HAL
open science

Optimisation d'un schéma de codage d'image à base d'une TCD. Application à un codeur JPEG pour l'enregistrement numérique à bas débit

Moussa Ammar

To cite this version:

Moussa Ammar. Optimisation d'un schéma de codage d'image à base d'une TCD. Application à un codeur JPEG pour l'enregistrement numérique à bas débit. Traitement du signal et de l'image [eess.SP]. Télécom ParisTech, 2002. Français. NNT : . tel-00005739

HAL Id: tel-00005739

https://pastel.hal.science/tel-00005739v1

Submitted on 5 Apr 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse

Présentée pour obtenir le grade de Docteur de
l'Ecole Nationale Supérieure des télécommunications
Spécialité : Signal et Images

Moussa AMMAR

Optimisation D'un Schéma De Codage D'image À Base D'une TCD. Application À Un Codeur JPEG Pour L'enregistrement Numérique À Bas Débit

Soutenue le 14 Janvier 2002 devant le jury composé de

Michel Barlaud	Président
Rémy PROST	Rapporteur
Béatrice PESQUET	Examinateur
Jacques PRADO	Examinateur
Pierre DUHAMEL	Directeur de Thèse

Lis, au nom de ton Seigneur qui a créé,
qui a créé l'homme d'une adhérence.
Lis! Ton Seigneur est le Très Noble.
qui a enseigné par la plume,
a enseigné à l'homme ce qu'il ne savait pas.
Prenez-garde! Vraiment l'homme devient rebelle,
dès qu'il estime qu'il peut se suffire à lui-même.
Mais, c'est vers ton Seigneur qu'est le retour.

Le Saint Coran

Table de matières

1 Introduction Générale **1**
1.1 La Compression d'Image, Pourquoi? 1
1.2 Mesure de la qualité d'image compressée 2
1.3 Objet de l'étude 3

I Optimisation du schéma de codage d'image **5**

2 Schéma de codage source. Application à l'image **7**
2.1 La compression d'image 7
2.2 Structure d'un Codeur/Décodeur 7
2.3 Les Transformées et les Sous-Bandes 8
2.3.1 Les Transformations Linéaires 9
2.3.2 Les Bancs de Filtres 13
2.4 La Quantification 16
2.4.1 La Quantification Scalaire 17
2.4.2 La Quantification Vectorielle 22
2.5 L'Allocation de Bits 23
2.6 Le Codage par plage (Run length Encoding) 27
2.6.1 Le Procédé du Codage par plage 27
2.7 L'Entropie 28
2.7.1 Définitions 28
2.7.2 Le Codage Entropique 30
2.8 Exemples de Codeurs 34
2.8.1 JPEG: Norme de Compression d'Image Fixe 34
2.8.2 MPEG et la Compression des Séquences d'Images Animées . . 37
2.8.3 H261 et H263 Normes pour le Codage Vidéo 38

3 Codeur JPEG à base d'une transformée TCD **41**
3.1 La TCD une Transformée linéaire orthogonale 41
3.2 La Compression Sans Perte en Absence de Toute Quantification . . . 43
3.3 La Compression Avec Perte en Présence de Bruit de Quantification . 44

3.3.1 Formulation 44
3.4 TCD^{-1}, Transformée de Synthèse Optimale pour la Compression avec Perte? 45
3.5 Présentation d'une Transformée de Synthèse Optimale 46
3.5.1 Les Filtres de Wiener 46
3.5.2 Performances de la MMSE par rapport à la TCD classique . . 49
3.5.3 Conclusions 49

4 Les Quantificateurs optimaux et l'Allocation de bits 55
4.1 Travaux Liés au Problème d'Allocation de Bits pour JPEG 55
4.2 Problème de Minimisation Sous Contrainte - Problème Sans Contrainte 56
4.2.1 Le Multiplicateur de Lagrange, Fonction du Coût à Minimiser 57
4.2.2 Une Description Graphique 58
4.3 Allocation de Bits avec les Filtres de Wiener à la Synthèse 61
4.3.1 Le Problème de Non-Orthogonalité de la MMSE avec la TCD 61
4.3.2 Une Distorsion Additive ou Non ? 61
4.4 La Convexité des courbes Débit/Distorsion. Solution sous-Optimale . 63
4.5 Conclusion 64

5 l'Optimisation d'un schéma de codage d'image à base d'une TCD 67
5.1 L'Optimisation Conjointe des Quantificateurs et de la Transformée de Synthèse 67
5.2 L'Algorithme d'Optimisation itératif A1 68
5.2.1 Comment Compléter les tables de Huffman? 69
5.3 Introduction de la Quantification Variable et des Facteurs d'échelle . 72
5.3.1 L'Annexe C du JPEG (Quantification Variable) 72
5.4 Le Calcul des facteurs d'échelle 73
5.4.1 Cas avec TCD (JPEG-3): Calcul simple 73
5.4.2 Cas avec MMSE 73
5.5 L'Algorithme itératif A2 73
5.6 Les Résultats de codage par le nouveau schéma optimisé 74
5.6.1 Comparaison du JPEG avec les différentes étapes des algorithmes A1 et A2 74
5.7 Conclusion 75

II Post-traitement à la reconstruction, Réduction des effets de blocs 87

6 Une nouvelle technique de post-traitement de l'image 89
6.1 Introduction et motivation 89
6.2 Travaux antérieurs liés au post-traitement 90
6.3 La Réduction des effets de blocs par minimisation de l'énergie des HF 91

6.3.1 Les Effets de blocs et Les Hautes Fréquences 91
6.3.2 Formulations 93
6.3.3 L'Algorithme B1 de réduction des effets de blocs dans un schéma de codage par transformée 95
6.4 Résultats d'application de l'algorithme B1 96
6.4.1 Comparaison de l'Algorithme SPIHT avec A1 + B1 98
6.5 Conclusion 99

Conclusion Générale et Perspectives **129**

Liste des Figures **133**

Liste des Tables **135**

Glossaire **137**

Bibliographie **143**

Remerciements

Je tiens en tout premier lieu à remercier chaleureusement le Professeur Pierre DUHAMEL, pour m'avoir accordé l'honneur de dirriger cette thèse qui s'est déroulée au sein du département TSI de l'ENST - Paris, dont il était chef du département. Pierre, merci pour m'avoir indiqué souvent le bon chemin, pour ton soutien moral et scientifique, pour ta patience et ta gentillesse, et enfin pour m'avoir supporté le long des années de cette thèse malgré tous les hauts et les bas.

Je tiens ensuite à remercier Jacques PRADO pour sa co-direction de cette thèse avec Pierre. Jacques, tu étais toujours à mes côtés face à tous les problèmes surtout les techniques. Merci pour tout le support et tout le temps que tu m'as réservés afin de faire avancer et aboutir les travaux de cette thèse.

Je tiens à exprimer toute ma gratitude envers Michel BARLAUD, Professeur à l'Université de Nice-Sophia Antipolis pour avoir accepté de présider le jury de ma soutenance de doctorat et pour son rôle de rapporteur. Tout particulièrement, j'apprécie énormement ses commentaires judicieux et sa lecture minutieuse de mon manuscript.

Je remercie enfin chaleureusement les autres membres du jury qui ont accepté de participer à cette soutenance: Rémy PROST, Professeur des Universités à INSA-Lyon, pour son rôle de rapporteur, et Béatrice PESQUET, Maître de Conférences à l'ENST-Paris, pour son rôle d'Examinateur et sa relecture consciencieuse de mon rapport.

Merci aussi à tous les membres du département TSI à l'ENST-Paris, malgré mes rares présences à l'Ecole.

Merci à mon épouse Joumana sans qui cette thèse ne serait pas.

Enfin, je dois ma profonde reconnaissance à mes parents au Liban, qui trouvent dans cette thèse l'aboutissement de longues années d'efforts et de soutien.

Chapitre 1

Introduction Générale

1.1 La Compression d'Image, Pourquoi?

Récemment, on a remarqué une croissance en terme de besoins de stockage et/ou de transmission des informations visuelles. Les diverses applications telles que la télécopie, la vidéo-conférence, l'imagerie médicale et satellitaires, la télévision haute définition, la télésurveillance et les services d'informations sur l'internet sont basées sur la fiabilité de sauvegarder et transmettre les images. Le stockage des images sur les disques durs des ordinateurs grand public ainsi que l'utilisation des technologies numériques pour le traitement et la retouche des images necessitent d'acquérir les images sous format numérique (encore appelé format électronique).
Le format numérique des images est extrêmement coûteux en taille mémoire, bien qu'il soit le plus adapté aux applications citées plus haut. Pour resoudre ce problème de coût qui peut limiter la faisabilité de stockage et de transmission des images, des techniques de compression d'images ont été élaborées pour compacter leur représentation numérique. A l'aide de ces techniques de compression, le stockage et la transmission des images seront plus efficaces et plus rapides.
Les techniques de compression se divisent en deux catégories principales "Compression sans perte" et "Compression avec perte". La "Compression sans perte" consiste à enlever la redondance dans les données juste nécessaire pour représenter l'image. Cette redondance est directement liée à la prédictibilité des éléments constituants de l'image. Par exemple, une image de couleur unie est totalement redondante du fait que la couleur fournit suffisamment d'informations pour représenter toute l'image. La "Compression sans perte" identifie les éléments constituants de l'image et exploite leur structure pour réduire la quantité des données. Une élimination simple de la redondance par la "Compression sans perte" ne fournit pas une représentation suffisament compacte pour plusieurs applications. Ainsi, pour obtenir une compres-

sion plus élevée, quelques informations seront négligées par la "Compression avec perte".
Les améliorations apportées par la "Compression avec perte" ne sont pas tout simplement dûes à l'élimination des données redondantes, mais plutôt l'abandon des informations estimées non pertinentes, comme par exemple les informations sur des détails non perceptibles facilement à l'œil nu. Ainsi la "Compression avec perte" fournit une représentation compacte *a priori* indiscernable visuellement de l'image originale, bien qu'en général l'image compressé est tout à fait différente de l'originale.
Cette étude portera sur l'optimisation des normes existantes de compression d'images avec perte, en particulier les normes à base de la Transformée en Cosinus Discrète (TCD) telle que $JPEG$ pour les images fixes et dans une certaine mesure la norme $MPEG$ pour les images animées et $H26x$ pour la transmission vidéo. Les performances de cette optimisation seront mesurées à deux niveaux, *qualitatif* (par une observation visuelle des images codées) et *quantitatif* (à l'aide du Rapport Signal/Bruit Crête $PSNR$). Selon ces critères de performance, les normes $MPEG$, $JPEG$ et $H26x$ ne représentent pas forcément l'état de l'art dans la recherche sur la compression d'images. Les techniques basées sur les ondelettes donnent des résultats sensiblement meilleurs que ceux des normes à base de la TCD. Les avantages des normes $MPEG$, $JPEG$ et $H26x$ sont le temps de codage modéré pour une faible complexité et une robustesse aux erreurs de transmission.
Ces normes resteront longtemps à la base de nombreuses applications commerciales et des outils de compression grand public. Les normes $JPEG$ et $MPEG$ sont souvent utilisées comme repères pour les nouveaux travaux, les comparaisons ne sont pas toujours objectives ce qui rend floues les améliorations apportées par les nouvelles techniques. Nous avons l'intention dans cette étude de fournir des points de repères plus objectifs qui aident à justifier l'optimisation des techniques de compression à base de la TCD. Concernant la norme $JPEG$ il parait que notre travail d'optimisation est plus performant que $JPEG$ sur le plan théorique et expérimental. Pour les normes $MPEG$ et $H26x$, des travaux d'optimisation supplémentaires seront encore nécessaires.

1.2 Mesure de la qualité d'image compressée

Dans un système de compression d'images, l'image compressée est toujours comparée à l'originale pour déterminer son rapport de ressemblance. A côté du critère qualitatif et subjectif (un observateur humain qui déclare deux images identiques), on trouve le critère quantitatif le plus utilisé l'Erreur Quadratique Moyenne "EQM" (Mean Square Error "MSE") ou son équivalent le Rapport Signal/Bruit Crête (Peak Signal to Noise Ratio $PSNR$). L'EQM est définie par

$$MSE = \frac{1}{M \times N} \sum_{m=1}^{M} \sum_{n=1}^{N} [\hat{x}(m,n) - x(m,n)]^2 \tag{1.1}$$

Le $PSNR$ est

$$PSNR = 10log_{10}\frac{255^2}{MSE} \tag{1.2}$$

où $(M \times N)$ est la taille de l'image à coder, $x(m,n)$ et $\hat{x}(m,n)$ sont les intensités des pixels de l'image d'origine et l'image codée respectivement.
En général, une image codée dont l'EQM est faible doit ressembler à l'image originale plus qu'une autre avec une valeur d'EQM plus importante. Or ce n'est pas toujours le cas, du fait que l'observation visuelle des images codées est subjective, elle est différente d'une personne à une autre. Nous adoptons dans cette étude, simultanément les deux critères qualitatif et quantitif pour mesurer les performances du schéma de compression optimisé, bien que les résultats sont souvent donnés en terme de $PSNR$.
Nous définissons, le taux de compression comme le nombre de bits nécessaire pour stocker l'image d'origine divisé par celui nécessaire pour l'image codée et le débit en bit-par-pixel (bpp) comme le nombre moyen de bits nécessaire pour représenter un pixel de l'image.
Avec ces définitions, l'objectif de notre optimisation du système de compression serait la minimisation de la distorsion (l'EQM mesurée sur l'image décodée) sous la contrainte de débit (ou de taux de compression). Tout de même, il est possible aussi de minimiser le taux de compression pour un certain niveau de distorsion (une qualité de l'image décodée). Ces deux problèmes equivalents entrent dans le cadre de l'analyse débit/distorsion qui consiste à l'optimisation d'un système de "Compression avec perte" en termes de minimisation sous contraintes.

1.3 Objet de l'étude

Pratiquement, cette étude traite deux problèmes. Premièrement, les normes $JPEG$, $MPEG$ et $H26x$ ainsi que d'autres techniques à base de la TCD sont largement utilisées d'une façon *ad hoc*. Les procédures de codage et de décodage ne sont pas exécutées d'une façon à optimiser l'erreur de reconstruction. Ainsi, nous pourrons obtenir des améliorations considérables sur la qualité des images codées si nous analysons et nous optimisons le système de compression dans le cadre d'une étude sur la théorie débit/distorsion. D'ailleurs ces normes sont définies fondamentalement pour les décodeurs, laissant plus de liberté pour l'optimisation du codeur tout en respectant certaines règles de syntaxe du flux de données.
Dans ce contexte, cette thèse traite le problème d'optimisation de la compression à deux niveaux:

1. Chercher la meilleure allocation de bits (optimisation débit/distorsion);

2. Calculer une transformée de synthèse plus adequate que la TCD pour reconstruire l'image avec moins de distorsion.

Deuxièmement, à très bas débit le codage par les normes à base de la TCD génère des artefacts visuels dans l'image décodée qui persistent même après l'optimisation du codeur. En particulier, on considère les effets de blocs qui se présentent comme des discontinuités rectangulaires artificielles entre les blocs. Celles-ci sont les résultats du traitement individuel (Transformation et Quantification) des blocs de pixels de l'image originale. Nous proposons dans cette thèse une nouvelle technique pour réduire ces effets de blocs et garantir une meilleure qualité d'image à très bas débits. Pendant que l'optimisation au niveau du codeur cherche le meilleur compromis débit/distorsion, l'optimisation au décodeur (reduction des effets de blocs) ou notre technique de post-traitement n'exige aucune information *a priori* sur le contenu de l'image donc pas de débit supplémentaire, d'où son intérêt.
La première partie de cette thèse est consacrée à l'optimisation du Codeur à base de la TCD alors que la deuxième partie traite le problème de post traitement au décodeur.

Partie I

Optimisation du schéma de codage d'image

Chapitre 2

Schéma de codage source. Application à l'image

2.1 La compression d'image

Il existe 2 types de compression: sans perte et avec perte. Dans ce chapitre nous expliquerons le sens de ces termes ainsi que les détails des algorithmes de compression avec perte où la Transformée en Cosinus Discrète TCD joue un rôle très critique.
La compression est une opération qui consiste à réduire le nombre de bits utilisés pour représenter un ensemble d'informations sans dégrader la qualité de ces informations (compression sans perte) ou pour représenter approximativement ces données (compression avec perte). L'utilisation de la compression (sous ses différents modes) est devenue un facteur économique pour l'archivage ou la transmission d'image surtout avec l'énorme demande de telle technique pour les applications multimédia, audio-visuelle ou autre.

2.2 Structure d'un Codeur/Décodeur

La structure d'un compresseur d'images fixes telle qu'elle sera étudiée dans cette thèse comprend les modules essentiels suivants:

1. La transformée d'analyse directe qui consiste à décorreler les informations dans l'image, les intensités des pixels. Cette opération fait passer les pixels de l'espace d'intensité vers un espace fréquentiel où l'image sera représentée par un ensemble de coefficients plus ou moins décorrelés entre eux. Dans les

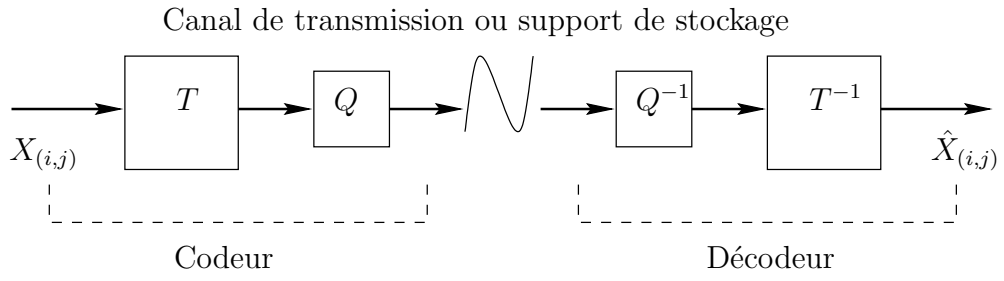

Figure 2.1: Schéma de Codage/Décodage

codeurs classiques, la transformation est normalement réversible et n'entraine en aucun cas une déformation des données. En effet, cette opération ne réduit pas la quantité d'information présente dans l'image (voir le paragraphe 2.3).

2. La quantification qui permet de représenter sur un nombre fini de bits les coefficients transformés. Cette opération entraine en conséquence une perte irréversible de la qualité objective de l'image (voir le paragraphe 2.4).

3. Le codage entropique où les coefficients quantifiés forment un train binaire le plus réduit possible (voir le paragraphe 2.7.2). Cette opération ne provoque aucune distorsion ou perte à l'information contenue dans l'image.

Le décodage ou la décompression est tout simplement le chemin inverse de la compresion, c-à-d décodage entropique, quantification inverse et transformation inverse ou plus généralement transformation de synthèse.
Le fait d'utiliser un canal de transmission ou un support de stockage limité en débit ou en taille nous oblige à contrôler le débit du codeur en réajustant les pas de quantification. Cette allocation de bits sera étudiée dans le paragraphe 2.5.
Comme le codage entropique est complètement réversible (en absence d'erreur de transmission), le schéma du codeur ainsi que celui du décodeur se résument par la figure 2.1.

2.3 Les Transformées et les Sous-Bandes

Le rôle de la transformée dans un schéma de compression est double. D'une part elle permet de décorreler les coefficients dans le domaine de la transformée et de concentrer la puissance du signal sur un nombre réduit de ces coefficients. D'autre part une transformée comme la TCD apporte de l'information sur le contenu "fréquentiel" du signal dans le sens où les coefficients représentent le contenu des bandes de fréquence du signal d'entrée. Cette information est précieuse lorsque l'on veut tenir compte de certaines connaissances que l'on a *a priori* sur le signal, ou sur l'utilisation qui en est faite (comme, par exemple, des caractéristiques perceptuelles).
Un codeur d'images classique utilise une transformée d'analyse (généralement

réversible) pour représenter les données (intensité des pixels) dans un domaine où les opérations de quantification et de codage entropique sont bien simplifiées.
La transformée est très importante en compression d'images du point de vue pratique, où une transformée médiocre peut rendre complexe la mise en œuvre d'un quantificateur et d'un codeur entropique performants. Avec ces atouts, on considère deux propriétés importantes qui distinguent une transformée performante, la décorrelation et la concentration d'énergie .
Dans une image, les valeurs des pixels dépendent les unes des autres, dans le sens où la valeur d'un pixel est fortement liée à celles de ces voisins, d'où l'importance de la décorrelation induite par la transformée qui rend l'optimisation de la quantification et du codage entropique des variables indépendantes plus simple que le cas sans décorrelation. Cependant cette décorrelation correspond à une indépendance au sens d'une statistique de second ordre et non à une indépendance globale. Celà signifie que les quantificateurs et le codage entropique calculés sont sous-optimaux du fait que des dépendances considérables peuvent encore exister entre les coefficients de la transformée.
L'autre critère de choix d'une meilleure transformée est celui de la concentration d'énergie qui a pour effet de représenter les informations dans une image par un minimum de coefficients. Avec ce nombre réduit de coefficients associés à un codage entropique performant (ou peut être optimisé), on pourra réaliser une compression éfficace.

2.3.1 Les Transformations Linéaires

Soit un vecteur $X(n) = [x_n, x_{n+1}, \ldots, x_{n+N-1}]^t$ de N échantillons d'un processus réel aléatoire stationnaire au sens large. Les échantillons sont fortement corrélés et un simple codage pour chacun d'eux est insuffisant. L'idéal est d'appliquer une transformation linéaire de façon à décorreler les coefficients avant de les coder.

Définitions d'une transformation linéaire

A partir du vecteur $X(n)$ (qui représentera un bloc d'image comme nous le verrons par la suite), on peut déterminer un autre vecteur $Y(n) = [y_n, y_{n+1}, \ldots, y_{n+N-1}]^t$ grâce à des combinaisons linéaires différentes telles que :

$$\begin{aligned}
y_n &= a_{0,0}x_n + a_{0,1}x_{n+1} + \cdots + a_{0,N-1}x_{n+N-1} \\
y_{n+1} &= a_{1,0}x_n + a_{1,1}x_{n+1} + \cdots + a_{1,N-1}x_{n+N-1} \\
\vdots & \quad \vdots + \vdots + \cdots + \vdots \\
y_{n+N-1} &= a_{N-1,0}x_n + a_{N-1,1}x_{n+1} + \cdots + a_{N-1,N-1}x_{n+N-1}
\end{aligned}$$

où les $a_{i,j}$ sont des coefficients réels ou complexes de la matrice de transformation A dite linéaire.

$$A = \begin{pmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N-1,0} & a_{N-1,1} & \dots & a_{N-1,N-1} \end{pmatrix}$$

Ainsi

$$Y(n) = AX(n) \tag{2.1}$$

De la même façon, il est peut être possible de reconstruire exactement les échantillons du vecteur X(n) à partir des échantillons transformés Y(n). Ceci est vrai s'il existe une autre transformation linéaire avec sa matrice associée B tel que $X(n) = BY(n)$. Or dans la mesure où la matrice A est inversible, l'inversion de l'équation (2.1) donne $X(n] = A^{-1}Y(n)$ et la matrice B n'est que l'inverse de A.
Dans le cas d'image, cette dernière sera découpée en blocs de $(N \times N)$ pixels et chacun de ces blocs sera traité individuellement (transformé, quantifié, ...). Il est possible comme dans le cas d'un signal monodimensionel de calculer la transformation linéaire d'un bloc de l'image, mais avant de procéder à cette opération, nous allons définir le vecteur colonne correspondant à ce bloc d'image. En parcourant les lignes du bloc de $(N \times N)$ pixels de gauche à droite et de haut en bas, on peut reformer avec ces pixels un vecteur colonne de N^2 échantillons. Dans la suite de cette thèse, un bloc de l'image sera toujours traité comme un vecteur formé dans l'ordre décrit ci-dessus.
Revenons à la transformation linéaire, le nouveau vecteur de N^2 échantillons, subira une transformation linéaire avec une matrice A spécialement conçue (comme on le verra par la suite), et si la matrice A est inversible on pourra reconstruire les échantillons du vecteur des pixels à partir des échantillons transformés.

Pourquoi a-t-on besoin d'une transformation linéaire pour la compression ?

En effet si les échantillons du vecteur transformé sont à valeurs très faibles, on se trouve devant 2 possibilités.
La première où les échantillons transformés mêmes les plus faibles sont transmis ou stockés (selon le besoin ou l'application). Dans ce cas une transformation inverse des échantillons transformés $Y(n)$ peut restituer exactement le vecteur $X(n)$. Ceci s'explique par le fait que les transformées directe et inverse se neutralisent parfaitement. Aucune compression n'est faite à ce stade. Au contraire si la transformée utilisée est à coefficients réels alors les échantillons de $Y(n)$ sont réels et nécessitent davantage de bits pour être codés.
Le second cas, où les échantillons à valeurs les plus faibles ne sont pas transmis ni stockés. Pour la transformée inverse qui opère sur un vecteur d'échantillons complet,

les échantillons qui manquent seront remplacés par des zéros au décodeur. Dans ce cas, une compression a eu lieu. Par conséquence, le vecteur de pixels restitués est différent de l'original $X(n)$. Nous aurions à contrôler cette différence (bruit ou distorsion) qu'il nous faudra quantifier. Nous verrons que l'EQM est la mesure de distorsion qu'on adopte pour notre codeur optimal.
Le problème est donc de trouver une transformée qui permet de garantir l'obtention de très petits coefficients transformés, ainsi que de contrôler la distorsion causée par l'abandon des plus petits [17].

La Transformée de Karhunen_Loève

Soit une image de dimension $N = (H \times W)$ pixels, par la suite on traite individuellement les pixels comme des variables aléatoires. Notre espace de probabilité est formée d'une collection de M images $\mathcal{P} = \{P_1, P_2, \ldots, P_M\}$. L'intensité du $n^{ième}$ pixel $P(n)$, $1 \leq n \leq N$, est une variable aléatoire à valeur non négative pour chacune des images individuelles $P_i \in \mathcal{P}$. Dans le cas d'une image à faible présence des hautes fréquences, les pixels sont fortement corrélés, où la valeur d'un pixel contient des informations sur les valeurs de son voisinage. Ainsi, si on représente la valeur d'un pixel avec suffisamment de données, on peut profiter de cette corrélation pour réduire le coût de transmission des valeurs des pixels voisins. Ceci est réalisable en transformant l'image dans un autre espace de coordonnées décorrélés sur l'ensemble $\mathcal{P}$ et en transmettant les valeurs décorrélées.
La matrice d'autocovariance des pixels dans $\mathcal{P}$,

$$A = (A(i,j))_{i,j=1..N}$$

contient des éléments hors diagonale

$$A(i,j) = \frac{1}{M} \sum_{m=1}^{M} \hat{P}_m(i) \hat{P}_m(j)$$

avec

$$\hat{P}_m(i) = P_m(i) - \frac{1}{M} \sum_{l=1}^{M} P_l(i)$$

La matrice A étant symétrique, elle peut être diagonalisée. Soit T une matrice orthogonale unitaire ($T^tT = TT^t = I$), si T permet de diagonaliser la matrice A tel que TAT^t soit diagonale, T est dite la matrice associée à la transformée de Karhunen_Loève.
Soit $A1 = TAT^t$, si T diagonalise A, $A1$ sera une matrice diagonale représentant la décorrélation entre les différents coefficients. Pour arriver à ce résultat les lignes de T doivent être les vecteurs propres de A où $Av_i = \lambda_i v_i$ avec v_i est le vecteur propre (normalisé) de A associé à la valeur propre λ_i. $A1 = TAT^t = TT^t\Lambda = \Lambda$ avec Λ est la matrice diagonale telle que $\Lambda(i,i) = \lambda_i$. Le nombre de $\Lambda(i,i)$ à valeur positive

représente le nombre de coefficients décorrélés. Cette transformée bien qu'elle soit optimale, elle n'est pas pratique pour une mise en œuvre vu la quantité de calcul qu'elle demande où la complexité peut atteindre $O(N^3)$ dans le cas général.
Parmi les inconvénients majeurs de la Transformation de Karhunen_Loève citons [17]:

1. La connaissance *a priori* des images à coder;

2. La matrice est généralement non séparable, d'où l'absence d'algorithme rapide et l'impossibilité de prévoir une double transformation monodimensionnelle équivalente;

3. Le calcul de la matrice d'autocovariance A, indispensable pour le calcul de la matrice de Karhunen_Loève, s'appuie sur des blocs dont les statistiques doivent être préalablement connues. Or, une modélisation rigoureuse des blocs d'images est impossible [28];

4. à cause de la précision finie des calculateurs, il arrive que la diagonalisation, qu'il est nécessaire d'effectuer sur A, pose des probèmes;

5. à chaque collection d'images correspond une transformation différente.

Il est vrai que des solutions existent pour pallier ces inconvénients mais elles font appel à des compromis et s'écartent ainsi de la solution optimale.

La Transformée en Cosinus Discrète TCD

Nous traitons directement le cas de la TCD à $2D$ ou la $TCDB$, comme elle est utilisée dans les codeurs d'images fixes $JPEG$ ou animées $MPEG$ et $H26x$.
Tout comme la Transformée de Fourrier Discrète TFD, la TCD est facilement implantée (matérielle ou logicielle). Elle est linéaire, unitaire avec conservation d'énergie, et décompose un signal en composantes fréquentielles différentes. La TCD manifeste une concentration d'énergie et une décorrélation excellentes.
La caractéristique de concentration d'énergie provient du fait que l'application de la TCD à un bloc d'image, où une grande partie de l'énergie se trouve dans les quelques composantes de basses fréquences, donne des coefficients à valeurs significatives dans la région de basses fréquences et des coefficients négligeables pour les hautes fréquences, sauf dans le cas où l'image contient beaucoup de détails qui augmentent l'énergie des hautes fréquences.
Alors que d'autres transformées réalisent une décomposition fréquentielle, elles ne possèdent pas cette propriété de concentration d'énergie. En effet, pour un processus aléatoire markovien de type 1 ayant une corrélation proche de 1, la TCD se rapproche le plus de la transformée KLT qui est optimale dans le sens de la concentration d'énergie. La KLT fournit des coefficients décorrélés mais malheureusement

comme elle dépend du signal et est difficile à implanter (logiciel et matériel), la TCD est plus intéressante, facile à mettre en œuvre et est adoptée pour la compression d'image.
En termes d'équations mathématiques, les transformées bidimensionnelles directe $TCDBD$ et inverse $TCDB^{-1}$ utilisées s'expriment respectivement par:

$$X(u,v) = \frac{2}{M}\alpha(u)\alpha(v)\sum_{m=0}^{M-1}\sum_{n=0}^{M-1} x(m,n)\cos(\frac{\pi u(2m+1)}{2M})\cos(\frac{\pi v(2n+1)}{2M})$$

et

$$x(m,n) = \frac{2}{M}\sum_{u=0}^{M-1}\sum_{v=0}^{M-1} \alpha(u)\alpha(v)X(u,v)\cos(\frac{\pi u(2m+1)}{2M})\cos(\frac{\pi v(2n+1)}{2M})$$

avec

$$\alpha(i) = \begin{cases} \frac{1}{\sqrt{2}} & i = 0 \\ 1 & i \neq 0 \end{cases}$$

Un signal transformé par la $TCDBD$ est projeté sur une base de fonctions $Cosinus$ à différentes fréquences horizontales et verticales. Ainsi chaque coefficient $X(u,v)$ représente dans l'échantillon $x(m,n)$ le degrés de présence de la fonction $Cosinus$ aux fréquences u, v dans les directions horizontale et vericale respectivement. Par définition, le coefficient $X(0,0)$ qui représente la valeur moyenne des pixels dans chaque bloc, est appelé DC alors que les 63 coefficients $X(u,v)$ qui restent sont nommés AC.
En compression d'image, les transformées $TCDB$ adoptées s'appliquent sur des blocs de dimensions (8×8) ou (16×16). Au delà de ces dimensions les avantages de la $TCDB$ sont minimes [22].

2.3.2 Les Bancs de Filtres

Considérons le schéma d'un banc de filtres à M sous-bandes composé de M filtres d'analyse et M autres de synthèse, donné ci-dessous. On note $H_k(z)$ et $F_k(z)$ les fonctions de transfert correspondantes. Les filtres utilisés sont à Réponse Impulsionnelle Finie (RIF), à coefficients réels, de même ordre N aussi bien à l'analyse qu'à la synthèse. Le banc de filtres tel qu'il est représenté (figure 2.2) est dit uniforme à sous-échantillonnage critique, comme les filtres d'analyse $H_k(z)$ pour $0 \leq k < M$ sont suivis par une opération de sous-échantillonnage d'un facteur M où M est le nombre de sous-bandes. A la synthèse on sur-échantillonne, interpole et on somme les coefficients.

