

**République algérienne démocratique et populaire**  
**Ministère de l'enseignement supérieur et de la recherche scientifique**  
**Université Saad Dahleb de Blida**  
**Faculté des sciences de l'ingénieur**  
**Département d'aéronautique**



Mémoire de fin d'études en vue de l'obtention  
du diplôme d'ingénieur d'état en Aéronautique

**Option : Installation**

**THEME**

**Réduction du bruit des images par  
filtrage spatial**

**Présenté par :**

**Mr: HADJ ABDELKADER Kheir eddine**

**Mr: BOUNOUAHINE Belhadj**

**Promotrice :**

**Melle: F.DOUDOU**

**ANNEE UNIVERSITAIRE 2007/2008**

## **Résumé**

Ce Projet de Fin des Etudes, concerne l'étude des filtres permettant l'amélioration de la qualité des images fournies afin de mieux en extraire les informations, et de mieux les interpréter. Il s'agit de mettre en évidence les techniques consistant à éliminer le problème dus aux différents bruits tout en conservant les données pertinentes de l'image.

Dans un premier temps, nous présentons une étude générale sur les images, puis nous décrivons les filtres passe-bas et les filtres passe-haut, ainsi de suite nous avons mis en œuvre les algorithmes qui représentent les équations des filtres cités précédemment et l'implémentation de ces filtres sous le langage de programmation MATLAB.

## **Abstract**

This Project of End of the Studies, relate to the study of the filters allowing the improvement of quality of the images provided in order to better extract information from them, and to better interpret them. It is a question of highlighting the techniques consisting in eliminating the problem due to different noises while preserving the relevant data of the image.

Initially, we present a general study on the image, then we describe the low-pass filters and the high-pass filters, so on we put the algorithms which represent the equations of the filters quoted previously and the implementation of these filters under the programming language MATLAB.



# REMERCIEMENTS

*Nous remercions tout d'abord dieu le tout puissant tout clément  
qui nous a aidé à élaborer ce modeste travail.*

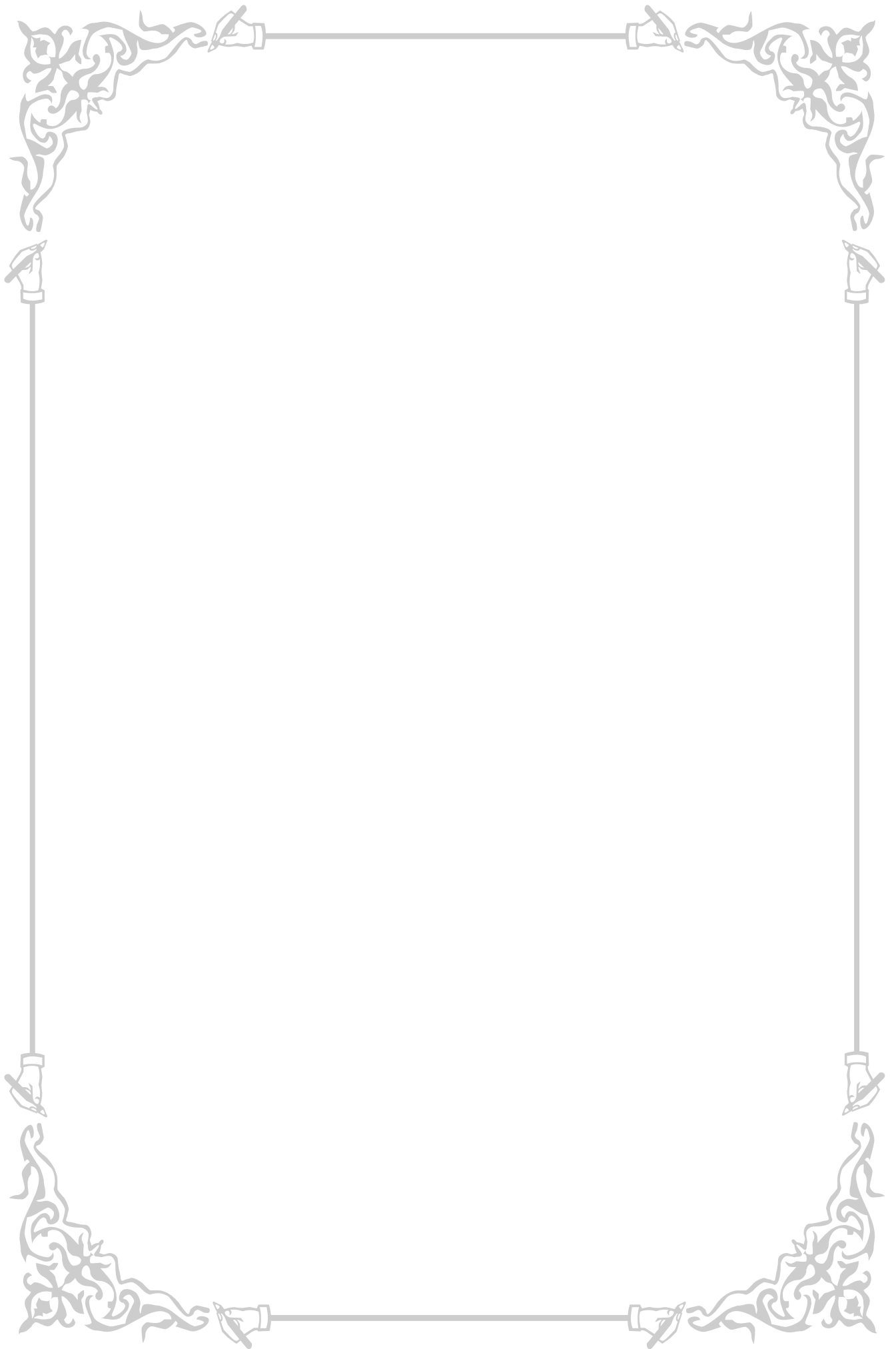
*Nous exprimons notre profonde et respectueuse gratitude*

*A notre promotrice Melle F.DOUDOU*

*Nous exprimons aussi nos remerciements les plus sincères*

*A l'ensemble des enseignants du département d'Aéronautique*

*Nous remercions également tous nos amis.*







## *Dédicace*

*Je dédie ce modeste travail, synonyme de  
concrétisation de tous mes efforts fournis pendant ces  
cinq dernières années :*

*A ma très chère mère ma raison de vivre, symboles de courage  
et de sacrifices, que dieu la garde.*

*A mon cher père.*

*A la joie et le sourire et la fleur de ma vie  
ma très chère femme.*

*A ma future petite famille.*

*A ma grande mère.*

*A mon grand père.*

*A mes frères.*

*A mes sœurs.*

*A toutes les familles Hadj abdelkader et Maameri.*

*A mon binôme Bounouahine belhadj.*

*A tous mes Amis :*

*Berrahi, Elkouf, Siouti, Hamza, Touati, Djébar, Abes,  
Boumedién, Abdelchakour, Mogtit, Cherif, Wahid,  
Mekhalfia, Abdellah, Saïf eddine, Beltach...*

*HADJ AËK Kheir eddine*









*Dédicace*

*Je dédie ce modeste travail, synonyme de  
concrétisation de tous mes efforts fournis pendant ces  
cinq dernières années :*

*A ma très chère Mère.  
A mes frères.  
A mes sœurs.*

*A toute la famille.*

*A mon binôme HADJ AËK Kheir eddine.*

*A tout mes Amis.*

*BOUNOUAHINE Belhadj*



# Sommaire

## Introduction générale

## Chapitre I : Généralités sur les images

I-1 Introduction.....	01
I-2- Etat de l'art.....	02
I-3- Image numérique.....	04
I-4- Définition et résolution.....	06
I-5- Couleurs.....	07
I-5-1- Mode RVB (Synthèse additive de la lumière).....	08
I-5-2- Mode CMJN (Synthèse soustractive de la lumière).....	08
I-5-3- Mode HSL.....	09
I-5-4- Mode CIE.....	10
I-5-5- Pseudo- Couleurs.....	11
I-6- Luminance et contraste.....	11
I-7- Transparence.....	11
I-8- Contours.....	12
I-9- Codage de la couleur.....	12
I-10- Poids d'une image.....	13
I-11- Images bitmap et images vectorielles.....	14
I-12- Compression des images.....	15
I-12-1- Compression sans pertes.....	15
I-12-1-1- Concaténation de points.....	16
I-12-1-2- Compression RLE.....	16
I-12-1-3- Codage de Huffman.....	16
I-12-1-4- Compression LZW.....	17
I-12-2- Compression avec pertes.....	18

I-13- Formats de fichiers images.....	19
I-13-1- Format BMP.....	20
I-13-2- Format PCX.....	21
I-13-3- Format GIF.....	21
I-13-4- Format PNG.....	22
I-13-5- Format TIF.....	22
I-13-6- Format JPEG.....	23
I-14- Qualité de l'image numérique.....	24
I-15- Caractéristiques statiques.....	24
I-15-1- Histogramme.....	24
I-15-2- Moyenne arithmétique.....	25
I-15-3- Variance.....	25
I-15-4- Erreur quadratique moyenne.....	26
I-16- Bruit dans une image.....	26
I-16-1- Origines du bruit.....	27
I-16-1-1- Bruits externes.....	27
I-16-1-2- Bruits internes.....	28
I-16-1-3- Bruits de numérisation.....	29
I-16-2- Modèles statistiques de bruit.....	30
I-16-2-1- Bruit additif.....	30
I-16-2-2- Bruit multiplicatif.....	30
I-16-3- Exemples de bruits.....	31
I-16-3-1- Bruit Gaussien.....	31
I-16-3-2- Bruit du chatoiement.....	32
I-16-3-3- Flou.....	32

## **Chapitre II : Filtrage spatial**

II-1- Traitement mathématique.....	33
II-2- Filtrage des images.....	35

II-2-1- Convolution.....	36
II-2-2- Filtres et Fenêtres.....	37
II-2-3- Filtres Passe-bas (Lissage).....	39
II-2-3-1- Filtre moyenne.....	39
II-2-3-2- Filtre mode.....	40
II-2-3-3- Filtre Gaussien.....	40
II-2-3-4- Filtre médian.....	41
II-2-3-5- Filtre médian à 3 états.....	42
II-2-3-6- Filtre maximum (conservative smoothing).....	43
II-2-4- Filtres Passe- haut (Accentuation).....	44
II-2-4-1- Filtre Gradient.....	45
a- Principe du Gradient.....	46
b- Filtres de PREWITT, SOBEL, ROBERTS et KIRSCH.....	47
b-1- Masques de Sobel.....	48
b-2- Masques de Roberts.....	48
b-3- Masques de Prewitt.....	48
b-4- Masques de Kirsh.....	49
b-5- Masques de Roberts pour 4 directions.....	49
b-6- Masques de Sobel pour 4 directions.....	49
b-7- Masques de Prewitt pour 4 directions.....	50
b-8- Masques de Kirsh pour 4 directions.....	50
II-2-4-2- Filtre Laplacien.....	50
II-2-4-3- Filtre compas.....	52
II-2-4-4- Filtre de Canny.....	52
II-2-4-5- Filtre de Marr Hildreth (Laplacien de la Gaussienne).....	53

### **Chapitre III : Implémentation et résultats**

III-1- Introduction.....	55
III-2- Plate forme de travail.....	55

III-3- Choix de MATLAB.....	55
III-4- Image Utilisée.....	56
III-5- Algorithmes et résultats.....	57
III-5-1- Bruitage d'image.....	57
III-5-2- Chaîne de filtrage.....	59
III-5-3- Calcul des paramètres de mesures statistiques.....	59
III-5-4- Influence de la taille de la fenêtre sur l'effet des filtres passe bas.....	61
III-5-4-1- Filtre moyenne.....	61
III-5-4-2- Filtre gaussien.....	61
III-5-4-3- Filtre médian.....	62
III-5-5- Filtrage passe bas.....	63
III-5-5-1 Famille des filtres médian.....	63
III-5-5-2- Comparaison des filtres du lissage.....	65
a- Bruit Sel et Poivre (densité=2%).....	65
b- Bruit Sel et Poivre (densité=70%).....	66
c- Bruit gaussien (variance=0.05).....	67
d- Bruit gaussien (variance=0.1).....	68
III-5-6- Filtrage passe haut (détection de contour).....	69
III-5-6-1- Filtre Gradient.....	69
III-5-6-2- Filtre Laplacien.....	70
III-5-6-3- Filtre de Canny.....	70
III-5-6-4- Filtre Laplacien de la gaussienne.....	71
III-5-7- L'effet des filtres passe-bas sur la détection des contours.....	71

## **Conclusion générale**

# Liste des figures

<b>Figure I-1</b> : Image définie sur un espace multidimensionnel.....	01
<b>Figure I-2</b> : Numérisation=Echantillonnage + Quantification.....	06
<b>Figure I-3</b> : Synthèse additive.....	08
<b>Figure I-4</b> : Synthèse soustractive.....	09
<b>Figure I-5</b> : Images vectorielle et bitmap.....	15
<b>Figure I-6</b> : Histogramme I-D.....	25
<b>Figure II-1</b> : Les phases du traitement d'une image.....	34
<b>Figure II-2</b> : Transformations de pixels.....	34
<b>Figure II-3</b> : Concept du filtrage.....	37
<b>Figure II-4</b> : Principe de la convolution (cas $DF=3$ ).....	37
<b>Figure II-5</b> : Taille d'un filtre.....	38
<b>Figure II-6</b> : Exemple d'application d'un filtre Laplacien sur une image astronomique.....	51
<b>Figure II-7</b> : Laplacien de la fonction de Gauss.....	53
<b>Figure III-1</b> : Exemple du filtrage par fenêtre $3 \times 3$ .....	57
<b>Figure III-2</b> : Bruitage d'image.....	58
<b>Figure III-3</b> : Organigramme de chaîne du filtrage.....	59
<b>Figure III-4</b> : Influence de la taille de la fenêtre sur l'effet du filtre moyenne.....	61
<b>Figure III-5</b> : Influence de la taille de la fenêtre sur l'effet du filtre gaussien.....	62
<b>Figure III-6</b> : Influence de la taille de la fenêtre sur l'effet du filtre médian.....	62
<b>Figure III-7</b> : Débruitage de bruit speckle par la famille des filtres médian.....	64
<b>Figure III-8</b> : Débruitage de bruit Sel et Poivre ( $D=2\%$ ) par les filtres du lissage.....	65
<b>Figure III-9</b> : Débruitage de bruit Sel et Poivre ( $D=70\%$ ) par les filtres du lissage.....	66
<b>Figure III-10</b> : Débruitage de bruit gaussien (variance=0.05) par les filtres du lissage.....	67
<b>Figure III-11</b> : Débruitage de bruit gaussien (variance=0.1) par les filtres du lissage.....	68
<b>Figure III-12</b> : Détection de contour par les filtres Gradient.....	69
<b>Figure III-13</b> : Détection de contour par le filtre Laplacien.....	70



<b>Figure III-14</b> : Détection de contour par le filtre de Canny.....	70
<b>Figure III-15</b> : Détection de contour par le filtre Laplacien de la gaussienne.....	71
<b>Figure III-16</b> : L'effet de filtre médian sur la détection des contours.....	72

# Liste des tableaux

<b>Tableau I-1</b> : Poids de l'image.....	13
<b>Tableau I-2</b> : Les formats de fichiers graphiques les plus utilisés.....	20
<b>Tableau I-3</b> : Exemples de bruits.....	31
<b>Tableau II-1</b> : Fenêtre 3x3.....	38
<b>Tableau II-2</b> : Les trois noyaux 3x3 de l'opérateur Laplacien.....	51

# Liste des symboles

C : contraste.

L1 et L2 : degrés de luminosité de deux zones d'une image.

H(m) : histogramme monodimensionnel.

E(f) : moyenne arithmétique.

N : nombre de pixels de l'image.

n : nombre de lignes de l'image.

P : nombre de colonnes de l'image.

Var(f) : variance.

, S : écart type.

MSE : erreur quadratique moyenne.

I(i,j), I<sub>j</sub> : image bruitée.

$\hat{R}(i, j)$  : image reconstruite ou estimée filtrée.

f(x,y), Pz(Z) : densité de probabilité.

F(x,y) : fonction de répartition.

A<sub>j</sub> : bruit additif de moyenne égale à 0.

R<sub>j</sub>, u(x) : image sans bruit.

B<sub>j</sub> : bruit multiplicatif de moyenne égale à 1.

m : moyenne.

n(x) : bruit additif Gaussien.

v(x) : image dégradée par le bruit additif gaussien.

h(x,y) : flou gaussien.

f(U) : densité de probabilité de Bruit du chatoiement.

h(x,y) : réponse impulsionnelle du filtre.

\* : loi de convolution.

f(x,y) : image originale.

g(x,y) : image modifiée.

DF : masque du filtre.

W : fenêtre carrée qui entoure le pixel que l'on étudie.

W<sub>c</sub> : poids affecté au pixel central de la fenêtre d'étude du filtre médian à 3 états.

dz,  $\nabla f(x,y)$  : Gradient.

$\|\nabla f(x,y)\|$  : amplitude du gradient.

$\emptyset$ , : direction du gradient.

$\text{mag}(\nabla f)$  : magnitude du gradient.

$W_x$  : direction (masque) horizontale.

$W_y$  : direction (masque) verticale.

$\nabla^2 f(x,y)$  : Laplacien.

$\mathcal{E}(x)$  : dérivée de la gaussienne.

$G$  : filtre.

$G(r)$  : fonction de Gauss.

$\nabla^2 G(r)$  : Laplacien de la fonction de Gauss.

$M$  : moyenne de bruit.

$V$  : variance de bruit.

$D$  : densité de bruit sel et poivre.

SNR : rapport signal sur bruit.

# **Introduction générale**

## **Introduction générale :**

Le traitement d'image issu des techniques du traitement du signal, né de l'idée et de la nécessité de remplacer l'observateur humain par la machine, a débordé de son cadre étroit pour devenir une science à part entière.

L'analyse d'image présente un intérêt dans quasiment tous les domaines où les problèmes de la recherche automatique d'information dans des images s'avère un besoin. Donnons en quelques exemples les images satellites sont devenues un formidable outil d'investigation en agronomie, en météorologie et même dans les domaines militaires. Pour la surveillance l'analyse de séquences d'images est largement utilisé pour le suivi d'objets, de personnes, la lecture labiale, la détection de changement de scènes, le suivi de cibles pour les militaires où reconnaissances de formes: reconnaissances de visages, d'empreintes digital, d'iris de l'œil, de textes, de dessins.

L'image cette collection d'informations ou les signaux provenant des capteurs ont alors été numérisés pour pouvoir être traités par l'ordinateur. A partir d'une image numérique, il convient d'extraire les informations pertinentes en regard l'application concernée, les traiter puis les interpréter. Le terme générique d'analyse d'images désigne l'ensemble de ces opérations. Parmi les différentes opérations de l'immense champ du traitement d'images nous avons retenu dans cette une approche qui à pour objectif l'amélioration des images par filtrage spatial.

La compréhension du traitement d'images commence par la compréhension de l'image, a ce titre nous avons exposé le détail relatif aux caractéristiques des images dans le premier chapitre, nous avons mis en valeur toute les information dominante relative à la compression des images pour mieux introduire les divers formats graphiques et nous avons tenté d'expliquer les mesures mathématiques associées à l'image.

Dans le chapitre deux, nous décrivons les filtres standards qu'on applique généralement aux images bruitées.

Dans le dernier chapitre, nous avons exploré l'effet de tous ces filtres sur une image aérienne, une comparaison des différents filtres montrant leurs effets a été réalisée et nous avons présenté les résultats et les diverses discussions relatives à l'implémentation de chaque filtre, l'évaluation des performances pour chaque filtre est faite à partir du critère visuel, complété et confirmé par des critères statistiques de qualité comme EQM (Erreur Quadratique Moyenne).

---

---

# **Chapitre I**

## **Généralités sur les images**

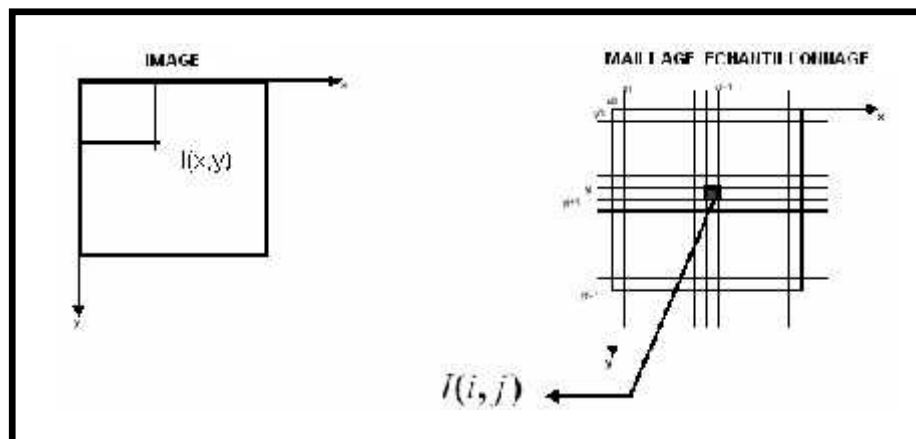
## I-1- Introduction :

A titre indicatif notons que près de 90% de l'information reçue par l'homme est visuelle. La production d'images de qualité, de même que leur traitement numérique (et si possible) automatique a donc une importance considérable. D'ailleurs, de nombreux domaines d'applications utilisent l'image comme source d'information ou de visualisation.

