

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS-SETIF

UFAS (ALGERIE)

MEMOIRE

Présenté à la Faculté des technologies

Département d'Electronique

Pour l'Obtention du Diplôme de

MAGISTER

Option : Communication

Par

M^{elle}: Nabila BRAHIMI

Thème

**Développement et implémentation des algorithmes de compression
d'images basés sur des transformées entières**

Sotenu le : 13/01/2011

devant la commission d'examen :

Président :	N. BOUZIT	Prof.	Université de Sétif
Rapporteur :	S. BOUGUEZEL	M. C	Université de Sétif
Examineurs :	A. KHELLAF	Prof.	Université de Sétif
	T. MOHAMADI	Prof.	Université de Sétif
	N. AMARDJIA	M. C	Université de Sétif

Remerciement

Au moment d'amener les voiles, le marin sait qu'il ne doit son arrivée au port qu'aux vents qui ont su l'y mener.

Tout d'abord je remercie le Bon DIEU, de m'avoir donné le courage pour terminer ce travail.

Les deux premières personnes que je souhaite remercier sont mes parents bien-aimés, qui n'ont jamais baissé les bras devant les difficultés.

Je tiens à exprimer ma gratitude à Mr. Saad Bouguezal, Maître de conférence à l'université Farhat Abbas à Sétif, pour avoir suivi de près mon travail, merci pour m'avoir indiqué souvent le bon chemin, pour ton soutien moral et scientifique.

J'adresse mes remerciements aux personnes du jury qui m'ont fait l'honneur d'accepter d'évaluer cette thèse:

Je tiens à exprimer toute ma gratitude envers Mr. N. BOUZIT, Professeur à l'Université de Ferhat Abbas à Sétif, pour avoir accepté de présider le jury de ma soutenance de magistère.

Je remercie aussi les autres membres du jury qui ont accepté de participer à cette soutenance : Mr. A. KHELLAF, Professeur à l'université de Ferhat Abbas, Mr. T. MOHAMADI, Professeur à l'université de Ferhat Abbas, Mr. N. AMARDJIA, Maître de conférence à l'université de Ferhat Abbas, pour leur rôle d'examineurs et pour leur relecture consciencieuse de ma thèse.

Merci à celles et ceux qui sont venus assister (ou qui auraient souhaité être là si il n'y avait pas eu toute cette distance) à mes pré-soutenances, soutenance et soirée de thèse!

Enfin, j'adresse mes plus sincères remerciements à tous mes proches et à tous mes amis, qui m'ont toujours soutenus et encouragés au cours de la réalisation de ce mémoire.

Merci à tous.

Dédicace

A toi **Azzeddine**, mon père,

Pour les sacrifices consentis pour ma scolarisation
et ton voeu ardent de me voir réussir dans la vie.

Avec toute ma tendresse.

A toi **Akila**, ma mère,

Pour ton amour, ton soutien, tes prières,
tes sacrifices endurés pour parvenir à l'aboutissement
de mes études.

Avec toute ma tendresse.

A vous **Mohammed (et sa femme Fahima), Hichem, Zahra (et son
mari Nacer), Wahiba, Saida, Faiza et Kanza**, mes chers
frères et sœurs,

Pour votre soutien spirituel.

Sincère gratitude.

A vous **Aya** et **Selma**, mes chères nièces,

Meilleurs voeux de succès dans votre vie.

A mes camarades d'auditoires et tous ceux de la Faculté des Sciences de
l'Université de Ferhat Abbas-Sétif.

Affectueuse reconnaissance.

Je dédie ce travail.

Nabila BRAHIMI

Table des matières

Liste des abréviations	I
Liste des figures	III
Liste des tableaux	V
Résumé	VII

Introduction générale

Chapitre I Compression d'images avec la DCT

I.1. Introduction	4
I.2. Généralités sur les images	4
I.3. Classification des méthodes de compression	6
I.3.1. Méthodes réversibles.....	6
I.3.2. Méthodes irréversibles	6
I.4. Transformée en Cosinus Discrète	7
I.4.1. DCT unidimensionnelle.....	8
I.4.2. DCT bidimensionnelle.....	9
I.5. Propriétés de la DCT	11
I.5.1. Décorrélation	11
I.5.2. Compactage d'énergie.....	12
I.5.3. Séparabilité	14
I.5.4. Symétrie	14
I.5.5. Orthogonalité.....	15
I.6. Algorithmes rapides de la DCT	16
I.7. Compression JPEG	18
I.7.1. Algorithme de compression JPEG.....	18
I.7.1.1. <i>Changement d'espace colorimétrique</i>	19
I.7.1.2. <i>Sous-échantillonnage</i>	20

I.7.1.3. <i>Découpage en blocs</i>	20
I.7.1.4. <i>Application de la DCT</i>	21
I.7.1.5. <i>Quantification</i>	21
I.7.1.6. <i>Codage entropique</i>	23
I.7.2. Algorithme de décompression JPEG.....	26
I.7.3. Exemple.....	26
I.8. Mesures de performances et de distorsion	28
I.8.1. Taux ou Facteur de compression.....	28
I.8.2. Système visuel humain : VHS (Visual Human System).....	29
I.8.3. Erreur quadratique moyenne	29
I.8.4. Rapport signal sur bruit en pic.....	30
I.9. Conclusion	30

Chapitre II Transformées en cosinus discrète entières

II.1. Introduction	31
II.2. Critères de développement des versions entières de la DCT	31
II.3. Méthode de développement des transformées DCT entières	32
II.3.1. Méthode basée sur la transformée de Hadamard.....	32
II.3.2. Méthode basée sur le principe de la symétrie dyadique.....	37
II.3.3. Méthode basée sur le principe lifting	40
II.3.4. Méthode basée sur les algorithmes à complexité variable.....	43
II.3.5. Méthode basée sur l'application de la fonction Signe	45
II.3.6. Méthode basée sur l'insertion appropriée des éléments nuls dans la SDCT.....	46
II.4. Conclusion	48

Chapitre III Transformées DCT entières proposées

III.1. Introduction	49
III.2. Transformées entières proposées	49
III.3. Algorithmes rapides pour les transformées proposées	51
III.3.1. Algorithme rapide pour la transformée T_1	52
III.3.2. Algorithme rapide pour la transformée T_2	54

III.4. Structure des matrices proposées.....	57
III.4.1. Structure de $C_{propos\ée\ 1}$	57
III.4.2. Structure de $C_{propos\ée\ 2}$	57
III.5. Conclusion.....	58

Chapitre IV Modélisation des transformées DCT entières

IV.1. Introduction.....	59
IV.2. Distribution énergétique des coefficients de la DCT conventionnelle et de ses versions entières.....	59
IV.3. Matrices de quantification proposées.....	62
IV.4. Modélisations statistiques proposées.....	62
IV.4.1. Histogramme des coefficients de la DCT entière.....	63
IV.4.2. Modèle empirique.....	64
IV.4.3. Modèles testés.....	64
IV.4.4. Estimation des paramètres des distributions.....	65
IV.4.5. Tests d'adéquation.....	66
IV.4.5.1. <i>Test de Kolmogorov-Smirnov</i>	67
IV.4.5.2. <i>Test de chi2</i>	67
IV.5. Résultats expérimentaux.....	68
IV.5.1. Modélisation des coefficients de la transformée entière de [11].....	69
IV.5.2. Modélisation des coefficients de la transformée de [12].....	75
IV.5.3. Modélisation des coefficients de la transformée entière proposée T_1	77
IV.5.4. Modélisation des coefficients de la transformée entière proposée T_2	79
IV.6. Résultats globaux.....	81
IV.7. Conclusion	83

Chapitre V Résultats expérimentaux et discussion

V.1. Introduction.....	84
V.2. Images de test utilisées.....	84
V.3. Résultats expérimentaux.....	85

V.3.1. Résultats obtenus pour la transformée proposée T_1	86
V.3.2. Résultats obtenus pour la transformée proposée T_2	87
V.3.3. Implémentation des transformées proposées dans la norme JPEG.....	92
V.3.4. Implémentation des matrices de quantification proposées dans JPEG.....	92
V.4. Conclusion	98