Le Banc de Filtres et la Transformée. Equivalence

Donnons l'expression des signaux dans les sous-bandes. On a

$$y_k(m) = u_k(mM) = \sum_{l=-\infty}^{+\infty} h_k(l)x(mM-l)$$

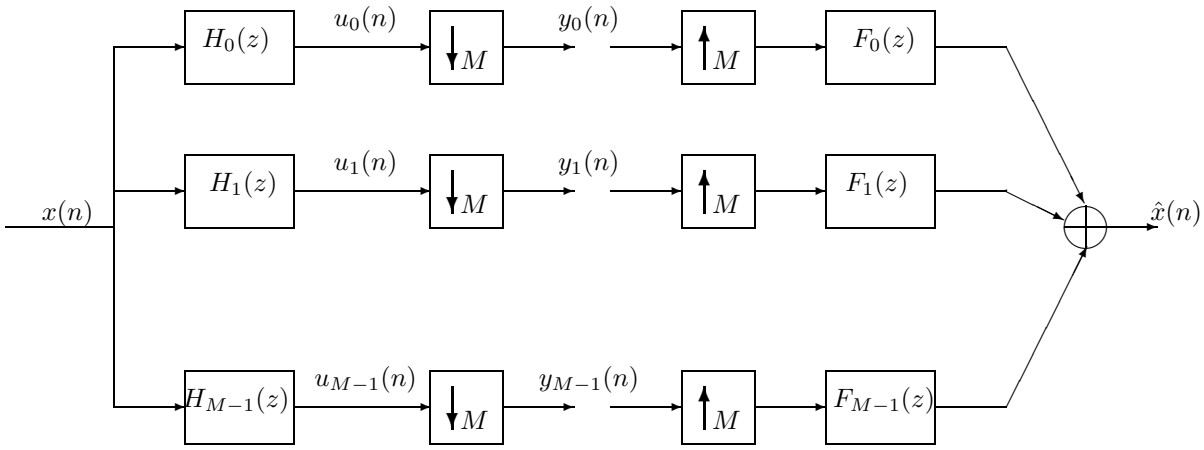

Figure 2.2: Banc de filtres avec sous-échantillonnage et sur-échantillonnage critiques

Comme les filtres du banc ont des réponses impulsionnelles finies de même longueur N, on obtient

$$y_k(m) = \sum_{l=0}^{N-1} h_k(l)x(mM - l) = \sum_{l=0}^{N-1} h_k(l)x_l(m)$$

en mettant en évidence la décomposition polyphasée de $x(n)$. Appelons

$$X(m) = [x_0(m), x_1(m), \ldots, x_{N-1}(m)]^t$$

et

$$Y(m) = [y_0(m), y_1(m), \ldots, y_{M-1}(m)]^t$$

On obtient la relation

$$Y(m) = \begin{bmatrix} h_0(0) & \ldots & h_0(N-1) \\ \vdots & \ldots & \vdots \\ h_{M-1}(0) & \ldots & h_{M-1}(N-1) \end{bmatrix} X(m) = TX(m)$$

Le banc de filtres d'analyse est donc équivalent à une transformation T caractérisée par une matrice rectangulaire $(M \times N)$ dont les vecteurs lignes sont les réponses impulsionnelles des filtres d'analyse.
A la synthèse comme à l'analyse, les filtres du banc de synthèse peuvent s'écrire sous forme matricielle pour désigner la matrice de transformation de synthèse.

Cas de la Transformée en Cosinus Discrète TCD

Lorsque le banc de filtres d'analyse est modulé avec

$$h_k(n) = 2h(n).\cos(\frac{\pi(2k+1)n}{2M})$$

la matrice associée à la transformée s'écrit sous la forme

$$T = 2A \begin{bmatrix} h(0) & 0 & \dots & 0 & 0 \\ 0 & h(1) & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & h(N-1) \end{bmatrix}$$

avec

$$A = \begin{bmatrix} 1 & \cos(\frac{\pi}{2M}) & \dots & \dots & \cos(\frac{\pi(N-1)}{2M}) \\ 1 & \cos(\frac{3\pi}{2M}) & \dots & \dots & \cos(\frac{3\pi(N-1)}{2M}) \\ \vdots & \dots & \dots & \dots & \vdots \\ 1 & \cos(\frac{\pi(2M-1)}{2M}) & \dots & \dots & \cos(\frac{\pi(2M-1)(N-1)}{2M}) \end{bmatrix}$$

Si on considère la matrice carrée $(M \times M)$ composée des M premiers vecteurs colonnes de A, ces vecteurs sont orthogonaux. Comme ils ne sont pas normés et que la norme du premier vecteur est égale à $\sqrt{M}$ et la norme des vecteurs suivants est égale à $\sqrt{M/2}$, on définit la matrice

$$B = \frac{1}{\sqrt{M}} C$$

où

$$C = \begin{bmatrix} 1 & \sqrt{2}\cos(\frac{\pi}{2M}) & \dots & \dots & \sqrt{2}\cos(\frac{\pi(M-1)}{2M}) \\ 1 & \sqrt{2}\cos(\frac{3\pi}{2M}) & \dots & \dots & \sqrt{2}\cos(\frac{3\pi(M-1)}{2M}) \\ \vdots & \dots & \dots & \dots & \vdots \\ 1 & \sqrt{2}\cos(\frac{\pi(2M-1)}{2M}) & \dots & \dots & \sqrt{2}\cos(\frac{\pi(2M-1)(M-1)}{2M}) \end{bmatrix}$$

B vérifie cette égalité

$$B^t B = I$$

En effet, il s'agit d'une (parmi quatre) définition de la Transformée en Cosinus Discrète [36].

Les Conditions de Reconstruction Parfaite pour un Banc de Filtres à M Sous-bandes

Considérons le banc de filtres présenté à la figure 2.3. A la synthèse, la reconstruction se caractérise par la matrice P et les opérateurs de sur-échantillonnage, de décalage et d'addition qui permettent successivement de transformer d'abord le vecteur $Y(m)$ en $\hat{X}(m)$ puis le vecteur $\hat{X}(m)$ en séquence $\hat{x}(n)$. La condition que le banc de filtres doit respecter pour avoir une reconstruction parfaite, est tout simplement de s'assurer que les échantillons après passage dans les lignes de retard ressortent du banc dans l'ordre dans lequel ils sont entrés. Donc on retrouve à la sortie les échantillons du signal d'entrée bien ordonnés, si et seulement si la matrice produit

$P(z)T(z)$ est égale à la matrice identité, éventuellement retardée. Cette condition est appelée condition de bi-orthogonalité. Le signal reconstruit vérifie $\hat{x}(n) = x(n-d)$ où d représente le délai de reconstruction. On peut caractériser les filtres de synthèse $F_k(z)$ représentés à la figure 2.2 en fonction de P indiqué figure 2.3. Si on considère que toutes les composantes du vecteur $Y(m)$ sont nulles sauf la k^{ieme}, le schéma de la figure 2.2 nous montre que le signal à la sortie est la réponse impulsionnelle $f_k(n)$ du filtre $F_k(z)$, plus précisement

$$[\hat{x}(nM)\dots\hat{x}(mM+M-1)] = [f_k(0)\dots f_k(M-1)]$$

On en déduit de la figure 2.3 les relations suivantes.

$$[\hat{x}_0(m)\dots\hat{x}_{M-1}(m)] = [P_{0,k}\dots P_{M-1,k}]$$

$$[\hat{x}(mM)\dots\hat{x}(mM+M-1)] = [\hat{x}_{M-1}(m)\dots\hat{x}_0(m)].$$

Ainsi on obtient

$$[f_k(0)\dots f_k(M-1)] = [P_{M-1,k}\dots P_{0,k}].$$

La k^{ieme} colonne renversée de la matrice P forme la réponse impulsionnelle du filtre de synthèse $F_k(z)$. Pour les relations

$$\hat{X}(m) = PY(m)$$

et

$$Y(m) = TX(m)$$

prenons le cas particulier de $T = P^t$ (où "t" dénote la transposition de P), on retrouve une simple transformation orthogonale (ou unitaire) ou un banc de filtres orthogonal. La condition $T = P^t$ impose la relation suivante entre les réponses impulsionnelles des filtres d'analyse et de synthèse

$$h_k(n) = f_k(M-1-n).$$

2.4 La Quantification

Comme on traite les signaux à temps discret, les coefficients à la sortie de la transformée (ou des filtres de synthèse dans le cas d'un banc de filtres) sont des valeurs réelles. Pour réaliser la compression, on aura besoin de représenter les valeurs réelles des coefficients par un ensemble de valeurs discrètes. Cette procédure est appelée quantification.

Dans le contexte de la compression d'image, la réduction de la résolution passe souvent inaperçue à l'œil nu par rapport à la résolution originale, ceci est réalisable par la quantification qui réduit éfficacement le nombre de bits de la représentation

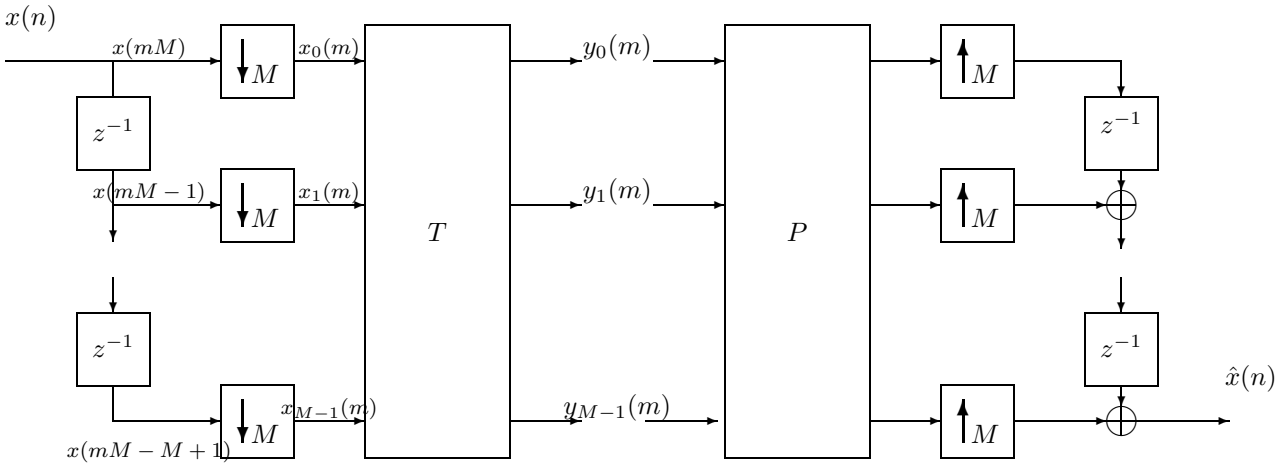

Figure 2.3: Schéma équivalent d'un banc de filtres d'analyse et de synthèse lorsque $N = M$

des coefficients, d'où son intérêt dans la chaîne de compression d'image. Une étude détaillée sur la quantification est réalisée par Gersho dans [12].
La quantification dans sa forme la plus simple consiste à quantifier chaque coefficient individuellement, on parle ainsi de la quantification scalaire. Une autre méthode plus complexe consiste à quantifier plusieurs coefficients à la fois, c'est la quantification vectorielle sous ses différentes alternatives qui fournit théoriquement la performance optimale pour tout codeur [12].

2.4.1 La Quantification Scalaire

Un exemple de quantification scalaire est donné à la figure 2.4. la dynamique d'entrée est divisée en intervalles $I_i = [x_{i-1}, x_i]$ et la valeur à la sortie y_i est typiquement choisie dans l'intervalle I_i. L'ensemble $\{y_i\}$ et y_i sont appelés respectivement le dictionnaire (Codebook) et centroïde.

La Quantification Scalaire Uniforme

Soit le signal à temps discret $x(n)$ prenant ses valeurs dans l'intervalle $[a, b]$ avec une loi uniforme, nous le considérons comme la réalisation d'un processus aléatoire $X(n)$ centré (ce signal est tout à fait irréaliste surtout dans le cas de l'image, mais il va nous permettre de simplifier le calcul et de donner une formule de la puissance de l'erreur de quantification en fonction de la puissance du signal et de la résolution). La résolution $R = \lceil log_2 N \rceil$ est fonction du nombre N d'intervalles divisant la dynamique d'entrée $[a, b]$. Pour définir un quantificateur scalaire uniforme, il faut:

1. partitionner l'intervalle $[a, b]$ en $N = 2^R$ intervalles distincts de même longueur $\Delta = (b - a)/2^R$;

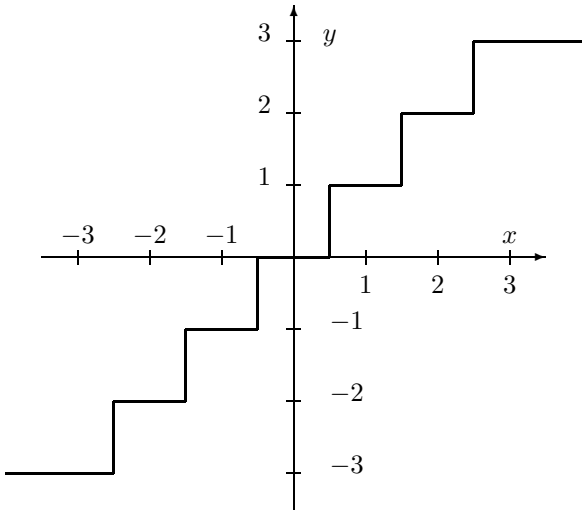

Figure 2.4: Quantification scalaire uniforme

2. associer un numéro à chaque intervalle "i";

3. définir un représentant par intervalle "y_i", par exemple le milieu de l'intervalle.

L'opération appelée "codage" consiste à trouver l'intervalle auquel appartient une valeur d'entrée $x(n)$ et à lui associer le numéro de cet intervalle $i(n) \in \{1 \dots N\}$ qui sera transmis ou stocké. L'opération de "décodage" consiste à associer au numéro $i(n)$ le $y_{i(n)}$ correspondant dans le dictionnaire.

$$\hat{x}(n) = y_{i(n)}$$

Ainsi la quantification est un processus à 2 étapes:

$$x(n) \stackrel{codage}{\longrightarrow} i(n) \stackrel{décodage}{\longrightarrow} y_{i(n)}$$

et l'erreur de quantification est exprimée par

$$q(n) = x(n) - \hat{x}(n)$$

où $x(n)$ et $\hat{x}(n)$ sont respectivement l'entrée et la sortie du quantificateur.
La performance d'un quantificateur est mesurée par la distance entre les valeurs d'entrée et de sortie au sens de l'Erreur Quadratique Moyenne EQM qu'on cherche souvent à minimiser ou dans un autre sens à maximiser le rapport Signal sur bruit

$$D = E[|X(n) - \hat{x}(n)|^2]$$

$$SNR = \frac{E[X^2(n)]}{E[|X(n) - \hat{x}(n)|^2]}$$

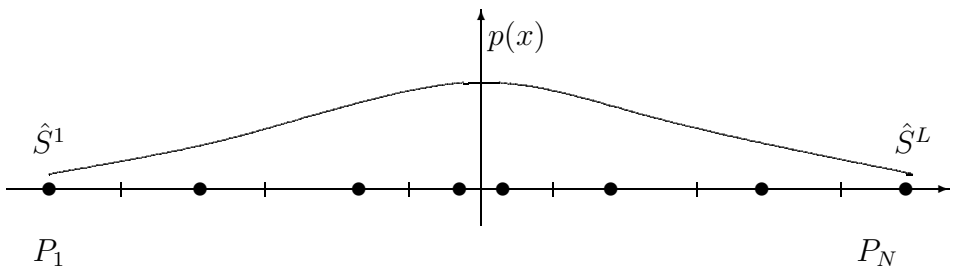

Figure 2.5: Quantification scalaire non-uniforme

A priori l'erreur de quantification q(n) est une fonction déterministe de $x(n)$. Elle est souvent modélisée par un processus aléatoire de bruit blanc $Q(n)$, décorrélé du signal $x(n)$ avec une distribution uniforme. La moyenne de cette erreur de quantification est nulle, sa variance est donnée par

$$\sigma_Q^2 = E[Q^2(n)] = \int_{-\Delta/2}^{+\Delta/2} x^2 \, \frac{1}{\Delta} \, dx$$

Pour N intervalles représentant la dynamique du signal d'entrée $[a, b]$, σ_Q^2 devient

$$\sigma_Q^2 = \frac{\Delta^2}{12} = \frac{1}{12} \frac{(b-a)^2}{N^2} = \frac{1}{12} \frac{(b-a)^2}{2^{2R}}$$

La variance du signal $X(n)$ est

$$\sigma_X^2 = E[X^2(n)] = \frac{(b-a)^2}{12}$$

$$\sigma_Q^2 = \sigma^2 \, 2^{-2R} = C \; 2^{-2R}$$

Le rapport Signal sur Bruit exprimé en dB est

$$SNR = 10 \, log_{10} \frac{E[X^2(n)]}{E[Q^2(n)]} = 10 \, log_{10} 2^{2R} = 6.02 \, R.$$

Ainsi le fait d'ajouter un bit/pixel revient à augmenter le SNR de $6dB$.

La Quantification Scalaire non-Uniforme

Comme on l'a déjà cité, les hypothèses sur la nature du signal d'entrée sont irréalistes du fait qu'une image ne peut jamais être considérée comme un processus aléatoire stationnaire avec une fonction de distribution de probabilité "*pdf*" uniforme dans un intervalle $[a, b]$. En revanche on pourra considérer l'image comme un signal localement uniforme mais avec des lois de probabilité non-uniformes. Si la "*pdf*" n'est plus uniforme, alors la quantificateur uniforme n'est plus le quantificateur

optimal. On note $f_X(x)$ la fonction de densité de probabilité du processus aléatoire $X(n)$. La distorsion entre l'entrée et la sortie est mesurée par l'erreur quadratique

$$d(x, \hat{x}) = (x - \hat{x})^2$$

et la distorsion moyenne devient l'Erreur Quadratique Moyenne EQM

$$D = E[d(x, \hat{x})] = \int_{-\infty}^{+\infty} d(x, \hat{x})\, f_X(x)\, dx$$

$$D = \sum_{i=1}^{N} \int_{x \in P_i} d(x, \hat{y}_i)\, f_X(x)\, dx$$

où P_i est l'élément de la partition associé au numéro i, sachant qu'on a N partitions (P_i $i = 1, ..., N$) de longueurs distinctes comme le montre la figure 2.5. La longueur est d'autant plus petite que la densité de probabilité correspondante est importante. La distorsion moyenne devient l'erreur quadratique moyenne (MSE)

$$D = \sigma_Q^2 = \sum_{i=1}^{N} \int_{x \in P_i} d(x, \hat{y}_i)\, f_X(x)\, dx \tag{2.2}$$

Un quantificateur optimal au sens de la MSE est celui qui minimise D dans (2.2) pour un nombre N donné de symboles de sortie $\hat{y}_i$.
En effet pour définir ce quantificateur, il s'agit de trouver la partition $\{P_1 \dots P_N\}$ et les symboles de sortie $\{\hat{y}_1 \dots \hat{y}_N\}$ qui minimisent D. Cette minimisation conjointe n'admet pas de solution simple. Il existe deux conditions nécessaires d'optimalité [12, 31]. Si l'on connaît les symboles de sortie $\{\hat{y}_1 \dots \hat{y}_N\}$ on peut calculer la meilleure partition $\{P_1 \dots P_N\}$. Si l'on se donne la partition, on peut en déduire les meilleurs représentants. En d'autre terme, la partie dite "codage" du quantificateur doit être optimale étant donné la partie "décodage" et réciproquement. Les 2 conditions nécessaires d'optimalité, lorsqu'on choisit comme mesure de distorsion l'erreur quadratique sont:

1. La règle du plus proche voisin.
2. La condition du centroïde.

La Règle du plus proche voisin:

Etant donné un dictionnaire de symboles $\{\hat{y}_1 \dots \hat{y}_N\}$, la meilleure partition est celle qui vérifie

$$P_i = \{x : (x - \hat{y}_i)^2 \leq (x - \hat{y}_j)^2 \ \forall\, j \in \{1 \dots N\}\}.$$

En effet, si l'on appele t_i la valeur définissant la frontière entre les partitions P_i et P_{i-1}, la minimisation de D la MSE relativement à t_i est obtenue par

$$\frac{\partial}{\partial t_i} \left[\int_{t_{i-1}}^{t_i} (x - \hat{y}_i)^2 f_X(x) dx + \int_{t_i}^{t_{i+1}} (x - \hat{y}_{i+1})^2 f_X(x) dx \right] = 0$$

$$(t_i - \hat{y}_i)^2 f_X(t_i) - (t_i - \hat{y}_{i+1})^2 f_X(t_i) = 0$$

ainsi

$$t_i = \frac{\hat{y}_i + \hat{y}_{i+1}}{2}$$

donc la meilleure partition est celle qui correspond aux points milieux entre deux symboles de sortie adjacents.

La Condition du centroïde:

Etant donné une partition $\{P_1 \dots P_N\}$, les meilleurs symboles $\{\hat{y}_1 \dots \hat{y}_N\}$ sont obtenus par le centre de gravité de la partie de la densité de probabilité placée dans la région P_i.

$$\hat{y}_i = \frac{\int_{x \in P_i} x f_X(x) dx}{\int_{x \in P_i} f_X(x) dx} = E(X/X \in P_i) \tag{2.3}$$

En effet, on remarque d'abord que la minimisation de D relativement à $\hat{y}_i$ ne fait intervenir qu'un élément de la somme donnée par (2.3). En écrivant ensuite

$$\frac{\partial}{\partial \hat{y}_i} \int_{x \in P_i} (x - \hat{y}_i)^2 f_X(x) dx = 0$$

$$2 \int_{x \in P_i} x f_X(x) dx + 2\, \hat{y}_i \int_{x \in P_i} f_X(x) dx = 0$$

ainsi on obtient la 1ère égalité de (2.3).
Comme

$$\int_{x \in P_i} x f_X(x) dx = \int_{x \in P_i} f_X(x) dx \int_{-\infty}^{+\infty} f_{X/P_i}(x) dx$$

où $f_{X/P_i}(x)$ est la pdf condititionnelle de X sachant que $X \in P_i$, on obtient

$$\hat{y}_i = \int_{-\infty}^{+\infty} f_{X/P_i}(x) dx$$

$$\hat{y}_i = E(X/X \in P_i).$$

La valeur que l'on doit choisir est la valeur moyenne de X dans l'intervalle considérée.
Notons qu'une telle quantification n'est pas nécessairement optimale pour la compression tant que le codage entropique n'est pas pris en compte.
Les deux conditions citées plus haut sont à la base de l'algorithme itératif de **Lloyd-Max** [30] qui améliore à chaque itération le dictionnaire, ayant la fonction de densité de probabilité et le nombre de symboles N.
Avec l'initialisation du dictionnaire à $\{\hat{y}_i^{(0)}\}$, l'algorithme comprend les 2 étapes suivantes:

1. Ayant $\{\hat{y}_i^{(n)}\}$, trouver la meilleure partition $\{P_i\}$, satisfaisant la condition du plus proche voisin.

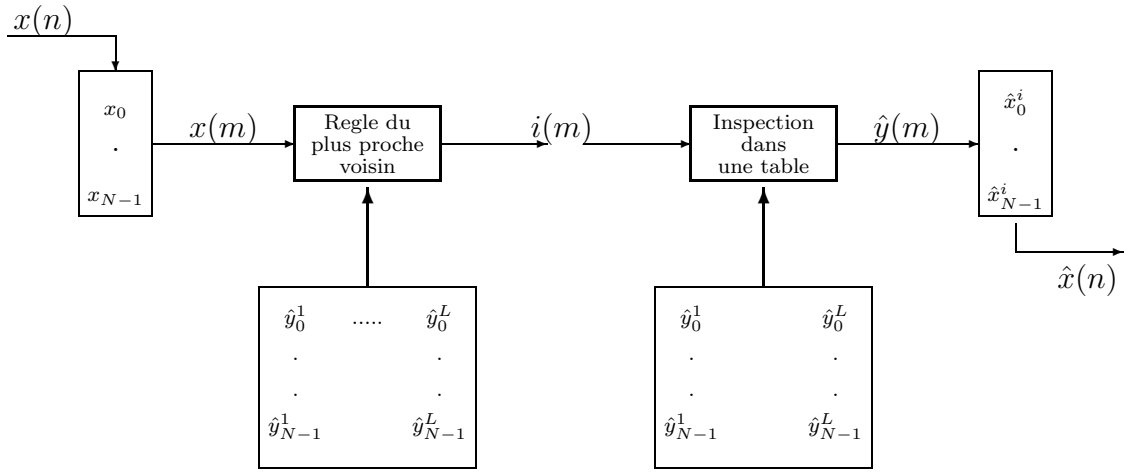

Figure 2.6: Quantification Vectorielle

2. Pour la partition $\{P_i\}$ (donnée ou celle trouvée à l'étape (1)), calculer le meilleur dictionnaire $\{\hat{y}_i^{(n+1)}\}$ satisfaisant la condition de centroïde.

L'algorithme converge quand la distorsion $D(n)$ sera quasiment constante. On assure généralement la décroissance de la distorsion moyenne mais on ne tend pas toujours vers le minimum golbal. On atteint simplement un minimum local.

2.4.2 La Quantification Vectorielle

La quantification vectorielle n'est qu'une généralisation de la quantification scalaire à plusieurs dimensions [15]. Elle permet de prendre en compte directement la corrélation contenue dans le signal, plutôt que de chercher d'abord à décorréler le signal, puis à quantifier ce signal décorrelé comme le fait la quantification scalaire prédictive.
La quantification vectorielle sera symbolisée par la figure 2.6 où chaque vecteur $x(m)$ est disponible à des instants multiples de N. Les vecteurs

$$\hat{y}^i = [\hat{y}_0^i \ \ldots \ \hat{y}_{N-1}^i]^t$$

représentent les vecteurs symboles à la sortie du quantificateur. L'ensemble des vecteurs $\hat{y}^i$ assimilable à une matrice est le dictionnaire. En effet, ce dictionnaire C n'est qu'une partition de l'espace R^N en L régions définies par

$$P_i = \{x; Q(x) = \hat{y}^i\}$$

où Q est une application de R^N dans un ensemble fini C contenant L vecteurs de dimension N

$$Q : R^N \longrightarrow C \quad C = \{\hat{y}^i, \ldots, \hat{y}^L\} \ \ où \ \ \hat{y}^i \in R^N$$

Un quantificateur vectoriel optimal au sens de la MSE est celui qui satisfait les deux conditions décrites pour les quantificateurs scalaires, c-à-d:

1. La règle du plus proche voisin.
2. La condition du centroïde.

Notons que ces deux conditions d'optimalité ne signifient pas une optimalité globale, mais plutôt locale pour un dictionnaire satistfaisant les deux conditions. La définition d'un dictionnaire de quantification vectorielle est une technique très sophistiquée qui consiste en une bonne initialisation suivi par une procédure itérative. L'exemple type de ce calcul est l'algorithme de Lloyd-Max détaillé dans [12].
L'inconvénient de la quantification vectorielle est sa complexité qui limite la taille des vecteurs $\hat{y}^i$ utilisés. Pour contourner ce problème on peut structurer le dictionnaire de façon à simplifier la recherche du vecteur convenable étant donné le vecteur d'entrée. Ceci est réalisable avec la quantification vectorielle arborescente. Une autre approche est d'utiliser des tranformations linéaires (telle que la décomposition en sous-bandes et les transfomées en ondelettes) et d'appliquer la quantification vectorielle sur les coefficients.
Moreau dans [31] décrit d'une façon détaillée les différentes alternatives pour la quantification vectorielle ainsi que leurs utilisations.
Reste à dire que nous privilegeons dans la suite l'utilisation des quantificateurs scalaires uniformes pour leur simplicité de mise en œuvre, employés conjointement avec une transformée performante et un codage entropique ces quantificateurs sont performant autant que les quantificateurs vectoriels [7].

2.5 L'Allocation de Bits

En regardant le schéma de codage par transformée la question qui se pose est comment choisir et avec quels critères les quantificateurs des coefficients transformés?
En effet, bien que ça semble être un problème classique d'allocation de ressources, il parait important de préciser les conditions d'application des algorithmes classiques d'allocation de bits. Dans ce type d'algorithme on essaye de minimiser la distorsion sous la contrainte d'un débit demandé. Dans le cas de transformées unitaires ou orthogonales (à l'analyse et la synthèse) l'erreur quadratique totale entre le signal d'entrée et celui à la sortie est la somme des erreurs quadratiques apportées par la quantification de chacun des coefficients. Plus précisement, appelons x et $\hat{x}$ respectivement les signaux d'entrée et reconstruit, y et $\hat{y}$ l'entrée et la sortie de l'étage de quantification.

$$y = Tx \ , \ \ \hat{x} = T^t \hat{y}$$

où T est la matrice associée à une transformée unitaire et $T^t = T^{-1}$ la transformée de synthèse

$$TT^t = T^t T = I.$$

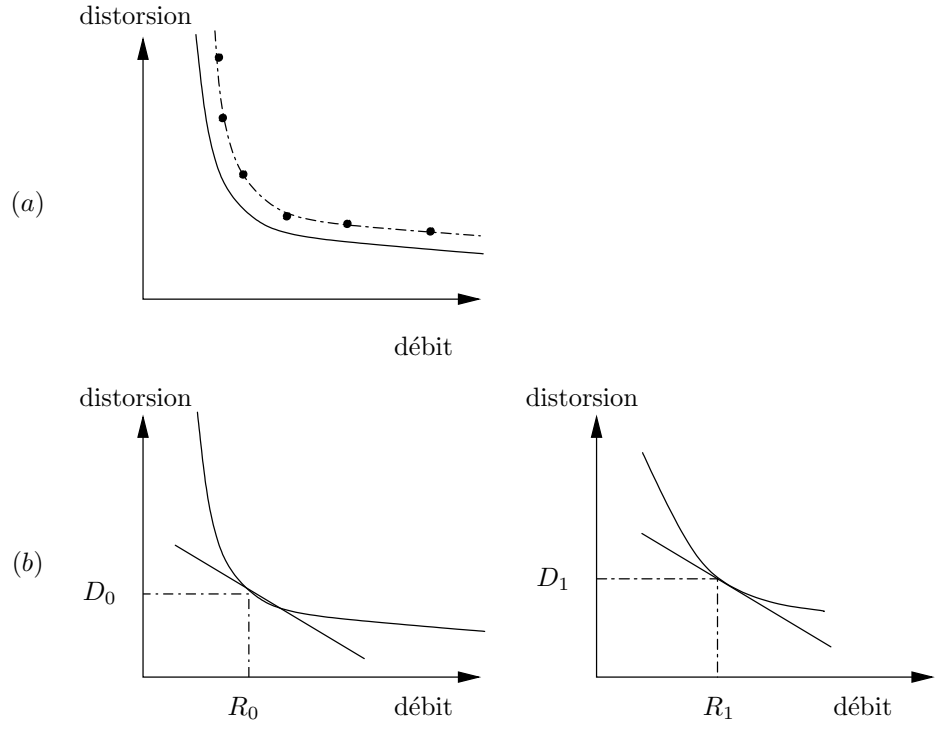

Figure 2.7: Débit/Distorsion et alloctaion de bits. (a) Courbe R/D théorique (en continue) et courbe opérationnelle (en pointillé). (b) Solution optimale à pente $-\lambda$ constante.

