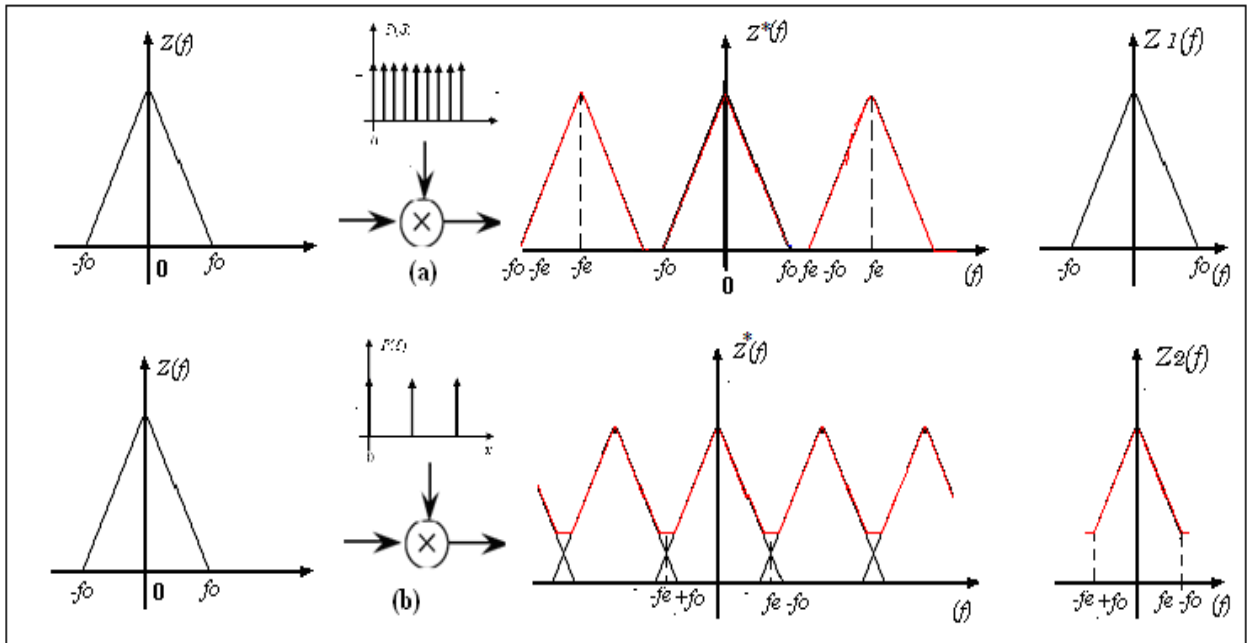


Figure A1. 7 Changement de la fonction élémentaire lors du recouvrement.

(a) Echantillonnage sans recouvrement (b) Echantillonnage avec recouvrement.

$Z_1(f)$: La fonction de base de l'échantillonnage sans le recouvrement .

$Z_2(f)$: La fonction de base de l'échantillonnage avec le recouvrement.

Le théorème de *Shannon-Nyquist* mentionne que si $z(x)$ est un signal à bande limitée tel que $Z(f)=0$ pour tout $f \geq f_0$, et si la fréquence d'échantillonnage est $f_e \geq 2f_0$ alors le signal original z peut être reconstruit à partir du signal échantillonné z^* , cette condition est considérée comme la condition nécessaire et suffisante de reconstruction.

En remarquant la fonction élémentaire du spectre du signal auto-chevauché de la figure A1.7, nous constatons l'absence de la distorsion par rapport au spectre du signal original pour les fréquences f situées entre $(-f_e + f_0)$ et $(f_e - f_0)$, autrement dit :

$$\forall f \in [(-f_e + f_0), (f_e - f_0)]: Z_2(f) = Z(f)$$

$$\Rightarrow \forall |f| \leq f_e - f_0 : Z_2(f) = Z(f).$$

Par conséquent, pour éliminer le recouvrement, le choix de la fréquence de coupure doit être inférieur à $(f_e - f_0)$ pour que $Z_2(f) = Z(f)$ pour tout f .

$$f_0 \leq f_e - f_0$$

$$\Rightarrow f_e \geq 2f_0$$

(A1.4)

On appelle la fréquence $f = 2f_0$ la fréquence limite de Shannon.

A1.5 Reconstruction idéale : interpolation de Whittaker

Le problème de reconstruction de signal analogique $z(x)$ à partir de son signal échantillonné en respectant la condition de Shannon-Nyquist $z^*(x)$ revient à déterminer la fonction élémentaire de $Z^*(f)$. Pour cela, un filtrage avec un filtre passe bas à bande limitée idéal $h(x)$ résout le problème, cette reconstruction est dite reconstruction par interpolation de Whittaker.

Soit $H(f)$ la réponse fréquentielle du filtre idéal passe bas $h(x)$ définie comme suit :

$$H(f) = \begin{cases} 1, & |f| \leq f_e/2 \\ 0, & |f| > f_e/2 \end{cases}$$

Le filtrage de $z^*(x)$ par $h(x)$ est une opération de convolution entre eux, qui se traduit par une multiplication de leurs fonctions de transferts :

$$z^*(x) \otimes h(x) \Leftrightarrow Z^*(f).H(f) \quad (\text{A1.5})$$

Nous avons de la relation (A1.3) :

$$Z^*(f) = \sum_{m=-\infty}^{m=+\infty} Z(f - mf_e)$$

D'où :

$$\begin{aligned} H(f).Z^*(f) &= H(f). \sum_{m=-\infty}^{m=+\infty} Z(f - mf_e) \\ \Rightarrow H(f).Z^*(f) &= H(f). \left[\sum_{m=-\infty}^{m=-1} Z(f - mf_e) + Z(f) + \sum_{m=1}^{m=+\infty} Z(f - mf_e) \right]. \end{aligned}$$

Pour $f_e \geq 2f_0$ nous n'avons pas de chevauchement, c'est-à-dire que les signaux $Z(f - mf_e)|_{m \in \mathbb{Z}}$ ne sont pas corrélés, et donc :

$$\Rightarrow H(f).Z^*(f) = 0. \sum_{m=-\infty}^{m=-1} Z(f - mf_e) + 1.Z(f) + 0. \sum_{m=1}^{m=+\infty} Z(f - mf_e)$$

$$\Rightarrow H(f).Z^*(f) = Z(f) \quad (\text{A1.6})$$

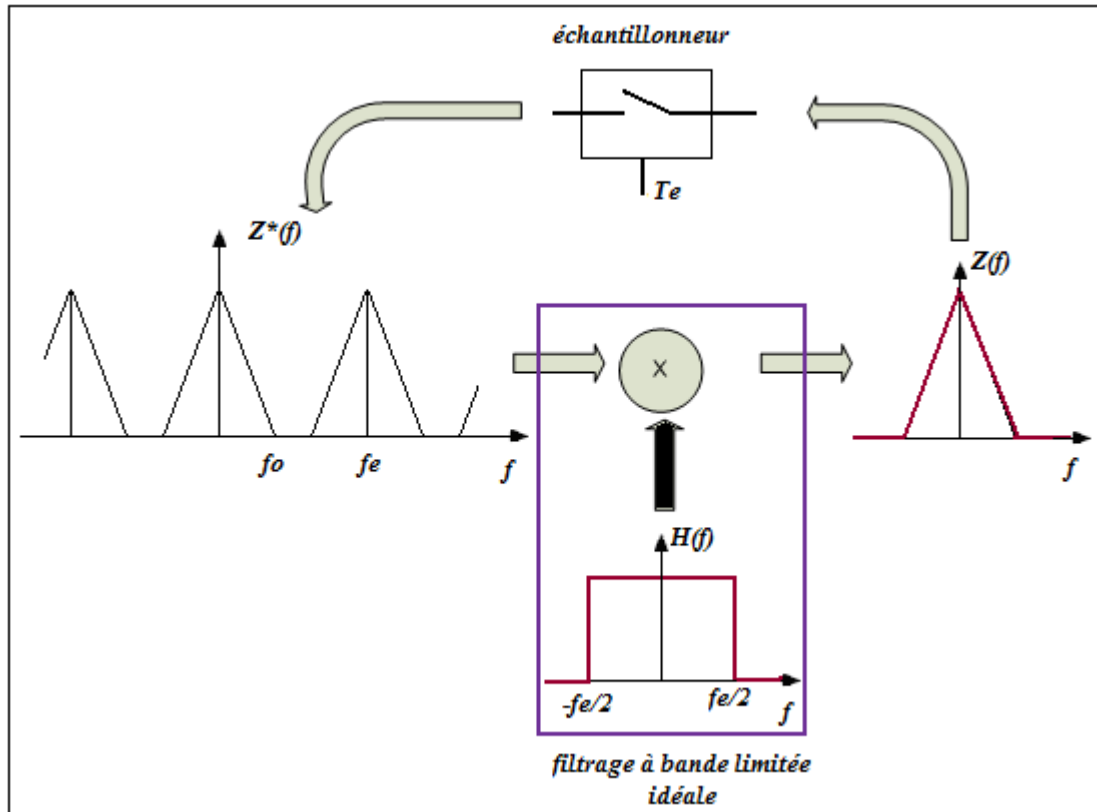


Figure A1. 8 Echantillonnage et reconstruction par filtre à bande limitée idéal.