Conclusion générale et perspectives

Annexe	101
Bibliographie	103

Liste des abréviations

AC : Alternative Component
BinDCT : Binary DCT
CDF : Cumulativ Distribution Function
CR : Compression Ratio
DC : Discret Component
DCT : Discret Cosine Transform
DCT-1D : DCT unidimensionnelle
DCT -2D : DCT Bidimensionnelle
DFT : Discret Fourier Transform
DPCM : Differential Pulse Code Modulation
DSP : Digital Signal Processor
DST : Discret Sine Transform
DWT : Discret W Transform
EM : Estimation, Maximisation
EOB : End Of Bloc
ICT : Integer Cosine Transform
IDCT : Inverse Discret Cosine Transform
IEC : International Electronic Commission
ISO : International Organization for Standardization
ITU : International Telecommunication Union
JPEG : Joint Photography Experts Group
KStest : Kolmogorov Smirnov Test
MPEG : Moving Picture Experts Group
MSE : Mean Square Error
PEEN : Percentage Energy Error Norm
PSNR : Pic Signal to Noise Ratio
RGB : Red Green Blue
RLE : Run Length Encoding
RMSE : Root Mean Square Error
SDCT : Signed Discret Cosine Transform
VHS : Visuel Human System

VLC : Variable Length Coding

WHT : Walsh-Hadamard Transform

ZRL : Zero Run Length

Liste des figures

Figure	Titre	Page
Figure (I.1) :	Structure d'une image numérique.....	5
Figure (I.2) :	Fonctions de base de la DCT-1D.....	9
Figure (I.3) :	Distribution des fréquences de la DCT.....	10
Figure (I.4) :	Fonctions de base de la DCT-2D.....	11
Figure (I.5) :	Image aux niveaux de gris, 'Lena.bmp'.....	12
Figure (I.6) :	Les valeurs de niveau gris des paires de pixels adjacents	12
Figure (I.7) :	Compactage d'énergie de la DCT.....	13
Figure (I.8) :	Calcul de la DCT -2D en utilisant la propriété de séparabilité.....	14
Figure (I.9) :	Chaîne de codage de l'algorithme JPEG	18
Figure (I.10) :	Préparation des coefficients quantifiés pour le codage entropique	24
Figure (I.11) :	Chaîne de décodage de l'algorithme JPEG.....	26
Figure (I.12) :	Exemple de DCT et de quantification.....	27
Figure (II.1) :	Représentation de la rotation en plan directe par trois étapes lifting...	40
Figure (II.2) :	Représentation de la rotation en plan inverse par trois étapes lifting..	41
Figure (II.3) :	Structure de la Bin DCT version C.....	42
Figure (II.4) :	Algorithme de la DCT approximée.....	43
Figure (III.1) :	Structure de la transformée proposée $C_{proposée 1}$	57
Figure (III.2) :	Structure de la transformée proposée $C_{proposée 2}$	57
Figure (IV.1) :	Distribution énergétique des coefficients des différentes transformées entières pour l'image « Lena 256×256 ».....	61
Figure (IV.2) :	Matrices de quantification proposées.....	62
Figure (IV.3) :	Principe de la modélisation statistique proposée.....	63
Figure (IV.4) :	Images de test standards utilisées.....	69
Figure (IV.5) :	Histogrammes des coefficients DCT entière de [11] pour l'image « Bridge 512×512 ».....	69
Figure (IV.6) :	Approximation de la loi empirique du coefficient (2,1) de la transformée de [11] de l'image « Bridge 512×512 » par différentes lois théoriques.....	74

Figure (IV.7) :	Histogrammes des coefficients DCT entière de [12] pour l'image « Bridge 512×512 ».....	75
Figure (IV.8) :	Approximation de la loi empirique du coefficient (2,1) de la transformée de [12] de l'image « Lena 512×512 » par différentes lois théoriques.....	76
Figure (IV.9) :	Histogrammes des coefficients DCT entière proposée T_1 pour l'image «Bridge 512×512 ».....	77
Figure (IV.10):	Approximation de la loi empirique du coefficient (2,1) la transformée proposée T_1 de l'image « Lena 512×512 » par différentes lois théoriques.....	79
Figure (IV.11):	Histogrammes des coefficients DCT entière proposée T_2 pour l'image « Bridge 512×512 ».....	79
Figure (IV.12):	Approximation de la loi empirique du coefficient (2,1) de la transformée proposée T_2 de l'image « Lena 512×512 » par différentes lois théoriques.....	81
Figure (V.1) :	Images de test naturelles utilisées.....	84
Figure (V.2) :	Histogrammes correspondants aux images de test utilisées	85
Figure (V.3) :	Chaine de compression et de décompression réduite utilisée.....	85
Figure (V.4) :	Comparaison en terme de PSNR en utilisant: Scaled DCT [28], transformée de [11], transformée de [12], notre transformée proposée T_1 et SDCT [10]	89
Figure (V.5) :	Images reconstruites en utilisant: Scaled DCT [28], la transformée dans [11], transformée de [12], notre transformée proposée T_1 et SDCT [10]	89
Figure (V.6) :	Comparaison en terme PSNR en utilisant: Scaled DCT [28], transformée de [12], notre transformée proposée T_2 et SDCT [10].....	90
Figure (V.7) :	Images reconstruites en utilisant Scaled DCT [28], matrice proposée T_2 , matrice DCT de [50] et SDCT [10]	92
Figure (V.8) :	Image « Lena » restituée en utilisant la transformée de [11].....	96
Figure (V.9) :	Image « Lena » restituée en utilisant la transformée de [12].....	96
Figure (V.10) :	Image « Lena » restituée en utilisant la transformée proposée T_1	97
Figure (V.11) :	Image « Lena » restituée en utilisant la transformée proposée T_2	97

Liste des tableaux

Tableau	Titre	Page
Tableau (I.1) :	Complexité arithmétique des algorithmes de DCT.....	16
Tableau (I.2) :	Codage de différence des coefficients DC.....	25
Tableau (I.3) :	Codage des coefficients AC.....	26
Tableau (III.1) :	Comparison entre les transformées proposées et les autres transformées en terme de complexité.....	56
Tableau (IV.1) :	Tailles des images de test standards utilisées.....	68
Tableau (IV.2) :	Valeurs statistiques de test KS pour les 15 premiers coefficients en ordre de zigzag pour la transformée de [11] pour les lois de Cauchy, Laplace et Gauss.....	70
Tableau (IV.3) :	Valeurs statistiques de test χ^2 pour les 15 premiers coefficients en ordre de zigzag pour la transformées dans [11] de Cauchy, Laplace et Gauss.....	71
Tableau (IV.4) :	Résultat total de KS test pour la transformée dans [11] de Cauchy, Laplace et Gauss.....	71
Tableau (IV.5) :	Résultat total de χ^2 test pour la transformée dans [11] de Cauchy, Laplace et Gauss.....	72
Tableau (IV.6) :	Valeurs statistiques de test KS pour les 15 premiers coefficients en ordre de zigzag pour la transformée de [11].....	72
Tableau (IV.7) :	Valeurs statistiques de test χ^2 pour les 15 premiers coefficients en ordre de zigzag pour la transformée de [11].....	73
Tableau (IV.8) :	Valeurs statistiques de test KS pour les 15 premiers coefficients en ordre de zigzag pour la transformée de [12].....	75
Tableau (IV.9) :	Valeurs statistiques de test χ^2 pour les 15 premiers coefficients en ordre de zigzag pour la transformée de [12].....	76
Tableau (IV.10):	Valeurs statistiques de test KS pour les 15 premiers coefficients en ordre de zigzag pour la transformée proposée \mathbf{T}_1	77

Tableau (IV.11):	Valeurs statistiques de test χ^2 pour les 15 premiers coefficients en ordre de zigzag pour la transformée proposée \mathbf{T}_1	78
Tableau (IV.12):	Valeurs statistiques de test KS pour les 15 premiers coefficients en ordre de zigzag pour la transformée proposée \mathbf{T}_2	80
Tableau (IV.13):	Valeurs statistiques de test χ^2 pour les 15 premiers coefficients en ordre de zigzag pour la transformée proposée \mathbf{T}_2	80
Tableau (IV.14):	Résultats consolidé pour les transformées étudiées.....	82
Tableau (IV.15):	Résultats totaux de la modélisation des transformées entières.....	82
Tableau (V.1) :	Comparaison objective en terme de PSNR (en dB) entre la transformée entières proposée \mathbf{T}_1 et les autres transformée entières existantes.....	88
Tableau (V.2) :	Comparaison en terme de PEEN entre la transformée entières proposée \mathbf{T}_1 et les autres transformée entières existantes.....	88
Tableau (V.3) :	Comparaison objective en terme de PSNR (en dB) entre la transformée entière proposée \mathbf{T}_2 et les autres transformées entières existantes.....	91
Tableau (V.4) :	Comparaison objective en terme de PEEN entre la transformée entière proposée \mathbf{T}_2 et les autres transformées entières existantes..	91
Tableau (V.5) :	Comparaison entres les différentes transformées en terme de PSNR et le CR.....	93
Tableau (V.6) :	Résultats comparatives entre les différentes matrices de quantification proposées et la matrice conventionnelle de JPEG..	95

