

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma –

Faculté des Mathématique, d'Informatique et des Sciences de la matière

Département d'Informatique



## Mémoire de Fin d'études Master

Filière : Informatique

Option : Académique

Thème :

13/873

---



---

# Compression Des Données Multimédias

---



---

**Encadré Par :**

Kouahla Mohamed Nadjib

**Présenté Par :**

Ghomrani Samir

Belgherbi Med Amine

2013/2014

# REMERCIEMENTS

*En préambule à ce mémoire nous remercions ALLAH qui nous aide et nous donne la patience et le courage durant ces longues années d'étude.*

*Nous souhaitant adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.*

*Ces remerciements vont tout d'abord au corps professoral et administratif du département d'informatique, pour la richesse et la qualité de leur enseignement et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.*

*Nous tenons à remercier sincèrement Monsieur Kouahla Mohamed Nadjib qui, en tant que Directeurs de mémoire, qui est toujours montrés à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'ils ont bien voulu nous consacrer et sans qui ce mémoire n'aurait jamais vu le jour.*

*Merci à tous et à toutes.*

# Dédicace



Je dédie ce mémoire à ...

*À la mémoire de mon très cher  
grand-père et ma chère grand-mère parti  
avant de voir son rêve se réaliser, puisse-t-  
il trouver dans ce mémoire concrétisation  
de ses espoirs qu'il avait mis en moi.*

*À mon cher papa: aucune dédicace  
ne saurait être assez éloquente pour  
exprimer ce que tu mérites pour  
tous les sacrifices que tu n'avais  
pas cessé de me donner depuis ma naissance  
, durant mon enfance et même à l'âge adulte.*

*À mes adorables Tantes habiba, houria, mounira, puisse-t-elle  
Trouver dans ce travail l'expression  
de mon grand Amour et ma plus  
Profonde gratitude  
À mon adorable Maman, puisse-t-elle  
Trouver dans ce travail l'expression  
de mon grand Amour et ma plus  
Profonde gratitude*

*À mon frère : Walid ainsi que sa femme Ibtissem et Khalil  
À mes sœurs: Marwa et Lidia  
Pour leur précieuse aide et encouragements*

*Mes pensées vont aussi à tous mes amis: Housseem, Raouf, abdou, basseem,  
Karim,  
tout au long de la réalisation de ce travail, m'ont manifesté leur soutien et leurs  
encouragements.*

# Dédicace



Je dédie ce mémoire à ...

*À la mémoire de mon très cher  
grand-père et ma chère grand-mère parti  
avant de voir son rêve se réaliser, puisse-t-  
il trouver dans ce mémoire concrétisation  
de ses espoirs qu'il avait mis en moi.*

*À mon cher papa: aucune dédicace  
ne saurait être assez éloquente pour  
exprimer ce que tu mérites pour  
tous les sacrifices que tu n'avais  
pas cessé de me donner depuis ma naissance  
, durant mon enfance et même à l'âge adulte.*

*À mes adorables Maman, puisse-t-elle  
Trouver dans ce travail l'expression  
de mon grand Amour et ma plus  
Profonde gratitude*

*À mes sœurs: Imene , Mejda et Sirine  
Pour leur précieuse aide et encouragements*

*Mes pensées vont aussi à tous mes amis: Housseem, Chemssou, Dadi, Saad,  
tout au long de la réalisation de ce travail, m'ont manifesté leur soutien et leurs  
encouragements.*

## Résumé

Dans ce mémoire, nous avons conçu un algorithme de compression en utilisant les ondelettes. Afin de résoudre les soucis de taille, qualité des données et temps de traitement, on a utilisé plusieurs paramètres. Une comparaison a été faite avec les résultats obtenus l'an dernier de compression JPEG avec la DCT.

Des tests expérimentaux ont permis de valider l'efficacité de cette méthode et d'évaluer ses performances déduisant ainsi qu'elle conduit aux meilleurs résultats possibles.

Malgré les résultats obtenues et en prenant en compte l'évolution constante dans le domaine des technologies multimédias, cet algorithme, mais aussi tous les algorithmes connus à nos jours ne satisfont pas les critères de temps et de qualité.

# *Table des matières*

Liste des figures .....	3
Liste des tableaux .....	4
Introduction Générale .....	5
Chapitre 1 : Compression Multimédia.....	7
1. Introduction .....	7
2. L'image .....	7
2.1 Caractéristiques d'une image.....	7
2.2 Généralités .....	7
3. La compression.....	9
3.1 Définition .....	9
3.2 Historique .....	9
3.3 Avantages de la compression .....	11
3.4 Inconvénients de la compression.....	11
3.5 Taux de compression.....	12
3.6 Différents types de compression.....	12
3.7 Compression JPEG .....	14
3.8 La qualité de compression.....	17
4. Conclusion.....	18
Chapitre 2 : Analyses Multi résolution et Transformée en ondelettes.....	19
1 Introduction .....	19
2 L'analyse multi résolution.....	19
2.1 Définition .....	19
2.2 Explication .....	19
2.3 Principe.....	20
3 Transformée en Cosinus Discrète par Bloc 8x8(DCT) .....	21
3.1 Rappels mathématiques.....	21
3.2 Pourquoi la DCT? .....	22
3.3 Algorithme de calcul pour la DCT .....	24
4 La compression en ondelette (DWT) .....	24
4.1 Définition.....	24

4.2	Historique .....	25
4.3	Avantage des ondelettes.....	25
4.4	Choix d'ondelettes.....	25
5	Conclusion.....	26
Chapitre 3 : Conception d'un système de compression.....		27
1.	Introduction .....	27
2.	Algorithme de compression d'images numériques .....	27
2.1.	Description de l'algorithme .....	27
2.2.	Description .....	27
2.3.	Application sur les composantes YCbCr .....	28
3.	Description des différentes fonctions mathématiques utilisées.....	31
3.1.	Description détaillée.....	31
4.	Implémentation de l'algorithme .....	34
4.1.	Environnement de travail.....	34
4.2.	Les classes utilisées.....	34
4.3.	Structure du programme.....	35
Chapitre 4 : Tests et résultats.....		37
1.	Introduction .....	37
2.	Tests de l'algorithme de compression numérique.....	37
2.1	Tests des performances de l'algorithme avec un autre algorithme.....	37
3.	Comparaison des résultats obtenus.....	38
3.1.	Les images utilisées .....	38
4.	Conclusion.....	53
Conclusion Générale.....		54
Bibliographie : .....		55

## Liste des figures

1.1 Présentation Compression avec perte dans JPEG.....	13
1.2 Graphe qui présente la qualité en variation avec la taille.....	14
1.3 Schéma générale de la compression d'image.....	14
1.4 Etapes de la compression JPEG.....	16
2.1 Représentation du passage d'une échelle à l'échelle plus petite suivante.....	21
2.2 Décroissance des coefficients.....	23
3.1 Etape de l'algorithme basé sur les composants RVB.....	28
3.2 Etapes de l'algorithme basé sur les composants YCbCr.....	30
3.3 Schéma général de la décomposition en ondelettes.....	32
3.4 Décomposition d'image par ondelettes.....	33
3.5 Affichage de la décomposition par le nombre d'itération.....	34
3.6 Reconstruction de l'image par ondelettes.....	34
4.1 Image Originale.....	39
4.2 Image compressé avec DCT Taille = 10 ko.....	39
4.3 Image compressé avec DWT Taille = 29 ko.....	40
4.4 Image Originale.....	41
4.5 Image compressé avec DCT.....	42
4.6 Image compressé avec DWT.....	43
4.7 Image Originale.....	44
4.8 Image compressé avec DCT.....	45
4.9 Image compressé avec DWT.....	46
4.10 Image Originale.....	49
4.11 Image compressé avec DCT.....	50
4.12 Image compressé avec DWT.....	51

## Liste des tableaux

1.1 Tableau de code morse .....	10
1.2 Définition d'intervalles de qualité d'image en PSNR(dB). .....	18
4.1 Taille des fichiers après chaque transformée .....	47
4.2 Les résultats des images testés en termes de vitesse de traitement .....	47
4.3 les résultats des images testés en termes de qualité d'image .....	48
4.4 Taille des fichiers après chaque transformée .....	52
4.5 Temps de traitement pour chaque transformée .....	52
4.6 Les résultats des images testés en termes de qualité d'image .....	52

## Introduction Générale

Soucieux de son confort, l'homme n'hésite pas à exploiter à l'épuisement tous les moyens qui peuvent lui rendre la vie plus simple et plus facile. Parmi ces moyens la communication qui se dresse en tête de liste, celle-ci se caractérise par une importante consommation de l'information.

La compression numérique devient alors une nécessité afin d'assurer l'archivage d'une part et de faciliter leur transmission d'autre part.

La formidable avancée technologique de ces dernières années, la baisse des coûts du matériel et l'accès à des plateformes performantes pour le grand public incite les fournisseurs de contenus à proposer de plus en plus de choix dans le mode de transmission pour l'image numérique.

En outre, le principe général de toute compression est la réduction de la taille des données, en construisant des messages qui résumeront l'information initiale par l'élimination des redondances.

Aujourd'hui, on distingue deux principaux types de compression qui semblent destinés à connaître un bel avenir et susceptibles de remplacer les formats actuels (en particulier la norme JPEG). Ces deux types de compression sont les Fractales et la compression par Ondelettes.

Le succès de ce dernier type de méthode s'explique par le fait que la décomposition en ondelettes de la plupart des signaux ne fait apparaître des coefficients significatifs qu'à un petit nombre de positions dans le plan temps-échelle, c'est-à-dire que l'énergie est localisée préférentiellement à certaines fréquences et certaines positions. Cette caractéristique est utile pour la compression d'image car un petit nombre de coefficients suffisent à en reconstruire l'essentiel.

La transformation par ondelettes ne compresse absolument rien. La compression vient plus tard, en utilisant d'autres procédés de quantification et de codage entropique.

Dans ce projet de fin d'études, nous allons exploiter ces différents domaines informatiques en plusieurs chapitres :

Le chapitre introductif de ce mémoire exploitera la compression ainsi que son application dans le domaine multimédia.

Le deuxième chapitre discutera l'analyse multi résolution, ensuite un rappel sur la transformé en cosinus discrète ainsi que l'utilisation des ondelettes.

Le troisième chapitre est consacré à la description d'un algorithme de compression d'image en utilisant le principe des ondelettes,

Le Quatrième chapitre est basé sur les différents tests appliqués sur les algorithmes conçu.

Enfin, nous clôturons ce mémoire par une conclusion générale en résumant les différents principes et techniques développés pour étudier l'aspect de compression des données multimédia en utilisant les ondelettes.

# Chapitre 1 : Compression Multimédia

## 1. Introduction

La compression de données est une activité ancienne, l'utilisation d'abréviations est une preuve. Les langues elles-mêmes utilisent des mots de longueurs variées, les plus fréquents étant les plus courts, afin de réduire la taille des phrases. La compression des images numériques, C'est elle qui a permis la diffusion des images sur Internet ou encore la propagation des appareils photos numériques. Elle constitue également la base de la compression vidéo.

L'image devient vite lourde de données, donc excessivement longue à transmettre et à traiter. Voilà pourquoi depuis quelques années, les centres de recherche en informatique dépensent de nombreuses heures sur des algorithmes de compression. Afin de limiter la taille, ou le poids d'une image, nous devons la compresser, c'est-à-dire éliminer les informations inintéressantes ou redondantes. Il existe de nos jours plus d'une vingtaine de formats de compression, spécifiquement dans la compression d'image (.gif, .jpeg, .bmp...).

## 2. L'image

### 2.1 Caractéristiques d'une image

Pour comprendre comment fonctionne la compression d'image, nous devons tout d'abord savoir ce qu'est une image, quelles sont les différentes représentations informatiques, par quels moyen peut-on réduire la taille des fichiers, comment représenter les couleurs bref autant de choses qu'il est nécessaire de détailler dans une première partie.

### 2.2 Généralités

#### a - Notion de pixel

En informatique, et ce pour des raisons de gain de place, une image est composée d'un ensemble de points, appelés pixel (abréviation de PICTURE Element). Ces pixels sont regroupés en lignes et en colonnes afin de former un espace à deux dimensions. Chaque point sera représenté par ses coordonnées (X, Y), avec X l'axe orienté de gauche à droite, et Y l'axe orienté de haut en bas [1].

