

Chapitre 2

Introduction à la compression des images

2.1. Introduction

L'homme a toujours voulu découvrir la beauté des planètes, les vues de l'espace. Il a donc envoyé des satellites capables de photographier ceux-ci. Mais, l'un des problèmes majeurs est la transmission de ces photos du satellite à la terre. Cette communication se fait à l'aide des ondes électromagnétiques.

La transmission des informations est d'autant plus facile que le nombre d'informations est faible. Il est donc avantageux d'en réduire le nombre. Pour atteindre cet objectif, une technique employée dans de nombreux domaines est la compression des informations. L'utilisation d'algorithmes de compression d'images permet une réduction importante de la quantité de données.

Ce chapitre est une introduction à la compression des images. Après une brève description des concepts élémentaires, un modèle de compression et les méthodes de compression sont présentés. Ensuite, l'aspect plus mathématique concernant les théorèmes fondamentaux utilisés pour la compression est considéré. Enfin, les techniques des compressions JPEG d'images fixe sont traitées.

2.2. Objectif de la compression

La représentation numérique des signaux comparée aux signaux analogues a besoin d'un nombre très grand de bits pour le stockage et la transmission. En effet, une vidéo de couleur de basse résolution (qualité de télévision), de 30 images par seconde avec chaque armature contenant 24 bits par pixel de couleur, a besoin de plus de 210 millions de bits par seconde de stockage. En conséquence, un film d'une heure digitalisé de couleur exigerait approximativement 95 Go de stockage. La capacité de stockage pour la télévision à haute définition (HDTV) ayant une résolution à 60 images par seconde est bien plus grande. Un

film d'une heure digitalisé de couleur de qualité HDTV exigera approximativement 560 Go de stockage [Ach].

La transmission de ces signaux numériques par les voies de transmission de largeur de bande limitée est un grand défi et parfois impossible. Bien que le coût de stockage ait diminué rigoureusement pendant la dernière décennie, dû à l'avancement significatif dans la microélectronique, la compression de données est la technique utilisée pour réduire les redondances dans la représentation de données. Cette redondance permet de diminuer les conditions de stockage des données et par conséquent les coûts de communication. La réduction de la condition de stockage est équivalente à augmenter la capacité de la largeur de bande du support. Ainsi le développement des techniques efficaces de compression continuera à être un défi de conception pour les futurs des systèmes de communication et des applications avancées de multimédia [Ach].

2.2.1. Avantages de la compression de données [Ach]

L'avantage principal de la compression est la réduction de la taille de stockage de données. Il offre également une approche attrayante pour réduire le temps de communication dans la transmission de grands volumes de données sur des liens de long-courrier par l'intermédiaire d'une utilisation efficace d'une largeur de bande plus élevée disponible dans les liaisons de transmission de données. Cette réduction du temps de communication se traduit par une réduction significative du coût de communication qui est une conséquence de la réduction du débit. La compression de données augmente également la qualité de la présentation de multimédia par des voies de transmission de largeur de bande limitée. Une chaîne de télévision simple d'émission de 6 MHz peut porter des signaux de HDTV pour fournir une meilleure qualité audio et une vidéo à des taux beaucoup plus élevés et une résolution augmentée sans conditions additionnelles de largeur de bande. La compression courte à rendement élevé a créé de nouvelles applications créatrices telles que la bibliothèque numérique, l'archivage numérique, la télémédecine et le divertissement numérique, etc. [Ach].

2.2.2. Inconvénients de la compression de données [Ach]

Bien que la compression de données offre de nombreux avantages, elle a quelques inconvénients selon le domaine d'application et la sensibilité des données. Par exemple, les frais généraux supplémentaires encourus en codant et en décodant des processus sont l'un des inconvénients les plus sérieux de la compression de données, qui décourage son utilisation

dans quelques secteurs (par exemple, dans beaucoup de grandes applications de base de données). Ces frais généraux supplémentaires sont habituellement exigés afin d'identifier ou interpréter les données comprimées. Dans de grandes bases de données statistiques ou scientifiques où les changements de la base de données ne sont pas très fréquents, le processus de décodage a un plus grand impact sur l'exécution du système que le procédé de codage. L'accès à un disque simple dans une grande base de données exige de décompresser la base de données entière avant d'accéder au disque désiré. Après l'accès et probablement la modification des données, la base de données est de nouveau comprimée pour être stocker. La compression de données réduit généralement la fiabilité des articles de données. Par exemple, une erreur à bit unique en code comprimé fera mal interpréter le décodeur produisant des données incorrectes. La transmission des données comprimées très sensibles (par exemple, l'information médicale) par une voie bruyante de transmission (telle que des médias sans fil) est risquée parce que les erreurs d'éclat présentées par le canal bruyant peuvent détruire les émissions de données. Un autre problème de la compression de données est la rupture des propriétés de données parce que les données comprimées sont différentes des données originales. Par exemple, le triage et la recherche des arrangements dans les données comprimées peuvent être inapplicables pendant que la commande lexicologique des données originales n'est plus préservée dans les données comprimées. Pendant l'exécution du matériel, la complexité supplémentaire de la compression de données peut augmenter le coût du système et réduire son efficacité, particulièrement dans les domaines d'applications qui exigent l'exécution de très basse puissance VLSI (Very Large Scale Integration) [Ach].