$Z^*(f)$: spectre du signal échantillonné.

$H(f)$: réponse fréquentielle du filtre idéal utilisé.

$Z(f)$: spectre du signal analogique reconstruit par filtrage à bande limitée idéal.

Sur le plan spatial, de (A1.5) et (A1.6) on aura :

$$z(x) = h(x) \otimes z^*(x) \quad (\text{A1.7})$$

Où $h(x)$ est la réponse impulsionnelle du filtre $H(f)$

On a :

$$h(x) = TF^{-1}(H(f))$$

$$\Rightarrow h(x) = \frac{\sin(\pi f_e x)}{\pi f_e x} \quad (\text{A1.8})$$

D'où :

$$z(x) = h(x) \otimes z^*(x)$$

$$\Rightarrow z(x) = h(x) \otimes \sum_{k=-\infty}^{\infty} z^*(kx_e) \cdot \delta(x - kx_e)$$

$$\Rightarrow z(x) = \sum_{k=-\infty}^{\infty} z^*(kx_e) \cdot h(x - kx_e)$$

$$\Rightarrow z(x) = \sum_{k=-\infty}^{\infty} z^*(kx_e) \cdot \frac{\sin(\pi f_e (x - kx_e))}{\pi f_e (x - kx_e)} \quad (\text{A1.9})$$

Nous remarquons que le filtre $h(x)$ s'annule pour des valeurs de x non nulles multiples de la période d'échantillonnage x_e : $\forall k \in (\mathbb{Z} - \{0\}) : h(kx_e) = 0$.

En reproduisant au niveau de chaque échantillon la réponse impulsionnelle du filtre $h(x)$, qui s'annule au niveau des autres échantillons, et qui laisse donc inchangées leurs valeurs, on détermine les valeurs intermédiaires exactes pour reconstruire le signal analogique comme l'illustre la figure A1.9.

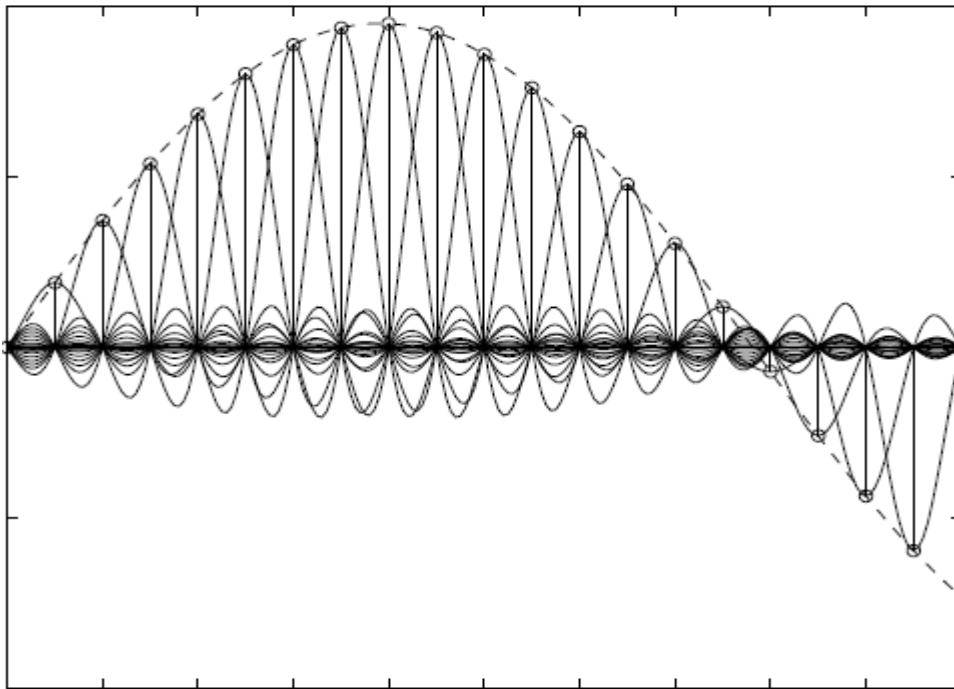


Figure A1. 9 Reconstruction par interpolation de Whittaker.

Cette interpolation idéale est, malheureusement, purement théorique et ne peut être appliquée réellement pour les deux raisons suivantes :

- Ce système n'est pas causal puisque sa réponse impulsionnelle est définie aussi dans l'intervalle $]-\infty, 0[$, donc l'effet de chaque échantillon existe même avant sa production.
- L'effet de chaque échantillon dure infiniment, conséquence de la définition de la réponse impulsionnelle sur $]0, +\infty[$.

Par ailleurs, l'utilisation d'un filtre $h'(x)$ tronqué va engendrer l'apparition des ondulations de Gibbs au niveau des points de discontinuité. En effet, avec la présence de la troncature, les valeurs de ces points ne seront pas construites avec toutes les harmoniques nécessaires pour reconstruire le signal, ce qui produira des erreurs sous forme d'ondulations.

D'autres algorithmes de reconstruction sont plus ou moins illustrés et utilisés suivant le taux d'erreurs acceptable, la puissance de calcul disponible et la complexité de traitement autorisée.

La reconstruction pratique la plus simple et la plus utilisée est la reconstruction par l'interpolation d'ordre zéro.

A1.6 Interpolation d'ordre zéro

En pratique, la reconstruction la plus utilisée pour la représentation du signal est faite, généralement, par un interpolateur d'ordre zéro, qui n'est rien d'autre qu'une mémoire ayant pour rôle de maintenir la valeur d'échantillon pour une durée égale à une période d'échantillonnage x_e .

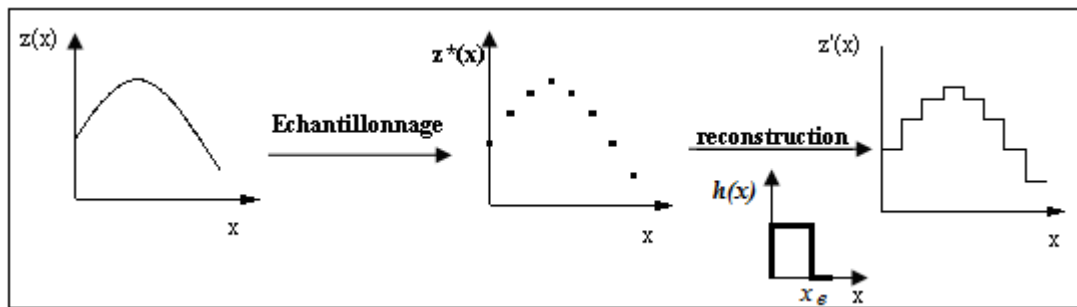


Figure A1. 10 Reconstruction par interpolation d'ordre zéro.

$z(x)$ signal analogique original. $z^*(x)$ signal échantillonné $z'(x)$ signal reconstruit.
 $h(x)$ réponse impulsionnelle du filtre d'interpolation d'ordre zéro.

Le modèle mathématique proposé pour une telle interpolation est le filtrage avec un filtre $h(x)$ dont la réponse impulsionnelle est une fenêtre rectangulaire de durée x_e :

$$h(x) = \begin{cases} 1 & 0 \leq x \leq x_e \\ 0 & \text{si non} \end{cases} .$$

On a :

$$z'(x) = h(x) \otimes z^*(x) \quad (\text{A1.10})$$

$$\Rightarrow z'(x) = h(x) \sum_{k=-\infty}^{\infty} z^*(x) \cdot \delta(x - kx_e)$$

$$\Rightarrow z'(x) = \sum_{k=-\infty}^{\infty} z^*(x) \cdot h(x - kx_e) \quad (\text{A1.11})$$

$$\Rightarrow \forall k \in Z, \forall x \in [kx_e, (k+1)x_e[: z'(x) = z^*(kx_e) \quad (\text{A1.12})$$

Sur le domaine spectral (A1.10) s'écrira :

$$Z'(f) = H(f) \cdot Z^*(f) \quad (\text{A1.12})$$

avec

$$H(f) = TF(h(f))$$

$$\Rightarrow H(f) = e^{-j\pi \cdot f \cdot x_e} \cdot \frac{\sin(\pi \cdot f \cdot x_e)}{\pi \cdot f \cdot x_e}$$

D'où

$$Z'(f) = Z^*(f) \cdot e^{-j\pi \cdot f \cdot x_e} \cdot \frac{\sin(\pi \cdot f \cdot x_e)}{\pi \cdot f \cdot x_e}$$

$$Z'(f) = \sum_{m=-\infty}^{m=+\infty} Z(f - mf_e) \cdot e^{-j\pi \cdot f \cdot x_e} \cdot \frac{\sin(\pi \cdot f \cdot x_e)}{\pi \cdot f \cdot x_e}. \quad (\text{A1.13})$$

Après l'étude de la fonction de transfert du filtre bloqueur $H(f)$, nous remarquons que la forme du premier lobe a une zone approximativement plate au voisinage du zéro, ce qui explique la faible distorsion de $Z'(f)$ au niveau de très faibles fréquences. La largeur de cette zone est proportionnelle à la largeur du premier lobe d'où le choix de la fréquence d'échantillonnage f_e : plus f_e est grande plus cette zone est large donc une distorsion plus faible et une reconstruction meilleure. Pour une fréquence d'échantillonnage importante, cette zone peut couvrir toute la fréquence de coupure f_0 jusqu'à obtenir une tendance de $Z'(f)$ vers $Z(f)$.