## b - Représentation de la couleur

En plus de ses coordonnées planaires, un pixel se caractérise par sa pondération, appelée aussi profondeur de codage, qui représente sa couleur ou son intensité. Cette valeur peut être codée sur un nombre  $n$  différent de bits (ou octet) selon les méthodes de codage de la couleur utilisées. Les standards les plus répandus sont  $n=1$  bit (noir ou blanc),  $n=4$  (16 couleurs ou 16 niveaux de gris),  $n=8$  bits (256 couleurs ou 256 niveaux de gris) ...

On appelle la palette de couleur, l'ensemble des couleurs que peut contenir une image. Il est fréquent de voir des images qui n'utilisent jamais certaines couleurs, il devient dès lors intéressant de limiter la palette de couleur en ne sélectionnant que la ou les couleurs utilisées réellement par l'image. Plusieurs types d'images existant les plus connues sont :

- Les images binaires (noir ou blanc)

Image la plus simple, un pixel peut prendre uniquement les valeurs noir ou blanc. C'est typiquement le type d'image que l'on utilise pour scanner du texte quand celui-ci est composé d'une seule couleur.

- Les images en niveaux de gris

En général, les images en niveaux de gris renferment 256 teintes de gris. Par convention la valeur zéro représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (intensité lumineuse maximale). Le nombre 256 est lié à la quantification de l'image. En effet chaque entier représentant un niveau de gris est codé sur 8 bits. Il est donc compris entre 0 et  $2^8 - 1 = 255$ . C'est la quantification la plus courante. On peut coder une image en niveaux de gris sur 16 bits ( $0 < n < 2^{16} - 1$ ) ou sur 2 bits : dans ce dernier cas le « niveau de gris » vaut 0 ou 1 : il s'agit alors d'une image binaire (Noir et Blanc).

- Les images couleurs

L'espace couleur est basé sur la synthèse additive des couleurs, c'est à dire que le mélange de trois composantes (par exemple (R, V, B)) donne une couleur. Un pixel est codé par trois valeurs numériques. La signification de ces valeurs dépend du type de codage choisi. Le plus utilisé pour la manipulation des images numériques est

l'espace couleur Rouge, Vert, Bleu (R, V, B) (RGB en anglais). La restitution des couleurs sur écran utilise cette représentation (synthèse additive).

### **c - Taille et définition d'une image**

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

Prenons l'exemple d'une image 1024 x 768, dont la couleur est codée sur 24 bits (1 octet pour les nuances de rouge, 1 pour le bleu et 1 octet pour le vert, codage True color ou RGB)

Nombre de pixels :

$$1024 \times 768 = 786432 \text{ pixels ;}$$

Poids de l'image :

$$786432 \times 3 = 2359296 \text{ octets ;}$$

Soit une image de  $2359296 / 1024 = 2304$  Ko, ou  $2304 / 1024 = 2,25$  Mo, ce qui est assez conséquent, surtout lorsqu'on veut transmettre l'image [2].

## **3. La compression**

### **3.1 Définition**

Les méthodes de compression et de codage réduisent le nombre de bits par pixel à stocker ou à transmettre, en exploitant la redondance informationnelle dans l'image.

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

- La qualité de reconstitution de l'image.
- Le taux de compression.
- La rapidité du codeur et décodeur (codec) [3].

### **3.2 Historique**

A- Le code Morse

Un des exemples les plus anciens de compression, c'est le code Morse, inventé par Samuel Finley Breese Morse (1791-1872). Le code fut conçu dès 1837, mais ne connut le succès qu'après les années 1845. Le code Morse est composé de trois symboles : le point(.), le trait(-) et une pause servant à délimiter les codes. Dans le code Morse, chaque lettre et chiffre est représenté par une série de points et de traits, terminés par une pause. On transmettait les messages traduits en code Morse par le télégraphe, appareil électromagnétique capable de représenter deux états, ouverts ou fermés. Quand l'appareil recevait un signal électrique, il émettait un son, et quand il ne recevait aucun signal, il demeurait muet. Par convention, un trait durait trois fois plus long temps qu'un point ou une pause, ce qui permettait de les différencier facilement [4].

A	· - -	N	- - ·	0	- - - - -
B	- · · ·	O	- - -	1	· - - - -
C	- · · · ·	P	· - - -	2	· · - - -
D	- · ·	Q	- - - ·	3	· · · - -
E	·	R	· - · ·	4	· · · · -
F	· · - · ·	S	· · · ·	5	· · · · ·
G	- - · ·	T	-	6	- · · · ·
H	· · · ·	U	· · -	7	- - - · ·
I	· ·	V	· · · -	8	- - - · · ·
J	· - - - -	W	· - - -	9	- - - - ·
K	- · - -	X	- · · - -	.	· - - - - -
L	· - · · ·	Y	- - - - -	,	- - - - - -
M	- -	Z	- - - · ·	?	· · - - - ·

Tableau 1.1 Tableau de code morse.

#### B- Le code Braille :

Le braille a été inventé à Paris en 1829 par Louis Braille. Louis Braille a perdu la vue à l'âge de trois ans à la suite d'un accident. Jeune écolier, il était ennuyé par les grosses lettres surélevées employées pour enseigner à lire et à écrire. Plus tard, un officier de l'Artillerie française, Charles Barbier de la Serre, lui a mis en tête l'idée de

lire à l'aide d'un code tactile. Après plusieurs années de recherche, Louis Braille a mis au point le système de lecture et d'écriture qui est aujourd'hui utilisé aux quatre coins du globe [5].

### **3.3 Avantages de la compression**

Le principal avantage de la compression est qu'elle réduit les besoins de stockage de données. Elle offre également une approche intéressante pour réduire le coût de la communication dans la transmission de gros volumes de données sur des liaisons long-courriers par une plus grande utilisation efficace de la bande passante disponible dans les liaisons de données. Ceci facilite considérablement la réduction du coût de la communication en raison de la réduction de débit de données. En raison de la réduction de débit de données, la compression des données augmente également la qualité de la présentation multimédia à travers les canaux de communication à bande passante limitée, en raison du débit de données réduit. Offertes par les techniques de compression, réseau informatique et l'utilisation d'Internet est de plus en plus l'image et graphique conviviale, plutôt que d'être seulement des données et des phénomènes de texte centrée. En bref, la compression de haute performance a créé de nouvelles possibilités d'applications créatives telles que la bibliothèque numérique, archivage numérique, la vidéoconférence, la télémédecine et de divertissement numérique pour n'en nommer que quelques-uns. Il existe de nombreux autres avantages secondaires dans la compression de données. Par exemple, il a de grandes implications en matière d'accès de base de données [6].

### **3.4 Inconvénients de la compression**

La compression de données présente aussi des inconvénients, en fonction du domaine d'application et de la sensibilité des données. Par exemple, la charge supplémentaire encourue par le processus de codage et de décodage est l'un des inconvénients les plus graves de la compression des données, ce qui décourage son utilisation dans certaines régions. Cette charge supplémentaire est généralement nécessaire afin d'identifier ou d'interpréter de façon unique les données compressées. La compression des données réduit généralement la fiabilité des enregistrements de données. Par exemple, une erreur d'un seul bit dans le code comprimé entraîne des données incorrectes [7].

### 3.5 Taux de compression

Le taux de compression est un outil essentiel dans la compression. Il nous indique le pourcentage de réduction de la taille du fichier original.

Ce critère est d'importance puisqu'il est directement dépendant de la technique de compression utilisée. Certaines méthodes proposent des taux de compression fixes (entré par exemple par l'utilisateur) ou non. On parlera alors de codage progressif où il est possible d'encoder mais surtout de décoder une image avec divers taux de compression. Afin de définir le taux de compression à proprement dit, nous introduisons au préalable la notion de débit comme étant le nombre moyen de bits par pixels nécessaires à la description de l'image.

$$\text{Débit} = \frac{\text{Taille de fichier(bits)}}{\text{Nombre de pixels}} \quad \text{en bits par pixel(bpp)}$$

Une image numérique est composée d'un ensemble de pixels présentant un nombre fixe d'états (niveaux de gris par exemple). A chacun de ces éléments est attribué un symbole ou code binaire. Pour les données RSO issues du capteur RAMSES, chaque pixel est composé de 2 réels, ce qui requiert 64 bits par pixel.

Le taux de compression  $\eta$  est défini comme le rapport entre le nombre de bits par pixels de l'image initiale et l'image comprimée (rapport des débits).

$$\eta = \frac{\text{Débit de l'image avant compression}}{\text{Débit de l'image après compression}}$$

Pour les méthodes de codage sans perte, le taux de compression est de l'ordre de 2 à 3 sur des images naturelles. Concernant les images RSO, le taux obtenu est encore plus faible à cause d'une plus forte dé-corrélation des pixels [8].

### 3.6 Différents types de compression

L'enjeu de la recherche sur la compression d'image est de trouver un moyen de diminuer la taille d'une image, tout en essayant de limiter la dégradation due à la compression.

Il existe deux grandes familles d'algorithmes de compression : compression sans perte (non-destructrice) et compression avec perte (destructrice) [9].

a- La compression sans perte

Une compression est dite « Sans perte » lorsqu'on peut obtenir les données initiales à partir des données compressées sans qu'il n'y ait eu aucune altération. Les programmes informatiques, codes sources, et rapports sont toujours encodés de cette façon [9].

b- La compression avec pertes

Il est question de compression « Avec pertes » lorsque certaines informations sont volontairement supprimées, par exemple les détails d'une image, ou les fréquences inaudibles pour un fichier sonore. Certains éléments sont quasiment imperceptibles par l'humain (par exemple les très basses ou hautes fréquences audio), il est alors intéressant de profiter de cette « faiblesse » pour réduire la taille du fichier. La qualité finale du média dépendra donc de la quantité de données altérée, le but étant de trouver un bon compromis entre « Qualité » et « Taux de compression » [9].



JPEG (100%) 188,227 Ko

JPEG (50%) 23,591 Ko

JPEG (1%) 4,449 Ko

Figure 1.1 Présentation Compression avec perte dans JPEG.

La figure 1.2 présente un exemple de compression avec perte dans JPEG, on permet de voir un taux de compression impressionnant avec une qualité d'image acceptable.

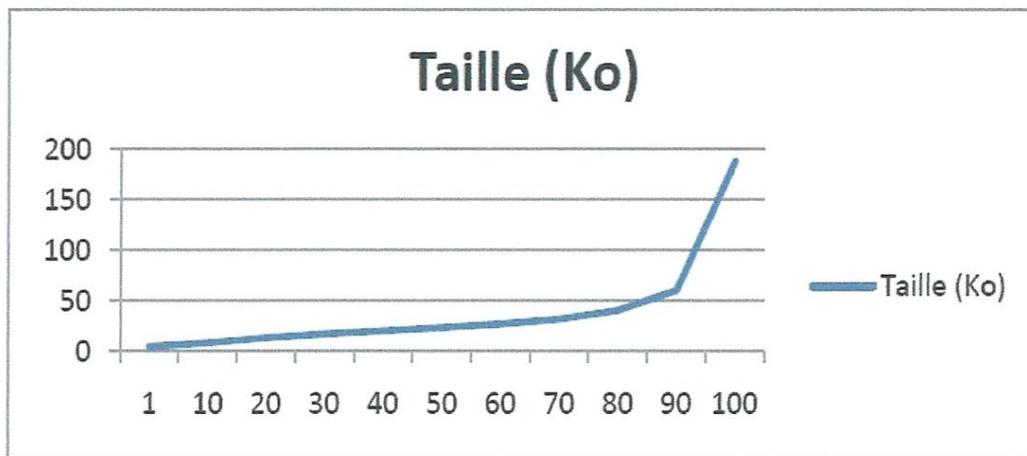


Figure 1.2 Graphe qui présente la qualité en variation avec la taille

La qualité diminue beaucoup plus vite que le gain en taille réalisé, un bon compromis est donc d'utiliser une qualité d'environ 75% lors de la compression au format JPEG.

