

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
UNIVERSITE MOULOU D MAMMERI, TIZI OUZOU



Faculté de Génie Electrique et d'informatique
Département d'Electronique

MEMOIRE

De fin d'étude

En vue de l'obtention du Diplôme d'Ingénieur d'Etat

En Electronique

Option : contrôle

Thème

**Compression d'images Météosat seconde
génération par le standard JPEG 2000**

Proposé et dirigé par :
Mr. LAHDIR Mourad

Réalisé par :
Mr. ARAB Hakim

Promotion: 2009/ 2010

Sommaire

INTRODUCTION GENERALE.....	1
-----------------------------------	----------

Chapitre I : Méthodes de compression d'images

I.1 Introduction.....	3
I.2 Définition et caractéristiques des images.....	3
I.2.1 Définition de l'image	3
I.2.2 L'image numérique.....	3
I.2.3 Les caractéristiques des images numériques.....	3
I.2.3.1 Pixel.....	3
I.2.3.2 Dimension ou définition de l'image.....	3
I.2.3.3 Résolution.....	3
I.2.3.4 Luminance.....	4
I.2.3.5 Contraste.....	4
I.2.3.6 Images a niveaux de gris.....	4
I.2.3.7 Images en couleurs.....	4
I.2.3.8 Contours et textures.....	5
I.2.3.9 Bruit.....	5
I.2.3.10 Redondance.....	5
I.3 La compression d'images.....	6
I.3.1 Définition de la compression d'images.....	6
I.3.2 Critères de performance de la compression d'images.....	6
I.3.2.1 Quotient de compression et taux de compression.....	6
I.3.2.2 Débits binaires.....	7
I.3.2.3 Entropie.....	7

I.3.2.4	Mesure de distorsion.....	7
I.3.2.5	Complexité.....	8
I.3.2.6	La courbe débit-distorsion.....	8
I.3.2.7	Critère psycho visuel en compression d'image.....	9
I.3.3	Les étapes de la compression d'images.....	9
I.3.3.1	Décorrélacion (transformation).....	9
I.3.3.2	Quantification.....	10
I.3.3.3	Codage entropique.....	10
I.4	Méthode de compression d'images.....	10
I.4.1	Méthode de compression réversible.....	10
I.4.1.1	Codage de Shannon-Fano.....	10
I.4.1.2	Codage de Huffman.....	11
I.4.1.3	Codage arithmétique.....	12
I.4.1.4	Codage par plan de bits.....	15
I.4.1.5	Codage prédictif.....	15
I.4.1.6	Codage Run Length Encoding (RLE).....	16
I.4.1.7	Codage Lempel-Ziv-Welch (LZW).....	17
I.4.2	Méthode de compression irréversible.....	18
I.4.2.1	Méthodes par transformée.....	18
I.4.2.1.1	La transformée de Fourier discrète (DFT).....	19
I.4.2.1.2	La transformée de Karhunen Loeve.....	20
I.4.2.1.3	La transformation en cosinus discrète (DCT).....	20
I.4.2.1.4	La transformation en ondelettes.....	21
I.4.2.1.5	Fractales.....	23
I.4.2.2	Quantification.....	24
I.4.2.2.1	Quantification scalaire (QS).....	24
I.4.2.2.2	Quantification vectorielle (QV).....	25

I.4.2.3 Codage par DPC (Différentiel Prédictive Coding).....	27
I.4.2.4 Méthodes hybrides	27
I.5 Les normes de compression.....	28
I.5.1 La norme JPEG.....	28
I.5.2 La norme JPEG2000.....	29
I.5.3 La norme MPEG.....	30

Chapitre II: Les images Météosat seconde génération (MSG)

II.1 Introduction.....	34
II.2 Les satellites météorologiques Météosat.....	34
II.2.1 Les satellites météorologiques.....	34
II.2.2 Les satellites Météosat.....	34
II.2.2.1 La première génération (Météosat 1 a 7).....	34
II.2.2.2 La deuxième génération (Météosat seconde génération : MSG).....	34
II.3 Paramètre influençons sur l'acquisition des images.....	35
II.3.1 Géométrie soleil satellite objet.....	35
II.3.2 Réflectance du corps visé.....	35
II.3.3 Éclairement du soleil.....	35
II.4 Caractéristique optique des objets au sol.....	35
II.4.1 Propriétés optiques des surfaces d'eaux.....	36
II.4.2 Propriétés optiques de la végétation.....	36
II.4.3 Propriétés optiques du sol.....	36
II.4.4 Propriétés optiques des nuages.....	36
II.5 Fonctionnement des satellites Météosat seconde génération (MSG).....	36
II.5.1 Canaux visible.....	37
II.5.2 Canaux infrarouge.....	38

II.5.3 Canaux vapeur d’eaux.....	39
II.5.4 Canal visible à haute résolution (HRV).....	39
II.6 Application des images MSG.....	40

Chapitre III : Application du standard JPEG 2000 pour la compression des images Météosat second génération

III.1 Introduction.....	42
III.2 Les étapes de l’algorithme de compression.....	43
III.2.1 Découpage en tuiles.....	43
III.2.2 La transformée en ondelettes discrète (TOD).....	43
III.2.3 Quantification scalaire.....	48
III.2.4 Codage entropique.....	49
III.2.5 Allocation de débit.....	51
III.3 Les étapes de l’algorithme de décompression.....	52
III.3.1 Le décodage entropique.....	52
III.3.2 La quantification inverse.....	53
III.3.3 Transformée en ondelette discrète inverse (TODI).....	53
III.4 Conclusion.....	54

Chapitre IV : Tests et résultats

IV.1 Introduction.....	55
IV.2 Présentation des données.....	56
IV.2.1 Image Lena.....	56
IV.2.2 Les image Météosat seconde génération (MSG).....	56
IV.3 Testes et résultats.....	58

IV.4 Interprétation des résultats.....	70
IV.5 Aspect visuel des images restitué.....	71
IV.6 Conclusion.....	76
CONCLUSION GENERALE.....	77
BIBLIOGRAPHIE.....	78

INTRODUCTION GENERALE

Introduction générale

Les images que nous manipulons proviennent de diverse source comme les satellites, la photographie, la visioconférence, la télévision et les applications médicales, ce qui représente un volume de données numériques très important. Malgré l'augmentation rapide de la puissance des processeurs, des capacités mémoires et la rapidité des supports de transmission, ils restent impuissants devant la quantité énorme de ces données. Pour archiver et analyser ces dernières, nous avons recourue aux techniques de compression d'images, qui permettent de réduire le nombre de bits représentant un pixel, tout en sauvegardant une bonne qualité visuelle de l'image. On distingue deux sortes de méthodes de compression d'images, la compression son pertes (réversible) et la compression avec pertes d'information (irréversible).

Les méthodes de compression réversibles présentent une reconstitution exacte des pixels de l'image originale et l'inconvénient est le taux de compression relativement faible. Les algorithmes utilisés à cet effet, se basent sur la redondance statistique de l'information contenue dans l'image. Parmi les méthodes les plus utilise nous citons : le codage de Huffman, le codage arithmétique, le codage RLE (Run Lenght Encodimg) et le codage LZW (Lempel Ziv Welch).

Les méthodes de compression irréversible autorisent une certaine perte d'information, dans le but d'amélioré le taux de compression. Les algorithmes développés dans ce cas sont basées sur la redondance psycho-visuelle inhérente à l'image. Ils permettent d'avoir un fort taux de compression toute en sauvegardent une bonne qualité visuelle des images restituées. Ces méthodes opèrent en quatre étapes : le découpage de l'image en blocs pour mieux exploiter le contenu local, le changement de représentation pour éliminer la redondance spatiale entre les pixels voisins et concentrée le maximum de l'énergie dans un nombre restreint de coefficients en utilisant des transformées tel que la DCT (Discret Cosine Transform), la KLT (Karhunan Loeve Transform) et la TOD (Transformée en ondelette discret). La quantification scalaire ou vectorielle de ces dernières pour régler la précision celant la perception du système visuelle humain et le codage entropique qui permet de minimiser la redondance statistique et de générer la séquence binaire à transmettre ou a sauvegardé.

Afin d'archiver les images Météosat seconde génération (MSG), pour des études climatique ou environnemental à long terme, les méthodes de compression d'images irréversible sont les seuls qui peuvent répondre à cette exigence. En effet les satellites Météosat seconde génération produisent douze images via ces douze canaux spectraux toute les quart d'heur, conduisant ainsi à un nombre important d'images.

Afin de comprimer les images MSG, nous avant opté pour une méthode de compression basée sur l'algorithme de JPEG 2000 qui utilise la transformation en ondelette. Dans notre étude, nous avons partagé notre travail en quatre chapitres organise dans l'ordre suivant :

Dans le premier chapitre, nous donnerons les notions générales nécessaires pour la compression d'images et les différentes méthodes utilisées dans ce domaine.

Dans le deuxième chapitre, nous présenterons les images MSG et nous donnerons quelques définitions et particularités de ces dernières.

Dans le troisième chapitre, nous présenterons les étapes de l'algorithme de la norme JPEG 2000 et nous donnerons une description de la théorie des ondelettes et du codeur entropique utilisé.

Dans le quatrième chapitre, nous présenterons les résultats des tests de l'algorithme développé dans le chapitre précédant ainsi que leurs interprétations.

Enfin, nous terminerons ce mémoire par une conclusion général et des perspectives.

CHAPITRE I

Méthodes de compression d'images

I.1 Introduction :

Avec la parole, l'image constitue l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec autrui. C'est un outil de communication universel, dont la richesse du contenu permet aux êtres humains de tous âges et de toutes cultures de se comprendre. Chacun, peut analyser l'image à sa manière, pour en dégager une impression ou extraire des informations.

I.2 Définition et Caractéristiques des images [1] [17] :

I.2.1 Définition de l'image :

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film, etc. C'est aussi un ensemble structuré d'informations qui, après affichage sur écran, ont une signification pour l'œil humain.

Elle peut être décrite sous la forme d'une fonction $I(x, y)$ de brillance analogique continue, définie dans un domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur. Sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation.

I.2.2 L'image numérique :

Une image numérique, telle qu'on peut la voir sur un écran d'ordinateur est une matrice de m lignes et n colonnes, soit $n \times m$ points appelés pixels et à chaque pixel est associé une ou plusieurs valeurs d'intensité qui se combinent pour déterminer la couleur.

I.2.3 Les caractéristiques des images numériques :

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants :

I.2.3.1 Pixel :

Contraction de l'expression anglaise " Picture éléments ": éléments d'image, le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression.

I.2.3.2 Dimension ou définition de l'image :

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes, nous donne le nombre total de pixels dans une image.

I.2.3.3 Résolution :

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre).

I.2.3.4 Luminance :

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface.

I.2.3.5 Contraste :

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

Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C , est défini par le rapport :

$$C = \frac{L_1 - L_2}{L_1 + L_2} \quad (\text{I-1})$$

I.2.3.6 Images à niveaux de gris :

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondante à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel est donc représenté par un octet (1 octet=8 bits).

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la couleur de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux.

Une image comportant deux couleurs (noir et blanc, image dite bitmap), sera codée sur 1 bit par pixel car 1 bit peut prendre 2 valeurs, notées symboliquement 0 et 1.

I.2.3.7 Images en couleurs [15] :

Même, s'il est parfois utile de pouvoir représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation de ces dernières s'effectue de la même manière que les images monochromes, avec cependant, quelques particularités.

a) La représentation en couleurs réelles :

Il existe plusieurs modes de codage informatique des couleurs, les plus utilisés, sont le mode RVB (Rouge, Vert, Bleu) et le mode CMJN (Cyan, Magenta, Jaune, Noir) utilisé principalement pour l'impression,

L'image RVB utilise 24 bits pour chaque point de l'image. Huit bits sont employés pour décrire la composante rouge (R), huit pour le vert (V) et huit pour le bleu (B). Il est ainsi, possible de représenté environ 16,7 millions de couleurs différentes. Cet espace est basé sur une synthèse additive des couleurs, c'est-à-dire que le mélange des trois composantes R, V, et B à leur valeur maximale donne du blanc, à l'instar de la lumière. Le mélange de ces trois couleurs à des proportions diverses permet de reproduire à l'écran une part importante du spectre visible.

L'image CMJN est la synthèse des quatre images cyan (C), magenta (M), jaune (J) et noire (N). On parle dans ce cas, de synthèse soustractive car les encres soustraient absorbent certaines franges de la lumière blanche et réfléchissent les autres. Dans ce type d'images, chaque pixel est représenté sur quatre octets : un octet pour chacune des couleurs : cyan (C), magenta (M), jaune (J) et noire (N), il peut donc comporter plus de 4 milliards de couleurs.

b) La représentation en couleurs indexées :

Afin de diminuer la charge de travail nécessaire pour manipuler des images en 24 bits, on peut utiliser le mode de représentation en couleurs indexée. Le principe consiste à déterminer le nombre de couleurs différentes utilisées dans l'image, puis à créer une table de ces couleurs en attribuant à chacune une valeur numérique correspondant à sa position dans la table. La table, appelée palette, comporte également la description de chacune des couleurs, sur 24 bits.

c) Autres modèles de représentation :

Le modèle RVB, représentant toutes les couleurs par l'addition de trois composantes fondamentales, n'est pas le seul possible. Il en existe de nombreux autres, l'un d'eux est particulièrement important, il consiste à séparer les informations de couleurs (chrominance) et les informations d'intensité lumineuse (luminance). Il s'agit du principe employé pour les enregistrements vidéo. La chrominance est représentée par deux valeurs (selon des modèles divers) et la luminance par une valeur.

I.2.3.8 Contours et textures :

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative, les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image, les points qui séparent deux textures différentes.

I.2.3.9 Bruit :

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur.

I.2.3.10 Redondance :

Pour comprimer des images, il faut éliminer les redondances présentes à l'intérieures, celles-ci sont de trois types :

1) Redondance spatiale :

La redondance spatiale ou corrélation entre pixels voisins, exprime le fait qu'un point ou une zone de l'image n'est généralement pas indépendante de son environnement ; il est ainsi fréquent de trouver des plages uniformes dans les images, dans ce cas là, la corrélation est totale. La redondance dans le domaine spatial n'est pas facilement identifiable et généralement ne fournit pas toujours un bon taux de compression. Il est donc nécessaire de faire une transformation pour obtenir une décorrélation de l'information spatiale et un groupement d'énergie fréquentielle.

2) Redondance spectrale :

La redondance spectrale est liée aux fréquences, elle est acquise avec les transformations de domaines. Le but de ses transformées est de décorréler l'information présente dans l'image.

3) Redondance temporelle :

La redondance temporelle s'explique par le fait que, dans une séquence d'images, il y a statistiquement peu de changements en passant d'une image à une autre.

I.3 La compression d'images :

Entre la photo, la vidéo et la télévision, nous consommons toujours plus d'images numériques. Mais, les tuyaux qui servent à les diffuser (liaisons ADSL et câble, réseaux de téléphonie mobile, etc.) comme les supports qui permettent de les stocker (cartes mémoire, disques durs, DVD, etc.) n'ont pas des capacités infinies, même, si elles augmentent continuellement. D'autant que cette boulimie de pixels va encore s'accroître avec le développement de nouvelles technologies, telles que ; la haute définition, la télévision sur téléphone mobile, etc. Pour satisfaire ces nouveaux besoins, les spécialistes cherchent à mettre au point des méthodes de compression d'image, afin de réduire le poids des données, avec une contrainte incontournable : ne pas trop dégrader la qualité.

I.3.1 Définition de La compression d'images :

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. Cette compression peut; soit conserver l'image intacte, on parle alors de compression sans perte. Soit autoriser une dégradation de l'image pour diminuer encore l'empreinte mémoire, on parle ici de compression avec perte.

I.3.2 Critères de performance de la compression d'images :

Les critères de performance évaluent la qualité d'une compression en mesurons les différences qui peuvent exister entre l'image originale et l'image compressée. Les principaux critères les plus utilise sont :

I.3.2.1 Quotient de compression et taux de compression :

Le quotient de compression, aussi parfois appelé rapport de compression, est une mesure de la performance d'un algorithme de compression de données, il est définie comme étant le rapport entre la taille de l'image originale et la taille de l'image compressée :

$$Rc = \frac{\text{nombre de bits de l'image originale}}{\text{nombre de bits de l'image compressée}} \quad (I-2)$$

Le taux de compression est défini à partir du quotient de compression par la formule suivante :

$$\begin{aligned} Tc &= \left(1 - \frac{1}{Rc}\right) \times 100 \\ &= \left(\frac{\text{nombre de bits de l'image originale} - \text{nombre de bits de l'image compressée}}{\text{nombre de bits de l'image originale}}\right) \times 100 \end{aligned} \quad (I-3)$$

Donc plus le taux de compression est élevé, plus la taille du fichier compressé résultant est faible.

I.3.2.2 Débits binaires :

Le débit binaire est défini comme le nombre de bits par pixel de l'image compressée. On le calcule par la formule suivante :

$$debit = \frac{\text{nombre de bits de l'image compressée}}{\text{nombre de pixels de l'image}} = \frac{\text{nombre de bits par pixels de l'image originale}}{Rc} \quad (I-4)$$

Rc : Le rapport de compression

Le débit binaire est exprimé en bits par pixel (bpp).

I.3.2.3 Entropie :

L'entropie de Shannon, est une fonction mathématique qui, intuitivement, correspond à la quantité d'information contenue dans une image. Plus l'image contient beaucoup d'informations, plus l'entropie est grande, et vice versa.

L'entropie d'un point P de l'image est définie par :

$$H(p) = -\sum_{i=0}^{N-1} p(i) \log_2(p(i)) \quad (I-5)$$

N : est le niveau de quantification de l'image.

i : Niveau de gris quelconque.

$p(i)$: Probabilité de présence du niveau i ,

$$\text{On a : } p(i) = \frac{n_i}{n} \quad (I-6)$$

n_i : Est le nombre de pixels présentant la valeur i et n : le nombre total de pixels de l'image

L'entropie définit une borne inférieure sur le nombre moyen de bits nécessaires pour coder toutes les données contenues dans l'image.

I.3.2.4 Mesure de distorsion :

La distorsion est une mesure de l'erreur commise entre l'image originale et l'image reconstruite. Pour les mesures de distorsion, on utilisera l'erreur quadratique moyenne MSE (Mean Square Error) entre l'image originale et l'image compressée de taille $M \times N$:

$$MSE = \frac{1}{MN} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (I(i, j) - \tilde{I}(i, j))^2 \quad (I-7)$$

$I(i, j)$: Position du pixel de l'image originale.

$\tilde{I}(i, j)$: Position du pixel de l'image compressée.

Pour les mesures de qualité, on utilisera le rapport signal sur bruit $PSNR$ (Peak Signal to Noise Ratio) en décibels défini à partir de l' MSE par :

$$PSNR = 10 \log \left(\frac{N_{G_{\max}}^2}{MSE} \right) = 10 \log \left(\frac{2^n - 1}{MSE} \right) \quad (I-8)$$

Ou :

$N_{G_{\max}}$: Niveau maximal du codage du gris de l'image.

n : Nombre de bits par pixel.

Si l'image est parfaitement reconstruite par le processus de compression, l'erreur quadratique moyenne MSE tend vers zéro ($MSE \rightarrow 0$) et le $PSNR$ tend vers l'infinie ($PSNR \rightarrow \infty$).

I.3.2.5 Complexité :

Le troisième critère de performance d'un système de compression est la complexité. La complexité calculatoire peut être mesurée par le temps d'exécution du processus de compression ou en nombre d'opérations par pixel : C'est le nombre moyen d'opérations qui sont nécessaires à la compression de l'image.

I.3.2.6 La courbe débit-distorsion :

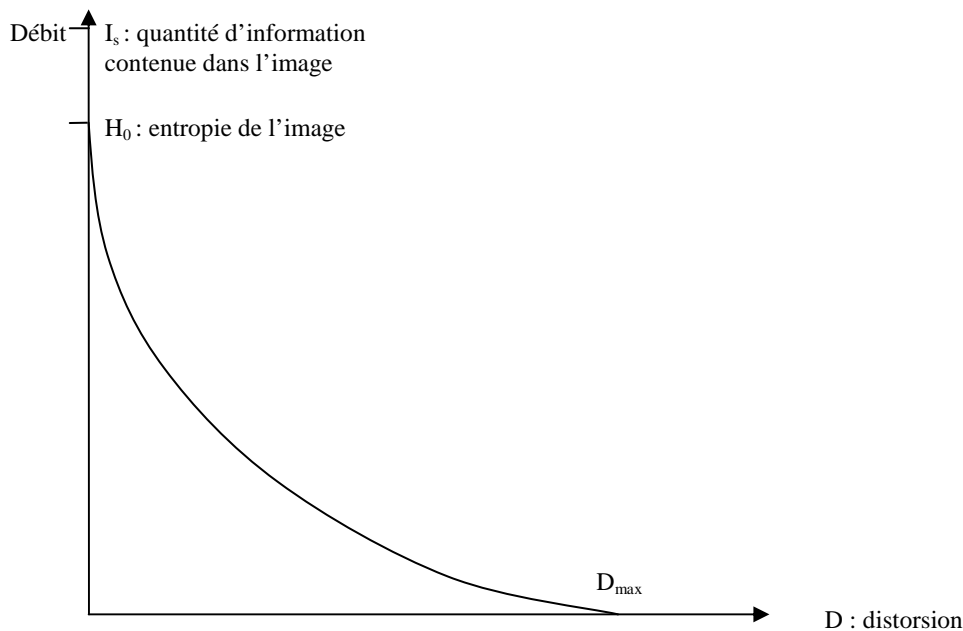


Figure I-1 : Courbe : débit - distorsion

La courbe est divisée en deux parties distinctes :

1) La première est située entre I_s et H_0 , cette dernière est confondue avec l'axe des ordonnées, ce qui veut dire que la distorsion est nulle, ceci correspond à une compression réversible.

2) La seconde partie est comprise entre H_0 et zéro (0), dans cette partie, l'image est codée avec un nombre de bits inférieur à l'entropie, indiquant, que cette dernière a subi des distorsions, ce qui correspond à une perte d'information invisible à l'œil nu, donc, l'image restituée est une approximation de l'image originale. Dans ce cas, la compression est irréversible.

I.3.2.7 Critère psycho-visuel en compression d'image :

Les méthodes de compression sans perte, ne causent aucun problème visuel, car, ils sont totalement réversibles. Par contre, les procédures de compression avec pertes diminuent la qualité de l'image. Plus le taux de compression est important, plus les distorsions apparaissent sur l'image.

Le point critique est la définition de la quantité de distorsions par rapport à la qualité de l'image. Le système visuel humain possède des caractéristiques particulières qui doivent être prises en compte. Notre œil est capable de distinguer environ sept millions de couleurs. Quand nous regardons une image notre système visuel doit résoudre beaucoup de contraintes : Perception 3D, ombres, objets cachés, etc. En effet, ce dernier tente de donner un sens visuel à chaque objet, car, notre perception est influencée par ce que nous nous attendons à voir, c'est le cas des illusions optiques par exemple.

I.3.3 Les étapes de la compression d'images :

Le schéma d'un système de compression classique est représenté sur la figure I-2. La première étape est la décorrélation, elle permet d'exploiter les redondances statistiques de l'image. Ensuite, vient l'étape de quantification, c'est la où la perte d'information survient. Enfin, le codage entropique des données quantifiées constitue la dernière étape, il permet d'organiser le train binaire en compressant les données sans altérer l'information.

L'étape qui distingue les deux méthodes de compression « réversible et irréversible », est la quantification, qui cause des pertes d'informations.

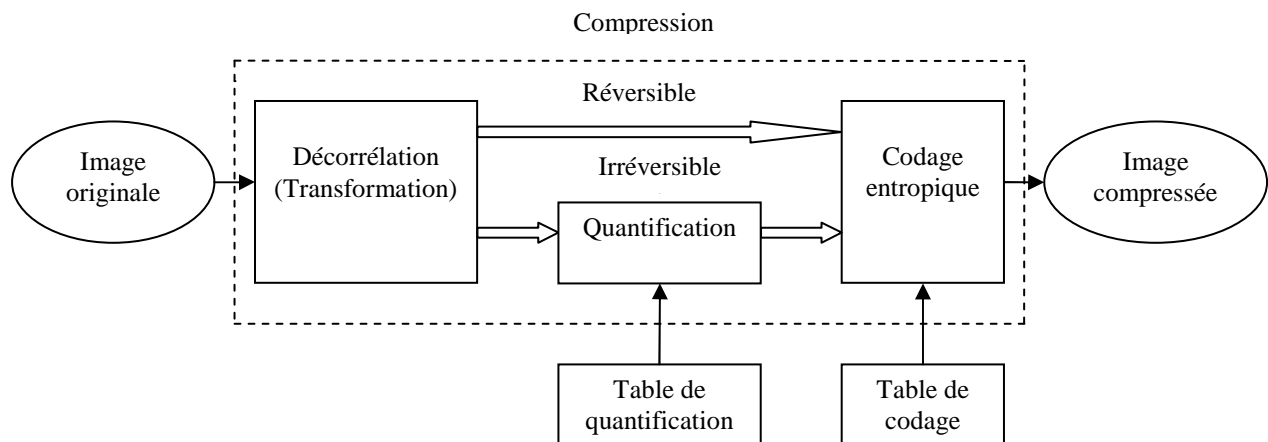


Figure I-2 : Schéma général de compression d'images

I.3.3.1 Décorrélacion (transformation) :

La première étape du processus, est la transformation de l'information initiale en une forme plus compressible à l'aide de transformées choisies pour leur réversibilité. Cette propriété est rendue possible par la réduction des corrélacions internes, en modifiant la distribution statistique de l'information ou encore, par regroupement d'informations similaires. Plusieurs fonctions mathématiques sont utilisées pour réaliser de telles transformations, on peut citer notamment ; la transformée discrète par cosinus (Discrete Cosine Transformation - DCT), la décomposition par ondelettes ou la conversion de l'espace de couleur en un autre qui présente une corrélacion moins forte.

I.3.3.2 Quantification :

La quantification est la seule étape non réversible du schéma de compression. Elle consiste à convertir l'image transformée en une série de symboles favorables à un encodage de plus faible volume et peut être vectorielle ou scalaire.

La quantification vectorielle permet la quantification simultanée de plusieurs coefficients. Elle est coûteuse en termes de complexité calculatoire et n'est utile que si les données ne sont pas décorréliées.

La quantification scalaire est moins complexe, on distingue deux variantes :

- la quantification scalaire uniforme : Les intervalles de quantification sont tous de la même taille, elle est optimale au sens de la distorsion pour une distribution uniforme des coefficients.
- la quantification scalaire non uniforme : Les intervalles de quantification ne sont pas tous de la même taille, elle est optimale au sens de la distorsion pour une statistique marginale donnée.

I.3.3.3 Codage entropique :

La troisième et dernière étape est le codage des symboles sous une forme compressée, il génère un flux d'information par assignation d'un code binaire à chaque symbole précédemment défini. La compression proprement dite, intervient uniquement lors de cette étape.

I.4 Méthode de compression d'images :

Avec les derniers progrès dans les différents domaines de l'informatique, nous consommons de plus en plus d'images numériques. Ce qui a poussé les chercheurs à développer différentes méthodes de compression d'images, qu'on peut classer en deux groupes distincts : Les méthodes de compression réversible qui restituent une image identique à l'originale et les méthodes irréversibles qui réduisent les éléments invisibles à l'œil humain, engendrant ainsi, une perte de données de l'image.

I.4.1 Méthode de compression réversible [11] [12] [14]:

La compression est dite sans perte, lorsqu'il n'y a aucune perte de données, il y'a autant d'informations avant et après la compression. Elle est seulement réécrite d'une manière plus concise. De façon générale, ces méthodes ne font que compacter l'information présente dans l'image.

Les méthodes de compression les plus utilisées sont : le codage de Shannon-Fano, le codage de Huffman, le codage arithmétique, le codage par plan de bits, le codage prédictif, le codage RLE et le codage LZW.

I.4.1.1 Codage de Shannon-Fano :

Le codage de Shannon-Fano est la première méthode de codage entropique efficace, développée en même temps par Claude Shannon et Robert Fano en 1949. C'est le premier code qui a exploité la redondance d'une source. Il consiste à faire en sorte que les éléments binaires composant les mots code, apportent une quantité d'information moyenne, la plus grande possible.

L'algorithme de Shannon-Fano est un arbre construit à partir de la racine en suivant les étapes suivantes :

- 1) Le calcul des probabilités de chaque symbole.

- 2) Le classement des symboles source par ordre de probabilités décroissant.
- 3) la division de l'ensemble des symboles en deux sous-ensembles, de sorte à ce que les probabilités cumulées des éléments constitutifs de chacun des deux soient les plus proches.
- 4) l'attribution du chiffre binaire «0» aux éléments du sous-ensemble situé en haut et du chiffre binaire «1» aux éléments du sous-ensemble situé en bas. Ce procédé est répété pour tous les sous-ensembles comportant au moins deux éléments, jusqu'à ce que ces derniers ne comportent plus qu'un seul symbole.
- 5) L'affectation de chaque symbole par le code généré par l'algorithme.

On remarque que le codage de Shannon-Fano nécessite deux passages successifs sur les données à compresser : Le premier pour calculer les probabilités, le second pour le codage.

Remarque : Dans le codage de Shannon-Fano on peut remplacer les probabilités par les fréquences.