La distorsion totale dans le sens de la MSE est exprimée par

$$D = E\left[(x-\hat{x})^t(x-\hat{x})\right] = E\left[(y-\hat{y})^t T T^t (y-\hat{y})\right]$$

$$D = E\left[(y-\hat{y})^t(y-\hat{y})\right] = E\left[\sum_{i=0}^{N-1}(y_i-\hat{y}_i)^2\right] = \sum_{i=0}^{N-1} D_i$$

$$D = \sum_{i=0}^{N-1} D_i. \tag{2.4}$$

Le problème d'allocation de bit est alors de minimiser la distorsion D sous la contrainte de débit

$$\sum_{i=0}^{N-1} R_i \leq R \tag{2.5}$$

où R est le débit total et R_i est le nombre de bits alloués pour le $i^{ème}$ coefficient.
Avant de spécifier une procédure d'allocation, rappelons nous tout d'abord de quelques aspects d'une solution optimale.
Le compromis essentiel en quantification est celui qui existe entre le débit et la distorsion par la théorie de Débit/Distorsion [16].
En effet les limites de débit-distorsion ne sont pas atteintes en pratique. Dans la

figure 2.7.a, on distingue la fonction théorique Débit/Distorsion (R,D) des points pratiquement calculés et formant une courbe (R,D) opérationnelle. Notons qu'au contraire de la fonction théorique, la courbe opérationnelle n'est pas nécessairement convexe. Pour plus de clarté, prenons l'exemple d'une quantification scalaire, la distorsion D_i est donnée en fonction de R_i par

$$D_i(R_i) \approx C_i \sigma_i^2 2^{-2R_i}$$

où C_i est une constante qui dépend de la *pdf* du $i^{ème}$ coefficient quantifié. Supposons qu'on code deux variables x_0 et x_1 où chacune peut avoir sa propre fonction (R,D). Les conditions de génération de (2.4) et (2.5) sont des propriétés clés pour optimiser l'allocation de bit (additivité de la distorsion et du débit suite à l'utilisation des transformées orthogonales).
Shoham et Gersho [13] proposent d'utiliser la fonction coût de Lagrange L qui regroupe les deux paramètres Débit et Distorsion à une valeur λ positive dite multiplicateur de Lagrange.

$$L = D + \lambda R$$

$$L_i = D_i + \lambda R_i$$

Mimimiser L (qui dépend maintemant de λ) revient à chercher le minimum de chaque L_i (propriété d'additivité) en remplaçant dans L_i, D_i par sa valeur en fonction de R_i. Le minimum de L_i est obtenu en mettant à 0 la dérivée de L_i par rapport R_i

$$\frac{\partial L_i}{\partial R_i} = \frac{\partial D_i(R_i)}{\partial R_i} + \lambda = 0$$

Ceci dit que la pente de chaque courbe (R,D) est égale à $-\lambda$. La solution est unique si les courbes sont convexes. Ainsi la solution optimale est obtenue en choisisant sur chacune des courbes le point qui correspond à la pente $-\lambda$, figure 2.7.b [13]. Il est clair que cette solution est intuitive. Prenons le cas de deux variables $i = 0, 1$, supposons que (R_0, D_0) et (R_1, D_1) n'ont pas la même pente et λ_0 est largement supérieure à λ_1. Pour un débit $R = R_1 + R_2 \leq R_d$, si on augmente R_0 de ϵ, il faudra diminuer R_1 de au moins la même quantité. Sur les courbes (R,D), on décrémente la distorsion D_0 et incrémente D_1. Comme le débit total $R \leq R_d$ au moment où la distorsion globale diminue, on répette cette procédure pour se rapprocher de plus en plus de la solution optimale où les deux pentes sont égales et au delà de ce point en ne gagne plus rien.
Une solution à pente constante est toujours possible pour une valeur R donnée. Les fonctions $D_i(R_i)$ ne sont pas toujours connues. Shoham et Gersho [13] utilisent les courbes (R_i, D_i) opérationnelles pour trouver la solution optimale parmi les points opérationnels.
Si on effectuait toutes les combinaisons possibles $(R_i(j), D_i(j))$ où $i = 0, ..., N-1$ est le nombre de coefficients et $j = 1, ..., M$ est le nombre de niveaux de quantification

pour chaque coefficient, on obtiendrait un nuage dense de points dans le plan (R, D). Appelons I une combinaison quelconque de N quantificateurs $I = (i_0, ..., i_{N-1})$, l'algorithme d'optimisation proposé dans [13] consiste en deux étapes.

1. Pour une pente $-\lambda$ donnée, on cherche le point optimum (R^*_{opt}, D^*_{opt}) uniquement sur l'enveloppe convexe de ce nuage et qui est l'intersection avec la droite $D = -\lambda R + L$.
   Ce point optimum est caractérisé par
   $$I_{opt}(\lambda) = arg \min_I [D(I) + \lambda\, R(I)]$$
   dans le cas de transformées orthogonales où $D = \sum_{i=0}^{N-1} D_i$ et $R = \sum_{i=0}^{N-1} R_i$
   $$D(I) = \sum_{k=0}^{N-1} D_k(i_k)$$
   $$R(I) = \sum_{k=0}^{N-1} R_k(i_k)$$
   $$I_{opt}(\lambda) = arg \min_I \left[\sum_{k=0}^{N-1} D_k(i_k) + \lambda \sum_{k=0}^{N-1} R_k(i_k)\right]$$
   soit
   $$I_{opt}(\lambda) = arg \sum_{k=0}^{N-1} \min_{i_k} [D_k(i_k) + \lambda R_k(i_k)]\,.$$
   On constate que cette technique permet de réduire le nombre de combinaisons possibles des différents quantificateurs de M^N à $(M \times N)$ où N minimisations indépendantes sont suffisantes.

2. A la deuxième étape on essaye d'optimiser la valeur de λ pour laquelle la 1ère étape fournit le point optimum. Toutes les droites déterminées par la minimisation précédente sont caractérisées par une équation qui ne dépend que de λ
   $$L_{I_{opt}}(\lambda) = D + \lambda R.$$
   En considérant toutes ces droites, il suffit de choisir celle qui maximise l'ordonnée du point d'intersection avec la droite $R = R_d$. En fait, comme cette ordonnée vaut
   $$D(\lambda) = L_{I_{opt}}(\lambda) - \lambda R_d = D_{opt}(\lambda) + \lambda(R_{opt}(\lambda) - R_d)$$
   la pente optimale λ_{opt} est donc la solution de
   $$\lambda_{opt} = arg \max_\lambda (D_{opt}(\lambda) + \lambda(R_{opt}(\lambda) - R_d).$$
   Les N composantes de $I_{opt}(\lambda_{opt})$ caractérisent les N quantificateurs optimaux.

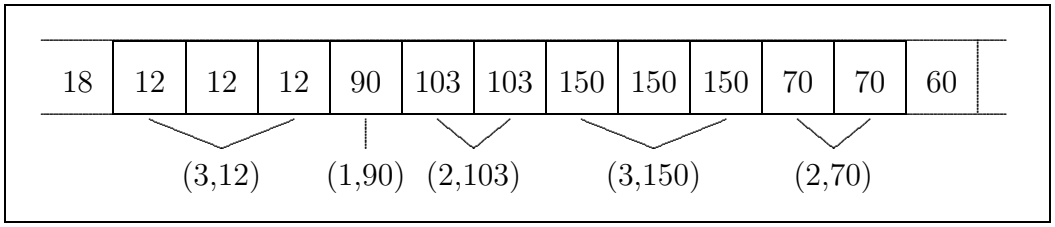

Figure 2.8: Exemple de Codage par plage RLE

2.6 Le Codage par plage (Run length Encoding)

Le procédé *Run-Length* ne relève pas d'une théorie mathématique très complexe. Il s'agit simplement de remplacer des éléments significatifs, successifs et identiques par un seul d'entre eux, suivi du nombre de répétitions. Un exemple de traitement *RLE* est donné à la figure 2.8. Ce procédé peut paraître simpliste et peu performant si on cherche à l'appliquer, par exemple, à un texte, les répétitions de lettres n'apporteraient qu'une compression dérisoire! En revanche, si on l'applique à une image, il est aisé de s'apercevoir que les plages de couleurs (ou niveaux de gris) homogènes sont souvent importantes, même si leur nombre est parfois faible.

2.6.1 Le Procédé du Codage par plage

Si n octets successifs sont dans un même état, il est aisé de transmettre l'octet répété et le nombre de répétitions. On pourra ainsi, dans la plupart des cas, coder sur 3 octets les n octets composant le signal initial. S'il est relativement simple de coder l'octet à répéter, suivi du nombre de répétitions dans l'octet suivant, cette méthode peut se révéler très pénalisante pour certains cas: à la limite si deux octets consécutifs sont toujours différents, la taille des données compréssées sera la double de celle des données initiales! Pour éviter cet inconvénient, les versions les plus avancées du *RLE* utilisent un code discriminant pour indiquer le début d'une séquence *octet à répéter + nombre de répétitions*, les octets isolés restant codés sous leur forme initiale.
Dans la norme *JPEG* le "Codage par plage" se focalise sur les répétitions des "zéros" dans une séquence (ou un balayage). Ainsi les coefficients *AC* quantifiés dans un balayage subbissent un codage par plage avant le passage par le codeur de **Huffman**. La figure 2.9 représente l'implantation du "Codage par plage" des coefficients *AC* dans la norme *JPEG*.

2	3	4	5	6	7	8	9	10	11	12	...	63	64
412	0	0	-23	9	0	0	0	0	0	6	...	0	0

La première ligne indique l'ordre du coefficient AC dans le balayage en zig-zag.
La deuxième ligne indique la valeur du coefficient AC quantifié.

(0,412), (2,-23), (0,9), (5,6), ..., EOB.

Figure 2.9: Codage par plage des coefficients *AC* dans JPEG

2.7 L'Entropie

2.7.1 Définitions

Soit une série d'images numériques du même type (mêmes dimensions, même nature telle que noire et blanche, même nombre de niveaux de gris...). Lorsque l'on regarde une image particulière de cette série, on a conscience d'avoir reçu une information. On admet qu'une grande image a toutes les chances d'apporter une plus grande quantité d'information qu'une petite. Pour quantifier , c'est-à-dire chiffrer, cette quantité d'information, la théorie de l'information à laquelle **Shannon** a laissé des travaux d'une importance essentielle [40, 11, 16] nous introduit la valeur prise par une variable aléatoire.

D'ailleurs chaque point de l'image sera considéré comme une variable aléatoire et un bloc de $N \times N$ sera considéré comme une suite de N^2 variables aléatoires indépendantes et identiquement distribuées prenant ses valeurs dans l'ensemble

$$A_x = \{x^1, \ldots, x^K\}$$

on parle alors de Source Discrète sans mémoire (Discret Memoryless Source DMS) [17, 31].

Information Propre

Pour $X(n)$ un point de l'image, nous définissons l'information propre de l'événement $\{X(n) = x^i\}$ par

$$I(x^i) = -log_2 p_X(x^i) \ bits/symbole \ \ i \in [1, K]$$

où $p_X(x^i)$ est la probabilité de la valeur x^i. Cette quantité est positive ou nulle.

Entropie du point

On appelle Information Propre Moyenne ou Entropie du point $X(n)$ l'espérance de $I(x^i)$

$$H(X) = E[I(x^i)] = -\sum_{i=1}^{K} p_X(x^i) log_2 p_X(x^i) \quad bits$$

Shannon [40] définit l'entropie comme le débit binaire moyen minimal qu'un système de codage devra respecter pour transmettre les échantillons de la source. D'une manière générale, les variables aléatoires d'un processus ne sont pas des DMS et en conséquence le calcul de l'entropie devra prendre en compte cette redondance. Quoi qu'il en soit on ne peut, dans la réalité, qu'approcher la valeur de l'entropie.

Entropie d'un bloc

Ce qui précède a permis de calculer la quantité d'information obtenue lorsque l'on connaît la valeur d'un point de l'image. Pour un bloc de $N \times N$, quelle est la quantité d'information que l'on tire de la connaissance des valeurs de ces N^2 points?
Considérons le vecteur aléatoire (en effet c'est le bloc de l'image reformaté en vecteur)

$$X_N = [X(0), X(1), \ldots, X(N-1)]^t$$

et appelons $p_{X_N}(x^{i_0}, x^{i_1}, \ldots, x^{i_{N-1}})$ la probabilité conjointe

$$p_{X_N}(x^{i_0}, x^{i_1}, \ldots, x^{i_N}) = P\{X(0) = x^{i_0}, \ldots, X(N-1) = x^{i_{N-1}}\}$$

Nous définissons l'entropie du processus aléatoire correspondant au vecteur X_N par [23]

$$H(X_N) = -\sum_{x^{i_0}, x^{i_1}, \ldots, x^{i_{N-1}}} p_{X_N}(x^{i_0}, x^{i_1}, \ldots, x^{i_N}) log_2 p_{X_N}(x^{i_0}, x^{i_1}, \ldots, x^{i_N}) \quad bits$$

Partant de là, le débit entropique est

$$\bar{H}(X) = \lim_{N \to \infty} \frac{1}{N} H(X_N)$$

Dans le cas où le processus X_N est une suite de variables aléatoires indépendantes et identiquement distribuées, l'entropie vaut

$$H(X_N) = -\sum_{x^{i_0}, x^{i_1}, \ldots, x^{i_{N-1}}} p_X(x^{i_0}) \ldots p_N(x^{i_N}) log_2[p_X(x^{i_0}) \ldots p_N(x^{i_N})] \quad bits$$

$$H(X_N) = -N \sum_{i=1}^{K} p_X(x^i) log_2 p_X(x^i) \quad bits$$

$$H(X_N) = NH(X)$$

Le débit entropique du bloc est donc égal à l'entropie du point. De façon générale, on a l'inégalité

$$0 \leq \bar{H}(X) \leq H(X) \leq log_2 K$$

Le débit entropique est limité par la valeur de l'entropie, il est considérablement faible pour les images redondantes.

2.7.2 Le Codage Entropique

La dernière étape dans un schéma de codage est le codage entropique. L'opération du codage entropique est parfaitement réversible c-à-d elle n'introduit aucune distorsion comme c'est le cas avec la quantification. Après la phase de quantification, les coefficients prennent des valeurs dans un enemble fini de symbole $\{\hat{y}_i\}$. L'idée est de trouver une sorte de bijection de l'ensemble $\{\hat{y}_i\}$ dans un nouvel ensemble $\{\hat{z}_i\}$ de manière à minimiser le nombre moyen de bits/coefficient. Pour établir une telle bijection, Les paramètres sont les probabilités de présence $P(\hat{y}_i)$ de chaque symbole $\{\hat{y}_i\}$ dans l'ensemble de coefficients. Les probabilités $P(\hat{y}_i)$ sont fixes et un codage de **Huffman** [19] est énormement utile. En revanche si les probabilités évoluent avec le temps, des méthodes adaptives telle que le codage arithmétique adaptatif sont mieux employées. Cette bijection transforme les mots de code à longueur fixe en d'autres à longueur variable, générant ainsi un flux de bits à longueur variable.

Le Codage de Huffman

Comme il est déjà cité plus haut, le but du codage entropique est de trouver une bijection $\hat{z}_i = F(\hat{y}_i)$ qui peut minimiser la longueur moyenne des mots de code $l(\hat{z}_i)$, ayant les probabilités de présence $p(\hat{y}_i)$ comme paramètres.

$$E\left[l(\hat{z}_i)\right] = \sum_{i=0}^{M-1} p(\hat{y}_i) l(\hat{z}_i). \tag{2.6}$$

De plus, on exige que les mots de code $\{\hat{z}_i\}$ soient uniquement décodables. Cette condition se traduit par une contrainte sur les mots de code $\{\hat{z}_i\}$, plus précisement aucun mot de code ne doit pas être le préfixe d'un autre mot. Ainsi le flux de données $\hat{y}_i$ sera uniquement décodable par une simple reconnaissance des $\hat{z}_i$ sans référence aux mots de code qui suivent, ce code est dit instantané [19]. Rappelons que l'entropie de l'ensemble $\{\hat{y}_i\}$ est donnée par (2.7)

$$H = -\sum_{i=0}^{M-1} p(\hat{y}_i) log_2(p(\hat{y}_i)). \tag{2.7}$$

Cette entropie est la limite inférieure de la longueur moyenne des mots de code (2.6). La méthode de construction des mots de code proposée par **Huffman** satisfait la condition de préfixe et fournit le longueur moyenne des mots de code la plus proche

Symboles	$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$	$\hat{y}_5$	$\hat{y}_6$
Probabilités	0.5	0.15	0.17	0.08	0.06	0.04

Table 2.1: Probabilités associées aux événements $\hat{y}_i$

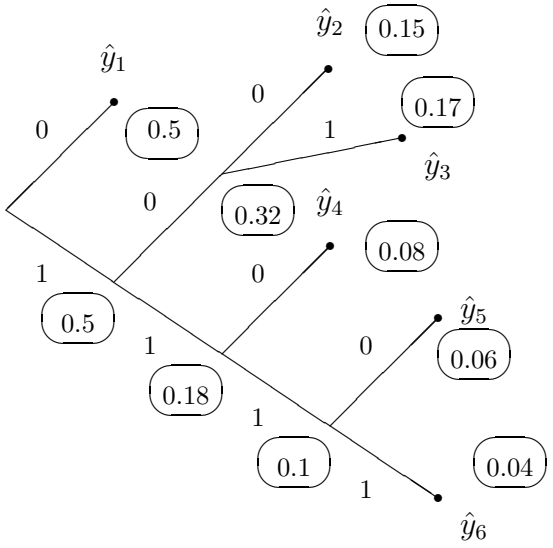

Figure 2.10: L'arbre binaire de l'algorithme de **Huffman**

de l'entropie.
L'algorithme de **Huffman** consiste à construire progressivement un arbre binaire en partant des noeuds terminaux vers la racine.

1. On part de deux listes $\{\hat{y}_1, \ldots, \hat{y}_M\}$ et $\{P(\hat{y}_1), \ldots, P(\hat{y}_M)\}$;

2. On sélectionne les deux symboles les moins probables, on crée deux branches dans l'arbre et on les étiquette par les deux symboles binaires 0 et 1;

3. On actualise les deux listes en rassemblant les deux symboles utilisés en un nouveau et en lui associant comme probabilité la somme des probabilités sélectionnées;

4. On recommence les deux étapes précédentes tant qu'il reste plus qu'un symbole dans la liste.

Le mot de code sera la séquence des 0 et 1 qui étiquettent les branches, partant de la racine vers le noeud terminal associé au symbole demandé (cf. table 2.2). Un exemple de calcul est donné à la figure 2.10.
Soit l'exemple d'une source ne pouvant prendre que 6 valeurs différentes et supposons connues les probabilités. Elles sont données dans la table (2.1).

Symboles	$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$	$\hat{y}_5$	$\hat{y}_6$
Mots de code	0	100	101	110	1110	1111

Table 2.2: Mots de code relatifs aux événements $\hat{y}_i$

L'entropie de cette source est égale à 2.06bits. Le code de Shannon entraîne une longueur moyenne égale à 2.28bits. La longueur moyenne est égale à 2.1bits, valeur très voisine de la limite théorique.
Bien que le codage de **Huffman** fournit des mots de code dont la longueur moyenne (2.6) est supérieure ou égale à l'entropie (2.7), le fait d'avoir un bit de plus pour l'entropie peut être inacceptable dans certaines applications, notamment en compression d'images où l'entropie n'est qu'une faible fraction de l'unité 1. Dans ce cas le codage de **Huffman** est inéfficace comme la longueur moyenne minimale est au moins égale à 1 (le mot de code le plus court avec un codage de **Huffman** est le bit 0 ou 1). Pour contourner ce type de problème, on peut regrouper des symboles pour former en nouvel ensemble de symboles et ainsi une longueur moyenne d'un bit sera distribuée sur plusieurs symboles d'origine.
En effet, dans les normes $JPEG$, $MPEG$ et $H261$ on trouve le codage de **Huffman** implanté de cette façon où un codage de type *Run-Length* (longueur de plage) précede le codage de **Huffman** pour regrouper les symboles.

Le Codage Arithmétique

Dans la théorie du codage **Arithmétique**, on attribue un "mot de code" à chaque ensemble de données. Ces "mots de code" correspondent à des sous-intervalles semi-ouverts de l'intervalle unité semi-ouvert $[0, 1[$. Nous construisons les "mots de code" en spécifiant suffisamment de bits pour distinguer le sous-intervalle correspondant à l'ensemble actuel de données, des autres sous-intervalles. Les codes les plus courts correspondent aux sous-intervalles les plus larges, donc aux ensembles de données les plus probables. Dans la pratique, le sous-intervalle est raffiné successivement en utilisant les probabilités individuelles des événements dans l'ensemble de données d'entrée. Les codes **Arithmétiques**, bien qu'ils réalisent souvent une compression meilleure que les codes préfixés, ils leurs manquent une correspondance directe entre les événements dans l'ensemble des données d'entrée et les bits du fichier de sortie codé.
Un codage statistique doit être accompagné, aux différents points de codage, d'un modèle d'estimation de probabilité pour chaque événement possible. Ce modèle de probabilité n'a pas besoin de décrire le processus qui génère les données, il doit tout simplement fournir une distribution de probabilité des données, qui n'est pas nécessairement précise. Le modèle peut être adaptatif (estimation dynamique de probabilité de chaque événement à la base des événements qui précèdent), semi-

adaptatif (en effectuant un passage préliminaire du fichier à coder pour établir les statistiques) ou non-adaptatif (en utilisant des probabilités fixes pour tous les fichiers). les modèles non-adaptatifs sont arbitraires et souvent non-performants [43]. Le codage adaptatif est du type "un seul passage" mais nécessite une structure de données plus compliquée. En revanche le codage semi-adaptatif exige un double passage ainsi que la transmission du modèle de données comme information supplémentaire. Si la transmission du modèle est bien optimisée un codage semi-adaptatif fournit une compression meilleure que le codage adaptatif. En général, le coût de transmission du moèle semi-adaptatif est presque le même que celui de l'apprentissage du modèle adaptatif [18].

Principe de base du Codage Arithmétique

L'algorithme de codage **Arithmétique** d'une suite de données comprend les étapes suivantes:

1. On divise l'intervalle unité semi-ouvert $[0, 1[$ en M sous-intervalles qui correspondent aux M symboles possibles. La taille de chaque sous-intervalle est proportionnelle à la probabilité estimée que le symbole correspondant soit le suivant dans la suite à coder.

2. On choisit le sous-intervalle, correspondant au premier symbole de la série à coder, pour le diviser en M sous-intervalles proportionnellement aux probabilités des symboles.

3. On répète l'étape 2 jusqu'au dernier symbole dans la suite à coder.

4. On représente les limites inférieure et supérieure du sous-intervalle correspondant au dernier symbole de la suite par leurs valeurs en fraction binaire, et on choisit le nombre binaire qui distingue le dernier sous-intervalle.

Pour mieux comprendre l'algorithme prenons l'exemple suivant:
Soit l'ensemble de cinq symboles $\{a, b, c, d, e\}$ dont on connait les probabilités $\{0.12, 0.09, 0.07, 0.42, 0.3\}$, on cherche à coder la suite "**beca**" avec un codeur **Arithmétique**.
L'intervalle $[0, 1[$ est divisé en cinq sous-intervalles relativement aux probabilités des symboles $\{a, b, c, d, e\}$. Le premier symbole dans la suite étant **b**, on choisit le sous-intervalle correspondant $[0.37, 0.46[$ pour le diviser en sous-intervalles proportionnels aux probabilités. Pour le symbole suivant de la suite **e**, le sous-intervalle est restreint à $[0.3763, 0.4033[$. Au dernier symbole de la suite le sous-intervalle devient $[0.3779632, 0.37819[$ (cf. figure 2.11), dont la largeur est égale à la probabilité de "**beca**" soit $0.12 \times 0.09 \times 0.07 \times 3 = 0.0002268$. En binaire, le sous-intervalle final est $[0.0110000011000..., 0.0110000011010...[$, où particulièrement tous

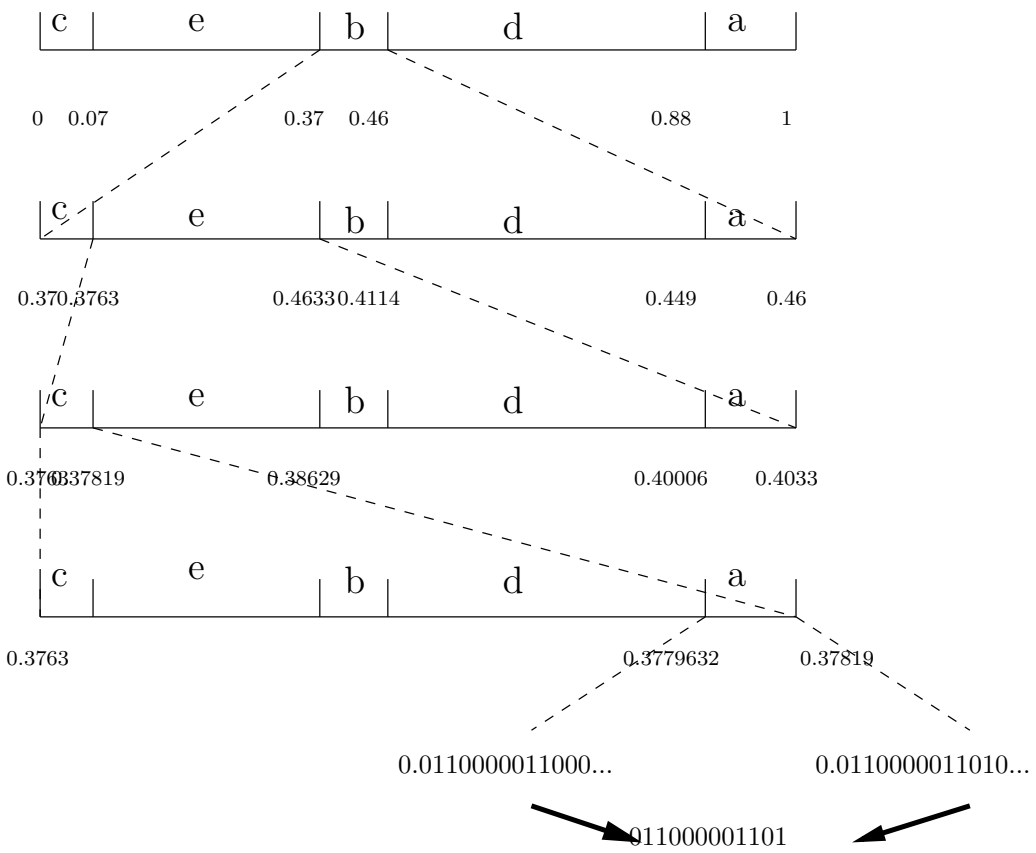

Figure 2.11: Illustration graphique du codage **Arithmétique**

le nombres binaires commençant par 0.011000001101 s'y trouvent. Ainsi pour représenter l'intervalle correspondant à la suite "**beca**", il suffit de transmettre la valeur 011000001101 sur 12bits. A titre de comparaison la suite "**beca**" sera codée sur 13bits avec un codeur de **Huffman**.

2.8 Exemples de Codeurs

2.8.1 JPEG: Norme de Compression d'Image Fixe

Beaucoup de techniques de compression avec perte existent déjà, cependant la famille des techniques de compression par transformée semble être la plus appréciée. Le meilleur exemple de cette famille est la norme $JPEG$ pour la compression des images fixes.
L'Organisation de Standard International ISO et l'UIT (Union Internationale de Télécom.) travaillaient ensemble depuis juin 1982 pour la normalisation de la compression/décompression des images fixes. Le projet n'était que le *Joint Photographic*

Expert Group JPEG. Juin 1987, on a présenté 10 techniques différentes pour la compression des images fixes en couleur ou à niveaux de gris. Après un ensemble de tests comparatifs, une technique de codage par TCD avait donné les meilleurs résultats subjectifs et c'est cette technique qui aurait été adopté pour la norme. Cette dernière portant le nom du projet $JPEG$ est devenue en 1992 une norme internationale ISO.
La norme $JPEG$ est développée pour la compression des images fixes en couleur ou à niveaux de gris. Elle ne traite pas les images binaires (noir et blanc) et est plus éfficace et performante pour les images à nuance constante et beaucoup moins pour les images à variation brusque de couleur. Les contraintes imposées sur la norme $JPEG$ sont nombreuses, de la bonne qualité d'image reconstruite à la complexité de l'algorithme raisonnable. Des contraintes relatives aux modes d'opérations ont également été ajouté. Il existe quatre modes de fonctionnement de la norme $JPEG$, le séquentiel sans perte, le séquentiel à base de la TCD, le progressif à base de la TCD et le mode hiérarchique. Pour les modes séquentiel et progressif à base de la TCD on trouve d'autres catégories (c.f. table 2.3). La différence fondamentale entre ces modes est le principe de codage avec ou sans perte.
Le mode séquentiel sans perte est la seul version de la norme $JPEG$ qui utilise un codage différentiel à la place du codage par TCD. Ce mode sans perte, bien qu'il soit simple à mettre en œuvre, permet d'avoir un taux de compression autour de $2:1$.
Dans le mode de codage séquentiel, les coefficients de l'image transformée par une TCD sont codés par un seul passage de manière à pouvoir décoder toutes les informations et reconstruire une image complète. Le mode de codage progressif effectue plusieurs passages sur les informations dans l'image transformée pour un codage par approximations succéssives ou par une sélection spectrale (les basses fréquences suivies par les hautes fréquences), ce qui permet une représentaion approximative de l'image avant la fin du décodage. Ces représentations approximatives de l'image augmentent la complexité du décodeur et réduisent même les performances du compresseur.
Dans le mode hiérarchique, on visualise progressivement l'image reconstruite partant d'une basse résolution jusqu'à la plus haute. Encore une fois, la reconstruction des images aux étapes intermédiaires du décodage bien qu'elle fournisse des bons résultats, elle réduit les performances du compresseur.

Mode Séquentiel à Base de la TCD - $JPEG$ de base

Le $JPEG$ de base est le mode séquentiel à base de la TCD. Il est le mode le plus utilisé pour la norme $JPEG$, d'ailleurs il est l'infrastructure de la norme $MPEG$.
Le principe de l'algorithme $JPEG$ pour une image à niveaux de gris (sachant qu'une image en couleur est considérée comme superposition des images de ce type) est le suivant (figure 2.12):
L'image est décomposée séquentiellement en blocs de (8×8) pixels subissant indi-

JPEG					
Séquantiel sans perte	**A base de la *TCD* avec perte**				
	Séquentiel		**Hiérarchique**	**Progressif**	
	***JPEG* de base**	**Arithmétique**		**Sélection Spectrale**	**Approximations succéssives**

Table 2.3: Les différents modes de fonctionnement de la norme *JPEG*

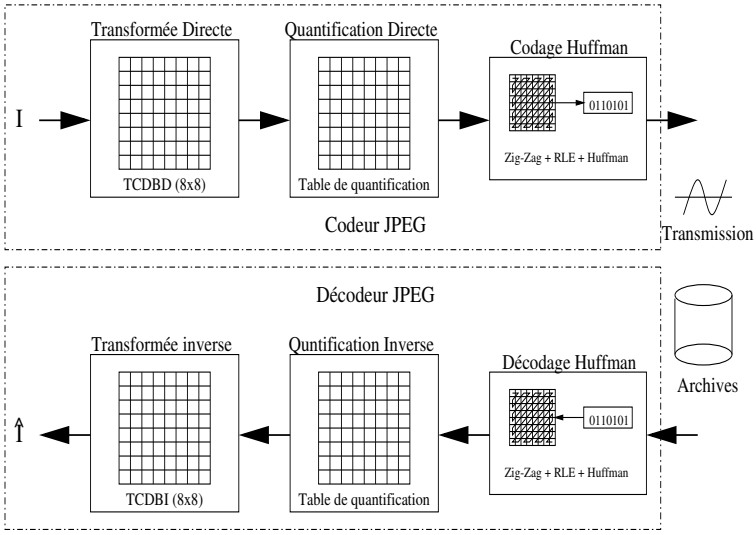

Figure 2.12: Schéma bloc du Codeur/Décodeur *JPEG*

viduellement le même traitement. Une transformée *TCD* bidimensionnelle est appliquée sur chaque bloc. Les coefficients du bloc transformé sont ensuite quantifiés en association avec une table de 64 éléments définissant les pas de quantification. Cette table permet de choisir un pas de quantification important pour certaines composantes jugées peu significatives visuellement. Dans le cas de l'image une grande partie de l'énergie est concentrée dans les fréquences spatiales les plus basses. Ainsi la quantification doit permettre de garder une certaine précision des coefficients basses fréquences et réduire celle des coefficients hautes fréquences. Les coefficients d'un bloc sont ordonnés suivant un balayage en zig-zag pour placer d'abord les coefficients correspondants aux fréquences les plus basses dans un flux de symboles. Ce dernier est ensuite traité par un codeur de longueur de plage (run-length) le codage entropique (**Huffman** ou **Arithmétique**) permettant d'utiliser les propriétés statistiques. Pour une étude détaillée de la norme *JPEG* nous conseillons les lecteurs de consulter l'ouvrage de Pennebaker & Mitchell [34].

2.8.2 MPEG et la Compression des Séquences d'Images Animées

En 1988 l'UIT et le $CCIR$ ont crée le comité commun $MPEG$ (Moving Picture Expert Group) pour la diffusion numériques d'images et du son pour la Télévision, l'enregistrement sur $CD-ROM$ de films et le multimédia. Ce groupe de travail a défini une norme internationale [2] en 1993. Actuellement la $MPEG$ consite en plusieurs normes de codage, $MPEG1$ pour les débits autour de 1.5Mbit/s, $MPEG2$ pour les débits situés entre 4 et 10Mbit/s, et la $MPEG4$ écemment normalisé.
En général, le codeur $MPEG$ sous ses différents aspects est composé de deux codeurs séparés, un pour la vidéo ($MPEG$-video) et l'autre pour l'audio($MPEG$-audio). Le $MPEG$-system décrit la synchronisation et le multiplexage des deux flux de bits compressés. Le $MPEG$ possède plusieurs caractéristiques importantes, en plus de la réduction de débit. Avec cette norme une scène filmée et compressée peut être visualisée dans les deux sens à grande ou à vitesse normale. L'accès aux images est aléatoire, ce qui permet de visualiser indépendament chaque image de la séquence comme si c'est une image fixe. En allant plus loin avec la norme $MPEG4$ et grâce à la notion d'objet, la séquence d'images est représentée comme un système d'objets indépendants et autonomes. La $MPEG4$ sera capable de couper des régions de l'image (un visage ou n'importe quel objet dans la scène) et de les compresser individuellement sans avoir à coder toute l'image.
L'approche de $MPEG$ compresse les images selon 3 modes de compression:

1. Mode Intra I (Intra-frame) où les images sont codées comme des images fixes, sans références à d'autres images. La compression dans ce mode est réalisé par la norme $JPEG$ mais en utilisant des pas de quantification adaptés au contenu de chaque macro-bloc de l'image (2×2 blocs de 8×8).