La Figure A1.11 montre une comparaison entre deux signaux reconstruits à partir de deux signaux échantillonnés avec des fréquences d'échantillonnage différentes en respectant toujours la condition de *Shannon*. Il est clair qu'en augmentant la fréquence d'échantillonnage on minimise la distorsion donc on s'approche de plus en plus du signal analogique original.

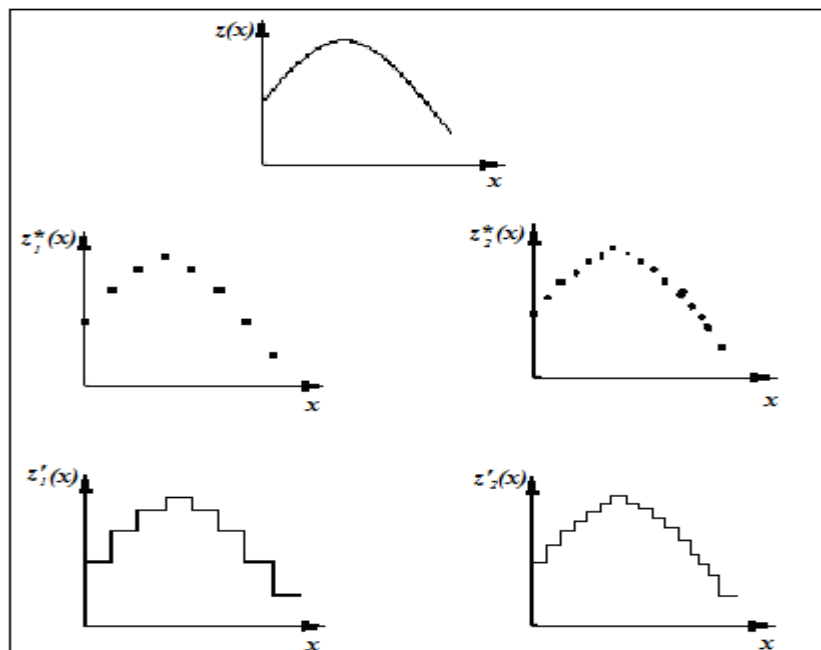


Figure A1. 11 Effet de la fréquence sur la reconstruction par interpolation d'ordre zéro

$z(x)$ signal analogique original.

z^*_1 signal échantillonné avec une fréquence f_1 .

z^*_2 signal échantillonné avec une fréquence $f_2 > f_1$.

z'_1 signal reconstruit à partir de z^*_1 .

z'_2 signal reconstruit à partir de z^*_2 .

Donc même en cas d'échantillonnage qui respecte la condition de *Shannon-Nyquist*, le signal reconstruit sera caractérisé par :

- une distorsion de la partie de $Z(f)$ entre $-f_0$ et f_0 due à la forme non parfaitement rectangulaire du filtre $H(f)$;
- une distorsion et une atténuation du signal autour des fréquences multiples de la fréquence d'échantillonnage dues à la forme de $H(f)$.

La figure A1.12 montre l'effet de cette reconstruction sur un signal échantillonné en respectant la condition de Shannon

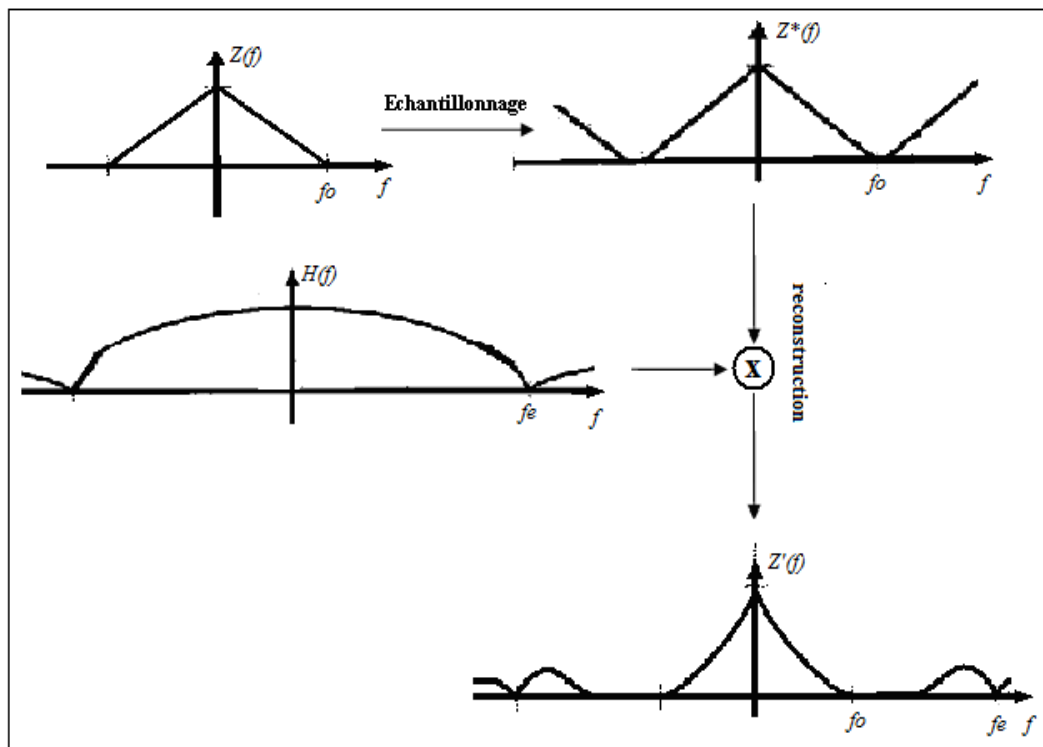


Figure A1. 12 Représentation fréquentielle de la reconstruction par interpolation d'ordre zéro.
 $Z(f)$ signal analogique original. $Z^*(f)$ signal échantillonné. $Z'(f)$ signal reconstruit.

Dans ce qui suit, nous allons appliquer la théorie d'échantillonnage sur des signaux bidimensionnels (images en 2D).

A1.7 Echantillonnage et reconstruction des images

Une image peut-être considérée comme la réalisation d'un signal aléatoire bidimensionnel $z(x,y)$ composé de deux variables indépendantes X et Y , ce signal peut être continu (le cas des scènes et des images analogiques) ou discret (cas des images numériques).

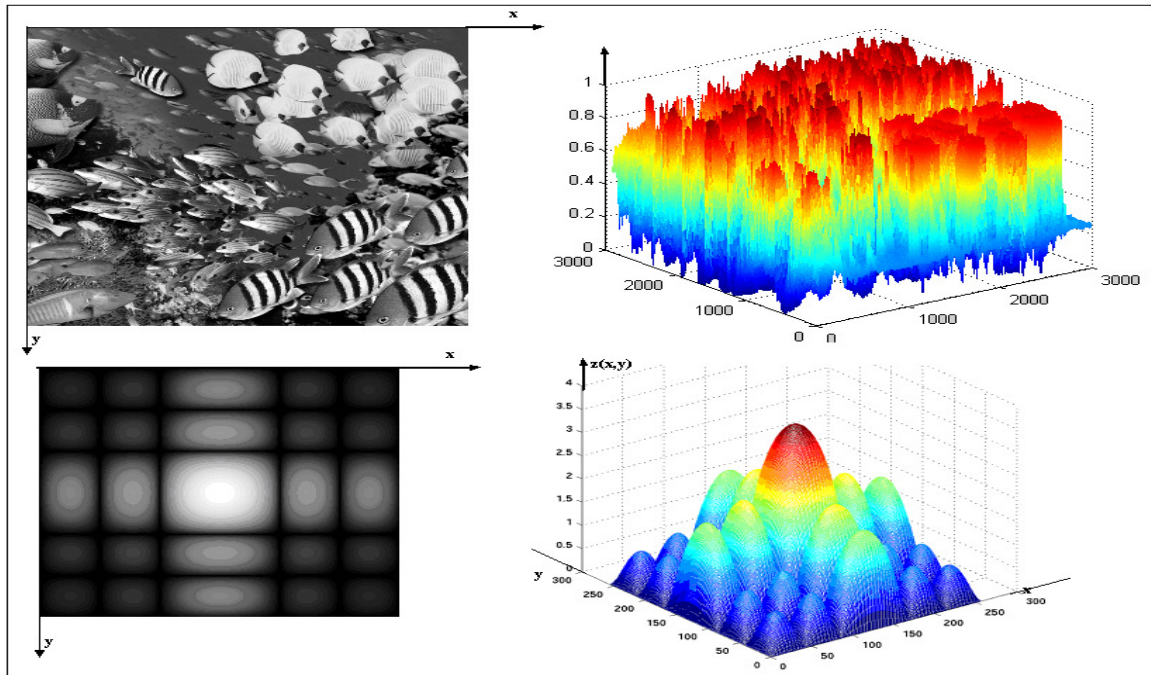


Figure A1.13 Représentation des images sous forme de signaux bidimensionnels.