Exemple : Soit un message de 36 caractères : A, B, C, D et E, dont les fréquences sont:
A : 7 fois, B : 6 fois, C : 5 fois, D : 14 fois, E : 4 fois

Le tableau suivant illustre les différentes étapes de l'algorithme de Shannon-Fano appliqué à l'exemple précédent :

Symbole	fréquence	1 ^{ère} étape	2 ^{ème} étape	3 ^{ème} étape	Codage	
D	14	21	14		00	
A	7		7		01	
B	6	15	6		10	
C	5		9		5	110
E	4				4	111

Tableaux I-1 : exemple de codage de Shannon-Fano

Le codage obtenu est : A : 01 B : 10 C : 110 D : 00 E : 111

Le but du codage de Shannon-Fano est de se rapprocher le plus possible de l'entropie de l'image, mais l'inconvénient de cette approche est que, lorsqu'il n'est pas possible de séparer un ensemble de symboles en deux sous-ensembles de probabilités à peu près égales (c'est-à-dire lorsque l'un des sous-ensembles est beaucoup plus probable que l'autre), les codes produits ne sont pas optimaux. Ce qui a conduit à le supplanter rapidement par le codage de Huffman.

I.4.1.2 Codage de Huffman :

C'est un algorithme élaboré par David Albert Huffman en 1952, lors de sa thèse de doctorat au MIT. Le but de cet algorithme est de réduire la redondance de la source en affectant des codes longs pour les symboles (pixels ou caractères par exemple) rares et des codes courts pour les symboles fréquents.

Contrairement au code de Shannon-Fano, la construction du code de Huffman s'effectue de façon ascendante (des feuilles vers la racine de l'arbre). Sa mise en œuvre est décrite par l'algorithme suivant :

1) Le trie des symboles par ordre de probabilités décroissant (par exemple de la gauche vers la droite).

2) Successivement, les deux symboles de plus faible probabilité d'apparition sont rattachés à un nœud dont le poids vaut la somme des probabilités des deux symboles. A ces derniers sont affectés respectivement les chiffres binaires « 0 » pour le plus fréquent et « 1 » pour le plus rare.

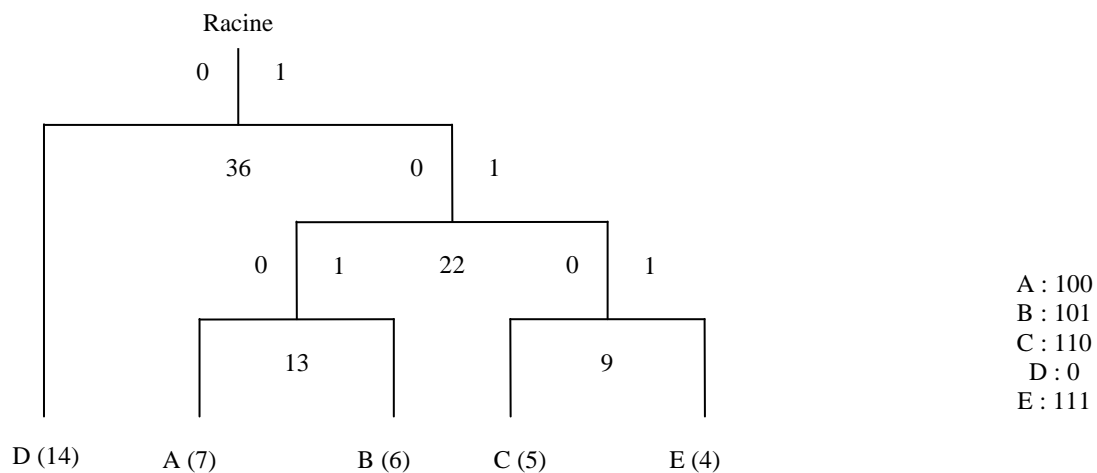
3) Le même procédé est répété en considérant les nœuds comme des symboles jusqu'à obtenir un seul nœud parent appelé racine.

4) Le code de chaque symbole s'obtient en écrivant de gauche vers la droite les chiffres binaire rencontrés en parcourant l'arbre de la racine vers le symbole. Ainsi, plus le symbole est rare plus il est profond dans l'arbre, plus son mot de code est long.

Dans le codage de Huffman, la table des codes est enregistrée dans le fichier compressé pour être utilisé dans le décodage.

Exemple : Soit un message de 36 caractères : A, B, C, D et E, dont les fréquences :

A : 7 fois B : 6 fois C : 5 fois D : 14 fois E : 4 fois



Arbre de Huffman

Table de codes

Figure I-3 : Exemple de codage de Huffman

I.4.1.3 Codage arithmétique :

Le codage arithmétique permet, à partir de la probabilité d'apparition des symboles d'une source, de créer un seul mot code qui soit associé à une séquence de longueur arbitraire de symboles. Ceci diffère du code de Huffman qui attribue des mots codes de longueurs variables à chaque symbole de la source.

Le code associé à une séquence est un nombre réel de l'intervalle $[0, 1[$. Ce dernier est construit par subdivisions récursives d'intervalles. Un intervalle est subdivisé pour chaque nouveau symbole qui appartient à la séquence. On obtient, en définitive, un sous intervalle de l'intervalle $[0, 1[$ tel que, tout nombre réel appartenant à cet intervalle représente la séquence à coder.

Soit, un ensemble de N symboles s_i de probabilité p_i . Pour coder une séquence de M symboles $S_{ki}, i \in \{1, 2, \dots, M\}$ de l'ensemble précédent par le codage arithmétique, on utilise l'algorithme suivant :

- 1) Le calcul de la probabilité p_i associée à chaque symbole S_i dans la séquence à coder.
- 2) L'initialisation d'un premier intervalle avec deux bornes : la borne inférieure $L_c = 0$, la borne supérieure $H_c = 1$.

La taille de cet intervalle est donc définie par : $T_c = H_c - L_c = 1$.

3) la division de l'intervalle initial en N sous intervalles $[L_k, H_k[$ en fonction des probabilités de chaque symbole S_k de la source. Ce sont les partitions initiales. La longueur de ces sous intervalles est donnée par : $H_k - L_k = p_k$, ou p_k est la probabilité du symbole S_k on a donc :

$$L_k = L_c + T_c \sum_{i=1}^{k-1} p_i \text{ et } H_k = L_c + T_c \sum_{i=1}^k p_i \quad (\text{I-9})$$

4) le choix du prochain sous intervalle correspondant au premier symbole S_{k1} qui apparaît dans la séquence à coder. L'intervalle initial $[L_c, H_c[$ est réinitialisé comme suit : $L_c = L_c + T_c L_{k1}$ et $H_c = L_c + T_c H_{k1}$

5) Cet intervalle est subdivisé à nouveau, selon le même procédé que celui utilisé dans l'étape 3.

6) Les étapes 3, 4 et 5 sont répétées jusqu'à obtenir le mot code représentant la séquence complète des symboles source.

Pour le décodage on utilise l'algorithme suivant :

1) L'initialisation du premier intervalle $[L_c, H_c[$ tel que $L_c = 0$ et $H_c = 1$. Sa taille est : $T_c = H_c - L_c = 1$

2) Le décodage du premier symbole S_k en trouvant le sous intervalle $[L_k, H_k[$ tel que : $L_k \leq \frac{C - L_c}{T_c} \leq H_k, 1 \leq k \leq N$ ou C est le code arithmétique de la séquence à décoder.

3) La mise à jour de l'intervalle initial $[L_c, H_c[$ comme au moment du codage de manière à décoder le second symbole : $L_c = L_c + T_c L_k$ et $H_c = L_c + T_c H_k$. La taille de cet intervalle est : $T_c = H_c - L_c$

4) Répéter les étapes 2 et 3 jusqu'à obtenir le décodage de tous les symboles de la séquence.

Exemple : Soit, l'ensemble des symboles suivants : $\{m_1, m_2, m_3, m_4\}$ de probabilités : $P(m_1)=0.3, p(m_2)=0.4, p(m_3)=0.2, p(m_4)=0.1$.

Soit à coder la séquence de symboles suivants : $m_2 m_3 m_3 m_1 m_4$. La figure suivante illustre les étapes de l'algorithme arithmétique :

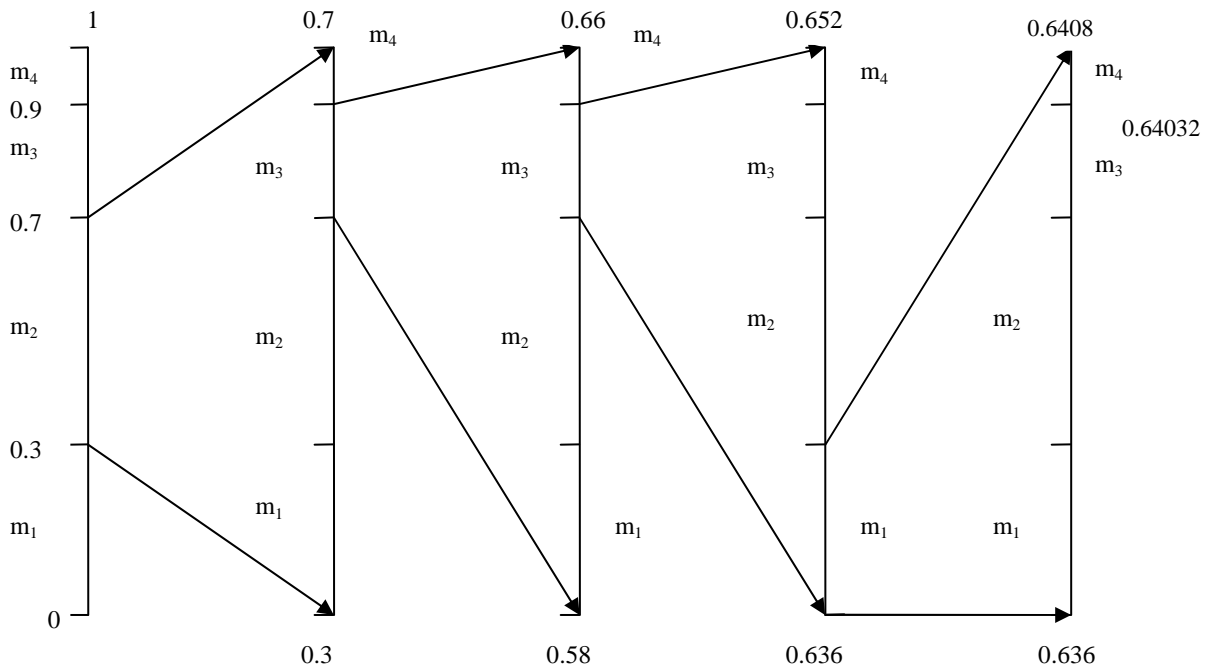


Figure I-4 : Exemple de subdivision de l'intervalle initial

La séquence $m_2 m_3 m_3 m_1 m_4$ pourra être codée par tout nombre réel appartenant à l'intervalle : $[0.64032, 0.6408[$.

Prenons par exemple le nombre réel $C = 0.6405$

Le décodage est quant à lui tout aussi simple:

On initialise les bornes de l'intervalle $[L_c, H_c[, L_c = 0$ et $H_c = 1$, et On calcule la longueur T_c de cet intervalle : $T_c = H_c - L_c = 1$.

On calcule le nombre $c_1 = \frac{C - L_c}{T_c} = 0,6405$, on a $c_1 \in [0.3, 0.7[$ donc le premier symbole est m_2 .

On met à jour le sous intervalle $[L_c, H_c[$ comme au moment du codage, de manière à décoder le second symbole :

$L_c = L_c + T_c L_k = 0 + 1 \times 0.3 = 0.3$ et $H_c = L_c + T_c H_k = 0 + 1 \times 0.7 = 0.7$. La longueur de cet intervalle est $T_c = H_c - L_c = 0.4$

On calcule le nombre $c_2 = \frac{C - L_c}{T_c} = 0,85125$. On a $c_2 \in [0.7, 0.9[$, le deuxième symbole est m_3 .

On répétant ces étapes, on obtient la séquence de tous les symboles codés.

I.4.1.4 Codage par plan de bits [4] :

Il s'agit ici, de réduire la redondance inter pixel. Le principe d'un codage en plans de bits consiste en la décomposition d'une image à m bits par pixel en une série de m images binaires, puis de compresser chacune de ces dernières, séparément.

Codé sur m bits par pixel, le niveau de ce dernier s'exprime comme un nombre binaire compris entre 0 et 2^{m-1} . On peut donc exprimer ce niveau sous forme d'un polynôme :

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \dots + a_12^1 + a_02^0$$

Chaque coefficient du polynôme (0 ou 1) correspond à la présence d'un point noir ou blanc dans un plan de niveau p .

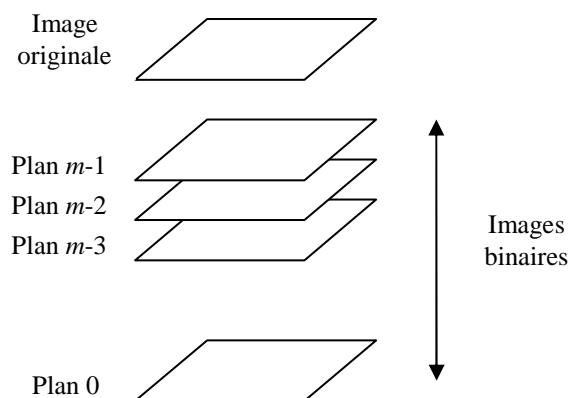


Figure I-5 : Schéma de décomposition d'une image en plans de bits

I.4.1.5 Codage prédictif [4] [14] :

Les algorithmes qui utilisent le codage par prédiction, exploitent la redondance spatiale. Il s'agit de prédire la valeur d'un pixel en fonction de la valeur de ces voisins et de ne coder que l'erreur de prédiction. Le gain en compression est accompli par la variation faible entre pixels voisins, sauf pour ceux situés sur les contours. Le voisinage peut être défini selon sa connexité (4-connexité ou 8-connexité) ou selon l'ordre du parcours choisi pour accéder aux pixels voisins. L'une des techniques de prédiction la plus simple est la DPCM (Differential Pulse Code Modulation). Cette technique effectue une prédiction à base d'une combinaison linéaire des valeurs des pixels voisins. Une version adaptative, ADPCM, qui utilise différentes formes de prédiction et de voisinage selon le contexte et le contenu de l'image qui a été présentée par Kyung et al.

La figure ci-dessous montre le diagramme fonctionnel d'un système de codage et de décodage prédictif; l'indice n correspond au rang d'un pixel.

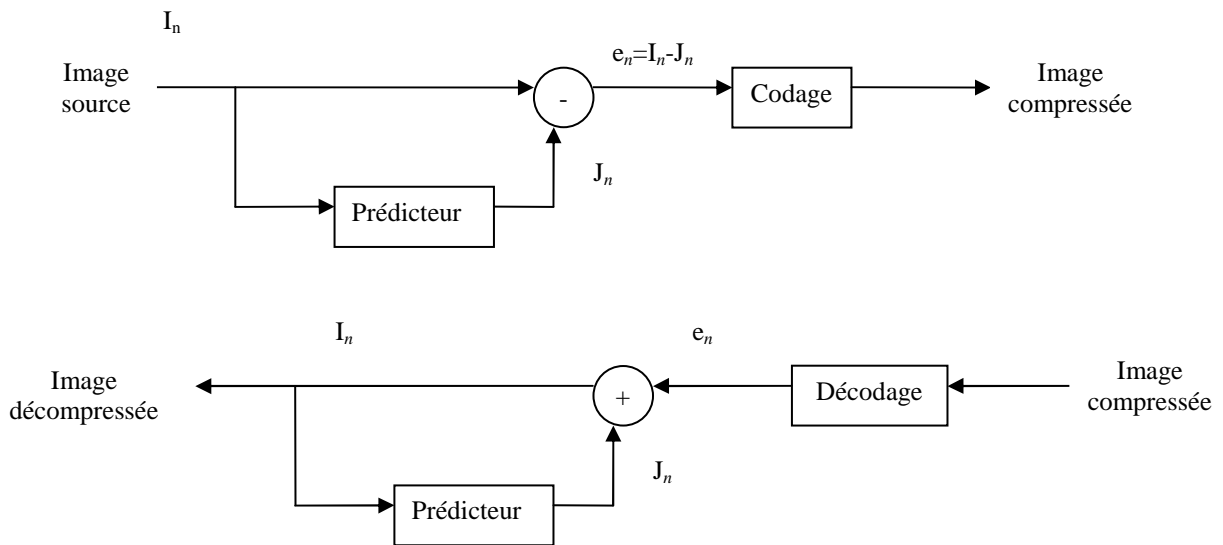


Figure I-6 : Schéma de compression et de décompression prédictif

I_n : Valeur du pixel.

J_n : Valeur du pixel prédit.

e_n : Erreur de prédiction.

I.4.1.6 Codage Run Length Encoding (RLE):

Le codage RLE aussi connu sous le nom de RLC (Run Length Coding) est recommandé lorsque nous observons des répétitions de symboles consécutifs.

L'idée générale de ce codage est de regrouper les pixels voisins ayant le même niveau de gris. Chaque groupement définit un couple de valeurs : (plage, n) où plage est le nombre de points voisins ayant la même valeur et n est cette valeur. Cette méthode n'a donc d'intérêt que pour des images possédant de nombreux éléments consécutifs redondants, notamment les images avec de larges parties uniformes.

Exemple : supposons la suite des symboles suivants : aabbccccbbbaaddd

Le codage RLE donnera la suite des couples suivants : (2,a),(3,b),(4,c),(4,b),(2,a),(3,d).

Donc le codage RLE de ces données est : 2a3b4c4b2a3d.

Le codage RLE est simple à implémenter et rapide d'exécution, il offre des taux de compression très élevés, pour des images avec de larges zones uniformes. Mais il n'est pas très efficace pour des images qui ne contiennent pas beaucoup de répétitions. Pour parer à ce problème, des règles particulières permettent de compresser lorsque les séquences comportent plus de trois répétitions d'un même élément, les autres sont conservées telles quelles, séparées des séquences compressées par un caractère spécial.

Le codage RLE code les informations contenues dans l'image en la balayant ligne par ligne, colonne par colonne ou en zigzag.

I.4.1.7 Codage Lempel-Ziv-Welch (LZW):

L'algorithme LZW a été proposé par Terry Welch en 1984 et constitue un raffinement des algorithmes LZ77 et LZ78 proposés initialement par Abraham Lempel et Jacob Ziv respectivement en 1977 et 1978. L'idée générale des algorithmes de cette famille est de parcourir la chaîne de symboles à coder et de remplacer les séquences de données par un indice en construisant au fur et à mesure un dictionnaire.

Le dictionnaire est construit dynamiquement à la lecture de la source, et n'a pas besoin d'être stocké dans le fichier compressé. En effet, l'algorithme de décompression reconstituera le dictionnaire de la même manière (par stockage d'éléments rencontrés).

La compression se fait à l'aide des éléments suivants :

- La chaîne de symboles à compresser, que l'on va parcourir linéairement.
- Un tampon dans lequel peuvent être placés des caractères en attente de codage.
- Un dictionnaire associant des chaînes de caractères à des codes numériques.

Le déroulement de la compression se résume alors en deux formules :

1) Quant le symbole lu dans la chaîne forme avec le tampon une séquence présente dans le dictionnaire, on émet le code correspondant à cette dernière et on essaie de la rallonger à l'aide du symbole suivant, pris dans la chaîne à compresser.

2) Quant le symbole lu dans la chaîne forme avec le tampon une séquence inconnue du dictionnaire, on émet le code correspondant au symbole du tampon actuel et on enrichit le dictionnaire d'une entrée pour la séquence inconnue au cas où il se représenterait et enfin, on relance la manœuvre avec un tampon vide contenant juste le dernier symbole lu.

Exemple : Voyons le déroulement de l'algorithme sur le mot "répétition" à travers le tableau suivant :

étape	Tampon	Caractère lu	Caractère émit	Code ASCII	Ajout au dictionnaire
0					
1	r	r	r	114	ré : 256
2	é	é	é	101	ép : 257
3	p	p	p	112	pé : 258
4	é	è	è	101	ét : 259
5	t	t	t	116	ti : 260
6	i	i	i	105	it : 261
7	t				
8	i	ti	ti	260	tio : 262
9	o	o	o	111	on : 263
10	n	n	n	110	

Tableaux I-2 : les étapes du codage LZW

Le code obtenu est le suivant : 114 101 112 101 116 105 260 111 110

Lors de la décompression on reconstruit le dictionnaire, il ne doit donc pas être transmis. Il est reconstruit au fur et à mesure de la lecture du fichier compressé.

Les étapes du décodage du code obtenu dans l'exemple précédent sont résumées dans le tableau suivant :

étape	Code lu (ASCII)	code latent	Mot décodé	Ajout au dictionnaire
0	114		r	
1	101	r	é	ré : 256
2	112	é	p	ép : 257
3	101	p	è	pé : 258
4	116	è	t	ét : 259
5	105	t	i	ti : 260
6	260	i	ti	it : 261
7	111	ti	o	tio : 262
8	110	o	n	on : 263

Tableaux I-3 : les étapes du décodage LZW

Notons que les codes binaires sont au début généralement émis sur 8 bits, jusqu'à ce que ces 8 bits ne suffisent plus à coder l'indice que l'on désire (par exemple l'indice 256, qu'il est nécessaire de coder sur 9 bits). On émet alors l'indice 0 pour signifier que l'on augmente le nombre de bits émis de 1.

Remarquant également, que lorsque le nombre de bits réserver pour le codage des symboles de la source ne suffit pas, l'algorithme émet le code zéro pour augmenter leur nombre de un.

Remarque : l'algorithme LZW a été breveté par UNISYS et ne pouvait pas donc être utilisé librement dans tous les pays, mais ce brevet a expiré en 2003.

I.4.2 Méthode de compression irréversible :

Afin d'atteindre des taux de compression plus importants, une compression irréversible est nécessaire. Le principe de ces méthodes de compression est basé sur l'élimination des redondances existantes dans l'image, ceci rend possible une compression plus grande au prix d'une dégradation toujours plus importante, qui affecte l'aspect visuel de l'image. Pour palier ce problème, les algorithmes de compression avec pertes éliminent les informations les moins perceptibles par l'œil humain, de manière à obtenir un résultat qui paraît identique à l'original.

Dans ce qui suit nous allons nous intéresser aux méthodes de compressions irréversibles les plus utilisées.

I.4.2.1 Méthodes par transformée :

Le codage par transformation effectue une transformation de l'image du domaine spatial dans lequel, les pixels sont fortement corrélés au sein d'un autre domaine que l'on pourrait appeler fréquentiel, ou cette corrélation est minimisée, car l'ensemble de l'information contenue dan

l'image est concentrée dans un nombre réduit de coefficients; plus précisément la transformation appliquée à l'image convertit l'ensemble des pixels en un ensemble de coefficients.

L'image supposée carrée et comportant $N \times N$ pixels est divisée en un ensemble de sous images, de taille $n \times n$. Sur chacune de ces sous images, on applique une transformation linéaire qui fournit une série de coefficients. $I(i, j)$ désignant un pixel d'une sous image et $J(u, v)$ représentant sa transformée par une transformation h , on a la relation linéaire suivante :

$$I(i, j) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} J(u, v) h(i, j, u, v) \quad (\text{I-10})$$

10)

Pour $i, j = 0, 1, \dots, n-1$

On peut aussi écrire, puisque l'ensemble des pixels de l'image initiale correspond à une matrice carrée I l'équation suivante :

$$I(i, j) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} J(u, v) H_{uv} \quad (\text{I-11})$$

$$\text{Ou : } H_{uv} = \begin{pmatrix} h(0, 0, u, v) & h(0, 1, u, v) & \dots & h(0, n-1, u, v) \\ h(1, 0, u, v) & h(1, 1, u, v) & \dots & h(1, n-1, u, v) \\ \vdots & \vdots & \ddots & \vdots \\ h(n-1, 0, u, v) & h(n-1, 1, u, v) & \dots & h(n-1, n-1, u, v) \end{pmatrix}$$

Ces transformations sont linéaires est orthogonales, ce qui veut dire que les vecteurs colonnes de la matrice H_{uv} sont deux à deux orthogonaux (leur produit scalaire est nulle). On a dans ce cas : $H_{uv}^{-1} = H_{uv}^T$ (la matrice inverse de H_{uv} est égale à sa matrice transposée).

I.4.2.1.1 La transformée de Fourier discrète (DFT) [13] [10]:

La transformée de Fourier discrète (TFD) est un outil mathématique de traitement du signal numérique, qui est l'équivalent discret de la transformée de Fourier continue qui est utilisée pour le traitement du signal analogique. Elle permet le compactage de l'énergie moyenne sur un nombre réduit de composants.

Pour une image de taille $M \times N$, la DFT à deux dimensions est donnée par l'équation suivante :

$$\begin{aligned} I_F(u, v) &= \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} I(k, l) e^{-2j\pi \left(\frac{uk}{M} + \frac{vl}{N} \right)} \\ &= \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} I(k, l) \left(\cos \left(2\pi \left(\frac{uk}{M} + \frac{vl}{N} \right) \right) - j \sin \left(2\pi \left(\frac{uk}{M} + \frac{vl}{N} \right) \right) \right) \end{aligned} \quad (\text{I-12})$$

Ou :

$I(k, l)$ est le niveau de gris du pixel situé à l'intersection de la ligne k et de la colonne l .

A partir de la transformée de Fourier, il est possible de reconstituer l'image originale en prenant la transformée inverse :

$$I(k, l) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} I_F(u, v) e^{2j\pi \left(\frac{uk}{M} + \frac{vl}{N} \right)} \quad (\text{I-13})$$

La TFD est calculée pour chaque pixel de l'image pour obtenir une image complexe, ce qui nécessite un nombre important de calculs et un temps d'exécution très long. En plus, l'espace de projection de cette transformée est complexe, ce qui réduit son utilisation.

I.4.2.1.2 La transformée de Karhunen Loeve :

La KLT (Karhunen Loeve Transformation) appliqué a une image couleur s'interprète comme un changement d'espace colorimétrique. Le nouvel espace optimise la décorrélation des couleurs et concentre l'information utile (énergie) sur un nombre minimal de coefficients.

Les coefficients de la KLT sont ceux qui contiennent la plus grande quantité d'énergie par rapport aux autres transformations, mais son utilisation pratique est difficile, car pour obtenir sa matrice de transformation, il faut calculer la matrice de covariance de l'image puis retrouver ces valeurs propres et ces vecteurs propres.

Il s'avère que la transformée en cosinus discrète permet d'obtenir des résultats comparables à ceux obtenus par la transformation optimale.

I.4.2.1.3 La transformation en cosinus discrète (DCT) [10] [12] [13] :

La transformation en cosinus discrète est une méthode proposée par Ahmed Natarajan et Rao en 1974, c'est une transformation linéaire qui est proche de la transformation de Fourier discrète (DFT), car elle permet de passer d'un domaine spatial à une représentation équivalente dans le domaine fréquentiel. La différence est que le noyau de projection utilisé pour la DFT est représenté par une exponentielle complexe alors que celui de la DCT (Discrete Cosine Transform) est simplement une base de cosinus. Les coefficients de la transformée ne sont donc pas complexes, mais réels, ce qui présente un avantage pour le codage et la quantification.

Dans le cas des images, plus la matrice sur laquelle la DCT est appliquée est grande plus le nombre d'opérations augmente, ce qui nécessite le découpage de l'image en matrices (bloc) de taille plus petite (8×8 ou 32×32 par exemple). Le nombre d'opérations augmente, non linéairement avec la taille de la matrice. Si on prend une taille plus petite, la qualité diminue. La matrice 8×8 est un bon compromis entre qualité et nombre d'opérations.

On utilise donc une DCT bidirectionnelle qui transformera un bloc de $N \times N$ pixels en un autre bloc de $N \times N$ coefficients, correspondant à l'amplitude des fonctions harmoniques dans l'espace transformé, ces coefficients sont calculés à l'aide de l'équation suivante:

$$T(u, v) = \frac{2}{N} c(u)c(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i, j) \cos\left(\frac{(2i+1)\pi u}{2N}\right) \cos\left(\frac{(2j+1)\pi v}{2N}\right) \quad (\text{I-14})$$

$$\text{Ou: } c(u), c(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{si } u = v = 1 \\ 1 & \text{si non} \end{cases}$$

Avec $u, v \in \{0, 1, \dots, N-1\}$

$I(i, j)$: La valeur du pixel de coordonnées i et j .

$T(u, v)$: La valeur du coefficient de la DCT.

La valeur du coefficient $T(0,0)$ est appelée le coefficient continu, il est proportionnel à la valeur moyenne des pixels du bloc de l'image.

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, on retourne à une représentation sous forme de pixels. La formule qui donne la transformée inverse est la suivante :

$$I(i, j) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u)c(v)T(u, v) \cos\left(\frac{(2i+1)\pi u}{2N}\right) \cos\left(\frac{(2j+1)\pi v}{2N}\right) \quad (\text{I-15})$$

Dans la matrice, après application de la DCT, les basses fréquences se trouvent en haut à gauche et les hautes fréquences en bas et à droite. Les hautes fréquences représentent les zones à forts contrastes dans l'image, ce qui signifie un changement brutal de couleurs. Une image classique admet une grande continuité entre les valeurs des pixels. Les hautes fréquences étant réservées à des changements rapides d'intensité de ces derniers, ceux-ci sont en général minimales dans une image. Ainsi, on parvient à représenter l'intégralité de l'information sur très peu de coefficients, correspondant à des fréquences plutôt basses. Donc le maximum d'informations sur le bloc de l'image se trouve concentré sur la partie supérieure à gauche de la matrice.