Résumé

Ce travail considère la compression des images fixes aux niveaux de gris par les DCT entières. Nous effectuons en premier lieu une étude bibliographique sur les techniques de compression d'images ainsi que sur cette catégorie de transformées et leurs méthodes de développement. Par une exploitation efficace de ces méthodes, nous développons deux nouvelles transformées entières et afin de les rendre attractives pour les applications en temps réel, des algorithmes rapides pour leurs calculs et des structures simples pour leurs implémentations sont aussi développés. Pour améliorer d'avantage les performances de ces transformées entières en termes de qualité de compression, nous proposons des nouvelles matrices de quantification appropriées pour les transformées proposées ainsi que pour quelques transformées existantes. Ces matrices de quantification présentent une amélioration supérieure à 0.5 dB en termes de PSNR par rapport aux résultats obtenus par la matrice de quantification adoptée par JPEG. Ce travail introduit aussi une modélisation statistique des coefficients des transformées entières proposées et existantes. Cette modélisation peut être exploitée pour développer des matrices de quantifications optimales pour chaque transformée entière. Les résultats expérimentaux obtenus en implémentant les transformées entières proposées et existantes dans le standard de compression JPEG montrent clairement l'efficacité et les performances des transformées et des matrices de quantification proposées.

Mots clés : compression d'images fixes, JPEG, DCT entières, matrice de quantification, modélisation statistique.

INTRODUCTION GENERALE

Introduction générale

Récemment, une croissance rapide en terme de besoin de stockage et/ou de transmission des informations visuelles est devenue très remarquable. Les diverses applications telles que la télécopie, la vidéoconférence, l'imagerie médicale et satellitaire, la télévision à haute définition, la télésurveillance et les services d'informations sur l'internet et le mobile 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 nécessitent 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 résoudre 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 deviennent plus efficaces et plus rapides.

Les techniques de compression se divisent en deux catégories principales [1], *compression sans pertes* et *compression avec pertes*. 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 constituant 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 constituant 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 suffisamment compacte pour plusieurs applications. Ainsi, pour obtenir une compression plus élevée, quelques informations seront négligées par la compression avec perte. Les améliorations apportées par la compression avec perte ne sont pas tout simplement dues 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 compressée est tout à fait différente de l'originale.

La compression avec perte est généralement réalisée dans le domaine fréquentiel par des méthodes basées sur les transformées discrètes. Deux standards sont adoptés par beaucoup de systèmes, la norme JPEG (*Joint Photographic Experts Group*) [2]-[4] qui est basée sur la DCT [5], [6] (*Discrete Cosine Transform*) et la norme JPEG2000 [7] qui est basée sur les ondelettes [6]. Dans ce travail, nous sommes intéressés par les techniques de compression des images fixes par transformées, en particulier celles qui sont basées sur la transformée DCT. Le processus de compression par cette transformée possède trois étapes principales, à savoir: (i) la transformation des pixels, l'image en entrée est transformée par la DCT afin de réduire les redondances entre les composantes voisines de l'image, (ii) la quantification et (iii) le codage entropique. Cette transformée qui est une transformation à valeurs réelles présente un inconvénient d'employer des opérations à virgule flottante qui sont très lentes et exigent plus de mémoire et consomment plus d'énergie dans l'implémentation hardware et software. Ces problèmes ont orienté la recherche vers une nouvelle direction qui est le développement de *versions entières de la DCT* [8]-[15]. Ces DCT entières n'exigent que des additions et/ou des décalages de bits. Par conséquent, elles présentent des complexités très réduites par rapport à celle de la DCT qui exige des additions et des multiplications en virgule flottante. En effet, les DCT entières remplacent la DCT conventionnelle dans beaucoup d'applications récentes qui nécessitent des vitesses de calcul très élevées.

Cette importance des DCT entières nous a motivé à entreprendre les recherches dans le domaine de compression d'images par ces transformées. L'objectif principale de notre travail est de chercher la possibilité d'améliorer les performances de la compression par les transformées entières. La méthodologie générale que nous avons considéré pour réaliser cet objectif comprend deux stratégies, (i) la recherche de nouvelles transformées entières qui présentent des bons compromis entre la complexité des calculs et les performances de compression, nous avons déjà publié dans [16] quelques résultats préliminaires obtenus par cette stratégie, et (ii) la modélisation statistique des coefficients des transformées entières pour construire des matrices de quantification optimales et propres à ces transformées. Noter bien que cette deuxième stratégie que nous avons introduit dans ce travail n'a jamais été considérée dans la littérature pour le cas des transformées entières.

Ce mémoire est composé de cinq chapitres. Le chapitre I présente une introduction générale sur les méthodes de compression d'image sans et avec pertes, en particulier celle qui est basée sur la DCT est donnée en détail. Des outils de mesure de performances et de

distorsions qui permettent la comparaison entre différentes méthodes de compression d'images sont aussi présentés. Une étude bibliographique sur les transformées DCT entières existantes [11], [12] et leurs méthodes de développement est considérée dans le chapitre II. L'exploitation de ces méthodes nous a conduit à développer des nouvelles transformées entières dans le chapitre III, des algorithmes rapides pour leurs calcul et des structures simples pour leurs implémentation sont aussi présentés. Le chapitre IV est consacré à la modélisation statistique des coefficients des transformées entières proposées et existantes. Cette modélisation permet de construire des nouvelles matrices de quantification propres à ces transformées. Afin de montrer l'efficacité et les performances de notre transformées entières et notre matrices de quantification, des résultats expérimentaux obtenues en implémentant les transformées entières proposées et existantes dans le standard de compression JPEG sont présentés dans le chapitre V pour quelques images de test.

CHAPITRE I

Compression d'images avec la DCT

I.1. Introduction

Dans de nombreux domaines, l'image numérisée remplace les images analogiques classiques telles que les photographies, radiographies ...etc. Le très grand nombre de données nécessaires pour la décrire rend la durée de transmission souvent trop longue et le stockage trop coûteux. Des problèmes se sont alors posés pour le stockage et surtout la transmission de ces images dans plusieurs domaines; dans l'astronomie pour la transmission des photographies prises par les sondes, dans les appareils photographiques numériques qui disposent d'une quantité limitée de mémoire, et bien sûr dans l'Internet et d'autres systèmes de communications mobiles. Pour remédier à ces problèmes, il est donc nécessaire de réduire la taille de l'image en extrayant l'information visuelle utile qui sera la seule codée. Cette réduction de taille des données sans ou avec une perte non significative d'informations est effectuée par une *technique de compression*.

En effet, il existe dans la littérature un nombre très important de techniques de compression. Dans ce chapitre, nous donnons une introduction générale sur les méthodes de compression d'images sans et avec pertes, en focalisant ensuite sur celle qui est basée sur la DCT (*Discrete Cosine Transform*) où nous présentons son principe, ces propriétés, ces algorithmes rapides et une description détaillée de l'algorithme JPEG (*Joint Photographic Experts Group*) qui l'exploite. Des outils de mesure de performances et de distorsions qui permettent la comparaison entre différentes méthodes de compression sont aussi présentés dans ce chapitre.

I.2. Généralités sur les images numériques

Une image numérique (aux niveaux de gris) peut être représentée comme une matrice de N colonnes et M lignes où chaque élément correspond au niveau de gris du pixel (Picture Element) de l'image. Une image de résolution 512×512 est donc une matrice de dimension 512×512 où le premier pixel est le premier élément de la matrice. Chaque élément est compris entre 0 (correspondant au noir) et 255 (correspondant au blanc).

On parle des images binaires si un seul bit décrit chaque pixel, 0 représentant un pixel noir et 1 un pixel blanc. En attribuant plusieurs bits aux pixels, un nombre plus élevé de niveaux de gris peut être distingué. La plupart du temps, les images monochromes sont codées

sur huit bits par pixel, puisque cela devient exactement un octet et les 256 ($2^8 = 256$) différentes intensités ainsi représentables sont largement suffisantes pour la perception humaine.

Pour les images couleurs, chaque pixel est caractérisé par 3 intensités lumineuses, celles des canaux rouge, vert et bleu, définissant des images $3 \times 8 = 24$ bits [17].

La figure ci-dessous donne la structure d'une image numérique, c'est l'image de « Goldhill » de 256×256 pixels.

