

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université Mouloud MAMMARI, Tizi-Ouzou**



**Faculté de Génie Electrique et d'Informatique**  
**Département d'électronique**

**Mémoire de Fin d'Etudes**

En vue de l'obtention du diplôme

*D'Ingénieur d'Etat en Electronique*  
*OPTION :Communication*

**Thème**

**Compression d'images météosats**  
**avec insertion spirale**

Proposé et dirigé par :

M<sup>r</sup> Mourad LAHDIR.

Présenté par : M<sup>elle</sup> Safia TAMAZIRT.

M<sup>r</sup> Ali RAMDANI.

M<sup>r</sup> Sofiane TANTAR.

Soutenu le : / 06 /2010

*Promotion 2009/2010*

## *Remerciements*

*En présentant ce travail, Nous tenons à remercier Mr Mourad LAHDIR, notre promoteur pour sa serviabilité, sa disponibilité et ses remarques constructives qui nous ont utiles tout au long de notre projet.*

*Nous tenons à remercier le président et les membres de jury pour l'honneur qu'ils nous ont fait en acceptant de juger notre travail*

*Enfin, que tous ceux qui ont, de près ou de loin, contribuer à la concrétisation de ce travail trouveront ici l'expression de notre vive reconnaissance.*

# *Dédicaces*

*Je dédie ce modeste travail à :*

- √ Mes chers parents qui m'ont bien encouragé*
- √ Mes grands parents maternels et paternels
  - √ Mes sœurs*
  - √ Ma sœur Karima*
  - √ Ma tante Saliha**
- √ Le mari de ma sœur Saïd et sa famille*
- √ Ma sœur Hasiba, son mari Moh et sa famille*
- √ Tous mes oncles et tantes et leurs familles*
- √ Tous mes amis (es) en particulier T.Sofiane et T.Safia
  - √ A la promotion 2010**

*R.Ali*

# Dédicaces

*Je remercie DIEU, le tous puissant de m'avoir accordé la santé et le courage pour accomplir ce travail.*

*Je dédié ce modeste travail :*

- ✚ En premier lieu à la mémoire de ma mère que dieu la recueillera dans son vaste paradis.*
- ✚ A mon très chère père, qui ma soutenu le long de ma vie.*
  
- ✚ A mes très chers frères (Ferhat. Akli et sa petite famille)*
- ✚ A mes très chères sœurs (Ourdia, Karima, Kahina et leurs petites familles)*
  
- ✚ A tout mes proches et amis (es).surtout R.Ali et T.Sofiane et Zahia.*

SAFIA

# *Dédicaces*

*Je dédie ce modeste travail à :*

- √ Mes chers parents qui m'ont bien encouragé*
- √ Mes grands parents maternels et paternels*
  - √ Mon frère Madjid*
  - √ Mes sœurs*
- √ Le mari de ma sœur Athmane et sa famille*
  - √ Tous mes oncles et tantes et leurs familles*
- √ Tous mes amis (es) en particulier R.Ali et T.Safia*
  - √ A la promotion 2010*

*T. Sofiane*

# *Sommaire*

# Sommaire

---

## Introduction Générale

## Chapitre I : Généralités sur le traitement d'images

I-1-Préambule.....	4
I-2-Les systèmes numériques de traitement d'images.....	4
I-3- Définition d'une image.....	5
I-3-1-Définition d'une image numérique.....	6
I-4-Caractéristique d'une image numérique.....	6
I-5-Définition de la compression d'images.....	7
I-5-1-Principe général de la compression des images.....	8
I-5-1-1- Décorrélacion.....	8
I-5-1-2-Quantification.....	8
I-5-1-3-Codage.....	10
I-6- Méthodes de compressions d'images.....	10
I-6-1- Compression sans perte (réversible).....	10
I-6-1-1 -Codage SHANNON FANO.....	10
I-6-1-2- Codage de Huffman.....	12
I-6-1-3- Arithmétique.....	13
I-6-1-4- Le codage LAMPEL ZIV WELCH (LZW).....	14
I-6-1-5-Codage en RLE (Run Length Encoding).....	14
I-6-2- Compression avec perte (irréversible).....	14
I-6-2-1 Compression par transformation.....	15
I-7- Mesure des performances .....	17
I-7-1-Taux de compression.....	18
I-7-2-Entropie.....	18
I-7-3-Mesure de distorsion.....	19
I-8- Les normes de compression d'image1 .....	20
I-8-1- La norme de compression JPEG (Joint Photographic Experts Groupe).....	20
I-8-2- La norme de compression JPEG 2000.....	21
I-8-3-La norme de compression H-261.....	22

# Sommaire

---

I-8-4-La norme de compression MPEG (Moving Picture Experts Groupe).....	22
-Discussion.....	23
<b>Chapitre II : La théorie des ondelettes</b>	
II-1-Préambule.....	25
II-2-Transformée de Fourier.....	25
II-2-1 Transformée de Fourier à fenêtre glissante (TFFG).....	26
II-3-Ondelettes.....	26
II-3-1-Définition de l'ondelette.....	27
II-3-2-Transformée en ondelette continue(CWT).....	27
II-3-2-a-Transformée inverse.....	28
II-4-Transformées en ondelette discrètes (dwt).....	28
II-5-Base d'ondelette orthogonale.....	29
II-6-Approximation de base.....	29
II-6-1-Espace d'approximation.....	29
II-6-2-Espace des détails.....	31
II-7- Analyses multi-résolutions.....	32
II-7-1-Définition.....	32
II-7-2- Interprétation des propriétés de l'analyse multi-résolution.....	33
II-7-3-base d'ondelettes de $L_2(\mathbb{R})$ .....	33
II-7-4- Algorithme de S.Mallat (Algorithme rapide).....	33
II-7-4-1-Décomposition (Analyse).....	34
II-7-4-2-Reconstruction (Synthèse).....	34
II-8-Bases biorthogonale.....	36
II-8-1-Définition.....	36
II-8-2-Analyse biorthogonale.....	36
II-9-Analyse multirésolution deux dimensions séparables.....	38
II-9-1-Principe.....	38
II-9-2-algorithmes de décomposition dyadique séparable et de reconstruction.....	39
-Discussion :.....	42

# Sommaire

---

## Chapitre III : Méthode adoptée

III-1-préambule.....	44
III-2-Principe général de la méthode.....	44
III-3-Insertion spirale des images.....	46
III-4-Compression avec JPEG2000.....	50
III-4-1-Introduction.....	50
III-4-2-Chaines de codage et de décodage JPEG2000.....	51
III-4-3-Prétraitement et transformée de couleur.....	51
III-4-4-Transformée en ondelette discrète.....	52
III-4-5-Quantification des sous bandes.....	52
III-4-6-Codage des sous-bandes.....	53
III-4-7-Codage par régions d'intérêt.....	53
III-4-8-Codage entropique.....	53
III-5-Reconstruction de l'image insérée et la séquence météoats.....	54
-Discussion :.....	59

## Chapitre IV : Tests et résultats

IV-1-Préambule.....	61
IV-2- Résultats et interprétations.....	61
IV-2-1- Résultat de la compression avec le JPEG2000.....	62
IV-2-1-1- Les résultats obtenus pour $P_c = 66.66\%$ .....	63
IV-2-1-2- Quelques figures obtenus pour $P_c = 66.66\%$ .....	67
IV-2-1-3- Les résultats obtenus pour $P_c = 33.33\%$ .....	72
IV-2-1-4- Quelques figures obtenus pour $P_c = 33.33\%$ .....	77
IV-3-Interprétation des résultats de compression avec JPEG2000.....	83
-Discussion.....	84
-Perspective .....	85
-Conclusion Générale .....	87
-Annexes	
-Références bibliographiques	

# Introduction Générale ?

De nos jours, la puissance des processeurs augmente plus vite que les capacités de stockage, et énormément plus vite que la bande passante de réseau informatique, qui malgré les nouvelles technologies, a du mal à augmenter car cela demande d'énormes changements dans les infrastructures tel que l'installation téléphonique et à cause de ces problèmes, on préfère réduire la taille des données.

Pour remédier à ces problèmes, les chercheurs ont pensé à traiter le signal d'où il vient, le terme générique « traitement du signal », car l'image est un signal bidimensionnel, on parle alors du traitement d'images. Le traitement d'images est l'ensemble des techniques opérant sur celle-ci dont le but de rendre ces images plus visibles à l'œil et aussi d'extraire des informations jugées pertinentes.

Le développement exponentiel dans le domaine de l'informatique et de traitement d'images ont rendu la quantité d'information énorme, alors pour remédier ce problème il faut penser à comprimer ces données afin de gagner l'espace pour d'autres applications au données pour les stocker car la capacité de stockage est faible. Alors pour compresser ces données où images, il faut poser cette question « à ce que on veut garder toutes les informations où pas ? » car dans le traitement des données nous avons deux méthodes de compressions celles qui sont réversibles c'est-à-dire on ne perd pas l'information lors de compression mais l'inconvénient de cette méthode est que nous n'arrivons pas à des taux de compressions élevés par contre la dans la deuxième méthode on perd l'information mais nous arrivons à des taux de compressions plus élevés, cette méthode est irréversible.

Le traitement d'images a progressé d'une manière rapide depuis sa naissance, à l'heure actuelle, les recherches sont arrivées à donner deux standards récents ; le JPEG (Joint Photographic Experts Group), est le standard qui utilise la décomposition en cosinus discrète TCD (Transformation Cosinus Discrète), et l'autre norme qui est le JPEG2000, cette forme est la plus récente et utilise la décomposition en ondelettes TOD (Transformation Ondelettes Discrètes), c'est cette dernière norme qu'on a utilisée dans notre travail.

Ainsi le développement dans le domaine de la météorologie (stockage des images, transmission des images.....) a poussé les chercheurs à étudier ces images. En général, la série des satellites géostationnaires Météosats, permet d'observer tout le disque terrestre dans le quelle se trouve les continents Européen et africain. Les images produites par ces satellites,

sont collectées toutes les demi-heures dans trois canaux par les satellites anciens et pour le satellite de seconde génération MSG (Meteosat Second Generation), les informations sont collectées toutes les quart-heures dans douze canaux. Avec toutes ces informations collectées en demi-heures pour les anciens satellites et les quart-heures pour le MSG, ont rendu les études sur la méthode de stockage et de transmission plus intéressante car l'essentiel pour les scientifiques est de trouver un compromis entre espace mémoire et temps.

Pour gagner de l'espace et le temps il faut compresser une séquence d'images au lieu de compresser image une par image c'est comme une modulation d'un signal. Notre travail est basé sur l'insertion spirale des images dans d'autres images.

Pour cela nous avons reparti notre travail en quatre chapitres :

Dans le premier chapitre, nous avons parlé d'une façon générale sur les méthodes de compression d'images les plus connues telle que le codage de Shannon-Fano, arithmétique et LZW...etc. et on le termine par une conclusion.

Par contre dans le deuxième chapitre nous avons consacré une étude un peu détaillée sur les ondelettes orthogonales et biorthogonales et aussi sur l'analyse multirésolution, ainsi une description théorique de la méthode de décomposition et de reconstruction. On termine ce chapitre par une conclusion.

Dans le troisième chapitre nous avons élaboré notre méthode adoptée et nous avons présenté les différents blocs de notre schéma de compression ainsi la méthode d'insertion, de reconstruction et les différents traitements mis en œuvre.

Dans le quatrième chapitre nous avons consacré notre étude sur les tests et les résultats avec quelques images.

Enfin nous terminons notre travail avec une conclusion générale.

# **Chapitre I**

## **Généralités sur le traitement d'images**

**I-1-Préambule :**

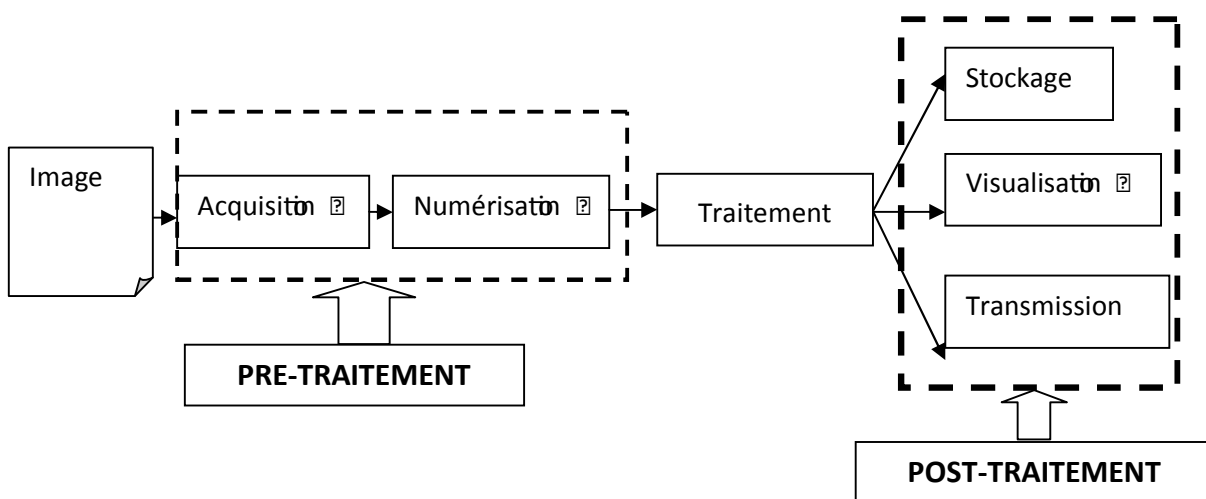
L'image est considérée parmi les moyens les plus privilégiés que l'homme a utilisé pour communiquer avec autrui. En effet c'est un moyen de communication universel dont la richesse de contenu a permis aux êtres humains chacun de sa manière d'analyser et de comprendre l'image.

Mais à l'heure actuelle l'imagerie intervient dans plusieurs domaines tels que le contrôle des voyageurs ainsi que leurs bagages, la médecine est aussi dans la météorologie; à cause des divers applications les images ont besoin des traitements, d'où il vient le terme générique « traitement d'image » qui veut dire ensemble des technique et méthodes opérant sur l'image afin d'améliorer son aspect visuel et d'extraire des informations jugées pertinentes.

Une autre technique qui est indispensable dans le traitement d'images c'est « la compression d'images », son rôle est de minimiser la quantité d'informations qui transporte une image mais tout en gardant sa qualité, donc éliminer les informations qui ne sont pas importantes et cela veut dire minimiser la redondance dans l'image originale pour ne pas charger la mémoire avec des informations inutiles, afin de gagner l'espace pour l'utilisation d'autres applications ou d'autres images.

**I-2-Les systèmes numériques de traitement d'images :**

Un système numérique de traitement d'image comprend un ensemble de technologies et de services, les fonctions élémentaires sont par exemple l'acquisition d'images ou de données, la numérisation, la compression, le stockage, la manipulation de données, l'affichage et la transmission ou encore la manipulation à distance d'instruments (figure I.1).



**Figure I-1** Schéma générale d'un système numérique de traitement d'images

**1-Pré- traitement :**

L'acquisition des images est la première étape dans une chaîne de traitement numérique des images, pour pouvoir manipuler une image sur un système informatique, il est donc nécessaire de subir à l'image une transformation qui la rend visible et manipulable par ce système, cette étape est appelée numérisation. Ces systèmes de saisis, dénommés optiques, peuvent être classés en deux catégories principales : caméras numériques et les scanners.

Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionne à l'instar des cameras vidéo, grâce aux capteurs CCD (Charge Coupled Device), les cartes d'acquisition reçoivent les images d'un capteur et les convertit en information binaire qui seront stockées dans un fichier.

**2-Traitement des images :**

Dans cette étape de chaîne de traitement, nous allons subir aux images plusieurs opérations : filtrage numérique, détection de contours, calcul de gradient, la transformée de Fourier ou la décomposition en Ondelettes.

**3- Post-traitement :**

Le post-traitement concerne les images traitées, c'est la dernière étape dans un système numérique de traitement d'images. Les images traitées seront transmises, stockées ou visualisées.

**I-3- Définition d'une image :**

On peut dire que l'image est obtenue par transformation d'une scène réelle par un capteur qui transforme l'image captée en un signal électrique après numérisation de signal. Dans une scène à imager, les objets émettent les ondes électromagnétiques qui sont des sources ou réfractent une onde électromagnétique.

D'une manière générale ; on définit une image comme étant un support d'information, elle présente une scène qui a été captée soit par un appareil photographique, soit par un satellite,....

Une image peut avoir d'autres définitions, en traitement du signal, on définit une image comme étant un signal bidimensionnel. Mathématiquement parlant, une image est une

application d'un sous-ensemble  $M \times N$  de  $R \times R$  vers l'ensemble des réels  $R$ , qui à chaque couple de réels  $(x, y)$  est associé  $F(x, y)$  :

$$\begin{cases} F(M, N) \longrightarrow R \\ (X, Y) \longrightarrow F(X, Y) \end{cases} \quad (\text{I-1})$$

### I-3-1-Définition d'une image numérique :

L'image numérique est la conversion de celle-ci de son état analogique (distribution continue d'intensité lumineuse dans un plan  $xoy$ ) vers une représentation matricielle bidimensionnelle de valeur numérique  $f(x, y)$ , chaque valeur de  $f(x, y)$  représente un pixel de l'image et sa valeur associe un niveau de gris sur  $m$  bits soit  $2^m$  niveau ( $0 = \text{noir}$ ,  $2^{m-1} = \text{blanc}$ ).

On peut dire aussi qu'une image numérique encore appelée image BITMAP, est une image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleur prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de scène à représenter.

### I-4-Caractéristique d'une image numérique :

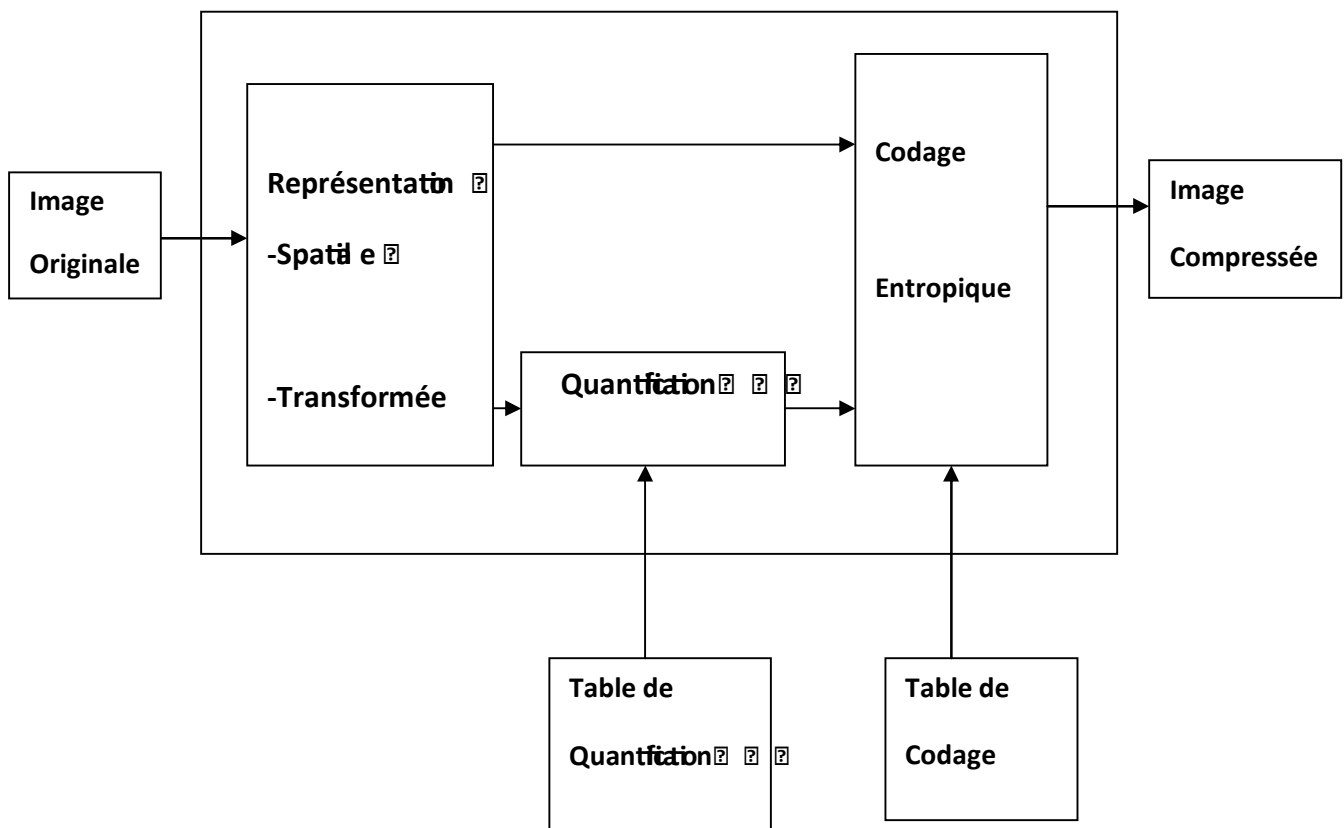
L'image est caractérisée premièrement par le PIXEL qui est le plus petit point carrée issu de terme anglais "Picture element". Ce PIXEL véhicule une quantité d'information et chaque pixel donne la nuance entre les images monochromes et images couleurs. Dans le cas d'une image monochrome un pixel est codé sur un octet, et dans le cas d'une image couleur (R.V.B) un pixel est codé sur trois octets, chaque octet pour une couleur; et dans le cas de trois dimension on parle d'un VOXEL. Ainsi une image est caractérisée par la résolution qui est la finesse ou la clarté des détails atteints par un moniteur ou une imprimante dans la production d'une image. L'autre caractéristique est la dimension qui est représentée sous forme d'une matrice dont les éléments sont des valeurs numériques représentant des intensités lumineuses ainsi que le nombre de lignes multiplié par le nombre de colonnes donne le nombre total des pixels dans une image. Pour la luminance, c'est le degré de luminosité des points de l'image, elle est définie comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, ainsi nous avons le bruit qui est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins. Nous avons aussi le contraste qui est défini comme étant l'opposition entre deux régions de l'image, l'une est

considérée claire et l'autre sombre. Ainsi nous avons l'histogramme qui est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans une image.

**I-5-Définition de la compression d'images :**

La compression d'image consiste à faire réduire la taille physique d'un bloc d'information, autrement dit réduire le nombre de bits par pixels et tout cela a pour but de la réduction du cout du stockage et la transmission rapide des données, en exploitant la redondance informationnelle dans l'image.

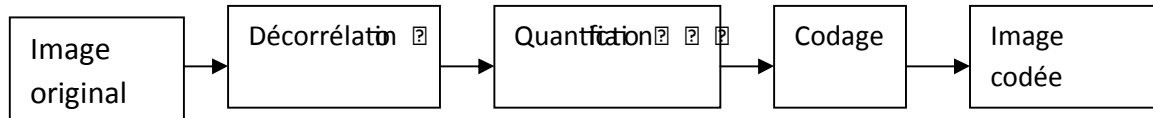
**Encodage**



**Figure I-2** Structure générale d'un schéma de compression d'image

**I-5-1-Principe générale de la compression des images :**

Le schéma fonctionnel de la compression est représenté dans la figure I-3 ci-dessous :



**Figure I-3-** schéma d'un codeur source

**I-5-1-1- Décorrélation :**

La dépendance existant entre chacun des pixels et ses voisins (la luminosité varie très peu d'un pixel à un pixel voisin) traduit une corrélation très forte sur l'image. On essaie alors de tirer une partie de cette corrélation, pour réduire le volume d'information en effectuant une décorrélation des pixels, cette décorrélation consiste à transformer les pixels initiaux en un ensemble de coefficients moins corrélés, c'est une opération réversible.

**I-5-1-2-Quantification :**

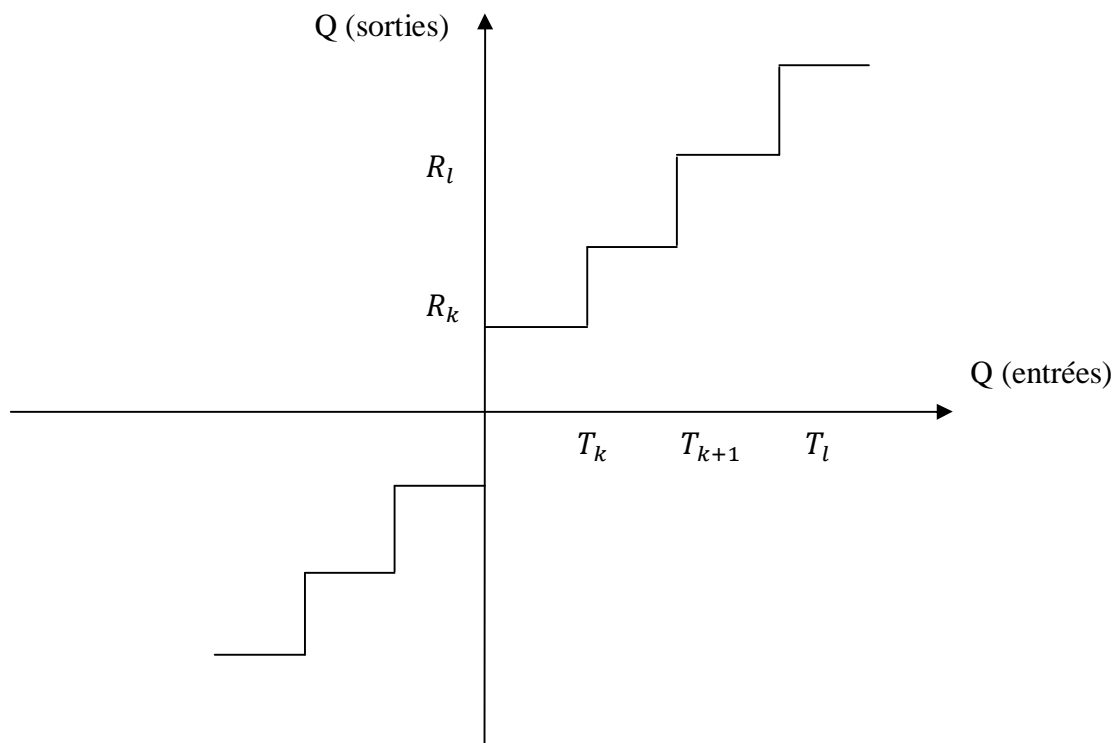
La quantification est la partie qui concerne la perte d'information de l'image et donc de la qualité de celle-ci, son objectif est d'atténuer les fréquences élevées pour lesquelles l'œil est peu sensible.

C'est à dire La quantification des coefficients obtenus a pour but de réduire le nombre de bits nécessaires pour leur représentation, elle représente une étape clé de la compression, elle approxime chaque valeur d'un signal par un multiple entier d'une quantité  $Q$ , appelée quantum élémentaire ou pas de quantification, si elle peut être scalaire ou vectorielle. L'un des résultats fondamentaux de Shannon concernant la relation débit /distorsion, montre que l'on obtient des meilleures performances en utilisant la quantification vectorielle.

**Ø Quantification scalaire :**

La quantification scalaire (QS) consiste à associer à une variable continue  $Q$  une variable discrète  $Q'$  pouvant prendre un nombre plus faible et fini de valeurs.

Si on représente par un diagramme la relation entre  $Q$  et  $Q'$  ( $Q$  : l'entrée et  $Q'$  : la sortie), on obtient une fonction en escalier, comme représenté à la figure I-4 ci-dessous :



**Figure I-4-** Quantification scalaire uniforme

On peut choisir les régions de décision (l'entrée)  $\{T_K, K = 1, 2, \dots, L\}$ , ainsi que les seuils de la décision (la sortie)  $\{R_J, J = 1, 2, \dots, L\}$ .

L'étape de quantification introduite inévitablement une distorsion.

#### Ø Quantification vectorielle :

La quantification vectorielle est définie comme étant une méthode permettant la représentation d'une image avec un minimum de bits possibles. L'algorithme de la **QV** consiste à diviser l'image en un ensemble de blocs, qui vont être comparés aux vecteurs qui forment le dictionnaire. Cette quantification consiste à repérer dans ce dernier des mots - codes disponibles, qui ressemblent le plus au bloc d'entrée. Le critère consiste en général à minimiser l'erreur quadratique et le décodage consiste simplement à remplacer l'indice par le mot - code correspondant dans le dictionnaire.

Le dictionnaire de bloc-types est construit par apprentissage, à l'aide d'un ensemble d'image type à compresser, ou à l'aide d'un treillis qui utilise les propriétés de la distribution statistique des symboles à coder un dictionnaire qui n'est performant que par la classe d'image avec laquelle il a été mis au point.

**I-5-1-3- Codage :**

Une fois les coefficients quantifiés, ils sont codés. un codeur doit satisfaire à priori les conditions suivantes :

- Unicité : deux messages différents ne doivent pas être codé de la même façon.
- Déchiffrabilité : deux mots de codes successifs doivent être distingués sans ambiguïté.