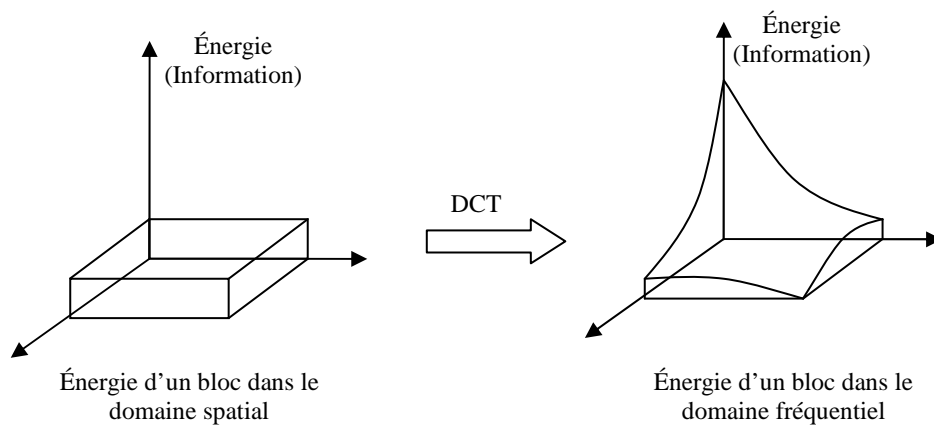


Figure I-7 : La représentation de l'évolution de l'énergie d'un bloc de l'image après l'application de la DCT

La DCT est la transformation qui se rapproche le plus de la transformée de Karhunen-Loève, qui fournit une décorrélation optimale des coefficients.

I.4.2.1.4 La transformation en ondelettes [4] [5] :

La transformée en ondelettes correspond à une analyse multi-résolution de l'image. Cette dernière est décomposée en un ensemble de sous bandes (images de résolution inférieure) représentant l'information portée par celle-ci. L'algorithme de Mallat permet de décomposer une image en quatre sous bandes : L'une représentant la version lissée et réduite de l'image source (basse résolution), ce qui correspond aux basses fréquences de cette dernière, tandis que les trois autres contiennent les informations de hautes fréquences, ce qui correspond aux détails verticaux, horizontaux et diagonaux de l'image originale, donc, les différents contours de celle-ci. La décomposition est réalisée par un ensemble de filtres orthogonaux passe-bas (FB) et passe-haut (FH). Ce processus est répété à plusieurs reprises en décomposant à chaque fois l'image basse fréquence, on accède ainsi, à des représentations de plus en plus grossières de l'image originale. Les

sous bandes de résolutions supérieures possèdent un contenu qui est relativement faible, alors que les sous bandes basse fréquences sont beaucoup plus riches, ce qui permet d'employer des algorithmes de quantification performants.

Le schéma suivant représente les étapes de décomposition d'une image à la résolution 1

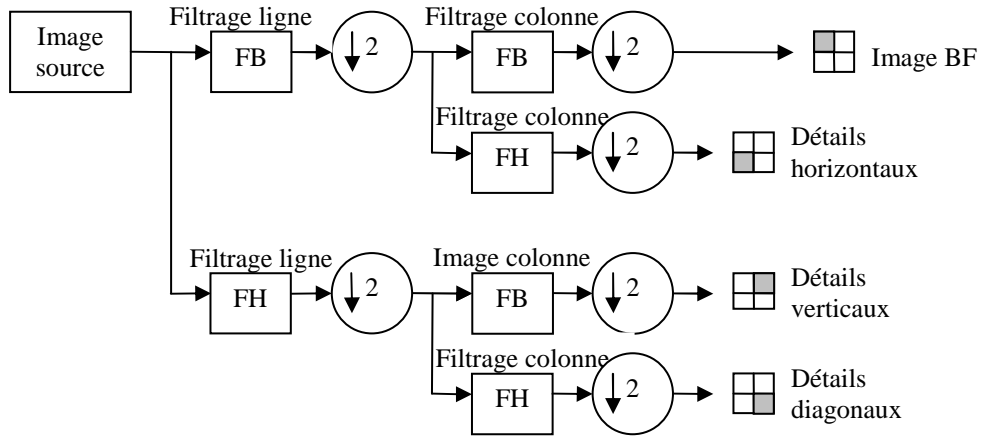


Figure I-8 : Décomposition multi-résolutions d'une image

La reconstruction de l'image originale à partir des différents sous bandes, nécessite également l'utilisation de filtres orthogonaux passe-bas et passe-haut. Le schéma suivant illustre les différentes étapes de ce processus :

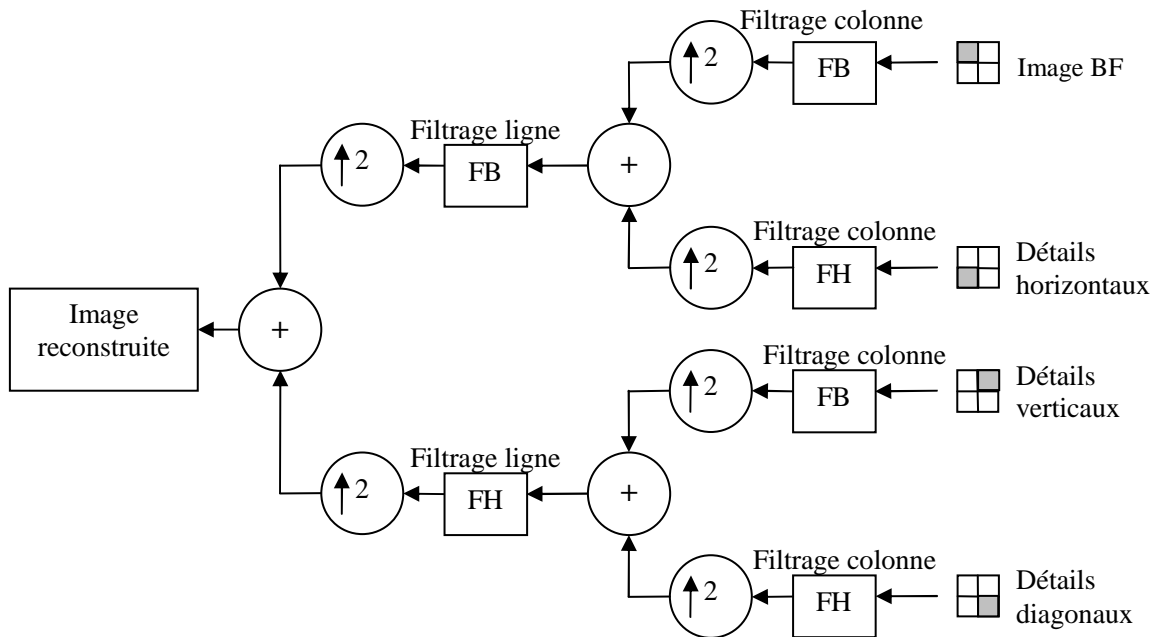


Figure I-9 : Reconstruction de l'image

L'avantage de la représentation multi-résolutions réside dans la dualité contenue fréquence. Contrairement aux autres transformées qui projettent le signal dans l'espace des fréquences, la transformée en ondelettes représente l'image conjointement dans son espace réel et dans son domaine fréquentiel. Les propriétés topologiques (orientations, agencement du contenu) sont ainsi conservées après la transformation multi-résolutions. Le codeur qui est basé sur une telle

transformation, peut ainsi, prendre en compte la redondance spatiale et fréquentielle de l'image et être plus efficace.

Remarque : La transformée en ondelettes fait partie des transformations en sous bandes, qui consistent à séparer les images en un ensemble de sous bandes, en utilisant des bancs de filtres puis de codés ses dernières en privilégiant ceux qui contiennent les basse fréquences riche en informations.

I.4.2.1.5 Fractales [8] [13]:

La géométrie fractale est une branche de la géométrie, qui cherche à décrire mathématiquement des objets ayant une forme complexe, très rugueuse ou fragmentée, telle qu'en on trouve dans la nature.

Un fractal est une structure géométrique qui se reproduit, dans une boucle infinie, par transformation affine (translation, rotation et mise à l'échelle). Cette structure se refait à toutes les échelles de forme réduite et légèrement déformée.

La compression par fractale est une technique de compression avec perte, développée par Michael Barnsley en 1986, elle consiste à coder une image comme une suite de transformations. En effet, les données d'une image contiennent généralement un ensemble plus ou moins grand d'éléments redondants (partie similaire sur l'image ou autosimilarités). Etant inutile de stocker ces éléments autant de fois qu'ils sont présents, la taille de l'image décroît.

Les étapes de la compression fractale (méthode de I Jacquin) sont les suivantes :

1) Partitionnement de l'image en m blocs cible R de taille $n \times n$ pixels (ces blocs ne peuvent pas se chevaucher).

2) Construction d'une liste de blocs source D de taille $2n \times 2n$ extraite de l'image (ces blocs peuvent se chevaucher).

3) Déterminé pour chaque bloc source, le meilleur couple (bloc source, bloc destination) minimisant une erreur. Cette erreur est généralement calculée en soustrayant les deux blocs. Pour réaliser l'opération de soustraction, il est nécessaire d'opérer une transformation du bloc source aux dimensions et à la géométrie du bloc destination. Pour cela on utilise les IFS (Iterated Function Systems), système de fonctions itérées. Un IFS est en fait un ensemble de transformations géométriques (rotations, translations...) répétées. Les transformations géométriques doivent être nécessairement contractantes, c'est à dire que le résultat est inclus dans le plan limité de départ (l'image est plus petite ou identique). Ces dernières possèdent la propriété remarquable de converger vers un élément fixe indépendamment de leur initialisation. Le point fixe est appelé attracteur (proche du bloc de l'image à coder).

4) Une fois que tous les couples qui minimisent l'erreur d'approximation ont été trouvés, le fichier de sortie contient alors ces différents couples, ainsi que les transformations effectuées.

Pour décoder l'image à partir de son code IFS, l'algorithme procède comme suit : L'image initiale est partitionnée en blocs carrés ($n \times n$), chacun d'eux est calculé à partir du bloc source et de la transformation affinée contractive, définies lors de l'étape de codage. Ainsi, l'image finale est obtenue après la reconstruction de tous les blocs.

Ce genre de compression permet un puissant zoom fractal, pour agrandir une image sans effet de pixellisation et son taux de compression est très avantageux, tout en conservant un aperçu fidèle. Mais elle reste toujours peu utilisée à cause de la lenteur de sa phase de compression, malgré toutes les améliorations.

I.4.2.2 Quantification [14] :

La quantification réduit la quantité d'information contenue dans l'image, en produisant des plages de valeur identique s'apprenant au codage, de façon à ce que cette réduction n'apporte pas de dégradations visuelles. On distingue deux types de quantifications : la quantification scalaire (QS) et la quantification vectorielle (QV).

I.4.2.2.1 Quantification scalaire (QS) :

La quantification scalaire est une application Q qui associe à une variable continue X une variable discrète Y .

Pour réaliser une bonne quantification on suit les étapes suivantes :

1) On fixe le nombre de niveaux de quantification N , ainsi l'ensemble des valeurs discrètes Y est défini : $Y = \{y_i ; i = 0, \dots, N-1\}$ ou $S_i = [s_i, s_{i+1}[$

2) On divise l'ensemble des valeurs continues X en N intervalles. On aura un ensemble d'intervalles : $S = \{S_i ; i = 0, \dots, N-1\}$

3) On associe à chaque valeur continue x appartenant à l'intervalle $S_i = [s_i, s_{i+1}[$ la valeur quantifiée y_i .

L'application Q est donc définie par la relation suivante :

$$Q: X \rightarrow Y$$

$$x \mapsto Q(x)$$

$$\text{Où } Q(x) = \begin{cases} y_0 & \text{si } x \in S_0 \\ y_1 & \text{si } x \in S_1 \\ \vdots & \\ y_{N-1} & \text{si } x \in S_{N-1} \end{cases} \quad (\text{I-16})$$

D'une manière générale, la largeur d'un intervalle L_i n'est pas constante: $L_i = s_{i+1} - s_i \neq \text{cste}$. Cette largeur est appelée le pas de quantification.

La quantification scalaire est uniforme lorsque les intervalles sont de longueur constante et les valeurs de sorties sont les milieux des intervalles de quantification : $y_i = \frac{s_i + s_{i+1}}{2}$. Le pas de quantification est donc fixe : $L_1 = L_2 = \dots = L_{N-1}$

Dans les autres cas, la quantification est non uniforme.

La quantification est dite à zone morte, quand l'intervalle autour de zéro est plus large. Ce dernier, désigne donc cette zone, qui permet à l'ensemble des valeurs de source considérées comme petites, d'être quantifiées à une seule et même valeur (généralement zéro). Ce type de quantification est donc non uniforme (ou asymétrique). Toutefois, si l'ensemble des autres intervalles sont uniformes, on qualifie généralement ce type de quantification d'uniforme à zone morte.

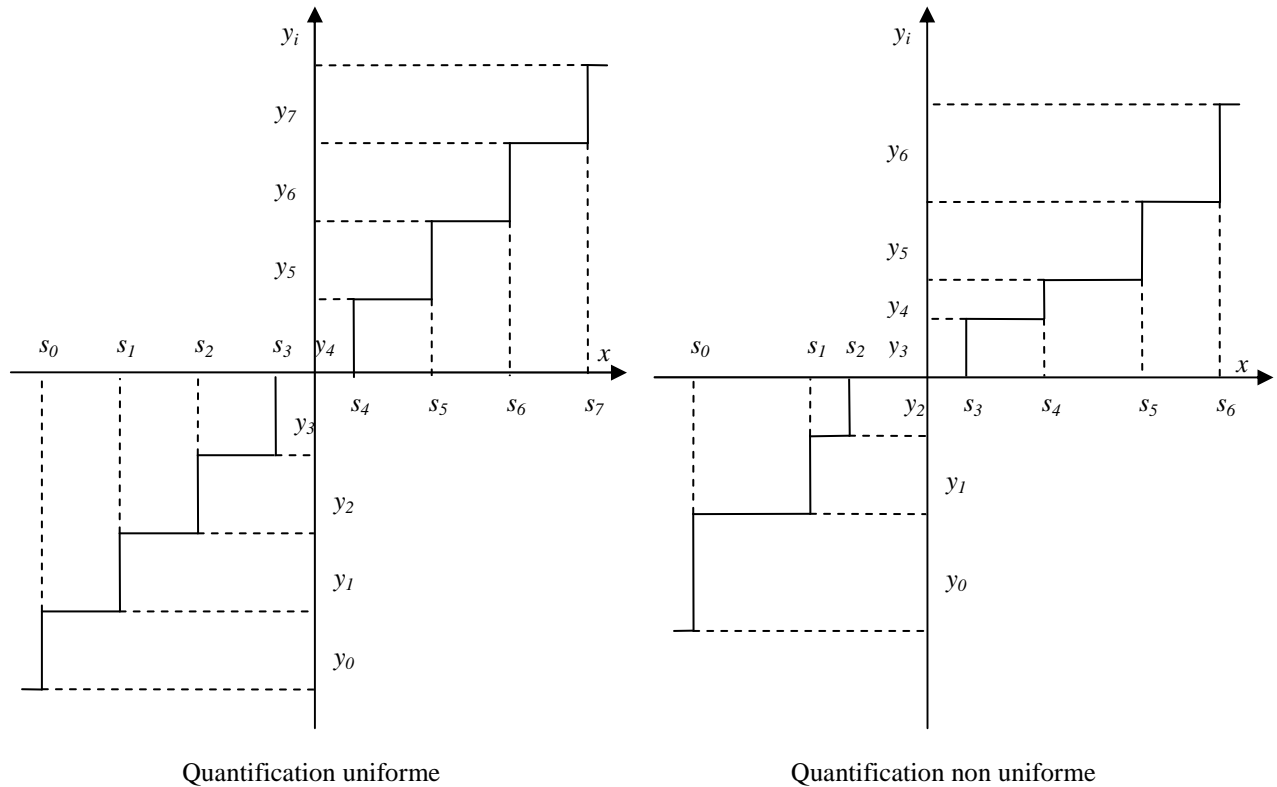


Figure I-10 : Graphes de Quantification

I.4.2.2.2 Quantification vectorielle (QV) :

La quantification vectorielle QV a été développée par Gersho et Gray en 1980. Son principe est issu du travail de Shannon, qui montre qu'il était toujours possible d'améliorer la compression de données, en codant non pas des scalaires mais, des vecteurs.

La quantification vectorielle dans son sens le plus large est l'approximation d'un signal d'amplitude continue par un signal d'amplitude discret.

D'un point de vue mathématique, un quantificateur vectoriel de dimension k et de taille M peut être défini comme une application Q de R^k vers D .

$$Q: R^k \rightarrow D$$

$$x \mapsto y_i = Q(x)$$

Ou :

R^k : Est un ensemble de vecteur x de dimension k .

D : Est un ensemble fini de M vecteurs y_i appeler dictionnaire.

$$D = \{y_i \in R^k ; i = 0, \dots, M - 1\}$$

Cette application Q détermine implicitement une partition de l'espace source R^k en M régions C_i entourant les vecteurs y_i . Ces dernières sont encore appelées classes, déterminées par : $C_i = \{x \in R^k ; Q(x) = y_i\}$

Chaque région contient un ensemble de vecteurs de R^k et tous les vecteurs x qui appartiennent à C_i sont représentés par le même vecteur y_i du dictionnaire.

Pour la compression, la quantification vectorielle se déroule en deux étapes : le codage et le décodage.

Le rôle du codeur consiste à découper la matrice de pixel de l'image ou les coefficients du domaine transformée (fréquentiel), en un ensemble de bloc (vecteurs), puis de comparé chacun avec l'ensemble des M vecteurs du dictionnaire D et de le remplacer par le plus proche. C'est uniquement l'adresse du vecteur sélectionnée qui sera transmise ou stockée.

Le décodage opère en remplaçant l'adresse reçue par le vecteur correspondant dans le dictionnaire, car le décodeur dispose d'une réplique du dictionnaire du codage.

La notion de proximité utilisée dans le codage, encore appelé mesure de distorsion entre les vecteurs x et y_i , elle correspond à la distance entre x et y_i au sens de la norme euclidienne, on a alors :

$$d(x, y_j) = \sqrt{\left(\sum_{j=0}^k (x_j - y_j)^2 \right)} \tag{I-17}$$

Où les $x_j, j=1,2,\dots,k$ sont les coordonnées du vecteur x et les $y_j, j=1,2,\dots,k$ sont les coordonnées de y_j

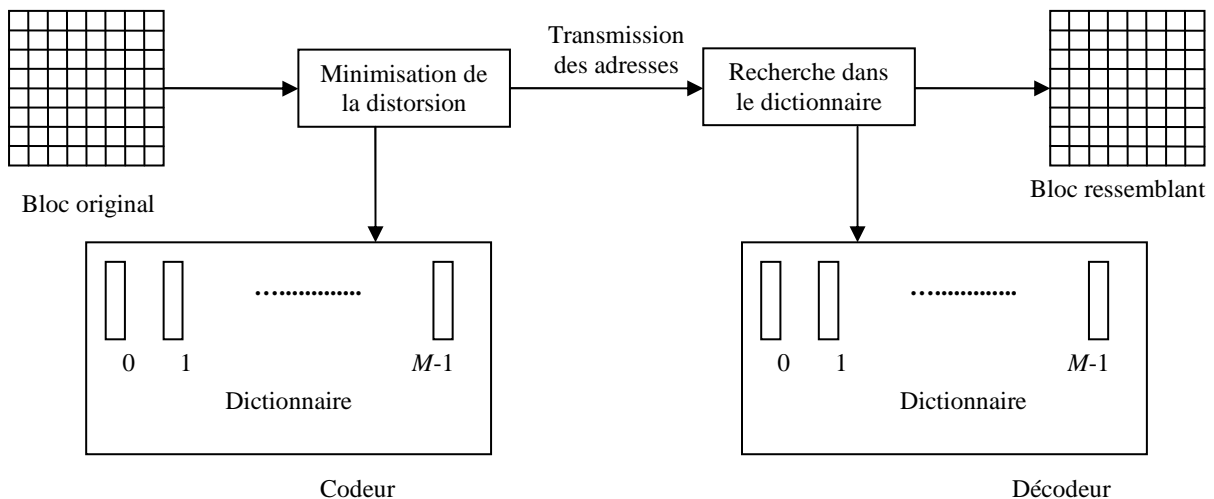


Figure I-11 : Schéma général du principe de la quantification vectorielle

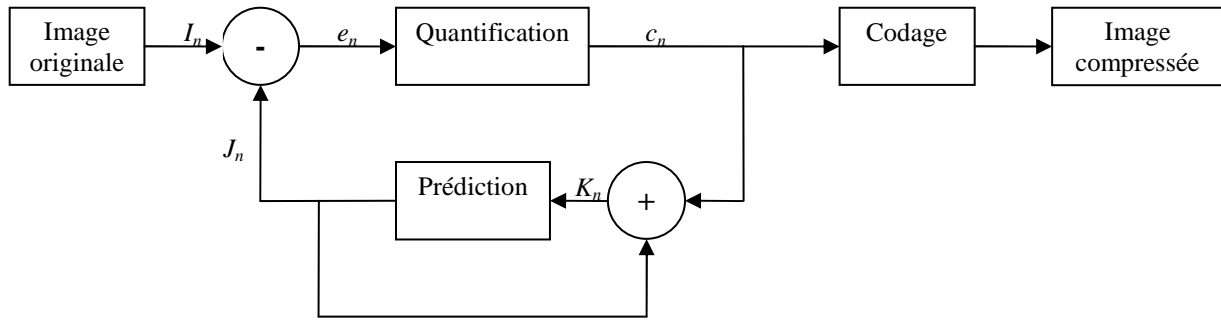
Un dictionnaire est défini par deux paramètres principaux :

- 1) Le nombre de vecteurs le constituant : M
- 2) La dimension de l'espace vectoriel : k

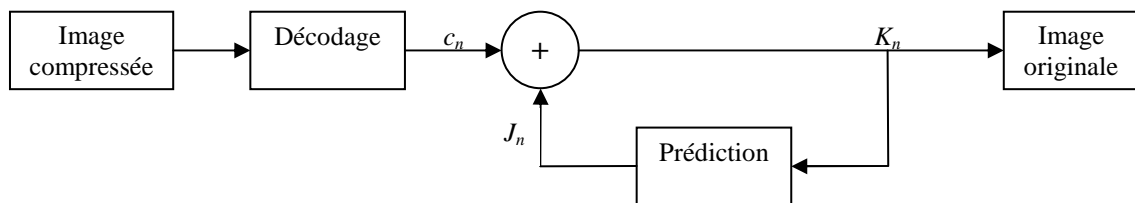
I.4.2.3 Codage par DPC (Différentiel Prédictive Coding) [14] :

Le codage par DPC exploite la redondance spatiale existante dans l'image. Il s'agit de prédire la valeur d'un pixel en fonction de celle de ces voisins et de ne coder que l'erreur de prédiction.

La différence entre cette technique et la prédiction sans perte, est que l'erreur de prédictions est quantifiée avant d'être codée. Le schéma suivant résume les étapes du codage DPC :



Compresseur DPC



Décompresseur DPC

Figure I-12 : Schéma du codage et de décodage DPC

Ou :

I_n : Valeur du pixel original.

K_n : Valeur du pixel reconstruit.

J_n : Valeur du pixel prédit.

$e_n = I_n - J_n$: Erreur de prédiction.

$c_n = K_n - J_n$: Erreur de quantification.

I.4.2.4 Méthodes hybrides :

Les méthodes de compression hybrides sont une combinaison des techniques du domaine transformée (fréquentiel) et celle du domaine spatial. La combinaison de ses deux techniques permet de regrouper leur avantage, ce qui contribue à l'amélioration du taux de compression, l'obtention d'images de restitution de meilleure qualité et l'amélioration du temps d'exécution de ses algorithmes.

I.5 Les normes de compression :

Une norme de compression d'image est une représentation informatique de celle-ci, associée à des informations sur la façon dont, elle est codée et fournissant éventuellement des indications sur la manière de la décoder et de la manipuler.

I.5.1 La norme JPEG [10] [14] :

Le comité JPEG (Joint Photographic Expert Group) a été créé en 1986 par la jonction de deux groupes, qui travaillaient sur la photographie, le groupe ISO (International Standards Organisation) et le groupe CEI (Commission Electronic International). Ce comité a produit la norme de compression d'images qui a été standardisée (ISO/IEC/10918-1/1994) et a reçue son nom JPEG.

La norme JPEG possède quatre modes de fonctionnement : séquentiel (baseline), progressif (extended

DCT-based), sans perte (Lossless) et hiérarchique (hierarchical). Entre ces quatre modes de compression, le séquentiel est le mode principal le plus répandu, ces étapes sont :

1) Le format de codage de couleur, qui est en général du RVB est converti au format YCbCr. Ce dernier n'est plus basé sur les trois couleurs, mais, sur la luminance (Y) et la chrominance (Cb et Cr). Or, physiquement, l'œil humain est peu sensible à la chrominance. On peut donc sous échantillonner les composants Cb et Cr, ce qui se traduit par une diminution de l'information et donc de la taille du fichier. Cette étape de changement d'espace colorimétrique est facultative ; néanmoins, elle donne les meilleurs taux de compression.

2) Découpage de l'image par bloc de 8x8 pixels, et application d'une DCT a chacun d'entre eux. Ce qui permet de concentrer l'information sur les coefficients de basse fréquence, situé en haut et à gauche du bloc transformée. Le premier coefficient de la DCT est appelé DC, les 63 autres sont appelés AC.

3) La compression avec pertes est faite dans l'étape de quantification, qui est réalisée à l'aide d'une matrice Q de quantification de 8x8 éléments. L'image subit des distorsions selon le niveau de compression désiré. Chaque coefficient DCT est divisé par la valeur correspondante dans Q et le résultat est arrondi à l'entier le plus proche. L'acuité du système visuel humain est plus faible à des hautes fréquences et sensible aux basses fréquences. Quelques tables standard, pour la quantification, ont été générées grâce à une série de caractéristiques psychos visuelles. Les valeurs prennent donc, en compte cette caractéristique et introduit majoritairement de la distorsion dans les hautes fréquences. La quantification ramène beaucoup de coefficients à zéro, surtout en bas et a droite dans la matrice, la ou sont les hautes fréquences. Seules quelques informations essentielles (coin en haut à gauche) sont gardées pour représenter le bloc. Il faut remarquer que plus la compression demandée est forte, plus des informations seront supprimées, mais la qualité en sera ressentie.

4) Après la quantification, la matrice contient un grand nombre de zéro. Une compression du type RLE est appliquée en parcourant la matrice en zigzag. Ce parcours récupère le plus grand nombre de zéro d'affilés, ce qui permet d'augmenter le taux de compression. Après cette étape, un codage Huffman est appliqué, ces deux compressions sont non destructives ; donc il n'y a pas de perte d'information.

Pour la décompression, on réalise simplement la méthode inverse : décodage, DCT inverse et la Déquantification.

Les trois autres types de codage sont une extension de celui de base, ils diffèrent de ce dernier principalement dans l'emploi du codage entropique en utilisant un codeur arithmétique ou par restitution progressive de l'image.

Il est à noter qu'il existe une forme de compression JPEG sans perte, elle est basée sur le codage prédictif suivi d'un codage entropique. Bien que, peu utilisé par la communauté informatique en général, il sert surtout pour la transmission d'images médicales. La compression est alors beaucoup moins efficace (taux de 50 % environ).

Il faut également signaler des versions dérivées de JPEG adaptées au codage de la vidéo animée, connue sous le nom de M-JPEG (Moving JPEG ou Motion JPEG). La méthode retenue consiste en un codage de chaque image selon le procédé JPEG, avec un taux de compression optimal, compte tenu de la bande passante disponible. Ces procédés souffrent cependant d'une absence de normalisation.

La norme JPEG présente trois défauts majeurs :

- Quand les taux de compression sont élevés, l'image compressée est fortement dégradée et les blocs sont visibles à l'œil nu. Ce qu'on appelle effet de pixellisation.
- l'algorithme de compression nécessite une puissance de traitement égale pour la compression et la décompression.
- la norme JPEG n'impose pas de format de fichier, ce qui ne résout pas tous les problèmes de compatibilité.

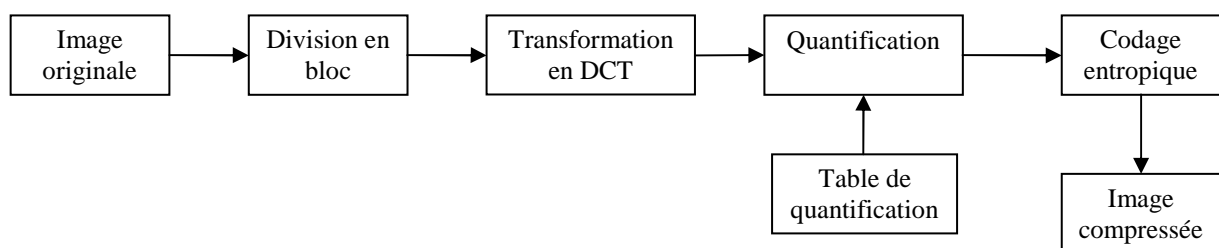


Figure I-13 : Schéma de compression JPEG

I.5.2 La norme JPEG 2000 [4] [6] :

L'appel à proposition pour la nouvelle norme de compression du troisième millénaire a commencé en mars 1997.