Une image est donc la représentation d'une scène ou d'un objet situé en général dans un espace tridimensionnel, acquise à l'aide de systèmes de production d'images (appareils photographique, caméra, satellite, scanner, sonar,.....). Sa forme peut être analogique (ex: négatif, photographie, vidéo...) ou numérique (images numérisées suivant divers formats (images compressées ou non...) ou obtenues par des capteurs fournissant des images numérisées), et dans ce cas un traitement par ordinateur est possible.

Une image peut être considérée comme une fonction  $I(X)$  définie sur un espace multidimensionnel [1].

- $X$  est un vecteur de coordonnées définissant une position dans un espace multidimensionnel (ex :  $X = (x, y)$  dans le 2D).
- $I(X)$  est une valeur scalaire.
- Le résultat de cette fonction sera la couleur associée à cette position.



**Figure I-1 :** Image définie sur un espace multidimensionnel



## I-2- Etat de l'art :

Les premières applications des images numériques remontent aux années 1920 et ne concernent que les premiers pas en transfert d'images et en correction et en amélioration d'images. Les principaux événements se sont déroulés selon la chronologie suivante :

- **1921** : Transmission d'images de journaux par câble sous marin entre Londres et New York (une image transmise en trois heures) par codage au départ et décodage à l'arrivée par une imprimante spécialisée. Chaque niveau de gris étant codé par un caractère particulier (déclenchant une impression plus ou moins dense en surface encrée). Harry G. Bartholomew et Maynard D. McFarlane effectuèrent la première numérisation d'image avec compression de données pour envoyer des fax de Londres à New York.
- **1922** : Utilisation de 5 niveaux de gris.
- **1929** : Passage à 15 niveaux de gris.
- **1929** : Premier brevet déposé sur la reconnaissance optique de caractère.
- Le besoin de traiter les images est rapidement devenu une évidence, dès les années **1950** en physique des particules pour scruter les composants de la matière, il fallait analyser 10 000 à 100 000 images par expérience dans les chambres à bulles pour déterminer les trajectoires de milliers de particules grâce à plusieurs caméras réparties.
- Dans le début des années **1960**, les chercheurs se sont intéressés à la lecture optique pour reconnaître les caractères dactylographiés d'un texte. En termes d'images, cette application semblait abordable : le nombre de caractères est limité et l'image est contrastée.
- Dans les années **1960**, les images prises du ciel (imagerie aérienne ou satellitaire) se multipliaient aussi. « On pouvait en tirer une multitude d'informations comme la progression de la désertification d'une région, l'évolution des zones urbaines ou des informations tactiques pour les militaires ». Le véritable essor du traitement d'images n'a lieu quand les ordinateurs commencèrent à être suffisamment puissants pour travailler sur des images. Peu après, la redécouverte de la transformée de Fourier rapide (FFT) révolutionne le domaine, en rendant possible les manipulations du contenu fréquentiel des signaux sur ordinateur.

- **Entre 1960 et 1965**, les Américains ont beaucoup investi dans ce domaine pour leur défense, Leurs travaux sur l'amélioration et l'interprétation d'images numériques spatiales se sont révélés de très bonne qualité. Les Russes relevaient ce même défi, à la même époque, mais à partir d'observation visuelle d'hommes embarqués dans des satellites. Ils ont beaucoup étudié les performances du système visuel humain dans l'espace mais, du coup, n'ont pas développé autant les algorithmes de reconnaissance de forme.
- **Dès 1964**, des chercheurs français, Georges Matheron (1930-2000) et Jean Serra, ont développé une autre approche théorique (baptisée morphologie mathématique) d'abord pour analyser des microphotographies de terrain et évaluer des teneurs en minéral, puis pour d'autres applications comme la cytologie (caractérisation et comptage de cellules) ».
- Le professeur René de Possel (1905-1974), alors directeur de l'Institut Blaise Pascal à Paris, avait mis au point un lecteur optique dès 1965, et fit fonctionner **en 1969** la première machine à lire automatiquement les textes imprimés.
- D'une manière générale, jusqu'à **la fin des années 1960**, les images, que ce soient des photos des satellites, des images d'ADN au microscope électronique ou des radiographies, étaient de mauvaise qualité, difficiles à exploiter. Les optiques étaient peu performantes, provoquant de nombreuses aberrations géométriques et chromatiques. Traiter des images a d'abord consisté à les restaurer en corrigeant tous ces défauts d'acquisition. En parallèle, se posaient deux problèmes : leur volume - plusieurs milliers de pixels par ligne sur plusieurs milliers de lignes - difficile à stocker, et surtout leur traitement, très lent.
- Que ce soit pour déchiffrer un texte dactylographié ou manuscrit, pour compter des chromosomes, reconnaître une tumeur, un char ou un avion de guerre, la compréhension de l'image, sa classification passe toujours par la reconnaissance d'une forme. Plusieurs approches théoriques ont été développées « Les premières consistaient à faire des calculs à partir de l'image et construire des représentations symboliques de plus en plus complexes.
- **En 1976**, on disposait d'un système complet de reconnaissance de 1000 caractères dactylographiés par seconde.

- Un peu partout dans le monde, les chercheurs ont mis au point des méthodes mathématiques permettant de détecter les contours des objets à partir des changements rapides de contraste dans l'image, des ombres et des lumières, des régions homogènes en couleur, en intensité, en texture. ce n'est que **dans les années 80** que ces théories ont réellement commencé à être développées. Avant, les ordinateurs étaient beaucoup trop limités en mémoire et vitesse de calcul pour traiter des images. On utilisait des formules mathématiques simplifiées à l'extrême. On passait autant de temps à faire de l'électronique pour parvenir à afficher nos images qu'à les interpréter.
- En **1980**, David Marr formalise le premier la détection de contour de manière précise.
- **Les années 1990** ont vu l'amélioration constante des opérateurs de filtrage des images.
- Aujourd'hui, même les ordinateurs domestiques les plus récents ne sont pas encore assez performants pour résoudre certains problèmes, en particulier en imagerie médicale. Les quantités d'information à stocker et à traiter restent problématiques, par exemple pour expertiser un cœur qui bat, la nature bénigne ou maligne d'une tumeur, ou pour répondre aux problèmes les plus complexes de vision par ordinateur, ceux qui nécessitent une analyse comparable à celle de notre intelligence. En revanche, beaucoup d'applications sont déjà courantes comme les traitements et l'amélioration sur les images des appareils photo numériques que chacun peut faire sur son ordinateur personnel, ou les effets spéciaux au cinéma pour mélanger des images virtuelles tridimensionnelles à une scène.

### **I-3- Image numérique :**

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation.

Les images numériques peuvent être séparées en deux grandes familles : les images numérisées d'origine extérieure et les images de synthèse (produites sur un ordinateur).

Une image digitalisée est une représentation informatique d'une image. La numérisation se fait dans deux espaces :

- L'espace spatial où l'image est numérisée suivant l'axe des abscisses et des ordonnées ; on parle d'échantillonnage.
- L'espace des couleurs où les différentes valeurs de luminosité que peut prendre un pixel sont numérisées pour représenter sa couleur et son intensité ; on parle de quantification.

L'image échantillonnée formera un ensemble de points appelés pixels (pixel est une abréviation de Picture Element). Le pixel représente le plus petit élément constitutif d'une image numérique. Ainsi l'image est découpée en points élémentaires, ou « pixel », et une couleur est attribuée à ce pixel. La couleur est représentée par un nombre, la correspondance couleur nombre étant faite par une « palette ».

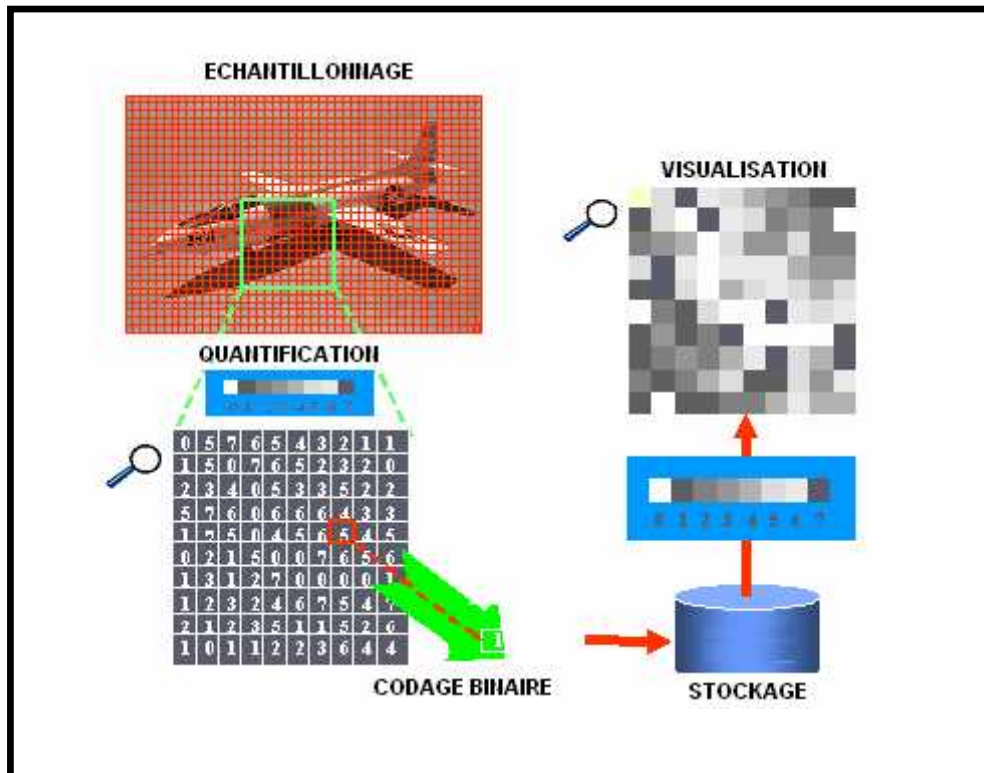
Etant donné que l'écran effectue un balayage de gauche à droite et de haut en bas, on désigne généralement par les coordonnées [0,0] le pixel situé en haut à gauche de l'image, cela signifie que les axes de l'image sont orientés de la façon suivante:

- L'axe X est orienté de gauche à droite.
- L'axe Y est orienté de haut en bas.

Il est inutile de donner les coordonnées des points : si l'on donne la largeur de l'image en nombre  $n$  de points, alors les  $n$  premiers points représentent la première ligne, les points  $n+1$  à  $2n$  représentent la deuxième ligne... Il suffit alors de fixer par convention l'ordre de balayage, en l'occurrence de gauche à droite et de haut en bas. Ceci donne une image au format carte de points, souvent appelée image bitmap. C'est donc un canevas de points dont chacun se voit attribuer une valeur colorée.

Les grandes différences entre les formats existants sont la profondeur de couleurs (1 bit : noir ou blanc, 8 bits : 256 couleurs, 24 bits : 16 millions de couleurs...) et le type de compression.

Le stockage de l'image en mémoire est réalisé en conservant les données attachées à chaque pixel dans un tableau ou matrice (chaque pixel correspondra une case de la matrice).



**Figure I-2 :** Numérisation=Echantillonnage + Quantification

#### I-4- Définition et résolution :

On appelle définition le nombre de points (pixel) constituant l'image, c'est-à-dire sa « dimension informatique » (le nombre de colonnes de l'image que multiplie son nombre de lignes). Une image possédant 640 pixels en largeur et 480 en hauteur aura une définition de 640 pixels par 480, notée 640x480.

En imagerie informatique (écran d'ordinateur, appareil photo numérique, imprimante, numériseur "scanner", etc.), une image est formée par un alignement horizontal et vertical de points lumineux (*pixels*). Pour que deux sources ponctuelles lumineuses apparaissent distinctement dans une telle image, il faut qu'elles soient sur deux pixels distincts, ou mieux, sur deux pixels séparés par au moins un pixel. Par conséquent la résolution, au sens optique, d'un système d'imagerie informatique, se mesure par les dimensions spatiales d'un pixel. La résolution est plus fine ou plus grossière selon que les dimensions des pixels sont plus petites ou plus grandes. Généralement, la largeur et la hauteur d'un pixel sont égales (un pixel est carré), donc on mesure la résolution par le nombre de pixels par unité de longueur.

La résolution, terme souvent confondu avec la "définition", détermine le nombre de points par unité de surface, exprimé en points par pouce (PPP, en anglais DPI pour Dots Per Inch); un pouce représentant 2.54 cm. La résolution permet ainsi d'établir le rapport entre le nombre de pixels d'une image et la taille réelle de sa représentation sur un support physique. Une résolution de 300 dpi signifie donc 300 colonnes et 300 rangées de pixels sur un pouce carré ce qui donne donc 90000 pixels sur un pouce carré. La résolution de référence de 72 dpi nous donne un pixel de  $1"/72$  (un pouce divisé par 72) soit 0.353mm.

Lorsque la résolution est deux fois plus fine, il y a deux fois plus de pixels à la fois en largeur et en hauteur, donc la taille informatique de l'image (nombre de pixels) sera multipliée par quatre ; plus généralement, une résolution n fois plus fine donne une image  $n^2$  fois plus volumineuse.

Sur un écran d'ordinateur, la taille des pixels, donc la résolution, est fixée une fois pour toute ; par exemple un écran dont les dimensions sont de rapport  $H/L = 4/5$  pourra avoir 1024 pixels en hauteur et 1280 en largeur. Une image numérisée à résolution plus fine apparaîtra tout simplement plus grande sur l'écran.

### **I-5- Couleurs :**

Afin de pouvoir manipuler correctement des couleurs et échanger des informations colorimétriques il est nécessaire de disposer de moyens permettant de les catégoriser et de les choisir.

En informatique, de la même façon, il est essentiel de disposer d'un moyen de choisir une couleur parmi toutes celles utilisables. Or, la gamme de couleurs possibles est très vaste et la chaîne de traitement de l'image passe par différents périphériques : par exemple un numériseur (scanner), puis un logiciel de retouche d'image et enfin une imprimante. Il est donc nécessaire de pouvoir représenter de façon fiable la couleur afin de s'assurer de la cohérence entre ces différents périphériques. On appelle ainsi «espace de couleurs» la représentation mathématique d'un ensemble de couleurs. Il en existe plusieurs, parmi lesquels les plus connus sont :

- Le mode **RVB** (Rouge, Vert, Bleu, en anglais RGB, Red, Green, Blue).
- Le mode **CMYK**( Cyan, Magenta, jaune, en anglais Cyan, Magenta, Yellow).

- Le mode **TSL** (Teinte, Saturation, Luminance, en anglais HSL, Hue, Saturation, Luminance).
- Le mode **CIE**.
- Pseudo-couleurs.

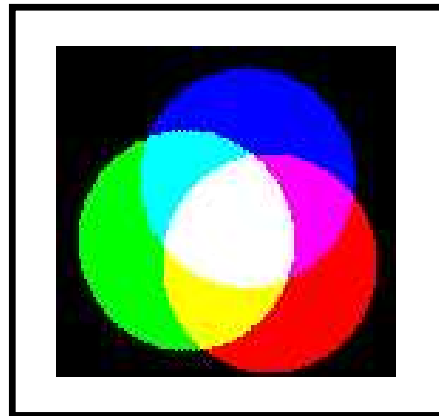
Le spectre de couleurs qu'un périphérique d'affichage permet d'afficher est appelé gamut ou espace colorimétrique. Les couleurs n'appartenant pas au gamut sont appelées couleurs hors-gamme.

### **I-5-1- Mode RVB (Synthèse additive de la lumière) :**

L'image RVB est obtenue par la superposition de trois rayonnements lumineux : le rouge (R), le vert (V) et le bleu (B).

Dans le cas d'un écran cathodique, ces 3 rayonnements sont obtenus en bombardant les luminophores photosensibles de l'écran.

La gamme des couleurs reproductibles par ce mode, quoique conditionnée par la qualité du matériel employé, est très étendue, et reproduit bien les couleurs saturées. En contrepartie, elle convient mal à la restitution des nuances délicates des lumières intenses et des tons pastel.



**Figure I- 3** : Synthèse additive

### **I-5-2- Mode CMJN (Synthèse soustractive de la lumière) :**

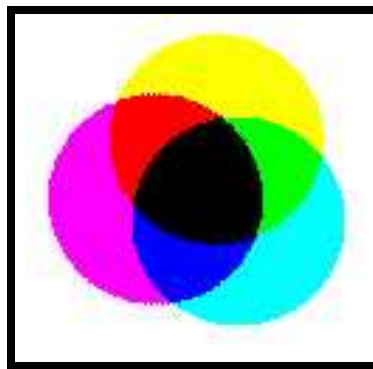
Le type de synthèse des couleurs que nous avons décrit plus haut se base sur l'addition de rayons lumineux, donc plus on met de couleurs, plus le résultat sera clair. C'est ce qu'on appelle la *synthèse additive* de la couleur. Il y a une autre forme de synthèse, où les divers éléments

constitutifs de la couleur agissent en enlevant certaines longueurs d'ondes, donc plus on met de couleurs, plus le résultat sera sombre. C'est la *synthèse soustractive* de la couleur.

Les couleurs sont obtenues par mélange des pigments colorés. Lorsqu'ils sont éclairés par la lumière blanche, les pigments absorbent une partie de la lumière qu'ils reçoivent ce qui les fait apparaître colorés.

Les trois couleurs «primaires» sont le cyan (C), le magenta (M) et le jaune (J). Les autres sont obtenues par mélange.

Les trois encres additionnées ne donnent pas plus qu'un brun sombre, que l'on se doit de renforcer par un quatrième passage d'encre noire (que l'on note donc en toute logique : «N», comme Noir). La gamme des couleurs reproductibles par le mode CMJN est plus restrictive que celle de la gamme RVB.



**Figure I-4 :** Synthèse soustractive

### **I-5-3- Mode HSL :**

Le modèle HSL (Hue, Saturation, Luminance, ou en français TSL), s'appuyant sur les travaux du peintre Albert H. Munsell, est un modèle de représentation dit "naturel", c'est-à-dire proche de la perception physiologique de la couleur par l'oeil humain. En effet, le modèle RGB aussi adapté soit-il pour la représentation couleur ou bien l'affichage sur les périphériques de sortie, ne permet pas de sélectionner facilement une couleur.

En effet, le réglage de la couleur en RGB dans les outils informatiques se fait généralement à l'aide de trois glisseurs ou bien de trois cases avec les valeurs relatives de chacune des composantes primaires, or l'éclaircissement d'une couleur demande d'augmenter proportionnellement les valeurs respectives de chacune des composantes. Ainsi le modèle HSL a-t-il été mis au point afin de pallier cette lacune du modèle RGB.



Le modèle HSL consiste à décomposer la couleur selon des critères physiologiques :

- La teinte (en anglais Hue), correspondant à la perception de la couleur (Exemple : chose mauve ou orange).
- La saturation, décrivant la pureté de la couleur, c'est-à-dire son caractère vif ou terne (Exemple : chose neuve ou délavée).
- La luminance, indiquant la quantité de lumière de la couleur, c'est-à-dire son aspect clair ou sombre (Exemple : chose au soleil ou à l'ombre).

Il existe d'autres modèles naturels de représentation proches du modèle HSL :

- **HSB** : Hue, Saturation, Brightness soit Teinte, Saturation, Brillance en français. La brillance décrit la perception de la lumière émise par une surface.
- **HSV** : Hue, Saturation, Value soit Teinte, Saturation, Valeur.
- **HSI** : Hue, Saturation, Intensity soit Teinte, Saturation, Intensité.
- **HCI** : Hue, Chrominance, Intensity soit Teinte, Chrominance, Intensité.

#### **I-5-4- Mode CIE :**

Les couleurs peuvent être perçues différemment selon les individus et peuvent être affichées différemment selon les périphériques d'affichage.

La Commission Internationale de l'Eclairage (CIE) a donc défini des standards permettant de définir une couleur indépendamment des périphériques utilisés. En 1976, la CIE développe le modèle colorimétrique **La\*b\*** (aussi connu sous le nom de *CIELab*), dans lequel une couleur est repérée par trois valeurs :

- L, la luminance, exprimée en pourcentage (0 pour le noir à 100 pour le blanc).
- a et b deux gammes de couleur allant respectivement du vert au rouge et du bleu au jaune avec des valeurs allant de -120 à +120.

Le mode Lab couvre l'intégralité du spectre visible par l'oeil humain et le représente de manière uniforme. Il permet de décrire l'ensemble des couleurs visibles indépendamment de toute technologie graphique.

### I-5-5- Pseudo- Couleurs :

Parfois il est visuellement plus parlant de représenter une image à niveaux de gris comme une image en couleurs, en remplaçant chaque niveau de gris par une couleur au moyen d'une fonction réalisant une table de couleurs. Dans ce cas, on parlera de fausses couleurs, ou de pseudo couleurs [2].

### I-6- Luminance et contraste :

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

Le contraste est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image.

Le contraste C peut-être défini de plusieurs façons :

- L'écart-type des variations des niveaux de gris dans l'image.
- La variation entre les valeurs de niveau de gris minimal et maximal :

$$C = \frac{[\max f(x, y) - \min f(x, y)]}{[\max f(x, y) + \min f(x, y)]} \quad (\text{I-1})$$

Si L1 et L2 sont les degrés de luminosité de deux zones d'une image alors :

$$C = \frac{L_2 - L_1}{L_1} \quad (\text{I-2})$$

### I-7- Transparence :

La transparence est une caractéristique permettant de définir le niveau d'opacité des éléments d'une image, c'est-à-dire la possibilité de voir à travers l'image des éléments graphiques située derrière celle-ci.