Images de gauche: images optiques.

Images de droite: représentations mathématiques des images de gauche.

Soit $z(x,y)$ une image analogique. $z^*(x,y)$ est dite image échantillonnée de $z(x,y)$ avec un pas x_e sur l'axe (X) et y_e sur l'axe (Y), lorsqu'elle est déterminée par un double échantillonnage de $z(x,y)$: une fois par rapport à l'axe (X) avec une période de x_e et une autre fois sur l'autre axe (Y) avec un pas de y_e .

D'où, et par analogie avec la relation (A1.1) on aura:

$$z^*(x,y) = z(x,y) \sum_{k=-\infty}^{k=+\infty} \delta(x - kx_e) \sum_{k'=-\infty}^{k'=+\infty} \delta(y - k'y_e). \quad (\text{A1.14})$$

Puisque les variables X et Y sont indépendantes :

$$(2.14) \Rightarrow z^*(x,y) = z(x,y) \sum_{k=-\infty}^{k=+\infty} \sum_{k'=-\infty}^{k'=+\infty} \delta(y - k'y_e) \delta(x - kx_e).$$

$$z^*(x,y) = z(x,y) \sum_{k=-\infty}^{k=+\infty} \sum_{k'=-\infty}^{k'=+\infty} \delta(x - x_e k, y - k'y_e). \quad (\text{A1.15})$$

$$\text{Où : } \delta(x,y) = \begin{cases} 1 & x=0 \wedge y=0 \\ 0 & x \neq 0 \vee y \neq 0 \end{cases}$$

Nous pouvons démontrer en suivant la même procédure de la démonstration (A1.3) que:

$$(2.15) \Rightarrow Z^*(fx, fy) = \sum_{m=-\infty}^{m=+\infty} \sum_{m'=-\infty}^{m'=+\infty} Z(fx - mf_{ex}, fy - m'f_{ey}) \quad . \quad (A1.16)$$

Où $\begin{cases} f_{ex} = 1/x_e \\ f_{ey} = 1/y_e \end{cases}$ sont les fréquences d'échantillonnage selon les axes Fx et Fy.

La relation (A1.16) montre que l'échantillonnage des images produit une périodicité de leurs spectres avec une période de f_{ex} sur l'axe Fx et avec une période de f_{ey} sur l'axe Fy

Cette propriété de périodicité nous permet, théoriquement, la reconstruction de l'image originale à partir d'une autre image discrète (échantillonnée) de la même scène, à condition que cette dernière soit échantillonnée en respectant le théorème de *Shannon-Nyquist* afin d'éviter le problème de repliement du spectre

Dans le cas des images (signaux bidimensionnels), la condition de *Shannon-Nyquist* montre que la reconstruction de l'image analogique originale, à bande limitée de fréquences de coupure f_{x0} et f_{y0} , à partir de son image échantillonnée à un couple de fréquence (f_{ex}, f_{ey}) n'est possible qu'en absence de chevauchement du spectre. Pour cela, et de la même manière que dans la relation (A1.4), on peut démontrer que pour éviter le chevauchement il faut seulement que :

$$\begin{cases} f_{ex} \geq 2f_{x0} \\ f_{ey} \geq 2f_{y0} \end{cases} \quad . \quad (A1.17)$$

La Figure A1.14 montre la périodicité de spectre des images échantillonnées et l'effet du choix du couple des fréquences d'échantillonnage sur l'apparition du repliement du spectre : le spectre Z_1 est obtenu en échantillonnant l'image analogique de spectre Z , en respectant la condition de *Shannon-Nyquist*, contrairement au spectre Z_2 obtenu en échantillonnant la même image analogique sans respecter la condition de *Shannon-Nyquist*. Nous remarquons l'absence du repliement dans le spectre Z_1 et l'apparition de ce phénomène dans Z_2 .

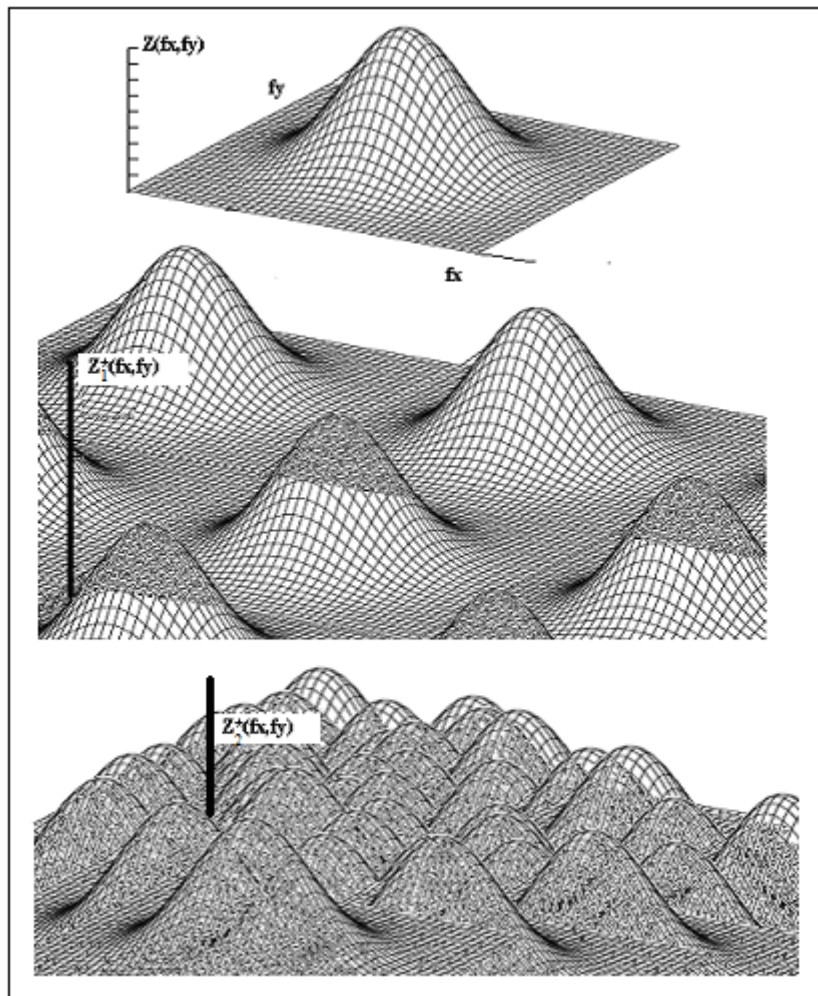


Figure A1. 14 Périodicité du spectre de l'image échantillonnée.

$Z(f_x, f_y)$: spectre de l'image analogique.

$Z_1^*(f_x, f_y)$: spectre de son image échantillonnée sans repliement.

$Z_2^*(f_x, f_y)$: spectre de son image échantillonnée avec repliement.

La Figure A1.15 représente deux images échantillonnées de la même image initiale, nous remarquons que l'image échantillonnée avec le facteur 2X2 ne présente pas de chevauchement contrairement à celle échantillonnée avec un facteur de 8X8.

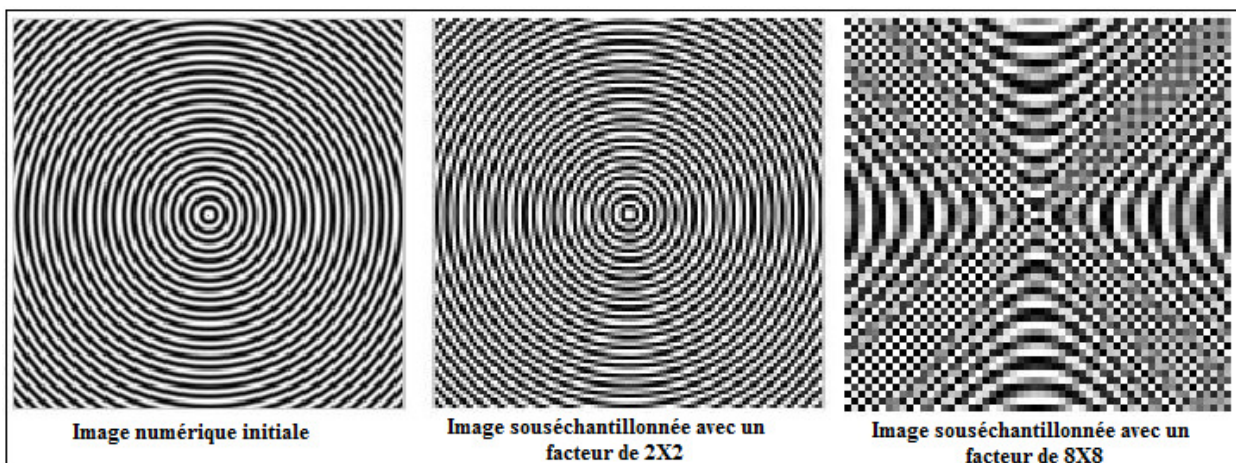


Figure A1. 15 Image échantillonnée sans et avec repliement.