**I-6- Méthodes de compressions d'images :**

Il existe deux grands domaines de compression:

- compression sans perte (réversible)
- compression avec perte (irréversible)

**I-6-1- Compression sans perte (réversible):**

La compression est dite sans perte ou réversible lorsqu'il n'y a aucune perte de données sur l'information d'origine. Le but de la compression est de réduire la taille moyenne de texte obtenu après la compression afin d'accéder au message d'origine. Les méthodes réversibles permettent des taux de compression de l'ordre de 50 à 75%. Afin de retrouver exactement les pixels de l'image numérique originale et les diverses techniques utilisant soit un modèle prédictive soit un modèle multi-résolution. Nous avons comme méthodes de codage statique (algorithme de Shannon Fano et de Huffman) codage en RLU, en terminant par la compression (LZW).

**I-6-1-1 -Codage SHANNON FANO :**

Le codage de Shannon Fano est un algorithme de compression de données sans perte dont la probabilité de chaque symbole doit être connue, il s'agit d'un codage entropique.

**Principe :**

Les codes sont définis à l'aide d'un algorithme spécifique selon un arbre de Shannon-Fano :

1-Classer les probabilités de chaque symbole dans une liste par ordre décroissant.

2-Repas la table des probabilités en deux parties, le total des probabilités de la partie supérieure doit être le plus proche possible du total de la moitié inférieure en poursuivant l'arborescence jusqu'à ce que toutes les fréquences soient isolées.

3-Attribuer dans l'arborescence le bit 0 à chaque moitié supérieure et le bit 1 aux moitiés inférieures.

4- Affecter à chaque symbole le code binaire généré par l'arborescence

**Exemple de codage de SHANNON FANO:**

Exemple : soit une image qui contient les codes suivants

a=111,b=100,c=144,d=250,e=180,f=155 et g=200

111	155	155	144	200	144	144	200
180	155	111	200	200	144	111	250
180	155	111	144	200	144	155	250
250	155	155	155	250	250	155	200
250	100	155	250	180	100	200	250
250	155	155	155	180	100	111	250
250	100	155	100	250	200	100	200
180	180	180	180	180	144	144	200

Symboles	a	b	c	d	e	f	g
Fréquences d'apparitions	5	6	8	12	9	14	10

**Figure I-5-a-** Image codée

**Figure I-5-b-**Fréquence d'apparition des codes

Symboles	Probabilités	Codes
<b>f</b>	<b>0.22</b>	<b>01</b>
<b>d</b>	<b>0.187</b>	<b>000</b>
<b>g</b>	<b>0.156</b>	<b>001</b>
<b>e</b>	<b>0.14</b>	<b>100</b>
<b>c</b>	<b>0.125</b>	<b>101</b>
<b>b</b>	<b>0.093</b>	<b>1100</b>
<b>a</b>	<b>0.078</b>	<b>1101</b>

**Figure I-5-c-** Construction de codage de Shannon Fano

**Figure I-5-Exemple de codage de SHANNON FANO**

**I-6-1-2- Codage de Huffman :**

Le codage de Huffman a rapidement remplacé le codage de Shannon Fano car il est plus efficace.

La méthode générale de construction de code de longueur variable a été découverte par Huffman dans le but est la compression des données : au lieu des sept ou huit bits pour représenter un caractère, on utilise un nombre moindre de bits pour les caractères les plus fréquents et un nombre plus grand pour ceux des moins fréquents.

L'algorithme considère chaque symbole à coder comme étant une feuille d'un arbre qui reste à construire, l'idée est d'attribuer aux deux symboles de plus faibles probabilités, les mots codés les plus longs, ces deux mots codés ne se différencient que par leur dernier bit.

**Principe :**

1-Distribuer les fréquences  $f(i)$  des symboles.

2-Classer les symboles dans l'ordre décroissant des fréquences.

Le résultat de l'algorithme ne se change pas si l'on remplace les fréquences  $F(i)$  par les probabilités.

3-regrouper par séquences les paires des symboles de plus faible probabilités en les relaçant si nécessaire, plus précisément par la somme des deux faibles fréquences :

$$S = F(i_n) + F(i_{n-1})$$

4-Choisir le plus petit indice  $K$  tel  $S$  sera supérieur ou égal à  $F(i,k)$ , remplacer  $K$  "" position de la valeur  $S$  et en décalant les autres d'une position vers le bas. Puis décrémenter  $n$  d'une unité, poursuivre jusqu'à ce que la table des fréquences ne comportent plus de deux éléments.

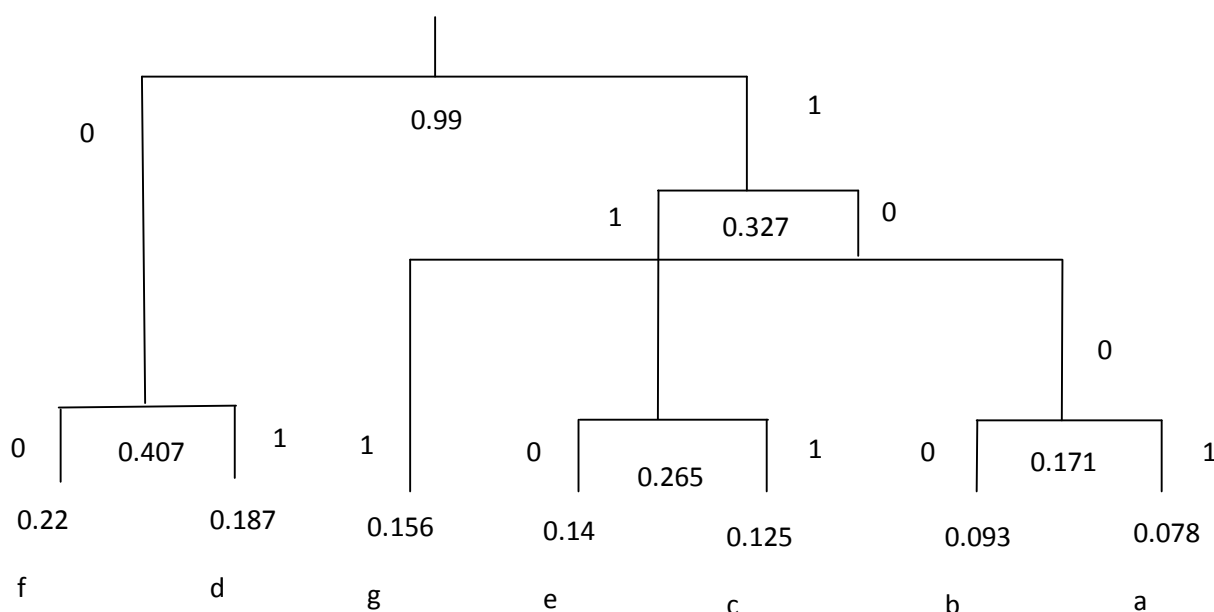
5-Coder avec retour arrière depuis le dernier groupe, en ajoutant un (0) ou un (1) pour différencier les symboles préalablement regroupés.

**Exemple de code de HUFFMAN :**

Une source S émet les symboles de a jusqu'a g avec les probabilités suivantes :

Symboles	a	b	c	d	e	f	g
Fréquences d'apparitions	5	6	8	12	9	14	10

**Figure I-6-a-** Fréquence d'apparition des codes de l'image



**Figure I-6-b-** Arbre HUFFMAN après construction.

**Figure I-6- Exemple de codage de HUFFMAN.**

**I-6-1-3- Arithmétique:**

Le codage arithmétique est un codage récent utilisant un modèle statistique, tout comme le codeur de Huffman. Contrairement à ce dernier, il produit un code pour une séquence de symboles tous entiers, et non pas un code par symbole.

Chaque nouveau symbole lu modifie de façon incrémentale le code de sortie. Ce code de sortie est un nombre à virgule flottante compris entre 0 et 1, dont le nombre de chiffres après la virgule correspond au nombre de symboles. Contrairement à Huffman, il n'est pas obligatoire que chaque code ait un nombre entier de bits. Par exemple un symbole de

probabilité 0.9 a pour entropie 0.15, mais Huffman affectera probablement un code de un bit (ou plus), et la séquence codée aura un nombre de bits plus long qu'en théorie.

Le codeur arithmétique est plus performant que le codeur de Huffman, mais il est plus complexe à implémenter.

#### **I-6-1-4- Le codage LAMPEL ZIV WELCH (LZW):**

LZW est un code imaginé par Lempel et Ziv en 1977 qui a donné de nombreuses variantes (LZ77, LZ78, ...), après l'amélioration de ces algorithmes ils ont arrivés à LZW, l'un des algorithmes les plus répandus. LZW peut travailler avec n'importe quel type de données. Il est rapide en compression et en décompression. Il encode au niveau du bit et non au niveau de l'octet. On remplace avec cette méthode quelque bit par un mot, une phrase ou même un paragraphe. Il est principalement utilisé dans les modems, le format GIF, la compression ZIP, ARG.

LZW utilise la substitution des motifs en basant sur un dictionnaire construit au fil de la compression. Le flot d'information à compresser est découpé en chaînes d'octet. Chaque chaîne est comparée au dictionnaire. Si elle n'est pas présente, elle est stockée et elle est ensuite écrite dans le flot de compressé.

Quand une chaîne, déjà rencontrée, apparaît dans le flot, elle est codée et transmise si elle a une longueur inférieure au plus grand mot du dictionnaire. Le dictionnaire est initialisé avec 256 valeurs de la table ASCII.

#### **I-6-1-5-Codage en RLE (Run Length Encoding) :**

La méthode de compression **RLE** est utilisée par de nombreux formats d'images, elle est basée sur la répétition d'éléments consécutifs, lorsque trois éléments ou plus se répètent consécutivement alors la méthode de compression RLE est utilisée. Par exemple la séquence "4444444" est remplacée par "4/7". Ce codage appelé RLE "Run Length Encoding" est intéressant pour des données comportant peu de valeurs différentes et de longues séquences.

#### **I-6-2- Compression avec perte (irréversible) :**

La compression dite avec perte ne s'applique qu'aux données 'perceptibles' en général, sonores ou visuelles. La perte d'information est irréversible, il est impossible de

retrouver les données d'origine après une telle compression donc l'image reconstruite étant dégradée par rapport à l'original.

Ces méthodes permettent de retrouver une approximation de l'image numérique et les pertes sont généralement indécélables à l'œil nu.

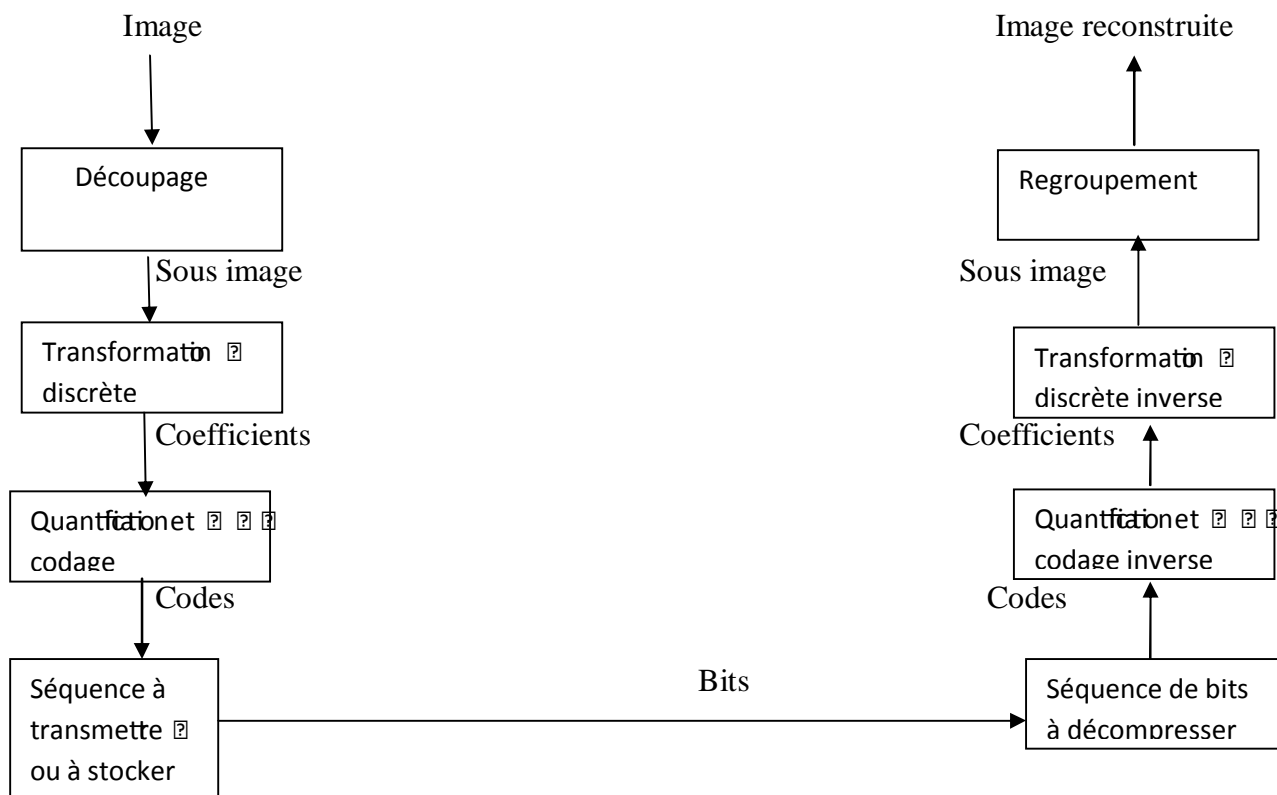
**I-6-2-1 Compression par transformation**

Dans les méthodes par transformation l'image est subdivisée en sous images (bloc) de taille réduite. Chaque bloc subit une transformation mathématique orthogonale réversible linéaire du domaine spatial vers le domaine fréquentiel.

Parmi les transformations linéaires existantes :

- Transformation en cosinus discrète TCD.
- La transformation KARHUNEN LOEVE KLT.
- La transformation en ondelettes discrètes TOD.

Le principe d'un système de codage par transformation et le suivant :



**Figure -I-7- Principe d'un système de codage par transformation**

- **Transformation en cosinus discrète DCT:**

Le passage par la DCT a été l'idée majeure pour la compression JPEG, en effet ce processus permet un changement de domaine d'étude, tout en gardant exactement la même fonction étudiée. On peut comparer la DCT à la transformée de fourrier. L'application de la DCT, fait passer l'information de l'image du domaine spatial en une représentation identique dans le domaine fréquentiel et on passe d'un domaine à un autre, parce qu'une image classique admet une grande continuité entre les valeurs des pixels. Les hautes fréquences étant réservées à des changements rapides d'intensité du pixel.

Formule pour calculer la DCT sur une matrice NxN

$$X_c(k, l) = \frac{4}{N^2} C(k, l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} X(m, n) \cos\left[\pi \frac{(2m+1)k}{2N}\right] \cos\left[\pi \frac{(2n+1)l}{2N}\right] \quad (I-2)$$

Avec : k=1, 2, 3,4.....N-1

C(0,0)=1/2

C(0,1)=C(k, 0)=1/√2

C(k, l)=1

- **La transformation KARHUNEN LOEVE KLT :**

L'idée de la transformation est de réaliser un changement de base pour obtenir de nouveaux axes où l'information contenue sur chaque axe est distribuée de façon optimale.

La transformation KLT est une transformation orthogonale. Elle est optimale en terme de comptage d'énergie .elle est définie par :

$$X_i = XU' \quad (I-3)$$

Ou Xi : représente les coefficients transformés

X : pixel de l'image originale

U : matrice de transformation transposé

En pratique cette transformation est rarement utilisée car on ne dispose pas d'algorithme de calcul rapide pour l'évaluer.

- **La transformation en ondelettes discrète TOD :**

La transformation en ondelette permet d'obtenir une représentation temps-fréquence. Elle utilise la décomposition spatiale en un ensemble de coefficient indépendant, elle conserve l'énergie du signal et possède aussi des algorithmes rapides.

La décomposition en ondelette aboutit à trois images de détails, et une image de basse résolution, cette dernière est à son tour décomposée en une image de détails et une image de basse résolution, et ainsi de suite, plus on avance dans la décomposition plus les fréquences sont basses.

- **La compression fractale :**

Toutes les images fractales sont des images caractérisées par leur auto similitude .Elle sont générées a partir d'une équation et de quelques paramètres .Dans cette compression pour chaque image il existe une formule de calcul qui prend du temps pour la compression, par contre la décompression est rapide donc elle est asymétrique.

Chaque image possède une formule qui permet de la construire mais nous n'avons aucune méthode pour déterminer cette formule .C'est seulement en 1986 un chercheur a proposé une méthode qui permet d'approcher cette formule d'une manière utile à la compression. La compression fractale considère l'image composée de plusieurs petites images. Cette technique cherche à trouver les relations reliant les petites images composantes de l'image initiale puis les représenter par des formules mathématiques .En pratique, pour des taux inférieurs à 50, la compression fractale donne des résultats plutôt inférieurs au JPEG. Pour des taux plus importants, c'est l'inverse.

### **I-7- Mesure des performances :**

Les principaux critères d'évaluation de toute méthode de compression sont :

- La qualité de reconstitution de l'image
- L'entropie
- La mesure de distorsion
- Le taux de compression
- La rapidité du codeur et du décodeur (codec)

**I-7-1-Taux de compression :**

Il sert à mesurer l'efficacité d'une méthode de compression d'images. Tel que il est utilisée dans les publications issues de la communauté de traitement d'image, en comparant sous la forme d'un rapport, la quantité de bits utilisés par l'image originale et celle utilisée par l'image comprimée.

$$RC = \frac{\text{Nombre de bits de l'image originale}}{\text{Nombre de bits de l'image comprimée}} \quad (I-4)$$

**RC** est le rapport de compression

$$TC = (1 - \frac{1}{RC}) * 100 \quad (I-5)$$

Et le rapport **TC** appelée taux de compression

- le taux est toujours supérieur ou égale à 1
- quand le taux est grand l'image est mieux compressée
- l'entropie donc une limite théorique au **TC** maximal qu'il est possible d'obtenir sans distorsion pour toute méthode de compression sur une image donnée.

**I-7-2-Entropie :**

L'entropie est une mesure de quantité d'information contenue dans une image. De ce fait, une image dont tous les pixels ont la même valeur, contient très peu d'information car elle est extrêmement redondante, son entropie est faible, par contre une image dont tous les pixels ont une valeur aléatoire, contient beaucoup d'information, son entropie est élevé, donc l'entropie représente la quantité d'information minimale nécessaire à une bonne représentation de l'image.

En pratique, le calcul de l'entropie  $H$  d'une image numérique nécessite celui des probabilités  $P(n)$  des niveaux de gris, est sont inversement proportionnelles.

$$H = - \sum_{n=0}^{2^R-1} P(n) \log_2 [P(n)] \quad (\text{bpp}) \quad (I-6)$$

R : le nombre de bit par pixel avec laquelle est codée l'image.

Bpp : Bit par pixel.

- Si L est la longueur moyenne d'un code, il faut que  $L \geq H$  pour présenter toute l'information nécessaire.
- L'entropie est maximale lorsque on a le maximum d'incertitude quant à l'apparition des symboles c'est à dire Si les probabilités des différents sont égaux.

### I-7-3-Mesure de distorsion :

La mesure de la distorsion utilisée généralement dans la compression d'images est l'erreur quadratique moyenne MSE (Mean Squar Error) telle que de plus MSE est forte, plus il y a de différences entre l'image originale et l'image compressée. Cette grandeur représente la moyenne des écarts aux carrés entre le pixel de l'image originale  $I(i, j)$ , et celui de l'image reconstruite  $I'(i, j)$ .

$$MSE = \frac{1}{M*N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i, j) - I'(i, j)]^2 \quad (I-7)$$

M : le nombre de lignes de la matrice image.

N : le nombre de colonnes de la matrice image.

**PSNR** : Le rapport signal sur bruit crête pour une image (**PSNR**) est utilisé pour mesurer les performances des méthodes de compression d'image. Il traduit une mesure d'erreur issue de l'erreur quadratique moyenne entre les deux images (MSE). La mesure du PSNR dont l'unité est le décibel est donné par l'équation suivante :

$$PSNR = 10 \log_{10} \frac{[Ng_{max}]^2}{MSE} \quad (dB) \quad (I-8)$$

$Ng_{max}$  : Dynamique de l'image.

Lorsque la reconstitution est parfaite :

- MSE  $\longrightarrow$  0
- PSNR  $\longrightarrow$   $\infty$

Ces mesures sont très utilisées car elles sont très simples à calculer, cependant elles ne rendent pas compte de la qualité visuelle engendrée par la compression.

**I-8- Les normes de compression d'image :****I-8-1- La norme de compression JPEG (Joint Photographic Experts Groupe):**

JPEG désigne avant tout une méthode de compression conçue pour échanger des images avec une qualité acceptable tout en limitant la taille des fichiers à un strict minimum. Elle est destinée à la compression des images fixes en couleurs et à niveau de gris ou de couleurs, elle est moins adaptée aux images en noir et blanc ou aux images à palettes de couleurs.

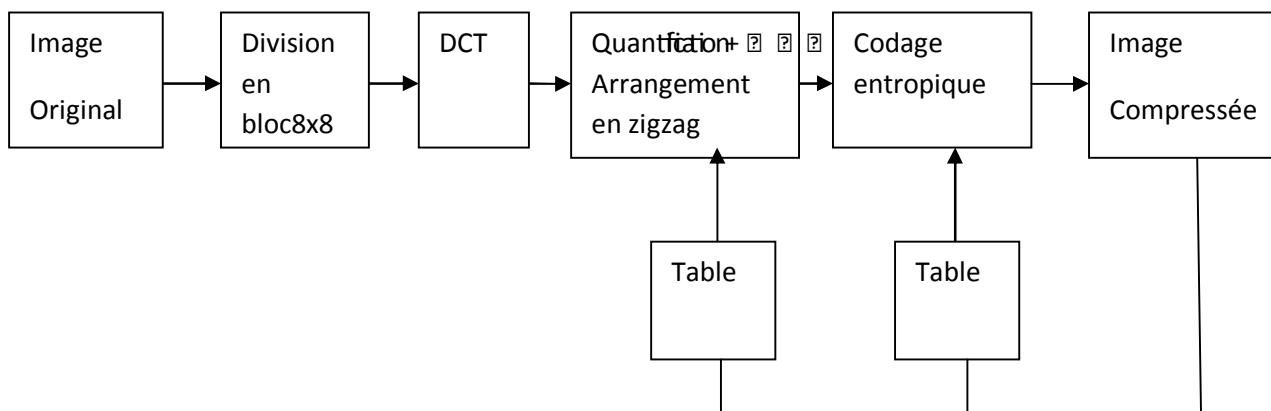
Le taux de compression peut être adapté selon les besoins d'utilisation (diffusion en ligne, stockage, etc.....) et de qualité requis. Plus le fichier est compressé plus son poids est faible, mais son potentiel de réutilisation est limité.

Les techniques définies par la norme JPEG se divisent en deux classes : les méthodes de compression avec pertes qui sont basées sur la TCD suivi d'une quantification et d'un codeur entropique. La seconde classe, concerne les processus de codage sans pertes, cette classe de codeurs n'est pas basée sur la TCD mais sur le codage MICD suivi d'un codage entropique. Pour les méthodes avec pertes, quatre codeurs ont été spécifiés : un codage de base où l'image compressée puis décompressée n'est plus identique à l'image originale, ce processus utilise la TCD et un codage de Huffman.

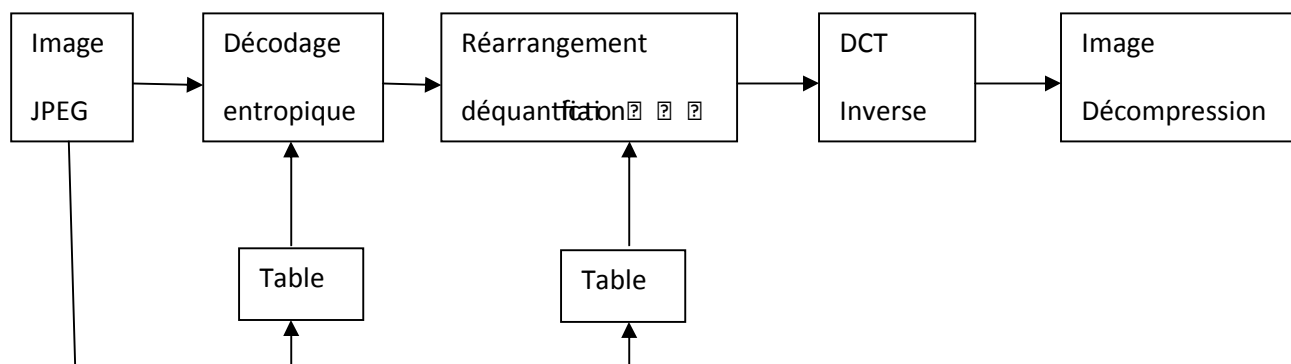
Les trois autres types de codage sont une extension de codage de base, ils diffèrent de codage de base principalement par le codage entropique en utilisant un codage arithmétique ou par restitution progressive de l'image.

La phase de décompression avec la norme JPEG se résume par la succession inverse des étapes nécessaires à la compression et les figures suivantes représentent la compression et la décompression des images avec cette méthode.

Les figures(9), (10) représentent la compression et la décompression avec JPEG :



**Figure I-8-Compression avec standard JPEG**



**Figure I-9-Décompression avec standard JPEG**

**I-8-2- La norme de compression JPEG 2000 :**

Lorsqu'on aborde le sujet de la compression d'image, la première norme qui nous vient à l'esprit est JPEG, celle-ci est en effet l'une des plus utilisées dans le domaine de la compression d'image photographiques et fut créée par le Joint Photographic Experts Groupe, il a maintenant plus de 10 ans. La compression JPEG est efficace lorsque des pertes sont tolérées, par contre cette compression n'est pas du tout efficace lorsqu'on suppose qu'il n'y ait pas de pertes. La compression JPEG a résolu ce problème ayant proposé une compression efficace dans les deux cas.

La version définitive de standard JPEG a pris forme en décembre 2000. Ce nouveau standard a pour objectif d'offrir de nouvelles fonctionnalités permettant de répondre à une demande croissante, à savoir :

- Obtenir des performances de compression supérieures à son précédent JPEG notamment pour des débits très faibles.
- Permettre d'organiser le fichier compressé de plusieurs manières.
- Avoir un mode de compression sans perte et performant.
- Fournir la possibilité de coder des parties d'une image avec une qualité supérieure à d'autres parties.

### **I-8-3-La norme de compression H-261 :**

La norme de compression **H-261** est destinée au codage des images animées pour la visiophonie (Téléphonie Visuelle), elle utilise un codage hybride qui combine entre le codage en transformée (**DCT**) pour réduire la redondance spatiale et le codage prédictif utilisé dans le but de réduire la redondance temporelle.

### **I-8-4-La norme de compression MPEG (Moving Picture Experts Groupe):**

On distingue les normes suivantes :

#### **MPEG-1 :**

MPEG-1 est un standard rend possible le stockage de séquences non entrelacées de petites taille sur des CD-ROM. Il a pour but de fournir un équivalent numérique au format d'enregistrement analogique VHS .MPEG 1 peut fournir des images de qualités TV. Avec l'augmentation des vitesses des lecteurs CD-ROM.

#### **MPEG-2 :**

MPEG-2 permet le codage de séquences à être diffusées de manière numérique (Câble satellite). MPEG-2 a plusieurs fonctions :

- La possibilité d'effectuer la lecture rapide, le retour, les ralentis, et des poses sur les séquences.

- MPEG-2 représente le standard de la télévision numérique.
- L'utilisation de différent échantillonnage de la chrominance.
- MPEG-2 fournit également la possibilité de découper la séquence en fonction d'une stabilité temporelle spatiale ou bien en fonction de la qualité.

#### **MPEG : 4**

MPEG-4 permet d'augmenter l'interactivité d'une scène animée, il a pour objectif de coder de manière efficace des séquences à très bas débits.

L'algorithme de compression MPEG-4 des séquences naturellement déjà utilisé actuellement sous des fichiers format DivX.

#### **Discussion :**

Nous avons entamé dans ce chapitre quelques généralités sur le traitement d'image, les différentes méthodes de compression telle que la compression par transformation et par quantification. Nous avons terminé ce chapitre par les différentes normes de compression d'images.

Dans le chapitre qui va suivre, nous allons nous intéresser à l'un des outils mathématiques les plus performants en traitement d'image en l'occurrence les ondelettes et l'application de sa transformée dans la compression d'images.

## Chapitre II

# La théorie des ondelettes

## II-1-Préambule :

La transformée de Fourier (1822) permet d'explorer la composition fréquentielle du signal par ses propriétés de lui appliquer facilement des opérateurs de filtrage.

La transformation en ondelettes est un outil qui découpe les données, les fonctions ou les opérateurs en composants fréquents suivant une résolution adaptée à l'échelle. Le premier à avoir proposé le nom de l'ondelette fut Jean Morlet en 1983.