Il existe deux modes de transparence :

- La transparence simple s'applique pour une image indexée et consiste à définir parmi la palette de couleurs une des couleurs comme transparente.

- La transparence par couche alpha (ou canal alpha, en anglais alpha channel) consiste à rajouter pour chaque pixel de l'image un octet définissant le niveau de transparence (de 0 à 255).

### **I-8- Contours :**

Les contours représentent la frontière entre les objets de l'image ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative.

### **I-9- Codage de la couleur :**

Une image est donc représentée par un tableau à deux dimensions dont chaque case est un pixel. Pour représenter informatiquement une image, il suffit donc de créer un tableau de pixels dont chaque case contient une valeur. La valeur stockée dans une case est codée sur un certain nombre de bits déterminant la couleur ou l'intensité du pixel, on l'appelle profondeur de codage (parfois profondeur de couleur).

Il existe plusieurs standards de codage de la profondeur :

- **bitmap noir et blanc** : En stockant un bit dans chaque case, il est possible de définir deux couleurs (noir ou blanc).
- **bitmap 16 couleurs ou 16 niveaux de gris** : En stockant 4 bits dans chaque case, il est possible de définir  $2^4$  possibilités d'intensités pour chaque pixel, c'est-à-dire 16 dégradés de gris allant du noir au blanc ou bien 16 couleurs différentes.
- **bitmap 256 couleurs ou 256 niveaux de gris** : En stockant un octet dans chaque case, il est possible de définir  $2^8$  intensités de pixels, c'est-à-dire 256 niveaux différents (nuances de gris ou de couleurs).
- **palette de couleurs (colormap)** : Grâce à cette méthode, il est possible de définir une palette, ou table des couleurs, contenant l'ensemble des couleurs pouvant être contenues dans l'image, à chacune desquelles est associé un indice. Le nombre de bits réservé au codage de chaque indice de la palette détermine le nombre de couleurs pouvant être utilisées. Ainsi en codant les indices sur 8 bits il est possible de définir 256 couleurs utilisables, c'est-à-dire que chaque case du tableau à deux dimensions représentant l'image va contenir un nombre indiquant l'indice de la couleur à utiliser. Cette image s'appelle une image en couleurs indexées.

- **Couleurs vraies (True color) ou couleurs réelles** : On définit chacune des composantes (RGB, pour rouge, vert et bleu). Chaque pixel est représenté par un entier comportant les trois composantes, chacune codée sur un octet, c'est-à-dire au total 24 bits (16 millions de couleurs). Il est possible de rajouter une quatrième composante permettant d'ajouter une information de transparence ou de texture, chaque pixel est alors codé sur 32 bits.

### I-10- Poids d'une image :

Pour connaître le poids (en octets) d'une image, il est nécessaire de compter le nombre de pixels que contient l'image, cela revient à calculer le nombre de cases du tableau, soit la hauteur de celui-ci que multiplie sa largeur. Le poids de l'image est alors égal à son nombre de pixels que multiplie le poids de chacun de ces éléments.

#### Exemples de calcul de poids :

Le calcul pour une image 640x480 en True color :

Nombre de pixels :  $640 \times 480 = 307200$

Puisque chaque pixel a 03 composantes et chaque composante a un octet alors :

Le poids de l'image en kilo octet est ainsi égal à :

$307200 \times 3 = 921600$  octets, Soit :  $921600 / 1024 = 900$  Ko.

Définition de l'image	Noir et blanc (1 bit)	256 couleurs (8 bits)	65000 couleurs (16 bits)	True color (24 bits)
320x200	7.8 Ko	62.5 Ko	125 Ko	187.5 Ko
640x480	37.5 Ko	300 Ko	600 Ko	900 Ko
800x600	58.6 Ko	468.7 Ko	937.5 Ko	1.6 Mo
1024x768	96 Ko	768 Ko	1.5 Mo	2.3 Mo

**Tableau I-1** : Poids de l'image

## I-11- Images bitmap et images vectorielles :

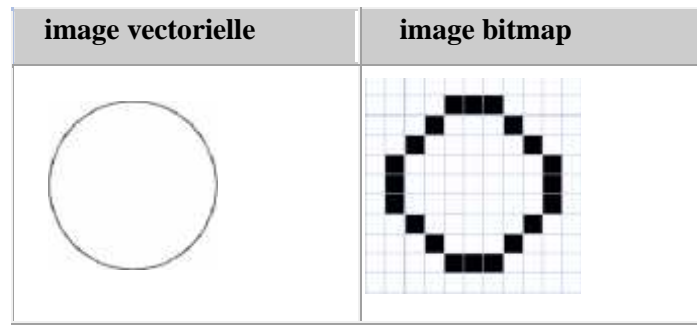
On distingue généralement deux grandes catégories d'images:

- **les images bitmap** : (appelées aussi images raster) Il s'agit d'images pixellisées, c'est-à-dire un ensemble de points (pixels) contenus dans un tableau, chacun de ces points possédant une ou plusieurs valeurs décrivant sa couleur.
- **les images vectorielles** : Les images vectorielles sont des représentations d'entités géométriques telles qu'un cercle, un rectangle ou un segment. Ceux-ci sont représentés par des formules mathématiques (un rectangle est défini par deux points, un cercle par un centre et un rayon, une courbe par plusieurs points et une équation). C'est le processeur qui sera chargé de "traduire" ces formes en informations interprétables par la carte graphique.

Etant donné qu'une image vectorielle est constituée uniquement d'entités mathématiques, il est possible de lui appliquer facilement des transformations géométriques (zoom, étirement, ...), tandis qu'une image bitmap, faite de pixels, ne pourra subir de telles transformations qu'au prix d'une perte d'information, appelée distorsion. On nomme ainsi pixellisation (en anglais aliasing) l'apparition de pixels dans une image suite à une transformation géométrique (notamment l'agrandissement). De plus, les images vectorielles (appelées cliparts lorsqu'il s'agit d'un objet vectoriel) permettent de définir une image avec très peu d'information, ce qui rend les fichiers très peu volumineux.

En contrepartie, une image vectorielle permet uniquement de représenter des formes simples.

S'il est vrai qu'une superposition de divers éléments simples peut donner des résultats très impressionnants, toute image ne peut pas être rendue vectoriellement, c'est notamment le cas des photos réalistes [3].



**Figure I-5 :** Images vectorielle et bitmap

### I-12- Compression des images :

De nos jours, la puissance des processeurs augmente plus vite que les capacités de stockage, pour pallier ce manque, il est courant de réduire la taille des données. Les images numériques font partie des données dont le stockage nécessite le plus de place. Ainsi une image de qualité photographique (600 points par pouce), de dimensions 10 x 15 cm, occupe environ 25 méga-octets, soit une suite de plus de 200 millions de 0 et de 1.

La compression consiste à réduire la taille physique de blocs d'informations. Un compresseur utilise un algorithme qui sert à optimiser les données en utilisant des considérations propres au type de données à compresser; un décompresseur est donc nécessaire pour reconstruire les données originelles grâce à l'algorithme inverse de celui utilisé pour la compression.

La compression peut se définir par le taux de compression, c'est-à-dire le quotient du nombre de bits dans l'image originale par le nombre de bits dans l'image compressée.

#### I-12-1- Compression sans pertes :

La compression est dite sans perte lorsqu'il n'y a aucune perte de données sur l'information d'origine. Il y a autant d'information après la compression qu'avant, elle est seulement réécrite d'une manière plus concise. La compression sans perte est dite aussi compactage.

L'information à compresser est vue comme la sortie d'une source de symboles qui produit des textes finis selon certaines règles. Le but est de réduire la taille moyenne des textes obtenus après la compression tout en ayant la possibilité de retrouver exactement le message d'origine.

### **I-12-1-1- Concaténation de points :**

La concaténation de point est une méthode permettant de stocker les points d'une manière optimale: pour une image monochrome il n'y a, par définition, que deux couleurs, un point de l'image peut donc être codé sur un seul bit pour gagner de l'espace mémoire.

### **I-12-1-2- Compression RLE :**

La méthode de compression RLE (Run Length Encoding, parfois notée RLC pour Run Length Coding) est utilisée par de nombreux formats d'images (BMP, PCX, TIFF). Elle est basée sur la répétition d'éléments consécutifs. Le principe de base consiste à coder un premier élément donnant le nombre de répétitions d'une valeur puis le compléter par la valeur à répéter.

#### **Exemple de compression RLE :**

Dans l'image, on trouve une séquence du type:

**157-157-157-157-157-007-007-007-007-007-007-007-007-007-007-007-007-007-007**

Ce qui donne une fois codé :

**005/157-14/007**

Le gain de compression est ainsi de  $(19-5)/19$  soit environ 73,7%.

En contrepartie pour la chaîne "107-20-20-175-175-20-45-20-125-45", dans lequel la redondance des caractères est faible, le résultat de la compression donne "1/107-2/20-2/175-1/20-1/45-1/20-1/125-1/45"; la compression s'avère ici très coûteuse, avec un gain négatif valant  $(10-16)/10$  soit -60%!

En réalité la compression RLE est régie par des règles particulières permettant de compresser lorsque cela est nécessaire et de laisser la chaîne telle quelle lorsque la compression induit un gaspillage.

### **I-12-1-3- Codage de Huffman :**

David Huffman a proposé en 1952 une méthode statistique qui permet d'attribuer un mot de code binaire aux différents symboles à compresser (pixels ou caractères par exemple). La longueur de chaque mot de code n'est pas identique pour tous les symboles: les symboles les plus fréquents (qui apparaissent le plus souvent) sont codés avec de petits mots de code, tandis que les symboles les plus rares reçoivent de plus longs codes binaires.

**Exemple de compression Huffman :**

Dans l'image, on trouve une séquence du type:

**157 255 157 157 007 001 007 255 255 089 089 157 089 157 089**

157 est remplacé par 01

089 est remplacé par 001

255 est remplacé par 0011

007 est remplacé par 00011

001 est remplacé par 000111

Ce qui donne une fois codée :

**01 0011 01 01 00011 000111 00011 0011 0011 001 001 01 001 01 001**

**I -12-1-4- Compression LZW :**

Abraham Lempel et Jakob Ziv sont les créateurs du compresseur LZ77, inventé en 1977 (d'où son nom). Ce compresseur était alors utilisé pour l'archivage (les formats ZIP, ARJ et LHA l'utilisent). En 1978 ils créent le compresseur LZ78 spécialisé dans la compression d'images (ou tout type de fichier de type binaire).

En 1984, Terry Welch de la société Unisys le modifia pour l'utiliser dans des contrôleurs de disques durs, son initiale vint donc se rajouter à l'abréviation LZ pour donner LZW.

LZW est un algorithme très rapide aussi bien en compression qu'en décompression, basé sur la multiplicité des occurrences de séquences de caractères dans la chaîne à encoder. La compression Lempel-Ziv-Welch est dite de type dictionnaire. Elle est basée sur le fait que des successions de caractères se retrouvent plus souvent que d'autres et qu'on peut donc les remplacer par un nouveau caractère.



Son principe consiste à substituer des motifs par un code d'affectation (indice) en construisant au fur et à mesure un dictionnaire. De plus, il travaille sur des bits et non sur des octets, il ne dépend donc pas de la manière de laquelle le processeur code les informations.

### **Exemple de compression LZW :**

Supposons que dans l'image, on trouve une séquence du type:

**1573157 2557255 1573157 0 070007 070007 2557255 08990899 1573157 08990899**

1573157 est remplacé par code 1.

2557255 est remplacé par code2.

070007 est remplacé par code3 .

08990899 est remplacé par code4

Ce qui donne une fois codée :

**code1 - code2 - code1 - 0 - code3 - code3 - code2 - code4 -code1 - code4**

### **I-12-2- Compression avec pertes :**

La compression avec pertes (en anglais lossy compression), par opposition à la compression sans pertes (lossless compression), se permet d'éliminer quelques informations pour avoir le meilleur taux de compression possible, tout en gardant un résultat qui soit le plus proche possible des données originales. Puisque l'œil ne perçoit pas nécessairement tous les détails d'une image, il est possible de retrancher des données, dans l'espace des fréquences, de telle sorte que le résultat soit très ressemblant à l'original, voire pareil, pour l'œil. Le tout est de savoir quelles données retrancher. L'image finale n'étant pas, numériquement parlant, identique à l'image initiale, d'où la perte en question. Les programmes de compression s'attachent à découvrir ces zones et à les coder de la façon aussi compacte que possible.

## Compression JPEG :

L'acronyme JPEG (Joint Photographic Expert Group prononcez jipègue ou en anglais djaypègue) provient de la réunion en 1982 d'un groupe d'experts de la photographie, dont le principal souci était de travailler sur les façons de transmettre des informations (images fixes ou animées).

Contrairement à la compression LZW, la compression JPEG est une compression avec pertes, ce qui lui permet, en dépit d'une perte de qualité, un des meilleurs taux de compression (20:1 à 25:1 sans perte notable de qualité).

Cette méthode de compression est beaucoup plus efficace sur les images photographiques (comportant de nombreux pixels de couleurs différentes) et non sur des images géométriques (à la différence de la compression LZW) car sur ces dernières les différences de nuances dues à la compression sont très visibles. Les étapes de la compression JPEG sont :

- **Rééchantillonnage** de l'image par un changement du mode de la couleur.
- **Découpage de l'image** en blocs de 8x8 points, puis l'application d'un opérateur mathématique : la fonction DCT (Discrete Cosinus Transform, transformation discrète en cosinus) qui décompose l'image en somme de fréquences.
- **Quantification de chaque bloc**, on compacte les nombreuses valeurs nulles par un codage RLE.
- **Compression de l'image** avec la méthode d'Huffman.

La nouvelle norme JPEG 2000, par exemple, arrive généralement à coder des images photographiques sur 1 bit par pixel sans perte visible de qualité sur un écran, soit une compression d'un facteur 24 à 1 [4].

## I-13- Formats de fichiers images :

Pour être affichée sur moniteur, l'image doit être codée, toutefois lorsque l'on veut stocker cette image dans un fichier, ce format n'est pas le plus pratique. On peut en effet vouloir une image qui prenne moins de place en mémoire, ou bien une image que l'on puisse agrandir sans faire apparaître de pixellisation.

Il est possible de stocker l'image dans un fichier avec une structure de donnée décrivant l'image à l'aide d'équation, et qui devra être décodée par le processeur avant que les informations soient envoyées à la carte graphique.

Il existe un grand nombre de formats de fichiers. Parmi les formats de fichiers graphiques les plus utilisés sont :

<b>Format</b>	<b>Compression</b>	<b>Dimensions maximales</b>	<b>Nombre maximal de couleurs</b>
BMP	aucune / RLE	65 536 x 65 536	16 777 216
GIF	LZW	65 536 x 65 536	256
JPEG	JPEG	65 536 x 65 536	Supérieur à 16 777 216
PCX	aucune / RLE	65 536 x 65 536	16 777 216
PNG	RLE	65 536 x 65 536	Supérieur à 16 777 216
TIFF/TIF	RLE/JPEG/LZW/...	$2^{32}-1$	Supérieur à 16 777 216

**Tableau I-2 :** Les formats de fichiers graphiques les plus utilisés

### **I-13-1- Format BMP :**

Le format BMP est un des formats les plus simples développé conjointement par Microsoft et IBM, ce qui explique qu'il soit particulièrement répandu sur les plates formes Windows.

Un fichier BMP est un fichier bitmap (ou en mode point, ou « raster »), c'est-à-dire un fichier d'image graphique stockant les pixels sous forme de tableau de points et gérant les couleurs soit en couleur vraie soit grâce à une palette indexée.

### **I-13-2- Format PCX :**

Le PCX est un format d'image numérique dont l'encodage est basé sur une forme de run-length encoding. Il a été mis au point par la société ZSoft, Corporation à Marietta, en Géorgie. C'était le format de base du logiciel PaintBrush équipant en standard les systèmes d'exploitation Microsoft Windows à partir des années 80. Le format PCX est un format bitmap permettant d'encoder des images dont la dimension peut aller jusqu'à 65536 par 65536 et codées sur 1 bit, 4 bit, 8 bit ou 24 bit (correspondant respectivement à 2, 16, 256 ou 16 millions de couleurs).

### **I-13-3- Format GIF :**

GIF a été mis au point par CompuServe en 1986 pour permettre le téléchargement d'images en couleur. Ce format utilise l'algorithme de compression LZW, nettement plus efficace que l'algorithme RLE utilisé par la plupart des formats alors disponibles (PCX et BMP).

Il existe deux versions de ce format de fichier développées respectivement en 1987 et 1989 :

- GIF 87a supportant la compression LZW, une palette de 256 couleurs et la possibilité d'avoir des images animées.
- GIF 89a ajoutant la possibilité de définir une couleur transparente dans la palette et de préciser le délai pour les animations.

Une image GIF peut contenir de 2 à 256 couleurs (2, 4, 8, 16, 32, 64, 128 ou 256) parmi 16 777 216 nuances de couleur : 8 bits par composante RVB, soit  $2^{24}$  nuances. La limitation à 256 couleurs n'est pas gênante pour les logos, les graphiques et la plupart des images synthétiques, ainsi que les photographies noir et blanc. En revanche une photographie couleur de qualité nécessite plus de nuances.

### **I-13-4- Format PNG :**

Le format PNG (Portable Network Graphics, ou format Ping) est un format de fichier graphique bitmap (raster). Il a été mis au point en 1995 afin de fournir une alternative libre au format GIF, format propriétaire dont les droits sont détenus par la société Unisys (propriétaire de l'algorithme de compression LZW), ce qui oblige chaque éditeur de logiciel manipulant ce type de format à leur verser des royalties. Ainsi PNG est également un acronyme récursif pour PNG's Not Gif.

Le format PNG permet de stocker des images en noir et blanc (jusqu'à 16 bits par pixels de profondeur de codage), en couleurs réelles (True color, jusqu'à 48 bits par pixels de profondeur de codage) ainsi que des images indexées, faisant usage d'une palette de 256 couleurs. De plus il supporte la transparence par couche alpha, c'est-à-dire la possibilité de définir 256 niveaux de transparence, tandis que le format GIF ne permet de définir qu'une seule couleur de la palette comme transparente. Il possède également une fonction d'entrelacement permettant d'afficher l'image progressivement.

La compression proposée par ce format est une compression sans perte (lossless compression) 5 à 25% meilleure que la compression GIF.

- Enfin PNG embarque des informations sur le gamma de l'image, ce qui rend possible une correction gamma et permet une indépendance vis-à-vis des périphériques d'affichage. Des mécanismes de correction d'erreurs sont également embarqués dans le fichier afin de garantir son intégrité.

### **I-13-5- Format TIF :**

Le format TIF ou TIFF (Tagged Image File Format) est un format de fichier graphique bitmap (raster). Il a été mis au point en 1987 par la société Aldus (appartenant désormais à Adobe).

Le format TIFF est un ancien format graphique, permettant de stocker des images bitmap (raster) de taille importante (plus de 4 Go compressées), sans perte de qualité. Il permet de stocker des images en noir et blanc, en couleurs réelles (True color, jusqu'à 32 bits par pixels) ainsi que des images indexées, faisant usage d'une palette de couleurs.

TIFF est un format extrêmement flexible, il est courant et lu par tous les logiciels de traitement d'image matricielle:

- Il permet d'utiliser de nombreux types de compression, avec ou sans perte de données.
- Il supporte de nombreux espaces colorimétriques : noir et blanc, monochrome, palettes de couleurs, RVB, CMJN, CIE Lab, etc.
- Il supporte de nombreuses informations additionnelles sur les couleurs utiles à la calibration colorimétrique : correction gamma, etc.
- Il supporte le stockage d'image par bloc, et aussi de multiples images par fichier, des images alternatives en basse résolution, des annotations sous forme de courbes et de texte, etc.

Cette considérable flexibilité fait que TIFF est utilisé dans des applications très diverses, des scanners industriels aux appareils photo numériques en passant par les imprimantes.

### **I-13-6- Format JPEG :**

JPEG (également appelé JPG) est un acronyme de Joint Photographic Experts Group, nom d'un comité créé en 1986. Ce comité a donné son nom à la norme ouverte de compression d'images numériques JPEG. Ensuite cette norme a donné son nom au format de données défini et au format de fichier le plus utilisé pour contenir ces données.

Le format de fichier embarquant un flux codé en JPEG est en réalité appelés JFIF (JPEG File Interchange Format, soit en français Format d'échange de fichiers JPEG), mais par déformation le terme de "fichier JPEG" est couramment utilisé.

Le JPEG est un format à perte, qui élimine donc des informations, mais un des points forts de JPEG est que son taux de compression est réglable. Un compromis doit cependant être fait entre le taux de compression et la qualité de l'image comprimée. En d'autres termes, le taux de compression ne doit pas être trop élevé, ni l'opération de compression être trop souvent répétée, sous peine de nuire gravement à la qualité générale de l'image. Certains logiciels offrent plusieurs choix préprogrammés de compression et d'autres permettent de l'affiner très précisément.

Le format JPEG sauvegarde davantage d'informations couleur que le format GIF et garantit de ce fait un nombre élevé de couleurs. La compression flexible rend possible une réduction de la taille du fichier JPEG sans avoir trop d'impact sur la qualité de l'image.