Même en l'absence du repliement, la reconstruction idéale de l'image originale reste purement théorique en utilisant une interpolation de *Whittaker*. Cette interpolation est réalisée par un filtrage de l'image échantillonnée avec un filtre dont la fonction de transfert est un parallélépipède parfait défini comme suit :

$$H(fx, fy) = \begin{cases} 1, & |fx| \leq fe_x / 2 \wedge |fy| \leq fe_y / 2 \\ 0, & |fx| > fe_x / 2 \vee |fy| > fe_y / 2 \end{cases} \quad (\text{A1.18})$$

D'où sa réponse impulsionnelle est

$$h(x) = xe.ye. \frac{\sin(\pi fe_x x)}{\pi x} \cdot \frac{\sin(\pi fe_y y)}{\pi y} \quad (\text{A1.19})$$

Cette reconstruction reste purement théorique : en effet, le filtre de *Whittaker* ne peut être implémenté à cause de non causalité du système et de l'influence permanente de chaque échantillon sur l'image originale.

En pratique, comme pour les signaux monodimensionnels, la reconstruction la plus utilisée pour représenter l'image échantillonnée est la reconstruction interpolation d'ordre zéro définie physiquement par les pixels. Chaque pixel maintient la valeur de chaque échantillon sur une surface élémentaire d'échantillonnage $x_e \times y_e$.

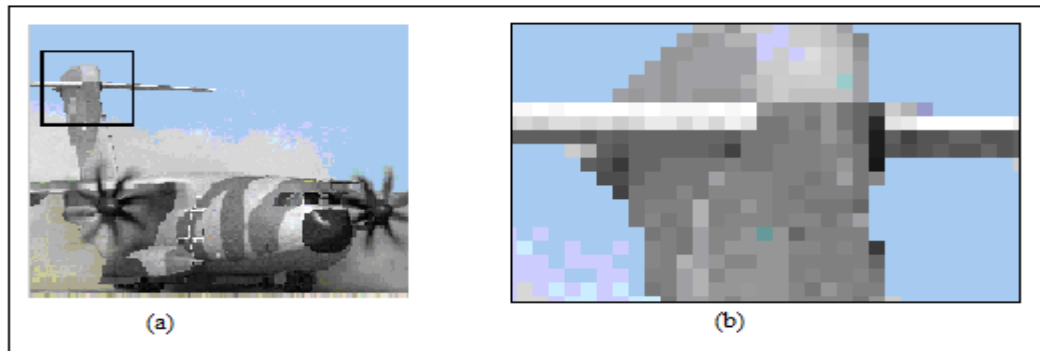


Figure A1.16 Représentation des pixels dans l'image échantillonnée.

(a) une image échantillonnée (b) représentation des pixels construisant la zone encadrée de (a).

Le modèle mathématique de cette reconstruction et par analogie avec la relation (A1.11) est déterminée par $z'(x,y)$ tel que :

$$\begin{aligned} z'(x, y) &= z^*(x, y) \otimes h(x, y) \\ \Rightarrow z'(x, y) &= \sum_{m=-\infty}^{\infty} \sum_{m'=-\infty}^{\infty} z^*(x, y) \cdot h(x - m.xe, y - m'.ye) \end{aligned} \quad (\text{A1.20})$$

Avec :

$$h(x, y) = \begin{cases} 1, & 0 \leq x \leq x_e \quad \wedge \quad 0 \leq y \leq y_e \\ 0 & \text{ailleurs} \end{cases}$$

Qui s'écrit dans le domaine fréquentiel comme suit :

$$Z'(fx, fy) = H(fx, fy).Z^*(fx, fy) \quad (\text{A1.21})$$

Tel que

$$H(fx, fy) = e^{-j\pi \cdot f \cdot x_e} \cdot e^{-j\pi \cdot f \cdot y_e} \cdot \frac{\sin(\pi \cdot f_x \cdot x_e)}{\pi \cdot f_x \cdot x_e} \cdot \frac{\sin(\pi \cdot f_y \cdot y_e)}{\pi \cdot f_y \cdot y_e} \quad (\text{A1.22})$$

Donc et à partir de la relation (A1.22), nous constatons que la représentation de l'image par l'interpolation d'ordre zéro présente des distorsions et des atténuations sur la forme originale de l'image. La distorsion est faible pour les basses fréquences à cause de la surface plate du $H(f_x, f_y)$ et devient de plus en plus importante en augmentant ces fréquences (voir la Figure (A1.12)). Cette surface plate augmente avec l'augmentation des fréquences d'échantillonnage : pour f_{x_e}, f_{y_e} très importantes, $H(f_x, f_y)$ tend vers un filtre de *Whittaker* idéal pour des fréquences inférieures aux fréquences de coupures, d'où la tendance vers une reconstruction idéale.

Par ailleurs, nous pouvons dire que pour avoir une meilleure reconstruction d'ordre zéro, on doit augmenter le nombre des échantillons par l'opération de suréchantillonnage qui minimise le taux d'erreurs entre les valeurs réelles et les valeurs ajoutées.

A1.8 Suréchantillonnage et Souséchantillonnage

Une image échantillonnée peut être représentée par une matrice tel que les colonnes représentent l'évolution sur l'axe X et les lignes représentent l'évolution sur l'axe Y. la notation $i(x, y)$ signifie la représentation matricielle de l'image échantillonnée.

On appelle une opération du suréchantillonnage l'opération d'ajout des échantillons à une image afin d'avoir une distorsion minimale et d'augmenter sa résolution. L'opération inverse est dite une opération de souséchantillonnage.

On peut, donc, dire qu'une opération de suréchantillonnage d'un facteur $K_x \times K_y$ consiste à effectuer une interpolation sur une image $i(x, y)$ en estimant les $(K_x - 1) \times (K_y - 1)$ valeurs intermédiaires entre les points adjacents afin d'avoir une image suréchantillonnée $i_u(x, y)$.

De même, on définit l'opération de souséchantillonnage de l'image échantillonnée $i(x, y)$ à la fréquence (f_{ex}, f_{ey}) d'un facteur $K_x \times K_y$ le calcul d'une image de la même scène échantillonnée avec une fréquence $(f_{ex}/K_x, f_{ey}/K_y)$ de l'image analogique originale, l'image résultante est notée par $i_d(x, y)$.

Ces deux opérations sont des opérations naturelles pour la vision humaine, En effet, lorsque l'œil, qui a une certaine résolution, capte une image numérique, la résolution de l'œil peut être différente de la résolution de cette image : si elle est inférieure le cerveau fait une opération de suréchantillonnage, sinon il fait une opération de souséchantillonnage.

D'après le théorème d'échantillonnage, l'opération de suréchantillonnage d'un signal n'est pas aussi simple que le souséchantillonnage : en effet, le passage d'une image $i(x,y)$ de dimension $N \times M$ vers une image souséchantillonnée $i_d(x,y)$ de dimension inférieure $N_d \times M_d$ se fait par un double produit matriciel en utilisant des matrices de passage P_1 et P_2 :

$$i_d(x,y) = P_1.i(x,y).P_2 \quad (\text{A1.23})$$

Où P_1 et P_2 sont des matrices de dimensions $N_d \times N$ et $M \times M_d$ tel que

$$P_1 = \begin{cases} 1 & x = y \\ 0 & \text{si non} \end{cases} \text{ et } P_2 = \begin{cases} 1 & x = y \\ 0 & \text{si non} \end{cases}$$

nous remarquons que lors d'une opération de souséchantillonnage, les matrices de passage ne sont pas toutes les deux carrées, donc l'opération de suréchantillonnage parfait, n'est pas possible puisque l'inverse d'une matrice non carrée n'existe pas, et donc la reconstruction idéale de $i(x,y)$ n'est pas évidente.

A1.9 Conclusion

Nous avons exposé, dans ce chapitre, les principaux éléments de la théorie d'échantillonnage et de la reconstruction des signaux monodimensionnels et bidimensionnels en se basant sur la condition d'échantillonnage de Shannon-Nyquist, ainsi que les limites de la reconstruction théorique par filtrage de Whittaker.

Dans le chapitre suivant, nous allons présenter différentes techniques pratiques de super-résolution monoframe par interpolation afin d'effectuer le suréchantillonnage des images basse résolution en produisant un minimum de distorsion.

Annexe II

DETECTION DES CONTOURS ET BINARISATION DES IMAGES

Nous avons exposé dans cette annexe les différents algorithmes de détection des contours et de binarisation utilisée dans l'application ENSP SUPERRESOLUTION.

A2.1 Détection des contours

A2.1.1 La superrésolution et les contours

Un signal de haute fréquence est un signal qui présente des variations rapides de l'amplitude. Dans le cas des images, les hautes fréquences se traduisent par une variation rapide, un saut ou une discontinuité au niveau de l'intensité de l'image.