Alors on utilise les ondelettes car la plupart des signaux du monde réel ne sont pas stationnaires, c'est justement dans l'évolution de leurs caractéristiques (statiques, fréquentielles, temporelles et spatiales) où réside l'essentiel de l'information qu'ils contiennent, or l'analyse de Fourier propose une approche globale, les intégrations sont faites de  $-\infty$  à  $+\infty$ , et toute notion de localisation temporelle (ou spatiale d'image) disparaît dans l'espace de Fourier, pour cela il faut trouver un compromis, une transformation qui renseigne sur le contenu fréquentiel tout en préservant la localisation afin d'obtenir une transformation temps/fréquence ou espace/échelle.

Donc la transformée en ondelettes ne sera pas intéressante comme outil d'analyse du signal que si la fonction analysante (ondelette) reste bien localisée dans le temps et en fréquence.

## II-2-Transformée de Fourier :

Les séries de Fourier sont contribuées d'une manière générale à analyser les signaux périodiques en utilisant des fonctions sinus, cosinus et exponentielles. L'analyse de Fourier permet de caractériser la régularité globale d'une fonction, quant à elle, permet une analyse dans les domaines temps-fréquence, elle est définie par :

$$\tilde{f}(w) = \int_{-\infty}^{+\infty} f(x) e^{-jwx} dx \quad (\text{II-1})$$

Telle que  $f(x)$  est la fonction à traiter. Cette analyse était considérée parfaite car elle permettait de déterminer les caractéristiques fréquentielles d'un signal. Toute en avançant dans la science, l'analyse de Fourier se heurte à des problèmes sur certains points, par exemple elle ne permet pas de faire simultanément une analyse en temps et en fréquence du signal, d'où il vient le terme générique « transformée de Fourier à fenêtre glissante ».

**II-2-1 Transformée de Fourier à fenêtre glissante (TFFG):**

Denis Gabor qui a introduit en 1946 cette approche (TFFG), basée sur un découpage du signal en morceau d'ondes, qui localise une information donnée par la TF en utilisant les fenêtres spatiales  $g(x)$  dans l'intégral de Fourier ; ainsi le terme « glissant » découle de la translation de  $g(x)$  sur tout le domaine spatial.

Cette fonction est donnée par la relation suivante :

$$\tilde{f}(w, \tau) = \int_{-\infty}^{+\infty} f(x) g(x - \tau) e^{-jwx} dx \quad (\text{II-2})$$

L'inconvénient de cette méthode est que la fenêtre a une longueur fixe, elle ne convient pas à la représentation des signaux ayant des composantes de tailles différentes de la taille de la fenêtre. De plus, le calcul de la TFFG doit s'effectuer pour chaque changement de la taille de la fenêtre d'où une charge de calculs importants.

Si on pose :

$$\Psi_{\tau, w}(x) = g(x - \tau) e^{-jwx} \quad (\text{II-3})$$

Alors on interprète cette transformation comme la projection de  $f$  sur la « base » des fonctions à fenêtres glissantes  $\Psi$  :

$$Tf \text{ glissante} = \langle f, \Psi_{\tau, w} \rangle \quad (\text{II-4})$$

La notion  $\langle f, g \rangle$  respecte le produit scalaire :

$$\langle f, g \rangle = \int_{-\infty}^{+\infty} f(x) g(x) dx \quad (\text{II-5})$$

Un certain nombre de fenêtres sont utilisées, les plus connues sont les fenêtres de Hanning et de Gauss

**II-3-Ondelettes :**

La transformation en ondelettes est très intéressante en traitement du signal, et en compression, grâce à son approche en multi résolution. La décomposition en ondelettes est une extension de la décomposition en sous-bandes, avec une manière particulière de choisir les filtres. La décomposition en ondelettes aboutit à une image de détails et une image de basse résolution. Cette dernière est à nouveau décomposée en une image de détail et une

image basse résolution, et ainsi de suite. Plus on avance dans la décomposition, plus l'échelle de "grossissement" est importante, et plus les fréquences sont basses. Les sous-images ainsi obtenues peuvent être codées avec des stratégies différentes. La méthode de décomposition en ondelettes présente des qualités dites de "régularité" des filtres de décomposition qui sont plus avantageuses que celles des filtres usuels de décomposition sous-bandes.

### II-3-1-Définition de l'ondelette:

Les ondelettes sont des fonctions élémentaires oscillantes et de moyenne nulle, générées à partir d'une ondelette analysante, aussi appelée ondelette mère. Cette dernière est une fonction de base que l'on va traduire et décaler pour recouvrir le plan temps-fréquence et analyser le signal. Elles sont normalisées suivant l'équation :

$$\psi_{a,b}(t) = a^{-1/2} * \psi((t - b)/a) \quad (\text{II-6})$$

Avec :

$\Psi(t)$  : fonction bien localisée en temps et en fréquence.

$a$ : Paramètre de dilatation.

$b$  : Paramètre de translation.

$a \in \mathbb{R}_+^*$  et  $b \in \mathbb{R}$

### II-3-2-Transformée en ondelette continue(CWT) :

La transformée en ondelette est dite continue car les paramètres de dilatation et de translation prennent des valeurs continues. La transformée en ondelette d'une fonction  $f$  est une projection de celle-ci sur la famille d'ondelette choisie. On définit la transformée en ondelette continue d'un signal  $f(x) \in L^2(\mathbb{R})$  :

$$W(a, b) = CWT(f)(a, b) = \psi_{(a,b,f)} = |a|^{-\frac{1}{2}} \int_{-\infty}^{+\infty} f(x) \overline{\psi\left(\frac{x-b}{a}\right)} dx \quad (\text{II-7})$$

La transformée inverse de la (TOC) n'est pas unique du fait de la redondance de l'information, ce pour cela que l'ondelette  $\Psi$  doit vérifier la condition d'admissibilité suivante :

$$C_\psi = \int_{-\infty}^{+\infty} \frac{|\psi(w)|^2}{w} dw < \infty \quad (\text{II-8})$$

### II-3-2-a-Transformée inverse :

La transformée en ondelette continue admet une formule de reconstruction dans  $L^2(\mathbb{R})$ :

$$f(x) = \frac{1}{C_\psi} \iint_{\mathbb{R}} w_f(a, b) \psi\left(\frac{x-b}{a}\right) \frac{dadb}{a^2} \quad (\text{II-9})$$

$$f(x) = \frac{1}{C_\psi} \quad (\text{II-10})$$

Dans le sens :

$$\lim_{\varepsilon \rightarrow 0^+} \frac{1}{C_\psi} \iint_{b \in \mathbb{R} \quad |a| > \varepsilon} w_f(a, b) \psi\left(\frac{x-b}{a}\right) \frac{dadb}{a^2} = f(x) \quad (\text{II-11})$$

### II-4-Transformées en ondelette discrètes (dwt) :

La transformation en ondelette discrète est une transformation bien localisée en temps et en fréquence, dont les paramètres sont discrétisés en vue d'une analyse spectrale des signaux numériques. Les paramètres (a et b) prennent des valeurs discrètes selon l'expression suivante :

$$(a, b) = (a_0^j, kb_0 a_0^j) \quad \text{Ou} \quad (j, k) \in \mathbb{Z}^2$$

Ou :  $a_0 > 1$  (le pas de dilatation).

$b_0 \neq 0$  (le pas de translation).

La transformée en ondelettes discrète d'une fonction est donnée par l'équation suivante :

$$W(a, b) = (\psi_{j,k}, f) = a_0^{-j/2} \int_{-\infty}^{+\infty} \psi(a_0^{-j}t - kb_0) f(t) dt \quad (\text{II-12})$$

En 1987, Y Meyer a démontré qu'il était possible de construire des bases d'ondelettes orthogonales en discrétisant les paramètres de dilatation et de translation a et b respectivement de la manière suivante :

$$a = 2^{-j} ; b = ka = k2^{-j} \quad j, k \in \mathbb{Z}$$

On obtient ainsi des bases dans  $L^2(\mathbb{R})$  de la forme :

$$\{\psi_{j,k}\}_{j,k \in \mathbb{Z}} = \{2^{j/2} \cdot \psi(2^j t - k)\}_{j,k \in \mathbb{Z}} \quad (\text{II-13})$$

### II-5-Base d'ondelette orthogonale :

Y. Mayer et S.Mallat proposent une méthode de construction de base d'ondelette orthonormée. Cette approche est appréciée dans le domaine du traitement de signal surtout en compression d'image.

Il existe des familles d'ondelettes discrètes formant des bases orthonormées de  $L^2(\mathbb{R})$ , la réduction de la redondance de la représentation en ondelettes est maximale.

Les paramètres  $s$  de dilatation et de translations sont choisis comme suit :

$$(a, b) = (a_0^j, kb_0 a_0^j) = (2^{-j}, k2^{-j}) \quad \text{avec } a_0=2 \text{ et } b_0=1$$

La décomposition de  $f(x) \in L^2(\mathbb{R})$  peut s'écrire alors :

$$f(x) = \sum_{j,k \in \mathbb{Z}} (f(x), \psi_{j,k}(x)) \psi_{j,k}(x) \quad (\text{II-14})$$

### Remarque :

- ∅ La transformée en ondelettes ne sera pas intéressante comme outil d'analyse du signal que si la fonction analysante (ondelette) reste bien localisée dans le temps et en fréquence.
- ∅ Les transformées continues sont obtenues en prenant le facteur d'échelle  $a$  et le pas de translation  $b$  dans l'ensemble des nombres réels. Ces transformations sont très redondantes car l'espace temps-fréquence est parcouru continûment. Alors dans la pratique cette transformation ne peut être effectuée que approximativement et il y a toujours en fait une discrétisation du calcul qui est opérée.

## II-6-Approximation de base :

### II-6-1-Espace d'approximation :

On parle dans l'espace  $L^2(\mathbb{R})$  de fonctions continues d'une variable réelle et de carré intégrable.

Une analyse à la résolution  $j$  de la fonction sera obtenue par action d'un opérateur  $A_j$  sur  $f$ , tel que :  $A_j f \in V_j$ . Donc  $V_j$  est un sous espace de  $L^2(\mathbb{R})$ ,  $A_j$  sera un projecteur (idempotent).

À l'aide de  $V_j$  emboîtés les uns dans les autres, tels que le passage de l'un à l'autre nécessite d'un changement d'échelle (zoom), on construit une analyse multi-résolution.

Exemple : soit  $f(x) \in V_j \Leftrightarrow f\left(\frac{x}{2}\right) \in V_{j+1}$  tel que  $V_{j+1} \subset V_j$ . ce que correspond à une dilatation 2,  $V_{j+1}$  contient des signaux plus « grossière » que l'espace  $V_j$ .

Soit un ensemble de sous espace de  $L^2$  tels que  $\dots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \dots V_{j+1} \subset V_j$

$$\begin{cases} \overline{\bigcup_{j \in Z} V_j} = L^2(R) \\ \overline{\bigcap_{j \in Z} V_j} = \{0\} \end{cases} \tag{II-15}$$

$$\forall n \in Z \quad f(x) \in V_j \Leftrightarrow f(2^{-1}x) \in V_{j+1} \text{ .Ou } f(2^j x) \in V_0 \tag{II-16}$$

$$\forall k \in Z \text{ si } f(x) \in V_0 \Leftrightarrow f(x - k) \in V_0 \quad (\text{Invariance par translation})$$

Cette ensemble définit une analyse multi-résolution de  $L^2(R)$

À partir de ces conditions on peut dire qu'il existe une base orthonormée de  $V_j$ . Cette fonction sera  $\varphi(x) \in L^2(R)$ . Et pour les fonctions de bases seront reconstruites suivant la relation suivante :

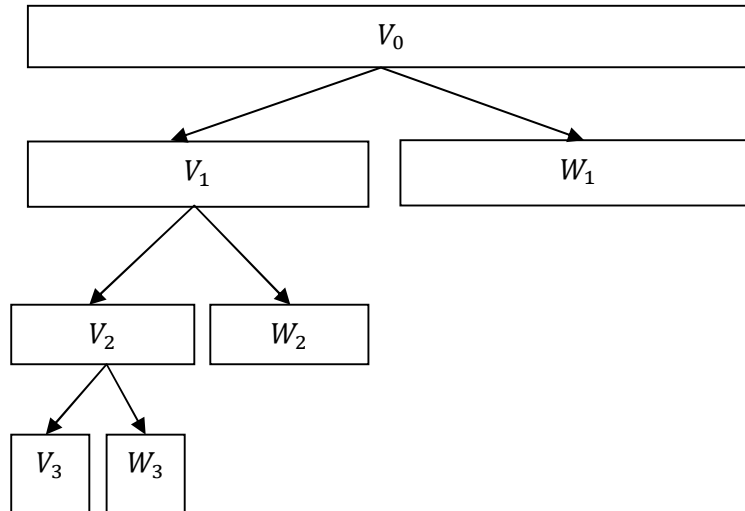
$$\varphi_{j,n}(x) = 2^{-\frac{j}{2}} \varphi(2^{-j}x - n) \quad \forall n \in Z \tag{II-17}.$$

Il suffit d'ailleurs que  $\varphi(\cdot, -n)$  soit une base de  $V_0$  .

Donc la base sera orthonormée si et seulement si :

$$\int_{-\infty}^{+\infty} \varphi(x) \varphi^*(x+n) dx = \delta(n) \quad \forall n \in Z \tag{II-18}$$

**Exemple d'un schéma de l'analyse multi-résolution :**



**Figure-II-1** : schéma de l'analyse multi-résolution

La relation d'orthogonalité pour les fonctions de base de même échelle donnée pourra s'écrire :

$$\langle \varphi_{j,n}, \varphi_{j,k} \rangle = \delta(n - k) \quad \forall n, k, j \in \mathbb{Z} \quad (\text{II-19})$$

### II-6-2-Espace des détails :

On peut définir pour chaque  $V_j$  son complément orthogonal  $W_j$  dans  $V_{j-1}$  tel que comme  $W_{j-1}$  est orthogonal à  $V_{j-1}$ , alors  $W_{j-1}$  sera orthogonal à  $W_j$ , on peut écrire cette propriété comme suite :  $\forall j, k \neq j$  alors  $W_j \perp W_k$ .

Les sous espaces  $W_j$  ne forment pas une famille d'espaces emboîtés, mais ils gardent les propriétés d'échelle et de translation.

Dans ces conditions on peut montrer qu'il existe une fonction appelée « ondelette » qui par dilatation et translation engendre une base orthonormée des  $W_j$  et donc de  $L^2$   $\Psi(x) \in L^2(\mathbb{R})$ .

Les fonctions de bases sont construites suivant cette relation :

$$\Psi_j(x) = 2^{-\frac{j}{2}} \Psi(2^{-j}x - n) \quad n \in \mathbb{Z} \quad (\text{II-20})$$

L'ortho normalité de la base d'ondelettes s'écrit :

$$\langle \Psi_{j,n}, \Psi_{i,k} \rangle = \delta(j-i)\delta(n-k) \quad i, j, n, k \in Z \quad (\text{II-21})$$

L'approximation à l'échelle immédiatement plus fine pourra donc être reconstruite en utilisant les détails du signal fournis par sa projection sur la base  $W_j$  suivant cette relation :

$$A_{j-1}f = A_j + \sum_n \langle f, \Psi_{j,n} \rangle \Psi_{j,n} \quad (\text{II-22})$$

Et on notera  $D_j$  le projecteur sur et le signal de détail sera décrit par la suite numérique :

Posant d'abord

$$D_j f = \sum_n \langle f, \Psi_{j,n} \rangle \Psi_{j,n} \quad (\text{II-23})$$

Et la formule de reconstruction s'écrit :

$$A_{j-1}f = A_j f + D_j f \quad (\text{II-24})$$

### II-7- Analyses multi-résolutions :

L'analyse multi-résolution est un concept qui a été développé par Stéphane Mallat. Autrement dit c'est un outil mathématique bien adapté à l'utilisation d'ondelette en analyse d'image. Cette analyse peut faire l'approximation de n'importe quel signal par changement d'échelle, à chaque échelle, le signal sera remplacé par l'approximation la plus adéquate que l'on puisse y tracer ou chaque approximation est une version lissée de la précédente. La fonction d'échelle dilatée donne une image du signal à des résolutions différentes, en allant des échelles les plus grosses vers les plus fines, on accède à des représentations de plus en plus précises du signal donné.

#### III-7-1-Définition:

L'analyse multi-résolution est une famille de sous-espaces de  $L^2(\mathbb{R})$  notée  $(V_j)$  emboîtés les uns et les autres, telle que le passage de l'un à l'autre soit le résultat d'un changement d'échelle (zoom). Ces sous espaces vérifient les propriétés suivantes :

$$1 - \forall j \in Z \quad V_j \subset V_{j+1} \quad (\text{II-25})$$

$$2 - \bigcap_{j \in Z} V_j = \{0\} \quad (\text{II-26})$$

$$3 - \bigcup_{j \in \mathbb{Z}} V_j \text{ est dense dans } L^2(\mathbb{R}) \quad (\text{II-27})$$

$$4- \forall f \in L^2(\mathbb{R}), \forall j \in \mathbb{Z}: \quad f(x) \in V_j \Leftrightarrow f\left(\frac{x}{2}\right) \in V_{j+1} \quad (\text{II-28})$$

$$5- \forall f \in L^2(\mathbb{R}), \forall k \in \mathbb{Z}^N: \quad f(x) \in V_0 \Leftrightarrow f(x - k) \in V_0 \quad (\text{II-29})$$

### III-7-2- Interprétation des propriétés de l'analyse multi-résolution :

- ∅ L'information contenue dans  $V_j$  est également incluse dans  $V_{j+1}$ , cela signifie que plus on monte en résolution plus le signal tend vers le signal originale (propriété 1 et 2)
- ∅ Inversement plus on descend en résolution, le signal contient moins d'information (propriété 2)
- ∅ Un signal appartenant à un sous espace vectoriel  $V_j$ , a ses détails dilatés d'un facteur de 2 quand il est considéré dans  $V_{j+1}$  (propriété 4)
- ∅ Tandis qu'un signal translaté reste invariant par rapport à la résolution (propriété 5)

### III-7-3-base d'ondelettes de $L^2(\mathbb{R})$ :

On note  $W_j$  l'espace complémentaire et orthogonale de  $V_j$  dans  $V_{j-1}$ , il vérifie donc :

$$V_{j-1} = V_j \oplus W_j \quad (\text{II-30}).$$

À partir de l'analyse multi-résolution on peut déduire les propriétés des espaces  $W_j$  :

$$-\forall i, j \in \mathbb{Z}, i \neq j \Rightarrow W_i \perp W_j \quad (\text{II-31})$$

$$-\bigoplus_{j \in \mathbb{Z}} W_j = L^2(\mathbb{R}) \quad (\text{II-32})$$

$$-f(2t) \in W_{j-1} \Leftrightarrow f(t) \in W_j \quad (\text{II-33})$$

$$-f(t) \in W_0 \Rightarrow \forall n \in \mathbb{Z}, f(t - 1) \in W_0 \quad (\text{II-34})$$

$-\exists \Psi \in W_0$  tel que  $[\Psi(t - n); n \in \mathbb{Z}]$  soit une base orthonormée de  $W_0$  .

### III-7-4- Algorithme de S.Mallat (Algorithme rapide) :

La réalisation d'une analyse multi-résolution d'un signal consiste à décomposer ce signal  $f(t)$  suivant des versions à différentes résolutions données par les coefficients d'approximation :

$$A_k^j = \int_{-\infty}^{+\infty} f(t) \varphi_k^j(t) dt \tag{II-35}$$

La synthèse consiste à reconstruire le signal à partir de ces différentes versions (les coefficients  $A_k^j$ ), et les fluctuations du signal entre deux échelles successives données par les coefficients de détails :

$$d_k^j = \int_{-\infty}^{+\infty} f(t) \psi_k^j(t) dt \tag{II-36}$$

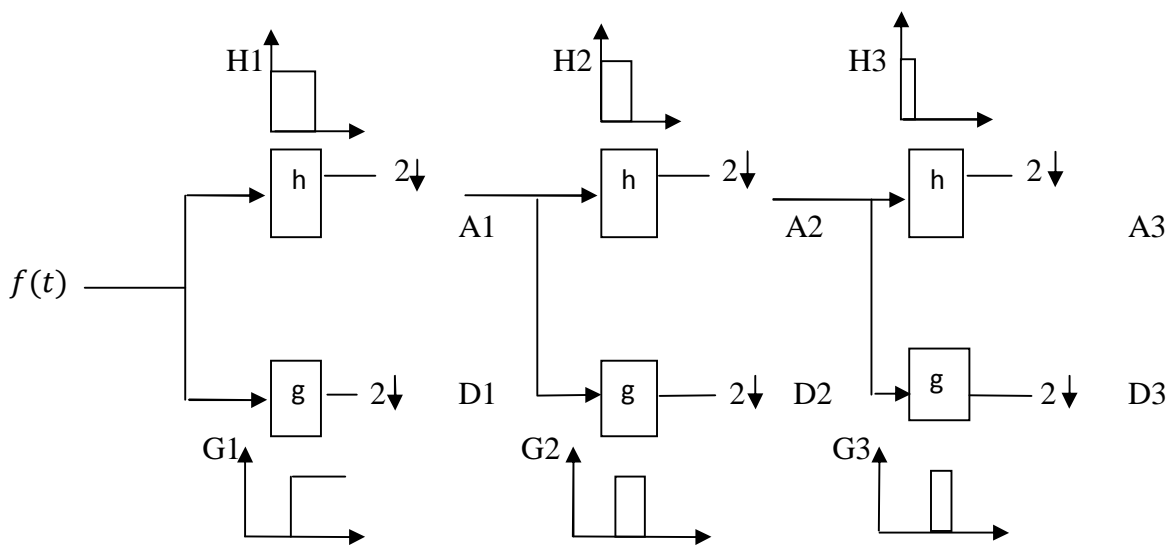
**III-7-4-1-Décomposition (Analyse) :**

L'étape qui permet la décomposition d'un signal en sous-bandes est appelée analyse, elle s'effectue en utilisant deux filtres à réponse impulsionnelle finie, l'un passe-haut ( $h_n$ )  $n \in Z$ , l'autre passe-bas ( $g_n$ )  $n \in Z$ .

Cette décomposition s'effectue par l'application de l'algorithme suivant :

$$\begin{cases} A_k^{j+1} = \sum_n h_{n-2k} A_n^j \\ d_k^{j+1} = \sum_n g_{n-2k} A_n^j \end{cases} \tag{II-37}$$

A chaque étape de l'algorithme, les coordonnées dans l'espaces  $V_{j+1}$  et  $W_{j+1}$  sont déduites de celles de l'approximation dans  $V_j$  par une convolution avec les suites  $(h_n)_{n \in Z}$  ou  $(g_n)_{n \in Z}$  suivies d'une décimation (ou enlève un échantillon sur deux).



**Figure -II-2-** algorithme de décomposition de  $f(t)$ .

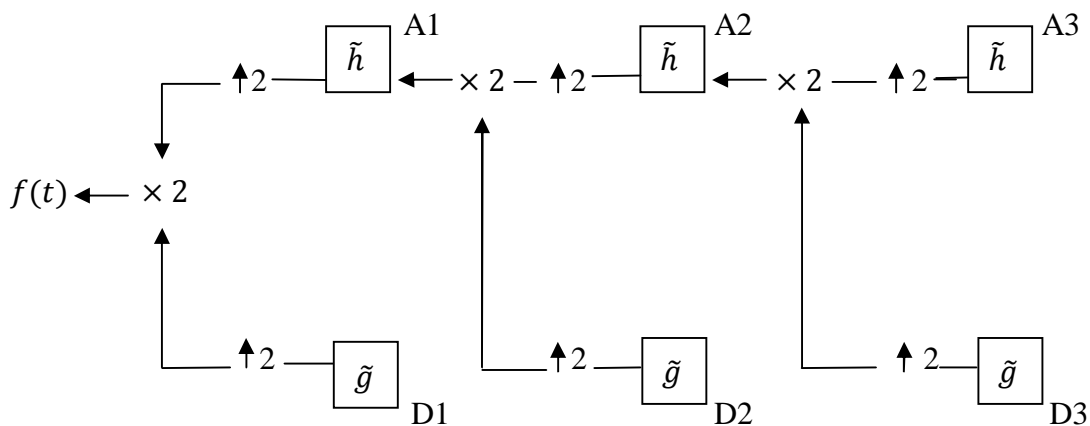
**III-7-4-2-Reconstruction (Synthèse) :**

L'étape qui permet reconstruction d'un signal en sous-bandes est appelée synthèse, elle s'effectue en utilisant deux filtres à réponse impulsionnelle finie, l'un passe-haut  $(h_n)_{n \in \mathbb{Z}}$ , l'autre passe-bas  $(g_n)_{n \in \mathbb{Z}}$ .

La synthèse d'une échelle est obtenue par la formule suivante :

$$A^{j-1} = 2 \sum_n h_{n-2k} A_k^j + 2 \sum_n g_{n-2k} d_k^j \tag{II-38}$$

Cette opération, qui permet de trouver l'approximation dans  $V_{j-1}$  à partir des coordonnées dans  $V_j$  et  $W_j$  consiste à échantillonner ces deux suites en intercalant des zéros, puis en interpolant avec les filtres  $(h_n)_{n \in \mathbb{Z}}$  ou  $(g_n)_{n \in \mathbb{Z}}$ .



**Figure –II-3-** Algorithme de reconstruction

- ↓ 2 : conserve qu'un échantillon sur 2.
- ↑ 2 : met un zéros entre chaque échantillons
- × 2 : multiplication par 2
- h : filtre passe-bas de décomposition.
- g : filtre passe-haut de décomposition.
- h-tilde : filtre passe-bas de reconstruction.
- g-tilde : filtre passe-haut de reconstruction.

**II-8-Bases biorthogonale :**

La notion biorthogonale a été introduite en 1987 par Tchamitchian et celle de l'ondelette biorthogonale, en 1992 par Cohen et All. Ces familles de fonction permettent l'analyse multi résolution du signal et assurent la possibilité d'une reconstruction parfaite du signal. L'analyse reste non redondante, possède les propriétés de symétrie et de simplicités requise, mais conduit à une corrélation entre les composants de signale plus grande que dans le cas des bases orthonormés.

**II-8-1-Définition :**

Les ondelettes biorthogonales sont construites à partir de notion de gamme ou structure oblique. On considère une famille  $(\Psi_i)_{i \in I}$  étant une structure oblique d'un espace de Hilbert H. alors il existe une oblique duale  $(\tilde{\Psi}_i)_{i \in I}$  si et seulement si on a :

$$f = \sum_i (f, \Psi_i)_{\mathcal{H}} \tilde{\Psi}_i = \sum_i (f, \tilde{\Psi}_i)_{\mathcal{H}} \Psi_i \quad (\text{II-39})$$

On dit que les familles  $(\Psi_i)_{i \in I}$  et  $(\tilde{\Psi}_i)_{i \in I}$  sont biorthogonale si :

$$(\Psi_i, \tilde{\Psi}_j)_{\mathcal{H}} = \delta(i, j) \quad (\text{II-40})$$

Ou  $\delta = (i, j)$  est le symbole de Kronecker.

**II-8-2-Analyse biorthogonale :**

On considère deux familles de fonction duales, l'une sera utilisée pour l'analyse (transformée directe) et l'autre pour la reconstruction ou synthèse (transformée inverse). Ces familles sont orthogonales entre elles mais ne sont pas orthogonales en elles mêmes. Alors nous avons besoins des deux familles duales de sous -espace emboîtées définies par les deux familles de fonctions d'échelles, chacune formant une base non orthogonale. Les propriétés à respecter sont l'indépendance linéaire, l'existence et l'unicité de la décomposition.

$$\begin{cases} V_j \subset V_{j-1} \\ \bar{V}_j \subset \bar{V}_{j-1} \end{cases} \quad (\text{II-41})$$

Les sous espace complémentaire sont définis différemment car ils sont complémentaire sous espace dual.

$$\begin{cases} W_j \perp \bar{V}_j \\ \bar{W}_j \perp V_j \end{cases} \quad (\text{II-42})$$

Le passage d'une résolution à la suivante est fait suivant cette décomposition :

$$\begin{cases} V_{j-1} = V_j \oplus W_j & \text{mais } V_j < W_j \\ \bar{V}_{j-1} = \bar{V}_j \oplus \bar{W}_j & \text{mais } \bar{V}_j < \bar{W}_j \end{cases} \quad \text{avec : } W_j \perp \bar{W}_{k \neq j \in \mathbb{Z}} \quad (\text{II-43})$$

Les propriétés d'orthogonalités des bases de sous-espaces seront :

$$\begin{cases} \langle \Psi_{j,n}, \bar{\Psi}_{i,m} \rangle = \delta(j-i)\delta(n-m) \\ \langle \varphi_{j,n}, \bar{\varphi}_{j,m} \rangle = \delta(n-m) \end{cases} \quad (\text{II-44})$$

L'orthogonalité vis-à-vis de la translation et dilatation n'est requise que d'un espace à son dual. Alors pour créer une analyse biorthogonale, il suffira de définir une famille complète (une base) de fonctions construites par dilatation-translation d'une fonction mère. La base duale et les filtres associés sont déterminés par les propriétés d'orthogonalités (équation (II-56)) éventuellement exprimées dans l'espace de Fourier.