La version définitive du standard a pris forme en décembre 2000. Ce nouveau standard a pour objectif d'offrir de nouvelles fonctionnalités permettant de répondre aux exigences des nouvelles applications, les plus diversifiées.

Le codage et décodage d'une image sous JPEG 2000 s'effectuent en quatre étapes principales : les trois étapes classiques en compression d'image (Transformation, Quantification, Codage) plus

une étape de prétraitement de l'image, qui a pour but de rendre l'opération de codage plus efficace. Ces différentes étapes sont :

1) L'étape de prétraitement consiste à transformer les valeurs des pixels en valeurs signées. Typiquement, on passe d'une représentation de $[0, 255]$ à $[-128, 127]$, afin d'avoir un signal centré autour de zéro. Puis, l'image est découpée en un pavage de tuiles (traduction de tile). Le but de cette opération est de réduire la complexité de l'algorithme pour des images de très grande taille mais aussi de faciliter la navigation à l'intérieur de celles-ci. Ces tuiles sont de tailles carré (64x64) ou (128x128) par exemple. Cette étape comporte également une transformation de l'espace des couleurs RVB en l'espace YCrCb, cette dernière permet également de coder les couleurs avec un nombre de bits inférieur.

2) La deuxième étape consiste à appliquer la transformée en ondelettes sur l'image ou sur ces tuiles, le but de cette transformation est de décomposer l'image suivant différentes résolutions. On obtient ainsi une sous bande lissée (basse fréquence) et des sous bandes détaillées (haute fréquence) selon les directions horizontales, verticales et diagonales.

3) Les sous bandes de résolutions supérieures possèdent un contenu qui est relativement faible, alors que celles de basses fréquences sont beaucoup plus riches. L'algorithme de compression JPEG 2000 adopte une quantification linéaire pour chacune d'elles. Le pas de quantification utilisé est cependant beaucoup plus faible pour les sous bandes basses fréquences, qui contiennent beaucoup d'information, que pour les sous bandes hautes fréquences.

4) 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 (les bits de poids forts) sera codée dans un premier temps, alors que les détails (les bits de poids faibles) sont codés après. Pour résumer, 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 les bits parent est non nuls.
- une étape qui code le reste des bits non nuls.

Une autre fonctionnalité importante du JPEG2000 est la compression par région d'intérêt. Ceci permet d'avoir des taux de compression différents dans certaines régions de l'image. Les zones importantes peuvent être compressées quasi sans pertes et les zones moins importantes avec un fort taux de compression.

Dans le cas d'une compression sans pertes, c'est l'ondelette de Gall qui est utilisée, sinon c'est l'ondelette de Daubechies.

Une variante de cette norme a aussi vu le jour, le MJ2 (Motion JPEG 2000). Celle-ci utilise aussi la compression par ondelettes, mais pour des images en mouvement, c'est à dire pour la vidéo numérique.

I.5.7 La norme MPEG [4] [10] :

Les réunions du groupe MPEG (Moving Picture Experts Group) ont démarré en 1988 dans le but de développer une première norme, MPEG-1 (ISO/CEI 11172-1 à 5), pour des applications de

stockage audio vidéo du type Vidéo CD. MPEG a ensuite rapidement produit une nouvelle série de normes internationales, MPEG-2, ciblant essentiellement les applications liées à la télévision numérique, d'autres séries de normes ont depuis été produites. MPEG rassemble de nombreux spécialistes provenant de l'industrie de l'électronique des composants, de l'informatique et des télécommunications.

Les spécifications de formats produits par MPEG sont ouvertes, mais non libres : leur utilisation est soumise au paiement de redevances.

Les codeurs MPEG sont basés sur la prédiction par compensation de mouvement et le codage différentiel. Les images codées de la séquence sont décomposées en trois types différents.

1) Les images de type intra I qui seront codées sans prédiction et sans compensation de mouvement par un algorithme de compression très proche de celui de JPEG.

2) Les images de type prédites P qui seront prédites à partir des images I ou P précédentes, en utilisant une compensation de mouvement basé sur le codage des vecteurs mouvements, indiquant le déplacement des différents éléments de l'image référence. Pour améliorer la qualité de reconstruction, la différence entre l'image originale et l'image prédite sera également codée, le codage utilisé pour coder cette différence sera le même que celui utilisé pour coder les images I.

3) Les images de type bidirectionnelles B qui seront également prédites avec deux compensations de mouvement, l'une provenant d'une image I ou P passée, l'autre provenant d'une image I ou P future. On ne peut donc la décoder qu'après avoir lu un peu plus loin le fichier. Comme pour les images P, la différence entre l'image prédite et l'image originale de la séquence sera également codée.

La succession des images de type I, P et B forme un groupe d'images appelées GOP (group of picture).

De part la structure des images codées (I, B et P), le fichier compressé MPEG comporte deux catégories d'informations :

- les informations images, qui sont les images I codées ou les images d'erreur provenant des images P et B.

- les informations mouvement, qui sont les vecteurs de mouvement nécessaires à la compensation de mouvement.

Le fonctionnement du codeur vidéo MPEG est représenté sur la figure suivante :

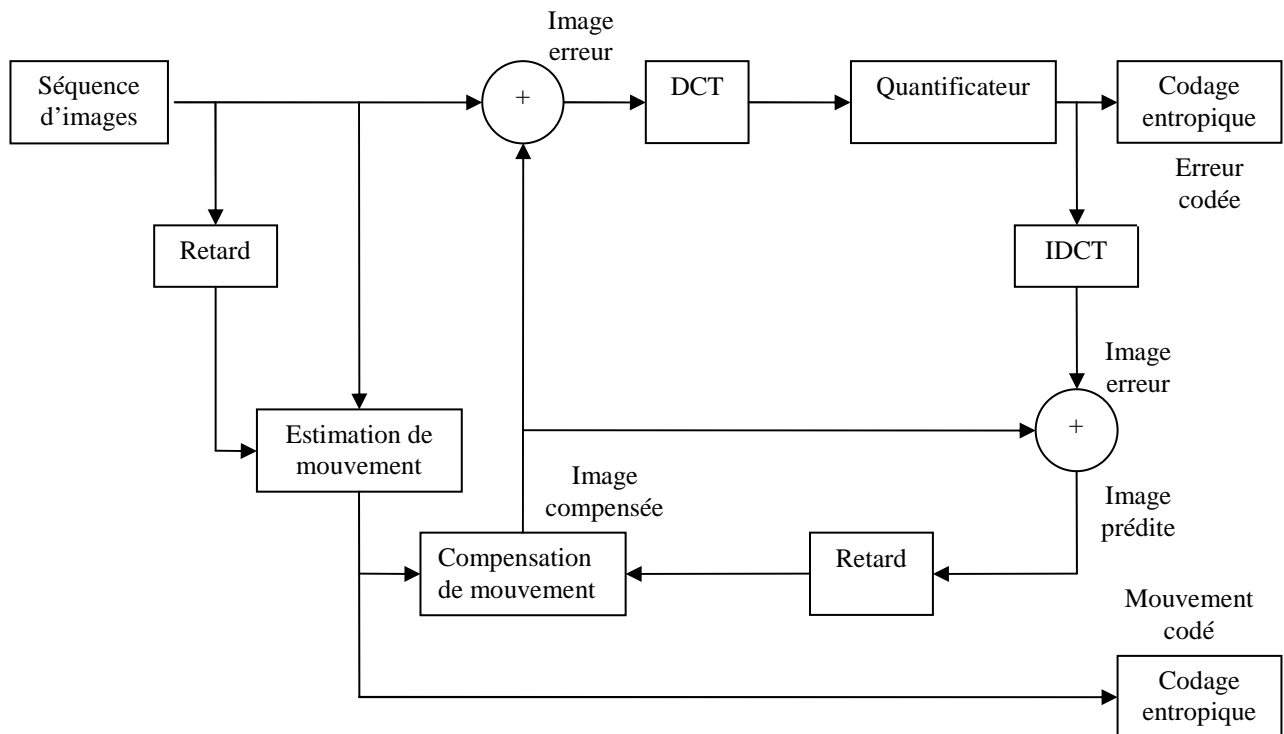


Figure I-14 : Structure du codeur MPEG

MPEG a développé les normes suivantes :

MPEG-1 : première norme audio et vidéo, utilisé plus tard pour les Vidéo CDs. Ce format offre une résolution à l'écran de 352×240 pixels à 30 images par seconde ou de 352×288 à 25 images par seconde, avec un débit d'environ 1,5 Mbit/s. Elle comprend le populaire format audio MPEG-1 partie 3 audio couche 3 (MP3).

MPEG-2 : norme applicable au codage de l'audio et la vidéo, ainsi que leur transport pour la télévision numérique : télévision numérique par satellite, télévision numérique par câble, télévision numérique terrestre, et (avec quelques restrictions) pour les vidéo disques DVD ou SVCD. C'est notamment le format utilisé jusqu'à présent pour la TV sur ADSL. Les débits habituels sont de 2 à 6 Mbit/s pour la résolution standard (SD), et de 15 à 20 Mbit/s pour la haute résolution (HD)

MPEG-III : MPEG-III est destiné à la télévision haute définition. Ses débits varient entre 30 et 40 Mbits/s.

MPEG-4 : MPEG-4 a pour objectif de coder de manière efficace des séquences à très bas débit. Il permet également d'augmenter l'interactivité d'une scène animée, la séquence pouvant être décomposée en plusieurs Objets Vidéo qui sont codés de manière indépendante. Chaque objet vide est codé par une information de texture et une information de masque permettant de définir la forme de l'objet. Les différents objets vidéo peuvent être agencés dans une scène en utilisant un compositeur.

L'algorithme de compression MPEG-4 des séquences naturelles est déjà utilisé actuellement sous des fichiers au format DivX ;).

Ses débits sont constants ou variable : pour des niveaux faible, on a moins de 64 kbits/s, entre 64kbits/s et 384kbits/s pour les niveaux intermédiaires et entre 384kbits/s et 4Mbits/s pour les haut niveaux.

CHAPITRE II

Les images Météosat seconde génération (MSG)

II.1 Introduction :

Dans notre étude nous allons compresser des images Météosat seconde génération MSG, pour mieux comprendre leurs propriétés et leurs différentes applications nous allons nous intéresser dans ce chapitre aux divers constituants des satellites MSG et les mécanismes d'acquisitions de ces images, ainsi qu'aux phénomènes physiques et optiques intervenant.

II.2 Les satellites météorologiques [2] :

Les satellites artificiels sont des engins placés en orbite autour de la terre, ou d'un astre quelconque, à des fins scientifiques, militaires, industrielles ou économiques.

II.2.1 Les satellites météorologiques Météosat:

Les satellites météorologiques permettent de réaliser des images de la terre et d'alimenter les stations météorologiques, de façon à prévoir l'évolution des phénomènes naturels. Il existe deux types de satellites météorologiques : les satellites géostationnaires et les satellites à défilement. Les premiers sont immobiles pour un observateur terrestre, car ils ont une trajectoire équatoriale circulaire à environ 36 000 km au-dessus de l'équateur, ils observent donc en continu une portion du globe terrestre ; les satellites à défilement ont en général des orbites beaucoup plus basses et survolent un grand nombre de régions du globe.

II.2.2 Les satellites Météosat :

Les satellites Météosat est une famille de satellites météorologiques géostationnaires de l'Agence spatiale européenne (ESA), dédiés à la prévision météorologique immédiate et à l'évaluation, à long et moyen terme, du climat. Il existe deux générations de satellites météoat : la première génération et la deuxième génération.

II.2.2.1 La première génération (Météosat 1 à 7) :

Le programme Météosat fait partie du système mondial d'observation de l'atmosphère terrestre, mis en place par l'organisation météorologique mondiale au milieu des années 70. Il comprend cinq satellites équ-répartis sur l'orbite géostationnaire, tous dans le plan de l'équateur. Les satellites Météosat, de première génération, ont été réalisés dans le Centre spatial de Cannes Mandelieu par un consortium (COSMOS) créé à cet effet, sous maîtrise d'œuvre d'Aérospatiale. Le premier satellite, Météosat 1, a été lancé le 23 novembre 1977, le deuxième, Meteosat-2 le 19 juin 1981, le troisième, Meteosat-3 le 15 juin 1988, le quatrième, Meteosat-4 le 6 mars 1989, le cinquième, Meteosat-5 le 3 mars 1991, le sixième, Meteosat-6 le 20 novembre 1993 et le septième, Meteosat-7 le 2 septembre 1997. Aujourd'hui, trois de ces satellites sont encore en fonction : Météosat 5, Météosat 6 et Météosat7.

Ces satellites ont un diamètre de 2,1 m, une hauteur de 3,2 m et une masse de 282 kg ; ils sont équipés d'un radiomètre à 3 canaux (visible, infrarouge thermique et infrarouge vapeur d'eau) et tournent sur leur axe avec une vitesse de 100 tours par minute. Les images produites par ces satellites sont collectées toutes les demi heures.

II.2.2.2 La deuxième génération (Météosat seconde génération : MSG) :

C'est la Division Satellites, qui remporte l'appel d'offres en 1990, pour cette seconde génération (MSG) de quatre satellites après les sept déjà réalisés au titre de la première génération Météosat. Comme pour les sept premiers, ils sont aussi réalisés dans le Centre spatial de Cannes

Mandelieu. Après six années d'études détaillées et de recherches et développements, le contrat pour la réalisation de trois modèles est signé le 16 octobre 1996 par Jean-Marie Luton, directeur général de l'Agence spatiale européenne et Yves Michot, président d'Aérospatiale, en présence du Dr Tillmann Mohr, directeur d'Eumetsat. Le premier satellite MSG 1 (Météosat 8) est lancé le 28 août 2002, le second MSG 2 (Météosat 9) a eu lieu dans la nuit du 21 au 22 décembre 2005, le troisième satellites MSG 3 en 2008 et le lancement du quatrième satellite MSG 4 est prévu pour 2012.

Les satellites de deuxième génération (MSG) affichent des dimensions bien supérieures à ceux de la précédente génération : un diamètre de 3,2 m, une hauteur de 2,4 m et une masse de 2 000 kg ; ils disposent d'un radiomètre à 12 canaux (3 dans le visible et 9 dans l'infrarouge) qui transmettent des données tous les quarts d'heure et ils tournent sur leur axe avec une vitesse de 100 tours par minute.

II.3 Paramètres influençant l'acquisition des images :

Pour la détection des objets terrestres ces satellites utilisent le rayonnement réfléchi ou émis par ces derniers. Quant à leur acquisition dans plusieurs intervalles de longueur d'ondes permet de les identifier par l'étude de leurs signatures spectrales.

L'acquisition d'information dépend de trois facteurs principaux.

II.3.1 Géométrie soleil satellite objet :

La position du rayonnement solaire incident par rapport au satellite influence la valeur de la luminance reçue d'un objet terrestre par le radiomètre visible.

II.3.2 Réflectance du corps visé :

Sur les images Météosat on remarque que la couleur des objets devient de plus en plus claire quand leur réflectance augmente.

II.3.3 Éclairement du soleil :

La valeur du niveau du gris des pixels de l'image dépend de l'heure et de la saison où elle est prise. Donc la couleur des images devient claire avec l'augmentation de l'éclairement du soleil. L'angle d'azimut du rayonnement solaire incident influence la luminance du rayonnement réfléchi.

II.4 Caractéristiques optiques des objets au sol [2] :

Pour reconnaître les différents éléments des images MSG, il faut analyser les propriétés optiques des objets qui constituent la surface de la terre.

II.4.1 Propriétés optiques des surfaces d'eaux :

L'eau absorbe une très grande partie du rayonnement qu'elle reçoit. Cette absorption augmente dans les grandes longueurs d'ondes, c'est la raison pour laquelle les étendus d'eaux apparaissent en noir. Dans le spectre visible on remarque une augmentation de la réflectance de l'eau dans le bleu. Cependant, la présence de la matière chlorophyllienne augmente la réflexion de l'eau dans le vert.

II.4.2 Propriétés optiques de la végétation :

La réflectance des végétaux varie en fonction de trois facteurs : la pigmentation, la structure cellulaire et le contenu en eau. La pigmentation augmente la réflectance dans la bande verte du domaine visible, la structure interne des feuilles provoque une augmentation de la réflectance dans le proche infrarouge et la présence de l'eau entraîne une baisse de la réflectance dans l'infrarouge moyen.

II.4.3 Propriétés optiques du sol :

Le sol est un mauvais réflecteur, il apparaît en gris sombre. Cependant sa réflectance varie en fonction de sa nature, sa couverture et de son humidité.

L'humidité ou la présence d'argile augmentent la réflectance du sol, tandis que la présence de matière organique et d'oxyde de fer provoquent sa diminution ce qui produit l'assombrissement du sol. La réflectance diminue aussi sous l'effet de la rugosité, car plus le sol est rugueux moins il réfléchit.

II.4.4 Propriétés optiques des nuages :

Les nuages sont de très bons réflecteurs et prennent toutes les teintes intermédiaires entre le gris clair et le blanc. La réflectance des nuages dépend de plusieurs facteurs qui sont les suivants : leurs constituants, la régularité de leurs sommets, leur épaisseur, leur densité et leur altitude.

II.5 Fonctionnement des satellites Météosat seconde génération (MSG) [2] [3] :

Le satellite tourne à 100 tours par minute autour d'un axe parallèle à l'axe nord-sud de la terre. Le télescope du radiomètre de Météosat vise la terre par l'intermédiaire d'un miroir et balaie à chaque révolution du satellite une étroite bande de la surface de la terre. L'angle de balayage correspondant, de 18°, est décrit en 30 ms. Pendant les 570 ms suivantes, le télescope vise l'espace et cette durée est mise à profit pour modifier l'orientation du miroir, de façon qu'au tour suivant, il balaie au sol une bande contiguë à la précédente, mais plus au nord. La phase de non acquisition de données est mise à profit pour calibrer les détecteurs. Au bout d'un quart d'heure le radiomètre est rapidement basculé vers le bas, prêt pour un nouveau scanne. Ce dernier est l'instrument principal dont les satellites météorologiques sont équipés. Il balaie la surface de la terre ligne par ligne. Chaque ligne consiste en une série d'images élémentaires ou pixel. Pour chaque pixel, le radiomètre mesure l'énergie radiative dans différentes bandes spectrales. Les images sont transmises, quasiment en temps réel, vers le centre de contrôle des satellites (ESOC) situé à Darmstadt, en Allemagne. Le satellite fournit des images tous les quarts d'heure, il effectue donc 96 scanne par jour.

Grâce aux 12 canaux spectraux de SEVIRI, le radiomètre imageur dans le visible et l'infrarouge embarqué sur les satellites, les apports du système MSG sont décisifs pour diverses applications dans le domaine de la prévision immédiate ou de la prévision numérique du temps, tout autant que pour la surveillance du climat et de l'environnement et la recherche.

SEVIRI transmet des images des structures météorologiques à une résolution de 3 km, et d'un km pour ce qui est de son canal à haute résolution dans le visible (HRV), ce qui signifie qu'il pourra distinguer des détails d'à peine un kilomètre. Ses 12 canaux permettent de procéder à ce qu'on appelle un "pseudo sondage" et d'obtenir des informations sur les profils des températures, la pression et l'humidité de l'atmosphère. La fréquence de son cycle de vue rapide, avec une nouvelle image toutes les 15 minutes, permet de suivre parfaitement les phénomènes météorologiques à évolution rapide, ce qui s'avère d'une grande utilité pour les prévisionnistes qui peuvent ainsi

reconnaître et prévoir les phénomènes météorologiques dangereux, orages, pluies violentes, brouillards ou le développement de dépressions aussi petites qu'intenses susceptibles d'aboutir à des tempêtes dévastatrices.

Outre SEVIRI, les satellites MSG ont à bord l'instrument GERB de mesure du bilan radiatif de la Terre qui fournit des informations précieuses sur le rayonnement de la planète, permettant aux chercheurs de calculer avec une grande précision le bilan radiatif terrestre, c'est-à-dire le rapport entre le rayonnement venant du Soleil et celui renvoyé dans l'espace. En complément de ces deux instruments, les satellites MSG sont dotés d'une charge utile de télécommunications très perfectionnée, nécessaire à l'exploitation et à la diffusion des données aux utilisateurs.



Figure II-1 : Le satellite MSG 2 (Météosat 9) avant sa mise en place à bord du lanceur Ariane 5

Le SEVIRI du satellite MSG comporte 12 canaux 3 dans le visible et 9 dans l'infrarouge :

II.5.1 Canaux visible :

Les images visibles représentent la quantité de lumière visible réfléchiée par les nuages ou la surface de la terre. Les nuages, la neige et la glace apparaissent en gris clair, cependant les nuages épais sont plus brillants que les nuages fins, les sols secs et nus ainsi que le sable apparaissent également en couleur claire tandis que les sols couverts de végétation apparaissent en couleur plus sombre. Les surfaces d'eau ont une réflectance très faible et apparaissent en conséquence en couleur gris sombre. Avec les canaux visibles il est difficile de distinguer les nuages bas des nuages élevés et leurs images sont complètement noires pendant la nuit. Pour cela, il faut utiliser les images infrarouges.

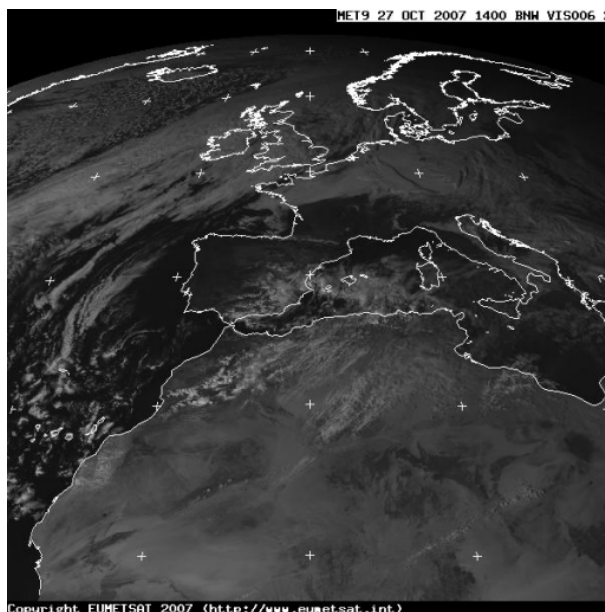


Figure II-2 : Image MSG visible prise le 27 octobre 2007 à 14 h

II.5.2 Canaux infrarouges :

Les images infrarouges représentent une mesure du rayonnement infrarouge émis par le sol ou les nuages. Ce dernier dépend de la température de l'objet observé. Plus il est chaud, plus il est noir et plus il est froid il blanchit. Les nuages élevés apparaissent plus blancs que les nuages bas car ils sont plus froids. Dans les zones sans nuages, l'assombrissement s'accroît avec la chaleur du sol et vice-versa.

Après avoir absorbé une fraction de l'énergie incidente, la surface terrestre rayonne à son tour restituant ainsi une partie de l'énergie emmagasinée. Ce rayonnement se fait dans une gamme de longueurs d'ondes comprise entre $3\mu\text{m}$ et $50\mu\text{m}$ avec un maximum pour $10\mu\text{m}$. Ces longueurs d'ondes appartiennent au domaine des infrarouges IR.

Une majeure partie de ces radiations émises par la terre est absorbées par la vapeur d'eau et certains gaz (CO_2 , N_2O , O_3 , O_2) présents dans l'atmosphère, piégeant ainsi cette énergie (effet de serre).

Cependant cette absorption atmosphérique est inégale selon les radiations émises par la terre. Les infrarouges de longueurs d'ondes comprises entre $8,5\mu\text{m}$ et $13\mu\text{m}$ traversent bien l'atmosphère. Ce sont elles, entre autres, que le satellite Météosat capte au moyen de son détecteur IR sensible aux énergies radiatives dans cette plage de longueurs d'ondes.

L'émission est fonction de la température et de la nature du corps émetteur. La couverture nuageuse, selon sa nature (glace, vapeur) et son épaisseur laissera passer plus ou moins ces rayons IR.

Sur les images infrarouges, il peut être difficile de distinguer entre les nuages bas et les zones sans nuages, parce que la différence de température entre ces nuages et le sol mouillé peut être assez insignifiante.

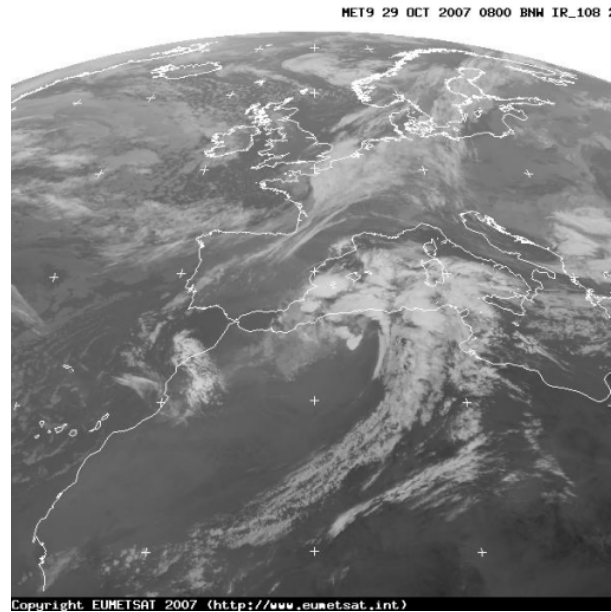


Figure II-3 : Image MSG Infrarouge prise le 29 octobre 2007 à 09 h

II.5.3 Canaux vapeurs d'eaux :

Ces images représentent une mesure du rayonnement infrarouge influencée par la vapeur d'eau dans l'atmosphère. Cela permet de déterminer les zones sèches et les zones humides.

Les infrarouges de longueurs d'ondes comprises entre $5 \mu\text{m}$ et $7,5 \mu\text{m}$ sont particulièrement absorbés quand l'atmosphère est riche en vapeur d'eau. Ce sont eux que le satellite Météosat capte par l'intermédiaire de son détecteur WV (Water Vapeur) sensible aux énergies radiatives comprises entre $5,5 \mu\text{m}$ et $7,5 \mu\text{m}$.

Lorsque l'atmosphère est pauvre en vapeur d'eau, ces rayons infrarouges la traversent et parviennent aux capteurs WV de MSG. Ce qui se raréfie, lorsque celle-ci est chargée en humidité. Cela explique pourquoi les régions sèches apparaissent en gris et les zones de fortes humidités en clair.

II.5.4 Canal visible à haute résolution (HRV) :

Il est clair que le canal HRV, représente une technologie très poussée, avec un intervalle d'échantillonnage amélioré de 1 Km. En le comparant aux autres canaux on constate une nette distinction entre la neige et les nuages. Cela pourra nous être très utile pour la détection automatique et la distinction entre les phénomènes dans une image MSG. Ce canal peut détecter les phénomènes météorologiques de petite taille comme les orages, les vents en altitudes et les différentes textures de nuages.

Les différentes caractéristiques des canaux du satellite MSG sont résumées dans le tableau suivant :

canal		Bande spectrale	utilisation
Visible	canal 0.8	0.56 μm à 0.71 μm	Nuage, végétation, vent à l'altitude des nuages, aérosols, albédo de surface.
	canal 0.6	0.74 μm à 0.88 μm	
Proche infrarouge : canal 1.6		1.5 μm à 1.78 μm	Discrimination neige nuage, aérosols, stress de végétation.
Infrarouge moyen : canal 3.9		3.48 μm à 4.36 μm	Nuage bas, brouillards, température de nuit de la surface de la mer et des sols.
Vapeur d'eau	Canal 6.2	5.35 μm à 7.15 μm	Contenue en vapeur d'eau et vent dans la moyenne et basse troposphère, calcul de l'indice de stabilité, altitude des cirrus.
	Canal 7.3	6.85 μm à 7.85 μm	
Infrarouge	canal 8.7	8.3 μm à 9.1 μm	Cirrus fin, phase de nuage
	Canal 9.7	9.38 μm à 9.94 μm	Ozone, vent au voisinage de la troposphère, calcul de l'indice de stabilité.
	Canal 10.8	9.8 μm à 11.8 μm	Température des sommets des nuages, indice de précipitation, détection des cirrus, inertie thermique des sols et hydrologie, température des surfaces des mers et des sols.
	Canal 12.0	11 μm à 13 μm	Discrimination des cirrus et de nuages semi transparents, calcul de l'indice de stabilité, altitude des sommets des nuages.
	Canal 13.4	12.4 μm à 14.4 μm	
Canal visible HRV (haute résolution)		0.56 μm à 0.74 μm	Vent à l'altitude des nuages, texture des nuages,

Tableau II-1 : Bande spectrale des 12 canaux du satellite MSG

II.6 Application des images MSG :

Les satellites MSG fournissent les images qui servent à l'élaboration des bulletins météorologiques diffusés sur les chaînes de télévision. Les animations montrant la formation d'anticyclones ou de dépressions sont ainsi devenues très familières du grand public. Mais au-delà des prévisions météorologiques de convenance personnelle, les données météorologiques satellitaires sont surtout devenues des paramètres essentiels dans les transports, l'agriculture, la pêche, la prévision et le suivi de phénomènes météorologiques violents, ainsi que l'évolution climatique de la planète à long terme.