### 3.7 Compression JPEG

a - schéma générale de la compression :

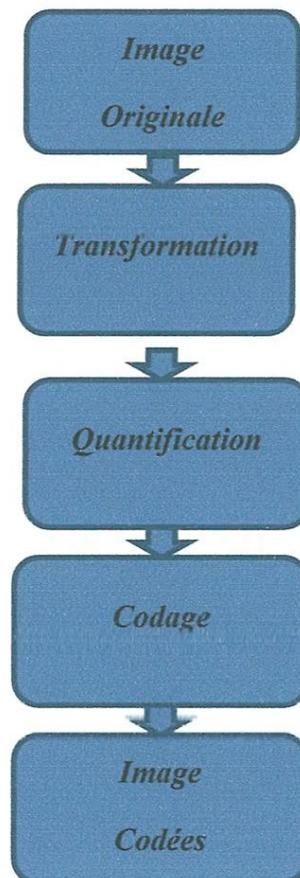


Figure 1.3 Schéma générale de la compression d'image.

A partir de ce schéma, nous allons revoir chacune de ses étapes à fin de préciser leur rôle.

#### ❖ Transformation

La dépendance existante entre chacun des pixels et ses voisins (la luminosité varie très peu d'un pixel à un pixel voisin) traduisent une corrélation très forte sur l'image. On essaie donc de tirer parti de cette corrélation, pour réduire le volume d'information en effectuant une opération de dé-corrélation des pixels.

La dé-corrélation consiste à transformer les pixels initiaux en un ensemble de coefficients moins corrélés, c'est une opération réversible [10].

#### ❖ Quantification

La quantification des coefficients a pour but de réduire le nombre de bits nécessaires pour leurs représentations. 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. Elle peut être scalaire ou vectorielle. Un des résultats fondamentaux des travaux de Shannon concernant la relation : (débit /distorsion) montrent que l'on obtient de meilleures performances en utilisant la quantification vectorielle [11].

#### ❖ Codage

Une fois les coefficients quantifiés, ils sont codés. Un codeur doit satisfaire a priori les deux conditions suivantes :

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

A partir de ce schéma fonctionnel qui nous avons explicité, nous passons à la classification des méthodes de compression d'image numériques [12].

#### b - La méthode JPEG:

Le JPEG (Joint Photographic Experts Group Standart) est une méthode sophistiquée de compression avec perte pour la couleur ou les nuances de gris. Il ne manipule pas les images noires et blanches mais des images à deux niveaux. Il travaille également mieux sur des images à ton continu, où les pixels adjacents ont les couleurs semblables. Un avantage de JPEG est l'utilisation de beaucoup de paramètres, permettant à l'utilisateur d'ajuster la quantité des données à perdre (et ainsi aussi le rapport de compression) sur un éventail. Souvent, il ne peut voir aucune dégradation

d'image même aux rapports de compression élevés. Il existe une autre méthode JPEG mais sans perte, utilisée dans des domaines compliqués comme la médecine.

c - Etapes de la compression JPEG :

Le schéma 1.4 montre les étapes principales de la méthode JPEG.

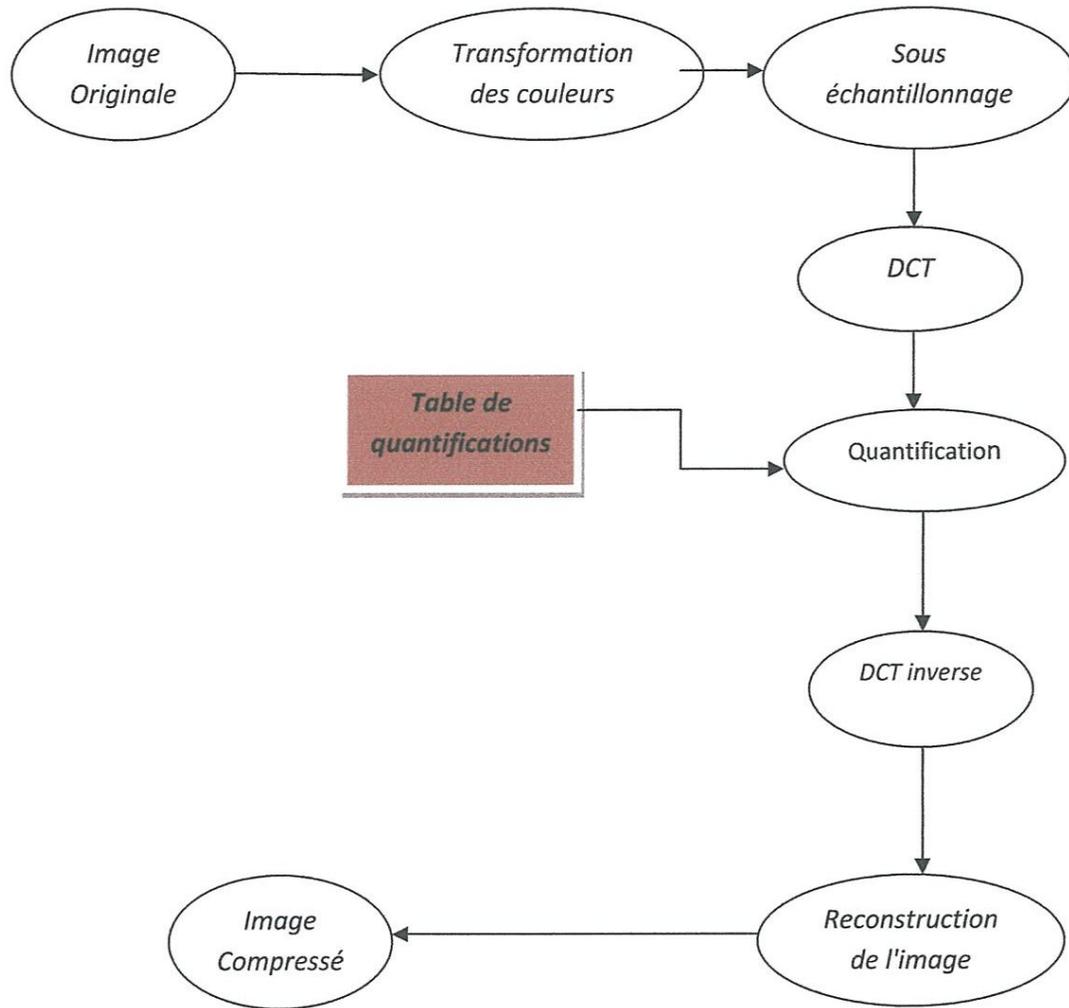


Figure. 1.4 .Etapes de la compression JPEG.

D'après le schéma, les principales étapes de la compression JPEG sont les suivantes :

- Ré-échantillonnage de la chrominance ou partitionnement, car l'œil ne peut discerner de différences de chrominance au sein d'un carré de 2x2 points.
- Découpage de l'image en blocs de 8x8 points, puis l'application de la fonction DCT.

- Quantification de chaque bloc, c'est-à-dire qu'il applique un coefficient de perte (qui permet de déterminer le ratio taille/qualité) qui annulera ou diminuera des valeurs de hautes fréquences. Le format de fichier embarquant un flux codé en JPEG est en réalité appelé JFIF (JPEG File Interchange Format, soit en français Format d'échange de fichiers JPEG), mais par déformation le terme de fichier JPEG est couramment utilisé [13].

### 3.8 La qualité de compression

L'évaluation de la qualité de l'image est très demandée à l'heure actuelle dans diverses applications de communications multimédias.

Les divers traitements de codage et de compression subis par les images, ainsi que l'impact des réseaux de communications qui sont traversés durant la transmission des données multimédias, introduisent des distorsions dans l'image et altèrent sa qualité [14]. Les méthodes d'évaluation subjective de la qualité basées sur l'opinion moyenne d'observateurs MOS (MOS : Mean Score/Opinion Observer) peuvent s'avérer efficaces en termes de précision de la qualité estimée, mais le processus d'évaluation associé à cette démarche est coûteux, car il nécessite du temps, des ressources humaines et une infrastructure adéquate. Les méthodes d'évaluations objectives de la qualité utilisent des métriques pour calculer mathématiquement la qualité de l'image [15]. Le PSNR (PSNR : Peak Signal to Noise Ratio) ou rapport signal à bruit crête est une métrique qui se base sur le calcul de l'erreur quadratique moyenne (EQM) entre les pixels de l'image de référence I et l'image distordue I' selon la formule suivante :

$$\text{PSNR} = 10 \log_{10} \left( \frac{255}{\text{Ems}} \right)$$

$$\text{EQM} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I_o(i,j) - I_r(i,j)\|^2$$

M et N étant le nombre de lignes et de colonnes des images I et I'. Une valeur de PSNR supérieure à 40 dB est en général un indicateur d'une très bonne qualité d'image (i.e. la dégradation subie est en général imperceptible) [16].

<b>Attribut</b>	<b>Très Bonne</b> (Very Good)	<b>Bonne</b> (Good)	<b>Acceptable</b> (Acceptable)	<b>Mauvaise</b> (Poor)
<b>PSNR (dB)</b>	$\geq 40$	40-35	30-35	$< 30$

Tableau 1.2 : Définition d'intervalles de qualité d'image en PSNR(dB).

#### **4. Conclusion**

La compression des images est désignée à prendre un rôle encore plus important en raison de l'évolution des réseaux et du multimédia. Son importance est surtout due au décalage qui existe entre les possibilités matérielles des dispositifs que nous utilisons (capacité des mémoires de masse, débits sur Internet, ...) et les besoins qu'expriment les utilisateurs (visiophonie, vidéo plein écran, transfert de quantités d'informations toujours plus importantes dans des délais toujours plus brefs.

# Chapitre 2 : Analyses Multi résolution et Transformée en ondelettes

## 1 Introduction

Qui a déjà téléchargé des images sur son ordinateur sait à quel point l'opération peut se révéler longue. D'où la nécessité de réduire la taille des fichiers informatiques. La nouvelle référence en matière de compression est issue d'un concept mathématique récent : les ondelettes.

## 2 L'analyse multi résolution

### 2.1 Définition

Toutes les bases d'ondelettes connues à ce jour sont données par la méthode AMR. L'AMR de l'espace des fonctions de carrés sommables consiste à le découper en une suite croissante de sous-espaces vectoriels fermés. Ces sous-espaces ont un certain nombre de propriétés : ils forment une suite emboîtée, leur intersection est réduite et leur réunion est dense. Chaque sous-espace est l'ensemble de toutes les approximations possibles d'un même signal à l'échelle associée au sous-espace. Le signal à analyser sera approximé par une succession de projections orthogonales sur les sous-espaces [17].

### 2.2 Explication

Pour construire un gros objet, par exemple un gros bonhomme, on place d'abord les grosses briques pour avoir une version grossière du personnage. Quelques grosses briques pour le corps, les bras, les jambes et la tête, puis on ajoute des briques moyennes pour le cou, les pieds, les mains, les articulations, puis des briques un peu plus petites pour les yeux, le nez, les doigts, puis éventuellement des toutes petites pour figurer les détails. Cette approche qui permet d'utiliser un nombre d'éléments très réduit est appelé multi résolution, car on construit un objet d'abord de manière grossière puis de plus en plus précise en augmentant graduellement la résolution. C'est dans cet esprit qu'ont été inventées les ondelettes qui sont des ensembles de pixels de différentes tailles. Pour être précis, les ondelettes ne sont pas des gros blocks rectangulaires mais des paquets plus ou moins gros qui ont la forme d'une vague, ou

d'une onde d'où leur nom. Les grosses ondelettes vont permettre de construire une version grossière de l'image et les petites ondelettes vont apporter des détails plus fins [18].

### **2.3 Principe**

On veut trouver une base orthonormale d'ondelettes dans laquelle il sera possible de décomposer le signal. Pour cela, on peut utiliser l'analyse multi résolution. Une telle analyse consiste à décomposer le signal sur une gamme très étendue d'échelles, opération que l'on peut comparer à une cartographie. A chaque échelle, le signal sera remplacé par l'approximation la plus adéquate que l'on puisse y tracer. En allant des échelles les plus grossières vers les échelles les plus fines, on accède à des représentations de plus en plus précises du signal donné. L'analyse s'effectue en calculant ce qui diffère d'une échelle à l'autre, c'est-à-dire les détails à une résolution donnée. Ceux-ci permettent, en corrigeant une approximation encore assez grossière, d'accéder à une représentation d'une qualité meilleure.

Les ondelettes correspondent à des degrés de résolutions: elles sont définies par la différence entre 2 fonctions échelles consécutives. Elles permettent de représenter les détails gagnés lors du passage d'une échelle à l'échelle plus petite suivante [19].