### **I-14- Qualité de l'image numérique :**

Elle dépend, d'une part, de la qualité des images d'origine et, d'autre part, des moyens mis en œuvre pour convertir un signal analogique en signal numérique. Elle dépend aussi de :

- La qualité des périphériques de numérisation de l'image, du nombre de niveaux de gris ou de couleurs enregistrées, etc.
- La qualité de l'affichage à l'écran : définition de l'écran, nombre de teintes disponibles simultanément, calibrage de l'écran, etc.

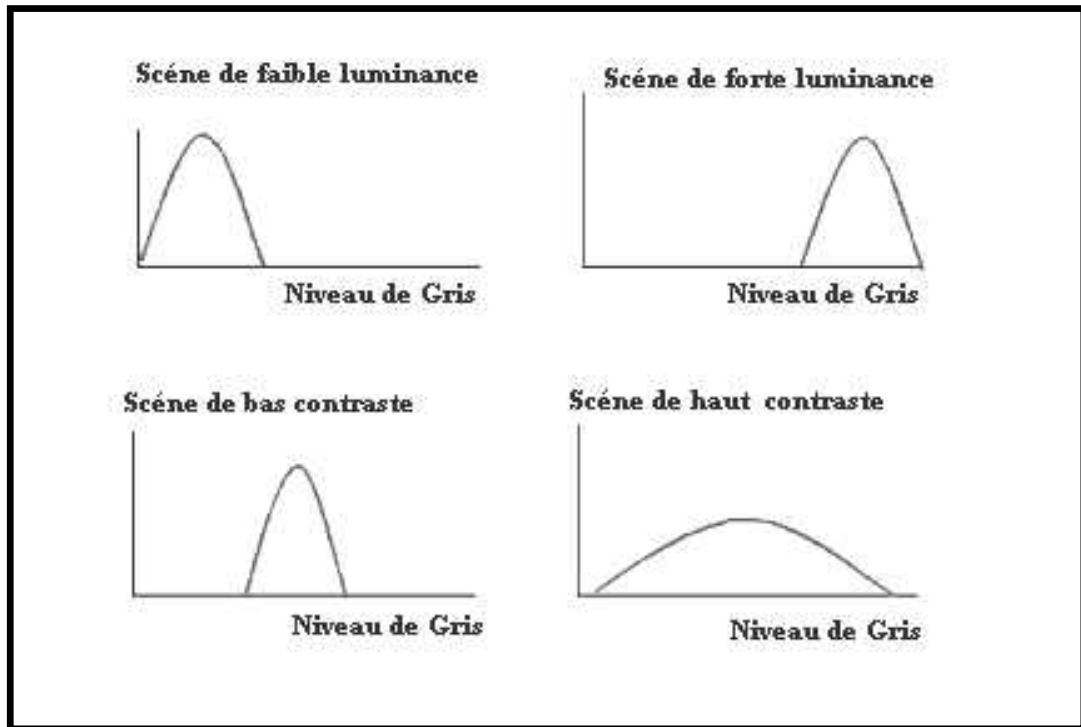
Les critères d'appréciation de la qualité d'une image, dépendent largement de la structure même de l'image réaliste ou conceptuelle et de son mode de représentation (bitmap ou vectorielle) [5].

### **I-15- Caractéristiques statiques :**

#### **I-15-1- Histogramme :**

Une première approche pour extraire et synthétiser l'information consiste à s'intéresser à l'information spectrale, pour cela on calcule la distribution des valeurs de niveaux de gris dans une image soit le nombre de pixels en fonction du niveau de gris.

L'histogramme monodimensionnel est une représentation graphique  $H(m)$  qui indique la fréquence d'apparition des différentes valeurs possibles de  $f(x, y)$ . En abscisse, sont représentés les niveaux de gris et en ordonnée, sont représentés le nombre pixels pour chaque niveau de gris.



**Figure I-6 :** Histogramme I-D

### I-15-2- Moyenne arithmétique :

La valeur moyenne  $E(f)$  des valeurs des pixels de l'image donne une idée sur la luminance globale (la brillance) de l'image.

$$E(f) = \frac{1}{N} \sum_{x=1}^n \sum_{y=1}^p f(x; y) \quad (\text{I-3})$$

Où :

$N = n \times p$  est le nombre de pixels de l'image.

### I-15-3- Variance :

$$\text{Var}(f) = \frac{1}{N} \sum_{x=1}^n \sum_{y=1}^p [f(x, y)^2 - E(f)^2] \quad (\text{I-4})$$



L'écart type étant :

$$\dagger = \text{Var}(f)^{\frac{1}{2}} \quad (\text{I-5})$$

La variance et l'écart type peuvent être calculés pour chaque image ou pour une partie de l'image. Ces deux grandeurs caractérisent la dynamique de la distribution, elles expriment le regroupement ou la dispersion autour de la valeur moyenne. Plus l'écart type est grand, plus les valeurs sont dispersées et plus les grandeurs sont concentrées autour de la moyenne et moins l'écart type est élevé.

#### **I-15-4- Erreur quadratique moyenne :**

Erreur Quadratique moyenne EQM (MSE : mean square error) indique l'erreur soit la différence moyenne des pixels dans toute l'image. Dans notre travail, la définition d'une grandeur MSE se rapporte à une plus grande différence entre l'originale et l'image bruitée. Ceci signifie qu'il y a une réduction significative du bruit. La formule pour le calcul de MSE est donnée par :

$$MSE = \frac{1}{N} \sum_{i=1}^n \sum_{j=1}^m (I(i, j) - \hat{R}(i, j))^2 \quad (\text{I-6})$$

Où :

$I(i, j)$  : l'image originale ou bruitée.

$\hat{R}(i, j)$  : L'image reconstruite ou estimée filtrée.

Les mesures MSE sont utilisées pour analyser les performances des filtres sur toutes les régions de l'image.

#### **I-16- Bruit dans une image :**

Le bruit caractérise les parasites ou interférences d'un signal, c'est-à-dire les parties du signal déformées localement. Ainsi le bruit d'une image désigne les pixels de l'image dont l'intensité est très différente de celles des pixels voisins.

Le bruit peut provenir de différentes causes :

- Environnement lors de l'acquisition.
- Qualité du capteur.
- Qualité de l'échantillonnage.

Le bruit est donc une variation aléatoire de la valeur des pixels au sein d'une image, variation indépendante de l'image en elle-même et qui joue de manière significative sur la qualité de l'image.

Il est essentiellement visible dans les zones relativement uniformes (comme des visages, des ciels bleus et des zones à textures plates) et se manifeste par une combinaison de motifs lumineux et granulés.

On peut dire que le bruit d'image est un champ aléatoire caractérisé par sa densité de probabilité  $f$  et sa fonction de répartition  $F$ .

### **I-16-1- Origines du bruit :**

Tous les algorithmes d'évaluation du bruit dans les images reposent sur la connaissance du type de bruit à analyser. Les bruits proviennent de différentes causes.

#### **I-16-1-1- Bruits externes :**

Les perturbations d'origine externe sont causées par :

- **Bruit lié au contexte de l'acquisition :**

Dans cette première catégorie, on retrouve tous les événements inattendus, modifiant les conditions de l'acquisition du signal. L'exemple le plus simple est le bougé. On peut aussi imaginer une modification ponctuelle des conditions d'éclairage, conduisant à une sur illumination de l'objet observé.

- **Les perturbations artificielles :**

Elles sont principalement dues aux parasites générés par les équipements industriels (Station et les arcs électrique, les antenne...etc).

- **Les perturbations naturelles :**

Elles sont associées à des phénomènes atmosphériques ou cosmiques. L'environnement dans lequel se situe l'acquisition n'est pas toujours maîtrisable. En télédétection, la couverture nuageuse qui peut dégrader la qualité des images

satellites. De même, la poussière dans un atelier de production peut être difficile à éliminer. Cependant, là encore, la connaissance a priori du phénomène perturbateur permet d'envisager une modélisation et donc une correction systématique.

- **Bruit de fluctuations :**

Les bruits de fluctuations peuvent être considérés comme étant la superposition d'un très grand nombre de perturbations impulsives, qui sont dues aux réflexions des ondes électromagnétiques sur les différents éléments de l'environnement (sol, végétation, habitations ...etc.). Généralement ce bruit est de nature non stationnaire.

- **Bruit lié à la nature de la scène :**

L'environnement dans lequel se situe l'acquisition n'est pas toujours maîtrisable. En télédétection, la couverture nuageuse peut dégrader la qualité des images satellites. Cependant, là encore, la connaissance a priori du phénomène perturbateur permet d'envisager une modélisation et donc une correction systématique.

Dans les images radar, les propriétés de la surface (taux d'humidité, rugosité) influent directement sur la netteté de l'image acquise.

#### I-16-1-2- Bruits internes :

- **Bruit thermique :**

C'est le bruit de fond généré dans les câbles et les composants électroniques.

- **Perturbations impulsionnelles :**

Les perturbations essentiellement de types impulsionnelles engendrées par les commutations de courants.

- **Bruit lié au capteur :**

Ce sont des bruits intrinsèques, produits par la chaîne de détection : le CCD, le radar à ouverture synthétique, les divers étages d'amplification, les circuits électroniques du système,...etc.

Le capteur, si il est de mauvaise qualité ou si il est mal utilisé, peut introduire toute sorte de bruits. On peut ainsi obtenir une distorsion de la gamme des niveaux de gris, provoquant une saturation ou bien une distorsion géométrique de l'image équivalente à l'effet d'un miroir grossissant par exemple. Une mauvaise mise au point peut provoquer un flou.

Bien entendu, la meilleure restauration possible consiste à agir directement au niveau du capteur. Cependant, si cela n'est pas possible, la connaissance de l'appareil, permet le plus souvent, de construire un modèle a priori du phénomène de dégradation; une inversion systématique est alors envisageable.

Plusieurs facteurs propres au système Radar et au territoire observé se manifestent dans les images en dégradant plus ou moins sa netteté, les paramètres influant sont les caractéristiques du radar (longueur d'onde, polarisation, angle d'incidence).

De plus les images radars présentent un chatoiement (Speckle en anglais) attribuables au fait que les irrégularités de la scène visée sont de même ordre de grandeur que la longueur d'onde d'éclairage. Ce chatoiement est un bruit multiplicatif car il augmente en fonction la valeur du signal reçu au capteur.

### **I-16-1-3- Bruits de numérisation :**

- **Bruit lié à l'échantillonnage :**

Une image est une forme discrète. Il est donc nécessaire de passer du domaine continu au domaine discret, au niveau de la scène et au niveau de l'intensité d'onde électromagnétique émanant de celle-ci.

D'une manière générale, le spectre des intensités est quantifié sur 256 niveaux de gris différents nécessitant 8 bits pour le stockage en chaque point. Si le capteur est bien calibré, cette quantification ne provoque pas de bruit important. Il faut savoir que cette précision est de loin supérieure à ce que l'oeil humain utilise dans ses tâches quotidiennes [6].

**I-16-2- Modèles statistiques de bruit :**

Dans les images, le bruit peut se manifester sous différents modèles :

**I-16-2-1- Bruit additif :**

Le bruit additif peut être défini de la façon suivante: Etant données une image non bruitée R et I la même image avec un bruit additif A, alors chaque pixel j est caractérisé par la relation :

$$I_j = A_j + R_j \quad (I-7)$$

Où :

$A_j$  est une variable aléatoire de moyenne égale à 0.

**I-16-2-2-Bruit multiplicatif :**

Le bruit multiplicatif se définit de façon analogue: Etant données une image non bruitée R et I la même image avec un bruit multiplicatif B, alors chaque pixel j est caractérisé par la relation:

$$I_j = B_j \cdot R_j \quad (I-8)$$

Où :

$B_j$  est une variable aléatoire de moyenne égale à 1.

La principale caractéristique de ce bruit est que les pixels d'une zone homogène seront d'autant plus bruités que leur niveau de gris est élevé.

**I-16-3- Exemples de bruits :**

Nom	Densité de Probabilité	Moyenne & Variance
Uniforme	$p_z(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$	$m = \frac{a+b}{2}, \quad \sigma^2 = \frac{(b-a)^2}{12}$
Gauss	$p_z(z) = \frac{1}{\sqrt{2\pi}b} e^{-\frac{(z-a)^2}{2b^2}} \quad -\infty < z < \infty$	$m = a, \quad \sigma^2 = b^2$
Sel & Poivre Salt & Pepper	$p_z(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases} \quad b > a$	$m = aP_a + bP_b$ $\sigma^2 = (a-m)^2P_a + (b-m)^2P_b$
Lognormal	$p_z(z) = \frac{1}{\sqrt{2\pi}bz} e^{-\frac{[\ln(z)-a]^2}{2b^2}} \quad z > 0$	$m = e^{a+(b^2/2)}, \quad \sigma^2 = [e^{b^2} - 1]e^{2a+b^2}$
Rayleigh	$p_z(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & z \geq a \\ 0 & z < a \end{cases}$	$m = a + \sqrt{\pi b/4}, \quad \sigma^2 = \frac{b(4-\pi)}{4}$
Exponentiel	$p_z(z) = \begin{cases} ae^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$	$m = \frac{1}{a}, \quad \sigma^2 = \frac{1}{a^2}$
Erlang	$p_z(z) = \frac{a^b z^{b-1}}{(b-1)!} e^{-az} \quad z \geq 0$	$m = \frac{b}{a}, \quad \sigma^2 = \frac{b}{a^2}$

**Tableau I-3 :** Exemples de bruits

**I-16-3-1- Bruit Gaussien :**

Le bruit additif Gaussien, qui dégrade une image u en v telle que :

$$v(x) = u(x) + n(x) \tag{I-9}$$

Où les n(x) sont des variables aléatoires indépendantes et identiquement distribuées suivant une loi Gaussienne centrée de variance  $\dagger^2$  :

$$P[n(x) \geq t] = \frac{1}{\dagger \sqrt{2f}} \int_t^\infty e^{-s^2/2\dagger^2} ds \tag{I-10}$$

Un tel modèle de bruit rend plutôt compte de la multitude de dégradations physiques qui affectent l'image (fluctuations lumineuses dues à la nature quantique du photon, imperfection de l'intégration des capteurs, ...)

### **I-16-3-2- Bruit du chatolement :**

Le bruit du chatolement (speckle) concerne toutes les images satellitaires acquises par un radar embarqué (RSO : Radar à synthèse d'ouverture). Il est dû à l'interférence des ondes réfléchies par les diffuseurs [7].

Dans l'imagerie RSO, le chatolement est généralement modélisé comme un fort bruit multiplicatif. En général, on caractérise le chatolement par une fonction  $f$  de densité de probabilité distribuée selon la loi de Rayleigh :

$$f(U) = \frac{U}{\dagger^2} \exp\left(-\frac{U}{2\dagger^2}\right) \quad (\text{I-11})$$

### **I-16-3-3- Flou :**

A titre d'exemple, nous citerons le flou gaussien qui est obtenu par convolution avec une gaussienne de "largeur"  $\dagger$  :

$$h(x, y) = \frac{1}{f\dagger^2} e^{-\frac{x^2 + y^2}{2\dagger^2}} \quad (\text{I-12})$$

# **Chapitre II**

## **Filtrage spatial**

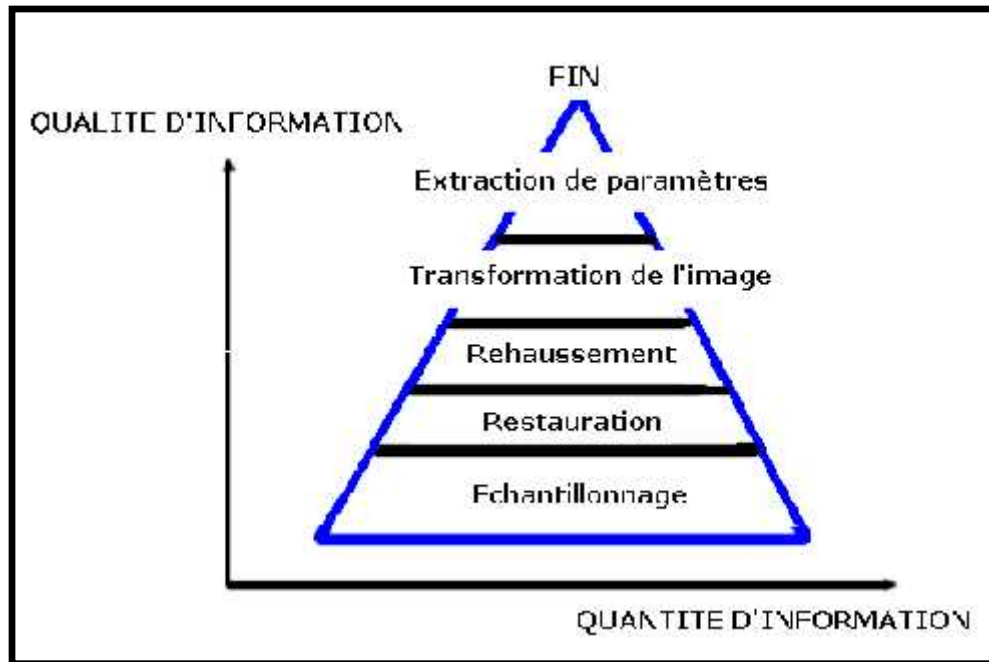


---

## II-1- Traitement mathématique :

Les méthodes mathématiques de traitement d'images peuvent être groupées dans trois catégories fonctionnelles :

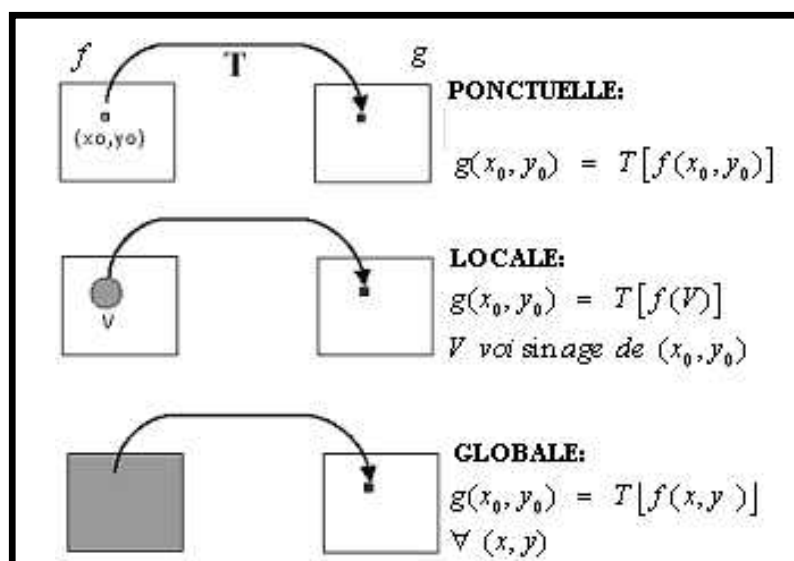
- Les prétraitements servent à rétablir les images et à compenser des erreurs de données, le bruit et des déformations géométriques. Les opérations principales sont composées de :
  - Reconstitution du rayage de lignes et de colonnes.
  - Filtrage du bruit aléatoire.
  - Correction des déformations géométriques.
  - Restauration.
  
- Les transformations changent l'impact visuel de l'image en améliorant le teneur de l'information.
  - Rehaussement d'image.
    - L'approche spatiale dont les méthodes opèrent directement sur l'ensemble des pixels composant l'image.
    - L'approche fréquentielle ou spectrale, basée sur la modification de la transformée de Fourier de l'image.
  - Compression de données.
  
- L'extraction de l'information, le zonage et la reconnaissance consiste à identifier et à regrouper des pixels par classes thématiques sur la base de leurs signatures numériques.
  - Production des images par composante principale.
  - Segmentation.
  - Classification.



**Figure II-1 :** Les phases de traitement d'une image

Les traitements d'images numérisées, se répartissent comme suit :

- Les **opérateurs ponctuels**, qui agissent sur les pixels indépendants.
- Les **opérateurs locaux**, qui définissent une nouvelle valeur de pixel en examinant son voisinage.
- Les **opérateurs globaux**, qui prennent en compte toute l'image donnée pour déterminer chaque pixel de l'image résultat.



**Figure II-2 :** Transformations de pixels

Ces opérateurs sont utilisés dans le rehaussement des images qui a pour objectif de rendre les images plus lisibles et d'exploiter au mieux les nuances présentes dans l'image.

Deux logiques sous-tendent au recours de techniques de rehaussement :

- Une logique purement visuelle, visant à profiter au maximum des capacités de la vision humaine et de celles des outils de visualisation (écrans, imprimantes).
- Une logique scientifique, visant à mettre en évidence les éléments pertinents de l'image en fonction d'une problématique et à l'inverse, à masquer les éléments moins significatifs [8].

Dans ce mémoire de projet de fin des études, les méthodes principales mathématiques utilisées dans l'amélioration des images notamment les opérations du filtrage spatiales sont décrites et illustrées dans les chapitres qui suivront.

## II-2- Filtrage des images :

Pour améliorer la qualité visuelle de l'image, on doit éliminer les effets des bruits (parasites) en lui faisant subir un traitement appelé filtrage.