2.2.3. Types de compression

La compression de données consiste à obtenir des fichiers plus léger, afin d'améliorer la vitesse de transfert sur internet ou limiter l'espace de stockage utilisé sur un disque dur. Il existe deux principaux types de compression:

- **La compression sans perte** : Appelée aussi « compactage ». Cette solution consiste simplement à coder les données binaires de manière plus concise dans un fichier. Elle permet ainsi de retrouver la totalité des informations après une procédure de décompactage.
- **La compression avec perte** : Concernant essentiellement les fichiers de média (image, son, vidéo), elle consiste en une réduction de l'information basée sur notre propre limite humaine à percevoir ces médias. Puisque l'œil ne perçoit pas nécessairement tous les

détails d'une image, il est possible de réduire la quantité de données de telle sorte que le résultat soit très ressemblant à l'original, voire identique, pour l'œil humain [Dug].

2.3. Concepts élémentaires

2.3.1. Image Digitale

2.3.1.1. Images vectorielles

Une image vectorielle ne stocke pas le résultat du dessin sous la forme de pixels (bitmap), mais de la façon de le dessiner par un ensemble d'objets géométriques (lignes, cercles, polygones, courbes de Bézier, texte, etc.) définis par différents attributs (coordonnées, couleur, épaisseur de trait, remplissage, etc.) [Tho]. La figure 2.1 montre un exemple d'une image vectorielle.

Les domaines d'utilisation de ces images sont très variés : dessin industriel, PAO, Systèmes d'Information Géographiques (SIG), Internet, etc.

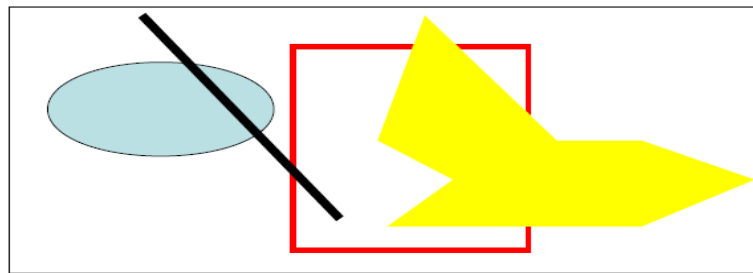


Figure 2.1 : Image vectorielle [Tho].

*** Avantages des images vectorielles [Tho]**

- Elles sont Adaptées au stockage d'images composées de formes géométriques.
- Elles peuvent aisément être redimensionnées.
- Elles prennent moins de place qu'une image bitmap.

*** Inconvénients des images vectorielles [Tho]**

- Elles peuvent difficilement stocker des images complexes comme des photographies.
- L'affichage d'une image vectorielle peut prendre plus de temps que l'affichage d'une image bitmap de complexité égale.

2.3.1.2. Images bitmap

Une image bitmap est un ensemble de points (pixels), caractérisée par sa discrétisation et sa quantification [Tho]. La figure 2.2 montre un exemple d'une image bitmap.

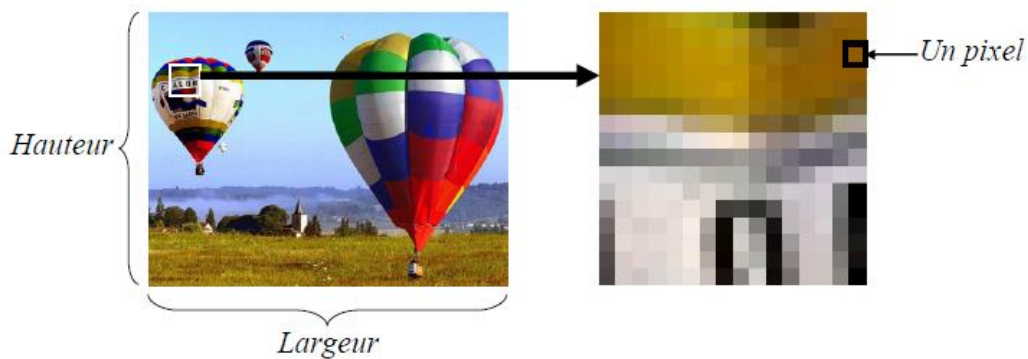


Figure 2.2 : Image bitmap [Sun].

* Avantages des images bitmap [Tho]

- Elles peuvent facilement être créées et stockées dans un tableau de pixels représentant l'image.
- La lecture/écriture d'un pixel facilite la représentation de l'image comme une grille.
- Elles peuvent facilement être affichées sur un écran ou être imprimées.

* Inconvénients des images bitmap [Tho]

- Les fichiers peuvent être très gros (nécessité de compression)
- Problème de changement d'échelle (apparition d'effets de marches d'escalier ou de flou avec interpolation).
- Les dimensions de l'image doivent être prévues pour la résolution de l'interface de sortie (écran, imprimante).