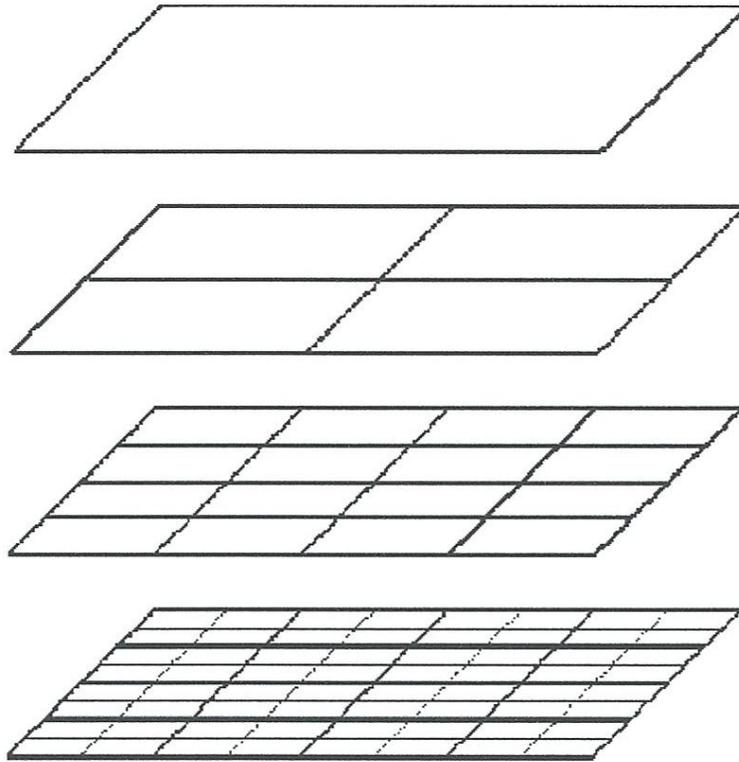


Figure 2.1 Représentation du passage d'une échelle à l'échelle plus petite suivante.

### 3 Transformée en Cosinus Discrète par Bloc 8x8(DCT)

#### 3.1 Rappels mathématiques

Le passage par la DCT a été l'idée majeure pour la compression JPEG. En effet ce processus appartient à une classe d'opérations mathématiques, tout comme la Transformée de Fourier. Elle permet un changement de domaine d'étude, tout en gardant exactement la même fonction étudiée. Dans notre cas, on étudie une image, c'est à dire une fonction à 3 dimensions : X et Y, indiquant le pixel, et Z avec la valeur du pixel en ce point. Dans le cas d'une image couleur, il faut donc considérer indépendamment 3 fonctions, pour chacun des canaux RGB [20].

L'application de la DCT, ou d'une Transformée de Fourier fait passer l'information de l'image du domaine spatial en une représentation identique dans le domaine fréquentiel. Pourquoi ce changement de domaine est-il si intéressant? Justement 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,

ceux-ci sont en général minimales dans une image. Ainsi on parvient à représenter l'intégralité de l'information de l'image sur très peu de coefficients, correspondant à des fréquences plutôt basses. La composante continue (valeur moyenne de l'image traitée) ayant une grande importance pour l'œil.

La DCT s'applique à une matrice carrée. Le résultat fournit est représenté dans une matrice de même dimension. Les basses fréquences se trouvant en haut à gauche de la matrice, et les hautes fréquences en bas à droite.

La transformation matricielle DCT étant Orthogonale, elle s'accompagne d'une méthode d'inversion pour pouvoir revenir dans le domaine spatial. Ainsi après avoir fait des modifications dans le domaine fréquentiel, éliminer des variations de l'image quasiment invisibles par l'œil humain, on retourne à une représentation sous forme de pixels.

Formule pour calculer la DCT sur une matrice NxN

$$DCT(i, j) = \frac{1}{\sqrt{2}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{pixel}(x, y) \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cos\left(\frac{(2y+1)j\pi}{2N}\right)$$

Formule pour calculer la IDCT sur une matrice NxN

$$\text{pixel}(x, y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i)C(j) DCT(i, j) \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cos\left(\frac{(2y+1)j\pi}{2N}\right)$$

$$C(x) = 1/\sqrt{2}$$

Si x vaut 0, et 1 si x>0 [20].

### 3.2 Pourquoi la DCT?

La DCT étant dans la même classe d'outils mathématiques que la Transformée de Fourier. Alors pourquoi les membres du groupe JPEG ont-ils fait le choix de la DCT? Ces deux méthodes permettent une décomposition de l'information dans une autre base : Une base de cosinus, ou la base de Fourier. Cependant, La décomposition dans la base de Fourier soulève plusieurs problèmes : si l'image présente des discontinuités,

alors la décroissance des coefficients de la transformée de Fourier n'est qu'en  $1/K$ ,  $K$  étant l'indice du coefficient [20].

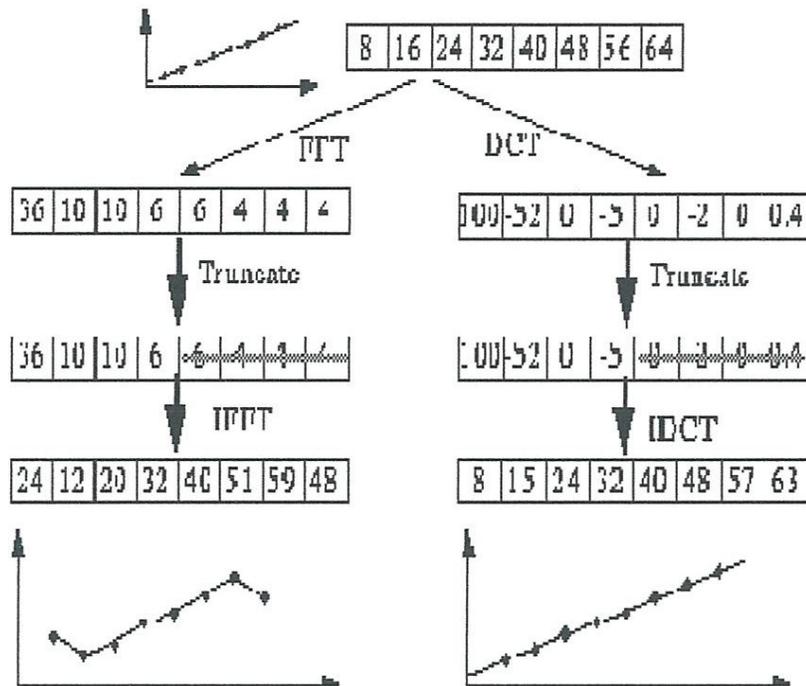


Figure 2.2: Décroissance des coefficients.

Dans cette figure, on se rend compte que pour restituer convenablement l'information, on a besoin de beaucoup plus de coefficients que pour une DCT. La décroissance des coefficients n'étant pas suffisante pour pouvoir négliger rapidement les coefficients de grands indices. De plus en tronquant les derniers coefficients, on risque de voir se produire à la fin le phénomène de Gibbs, qui se traduit par une oscillation au niveau des discontinuités.

D'autre part, la fonction doit être périodique pour la transformée de Fourier, sinon on se retrouve avec une discontinuité au bord. Là encore se posera le phénomène de Gibbs qui risque fort de dégrader l'image.

Le fait de décomposer la fonction sur une base de cosinus fait que la fonction sera symétrique par rapport à  $-\frac{1}{2}$ . Cependant la DCT pose un problème d'optimisation. En effet le calcul d'un coefficient nécessite  $N^2$  multiplications, or il y a  $N^2$  coefficients à calculer. Le coût d'une telle décomposition devient alors démesuré si notre image est de taille  $512 \times 512$ . Ainsi, au lieu de traiter toute l'image, on découpe celle-ci en blocs

8x8. Ce choix représente un compromis performance qualité : en effet, en augmentant la taille de ces blocs, la compression serait meilleure, mais le coût en temps a été jugé trop grand. Sur chacun de ces blocs, on procède ensuite à une DCT.

Cela permet d'avoir un algorithme rapide. Cependant la DCT par blocs 8x8 est justement un des facteurs limitant de la compression JPEG : en effet lorsqu'on augmente la compression, on voit apparaître ces blocs [20].

### 3.3 Algorithme de calcul pour la DCT

Une bonne manière d'implémenter la DCT par blocs serait de créer une matrice de la transformée de cosinus, C.

$$C_{i,j} = \begin{cases} \frac{1}{\sqrt{N}}, & \text{Si } i = 0 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{(2j+1)i\pi}{2N}\right), & \text{Si } i > 0 \end{cases}$$

Une fois cette matrice créée, on crée la matrice Ct qui correspond à la transposée de C. Ensuite la transformée par bloc 8x8 se réduit à deux multiplications matricielles :

$$DCT = C * Pixels * Ct$$

Donc le coût pour calculer la DCT sur un bloc se réduira à 2N multiplications sur des entiers, et 2N additions sur des entiers : cela représente un gain important par rapport à une simple double boucle sur les indices comme pourrait le laisser suggérer la formule initiale de la DCT. On passe d'un coût de  $O(N^2)$  à un coût très intéressant de  $O(N)$  pour calculer UN bloc [20].

## 4 La compression en ondelette (DWT)

### 4.1 Définition

Pour obtenir une image parfaitement identique à l'originale on doit utiliser toutes les ondelettes mais dans la plupart des zones unis il n'y a pas beaucoup de détails et on peut se permettre de ne pas les coder. En pratique une compression en ondelettes

permet de coder précisément les zones complexes et coder grossièrement les zones uniformes. En d'autres termes, on ne code les détails que là où ils sont nécessaires, comme un peintre qui passe plus de temps à réaliser les petits détails que les grands aplats [21].

## **4.2 Historique**

L'histoire des ondelettes commence à la fin des années 1970. Leur découverte est due à un ingénieur d'Elf-Aquitaine, Jean Morlet, alors à la recherche de nouvelles manières d'analyser des signaux issus de relevés sismiques en prospection pétrolière. D'une efficacité redoutable mais extrêmement complexe, l'analyse par ondelettes effraya les dirigeants d'Elf-Aquitaine qui craignaient une escroquerie semblable à celle des « avions renifleurs » censés détecter le pétrole fossile. Invité à prendre une retraite anticipée, Jean Morlet retourna à son laboratoire d'origine de l'Ecole polytechnique de Palaiseau. Là, l'une de ses vieilles connaissances, Roger Balian, professeur de physique statistique, le réorienta vers Alex Grossmann, un physicien théoricien de Marseille. Les deux chercheurs se lancèrent alors dans l'étude de ces étranges objets mathématiques [22].

## **4.3 Avantage des ondelettes**

Les avantages de cette technique sont nombreux. Tout d'abord, on ne perçoit plus de limites de blocs sur l'image finale. Ensuite, il est possible de définir à l'avance le taux de compression. Pour les grosses images, ce dernier peut être très important une image de 50 Mo peut être facilement ramenée à 1 Mo. Enfin, la restitution de l'image la décompression se fait de manière progressive : l'image se reconstitue petit à petit à l'écran, à la manière d'une mise au point, avec un confort visuel non négligeable lorsque l'on télécharge une page Web [23].

## **4.4 Choix d'ondelettes**

Le choix d'ondelettes n'est pas facile car il n'y a pas une ondelette qui soit meilleure qu'une autre. Tout dépend de l'application utilisée. certains disent, l'ondelette la plus simple (Haar) sera optimale. Pour d'autres applications, ce sera le pire des choix possibles. En pratique, il semblerait que l'élément le plus important soit le nombre de moments nuls. Pour la plupart des applications, il est désirable d'avoir le plus de coefficients d'ondelettes nuls et donc plus de moments nuls implique une meilleure transformation. Cependant, les ondelettes ayant un plus grand nombre de moments

ont aussi un support plus grand ce qui signifie que si la fonction ou le signal a des discontinuités brusques, elle sera plus sujette aux phénomènes de Gibbs [24].

## **5 Conclusion**

Les principes de base de l'analyse multi résolution et des ondelettes ont été posés par Meyer (Meyer 1986,1988) pour les aspects mathématiques, puis Stéphane Mallat (Mallat 1989) pour les aspects signaux et images.