En fait, on pourrait comparer le travail de filtrage d'une image à celui du chercheur d'or tamisant les graviers d'une rivière : selon le calibre des pierres recherchées, le prospecteur utilisera des tamis avec des mailles plus ou moins grossières. Pour ne garder que les plus grosses pierres, il faudra un tamis à mailles larges ; en imagerie numérique, ce travail consiste à ne conserver que les détails les moins fins de l'image est appelé un filtrage **passé-bas**. Au contraire, pour sélectionner les cailloux les plus fins, il faudra plusieurs tamis successifs pour éliminer les pierres trop grosses ; ce que l'on appelle un filtrage **passé-haut**.

Dans une image les détails se répètent fréquemment sur un petit nombre de pixels, on dit qu'ils ont une **fréquence spatiale élevée** c'est le cas des bords et des contours dans une image. Au contraire, les **fréquences spatiales basses** correspondent à des variations qui se répètent peu car, diluées sur de grandes parties de l'image.

La plupart des filtres agissent de manière sélective sur ces fréquences pour les sélectionner, en vue de les amplifier ou de les réduire.

Contrairement aux manipulations d'histogrammes, qui sont des transformations ponctuelles sur tous les pixels de l'image, les filtres sont des opérations mathématiques appelés produits de convolution qui mettent en jeu l'environnement (voisinage) de chaque pixel.

### II-2-1- Convolution :

Filtrer une image consiste à convoluer sa fonction d'intensité  $f$  avec une fonction  $h$  appelée réponse impulsionnelle du filtre.

$$g(x, y) = h(x, y) * f(x, y) \quad (\text{II-1})$$

Le filtrage linéaire consiste donc à modifier des valeurs de l'image par application d'un opérateur dans le domaine de l'espace image :

$$g(x, y) = ( \text{filtre} * f)(x, y) = \sum_i \sum_j \text{filtre}(i, j) \text{filtre}(x-i, y-j) \quad (\text{II-2})$$

$f$  : Image originale.

$h$  : Filtre.

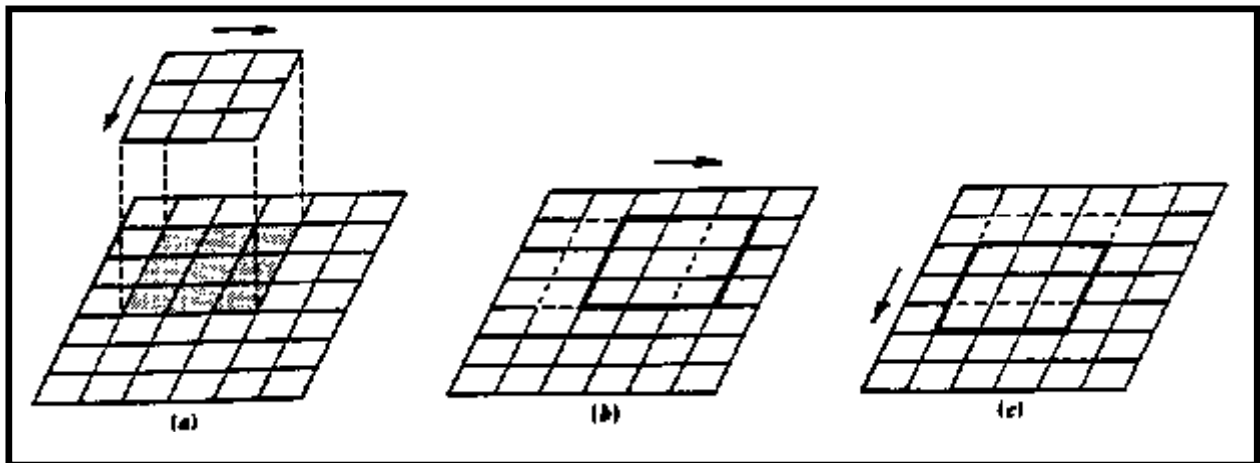
$g$  : Image modifiée.

\* : loi de convolution.

$$g(x, y) = \sum_{i=-(DF-1)/2}^{(DF-1)/2} \sum_{j=-(DF-1)/2}^{(DF-1)/2} \text{filtre}(i, j) \text{filtre}(x-i, y-j) \quad (\text{II-3})$$

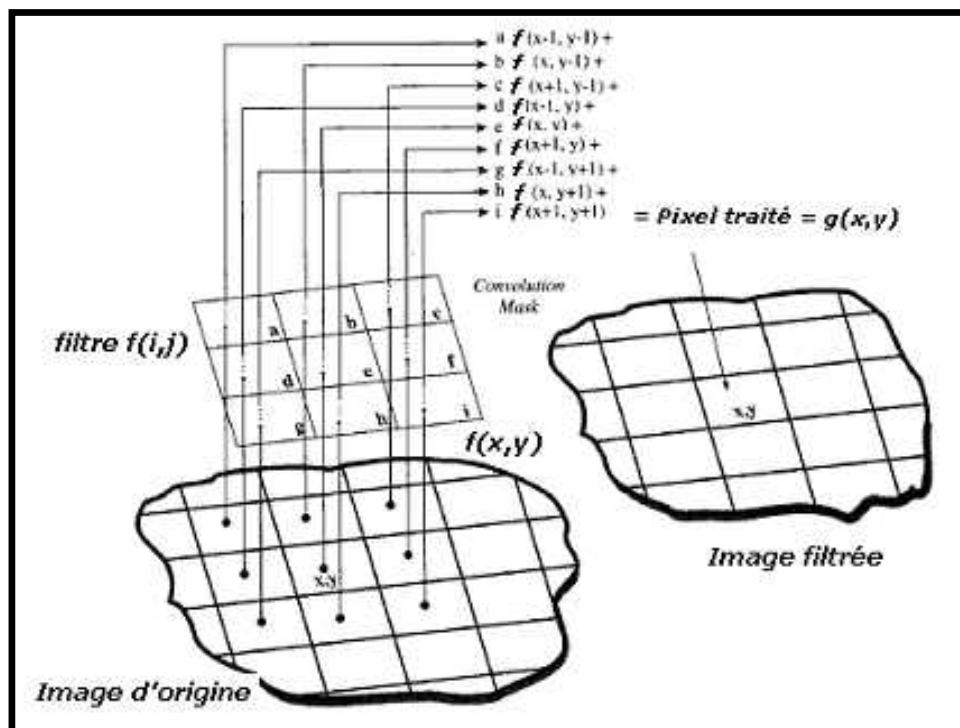
La méthode consiste à déplacer une "fenêtre" d'une dimension en général de (3 sur 3, 5 sur 5, ..., 2p+1 sur 2p+1) au-dessus de chaque pixel de l'image. On applique alors la convolution pour le remplacement de la valeur du pixel par une combinaison linéaire de ses voisins dans la fenêtre. Cela correspond à la série d'opérations (additions, multiplications) arithmétiques appliquées localement sur les valeurs des pixels dans la fenêtre de base. Après ce traitement mathématique, on remplace la valeur du pixel central par le résultat obtenu.

La fenêtre est déplacée le long des colonnes et des lignes de l'image, un pixel à la fois, répétant le calcul jusqu'à ce que l'image entière ait été filtrée. En modifiant le calcul effectué à l'intérieur de la fenêtre, il est possible de rehausser ou de supprimer différents types de caractéristiques présents dans une image.



**Figure II-3 :** Concept du filtrage

En général le masque qu'on applique est de dimension DF impair et symétrique.

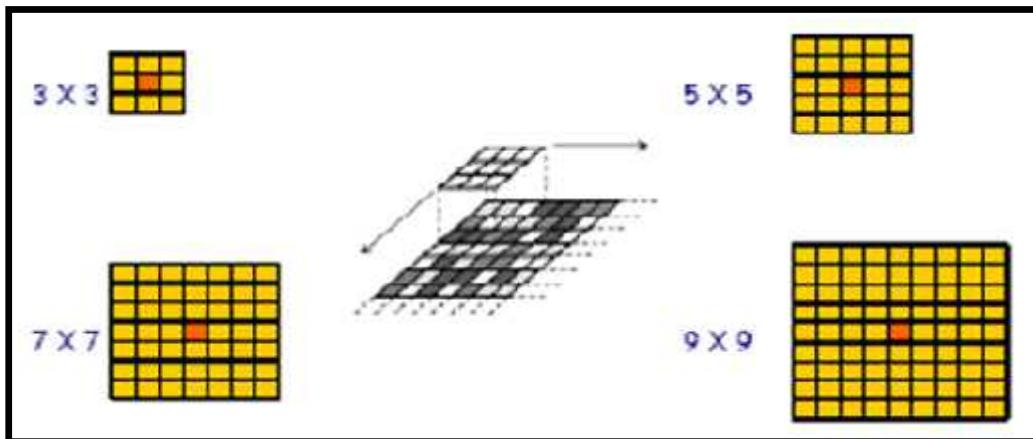


**Figure II-4 :** Principe de la convolution (cas DF=3)

### II-2-2- Filtres et Fenêtres :

Le filtre appelé noyau de convolution, correspond à la fenêtre dans laquelle le rehaussement est appliqué.

Théoriquement, cette matrice peut prendre n'importe quelle forme et taille, mais on utilise généralement des filtres composés d'un nombre impair de pixels.



**Figure II-5** : Taille d'un filtre

La technique du filtrage est basée sur la relation statistique entre le pixel central  $(x,y)$  et les pixels environnants.

Les fenêtres de filtrages sont des sous-ensembles de pixels  $X$  par  $Y$  se déplaçant dans l'image. La réduction du bruit par filtrage consiste à extraire une fenêtre de quelques pixels au-dessus de chaque pixel de l'image.

La taille de la fenêtre de filtrage doit être impaire et peut atteindre la dimension maximale de  $33 \times 33$  (cas des images radars), l'emploi d'une grande fenêtre signifie qu'une vaste portion de l'image sera utilisée dans le calcul, ce qui nécessitera en fonction de la complexité de l'algorithme un temps de traitement élevé.

Le fait que la fenêtre de filtrage soit grande, peut provoquer une perte de détails dans l'image due à un lissage excessif alors qu'une fenêtre de filtrage trop petite peut s'avérer inefficace dans la réduction du chatoyement. En pratique, on utilise en général des fenêtres de filtrage d'ordre  $3 \times 3$ ,  $5 \times 5$  ou  $7 \times 7$ .

$x-1, y-1$	$x-1, y$	$x-1, y+1$
$x, y-1$	<b><math>x, y</math></b>	$x, y+1$
$x+1, y-1$	$x+1, y$	$x+1, y+1$

**Tableau II-1** : Fenêtre  $3 \times 3$

Un filtre idéal est supposé atteindre les objectifs suivants :

- Eliminer le bruit, donc lisser le plus possible les zones.
- Conserver et rehausser les structures de l'image (lignes, contours, zones à très fortes réflectivités...).
- Préserver la texture de la scène [9].

Parmi les filtres spatiaux, on distingue : les **filtres passe-bas** (lissage), filtres **passe-haut** (accentuation) et les filtres **passe-bande** (différenciation). Ces filtres peuvent être linéaires ou non linéaires.

Un filtre non linéaire affectera au pixel central une valeur qui n'est pas une combinaison linéaire des pixels voisins.

### II-2-3- Filtres Passe-bas (Lissage) :

Ces filtres n'affectent pas les composantes de basse fréquence dans les données d'une image, mais doit atténuer les composantes de haute fréquence.

L'opération de lissage est souvent utilisée pour atténuer le bruit et les irrégularités de l'image. Elle peut être répétée plusieurs fois, ce qui crée un effet de flou. En pratique, il faut choisir un compromis entre l'atténuation du bruit et la conservation des détails et contours significatifs.

**Inconvénient** : On lisse aussi les contours.

#### II-2-3-1- Filtre moyenne :

C'est un cas particulier de filtre de convolution « passe-bas », qui remplace chaque pixel par la valeur moyenne de ses voisins (le pixel lui-même y compris). Pour un voisinage  $S$  de  $(x,y)$  on a :

$$g(x, y) = \text{moyenne} \left\{ f(n, m) \right\}_{(n,m) \in S} \quad (\text{II-4})$$

Cette méthode permet de **lisser** les images, c'est à dire de diminuer les différences de niveaux de gris entre pixels voisins en modifiant les niveaux de gris trop différents de leur entourage direct. Cette méthode très simple est censée supprimer le bruit représenté par des niveaux de gris **anormaux**.

Afin de conserver la moyenne de l'image  $f(x, y)$ , la somme des éléments du filtre est normalisée à 1.

**Les inconvénients** évidents de ce filtre de moyenne sont les suivants :

- Un pixel isolé avec un niveau de gris anormal pour son voisinage va perturber les valeurs moyennes des pixels de son voisinage.
- Sur une frontière de régions le filtre va estomper le contour et le rendre flou, ce qui est gênant en visualisation bien sûr mais éventuellement aussi pour un traitement ultérieur qui nécessiterait des frontières nettes.
- Plus le filtre grossit plus le lissage devient important et plus le flou s'accroît.

### II-2-3-2- Filtre mode :

Le filtre mode calcule le mode statistique des niveaux de gris dans la fenêtre du filtre qui entoure le pixel à traiter. La dimension de la fenêtre est impaire et peut au maximum être égale à 7x7. Pour un voisinage S de (x,y) :

$$g(x, y) = \operatorname{mode} \{f(n, m)\}_{(n, m) \in S} \quad (\text{II-5})$$

### II-2-3-3- Filtre Gaussien :

Il a été utilisé par David Marr, l'un des pionniers du traitement de l'image. Le lissage par un filtre gaussien est justifié lorsque l'on ne connaît rien a priori sur les interactions entre pixels de l'image. On fait donc l'hypothèse d'une loi d'interaction moyenne, c'est à dire gaussienne.

Le filtre Gaussien est un filtre isotrope (les propriétés de leur noyau de convolution sont identiques dans toutes les directions) spécial avec des propriétés mathématiques bien précises.

La fonction Gaussienne est définie par :

$$g(x, y) = \text{Gaussienne}(x, y) = e^{-\frac{f(x^2+y^2)}{f^2}} \quad (\text{II-6})$$



Le paramètre sigma s'appelle la déviation standard, et détermine la largeur de la cloche Gaussienne.

L'effet de ce filtre sur l'image est assez similaire au filtre moyenne, mais la moyenne est pondérée en ce sens ou les pixels près du centre ont un effet ou un "poids" plus important que ceux qui sont situés plus loin.

En général un filtre Gaussien avec un  $\sigma < 1$  est utilisé pour réduire le bruit, et si  $\sigma > 1$  c'est dans le but de fabriquer une image qu'on va utiliser pour faire un "masque flou" personnalisé qu'on applique souvent aux images satellitaires. Il faut noter que plus le sigma est grand, plus la cloche Gaussienne est large et plus le flou appliqué à l'image sera marqué.

Idéalement on devrait prévoir un filtre de taille  $(2\sigma + 1) \times (2\sigma + 1)$ .

**Les inconvénients** du filtre gaussien sont :

- Gros lissage  $\Rightarrow$  filtre très large  $\Rightarrow$  gros problèmes de bord, et gros coût de calcul.
- Mélange de structures voisines.
- Perte de contraste.
- Délocalisation des bords.

#### **II-2-3-4- Filtre médian :**

Le filtre médian est utilisé pour atténuer des pixels isolés, d'une valeur très différente de leur entourage. C'est le cas par exemple des pixels thermiques chauds ou froids qui ont une intensité très différente de leurs voisins.

Le filtre médian n'est pas à proprement parler un produit de convolution, mais sa mise en oeuvre sur l'image est assez similaire puisqu'un noyau est appliqué sur l'image et collecte les valeurs des pixels.

Le résultat obtenu avec ce filtre est l'élimination des pixels isolés, en évitant de créer un flou trop important dans l'image.

Par contre, à cause de ses propriétés non-linéaires il peut ne pas préserver les qualités photométriques et astrométriques de l'image originale.

Le filtre médian réalise un lissage de l'image un peu plus performant que le filtre moyenne en ce qui concerne les détails dans l'image.

Chaque pixel est traité en considérant ses voisins sur un voisinage donné. Le pixel lui-même et ses voisins forment alors un ensemble dont on calcule la « médiane ». Le pixel sera alors remplacé par cette valeur médiane.

$$g(x, y) = \underset{(n,m) \in S}{\text{médian}} \{f(n, m)\} \quad (\text{II-7})$$

**Intérêt** du filtre médian :

- Un pixel non représentatif dans le voisinage affectera peu la valeur médiane.
- Elimine les bruits de type sel & poivre (salt-and-pepper).
- La valeur médiane choisie étant le niveau de gris d'un des pixels considérés, on ne crée pas alors de nouveaux niveaux de gris dans l'image. Ainsi lorsque le filtre passe sur un contour très marqué il le préservera mieux.

**Inconvénients** du filtre médian:

- Coûteux et nécessite un tri avant l'opération du filtrage.

Vu que la médiane, en tant que mesure statistique, représente une alternative à la moyenne, le filtre médian constitue une forme de lissage qu'on utilise à la place du lissage linéaire.

### II-2-3-5- Filtre médian à 3 états :

Le principe du filtre médian à 03 états est, dans un premier temps, il faut déterminer si le pixel est corrompu ; puis lui appliquer soit un filtre médian, soit un filtre médian pondéré, soit de le laisser inchangé. Le filtre médian standard permet effectivement la suppression du bruit impulsionnel, mais efface les détails fins et arrondit les angles. Afin de trouver un bon compromis entre les détails et la suppression du bruit, Ko et Lee ont proposé le filtre CWM (Center Weight Median), dans lequel on affecte un poids  $w_c$  plus ou moins important au pixel central de la fenêtre d'étude.

$$g(x, y) = \underset{W}{\text{médian}} (w_c \cdot f(x, y)) \quad (\text{II-8})$$

W : est la fenêtre carrée qui entoure le pixel que l'on étudie.

**Remarques :**

- Si  $w_c = 1$ , le filtre CWM est en fait un simple filtre médian.
- Si  $w_c$  est supérieur ou égal à la taille de la fenêtre alors le filtre CWM devient le filtre "identité".

Les étapes de l'algorithme sont les suivantes :

- Pour chaque pixel de l'image, on détermine à quel point celui-ci est bruité en lui appliquant les filtres CWM et SM (médian standard) et en comparant les valeurs obtenues avec la valeur du pixel d'origine.
- Trois valeurs sont possibles pour remplacer le pixel étudié : la valeur d'origine (ie le pixel n'est pas bruité), la valeur de sortie  $g(x, y)^{SM}$  par le filtre SM, et la valeur de sortie  $g(x, y)^{CWM}$  par le filtre CWM.
- On choisit l'une de ces trois sorties en utilisant un seuil T comme suit :

$$g(x, y) = \begin{cases} f(x, y) & \text{si } T \geq d_1 \\ g(x, y)^{CWM} & \text{si } d_2 \leq T < d_1 \\ g(x, y)^{SM} & \text{si } T < d_2 \end{cases} \quad (\text{II-9})$$

Où :  $d_1 = |f(x, y) - g(x, y)^{SM}|$ , et  $d_2 = |f(x, y) - g(x, y)^{CWM}|$

**II-2-3-6- Filtre maximum (conservative smoothing) :**

Ce filtre de lissage est non linéaire, il supprime bien le bruit de type "poivre et sel" c'est à dire qu'il "adoucit" les pixels isolés ayant un niveau de gris très différent des niveaux de gris de leur voisinage et il a la particularité de bien préserver les contours très marqués.

Ce filtre s'assure en fait que tout pixel a son niveau de gris placé dans la gamme de ses voisins.

[10].

**Méthode:**

On considère le niveau de gris du pixel à traiter, et d'autre part tous ses voisins (à l'exception de lui même).

---

Sur les voisins on calcule le niveau min et le niveau max., si le niveau de gris du pixel à traiter est compris entre le min et le max, alors on le laisse inchangé sinon on le remplace par le max.

#### **II-2-4- Filtres Passe- haut (Accentuation) :**

Le filtre digital passe-haut a les caractéristiques inverses du filtre passe-bas .Ce filtre n'affecte pas les composantes de haute fréquence d'un signal, mais doit atténuer les composantes de basse fréquence.

Les filtres passe-haut ont pour rôle essentiel de mettre en évidence les zones transitoires comme la frontière entre deux régions (contour). Ils sont basés sur la différenciation de pixels voisins (nulle si ceux-ci sont de même intensité, importante quand ils sont à cheval sur 2 régions distinctes).

Dans le cas de la détection de contours, l'image est perçue comme un paysage où la topographie du relief est étudiée afin de mettre en évidence la limite, le voisinage et la discontinuité. Lorsqu'une brusque variation de luminance (ou de niveau de gris) est détectée, celle-ci fait l'objet d'un pourtour (droite, arc de cercle, courbe gauche) caractérisant une forme ou un objet. L'extraction de contours, d'un point de vue algorithmique, est une fonction économique qui permet d'alléger considérablement les processus de reconnaissance d'objet et de localisation tridimensionnelle.

Il existe cinq types de contours modélisés dans l'ensemble des littératures. Cela permet de caractériser une discontinuité par l'étude de la variation de luminance. La difficulté réside à définir le type de détecteur. Il n'existe pas de détecteur de contours universel. Car chaque image est différenciée par sa nature (luminosité, environnement extérieur ou intérieur...). Ainsi l'image se verra attribuer un détecteur spécifique correspondant au mieux aux exigences et aux besoins des traitements de haut niveau (reconnaissance, localisation...).

Deux approches :

- Approche gradient : détermination des extréma locaux dans la direction du gradient.
- Approche laplacien : détermination des passages par zéro du laplacien.