2.3.2. Mesures de performance

2.3.2.1. Taux de compression

Les algorithmes de compression améliorent le taux de compression en minimisant le débit de l'image compressée. Le débit d'une image est le nombre de bits par pixel, plus ce débit est minimisé, plus le taux de compression est maximisé. La mesure qui définit le taux de compression exprimé en pourcentage est définie par [Gab] :

$$CR = \frac{\text{Nombre de bits de l'image compressée}}{\text{Nombre de bits de l'image originale}} \quad (2.1)$$

2.3.2.2. Entropie

L'entropie est la quantité d'informations que contient une image. Par exemple, une image dont tous les pixels sont identiques contient très peu d'informations car elle est très redondante, son entropie est faible. Par contre, une image dont tous les pixels ont une valeur aléatoire contient beaucoup d'informations, par conséquent son entropie est forte.

En pratique, l'entropie d'une image numérique est inversement liée à la probabilité d'apparition des pixels de l'image. Plus une valeur d'un pixel est rare, plus sa probabilité d'apparition est faible, cela contribue à une entropie globale plus grande. La définition mathématique de l'entropie d'une image est donnée par [Gab] :

$$H = -\sum_{j=0}^n \Pr(a_j) \log(\Pr(a_j)) \quad (2.2)$$

Où $\Pr(a_j)$ est la probabilité d'occurrence du symbole a_j (pixel d'une image dans notre cas).

2.3.2.3. Mesures de distorsion

La distorsion est l'erreur introduite par l'opération de compression, due au fait qu'éventuellement l'image reconstruite n'est pas exactement identique à l'image originale. Les mesures de distorsion ne concernent que les méthodes de compression avec perte d'informations. La distorsion en compression d'image est généralement mesurée par l'erreur quadratique moyenne MSE (Mean Square Error) [Gab] :

$$\text{MSE} = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [f(i, j) - f'(i, j)]^2 \quad (2.3)$$

L'erreur quadratique moyenne Normalisé NMSE (Normalized Mean Square Error) :

$$\text{NMSE} = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [f(i, j) - f'(i, j)]^2}{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [f(i, j)]^2} \quad (2.4)$$

Le PSNR (Peak Signal-to-Noise-Ratio) Rapport signal sur bruit crête :

$$\text{PSNR} = 20 \cdot \log \left(\frac{255}{\sqrt{\text{MSE}}} \right) \quad (2.5)$$

Où les images ont $N \times M$ pixels, et $f(i, j)$ représente l'image originale, et $f'(i, j)$ représente l'image reconstruite.

2.3.3. Redondances [Gab]

Les images en niveaux de gris ont trois types de redondances:

Redondance inter-pixels : elle concerne les niveaux de couleurs que se répètent dans une image donnée et la corrélation qui existe entre les pixels adjacents. On peut donc représenter une image en n'utilisant que la différence qu'il y a entre un pixel et un autre.

Redondance de codes (symboles) : une redondance apparaît dans le code d'une image si ses niveaux de couleurs sont codés d'une façon qui utilise plus de symboles que strictement nécessaire.

Redondance psycho-visuelle : la visualisation d'une image ne requiert pas l'analyse quantitative de chaque pixel. Seuls certains éléments clés, comme par exemple les lignes de changement de couleur vont permettre au cerveau de reconstituer l'image. Les autres informations sont dites psycho-visuellement redondantes. Elles peuvent donc être éliminées sans affecter de façon significative la lisibilité de l'image. L'élimination de ces redondances introduit une perte quantitative d'informations : c'est la quantification. Cette opération est irréversible et conduit à une compression avec perte.

Les images couleurs ont en plus des redondances citées : la redondance du spectre. Les images vidéo ont plus des redondances citées la redondance temporelle.

2.4. Modèle de compression

Le modèle de compression théorique d'une image individuelle est illustré dans la figure 2.3. Chaque algorithme ou méthode de compression suit ce modèle. L'étape de «Quantification» est propre aux méthodes avec perte d'information. En général, la réduction de données est réalisée par la diminution (idéalement l'élimination) des différents types de redondances. Le mappage de pixels réduit la redondance inter-pixels. La quantification réduit la redondance psycho-visuelle. L'étape finale de l'encodage dans le modèle diminue la redondance de codes [Ber].

Les étapes de quantification et mappage de pixels réduisent l'entropie de l'image de sorte que l'étape de l'encodage puisse réaliser une meilleure compression [Kar].

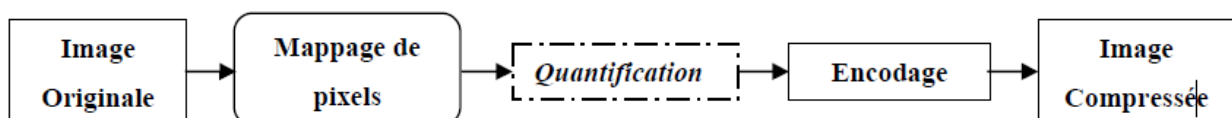


Figure 2.3 : Modèle de base de compression d'une image individuelle.