2. Mode Prédictif P (Predictive-frame) où les images sont codées en tenant compte d'une prédiction de la compensation de mouvement par rapport à l'image précédente.

3. Mode Bidirectionnel B (Bidirectional-frame) où les images sont codées en tenant compte d'une prédiction de la compensation de mouvement dans les deux sens, images précédentes et suivantes.

En effet, dans une séquence vidéo les valeurs des pixels du fond de l'image varient légèrement d'une image à une autre, à moins que la caméra soit mobile. Ainsi les images gardent un fond pratiquement invariant sur une douzaine d'images successives. Pour coder les images B et P, la $MPEG$ exploite cet avantage de redondance temporelle. Avec la compensation de mouvement qui applique une technique de "Block-Matching", le $MPEG$ calcule des vecteurs de mouvement pour les blocs des images B et P en utilisant les blocs correspondants des autres images. Ces vecteurs

de mouvement en plus des blocs compensés permettent de calculer les images estimées qui ne ressemblent pas exactement aux images actuelles. La différence entre l'image compensée et celle actuelle doit être codée comme une image intra à la façon de $JPEG$. Comme on vient de constater le $JPEG$ est bien à l'appui de la norme $MPEG$. La quantification de chaque macro-bloc (2×2 blocs de 8×8) est réalisée avec des pas de quantification obtenus par une simple multiplication par un facteur d'échelle des pas de quantification standards. Les facteurs d'échelle seront transmis au décodeur offrant ainsi une quantification plus flexible que celle dans le $JPEG$ de base, donc une compression plus performante. Actuellement une extension de la norme $JPEG$ prévoit des facteurs d'échelle pour la tables de quantification au niveau du bloc [1]. Nous reviendrons sur ce point au chapitre 5.

2.8.3 H261 et H263 Normes pour le Codage Vidéo

La $H261$ est la norme de codage vidéo spécifiée par l'$UIT - T$ en 1990. Elles est conçue pour les services audio-visuels à des débits multiples de 64Kb/s et est appelée parfois $p \times 64Kb/s$ (où $p = 1 \ldots 30$). Plus précisement, cette norme a été développée pour la transmission vidéo sur les réseaux $RNIS$. Actuellement, la $H261$ est la norme de compression vidéo la plus utilisée pour la vidéophonie (partie vidéo) sur les réseaux commutés tel que le $RNIS$ avec son accès de base ou primaire. Cette norme est utilisée conjointement avec d'autres normes de contrôle, de signalisation et de tramage ($H221$, $H230$, $H242$) à côté des normes de codage audio ($G711$, $G722$ et $G728$). La norme $H261$ spécifie les méthodes de codage et de décodage de la composante vidéo dans un service audiovisuel. Parmi ces spécifications on trouve:

- Format de l'image: Le codeur fonctionne sur des images non entrelacées qui seront codées comme une composante luminance Y et deux composantes de différences de couleurs Cr et Cb, où les matrices Cr et Cb sont le quart en dimension de la matrice Y. La $H261$ supporte deux définitions d'images $QCIF$ (144×176 pixels) et CIF (288×352 pixels en PAL).

- Les composantes du codeur source:

  1. Les Prédictions: La $H261$ définit comme en $MPEG$ deux types de codage, $INTRA$ où l'image est codée toute entière sans référence à une autre image et le codage $INTER$ où on code la différence d'une image avec sa prédiction. La prédiction d'une image est calculée comme en $MPEG$ après une recherche de ressemblence de chaque bloc de (8×8) dans une région de (16×16) (Macro-bloc). La compensation de mouvement dans le codeurs $H261$ n'est qu'une option, la norme ne définit ni la zone de recherche ni même la façon de calculer les vecteurs de mouvement. Cependant, elle précise que les composantes horizontale et verticale de ces vecteurs doivent être des valeurs entières entre $+15$ et -15.

2. Transformée des blocs: Une TCD bidimensionnelle comme celle utilisée dans $JPEG$ et $MPEG$ est employée dans la $H261$ pour transformer les blocs de (8×8) pixels en 64 composantes de fréquence.
3. Quantification: Dans cette opération, on cherche à allouer les bits disponibles aux composantes de le TCD $2D$ les plus significatives, suivant des critères psychovisuels. Le nombre de niveaux de quantification est de 1 pour les coefficients DC et 31 pour les AC.
4. Codage entropique: Le codage de Huffman est adopté pour réaliser cette étape.
5. Multiplexage: Le multiplexeur vidéo structure les données en un flux de bit hiérarchique universel. Cette structure est composée de quatre couches:

   * Couche Image (Picture Layer) qui corespond à une image vidéo;

   * Groupe de blocs qui correspond à 1/12 d'une image CIF et 1/3 d'une image $QCIF$;

   * Macro-bloc qui représente (2×2) blocs de (8×8) pixels de luminance et 2 blocs de (8×8) de composantes chromatiques correspondantes.

   * Bloc, ou l'ensemble de (8×8) pixels.
6. Le tramage de correction d'erreur: le contrôle de parité BCH (511,493) est utilisé pour protéger le flux de bits transmis sur le $RNIS$, il est optionel pour les décodeurs.

La norme $H261$ est bien adaptée pour les applications de visioconférence et la vidéophonie où l'image est relativement d'une faible définition. Toutefois elle prévoit dans son "Annexe D" la possibilité de transmettre des images fixes d'une définition 4 fois celle du CIF dans un flux vidéo. La norme H261 est bien détaillée dans [3].

H263 La Nouvelle Norme de Compression Vidéo à Faible Débit

Cette norme [4] a été conçue principalement pour des services audiovisuels à bas débit mais elle peut être employée pour des débits pouvant atteindre ceux de la $H261$ ($p \times 64Kb/s$ où $p = 1 \ldots 30$). Même si les grandes lignes de l'algorithme $H263$ ressemblent à celles de la $H261$, les modifications apportées sur la $H263$ permettent nettement d'améliorer la correction d'erreur et ainsi ses performances.
Les principales différences entre la $H261$ et la $H263$ sont:

- La compensation de mouvement est réalisée avec une précision d'1/2 pixel pour la $H263$ contre 1 pixel pour le $H261$.
- Quelques éléments de la structure hiérarchique sont optionnels pour la $H263$ permettant ainsi au codec de travailler à des débits plus bas et un recouvrement d'erreur plus éfficace.

- Quatre autres options sont incluses pour améliorer la performance:

  1. les vecteurs de mouvement non-restreints qui peuvent représenter des blocs en dehors de l'image et où les pixels de bords sont utilisés pour la prédiction des pixels inexistants. En cas de mouvement sur les bords de l'image (surtout pour les petites définitions) cette option réalise un gain significatif;
  2. le codage **Arithmétique** utilisé au lieu du codage à longueur variable permet de gagner en débit au voisinage de 10% sur les images $INTRA$ pour un même rapport Signal/Bruit SNR;
  3. une prédiction avancée, où on utilise une compensation de mouvement avec chevauchement de blocs. Pour un certain macro-bloc, un ensemble de 4 vecteurs de (8×8) est utilisé au lieu d'un seul vecteur de (16×16) et le codeur aura le choix entre l'ancienne et la nouvelle technique de prédiction. La prédiction avancée génère plus de débit mais améliore considérablement la qualité surtout que la compensation de mouvement avec chevauchement de blocs (Overlapped Block Motion Compensation $OBMC$) réduit subjectivement les effets de blocs;
  4. une prédiction bidirectionnelle (comme celle observée dans la norme $MPEG$) pour calculer les images prédictibles P et Bidirectionnelles B.

  Avec ces 4 options, la $H263$ génère la moitié du débit pour une même qualité d'image que la $H261$.

- La $H263$ supporte 5 définitions de l'image source. En plus de CIF et $QCIF$, on trouve $SQCIF = 1/2QCIF$, $4CIF$ et $16CIF$. Les deux dernières résolutions prouvent la puissance du codeur $H263$ à défier d'autres normes de codage vidéo à haut débit telle que la $MPEG$.

Chapitre 3

Codeur JPEG à base d'une transformée TCD

Dans le chapitre 2, nous avons décrit l'algorithme $JPEG$ pour le codage d'image fixe. Cet algorithme est basé sur une transformée de type $TCDB$ appliquée sur les blocs d'image de (8×8) pixels. Dans ce chapitre on montrera que la $TCDB^{-1}$ au décodeur n'est pas optimale en présence de bruit de quantification alors qu'elle l'est en absence de toute de quantification. On introduira une nouvelle transformée de synthèse conçue pour réduire la distorsion dûe aux bruits de quantification. Cette optimisation étant un problème bien connu en traitement de signal, elle nous a conduit au calcul de filtres de Wiener pour des signaux scalaires tels que l'intensité des pixels de l'image. Dans le paragraphe 3.5.1 nous présentons le développement sous forme matricilelle des filtres de Wiener.

3.1 La TCD une Transformée linéaire orthogonale

Le passage par une transformée directe n'est qu'une étape préparative nécessaire pour la compression. La transformée inverse ou généralement de synthèse est indispensable pour retrouver les échantillons dans leur domaine d'origine (intensité de pixel). La TCD utilisée est inversible, la matrice associée à la transformée inverse vérifie la relation

$$A^{-1}A = I$$

où A est la matrice de TCD directe, or le calcul montre que

$$A^t A = I \tag{3.1}$$

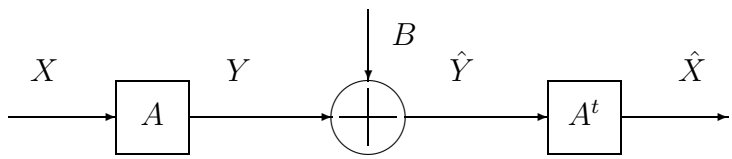

Figure 3.1: Schéma de Codage/Décodage par TCD

ce qui signifie que l'inversion de A est obtenue facilement par une simple transposition. Les matrices qui vérifient la relation (3.1) sont dites unitaires et orthogonales dans le cas de la TCD. Ces matrices sont intéressantes pour les propriétés suivantes:

- Une inversion simple qui se réalise par une transposition

$$A^{-1} = A^t$$

- L'erreur de reconstruction ou la distorsion mesurée par l'Erreur Quadratique Moyenne EQM (MSE) est égale à l'erreur de quantification du signal transformé.

Reprenons le schéma d'un codeur par TCD.

$$X(m) = \begin{bmatrix} x_0(m) \\ \vdots \\ x_{N-1}(m) \end{bmatrix}$$

est le signal d'entrée,

$$Y(m) = \begin{bmatrix} y_0(m) \\ \vdots \\ y_{N-1}(m) \end{bmatrix} = A \begin{bmatrix} x_0(m) \\ \vdots \\ x_{N-1}(m) \end{bmatrix}$$

le signal transformé et

$$B(m) = \begin{bmatrix} b_0(m) \\ \vdots \\ b_{N-1}(m) \end{bmatrix}$$

est le bruit ou l'erreur de quantification qu'on suppose additive par la suite.

$$N = 64 = 8 \times 8$$

Les coefficients $y_k(m)$ représentent les échantillons du processus aléatoire associé à la k^{ieme} des N sous-bandes.

$$\hat{X}(m) = A^{-1}\hat{Y}(m)$$

où $\hat{Y}(m)$ est le signal transformé et quantifié, et $\hat{X}(m)$ est la reconstruction par la transformée de synthèse.

$$\begin{aligned}
MSE &= \frac{1}{N}\sum_{k=0}^{N-1} E\left[(x_k(m)-\hat{x}_k(m))^2\right] \\
&= \frac{1}{N}E\left[(X(m)-\hat{X}(m))^t(X(m)-\hat{X}(m))\right] \\
&= \frac{1}{N}E\left[(A^{-1}A(X(m)-\hat{X}(m)))^t(A^{-1}A(X(m)-\hat{X}(m)))\right] \\
&= \frac{1}{N}E\left[(A^{-1}(Y(m)-AA^{-1}\hat{Y}(m)))^t(A^{-1}(Y(m)-AA^{-1}\hat{Y}(m)))\right] \\
&= \frac{1}{N}E\left[(Y(m)-\hat{Y}(m))^tAA^{-1}(Y(m)-\hat{Y}(m))\right] \\
&= \frac{1}{N}E\left[(Y(m)-\hat{Y}(m))^t(Y(m)-\hat{Y}(m))\right] \\
&= \frac{1}{N}\sum_{k=0}^{N-1} E\left[(y_k(m)-\hat{y}_k(m))^2\right] \\
&= \frac{1}{N}\sum_{k=0}^{N-1} E\left[(b_k(m))^2\right] \\
&= \sigma_Q^2 \qquad (3.2)
\end{aligned}$$

L'erreur de reconstruction est égale à l'erreur de quantification.

3.2 La Compression Sans Perte en Absence de Toute Quantification

En absence de la quantification l'équation (3.2) montre que l'erreur de reconstruction est nulle, ce qui signifie la reconstruction parfaite des blocs de l'image au décodeur. Comme on vient de voir dans le chapitre 2, le codage entropique effectué sur les symboles formés de longueur de plage et d'amplitude peut être vu comme une compression sans perte, réversible. Ceci est réaliste du fait que si les transformées d'analyse et de synthèse se neutralisent, comme c'est le cas avec la TCD, un codage entropique comme celui de **Huffman** précédé d'un codage à plage de zéro ou par répétition est complètement réversible aussi. La reconstruction sera *a priori* sans erreur (sauf erreur de sauvegarde ou du canal de transmission). Le taux de compression sans perte ne dépasse pas le rapport 3:1 et une telle compression ne servira pas beaucoups à l'enregistrement ni à la transmission d'image à faible débit (l'objectif de cette thèse).

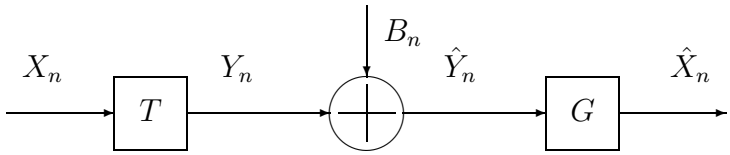

Figure 3.2: Codage par transformée

3.3 La Compression Avec Perte en Présence de Bruit de Quantification

Ce genre de compression réaliste est le plus répandu, malgré les erreurs de reconstruction ou la perte des détails dans l'image. En effet ces techniques de compression sont les bienvenues dans les applications multimédia pour lesquelles beaucoup des travaux sont menés afin de trouver des compromis entre la qualité d'image et le débit (ou la taille du fichier image).
Pour la compression avec perte, on utilise tous les éléments de la chaîne de codage y compris la quantification, source de l'erreur de reconstruction. Prenons le cas d'un codeur $JPEG$ à base de $TCDB$, la $TCDB$ permet une reconstruction parfaite en absence des erreurs de quantification. Cependant la transformée inverse $TCDB^{-1}$ est sous-optimale pour les coefficients quantifiés et l'erreur de quantification se traduit par une dégradation de la qualité facilement visible à bas débits.
Dans le paragraphe 3.1 on vient de voir que l'erreur de reconstruction au décodeur est égale à celle de quantification lorsqu'une transformée orthogonale telle que la $TCBD$ est utilisée à l'analyse et à la synthèse. L'équation (3.2) montre que la distorsion mesurée par l'EQM est additive sur l'ensemble des 64 sous-bandes de la $TCDB$. Cette propriété d'additivité de la distorsion est directement liée à l'orthogonalité des transformées d'analyse et de synthèse.

3.3.1 Formulation

Considérons le schéma ci-dessus qui représente un codeur source où $X(n)$ est un bloc de $N \times N$ pixels de l'image originale, T la matrice associée à la transformée d'analyse et G la matrice de la transformée de synthèse. Dans la suite $B(n)$ le bruit de quantification est supposé additif (quantification scalaire uniforme comme c'est le cas dans le $JPEG$). $Y(n)$ est le bloc des coefficients transformés et $\hat{Y}(n)$ est le bloc quantifié correspondant.

$$Y(n) = TX(n); \;\; \hat{Y}(n) = Y(n) + B(n)$$

Soit $\hat{X}(n)$ le bloc de l'image reconstruite.

$$\hat{X}(n) = G\hat{Y}(n)$$

Pour des raisons de simplification nous considérons chaque bloc de $N \times N$ comme un vecteur linéaire de N^2 éléments, ainsi les matrices de transformations T et G sont linéaires de dimensions $N^2 \times N^2$.
La distorsion entre $\hat{X}(n)$ et $X(n)$ (dans le sens de l'EQM) sous forme matricielle est donnée par

$$d = (\hat{X}(n) - X(n))^t(\hat{X}(n) - X(n))/N^2$$

alors que la distorsion globale est

$$D = trace(E\left[(\hat{X}(n) - X(n))(\hat{X}(n) - X(n))^t\right])/N^2$$

$$E\left[(\hat{X}(n) - X(n))(\hat{X}(n) - X(n))^t\right] = \\ E[((GT - I)X(n) + GB(n))((GT - I)X(n) + GB(n))^t]$$

Dans le cas de la TCD

$$G = T^{-1} = T^t$$

la distorsion globale devient

$$D = trace[GR_{BB}G^t]/N^2$$

où R_{BB} est la matrice d'autocorrelation du bruit de quantification. Ainsi la distorsion qui dépend directement de l'énergie du bruit de quantification n'atteint pas sa valeur minimale en utilisant la TCD^{-1} à la synthèse car cette énergie est directement liée aux quantificateurs des sous-bandes. Cette équation de la distorsion permet de se poser la question sur l'optimalité de la TCD^{-1} et la possibilité de trouver une autre transformée de synthèse optimale, permettant de minimiser la valeur de la distorsion.

3.4 TCD^{-1}, Transformée de Synthèse Optimale pour la Compression avec Perte?

Dans cette section, nous supposons que les pas de quantification correspondants aux sous-bandes sont fixes, ce qui permet de définir un débit constant. Nous choisissons la minimisation de la distorsion comme un critère pour calculer la transformée de synthèse optimale. Lorsqu'une TCD^{-1} est utilisée pour reconstruire un bloc codé, cette transformée ne tient pas compte de l'erreur de quantification ajoutée à l'ensemble de coefficients de ce bloc. Dans ce cas la reconstruction est une opération non-optimisée dans le contexte de codage.

Alors si la TCD^{-1} n'est plus la transformée de synthèse optimale, comment peut-on en calculer une et pour quel critère ?

L'optimisation de la synthèse dans un schéma de codage en présence de quantification est initialement étudiée par Dembo et Malah [8]. Dans leurs travaux, ils utilisent un modèle de bruit de quantification pour calculer un banc de filtres de synthèse. Haddad et al. [44, 38] traitent le problème d'optimisation en fixant un niveau de bruit de quantification. L'innovation dans leurs travaux demeure dans l'emploi d'une matrice compensatrice diagonale, dans le banc de filtres. Les coefficients de cette matrice sont ajoutés pour réduire l'erreur de reconstruction. Dans la même période Kovačević [27] procède de la même façon pour optimiser cette matrice de compensation.
Orchard dans [32] montre le calcul de la transformée de synthèse comme un problème d'optimisation pour un critère de concentration d'énergie. La matrice associée à la transformation linéaire optimisée est considérée comme un estimateur linéaire optimal d'une structure indépendante de la transformée d'analyse et non-orthogonale à la matrice associée à cette dernière. Gosse et Duhamel [14, 25] optimisent conjointement, dans un algorithme itératif, le banc de filtres de synthèse et les quantificateurs, pour un critère de minimisation de la distorsion sous une contrainte de débit binaire total donné. Dans [42] les auteurs proposent pour le même contexte des versions modulées d'un filtre prototype pour simplifier l'implantation du banc de filtres. Ils définissent en plus une variable de contrôle de gain *(gain control factor)* pour chaque filtre.
Une partie du travail de cette thèse est la suite des travaux de K.Gosse et P.Duhamel pour une application au codage d'image. Dans la suite de ce chapitre nous traitons le problème d'optimisation de la synthèse laissant le problème d'optimisation des pas de quantification pour le chapitre suivant.

3.5 Présentation d'une Transformée de Synthèse Optimale

Le contexte de notre étude est très proche de la norme $JPEG$, au moins pour le codeur. Pour l'instant nous supposons que les pas de quantification et les filtres d'analyse sont fixés, nous optimisons dans le décodeur le banc de filtres de synthèse qui minimise le critère exprimé par l'Erreur Quadratique Moyenne EQM (MSE) pour les blocs de l'image reconstruite par ce banc de filtres optimisé.

3.5.1 Les Filtres de Wiener

Reprenons l'équation de la distorsion calculée sur l'ensemble des blocs de $N \times N$ pixels de l'image reconstruite

$$D = trace(E\left[(\hat{X}(n) - X(n))(\hat{X}(n) - X(n))^t\right])/N^2 \tag{3.3}$$

$$\begin{aligned}
E[(\hat{X}(n)-X(n))(\hat{X}(n)-X(n))^t] &= E[((GT-I)X(n)+GB(n))((GT-I)X(n)+ \\
&\quad GB(n))^t] \\
&= (GT-I)E[X(n)X^t(n)](GT-I)^t + \\
&\quad (GT-I)E[X(n)X^t(n)]G^t + \\
&\quad GE[B(n)X^t(n)](GT-I)^t + \\
&\quad GE[B(n)B^t(n)]G^t \\
&= GTR_{XX}T^tG^t - GTR_{XX} - R_{XX}T^tG^t + \\
&\quad R_{xx} + GTR_{XB}G^t - R_{XB}G^t + GR_{XB}^tT^tG^t - \\
&\quad GR_{XB}^t + GR_{BB}G^t \qquad (3.4)
\end{aligned}$$

où R_{XX} et R_{BB} sont respectivement la matrice d'autocorrelation des blocs de l'image originale et celle des blocs de bruits de quantification. R_{XB} est la matrice de correlation des blocs de l'image et de bruit de quantification.

$$R_{XB} = E[X(n)B^t(n)] = (E[B(n)X^t(n)])^t = R_{BX}^t$$

$R_{\hat{Y}\hat{Y}}$ étant la matrice d'autocorrelation des blocs quantifiés, elle est définie par

$$\begin{aligned}
R_{\hat{Y}\hat{Y}} &= E[\hat{Y}(n)\hat{Y}^t(n)] \\
&= E[(TX(n)+B(n))(TX(n)+B(n))^t] \\
&= TR_{XX}T^t + R_{BX}T^t + TR_{XB} + R_{BB}
\end{aligned}$$

Cette matrice étant symétrique, elle est diagonalisable et *a priori* souvent inversible. L'équation (3.4) devient

$$\begin{aligned}
E\left[(\hat{X}(n)-X(n))(\hat{X}(n)-X(n))^t\right] &= G(TR_{XX}T^t + TR_{XB} + R_{XB}^tT^t + R_{BB})G^t - \\
&\quad GR_{\hat{Y}\hat{Y}}R_{\hat{Y}\hat{Y}}^{-1}TR_{XX} - R_{XX}T^tR_{\hat{Y}\hat{Y}}^{-1}R_{\hat{Y}\hat{Y}}G^t - \\
&\quad R_{XB}R_{\hat{Y}\hat{Y}}^{-1}R_{\hat{Y}\hat{Y}}G^t + R_{XX}T^tR_{\hat{Y}\hat{Y}}^{-1}TR_{XX} - \\
&\quad R_{XX}T^tR_{\hat{Y}\hat{Y}}^{-1}TR_{XX} + R_{XX} - \\
&\quad GR_{\hat{Y}\hat{Y}}R_{\hat{Y}\hat{Y}}^{-1}R_{XB}^t \\
&= GR_{\hat{Y}\hat{Y}}G^t - GR_{\hat{Y}\hat{Y}}R_{\hat{Y}\hat{Y}}^{-1}TR_{XX} - \\
&\quad R_{XX}T^tR_{\hat{Y}\hat{Y}}^{-1}R_{\hat{Y}\hat{Y}}G^t - R_{XB}R_{\hat{Y}\hat{Y}}^{-1}R_{\hat{Y}\hat{Y}}G^t + \\
&\quad R_{XX}T^tR_{\hat{Y}\hat{Y}}^{-1}TR_{XX} - R_{XX}T^tR_{\hat{Y}\hat{Y}}^{-1}TR_{XX} + \\
&\quad R_{XX} - GR_{\hat{Y}\hat{Y}}R_{\hat{Y}\hat{Y}}^{-1}R_{XB}^t
\end{aligned}$$

$$E\left[(\hat{X}(n)-X(n))(\hat{X}(n)-X(n))^t\right] = GR_{\hat{Y}\hat{Y}}(G^t - R_{\hat{Y}\hat{Y}}^{-1}TR_{XX} - R_{\hat{Y}\hat{Y}}^{-1}R_{XB}^t) -$$

$$
\begin{aligned}
& R_{XX}T^tR_{\hat{Y}\hat{Y}}^{-1}(R_{\hat{Y}\hat{Y}}G^t - TR_{XX}) + R_{XX}(I - \\
& T^tR_{\hat{Y}\hat{Y}}^{-1}TR_{XX}) - R_{XB}R_{\hat{Y}\hat{Y}}^{-1}R_{\hat{Y}\hat{Y}}G^t - \\
& R_{XX}T^tR_{\hat{Y}\hat{Y}}^{-1}R_{XB}^t + R_{XX}T^tR_{\hat{Y}\hat{Y}}^{-1}R_{XB}^t + \\
& R_{XB}R_{\hat{Y}\hat{Y}}^{-1}R_{XB}^t - R_{XB}R_{\hat{Y}\hat{Y}}^{-1}R_{XB}^t + \\
& R_{XB}R_{\hat{Y}\hat{Y}}^{-1}TR_{XX} - R_{XB}R_{\hat{Y}\hat{Y}}^{-1}TR_{XX} \\
= \; & (G - R_{XX}T^tR_{\hat{Y}\hat{Y}}^{-1} - R_{XB}R_{\hat{Y}\hat{Y}}^{-1})R_{\hat{Y}\hat{Y}}(G^t - \\
& R_{\hat{Y}\hat{Y}}^{-1}R_{XB}^t - R_{\hat{Y}\hat{Y}}^{-1}TR_{XX}) + R_{XX}(I - \\
& T^tR_{\hat{Y}\hat{Y}}^{-1}TR_{XX} - T^tR_{\hat{Y}\hat{Y}}^{-1}R_{XB}^t) - \\
& R_{XB}R_{\hat{Y}\hat{Y}}^{-1}(TR_{XX} + R_{XB}^t) \qquad (3.5)
\end{aligned}
$$

Avec les relations (3.3) et (3.5) la distorsion devient

$$
\begin{aligned}
D \;=\; & \frac{1}{N^2}[\sum_{i=0}^{N^2-1}(G^t - R_{\hat{Y}\hat{Y}}^{-1}R_{XB}^t - R_{\hat{Y}\hat{Y}}^{-1}TR_{XX})_iR_{\hat{Y}\hat{Y}}(G - R_{XX}T^tR_{\hat{Y}\hat{Y}}^{-1} - R_{XB}R_{\hat{Y}\hat{Y}}^{-1})_i + \\
& trace(R_{XX}(I - T^tR_{\hat{Y}\hat{Y}}^{-1}TR_{XX} - T^tR_{\hat{Y}\hat{Y}}^{-1}R_{XB}^t) - R_{XB}R_{\hat{Y}\hat{Y}}^{-1}(TR_{XX} + RXB^t))]
\end{aligned}
$$

où l'indice i représente l'index des vecteurs de composantes polyphases. La transformée de synthèse optimale correspondant à une distorsion minimale est obtenue en annulant la dérivée de D par rapport aux composantes polyphases G_i de la matrice G. Ainsi pour

$$\frac{\partial D}{\partial G_i} = 0$$

on obtient

$$G_i = (R_{XX}T^tR_{\hat{Y}\hat{Y}}^{-1} + R_{XB}R_{\hat{Y}\hat{Y}}^{-1})_i$$

et

$$G = R_{XX}T^tR_{\hat{Y}\hat{Y}}^{-1} + R_{XB}R_{\hat{Y}\hat{Y}}^{-1} = T^tR_{Y\hat{Y}}R_{\hat{Y}\hat{Y}}^{-1} \qquad (3.6)$$

Cette relation montre que la matrice associée à la transformée de synthèse dépend du bruit de quantification, ainsi que de la correlation des blocs de l'image et les blocs correspondants du bruit de quantification.

N.B.: L'équation (3.6) exige l'inversion de la matrice d'autocorrelaion $R_{\hat{Y}\hat{Y}}$. Or pour des pas de quantifications assez élevés, parfois cette matrice n'est plus inversible telle qu'elle est (matrice singulière). Nous avons procédé à une technique qui n'affecte pas la solution finale mais qui permet de régler ce problème de calcul et de diminuer le débit nécessaire pour la transmission de la matrice associée à la transformée de synthèse optimale. En effet sur les N^2 sous-bandes il ne sera pas nécessaire de calculer les filtres de synthèse correspondants aux sous-bandes à coefficients nuls, on pourra utiliser les filtres de la $TCDB^{-1}$ correspondants à ces sous-bandes sans avoir besoin de les transmettre (il sera juste nécessaire de transmettre l'indice des sous-bandes à coefficients nuls). Par suite on calcule l'inverse d'une sous matrice (non-singulière) de la matrice $R_{\hat{Y}\hat{Y}}$.

	$PSNR$(dB)			
	TCD^{-1} ($JPEG$)		$MMSE$ (**Wiener**)	
débit (bpp)	Lena	Barbara	Lena	Barbara
0.15	27.11	24.14	27.47	24.90
0.25	29.70	25.97	30.01	26.64
0.50	32.79	30.14	33.08	30.94
0.75	34.43	32.19	34.72	32.87
1.00	35.65	34.45	35.80	34.99

Table 3.1: Comparaison des performances du banc de filtres Wiener ($MMSE$) et de la TCD^{-1} pour les images "Lena" et Barbara.

3.5.2 Performances de la MMSE par rapport à la TCD classique

Exemples de résultats de codage

Pour évaluer les performances de la $MMSE$ par rapport à la $TCDB^{-1}$ nous avons calculé pour différentes tables de quantification un banc de filtres de Wiener à la synthèse ($MMSE$). Les images que l'on a utilisées pour les simulations sont les images standard $JPEG$: LENA et GOLDHILL de taille 512×512, BARBARA, BOATS et ZELDA de taille 576×720, et DESK de taille 256×256. Toutes ces images test sont initialement codées à 8 bits par pixel (bpp). L'évaluation numérique des performances du banc de filtres de Wiener est représentée par les courbes du $PSNR$ calculé pour l'image décodée avec la $MMSE$ et la TCD^{-1} ($JPEG$) en fonction du nombre de bpp nécessaire, après codage de **Huffman**, pour coder entièrement l'image.
Nous trouvons dans les tables (3.1, 3.2 et 3.3) les résultats de codage des images citées ci-dessus pour des débits différents. Nous constatons (c.f. table 3.1) que le calcul du banc de filtres de Wiener ($MMSE$) améliore le $PSNR$ de 0.15 à 0.80 dB selon le débit. Pour les images où les basses fréquences dominent, telles que LENA, BOATS, Goldhill et ZELDA, la performance des filtres de Wiener diminue quand le débit se rapproche de 1bpp. Les figures 3.3 et 3.4 représentent les courbes débit/$PSNR$ obtenues avec plusieurs tables de quantification pour les images LENA et BARBARA.

3.5.3 Conclusions

Dans ce chapitre, nous avons proposé une technique pour calculer une transformée de synthèse optimale selon le principe des filtres matriciels de Wiener ($MMSE$) qui remplace la Transformée en Cosinus Discrète inverse (TCD^{-1}) dans le décodeur $JPEG$. Ce banc de filtres de synthèse optimal permet d'améliorer le $PSNR$ calculé

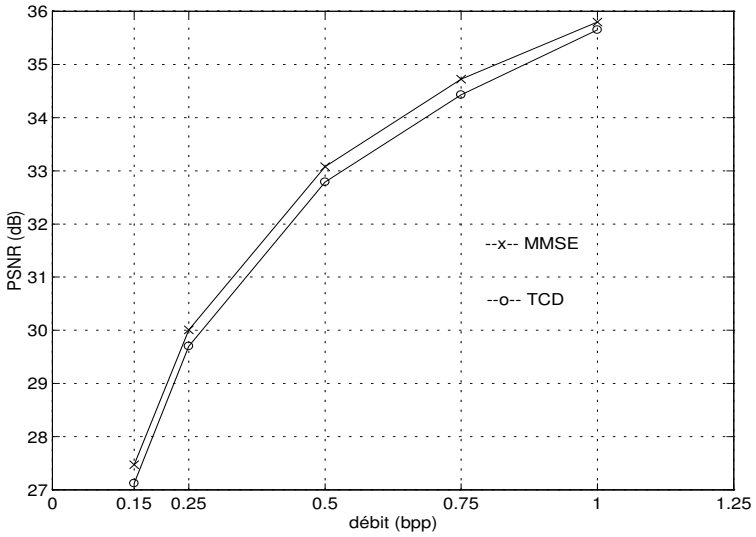

Figure 3.3: Rapport Signal/Bruit PSNR (dB) en sortie des décodeurs (TCD^{-1} et $MMSE$) en fonction du débit total alloué (bpp) de l'image "Lena".