Ces approches reposent sur le fait que les contours correspondent des discontinuités d'ordre 0 de la fonction d'intensité.

Les différentes approches existantes se classent ensuite suivant la manière d'estimer les dérivées de la fonction d'intensité :

- Différences finies.
- Filtrage optimal.
- Modélisation de la fonction d'intensité.

### II-2-4-1- Filtre Gradient :

Un gradient permet de visualiser les variations d'un phénomène, ainsi un dégradé de couleur peut s'appeler un gradient de couleur.

Mais parfois il est intéressant d'avoir de mettre en avant des détails qui changent selon une direction donnée, dans ce but on utilise des filtres de type gradient, qui sont en fait des dérivées partielles le long d'une direction particulière, en général l'un des deux axes cartésiens X ou Y de l'image.

Si Les dérivées partielles sont supposées continues en un point quelconque de l'image, alors il est admis que la différentielle de la fonction Image en un point  $M(x,y)$  est une fonction linéaire qui est notée :

$$dz = f'_x(x,y) \cdot dx + f'_y(x,y) \cdot dy \quad (\text{II-10})$$

Écrit plus généralement sous cette forme :

$$dz = \frac{\partial f(x,y)}{\partial x} \cdot dx + \frac{\partial f(x,y)}{\partial y} \cdot dy \quad (\text{II-11})$$

$dz$  ou  $\nabla f(x,y)$  : est appelé Gradient et a pour amplitude :

$$\|\nabla f(x,y)\| = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2} \quad (\text{II-12})$$

Sa direction est donnée par :

$$\phi = \text{Arc tan} \frac{\frac{\partial f(x,y)}{\partial y}}{\frac{\partial f(x,y)}{\partial x}} \quad (\text{II-13})$$

L'analyse du vecteur gradient permet de mettre en évidence un contour. En effet l'amplitude du gradient indique une plus ou moins forte discontinuité et sa direction est par définition normale au contour. Le gradient est un opérateur directionnel.

Localement à un point de l'image, les dérivées partielles font l'objet d'une approximation par une méthode de différentiation.

$$\frac{\partial f(x, y)}{\partial x} \text{ est évalué par } \Delta_x f(x, y) = f(x+1, y) - f(x, y) \quad (\text{II-14})$$

$$\frac{\partial f(x, y)}{\partial y} \text{ est évalué par } \Delta_y f(x, y) = f(x, y+1) - f(x, y) \quad (\text{II-15})$$

D'un point de vue géométrique, le gradient évalué sur un point indique la direction où la variation des niveaux de gris est la plus grande. Ce vecteur a une longueur très faible sur les points se trouvant à l'intérieur d'une région homogène.

Par contre, sur les points se trouvant à la limite entre deux régions, le gradient a une longueur importante, il est orienté vers la région la plus claire et se trouve perpendiculaire au contour.

Ainsi un filtre Gradient permet de mettre en évidence les variations de niveaux de gris suivant un axe variable, ce qui aura pour effet de mettre en évidence les fronts et de révéler les textures.

Pour une direction donnée, des filtres Gradient seront utilisés pour augmenter ou bien réduire les fronts sur cette direction, ce sont en fait des dérivées partielles le long d'une direction particulière, en général l'un des deux axes cartésiens X ou Y de l'image.

D'un point de vue pratique, les deux dérivées partielles suivant la largeur et la hauteur peuvent être approximées respectivement par des opérateurs de convolution. Plus le noyau de convolution est grand, plus larges sont les contours.

### a- Principe du Gradient :

- **Gradient :**

Soit  $f(x, y)$ , alors :

$$\nabla f = \begin{pmatrix} G_x \\ G_y \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix} \quad (\text{II-16})$$

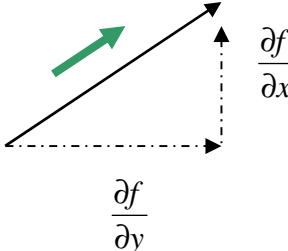
- **Magnitude du gradient :**

$$\text{mag}(\nabla f) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (\text{II-17})$$

- **Approximation de la Magnitude :**

$$\text{mag}(\nabla f) \approx \left| \left(\frac{\partial f}{\partial x}\right) \right| + \left| \left(\frac{\partial f}{\partial y}\right) \right| \quad (\text{II-18})$$

- **Direction du gradient :**

$$\theta = \text{Arc tan} \left[ \frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right] \quad (\text{II-19})$$


Les approximations les plus simples des dérivées directionnelles se font par différences finies calculées par convolution avec des noyaux très simples.

Dans le chapitre extraction de contour, plusieurs filtres de type gradient seront étudiés.

### **b- Filtres de PREWITT, SOBEL, ROBERTS et KIRSCH :**

Ces filtres, qui portent tous le nom de leurs inventeurs, sont tous conçus dans le même but : détecter avec la plus grande précision les contours naturels "cachés" dans une image.

A l'origine ils ont été développés dans le cadre des appareils de vision nocturne.

Le filtre de Sobel utilise par exemple deux noyaux 3x3, l'un pour l'axe horizontal (X) et l'autre pour l'axe vertical (Y) Chacun des noyaux est en fait un filtre gradient, qui sont tous les deux combinés pour créer l'image finale.

Ces masques sont généralement utilisés pour la détection des contours. Il s'agit des masques de Roberts, Sobel, Prewitt et Kirsh.

**b-1- Masques de Sobel :**

On fait passer le masque de convolutions pour extraire les contours horizontaux puis verticaux. On additionne ensuite les deux images obtenues.

$$W_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad W_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (\text{II-20})$$

$W_x$ : direction horizontale,  $W_y$ : direction verticale.

On remarque que Sobel est la combinaison d'un lissage dans un sens et d'un rehaussement dans l'autre :

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times [1 \ 0 \ -1] \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \times [1 \ 2 \ 1]$$

**b-2- Masques de Roberts :**

$$W_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad W_y = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad (\text{II-21})$$

$W_x$ : direction horizontale,  $W_y$ : direction verticale.

**b-3- Masques de Prewitt :**

$$W_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad W_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (\text{II-22})$$

$W_x$ : direction horizontale,  $W_y$ : direction verticale.



**b-4- Masques de Kirsh :**

$$W_x = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad W_y = \begin{bmatrix} -3 & -3 & 3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad (\text{II-23})$$

$W_x$ : direction horizontale,  $W_y$ : direction verticale.

**b-5- Masques de Roberts pour 4 directions :**

$$W_{0^\circ} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad W_{45^\circ} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -2 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (\text{II-24})$$

$$W_{90^\circ} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad W_{135^\circ} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

**b-6- Masques de Sobel pour 4 directions :**

$$W_{0^\circ} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad W_{45^\circ} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad (\text{II-25})$$

$$W_{90^\circ} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad W_{135^\circ} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

**b-7- Masques de Prewitt pour 4 directions :**

$$\begin{aligned}
 W_{0^\circ} &= \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} & W_{45^\circ} &= \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \\
 W_{90^\circ} &= \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} & W_{135^\circ} &= \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}
 \end{aligned}
 \tag{II-26}$$

**b-8- Masques de Kirsh pour 4 directions :**

$$\begin{aligned}
 W_{0^\circ} &= \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & W_{45^\circ} &= \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \\
 W_{90^\circ} &= \begin{bmatrix} -3 & -3 & 3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} & W_{135^\circ} &= \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}
 \end{aligned}
 \tag{II-27}$$

**II-2-4-2- Filtre Laplacien :**

Le Laplacien de la fonction Image est défini par :

$$\nabla^2 f = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}
 \tag{II-28}$$

Ainsi en utilisant également les propriétés de différentiation l'opérateur Laplacien peut s'écrire sous la forme suivante :

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)
 \tag{II-29}$$

Dans le cas du traitement d'image, l'image de départ  $f(x, y)$  n'est pas une fonction continue, mais une fonction discrète à cause de la numérisation effectuée. Mais on peut tout de même obtenir la dérivée seconde en bonne approximation.

Il existe trois noyaux typiques de taille 3x3 qui peuvent remplir ce rôle :

0	-1	0
-1	4	-1
0	-1	0

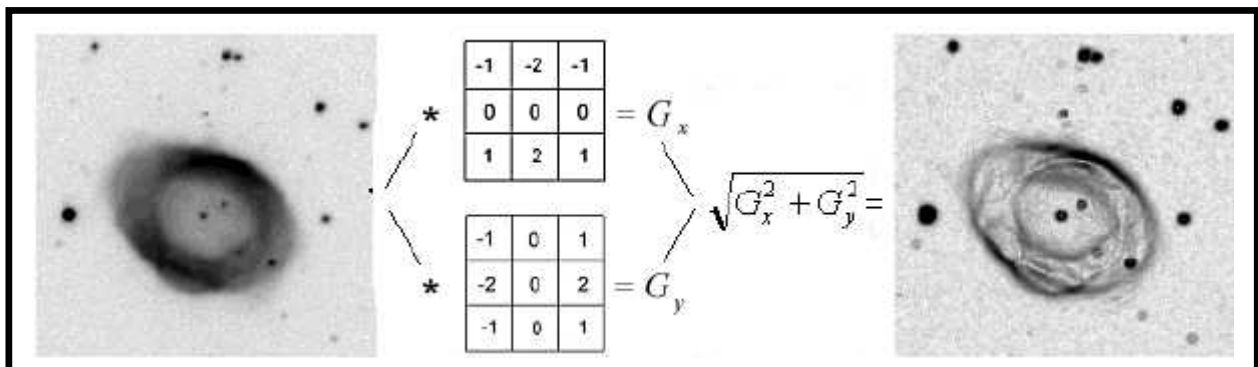
-1	-1	-1
-1	8	-1
-1	-1	-1

1	-2	1
-2	4	-2
1	-2	1

**Tableau II-2 :** Les trois noyaux 3x3 de l'opérateur Laplacien

Nous remarquons que l'opérateur Laplacien est symétrique quelque soit l'orientation choisie, il n'est donc pas directionnel. En effet, si par hypothèse de départ, la fonction image est supposée continue. Alors les propriétés de la dérivée seconde d'une fonction sont utilisées pour caractériser un contour par la présence d'un extremum local par le passage à zéro de la dérivée seconde.

Le filtre Laplacien est un filtre de convolution particulier utilisé pour mettre en valeur les détails qui ont une variation rapide de luminosité, il est fortement sensible aux bruits. Ce qui nous conduisons à essayer d'atténuer les bruits par l'emploi de filtre de lissage. Le Laplacien est donc idéal pour rendre visible les contours des objets, d'où son utilisation dans la reconnaissance de formes dans des applications militaires, puis civiles.



**Figure II-6 :** Exemple d'application d'un filtre Laplacien sur une image astronomique

**II-2-4-3- Filtre compas :**

$$\begin{array}{cccc}
 \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} & \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix} \\
 \uparrow N & \nwarrow NW & \leftarrow W & \searrow SW
 \end{array}$$

(II-30)

$$\begin{array}{cccc}
 \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix} \\
 \downarrow S & \swarrow SE & \rightarrow E & \nearrow NE
 \end{array}$$

Le gradient est défini par :

$$g(x, y) = \max_k \left| g_k(x, y) \right| \quad (\text{II-31})$$

**II-2-4-4- Filtre de Canny :**

Le détecteur de bords de Canny représente une approche un peu plus sophistiquée à la détection de bords. Il repose sur le fait que pour n'importe quel détecteur de bords, il y a un compromis entre la réduction du bruit et la détection de bords. Une réduction améliorée du bruit est réalisée au détriment d'une bonne localisation et vis versa. Le détecteur de Canny fournit le meilleur compromis entre ces deux conditions. Le filtre de Canny est optimal pour des bords en forme de marche en présence de bruit gaussien. Bien sûr dans la réalité les bords ne sont pas tous des marches et le bruit dans la nature n'est pas purement gaussien. Cependant le détecteur de Canny fournit de bons résultats avec des images réelles. Dans Canny la dérivée de la gaussienne est approchée par :

$$\psi(x) \approx -x \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (\text{II-32})$$

### II-2-4-5- Filtre de Marr Hildreth (Laplacien de la Gaussienne) :

Afin de limiter les réponses dues au bruit de l'image I, elle est préalablement filtrée par un filtre G. On obtient alors grâce aux propriétés de l'opérateur produit de convolution :

$$\nabla^2(I * G) = (\nabla^2 I) * G = I * (\nabla^2 G) \quad (\text{II-33})$$

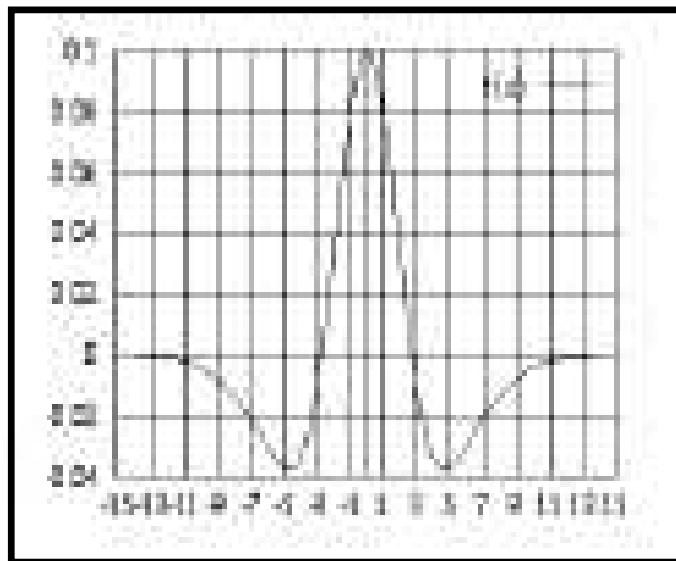
Les opérateurs de filtrage et de dérivation se font donc en une seule étape de calcul. Le plus souvent, on fait appel à un filtrage Gaussien :

$$G(r) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad r = \sqrt{x^2 + y^2} \quad (\text{II-34})$$

Dont le Laplacien est :

$$\nabla^2 G(r) = -\frac{1}{\sigma^4} \cdot \left(1 - \frac{r^2}{2\sigma^2}\right) \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (\text{II-35})$$

Plus connue sous le nom de « chapeau mexicain ».



**Figure II-7 :** Laplacien de la fonction de Gauss

Le paramètre  $\sigma$  sert à régler la résolution à laquelle les contours sont détectés.

Dans le cas discret, la plus simple approximation du Laplacien d'une Gaussienne est le filtre connu sous le nom de masque Laplacien.

Le filtre obtenu par convolution avec le Laplacien d'une gaussienne est connu sous le nom de LOG [11].

# **Chapitre III**

## **Implémentation et résultats**

### **III-1- Introduction :**

Ce chapitre est destiné à mettre en œuvre des algorithmes qui représentent les équations des filtres cités précédemment, l'implémentation de ces filtres est faite sous le langage de programmation MATLAB.

### **III-2- Plate forme de travail :**

Le matériel utilisé est un ordinateur Pentium IV (système d'exploitation Windows XP), la fréquence de son microprocesseur est 2400MHz et sa RAM est de 256 Mo.

Tous les algorithmes analysés ont été programmés sous MATLAB 7.1.

Ce langage de programmation est un environnement puissant, complet et facile à utiliser, destiné au calcul scientifique. Il est basé sur des techniques de raisonnement matriciel, développé au sein de la société MATHWORKS.

### **III-3- Choix de MATLAB :**

MATLAB intègre des routines dédiées au traitement des images et apporte à son utilisateur une aisance dans le calcul numérique et la visualisation.

Le traitement d'images permet de modifier le contenu des images afin de tirer l'information utile pour une application particulière. MATLAB offre de nombreuses possibilités de traitement avec une palette très fournie d'outils prêts à l'emploi.

L'inconvénient majeur de MATLAB réside dans sa relative lenteur pour effectuer certaines opérations de calculs (par exemple la transformée de Fourier). Toutefois MATLAB permet de déployer rapidement des tests pour vérifier la validité d'une méthode de traitement d'images. La manipulation d'images revient à la manipulation de matrices qui est très facile grâce au langage de haut niveau de MATLAB.

MATLAB est capable de lire et de décoder les fichiers images JPEG, TIFF, BMP, PCX,... Une image sous MATLAB peut être représentée sous plusieurs formes, mais toujours sous forme d'une matrice. Avant de traiter une image en MATLAB, il faut la lire et décoder son format afin de la transformer en une matrice de valeurs.



Les valeurs des images lues sous MATLAB sont entières, mais dans certaines circonstances, on a besoin de travailler sur des valeurs réelles. La transformation pour passer d'entier à réel utilise la fonction **im2double**.

La couleur est une donnée importante pour une image, elle modifie la perception que l'on a de l'image. L'espace de représentation standard décompose une image en trois plans de couleur: le rouge, le vert et le bleu - Red/Green/Blue RGB en anglais. Les couleurs finales sont obtenues par synthèse additive de ces trois couleurs primaires. Il existe cependant des problèmes qui peuvent nécessiter de changer d'espace de couleur pour percevoir différemment l'image.

Il y a des images où la couleur importe peu, par exemple des photographies de cellules vivantes (pseudo-transparentes), des images radar, des images satellites...

Dans ce cas, l'espace RGB n'est plus utilisé. Avec MATLAB, on peut choisir d'autres espaces comme HSV Hue/Saturation/Value ou YCbCr Luminance/Chrominance bleue/Chrominance rouge.

MATLAB permet le passage d'une image couleur à une image en niveaux de gris. La fonction **rgb2gray** permet d'obtenir une image en niveaux de gris en partant d'une image couleur. Dans certains cas, cette transformation est très utile

Les opérations géométriques classiques sont permises avec la boîte à outils de traitement d'images: rotation, changement de taille, découpage...

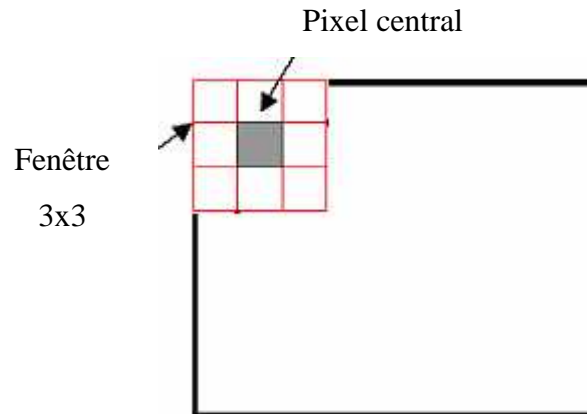
La détection de contours permet de repérer les différents objets qui constituent la scène de l'image. MATLAB intègre de nombreuses méthodes pour trouver les contours des objets, la plupart sont basées sur les dérivées premières et secondes de l'image [12].

#### **III-4- Image Utilisée :**

L'image que nous avons choisie pour effectuer les comparaisons est une image aérienne montrant le survol d'un avion de combat **Mig29SMT**, cette image représente une riche palette de textures (ciel, avion, terrains, routes au sol, ...etc.).

Il semble très intéressant du point de vue de variétés des zones homogènes et hétérogènes.

Dans notre approche, nous avons employé une image de taille (576x768) pixels, le filtre choisi commencera son calcul dans la fenêtre du filtre à partir du coin gauche supérieur de l'image.



**Figure III-1** : Exemple du filtrage par fenêtre 3x3

Si le filtre choisi est un filtre médian par exemple, la valeur du premier pixel dans l'image initiale sera remplacée par la valeur médiane de ses pixels environnants dans la fenêtre du filtre. On emploie la même approche pour essayer de remplacer tous les pixels pendant que la fenêtre glisse à travers l'image comme représentée sur figure (III-1).

### III-5- Algorithmes et résultats :

#### III-5-1- Bruitage d'image :

L'image utilisée contient plusieurs types de zones pour une meilleure appréciation des résultats des filtres appliqués.

Soient : **I**= image originale et **J** = image bruitée, premièrement on injecte différents types de bruit de par les fonctions prédéfinies suivantes :

- **J= IMNOISE(I,'speckle',V)**

V : variance de bruit injecté.

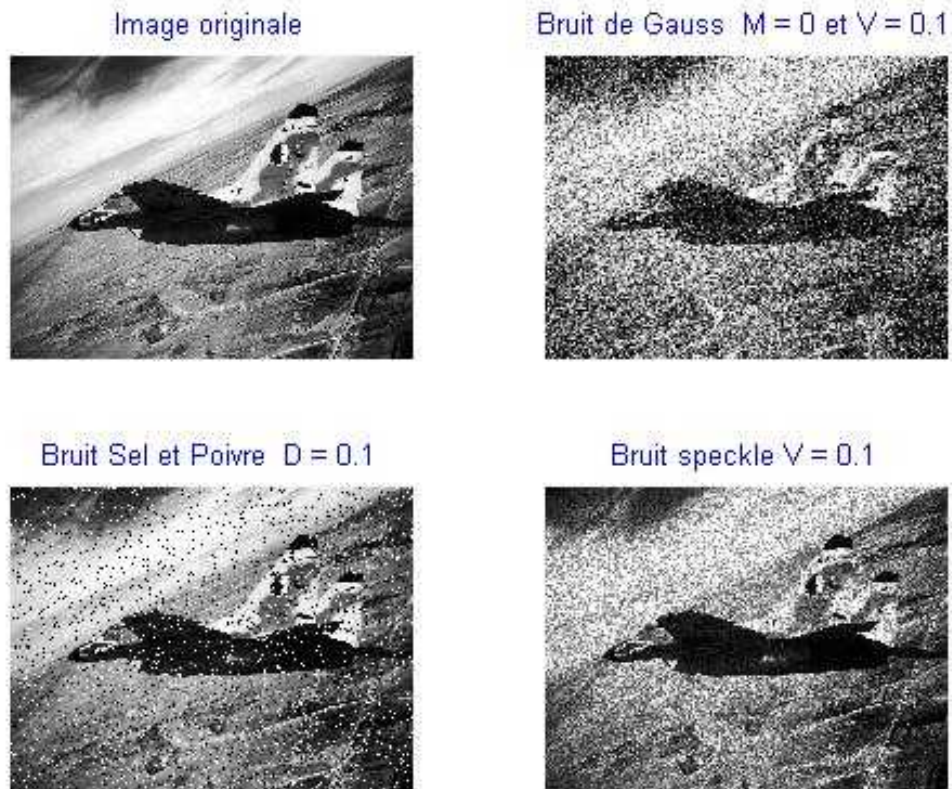
On rajoute un bruit de type multiplicatif à l'image I en utilisant l'équation  $J = I + n \cdot I$ , n étant un bruit aléatoire uniformément distribué, de moyenne nulle et de variance V. par défaut  $V = 0.04$ .