L'analyse et la synthèse d'un signal se feront suivant des algorithmes analogues à ceux du cas orthogonal :

$$A_{j-1} = A_j + D_j \quad (\text{II-45})$$

$$\text{Avec : } \begin{cases} A_j f = \sum \langle f, \bar{\varphi}_{j,n} \rangle \varphi_{j,n} \\ D_j f = \sum \langle f, \bar{\Psi}_{j,n} \rangle \Psi_{j,n} \end{cases} \quad (\text{II-46})$$

Les algorithmes récursifs seront identiques, avec pour l'analyse :

$$\begin{cases} a_n^j = \sum_l \tilde{h}(2n-l) a_l^{j-1} \\ d_n^j = \sum_l \tilde{g}(2n-l) a_l^{j-1} \end{cases} \quad (\text{II-47})$$

Et on reconstruction :

$$a_n^{j-1} = \sum_k a_k^j h(n-2k) + \sum_k d_k^j g(n-2k) \quad (\text{II-48})$$

$$\text{Avec : } \begin{cases} a_n^j = \langle f, \bar{\varphi}_{j,n} \rangle \\ d_n^j = \langle f, \bar{\Psi}_{j,n} \rangle \end{cases} \quad (\text{II-49})$$

Ø Schéma illustrant les algorithmes d'analyse et de synthèse :

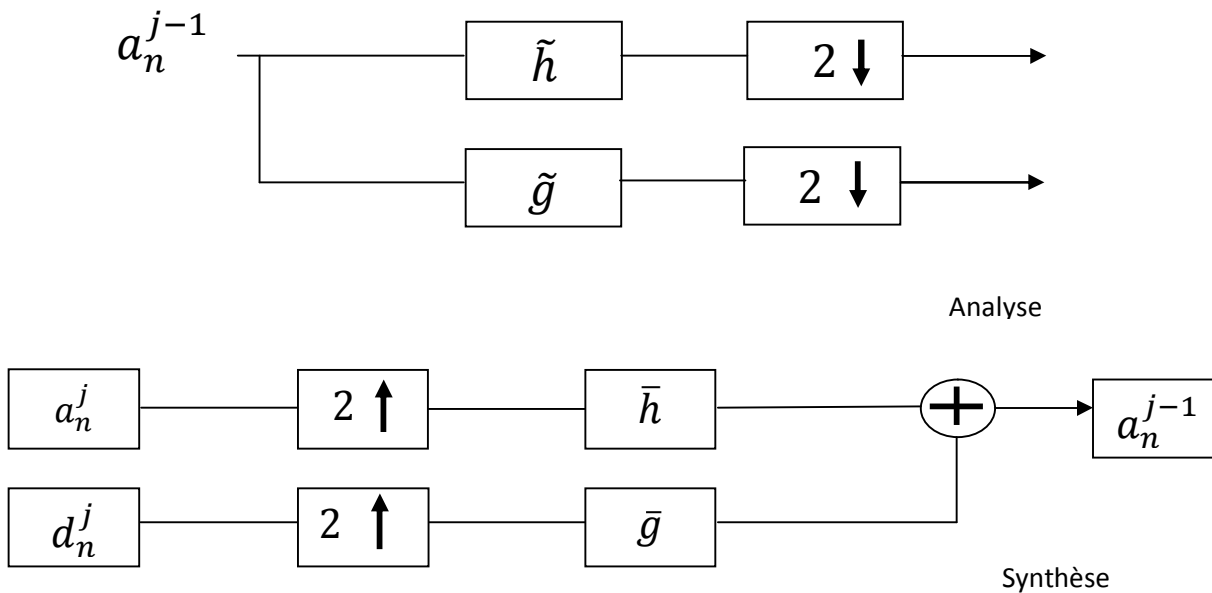


Figure II-4: algorithme d'analyse et de synthèse

**II-9-Analyse multi-résolution deux dimensions séparables :**

**II-9-1-Principe :**

Maintenant on va passer au cas bidimensionnel et spécifiquement aux images. Le principe est d'utiliser le produit tensoriel des opérateurs définis en une dimension.

L'avantage de cette méthode est qu'elle simplifie les algorithmes de décomposition et de reconstruction, mais présente l'inconvénient de privilégier les lignes et les colonnes. Ainsi il existe l'algorithme de décomposition en quinconce.

Nous nous limiterons au cas dyadique qu'est le plus utilisé dans la pratique.

Soit la matrice de dilatation qui prend la valeur suivante :  $J = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = 2I$  ( $I$  est la matrice identique)

On aura  $JX = 2X$  et  $J^j X = 2^j X$

Et la famille de fonctions dilatées –translatées sera notée comme suit :

$$\varphi_{j,m,n}(x, y) = 4^{-\frac{j}{2}} \varphi(2^{-j}x - m, 2^{-j}y - n) \tag{II-50}$$

Donc le facteur de dilatation surfacique est 4.

Comme dans le cas de dimension un (D1), on reconstruit l'analyse multi-résolution avec les mêmes axiomes en remplaçant la variable par un vecteur.

$$\forall k \in Z^2 \text{ si } f(X) \in V_0 \Leftrightarrow f(X - k) \in V_0 \quad (\text{II-51})$$

$$\forall j \in Z \text{ si } f(X) \in V_j \Leftrightarrow f(J^{-1}) \in V_{j+1} \text{ ou } f(J^j) \in V_0 \quad (\text{II-52})$$

Soit  $\{V_j^{(2)}\}_{j \in Z}$  une analyse multirésolution séparable de  $L^2(R^2)$ ,  $\varphi(x, y) = \varphi(x)\varphi(y)$  la fonction d'échelle de  $\{V_j^{(2)}\}_{j \in Z}$  et le base de sous espace de détail (ondelettes) sera constitué par 3 familles d'ondelettes :

$$\begin{cases} \psi^h(x, y) = \varphi(x)\psi(y) = \psi^1(x, y) \\ \psi^v(x, y) = \psi(x)\varphi(y) = \psi^2(x, y) \\ \psi^d(x, y) = \psi(x)\psi(y) = \psi^3(x, y) \end{cases} \quad (\text{II-53})$$

Soit  $\{V_j^{(2)}\}_{j \in Z}$  une analyse multirésolution séparable de  $L^2(R^2)$ ,

$\varphi(x, y) = \varphi(x)\varphi(y)$  la fonction d'échelle de  $\{V_j^{(2)}\}_{j \in Z}$  et le base de sous espace de détail (ondelettes) sera constitué par 3 familles d'ondelettes :

$\{\psi_{jnm}^1, \psi_{jnm}^2, \psi_{jnm}^3\}_{(n,m) \in Z^2}$  est une base orthonormé de  $W^2$

$\{\psi_{jnm}^1, \psi_{jnm}^2, \psi_{jnm}^3\}_{(j,n,m) \in Z^3}$  est une base orthonormée de  $L^2(R^2)$  avec :

$$\begin{aligned} \psi_{jnm}^1(x, y) &= 2^{-j/2} \varphi(2^{-j}x - n) \psi(2^{-j}y - m) \\ \psi_{jnm}^2(x, y) &= 2^{-j/2} \psi(2^{-j}x - n) \varphi(2^{-j}y - m) \\ \psi_{jnm}^3(x, y) &= 2^{-j/2} \psi(2^{-j}x - n) \psi(2^{-j}y - m) \end{aligned} \quad (\text{II-54})$$

### II-9-2-algorithmes de décomposition dyadique séparable et de reconstruction :

L'obtention des coefficients d'approximation et de détails d'une fonction  $f(u, v)$  se fait de la même manière, comme dans le cas d'une dimension, ces expressions sont données comme suit :

$$C_{k,l}^j = \langle f, \varphi_{j,k,l} \rangle = 2^{-j} \langle f(u, v), \varphi(2^{-j}u - k, 2^{-j}v - l) \rangle \quad (\text{II-55})$$

$$\begin{aligned}
D_{k,l}^{1,j} &= \langle f, \psi_{j,k,l}^1 \rangle = 2^{-j} \langle f(u, v), \psi^1(2^{-j}u - k, 2^{-j}v - l) \rangle \\
D_{k,l}^{2,j} &= \langle f, \varphi_{j,k,l} \rangle = 2^{-j} \langle f(u, v), \psi^2(2^{-j}u - k, 2^{-j}v - l) \rangle \\
D_{k,l}^{3,j} &= \langle f, \varphi_{j,k,l} \rangle = 2^{-j} \langle f(u, v), \psi^3(2^{-j}u - k, 2^{-j}v - l) \rangle \\
C_{k,l}^j &= \sum_n \sum_m h(2k - n)h(2l - m)C_{n,m}^{j+1} \\
D_{k,l}^{1,j} &= \sum_n \sum_m h(2k - n)g(2l - m)C_{n,m}^{j+1} \\
D_{k,l}^{2,j} &= \sum_n \sum_m g(2k - n)h(2l - m)C_{n,m}^{j+1} \\
D_{k,l}^{3,j} &= \sum_n \sum_m g(2k - n)g(2l - m)C_{n,m}^{j+1}
\end{aligned} \tag{II-56}$$

La reconstruction se fait selon la formule (II-57) suivante :

$$\begin{aligned}
C_{k,l}^{j+1} &= \sum_n \sum_m [\tilde{h}(k - 2n)\tilde{h}(l - 2m)C_{n,m}^j \\
&\quad + \tilde{h}(k - 2n)\tilde{g}(l - 2k)D_{n,m}^{1,j} \\
&\quad + \tilde{g}(k - 2n)\tilde{h}(l - 2m)D_{n,m}^{2,j} + \tilde{g}(k - 2n)\tilde{g}(l - 2n)D_{n,m}^{3,j}]
\end{aligned}$$

Les algorithmes de décomposition et de reconstruction se déduisent de même façon que dans le cas d'une dimension. La séparabilité a pour objectif de rendre le calcul simple et rapide. Le filtrage bidimensionnel séparable est effectué sur une image en appliquant le filtrage monodimensionnel et sous échantillonnage de manière indépendante sur les lignes et sur les colonnes. Ainsi une image est décomposée en quatre sous images : l'image d'approximation, l'image de détails horizontaux, l'image de détails verticaux et l'image de détails diagonaux. L'inconvénient de la décomposition séparable qu'il est anisotrope.

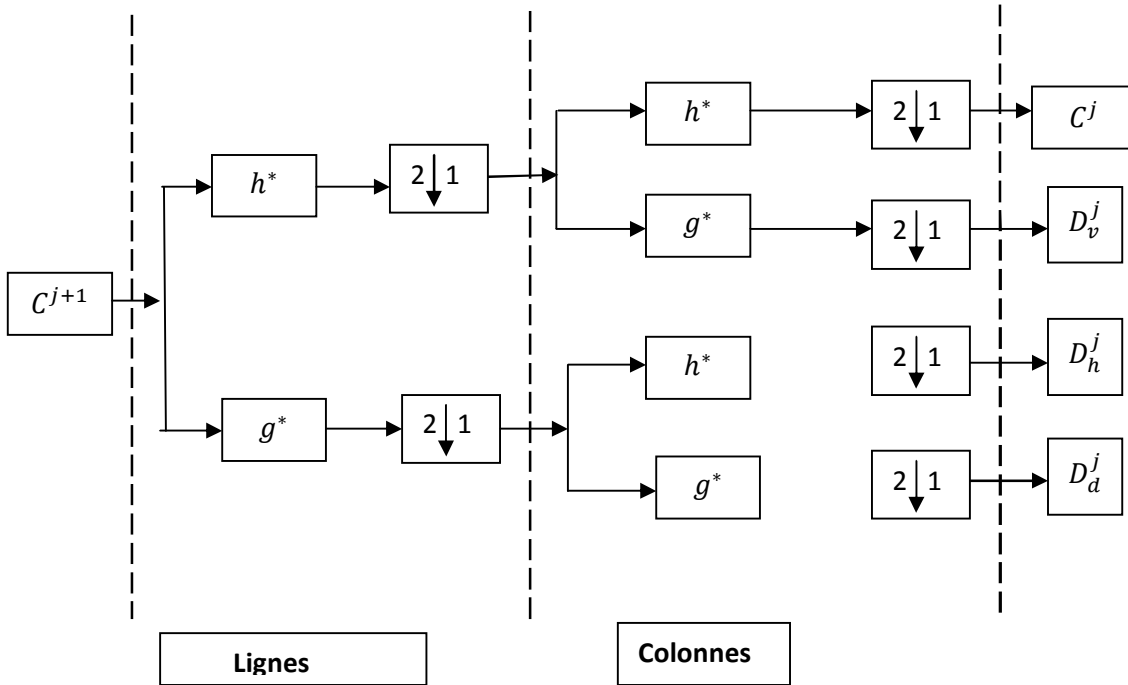


Figure II-5 : schéma de décomposition

$A^3$	$DH^3$	$DH^2$	$DH^1$
$DV^3$	$DD^3$		
$DV^2$		$DD^2$	
$DV^1$			$DD^1$

$w_3 \otimes v_2$	$w_2 \otimes v_2$	$w_2 \otimes v_2$	$w_1 \otimes v_1$
$w_2 \otimes v_2$	$w_2 \otimes v_2$		
$v_2 \otimes w_2$		$w_3 \otimes v_3$	
$v_1 \otimes w_1$			$w_1 \otimes w_1$

Figure II-6 : algorithme de décomposition en 2D dyadique

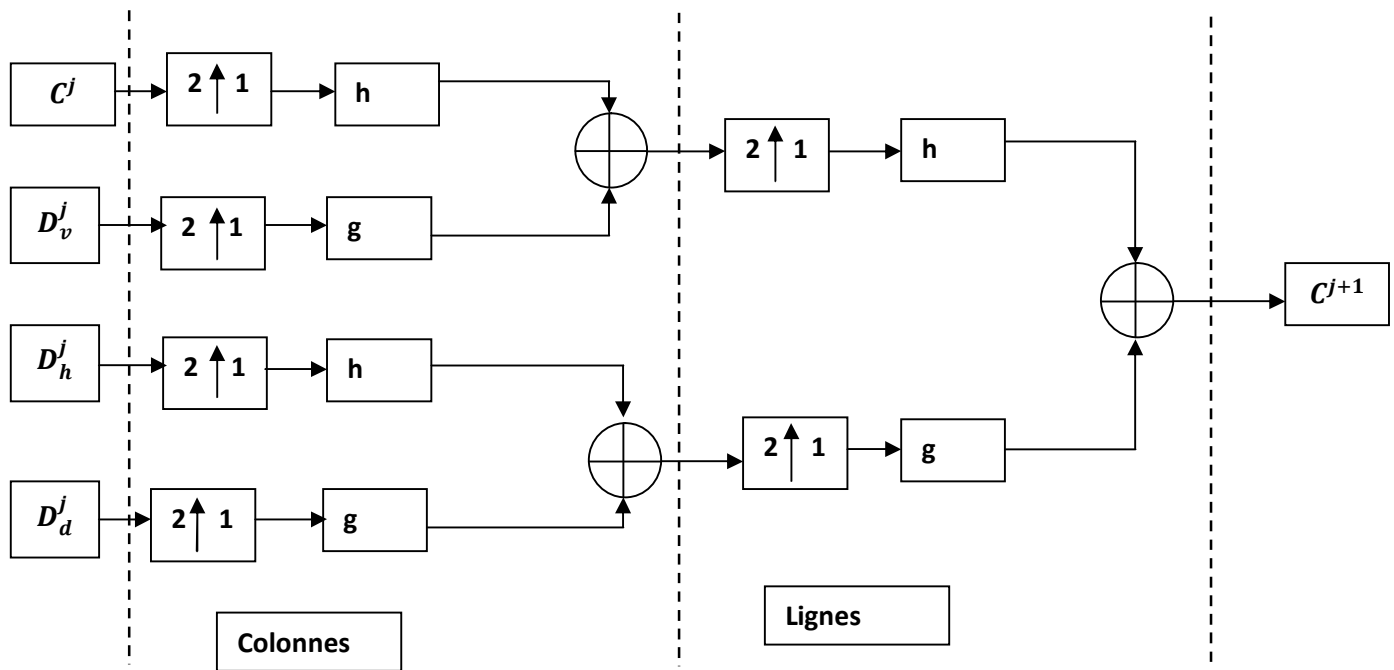


Figure II-7 : schéma de reconstruction

### Conclusion :

Nous avons entamé dans ce deuxième chapitre la théorie des ondelettes d'une manière générale et brève en commençant d'un petit rappel sur la transformée de Fourier, après la transformée d'ondelette continue (TOC) et la transformée d'ondelette discrète (TOD).

Ainsi nous avons parlé sur l'analyse multi-résolution, S.Mallat a relié les ondelettes orthogonales aux filtres utilisés dans le traitement du signal et on termine par la base biorthogonale.

## **Chapitre III**

# **Méthode adoptée**

**III-1-préambule :**

L'inconvénient majeur de la méthode de compression image par image réside dans l'espace mémoire occupé et le temps à prendre lors de compression, c'est-à-dire si nous avons trente images à compresser et chaque image prend un temps de compression égal à  $T_1$  alors le temps nécessaire pour compresser les trente images égales à trente fois  $T_1$ , et si chaque image a une taille égale à  $K$  octet donc nous avons besoin d'une zone mémoire qui est égale à trente fois  $K$  donc ça est coûteux.

Alors notre objectif est d'établir une méthode de compression d'une séquence d'images basée sur l'insertion spirale des images dans d'autres images, afin de gagner le temps et l'espace mémoire (stockage) tout en conservant une qualité visuelle acceptable, dans le but de transmission et d'un stockage à bas débit.

**III-2-Principe général de la méthode :**

Le principe général de la méthode consiste à insérer une image dans une séquence d'images météo, afin de compresser la séquence d'images avec l'image à insérer. On peut décrire ce principe comme suite :

a)-Chargement de l'image à insérer dans la séquence d'images météo.

b)-Insertion de l'image à insérer dans la séquence météo de telle façon à prendre toutes l'image à insérer, c'est-à-dire prendre les pixels de l'image et les mettre dans la position correspondante de la séquence météo. Cela peut être fait en convertissant l'ensemble des images (la séquence d'images météo et l'image à insérer) sous forme de vecteurs, après insertion on a fait l'opération inverse c'est-à-dire rendre le vecteur sous forme d'image pour la compression et décompression avec JPEG2000.

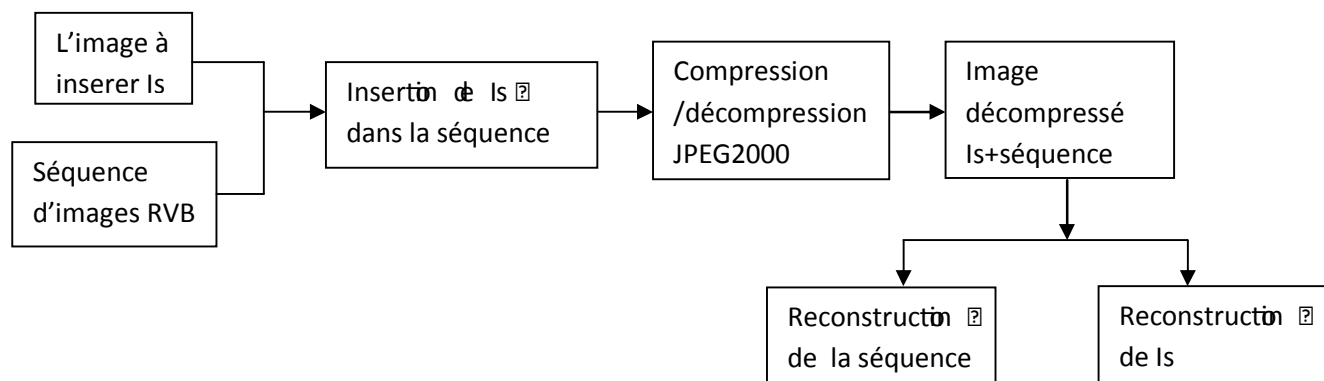
c)-compression et décompression avec JPEG2000 :

L'utilisation de JPEG2000 n'a pas été un choix arbitraire car dans le chapitre précédent on a parlé sur l'ondelette orthogonale et biorthogonale, et JPEG2000 utilise la décomposition en ondelettes pour la compression, ainsi cette norme est la norme la plus récente par rapport aux autres normes telle que JPEG. La raison de faire ce choix réside dans l'efficacité de cette norme par rapport aux autres normes.

d)-La restauration de la séquence d'images météo-sat avec insertion :

Après avoir compressé et décompressé avec JPEG2000, à la sortie on a récupéré la séquence météo-sat avec l'image à insérer et à partir de cette séquence météo-sat on a fait l'opération inverse, ce qui revient à dire qu'on a enlevé les pixels insérés dans la séquence météo-sat (les pixels qui se trouvent dans la position paire dans la séquence d'images) pour former une image, cette image est celle insérée au début. Ainsi on a restitué la séquence d'images météo-sat de la façon suivante : on a pris la moyenne de deux pixels voisins pour former le pixel paire enlevé de la séquence météo-sat.

Alors pour bien comprendre on a donné le schéma l'organigramme général de la méthode et on a détaillé chaque bloc.



**Figure III-1- Schéma bloc de la méthode adoptée**

Avant de passer aux illustrations, on a défini quelques paramètres :

L'image à insérer(Is) : C'est une image en niveau de gris de taille 512\*512, chaque pixel est codé sur 8 bits ce qui donne 256 niveaux de gris (exemple image LENA).

Pour la séquence d'images météo-sat (RVB) : C'est une séquence d'images formées de trois images, une image visible qui correspond à la palette Rouge, une image infrarouge qui correspond à la palette Bleu et la somme de ces deux images divisées par deux donne la palette Verte. Cette séquence est une image couleur (RVB) (exemple image RVB), les palettes sont de même caractéristique que l'image à insérer (taille est de 512\*512 et de 256 niveaux de gris).



a) Image LENA



b) image RVB

**Figure III-2- Les images utilisées dans la méthode adoptée****III-3-Insertion spirale des images :**

On a fait l'insertion d'une image dans une séquence d'images météo-sat d'une façon spirale. Tout d'abord on a intérêt à transformer l'image à insérer et la séquence d'images sous forme des vecteurs, on a fait cette transformation pour rendre l'insertion facile. Alors on a pris un tiers des pixels de vecteur de l'image à insérer et on les a insérés dans chaque palette de la séquence d'images dans la position paire d'une façon spirale. Ces palettes on les a transformées sous formes des vecteurs, l'organigramme de la figure III-3 montre la manière de transformation des images sous forme des vecteurs. Après avoir fait toutes ces transformations, on a passé à l'étape de l'insertion des pixels de  $I_{sv1}$  ( $I_{sv1}$  c'est le vecteur de l'image à insérer) dans différents vecteurs  $R1v$ ,  $V1v$  et  $B1v$  ( $R1v$ ,  $V1v$  et  $B1v$  ce sont respectivement les vecteurs des palettes Rouges, Verts et Bleus), le vecteur  $I_{sv1}$  est divisé par trois pour insérer chaque tiers dans une palette. On a fait l'insertion des pixels dans la position paire de chaque vecteur  $R1v$ ,  $V1v$  et  $B1v$ . Puisque nous avons limité l'insertion à un tiers de  $I_{sv1}$  dans chaque palette la région qui n'a pas été touché est appelé « région d'intérêt ».

Après avoir insérer les pixels dans la séquence d'images on a passé à la transformation inverse c'est-à-dire qu'on a converti les vecteurs de la séquence sous forme d'images, le but de cette transformation est de pouvoir compresser cette séquence. L'organigramme de la figure III-5 montre la méthode de transformation inverse c'est-à-dire de vecteur vers l'image.

Ces opérations ont illustré par les organigrammes suivant :

Dans cette première partie on va transformer l'image à insérer (Is) sous forme d'un vecteur (Isv1) d'une façon spirale,  $Isv1 = [a \ b \ c \ d]$  :

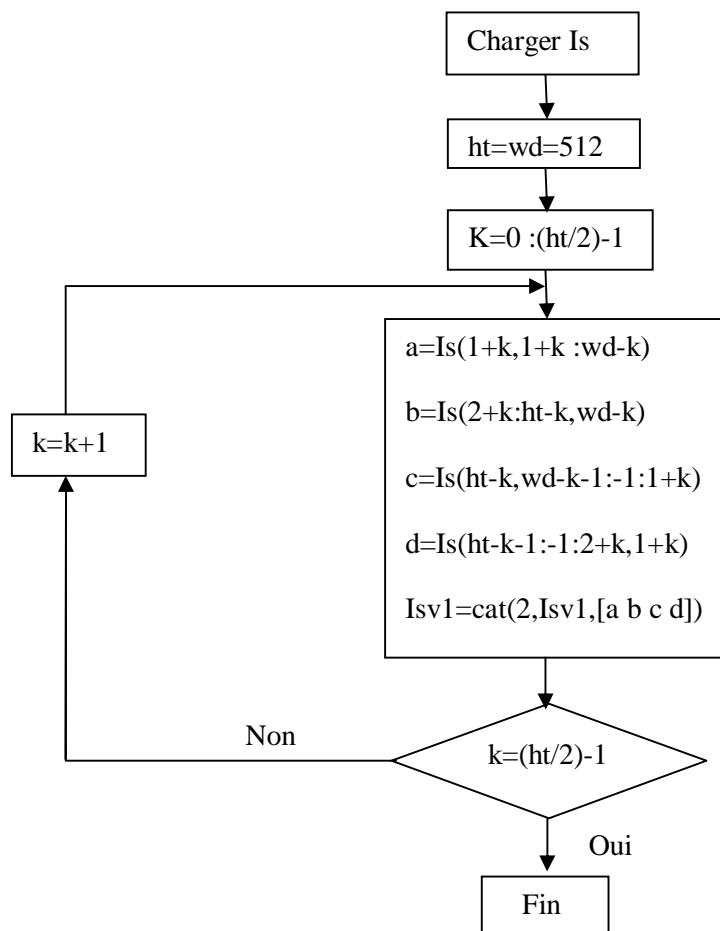
$ht = c$ 'est le nombre de ligne l'image.

$wd = c$ 'est le nombre de colonne l'image.

R1v : est le vecteur de la palette Rouge

V1v : est le vecteur de la palette Verte

B1v : est le vecteur de la palette Bleu

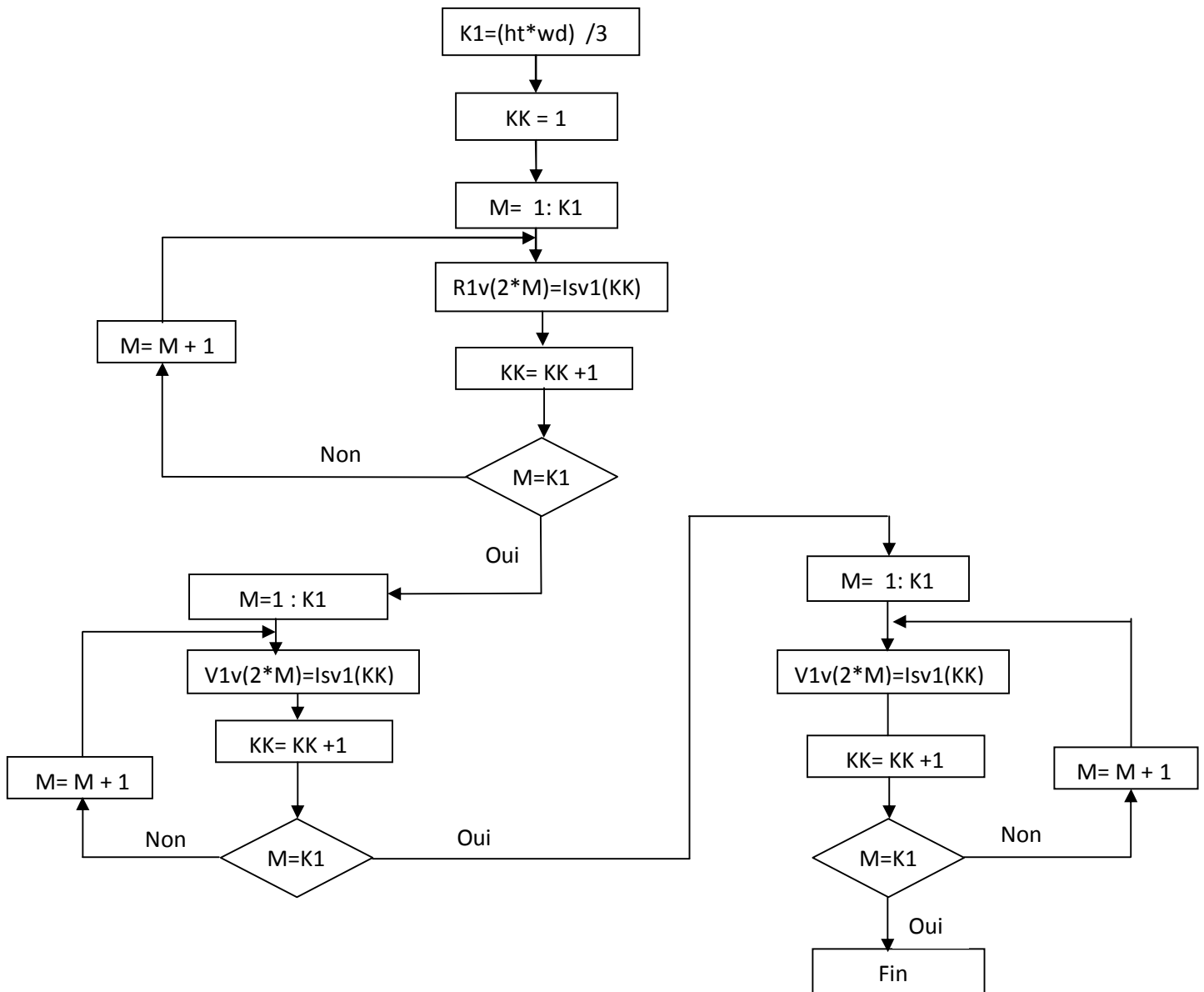


**Figure III-3 Organigramme de transformation de l'image (Is) en vecteur spirale (Isv1).**

Organiser chaque palette R1, V1 et B1 de la séquence RVB1 en vecteur spirale, et cela se fait de la même façon que l'organigramme de la Figure III-3 de l'image à insérer (exemple de image LENA).