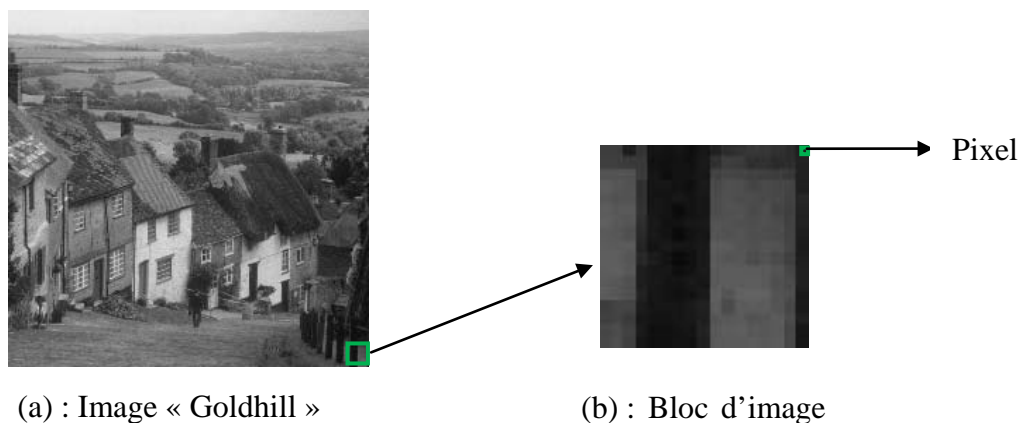


Figure (I.1) : Structure d'une image numérique

Le stockage en mémoire d'une image couleur de taille $M \times N$ exige $M \times N \times 24$ bits. Ceci représente une grande quantité: une image de 4 megapixels nécessite ainsi $4 \times 2^{20} \times 24$ bits soit 12 mégaoctets, ne permettant le stockage que d'une dizaine de photos sur une carte de 128 mégaoctets.

L'intérêt de la compression est donc évident. L'image comprimée occupe moins de place sur une unité de stockage. Elle prend moins de temps de transmission sous forme comprimée sur le même canal, ou bien, elle a besoin d'une bande passante plus petite pour arriver à destination en même temps que la même image non comprimée [17].

La compression d'image peut s'employer avec des contraintes et des attentes très différentes, selon l'utilisation à laquelle les images sont destinées. On peut vouloir réduire le nombre de bits d'une image avec une contrainte sur la capacité de stockage, etc.... La compression d'images est utilisée pour des applications de loisir, du traitement d'images, de la recherche visuelle rapide dans une base d'images, du diagnostic médicale, etc....

I.3. Classification des méthodes de compression

Selon les besoins, les techniques de compression d'images se différencient par le fait qu'elles permettent de comprimer avec ou sans perte d'informations. Alors, on distingue deux grandes catégories.

I.3.1. Méthodes réversibles

En général, les images (comme souvent d'autres données) présentent des redondances, ce qui est exploité par les méthodes réversibles. Dans ces méthodes, chaque bit d'information est conservé durant le processus de décompression, l'image reconstruite après la compression est une copie exacte de l'originale. En général, les méthodes réversibles (sans distorsion ou sans pertes d'informations) garantissent la restitution parfaite des images. Parmi les approches sans perte on trouve:

Les approches VLC (*Variable Length Coding*) exploitent la redondance statique des données, ils attribuent des mots de code courts aux symboles probables, et des mots plus longs aux symboles moins fréquents. L'algorithme de Huffman est une méthode simple qui se base sur ce principe. Il encode les différents symboles par des mots binaires de longueurs variables, la longueur dépend de la probabilité d'apparition du symbole. Bien qu'il soit optimal parmi les approches VLC sa performance est inférieure à celle du codage arithmétique [18] et ceci est dû du fait que les codes de Huffman doivent avoir une longueur entière de bits, ce qui ne permet pas toujours de réaliser une compression optimale, dont le codage arithmétique se singularise par sa capacité à coder chaque symbole sur un nombre non-entier de bits.

D'autres approches exploitent un autre type de redondance, la redondance de codage, notamment les répétitions; au lieu de coder un symbole répété plusieurs fois, il le code juste une fois et il indique le nombre de répétitions. L'algorithme RLE [18] (*Run Length Encoding*) est un bon exemple de ces approches.

I.3.2. Méthodes irréversibles

Les méthodes irréversibles ont également l'objectif de réduire ou supprimer les redondances présentes dans l'image, et en même temps elles exploitent le fait que l'œil n'est

pas sensible à certaines dégradations. Par exemple, le nombre de niveaux de gris que l'œil peut distinguer est largement inférieur à 256, il n'est même pas de 64. En tenant compte de ces arguments, les méthodes irréversibles (avec distorsion ou avec pertes d'information) modifient plus ou moins la valeur des pixels.

Parmi les deux possibilités de compression, nous avons choisi d'examiner et d'appliquer les approches avec pertes et plus précisément celles qui sont basées sur la transformée DCT. En effet, la performance en terme de taux de compression des algorithmes réversibles est très limitée.

Les méthodes par transformées figurent parmi les techniques de compression les plus employées. Elles permettent d'obtenir des taux de compression élevés tout en conservant une bonne qualité d'image [19]. L'intérêt que présentent de nombreuses transformées pour comprimer l'information est de projeter le signal sur une base de fonctions orthogonales, c'est-à-dire de distribuer l'énergie de ce signal sur des composantes décorrélatées entre elles. Il existe de nombreuses transformées orthogonales dont les propriétés sont généralement différentes.

Malgré qu'il existe un nombre considérable de transformées discrètes dans la littérature, on trouve que les transformées les plus utilisées dans le domaine de compression d'images sont la DCT et les Ondelettes. Les Ondelettes sont considérées dans le standard de compression JPEG 2000 [7], et la DCT dans les standards JPEG [2, 3] et MPEG [20] (*Moving Picture Experts Group*) pour la compression des images fixes et animées, respectivement. La popularité remarquable de la DCT est due essentiellement à sa capacité de compactage d'énergie (presque optimale) et à l'existence des algorithmes pour son calcul rapide et efficace. Du fait de toutes ces propriétés, nous avons retenu cette transformée, que nous décrivons dans la section suivante.

I.4. Transformée en Cosinus Discrète

Ahmed, Natarajan et Rao [5] ont été les premiers à mettre en équation la DCT en 1974. C'est une transformation mathématique qui transforme un ensemble de données d'un domaine spatial en un spectre de fréquence et inversement par la IDCT (*Inverse Discrete Cosine Transform*).

I.4.1. DCT unidimensionnelle

La DCT unidimensionnelle (DCT-1D) $F(k)$ d'une séquence $f(i)$ de N échantillons est donnée par l'équation (I.1):

$$F(k) = \alpha(k) \sum_{i=0}^{N-1} f(i) \cos \left[\frac{\pi(2i+1)k}{2N} \right] \quad (\text{I.1})$$

avec $k = 0, 1, 2, \dots, N-1$, et $\alpha(k)$ est défini comme:

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}} & \text{pour } k = 0 \\ \sqrt{\frac{2}{N}} & \text{pour } k \neq 0 \end{cases} \quad (\text{I.2})$$

La transformée inverse de la DCT-1D a pour équation:

$$f(i) = \sum_{k=0}^{N-1} \alpha(k) F(k) \cos \left[\frac{\pi(2i+1)k}{2N} \right] \quad (\text{I.3})$$

Il est clair d'après (I.1) et (I.2) que pour $k=0$,

$$F(k=0) = \sqrt{\frac{1}{N}} \sum_{i=0}^{N-1} f(i) \quad (\text{I.4})$$

Ainsi, le premier coefficient de la transformée est la valeur moyenne de la séquence d'échantillons (pixels). Dans la littérature, cette valeur est désignée sous le nom du coefficient DC (pour *Direct Component*) ou coefficient continu. Tous les autres coefficients de la transformée sont nommés les coefficients AC (*Alternative Component*) qui représentent les amplitudes des fréquences spatiales.

Le dessin de $\sum_{i=0}^{N-1} \cos \left[\frac{\pi(2i+1)k}{2N} \right]$ pour $N=8$ et en variant la valeur de i est montré sur

la figure (I.2). Selon l'observation précédente, la première forme d'onde au dessus-gauche ($k=0$) rend une valeur constante (DC), tandis que, toutes les autres formes d'onde ($k = 1, 2, \dots, 7$) ont des fréquences progressivement croissantes [21].