- **J = IMNOISE(I,'gaussian',M,V)**

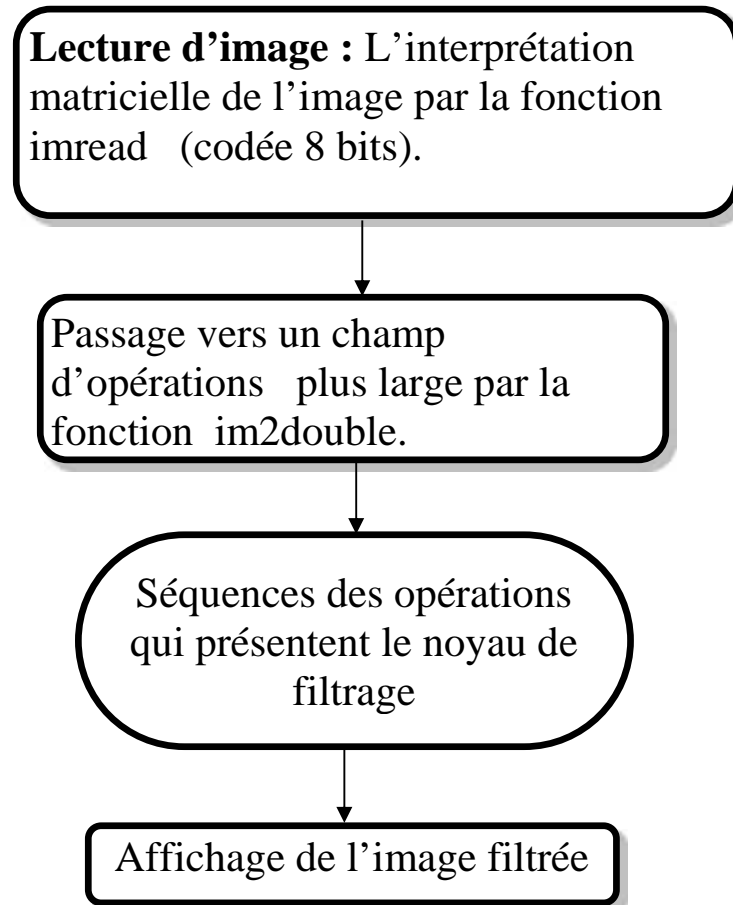
On rajoute à l'image I un bruit blanc de densité gaussienne, de moyenne M et de variance V à l'image I. par défaut vaut zéro et  $V = 0.01$ .

- **J = IMNOISE(I,'salt & pepper',D)**

On rajoute à l'image I un bruit de type sel et poivre ayant l'aspect d'une granulation ("salt and pepper"). D étant la densité du bruit égale par défaut à 0.05.



**Figure III-2 : Bruitage d'image**

**III-5-2- Chaîne de filtrage :****Figure III-3** : Organigramme de chaîne du filtrage**III-5-3- Calcul des paramètres de mesures statistiques :**

On a utilisé l'algorithme (cité au-dessous) qui permet de calculer les paramètres de l'image bruitée  $I(i,j)$  :

$M$  = Moyenne de l'image.

$S$  = Ecart type de l'image.

$MSE$  = Erreur quadratique moyenne.

**Algorithme :****Début**

I (i,j)

som = 0;

pour i = 1 jusqu'à n

pour j = 1 jusqu'à m

som = som+I(i,j);

fin

fin

 $\mathbf{M} = \text{som}/(\text{nxm})$ 

som = 0;

pour i = 1 jusqu'à n

pour j = 1 jusqu'à m

som = som+(I(i,j)^2-M^2);

fin

fin

 $\mathbf{S} = \sqrt{\text{som}/(\text{nxm})}$ 

som = 0;

pour i = 1 jusqu'à n

pour j = 1 jusqu'à m

som = som+(I(i,j) -  $\hat{R}(i, j)$ )^2;

fin

fin

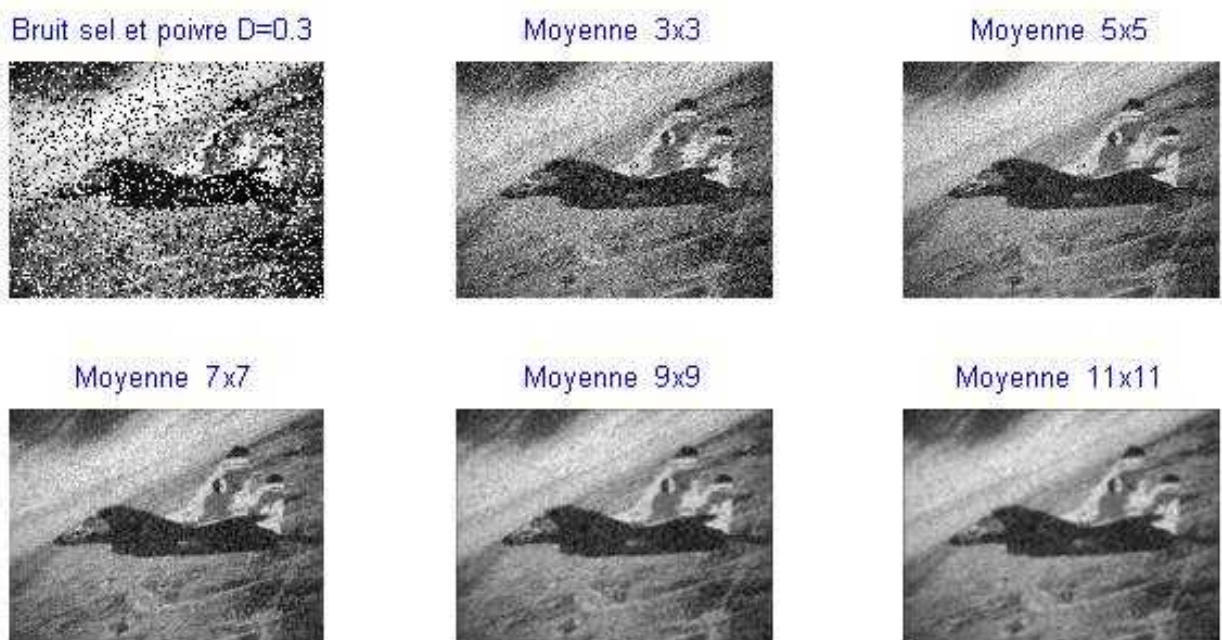
 $\mathbf{MSE} = \text{som}/\text{nxm}$ **Fin**

### III-5-4- Influence de la taille de la fenêtre sur l'effet des filtres passe bas :

#### III-5-4-1- Filtre moyenne :

On applique un filtre moyenne sous plusieurs tailles de fenêtres ((3x3), (5x5), (7x7), (9x9), (11x11)) contre le bruit sel et poivre  $D = 0.3$ .

La Figure (III-4) montre que l'application d'un filtre moyenne sous plusieurs fenêtres donne une dégradation à mesure que la dimension du noyau augmente.



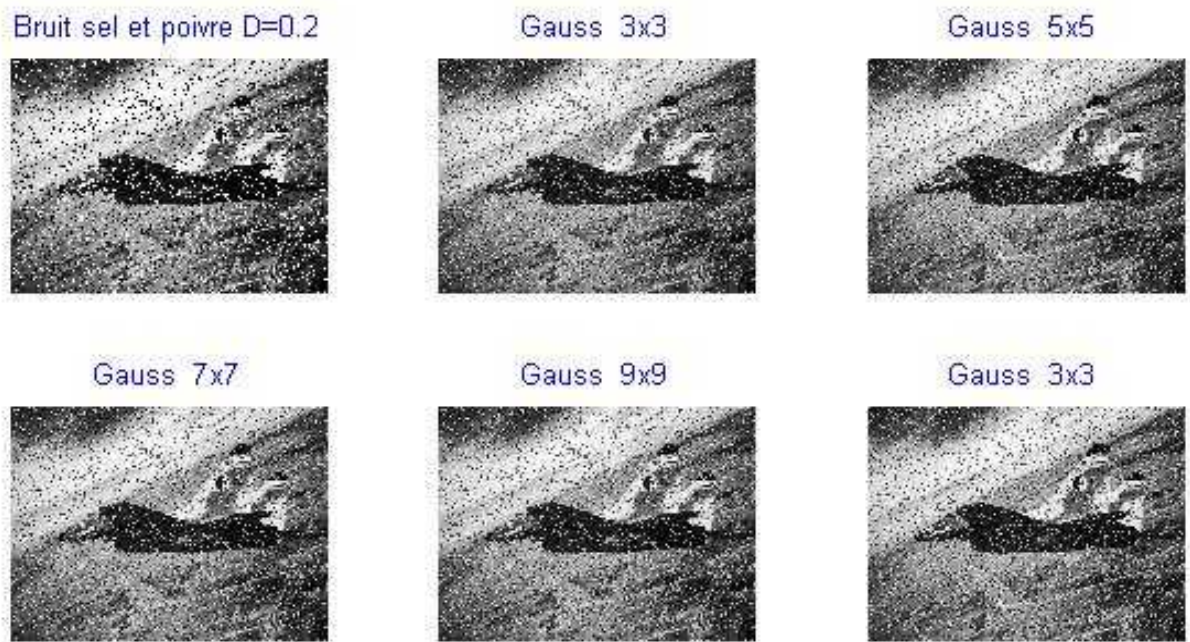
**Figure III-4 :** Influence de la taille de la fenêtre sur l'effet du filtre moyenne

#### III-5-4-2- Filtre gaussien :

On applique un filtre gaussien sous plusieurs tailles de fenêtres ((3x3), (5x5), (7x7), (9x9), (11x11)) contre le bruit sel et poivre  $D = 0.2$ .

La Figure (III-5) montre que la netteté de l'image a été améliorée au départ et puis l'aspect tend à se dégrader lorsqu'on augmente la taille de la fenêtre.

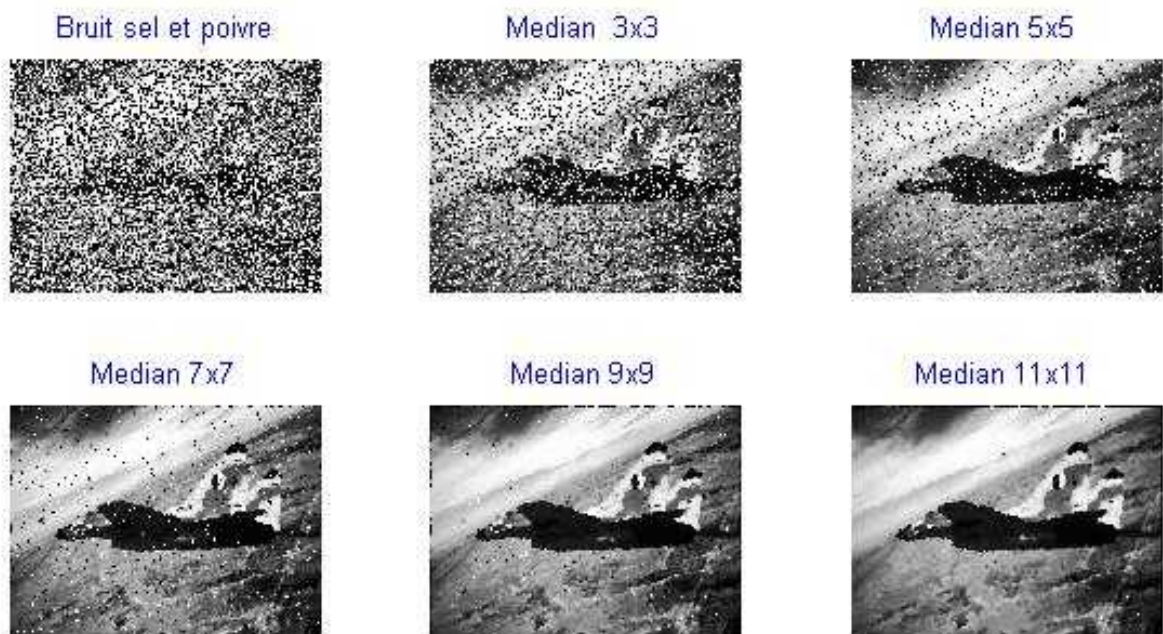




**Figure III-5:** Influence de la taille de la fenêtre sur l'effet du filtre gaussien

### III-5-4-3- Filtre médian :

On applique un filtre médian sous plusieurs tailles de fenêtres (( $3 \times 3$ ), ( $5 \times 5$ ), ( $7 \times 7$ ), ( $9 \times 9$ ), ( $11 \times 11$ )) contre le bruit sel et poivre  $D = 0.7$ .



**Figure III-6 :** Influence de la taille de la fenêtre sur l'effet du filtre médian

La Figure (III-6) montre que le lissage (3x3) a amélioré l'image mais il a laissé un certain bruit, avec une fenêtre (5x5) on arrive à enlever toutes les granulations externes mais à partir de (7x7) les formes des contours commencent à être dégradées, le (11x11) a effacé les contours de l'avion et a embrouillé les tracés de routes au sol.

### **III-5-5- Filtrage passe bas :**

#### **III-5-5-1 Famille des filtres médian :**

On teste le filtre TSM sur une image bruitée par un bruit de chatolement, uniformément réparti sur [0 ; 255]. On utilise une fenêtre 3x3.

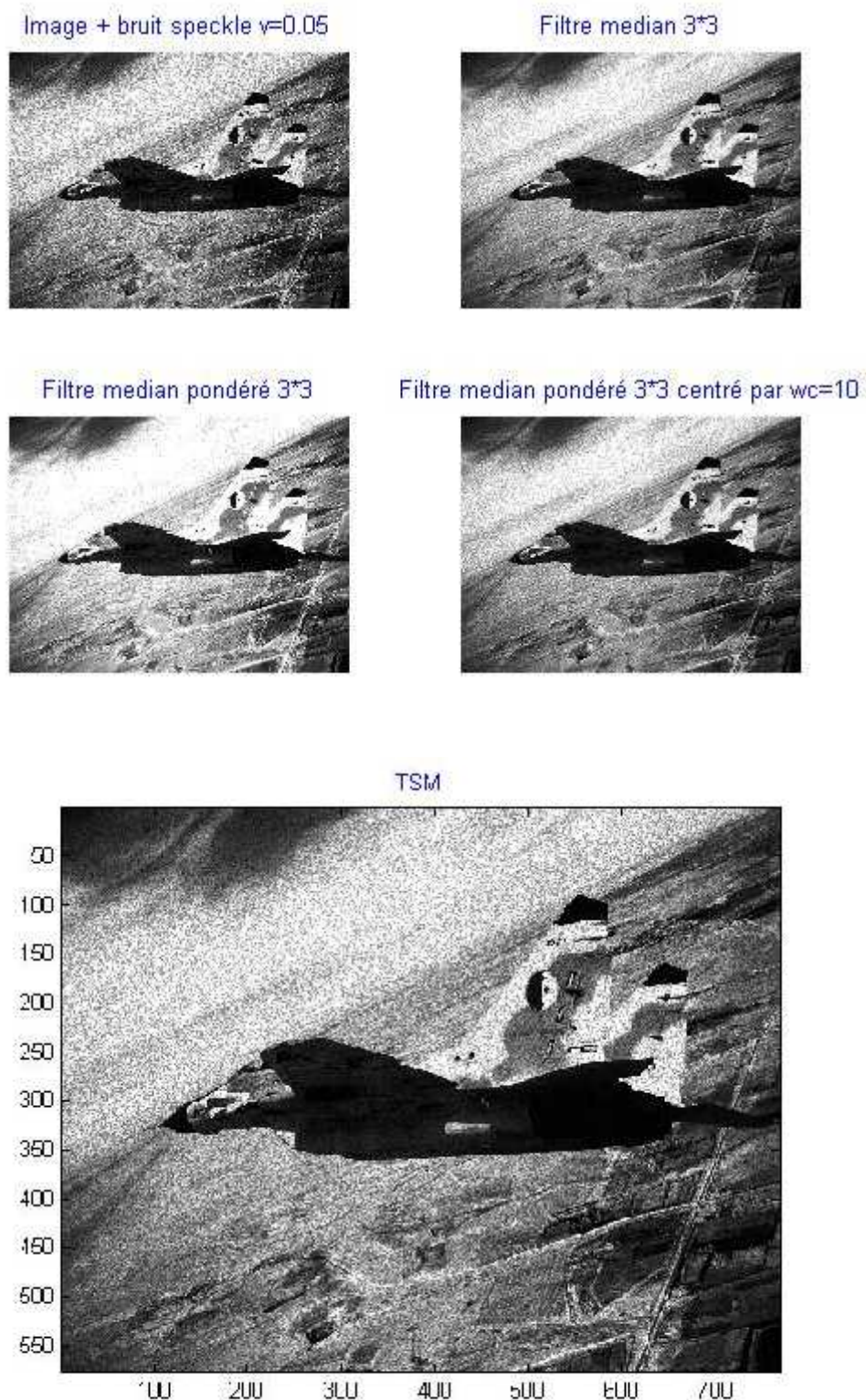
D'après la figure (III-7) on remarque que les meilleures performances de filtre médian pondéré centré sont obtenues pour WC=10.

En termes de préservation des détails, par ordre décroissant des performances on a : filtre identité, filtre CWM, filtre SM.

En termes de suppression du bruit, par ordre croissant des performances on a : filtre identité, filtre CWM, filtre SM.

Le filtre TSM permet ainsi de marier habilement les avantages des trois filtres précédents.





**Figure III-7** : Débruitage de bruit speckle par la famille des filtres médian

### III-5-5-2- Comparaison des filtres du lissage :

Cette étude sur le débruitage contribuera à donner des éléments de compréhension par la pratique quant à la comparaison des performances des filtres que nous présentons.

Les bruits envisagés seront ici des bruits :

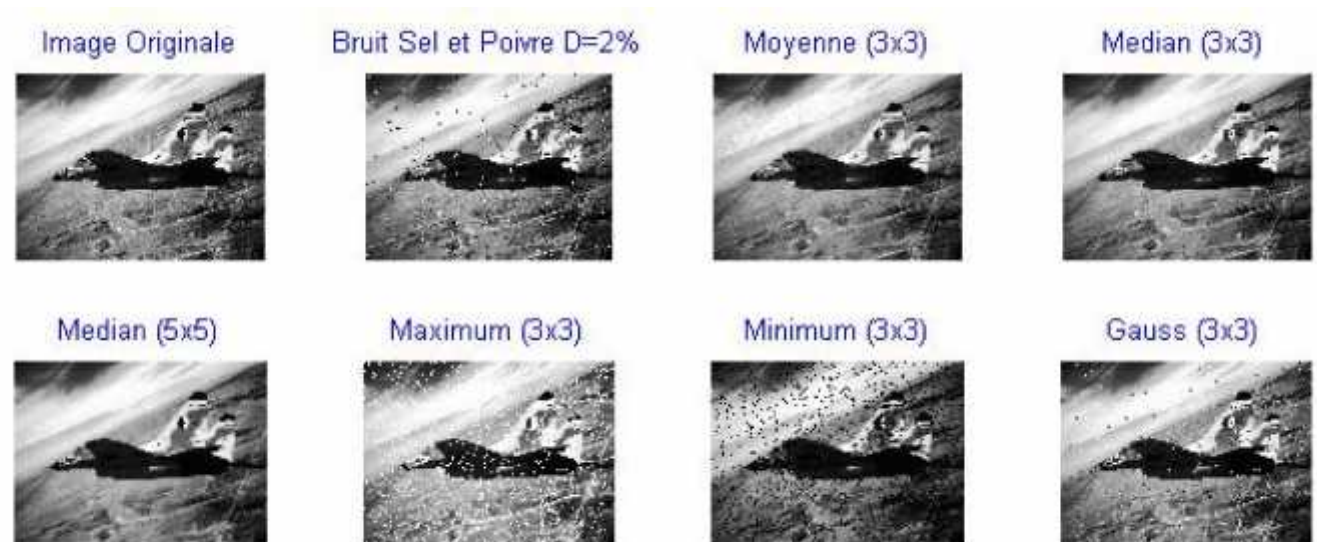
- Sel et Poivre.
- Gaussien.

Les critères de comparaisons seront successivement basés sur :

- L'appréciation subjective psycho visuelle.
- La qualité des filtrages en tant que prétraitement pour la détection des contours.
- Le rapport signal sur bruit (SNR : Signal to Noise Ratio).