	$PSNR$**(dB)**			
	TCD^{-1} $(JPEG)$		$MMSE$ **(Wiener)**	
débit (bpp)	Goldhill	Boats	Goldhill	Boats
0.15	27.25	28.06	27.55	28.50
0.25	29.14	30.67	29.44	31.07
0.50	31.72	34.45	32.00	34.77
0.75	33.22	36.94	33.51	37.18
1.00	34.55	38.60	34.73	38.66

Table 3.2: Comparaison des performances du banc de filtres Wiener ($MMSE$) et de la TCD^{-1} pour les images Goldhill et Boats.

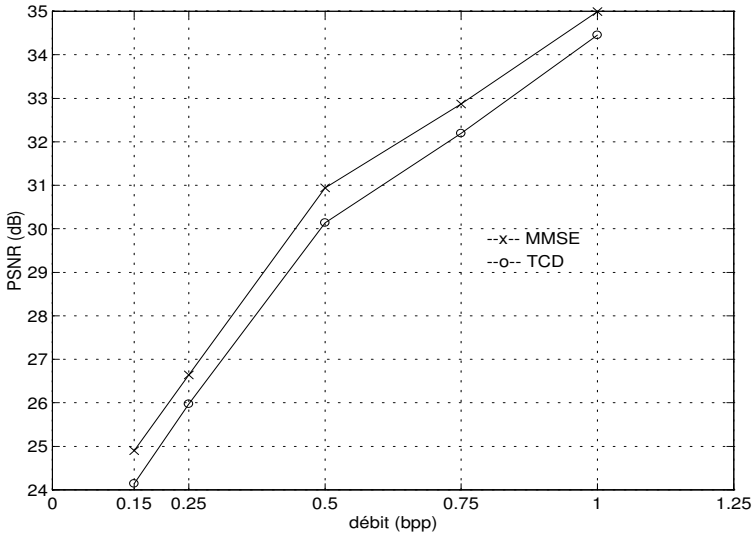

Figure 3.4: Rapport Signal/Bruit PSNR (dB) en sortie des décodeurs (TCD^{-1} et $MMSE$) en fonction du débit total alloué (bpp) de l'image "Barbara".

	$PSNR$**(dB)**			
	TCD^{-1} $(JPEG)$		$MMSE$ **(Wiener)**	
débit (bpp)	Desk	Zelda	Desk	Zelda
0.10	-	30.34	-	30.77
0.25	25.45	35.25	26.02	35.64
0.50	31.09	38.42	31.77	38.52
0.75	34.31	39.85	35.19	39.85
1.00	36.88	-	37.62	-

Table 3.3: Comparaison des performances du banc de filtres Wiener ($MMSE$) et de la TCD^{-1} pour les images Desk et Zelda.

Figure 3.5: Images originales LENA(512×512) et BARBARA(576×720)

à la reconstruction par rapport à celui donné par le $JPEG$ de base. La $TCDB^{-1}$ dans un décodeur $JPEG$ est à Reconstruction Parfaite **RP** en absence de toute quantification. En présence de la quantification, la $TCDB^{-1}$ ne tient pas compte de l'erreur dûe à cette quantification. Une connaissance sur les statistiques de l'image et l'erreur de quantification nous permet de bien optimiser la synthèse en atténuant les erreurs de reconstruction. Pour tirer profit de cette technique d'optimisation nous verrons dans le chapitre 5 comment l'utiliser dans un algorithme itératif conjointement avec une allocation de bit optimale.

Figure 3.6: Images originales GOLDHILL(512×512) et BOATS(576×720)

Figure 3.7: Image originales DESK(256×256) et ZELDA(576×720)

Chapitre 4

Les Quantificateurs optimaux et l'Allocation de bits

Dans le contexte général de cette thèse on s'est fixé le but d'enregistrement des images sur des supports à faibles et moyennes capacités, tout en améliorant la qualité des images décodées. Dans le chapitre précédent, on a calculé une transformée de synthèse optimale dans le sens de la $MMSE$, qui remplace la TCD^{-1} dans un décodeur $JPEG$ de base. Pour l'optimisation du banc de filtres de synthèse on supposait fixes les quantificateurs des coefficients de la TCD, ce qui fournissait un débit total inférieur ou égal à une limite donnée. En effet, la valeur du débit maximal est fonction de la capacité totale du support d'enregistrement ou de sauvegarde. Le choix des pas de quantification est primordial pour définir la niveau de qualité des images décodées. Une approche logique pour améliorer la performance du codeur et la qualité des images est l'optimisation des pas de quantification en fonction des statistiques de l'image à coder.
Dans ce chapitre nous traitons le problème d'optimisation des pas de quantification encore appelé allocation optimale de bits, pour une TCD^{-1} et un banc de filtres de synthèse optimal (Wiener).

4.1 Travaux Liés au Problème d'Allocation de Bits pour JPEG

Dans l'algorithme $JPEG$ de base, les tables de quantification proposées sont générées d'une façon empirique pour répondre à des critères psycho-visuels sans prendre en considération le compromis débit/distorsion. Elles affectent directement la performance de la compression qui devient sous-optimale. Le besoin d'optimiser les pas de quantification pour un débit fixe a été traité par Gersho et Wu [46]. Ils

proposent un algorithme recursif pour générer la table de quantification associée à une image codée avec le $JPEG$ de base sous contrainte de débit fixe. Un autre travail mené par Hung et Meng [20] consiste à modéliser la fonction de distribution des coefficients TCD par une fonction Laplacienne et à trouver une solution optimale locale individuellement pour chaque pas de quantification. Cette technique novatrice impose un modèle statistique aux coefficients TCD ce qui n'est pas toujours valable en réalité. En revenant sur l'approche de Gersho et Wu, cette technique cherche pour un compromis débit/distorsion basé sur les statistiques réelles de l'image à coder, les pas de quantification optimaux adaptés au contenu de l'image. En partant d'une table de quantification avec des grands pas de quantification correspondant à une grande distorsion et bas débit, l'algorithme diminue à chaque fois la valeur d'un seul pas de quantification jusqu'à atteindre le débit demandé. Dans chaque itération l'objectif est de trouver le meilleur compromis débit/distorsion pour une diminution du débit total. Cette approche semble être une version généralisée de l'algorithme d'allocation de bit proposé à l'origine par Shoham et Gersho [13].
Ramchandran et Vetterli [45] ont travaillé sur le problème d'allocation optimale de bit pour un codeur à base de paquets d'ondelettes. Leur algorithme est aussi inspiré des travaux de Shoham et Gersho. Basé sur la propriété d'additivité de la distorsion et du débit sur l'ensemble des sous-bandes, cet algorithme bénéficie de l'orthogonalité des bancs de filtres d'analyse et de synthèse. Le principe de cette technique consiste à choisir, parmi un ensemble de quantificateurs q_i à déterminer au préalable pour chaque sous-image, ceux qui résolvent le problème de minimisation

$$min(D = \sum_i D_i) \tag{4.1}$$

sous la contrainte d'obtenir un débit total inférieur ou égal à une valeur fixe donnée R_d

$$R = \sum_i R_i \leq R_d \tag{4.2}$$

4.2 Problème de Minimisation Sous Contrainte - Problème Sans Contrainte

Le problème de minimisation de la distorsion globale sous la contrainte de débit total inférieur ou égal à une valeur donnée peut être résolu plus facilement si on le convertit en un problème sans contrainte utilisant les multiplicateurs de Lagrange.
Sachant que si nous nous plaçons dans le cas de la TCD^{-1} à la synthèse on garde la propriété d'orthogonalité des transformées d'analyse et de synthèse, ainsi que l'additivité de la distorsion et du débit sur l'ensemble des sous-bandes pour des blocs de $N \times N$ pixels. Comme on vient de le dire on convertira le problème de minimisation de (4.1) sous la contrainte (4.2) en un problème plus simple sans

contrainte, en fusionant le débit R et la distorsion D avec le multiplicateur de Lagrange λ dans la fonction coût

$$J(\lambda) = D + \lambda R$$

Le problème se réduit à une simple minimisation de la nouvelle fonction $J(\lambda)$

4.2.1 Le Multiplicateur de Lagrange, Fonction du Coût à Minimiser

Everett [9] fut le premier à utiliser le multiplicateur de Lagrange λ pour résoudre le problème d'allocation optimale de ressources. Soit Q une table de quantification appartenant à l'ensemble S des tables pour les N^2 sous-bandes, pour tout $\lambda \geq 0$ la solution $Q^*(\lambda)$ du problème de minimisation de l'équation (4.1) sous la contrainte (4.2) est la solution de minimisation de la fonction coût

$$Q^*(\lambda) = arg \min_Q [D(Q) + \lambda R(Q)]$$

En effet pour toute $Q \in S$, la solution Q^* respecte l'inégalité suivante

$$D(Q^*) + \lambda R(Q^*) \leq D(Q) + \lambda R(Q)$$

$$D(Q^*) - D(Q) \leq \lambda [R(Q) - R(Q^*)] \tag{4.3}$$

Comme cette inéquation est valable pour toute Q dans S elle restera aussi vraie dans tout sous-ensemble S^* de S, particulièrement dans le sous-ensemble défini par

$$S^* = \{Q : R(Q) \leq R(Q^*)\}$$

comme $\lambda \geq 0$ on aura

$$D(Q^*) - D(Q) \leq 0 \;\; pour \;\; Q \in S^*$$

et Q^* sera la solution du problème sous la contrainte (4.2)

$$R_d = R(Q^*)$$

La fonction du coût à minimiser

$$J(\lambda) = D(Q) + \lambda R(Q)$$

s'écrit comme suit

$$\begin{aligned} J(\lambda) &= \sum_i D_i + \lambda \sum_i R_i \\ &= \sum_i (D_i + \lambda R_i) \end{aligned}$$

$$J(Q^*) = min \sum_i (D_i + \lambda R_i) = \sum_i min(D_i + \lambda R_i)$$

Cette formulation est intéressante, elle décompose le problème d'origine en optimisations parallèles indépendantes pour les N^2 sous-bandes.
Notons que la solution du problème sous contrainte n'est pas toujours garantie, en revanche pour certaines valeurs de

$$\lambda \geq 0$$

il existe un problème sous contrainte dont la solution $Q^*(\lambda)$ est identique à celle du problème sans contrainte lorsque

$$R(Q^*) = R_d$$

La valeur de λ étant variable de $0 \longrightarrow +\infty$, l'ensemble des solutions $Q^*(\lambda)$ forme un nuage de points $(R(Q^*), D(Q^*))$ à enveloppe convexe. L'optimisation consiste à chercher le point du nuage à distorsion minimale et débit inférieur ou égal au budget donné R_d, correspondant à la valeur λ. Le multiplicateur de Lagrange λ peut être considéré comme un index de qualité où $\lambda = 0$ correspond au couple (haut débit, faible distorsion) et $\lambda = \infty$ au (faible débit, forte distorsion) figure 4.1.

4.2.2 Une Description Graphique

Pour resoudre le problème d'optimisation de l'allocation de bits nous rappelons une version modifié de l'algorithme décrit dans [45] et nous exposons la solution graphique de ce problème pour la simplification.
Pour chacune des N^2 sous-bandes nous disposons d'un ensemble de quantificateurs q_i. On commence par déterminer toutes les valeurs R_i et D_i pour tous les quantificateurs possibles q_i de chaque sous-bande i. Ce calcul n'est pas forcement très coûteux, surtout si on se limite à des choix restreints de quantificateurs. Dans le cas de la TCD^{-1} à la synthèse chaque sous-bande est traitée indépendamment des autres, comme le débit et la distorsion sont additifs sur l'ensemble des N^2 sous-bandes.
Si l'on effectuait toutes les combinaisons possibles des $(R_i(q_i), D_i(q_i))$, on obtiendrait un nuage dense de points dans le plan (R, D), l'algorithme cherche le point optimum (R^*, D^*) uniquement sur l'enveloppe convexe de ce nuage qui est une ligne brisée représentée à la figure 4.2. Ainsi le point optimum sur l'enveloppe convexe est celui qui se trouve immédiatement à gauche de la droite $R = R_d$ (point C sur la figure 4.2). L'algorithme d'allocation optimale de bits prend alors la forme de deux optimisations successives:

1. **Calcul d'un point sur l'enveloppe convexe en fonction d'une pente $-\lambda$:** Pour une valeur donnée de $\lambda \geq 0$ on cherche l'unique point de l'enveloppe convexe qui lui est associé. Il est défini comme l'intersection, réduite à un seul point, d'une droite de pente $-\lambda$ (en pointillé sur la figure 4.2) avec

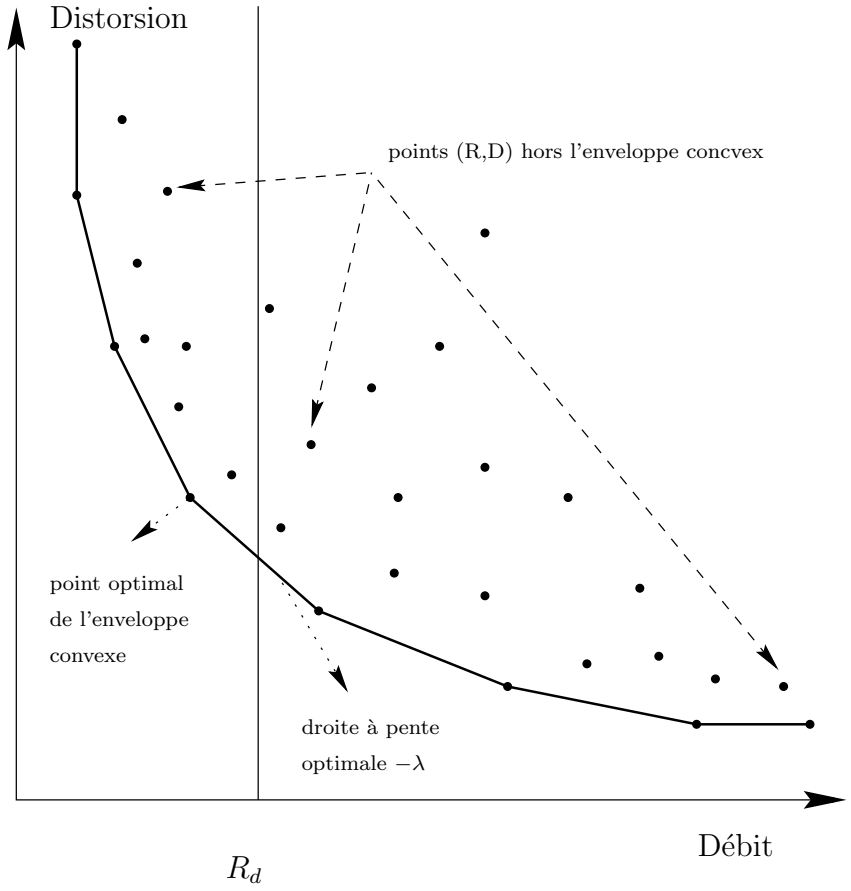

Figure 4.1: Courbe Débit/Distorsion opérationnelle et son enveloppe convexe.

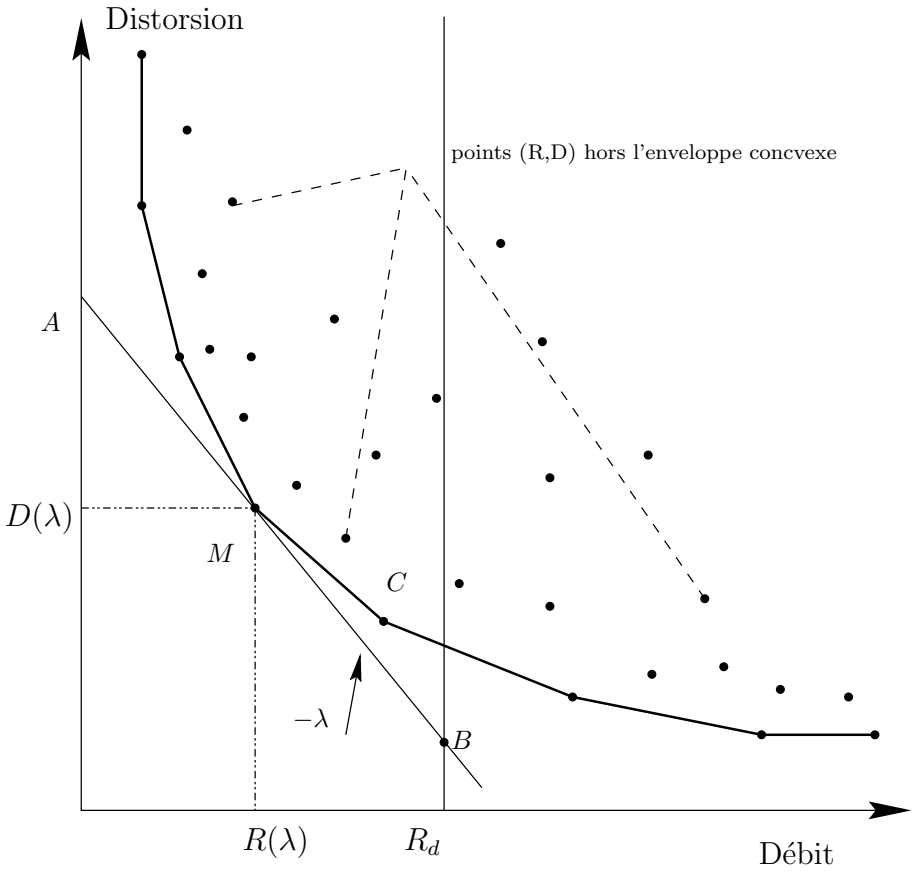

Figure 4.2: Interprétation graphique de l'algorithme d'allocation de bits.

l'enveloppe convexe. Si $-\lambda$ se trouve être la pente d'un des segments constituant l'enveloppe convexe, l'intersection est en fait tout ce segment, on choisit alors le point situé à sa gauche. Cette détermination revient à trouver le point $(R(\lambda), D(\lambda))$, parmi tous les points du nuage, qui minimise la distorsion du point A sur la figure 4.2. Il est facile de voir que cette quantité vaut $D + \lambda R$. (En cas d'égalité, on choisit le point dont la valeur de R est la plus faible.) Autrement dit, on résout le problème :

$$\min_{q_i}(D + \lambda R) = \sum_i (D_i + \lambda R_i)$$

Cette minimisation s'effectue facilement en minimisant la valeur $(D_i + \lambda R_i)$, pour un λ donné, indépendamment pour chacune des N^2 sous-bandes.

2. **Calcul de λ optimal:** Il reste à déterminer une valeur de λ pour laquelle l'étape ci-dessus fournit le point optimum (point C sur la figure 4.2). Il est facile de voir en considérant toutes les droites de pente variable passant par leurs points associés sur l'enveloppe convexe, que cela revient à maximiser

l'ordonnée du point B de la figure 4.2 qui est une fonction concave de λ

$$D + \lambda(R - R_d)$$

Ainsi la pente optimale λ^* est donnée par

$$\lambda^* = arg \max_{\lambda}[D(\lambda) + \lambda(R(\lambda) - R_d)]$$

où $R(\lambda)$ et $D(\lambda)$ sont fournis par l'étape ci-dessus. Cette maximisation est effectuée par un algorithme de type Newton sur les pentes [45], qui, en pratique converge très rapidement vers la solution optimale $R^* = R(\lambda^*)$, $D^* = D(\lambda^*)$. Une analyse plus poussée de cet algorithme basée sur des fonctionnelles lagrangiennes est donnée dans [45].

4.3 Allocation de Bits avec les Filtres de Wiener à la Synthèse

4.3.1 Le Problème de Non-Orthogonalité de la MMSE avec la TCD

Dans le paragraphe précédent, on s'intéressait aux propriétés d'additivité de la distorsion et du débit avec les transformées orthogonales à l'analyse et la synthèse telle que la TCD. La transformée en $MMSE$ (Erreur Quadratique Moyenne Minimale) ou le banc de filtres de Wiener calculé dans le chapitre 3, ne conserve plus la propriété d'additivité de la distorsion sur l'ensemble des sous-bandes.
De ce fait l'algorithme décrit dans le paragraphe précédent et applicable pour les codeurs/décodeurs à base des transformées orthogonales n'est plus valable tel qu'il est pour un banc de filtres de Wiener. (On verra dans la suite que sous des conditions bien précises sur la distorsion, la correlation du bruit de quantification et les signaux dans les sous-bandes on pourra éventuellement appliquer l'algorithme sans modifications.)

4.3.2 Une Distorsion Additive ou Non ?

En effet l'application de l'algorithme décrit ci-dessus est conditionnée par la propriété de distorsion additive qui est elle même le resultat de l'orthogonalité des filtres de synthèse et ceux d'analyse. L'additivité de la distorsion est possible dans le cas de la transformée $MMSE$ si le bruit dans une sous-bande est décorrélé du signal dans une autre sous-bande.
Reprenons le schéma de codage à la figure 3.2, la distorsion mesurée en Erreur Quadratique Moyenne EQM (MSE) sur les blocs de l'image est donnée par :

$$D = \frac{1}{N^2} E[\mid \hat{X}(n) - X(n) \mid^2]$$

$$Y(n) = TX(n); \hat{Y}(n) = Y(n) + B(n)$$

Soit $\hat{X}(n)$ le bloc de l'image reconstruite.

$$\hat{X}(n) = G\hat{Y}(n)$$

Le bruit $B(n)$ est supposé encore additif.

$$\begin{aligned} \mid \hat{X}(n) - X(n) \mid^2 &= (\hat{X}(n) - X(n))^t(\hat{X}(n) - X(n)) \\ &= \hat{X}^t(n)\hat{X}(n) - 2\hat{X}^t(n)X(n) + X^t(n)X(n) \\ &= \hat{Y}^t(n)G^tG\hat{Y}(n) - 2\hat{Y}^t(n)G^tX(n) + X^t(n)X(n) \\ &= trace(G\hat{Y}(n)\hat{Y}^t(n)G^t - 2G\hat{Y}(n)X^t(n)) + X^t(n)X(n) \end{aligned}$$

$$D = \frac{1}{N^2}E[trace(G\hat{Y}(n)\hat{Y}^t(n)G^t - 2G\hat{Y}(n)X^t(n))] + \frac{1}{N^2}E[X^t(n)X(n)]$$

Or

$$\frac{1}{N^2}E[X^t(n)X(n)] = \sigma_X^2 + \overline{X_n}^2$$

où σ_X^2 et $\overline{X_n}$ sont respectivement la variance et la moyenne du signal d'entrée.

$$\begin{aligned} D = & \frac{1}{N^2}\sum_{i=0}^{N^2-1}(G_i^tTE[X(n)X^t(n)]T^tG_i - 2G_i^tTE[x_{nN^2+i}X(n)]) + \\ & \frac{1}{N^2}\sum_{i=0}^{N^2-1}(G_i^tE[B(n)B^t(n)]G_i + 2G_i^tE[Y(n)B^t(n)]G_i + 2G_i^tE[x_{nN^2+i}B(n)]) + \\ & \sigma_X^2 + \overline{X_n}^2 \end{aligned}$$

où G_i^t est le $i^{ème}$ vecteur ligne de la matrice G,

$$\begin{aligned} D = & \sigma_X^2 + \overline{X_n}^2 + \frac{1}{N^2}\sum_{i=0}^{N^2-1}(G_i^tTR_{XX}T^tG_i - 2G_i^tTE[x_{nN^2+i}X(n)]) + \\ & \frac{1}{N^2}\sum_{i=0}^{N^2-1}(G_i^tR_{BB}G_i + 2G_i^tR_{BY}G_i + 2G_i^tE[x_{nN^2+i}B(n)]) \end{aligned} \tag{4.4}$$

R_{XX} et R_{BB} sont les matrices d'autocorrélation du signal d'entrée et du bruit de quantification respectivement, R_{BY} est la matrice de corrélation du bruit de quantification et des signaux dans les sous-bandes. Pour une image à coder, si les transfomées d'analyse et de synthèse sont données, l'expression de la distorsion dans l'équation (4.4) contient un terme constant D_f et un autre variable D_v en fonction de la quantification et du bruit généré.

$$D_f = \sigma_X^2 + \overline{X_n}^2 + \frac{1}{N^2}\sum_{i=0}^{N^2-1}(G_i^tTR_{XX}T^tG_i - 2G_i^tTE[x_{nN^2+i}X(n)])$$

$$D_v = \frac{1}{N^2} \sum_{i=0}^{N^2-1} (G_i^t R_{BB} G_i + 2G_i^t R_{BY} G_i + 2G_i^t E[x_{nN^2+i} B(n)])$$

Le terme D_f est dû à la reconstruction non-parfaite, il devient nul lorsque les filtres de synthèse G_i sont à reconstruction parfaite avec le banc de filtres d'analyse.
Le deuxième terme D_v représente l'effet du bruit de quantification, sa valeur tend vers zéro lorsque à très haut débit (ou en absence de toute quantifiaction) les signaux $\hat{Y}(n)$ se rapprochent des $Y(n)$ et les filtres de synthèse convergent vers un banc de filtres à reconstruction parfaite. Pour des applications d'enregistrement, la compression est réalisée à bas débit et le terme D_v ne peut s'annuler en présence du bruit de quantification. L'algorithme d'allocation optimale de bit du paragraphe 4.2 nécessite pour son application l'additivité de la distorsion. Il est clair que pour respecter cette condition, il suffit que le terme D_v ait une forme additive. Si on impose la condition de décorrélation du bruit d'une sous-bande avec le signal d'une autre sous-bande, la matrice $R_{BB} + 2R_{BY}$ aura une forme diagonale

$$R_{BB} + 2R_{BY} = \begin{bmatrix} \alpha_{00} & \dots & 0 \\ \vdots & \alpha_{jj} & \vdots \\ 0 & \dots & \alpha_{N^2-1N^2-1} \end{bmatrix}$$

$$E[x_{nN^2+i} B(n)] = \begin{bmatrix} \beta_0(m) \\ \vdots \\ \beta_{N^2-1}(m) \end{bmatrix}$$

Ainsi

$$D_v = \frac{1}{N^2} \sum_{i=0}^{N^2-1} \sum_{j=0}^{N^2-1} (G_i^2(j)\alpha_{jj} + 2G_i(j)\beta_j)$$

La distorsion globale $D = D_f + D_v$ est dans ce cas additive sur les N^2 sous-bandes, et l'algorithme d'allocation de bit proposé pour le cas des transformées orthogonales est applicable.

4.4 La Convexité des courbes Débit/Distorsion. Solution sous-Optimale

La condition imposée sur la décorrélation du bruit de quantification d'une sous-bande et le signal d'une autre sous-bande semble pratique pour la recherche d'une allocation optimale de bits avec un banc de filtres de Wiener, sans des coûts supplémentaires par rapport au cas avec des bancs de filtres ou des transformées orthogonaux. En effet, cette condition est largement moins contraignante que celle de l'additivité. Toutefois, le calcul du banc de filtres de Wiener dans le chapitre précédent (la transformée $MMSE$) tient compte de la corrélation signal/bruit entre

sous-bandes, parsuite nous proposons un autre algorithme d'allocation de bits sous-optimal, qui fournit des résultats plus précis sans imposer des conditions préalables sur la quantification.
L'algorithme appliqué est inspiré des travaux dans [45] et [13]. En effet, on calcule pour chaque sous-bande les points de la courbe débit/distorsion, en quantifiant toutes les sous-bandes par des quantificateurs fixes et en faisant juste varier le quantificateur de la sous-bande traitée. Dans ce cas, la distorsion mesurée à la reconstruction contiendra une valeur d'erreur quasi-fixe relative aux autres sous-bandes quantifiées. La courbe débit/distorsion de chaque sous-bande garde sa convexité et sa décroissance avec le débit malgré la translation sur l'axe de distorsion dûe à la valeur quasi-fixe d'erreur de quantification des autres sous-bandes. La convexité de la courbe débit/distorsion composite étant essentielle pour trouver une solution avec l'algorithme de [13], nous pouvons ainsi appliquer cet algorithme avec le banc de filtres de Wiener pour chercher une solution sous-optimale.
L'avantage de cette technique est que les termes de corrélations signal/bruit ont maintenant le même ordre de grandeur et les résultats d'optimisation de l'allocation de bits sont plus robustes par rapport à la méthode sous contrainte de décorrélation signal/bruit entre les sous-bandes.

4.5 Conclusion

Dans ce chapitre, nous avons passé en revue un algorithme d'allocation optimale de bits basé sur les fonctionnelles lagrangiennes et l'avons appliqué dans un schéma de codage d'image utilisant des bancs de filtres (ou transformée) orthogonaux. Cet algorithme permet de résoudre le problème d'allocation de bits sous contrainte de débit inférieur ou égal à une valeur donnée. Il reformule le problème sous-contrainte en un ensemble de problèmes d'optimisations sans contraintes, parallèles et indépendants pour les sous-bandes.
Une description graphique aide à mieux comprendre le mécanisme de l'algorithme à deux itérations.
Nous avons également donné les conditions d'additivité de la distorsion sur l'ensemble des sous-bandes et de convexité de la courbe débit/distorsion pour chacune d'elles. L'application de l'algorithme d'allocation de bits pour un codeur à base de transformée dépend des conditions ci-dessus. Dans le cas de banc de filtres de Wiener à la synthèse, la première condition sur l'orthogonalité n'est plus satisfaite et la distorsion n'est plus additive que si le bruit de quantification d'une sous-bande est décorrélé du signal dans chacune des autres sous-bandes. Or cette condition est irréaliste du fait que le calcul des filtres de Wiener (la transformée $MMSE$) est basée sur la corrélation signal/bruit entre les sous-bandes. Nous montrons que les courbes débit/distorsion maintiennent leur convexité, si nous quantifions toutes les sous-bandes par des quantificateurs fixes, sauf pour celle dont on trace la courbe

débit/distorsion on fait varier le pas de quantification dans une intervalle donnée. Cette technique de quantification permet de mesurer la distorsion globale qui contient une valeur quasi-fixe dûe à la quantification des autres sous-bandes. Cette valeur d'erreur déplace la courbe débit/distorsion sur l'axe de distorsion sans modifier son allure convexe. Cette technique semble donner des résultats plus robustes à la condition de décorrélation citée plus haut.

Chapitre 5

l'Optimisation d'un schéma de codage d'image à base d'une TCD

Dans les chapitre 3 et 4, nous avons proposé respectivement une nouvelle transformée de synthèse optimale et un algorithme d'allocation optimale de bits adapté à cette nouvelle transformée. Pour une meilleure optimisation du schéma de codage, nous développons dans ce chapitre un algorithme itératif baptisé **A1** pour l'optimisation conjointe des quantificateurs et de la transformée de synthèse. Cet algorithme nécessite quelques modifications dans la façon de générer les tables de **Huffman**. Celles-ci seront calculées selon les statistiques dans les sous-bandes alors que dans un codeur $JPEG$ les tables de **Huffman** sont fixes et calculées en fonction des statistiques dans les blocs en zig-zag de l'image.
Dans l'annexe **C** du $JPEG$, la norme prévoit une quantification variable des blocs de coefficients. En effet, outre la table de quantification globale pour tous les blocs, on calcule pour chacun d'eux un facteur d'échelle pour pondérer la table de quantification, dans une intervalle donnée de valeurs discrètes. Suite à cette extension de la norme $JPEG$, nous proposons une technique similaire pour le cas d'un banc de filtres de Wiener. Tout en fixant la matrice de quantification globale, nous optimisons conjointement les facteurs d'échelle et la transformée de synthèse pour le même critère que celui de l'algorithme **A1** et sous la même contrainte de débit total. L'algorithme proposé sous le nom de **A2** n'est que la suite naturelle de **A1**.

5.1 L'Optimisation Conjointe des Quantificateurs et de la Transformée de Synthèse

Pour une image donnée, le choix des pas de quantification des sous-bandes définit le débit total que génère la compression de cette image. La distorsion calculée à la

reconstruction dépend du choix des quantificateurs ainsi que du banc de filtres de synthèse. Plus précisement, un algorithme d'allocation optimale de bits fait appel à des fonctions spécifiques pour évaluer la distorsion engendrée par l'utilisation de tel ou tel quantificateur pour une sous-bande.
Dans le cas du banc de filtres de Wiener à la synthèse, ces fonctions d'évaluation de la distorsion calculent la distorsion totale induite sur l'image reconstruite. On aura donc besoin d'une transformée de synthèse préalablement optimisée pour certains quantificateurs des sous-bandes afin de pouvoir tracer les courbes débit/distorsion. La question qui se pose est quelle sera cette transformée de synthèse optimale (ou ce banc de filtres de Wiener) qui nous permet de démarrer notre optimisation des pas de quantification?
Avant de répondre à cette question, nous citons les travaux de K. GOSSE et P. DUHAMEL [14] et [25] en optimisation conjointe des quantificateurs et des filtres de synthèse sous contrainte de distorsion minimale, qui ont servi d'appui à la réalisation de nos algorithmes.
Le calcul d'un banc de filtres de Wiener dépend comme nous avons vu des quantificateurs des sous-bandes et de l'erreur due au bruit de quantification correspondant. Un banc de filtres optimal nécessite des pas de quantification optimaux pour une distorsion minimale sous contrainte d'un débit donné. Le problème d'optimisation est alors double:

1. Ayant un banc de filtres de synthèse optimal, calculer les quantificateurs optimaux;

2. Pour les quantificateurs optimaux calculés, trouver le banc de filtres de Wiener qui minimise la distorsion.

Pour resoudre ce problème double, nous proposons un algorithme itératif qui améliore à chacune de ses étapes le rapport Signal/Bruit (Peak Signal to Noise Ratio $PSNR$) ou en d'autres termes qui minimise la distorsion globale. Cet algorithme à améliorations succéssives optimise itérativement chaque famille de paramètres séparément et tour à tour.