2.5. Méthodes de compression

Il existe deux méthodes de compression : les méthodes avec perte et les méthodes sans perte d'informations. Cette classification s'intéresse à la présence ou non d'une perte d'information introduite par l'opération de compression [Huf].

2.5.1. Méthodes sans perte d'information (Réversible) [Gab]

Les méthodes sans perte d'informations s'appliquent en général dans le domaine spatial. Nous décrivons les algorithmes les plus connus ci-après.

2.5.1.1. Codage entropique

Le but du codage entropique est de s'approcher le plus possible de l'entropie de la séquence de symboles en affectant les codes les plus courts possibles aux symboles de probabilité élevée et vice versa. Ces systèmes sont conçus de manière à ce que les codes résultants, bien de longueur variable, puissent être décodés de façon unique.

La première méthode de ce type a été l'algorithme de Shannon-Fano, qui est basée sur la simple connaissance de la probabilité d'occurrence de chaque symbole dans un message.

L'algorithme construit une table de codes irréductibles comme suit :

- 1- Les probabilités d'occurrence de chaque message sont placées dans une liste dans un ordre décroissant. La liste constitue la racine d'un arbre qui pour l'instant est une feuille.
- 2- Couper la liste en deux groupes de symboles S_0 et S_1 dont les probabilités totales sont aussi voisines que possible.
- 3- Le groupe S_0 est codé par un 0, le groupe S_1 par un 1.
- 4- Si un groupe S_i n'a qu'un seul élément, il est appelé « feuille terminale » et est inchangée. Sinon, la procédure reprend à l'étape 2 sur le groupe S_i .

La procédure de codage construit un arbre dont les suites de bits 1 ou 0 partant de la racine vers chacune des feuilles constituent les mots code du code.

Le codage de Huffman a rapidement remplacé le codage de Shannon-Fano car il est plus efficace. Le codage de Huffman a la particularité de produire des codes de préfixe unique, ce qui permet de les décoder sans ambiguïté. La façon de construire les codes binaires est différente de Shannon-Fano. On classe aussi par ordre de probabilité décroissante les

symboles. A l'inverse de la technique précédente, on part des deux derniers symboles dont la probabilité est la plus faible, et on les groupe en une partie à laquelle on affecte la probabilité résultante. On reclasse l'ensemble de probabilités obtenues et on regroupe de nouveau les parties de plus faible probabilité. On construit ainsi un arbre jusqu'à ce qu'il ne reste plus que deux probabilités. On part de celles-ci pour construire les codes binaires des symboles. La table des codes doit être transmise avec les symboles codés, afin de l'utiliser pour le décodage [Huf].

2.5.1.2. Méthodes différentielles et prédictives

Ces méthodes exploitent la redondance entre un symbole et ses voisins, qui en général se ressemblent beaucoup. Par exemple, on code le premier symbole, on calcule la différence avec le second symbole et on code cette différence. Celle dernière nécessite moins de bits que les symboles eux-mêmes car cette différence est souvent faible. On code ensuite la différence entre le deuxième symbole et le troisième etc.

Dans des systèmes plus complexes et performants, on établit une fonction de prédiction qui permet d'estimer la valeur d'un symbole en fonction de la valeur des symboles voisins. On code alors l'erreur de prédiction. Cette erreur est l'écart entre la vraie valeur du symbole et la valeur prédite. La fonction de prédiction peut être plus ou moins complexe selon l'ordre de prédiction (nombre de symboles impliqués dans le calcul de prédiction), la topologie (position des symboles voisins utilisés dans le calcul) et l'utilisation de pondération (poids affectés aux symboles du calcul de prédiction selon leur position relative par rapport au symbole prédit ou selon les propriétés statistiques de l'ensemble de symboles). La façon de coder les erreurs de prédiction est souvent basée sur la quantification scalaire, que nous verrons plus loin. Le codage prédictif ainsi décrit correspond à la modulation par impulsion et codage différentiel (MICD, ou DPCM en anglais) [One].

2.5.1.3. Codage par plage (Run Length Encoding)

Il s'agit d'une plage est une succession de symboles ayant la même valeur. Pour chaque plage rencontrée dans la séquence, on code sa valeur et le nombre de symboles qu'elle comprend. Pour que cette méthode soit utile, il faut que la longueur moyenne des plages soit suffisamment élevée. Le Run Length Encoding (RLE) est utilisé par les fax dont les images sont binaires [Ber].