Nous avons observé durant notre travail sur les algorithmes de superrésolution que plus de 75% des erreurs produites lors de l'application des algorithmes de superrésolution sont situées au niveau des contours : En effet, les fonctions et les filtres d'interpolation utilisés sont des filtres passe bas qui atténuent (voire éliminent) certains signaux HF de l'image d'où la production des erreurs d'interpolation.

La Figure A2.1 compare deux images *image1* et *image2* où *l'image2* est obtenue par l'interpolation d'une image souséchantillonnée de *l'image1* avec un facteur 4X4 .

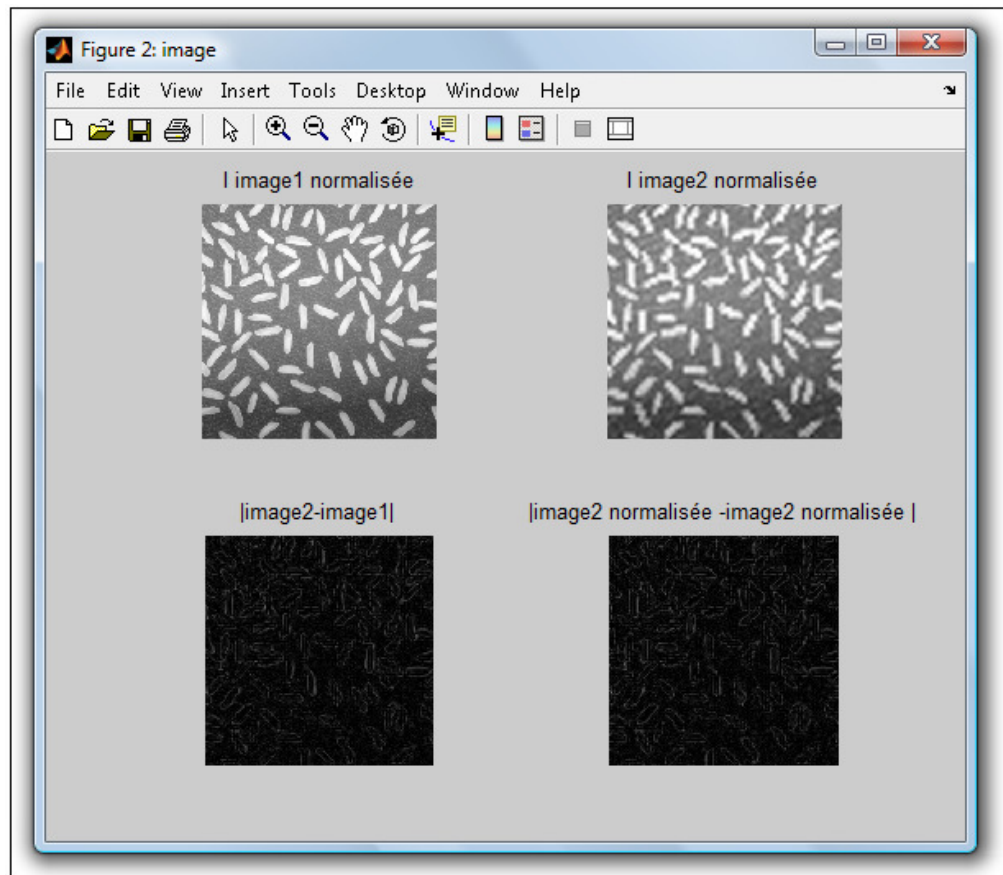


Figure A2.1 Différence entre une image de référence et une image obtenue par superrésolution.

On définit un contour comme une discontinuité dans la variation de l'intensité d'une image. On peut trouver deux types de contours : le premier où les contours se situent entre des pixels appartenant à deux régions ayant des intensités moyennes différentes, il s'agit des contours avec saut d'amplitude, le second correspond à une variation locale d'intensité présentant un maximum ou un minimum, c'est le cas des contours en toit.

Un contour est généralement caractérisé par :

- son épaisseur qui doit être la plus fine possible,
- La continuité du contour du même objet.

Plusieurs approches existent pour la détection de contours ; citons à titre d'exemple, l'approche surfacique, l'approche morphologique, l'approche dérivative et les approches basées sur les méthodes markoviennes, les réseaux de neurones ou les ondelettes.

L'approche dérivative est l'approche la plus immédiate pour détecter et localiser les variations du signal. Les contours sont assimilés aux points de fort gradient ou de dérivée seconde nulle. Pour cela nous avons choisi d'utiliser cette approche pour la détection des contours.

A2.1.2 Détection du contour par dérivation

La Figure A2.2 représente la réaction des première et deuxième dérivées de l'image au niveau du contour : nous remarquons que la première dérivée atteint un extremum local à ce niveau, ce qui se traduit par l'annulation de la dérivée seconde.

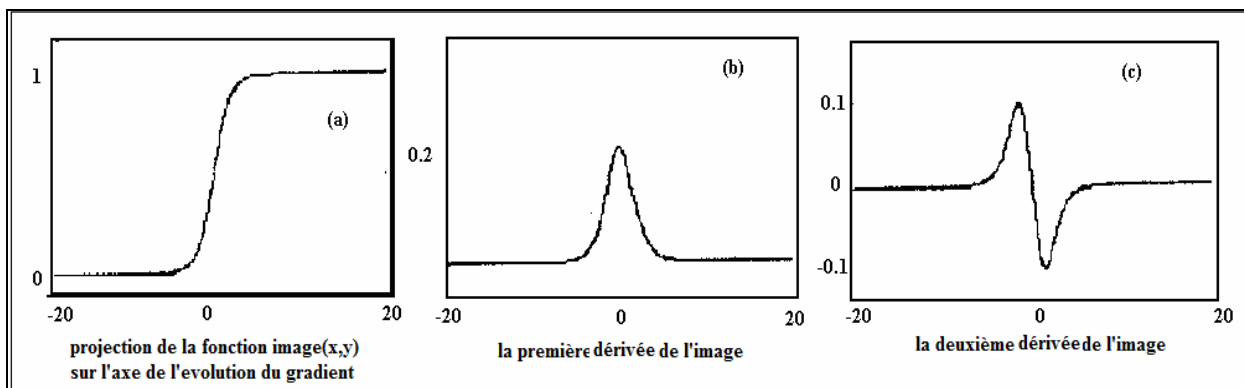


Figure A2.2 Réactions de la première et de la deuxième dérivées de l'image au niveau du contour

D'où la détection du contour peut être faite par la localisation des extremums de la première dérivée « approche gradient » ou par le passage par zéro de la deuxième dérivée de l'image « approche Laplacien ». Nous avons choisi, à ce niveau, d'utiliser l'approche gradient dans notre application.

Le gradient est un vecteur qui détermine la variation d'une fonction de plusieurs variables. Dans le cas d'une image, $i(x,y)$ on définit son gradient comme :

$$\overrightarrow{grad}(i(x,y)) = \begin{pmatrix} \frac{\partial i(x,y)}{\partial x} \\ \frac{\partial i(x,y)}{\partial y} \end{pmatrix} \quad (A2.1)$$

La Figure A2.3 montre la représentation des vecteurs gradients sur deux images à niveaux de gris.

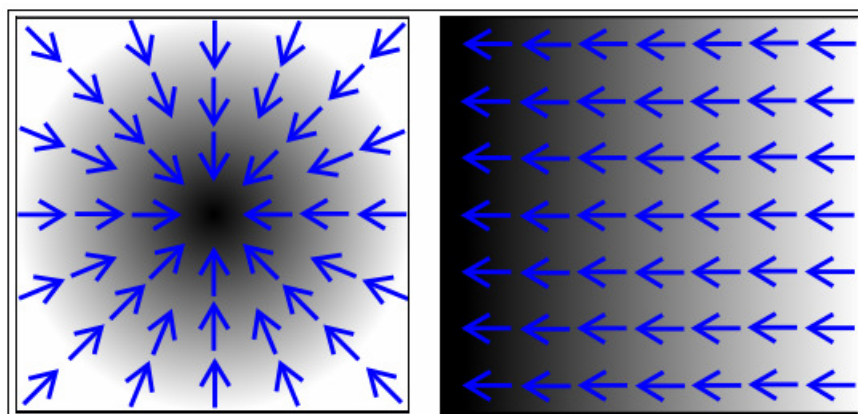


Figure A2.3 Représentation du gradient sur deux images à niveau de gris

Soit $i(x_0, y_0)$ l'intensité du pixel $p(x_0, y_0)$ de l'image i , l'orientation du gradient en ce point s'écrit :

$$\phi(x_0, y_0) = \arctan\left(\frac{\partial i(x_0, y_0)}{\partial y} / \frac{\partial i(x_0, y_0)}{\partial x}\right) \quad (\text{A2.2})$$

Et le module du gradient peut être défini comme :

$$\left| \overrightarrow{\text{grad}}(i(x_0, y_0)) \right| = \sqrt{\left(\frac{\partial i(x_0, y_0)}{\partial y}\right)^2 + \left(\frac{\partial i(x_0, y_0)}{\partial x}\right)^2} \quad (\text{A2.3})$$