Dans ce chapitre nous avons rappelé le principe de la multi résolution avec une définition de ce dernier en passant par la transformé en cosinus discrète avec un rappel mathématiques en situant son algorithme (DCT).

L'une des avantages de la DWT c'est qu'il est possible de définir à l'avance le taux de compression.

# Chapitre 3 : Conception d'un système de compression

## 1. Introduction

Dans le but de minimiser la taille des données multimédias, nous allons décrire nos algorithmes basés sur la compression avec perte vue dans le premier et le deuxième chapitre. Ainsi que l'utilisation des ondelettes vues dans la deuxième partie du même chapitre.

Notre étude se basera sur la comparaison entre DCT (Discrete Cosinus Transformation) et la DWT (Discrete Wavelet Transformation).

## 2. Algorithme de compression d'images numériques

### 2.1. Description de l'algorithme

Nous avons essayé, à partir des différents types de compressions vues dans le premier chapitre, de concevoir un algorithme basé sur un taux élevé de compression avec une perte minimale. Notre algorithme est basé sur les images en couleurs, et cela pour extraire les différents composants de la couleur pour appliquer des fonctions mathématiques bien connues.

### 2.2. Description

Les couleurs dans les images numériques sont connues pour leurs compositions de trois couleurs principales : rouge (R), vert (V) et bleu (B). Ces composantes spécifiées dans les images numériques sont les résultats de travaux faits par Newton dans la spécification des couleurs principales [25]. Grâce à cela, nous avons conçu notre algorithme par les étapes décrites dans le schéma 3.1. Les différentes étapes de l'algorithme sont :

- Extraction des composantes RVB de l'image ;
- Initialisation de la matrice de quantification dépendante d'une valeur définie par l'utilisateur. Les détails de l'initialisation sont détaillés plus loin ;
- Choix de la taille du bloc. La valeur de la taille doit être égale à  $2^i$  ;
- Application pour chaque bloc de chaque composante de l'image une DWT, une quantification et une DWT inverse ;
- Reconstruction de l'image avec les nouvelles composantes reconstruites ;
- Les différentes fonctions mathématiques utilisées seront expliquées plus loin.

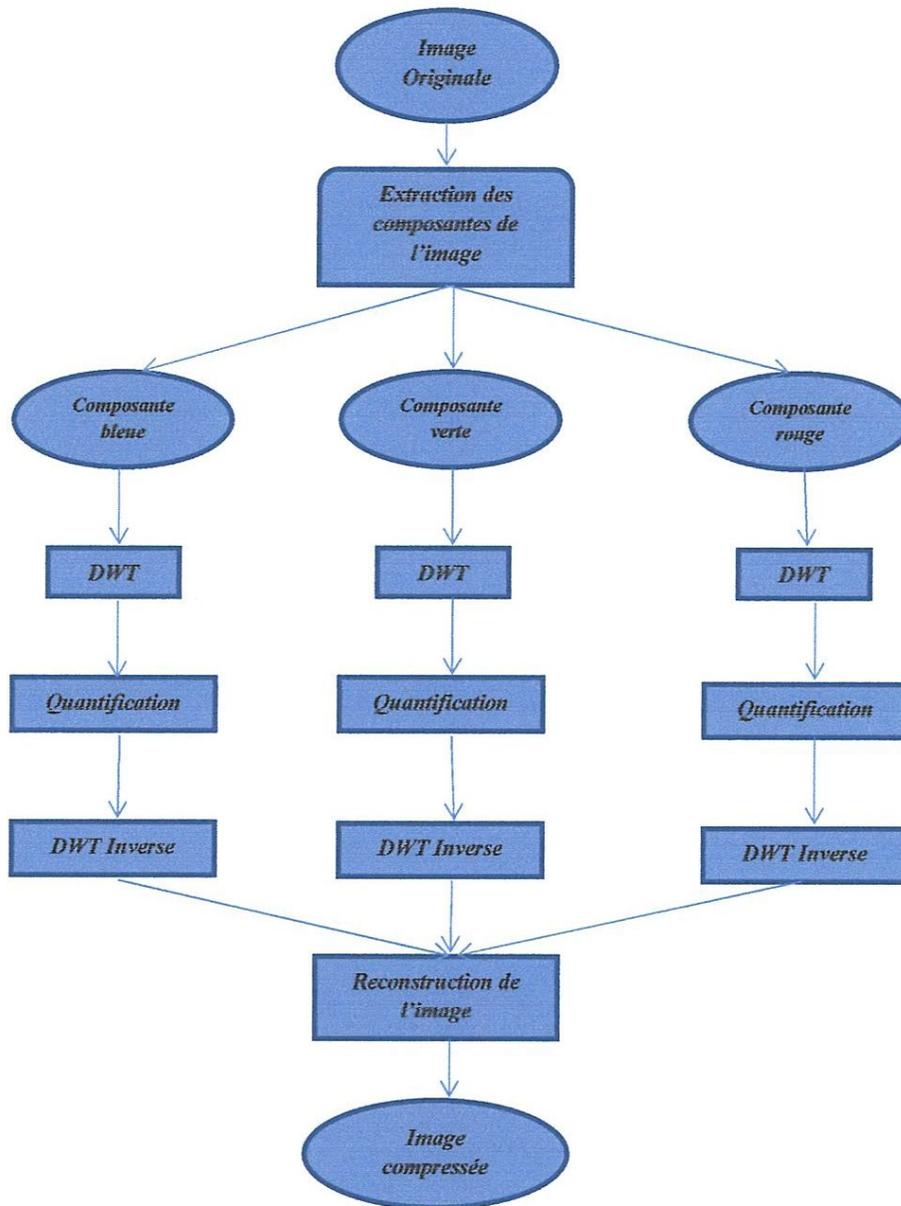


Figure 3.1 Etape de l'algorithme basé sur les composants RVB.

### 2.3. Application sur les composantes YCbCr

Les composantes YCbCr sont des composantes utilisées beaucoup dans la compression multimédia. Chaque composante représente un critère de l'image .

- Le Y représente l'intensité de l'image ;

- Les deux composantes Cb et Cr représentent la chrominance des couleurs bleues(Cb) et rouges (Cr) [25] ;

Pour extraire ces composantes de l'image, nous avons besoin des composantes RVB principales puis appliquer une transformation mathématique. Ces transformations sont comme le suivant [25] :

$$\begin{aligned} Y &= (0.257 * R) + (0.504 * V) + (0.098 * B) + 16 \\ Cb &= - (0.148 * R) - (0.291 * V) + (0.439 * B) + 128 \\ Cr &= (0.439 * R) - (0.368 * V) - (0.071 * B) + 128 \end{aligned}$$

Pour récupérer les valeurs RVB, il suffit d'une transformation inverse comme suit :

$$\begin{aligned} R &= 1.164 * (Y - 16) + 1.596 * (Cr - 128) \\ G &= 1.164 * (Y - 16) - 0.391 * (Cb - 128) - 0.813 * (Cr - 128) \\ B &= 1.164 * (Y - 16) + 2.018 * (Cb - 128) \end{aligned}$$

Cette transformation est pour les raisons suivantes : tout a un rapport avec l'œil humain. En effet, l'œil humain est très sensible aux changements de couleurs, mais peu sensible aux intensités lumineuses de l'image. Et comme la compression des images est en générale avec une perte, cette perte n'influença pas l'œil humain.

Le schéma 3.2 montre les différentes étapes de l'algorithme basé sur les composantes YCbCr.

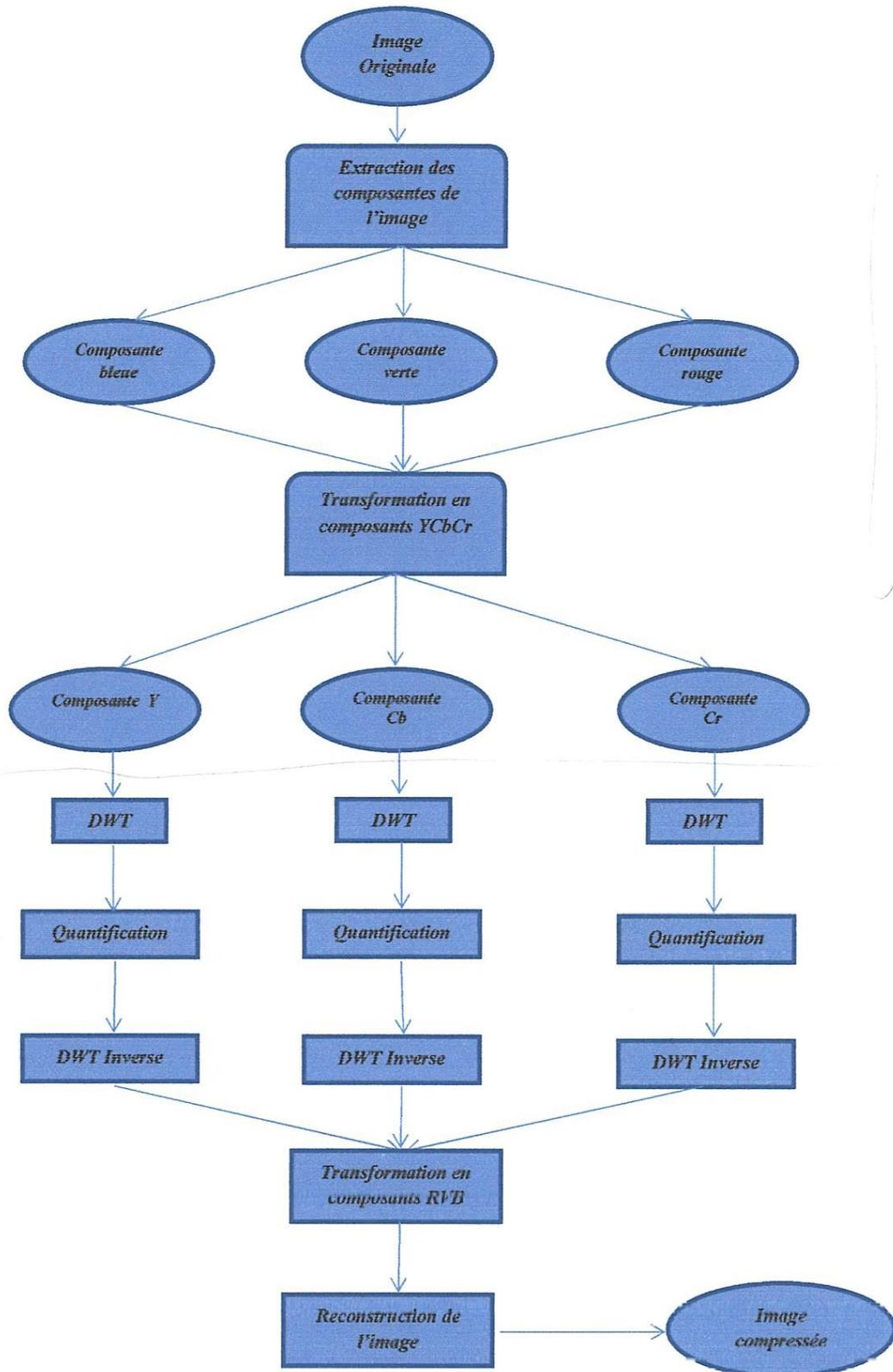


Figure 3.2 – Etapes de l’algorithme basé sur les composants YCbCr.

Les différentes étapes de l'algorithme sont les suivantes :

- Extraction des composantes RVB;
- Transformation des composantes RVB en composantes YCbCr avec les formules décrites auparavant ;
- Application pour chaque canal une DWT, une quantification ainsi qu'une quantification inverse ;
- Transformation inverse des composantes YCbCr en composantes RVB;
- Reconstruction de l'image avec les nouvelles composantes.

### **3. Description des différentes fonctions mathématiques utilisées**

Dans cette partie, nous allons voir les différentes fonctions mathématiques utilisées dans notre algorithme.

Cet algorithme calcule les coefficients d'ondelettes issus de la transformation d'une image au moyen d'une transformée biorthogonale séparable.

Il permet la décomposition de l'image d'origine avec un facteur de résolution égal à 2 et sur J niveau de résolution.

A chaque niveau de résolution sont associées trois sous-images de coefficients d'ondelettes chacune correspondant à une orientation de contour privilégié. L'algorithme de décomposition ainsi que celui de reconstruction associée sont fournis.