2.5.1.4. Codage arithmétique

Le codage arithmétique est un codage utilisant un modèle statistique comme le codeur de Huffman. Mais, contrairement à ce dernier, il produit un code pour une séquence de symboles tout entière et non pas un code par symbole. Chaque nouveau symbole lu modifie de façon incrémentale le code de sortie. Ce code de sortie est un nombre à virgule flottante compris entre 0 et 1, dont le nombre de chiffres après la virgule correspond au nombre de symboles. Contrairement à Huffman, il n'est pas obligatoire que chaque code ait un nombre entier de bits. Par exemple un symbole de probabilité 0.9 a pour entropie 0.15, mais Huffman affectera probablement un code de un bit (ou plus) et la séquence codée aura un nombre de bits plus long qu'en théorie. Le codeur arithmétique est plus performant que le codeur de Huffman, mais il est plus complexe à implémenter [Ber].

2.5.1.5. Méthodes par dictionnaire adaptatif

Dans les méthodes par dictionnaire adaptatif, le codeur lit un groupe de symboles et recherche des équivalences avec des chaînes de symboles rencontrées dans un ensemble de symboles précédents. Le premier algorithme de ce type a été mis au point par Lempel et Ziv en 1977 [Ziv77] et a donné lieu au programme LZ77. Celui-ci lit un flot de symboles et cherche des chaînes équivalentes dans une fenêtre de 4Ko précédant le flot d'entrée. Les équivalences sont remplacées par des codes. Lempel et Ziv ont développé une nouvelle version de leur algorithme en 1978 dénommée LZ78 où le dictionnaire est construit à partir de tous les symboles précédemment rencontrés et non par une fenêtre [Ziv78]. Le succès de ces méthodes s'explique par la rapidité des calculs.

2.5.2. Méthodes avec perte d'informations (Irréversible) [Gab]

2.5.2.1. Sous-échantillonnage

Le sous-échantillonnage consiste à ne conserver qu'une partie des données. Par exemple, si on ne stocke qu'un pixel de l'image sur deux, on obtient un taux de compression de 4:1 (4 bits représentés par 1 bit). L'image reconstruite s'obtient par interpolation, par exemple en remplaçant chaque pixel manquant par la moyenne de deux pixels adjacents. Cette méthode extrêmement simple est à employer avec précaution car la distorsion n'est pas contrôlée.

2.5.2.2. Quantification scalaire

La quantification scalaire consiste à remplacer un nombre très grand de symboles par un nombre restreint de codes. C'est une opération irréversible très largement employée en compression. Par exemple, le fait d'arrondir des nombres flottants à l'entier le plus proche est une quantification scalaire. En général, un quantificateur scalaire est une fonction en escalier. L'intervalle de toutes les valeurs possibles de symboles d'entrée est divisé en n intervalles adjacents appelés intervalles de décision. Chaque symbole appartient à l'un de ces intervalles et sa valeur codée correspond au numéro de son intervalle. Tous les symboles d'entrée appartenant à un même intervalle sont codés et reconstruits avec la même valeur. La valeur reconstruite est déduite directement des bornes de l'intervalle d'entrée. Pour cela on essaie d'adapter le mieux possible la position des bornes des intervalles de décision et la règle de reconstruction aux propriétés de la séquence de symboles. Tout le problème consiste à minimiser la distorsion introduite par la quantification tout en restreignant au maximum le nombre de codes.

2.5.2.3 Quantification vectorielle

La quantification vectorielle découpe la matrice de symboles (pixels ou coefficients du plan des fréquences) en petits blocs. Chaque bloc est codé par son indice dans un dictionnaire de blocs types. Le codage consiste à repérer dans le dictionnaire des mots-codes disponibles à celui qui ressemble le plus au bloc d'entrée. Le décodage consiste simplement à remplacer l'indice par le mot-code correspondant dans le dictionnaire.

La création du dictionnaire de bloc types est une étape préalable et cruciale, il est construit par apprentissage à l'aide d'un ensemble d'images type à compresser, ou à l'aide d'un treillis qui utilise les propriétés de la distribution statistique des symboles à coder. Un dictionnaire n'est performant que pour la classe d'image avec laquelle il a été mis au point [Gab].

2.5.2.4. Méthodes par transformation [Gab]

Les méthodes par transformation sont les techniques de compression les plus utilisées. Elles offrent des taux de compression élevés tout en conservant une bonne qualité de l'image. Ces méthodes font appel successivement à plusieurs principes de compression. Elles constituent des standards internationaux pour le codage des images fixes et de la vidéo. La compression par transformation consiste à décomposer les pixels fortement corrélés de

l'image en ensembles de coefficients spectraux partiellement décorrélés avec de l'énergie concentrée dans un nombre restreint de coefficients. Ce compactage de l'énergie permet d'assigner aux coefficients spectraux les plus énergétiques un nombre de bits plus élevé par rapport aux autres.

2.5.2.5. Codage en sous-bandes

Dans le codage en sous-bandes, l'image d'entrée est décomposée en un certain nombre de bandes par filtrage passe-bas passe-haut. Le résultat est un nombre de sous-images avec des bandes passantes plus étroites que l'image d'origine et par conséquent elles sont compressées plus efficacement [Gab].