5.2 L'Algorithme d'Optimisation itératif A1

L'algorithme **A1** proposé consiste en 2 phases:

- <u>Phase 1 Initialisation:</u>

    1. On initialise les pas de quantifications des sous-bandes à des valeurs multiples de celles proposées dans la norme $JPEG$, de sorte à avoir un débit total très proche de celui qu'on cherche;

2. Pour les pas de quantifications de l'étape 1, nous calculons le banc de filtres de Wiener ($MMSE$) qui minimise la distorsion;
3. On génère les tables de **Huffman** pour les sous-bandes AC et DC selon les statistiques dans les sous-bandes. Notons que cette étape diffère du $JPEG$ du fait que les tables de **Huffman** sont dynamiques. Elles dépendent de l'image à coder et tiennent compte des statistiques dans les sous-bandes et non plus dans les blocs ordonnés en zig-zag.
   Comme le calcul des tables est très coûteux en temps de calcul, nous suggérons de les compléter pour pouvoir les utiliser en traçant les courbes débit/distorsion (nécessaires pour l'allocation optimale de bits). Nous verrons que cette manipulation des tables de **Huffman** est très simple et ne coûte que quelques octets de plus par rapport aux tables non-complétées.

- Phase 2:
  Après la phase d'initialisation, il ne reste qu'à itérer entre le calcul des pas de quantification optimaux et la transformée de synthèse optimale.

  1. En utilisant la transformée de synthèse optimale ($MMSE$) et les tables de **Huffman** calculées à la phase d'initialisation (ou à l'étape 2 de la phase 2), on calcule avec l'algorithme d'allocation optimale de bits du chapitre 4, les pas de quantification optimaux;
  2. Les quantificateurs calculés à l'étape 1 (phase 2), étant optimaux pour un débit donné, on recherche la transformée de synthèse ($MMSE$). Les tables de **Huffman** étant dynamiques, on les recalcule en fonction des nouvelles statistiques dans les sous-bandes quantifiées.
     Si la distorsion globale reste constante par rapport à l'itération précédente on arrête, sinon on itère les étapes 1 et 2 de la phase 2 jusqu'à convergence. La convergence est assurée puisqu'à chaque itération la distorsion diminue ou le débit se rapproche de la valeur maximale qu'il peut atteindre. Il est toujours possible d'atteindre un minimum local avec l'algorithme **A1**.

5.2.1 Comment Compléter les tables de Huffman?

Les tables de **Huffman** sont générées en fonction des statistiques dans un vecteur d'échantillons donné. La structure de ces vecteurs d'échantillons est différente selon qu'on traite les coefficients DC ou les AC dans le schéma de codage à base d'une TCD. Pour les coefficients AC par exemple les vecteurs qui serviront à calculer les tables de **Huffman** sont des séries de couples $< \textit{longueur de plage}, \textit{amplitude} >$.
Dans notre algorithme d'allocation de bits, pour tracer la courbe débit/distorsion d'une sous-bande, son quantificateur prend sa valeur dans un intervalle discret.

Cette variation du quantificateur entraine un changement dans les coefficients quantifiés et les valeurs $< longueur\ de\ plage, amplitude >$. Pour coder (avec **Huffman**) cet éventail de valeurs, on pourrait recalculer les tables de **Huffman** pour chaque variation du quantificateur, ce qui serait très coûteux en temps de calcul et en mémoire. L'idéal est d'utiliser une seule table de **Huffman** qui couvre toutes les valeurs $< longueur\ de\ plage, amplitude >$ possibles et qui soit bien sûre adaptée au contenu de l'image à coder.
Pour trouver une telle table de **Huffman**, on a ajouté des codes à longueur fixe à une table préalablement calculée relativement à un ensemble donné $\mathcal{Q}$ de quantificateurs des sous-bandes. Ces codes à longueur fixe correspondent à des couples $< longueur\ de\ plage, amplitude >$ qui n'apparaissent pas en quantifiant l'image par l'ensemble $\mathcal{Q}$ mais qui peuvent éventuellement exister pour un autre ensemble des quantificateurs. En effet, on attribue à ces nouveaux couples $< longueur\ de\ plage, amplitude >$ une même probabilité d'occurence très faible que nous utilisons pour ajouter des codes à longueur fixe selon l'arbre binaire de **Huffman**. Ces nouveaux codes (à longueur fixe) nous permettent simplement d'évaluer le débit sans qu'il y ait besoin de les transmettre ou de les stocker. La connaissance de la longueur du mot de code est suffisante pour évaluer le débit des sous-bandes quantifiées. A la fin de chaque itération, nous recalculons une table de Huffamn adaptée aux coefficients quantifiés par les quantificateurs optimaux et nous la comparons à celle completée par les codes à longueur fixe pour garantir une amélioration dans l'optimisation débit/distorsion.
Pour illustrer cette technique reprenons l'exemple du codage de **Huffman** dans le chapitre 2. Les symboles $\hat{y}_1$, $\hat{y}_2$, $\hat{y}_3$, $\hat{y}_4$, $\hat{y}_5$ et $\hat{y}_6$ représentent respectivement les valeurs 1, 2, 4, 5, 7 et 9. Nous cherchons à compléter la table de **Huffman** calculée pour coder tous les entiers de l'intervalle [1, 10]. On attribue une probabilité P très faible aux entiers 3, 6, 8, 10 et on normalise les probabilités par la somme $(1 + 4P)$ (4 le nombre de codes à ajouter). Une façon pour calculer la probabilité P est de diviser par $(100 \times n)$ la valeur absolue de la différence des deux probabilités les plus faibles (n est le nombre de codes à ajouter, $n = 4$ dans notre exemple).

$$P = \frac{(0.06 - 0.04)}{4 \times 100}$$

Les probabilités associées aux entiers de l'intervalle [1, 10] seront celles données dans la table (5.1). En applicant l'algorithme de **Huffman**, on construit l'arbre binaire de la figure 5.1, où les codes ajoutés pour les entiers 3, 6, 8, 10 auront la longueur fixe de 7bits (cf. table 5.2).
N.B.: Suite à cette opération, la longueur de mot de code du symbole 9 augmente de 1bit. L'estimation du débit des sous-bandes quantifiées par l'ensemble $\mathcal{Q}$ sera légèrement modifiée car la probabilité associée au symbole 9 est la plus faible.

Symboles	1	2	3	4	5	6	7	8	9	10
Probabilités	0.4999	0.14997	0.00005	016997	0.07998	0.00005	0.05999	0.00005	0.03999	0.00005

Table 5.1: Probabilités associées aux entiers de l'intervalle [1,10]

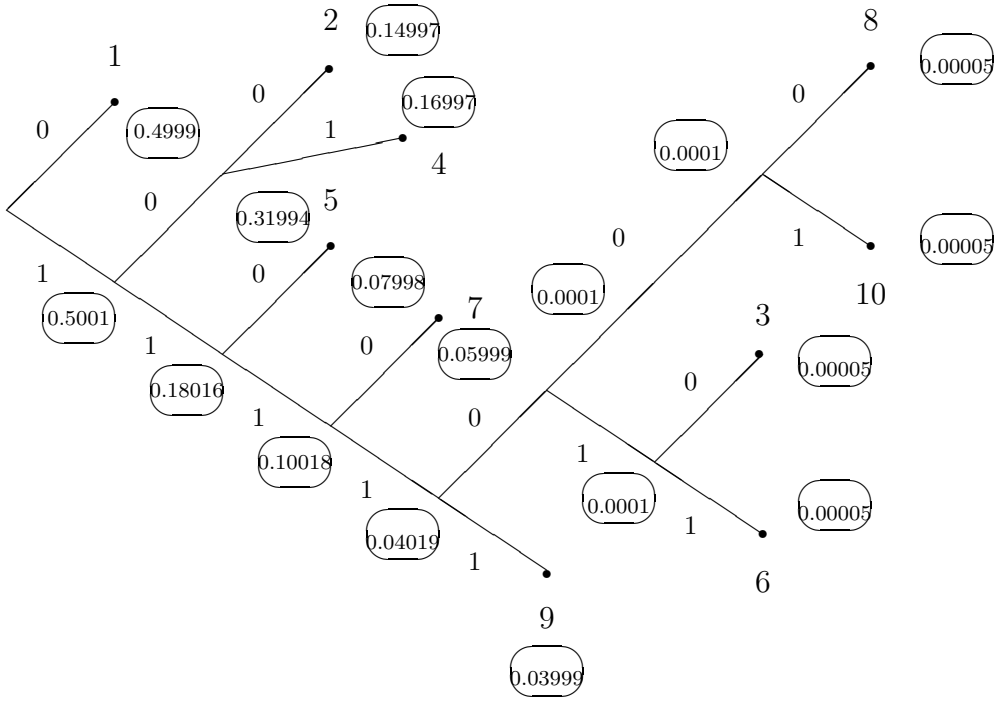

Figure 5.1: L'arbre binaire complété de l'algorithme de **Huffman**

Symboles	1	2	3	4	5	6	7	8	9	10
Mots de code	0	100	1111010	101	110	1111011	1110	1111000	11111	1111001

Table 5.2: Mots de code relatifs aux entiers de l'intervalle [1,10]

5.3 Introduction de la Quantification Variable et des Facteurs d'échelle

Une table de quantification, telle qu'elle est calculée par l'algorithme **A1** est globale pour tous les blocs. Elle permet de minimiser la distorsion globale sur l'image reconstruite. Le contenu d'une image varie d'un bloc à un autre, on y trouvera des hautes fréquences dans des blocs appartenant à des contours ou des basses fréquences dans les régions homogènes. La quantification moyenne de tous les blocs de l'image ne tient pas compte de la variation fréquentielle locale propre à chaque bloc. Pour pallier cette insuffisance de la quantification moyenne, nous proposons une quantification locale variable pour chaque bloc selon la pertinence de son contenu d'informations.

5.3.1 L'Annexe C du JPEG (Quantification Variable)

Cette annexe spécifie la technique de quantification variable dans une image ainsi que la modification de syntaxe des flux $JPEG$ de base. La procédure de quantification variable consiste à calculer, par bloc de 8×8, un facteur d'échelle de la table de quantification globale.
La table des symboles utilisée pour décoder les coefficients différentiels DC est augmentée d'un nouveau symbole appelé QS_CHANGE. Dans un bloc ordonné en zig-zag, le nouveau symbole s'il s'y trouve n'affecte que les coefficients de ce bloc. Il indique que les 5 bits qui suivent sont utilisés pour désigner un nouveau facteur d'échelle de la table de quantification. Les 5 bits définissent le paramètre $SCALE_CODE$ qui est un pointeur vers une table de correspondance (Look Up Table LUT). Cette table contient les valeurs possibles du facteur d'échelle Q_SCALE à utiliser juste pour pondérer les quantificateurs AC de la table de quantification.
Deux types de tables de facteurs d'échelle sont autorisées: Linéaire et Logarithmique, sachant qu'un segment marqueur DQS est employé pour la sélection d'une des 2 tables. La quantification des coefficients AC tiendra compte de cette quantification variable. L'équation suivante définit cette nouvelle quantification:

$$Yq_{i,j} = Round((X_{i,j} \times 16)/(Q_{i,j} \times Q_SCALE))$$

et la quantification inverse sera

$$Y = Yq_{i,j} \times Q_{i,j} \times Q_SCALE/16$$

Ainsi la valeur de Q_SCALE qui représente la quantification sans facteur d'échelle est 16. Pour plus de détails le lecteur peut se référer à la norme $JPEG$ [1].

5.4 Le Calcul des facteurs d'échelle

5.4.1 Cas avec TCD (JPEG-3): Calcul simple

En présence d'une TCD à la synthèse ($JPEG$) un simple algorithme d'allocation optimale de bits nous permet de calculer les pas de quantification les mieux adaptés à un critère donné (distorsion globale,...) sous une contrainte de débit. L'utilisation des multiplicateurs de Lagrange pour simplifier le problème d'optimisation des quantificateurs nous a permis de le transformer en un ensemble d'optimisations parallèles plus simples à traiter. De la même façon pour le critère d'une minimisation de la distorsion et sous la contrainte de débit on peut calculer un facteur d'échelle par bloc de coefficients.
En choisissant des facteurs d'échelle linéaires ou logarithmiques, on peut trouver pour chacun des blocs de l'image une courbe débit/distorsion et calculer la pente optimale et le point du nuage qui fournit la distorsion la plus faible sous la contrainte de débit.

5.4.2 Cas avec MMSE

Le calcul des facteurs d'échelle dans le cas d'un banc de filtres de Wiener à la synthèse est un peu plus compliqué que celui de la TCD^{-1} ($JPEG$).
La transformée $MMSE$ préalablement calculée (en appliquant l'algorithme **A1** ou par un simple calcul d'un banc de filtres de Wiener) nous permet de tracer les courbes débit/distorsion pour chacun des blocs de l'image. Pour l'ensemble des points du nuage on peut trouver, à l'aide des fonctionnelles lagrangiennes, un facteur optimal par bloc. Cette quantification variable génère un bruit qui sera probablement atténué par le banc de filtres optimisé pour un autre ensemble de quantificateurs. Un nouveau calcul de la transformée optimale est nécessaire afin de garantir la minimisation de la distorsion globale une fois la quantification variable est mise en place. En principe, une nouvelle itération de calcul des facteurs d'échelle suivie par une nouvelle recherche d'un banc de filtres de Wiener ne doivent que réduire l'erreur de reconstruction et améliorer la qualité de l'image reconstruite.
La table de quantification préalablement optimisée pour l'ensemble des sous-bandes reste fixe lors de l'optimisation conjointe des facteurs d'échelle et de la transformée $MMSE$.

5.5 L'Algorithme itératif A2

L'algorithme d'optimistaion des facteurs d'échelle et du banc de filtres de Wiener est baptisé **A2**. Il consiste en 3 étapes et est la suite naturelle de l'algorithme **A1**.

1. Initialiser tous les facteurs d'échelle à 16 (pas de quantification variable), générer et compléter la table de **Huffman** des coefficients AC comme en

$JPEG$, selon les statistiques dans les blocs en zig-zag. Si l'algorithme **A2** n'est pas précédé par l'application de **A1**, on procède à un calcul de la transformée de synthèse ($MMSE$) qu'on utilisera dans l'étape 2;

2. A l'aide de la table de **Huffman** et le banc de filtres de Wiener, tracer la courbe débit/distorsion pour chaque bloc de l'image. En utilisant les fonctionnelles Lagrangiennes, chercher les facteurs d'échelle optimaux;

3. Calculer la transformée $MMSE$ pour la quantification variable de l'étape 2 et itérer les étapes 1, 2 et 3 tant que l'amélioration du $PSNR$ est supérieure à 0.01dB.

Les simulations montrent qu'au bout de 3 à 8 itérations la valeur du $PSNR$ se stabilise et l'algorithme converge vers un minimum local.

5.6 Les Résultats de codage par le nouveau schéma optimisé

Pour mesurer les performances de l'optimisation conjointe des quantificateurs et du banc de filtre de synthèse, les images codées sont évaluées qualitativement par observation visuelle et quantitativement par le Rapport Signal/Bruit $PSNR$ en dB. Nous faisons par la suite la distinction entre les algorithmes **A1** et **A2** pour le calcul des quantificateurs et des facteurs d'échelle respectivement. Pour tracer les courbes débit/distorsion (R/D), on reprend le contexte et les conditions de simulations du chapitre 3. Les résultats de l'optimisation conjointe du banc de filtres de synthèse et des quantificateurs dans l'algorithme **A1** seront comparés avec ceux de l'optimisation de l'allocation de bits pour un codeur JPEG de base. Quand à l'optimisation conjointe des facteurs d'échelle et du banc de filtres de Wiener dans l'algorithme **A2**, les résultats correspondants seront comparés avec ceux de la procédure de quantification variable étendue à la norme $JPEG$.

5.6.1 Comparaison du JPEG avec les différentes étapes des algorithmes A1 et A2

Les tables (5.3, 5.4) établissent respectivement les résultats de codage des images test LENA et BARBARA. Chacune de ces deux images est codée par le $JPEG$ de base, le $JPEG$ à quantification optimisée et enfin par l'algorithme **A1**. Les tables de **Huffman** utilisées dans les trois cas sont adaptatives aux statistiques des images quantifiées.
On observe sur l'image LENA codée à 0.25bpp par le $JPEG$ à quantification optimisée (figure 5.3) une légère hausse du $PSNR$ (0.22dB) sans une amélioration visuelle remarquable de la qualité d'image par rapport au cas de codage avec le $JPEG$

de base (figure 5.2). Pour le même débit (0.25bpp), l'application de l'algorithme **A1** sur LENA résulte en une augmentation du $PSNR$ (0.49dB) par rapport au $JPEG$ de base et une meilleure qualité visuelle qui se manifeste par l'atténuation du bruit sur les contours et par une reconstruction optimale de quelques détails hautes fréquences (figure 5.4).
L'image de test BARBARA constitue un meilleur exemple pour l'application de l'algorithme **A1** car elle contient beaucoup de détails hautes fréquences. Un codage par le $JPEG$ à quantification optimisée à 0.25bpp, améliore le $PSNR$ de 0.43dB (figure 5.7), alors que l'algorithme **A1** fournit au même débit un $PSNR$ de 27.33dB (soit une hausse de 1.36dB par rapport au $JPEG$ de base) et reconstruit quelques détails hautes fréquences avec moins de distorsion (figure 5.8). La quantification variable proposée en extension de la norme $JPEG-3$ [1], dans le but d'améliorer la qualité et le $PSNR$ de l'image codée, utilise pour chaque bloc de l'image un facteur de pondération de la matrice de quantification. Les simulations de cette technique montrent qu'on peut gagner plus de 0.5dB sur l'image BARBARA codée à 1bpp. Par analogie l'algorithme **A2** pour l'optimisation conjointe des facteurs d'échelle et de la transformée de synthèse nous a permis d'augmenter encore le $PSNR$ par rapport au résultat de codage par l'algorithme **A1**. Les courbes débit/distorsion (figures 5.12 et 5.13) récapitulent les résultats d'optimisation conjointe par les algorithmes **A1** et **A2** en comparaison avec le $JPEG$ à quantification optimisée et le $JPEG$ à quantification variable.

5.7 Conclusion

Ce chapitre montre comment améliorer les performances d'un système de codage $JPEG$ en optimisant conjointement le banc de filtres de synthèse selon le critère $MMSE$ (filtrage matriciel de Wiener) et les quantificateurs des sous-bandes de la $TCDB$. Le rôle d'une allocation optimale de bits est de réduire l'énergie des hautes sous-bandes quantifiées et ainsi ne pas les transmettre pour respecter un débit maximal. Une optimisation de la synthèse signifie une meilleure reconstruction de l'image à partir des sous-bandes transmises. Pour un débit maximal donné, il faut chercher un compromis entre filtres de synthèse et quantificateurs, d'où l'intérêt de l'algorithme **A1** présenté à la Section 5.2. Nous introduisons enfin l'algorithme **A2** pour la mise en oeuvre d'une quantification variable dans une image, optimisée conjointement avec les filtres de synthèse. L'algorithme **A2** permet de gagner encore au niveau du $PSNR$ et d'améliorer la qualité d'image mais le nombre d'itérations (de 5 à 10 itérations) ainsi que la complexité du calcul ne justifie pas le gain en $PSNR$ de 0.2 à 0.5dB.

	Lena $PSNR$ (dB)		
débit (bpp)	$JPEG$ (TCD^{-1})	**Quantificateurs optimaux** $JPEG$	**Optimisation conjointe Algo. A1**
0.15	27.11	27.38	27.94
0.25	29.70	29.92	30.19
0.50	32.79	32.95	33.35
0.75	34.43	34.76	35.03
1.00	35.65	36.04	36.30

Table 5.3: Comparaison du codage par l'algorithme **A1** de l'image "Lena" avec le $JPEG$ de base et une optimisation de ses quantificateurs

	Barbara $PSNR$ (dB)		
débit (bpp)	$JPEG$ (TCD^{-1})	**Quantificateurs optimaux** $JPEG$	**Optimisation conjointe Algo. A1**
0.15	24.14	24.58	25.22
0.25	25.97	26.40	27.33
0.50	30.14	30.48	31.18
0.75	32.19	33.28	34.12
1.00	34.45	35.60	36.36

Table 5.4: Comparaison du codage par l'algorithme **A1** de l'image "Barbara" avec le $JPEG$ de base et une optimisation de ses quantificateurs

	Lena $PSNR$ (dB)		
débit (bpp)	$JPEG$ (TCD^{-1})	**Quantification variable** $JPEG$	**Optimisation conjointe Algo. A2**
0.15	27.11	27.42	27.85
0.25	29.70	29.98	30.30
0.50	32.79	33.28	33.54
0.75	34.43	35.10	35.35
1.00	35.65	36.46	36.58

Table 5.5: Comparaison du codage par l'algorithme **A2** de l'image "Lena" avec le $JPEG$ de base sa quantification variable

Figure 5.2: "Lena" codée à 0.25bpp par le JPEG de base PSNR=29.70dB

Figure 5.3: "Lena" codée à 0.25bpp par le JPEG avec des quantificateurs optimisés PSNR=29.92dB

Figure 5.4: "Lena" codée à 0.25bpp par l'algorithme **A1** PSNR=30.19dB

Figure 5.5: "Lena" codée à 0.25bpp par l'algorithme **A2** PSNR=30.30dB

Figure 5.6: Vue rapprochée de l'image "Barbara" codée à 0.25bpp par le JPEG de base PSNR=25.97dB

	Barbara $PSNR$ **(dB)**		
débit (bpp)	$JPEG$ (TCD^{-1})	**Quantification variable** $JPEG$	**Optimisation conjointe Algo. A2**
0.15	24.14	24.52	25.22
0.25	25.97	26.52	27.35
0.50	30.14	30.82	31.41
0.75	32.19	33.80	34.36
1.00	34.45	36.13	36.61

Table 5.6: Comparaison du codage par l'algorithme **A2** de l'image Barbara avec le $JPEG$ de base et sa quantification variable

Figure 5.7: Barbara codée à 0.25bpp par le JPEG avec des quantificateurs optimisés PSNR=26.40dB

Figure 5.8: Barbara codée à 0.25bpp par l'algorithme **A1** PSNR=27.33dB

Figure 5.9: Barbara codée à 0.25bpp par l'algorithme **A2** PSNR=27.35dB

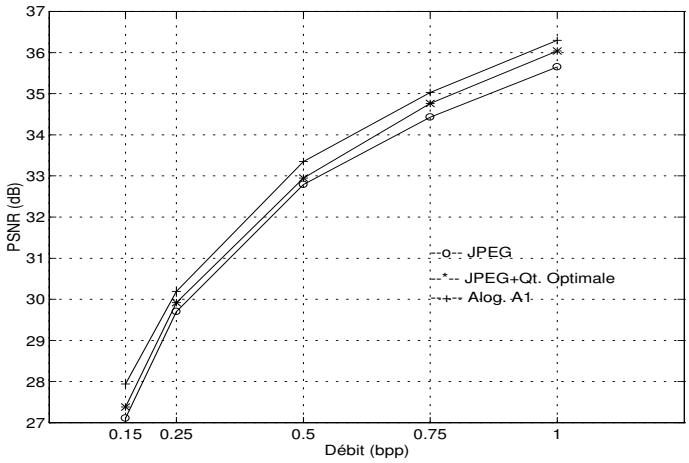

Figure 5.10: Rapport Signal/Bruit PSNR (dB) de l'image "Lena" codée avec le $JPEG$ de base, le $JPEG+$ Quantificateurs Optimisés et l'algorithme **A1**, en fonction du débit total alloué.

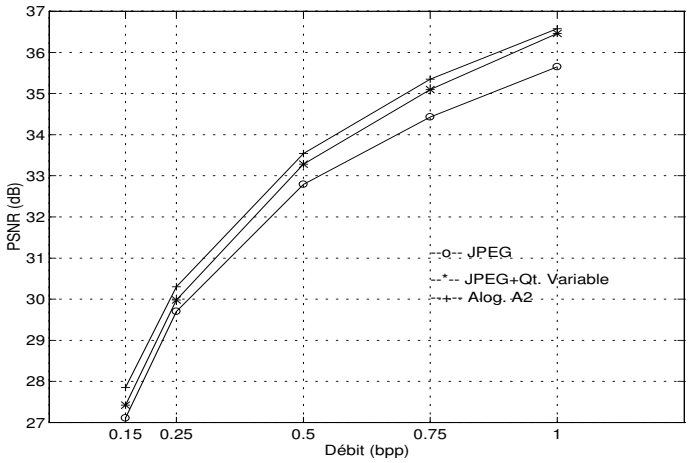

Figure 5.11: Rapport Signal/Bruit PSNR (dB) de l'image "Lena" codée avec le $JPEG$ de base, le $JPEG+$ Quantification Variable et l'algorithme **A2**, en fonction du débit total alloué.

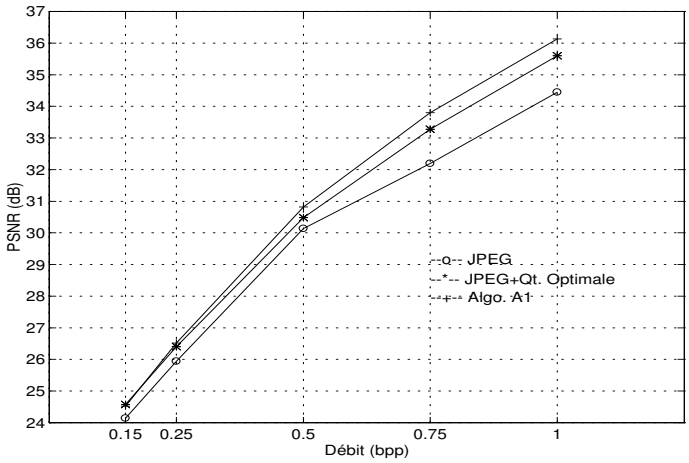

Figure 5.12: Rapport Signal/Bruit PSNR (dB) de l'image "Barbara" codée avec le *JPEG* de base, le *JPEG+* Quantificateurs Optimisés et l'algorithme **A1**, en fonction du débit total alloué.

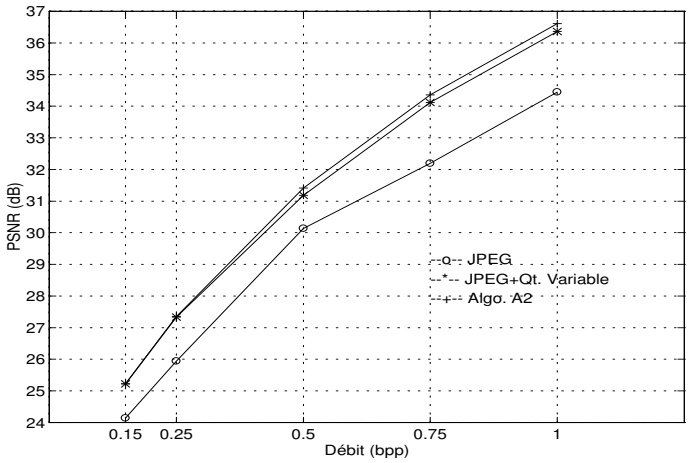

Figure 5.13: Rapport Signal/Bruit PSNR (dB) de l'image "Barbara" codée avec le *JPEG* de base, le *JPEG+* Quantification Variable et l'algorithme **A2**, en fonction du débit total alloué.

Partie II

Post-traitement à la reconstruction, Réduction des effets de blocs

Chapitre 6

Une nouvelle technique de post-traitement de l'image

6.1 Introduction et motivation

Avec l'avènement des normes de compresssion $JPEG$ pour les images fixes et le $MPEG$ pour les images animées, l'usage de la $TCDB$ (Transformée en Cosinus Discrète des Blocs) est devenu un outil simple et efficace pour la compression d'image. Dans l'approche de base, on divise l'image en blocs avant de transformer, quantifier et coder chacun de ces derniers. Pour des taux de compression élevés, ce traitement individuel des blocs génère des discontinuités artificielles entre eux, qui représentent les hautes fréquences dans le bruit de quantification. La qualité subjective de l'image reconstruite s'améliore par une réduction de ces effets de bloc, aux dépens d'une légère apparition d'un bruit.
Dans ce contexte, nous proposons dans la suite de ce chapitre une nouvelle approche pour la réduction des effets de bloc. Cette technique consiste à ajuster un ensemble de coefficients $TCDB$ dans un bloc de (8×8), selon un critère de minimisation de l'énergie des hautes fréquences situées dans le bloc de coefficients $TCDB$ (24×24), obtenu par transformation du bloc courant et de 8 blocs de son voisinage (cf. figure 6.2).
L'algorithme proposé s'applique sur les coefficients $TCDB$ quantifiés, avant reconstruction par la $TCDB^{-1}$ (dans le cas de $JPEG$) ou par le banc de filtres de Wiener (dans le cadre de notre optimisation du schéma de codage).
Le $PSNR$ n'est pas toujours le bon critère pour choisir la meilleure image codée. En effet cette valeur quantitative ne représente pas parfaitement la valeur qualitative d'une image. Ainsi, les performances de notre approche seront évaluées selon les deux critères quantitatif ($PSNR$) et qualitatif (qualité visuelle). Notons que

l'algorithme de réduction des effets de blocs est baptisé **B1**.

6.2 Travaux antérieurs liés au post-traitement

Dans le cadre de réduction des effets de blocs, nombreux sont les techniques et les algorithmes développés pour améliorer la qualité de l'image décodée. On distingue deux catégories de techniques.
La première qui modifie la structure de transformation en utilisant une transformée autre que la $TCDB$, la deuxième qui regroupe les techniques de post traitement. Parmi les transformées utilisées dans la première catégorie on trouve la transformée des blocs avec recouvrement qui permet de créer une sorte de dépendance entre les blocs [10, 33]. La transformée LOT (Lapped Orthogonal Transform) [29] est particulièrement intéressante car elle permet le recouvrement des blocs adjacents sans augmenter le débit. Dans la première catégorie Zhang [49] propose un algorithme de transformées combinées qui consiste à diviser l'image en deux ensembles, forte et faible corrélation et à appliquer une technique de compression sans perte sur le premier ensemble et la $TCDB$ sur le deuxième. Les techniques de la première catégorie [10, 33, 29, 49] fonctionnent pour des schémas de codage propres à chacune d'elle, leur utilisation reste restreinte dans les produits commercialisés de compression d'image tel que $JPEG$, $MPEG$ et $H26x$.
Les techniques de post-traitement semblent plus intéressantes du fait que les schémas de codage ne sont pas modifiés par leur mise en œuvre. Les effets de blocs visibles sont dûs à des composantes spatiales hautes fréquences résultats de discontinuités artificielles entre les blocs. Reeves et Lim [37] proposent d'appliquer un filtre Gaussien (3×3) passe-bas seulement sur les pixels de bord de chaque bloc pour ne pas lisser les détails à l'intérieur des blocs.
Ramamurthi et Gersho [35] appliquent une technique de filtrage de type statistique. Ils proposent un filtrage non-linéaire variant dans l'espace pour atténuer le bruit en escalier (Staircase Noise) et les effets de grille (Grid Effects). A la base des caractéristiques du système visuel humain un filtre RIF $2D$ est employé pour réduire les effets de blocs dans les régions monotones et un filtre RIF $1D$ est appliqué dans la direction parallèle aux contours pour atténuer le bruit en escalier dans les régions contenant des contours.
Stevenson [41] développe une technique de reconstruction de l'image compressée avec la $TCDB$ en utilisant un modèle de distribution de type MAP (Maximum A Posteriori probability). Dans [51, 47, 48, 50] une approche itérative est proposée pour la reconstruction de l'image au décodeur. Cette technique définit un ensemble convexe de contraintes pour représenter la régularité entre les blocs, et l'image reconstruite est obtenue par des projections alternées dans cet ensemble de contraintes et dans un autre ensemble défini par les informations *a priori* sur les coefficients $TCDB$. La complexité de ces techniques augmentent aux dépens d'une amélioration de la qualité

de l'image reconstruite [48]. Katsaggelos et al. [5, 26, 6] proposent une technique de restauration de l'image décodée basée sur la théorie des moindres carrés sous contrainte (Constrained Least Squares) introduite par Hunt [21]. Cette approche utilise pour la reconstruction une fonction objective quadratique à minimiser. Cette fonction regroupe les connaissances *a priori* sur la régularité dans l'image originale et les informations sur les coefficients $TCDB$.
Jeon et al. [24] considèrent les effets de bloc comme une perte de précision des coefficients $TCDB$ par la procédure de quantification. Ils proposent d'ajuster quelques coefficients $TCDB$ afin de compenser leur perte de précision et réduire au minimum la discontinuité des valeurs des pixels aux bords des blocs adjacents.
Dans la logique de compensation des coefficients de la $TCDB$ nous proposons une nouvelle approche sous forme d'un algorithme itératif. L'idée consiste à compenser un ensemble de coefficients $TCDB$ dans chaque bloc (8×8) afin de minimiser l'énergie d'une sélection de hautes fréquences dans le bloc de coefficients $TCDB$ (24×24) construit autour du bloc (8×8) de l'image reconstruite.