Les données fournies par les satellites MSG permettent également de prévoir l'évolution des sécheresses, de la désertification, ainsi que de surveiller des volcans en sommeil et la couverture végétale. Par ailleurs, l'apparition et la propagation d'épidémies étant généralement corrélées avec

des conditions climatiques spécifiques, la précision des données des satellites MSG permettent de repérer et d'anticiper les zones à risque.

CHAPITRE III

Application du standard JPEG 2000 pour la compression des images MSG

III.1 Introduction

Dans le domaine de la compression d'images, la norme la plus connue est certainement JPEG, créée par le Joint Photographic Experts Group, depuis plus de quinze ans. Cependant, le développement actuel des réseaux informatiques et l'augmentation vertigineuse de la vitesse des processeurs font apparaître de nombreuses nouvelles potentialités pour l'imagerie numérique. Que ce soit dans le domaine médical, commercial ou militaire, de nouvelles applications voient le jour, chacune avec ses spécificités. Et la norme JPEG, ne permet plus de répondre à la variété de ces nouvelles contraintes. Pour cela, le groupe JPEG a commencé en 1997 à élaborer une nouvelle norme de codage d'image plus flexible et plus performante : JPEG2000, a été finalisé en décembre 2000.

La norme JPEG2000 propose un certain nombre de fonctionnalités telle que :

- Compression avec et sans pertes.
- Excellente qualité d'image.
- Ordonnancement du bitstream.
- Région d'intérêt.
- Accès et décompression aléatoire du bitstream.
- Protection contre les erreurs.
- Traitement d'une large variété d'images.

Cette norme comprend 12 parties sous la même dénomination JPEG 2000, c'est cependant à la première partie que nous allons nous intéresser dans ce chapitre, afin de l'appliquer pour la compression d'images MSG.

III.2 Les étapes de l'algorithme de compression :

Les étapes de l'algorithme de compression sont résumées dans le schéma suivant :

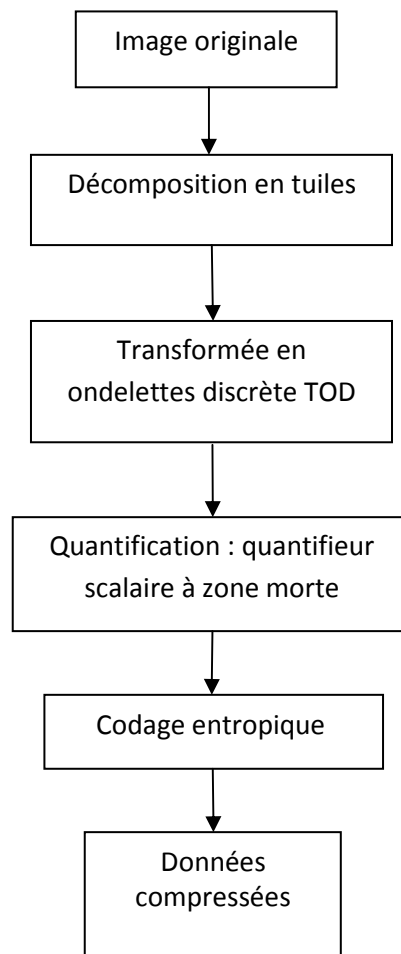


Figure III-1 : Les étapes de l'algorithme de codage

III.2.1 Découpage en tuiles [16] :

Avant l'application de la TOD, chaque image est découpée en un pavage de tuiles (traduction de tile). Le but de cette opération est de réduire la complexité de l'algorithme pour des images de très grande taille, mais aussi de faciliter la navigation à l'intérieur de celles-ci. Ces tuiles sont de tailles carré (64×64) ou (128×128) par exemple.

L'étape qui consiste à effectuer la transformation en couleur est supprimée, parce que les images utilisées sont en niveau de gris et ne nécessite aucune transformation en couleur.

III.2.2 La transformée en ondelettes discrète (TOD) [7] [8] [9] :

La transformation en ondelettes discrète provient de l'analyse multi-résolution qui a été développée par Stéphane Mallat et Yves Meyer en 1989. Comme son nom l'indique, le but de cette théorie est de décomposer un signal suivant différentes résolutions. On procède ainsi, à une décorrélation de l'information qu'il contient.

L'analyse classique de Fourier ne permet pas de réaliser l'étude des composantes transitoires de durées différentes d'un signal. Pour cela, la transformée en ondelettes décompose les signaux sur une famille d'ondelettes translatées et dilatées.

Une ondelette est une fonction mathématique ψ de $L^2(\mathbb{R})$ de moyenne nulle :

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad (\text{III-1})$$

Elle est normalisée et centrée au voisinage de $t=0$:

$$\int_{-\infty}^{+\infty} |\psi(t)|^2 dt = 1. \quad (\text{III-2})$$

Une famille d'ondelettes $\psi_{a,b}(t)$ s'obtient en dilatant d'un facteur a et en translatant d'un facteur b l'ondelette mère ψ :

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right). \quad (\text{III-3})$$

avec : $(a, b) \in \mathbb{R}_+^* \times \mathbb{R}$.

L'idée principale de la transformée en ondelette est de représenter une fonction f quelconque comme une superposition d'ondelettes. De plus, cette transformée contracte et dilate l'axe des temps, ce qui permet d'introduire la notion d'échelle ou de résolution.

La transformée en ondelette continue d'une fonction réelle f est donnée par l'équation suivante :

$$TOC_f(a, b) = \langle f, \psi_{a,b} \rangle = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} \psi^*\left(\frac{t-b}{a}\right) f(t) dt \quad (\text{III-4})$$

Avec : ψ^* : représente le conjugué de ψ .

Si l'ondelette vérifie une condition, dite d'admissibilité :

$$C_\psi = \int_0^{+\infty} \frac{|\widehat{\psi}(w)|^2}{w} dw < +\infty. \quad (\text{III-5})$$

Alors la transformée en ondelettes est complète et conserve l'énergie. Donc elle admet une transformation inverse donnée par l'équation suivante :

$$f(t) = \int_0^{+\infty} \int_{-\infty}^{+\infty} \langle f, \psi_{a,b} \rangle \psi_{a,b}(t) da \frac{db}{a^2} \quad (\text{III-6})$$

Pour appliquer les ondelettes à la compression d'images, il faut les discrétisées, cela implique que les opérateurs de dilatation et de translation a et b prennent des valeurs discrètes. Elle se fait suivant les expressions suivantes : $a=a_0^j$ et $b=b_0 k a_0^j$ avec : $(j, k) \in \mathbb{Z}^2$

Avec : $a_0 > 1$ et $b_0 \neq 0$

La nouvelle famille de la TOD est représentée par la formule suivante :

$$\psi_{j,k}(t) = a_0^{-\frac{j}{2}} \psi(a_0^{-j} t - kb_0) \quad (\text{III-7})$$

Et la TOD d'une fonction f est représentée par la formule suivante :

$$TOD_f(j, k) = \langle f, \psi_{j,k} \rangle = a_0^{-\frac{j}{2}} \int_{-\infty}^{+\infty} f(t) \psi(a_0^{-j} t - kb_0) dt \quad (\text{III-8})$$

Mayer à découvert en 1985 que le choix des opérateurs de dilatation et de translation dyadique :

$a = 2^j$ et $b = k \times 2^j$, réduit la redondance présente dans l'information. La nouvelle base orthogonale de $L^2(\mathbb{R})$ est définie comme suit :

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}} \psi(2^{-j} t - k) \quad (\text{III-9})$$

Les coefficients d'ondelette sont alors calculés par l'équation suivante :

$$C_{j,k} = \langle f, \psi_{j,k} \rangle = 2^{-j} \int_{-\infty}^{+\infty} f(t) \psi(2^{-j} t - k) dt \quad (\text{III-10})$$

La fonction f est reconstruite par la formule suivante :

$$f(t) = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} C_{j,k} \psi_{j,k} \quad (\text{III-11})$$

Pour introduire la notion de résolution, on emploie une fonction φ d'échelle définie, comme pour la fonction ψ par l'équation suivante :

$$\varphi_{j,k}(t) = 2^{-\frac{j}{2}} \varphi(2^{-j} t - k) \quad (\text{III-12})$$

Avec : $(j, k) \in \mathbb{Z}^2$

Ces fonctions forment une famille orthogonale pour j fixe. Elles possèdent la propriété suivante :

$$\varphi(t) = \sum_n \alpha_n \varphi(t - 2n) \quad (\text{III-13})$$

Ce qui entraîne que l'ondelette associée suit la relation suivante :

$$\psi(t) = \sum_n (-1)^n \alpha_{n+1} \varphi(2t + n) \quad (\text{III-14})$$

On a donc, en projetant sur cette famille de fonctions $\varphi_{j,k}$, une approximation du signal à la résolution 2^j . La perte d'information, lors du passage d'une résolution 2^{j-1} à une résolution 2^j plus grossière est alors décrite par les coefficients $C_{j,k} = \langle f, \psi_{j,k} \rangle$ définis précédemment.

Les signaux d'images que nous utilisons en pratique sont des signaux échantillonnés. Les échantillons de ces signaux d'origine sont considérés comme les coefficients d'approximations $s_{0,n}$ de plus grande résolution ; cette résolution est déterminée par la fréquence d'échantillonnage des signaux.

Soient h_n et g_n des Filtres tels que :

$$h_n = 2^{\frac{1}{2}} \int \varphi(x-n) \varphi(2x) dx \text{ et } g_l = (-1)^l h_{-l+1} \quad (\text{III-15})$$

L'algorithme de décomposition du signal est alors décrit par les équations suivantes :

$$c_{j,n} = \sum_k g_{2n-k} s_{j-1} \text{ et } c_{j,n} = \sum_k h_{2n-k} s_{j-1} \quad (\text{III-16})$$

Daubechies a montré que les coefficients h_n correspondent aux coefficients α_n de la formule (III-13). La condition nécessaire et suffisante pour avoir une base orthogonale d'ondelettes est alors

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2 \text{ et } G(\omega) = \bar{H}(\omega + \pi) e^{-i\omega} \quad (\text{III-17})$$

H et G sont les transformées de Fourier respectives des filtres h et g . Ces deux filtres orthogonaux H et G, de somme constante, décomposent le signal en deux sous-bandes.

Ces filtres, utilisés aussi pour la synthèse du signal, donnent alors une reconstruction exacte :

$$s_{j-1,k} = \sum_k (h_{2n-k} s_{j,n} + g_{2n-k} c_{j,n}) \quad (\text{III-18})$$

Il existe de nombreuses extensions de la transformée en ondelette de une dimension a plusieurs dimensions. Dans notre application, nous avons utilisé une transformée en ondelette qui privilégie les directions horizontales et verticales.

Dans un schéma d'analyse de deux dimensions par ondelette, on introduit une fonction d'échelle $\Phi(x, y)$ séparable telle que:

$$\Phi(x, y) = \varphi(x) \varphi(y) \quad (\text{III-19})$$

Ou φ est une fonction d'échelle à une dimension.

Soit $\psi(x)$ l'ondelette monodimensionnelle associée a la fonction d'échelle φ . Alors, les trois ondelettes à deux dimensions sont définies de la manière suivante :

$$\Psi^H(x, y) = \varphi(x) \psi(y) \quad (\text{III-20})$$

$$\Psi^V(x, y) = \psi(x) \varphi(y) \quad (\text{III-21})$$

$$\Psi^D(x, y) = \psi(x) \psi(y) \quad (\text{III-22})$$

En pratique, la TOD consiste à appliquer sur l'image deux filtres a réponse impulsionnels finie, un filtre passe bas (h) et un filtre passe haut (g) suivie d'une décimation par deux.

Le filtre passe bas consiste à atténuer les composantes de l'image ayant une fréquence haute correspondante à une version grossière de l'image originale, dans laquelle les valeurs des pixels sont très proches, ou aucune variation brusque n'est observée. Ce type de filtrage est généralement

utilisé pour atténuer le bruit de l'image, c'est la raison pour laquelle on parle habituellement de lissage.

Le filtre passe-haut, atténuent les composantes de basse fréquence de l'image et permet notamment d'accentuer les détails et le contraste, c'est la raison pour laquelle le terme de filtre d'accentuation est parfois utilisé.

Les étapes du filtrage de l'image sont les suivantes :

- Dans un premier temps, chaque ligne de l'image est décomposée en utilisant horizontalement les filtres h et g suivie par une décimation. Cela produit deux sous-bandes, l'une correspondant aux basses fréquences horizontales L disposée à gauche, l'autre aux hautes fréquences horizontales H, disposée à droite.

- Dans un second temps, les mêmes filtres sont appliqués aux colonnes de ces deux bandes. Chacune des deux bandes de départ est alors décomposée en deux nouvelles sous-bandes, l'une correspondant aux basses fréquences verticales (LL, LH) disposée dans la partie supérieure de la matrice, l'autre aux hautes fréquences verticale (HL, HH) disposée dans la partie inférieure.

Nous obtenons à la fin quatre sous bandes :

- une sous-bande issue des deux filtres passe-bas représentant la réduction de l'image source (approximation LL).
- une sous-bande issue d'un double filtrage passe-haut (détails diagonaux HH).
- une sous-bande issue d'un filtrage passe-haut horizontal et un filtrage passe-bas vertical (détails horizontaux HL)
- une sous-bande issue d'un filtrage passe-bas horizontal et un filtrage passe-haut vertical (détails verticaux LH).

La (figure III.4) illustre l'effet des différents filtres sur l'image originale.

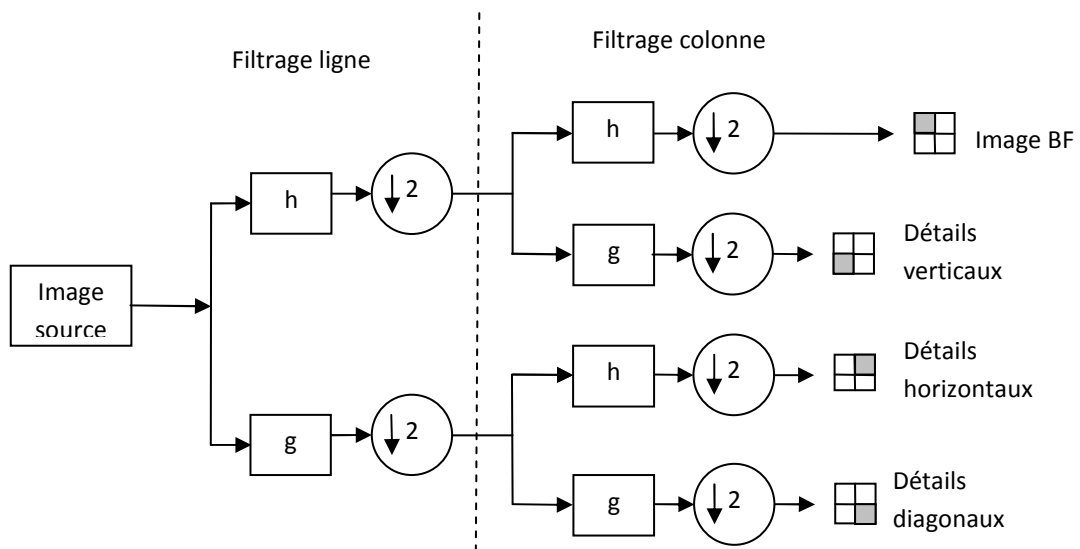


Figure III-2 : Décomposition multi-résolutions d'une image

Ce processus peut être répété à plusieurs reprises, en décomposant à chaque fois l'image basse fréquence, on accède ainsi à des représentations de plus en plus grossières de l'image originale.

III.2.3 Quantification scalaire [4] :

Les coefficients de la transformée en ondelettes sont quantifiés par un quantificateur scalaire uniforme à zone morte.

L'intérêt de la zone morte provient des très nombreux coefficients ondelettes non-nuls, mais proches de zéro. Ces coefficients n'apportent que très peu d'information pertinente, et leur codage entropique impliquerait un important surcoût au vu de la qualité gagnée. La zone morte permet donc de se débarrasser de ces coefficients en les quantifiant à zéro.

Chaque coefficient de la TOD : $c_b(u, v)$ d'une sous-bande est quantifié à la valeur $q_b(u, v)$ suivant l'équation suivante :

$$q_b(u, v) = \text{sign}(c_b(u, v)) \left(\frac{|c_b(u, v)|}{\Delta_b} \right) \quad (\text{III-23})$$

Avec :

$\text{sign}(c_b(u, v))$: Désigne le signe de $c_b(u, v)$.

Δ_b : Le pas de quantification. $C_b(u, v)$

Le pas de quantification dépend de la sous bande et du niveau de décomposition. La dimension du pas de quantification Δ_b est représentée par rapport à la dimension dynamique de la sous-bande b.

$$\Delta_b = \left(1 + \frac{\mu_b}{2^{11}} \right) \cdot 2^{R_b - \varepsilon_b} \quad (\text{III-24})$$

R_b : la dynamique du signal d'origine (nombre de bits).

ε_b : la dynamique voulue des coefficients.

μ_b : un facteur multiplicatif permettant d'avoir des valeurs de Δ_b différentes des multiples 2^N , avec N entier positif.

Dans le cas d'un codage sans perte, le pas de quantification est égal à 1, ce qui signifie qu'aucune quantification n'est appliquée.

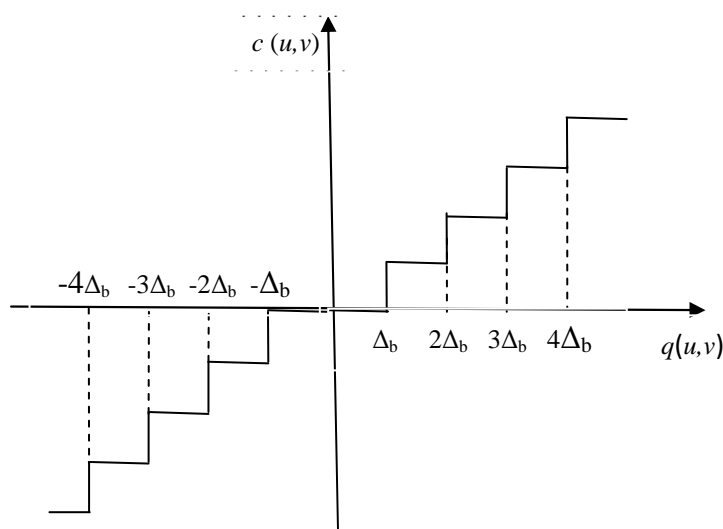


Figure III-3 : Quantification scalaire uniforme avec zone morte

III.2.4 Codage entropique [16] :

Une fois que les coefficients issus de la transformée en ondelettes ont été quantifiés, chaque sous-bande est découpée en entiers rectangulaires appelés code-blocks. Chacun de ces code-blocks est alors compressé indépendamment des autres au moyen d'un codeur entropique adaptatif avec contexte.

Ce codeur parvient à compresser une séquence de 0 et de 1 en réduisant la redondance présente dans cette séquence. Le fait qu'il soit entropique signifie qu'il réalise cette réduction de redondance en utilisant les probabilités d'occurrence de chaque symbole. Le fait qu'il soit adaptatif signifie que ces probabilités d'occurrence sont adaptées dynamiquement au cours du processus de codage. Enfin, le fait qu'il soit avec contexte signifie que la probabilité d'un symbole dépend de la valeur de ces voisins (son contexte).

Le codeur utilisé est constitué de deux parties bien distinctes mais interagissant l'une avec l'autre :

1) Le codeur EBCOT (Embedded Bloc Coding Optimized Truncation):

L'algorithme EBCOT a été créé en majorité par David Taubman, le principe de base de cet algorithme est le suivant ; Après la quantification scalaire, les coefficients issus des différentes sous-bandes sont rangés en blocs, appelés code-blocs (CB), de forme rectangulaire, de taille paramétrable, leurs hauteurs et largeurs correspondant à une puissance de deux et le produit largeur hauteur ne devant pas dépasser 4096 avec une hauteur minimale de 4. Chaque code-bloc est codé indépendamment, sans aucune référence aux autres blocs de la même sous-bande ou une autre.

A travers une sous-bande, on regroupe les code-blocs en precincts. Par ce regroupement, on obtient l'ensemble des coefficients d'un niveau de résolution pour une position spatiale donnée. La taille de ces precincts peut varier d'une sous-bande à l'autre. Les code-blocs sont en fait des tableaux de coefficients représentés par un tableau tridimensionnel binaire position-amplitude, constitué de N_b bit-planes où N_b représente la dynamique de la sous-bande. Chaque bit-plane est balayé dans un ordre prédéfini. Les données sont rangées par bandes de quatre valeurs de haut, balayées de haut en bas puis de la gauche vers la droite jusqu'à la prochaine bande. Le bit représentant le coefficient dans le bit-plane courant ne pourra être codé que par une seule des trois passes.

En commençant par le plan de bits de poids fort (MSB), tous ces plans sont codés en fonction de leur signifiante et de leur contexte par trois passes successives : Passage de signification (significance pass), passage d'amélioration (refinement pass) et passage de nettoyage (cleanup pass). Les bits issus de ces différentes passes, ainsi que le contexte associé, sont ensuite envoyés à un codeur arithmétique MQ. A chaque coefficient d'un code-bloc est associé une variable d'état appelée signifiante (state significance). Cette variable devient vraie (coefficient signifiant) quand le premier bit devient 1 (MSB du coefficient). Pour chaque coefficient, on définit également un vecteur de contexte qui se réfère à l'état de ses huit voisins. Tous les coefficients voisins de l'extérieur d'un code-bloc sont considérés non signifiants. Le nombre de contexte possible est cependant réduit à 19 (0 à 18). Les règles d'assignement du contexte sont dépendantes du type de passe.

Le premier bit-plane signifiant rencontré est codé seulement avec le passage de nettoyage (cleanup pass). Les bit-planes suivants sont codés en trois passes. Chaque bit d'un plan est codé une seule fois par l'une des trois passes.

- **Significance pass (passage de signification) :** Cette passe inclut uniquement les bits de coefficients dits insignifiants et ayant des contextes différents de zéro, cet a dire ayant au moins un voisin signifiant.

- **Refinement pass (passage d'amélioration) :** Cette passe de codage est la plus simple. Seuls les bits de coefficients déjà signifiants dans le bit-plane précédent sont codés.

- **Cleanup pass (passage de nettoyage) :** La passe cleanup code tous les bits d'un plan n'ayant pas été codés par les deux autres passes, c'est-à-dire les bits de coefficient insignifiants ayant un label de contexte égal à zéro (pas de voisin signifiant).

2) codeur arithmétique binaire adaptatif (MQ) :

Le fait que le codeur MQ soit adaptatif signifie que les probabilités des 2 types de symbole à encoder évoluent au cours du processus de codage. Les différentes améliorations apportées par le MQ, par rapport au codage arithmétique classique sont :

- l'abandon des symboles "0" et "1" au profit des symboles *MPS* (Most Probable Symbol) et *LPS* (Least Probable Symbol). Dans un codeur MQ, on n'encode donc pas a proprement parler un "0" ou un "1" mais plutôt le symbole le plus ou le moins probable. Comme nous le verrons, les valeurs du *MPS* et du *LPS* peuvent changer au cours du processus de codage. Par convention, l'intervalle du *LPS* est placé dans la partie inférieure de l'intervalle de probabilité. Puisqu'il s'agit d'un alphabet binaire les équations utilisées par cet algorithme sont :

$$\text{Pour le LPS : } A_{i+1} = A_i \times Q \text{ et } C_{i+1} = C_i \quad (\text{III-25})$$

$$\text{Pour le MPS : } A_{i+1} = A_i \times (1 - Q) \text{ et } C_{i+1} = C_i + A_i \times Q \quad (\text{III-26})$$

A_i est la largeur de l'intervalle de probabilité suite a l'encodage du i^{eme} symbole.

C_i est la borne inférieure de l'intervalle suite à l'encodage du i^{eme} symbole.

Q est la probabilité du LPS.

- Une seconde modification par rapport à l'algorithme de base résout le problème de précision des valeurs de A_i et C_i en renormalisant périodiquement leur valeurs et de cette façon, on peut maintenir ces valeurs a l'intérieur de bornes définies au préalable pour la représentation a précision

fixe. La première étape de cette normalisation consiste à stocker les valeurs de A_i et C_i dans des registres fixe A et B de longueur fixe. Puis de les décaler vers la gauche, ce qui correspond à une multiplication par deux, de cette manière on évite les multiplications. On remarque qu'à chaque renormalisation, un bit supplémentaire du mot-code (le MSB de C) sort du codeur et allonge petit-à-petit la chaîne des symboles codes. En pratique toutefois, on attendra d'avoir extrait tout un bit de C avant de le concaténer avec la chaîne déjà formée. En choisissant de maintenir A_i à une valeur proche de l'unité ($A_i \approx 1$) et en remplaçant la valeur de Q par une valeur Q_e qui est une approximation de ($A_i \times Q$), les équations précédentes peuvent être réduites comme suit :

$$\text{Pour le LPS : } A_{i+1} = Q_e \text{ et } C_{i+1} = C_i \quad (\text{III-27})$$

$$\text{Pour le MPS : } A_{i+1} = A_i - Q_e \text{ et } C_{i+1} = C_i + Q_e \quad (\text{III-28})$$

III.2.5 Allocation de débit :

La séquence binaire issue du MQ est divisée en un certain nombre de paquets. Chacun d'eux contient le bitstream correspondant à une même composante, un même niveau de résolution, une même couche de qualité et une même zone spatiale du niveau de résolution. Les zones spatiales de chaque niveau de résolution sont appelées precincts. Chacun des paquets est précédé d'un en-tête contenant des informations permettant d'identifier très précisément les données véhiculées par ce paquet. L'ordre dans lequel les paquets apparaissent dans le codestream est appelé ordre de progression. Quatre ordres de progression différents sont définis. Ils permettent lors du décodage d'obtenir en priorité soit les données d'une même composante, soit celles d'un même niveau de résolution, soit celles d'une même couche de qualité, soit celles d'une même zone spatiale de l'image.

III.3 Les étapes de l'algorithme de décompression [4] :

Les étapes de l'algorithme de décompression du standard JPEG 2000 sont résumées dans le schéma suivant :

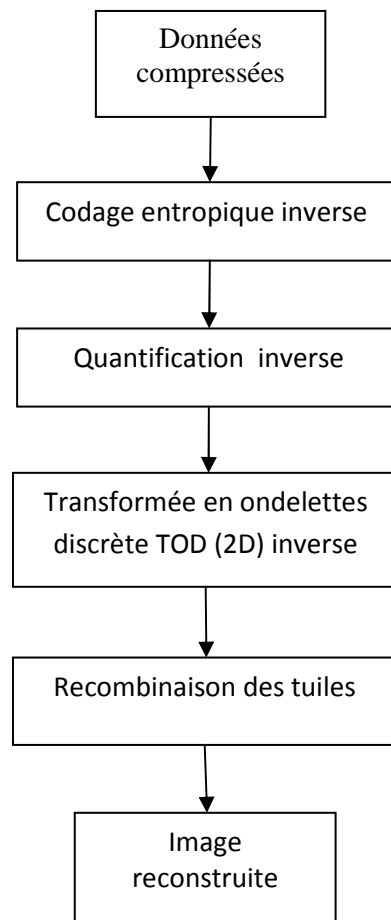


Figure III-4 : Les étapes de l'algorithme de décompression

III.3.1 Le décodage entropique :

Lors du décodage d'un bit, le décodeur reçoit le contexte de ce bit, le même que celui qui avait été utilisé pour le coder. Muni de ce contexte, il peut déterminer la probabilité Q_e associée à ce bit. Utilisant les valeurs des registres A et C (qui suivent exactement la même évolution qu'à l'encodage), il peut déterminer si la nouvelle valeur de C_i se situe dans l'intervalle du MPS ou du LPS. Sachant cela, et connaissant aussi la valeur du MPS associée au contexte qu'on lui a donné, il peut fournir à l'EBCOT la bonne valeur du bit décodé. Il met également à jour les valeurs A_i et C_i . Dans le cas où A_i est inférieur à 0.75, une ou plusieurs renormalisations sont opérées.

Le processus de décodage EBCOT est semblable à l'encodage. La différence principale réside dans le sens des interactions entre l'EBCOT et le MQ. À l'encodage, l'EBCOT envoie la paire de données au MQ afin que celui-ci génère le bit stream. Au décodage, l'EBCOT n'envoie que le contexte et reçoit du MQ la valeur du symbole à décodé

\

III.3.2 La quantification inverse :

Pour reconstruire les sous bande à partir des séquences générées par le décodeur entropique on utilise l'équation suivantes :

$$\tilde{c}_b(u, v) = \begin{cases} [q_b(u, v) + r \text{sign}(q_b(u, v))] \Delta_b & \text{si } q_b(u, v) \neq 0 \\ 0 & \text{autrement} \end{cases} \quad \text{(III-29)}$$

Où :

$q_b(u, v)$ est l'index de quantification.

Δ_b est le pas de quantification.

$\tilde{c}_b(u, v)$ est la valeur reconstruite du signal.

$\text{Sign}(q)$ dénote le signe de q , et r est la polarisation de reconstruction. Une valeur populaire pour r est 0.375.