2.5.2.6. Codages imbriqués

La transformation en ondelettes génère des sous-bandes qui correspondent à des projections orthogonales dans des espaces vectoriels disjoints. Il subsiste des ressemblances structurelles dans les sous-bandes détails dans la même direction à des résolutions successives (figure 2.4). Le principe de l'arbre de zéro proposé par Shapiro en 1993, permet d'exploiter cette ressemblance.

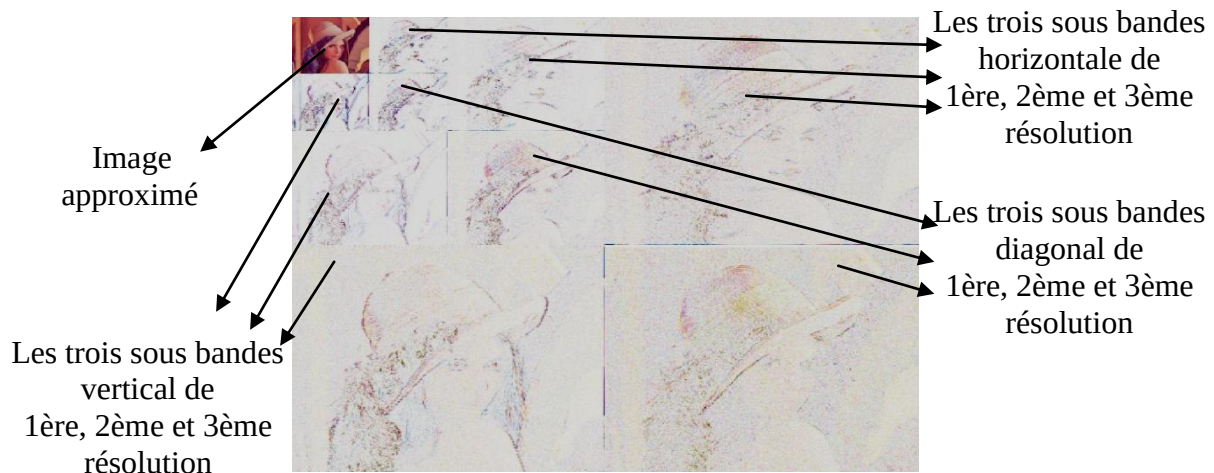


Figure 2.4 : Exemple de transformation en sous-bande de l'image Lena.

2.6. Techniques de compression

2.6.1. Compression JPEG standard

JPEG est l'acronyme de Joint Photographic Experts Group. Il s'agit d'un comité d'experts qui édite des normes de compression pour l'image fixe. La norme communément appelée JPEG, de son vrai nom ISO/CEI 10918-1 UIT-T Recommendation T.81, est le résultat de l'évolution de travaux qui ont débuté dans les années 1978 à 1980 avec les premiers essais en laboratoire de compression d'images [Web1].

Le processus de compression et de décompression JPEG irréversibles comporte six étapes principales représentées dans la figure 2.5 [Web1]:

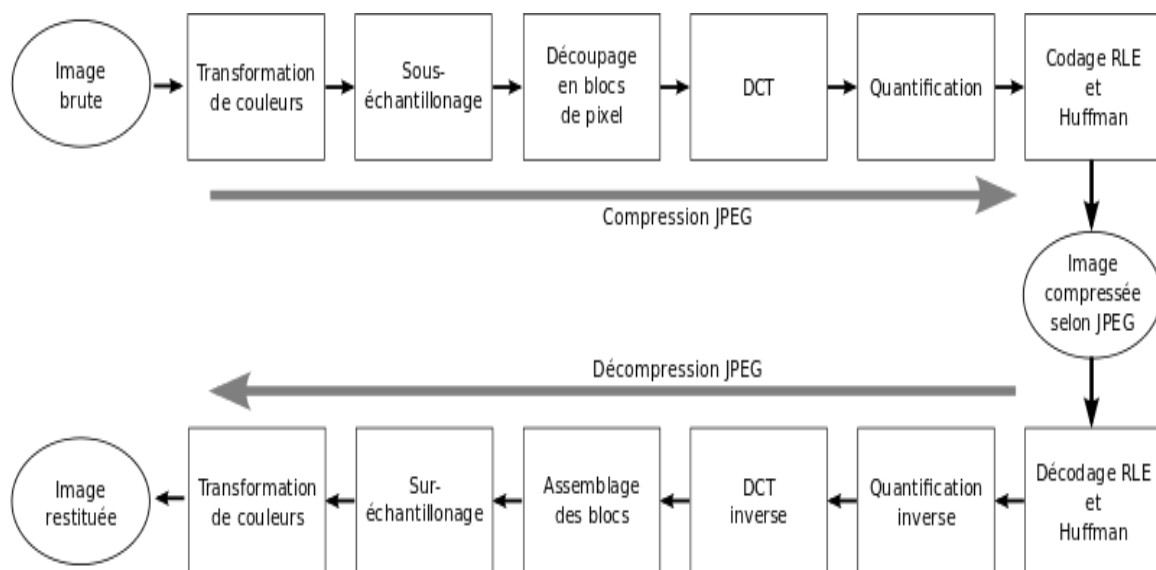


Figure 2.5 : Etapes de compression et de décompression JPEG irréversibles.