Ou défini comme :

$$\left| \overrightarrow{\text{grad}}(i(x_0, y_0)) \right| = \max\left\{\left|\frac{\partial i(x_0, y_0)}{\partial y}\right|, \left|\frac{\partial i(x_0, y_0)}{\partial x}\right|\right\} \quad (\text{A2.4})$$

Dans le cas des images numériques, les dérivées partielles de l'image peuvent être définies par :

$$\begin{cases} \frac{\partial i(x_0, y_0)}{\partial x} \approx \frac{\Delta i}{\Delta x} = i(x_0 + 1, y_0) - i(x_0 - 1, y_0) \\ \frac{\partial i(x_0, y_0)}{\partial y} \approx \frac{\Delta i}{\Delta y} = i(x_0, y_0 + 1) - i(x_0, y_0 - 1) \end{cases} \quad (\text{A2.5})$$

Ou par les formules :

$$\begin{cases} \frac{\partial i(x_0, y_0)}{\partial x} \approx \frac{\Delta i}{\Delta x} = i(x_0 + 1, y_0) - i(x_0 - 1, y_0) \\ \frac{\partial i(x_0, y_0)}{\partial y} \approx \frac{\Delta i}{\Delta y} = i(x_0, y_0 + 1) - i(x_0, y_0 - 1) \end{cases} \quad (\text{A2.6})$$

Pour la relation (A2.5), nous utilisons le pixel $p(x_0, y_0)$ dans le calcul du gradient. Cela implique que le gradient ne contient pas une information sur le pixel précédent contrairement à la relation (A2.6) qui l'utilise. Cette dernière relation calcule les dérivées partielles d'une image par la simple convolution de l'image avec le filtre $h_x = [1 \ 0 \ -1]$ pour la dérivation selon l'axe des x et avec le transposé du filtre précédent $h_y = h_x^t$ pour la dérivation selon l'axe des y.

La Figure A2.4 montre le gradient calculé en utilisant les relations (A2.3) et (A2.6) :

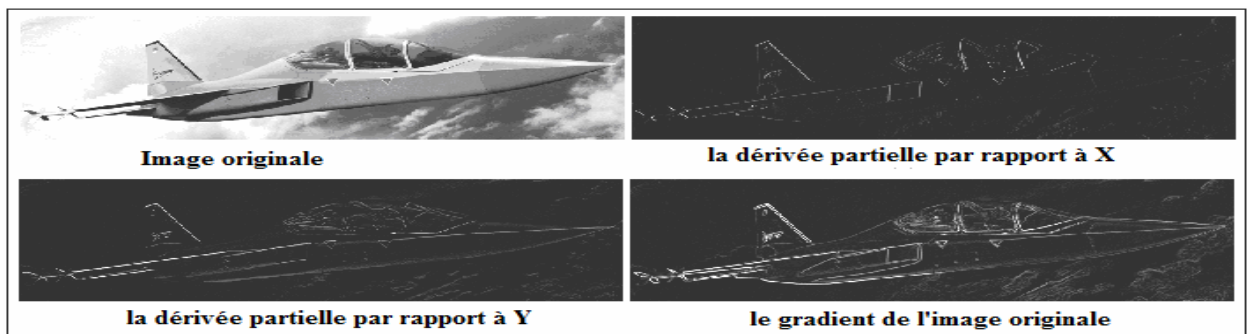


Figure A2.4 Calcul du gradient d'une image par le filtre de dérivation de la relation (6)

Le contour calculé par la dérivation directe de l'image peut présenter des discontinuités. Pour minimiser cet inconvénient, plusieurs opérateurs de calcul du gradient sont apparus citons :

♦ L'opérateur du Prewitt

Le calcul des dérivées partielles avec l'opérateur de Prewitt se fait par le filtrage de l'image avec les filtres :

$$h_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \quad (\text{A2.7})$$

$$h_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \quad (\text{A2.8})$$

où h_x représente le filtre dérivateur suivant l'axe X et h_y , le filtre dérivateur suivant l'axe Y.

Nous remarquons que chacun de ces deux filtres est composé d'une fonction de lissage $\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ et une autre fonction de dérivation $\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$ tirée de la relation (A2.6).

La Figure5 représente le gradient d'une image calculé par l'opérateur du Prewitt

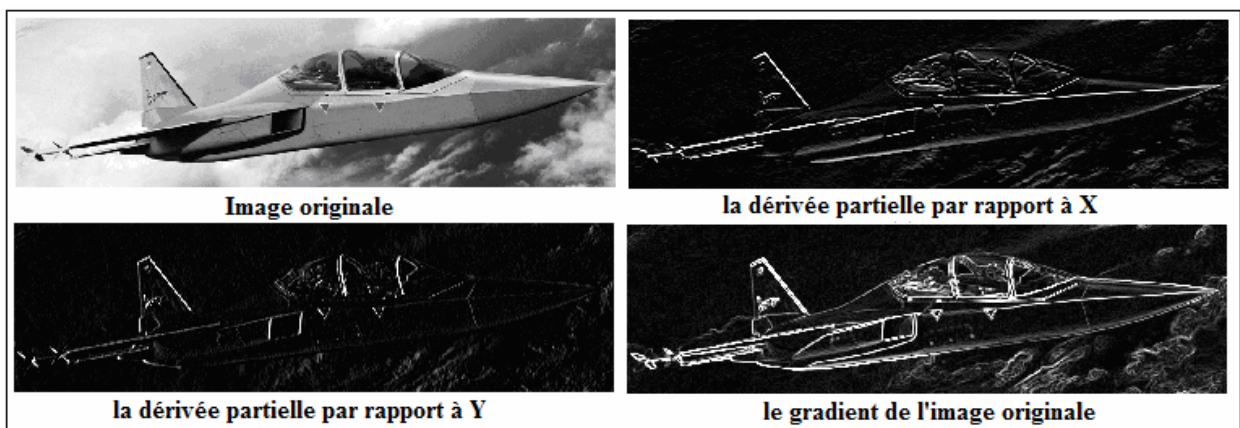


Figure A2.5 Calcul du gradient d'une image en utilisant l'opérateur de Prewitt

♦ L'opérateur de Sobel

De la même façon que l'opérateur de Prewitt, l'opérateur de Sobel est composé d'une fonction de dérivation $\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$ et d'une fonction de lissage $\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$

Les filtres dérivatifs s'écrivent comme suit :

$$h_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \quad (\text{A2.9})$$

$$h_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \quad (\text{A2.10})$$

La Figure A2.6 montre le gradient d'une image calculé par l'opérateur de Sobel

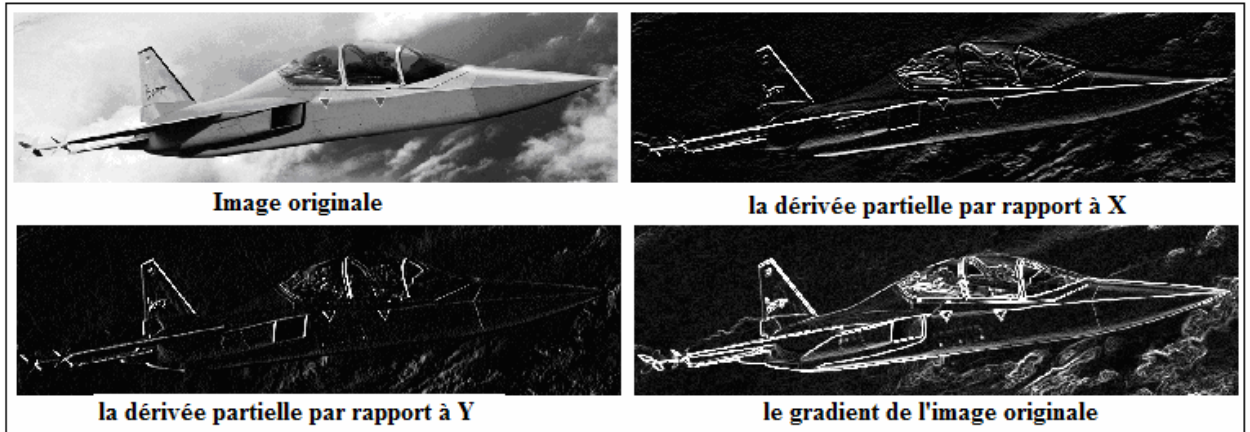


Figure A2.6 Calcul du gradient d'une image en utilisant l'opérateur de Sobel

Nous remarquons que l'opérateur de Sobel utilise une fonction de lissage plus forte, la comparaison entre les résultats des trois opérateurs de dérivation montre que le gradient calculé par l'opérateur de Sobel présente moins de discontinuité que les autres opérateurs, mais il peut produire parfois des zones floues dans l'image gradient.