III.3.3 Transformée en ondelette discrète inverse (TODI) :

Cette étape représente l'opération inverse de la transformée en ondelettes. La transformée inverse a pour but de reconstruire l'image originale à partir des sous bandes détails et de l'image de basse résolution. La reconstruction est donc obtenue par la somme du filtrage h (filtre passe bas) des coefficients d'approximations et du filtrage g (filtre passe haut) des coefficients de détails auxquels sont intercalés des zéros. La reconstruction de l'image originale à partir des différents sous bandes est illustrée par le schéma suivant :

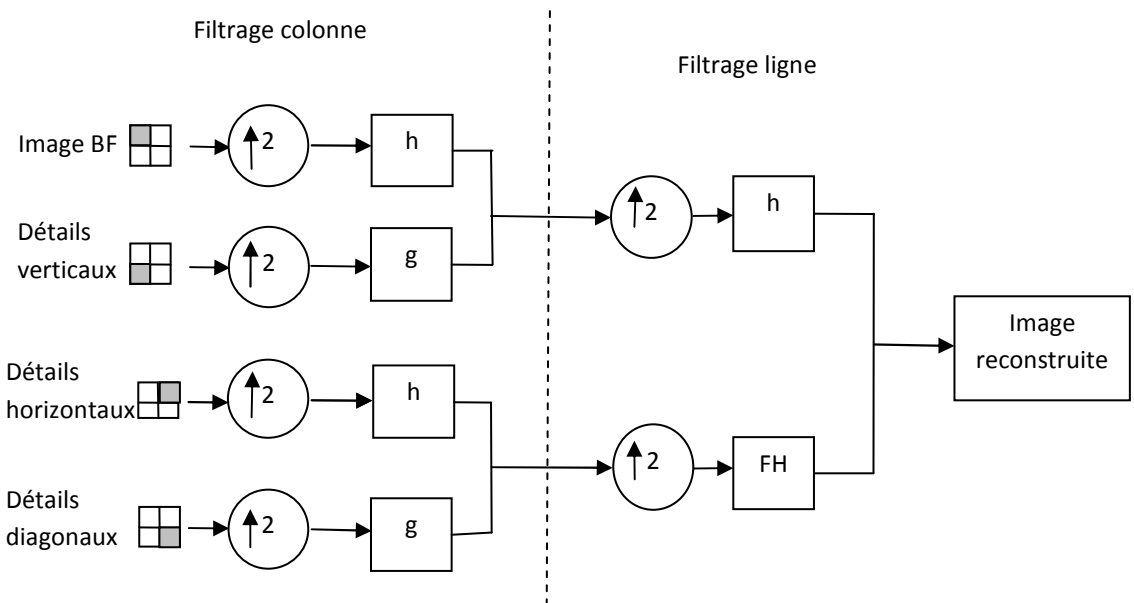


Figure III-5 : Reconstruction multi-résolutions d'une image

\

III.4 Conclusion :

Dans ce chapitre, nous avons défini les différentes étapes de l'algorithme JPEG 2000, qui seront appliquées pour la compression des images MSG. Les résultats du standard sur ces images seront exposés et interprétés dans le chapitre suivant.

CHAPITRE IV

Tests et résultats

IV.1 Introduction :

L'algorithme de compression développé dans le chapitre précédent a été mis en œuvre en utilisant le langage de programmation Matlab 7.5.0 sous l'environnement Windows. Le choix de ce langage réside dans sa grande souplesse.

Pour évaluer cette algorithme, nous allons effectué plusieurs tests sur des images météoat seconde génération (MSG) du domaine visible et infrarouge.

Dans le but d'interpréter les différents résultats obtenus par l'application de l'algorithme de compression, nous rappellerons les définitions des critères d'évaluation des performances de toutes méthodes de compression :

- Le rapport de compression qui est définie comme étant le quotient entre le nombre de bits de l'image originale et le nombre de bits de l'image compressée :

$$R_c = \frac{\text{nombre de bits de l'image originale}}{\text{nombre de bits de l'image compressée}} \quad (\text{IV-1})$$

- Le taux de compression en pourcentage est calculé par l'équation suivante :

$$T_c = \left(1 - \frac{1}{R_c}\right) \times 100 \quad (\text{IV-1})$$

- Le débit binaire représente le nombre de bits par pixel de l'image compressée :

$$R_c = \frac{\text{nombre de bits de l'image compressée}}{\text{nombre de pixel de l'image}} \quad (\text{IV-3})$$

- L'erreur quadratique moyenne qui mesure la distorsion entre l'image originale et l'image compressée :

$$MSE = \frac{1}{MN} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left(I(i, j) - \tilde{I}(i, j)\right)^2 \quad (\text{IV-4})$$

Ou M et N sont les dimensions de l'image.

$I(i, j)$: Position du pixel de l'image originale.

$\tilde{I}(i, j)$: Position du pixel de l'image compressée.

- Le rapport signal sur bruit crête en décibel (dB) qui mesure la qualité de l'image compressée :

$$PSNR = 10 \log \left(\frac{N_{Gmax}^2}{MSE} \right) \quad (\text{IV-5})$$

Ou N_{Gmax} : niveau maximal du codage du gris de l'image.

- La complexité calculatoire est le temps d'exécution de l'algorithme de compression.

IV.2 Présentation des données :

Afin d'évaluer les performances de l'algorithme décrit dans le chapitre précédent, deux types d'images sont utilisées :

IV.2.1 Image Lena :

L'image Lena qui est un portrait de référence très utilisé en compression d'images, elle est composée de diverses textures. Cette image est de taille 512 × 512, elle est codée sur 8bits (images à niveaux de gris).



Figure IV-1 : l'image Lena

IV.2.2 Les image Météosat seconde génération (MSG) :

Dans notre étude nous utiliserons deux types d'images MSG à niveaux de gris codés sur 8bits de taille 512 × 512 :

1) Deux images du canal visible, l'une est prise le 27 octobre 2007 à 14h et l'autre le 29 octobre 2007 à 9h.

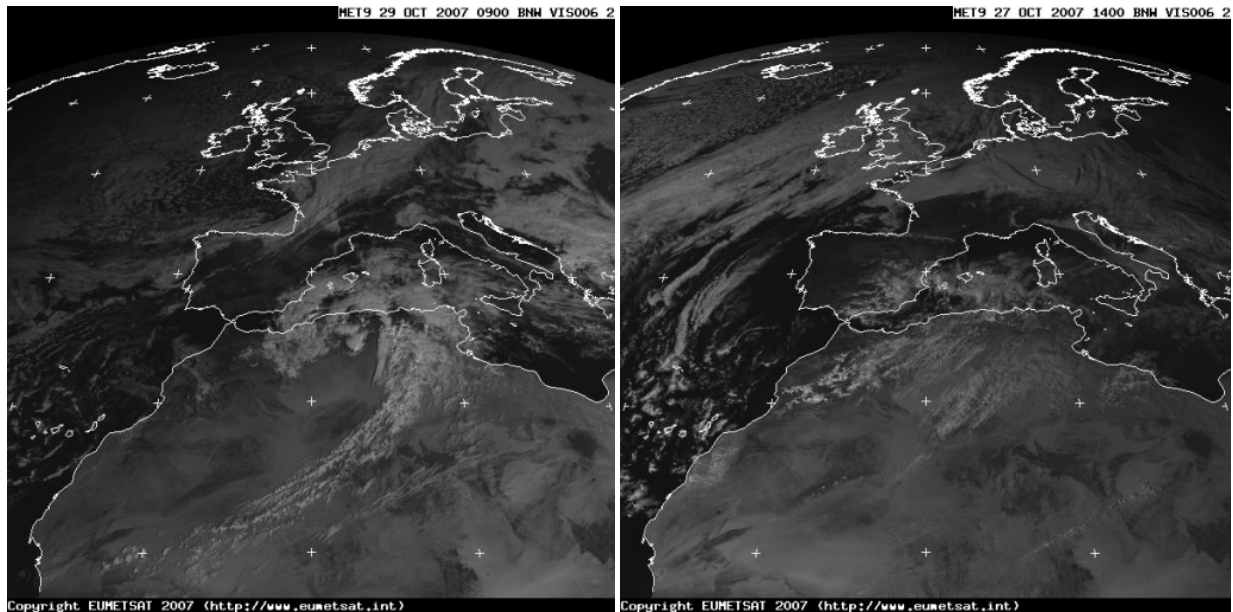


Image prise à 9h

Image prise à 14h

2) Deux images infrarouges, l'une est prise le 27 octobre 2007 à 14h et l'autre le 29 octobre 2007 à 8h.

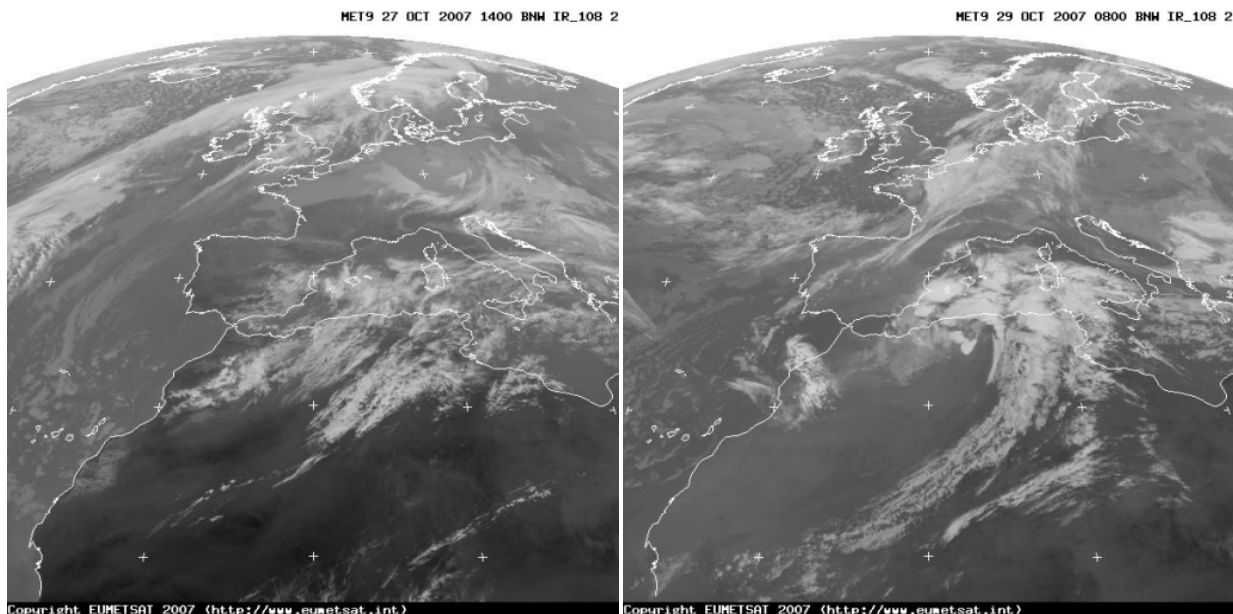


Image prise à 14h

Image prise à 8h

Figure IV-3 : images infrarouges du satellite Météosat MSG

IV.3 Testes et résultats :

Pour évaluer les performances de l'algorithme JPEG 2000 nous l'avons testé premièrement sur l'image Lena, les différents résultats de ce test sont résumés dans le tableau suivant :

Débit (bpp)	MSE	PSNR (dB)	temps de compression (seconds)
0.25	2.976524	43.393709	2.132184
0.5	0.277279	53.701635	1.520176
0.75	0.013969	66.679019	1.500594
1	0.013969	66.679019	1.450982

Tableau IV-1 : résultats obtenu avec l'image Lena

Les graphes suivants donnent la variation des différents paramètres de performances :

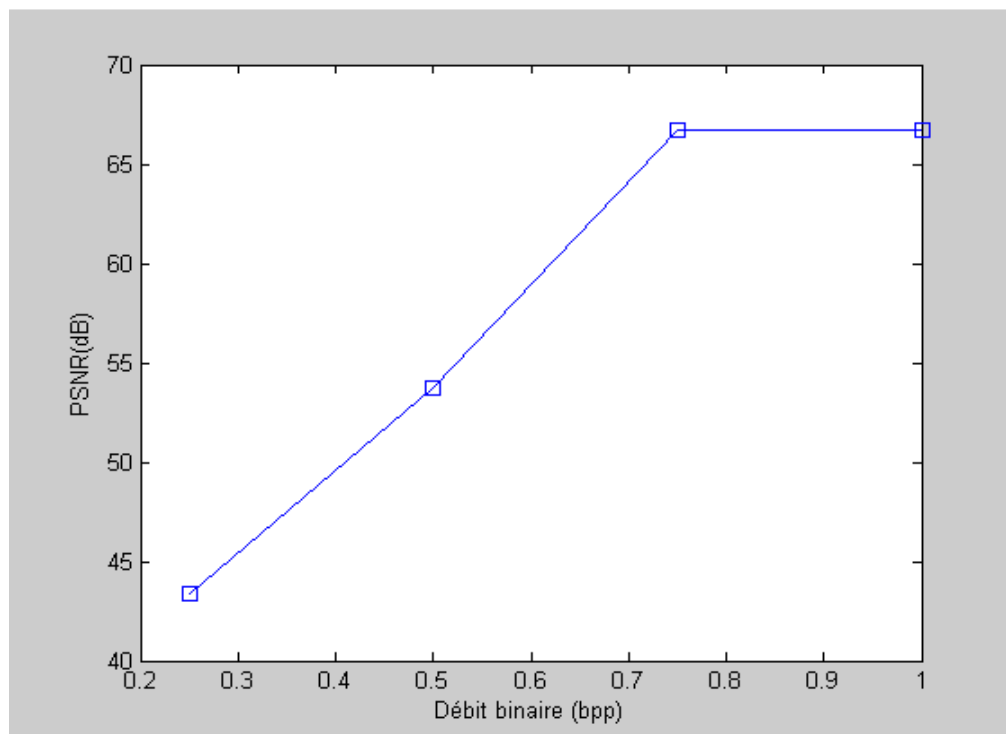


Figure IV-4 : variation du PSNR en fonction débit binaire

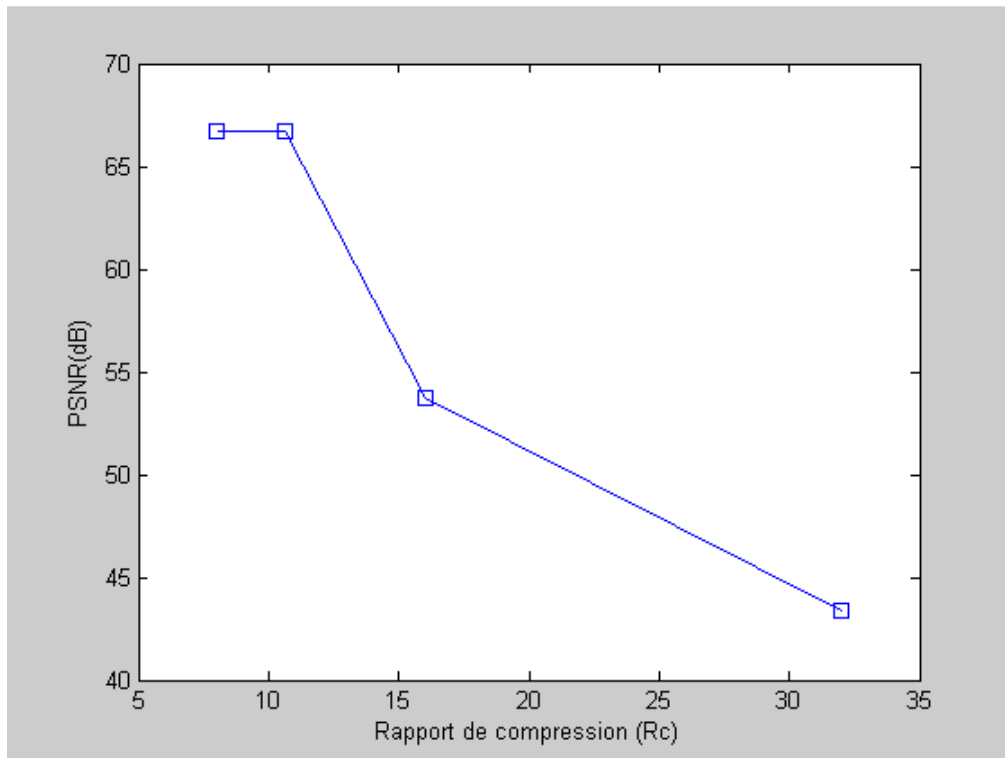


Figure IV-5 : variation du PSNR en fonction du rapport de compression

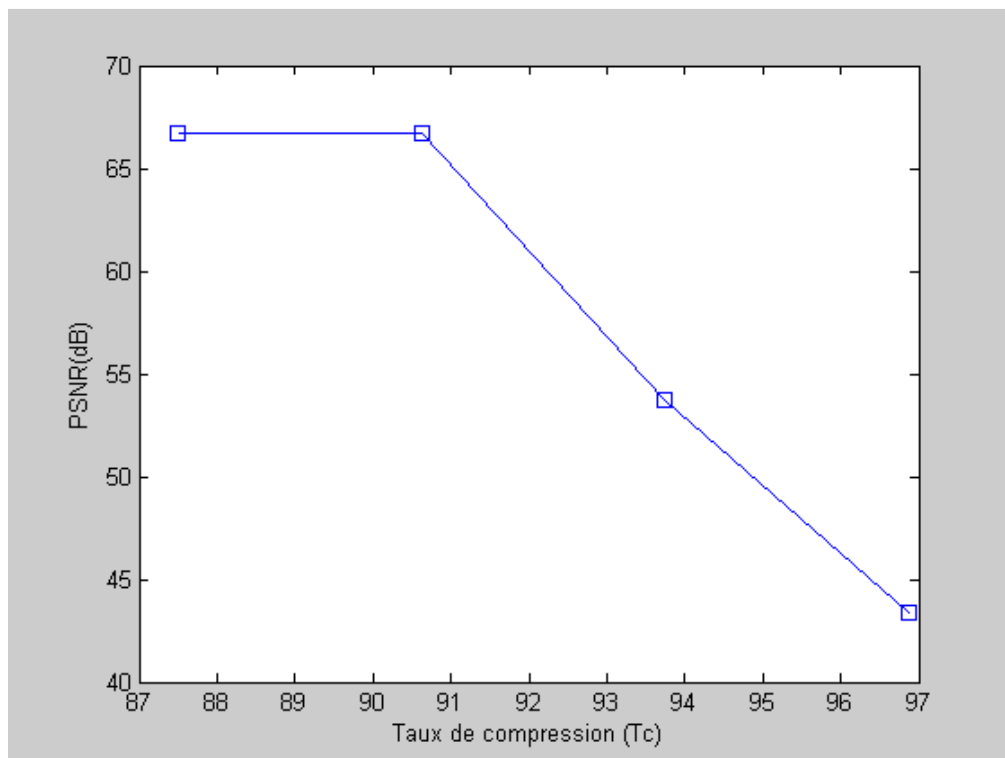


Figure IV-6 : variation du PSNR en fonction du taux de compression

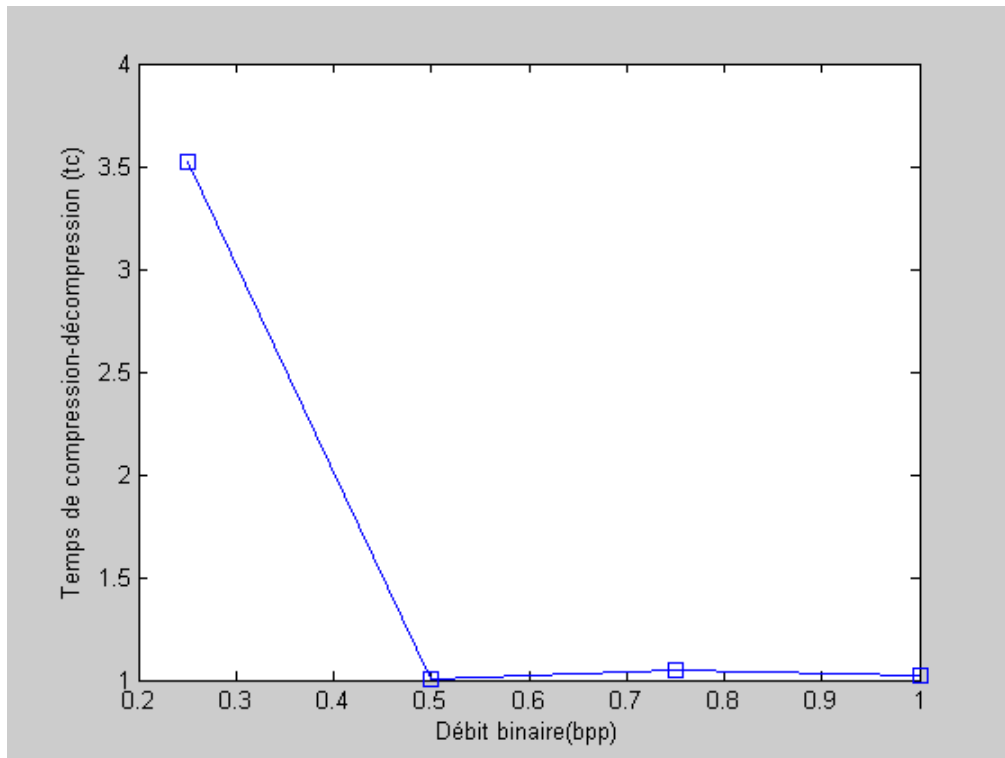


Figure IV-7 : variation du temps de compression et de décompression en fonction du débit binaire.

En deuxième lieu nous avons testé l'algorithme sur les images MSG. Les différents résultats de ce test sont résumés dans les tableaux et les graphes suivants :

1) Pour l'image MSG du canal visible prise à 9h :

Débit (bpp)	MSE	PSNR (dB)	temps de compression (seconds)
0.25	2.746040	43.743734	0.833380
0.5	0.063866	60.078129	0.647225
0.75	0.000122	87.264703	0.645423
1	0.000122	87.264703	0.643203

Tableau IV-2 : résultats obtenus avec l'image MSG du canal visible prise à 9h.

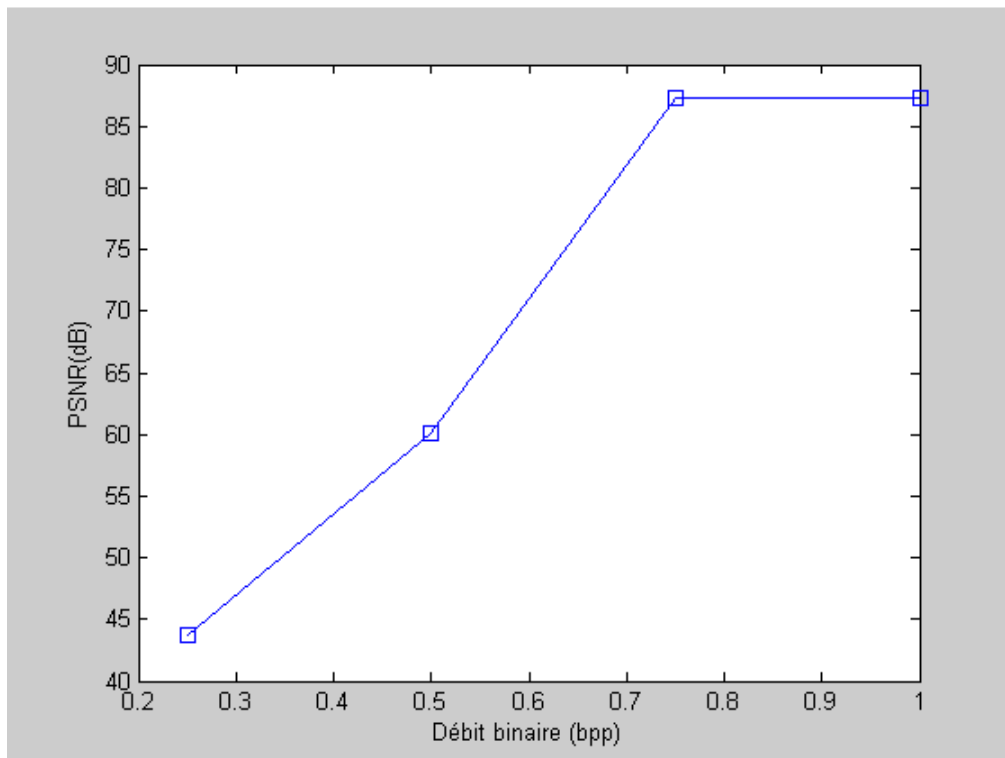


Figure IV-8 : variation du PSNR en fonction débit binaire

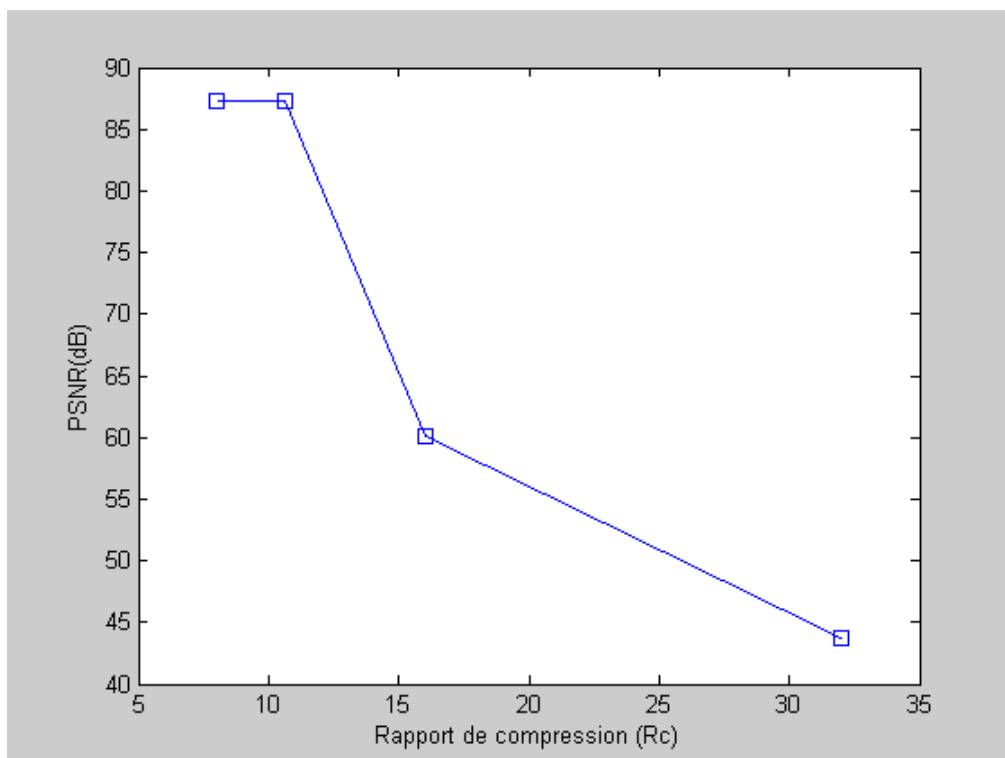


Figure IV-9 : variation du PSNR en fonction du rapport de compression

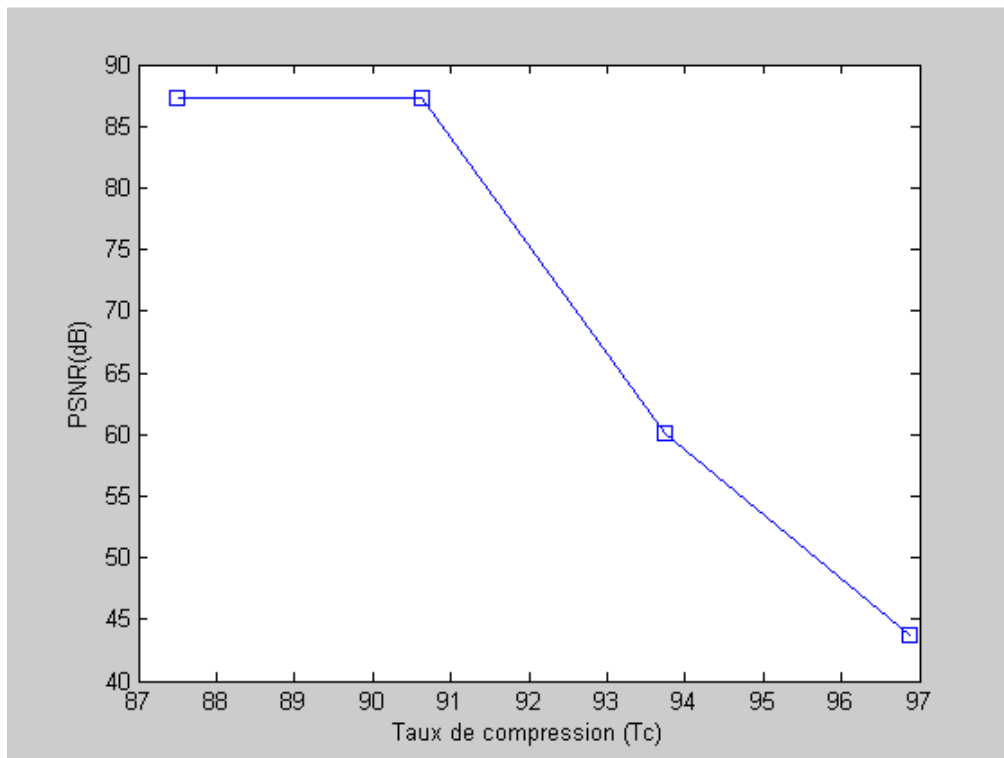


Figure IV-10 : variation du PSNR en fonction du taux de compression

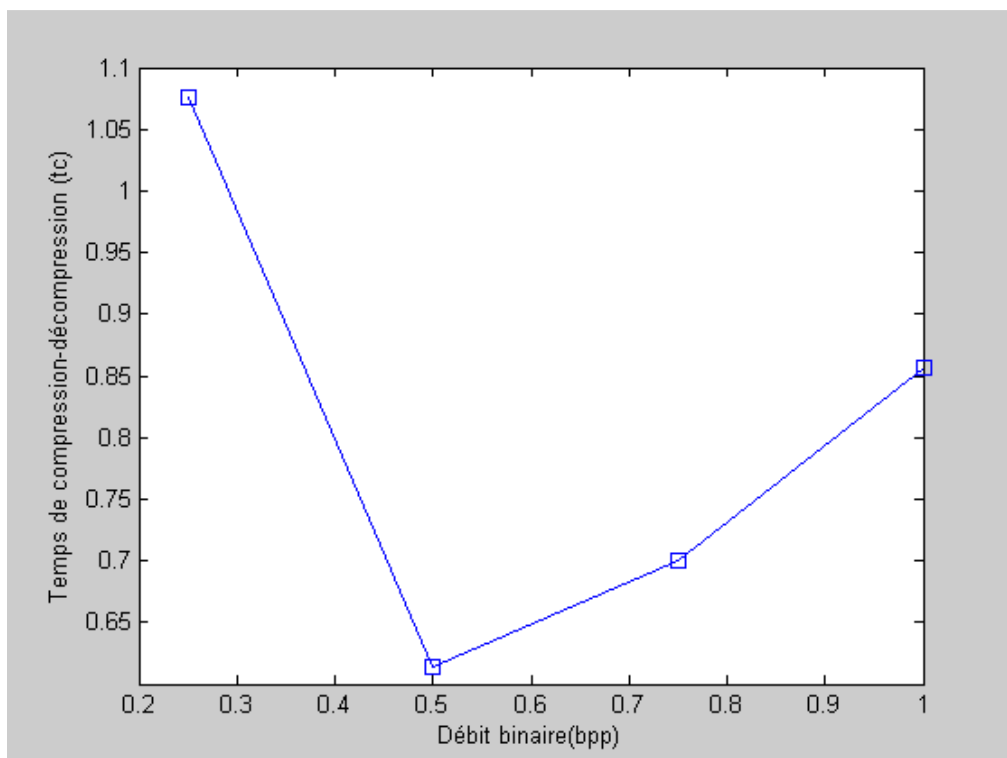


Figure IV-11 : variation du temps compression et décompression en fonction du débit binaire