#### a- Bruit Sel et Poivre (densité=2%) :

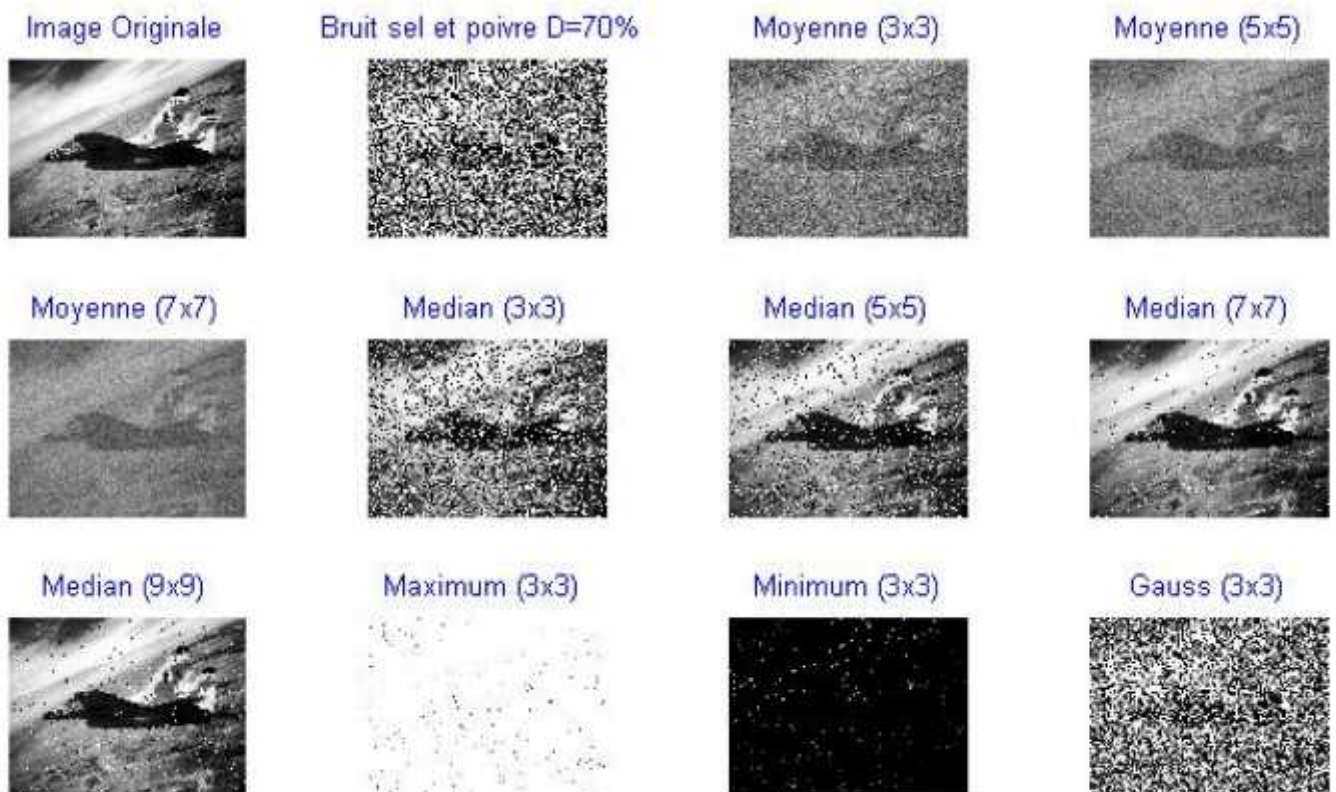
- Image bruitée.
- Débruitée par un filtre moyenne (3x3).
- Débruitée par un filtre moyenne (5x5).
- Débruitée par un filtre médian (3x3).
- Débruitée par un filtre médian (5x5).
- Débruitée par un filtre maximum (3x3).
- Débruitée par un filtre minimum (3x3).
- Débruitée par un filtre gaussien (3x3).



**Figure III-8** : Débruitage de bruit Sel et Poivre (D=2%) par les filtres du lissage

**b- Bruit Sel et Poivre (densité=70%) :**

- Image bruitée.
- Débruitée par un filtre moyenneur (3x3).
- Débruitée par un filtre moyenneur (5x5).
- Débruitée par un filtre moyenneur (7x7).
- Débruitée par un filtre médian (3x3).
- Débruitée par un filtre médian (5x5).
- Débruitée par un filtre médian (7x7).
- Débruitée par un filtre médian (9x9).
- Débruitée par un filtre maximum (3x3).
- Débruitée par un filtre minimum (3x3).
- Débruitée par un filtre Gauss (3x3).



**Figure III-9 :** Débruitage de bruit Sel et Poivre (D=70%) par les filtres du lissage

Nous allons ici comparer les filtres sur la qualité subjective (psychovisuelle) des images débruitées qu'ils fournissent, et ce en fonction de type du bruit et de son intensité.



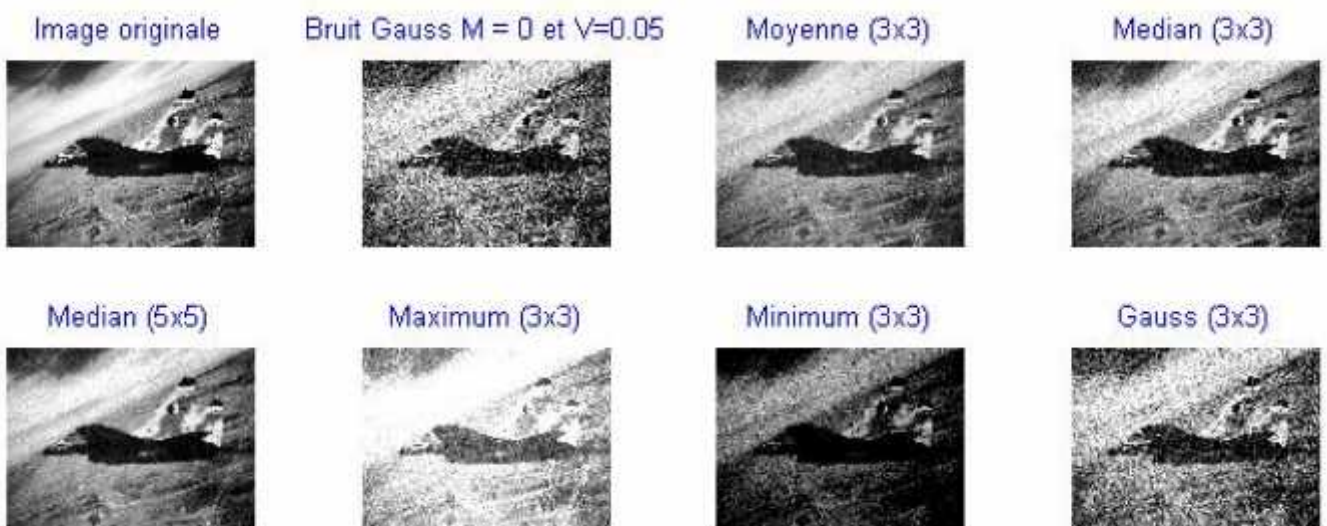
D'après la figure (III-7) on voit bien que le bruit impulsionnel (Sel et Poivre) est peu dérangeant pour l'oeil. Le débruitage est satisfaisant pour tous les filtres avec une préférence pour le médian. Les filtres créent des déformations psychovisuelles bien différentes de celles du bruit initial, le moyeneur introduit un flou dans l'image.

D'après la figure (III-8) on remarque que la gêne visuelle devient beaucoup plus importante et le débruitage beaucoup plus délicat. Le moyeneur laisse apparaître des traces blanches aux endroits de grande concentration de points corrompus, le médian introduit dans l'image un grand flou en fonction de la taille de la fenêtre mais conserve cependant d'assez bons contours.

Les filtres maximum et minimum tendent à tirer l'image vers une domination noirâtre ou blanchâtre.

### c- Bruit gaussien (variance=0.05) :

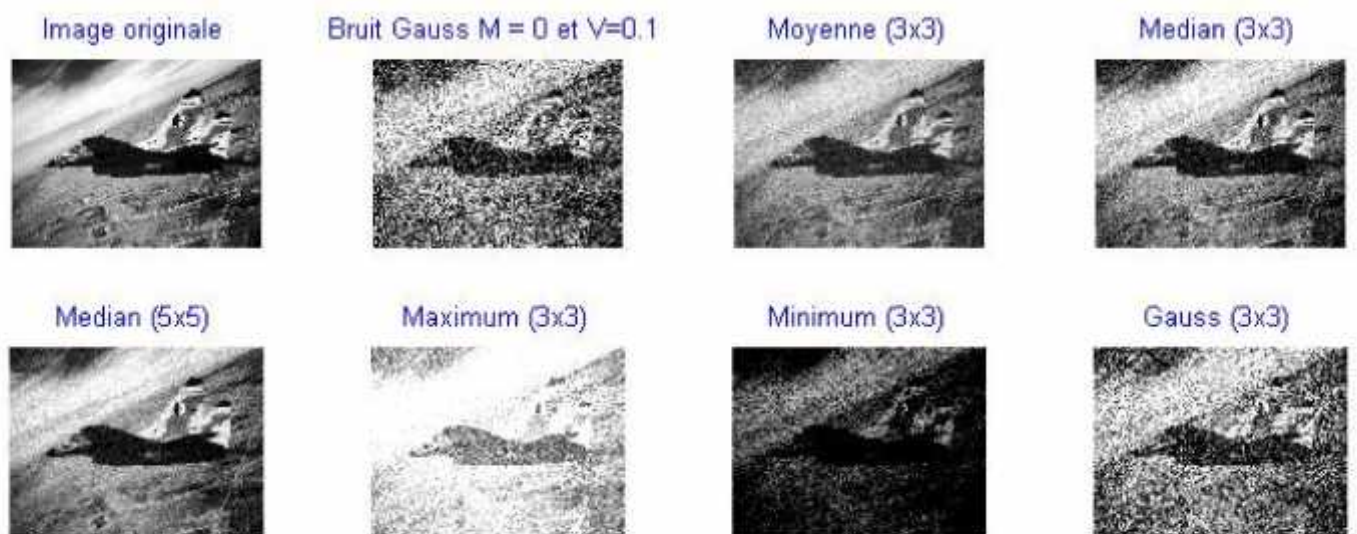
- Image bruitée.
- Débruitée par un filtre moyeneur (3x3).
- Débruitée par un filtre médian (3x3).
- Débruitée par un filtre médian (5x5).
- Débruitée par un filtre maximum (3x3).
- Débruitée par un filtre minimum (3x3).
- Débruitée par un filtre Gauss (3x3).



**Figure III-10** : Débruitage de bruit gaussien (variance=0.05) par les filtres du lissage

**d- Bruit gaussien (variance=0.1) :**

- Image bruitée.
- Débruitée par un filtre moyenneur (3x3).
- Débruitée par un filtre médian (3x3).
- Débruitée par un filtre médian (5x5).
- Débruitée par un filtre maximum (3x3).
- Débruitée par un filtre minimum (3x3).
- Débruitée par un filtre Gauss (3x3).



**Figure III-11** : Débruitage de bruit gaussien (variance=0.1) par les filtres du lissage

La figure (III-9) montre que le bruit gaussien est très peu dérangeant pour l'oeil. Il semble que les déformations apportées par les différents filtres de débruitage sont plus gênantes que le bruit initial, et de ce fait rendent le filtrage absurde d'un point de vue psychovisuel.

D'après la figure (III-10) on remarque que le bruit gaussien devient beaucoup plus gênant pour l'oeil, malheureusement l'inefficacité des filtres est à la hauteur de l'intensité du bruit.

## Conclusion :

Il apparaît tout d'abord que la qualité des filtres dépend des critères d'appréciation choisis (psychovisuel, SNR, ...). On constate ensuite que l'usage que l'on veut faire par l'image débruitée conditionne complètement le choix du filtre cas de la détection de contours.

On peut toutefois mettre en avant la robustesse du filtre médian qui donne de bons résultats dans tous les cas envisagés. On constate que c'est le seul filtre qui permet de débruiter une image corrompue simultanément par les deux bruits.

### III-5-6- Filtrage passe haut (détection de contour) :

#### III-5-6-1- Filtre Gradient :

- Image originale.
- Détection de contour par un filtre de PREWITT.
- Détection de contour par un filtre de SOBEL.
- Détection de contour par un filtre de ROBERTS.



**Figure III-12 :** Détection de contour par les filtres Gradient

### III-5-6-2- Filtre Laplacien :

- Image originale.
- Détection de contour par un filtre LAPLACIEN.



**Figure III-13** : Détection de contour par le filtre Laplacien

### III-5-6-3- Filtre de Canny :

- Image originale.
- Détection de contour par un filtre de CANNY.



**Figure III-14** : Détection de contour par le filtre de Canny



### III-5-6-4- Filtre Laplacien de la gaussienne :

- Image originale.
- Détection de contour par un filtre LAPLACIEN DE LA GAUSSIENNE.



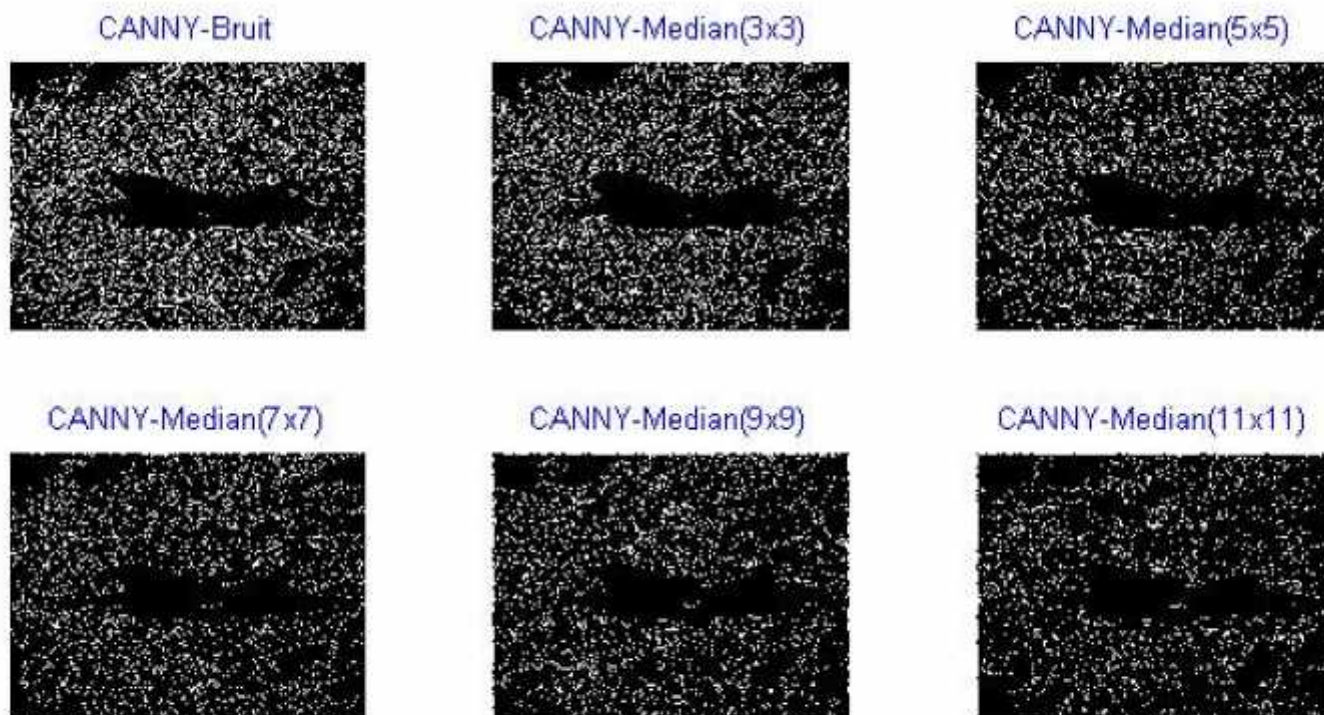
**Figure III-15 :** Détection de contour par le filtre Laplacien de la gaussienne

Les figures (III-12), (III-13), (III-14), et (III-15) montrent que tous les filtres donnent une détection des contours naturels cachés dans l'image avec une préférence et une précision pour le détecteur de Canny qui fournit de bons résultats.

### III-5-7- L'effet des filtres passe-bas sur la détection des contours :

- Image bruitée par un bruit speckle (variance=0.2).
- Débruitée par un filtre médian (3x3).
- Débruitée par un filtre médian (5x5).
- Débruitée par un filtre médian (7x7).
- Débruitée par un filtre médian (9x9).
- Débruitée par un filtre médian (11x11).
- Détection de contour par un filtre de CANNY.





**Figure III-16 :** L'effet de filtre médian sur la détection des contours

La Figure (III-16) montre que le lissage (3x3) donne une bonne détection des contours mais il a laissé un certain bruit, avec une fenêtre (5x5) on arrive à enlever toutes les granulations externes avec une meilleur détection des contours, à partir de (7x7) les formes des contours commencent à être dégradées, le (11x11) a effacé les contours de l'avion et a embrouillé les tracés de routes au sol.

### **Conclusion :**

Le détecteur de Canny fournit un meilleur compromis entre la réduction du bruit et la détection de contour.

# **Conclusion générale**

### **Conclusion générale :**

On a vu plusieurs filtres standard de réduction du bruit disponibles pour le traitement des images. Certains donnent une meilleure interprétation visuelle alors que d'autres offrent une meilleure atténuation du bruit ou un meilleur lissage.

Le filtrage dépend de l'intensité de bruit dans l'image, de la taille de la fenêtre et de la nature de texture de la scène.

Le degré de lissage est proportionnel à la taille de la fenêtre, à mesure d'augmenter la taille de la fenêtre du filtre, le lissage devient plus important, il donne pour les filtres classiques un flou intense.

L'intensité de bruit est aussi un acteur principal dans le traitement des images en vue de leurs améliorations, cette intensité est proportionnelle au filtrage, pour des images trop dégradées, cela conduit à la difficulté de l'estimation de filtrage, alors on fait plutôt appel aux techniques de restauration d'images.

Pour une meilleure interprétation et jugement sur les filtres il est recommandé de bien spécifier le domaine d'application ou la subdivision de l'image en portions selon la texture présente. Certains filtres sont applicables seulement sur les zones uniformes, avec l'inconvénient principal qui est leur ignorance du bruit dans les zones les plus proches des arêtes et des lignes en donnant un flou et leurs généralisations à des images quelconques provoquent une lourde perte d'information.

L'utilisation de chaque filtre dépend de la spécificité de l'image et de l'application recherchée. Le travail que nous avons présenté concerne le filtrage classique, il sera intéressant de faire l'inventaire exhaustif des filtres en fonction de leurs utilités et de reconduire cette étude pour donner une classification des filtres en prenant en considération simultanée l'objectif à atteindre, la texture de l'image, la forme et le modèle du bruit et la taille des fenêtres à utiliser.

---

---

# Bibliographie

- [1] [www.irit.fr/ACTIVITES/MasterPro\\_IIN/RESSOURCES/Cours/images](http://www.irit.fr/ACTIVITES/MasterPro_IIN/RESSOURCES/Cours/images).
- [2] G. KAGAN, T. BUQUET, « Numérisation d'image », Service photo de l'IRHT(CNRS), Juin 2001, <http://www.irth.cnrs.fr/formation/cours/intro.htm>.
- [3] «Le traitement des images tome 2», [www.tsi.enst.fr/maitre/textes/poly-amin-tome2.pdf](http://www.tsi.enst.fr/maitre/textes/poly-amin-tome2.pdf).
- [4] PHILIPPE MOUTESIMOS, «Une méthodologies en vue de l'interprétation automatique d'images aérienne», Thèse de Doctorat, Université de Nice.
- [5] Jong-sen Lee, «Digital image enhancement and noise filtering by use of local statistics», IEEE Trans, Pattern Analysis and Machine Intelligence, Vol. 2, No.2, pp.165-168, 1980.
- [6] P. Dewaele et al, «Comparison of some speckle reduction techniques for SAR images», IGARSS, 10:2417-2422, May 1990.
- [7] Z. Shi and K. B. Fung, «A comparison of Digital Speckle Filters», Proc. of IGARSS 94, Canada Centre for Remote Sensing, pp.2129-2133, August 8-12, 1994.
- [8] S. BRES, P-A CHAMPIN, J-M HERAUD, V. HERELIER, J-M JOLION, E. LOUPIAS, «Traitement D'images et vision artificielle», Projet TELESUN, Laboratoire Reconnaissance de formes et Vision, INSA Lyon, 1997, <http://telesun.insa-lyon.fr/>.
- [9] Y. Huang, J. L. Van Genderen and B. S. Van Veen, «The ITC filter: a new adaptive filter for SAR speckle reduction», Proceedings of the second international airborne remote sensing conference and exhibition, 1997.
- [10] Lopes A, Touzi R, Laur H, «Structure Detection and Statistical Adaptive Speckle Filtering in SAR Images», International Journal of Remote Sensing, vol. 14, no. 9, pp. 1735-1758, 1993.
- [11] M.ALI CHRIF et N Doumi «La détection de contours dans l'image à niveau de gris ».
- [12] [www.denis.elbaz.free.fr/Realisations/Filtrage D'image](http://www.denis.elbaz.free.fr/Realisations/Filtrage_D'image).