L'insertion des pixels de vecteur à insérer (Isv1) dans les vecteurs R1v, V1v et B1v se fait de la manière suivante :

- a) Prendre (1/3) de Isv1.
- b) Insertion chaque (1/3) de Isv1 dans chaque vecteur R1v, V1v et B1v.
- c) L'insertion se fait dans la zone paire de chaque vecteur R1v, V1v et B1v.



**Figure III-4 Organigramme d'insertion des pixels de Isv1 dans les vecteurs R1v, V1v et B1v.**

Après avoir insérer toutes les pixels dans les vecteurs de la séquence d'images on a passé à l'étape de transformation des vecteurs sous formes d'images, l'organigramme de la figure suivante montre la méthode d'insertion.

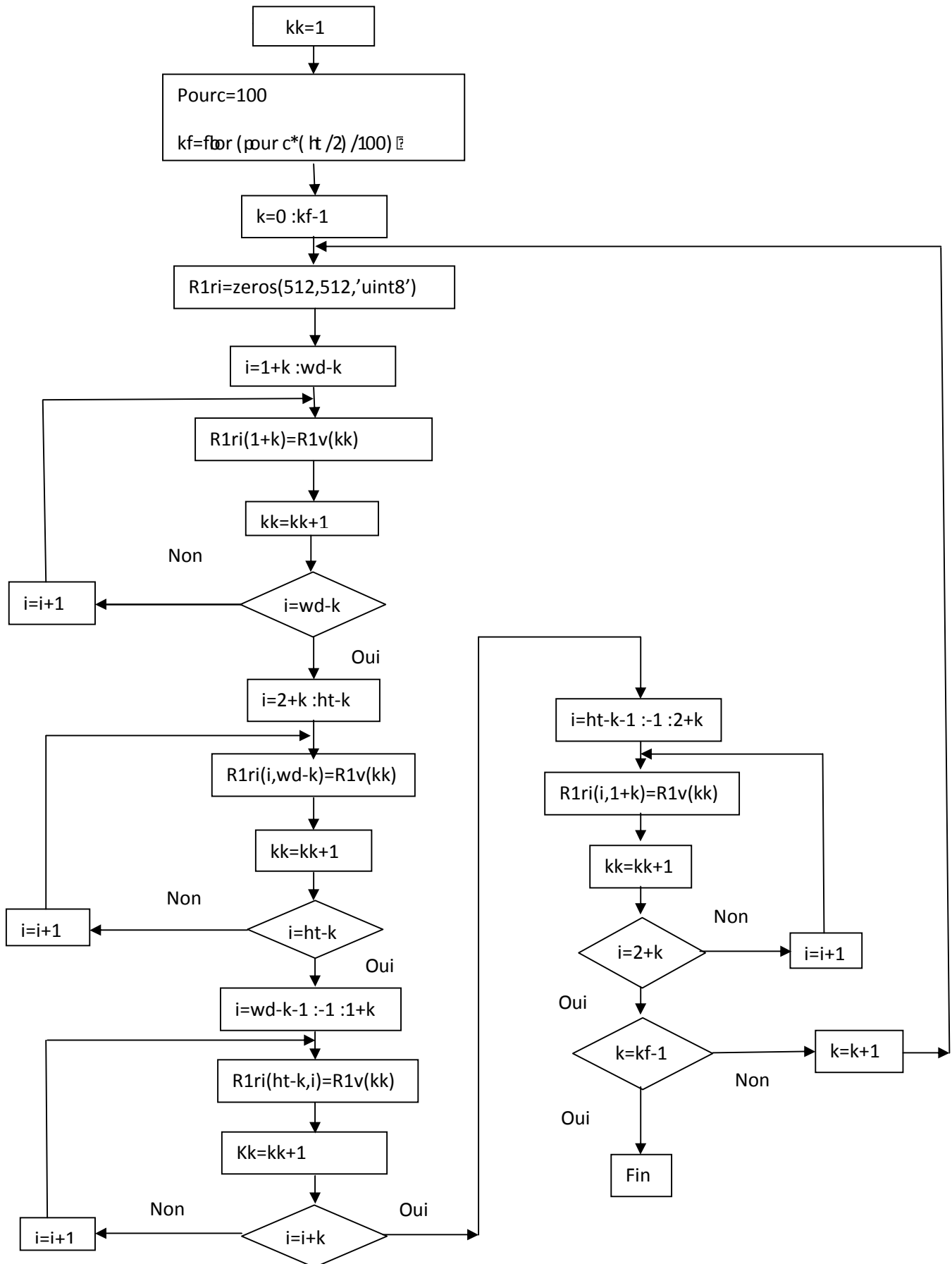
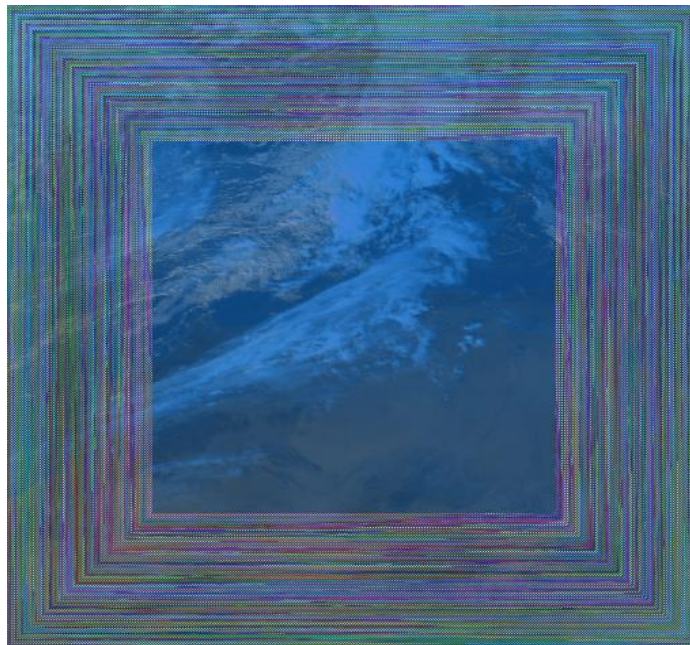


Figure III-5 organigramme de transformation de vecteurs sous forme des images

De la même procédure on va transformer V1v et B1v sous forme des images V1ri et B1ri respectivement afin de reconstruire l'image RVB1ri (l'image RVB1 avec insertion).

L'objectif de restituer RVB1ri est de la compresser avec JPEG2000 et à partir de l'image RVB1ridec (RVB1ridec est l'image RVB1ri compressée/décompressée avec JPEG2000) on a fait la reconstruction de l'image (LENA), avant d'obtenir cette image, on a transformé RVB1ridec en trois palettes nommées respectivement :R1dec, V1dec et B1dec et cela pour faciliter de les rendre sous forme des vecteurs nommées respectivement R1vdec, V1vdec et B1vdec dont l'objectif d'extraire les pixels paires de chaque vecteur et l'on insère dans un vecteur qui sera nommé Isv1. A partir de Isv1 on a reconstruit l'image insérée. Afin d'obtenir les vecteurs R1vdec, V1vdec et B1vdec on prend la même procédure que l'organigramme de la figure (III-3).

L'image de la figure suivante montre la séquence RVB avec insertion de l'image LENA.



**Figure III-6 image RVB avec insertion**

### **III-4-Compression avec JPEG2000 :**

#### **III-4-1-Introduction :**

JPEG2000 est le standard qui a succédé le JPEG. L'avantage de JPEG2000 par rapport aux autres normes de compression est qu'il a procédé par dégradation sélective de certaines zones moins stratégiques de l'image.

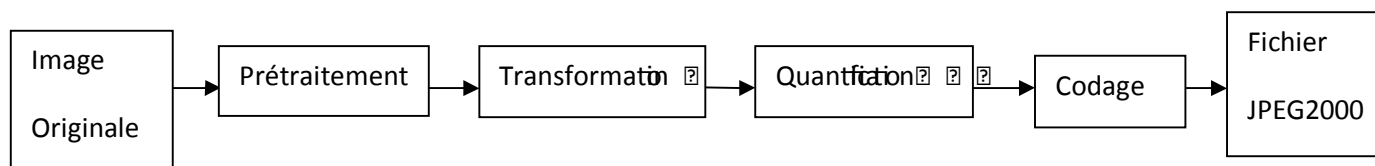
Ce nouveau standard a pour objectif d'offrir une nouvelle fonctionnalité permettant de répondre à une demande croissante à savoir :

- ∅ Obtenir des performances de compression supérieures à son prédécesseur JPEG, notamment pour des débits très faibles.
- ∅ Permettre d'organiser le fichier compressé de plusieurs manières, notamment en fonction de la résolution désirée ou de la qualité de reconstruction.
- ∅ Fournir la possibilité de coder des parties d'une image avec une qualité supérieure à d'autres parties.

#### III-4-2-Chaines de codage et de décodage JPEG2000 :

La chaîne de codage de JPEG-2000 commence par un pré-traitement de l'image originale. Ce prétraitement permet de changer la représentation de l'image (facultatif), tel qu'une transformation de couleur RGB à  $Y C_r C_b$ . Ensuite, on découpe l'image en « Tuile ».

Chaque « Tuile » subit une transformée en ondelettes. Une quantification est par la suite appliquée sur chaque sous bande obtenue suite à la transformée en ondelettes (TO). Les coefficients obtenus sont traités par le codeur entropique qui permet d'obtenir les données compressées. Ces données sont finalement mises en forme pour aboutir à un fichier JPEG-2000.



**Figure III-7-** Diagramme de la chaine de l'Algorithme JPEG2000.

#### III-4-3-Prétraitement et transformée de couleur :

Cette transformée est optionnelle et quand l'image est en espace de couleur RVB on la convertit généralement dans un espace luminance/chrominance  $Y C_r C_b$  qui permet de coder les couleurs avec un nombre de bits inférieur et puis encore chaque image est découpée en un ensemble de taille carrée (64\*64) ou (128\*128) .

Cette transformée peut être réversible et alors les données restent entières (ce qui autorise un éventuel codage sans perte), où irréversible et les données sont alors représentées en virgule flottante (ce qui implique déjà des pertes lors du codage).

#### III-4-4-Transformée en ondelette discrète :

C'est ici que l'algorithme JPEG 2000 est radicalement différent de celui de JPEG. Alors que le JPEG utilise une transformée de Fourier discrète, par contre JPEG 2000 utilise les ondelettes.

L'image subit une série de décomposition grâce à des filtres. Pour donner une grosse idée du travail effectué, on peut dire que l'image est divisée d'abord en « tuiles » de taille fixe. Chaque tuile est rééchantillonnée de manière à ce que la longueur et la largeur soient divisées par 2. Les informations perdues dans l'opération sont enregistrées sous forme de coefficients d'ondelettes. On recommence ensuite l'opération avec la nouvelle image et ainsi de suite tant qu'il reste des pixels. La méthode est légèrement différente selon la compression utilisée avec ou sans perte (ce n'est pas la même ondelette qui est utilisée).

#### III-4-5-Quantification des sous bandes :

Les sous-bandes de résolutions supérieures possèdent un contenu relativement faible alors que les sous-bandes de basses fréquences sont beaucoup plus riches. JPEG2000 adopte une quantification linéaire pour chaque sous-bande. Le pas de quantification utilisé est cependant beaucoup plus faible pour les sous-bandes de basses fréquences.

La quantification dans la norme JPEG2000 peut s'effectuer de deux manières :

- Ø Soit les coefficients d'ondelettes sont quantifiés avant le codage entropique des blocs.
- Ø Soit les coefficients ne sont pas quantifiés et sont directement compressés bloc par bloc.

Dans ce dernier cas, des marqueurs correspondant à des points de troncature sont insérés dans le code compressé. Ces points de troncature répondent à des débits demandés en n'utilisant qu'une partie de l'information. Ces marqueurs serviront à la segmentation du code et à la répartition par ordre d'importance dans le fichier binaire. Pour obtenir la taille du fichier désiré, il suffira de tronquer la partie indésirable. Pour un codage avec pertes, la précision sur les coefficients est réduite par quantification. Les coefficients de faible amplitude sont mis à zéro et ainsi fixent le taux de compression minimum.

**III-4-6-Codage des sous-bandes :**

Le codage de chaque sous-bande s'effectue par plan de bits, cette opération permet d'obtenir une « scalabilité » du fichier généré, car l'information importante (i.e. les bits de poids forts) sera codée dans un premier temps alors que les détails (i.e. les bits de poids faibles) sont codés après.

**III-4-7-Codage par régions d'intérêt :**

L'algorithme JPEG2000 permet également de coder différentes régions de l'image avec des qualités différentes.

Cette fonctionnalité est mise en œuvre en augmentant le nombre de bits de poids forts des coefficients appartenant à une zone d'intérêt. De part l'orientation du codage, ces coefficients seront alors codés en priorité. Il existe deux manières différentes de coder une région d'intérêt :

- ∅ Par insertion d'un masque de forme dans le fichier compressé.
- ∅ Par doublement du nombre de bits appartenant aux zones d'intérêts.

**III-4-8-Codage entropique :**

Les coefficients quantifiés sont préalablement groupés, dans chaque sous-bande, en blocs rectangulaires (code-blocks) typiquement de taille 64x64 ou 32x32. Puis chaque *code-block* est codé plan de bits par plan de bits, en commençant par les bits de poids le plus fort. En réalité, dans chaque plan de bits, les bits sont d'abord séparés en trois groupes, en fonction de leur voisinage, puis codés en trois passes (coding passes) successives. Il est important de constater ici que le bit Stream obtenu après un tel type de codage peut être tronqué à la fin de chacune des coding passes ( il y a donc trois points de troncature possible par plan de bits) qui correspondent alors à des qualités progressives de chaque code-block. Cette propriété est la base de l'algorithme EBCOT et elle est largement exploitée par tous les types d'allocations de débits tels que celui présenté ci-dessous.

- ✓ Cette technique de codage est effectuée en trois étapes :
- ∅ Une étape qui code les bits non nuls isolés.
- ∅ Une étape qui code les bits dont le bit paire est non nul.

∅ Une étape qui code le reste des bits non nuls.

### **III-5-Reconstruction de l'image insérée et la séquence météoats :**

Le paragraphe précédent a donné une généralité sur la compression avec JPEG2000. Donc après avoir compressé et décompressé la séquence d'images météoats, on a passé dans ce paragraphe à l'étape de reconstitution de l'image insérée et la séquence d'images météoats.

Premièrement on a restitué l'image insérée sous forme d'un vecteur à partir de la séquence d'images transformée sous forme des vecteurs c'est-à-dire un vecteur pour chaque palette, la procédure de transformation sous forme d'un vecteur s'est fait de la même façon que l'organigramme de la figure III-3. Après on est passé à la récupération des pixels insérés dans la position paire de chaque vecteur de la séquence d'images, ces vecteurs sont nommés R1vdec, V1vdec et B1vdec respectivement aux palettes Rouge, Vert et bleu. On a appelé le vecteur récupéré Isv1, la procédure de ces opérations est donnée par l'organigramme de la figure (III-8) ci-dessous :

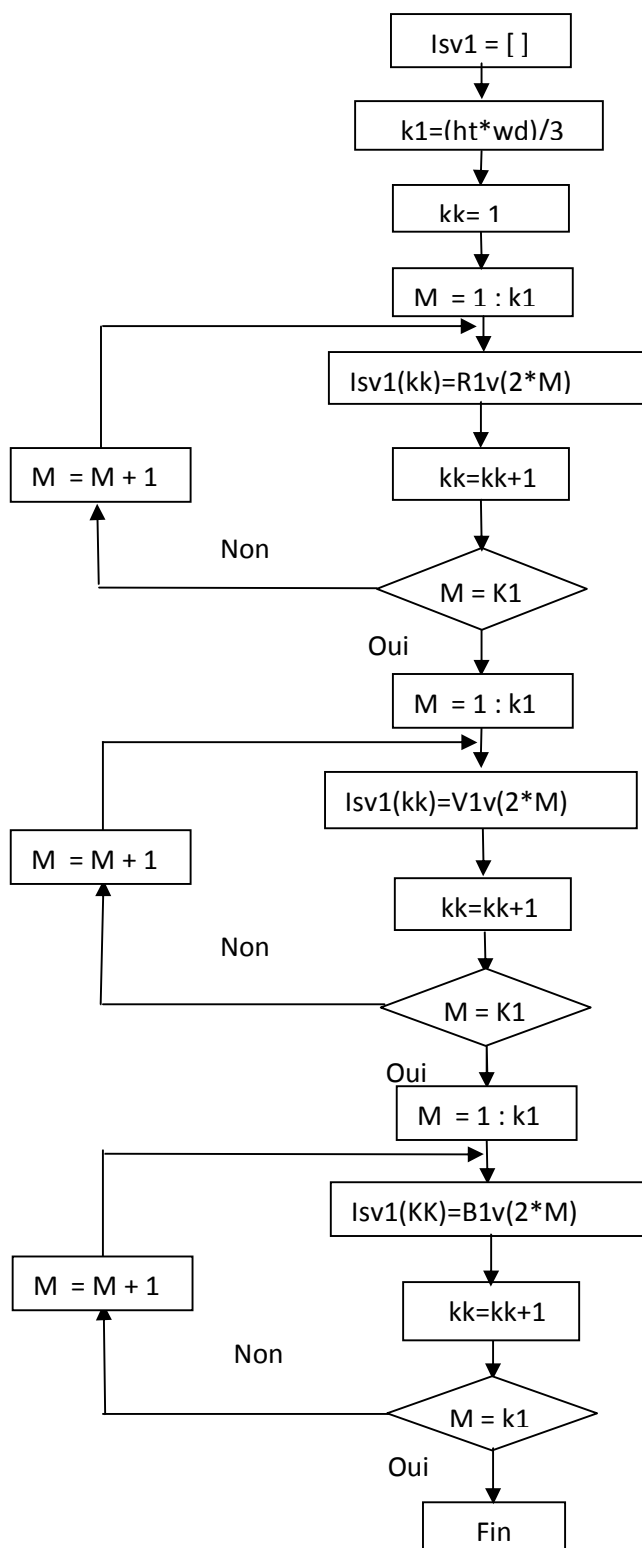


Figure III-8 organigramme de reconstruction de Isv1

Après on est passé à l'étape de restitution de l'image insérée à partir de vecteur  $I_{sv1}$  formé précédemment, on a illustré cette étape de reconstruction par l'organigramme ci-dessous :

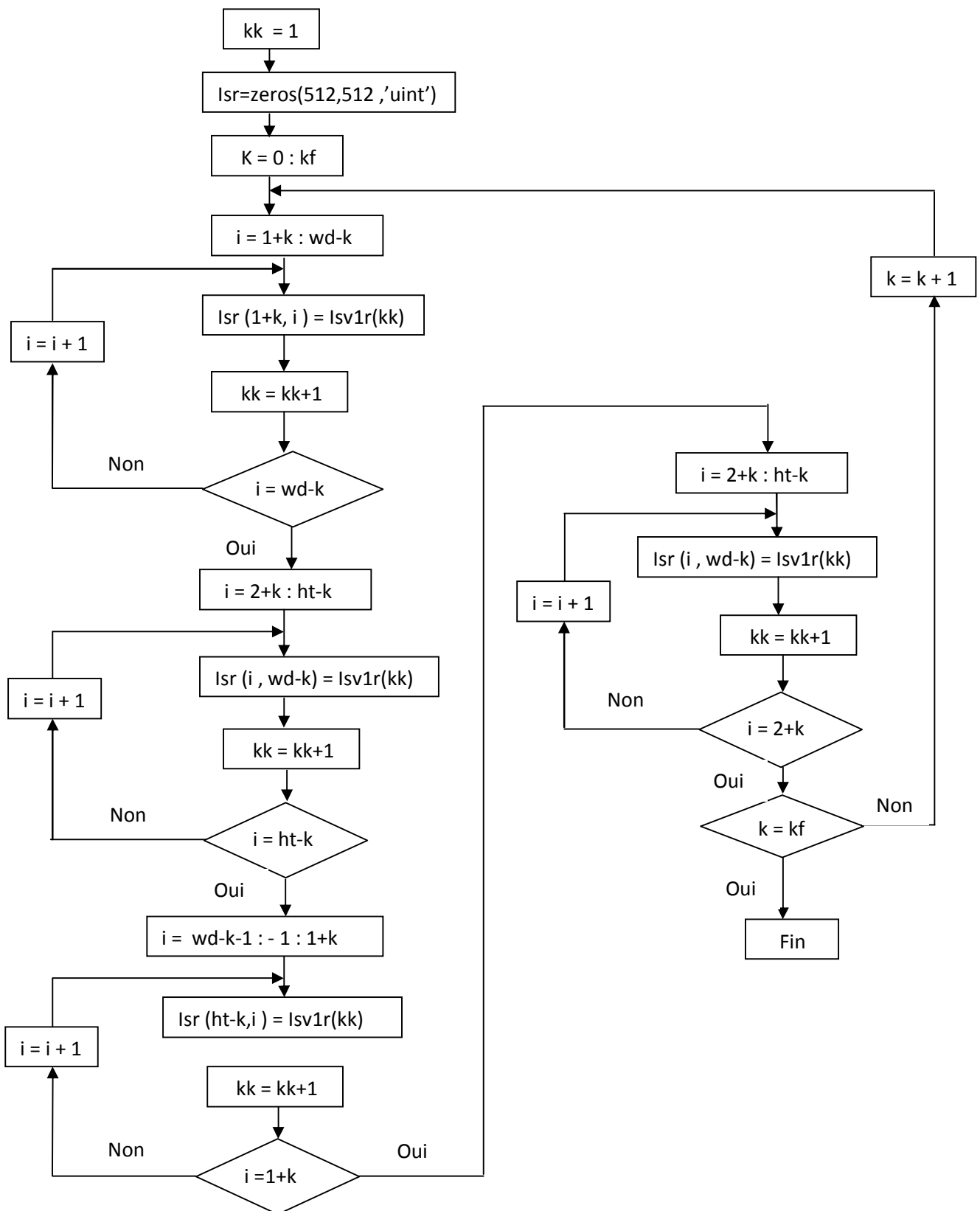


Figure III-9 organigramme de l'image reconstruite



**Figure III-10 : image LENA reconstruite**

De même on a reconstitué la séquence d'images mais avant de faire cette étape il a fallu récupérer les pixels enlevés de chaque vecteur  $R1vdec$ ,  $V1vdec$  et  $B1vdec$ , alors pour la restauration de ces pixels on a pris la moyenne de deux pixels voisins pour former le pixel de la position paire désinsérée (exemple : pour former le pixel de la position deux, on a pris la moyenne de deux pixels de la position trois et un). Toutes ces transformations ont été illustrées par l'organigramme ci-dessous :

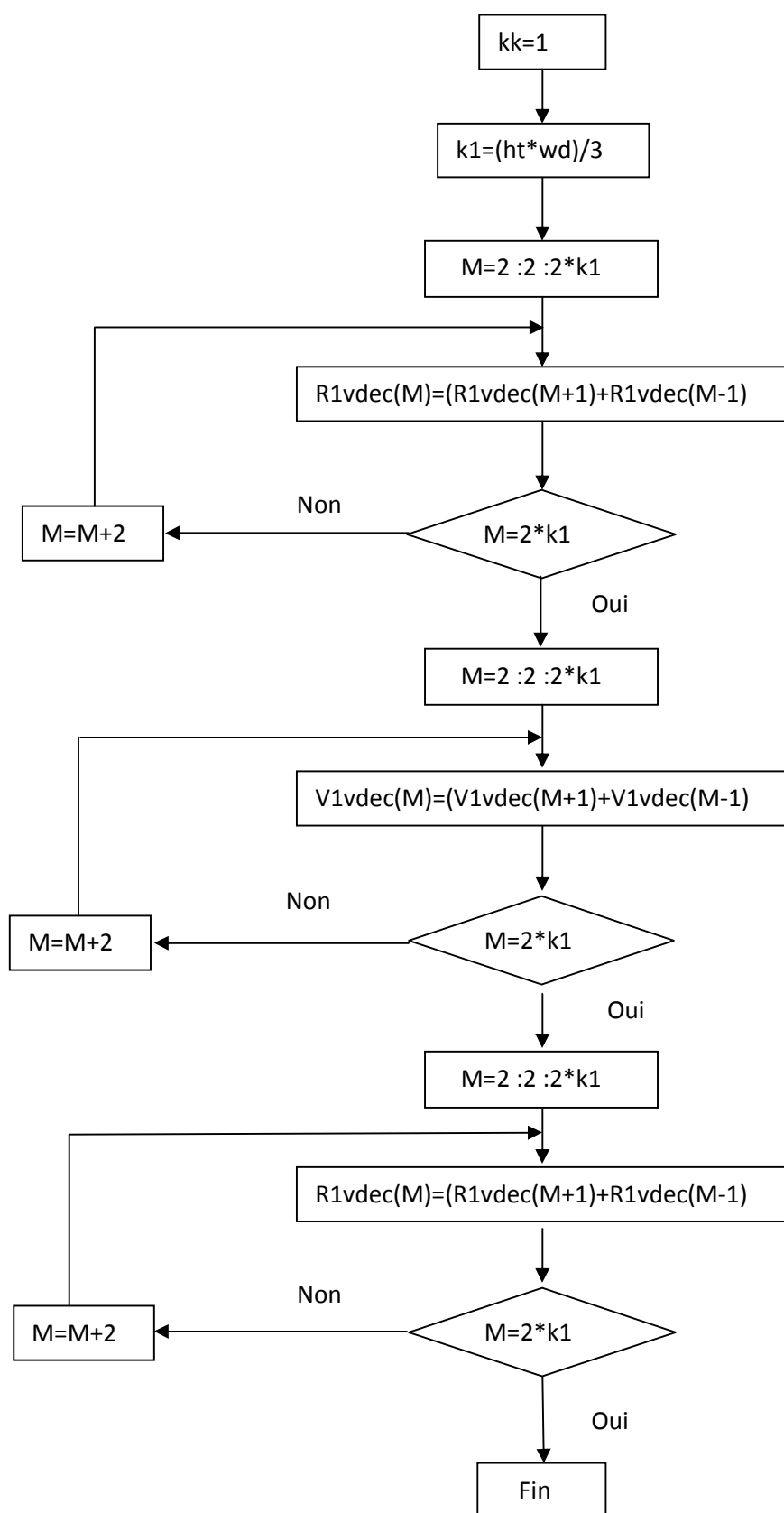
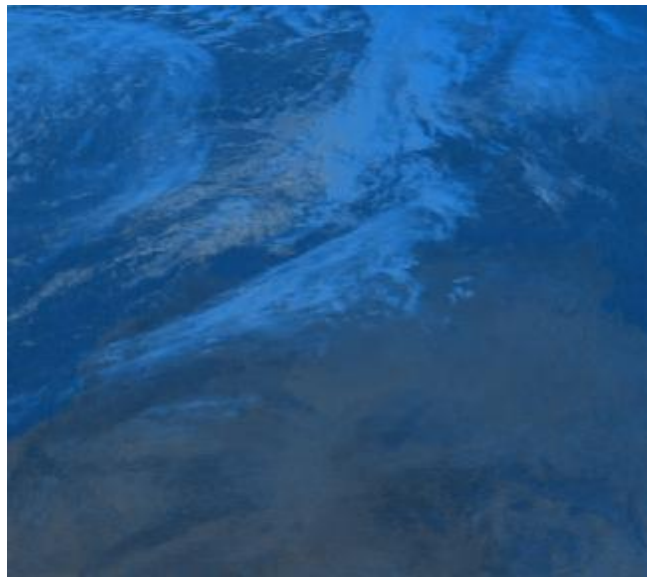


Figure III-11 organigramme de reconstruction des vecteurs R1vdec, V1vdec et B1vdec

Après avoir récupéré les pixels désinsérés on est passé à l'étape de transformation des vecteurs R1vdec, V1vdec et B1vdec sous forme d'images, cette transformation est très importante car à partir de ces images restituées on a pu faire une comparaison entre l'image originale avant insertion et l'image restituée. La restitution de la séquence d'images se fait de la même procédure que l'image à insérer comme le montre l'organigramme de la figure III-9.