6.3 La Réduction des effets de blocs par minimisation de l'énergie des HF

6.3.1 Les Effets de blocs et Les Hautes Fréquences

La méthode la plus simple pour réduire les effets de blocs est l'élimination des discontinuités entre les blocs. Ces discontinuités comme nous l'avons indiqué sont les résultats d'une erreur de quantification qui s'ajoute au signal transformé et qui génère à la reconstruction un bruit de nature haute fréquence. Pour mettre en évidence ce bruit haute fréquence, une $TCDB$ (24×24) est appliquée sur un macrobloc (9 blocs de 8×8) de l'image non-corrigée où on suppose que seul le bloc central produit la discontinuité avec son voisinage. Les coefficients correspondant aux hautes sous-bandes de la $TCDB$ (24×24) contiennent l'énergie de l'erreur de quantification du bloc central de coefficients $TCDB$ (8×8) quantifiés. On cherche à minimiser cette énergie pour réduire la discontinuité du bloc central avec ses 8 voisins, par exemple on peut modifier quelques coefficients dans le bloc central de coefficients $TCDB$ (8×8) quantifiés. La sensibilité du Système Visuel Humain aux basses fréquences justifie le choix de compenser les coefficients des basses sous-bandes car les effets de blocs sont perçus comme du bruit haute fréquence. La réduction des effets de blocs consistera alors à calculer le vecteur de compensation du bloc central selon un critère de minimisation d'énergie et sous la contrainte de l'amplitude du vecteur en question.

$\mathbf{X}$

X_{k-l-1}	X_{k-l}	X_{k-l+1}
X_{k-1}	X_k	X_{k+1}
X_{k+l-1}	X_{k+l}	X_{k+l+1}

Figure 6.1: La fenêtre d'observation $\mathbf{X}$ formée du bloc X_k et de son voisinage de 8 blocs.

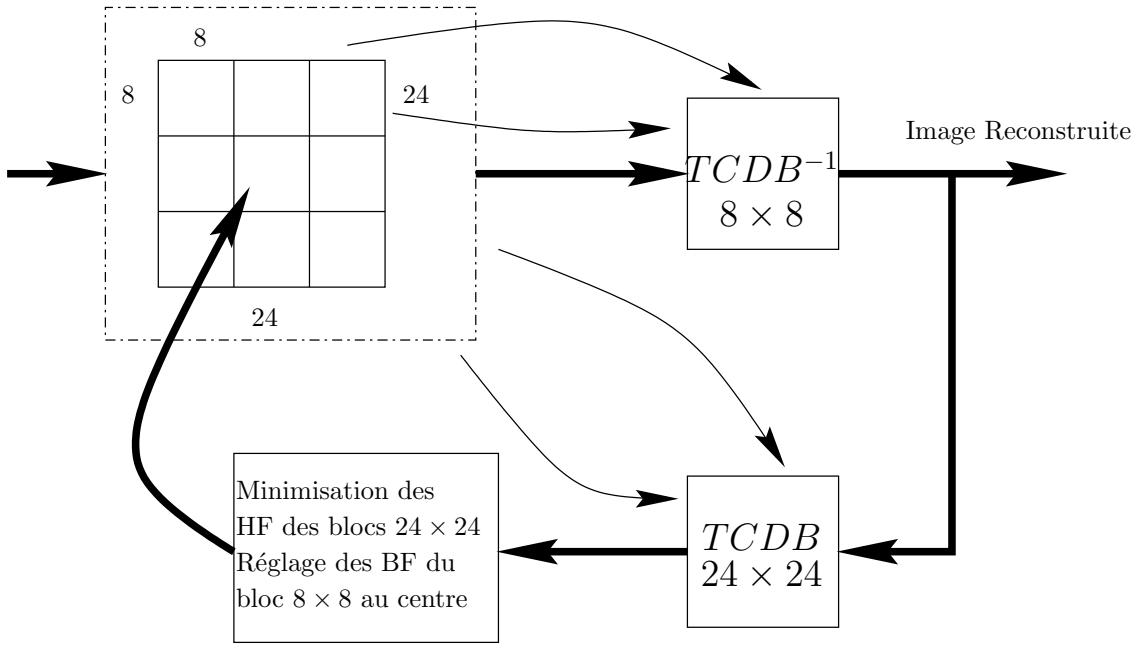

Figure 6.2: Réduction des effets de blocs par minimisation de l'énergie de Hautes Fréquences.

6.3.2 Formulations

Au décodeur, l'image déquantifiée de dimensions $(M \times N)$ est divisée en $K = (M/8 \times N/8)$ blocs de (8×8) coefficients, l est le nombre de blocs par ligne $l = N/8$. On considère la fenêtre d'observation formée du bloc X_k $(1 \leq k \leq K)$ au centre et des huit blocs de son voisinage (cf. figure 6.1). Soit $\mathbf{X}$ le nouveau bloc de (24×24) éléments. On procède à la reconstruction des blocs (8×8) de $\mathbf{X}$ par une transformée de synthèse ($TCDB^{-1}$ ou $MMSE$ 8×8). L'image reconstruite est de nouveau transformée par une Transformée en Cosinus Discrète des Blocs Directe $TCDBD$ (24×24). L'idée est de compenser l'erreur de quantification ajoutée au bloc X_k de $\mathbf{X}$ par un signal de compensation, sous la condition de réduire l'énergie d'une sélection de hautes fréquences dans le bloc compensé (24×24). La compensation portera sur les $(ne \times ne)$ coefficients situés en haut à gauche dans un bloc de (8×8) (le coefficient DC et quelques coefficients AC) où le choix de la valeur de ne est justifié ci-après.
Le signal de compensation du bloc X_k est la matrice

$$\Omega = \{\omega_{(i,j)} \; i,j \in [1,8]\} \begin{cases} \omega_{(i,j)} = 0 & ne+1 \leq i,j \leq 8 \\ -q_{(i,j)}/n < \omega_{(i,j)} < +q_{(i,j)}/n & 1 \leq i,j \leq ne \end{cases}$$

où $Q = \{q_{(i,j)} \; i,j \in [1,8]\}$ est la matrice de quantification et n est une variable paramétrable qui permet de définir le degré de lissage de l'image $(n \geq 2)$.
Comme convenu dans le chapitre 2, chaque bloc de $(m \times m)$ coefficients est traité comme un vecteur dans l'espace R^{m^2} et la transformée en bloc inverse sera linéaire de R^{m^2} dans R^{m^2}. On utilise le symbole "(:)" pour désigner la conversion d'une matrice bidimensionnelle $(2D)$ en un vecteur unidimensionnel $(1D)$. Soit $\mathbf{T}$ la matrice (64×64) associée à la tranformée de synthèse des blocs (8×8). La reconstruction du vecteur $\mathbf{X(:)}$ de dimension (576×1) est donc exprimée par $\mathbf{Y}(:) = \mathbf{A}\mathbf{X}(:)$ où $\mathbf{A}$ est la matrice spéciale de dimension $(24^2 \times 24^2)$ telle que

$$\mathbf{A} = \begin{pmatrix} \mathbf{T} & 0_{64} & 0_{64} & 0_{64} & 0_{64} & 0_{64} & 0_{64} & 0_{64} \\ 0_{64} & \mathbf{T} & 0_{64} & 0_{64} & 0_{64} & 0_{64} & 0_{64} & 0_{64} \\ 0_{64} & 0_{64} & \mathbf{T} & 0_{64} & 0_{64} & 0_{64} & 0_{64} & 0_{64} \\ 0_{64} & 0_{64} & 0_{64} & \mathbf{T} & 0_{64} & 0_{64} & 0_{64} & 0_{64} \\ 0_{64} & 0_{64} & 0_{64} & 0_{64} & \mathbf{T} & 0_{64} & 0_{64} & 0_{64} \\ 0_{64} & 0_{64} & 0_{64} & 0_{64} & 0_{64} & \mathbf{T} & 0_{64} & 0_{64} \\ 0_{64} & 0_{64} & 0_{64} & 0_{64} & 0_{64} & 0_{64} & \mathbf{T} & 0_{64} \\ 0_{64} & 0_{64} & 0_{64} & 0_{64} & 0_{64} & 0_{64} & 0_{64} & \mathbf{T} \end{pmatrix}$$

et 0_{64} est la matrice Nulle de dimension (64×64).
La matrice associée à la $TCDBD$ (24×24) utilisée pour transformer le bloc $\mathbf{Y}$ est $Ak_{576} = kron(A_{24}, A_{24})$ où A_{24} est la matrice associée à la TCD $(1D)$ de R^{24} dans R^{24}. $\mathbf{Z}(:) = Ak_{576}\mathbf{Y}(:)$.
Pour réduire l'énergie des hautes fréquences dans le bloc $\mathbf{Z}$, nous sélectionnons les

$(ns \times ns)$ plus hautes fréquences $H.F$ situées en bas à droite du bloc $\mathbf{Z}$. Le masque S sera utilisé pour sélectionner les ns hautes fréquences $H.F$ dans un vecteur de R^{24}, alors que S_{576} est le mask pour sélectionner $(ns \times ns)$ $H.F$ dans le bloc $\mathbf{Z}$ (24×24) dont on cherche à minimiser l'énergie.

$$S = \begin{pmatrix} 0_{(24-ns)\times(24-ns)} & 0_{(24-ns)\times ns} \\ 0_{ns\times(24-ns)} & I_{ns\times ns} \end{pmatrix}$$
$$S_{576} = \begin{pmatrix} diag_{24-ns}(S, S, S, ...S) & 0_{24}...0_{24} \\ 0_{24}...0_{24} & diag_{ns}(I_{24}, I_{24}, I_{24}, ...I_{24}) \end{pmatrix}$$

0_{24} et I_{24} représentent respectivement les matrices Nulle et Identité de même dimension que S.
Nous définissons la matrice $\mathbf{O}$ et le bloc $\mathbf{X}$ tel que

$$\mathbf{O}(:) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \Omega(:) \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \mathbf{X}(:) = \begin{pmatrix} X_{k-l-1}(:) \\ X_{k-l}(:) \\ X_{k-l+1}(:) \\ X_{k-1}(:) \\ X_k(:) \\ X_{k+1}(:) \\ X_{k+l-1}(:) \\ X_{k+l}(:) \\ X_{k+l+1}(:) \end{pmatrix}$$

$$\begin{aligned}
\hat{\mathbf{X}}(:) &= \mathbf{X}(:) + \mathbf{O}(:); && \text{l'équivalent compensé du bloc } \mathbf{X} \\
\hat{\mathbf{Y}}(:) &= \mathbf{A}\hat{\mathbf{X}}(:); && \text{la reconstruction du bloc compensé } \hat{\mathbf{X}} \\
\hat{\mathbf{Z}}(:) &= Ak_{576}\hat{\mathbf{Y}}(:); && \text{le bloc transformé de } \hat{\mathbf{Y}} \text{ par } TCDBD\ (24 \times 24) \\
\hat{\mathbf{Z}}_{HF}(:) &= S_{576}Ak_{576}\mathbf{A}(\mathbf{X}(:) + \mathbf{O}(:)); && \text{une sélection de } (ns \times ns)\ HF \text{ de } \hat{\mathbf{Z}}
\end{aligned}$$

Ainsi l'énergie des $(ns \times ns)$ HF dans $\hat{\mathbf{Z}}_{HF}(:)$ est

$$\begin{aligned}
\left[(\hat{\mathbf{Z}}_{HF}(:))^t\hat{\mathbf{Z}}_{HF}(:)\right] &= (\mathbf{X}(:) + \mathbf{O}(:))^t\mathbf{A}^t Ak_{576}^t S_{576}^t S_{576} Ak_{576}\mathbf{A}(\mathbf{X}(:) + \mathbf{O}(:)); \\
\mathbf{H} &= \mathbf{A}^t Ak_{576}^t S_{576}^t S_{576} Ak_{576}\mathbf{A}; \\
\mathbf{H}^t &= \mathbf{H}; \\
\left[(\hat{\mathbf{Z}}_{HF}(:))^t\hat{\mathbf{Z}}_{HF}(:)\right] &= (\mathbf{X}(:))^t\mathbf{H}\mathbf{X}(:) + (\mathbf{X}(:))^t\mathbf{H}\mathbf{O}(:) + (\mathbf{O}(:))^t\mathbf{H}\mathbf{X}(:) + \\
&\quad (\mathbf{O}(:))^t\mathbf{H}\mathbf{O}(:); && (6.1) \\
\mathbf{G} &= \mathbf{H}\mathbf{X}(:); && (6.2) \\
\left[(\hat{\mathbf{Z}}_{HF}(:))^t\hat{\mathbf{Z}}_{HF}(:)\right] &= cte + \mathbf{G}^t\mathbf{O}(:) + (\mathbf{O}(:))^t\mathbf{G} + (\mathbf{O}(:))^t\mathbf{H}\mathbf{O}(:);
\end{aligned}$$

Soit $\mathcal{G}$ un sous-vecteur de $\mathbf{G}$ de dimension (64×1) tel que

$$\begin{aligned} \mathcal{G} &= \mathbf{G}(64*4+1:64*5); \\ \mathbf{G}^t\mathbf{O}(:) &= \mathcal{G}^t\Omega(:); \end{aligned}$$

Comme les coefficients $X_k(i,j)$ pour $1 \leq i,j \leq ne$ sont les seuls à compenser, nous définissons o de dimension $(ne \times ne)$ la sous-matrice de Ω qui contient seulement les $\omega_{(i,j)}$ $(1 \leq i,j \leq ne)$ non-nulles, et g le sous-vecteur de $\mathcal{G}$ qui respecte la relation suivante:

$$g^t o(:) = \mathcal{G}^t\Omega(:);$$

Pour la matrice $\mathbf{H}$ nous réalisons la même opération de troncature pour avoir la sous-matrice h de dimension $(ne^2 \times ne^2)$ telle que

$$(o(:))^t h o(:) = (\mathbf{O}(:))^t \mathbf{H}\mathbf{O}(:);$$

L'équation de l'énergie des $(ns \times ns)$ HF devient

$$\left[(\hat{\mathbf{Z}}_{HF}(:))^t\hat{\mathbf{Z}}_{HF}(:)\right] = cte + 2\sum_{i=1}^{ne^2} g_{(i)}\ o(:)_{(i)} + \sum_{i=1}^{ne^2}\sum_{j=1}^{ne^2} o(:)_{(i)}\ o(:)_{(j)}\ h_{(i,j)} \quad (6.3)$$

Il est claire que l'équation (6.3) contient une forme quadratique dont la solution sous les contraintes

$$-q_{(i,j)}/n < o_{(i,j)} < +q_{(i,j)}/n \qquad 1 \leq i,j \leq ne \quad (6.4)$$

où $n \geq 2$ et

$$\mathbf{T}(X_k(:) + \Omega(:)) \geq 0 \quad (6.5)$$

est le résultat d'une approximation quadratique

6.3.3 L'Algorithme B1 de réduction des effets de blocs dans un schéma de codage par transformée

La technique proposée se concrétise par l'algorithme **B1** de réduction des effets de blocs. Cet algorithme de post-filtrage est itératif et est constitué des 5 étapes suivantes:

1. Sélectionner dans le bloc X_k les $(ne \times ne)$ coefficients à compenser et dans le bloc $\mathbf{X}$ les $(ns \times ns)$ hautes fréquences dont on cherche à minimiser l'énergie;

2. Calculer S et S_{576} en fonction de "ns", ensuite les matrices $\mathbf{H}$ et h en fonction de "ne";

3. Pour le bloc au centre X_k avec les équations (6.1) et (6.2) trouver l'équation (6.3);

4. Résoudre le problème d'approximation quadratique de (6.3) sous les contraintes (6.4) et (6.5) pour calculer les ω du bloc X_k;

5. Déplacer la fenêtre d'observation $\mathbf{X}$ d'un bloc, du haut vers le bas et de gauche à droite, où pour chaque itération calculer les ω du bloc au centre X_k comme dans les étapes 3 et 4.

Notons que les blocs compensés dans des itérations précédentes sont pris en compte pour la compensation du bloc au centre X_k de la fenêtre d'observation $\mathbf{X}$ dans sa position actuelle. Les blocs aux bords de l'image ne sont pas traités dans cet algorithme de post-traitement.

6.4 Résultats d'application de l'algorithme B1

Dans le cadre d'une continuité d'optimisation et d'amélioration du $PSNR$ et de la qualité d'image décodée, et pour démontrer la performance de notre technique proposée pour la réduction des effets de bloc, nous l'avons testée sur les résultats de codage des images LENA et BARBARA par le $JPEG$ standard, le $JPEG$ avec des quantificateurs optimisés et par l'algorithme **A1** d'optimisation conjointe des quantificateurs et de la transformée de synthèse.
Les tests sont faits aux différents débits proposés aux chapitres 3 et 5.
Pour le choix des coefficients basses fréquences à compenser dans les blocs de la $TCDB$ (8×8), observons pour l'image LENA la distribution spectrale de l'erreur quadratique moyenne de quantification des coefficients de la $TCDB$ (figure 6.4) à différents débits. Il est clair que l'erreur est concentrée sur les composantes basses fréquences où on peut identifier l'ensemble de (6×6) coefficients $TCDB$ à compenser. La distorsion perceptuelle des effets de blocs est dûe essentiellement à l'erreur de quantification survenue sur les composantes fréquentielles les plus basses, on se contentera de compenser au maximum les (6×6) premiers coefficients de la TCD parmi les (8×8). Pour comparer cette technique avec d'autres, nous avons appliqué sur les images LENA et BARBARA reconstruites un filtre moyenneur de dimension (3×3) $(filtre = 1/9[1\,1\,1; 1\,1\,1; 1\,1\,1])$ qui est un cas extrême de filtre passe-bas. Avec ce filtrage, et malgré la reduction considérable des artefacts, les effets de blocs restent encore visibles sans compter l'aspect flou qui domine les images filtrées. Avec ce test, nous avons remarqué qu'il existe une certaine limite d'utilisation du filtrage linéaire pour réduire les effets de blocs. Pour comprendre, la raison pour laquelle un filtrage linéaire n'élimine pas totalement les effets de blocs, nous avons tracé le profil d'une portion de la ligne 436 de l'image LENA codée à 0.25bpp (figure 6.3). Le filtre moyenneur (3×3) utilisé réduit le changement brusque de la valeur des pixels sur les bords des blocs. Toutefois l'écart de valeurs des pixels aux bords des blocs reste

encore très important pour être imperceptible. Cet écart est à l'origine des effets de voile. Pour distribuer cet écart sur l'ensemble des pixels, on peut utiliser un filtre de plus grande dimension, mais ceci rend l'image reconstruite encore plus floue.
En revanche, la technique proposée calcule pour chaque bloc (8×8) de l'image déquantifiée un signal qui compense l'erreur de quantification afin de réduire la présence des hautes fréquences dans une fenêtre d'observation plus large (24×24) construite autour du bloc à compenser. La compensation de l'erreur de quantification est conditionnée par l'équation (6.4) où la valeur de n augmente avec les fréquences les plus hautes (n_{dc} pour le coefficient **DC**, n_{ac} et n_{ac1} pour les coefficients **AC**) figure 6.5.
Les tables (6.1, 6.2) montrent les résultats en $PSNR$ de l'application de l'algorithme **B1** respectivement sur les images LENA et BARBARA, codées (dans le chapitre 5) par le $JPEG$ standard, le $JPEG$ et ses quantificateurs optimisés et par l'algorithme **A1** d'optimisation conjointe. Dans les trois cas et pour les deux images, en prenant le meilleur $PSNR$ comme critère de sélection, l'algorithme **B1**, quand il est appliqué, réduit considérablement les discontinuités entre les blocs sans pouvoir les éliminer complètement comme le montre l'exemple de l'image LENA à 0.25bpp (figures 6.9 et 6.13).
Cependant il est possible d'obtenir une meilleure qualité visuelle avec un autre réglage des paramètres ne, ns, n_{dc}, n_{ac} et n_{ac1}. En effet pour obtenir la meilleure qualité visuelle nous avons appliqué l'algorithme **B1** avec des valeurs de ne, n_{dc}, n_{ac} et n_{ac1} qui autorisent une compensation plus importante et dégressive pour un nombre réduit des coefficients basses fréquences de la $TCDB$ (8×8). Les images reconstruites avec le critère visuel montrent que l'algorithme **B1** effectue un lissage dans les zones correspondantes aux basses fréquences de l'image d'origine, perturbées par la forte quantification, sans trop affecter les zones de hautes fréquences de l'image. La technique proposée de post-traitement fournit, dans ce cas, une qualité visuelle largement supérieure à celle trouvée en cherchant le meilleur $PSNR$ pour la même image et le même débit, malgré une légère baisse de la valeur du $PSNR$ (figures 6.9, 6.13, 6.24 et 6.28). Les expériences ont déjà montré que la mesure de la qualité visuelle par le $PSNR$ n'est pas forcement adéquate.
Pour les tests effectués sur les images LENA et BARBARA, compenser les (2×2) coefficients basses fréquences de la $TCDB$ (8×8) avec un critère qualitatif et subjectif ou les (5×5) avec un critère quantitatif ($PSNR$), en minimisant l'énergie des (16×16) sous-bandes les plus hautes de la $TCDB$ (24×24) est un bon choix pour lequel le post-traitement est très performant avec un temps de calcul relativement faible.
La technique proposée dans l'algorithme **B1** est d'autant plus performante pour les images codées avec une transformée (16×16) que pour celles codées avec une $TCDB$ (8×8). En effet, il est évident que la décomposition de l'image en blocs de (16×16) génère, pour le même débit, moins de bruit hautes fréquences que dans le cas des blocs (8×8), du fait que la taille des blocs est 4 fois plus large et que le système

visuel humain est moins sensible dans ce cas. Nous avons remplacé les $TCDB$ et $TCDBI$ (8×8) dans $JPEG$ standard par des $TCDB$ et $TCDBI$ (16×16) respectivement. Les $TCDB$ (16×16) avec des quantificateurs optimisés nous permettent de gagner plus de 0.6dB sur LENA à 0.25bpp (figure 6.14) par rapport au cas avec les $TCDB$ (8×8) et les quantificateurs optimisés. En appliquant l'algorithme **A1** pour l'optimisation conjointe des quantificateurs et de la transformée de synthèse (16×16), LENA à 0.25bpp (figure 6.17) gagne plus de 1.12dB par rapport au cas de l'algorithme **A1** appliqué sur des blocs (8×8). L'utilisation d'une trasnformée de synthèse (16×16) $TCDBI$ ou $MMSE$ donne une meilleure restitution des zones à hautes fréquences et moins des effets de blocs dans les zones à basses fréquences, sans oublier toutefois que le banc de filtres de Wiener réduit davantage la distorsion par rapport à une simple $TCDBI$ (16×16).
Les avantages du domaine de transformée (16×16) ne se limitent pas à l'algorithme **A1**. L'application de l'algorithme **B1** sur les figures 6.14 et 6.17 (figures 6.16 et 6.19) pour le critère de meilleure qualité visuelle ($ne = 4$, $ns = 32$, $n_{dc} = 2$, $n_{ac} = 4$ et $n_{ac1} = 8$) réduit énormement les discontinuités entre les blocs dans les zones basses fréquences.

6.4.1 Comparaison de l'Algorithme SPIHT avec A1 + B1

Cette comparaison porte sur les résultats de codage des images LENA et BARBARA par le codeur $SPIHT$ [39] téléchargé sur le site *www.cipr.rpi.edu/research/SPIHT* et par l'application des algorithmes **A1** et **B1**.
En effet $SPIHT$ est l'état de l'art des techniques de codage d'image par les ondelettes. Cet algorithme permet de coder et reconstruire une image quasiment identique à l'originale au débit le plus faible. Comme les deux techniques ($SPIHT$ et $TCDB$ optimisée) sont complètement différentes, la comparaison entre elles tient uniquement sur l'aspect visuel des images codées et la valeur du $PSNR$. La figure 6.20 montre LENA codée par $SPIHT$ à 0.25bpp ($PSNR = 32.17dB$). Les zones à basses fréquences sont restituées avec $SPIHT$ sans dégradation remarquable, alors que le traitement individuel des blocs avec la $TCDB$ (algorithme **A1**) génère une dégradation de la qualité (effets de blocs) que l'algorithme **B1** corrige à 90% en lissant la transition entre les blocs. L'utilisation des transformées du domaine fréquentiel (16×16) pour modérer la sensibilité du système visuel humain, facilite la reconstruction des détails à hautes fréquences, que $SPIHT$ ne le permet pas. Nous avons remarqué que les contours sont plus lisses avec Les algorithmes **A1** et **B1** tandis que $SPIHT$ reconstruit les mêmes contours avec un effet de marches d'escalier.
En comparant le codage de BARBARA par **A1** + **B1** (figure 6.29) et par SPIHT (figure 6.30) à 0.25bpp, nous avons constaté que l'optimisation **A1** + **B1** ressort mieux les zones à hautes fréquences (nappe, pantalon, reliures des livres dans la biblithèque, etc...) que $SPIHT$, alors que ce dernier permet une reconstruction

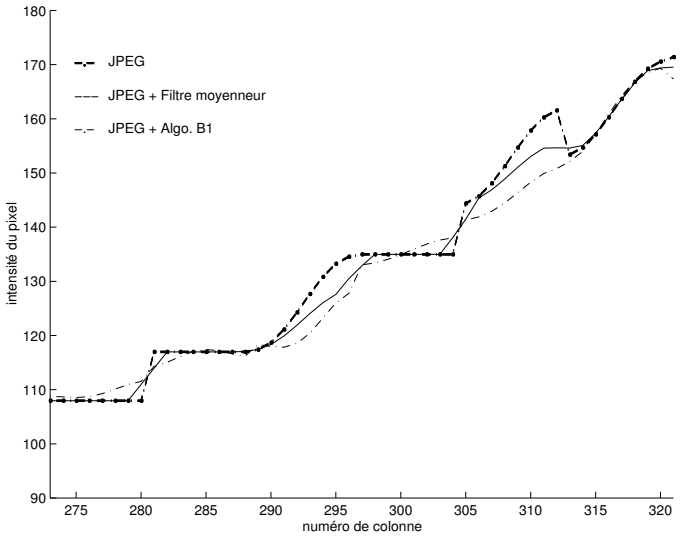

Figure 6.3: Profile d'une portion de la ligne 436 de l'image "Lena" codée à 0.25bpp avec $JPEG$ standard, $JPEG$ + un filtre moyenneur et $JPEG$ + post-traitement de l'algo. **B1**.

	Lena $PSNR$ **(dB)** $ne = 5$, $ns = 16$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$					
débit	$JPEG$		**Quantificateurs optimaux**		**Optimisation conjointe**	
(bpp)	**Standard**	**(Std + Algo. B1)**	$JPEG$	$JPEG$ **+ Algo. B1**	**Algo. A1**	**Algo. A1 + B1**
0.15	27.11	27.67	27.38	27.80	27.94	28.21
0.25	29.70	30.07	29.92	30.27	30.19	30.38
0.50	32.79	32.93	32.95	33.10	33.35	33.41
0.75	34.43	34.51	34.76	34.85	35.04	35.07
1.00	35.65	35.71	36.04	36.17	36.30	36.33

Table 6.1: Comparaison du codage par l'algorithme **A1** de l'image "Lena" avec le $JPEG$ standard, le $JPEG$ et une optimisation de ses quantificateurs, avant et après l'application de l'algorithme **B1** (critère du meilleur $PSNR$)

plus lisse des zones à basses fréquences.
Enfin, $SPIHT$ réalise comme prévu un meilleur compromis entre le $PSNR$ et la qualité subjective.

6.5 Conclusion

Dans ce chapitre nous avons proposé une nouvelle technique de post traitement pour la réduction des effets de blocs en minimisant la discontinuité entre les blocs de l'image codée avec Codeur à transformée. La méthode proposée calcule pour chaque bloc de coefficients $TCDB$ (8×8) un vecteur de compensation qui permet après le post-traitement d'atténuer la discontinuité de ce bloc avec ses 8 voisins. Le calcul du vecteur de compensation est basé sur la minimisation de l'énergie d'une sélection des hautes fréquences dans le domaine de la $TCDB$ (24×24).

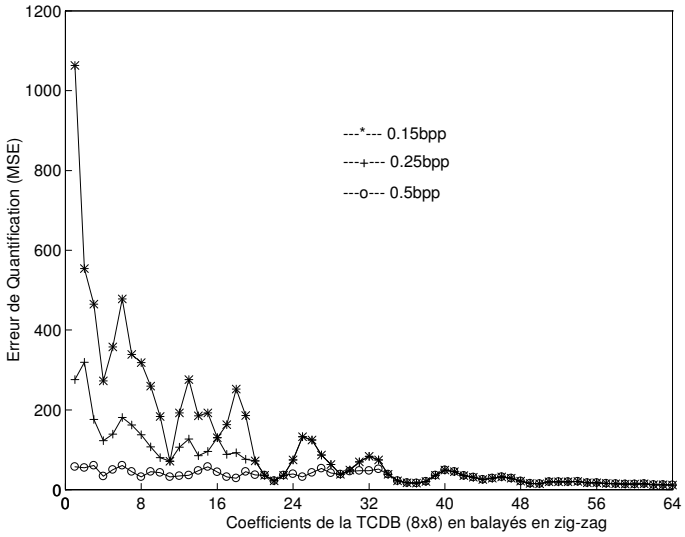

Figure 6.4: Erreur Quadratique Moyenne de Quantification des coefficients de la $TCDB$ (8×8) en ordre Zig-Zag, calculée pour L'image "Lena" à 0.15, 0.25 et 0.50bpp.

n_{dc}	n_{ac}	n_{ac1}	n_{ac1}	...	...	...	...
n_{ac}	n_{ac}	n_{ac1}	n_{ac1}	...	...	...	...
n_{ac1}	n_{ac1}	n_{ac1}	n_{ac1}	...	...	...	...
n_{ac1}	n_{ac1}	n_{ac1}	n_{ac1}	...	...	...	...
...	...	...	...	...	...	...	...

Figure 6.5: Variation de n selon la fréquence dans le bloc X_k de (8×8).