La décomposition et la reconstruction sont effectuées par des opérations de corrélations/convolutions séparable à l'aide de filtres numérique. Le sous-échantillonnage et les problèmes de bord sont pris en compte et optimisés dans cette version du logiciel.

#### **3.1. Description détaillée**

Entrées : Image  $N*N$  ; niveau de résolution, filtres associés aux ondelettes

Sorties :  $3*J+1$  sous-image à coder.

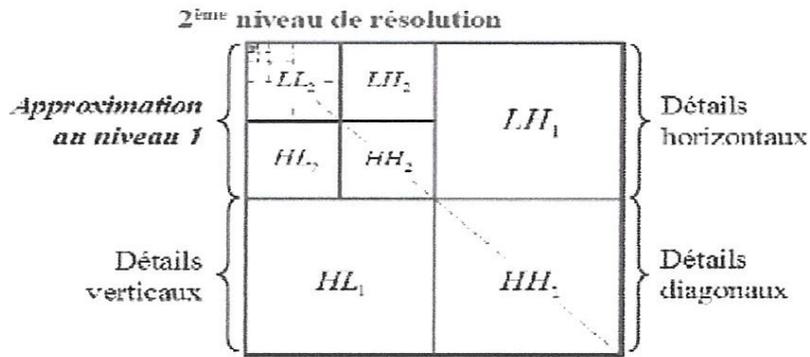


Figure 3.3 Schéma général de la décomposition en ondelettes.

La décomposition s'effectue selon les trois directions des contours de l'image ; horizontale, verticale et diagonale, tout ne conservant constant le nombre de pixels décrivant l'image.

L'image est donc décomposée en une miriade de sous bandes, c'est-à-dire des images de résolution inférieure.

a- Transformée en ondelettes de Haar

Soit deux mesures (l, r). On note à la moyenne et d la différence de ces deux mesures. Pour calculer la transformée en ondelettes de Haar d'une matrice de  $n = 2k$  données, il faut:

- ✓ Moyenner chaque paire de données ( $n / 2$  moyennes).  $a = (l + r) / 2$  = la fonction d'échelle.
- ✓ Calculer la différence entre chaque donnée et sa moyenne ( $n / 2$  différences).

$$d = l - a = a - r = (l - r) / 2 = \text{la fonction d'ondelette}$$

- ✓ placer les moyennes dans la première moitié de la matrice
- ✓ placer les différences dans la seconde moitié de la matrice;
- ✓ répéter le processus sur la première moitié des données (c'est pourquoi le nombre de données n doit être une puissance de 2).

La transformée inverse s'exprime par le calcul suivant:

- ✓  $l = a + d;$

✓  $r = a - d$  ;

b- La quantification :

Après une transformation avec des ondelettes, nous nous retrouvons avec une matrice de coefficients correspondant à différents niveaux de détails. Suite à des constats lors d'études psycho visuelles, nous savons que les hautes fréquences sont moins importantes que les basses fréquences. Voici la formule utilisée pour la quantification :  $q = \text{sign}(v) \cdot (\text{int}(|y| / \Delta b))^q$  est la valeur quantifiée,  $y$  la valeur à quantifier et  $\Delta b$  le pas de quantification.

Pour la dé-quantification, on utilise cette formule :

✓  $z = (q + r \cdot \text{sign}(q)) \cdot \Delta b$

$r$  est une valeur fixée arbitrairement selon chacun, mais elle est souvent proche de 0,5.

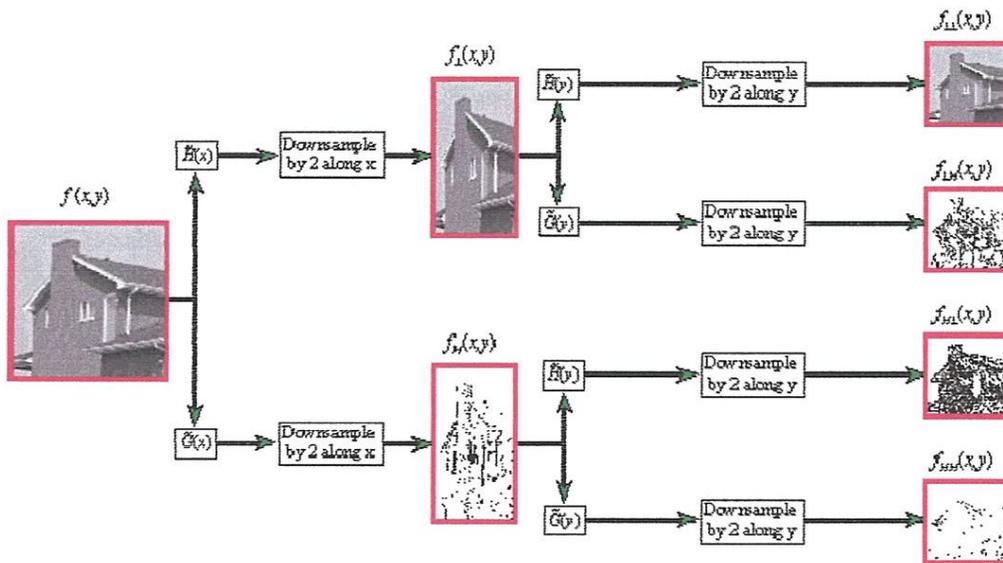


Figure 3.4 décomposition d'image par ondelettes.

Le résultat est une image d'approximation qui a une résolution divisée par 2 et trois images de détail qui donnent les erreurs entre l'image original et l'image d'approximation. Cette transformation est répétée autant de fois que nécessaire pour obtenir le nombre voulu de sous-bandes [26].

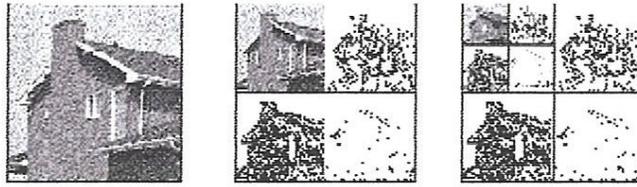


Figure 3.5 affichage de la décomposition par le nombre d'itération.

Schéma général de la reconstruction :

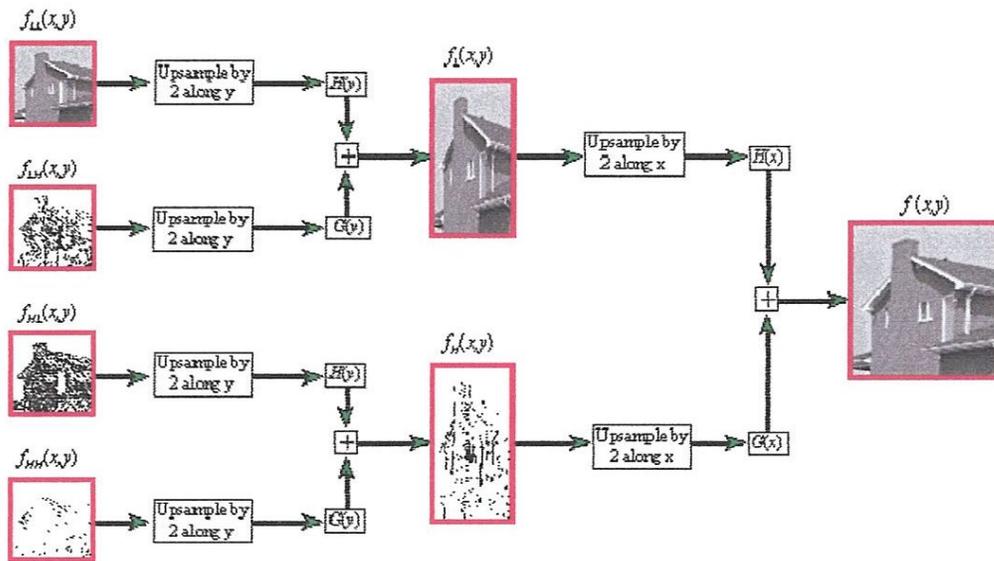


Figure 3.6 reconstruction de l'image par ondelettes.

## 4. Implémentation de l'algorithme

### 4.1. Environnement de travail

Le langage de programmation utilisé pour l'implémentation de ces algorithmes est le langage Java 1.06 version 64 bits sous l'environnement Eclipse Kepler version 64 bits.

### 4.2. Les classes utilisées

Le langage Java contient plusieurs classes prêtes à être utilisées par l'utilisateur. Pour notre algorithme nous avons utilisé les classes suivantes :

- java.awt.image.PixelGrabber: Classe permettant de convertir une image en tableau de pixel ;
- java.awt.Color : Contient les Codes de couleurs prédéfinis dans Java ;

- java.awt.Dimension; Permet d'avoir les dimensions d'une image ;
- java.awt.Image; permet de lire les images ;
- java.io.IOException : Classe géant les exceptions d'entrées –sorties ;
- java.awt.image.BufferedImage : Classe permettant de stocker une image dans un buffer;
- javax.imageio.ImageIO : Classe permettant l'enregistrement d'une image dans un format particulier ;

### 4.3.Structure du programme

L'implémentation de notre programme est structurée en trois classes :

- ❖ Main.java : exécute l'algorithme particulier ;
- ❖ RGB\_Decomposition.java: contient toutes les fonctions procédant à décomposition du RGB vers YCbCr et vice versa;
- ❖ ImagePanel.java : contient la fonction qui importe et lit une image ;
- ❖ code\_Decode : Contient notre algorithme principal avec l'utilisation des ondelettes.
- ❖ Listener.java : contient toutes les actions des boutons qui se trouvent dans la classe main.java

Le fonctionnement du programme est le suivant : on importe une image pour le traitement mathématique.

Nous allons exploiter les Cinq classes principales contenant les mécanismes de compression :

#### ➤ La classe RGB\_Decomposition.java

Cette classe contient un constructeur « **RGB\_Decomposition** » qui contient à son tour l'instanciation de la classe main et deux méthodes, la première méthode **convert\_to\_YCbCr** public void convert\_to\_YCbCr(int r[], int g[], int b[]) qui permet la conversion des couleurs du RGB vers YCbCr avec les formules déclaré auparavant

Par contre la deuxième méthode **Convert\_to\_RGB** public void convert\_to\_RGB(float Y[], float Cb[], float Cr[]) fait l'inverse de la dernière car elle fait la conversion des couleurs YCbCr vers RGB avec les formules déclaré auparavant.

#### ➤ La classe code\_Decode

Cette classe contient les fonctions mathématiques fondamentales dans la compression : la DWT et la DWT inverse. Mais leur implémentation n'est pas comme déclarée comme dans les formules vues auparavant. En effet, la DWT est une formule qui prend beaucoup de temps dans le calcul.

#### **Méthode "Prepare"**

```
public static void prepare(float[] pixels, int width, int height, int steps)
```

Cette méthode contient l'algorithme de la DWT " ondelettes" Comme expliqué auparavant

#### **Méthode Back**

```
public static void back(float[] pixels, int width, int height, int steps) {
```

A partir de la méthode Back on fait la reconstitution de l'image à partir des formules déclarée auparavant.

#### **Méthode Power**

```
Public static int power(int n, int m)
```

Cette fonction permet d'élever la puissance ex:  $2^3=8$

#### **méthode Quantification**

```
Public static float[] quantification(float[] pixels, int step, int width, int height)
```

Cette fonction permet de faire la quantification des résultats trouvés

#### ➤ **La classe ImagePanel.java**

Cette classe a un rôle intermédiaire car elle permet de chargé l'image pour l'utilisé après dans la compression.

#### ➤ **La classe Listener.java**

Le rôle de cette classe est aussi secondaire car on y trouve tous les boutons qui vont être utilisé pour l'interface et qui seront instancié dans la classe main.

#### ➤ **La classe Main.java**

La classe main à un rôle spécifique on trouve les instanciations de tous les autres classes

# Chapitre 4 : Tests et résultats

## 1. Introduction

Cette partie de ce rapport est consacrée aux validations des tests de chaque algorithme décrit auparavant.

La machine utilisée pour les tests est un PC portable de marque ASUS K53SD avec un processeur Intel(R) Core(TM) I5 2450m CPU @1.70GHz 1.70GHz, avec une mémoire vive de 8Go, sous le système d'exploitation Windows 7 Professionnel 64 bits.