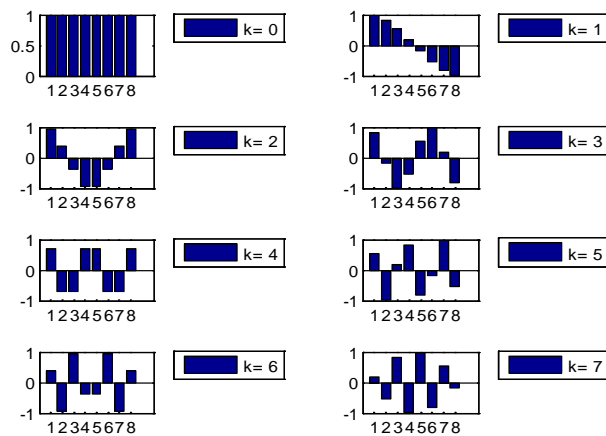


Figure (I.2) : Fonctions de base de la DCT-1D

I.4.2. DCT bidimensionnelle

La DCT-1D est utilisée en traitement des signaux unidimensionnels tels que les signaux de la parole. Pour l'analyse d'un signal bidimensionnel (2D) comme les images, on a besoin d'une version 2D de la DCT.

La DCT-2D est effectuée sur une matrice carrée $N \times N$ de pixels et donne une matrice carrée $N \times N$ de coefficients fréquentiels. Comme pour la DCT-1D, l'élément (0,0) est appelé la composante DC et les autres éléments sont les composantes AC.

Par convention, les 64 valeurs transformées (de chaque bloc de 8×8) sont positionnées d'une certaine manière, ainsi la valeur moyenne de tous ces coefficients est placée en haut à gauche de ce bloc. Plus on s'éloigne des coefficients continus plus leurs grandeurs tendent à diminuer, ce qui signifie que la DCT concentre l'énergie d'image en haut à gauche de la matrice transformée, les coefficients en bas et à droite de cette matrice contiennent moins d'information utile.

On peut représenter la distribution des fréquences de la DCT d'une matrice de 8×8 éléments par la figure (I.3).

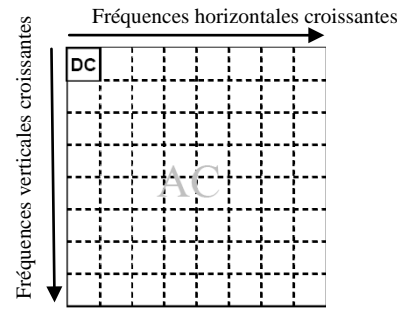


Figure (I.3): Distribution des fréquences de la DCT

Les deux équations qui suivent donnent, respectivement, la DCT-2D directe et inverse [21]:

$$F(k, m) = \alpha(k) \alpha(m) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) \cos \left[\frac{\pi(2i+1)k}{2N} \right] \cos \left[\frac{\pi(2j+1)m}{2N} \right] \quad (\text{I.5})$$

où $k, m = 0, 1, 2, 3, \dots, N-1$ sont des indices des fréquences spatiales dans la direction horizontale et verticale de l'image, et $\alpha(k)$ et $\alpha(m)$ sont donnés par (I.2).

- $f(i, j)$ représente la valeur d'un pixel de l'image initiale pour i et j donnés.
- $F(k, m)$ représente les coefficients de la DCT.
- N représente la taille du bloc.

La transformée en inverse est donnée comme:

$$f(i, j) = \sum_{k=0}^{N-1} \sum_{m=0}^{N-1} \alpha(k) \alpha(m) F(k, m) \cos \left[\frac{\pi(2i+1)k}{2N} \right] \cos \left[\frac{\pi(2j+1)m}{2N} \right] \quad (\text{I.6})$$

où $i, j = 0, 1, 2, 3, \dots, N-1$.

Dans le cas d'un bloc de taille 4×4 et 8×8 les différentes fonctions de bases qui représentent l'espace transformé sont, respectivement, au nombre de 16 et 64. Ces fonctions sont représentées sur la figure (I.4). Les fonctions situées en haut à gauche représentent les basses fréquences de la transformée, les fréquences spatiales augmentent au fur et à mesure que l'on se déplace vers le coin inférieur droit du bloc.

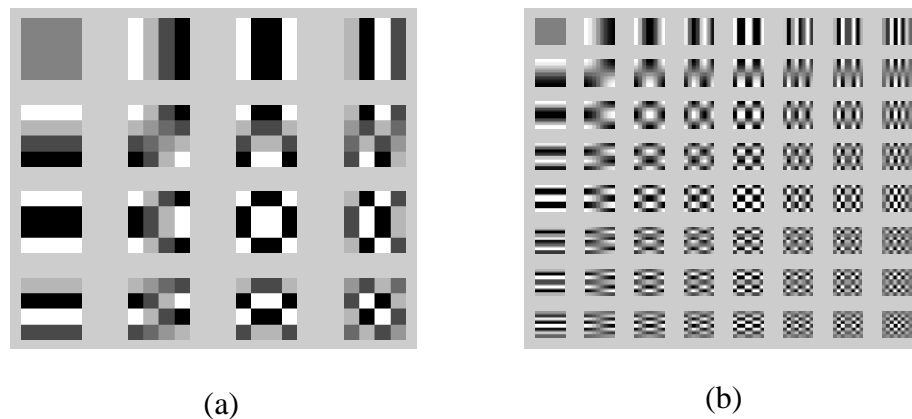


Figure (I.4) : Fonctions de base de la DCT-2D

(a) : Pour $N=4$

(b) : Pour $N=8$

I.5. Propriétés de la DCT

Dans la section précédente, nous avons développé la base mathématique de la DCT. Cette section expose (avec des exemples) quelques propriétés de la DCT qui sont particulièrement importantes aux applications de traitement d'image.

I.5.1. Décorrélation

Comme on a discuté précédemment, l'avantage principal de la transformation d'image est la réduction de redondance entre ces pixels voisins. Ceci mène à des coefficients transformés non-corrélés qui peuvent être codés indépendamment.

La figure (I.5) montre une image typique aux niveaux de gris. L'image est de taille 256×256 avec chaque pixel a un niveau de gris codé sur 8 bits pour une gamme de [0-255]. Même avec un grand degré de détail dans beaucoup de régions, la valeur du niveau de gris de n'importe quel pixel indiqué tend à être semblable à ses pixels voisins. Pour illustrer ce rapport, on a tracé les valeurs de niveau de gris des paires de pixels adjacents avant et après l'application de la DCT, comme montre la figure (I.6).



Figure (I.5) : Image aux niveaux de gris, 'Lena.bmp'

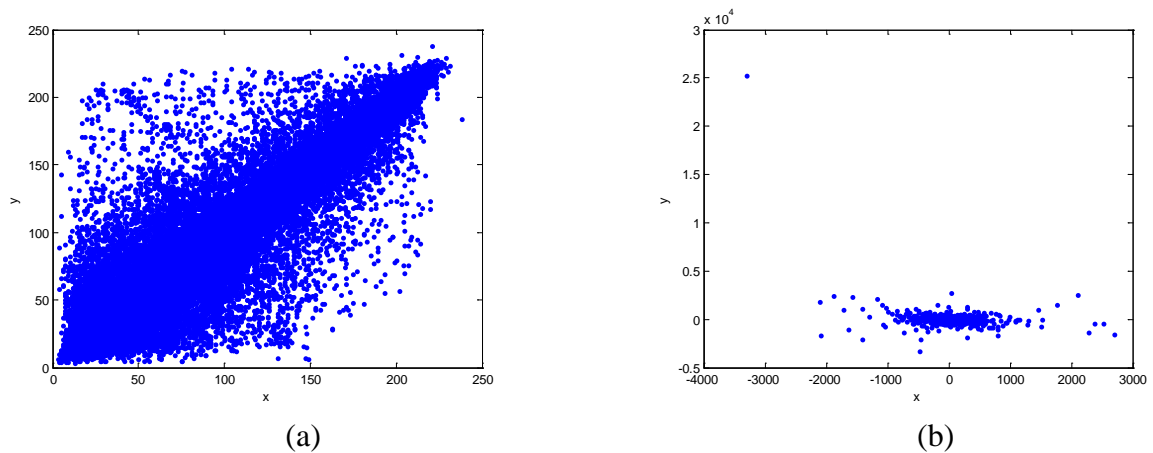


Figure (I.6) : Les valeurs de niveau de gris des paires de pixels adjacents

(a) : Avant la DCT

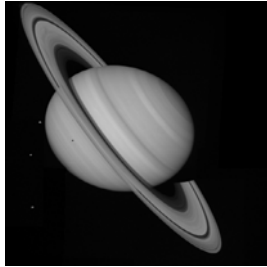
(b) : Après la DCT

Chaque point représente un pixel dans l'image, avec la coordonnée 'x' présente son niveau de gris et la coordonnée 'y' présente le niveau de gris de son voisin droit. La relation diagonale forte pour la ligne $x=y$ (figure (I.6 (a))) montre clairement la corrélation forte entre les pixels voisins et lorsqu'on applique la transformation DCT sur l'image entière, le résultat est montré sur la figure (I.6.(b)), les deux pixels sont décorrélés, i.e. savoir la valeur du premier pixel n'aide pas l'estimation de la valeur du second. Par conséquent, la DCT présente une excellente propriété de décorrélation.