Figure A2.7 Comparaison entre les trois méthodes de calcul du gradient

♦ L'opérateur MDIF

L'opérateur MDIF est un autre opérateur qui se base sur la combinaison entre le filtre dérivateur de premier ordre et une fonction de lissage. Dans ce cas, la fonction de lissage est

choisie pour : $l = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

On a donc :

$$\begin{cases} \frac{\partial i(x, y)}{\partial x} \approx h_x \otimes i(x, y) \\ \frac{\partial i(x, y)}{\partial y} \approx h_y \otimes i(x, y) \end{cases} \quad (\text{A2.11})$$

Par analogie à la relation (A2.10), on trouve que les filtres dérivatifs de l'opérateur MDIF sont définis par :

$$h_x = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 3 & 0 & -3 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 0 & 1 & 0 & -1 & 0 \end{bmatrix} \text{ et } h_y = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -3 & -2 & -1 \\ 0 & -1 & -1 & -1 & 0 \end{bmatrix} \quad (\text{A2.12})$$

La Figure A2.8 montre le gradient d'une image calculé par l'opérateur MDIF

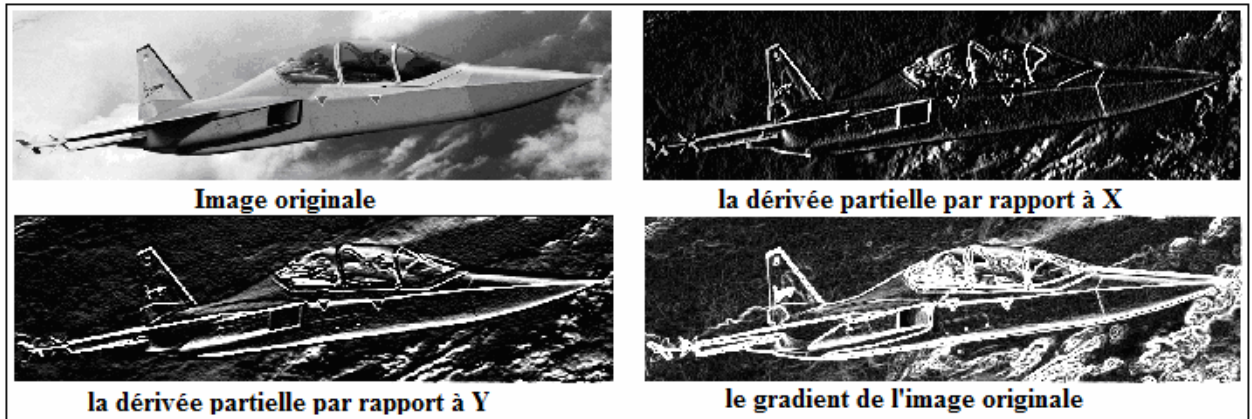


Figure A2.8 Calcul du gradient d'une image en utilisant l'opérateur MDIF

♦ L'opérateur de gradients directionnels de Kirsch

L'opérateur de gradients directionnels de Kirsch se base sur l'utilisation d'un filtre parmi les huit filtres de dérivation dont chaque masque représente la dérivation suivant une direction.

Ces filtres sont obtenus par une rotation de $\frac{\pi}{4}$ et définis par :

$$h_0 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}, h_1 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}, h_2 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}, h_3 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix},$$

$$h_4 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}, h_5 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}, h_6 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}, h_7 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

Tel que $h_0, h_1, h_2, h_3, h_4, h_5, h_6, h_7$ correspondent aux directions $0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}$.

Le gradient est donné par la maximisation de l'image filtrée par ces filtres :

$$\left| \overrightarrow{\text{grad}}(i(x, y)) \right| = \max_{\alpha} \{ |h_{\alpha} \otimes i(x, y)| \} \quad (\text{A2.13})$$

Et l'orientation du gradient est l'orientation du masque correspondant.

La Figure A2.9 représente une image gradient calculée par l'opérateur de Kirsch.

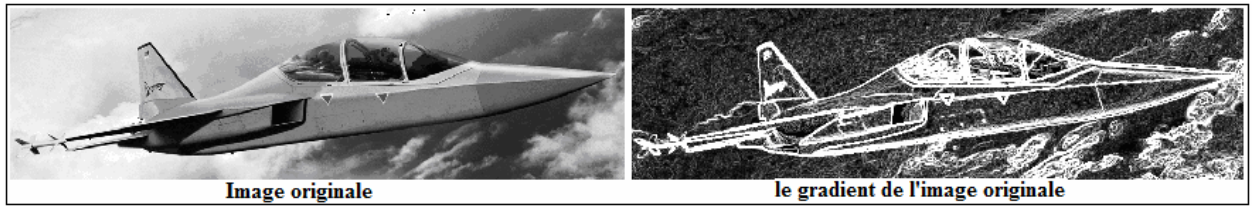


Figure A2.9 Calcul du gradient d'une image par l'opérateur de gradients directionnels de Kirsch

Nous avons remarqué que les contours calculés par les opérateurs MDIF et Kirsch assurent la continuité, mais malheureusement ces contours sont épais.

A2.2 Binarisation

La binarisation consiste à transformer une image donnée I en une image binaire Ib en comparant l'intensité de chaque pixel avec un seuil s dit seuil de binarisation :

$$Ib(x, y) = \begin{cases} 1 & \text{si } I(x, y) \geq s \\ 0 & \text{si } I(x, y) < s \end{cases}$$

Le choix du seuil de binarisation influe directement sur la quantité d'informations présentes dans l'image binaire : Afin de maximiser la quantité des informations présentes dans une image binarisée, on a développé un algorithme de calcul automatique de ce seuil dit algorithme d'Otsu. Cet algorithme minimise l'écart quadratique entre l'image en niveaux de gris et son image binaire.

Soit s le seuil de binarisation à calculer par l'algorithme d'Otsu, pour tous niveaux de gris k de l'image originale nous définissons $n(k)$ le nombre des pixels de l'image originale ayant une intensité égale à k et son image $\mu(k)$ par binarisation tel que :

$$\mu(k) = \begin{cases} \mu_{\text{inf}} & \text{si } k < s \\ \mu_{\text{sup}} & \text{si } k \geq s \end{cases}$$

Avec μ_{inf} représente la valeur moyenne de la classe U_{inf} des niveaux de gris inférieurs à s et μ_{sup} représente la valeur moyenne de la classe U_{sup} des niveaux supérieurs à s .

L'écart quadratique entre l'image en niveaux de gris et son image binaire est défini par la relation :

$$\varepsilon(I, Ib) = \sqrt{E\{(K - \mu(K))^2\}}$$

Minimiser cet écart quadratique entre les deux images revient à minimiser :

$$\sigma = E\{(K - \mu(K))^2\}$$

$$\Rightarrow \sigma = \sum_{k=1}^{s-1} n(k) \cdot (k - \mu_{\text{inf}})^2 + \sum_{k=s} n(k) \cdot (k - \mu_{\text{sup}})^2$$

$$\begin{aligned}
\sigma \text{ est minimale} &\Rightarrow \begin{cases} \frac{\partial}{\partial \mu_{\text{inf}}} \sigma(s, \mu_{\text{inf}}, \mu_{\text{sup}}) = 0 \\ \frac{\partial}{\partial \mu_{\text{sup}}} \sigma(s, \mu_{\text{inf}}, \mu_{\text{sup}}) = 0 \end{cases} \\
&\Rightarrow \begin{cases} -2 \sum_{k=1}^{s-1} n(k) \cdot (k - \mu_{\text{inf}}) = 0 \\ -2 \sum_{k=s} n(k) \cdot (k - \mu_{\text{sup}}) = 0 \end{cases} \\
&\Rightarrow \begin{cases} \mu_{\text{inf}} = \frac{\sum_{k=1}^{s-1} k \cdot n(k)}{\sum_{k=1}^{s-1} n(k)} \\ \mu_{\text{sup}} = \frac{\sum_{k=s} k \cdot n(k)}{\sum_{k=s} n(k)} \end{cases} \tag{A2.13}
\end{aligned}$$

Nous remarquons que les valeurs calculées de μ_{inf} et μ_{sup} sont en fonction de s , en les remplaçant dans σ nous trouvons :

$$\begin{aligned}
\sigma &= \sum_{k=1}^{s-1} k^2 n(k) - 2 \sum_{k=1}^{s-1} k \cdot \mu_{\text{inf}}(s) n(k) + \sum_{k=1}^{s-1} \mu_{\text{inf}}^2(s) n(k) + \sum_{k=s} k^2 n(k) - 2 \sum_{k=s} k \cdot \mu_{\text{sup}}(s) n(k) + \sum_{k=s} \mu_{\text{sup}}^2(s) n(k) \\
\sigma &= \sum_{k=1}^{s-1} k^2 n(k) - \left(\sum_{k=1}^{s-1} \mu_{\text{inf}}^2(s) n(k) + \sum_{k=s} \mu_{\text{sup}}^2(s) n(k) \right)
\end{aligned}$$

Minimiser σ revient à maximiser L tel que

$$L = \sum_{k=1}^{s-1} \mu_{\text{inf}}^2(s) n(k) + \sum_{k=s} \mu_{\text{sup}}^2(s) n(k) \tag{A2.14}$$

D'où le seuil d'Otsu est le seuil qui maximise la valeur de L définie par la relation (A2.14).