## 2. Tests de l'algorithme de compression numérique

### 2.1. Tests des performances de l'algorithme avec un autre algorithme

Nous allons, en premier lieu, tester notre algorithme avec un autre algorithme pour comparer ses performances. Nous avons choisi l'algorithme de compression sécurisé développé l'an dernier au sein de notre département d'informatique à l'université de Guelma [27].

a) L'application Compression Sécurisé

Compression sécurisé est un logiciel de compression multimédia multifonctionnel. Il a la capacité de compresser plusieurs formats d'image avec l'utilisation des transformées en cosinus discrètes(DCT).

Les principaux formats supportés par Compression Sécurisé sont:

➤ L'image : JPG / BMP / PNG / GIF ;

L'utilisation de ce logiciel est de comparer les résultats de compression entre Compression sécurisé et notre algorithme.

b) Comparaison des résultats des deux algorithmes

Nous allons comparer les résultats de notre algorithme avec celui de Compression Sécurisé en se basant sur trois critères :

- ✓ La qualité de l'image ;
- ✓ le taux de compression ;
- ✓ Le temps de compression

Pour cela, Compression Sécurisé est très simple à utiliser. Nous sélectionnons une image quelconque et choisissons le format de sortie. La qualité de l'image n'est pas

une option dans ce logiciel. Il prend une image et la convertit automatiquement au format voulu.

### **3. Comparaison des résultats obtenus**

Pour la comparaison des résultats, nous allons utiliser des outils mathématiques tels que PSNR (Peak Signal to Noise Ratio) et le taux de compression :

#### **3.1. Les images utilisées**

Nous avons utilisé une base de données d'images, composé de plusieurs images de petites tailles (Lena, Barbara, Flower) qu'on peut trouver sur net. De grandes tailles (big\_tree, bridge, deer, fireworks, spider\_web,...) téléchargé depuis le site <http://www.imagecompression.info>.

Notre algorithme et celui de compression sécurisé donnent en sortie un seul format .jpeg. Donc, avec compression sécurisé, nous allons compresser l'image récupérée avec le même format et comparer les résultats des deux images obtenues. Pour la comparaison, nous allons nous baser sur la qualité des deux images obtenues, le temps d'exécutions ainsi que le taux de compression de chaque image.

Dans notre algorithme, nous avons fixé le nombre d'itérations à deux pour avoir de meilleurs résultats. Dans la "compression sécurisé" nous avons fixé la taille de bloc à 8x8 et la qualité 25 pour avoir des meilleurs résultats.

#### **Pour les images de petites tailles**

##### **Test 1**



Figure. 4.1 - Image Originale

Avec l'utilisation de compression sécurisé, nous avons eu l'image représenté dans la figure 4.2 avec l'utilisation de la transformée en cosinus discrète et l'image dans la figure 4.3 avec l'utilisation des ondelettes.

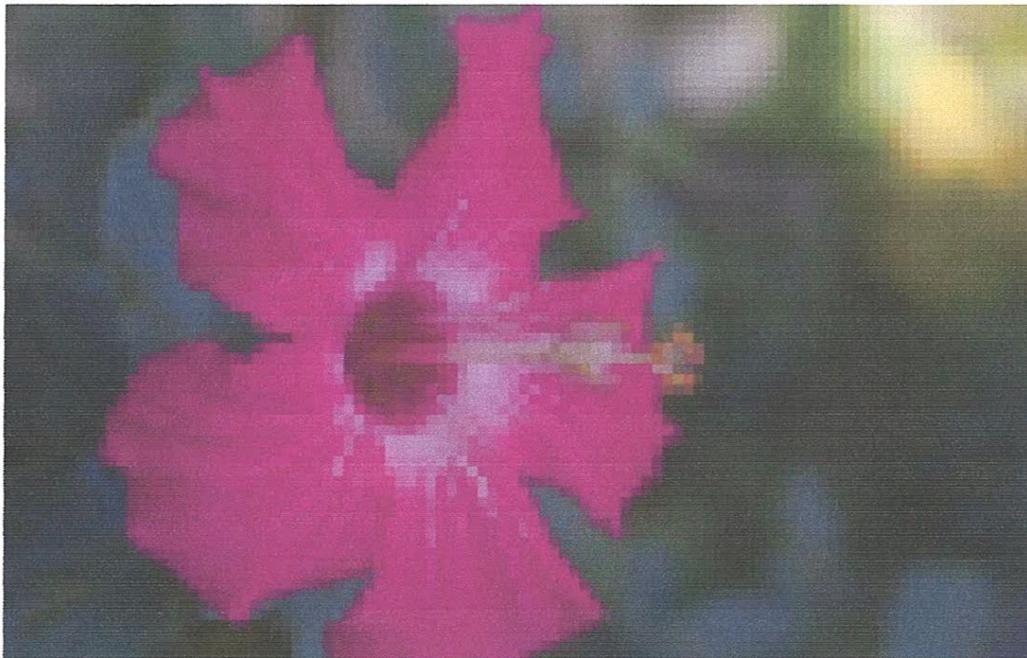


Figure. 4.2 - Image compressé avec DCT Taille = 10 ko



Figure. 4.3 - Image compressé avec DWT Taille = 29 ko

### **Discussion**

Le choix de l'image "flower" a été en raison de :

- les tests dans la compression sécurisée ont été faits avec cette image.
- on voit les résultats en termes des trois critères "taux, temps, qualité"

### **TEST 2**

Pour mieux éclaircir notre travail, nous allons faire un autre test, ce test sera sur une autre image "barbara "

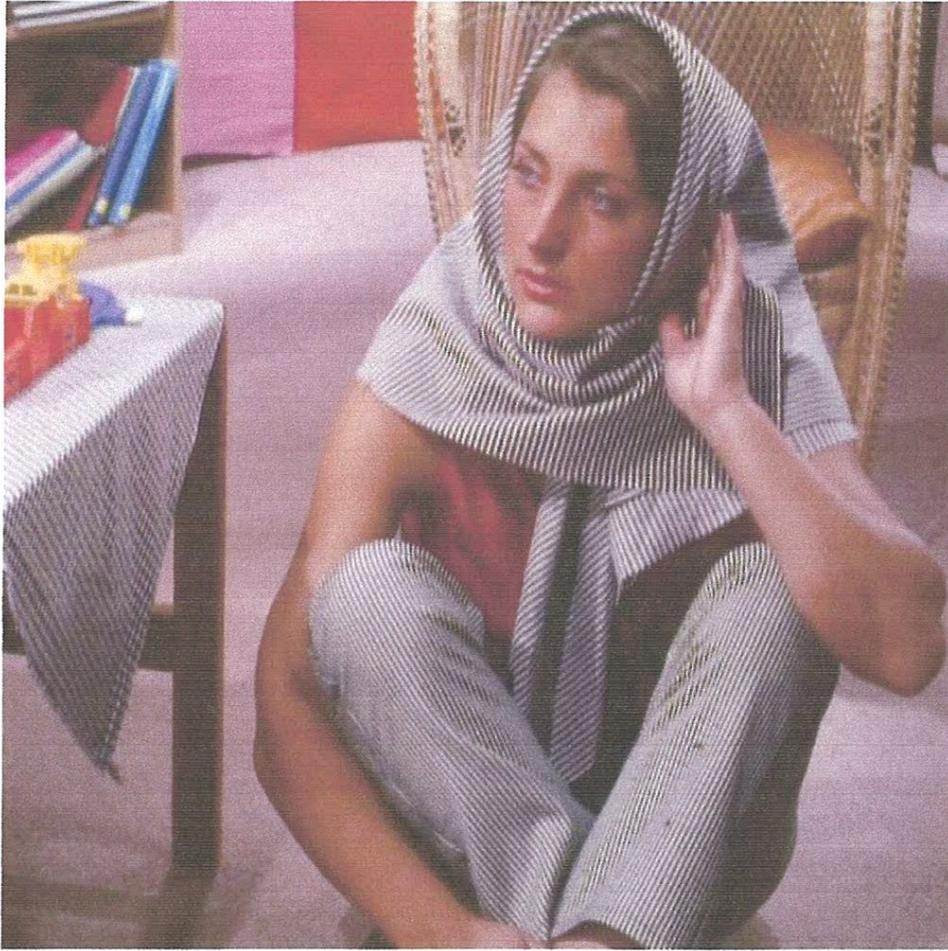


Figure 4.4 - Image Originale.

Avec l'utilisation de compression sécurisé, nous avons eu l'image représenté dans la figure 4.5 avec l'utilisation de la transformée en cosinus discrète et l'image dans la figure 4.6 avec l'utilisation des ondelettes.