2) Pour l'image MSG du canal visible prise à 14h :

Débit (bpp)	MSE	PSNR (dB)	temps de compression (seconds)
0.25	2.615982	43.954456	0.842596
0.5	0.033867	62.833051	0.642377
0.75	0.000046	91.524390	0.638305
1	0.000046	91.524390	0.576383

Tableau IV-3 : résultats obtenus avec l'image MSG du canal visible prise à 14h

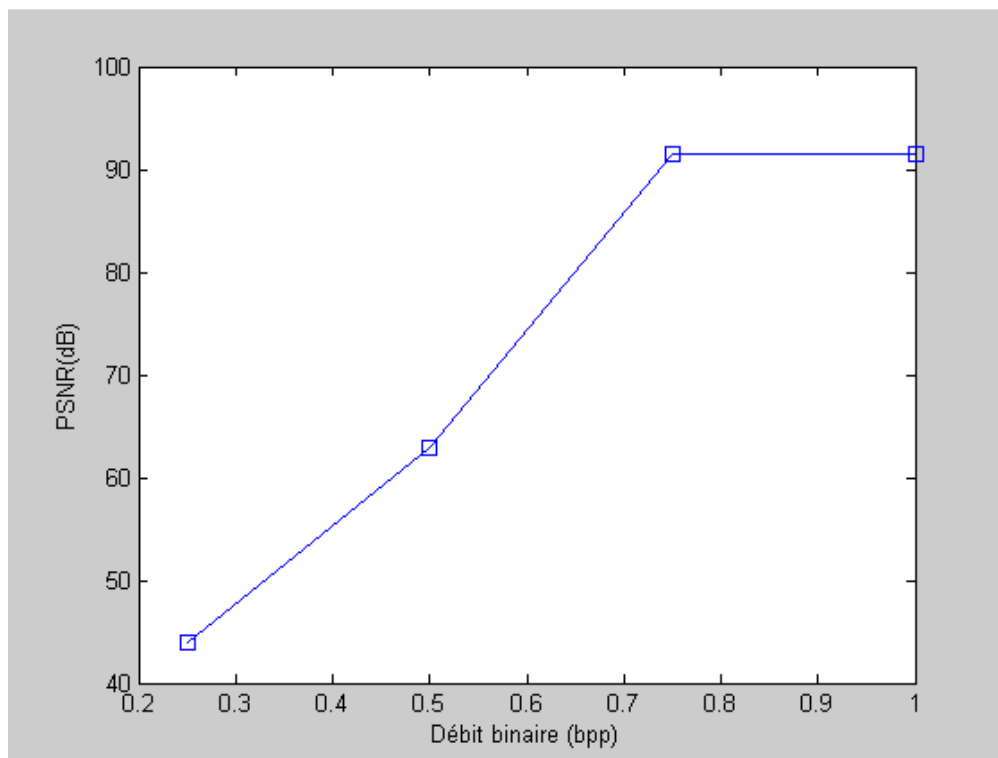


Figure IV-12 : variation du PSNR en fonction du débit binaire

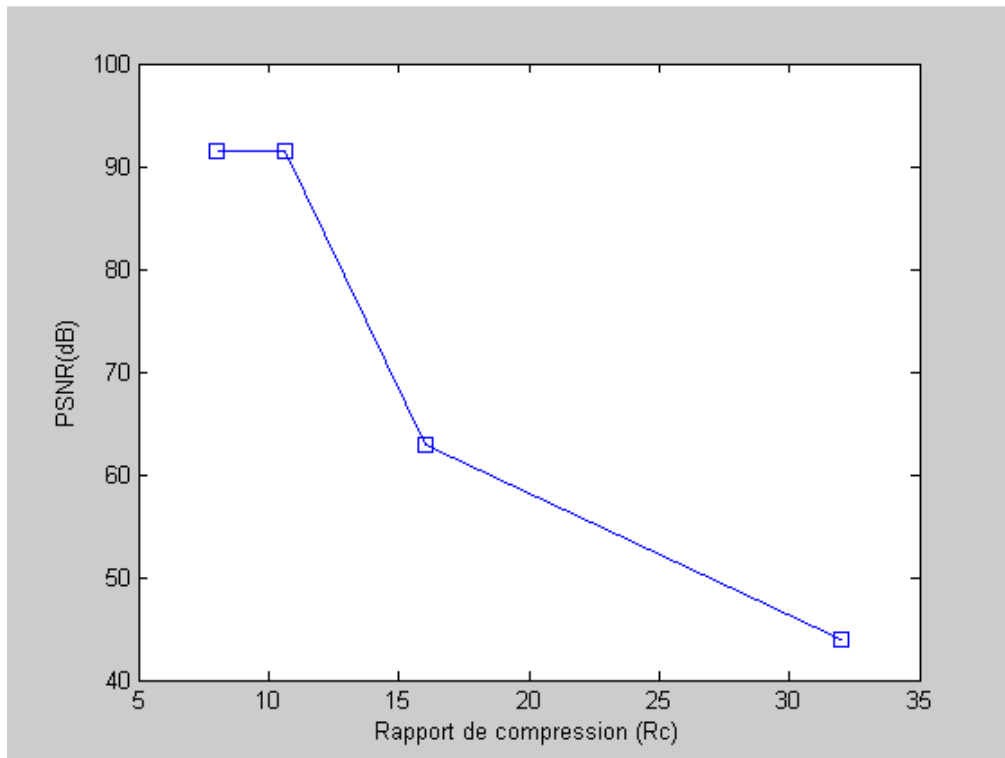


Figure IV-13 : variation du PSNR en fonction du rapport de compression

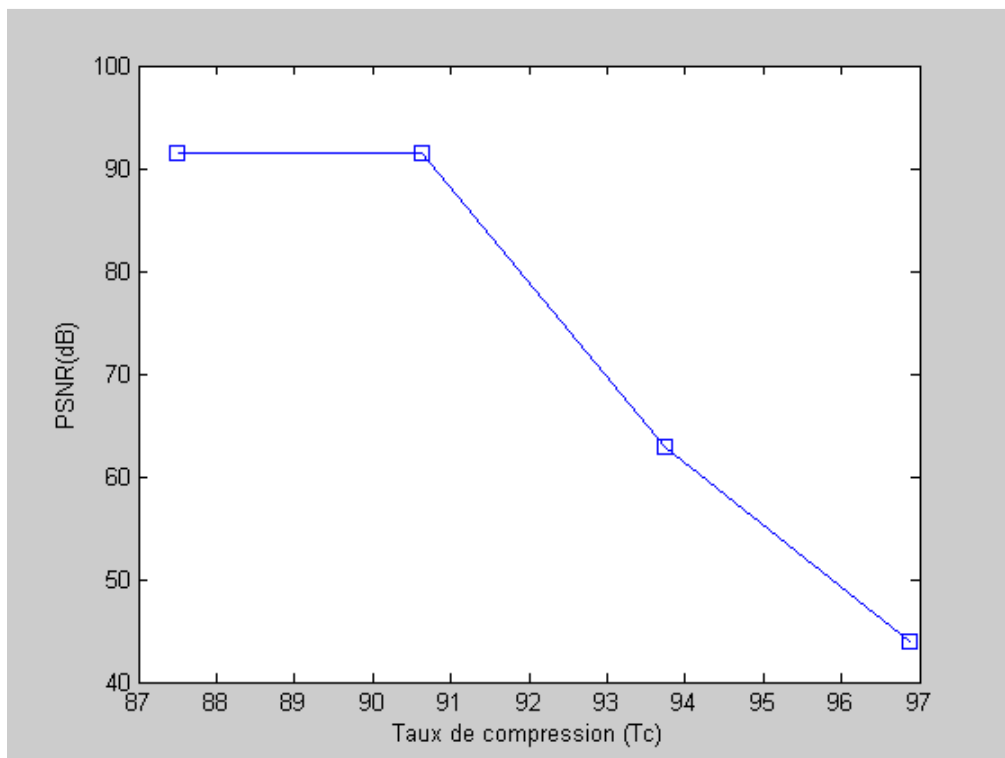


Figure IV-14 : variation du PSNR en fonction du taux de compression

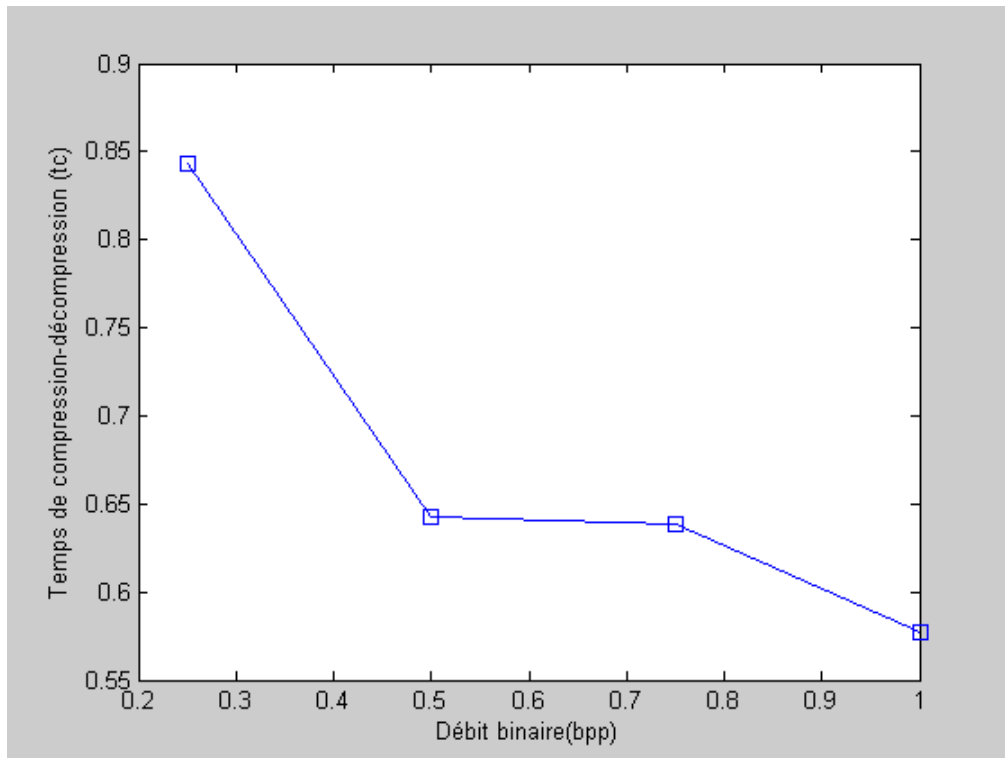


Figure IV-15 : variation du temps de compression et de décompression en fonction du débit binaire

3) Pour l'image MSG du canal infrarouge prise à 8h :

Débit (bpp)	MSE	PSNR (dB)	temps de compression (seconds)
0.25	2.193108	44.720204	1.253323
0.5	0.000008	99.305903	0.649759
0.75	0.000008	99.305903	0.618119
1	0.000008	99.305903	0.647150

Tableau IV-4 : résultats obtenus avec l'image MSG du canal infrarouge prise à 8h

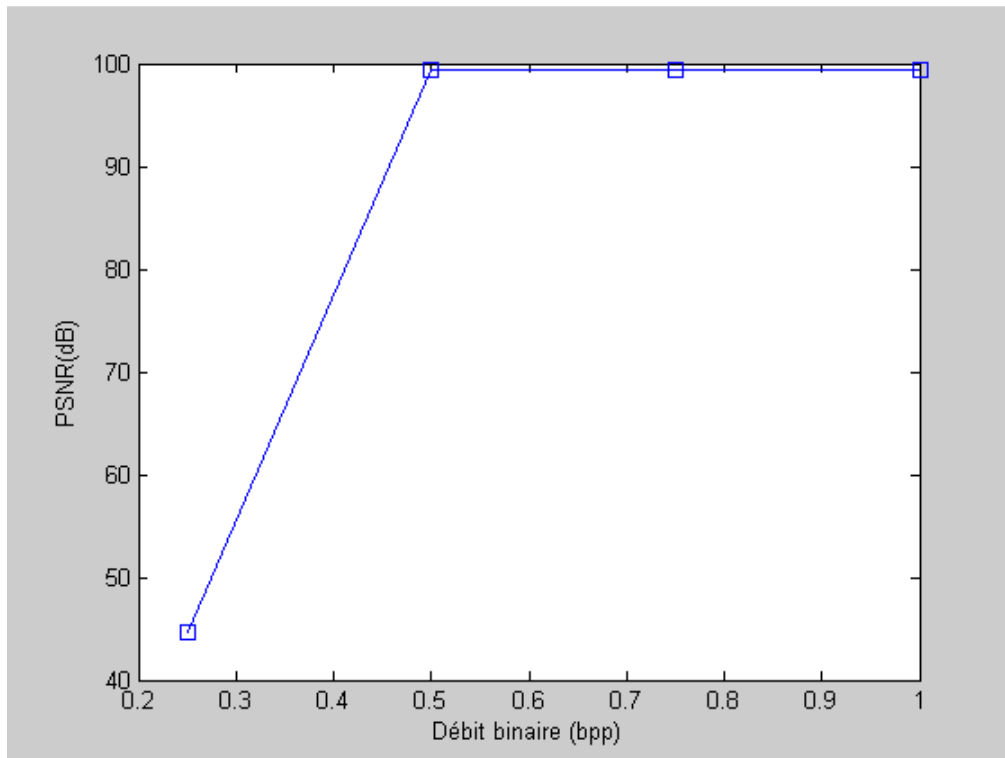


Figure IV-16 : variation du PSNR en fonction du débit binaire

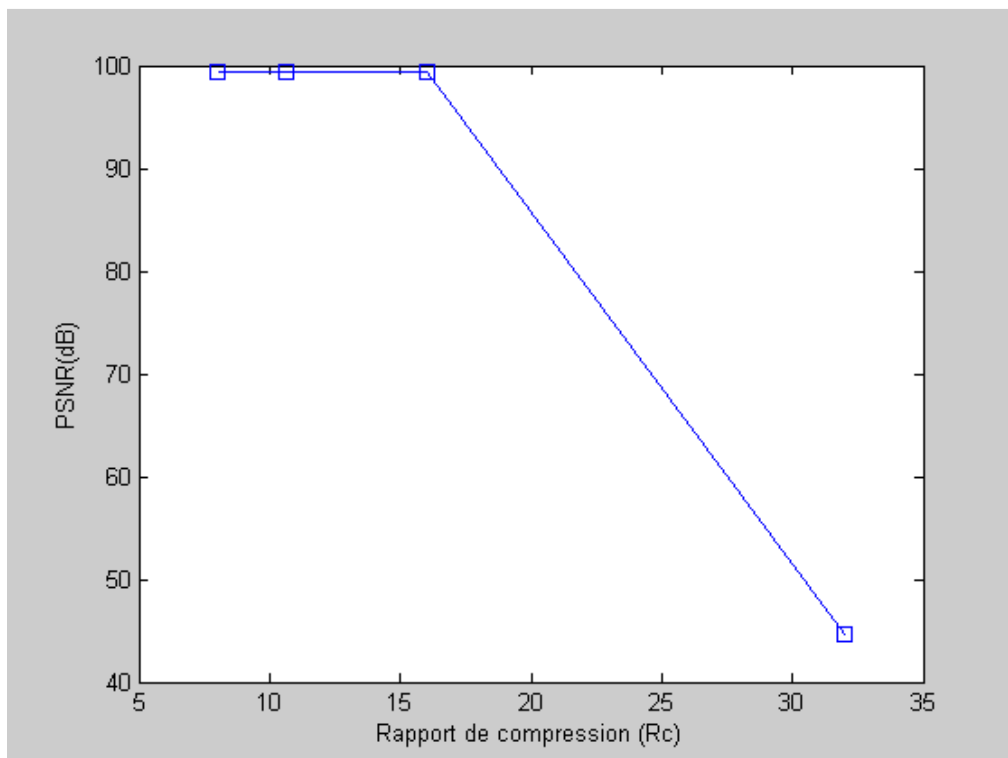


Figure IV-17 : variation du PSNR en fonction du rapport de compression

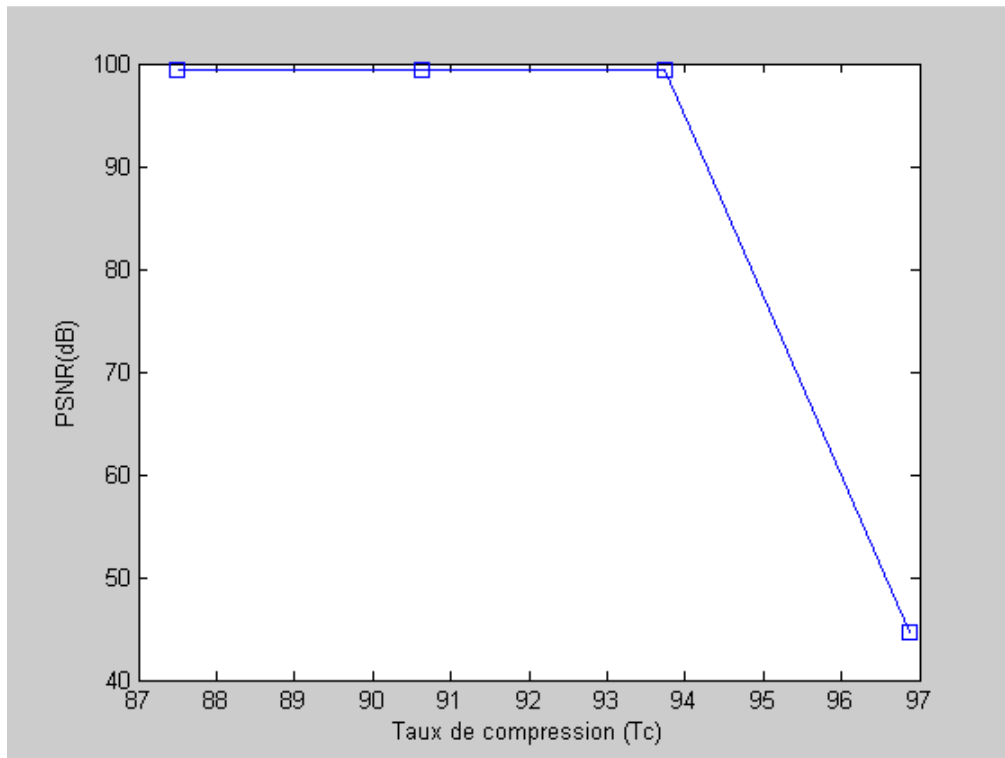


Figure IV-18 : variation du PSNR en fonction du taux de compression

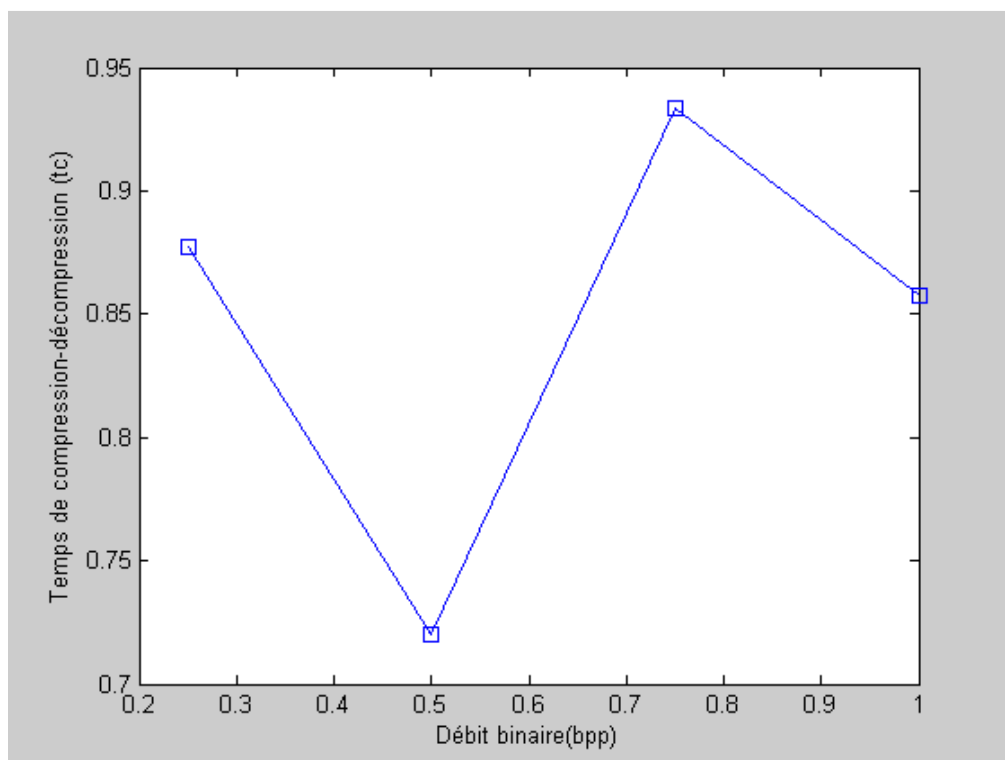


Figure IV-19 : variation du temps de compression et de décompression

en fonction du débit binaire

4) Pour l'image MSG du canal infrarouge prise à 14h :

Débit (bpp)	MSE	PSNR (dB)	temps de compression (seconds)
0.25	1.951950	45.226117	0.815432
0.5	0.000050	91.176769	0.585870
0.75	0.000050	91.176769	0.661944
1	0.000050	91.176769	0.930894

Tableau IV-5 : résultats obtenus avec l'image MSG du canal infrarouge prise à 14h

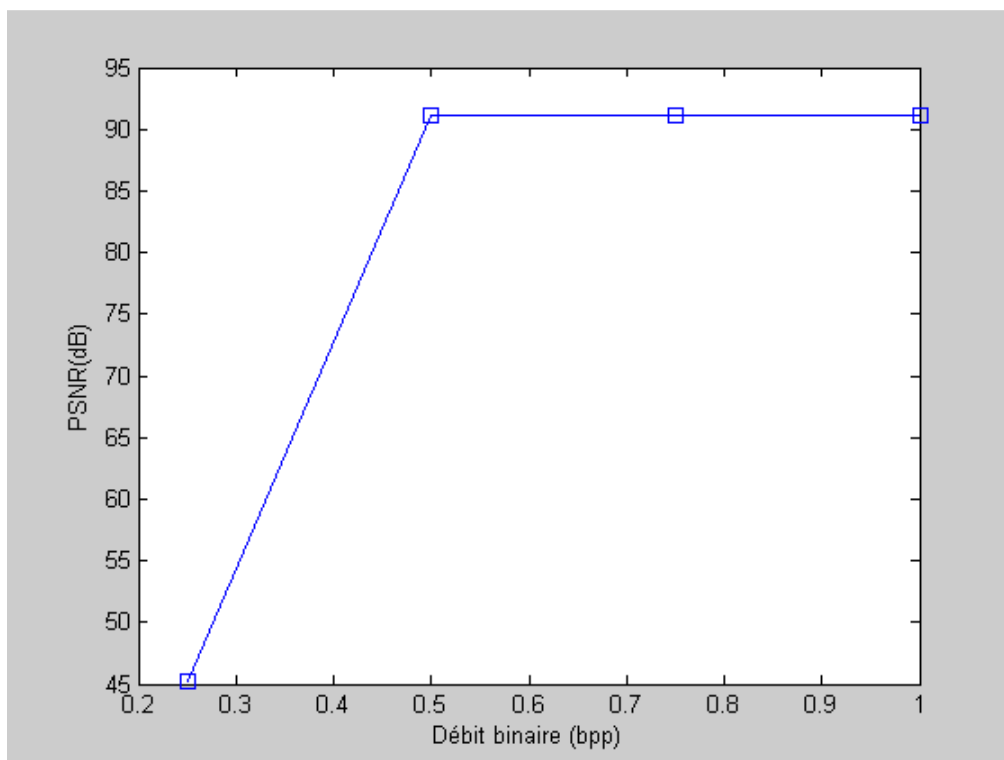


Figure IV-20 : variation du PSNR en fonction du débit binaire

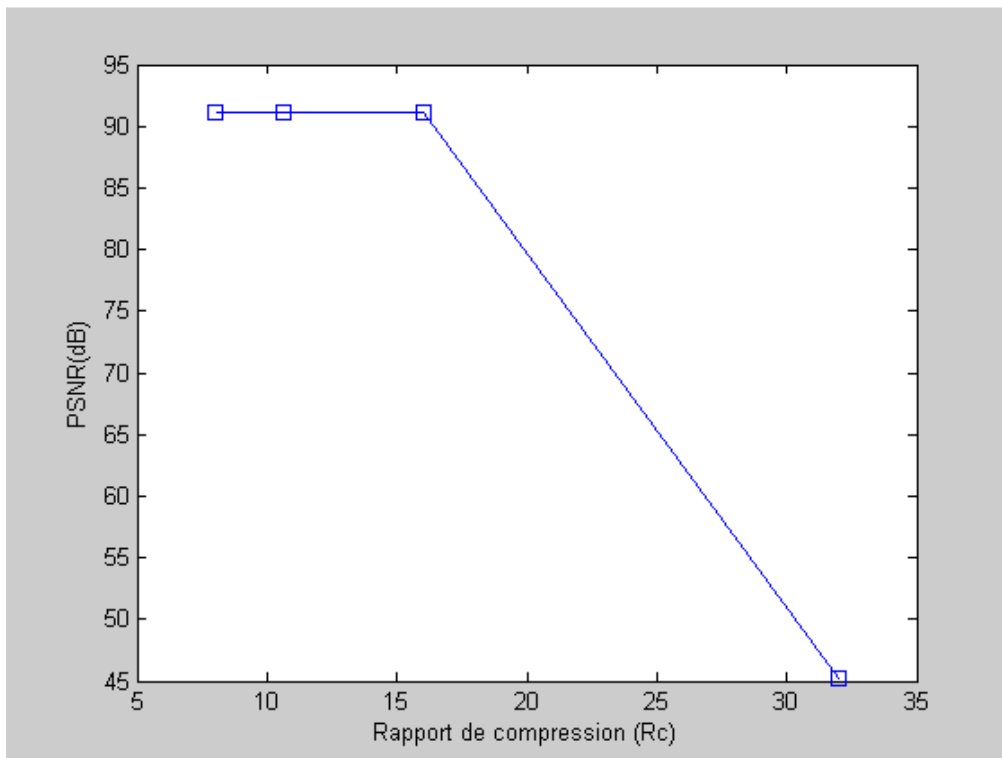


Figure IV-21 : variation du PSNR en fonction du rapport de compression

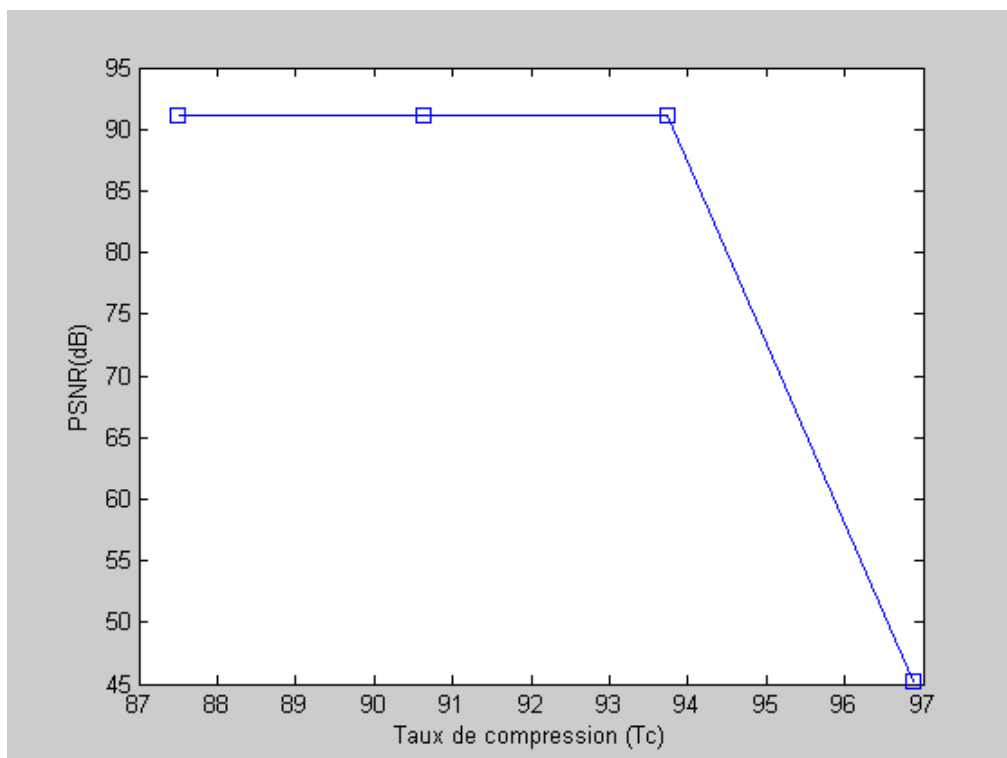


Figure IV-22 : variation du PSNR en fonction du taux de compression

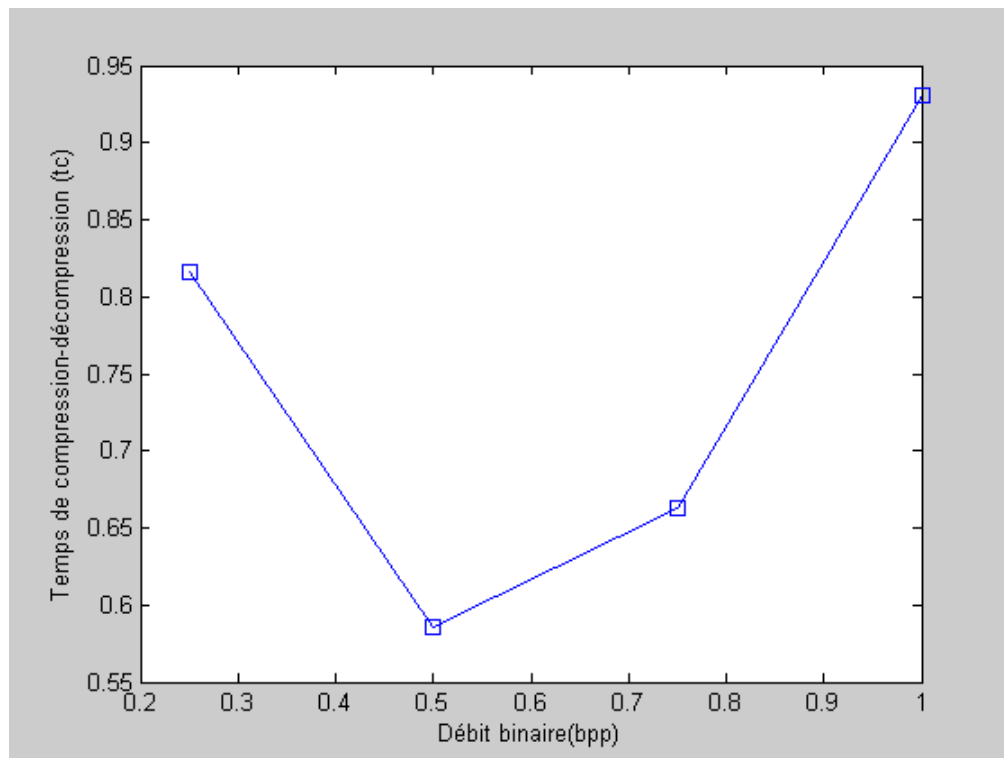


Figure IV-23 : variation du temps de compression et de décompression en fonction du débit binaire