Figure 6.6: "Lena" Codée par JPEG avec Quantificateurs optimisés à 0.25bpp ($PSNR = 29.92dB$)

Figure 6.7: "Lena" Codée par JPEG avec Quantificateurs optimisés + filtre moyenneur à 0.25bpp

Figure 6.8: "Lena" Codée par JPEG avec Quantificateurs optimisés à 0.25bpp + **B1** (meilleur $PSNR$, $ne = 5$, $ns = 16$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$, $PSNR = 30.27dB$)

Figure 6.9: "Lena" Codée par JPEG avec Quantificateurs optimisés à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 2$, $ns = 16$, $n_{dc} = 2$, $n_{ac} = 4$, $PSNR = 29.80dB$)

Figure 6.10: "Lena" Codée par l'Algo. **A1** à 0.25bpp ($PSNR = 30.19dB$)

Figure 6.11: "Lena" Codée par l'Algo. **A1** + filtre moyenneur à 0.25bpp

Figure 6.12: "Lena" Codée par l'Algo. **A1** à 0.25bpp + **B1** (meilleur $PSNR$, $ne = 5$, $ns = 16$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$, $PSNR = 30.38dB$)

Figure 6.13: "Lena" Codée par l'Algo. **A1** à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 2$, $ns = 16$, $n_{dc} = 2$, $n_{ac} = 2$, $PSNR = 29.76dB$)

Figure 6.14: "Lena" Codée par JPEG ($TCDB$ 16×16) avec Quantificateurs optimisés à 0.25bpp ($PSNR = 30.54dB$)

Figure 6.15: "Lena" Codée par JPEG ($TCDB$ 16×16) avec Quantificateurs optimisés à 0.25bpp + **B1** (meilleur $PSNR$, $ne = 6$, $ns = 32$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$, $PSNR = 30.62dB$)

Figure 6.16: "Lena" Codée par JPEG ($TCDB$ 16×16) avec Quantificateurs optimisés à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 4$, $ns = 32$, $n_{dc} = 2$, $n_{ac} = 4$, $n_{ac1} = 8$, $PSNR = 30.45dB$)

Figure 6.17: "Lena" Codée par l'Algo. **A1** (16×16) à 0.25bpp ($PSNR = 31.31dB$)

Figure 6.18: "Lena" Codée par l'Algo. **A1** (16×16) à 0.25bpp + **B1** (meilleur $PSNR$, $ne = 6$, $ns = 32$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$, $PSNR = 31.34dB$)

	Barbara $PSNR$ **(dB)** $ne = 5$, $ns = 16$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$					
débit	$JPEG$		**Quantificateurs optimaux**		**Optimisation conjointe**	
(bpp)	**Standard**	**(Std + Algo. B1)**	$JPEG$	$JPEG$ + **Algo. B1**	**Algo. A1**	**Algo. A1 + B1**
0.15	24.14	24.46	24.58	24.83	25.22	25.38
0.25	25.97	26.18	26.40	26.63	27.33	27.47
0.50	30.14	30.26	30.48	30.66	31.18	31.24
0.75	32.19	32.29	33.28	33.38	34.12	34.18
1.00	34.45	34.52	35.60	35.78	36.36	36.40

Table 6.2: Comparaison du codage par l'algorithme **A1** de l'image "Barbara" avec le $JPEG$ standard, le $JPEG$ et une optimisation de ses quantificateurs, avant et après l'application de l'algorithme **B1** (critère du meilleur $PSNR$)

Deux critères sont utilisés pour évaluer les performances de l'algorithme, le $PSNR$ et la qualité visuelle des images traitées.
La technique proposée est trés simple et n'exige aucune connaissance *a priori* de l'image à traiter. Avec un Codeur $JPEG$ et un décodeur optimisé (algorithmes **A1** et **B1**), nous avons montré sa puissance en réduction des effets de blocs et en amélioration de la qualité visuelle des images décodées sans sacrifier de leur netteté. Une comparaison des résultats obtenus avec **A1** + **B1** et ceux du codage $SPIHT$ montre que les qualités des images codées avec ces deux techniques restent comparables même si $SPIHT$ réalise en général un meilleur compromis $PSNR$/Qualité subjective.

Figure 6.19: "Lena" Codée par l'Algo. **A1** (16×16) à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 4$, $ns = 32$, $n_{dc} = 2$, $n_{ac} = 4$, $n_{ac1} = 8$, $PSNR = 31.06dB$)

Figure 6.20: "Lena" Codée par SPIHT à 0.25bpp, $PSNR = 32.17dB$

Figure 6.21: "Barbara" Codée par JPEG avec Quantificateurs optimisés à 0.25bpp ($PSNR = 26.40dB$)

Figure 6.22: "Barbara" Codée par JPEG avec Quantificateurs optimisés + filtre moyenneur à 0.25bpp

Figure 6.23: "Barbara" Codée par JPEG avec Quantificateurs optimisés à 0.25bpp + **B1** (meilleur $PSNR$, $ne = 5$, $ns = 16$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$, $PSNR = 26.63dB$)

Figure 6.24: "Barbara" Codée par JPEG avec Quantificateurs optimisés à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 2$, $ns = 16$, $n_{dc} = 2$, $n_{ac} = 4$, $PSNR = 26.25dB$)

Figure 6.25: "Barbara" Codée par l'Algo. **A1** à 0.25bpp ($PSNR = 27.32dB$)

Figure 6.26: "Barbara" Codée par l'Algo. **A1** + filtre moyenneur à 0.25bpp

Figure 6.27: "Barbara" Codée par l'Algo. **A1** à 0.25bpp + **B1** (meilleur $PSNR$, $ne = 5$, $ns = 16$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$, $PSNR = 27.47dB$)

Figure 6.28: "Barbara" Codée par l'Algo. **A1** à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 2$, $ns = 16$, $n_{dc} = 2$, $n_{ac} = 2$, $PSNR = 27.14dB$)

Figure 6.29: "Barbara" Codée par l'Algo. **A1** à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 2$, $ns = 16$, $n_{dc} = 2$, $n_{ac} = 2$, $PSNR = 27.14dB$)

Figure 6.30: "Barbara" Codée par SPIHT à 0.25bpp, $PSNR = 28.19dB$

Conclusions Générales et Perspectives

La nécessité de comprimer des données de plus en plus volumineuses et difficiles à transmettre ou à stocker est accrue avec le développement des techniques de communications. Les performances d'un système de compression sont évaluées par sa complexité calculatoire, son coût de stockage, le débit (ou le taux de compression) qu'il permet d'atteindre et la dégradation qu'il engendre sur les données compressées. Les techniques de compression ont fait l'objet de nombreuses recherches pour tenter d'optimiser ces différents aspects. Il demeure cependant difficile d'optimiser tous ces critères en même temps.
Au cours des années de cette thèse, nous avons traité le problème d'optimisation du schéma de compression d'image à base d'une transformée notamment la TCD et la norme $JPEG$ selon un critère d'Erreur Quadratique Moyenne Minimale $EQMM$ (ou $MMSE$). La norme $JPEG$ semble amener des réponses aux besoins des applications qui nécessitent un compactage simple et éfficace des images fixes pour des utilisations ultérieures. Ceci est vrai pour les moyens et faibles taux de compression, alors qu'à des taux assez élevés le $JPEG$ avec son implantation simple de base se trouve impuissant devant l'énorme dégradation de la qualité visuelle de l'image. Plusieurs paramètres sont à l'origine de cette dégradation visuelle de l'image, parmi les plus importants on trouve, les quantificateurs des coefficients de la TCD qui définissent le taux de compression et le banc de filtres de synthèse qui définit le mode de reconstruction (Reconstruction Parfaite ou autre). L'optimisaton de ces deux paramètres demeure une question délicate à traiter vu leurs effets directs sur la qualité de l'image codée. Un schéma de $JPEG$ optimisé de ses deux côtés Codeur et Décodeur est nécessaire pour une comparaison équitable avec d'autres techniques optimisées de compression.

Le travail présenté dans la première partie de ce mémoire, porte sur la définition d'un nouveau banc de filtres de synthèse optimisé conjointement avec une allocation de bits optimale dans le cadre d'une optimisation Débit/Distorsion. La TCD^{-1} étant à reconstruction parfaite en absence de toute quantification, n'est plus la transformée

de synthèse optimale dans le sens de distorsion minimale quand elle est employée dans un décodeur $JPEG$ de base. En effet, le calcul de la distorsion (Chapitre 3) sur une reconstruction de l'image codée avec $JPEG$ (surtout à bas débit) montre que la TCD^{-1} ne réduit jamais l'effet de l'erreur de quantification. L'écriture de la distorsion moyenne entre l'image et sa reconstruction, en fonction des filtres d'analyse et de synthèse, des coefficients TCD avant et après quantification et du bruit de quantification (qu'on suppose de nature additive), nous permet de caluler les filtres de synthèse qui minimisent la distorsion au sens de l'Erreur Quadratique Moyenne Minimale. Ceci nous conduit à la définition des filtres de Wiener. Une meilleure allocation de bits permet, pour un débit total donné, de bien redistribuer les quantificateurs sur les sous-bandes en tenant compte de la pertinence de quelques unes et de la redondance des autres. Dans le Chapitre 4 le problème d'une allocation de bits optimale est traité dans le cas de la TCD^{-1} ou d'une transformée de synthèse optimale ($MMSE$). En regroupant les deux optimisations dans l'algorithme itératif **A1** où dans chacune de ses deux principales étapes nous considérons un ensemble de paramètres fixes et nous optimisons l'autre ensemble, nous avons montré dans le Chapitre 5 l'amélioration de la qualié visuelle et de la valeur du $PSNR$ pour un ensemble d'images de test. Une extension de la norme $JPEG$ [1] prévoit la quantification variable à l'échelle des blocs. Par analogie, nous calculons des facteurs d'échelle de la matrice de quantification (globale) préalablement optimisée conjointement avec le banc de filtres de synthèse en appliquant l'algorithme **A1**. Le calcul de ces facteurs d'échelle est réalisé par un algorithme d'optimisation conjointe **A2** avec la transformée de synthèse encore selon un critère de l'Erreur Quadratique Moyenne Minimale ($MMSE$). Les résultats d'application de l'algorithme **A2** sur les images de test montrent une légère amélioration de la qualité d'image face à une quantité de calcul qui nécessite encore une optimisation algorithmique.
Dans le cadre de la première partie de cette thèse, plusieurs points sont encore nécessaires à développer. Certains de ces points concernent l'aspect algorithmique des techniques présentées. D'autres points sont induits par l'application des algorithmes **A1** et **A2** dans un contexte d'images vidéo, à savoir le développement d'un algorithme d'apprentissage, qui dépend de l'ensemble des séquences à coder, pour calculer une transformée de synthèse $MMSE$ et une table de quantification optimisées par rapport au contenu des images de la séquence (avec les algorithmes **A1** et **A2**) périodiquement et d'une façon adaptative.

La deuxième partie de ce mémoire de thèse, est consacrée au problème de correction des effets de blocs observés sur les images codées par le $JPEG$ (ou autre codeur à base de la $TCDB$) à faible et très faible débit. Les effets de blocs, auxquels le système visuel humain est sensible, sont dûs essentiellement à des discontinuités entre les blocs de l'image traités individuellement. La phase de quantification des coefficients de l'image transformée entraine des conséquences irréversibles sur ces coefficients, la perte de leur précision initiale. Cette modification des valeurs des

coefficients $TCDB$ se transforme en une variation sur les niveaux des pixels de l'image à la reconstruction. Une variation brusque des valeurs des pixels sur les bords d'un bloc par rapport à son voisinage est considérée comme une accentuation des hautes fréquences dans l'image entière. En minimisant l'énergie de ces présumées hautes fréquences on doit pouvoir diminuer les discontinuités entre les blocs et ainsi réduire les effets de blocs. Nous avons proposé un algorithme **B1** qui en minimisant l'énergie d'une sélection de hautes fréquences dans une fenêtre fréquentielle (24×24) permet de compenser la perte de définition d'un ensemble de coefficients basses fréquences dans l'image quantifiée. Le calcul de ce vecteur de compensation est basé sur une approximation quadratique dont l'optimisation reste à développer. Les résulats d'application de cette technique, qui n'exige aucune information au préalable sur les statistiques de l'image à corriger, sont montrés au chapitre 6.
Une meilleure application de l'algorithme **B1** serait celle qui s'adapterait aux informations dans l'image (Zones homogènes, contours, ...) en ajustant le nombre de coefficients basses fréquences dans les blocs (8×8) de l'image quantifiée, le nombre de hautes fréquences dont on minimise l'énergie et la dynamique du vecteur de compensation.
La combinaison des algorithmes **A1** et **B1** garantit une optimisation du système de codage à base de la $TCDB$, qui se traduit par une valeur plus élevée du $PSNR$ et une meilleure qualité de l'image à la reconstruction. Les tests effectués sur LENA montre qu'on gagne plus de 0.8dB par rapport au $JPEG$ de base et moins des effets de blocs dans les zones à basses fréquences (visage, épaule, miroir, etc...). Quand à l'image BARBARA, le gain en $PSNR$ de l'application des algorithmes **A1** et **B1** par rapport au $JPEG$ de base est de l'ordre de 2dB et les zones à hautes fréquences (foulard, nappe de table, etc...) sont mieux restituées avec notre optimisation.
En comparaison avec $SPIHT$ nous avons remarqué l'avantage de notre optimisation (**A1** + **B1**) à restituer des zones à hautes fréquences avec plus de détails et de contours plus lisses, que l'algorithme progressif à base des ondellettes les reconstruit respectivement avec moins de précisions et un effet de marches d'escalier. Toutefois $SPIHT$ reste plus performant pour trouver un compromis entre le $PSNR$ et la qualité visuelle de l'image codée.

Liste des Figures

2.1 Schéma de Codage/Décodage 8
2.2 Banc de filtres avec sous-échantillonnage et sur-échantillonnage critiques 14
2.3 Schéma équivalent d'un banc de filtres d'analyse et de synthèse lorsque $N = M$. 17
2.4 Quantification scalaire uniforme 18
2.5 Quantification scalaire non-uniforme 19
2.6 Quantification Vectorielle 22
2.7 Débit/Distorsion et alloctaion de bits. (a) Courbe R/D théorique (en continue) et courbe opérationnelle (en pointillé). (b) Solution optimale à pente $-\lambda$ constante. 24
2.8 Exemple de Codage par plage RLE 27
2.9 Codage par plage des coefficients AC dans JPEG 28
2.10 L'arbre binaire de l'algorithme de **Huffman** 31
2.11 Illustration graphique du codage **Arithmétique** 34
2.12 Schéma bloc du Codeur/Décodeur $JPEG$ 36

3.1 Schéma de Codage/Décodage par TCD 42
3.2 Codage par transformée 44
3.3 Rapport Signal/Bruit PSNR (dB) en sortie des décodeurs (TCD^{-1} et $MMSE$) en fonction du débit total alloué (bpp) de l'image "Lena". 50
3.4 Rapport Signal/Bruit PSNR (dB) en sortie des décodeurs (TCD^{-1} et $MMSE$) en fonction du débit total alloué (bpp) de l'image "Barbara". 51
3.5 Images originales LENA(512×512) et BARBARA(576×720) 52
3.6 Images originales GOLDHILL(512×512) et BOATS(576×720) 53
3.7 Image originales DESK(256×256) et ZELDA(576×720) 53

4.1 Courbe Débit/Distorsion opérationnelle et son enveloppe convexe. 59
4.2 Interprétation graphique de l'algorithme d'allocation de bits. 60

5.1 L'arbre binaire complété de l'algorithme de **Huffman** 71
5.2 "Lena" codée à 0.25bpp par le JPEG de base PSNR=29.70dB 77
5.3 "Lena" codée à 0.25bpp par le JPEG avec des quantificateurs optimisés PSNR=29.92dB 78
5.4 "Lena" codée à 0.25bpp par l'algorithme **A1** PSNR=30.19dB 79
5.5 "Lena" codée à 0.25bpp par l'algorithme **A2** PSNR=30.30dB 80

5.6 Vue rapprochée de l'image "Barbara" codée à 0.25bpp par le JPEG de base PSNR=25.97dB . . . 81
5.7 Barbara codée à 0.25bpp par le JPEG avec des quantificateurs optimisés PSNR=26.40dB . . . 82
5.8 Barbara codée à 0.25bpp par l'algorithme **A1** PSNR=27.33dB . . . 83
5.9 Barbara codée à 0.25bpp par l'algorithme **A2** PSNR=27.35dB . . . 84
5.10 Rapport Signal/Bruit PSNR (dB) de l'image "Lena" codée avec le $JPEG$ de base, le $JPEG+$ Quantificateurs Optimisés et l'algorithme **A1**, en fonction du débit total alloué. . . . 85
5.11 Rapport Signal/Bruit PSNR (dB) de l'image "Lena" codée avec le $JPEG$ de base, le $JPEG+$ Quantification Variable et l'algorithme **A2**, en fonction du débit total alloué. . . . 85
5.12 Rapport Signal/Bruit PSNR (dB) de l'image "Barbara" codée avec le $JPEG$ de base, le $JPEG+$ Quantificateurs Optimisés et l'algorithme **A1**, en fonction du débit total alloué. . . . 86
5.13 Rapport Signal/Bruit PSNR (dB) de l'image "Barbara" codée avec le $JPEG$ de base, le $JPEG+$ Quantification Variable et l'algorithme **A2**, en fonction du débit total alloué. . . . 86

6.1 La fenêtre d'observation **X** formée du bloc X_k et de son voisinage de 8 blocs. . . 92
6.2 Réduction des effets de blocs par minimisation de l'énergie de Hautes Fréquences. 92
6.3 Profile d'une portion de la ligne 436 de l'image "Lena" codée à 0.25bpp avec $JPEG$ standard, $JPEG$ + un filtre moyenneur et $JPEG$ + post-traitement de l'algo. **B1**. . . . 99
6.4 Erreur Quadratique Moyenne de Quantification des coefficients de la $TCDB$ (8×8) en ordre Zig-Zag, calculée pour L'image "Lena" à 0.15, 0.25 et 0.50bpp. . . . 100
6.5 Variation de n selon la fréquence dans le bloc X_k de (8×8). . . . 100
6.6 "Lena" Codée par JPEG avec Quantificateurs optimisés à 0.25bpp ($PSNR = 29.92dB$) . . . 101
6.7 "Lena" Codée par JPEG avec Quantificateurs optimisés + filtre moyenneur à 0.25bpp . . . 102
6.8 "Lena" Codée par JPEG avec Quantificateurs optimisés à 0.25bpp + **B1** (meilleur $PSNR$, $ne = 5$, $ns = 16$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$, $PSNR = 30.27dB$) . . . 103
6.9 "Lena" Codée par JPEG avec Quantificateurs optimisés à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 2$, $ns = 16$, $n_{dc} = 2$, $n_{ac} = 4$, $PSNR = 29.80dB$) 104
6.10 "Lena" Codée par l'Algo. **A1** à 0.25bpp ($PSNR = 30.19dB$) . . . 105
6.11 "Lena" Codée par l'Algo. **A1** + filtre moyenneur à 0.25bpp . . . 106
6.12 "Lena" Codée par l'Algo. **A1** à 0.25bpp + **B1** (meilleur $PSNR$, $ne = 5$, $ns = 16$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$, $PSNR = 30.38dB$) . . . 107
6.13 "Lena" Codée par l'Algo. **A1** à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 2$, $ns = 16$, $n_{dc} = 2$, $n_{ac} = 2$, $PSNR = 29.76dB$) . . . 108

6.14 "Lena" Codée par JPEG ($TCDB$ 16×16) avec Quantificateurs optimisés à 0.25bpp ($PSNR = 30.54dB$) 109
6.15 "Lena" Codée par JPEG ($TCDB$ 16×16) avec Quantificateurs optimisés à 0.25bpp + **B1** (meilleur $PSNR$, $ne = 6$, $ns = 32$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$, $PSNR = 30.62dB$) 110
6.16 "Lena" Codée par JPEG ($TCDB$ 16×16) avec Quantificateurs optimisés à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 4$, $ns = 32$, $n_{dc} = 2$, $n_{ac} = 4$, $n_{ac1} = 8$, $PSNR = 30.45dB$) 111
6.17 "Lena" Codée par l'Algo. **A1** (16×16) à 0.25bpp ($PSNR = 31.31dB$) 112
6.18 "Lena" Codée par l'Algo. **A1** (16×16) à 0.25bpp + **B1** (meilleur $PSNR$, $ne = 6$, $ns = 32$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$, $PSNR = 31.34dB$) 113
6.19 "Lena" Codée par l'Algo. **A1** (16×16) à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 4$, $ns = 32$, $n_{dc} = 2$, $n_{ac} = 4$, $n_{ac1} = 8$, $PSNR = 31.06dB$) 115
6.20 "Lena" Codée par SPIHT à 0.25bpp, $PSNR = 32.17dB$ 116
6.21 "Barbara" Codée par JPEG avec Quantificateurs optimisés à 0.25bpp ($PSNR = 26.40dB$) 117
6.22 "Barbara" Codée par JPEG avec Quantificateurs optimisés + filtre moyenneur à 0.25bpp 118
6.23 "Barbara" Codée par JPEG avec Quantificateurs optimisés à 0.25bpp + **B1** (meilleur $PSNR$, $ne = 5$, $ns = 16$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$, $PSNR = 26.63dB$) 119
6.24 "Barbara" Codée par JPEG avec Quantificateurs optimisés à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 2$, $ns = 16$, $n_{dc} = 2$, $n_{ac} = 4$, $PSNR = 26.25dB$) 120
6.25 "Barbara" Codée par l'Algo. **A1** à 0.25bpp ($PSNR = 27.32dB$) 121
6.26 "Barbara" Codée par l'Algo. **A1** + filtre moyenneur à 0.25bpp 122
6.27 "Barbara" Codée par l'Algo. **A1** à 0.25bpp + **B1** (meilleur $PSNR$, $ne = 5$, $ns = 16$, $n_{dc} = 8$, $n_{ac} = 8$, $n_{ac1} = 16$, $PSNR = 27.47dB$) 123
6.28 "Barbara" Codée par l'Algo. **A1** à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 2$, $ns = 16$, $n_{dc} = 2$, $n_{ac} = 2$, $PSNR = 27.14dB$) 124
6.29 "Barbara" Codée par l'Algo. **A1** à 0.25bpp + **B1** (meilleure qualité subjective, $ne = 2$, $ns = 16$, $n_{dc} = 2$, $n_{ac} = 2$, $PSNR = 27.14dB$) 125
6.30 "Barbara" Codée par SPIHT à 0.25bpp, $PSNR = 28.19dB$ 126

Liste des Tables

2.1 Probabilités associées aux événements $\hat{y}_i$ 31
2.2 Mots de code relatifs aux événements $\hat{y}_i$ 32
2.3 Les différents modes de fonctionnement de la norme $JPEG$ 36

3.1 Comparaison des performances du banc de filtres Wiener ($MMSE$) et de la TCD^{-1} pour les images "Lena" et Barbara. 49
3.2 Comparaison des performances du banc de filtres Wiener ($MMSE$) et de la TCD^{-1} pour les images Goldhill et Boats. 50
3.3 Comparaison des performances du banc de filtres Wiener ($MMSE$) et de la TCD^{-1} pour les images Desk et Zelda. 51

5.1 Probabilités associées aux entiers de l'intervalle [1,10] 71
5.2 Mots de code relatifs aux entiers de l'intervalle [1,10] 71
5.3 Comparaison du codage par l'algorithme **A1** de l'image "Lena" avec le $JPEG$ de base et une optimisation de ses quantificateurs 76
5.4 Comparaison du codage par l'algorithme **A1** de l'image "Barbara" avec le $JPEG$ de base et une optimisation de ses quantificateurs 76
5.5 Comparaison du codage par l'algorithme **A2** de l'image "Lena" avec le $JPEG$ de base sa quantification variable 76
5.6 Comparaison du codage par l'algorithme **A2** de l'image Barbara avec le $JPEG$ de base et sa quantification variable 81

6.1 Comparaison du codage par l'algorithme **A1** de l'image "Lena" avec le $JPEG$ standard, le $JPEG$ et une optimisation de ses quantificateurs, avant et après l'application de l'algorithme **B1** (critère du meilleur $PSNR$) 99
6.2 Comparaison du codage par l'algorithme **A1** de l'image "Barbara" avec le $JPEG$ standard, le $JPEG$ et une optimisation de ses quantificateurs, avant et après l'application de l'algorithme **B1** (critère du meilleur $PSNR$) 114

Glossaire

bpp	: bits par pixel
$d(x, \hat{x})$	: Distorsion entre x et x̂.
DMS	: Discrete Memoryless Source.
$E[x]$	: Espérance de x
EOB	: End Of Block.
EQM	: Erreur Quadratique Moyenne.
$EQMM$	: Erreur Quadratique Moyenne Minimale.
$H261$	: Recommandation de l'UIT d'un Codec video pour services audiovisuels à $p \times 64Kbit/s$.
$JPEG$	: Joint Photographic Expert Group.
KLT	: Transformée de Karhunen_Loève.
L	: Fonction Coût de Lagrange.
$MMSE$	: Minimum Mean Square Error.
$MPEG$	: Moving Pictures Expert Group.
MSE	: Mean Square Error.
pdf	: Fonction de distribution des probabilités.
$PSNR$	: Peak Signal/Noise Ratio (Rapport Signal à Bruit crête).
(R, D)	: Couple Débit/Distorsion.
RIF	: Réponse Impulsionnelle Finie.
RLE	: Run-Length Encoding (Codage par Plage).
SNR	: Signal/Noise Ratio (Rapport Signal à Bruit).
TCD	: Transformée en Cosinus Discrète d'un vecteur.
$TCDB$	: Transformée en Cosinus Discrète d'un Bloc ou Bidimensionnelle.
$TCDBIou(^{-1})$	: Transformée en Cosinus Discrète d'un Bloc ou Bidimensionnelle Inverse.
TFD	: Transformée de Fourrier Discrète.

Bibliographie

[1] *ITU-T Rec. T.84 ISO/IEC 10918-3*. "Compression and Coding of Continuous-tone Still Images - Part 3: Extensions".

[2] *MPEG*, 1993. "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5Mbits/s. Part2, Video.".

[3] *Rec. UIT-T H.261*, Mars 1993. "Utilisation des Lignes pour la Transmission des Signaux Autres que Téléphoniques, Codec Vidéo pour Services Audiovisuels à $p \times 64Kbit/s$".

[4] *Rec. UIT-T H.263*, Mars 1996. "Transmission de Signaux non Téléphoniques, Codage Vidéo pour Communications à Faible Débit".

[5] R. M. Mersereau A. K. Katsaggelos, J. Biemond and R. W. Schafer. "Nonstationary Iterative Image Restoration". in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages pp. 696–699, March 1985. Tampa, Florida.

[6] R. M. Mersereau A. K. Katsaggelos, J. Biemond and R. W. Schafer. "A Regularized Iterative Image Restoration Algorithm". *IEEE Trans. Signal Processing*, vol. 39(4):pp. 914–929, Dec. 1991.

[7] M. Crouse. "Optimizing Image Compression Standards", 1995.

[8] A. Dembo and S. Malah. "Statistical Design of Analysis/Synthesis Systems with Quantization". *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36(3):pp. 328–341, Mar. 1988.

[9] H. Everett. "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources". *Operation Res.*, vol. 11:pp. 399–417, 1963.

[10] P. Farrelle and A. Jain. "Recursive Block-Coding – A New Approach to Transform Coding". *IEEE Trans. Commun.*, vol. 34:pp. 161–179, Feb. 1986.

[11] Robert G. Gallager. *"Information Theory and Reliable Communication"*. John Wiley and Sons, 1968.

[12] A. Gersho and R.M. Gray. *"Vector Quantization and Signal Compression"*. Kluwer Academic Publishers, Boston, MA, 1992.

[13] A. Gersho and Y. Shoham. "Efficient Bit Allocation for an Arbitrary Set of Quantizers". *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36(9):pp. 1445–1453, Sept. 1988.

[14] K. Gosse and P. Duhamel. "Perfect Reconstruction versus MMSE Filter Banks in Source Coding". *IEEE Trans. on Signal Processing*, vol. 45(9):pp. 2188–2202, Sept. 1997.

[15] R.M. Gray. "Vector Quantization". *IEEE ASSP Mag.*, pages pp. 4–29, April 1984.

[16] R.M. Gray. *"Source Coding Theory"*. Kluwer Academic Publishers, Boston, MA, 1990.

[17] J. P. Guillois. *"Techniques de Compression des Images"*. Hermès, Paris, 1996.

[18] P. G. Howard and J. S. Vitter. "Analysis of Arithmetic Coding for Data Compression". *Information Processing and Managment*, vol. 28(7):pp. 749–763, 1992.

[19] D. A. Huffman. "A Method fo the Construction of Minimum Redundancy Codes". *Proc. of the Inst. of Radio Engineers*, vol. 40:pp. 1098–1101, 1952.

[20] A. Hung and T. Meng. "Optimal Quantizer Step Sizes for Transform Coders". in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. X:pp. 2621–2624, April 1991. Toronto, Canada.

[21] B.R. Hunt. "The Application of Constrained Least Squares Estimation to Image Restoration by Digital Computer". *IEEE Trans. Comput.*, vol. 22:pp. 805–812, Sept. 1973.

[22] A. Jain. *"Fundamentals of Digital Image Processing"*. Prentice-Hall Inc., Englewoods Cliffs, NJ, 1989.

[23] N.J. Jayant and P. Noll. *"Digital Coding of Waveforms"*. Prentice-Hall Inc., Englewoods Cliffs, NJ, 1984.

[24] B. Jeon and J. Jeong. "Blocking Artifacts Reduction in Image Compression with Block Boundary Discontinuity Criterion". *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8:pp. 345–357, Jun. 1998.

[25] F. Moreau de SAINT-MARTIN K. Gosse and P. Duhamel. "Filter Bank Design for Minimum Distortion in Presence of Subband Quantization". in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, May 1996. Atlanta,USA.

[26] A. K. Katsaggelos. "A General Formulation of Adaptive Image Restoration Algorithms". in *Proc. Conf. Inform. Sciences Syst.*, pages pp. 42–47, March 1986. Princton,NJ.

[27] J. Kovačevič. "Subband Coding Systems Incorporating Quantizer Models". *IEEE Trans. Image Processing*, vol. 4(5):pp. 543–553, May 1995.

[28] H. Maitre and F. Schmitt. "Propriétés Statistiques des Images, Echantillonnage, Fractales, Perception des Images". Technical report, Polycopié Télécom Paris, 1987.

[29] H. S. Malvar and D. H. Staelin. "The LOT: Transform Coding without Blocking Effects". *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37:pp. 553–559, Apr. 1989.

[30] J. Max. "Quantizing for Minimum Distortion". *IRE Trans. Inform. Theory*, vol. IT-6:pp. 7–12, March 1991.

[31] N. Moreau. *"Techniques de Compression des Signaux"*. Masson et CNET-ENST, Paris, FRANCE, 1994.

[32] M. T. Orchard and O. G. Guerlyuz. "Optimized Nonorthogonal Transforms for Image Compression". *IEEE Trans. Image Processing*, vol. 6(4):pp. 507–522, April 1997.

[33] D. Pearson and M. Whybray. "Transform Coding of Images using Interleaved Block". in *Proc. Inst. Elect. Eng.*, vol. 131:pp. 466–472, Aug. 1984.

[34] W. Pennebaker and J. Mitchell. *"JPEG Still Image Data Compression Standard"*. Van Nostrand Reinhold, New York, NY, 1993.

[35] B. Ramamurthi and A. Gersho. "Nonlinear Space-Variant Postprocessing of Block Coded Images". *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34:pp. 1258–1257, Oct. 1986.

[36] K. R. Rao and P. Yip. *"Discrete Cosine Transform: Algorithms, Advantages, Applications"*. Academic Press, 1990.

[37] H. Reeves and J. S. Lim. "Reduction of Blocking Effects in Image Coding". *Opt. Eng.*, vol. 23:pp. 34–37, Jan. 1984.

[38] R.Haddad and K. Park. "Modeling, Analysis and Optimum Design of Quantized M-Band Filter Banks". *IEEE Trans. on Signal Processing*, vol. 43(11):pp. 2540–2549, Nov. 1995.

[39] A. Said and W. A. Pearlman. "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees". *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6:pp. 243–250, Jun. 1996.

[40] C. E. Shannon. "A Mathematical Theory of Communication". *Bell Syst. Tech. Journal*, vol. 27:pp. 398–403, July 1948.

[41] R. L. Stevenson. "Reduction of Coding Artifacts in Transform Image Coding". in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages pp. 401–404, Mar. 1993. Minneapolis, MN.

[42] P. Duhamel T. Karp, K. Gosse and A. Mertins. "Design of Modulated Filter Banks for Minimum Output Distorsion in presence of Subband Quantization". in *Proc. IEEE Asilmor Conference on Signals, Systems and Computers*, Nov. 1996. Pacific Grove, CA, USA.

[43] J.G. Cleary T.C. Bell and J.H. Witten. *"Text Compression"*. Prentice-Hall Inc., Englewoods Cliffs, NJ, 1990.

[44] N. Uzun and R. A. Haddad. "Modeling, Analysis and Compensation of Quantization Effects in M-Band Subband Codecs". in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. III:pp. 173–176, April 1993. Minneapolis,USA.

[45] M. Vetterli and K. Ramchandran. "Best Wavelet Packet Bases in a Rate-Distortion Sense". *IEEE Trans. Image Processing*, vol. 2(2):pp. 160–175, April 1993.

[46] S. Wu and A. Gersho. "Rate-constrained Picture-Adaptive Quantization for JPEG Baseline Coders". in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. V:pp. 389–392, April 1993. Minneapolis,USA.

[47] N. P. Galatsanos Y. Yang and A. K. Katsaggelos. "Regularized Reconstruction to Reduce Blocking Artifacts of Block Discrete Cosine Transform Compressed Images". *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3(6):pp. 421–432, Dec. 1993.

[48] N. P. Galatsanos Y. Yang and A. K. Katsaggelos. "Projection-based Spatially Adaptive Reconstruction of Block-Transform Compressed Images". *IEEE Trans. Image Processing*, vol. 4(7):pp. 896–908, Jul. 1995.

[49] R. Pickholtz. Y. Zhang and M. Loew. "A new Approach to Reduce the Blocking Effect of Transform Coding". *IEEE Trans. Commun.*, vol. 41:pp. 299–302, Feb. 1993.

[50] M. T. Orchard Z. Xiong and Y. Zhang. "A Simple Deblocking Algorithm for JPEG Compressed Images Using Overcomplete Wavelet Representations". *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7(2):pp. 433–437, Apr. 1997.

[51] A. Zakhor. "Iterative Procedures for Reduction of Blocking Effects in Transform Image Coding". *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2(1):pp. 91–95, Mar. 1992.