2.6.2. Compression JPEG-LS

JPEG-LS (souvent surnommé Lossless JPEG) est une norme de compression sans perte (donc réversible), basée sur l'algorithme LOCO-I (Low Complexity Lossless Compression for Images) et évaluée par le Joint Photographic Experts Group [Web2].

Dans JPEG-LS la compression est réalisée par la combinaison d'un codage adaptatif (extension des codes de Golomb) avec un codeur entropique proche du codeur de Huffman pour les zones à faible entropie. JPEG-LS a été publiée comme norme internationale par l'ISO (ISO/CEI 14495-1) et l'UIT (T.87) [Web2]. Les étapes de compression réversibles JPEG-LS sont représentées dans la figure 2.6.

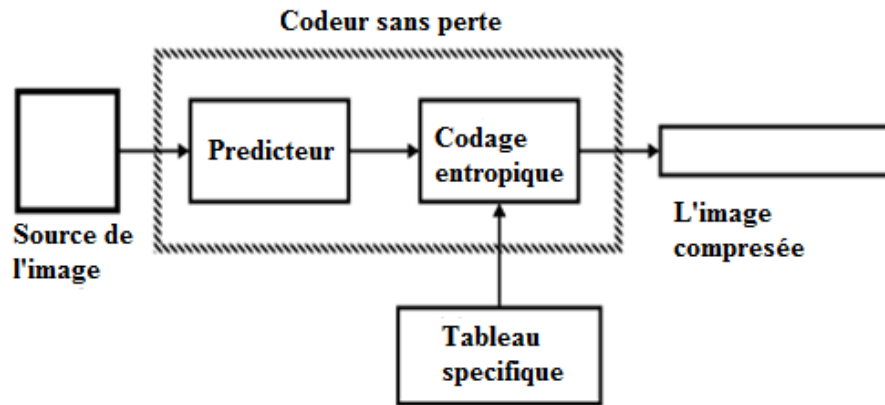


Figure 2.6 : Etapes de compression réversibles JPEG-LS [Web2].

2.6.3. Compression JPEG 2000

JPEG 2000 ou ISO/CEI 15444-1 est une norme de compression d'images produite par le groupe de travail Joint Photographic Experts Group dirigé par Hervé Grelet. JPEG 2000 est capable de travailler avec ou sans perte en utilisant une transformée en ondelettes (méthode d'analyse mathématique du signal). Les performances de JPEG 2000 en compression avec et sans perte sont supérieures à celle de la méthode de compression JPEG ISO/CEI 10918-1 (JPEG baseline). On obtient donc des fichiers d'un poids inférieur pour une qualité d'image égale. De plus, les contours nets et contrastés sont mieux rendus [Web3].

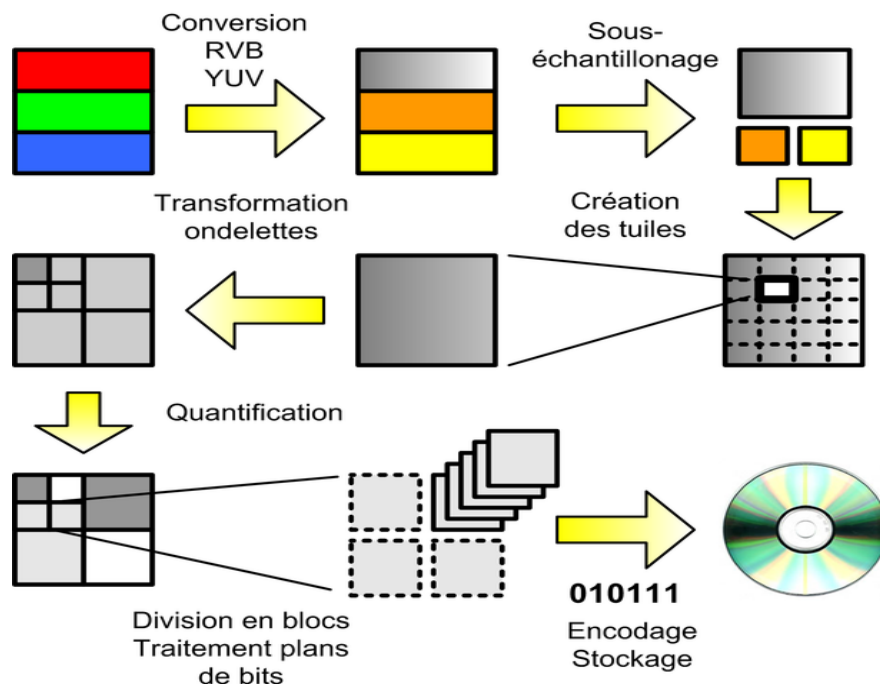


Figure 2.7 : Etapes de compression JPEG 2000 [Web3].