La figure suivante donne la séquence d'image reconstruite :



**Figure III-12 : La séquence d'image reconstruite**

### **Conclusion :**

Ce chapitre nous a permis d'illustrer notre méthode adoptée, en commençant par un schéma global qui a décrit les différentes étapes prises ainsi on a donné quelques organigrammes dont le but de comprendre les différentes opérations illustrées.

Afin de tester l'efficacité de notre méthode on a donné les résultats de notre méthode dans le chapitre suivant.

## **Chapitre IV**

# **Tests et résultats**

**IV-1-Préambule :**

Dans ce chapitre, nous avons présenté des résultats sur l'insertion spirale des images, ainsi nous avons utilisé des images de résolution 256\*256 à niveau de gris et chaque pixels est codé sur 8 bits, les images traitées dans ce chapitre sont des images météoats et l'image LENA.

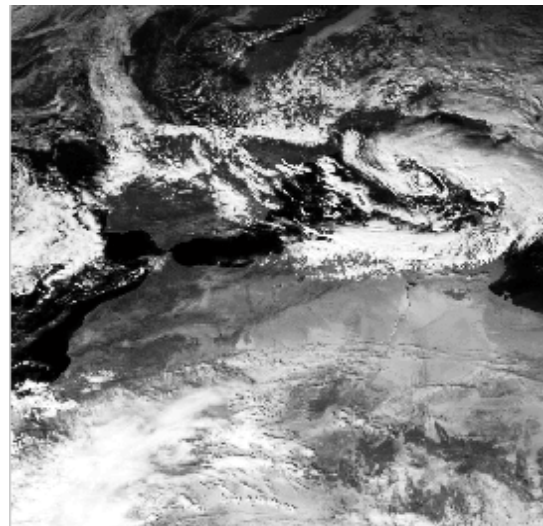
L'algorithme d'insertion a été illustrée grâce au langage MALAB 7.40 (R2007a), c'est un logiciel simple et très efficace, optimisé pour le traitement des matrices, d'où son nom « MATRICE LABORATORY ».

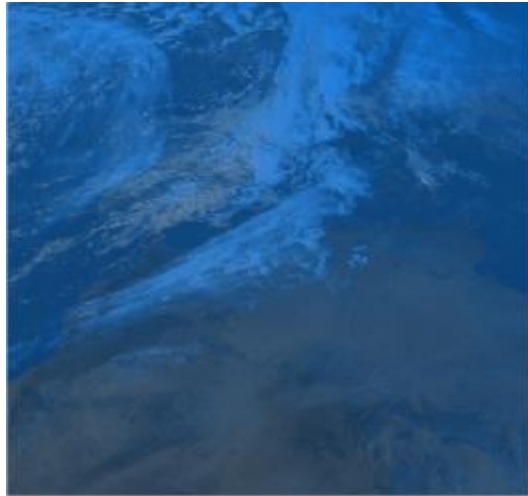
Afin d'évaluer les résultats obtenus, nous avons considéré les critères d'évaluation, tel que le MSE et le PSNR ( **Peak Signal Noise Ratio** ), ce rapport est très utilisé en traitement d'images, il montre la distorsion de l'image traitée par rapport à celle originale.

Cela permettra d'évaluer les performances de la méthode utilisée en fonction du nombre de bits par pixel (bpp), qui nous informe directement sur le taux de compression.

**IV-2- Résultats et interprétations :**

Pour évaluer les différents résultats obtenus, nous avons utilisé plusieurs images tests qui sont représentées dans la figure ci-dessous :

**a) Image LENA****b) Image Visible**



c) La séquence d'images météosats (Seq1)

d) La séquence d'images météosats (Seq2)

**Figure IV-1- Les images originales utilisées dans notre test.**

Pour les séquences d'images météosats nous avons utilisé des images visibles et infrarouges qui ont capté une partie de l'Europe et l'Afrique du nord dont le temps est différent tel que pour Seq1 l'image visible et infrarouge (J6V14,J6I14) ont été capté en 1996 à 14 heure et pour Seq2 l'image visible et infrarouge (J6V15,J6I15) ont été capté en 1996 à 15 heure.

#### **IV-2-1- Résultat de la compression avec le JPEG2000 :**

Nous avons appliqué le JPEG2000 sur les images tests de la figure IV-1, ce sont des images de résolution 256\*256 codées sur 256 niveaux de gris.

L'insertion de l'image lena se fait suivant le pourcentage (Pc) choisi sur la séquence d'images météosats, deux cas se présentent :

- Ø Premier cas : l'insertion se fait dans les 66.66% de la séquence.
- Ø Deuxième cas : l'insertion se fait dans les 33.33% de la séquence.

Les tableaux suivants représentent les résultats obtenus en estimant le rapport signal sur bruit de crête (PSNR) entre l'image originale et celle compressée, ainsi que le MSE en fonction de nombre de bits par pixel (bpp), le bpp varie de 0.01 : 0.02 : 0.22.

**IV-2-1-1- Les résultats obtenus pour  $P_c = 66.66\%$  :****1) Pour l'image LENA :**

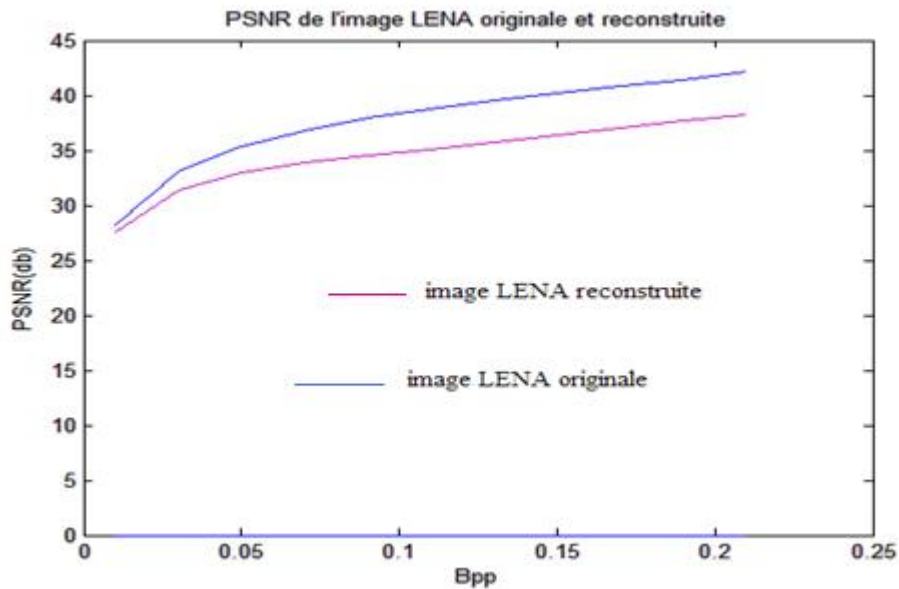
Le tableau suivant indique la variation de PSNR et MSE en fonction de nombre de bits par pixel (bpp), avec bpp varie 0.01 :0.02 :0.22.

bpp	MSE lena avant insertion	PSNR (db) lena avant insertion	MSE lena après insertion	PSNR (db) lena après insertion	Taux en %
0.01	77.2942	28.2499	115.378	27.5096	99.88
0.03	32.3397	33.0334	46.6052	31.4465	99.62
0.05	16.1632	35.3061	32.9607	32.9508	99.38
0.07	13.479	36.8342	26.9688	33.8223	99.12
0.09	10.3835	37.6973	22.7565	34.5597	98.88
0.11	8.61	38.7808	19.9513	35.1311	98.62
0.13	7.3072	39.4933	17.177	35.7813	98.38
0.15	6.2452	40.1754	14.9628	36.3807	98.12
0.17	5.3186	40.8743	12.791	37.0618	97.88
0.19	4.699	41.4107	11.1489	37.6585	97.62
0.22	3.9863	42.1251	9.7398	38.2453	97.38

**Tableau IV-1-** Résultats obtenus de la compression avec le JPEG2000.

Temps d'exécution de LENA originale:  $t=2.8856s$

Temps d'exécution de LENA reconstruite :  $t=3.0253s$



**Figure IV-2- Variation de PSNR en fonction de bpp pour LENA.**

## 2) Pour l'image Visible:

Le tableau suivant indique la variation de PSNR et MSE de l'image Visible en fonction de nombre de bits par pixel (bpp), avec bpp varie de 0.01 : 0.02 : 0.22.

bpp	MSE de l'image visible avant insertion	PSNR de l'image visible avant insertion	MSE de l'image visible après insertion	PSNR de l'image visible après insertion	Taux en %
0.01	97.2442	28.2499	115.378	27.5096	99.88
0.03	32.3397	33.0334	46.6052	31.4465	99.62
0.05	19.1632	35.3061	32.9607	32.9508	99.38
0.07	13.4779	36.8342	26.9681	33.8223	99.12
0.09	10.3835	37.9673	22.7567	34.5597	98.88
0.11	8.61	38.7808	19.9513	35.1311	98.62
0.13	7.3072	39.4933	17.177	35.7813	98.38
0.15	6.2452	40.1554	14.9628	36.3807	98.12
0.17	5.3168	40.8743	12.791	37.0618	97.88
0.19	4.699	41.4107	11.1489	37.6585	97.62
0.22	3.9863	42.1251	9.7398	38.2453	97.38

**Tableau IV-2- Résultats obtenus de la compression avec le JPEG2000.**

Temps d'exécution de l'image visible originale :  $t=2.9652s$

Temps d'exécution de l'image visible reconstruite :  $t=3.1795s$

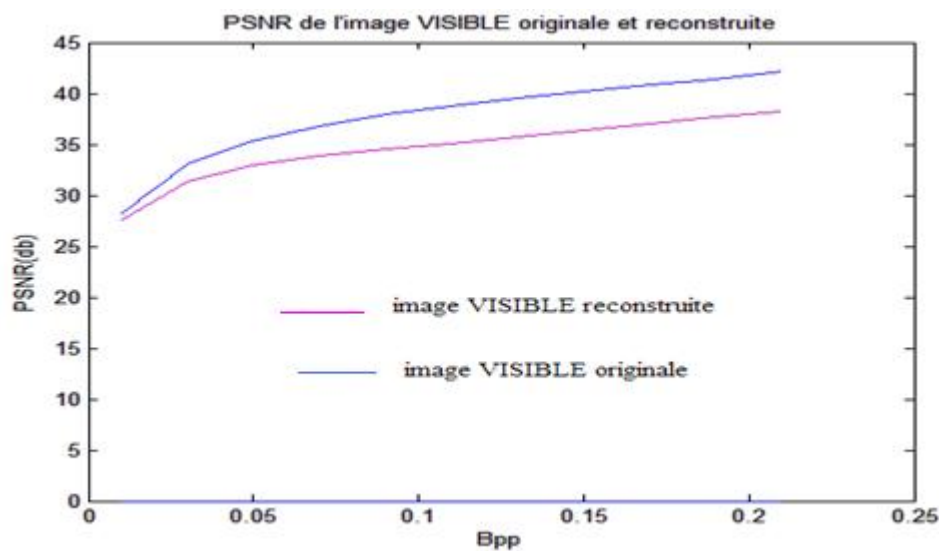


Figure IV-3- Variation de PSNR en fonction de bpp pour Visible.

### 3) Pour la Séquence d'image météosats (Seq1) :

Le tableau suivant indique la variation de PSNR et MSE de la séquence d'image météosats en fonction de nombre de bits par pixel (bpp), avec bpp varie de 0.01 : 0.02 : 0.22.

bpp	MSE Seq1 originale	PSNR (db) Seq1 originale	MSE Seq1 reconstruite	PSNR(db) Seq1 reconstruite	Taux en %
0.01	9.6421	38.2891	11.694	37.4512	99.88
0.03	4.1283	41.9.73	5.5928	40.6545	99.62
0.05	2.4524	44.2349	3.7739	42.3629	99.38
0.07	1.936	45.2618	2.875	40.5444	99.12
0.09	1.298	46.9981	2.3945	44.3386	98.88
0.11	1.2017	47.3328	2.1417	44.8232	98.62
0.13	1.0503	47.9178	1.9746	45.5911	98.38
0.15	0.8445	48.8648	1.3144	46.9435	98.12
0.17	0.6114	50.2678	1.1393	47.5643	97.88
0.19	0.5843	50.4642	0.9728	48.2505	97.62
0.22	0.4858	51.2677	0.7497	49.3822	97.38

Tableau IV-3- Résultats obtenus de la compression avec le JPEG2000.

Temps d'exécution de Seq1 originale :  $t=6.2694s$

Temps d'exécution de Seq1 reconstruite :  $t=6.5397s$

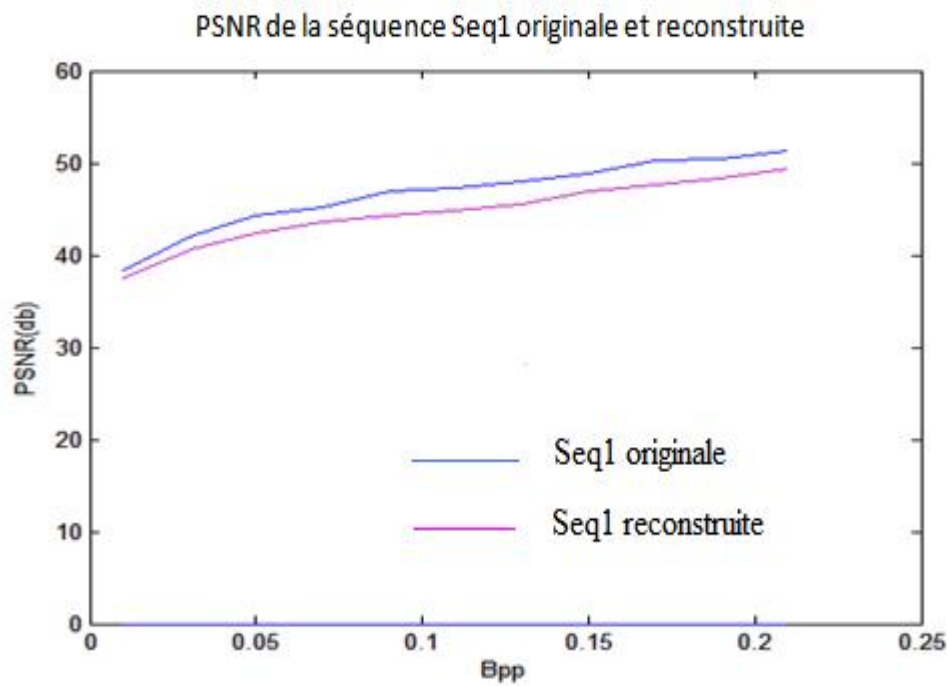


Figure : IV-4- Variation de PSNR en fonction de bpp pour la Seq.

IV-2-1-2- Quelques figures obtenus pour  $P_c = 66.66\%$  :

## 1) Pour l'image LENA :



a) Image originale compressée  
bpp=0.11  
temps d'exécution=0.554 s



b) Image reconstruite compressée  
bpp=0.11  
temps d'exécution=0.569 s



c) Image originale compressée  
bpp=0.22  
temps d'exécution=0.593 s



d) Image reconstruite compressée  
bpp=0.22  
temps d'exécution=0.633 s



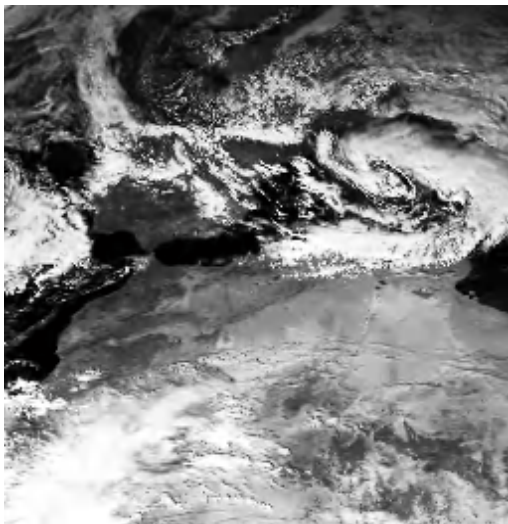
e) Image reconstruite  $\text{bpp}=0.11$



f) Image reconstruite  $\text{bpp}=0.22$

**Figure IV-5-**Résultats obtenus de la compression avec JPEG2000 sur l'image LENA.

**2) Pour l'image Visible :**



a) Image originale compressée

$\text{bpp}=0.11$

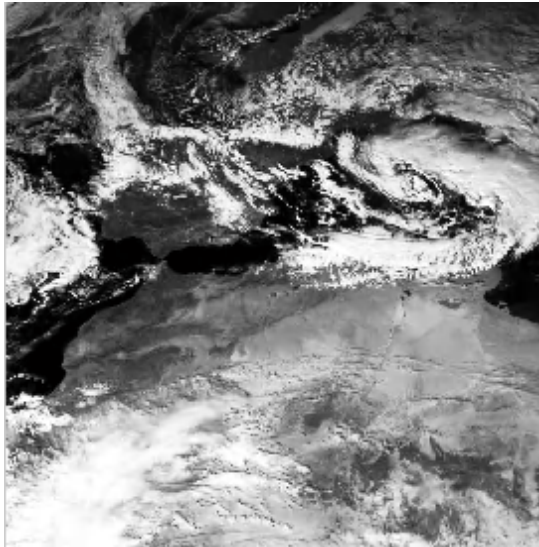
temps d'exécution=0.563 s



b) Image reconstruite compressée

$\text{bpp}=0.11$

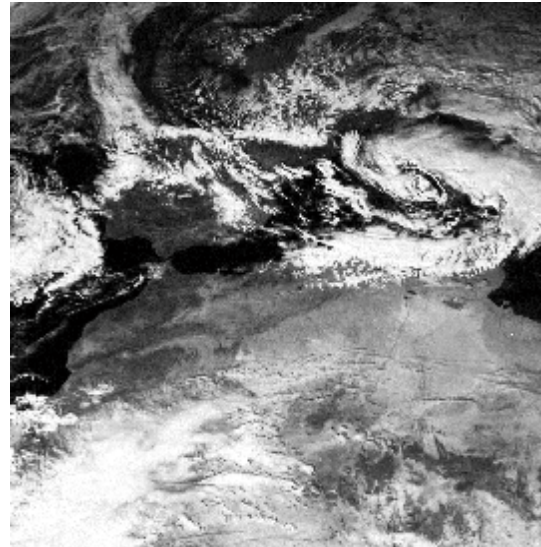
temps d'exécution=0.582 s



**c) Image originale compressée**

**bpp=0.22**

**temps d'exécution=0.551s**



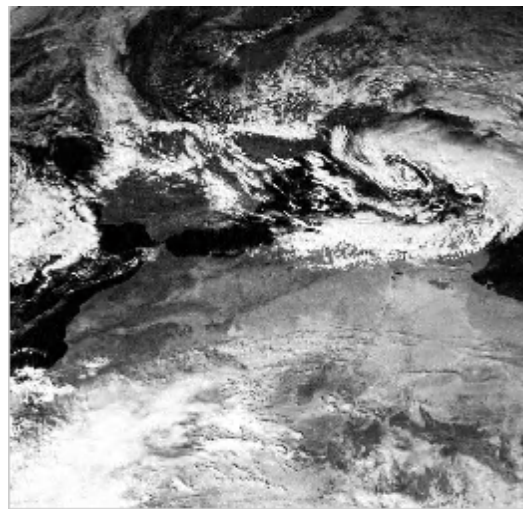
**d) Image reconstruite compressée**

**bpp=0.22**

**temps d'exécution=0.602 s**



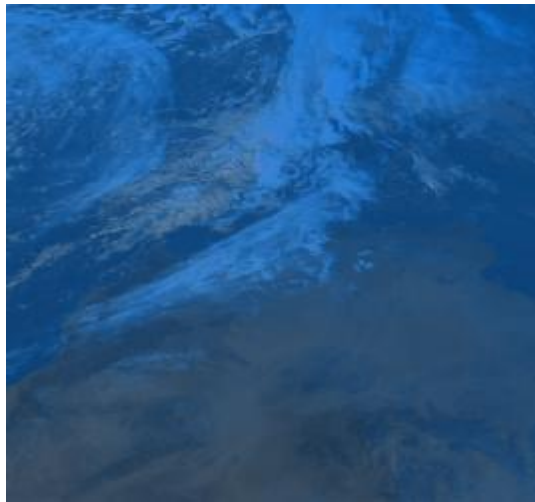
**e) Image reconstruite bpp=0.11**



**f) Image reconstruite bpp=0.22**

**Figure IV-6-Résultats obtenus de la compression avec JPEG2000 sur l'image Visible.**

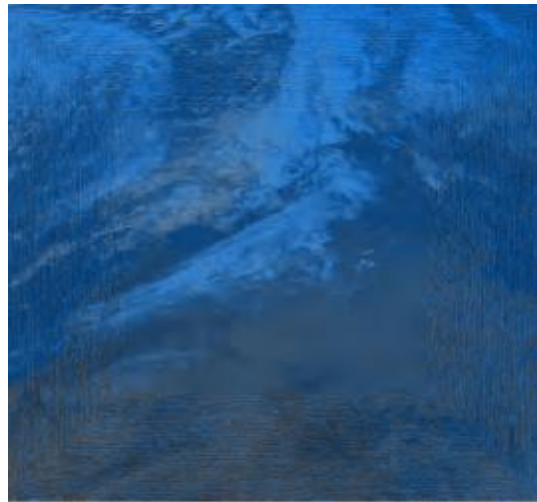
3) Pour la séquence :



a) Seq originale compressée

**bpp=0.11**

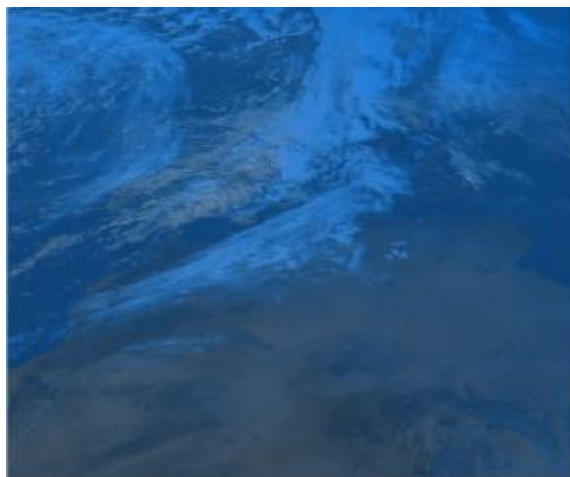
**temps d'exécution=1.118 s**



b) Seq reconstruite compressée

**bpp=0.11**

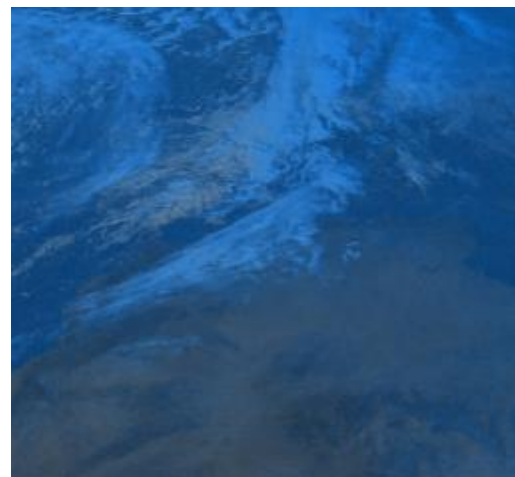
**temps d'exécution=1.051 s**



c) Seq originale compressée

**bpp=0.22**

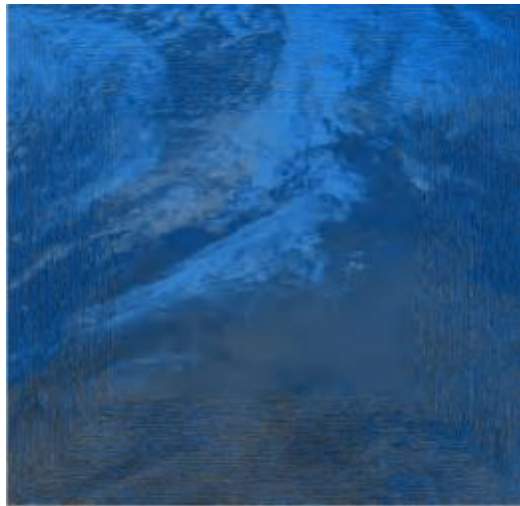
**temps d'exécution=1.312 s**



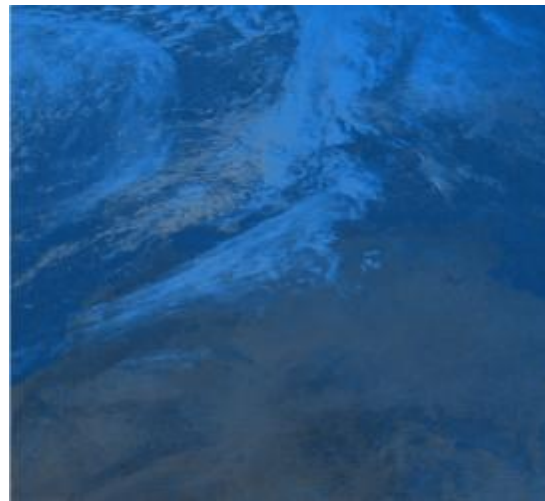
d) Seq reconstruite compressée

**bpp=0.22**

**temps d'exécution=1.112 s**



e) Seq reconstruite  $\text{bpp}=0.11$



f) Seq reconstruite  $\text{bpp}=0.22$

**Figure IV-7**-Résultats obtenus de la compression avec JPEG2000 sur la séquence.

**IV-2-1-3- Les résultats obtenus pour  $P_c = 33.33\%$  :****1) Pour l'image LENA :**

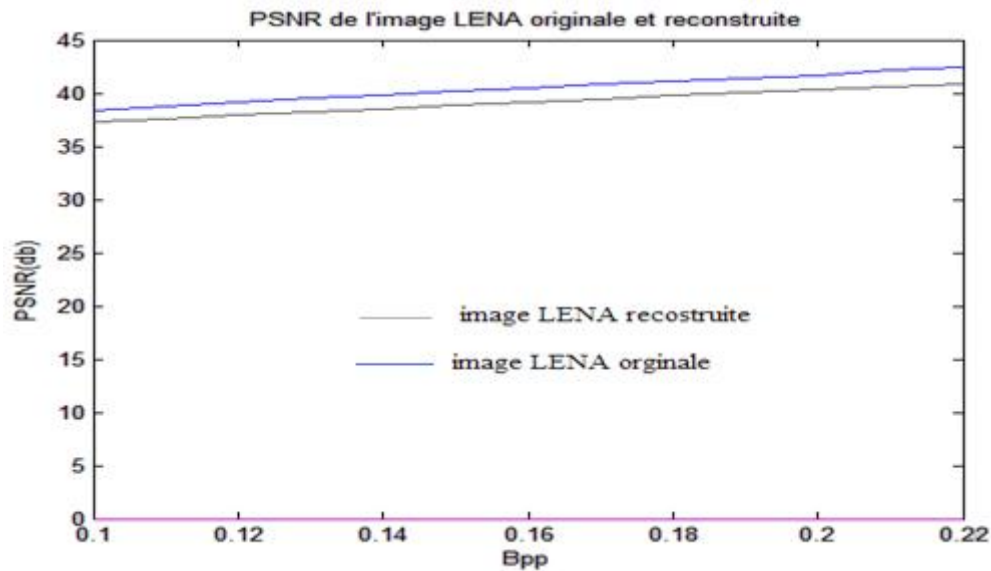
Le tableau suivant indique la variation de PSNR et MSE de la même image LENA en fonction de nombre de bits par pixel (bpp), en faisant varier le Bpp de 0.1 :0.01 :0.22.

bpp	MSE de l'image LENA avant insertion	PSNR (db) de l'image LENA avant insertion	MSE de l'image LENA après insertion	PSNR (db) de l'image LENA après insertion	Taux en %
0.1	9.3668	38.4149	12.1635	37.2802	98.75
0.11	8.61	38.7808	11.2562	37.6169	98.62
0.12	7.9424	39.1313	10.4426	37.9427	98.50
0.13	7.3072	39.4933	9.7031	38.2617	98.38
0.14	6.7363	39.8466	9.1171	38.5322	98.25
0.15	6.2452	40.1754	8.433	38.871	98.2
0.16	5.7812	40.5106	7.9081	39.1501	98.00
0.17	5.3168	40.8743	7.4174	39.4283	97.88
0.18	4.9925	41.1476	6.8932	39.7466	97.75
0.19	4.629	41.4107	6.4593	40.029	97.62
0.20	4.4353	41.6616	6.0581	40.3075	97.50
0.21	3.98863	41.1251	5.6753	40.591	97.38
0.22	3.6913	42.459	5.3265	40.8664	97.25

**Tableau IV-4-** Résultats obtenus de la compression avec le JPEG2000.

Temps d'exécution LENA originale: t=3.6606s

Temps d'exécution LENA reconstruite: t=3.8041s



**Figure IV-8- Variation de PSNR en fonction de bpp pour LENA.**

## 2) Pour l'image Visible :

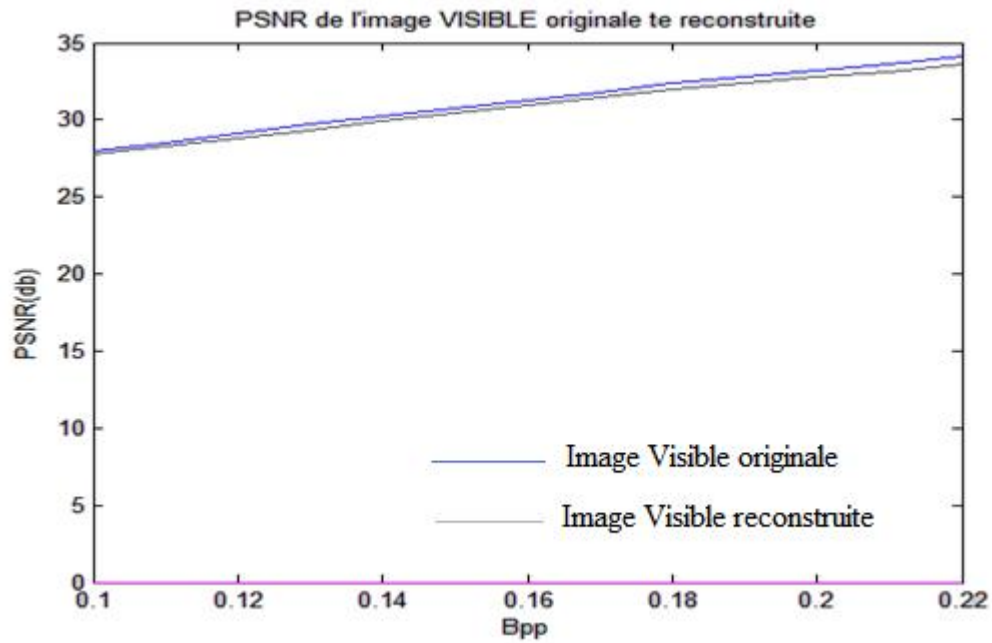
Le tableau suivant indique la variation de PSNR et MSE de la même image LENA en fonction de nombre de bits par pixel (bpp), en faisant varier le bpp de 0.1 :0.01 :0.22.

bpp	MSE de l'image visible avant insertion	PSNR de l'image visible avant insertion	MSE de l'image visible après insertion	PSNR de l'image visible après insertion	Taux en %
0.1	103.4807	27.9822	108.235	27.7871	98.75
0.11	90.9888	28.5409	96.9685	28.2735	98.62
0.12	99.3148	29.1373	84.5583	28.8592	98.50
0.13	69.3818	29.7183	75.1848	29.3695	98.38
0.14	61.9789	30.2084	66.7307	29.8875	98.25
0.15	54.6307	30.7564	58.8457	30.4337	98.2
0.16	48.0493	31.3139	52.2841	30.9471	98.00
0.17	42.7794	31.8185	46.1781	31.3864	97.88
0.18	37.5042	32.39	40.9789	32.0052	97.75
0.19	34.0623	32.8081	37.469	32.3941	97.62
0.20	31.0615	33.2086	34.1759	32.7936	97.50
0.21	28.4274	33.5934	31.4487	33.1548	97.38
0.22	25.1316	34.1286	28.3813	33.6005	97.25

**Tableau IV-5- Résultats obtenus de la compression avec le JPEG2000.**

Temps d'exécution de l'image visible originale :  $t=3.9533$  s.