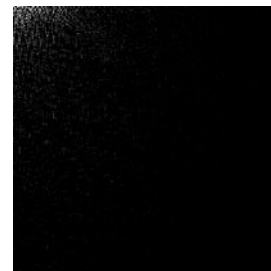
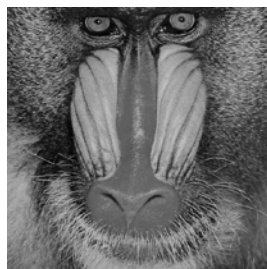
I.5.2. Compactage d'énergie

L'efficacité d'une transformation peut être directement mesurée par sa capacité de compactage d'énergie des données d'entrée dans des coefficients le faible possible. Ceci

permet au quantificateur de jeter les coefficients avec des amplitudes relativement petites sans présenter la déformation visuelle dans l'image reconstruite.



(a)



(b)



(c)

Figure (I.7) : Compactage d'énergie de la DCT, (a) Saturn et sa DCT, (b) Baboon et sa DCT, (c) Child et sa DCT

L'application de la DCT sur ces images fournit un compactage très bon d'énergie dans la région de basse fréquence de l'image transformée. Par conséquent, d'après les images précédentes, on peut remarquer que la DCT rend un excellent compactage d'énergie pour des images corrélées.

I.5.3. Séparabilité

L'équation (I.5) de la DCT peut être exprimée comme [21] :

$$F(k, m) = \alpha(k) \alpha(m) \sum_{i=0}^{N-1} \cos \left[\frac{\pi(2i+1)k}{2N} \right] \sum_{j=0}^{N-1} f(i, j) \cos \left[\frac{\pi(2j+1)m}{2N} \right] \quad (\text{I.9})$$

$$F(k, m) = \alpha(k) \sum_{i=0}^{N-1} G(i, m) \cos \left[\frac{\pi(2i+1)k}{2N} \right] \quad (\text{I.10})$$

$$G(i, m) = \alpha(m) \sum_{j=0}^{N-1} f(i, j) \cos \left[\frac{\pi(2j+1)m}{2N} \right] \quad (\text{I.11})$$

où $k, m = 0, 1, 2, \dots, N-1$, et $\alpha(k)$ et $\alpha(m)$ sont donnés par l'équation (I.2). Cette propriété, connue par la *séparabilité*, a l'avantage principal où $F(k, m)$ peut être calculé dans deux étapes par les opérations successives unidimensionnelles (1D) sur les lignes et les colonnes d'image. Cette idée est illustrée graphiquement sur la figure (I.8). Les arguments présentés peuvent être identiquement appliqués pour le calcul inverse de la DCT.

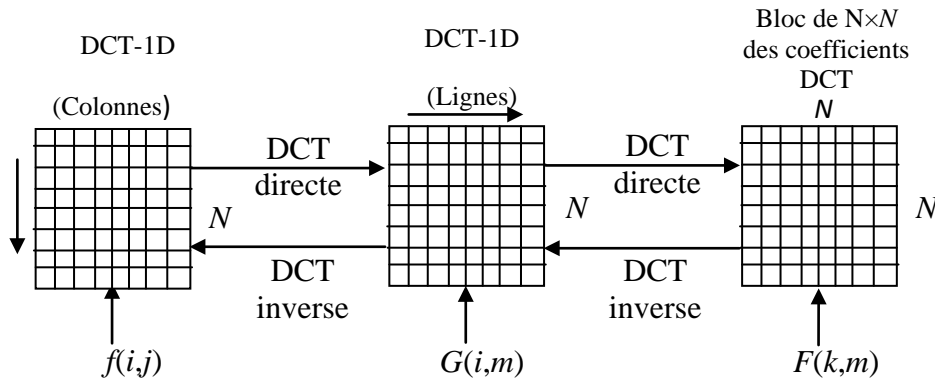


Figure (I.8) : Calcul de la DCT -2D en utilisant la propriété de séparabilité

I.5.4. Symétrie

Un autre regard sur l'équation (I.5) indique que la multiplication d'une matrice par la DCT nécessite les mêmes opérations pour les lignes que pour les colonnes, une telle

transformation s'appelle *une transformation symétrique*. Une transformation séparable et symétrique peut être exprimée sous la forme :

$$\mathbf{F} = \mathbf{T}_N \times \mathbf{f} \times \mathbf{T}_N^t \quad (\text{I.12})$$

où \mathbf{F} est la matrice transformée, \mathbf{f} est la matrice d'image originale et \mathbf{T}_N est la matrice de la transformée de taille $N \times N$ avec des éléments $T_N(i, j)$ donné par l'équation (I.13) [21].

$$T_N(i, j) = \begin{cases} \frac{1}{\sqrt{N}} & \text{si } i = 0 \\ \sqrt{\frac{2}{N}} \cos \left[\frac{(2j+1)i\pi}{2N} \right] & \text{si } i > 0 \end{cases} \quad (\text{I.13})$$

\mathbf{T}_N^t est la matrice transposée de \mathbf{T}_N .

C'est une propriété extrêmement utile puisqu'elle implique que la matrice transformée peut être pré-calculée et s'est appliquée alors à l'image, fournissant de ce fait une amélioration d'efficacité de calcul.

1.5.5. Orthogonalité

La matrice \mathbf{T}_N de la DCT est orthogonale par ce qu'elle vérifie [3] :

$$\mathbf{T}_N \times \mathbf{T}_N^t = \mathbf{I}_N \quad (\text{I.15})$$

où \mathbf{I}_N représente la matrice d'identité de taille $N \times N$. Cette propriété peut être vérifiée directement par :

$$\mathbf{v}_i \mathbf{v}_j^t = \sum_{k=0}^{N-1} \mathbf{v}_{ik} \mathbf{v}_{kj} = 0 \quad \forall i \neq j, \quad i, j = 0, 1, \dots, N-1 \quad (\text{I.16})$$

où \mathbf{v}_i et \mathbf{v}_j sont les vecteurs de base de \mathbf{T}_N . Ainsi, la matrice inverse de \mathbf{T}_N est égale à sa transposée, i.e. $\mathbf{T}_N^{-1} = \mathbf{T}_N^t$. Par conséquent, et en plus de ses caractéristiques de décorrélation, cette propriété fournit une certaine réduction dans la complexité de pré-calcul.

En exploitant l'orthogonalité de \mathbf{T}_N , l'inverse de la transformation donnée par l'équation (I.12) est donnée par:

$$\mathbf{f} = \mathbf{T}'_N \times \mathbf{F} \times \mathbf{T}_N \quad (\text{I.14})$$

I.6. Algorithmes rapides de la DCT [3, 18]

Le calcul de la DCT est la partie la plus longue pour coder une image en JPEG. Par conséquent, c'est la meilleure partie qui mérite d'être considérée par les investigateurs afin d'optimiser les calculs et augmenter la vitesse dans la compression JPEG.

Beaucoup de différents algorithmes rapides ont été proposés dans la littérature pour faire cette optimisation, par exemple [22]-[27]. La plupart de ces algorithmes exigent 12 multiplications et 29 additions pour calculer la DCT d'un vecteur de 8 points (voir le tableau (I.1)), sauf celui de Loeffler qui exige 11 multiplications.

Ref.auteurs	Chen [22]	Wang [23]	Lee [24]	Vetterli [25]	Suehiro [26]	Loeffler [27]
Multiplication	16(13)	13	12	12	12	11
Addition	26(29)	29	29	29	29	29

Tableau (I.1) : Complexité arithmétique des algorithmes de la DCT

- L'algorithme rapide de Chen [22] présente un algorithme récursif pour la factorisation de la matrice DCT de n'importe quelle taille N , avec $N = 2^m$ et $m \geq 2$, à des rotations en plan et des papillons (butterflies). La factorisation a une structure très régulière et elle est 6 fois plus rapide que l'algorithme rapide de la DCT basé sur la DFT (*Discrete Fourier Transform*). Pour $N=8$, le nombre de multiplications et d'additions peut être facilement changé en nombre montré dans la parenthèse dans le tableau (I.1) en employant la méthode de 3 multiplications et 3 additions par rotation qu'est utilisée dans toutes les autres publications.
- La méthode est généralisée en suite par Wang [23] pour toutes les versions de la DCT, Transformée en Sinus Discrète (DST), la transformée W discrète (DWT) et la transformée de Fourier discrète (DFT).