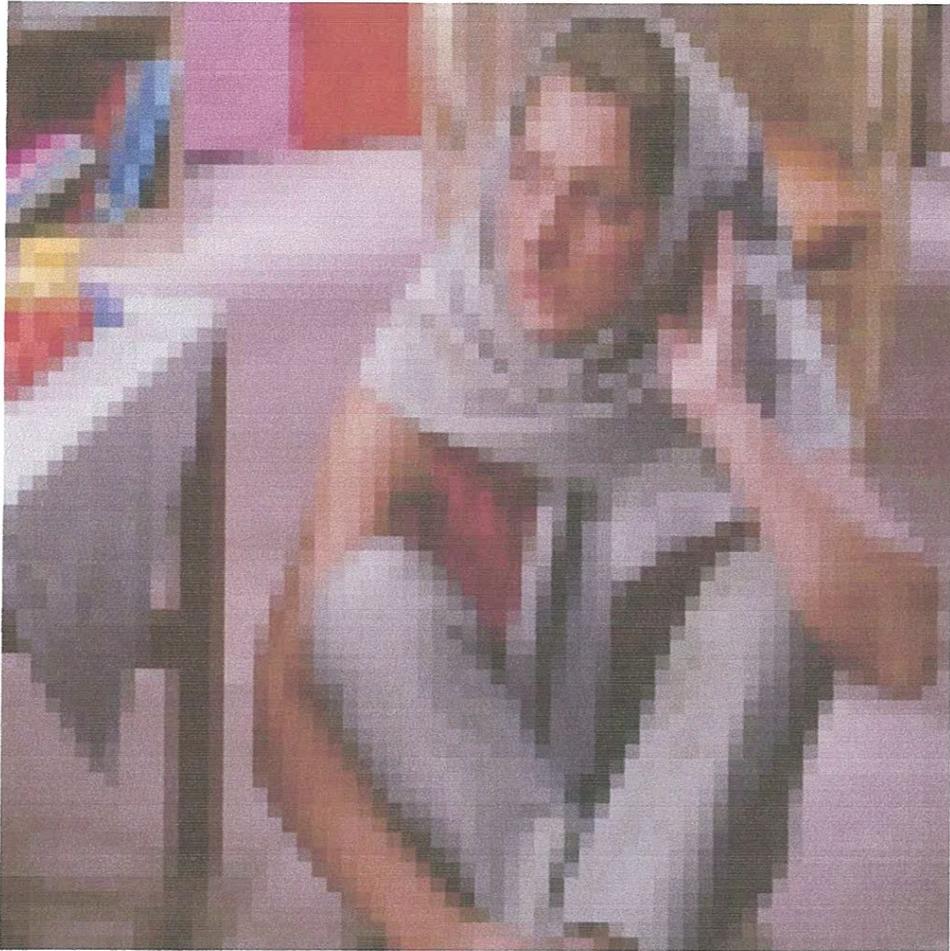


Figure 4.5 - Image compressé avec DCT

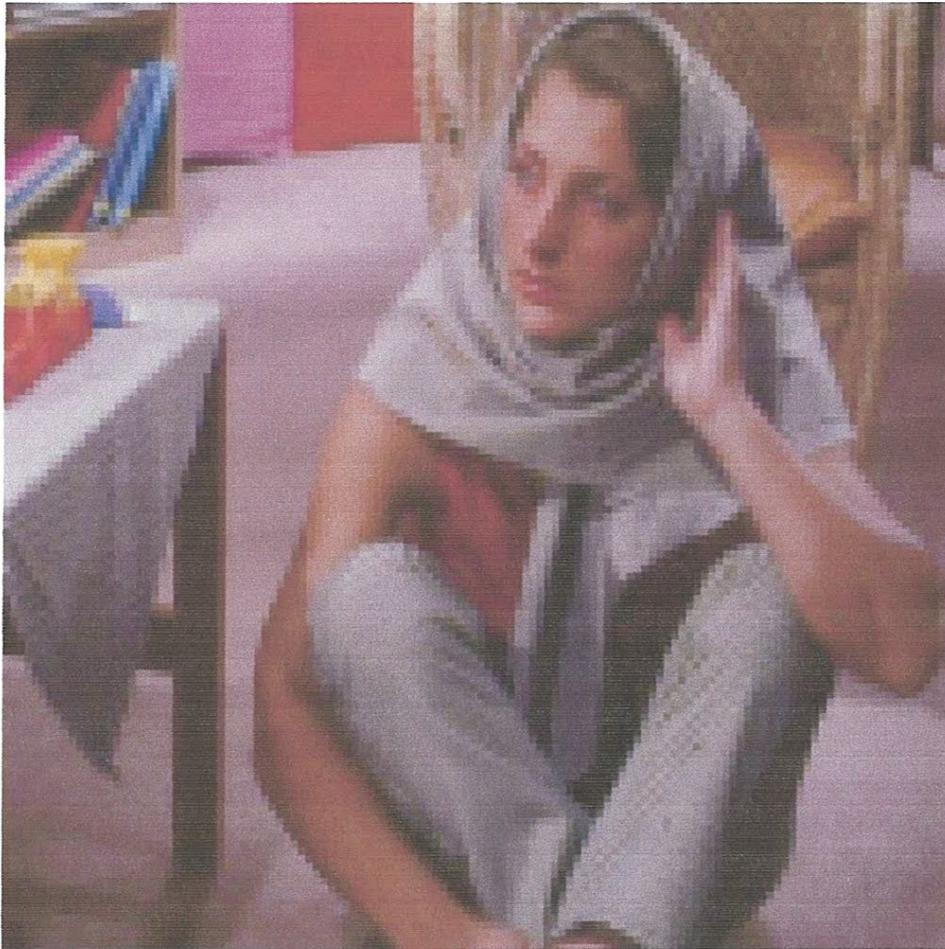


Figure. 4.6 - Image compressé avec DWT

### **Discussion**

Dans ce deuxième test, nous avons choisi L'image de test "barbara" Pour voir mieux la différence des résultats en termes de qualité.

### **Test 3**

#### **Lena**

Dans un autre test avec l'utilisation de compression sécurisé, nous avons eu l'image représenté dans la figure 4.8 avec l'utilisation de la transformée en cosinus discrète et l'image dans la figure 4.9 avec l'utilisation des ondelettes.



Figure 4.7 - Image Originale



Figure 4.8 - Image compressé avec DCT



Figure 4.9 - Image compressé avec DWT

#### Discussion Générale

Pour la compression sécurisée, nous avons fixé la taille de bloc 8x8 Pixels et la qualité = 25 pour avoir des meilleurs résultats.

D'après une première observation, nous pouvons dire que la différence est claire entre la qualité des images dans les trois tests, on voit que la DWT donne des meilleurs résultats par rapport à la DCT

Les tableaux 4.1 4.2 4.3 Nous montrent la variation de la taille de fichier en fonction du taux de compression, qualité d'image, vitesse de traitement.

- En termes de Taux de compression :

	Taille originale	DCT	DWT
Flower	1400 ko	13.9 ko	24.5 ko
Barbara	52.4 ko	10 ko	27.4 ko
Lena	48.3 ko	10 ko	24 ko

Tableau 4.1 Taille des fichiers après chaque transformée.

Dans ce tableau, nous observons que le taux de compression des deux algorithmes DCT et DWT ont dépassé les 98%, et d'après les premières observations faites, on voit qu'il y a une différence entre les résultats mais vu qu'ils ont dépassé les 98 % alors on peut négliger cette différence.

- En termes de Vitesse de traitement :

Nous allons maintenant comparer les résultats en termes de vitesse de traitement, on voit clairement que la vitesse d'exécution de la DWT est trois fois plus rapide que celles de la DCT, cela est dû à plusieurs critères:

- ✓ Optimisation du code source : on a utilisé des instructions simples.
- ✓ Les ondelettes sont connues par leurs rapidités.

	DCT	DWT
Flower	413.0 ms	344.0 ms
Barbara	469.0 ms	874.0 ms
Lena	766.0 ms	203.0 ms

Tableau 4.2 Les résultats des images testés en termes de vitesse de traitement.

- En termes de Qualité d'image:

Pour une bonne qualité d'une image, la valeur du PSNR doit être entre 30dB et 40dB. Dans le tableau tab 4.3, nous observons que les images obtenues avec DWT sont impressionnante. Par contre, les images obtenues avec compression sécurisé sont loin derrière, qui veut dire la valeur du PSNR de la DCT est inférieure par rapport aux tailles précédentes.

	DCT	DWT
Flower	21.6 dB	34.2 dB
Barbara	18.12 dB	31.13 dB
Lena	23.95 dB	32.26 dB

Tableau 4.3 Les résultats des images testés en termes de qualité d'image

D'après cette première observation, nous pouvons dire que les ondelettes nous donnent une bonne qualité d'image. Cela donne un avantage pour la DWT malgré qu'elle gagne pas beaucoup en taux de compression par rapport a la DCT mais en termes de temps et de qualité elle en gagne.

### **Pour les images de grandes tailles :**

#### **Test 1:**

#### **BIG\_TREE**



Figure 4.10 - Image originale



Figure 4.11- Image compressé avec DCT



Figure 4.12- Image compressé avec DWT

#### Discussion Générale

Nous avons fait le test pour une image de taille grande pour voir clairement la différence entre la DCT et les ondelettes. D'après une première vue sur le test sur Big\_tree, on peut voir que la différence se fixe dans la luminosité en termes de qualité

Les tableaux 4.4 4.5 4.6 Nous montrent la variation de la taille de fichier en fonction du taux de compression, qualité d'image, vitesse de traitement.

- En termes de Taux de compression :

	Taille originale	DCT	DWT
Big_tree	8700 ko	542 ko	1400 ko

Tableau 4.4 Taille des fichiers après chaque transformée

Comme dans les images de petites tailles, le taux de compression est très grand pour une image de grande taille. Dans le tableau 4.4, nous observons que le taux de compression des deux algorithmes DCT et DWT ont dépassé les 98%.

- En termes de Vitesse de traitement :

	DCT	DWT
Big_tree	13437.0 ms	344.0 ms

Tableau 4.5 Temps de traitement pour chaque transformée

Pour les images de grandes tailles, la vitesse de traitement est impressionnante pour les ondelettes, elle prend un temps incroyable pour la DCT.

- En termes de Qualité d'image:

	DCT	DWT
Big_tree	35.6 dB	29.2dB

Tableau 4.6 Les résultats des images testés en termes de qualité d'image

Si les ondelettes ne peuvent pas gagner plus que la DCT en termes de taux de compression c'est en termes de qualité qu'on voit la différence car elle on gagne beaucoup.

D'après cette première observation, nous pouvons dire que les ondelettes nous donnent une bonne qualité d'image. Cela donne un avantage pour la DWT.

## 4. Conclusion

D'après les tests effectués, nous pouvons conclure les points suivants:

- ✓ Notre algorithme est très rapide et nous donne des meilleurs résultats en termes de temps et qualité.
- ✓ Pour des résultats performantes, les ondelettes de Haar est le meilleur choix.

## Conclusion Générale

Dans notre rapport, nous avons exploité le principe de la compression dans le domaine multimédia, qui est une solution exemplaire pour différents problèmes liés à la transmission et stockage des données.

Notre algorithme est basé sur la compression standard des images en conservant la qualité visuelle de l'image avec un taux élevé de compression, une bonne qualité et une grande vitesse de traitement. Nous avons effectué des tests réels et une comparaison importante avec une application « compression sécurisé »

En général, on peut dire que l'approche proposée permet d'atteindre de très bons résultats du point de vue qualité des images reconstruites d'une part et le taux de compression d'autre part. En effet, elle permet de réduire énormément la quantité de données contenues dans les images tout en conservant leurs propriétés originales.

En perspective de notre travail, l'application du principe des ondelettes pour les vidéos serait souhaitable, étant donné qu'une vidéo est une séquence d'image.

# Bibliographie :

- [1] S. mourad. Compression des images numériques par la technique des ondelettes. Université Ferhat Abbas Sétif, Algérie, 2011.
- [2] A. Chouha. Traitement et Transfert d'images Par Réseau de Capteurs sans Fil. Thèse de magister, Université de Hadj Lakhdar Batna, 2011.
- [3] A. Zitouni. Ondelettes et techniques de compression d'image numérique. Thèse de Doctorat, Université de Mohamed Khider Biskra, 2013.
- [4] S. Piegion. Contributions à la compression de données. Thèse de Doctorat, Université de Montréal, 2001.
- [5] La compression de données conservatrice : <http://d.nouchi.free.fr/TER/c1.html>, consulté le : 20-03-2014.
- [6] Analysis Of Image Compression Methods Based On Transform and Fractal Coding. Mémoire de master, National Institute of Technology, Rourkela, 2013.
- [7] Analysis Of Image Compression Methods Based On Transform and Fractal Coding. Mémoire de master, National Institute of Technology, Rourkela, 2013.
- [8] C.Valade. Compression d'image complexe avec pertes : Application à l'imagerie Radar. Thèse de Doctorat, Ecole Nationale Supérieure Des Télécommunications Paris, 2006.
- [9] A. Thil. Initiation à la Recherche Algorithmes de compression de données et de traitement d'images, université de Metz, 2007.
- [10] A. Francklin, S. Tegueu. Compression d'images fixes: comparaison des méthodes par transformations en ondelettes et celle par curvelets, rapport de licence, Institut universitaire de technologie Fotso Victor de Bandjoun, 2009.
- [11] S. mourad. Compression des images numériques par la technique des ondelettes. Université Ferhat Abbas Sétif, Algérie, 2011.
- [12] E. Incerti. Compression d'image, algorithmes et standards, Edition Vuibert, Paris, 2003.
- [13] F. Gargne, C. KNOFF Gaëtan. Codage, Compression et Cryptologie, LECOURTOIS, Université de Nice-Sophia Antipolis.
- [14] E. Christophe, Cours : Imagerie Satellite et Compression. 2005.
- [15] K. Barhoumi. Approche par marquage pour l'évaluation de la qualité d'image dans les applications multimédias. Thèse de master, Université de Québec, 2012.

- [16] Yun Q. Shi, Huifang Sun. Images and video compression for multimedia Engineering, Fundamentals, Algorithms and Standards, Second Edition, CRC Press 1999.
- [17] G. Obozinski, Cours sur le Traitement de l'information et Vision Artificielle, Eléments de théorie des ondelettes, paris Tech, 2014.
- [18] La compression d'images par ondelettes :  
<https://www.ceremade.dauphine.fr/~peyre/java/image/> consulté le : 21-03-2014
- [19] Construction d'une analyse multi résolution :  
<http://www.tsi.telecomparistech.fr/pages/enseignement/ressources/beti/ondelettes/AMR.htm> consulté le: 22-03-2014
- [20] <http://www-ljk.imag.fr/membres/Valerie.Perrier/SiteWeb/node9.html> consulté le: 01-04-2014
- [21] La compression d'images par ondelettes :  
<https://www.ceremade.dauphine.fr/~peyre/java/image/> 02-04-2014
- [22] Des ondelettes pour compresser les images  
<http://www.larecherche.fr/savoirs/b-a-ba/ondelettes-compresser-images-01-01-2002-71149> consulté le: 08-04-2014
- [23] Compresser les images  
<http://www.larecherche.fr/savoirs/b-a-ba/ondelettes-compresser-images-01-01-2002-71149> consulté le: 15-04-2014
- [24] Z-E.BAARIR, A.ouafi. Etude de la transformée en ondelettes dans la compression d'image fixe. Courier de savoir, N°05.pp.69-74, 2004
- [25] Yun Q.Shi, Huifan Sun. Image and Video Compression for Multimedia Engineering, Second Edition, 2008
- [26] [http://www.irisa.fr/temics/Accord/CDROM/i3s/decomp\\_dyad/intro.html](http://www.irisa.fr/temics/Accord/CDROM/i3s/decomp_dyad/intro.html) consulté le: 16-04-2014.
- [27] K. Maadoul et A. Harroud. Compression multimédia et Cryptologie. Mémoire de master. Université 08 mai 1945, Guelma, 2013.