Temps d'exécution de l'image visible reconstruite :  $t=3.9765$  s.



**Figure IV-9-Variation de PSNR en fonction de bpp pour l'image Visible.**

### 3) Pour la séquence d'image météosats :

Le tableau suivant indique la variation de PSNR et MSE de la séquence d'image (Seq1, Seq2) originale en fonction du nombre de bits par pixel (bpp), en faisant varier le Bpp de 0.1 :0.01 :0.22.

Bpp	MSE Seq1 avant insertion	PSNR Seq1 avant insertion	MSE Seq2 avant insertion	PSNR Seq2 avant insertion	PSNRmoy avant insertion	Taux en %
0.1	1.2587	47.1316	0.9669	48.277	47.7043	98.75
0.11	1.2017	47.3328	0.8586	48.7931	48.063	98.62
0.12	1.1233	47.6257	0.7574	49.3374	48.4815	98.50
0.13	1.0503	47.9178	0.6394	50.0734	48.9956	98.38
0.14	0.9317	48.4382	0.5948	50.3873	49.1427	98.25
0.15	0.8445	48.8648	0.5806	50.4919	49.6784	98.2
0.16	0.6755	49.8348	0.5469	50.7519	50.2933	98.00
0.17	0.6114	50.2678	0.4891	51.2366	50.7522	97.88
0.18	0.6022	50.3337	0.4213	51.8851	51.1994	97.75
0.19	0.5843	50.4652	0.3868	52.2563	51.3602	97.62
0.20	0.5584	50.6611	0.3626	52.5365	51.5988	97.50
0.21	0.4856	51.2677	0.3512	52.6759	51.9718	97.38
0.22	0.4046	52.0602	0.338	52.8415	52.4508	97.25

**Tableau IV-6-** Résultats obtenus de la compression avec le JPEG2000.

Temps d'exécution Seq1:  $t=7.8558s$

Temps d'exécution Seq2 :  $t= 7.4705 s$

Le tableau suivant indique la variation de PSNR et MSE de la séquence d'images (Seq1, Seq2) reconstruite en fonction du nombre de bits par pixel (bpp), en faisant varier le Bpp de 0.1 :0.01 :0.22.

bpp	MSE Seq1 Après insertion	PSNR Seq1 Après insertion	MSE Seq2 Après insertion	PSNR Seq2 Après insertion	PSNRmoy Après insertion	Taux en %
0.1	0.8813	48.6797	0.7277	49.5112	49.0954	98.75
0.11	0.8196	48.9946	0.6529	49.9827	49.4886	98.62
0.12	0.7348	49.4691	0.6276	50.1538	49.8114	98.50
0.13	0.688	45.7549	0.5022	51.1219	50.4384	98.38
0.14	0.6347	50.105	0.4548	51.5528	50.8289	98.25
0.15	0.4831	51.2905	0.3757	52.3829	51.8367	98.2
0.16	0.4606	51.4974	0.3091	53.2294	52.3634	98.00
0.17	0.3788	52.3464	0.2738	53.7566	53.0515	97.88
0.18	0.3199	53.0811	0.2581	54.0125	53.6468	97.75
0.19	0.2804	53.6537	0.2242	45.6251	54.1394	97.62
0.20	0.2677	53.8538	0.1924	55.2886	54.5712	97.50
0.21	0.2366	54.3906	0.1622	56.0308	55.2107	97.38
0.22	0.209	54.9304	0.076	59.3209	57.1251	97.25

**Tableau IV-7-** Résultats obtenus de la compression avec le JPEG2000.

Temps d'exécution de Seq1 reconstruite :  $t=7.6111s$

Temps d'exécution de Seq2 reconstruite :  $t=7.6672s$

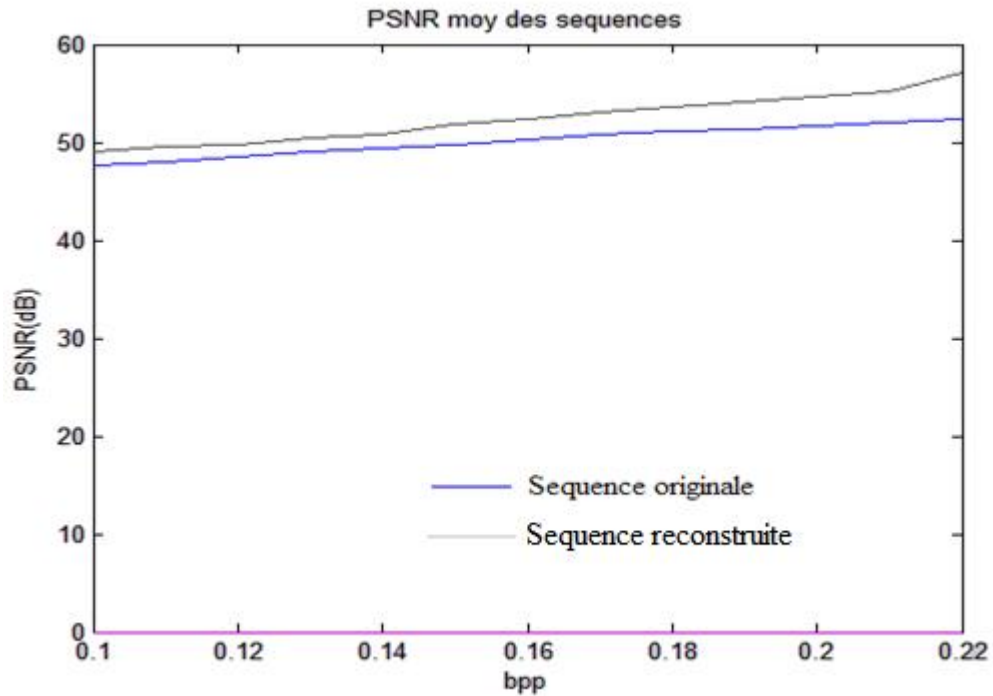


Figure IV-10- Variation de PSNRmoy en fonction de bpp pour la séquence (Seq1, Seq2)

IV-2-1-4- Quelques figures obtenus pour  $P_c = 33.33\%$  :

1) Sur l'image LENA :



a) Image originale compressée

**bpp=0.11**

**temps d'exécution=0.558 s**



b) Image reconstruite compressée

**bpp=0.11**

**temps d'exécution=0.661 s**



**c) Image originale compressée**

**bpp=0.22**

**temps d'exécution=0.624 s**



**d) Image reconstruite compressée**

**bpp=0.22**

**temps d'exécution=0.607 s**



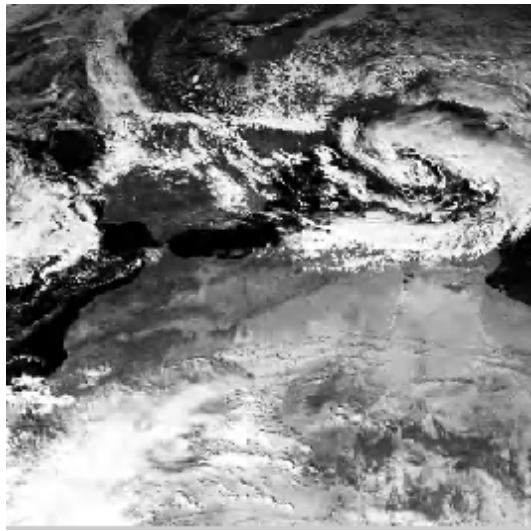
**e) Image reconstruite bpp=0.11**



**f) Image reconstruite bpp=0.22**

**Figure IV-11-Résultats obtenus de la compression avec JPEG2000 sur l'image LENA.**

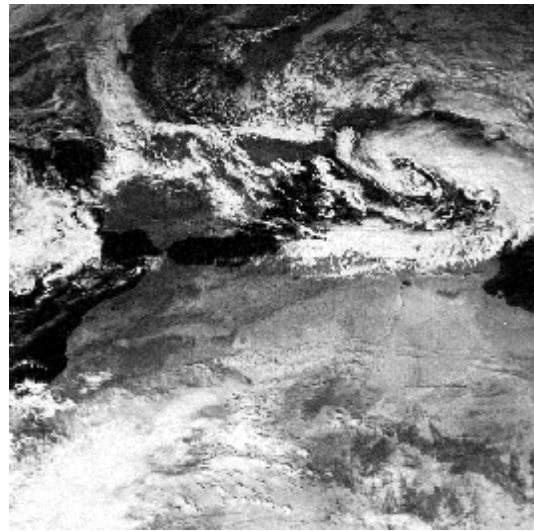
## 2) Sur l'image Visible :



a) Image originale compressée

bpp=0.11

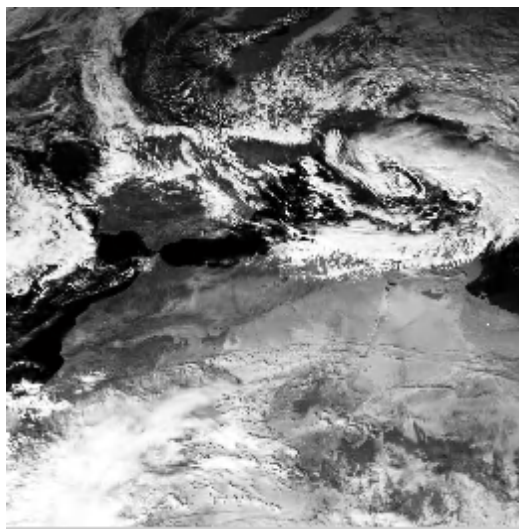
temps d'exécution=0.494 s



b) Image reconstruite compressée

bpp=0.11

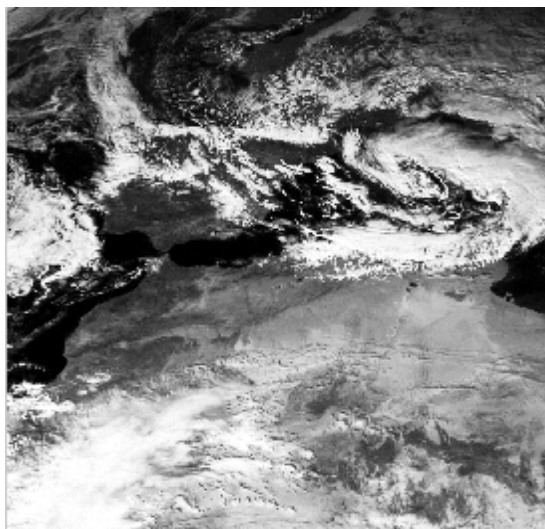
temps d'exécution=0.606 s



c) Image originale compressée

bpp=0.22

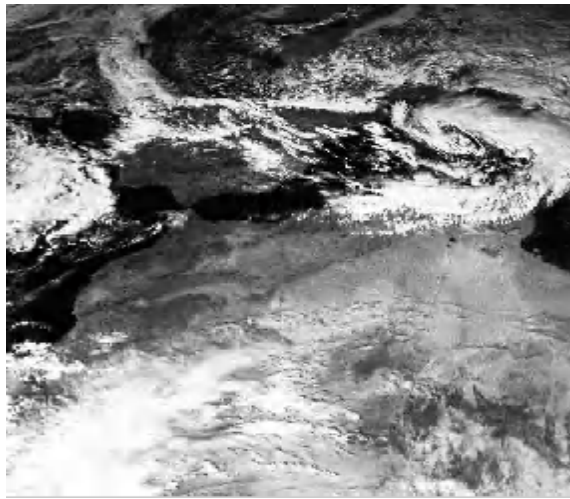
temps d'exécution=0.578 s



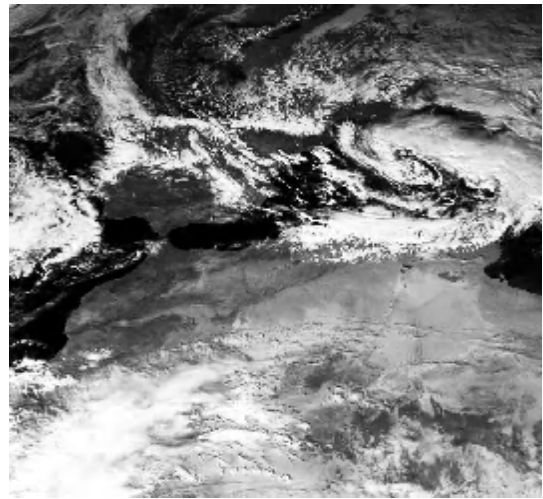
d) Image reconstruite compressée

bpp=0.22

temps d'exécution=0.619 s



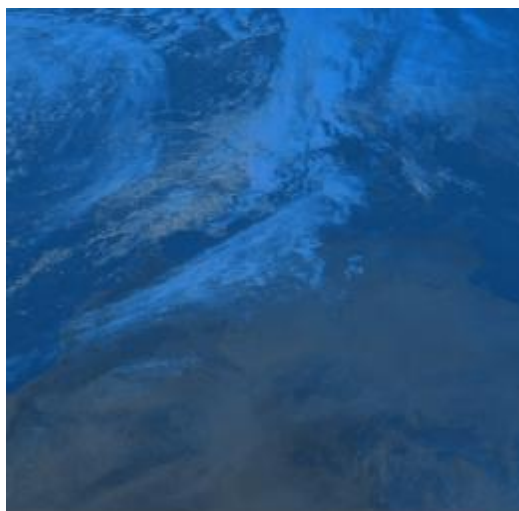
e) Image reconstruite  $\text{bpp}=0.11$



f) Image reconstruite  $\text{bpp}=0.22$

**Figure IV-12-Résultats obtenus de la compression avec JPEG2000 sur l'image Visible.**

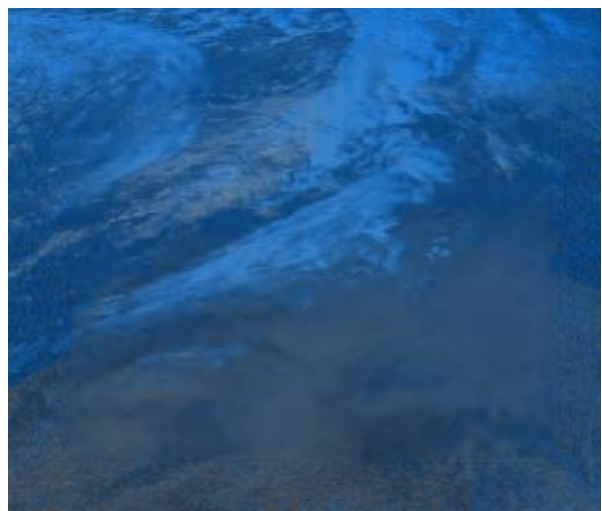
2) Sur la séquence Seq1 :



a) Image seq1 originale compressée

$\text{bpp}=0.1$

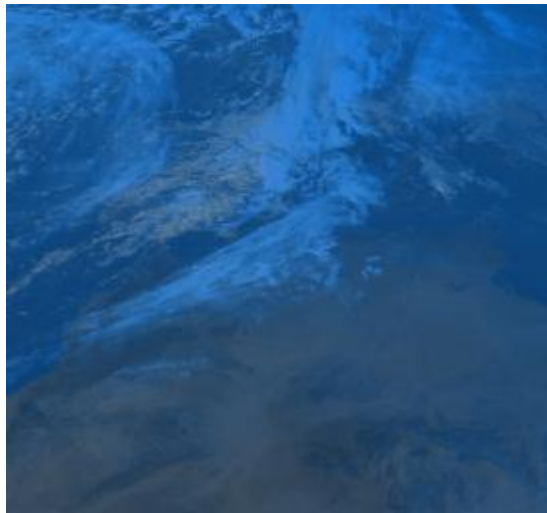
temps d'exécution=1.172 s



b) Image seq1 reconstruite compressée

$\text{bpp}=0.1$

temps d'exécution=1.132 s



**c) Image seq1 originale compressée**

**bpp=0.22**

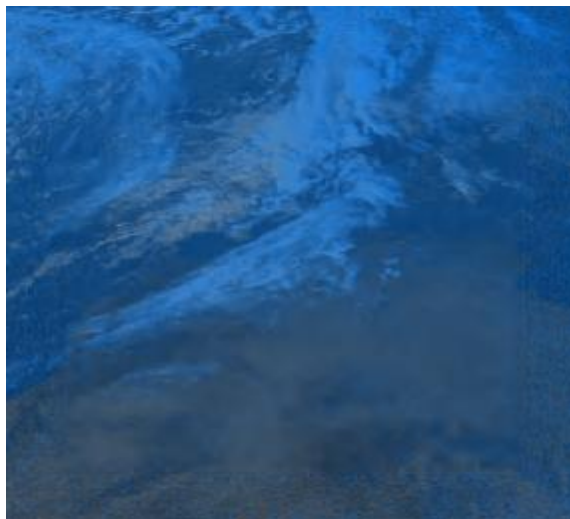
**temps d'exécution=1.266 s**



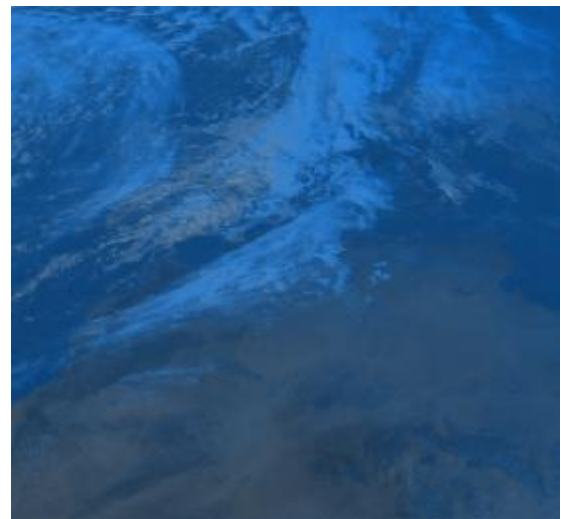
**d) Image seq1 reconstruite compressée**

**bpp=0.22**

**temps d'exécution=1.092 s**



**e) Image seq1 reconstruite bpp=0.1**



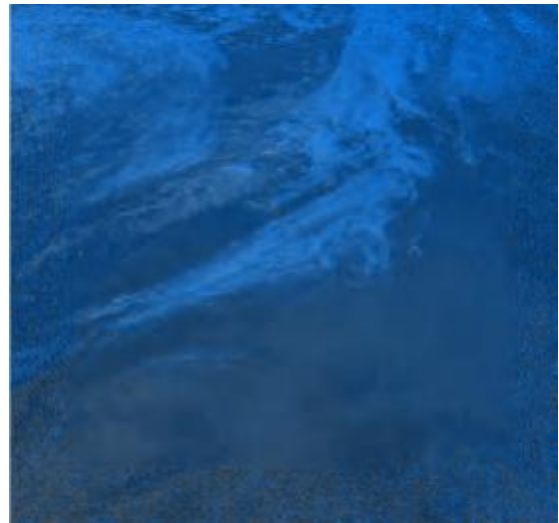
**f) Image seq1 reconstruite bpp=0.22**

**Figure IV-13-Résultats obtenus de la compression avec JPEG2000 sur la séquence1.**

## 3) Sur la séquence Seq2 :



a) Image seq2 originale compressée

**bpp=0.11****temps d'exécution=0.972 s**

b) Image seq2 reconstruite compressée

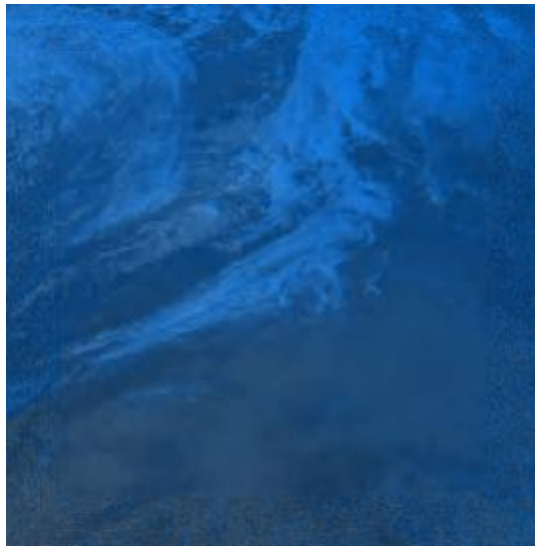
**bpp=0.11****temps d'exécution=1.030 s**

c) Image seq2 originale compressée

**bpp=0.22****temps d'exécution=1.320 s**

d) Image seq2 reconstruite compressée

**bpp=0.22****temps d'exécution=1.090 s**



e) Image seq2 reconstruite bpp=0.11



f) Image seq2 reconstruite bpp=0.22

**Figure IV-14**-Résultats obtenus de la compression avec JPEG2000 sur la séquence2.

### IV-3-Interprétation des résultats de compression avec JPEG2000:

A l'analyse des différents tableaux IV-1, IV-2, IV-3, IV-4, IV-5, IV-6, IV-7 ainsi que les différents graphes appropriés nous pouvons constater que:

**Pour le cas où  $P_c = 66.66\%$  :**

D'après les résultats obtenus (voir les figures : IV-2 et IV-3) nous avons remarqué que la qualité de l'image compressée est d'autant meilleure que le PSNR augmente et on remarque aussi que le rapport signal sur bruit augmente proportionnellement avec le bpp, et aussi le PSNR de l'image originale et plus grand que le PSNR de l'image compressée.

D'après le graphe de la figure IV-4 on remarque que pour des bpp très faibles, le PSNR de l'image originale et compressée sont très voisins et à partir de  $\text{bpp} = 0.03$  le PSNR augmente en fonction de bpp, ainsi que avec l'augmentation de bpp le PSNR de l'image originale et compressée se divergent,

**Pour le cas ou  $P_c = 33.33\%$  :**

D'après le graphe de la figure IV-8 nous avons remarqué que la qualité de l'image compressée est d'autant meilleure que le PSNR augmente et on remarque aussi que le rapport signal sur bruit augmente proportionnellement avec le bpp.

D'après le graphe de la figure IV-9 on remarque que les PSNR de l'image Visible originale et compressée sont très proches et pour les très faibles bpp les PSNR sont confondus. Ces résultats nous constatons que le  $PSNR_{moy}$  augmente avec l'augmentation de bpp.

Le graphe de la figure IV-10 a montré que le PSNR de la séquence (Seq1, Seq 2) originale et compressée augmente proportionnellement en fonction de bpp et dans la zone des faibles bpp le PSNR de la séquence originale et compressée sont voisin et se divergent avec l'augmentation de bpp.

Et l'image reconstituer dans le cas ou on a utilisé deux séquence (Seq1, Seq2) est mieux que pour le d'utilisation une séquence.

L'analyse des résultats obtenus par cette méthode, montre que le taux de compression influe directement sur la qualité visuelle des images de la séquence reconstituée et sur l'image insérée. Ainsi, la séquence reconstruite et l'image insérée reconstruite compressée à un taux de 0.11 bpp laisse des artefacts sur ces images d'une façon spirale. Tandis que pour un taux de compression de 0.22, les images reconstruites ne laissent aucuns artefacts.

**Conclusion :**

En ce référant aux différents résultats obtenus dans les divers cas étudiés, nous pouvons déduire que la qualité de l'image reconstruite après insertion spirale dans la séquence d'images Seq est directement liée aux taux de compression. En effet pour un codage voisinage de  $bpp=0.22$  notre technique donne des bonnes résultats en terme de qualité d'image et rapport signal sur bruit. Cependant, la qualité de la séquence est dégradée avec la diminution de bpp.

**Perspective :**

Notre méthode peut être utilisée dans différents domaines tel que les images satellitaires, cette méthode peut compresser une séquence d'image au lieu de compresser une image par image tout en gardant la qualité des images insérées lors de la restauration, ainsi on peut l'appliquer dans les images médicales car on peut fixer une région d'intérêt qui sera importante pour nous est d'insérer des images dans les autres régions qui sont moins importantes.

# *Conclusion Générale*

Les travaux présentés dans ce mémoire ont été élaborés dans le but d'améliorer nos connaissances dans un domaine de traitement d'images à savoir la compression d'images d'une façon spirale.

Alors, dans le premier chapitre nous sommes intéressés au traitement d'images d'une façon générale, ensuite nous avons exposé des différentes techniques de compression d'images telles que Huffman, Shannon-Fano et LZW etc.

Dans le second chapitre, nous avons donné quelques connaissances sur les ondelettes et des notions générales sur l'analyse multi-résolution, ainsi nous avons passé à l'ondelette biorthogonale et à l'analyse multi-résolution de deux dimensions.

Après avoir donné des connaissances théoriques nous avons abordé dans le troisième chapitre notre méthode adoptée qui consiste à l'insertion spirale des images dans d'autres images.

Pour tester l'efficacité de notre méthode nous avons donné dans le quatrième chapitre quelques résultats et leurs interprétations.

Dans ce travail, nous avons proposé une méthode de compression d'une séquence d'images, permettant de réduire l'énorme volume de compresser image par image, tout en gardant une qualité visuelle acceptable en utilisant une méthode de compression d'une séquence d'image d'une façon spirale.

L'insertion spirale des images dans une séquence d'images a permis de réduire le nombre d'images à compresser ainsi le temps nécessaire pour la compression de ces images.

Alors nous avons effectué plusieurs tests, dont les résultats sont satisfaisants de point de vue du taux de compression et la qualité visuelle des images reconstruites.

Notre travail peut être un sujet à d'autres améliorations, exemple on peut prendre une séquence de luminance chrominance (Y Cr Cb) au lieu d'une séquence couleur (RVB), les résultats seront plus acceptables.

# *Annexes*

Le standard JPEG-2000 est le standard le plus récent de compression d'images. Cet algorithme est plus avantageux que le JPEG car il procède par dégradation sélectives de certaines zones moins stratégiques de l'image.