2.6.4. Compression JPEG XR

Une nouvelle norme de compression d'image désignée sous le nom JPEG extended Range (XR JPEG) a été mise au point. JPEG XR (UIT-T T.832 | ISO / CEI 29199-2) s'adresse principalement à accroître les capacités de sortie des techniques de codage et fournit de hautes performances à faible coût de calcul. Les étapes de compression JPEG XR sont presque les mêmes au niveau supérieur par rapport aux normes de compression existantes mais les opérations de niveau inférieur sont différentes telles que les opérations de transformation, de quantification, de numérisation et les techniques d'entropie de codage. Il prend en charge la compression sans perte et celle avec perte. Les étapes de la compression JPEG XR sont présentées dans la figure 2.8 [Sun].

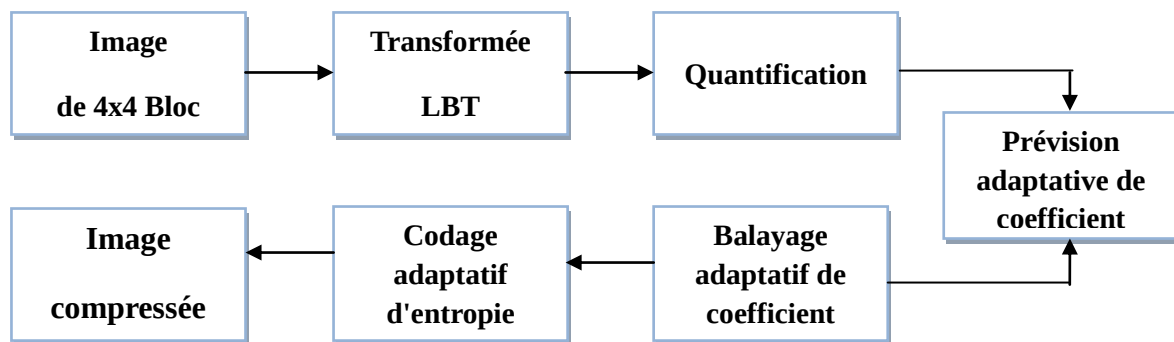


Figure 2.8 : Etapes de Compression JPEG-XR.

JPEG XR utilise Lapped Biorthogonal Transform (LBT) pour convertir des échantillons d'image en domaine fréquentiel. LBT est une transformée entière et elle est moins coûteuse en calcul que la DWT utilisée dans JPEG2000. Elle réduit les artefacts de blocage à des débits plus faibles par rapport à JPEG. Elle améliore de manière significative l'exécution globale de la compression de JPEG XR. L'exécution basée par logiciel est généralement employée pour le traitement en différé et conçue pour implémentation sur des processeurs d'usage universel [Sun].

2.6.5. Compression EZW

L'algorithme EZW peut être résumé en trois étapes, comme suit :

- ✓ La définition des cartes de signifiante indiquant les positions des coefficients significatifs par rapport à un seuil donné.

- ✓ Une approximation successive, par passes, des coefficients significatifs, qui permet donc une notion de progressivité du codage selon un critère d'arrêt de débit-distorsion.
- ✓ Un codeur arithmétique dynamique de la chaîne de symboles.

Le codeur EZW offre la propriété de la transmission progressive de l'image codée tout en apportant d'excellentes performances débit/distorsion par rapport à la norme JPEG. Le schéma bloc d'un tel codeur est donné dans la figure 2.9.

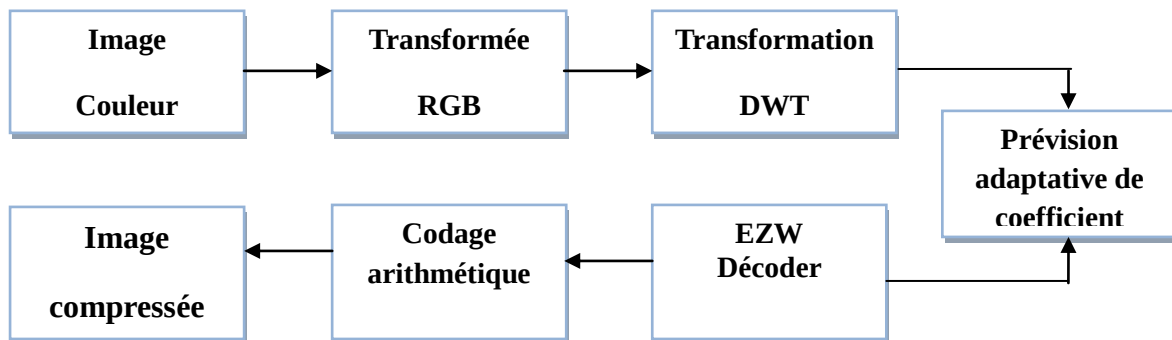


Figure 2.9 : Etapes de Compression EZW.

2.7. Conclusion

Dans ce chapitre, nous avons présenté des concepts élémentaires et des notions générales de traitements de données et des images. Nous avons commencé par les avantages et les inconvénients de la compression de données et nous avons présenté les méthodes de compression classiques et modernes.