IV.4 Interprétation des résultats :

A partir des différents graphes précédents, nous sommes parvenus aux conclusions suivantes :

- Le PSNR s'accroît de manière proportionnelle par rapport au débit binaire et se stabilise à une certaine valeur quelque soit le débit. Cette stabilité du PSNR révèle une invariabilité de l'aspect visuel des images reconstituées. Prouvant ainsi, que l'algorithme JPEG 2000 décode de manière très performante l'information contenue dans l'image, ce qui lui permet de supprimer le maximum de détails sans dégrader l'image. De plus, on remarque que les valeurs du PSNR obtenues sur les images infrarouges, sont plus élevées que celles obtenues sur les images visibles et atteignent leur valeur stable plus rapidement. Ceci est dû principalement, au contenu de l'image. En effet, l'image visible est plus contrastée que l'image infrarouge, donc la perte de détails fins serait plus élevée dans celle-ci.
- Le PSNR diminue, quand le taux de compression s'accroît. Ceci s'explique par le fait, que quand, le taux de compression est élevé, l'image est fortement quantifiée. Signifiant, ainsi, qu'une grande quantité d'informations de hautes fréquences (détails) est retirée de l'image, pour atténuer son empreinte mémoire. Ce qui engendre une grande dégradation de l'image restituée, entraînant ainsi, la diminution de la valeur du PSNR.
- Le temps de compression et de décompression reste relativement faible, car de manière générale le processus dure moins d'une seconde. En examinant les graphes, on remarque que le temps de réponse diminue quand le débit augmente.

IV.5 Aspect visuel des images restituées :

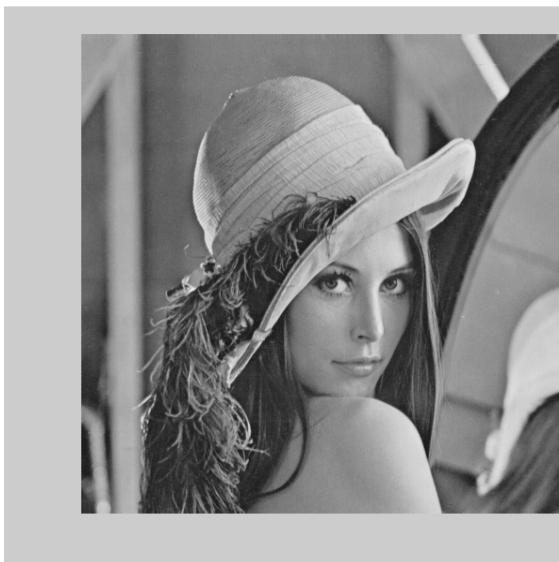
Dans le domaine de la compression d'images, le critère le plus important pour leurs estimations après reconstitution, est le critère visuel. Seul l'œil et l'analyse que fait le cerveau, des images captées, peuvent juger la qualité de ces dernières. Pour cette raison, nous allons vous présenté dans ce qui suit les différentes images restituées par l'algorithme JPEG 2000, pour quelques valeurs du débit binaire.



débit=1bpp



débit=0.75bpp



débit=0.5bpp



débit=0.25bpp

Figure IV-24 : Images Lina reconstituées avec l'algorithme de décompression JPEG2000 avec différentes valeurs de débit binaire

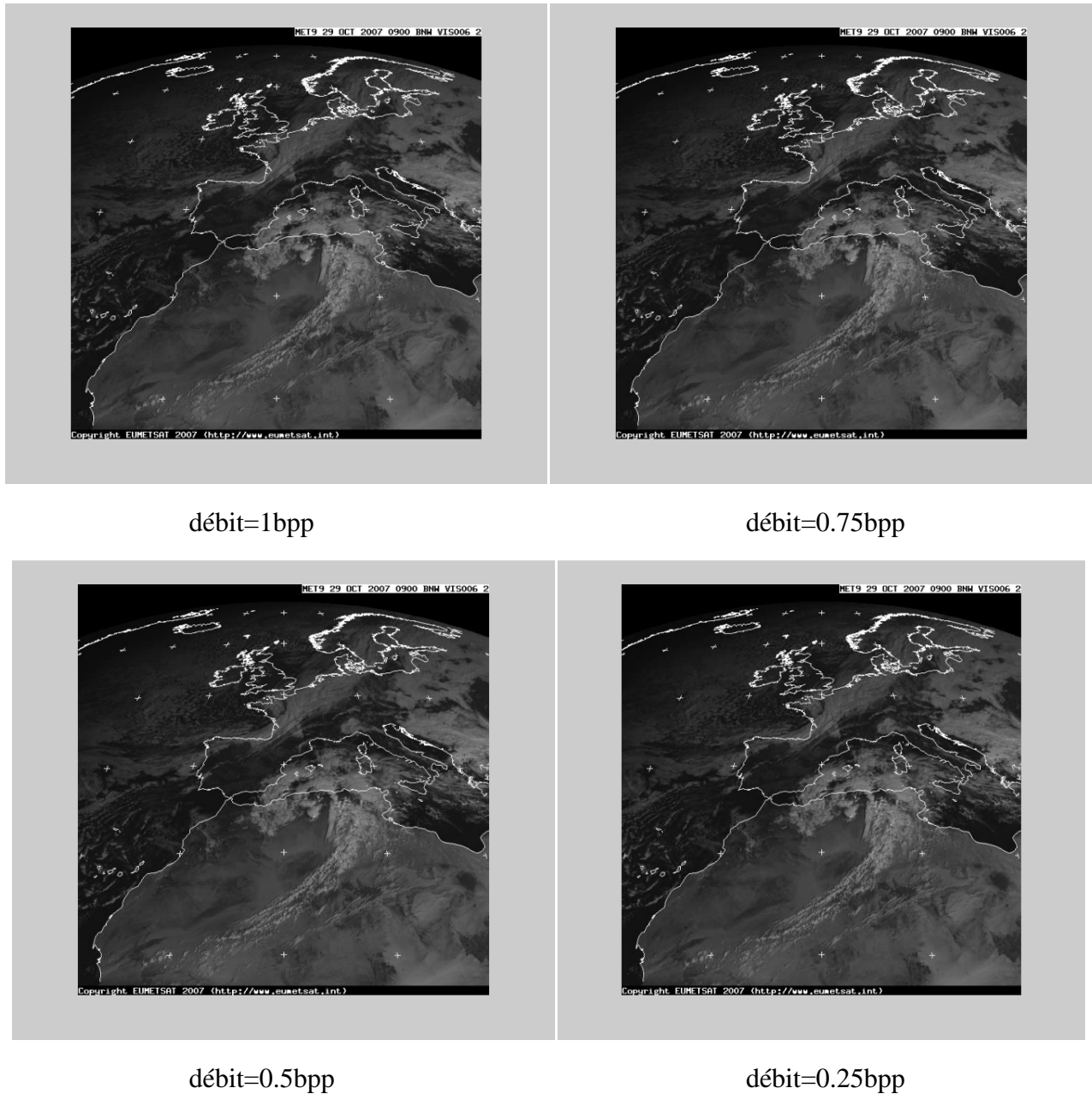


Figure IV-25 : Images MSG du canal visible (9h) reconstituées avec l’algorithme de décompression JPEG2000 avec différentes valeurs de débit binaire

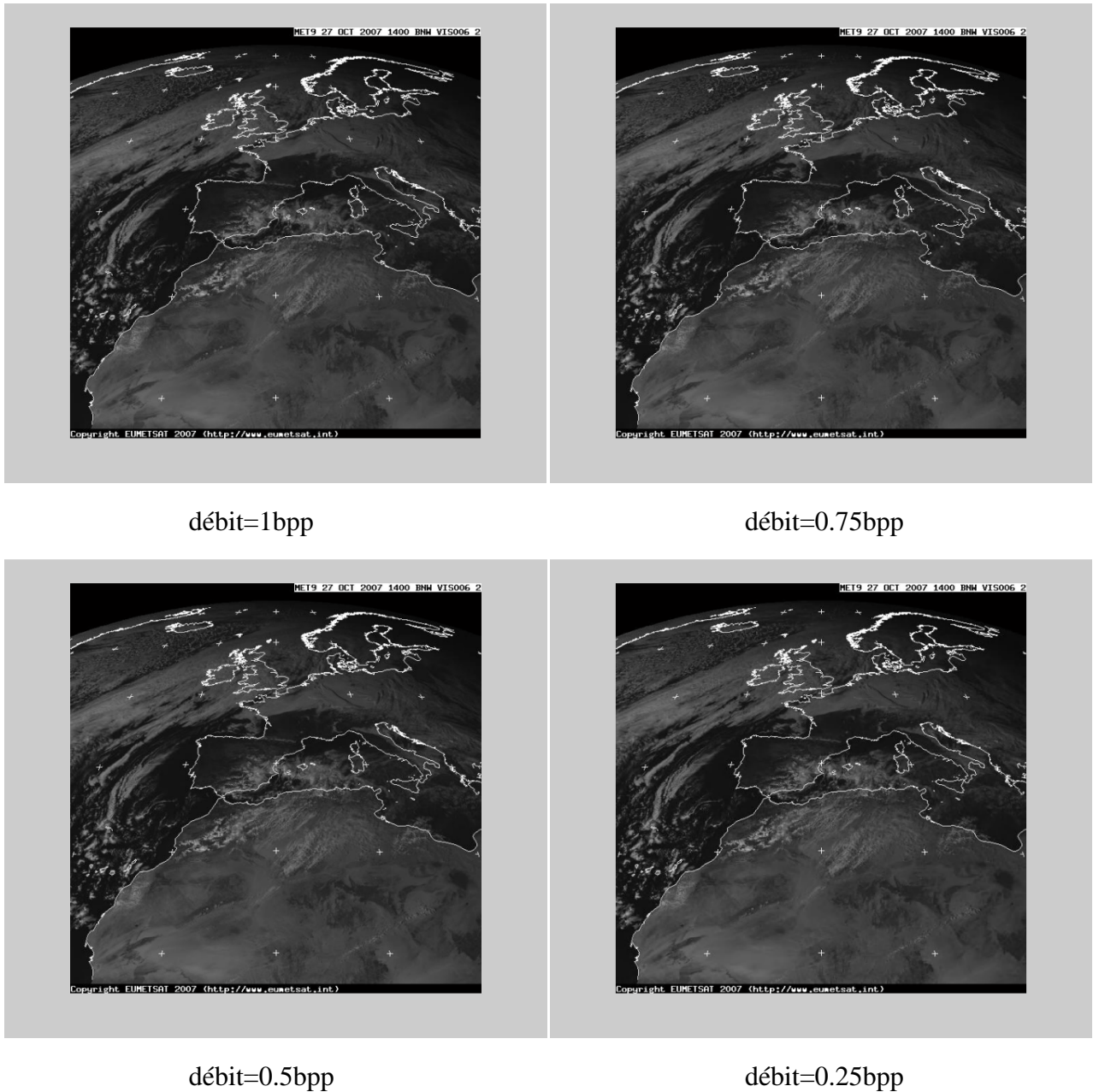


Figure IV-26 : Images MSG du canal visible (14h) reconstituées avec l’algorithme de décompression JPEG2000 avec différentes valeurs de débit binaire

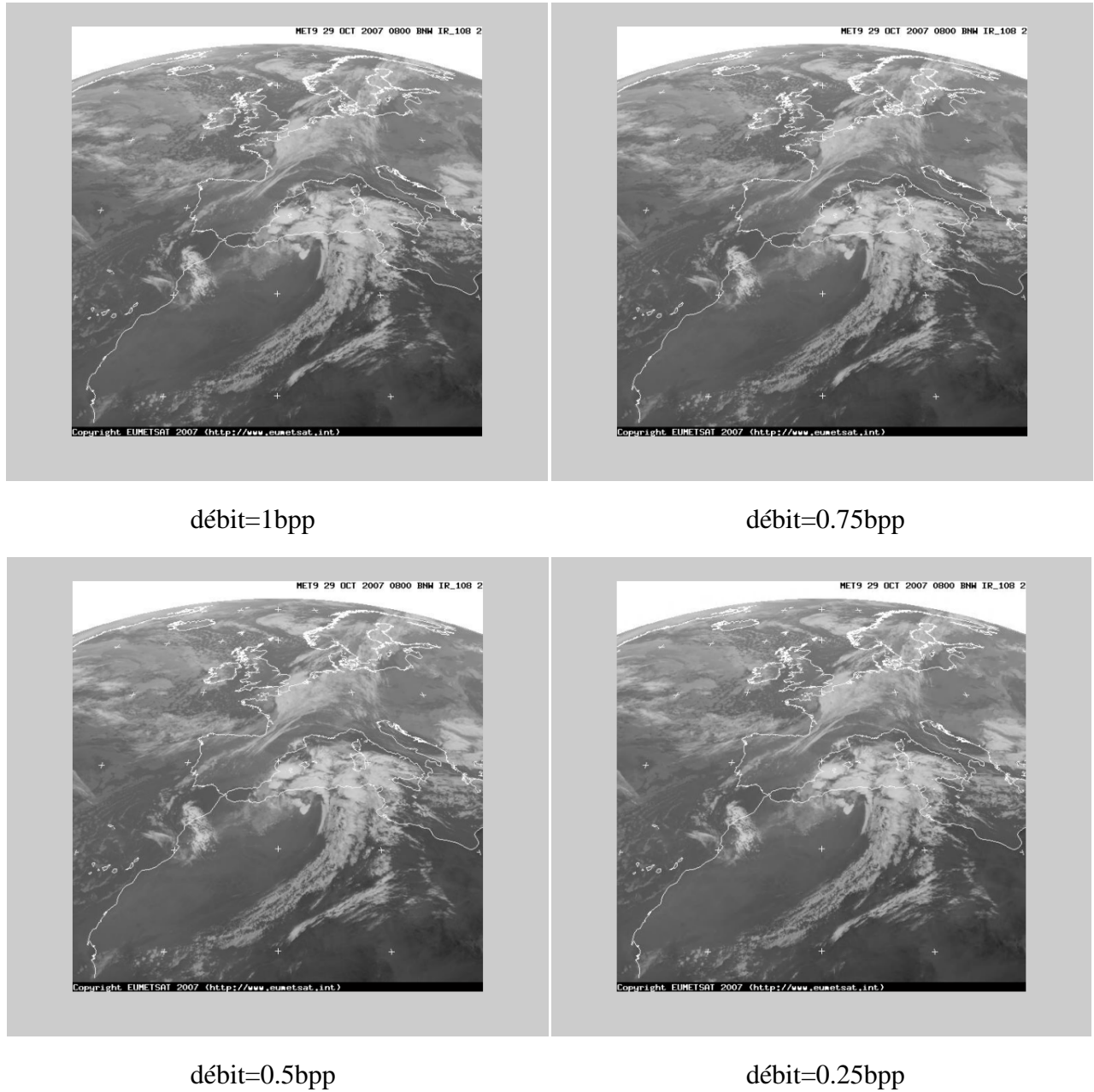


Figure IV-27 : Images MSG du canal infrarouge (8h) reconstituées avec l’algorithme de décompression JPEG2000 avec différentes valeurs de débit binaire

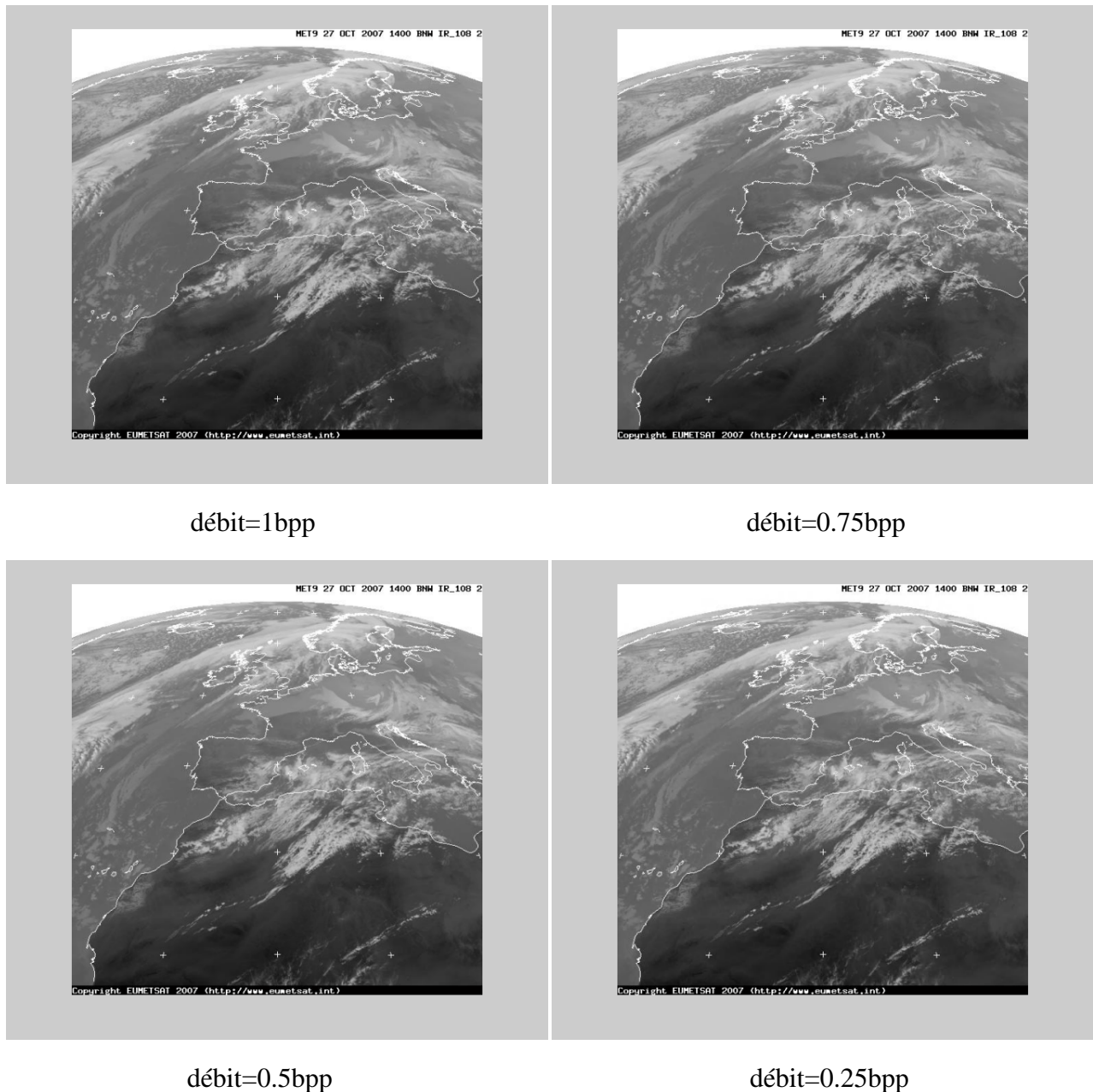


Figure IV-28 : Images MSG du canal infrarouge (14h) reconstituées avec l'algorithme de décompression JPEG2000 avec différentes valeurs de débit binaire

D'après les figures précédentes, on remarque que les images restituées sont très proches des images originales. Elles sont de bonne qualité, car même pour celles qui sont fortement compressées (correspondantes au débit binaire de valeur égale à 0.25bpp), l'effet de lissage est totalement imperceptible à l'œil nu. Ce qui démontre que l'algorithme JPEG 2000 permet d'atteindre des taux de compression très élevés, tout en gardant une bonne qualité visuelle des images reconstituées.

IV.6 Conclusion :

Les résultats obtenus montrent que l'algorithme JPEG 2000 permet d'avoir un fort taux de compression, tout en sauvegardant un bon aspect visuel des images restituées.

Pour ce qui concerne le temps de codage et de décodage de l'algorithme, on remarque qu'il ne dépasse pas les 2 secondes, ce qui veut dire que l'algorithme JPEG 2000 est très rapide.

On conclue donc, que l'algorithme du standard JPEG 2000 est performant et rapide, ce qui justifie son utilisation fréquente en compression d'images.

CONCLUSION GENERALE

Conclusion générale

Le travail exposé dans ce mémoire et la compression des images Météosat seconde génération (MSG) en utilisant l'algorithme de compression du standard JPEG 2000.

Dans un premier temps, on a fait une étude générale des différentes méthodes de compression d'images et des standards de compression existants. On s'est intéressé aux différentes propriétés caractéristiques des images MSG. Une étude des différentes étapes de compression et de décompression de l'algorithme de la norme JPEG 2000 a été présentée.

Les performances du standard JPEG 2000 sur les images MSG sont évaluées au chapitre IV. On a utilisé cinq paramètres de compression qui sont : le débit binaire, le taux de compression, le MSE, le PSNR et le temps d'exécution du processus codage-décodage.

Les différents résultats obtenus ont montré, que le standard JPEG 2000 peut atteindre des taux de compression très élevés (faible débit binaire), tout en gardant une bonne qualité visuelle de l'image restitué et un temps d'exécution considérablement faible.

L'étude réalisée dans ce projet nous a permis de découvrir de manière générale le domaine du traitement d'images numériques et en particulier la compression d'images satellitaires. On s'est aussi familiarisé avec certaines commandes de programmation sous Matlab pour le traitement d'images.

BIBLIOGRAPHIE

- [1] : A. Tremeau, Ch. Ferandez-Maloigne, P. Bonton « Image numérique couleur : de l'acquisition au traitement » TRS 72, Vol 02, 2004.
- [2] M^{elle} Drif Naima « Segmentation d'images Météosat en ciel clair en vue de la construction d'une image de référence », mémoire de magister, télédétection département d'électronique. UMMTO, 2008.
- [3] SEDDI.F « Segmentation d'image météoat par une méthode statistique supervisée » thèse ingénieur en électronique, UMMTO, 2005
- [4] M. Benbahloul et T.Moukhtari « Codage d'image fixe selon JPEG 2000 », thèse d'ingénieur département électronique, UMMTO, 2003.
- [5] M^{me} Ouyed Ouiza « Etude comparative des méthodes de compression d'images : application aux images MSG » thèse de magister, département électronique, UMMTO, 2008.
- [6] I.Aouadi O.Hammami V.Martin « Bi-codeur entropique JPEG-2000 pour applications mobiles » École Nationale Supérieure de Techniques Avancées, Paris (support de cours) 2004.
- [7] H.Messaoudi,N.Ould-driss et S.Zerhraoui « segmentation d'image par la transformée en ondelettes sous Matlab », thèse d'ingénieur département électronique, UMMTO, 2003.
- [8] N.Nait Amara et R.Chouchane « compression d'image fixe par classification de régions en associant les ondelettes et le fractales » thèse d'ingénieur département électronique, UMMTO .2008
- [9] Z .Drici , T.Selmani et M.Yahiatene « Application des ondelettes biorthogonales pour le codage d'image numériques » thèse d'ingénieur , département électronique, UMMTO, 2002.
- [10] Sabiha Ould Yahia et Fazia Abrous « Compression d'images fixes par la Transformée en Cosinus Discret (TCD) » thèse d'ingénieur, département électronique, UMMTO, 1999.
- [11] M^{elle} Bakri Naima et M^r Bendifallah Oussama « La compression d'images fixes par la transformation LOT » thèse d'ingénieur, département électronique, UMMTO, 1999.
- [12] M^r Mouzarine Hocine « Codage des images numériques par TCD et fractales » thèse d'ingénieur, département électronique, UMMTO, 2002.
- [13] M^{elle} Akrouf Leila « compression d'images numérique par fractales dans le domaine DCT » thèse d'ingénieur, département électronique, UMMTO, 2003.
- [14] José Marconi M. Rodrigues « Transfert sécurisé d'images par combinaison de techniques de compression, cryptage et marquage » thèse de doctorat, université de Montpellier II.

[15] Mohammed Benabdellah « Outils de compression et de crypto-compression : Applications aux images fixes et video » these de doctorat, faculte des sciences, universite Mohammed V-Agdal.

[16] Antonin Descampe et François-Olivier Decaux « étude et conception d'un décodeur hardware JPEG 2000 destine au cinéma numérique » ingénieur civil électricien, Faculté des sciences appliquées, Université Catholique de Louvain.

[17] <http://www.steps-method.com>

Résumé

Le but de notre travail est l'étude de la compression d'images MSG par le standard JPEG 2000, afin de déterminer les différentes caractéristiques de ce dernier. Cette étude est divisée en quatre parties différentes.

Dans le premier chapitre on a fait une étude générale des caractéristiques des images numériques, des méthodes réversibles et irréversibles de compression et de quelques normes de compression (JPEG, JPEG 2000 et MPEG).

Dans le deuxième chapitre, nous avons étudié les caractéristiques des satellites Météosat MSG, on a expliqué le fonctionnement de leurs canaux spectraux et on s'est intéressé aux caractéristiques optiques des objets au sol et des paramètres influençant sur l'acquisition des images par ces satellites.

Dans le troisième chapitre on a expliqué les différentes étapes de compression et de décompression de l'algorithme du standard JPEG2000 qui sont les suivantes : le prétraitement, la transformée en ondelettes discrète, le codage entropique et l'allocation de débit.

Dans le quatrième chapitre on a présenté les différents résultats obtenus par l'application de l'algorithme de JPEG 2000 sur les images MSG.

On a clôturé cette étude par une conclusion générale dans laquelle sont présentés les résultats obtenus par notre étude.