Il utilise une transformée en ondelettes et effectue un codage optimisé de façon à satisfaire de nombreuses fonctionnalités grâce entre autres à son étape de codage entropique. Le standard décrit le fonctionnement du traitement mais n'impose aucune implémentation particulière. Les implémentations du standard ou des parties du standard font donc l'objet d'études variées pour obtenir des solutions ayant un bon rapport cout-performances, à savoir :

- Ø Un découpage de l'image originale en sous-images de format rectangulaire, permet aussi de proposer un pré-découpage de grandes images pour proposer une compression indépendante de chacune de ces sous-images.
- Ø Il est possible de spécifier une région d'intérêt, une zone de l'image va être rendue prioritaire : la meilleure qualité y sera donc obtenue plus rapidement que pour le reste de l'image.

### **Les parties de la norme JPEG 2000 :**

JPEG 2000 est un nouveau système de codage d'images qui utilise les connaissances actuelles dans le domaine de la technologie des ondelettes.

Précisons, au cœur de JPEG il y a la transformée de Fourier discrète. Au cœur de JPEG 2000, il y a une décomposition en ondelettes. Le principe même du codage est différent. Il n'est pas une amélioration de JPEG, il constitue une autre manière d'analyser, de décomposer l'image pour la compresser avec ou sans pertes en apportant des fonctionnalités complètement nouvelles, en particulier pour les transmissions à faible bande passante. Cette nouvelle norme est composée de **11 parties** dont certaines sont publiées et d'autres non. Passons-les en revue rapidement :

- **Partie 1 :**

Publiée par l'ISO, elle définit la technologie minimale de l'algorithme de décodage ainsi que le format du codestream devant être compris par tous les produits se réclamant compatibles avec la norme. En outre, cette partie introduit le format de fichier JP2 qui permet d'associer au codestream de l'information additionnelle sur l'image.

- **Partie 2 :**

Celle-ci définit des extensions a la partie précédente afin d'améliorer les performances ou augmenter le nombre de fonctionnalités. Elle est surtout dédiée a des applications plus spécifiques telles que le codage de données hyper-spectrales (ex : image satellite).

- **Partie 3 :**

Elle correspond à Motion JPEG 2000. Elle définit un format de codage de séquences audiovisuelles et d'animations en tant que succession d'images indépendantes compressées par un système compatible avec JPEG 2000 partie1.

- **Partie 4 :**

Elle définit les règles de compatibilité avec la norme JPEG 2000.

- **Partie 5 :**

Elle est constituée de logiciels de référence. Sa publication est immense ; il existe en particulier deux implémentations de JPEG 2000, l'une en C (projet Jasper) l'autre en Java (projet JJ2000) élaboré par CANON Research France et Ericsson.

- **Partie 6 :**

Elle spécifie un format de fichier pour les images dite compound c.à.d. composé à la fois de textes, d'images, de graphique...etc.

- **Partie 7 :**

Elle définit un support minimal de compatibilité pour des appareils à puissance limitée (ex : appareils photo numérique ...).

- **Partie 8 (JPSEC) :**

Consacrée aux problèmes de sécurité, a été lancée en décembre 2001 à Sidney. Aucune version préparatoire n'est encore publiée.

- **Partie9 (JPIP) :**

Elle est consacrée aux outils d'interactivité, aux API et protocoles a été lancée en mars 2003.

- **Partie10 (JPIP) : et partie11 (JPWL) :**

Elles ont été lancées en décembre 2001 à Sidney. Aucune version préparatoire n'est encore publiée.

- **Partie 12(ISO Base Media File Format) :**

Elle est en fait une modification de la partie trois est en cours d'élaboration. Il est à noter qu'à ce jour, seule la partie 1 a atteint le stade de norme internationale. C'est d'ailleurs essentiellement cette partie que nous détaillons.

### **Les différentes caractéristiques du JPEG 2000 :**

- **Décodage progressif :**

Par nature les codestream JPEG 2000 sont à résolutions multiples. L'image s'y trouve compressée résolution par résolution sans redondance. On peut donc utiliser la résolution la plus appropriée au système d'affichage. Il est aussi possible d'ordonner l'information dans le codestream de manière croissante en qualité grâce au concept des layers ou encore de grouper l'information par rapport à la position qu'elle occupe dans une résolution de l'image.

- **Les régions d'intérêt :(ROI Region Of Interest)**

La norme permet une affectation hétérogène du budget de qualité d'une image. On pourra décider qu'une zone de l'image, de forme rectangulaire ou elliptique, doit être privilégiée par rapport au reste de l'image en étant reproduite avec moins de perte voire même sans pertes.

- **Codage avec ou sans pertes :**

Il faut insister sur le fait que le même algorithme permet de compresser avec ou sans perte. Nous verrons d'ailleurs tout à l'heure que la compression sans pertes de JPEG 2000 est très intéressante et à mon avis promise à un grand avenir.

- **Détection d'erreurs en milieu bruité :**

Les codestream JPEG 2000 sont très résistants et la norme a prévu un système de détection d'erreur très efficace.

- **Accès aléatoire aux différentes zones de l'image :**

Il est possible de ne décoder qu'une certaine zone d'une image quand elle est très grande.

- **Traitement d'une grande variété de types d'images :**

La norme JPEG 2000 s'applique aux images photographiques en couleurs mais aussi aux images en niveaux gris et aux images composées de zones différentes.

- **Compatibilité avec les profils ICC :**

JPEG 2000 emporte des informations précises sur les couleurs et est en particulier compatible avec les profils ICC qui sont devenus incontournables dans le pré-presse.

- **Métadonnées :**

JPEG 2000 permet d'enregistrer des métadonnées très variées grâce à un système optionnel perfectionné.

- **Images 32 bits :**

JPEG 2000 contrairement à JPEG permet de traiter des images 32bits.

- **Transparence :**

Contrairement à JPEG, JPEG 2000 permet d'enregistrer une image comportant de la transparence.

Une ondelette est une forme d'onde de la durée limitée qui a une valeur moyenne zéro, ou les ondelettes sont des ondes localisées par temps court. En regardant des images des ondelettes et des ondes sinusoïdales, on peut voir intuitivement que des signaux avec les changements pointus pourraient mieux être analysés avec une ondelette irrégulière qu'avec un sinusoïde douce.

### **propriétés de la fonction d'ondelette : [10]**

La théorie des ondelettes permet d'associer aux fonctions d'échelle et d'ondelette un certain nombre de propriétés intéressantes pour l'analyse et la synthèse des signaux :

#### **Support :**

Si la fonction d'échelle  $\varnothing$  et l'ondelette  $\psi$  sont à support compact, les filtres H et G sont à réponse impulsionnelle finie (FIR), et ils sont favorables à une implantation informatique. Dans le cas contraire, une décroissance rapide des filtres permet tout de même de les approcher raisonnablement par des filtres FIR.

#### **Coefficients rationnels :**

Dans une implantation informatique, il est souhaitable de réaliser les transformations le plus rapidement possible avec le moins d'occupation de l'espace mémoire. Il sera donc intéressant de ne pas effectuer les calculs en réel. Les filtres ayant des coefficients  $h_k$  et  $g_k$  rationnels (ou même dyadiques) sont donc très recherchés. Certaines précautions permettent même d'effectuer les transformations directement avec des entiers.

#### **Orthogonalité :**

Si les ondelettes sont orthogonales, la transformée en ondelettes est stable et réversible. Les opérateurs de projections dans les différents sous-espaces  $V_j$  et  $W_j$  conduisent à des approximations optimales au sens de  $L^2$ .

#### **Symétrie :**

Si les fonctions d'échelle et d'ondelette sont (anti-)symétriques, les filtres sont alors à phase linéaire. On observe une distorsion de la phase à la reconstruction lorsque cette propriété est absente.

En fait, il n'existe pas d'ondelettes orthogonales qui soient à la fois symétriques et à support compact. C'est pourquoi, les ondelettes biorthogonales ont été introduites et sont abondamment utilisées.

**Régularité :**

La régularité de l'ondelette est importante en compression. Les compressions correspondent souvent à des mises à zéro de certains coefficients d'ondelettes, ce qui équivaut à négliger les composantes  $d_j = \langle f, \psi_{j,k} \rangle$  de la fonction originale. La régularité de l'ondelette permet de masquer les erreurs de reconstruction. De plus, une plus grande régularité entraîne une meilleure localisation fréquentielle des filtres.

**Moments nuls :**

L'efficacité d'une ondelette pour la compression des signaux est caractérisée par sa capacité à représenter sur un nombre réduit de coefficient non nuls la plus grande partie de l'information. Autrement dit, une ondelette est plus efficace si elle donne un grand nombre de coefficients détails qui voisine zéro.

Cette efficacité peut être définie par le nombre de ces premiers moments nuls [18] :

$$N = \max \{n \in \mathbb{N}, \int_{-\infty}^{+\infty} t^n \psi(t) dt\}$$

**Quelques types d'ondelettes :****L'ondelette de Haar :**

C'est la plus simple des ondelettes définie sur l'intervalle  $[0,1]$  ou parfois sur  $[\frac{-1}{2}, \frac{1}{2}]$ , c'est la fonction H constante par morceaux qui vaut :

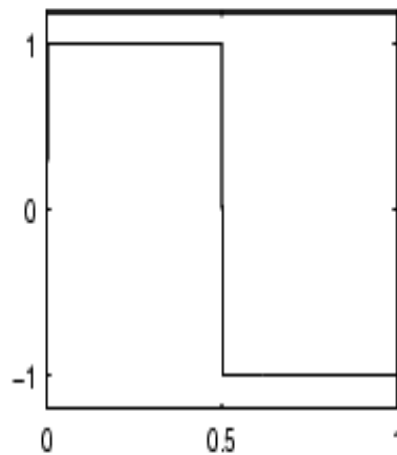
$$H(x) = \begin{cases} -1 & \text{si } x \in [0, \frac{1}{2}] \\ 1 & \text{si } x \in [\frac{1}{2}, 1] \end{cases}$$

Cette ondelette est très simple elle est discontinue et donc facile à mettre en œuvre algorithmiquement.

De plus, son support est compact c.à.d. elle est bien localisée en espace. Elle n'a qu'un seul moment nul et discontinue ; sa transformée de Fourier est de la forme suivant :

$$H(\omega) = j e^{-j\pi\omega \frac{\sin^2 \frac{\pi\omega}{2}}{\frac{\pi\omega}{2}}},$$

Cette dernière oscille beaucoup et sa localisation en fréquence n'est pas bonne.



**Figure II.8 :** Ondelette de Haar 

**Ondelettes de Daubechies :**

Ingrid Daubechies (1990) a complété l'œuvre de Haar en 1987 [Daubechies, 1988]. Pour toute valeur  $r$ , Daubechies construit une base orthonormée de  $L^2(\mathbb{R})$  de la forme,

$$2^{\frac{j}{2}}\psi_r(2^j x - K), j \in \mathbb{Z}, K \in \mathbb{Z}$$

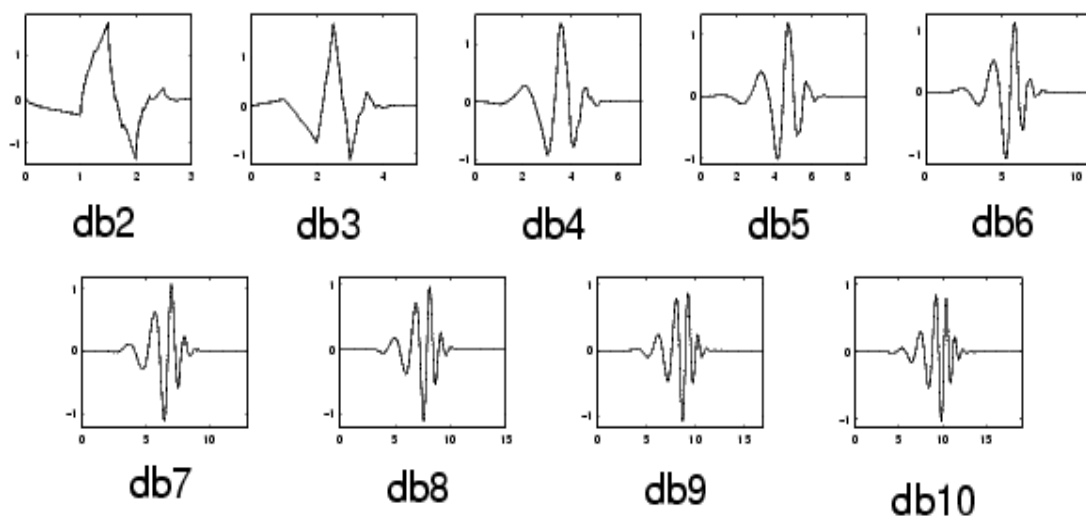
En effet,  $\psi$  est définie sur un support compact  $[0, 2r + 1]$  et satisfait l'équation suivante:

$$\int_{-\infty}^{+\infty} \psi_r(x) dx = \int_{-\infty}^{+\infty} x^r \psi_r(x) dx$$

Où  $\psi_r(x)$  possèdent  $r$  dérivées continues.

Lorsque  $r = 0$ ,  $\psi_r(x)$  est définie sur  $[0, 1]$ , on retombe sur le système de Haar.

Les noms des ondelettes de famille de Daubechies sont écrits le dbN, où N est l'ordre, et le DB le « nom de famille » de l'ondelette. L'ondelette db1, comme mentionnée ci-dessus, est identique que l'ondelette de Haar. Voici les fonctions livre par pouce carré d'ondelette des neuf prochains membres de la famille :



**Figure II.11 :** les Ondelettes de  **Daubechies**

**L'ondelette de Morlet :**

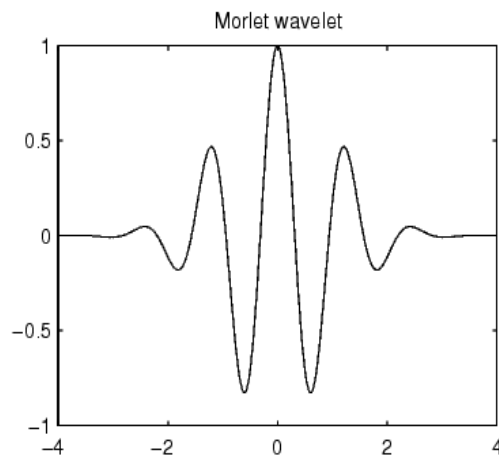
C'est une fonction gaussienne modulée, elle est représentée sous la forme suivant :

$$\psi(t) = e^{-\pi t^2} e^{2j\pi k_0 t}$$

Par exemple, pour  $k_0=5$  sa transformée de Fourier est une gaussienne décalée de  $k_0$  et vaut

$$\hat{\psi}(\omega) = e^{-\pi(\omega-k_0)^2}$$

$\hat{\psi}(0)$  ne vaut pas zéro :  $\psi$  n'est pas de moyenne nulle, si  $k_0$  est suffisamment grand,  $\hat{\psi}(0)$  négligeable ; par exemple pour  $k_0=5$ ,  $\hat{\psi}(\omega)$  est environs  $10^{-8}$  ; pour  $\omega \leq 0$   $\hat{\psi}(\omega)$  est négligeable. Cette fonction est donc couramment utilisée comme une ondelette complexe analytique.



**Figure II.9 :** Ondelette de Morlet

**Ondelettes de Meyer :**

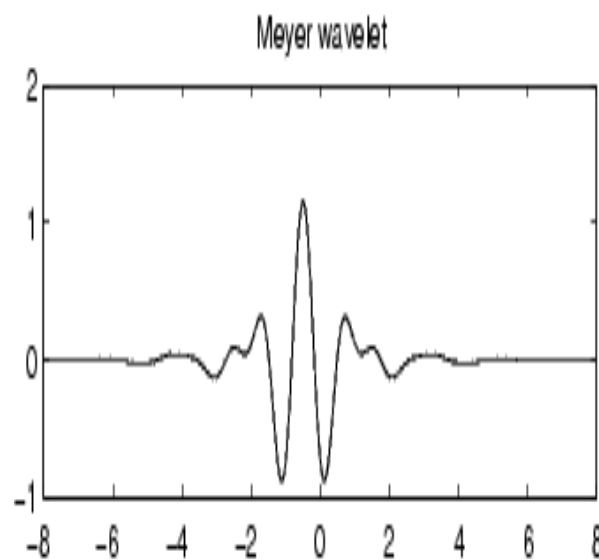
La fonction d'ondelette de Meyer est définie dans le domaine de fréquence comme suit :

$$\hat{\psi}(\omega) = (2\pi)^{-1/2} e^{i\omega/2} \sin\left(\frac{\pi}{2} v\left(\frac{3}{2\pi} |\omega| - 1\right)\right) \quad \text{si} \quad \frac{2\pi}{3} \leq |\omega| \leq \frac{4\pi}{3}$$

$$\hat{\psi}(\omega) = (2\pi)^{-1/2} e^{i\omega/2} \cos\left(\frac{\pi}{2} v\left(\frac{3}{4\pi} |\omega| - 1\right)\right) \quad \text{si} \quad \frac{4\pi}{3} \leq |\omega| \leq \frac{8\pi}{3}$$

$$\hat{\psi}(\omega) = 0 \quad \text{Si} \quad |\omega| \notin \left[\frac{2\pi}{3}, \frac{8\pi}{3}\right]$$

Telle que  $v(a) = a^4(35 - 84a + 70a^2 - 20a^3)$  ;  $a \in [0,1]$ .



**Figure II.10 : Ondelette de Meyer**

**Ondelettes orthogonales :**

La fonction d'échelle est une solution à une équation fonctionnelle fractale, appelée l'équation de raffinement :

$$\phi(x) = \sum_{k=0}^{N-1} a_k \phi(2x - k)$$

Où la séquence  $(a_0, a_1, \dots, a_{N-1})$  des entiers flottants est appelé la séquence d'échelle ou le masque d'échelle. L'ondelette exacte est obtenue par la combinaison linéaire similaire :

$$\psi(x) = \sum_{k=0}^{N-1} b_k \phi(2x - k)$$

Où la séquence  $(b_0, b_1, \dots, b_{M-1})$  des entiers flottants est appelé la séquence d'ondelette ou le masque d'ondelette.

Une condition nécessaire pour l'orthogonalité des ondelettes est que l'ordre d'échelle est orthogonal à tous ses décalages par un chiffre pair des coefficients :

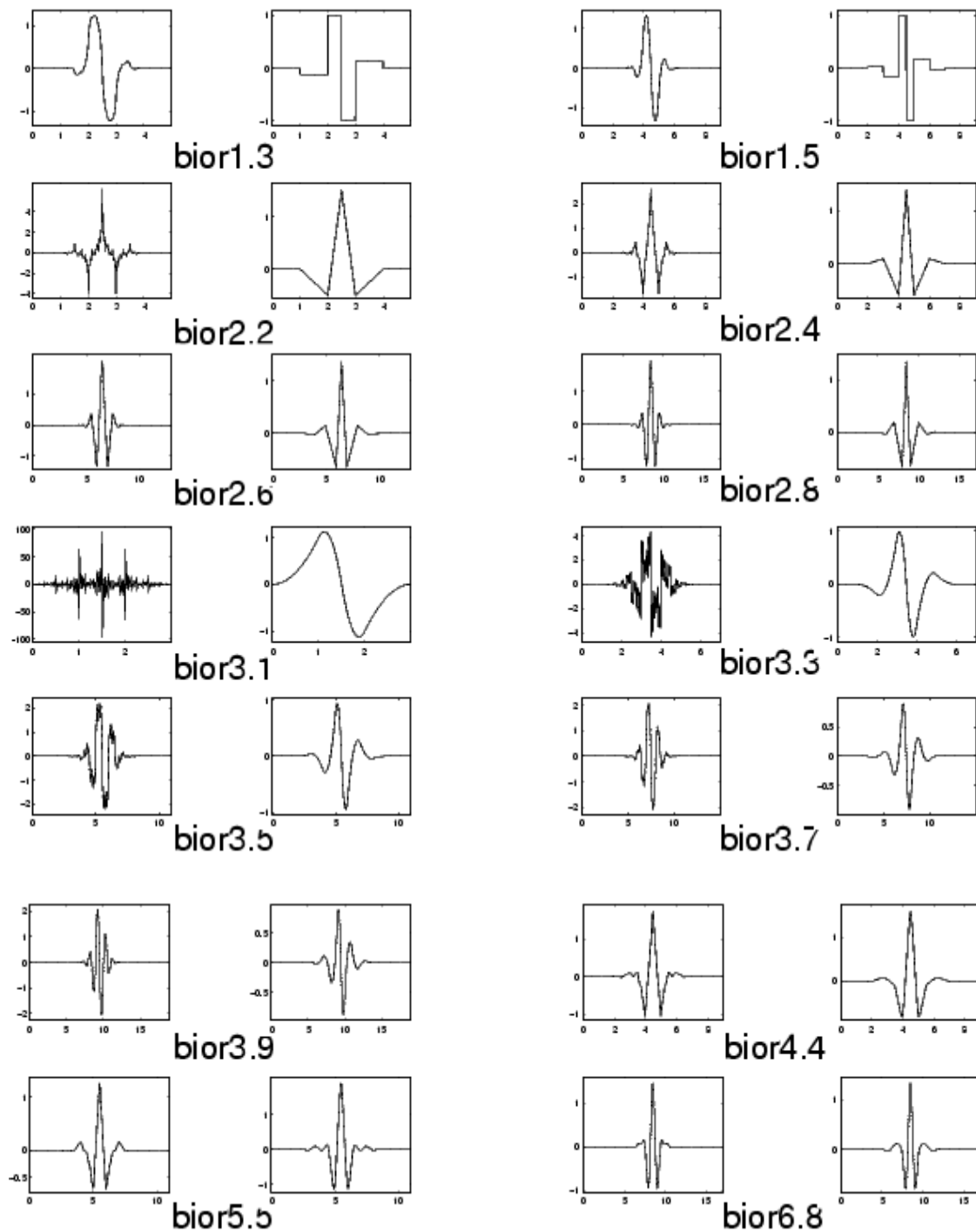
$$\sum_{n \in \mathbb{Z}} a_n a_{n+2m} = 2\delta_{m,0}$$

Dans ce cas-ci, il y a le même nombre  $M=N$  de coefficients dans la dilatation comme dans l'ordre d'ondelette, l'ordre d'ondelette peut être déterminé comme :

$$b_n = (-1)^n a_{N-n-1}$$

**Ondelettes biorthogonales :**

Cette famille d'ondelette présente la propriété de la phase linéaire, qui est nécessaire pour la reconstruction de signal et d'image. En employant deux ondelettes, une pour la décomposition et l'autre pour la reconstruction.



**Figure II.12 :** les Ondelettes **biorthogonales**

i	Coefficients des filtres d'analyse		Coefficients des filtres de synthèse	
	Filtre passe-bas	Filtre passe-haut	Filtre passe-bas	Filtre passe-haut
0	0.6029490182364	1.1150870524570	1.1150870524570	0.6029490182364
$\pm 1$	0.2668641184429	-0.5912717631142	0.5912717631142	-0.2668641184429
$\pm 2$	-0.0782232665290	-0.0575435262285	-0.0575435262285	-0.0782232665290
$\pm 3$	-0.0168641184287	0.0912717631142	-0.0912717631142	0.0168641184287
$\pm 4$	0.0267487574108			0.0267487574108

**Table de coefficients du filtre de Daubechies 9/7**

Correspondance entre les biorthogonales Nr.Nd (bior Nr.Nd) et Daubechies N/M :

Tel que :

**Nr et N** représentent le nombre de coefficients des filtres de reconstruction.

**Nd et M** représentent le nombre de coefficients des filtres de décomposition.

Biorthogonales Nr.Nd	Daubechies N/M
bior 1.1	2/2
bior 1.3	6/2
bior 1.5	10/2
bior 2.2	5/3
bior 2.4	9/3
bior 2.6	13/3
bior 2.8	17/3
bior 3.1	4/4
bior 3.3	8/4
bior 3.5	12/4
bior 3.7	16/4
bior 3.9	20/4
bior 4.4	9/7
bior 5.5	9/11
bior 6.8	17/11

## Notations

Le produit scalaire de deux fonctions  $f(x)$  et  $g(x)$  est donné par la relation suivante :

$$\langle f(x), g(x) \rangle = \int_{+\infty}^{-\infty} g(x) \cdot f(x) dx$$

La norme définit dans  $L^2(R)$  de  $f(x)$  est donnée par :

$$\|f(x)\| = \int_{+\infty}^{-\infty} \|f(x)\|^2 dx$$

La convolution de deux fonctions  $f(x) \in L^2(R)$  et  $g(x) \in L^2(R)$  est définie par :

$$f * g(u) = (f(x) * g(x))(u) = \int_{+\infty}^{-\infty} f(x) \cdot g(u - x) dx$$

La transformée de FOURIER de  $f(x) \in L^2(R)$  est définie comme suit :

$$\tilde{f}(w) = \int_{+\infty}^{-\infty} f(x) e^{-iwx} dx$$

$L^2(R)$  : Espace vectoriel des fonctions à deux dimensions  $f(x, y)$ , mesurables et à carrés intégrable.

Le produit interne de deux fonctions  $f(x, y)$  et  $g(x, y)$  appartenant à  $L^2(R^2)$  est défini par l'équation suivante :

$$\langle f(x, y), g(x, y) \rangle = \int_{+\infty}^{-\infty} \int_{+\infty}^{-\infty} g(x, y) \cdot f(x, y) dx dy$$

La transformée de FOURIER de  $f(x, y) \in L^2(R^2)$  est donné comme suit :

$$\tilde{f}(w_x, w_y) = \int_{+\infty}^{-\infty} \int_{+\infty}^{-\infty} f(x, y) e^{-i(w_x x + w_y y)} dx dy$$

# *Références bibliographiques*

## R é f é r e n c e s b i b l i o g r a p h i q u e s

[1] **S. OULD YAHIA, F.ABROUS**, Compression d'images fixes par la Transformée en Cosinus Discrète (TCD, mémoire d'ingénieur d'état en électronique. UMMTO 1999.

[2] **S. LOUNIS, C. DJAB**, Compression d'une séquence d'images par Ondelettes, mémoire d'ingénieur d'état en électronique. UMMTO 1999.

[3] **N. NAIT AMARA, R.CHOUCHANE**, Compression d'images fixes par classification de régions en associant les ondelettes et les fractales, mémoire d'ingénieur d'état en électronique. UMMTO 2008.

[4] **D. HAMEL, N. MESLEM**, Traitement d'image sous MATLAB par la transformée en Ondelettes, mémoire d'ingénieur d'état en électronique. UMMTO 2002.

[5] **FRE DRIQUE TRUCHETET** « ondelette pour le signal numérique » édition 1998.

[6] **O. KAMEL, A.MOURAD** « Génération d'un dictionnaire pour le codage des coefficients en ondelettes des images numériques » mémoire d'ingénieur d'état en électronique. UMMTO 2003

[7] **ISODORE PAUL AKAM BITA** « sur une approche de l'analyse en composantes indépendantes à la compression des images multicomposantes ». 12 Février 2007 université Joseph Fourier de Grenoble

[8] **MOURAD LAHDIR**« compression d'image en sous bandes associons une quantification vectorielle et une quantification en cosinus, application aux images météosats » thèse de magistère en électronique octobre 1999UMMTO.

- [9] **C.GASQUET et P.WITOMSKI** « Analyse de Fourier et applications : filtrage calcule numérique et ondelettes ». Université Joseph Fourier (Grenoble I). 2004.
- [10] **S.LAKROUM** « Transmission progressive des images ».mémoire d'ingénieur en électronique. UMMTO 2002.
- [11] **M.KUNT** Traitement numérique des signaux, Edition Dunod 1981.
- [12] **A.COHEN**, Ondelettes et traitement numérique du signal, Edition MASSON PARIS 1992.
- [13] **Eric INCERTI**, compression d'images et standard, Edition Vuibert informatique PARIS 2003.
- [14] **M.IDRI** « compression irréversible d'image numérique a base d'ondelettes entières » mémoire d'ingénieur en électronique, UMMTO 2007.
- [15] **M.LAHDIR** «Nouvelle approche de compression d'images basée sur les ondelettes et les fractales : application aux images météoat » thèse doctorat, UMMTO, 2007
- [16] **X.DELAUNY** « compression d'image satellite par poste transformées dans le domaine ondelettes » thèse doctorat, Université de Toulouse, 2008

**Résumé :**

Ce travail rentre dans le cadre des activités du Laboratoire d'Analyse et de Modélisation des Phénomènes Aléatoires (LAMPA) du département d'électronique. Dans ce projet, nous proposons un schéma de compression des images Météosat basé sur la technique à insertion spirale. Dans une séquence Météosat formée d'images visible et infrarouge, les pixels d'un certain nombre d'images sont entrelacés dans d'autres images de la même séquence, permettant ainsi de minimiser le taux de données MSG très important à compresser.

Cette insertion spirale qui est effectuée au bord des images, permet d'avantager la région centrale de la séquence. Celle-ci représente la région d'intérêt (ROI) à préserver lors de la compression. La séquence réduite est compressée de façon efficace par l'algorithme de JPEG2000. Après décompression, les images seront évaluées par des critères objectifs. L'application de ce schéma de compression permettra de vérifier l'amélioration notable aux niveaux des résultats (taux de compression atteint, qualité de restitution et temps d'exécution).

**Mots clés :**

codage d'images, insertion spirale, Météosat seconde génération (MSG), JPEG2000