- L'algorithme de Lee [24] a les premières phases très régulières, mais il a une structure irrégulière au dernier étage, il a besoin des valeurs inverse des coefficients de cosinus, ceci peut mener aux problèmes numériques de débordement.
- Vetterli [25] emploie une formule récursive pour son algorithme; cependant, les fonctionnements additionnels exigés pour connecter les cases périodiquement calculées mènent à une augmentation de complexité dans la structure de son algorithme.
- L'algorithme de Suehiro [26] exige moins de multiplications que celui de Wang [22], mais sa solution nécessite toujours l'application de la méthode de Wang pour obtenir des algorithmes pour DST, DWT et DFT de l'algorithme DCT.
- Une réduction dans le nombre de multiplications a été introduite par Loeffler et al. [27], l'algorithme proposé exige 11 multiplications au lieu de 12 multiplications nécessaires dans les autres algorithmes.

Le nombre minimal de multiplications exigé dans le calcul de la DCT-1D a été concrétisé par l'algorithme proposé dans [28]. La section suivante présente brièvement le principe général de cet algorithme.

➤ Algorithme le plus rapide de la DCT

E. Feig et all. [28] proposent un algorithme rapide pour calculer la DCT de taille $p \times q$, où p et q sont relativement des premiers, en appliquant le produit de Kronecker sur les matrices de tailles p et q . Si la SDCT (Scaled DCT) de taille $p \times q$ est construite à partir de SDCT de taille p et q , alors le nombre de multiplications, additions et décalages pour le calcul de DCT à $N = p \times q$ points, est [28]:

$$\begin{aligned}
 m_{pq} &\leq pm_q + qm_p + (pq - p - q + 1)/2 \\
 a_{pq} &= pa_q + qa_p + pq - p - q + 1 \\
 s_{pq} &= ps_q + qs_p
 \end{aligned}
 \tag{I.17}$$

où m_{pq} , a_{pq} et s_{pq} dénotent , respectivement, le nombre de multiplications, additions et de décalages. Cette matrice DCT de N points est factorisée d'après [28] comme:

$$\tilde{\mathbf{C}}_N = \mathbf{D}_N \mathbf{S}_{N,3} \mathbf{S}_{N,2} \mathbf{S}_{N,1}
 \tag{I.18}$$

où \mathbf{D}_N dénote une matrice diagonale d'ordre N et $\mathbf{S}_{N,3}, \mathbf{S}_{N,2}$ et $\mathbf{S}_{N,1}$ sont des matrices moins complexes. La factorisation de la matrice SDCT de 8×8 est donnée dans [28], elle exige 5 multiplications et 29 additions.

Les propriétés de la DCT ont été bien exploitées par la norme internationale de la compression d'images fixes JPEG. La section suivante est consacrée à une description détaillée de cette norme.

I.7. Compression JPEG

JPEG (l'acronyme de *Joint Photographic Experts Group*) est la première norme internationale de la compression d'images pour les images fixes en couleur et en niveaux de gris. Cette norme est un résultat des efforts de collaboration par l'Union de Télécommunication Internationale (ITU), Organisation International de Standardisation (ISO), et Commission Electrotechnique Internationale (IEC) [7].

I.7.1. Algorithme de compression JPEG

La chaîne de codage de standard JPEG est représentée sur la figure (I.9). La transformée choisie pour la norme est la transformée en cosinus discrète, elle n'est pas appliquée sur toute l'image mais sur des blocs de l'image de taille fixes (8×8). Suivie ensuite d'une étape de quantification des coefficients DCT et d'un codage de ceux-ci.

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

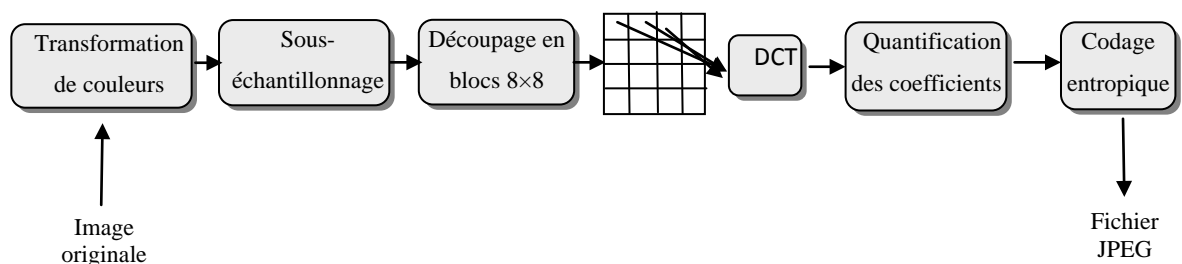


Figure (I.9) : Chaîne de codage de l'algorithme JPEG [20]

On peut diviser l'algorithme JPEG en 5 différentes étapes:

1. Transformer l'image en un espace de couleurs optimale.
2. Sous-échantillonnage de la chrominance en moyennant des groupes de pixels ensemble.

3. Appliquer la DCT à chaque bloc de pixels pour retirer la redondance des données de l'image.
4. Quantifier chaque bloc de coefficients DCT avec des fonctions optimisées pour l'œil humain.
5. Coder le résultat en utilisant l'algorithme de Huffman pour retirer les redondances résiduelles.

Le but des prochains paragraphes est de présenter de manière plus précise les différentes étapes de cet algorithme.

I.7.1.1. Changement d'espace colorimétrique

Dans une image non comprimée, chaque pixel est habituellement caractérisé par 3 composantes de base dans l'espace colorimétrique *RGB* (*Red Green Blue*). Dans cette étape, on va passer d'un espace colorimétrique du type *RGB* à un autre de type de luminance/chrominance *YC_bC_r*. (Cette étape est sautée pour des images aux niveaux de gris).

L'intérêt d'utiliser ce type de représentation est que la corrélation entre les composantes est réduite de telle façon que chaque composante peut être analysée séparément et sous-échantillonnée la chrominance sans perdre la qualité visuelle de l'image reconstruite.

La transformation du *RGB* à *YC_bC_r* est faite par l'expression mathématique suivante [7] :

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.29900 & 0.587000 & 0.114000 \\ -0.168736 & -0.331264 & 0.500002 \\ 0.500000 & -0.418688 & -0.081312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (\text{I.19})$$

Afin de représenter *C_b* et *C_r* dans des nombres entiers de 8 bits non signés, ils sont décalés en ajoutant 128 à chaque échantillon suivi d'une saturation de la valeur dans la gamme [0, 255]. Par conséquent, la transformation ci-dessus peut être exprimée comme [7].

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.29900 & 0.587000 & 0.114000 \\ -0.168736 & -0.331264 & 0.500002 \\ 0.500000 & -0.418688 & -0.081312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} \quad (\text{I.20})$$

En conséquence, la transformation inverse de *YC_bC_r* à *RGB* est faite par (I.21) [7].

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0 & 0.0 & 1.40210 \\ 1.0 & -0.34414 & -0.71414 \\ 1.0 & 1.77180 & 0.0 \end{pmatrix} \begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} - \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} \quad (\text{I.21})$$

I.7.1.2. Sous-échantillonnage

Après la conversion de l'espace de couleur, la majorité d'information spatiale de l'image est contenu dans la composante de luminance (Y). Les composants de chrominance (C_b and C_r) contiennent l'information de couleur de l'image et nous perdons peu d'informations par le sous-échantillonnage de ces composants. Son principe est donné ci-dessous [7]:

- On va découper l'image en blocs de pixels, le plus souvent des blocs de 2×2 . Pour ces 4 pixels du bloc on va calculer la moyenne de C_b et de C_r . Celle-ci est stockée une seule fois par bloc de 4 pixels. On peut se contenter de ce sous-échantillonnage car l'œil humain est moins sensible à la chrominance qu'à la luminance. Cette compression dégrade peu la qualité de l'image et permet de diviser la taille de l'image par 2.

I.7.1.3. Découpage en blocs

L'étape suivante consiste à diviser l'image en blocs de 8×8 sur lesquels vont s'appliquer indépendamment les étapes suivantes. Si la largeur et la longueur de l'image ne sont pas des multiples de 8, on rajoute des colonnes en répétant la dernière jusqu'à atteindre un multiple de 8 (même traitement pour les lignes). Les raisons de ce découpage sont:

- ❖ L'application de la DCT à tous $N \times N$ pixels d'une image produit une meilleure compression mais comporte également beaucoup d'opérations arithmétiques et donc lente; et l'application de la DCT aux unités de données réduit le taux de la compression mais plus rapide.
- ❖ L'expérience montre que, dans une image, les corrélations entre les pixels sont de gamme courte. Un pixel dans une telle image a une valeur (composant de couleur ou niveau de gris) qui proche de ses voisins, mais n'a rien pour faire avec les valeurs des voisins lointains.

I.7.1.4. Application de la DCT

Le concept général de la transformée est discuté dans la section (I.4). La norme JPEG applique la DCT pas à l'image entière mais aux unités de données (blocs) de 8×8 pixels.

$$F(k, m) = \frac{1}{4} \alpha(k) \alpha(m) \sum_{i=0}^7 \sum_{j=0}^7 f(i, j) \cos \left[\frac{\pi(2i+1)k}{16} \right] \cos \left[\frac{\pi(2j+1)m}{16} \right] \quad (\text{I.22})$$

$$\alpha(i) = \begin{cases} \frac{1}{\sqrt{2}} & \text{pour } k = 0 \\ 1 & \text{pour } k \neq 0 \end{cases} \quad \text{et } 0 \leq i = (k, m) \leq 7 \quad (\text{I.23})$$

La matrice de la DCT pour $N=8$ est donnée d'après l'équation (I.13) par :

$$\mathbf{T}_8 = \begin{bmatrix} \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} \\ \frac{1}{2} \cos \frac{1}{16} \pi & \frac{1}{2} \cos \frac{3}{16} \pi & \frac{1}{2} \cos \frac{5}{16} \pi & \frac{1}{2} \cos \frac{7}{16} \pi & \frac{1}{2} \cos \frac{9}{16} \pi & \frac{1}{2} \cos \frac{11}{16} \pi & \frac{1}{2} \cos \frac{13}{16} \pi & \frac{1}{2} \cos \frac{15}{16} \pi \\ \frac{1}{2} \cos \frac{2}{16} \pi & \frac{1}{2} \cos \frac{6}{16} \pi & \frac{1}{2} \cos \frac{10}{16} \pi & \frac{1}{2} \cos \frac{14}{16} \pi & \frac{1}{2} \cos \frac{18}{16} \pi & \frac{1}{2} \cos \frac{22}{16} \pi & \frac{1}{2} \cos \frac{26}{16} \pi & \frac{1}{2} \cos \frac{30}{16} \pi \\ \frac{1}{2} \cos \frac{3}{16} \pi & \frac{1}{2} \cos \frac{9}{16} \pi & \frac{1}{2} \cos \frac{15}{16} \pi & \frac{1}{2} \cos \frac{21}{16} \pi & \frac{1}{2} \cos \frac{27}{16} \pi & \frac{1}{2} \cos \frac{33}{16} \pi & \frac{1}{2} \cos \frac{39}{16} \pi & \frac{1}{2} \cos \frac{45}{16} \pi \\ \frac{1}{2} \cos \frac{4}{16} \pi & \frac{1}{2} \cos \frac{12}{16} \pi & \frac{1}{2} \cos \frac{20}{16} \pi & \frac{1}{2} \cos \frac{28}{16} \pi & \frac{1}{2} \cos \frac{36}{16} \pi & \frac{1}{2} \cos \frac{44}{16} \pi & \frac{1}{2} \cos \frac{52}{16} \pi & \frac{1}{2} \cos \frac{60}{16} \pi \\ \frac{1}{2} \cos \frac{5}{16} \pi & \frac{1}{2} \cos \frac{15}{16} \pi & \frac{1}{2} \cos \frac{25}{16} \pi & \frac{1}{2} \cos \frac{35}{16} \pi & \frac{1}{2} \cos \frac{45}{16} \pi & \frac{1}{2} \cos \frac{55}{16} \pi & \frac{1}{2} \cos \frac{65}{16} \pi & \frac{1}{2} \cos \frac{75}{16} \pi \\ \frac{1}{2} \cos \frac{6}{16} \pi & \frac{1}{2} \cos \frac{18}{16} \pi & \frac{1}{2} \cos \frac{30}{16} \pi & \frac{1}{2} \cos \frac{42}{16} \pi & \frac{1}{2} \cos \frac{54}{16} \pi & \frac{1}{2} \cos \frac{66}{16} \pi & \frac{1}{2} \cos \frac{78}{16} \pi & \frac{1}{2} \cos \frac{90}{16} \pi \\ \frac{1}{2} \cos \frac{7}{16} \pi & \frac{1}{2} \cos \frac{21}{16} \pi & \frac{1}{2} \cos \frac{35}{16} \pi & \frac{1}{2} \cos \frac{49}{16} \pi & \frac{1}{2} \cos \frac{63}{16} \pi & \frac{1}{2} \cos \frac{77}{16} \pi & \frac{1}{2} \cos \frac{91}{16} \pi & \frac{1}{2} \cos \frac{105}{16} \pi \end{bmatrix} \quad (\text{I.24})$$

I.7.1.5. Quantification

L'étape qui suit l'application de la DCT est la quantification des coefficients DCT, afin de se ramener à un nombre limité de valeurs différentes. Elle est la seule étape irréversible de l'algorithme de compression JPEG. Elle consiste à diviser chaque élément de la matrice obtenue après la DCT ($F(k, m)$) par son élément correspondant d'une matrice de quantification, \mathbf{Q} , caractérisant la sensibilité de l'œil à chaque zone de fréquence. Il sera

également arrondi à l'entier le plus grand inférieur au nombre que l'on a après la division par le coefficient de quantification.

La norme de JPEG tient compte jusqu'à quatre matrices de quantification, et l'utilisateur peut choisir n'importe quelle de ces quatre matrices pour quantifier chaque composant de couleur. Les 64 nombres qui constituent chaque matrice de quantification sont tous des paramètres de JPEG. La norme JPEG emploie normalement les deux approches suivantes :

1. *Matrices de quantification par défaut* : Deux matrices, pour la luminance (niveaux de gris) et les composantes de chrominance, sont des résultats de beaucoup d'expériences effectuées par les fondateurs de JPEG [18]. Elles sont incluses dans la norme de JPEG et elles sont reproduites ici dans les deux matrices dans l'Annexe.

2. *Une matrice de quantification simple* Q est calculée en se basant sur un paramètre R indiqué par l'utilisateur. Une expression simple comme (I.25) garantit que les Q Cs commencent petit au coin gauche supérieure et deviennent plus grand au coin inférieur droit.

$$Q(k, m) = 1 + (k + m) \times R \quad (\text{I.25})$$

- $Q(k, m)$: Valeur du pas de quantification dans un bloc, k et $m = 0, 1, 2, \dots, N - 1$.
- R : Facteur de qualité de la quantification.

Si les éléments des matrices de quantification de la luminance et de la chrominance sont représentés par $Q(k, m)$, un coefficient DCT quantifié à la position (k, m) est donné par :

$$F_q(k, m) = \left\lfloor \frac{F(k, m)}{Q(k, m)} \right\rfloor \quad (\text{I.26})$$

où $F_q(k, m)$ est la valeur du coefficient transformé après la quantification, $\lfloor \rfloor$ est implique une action d'arrondir la division au nombre entier le plus proche le plus petit.

Au décodeur, l'étape de la déquantification est réalisée comme suit:

$$\tilde{F}(k, m) = F_q(k, m) \times Q(k, m) \quad (\text{I.27})$$

où $\tilde{F}(k, m)$ est le coefficient déquantifié.

Si la quantification est faite correctement, très peu de nombres non zéro seront laissés dans la matrice de coefficients de DCT, et ils seront typiquement concentrés dans la région gauche supérieure. Ces nombres sont les sorties du JPEG, mais ils sont encore comprimés avant d'être écrits sur le fichier de sortie. Dans la littérature de JPEG cette compression s'appelle "codage entropique" qui est présentée en détail dans la section suivante.

I.7.1.6. Codage entropique

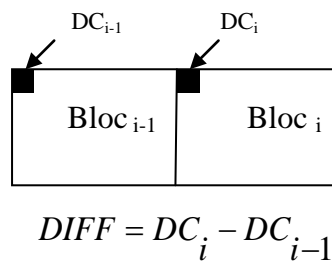
Enfin, la dernière étape est le codage des coefficients quantifiés, c'est une opération sans perte. Un codage entropique consiste à coder les symboles qui ont une probabilité d'apparition forte avec de mots de longueur plus faible que les symboles qui ont une probabilité d'apparition faible.

Le coefficient DC et les 63 coefficients AC sont codés séparément comme représenté sur la figure (I.10). Les coefficients DC sont codés en DPCM [18] (*Differential Pulse Code Modulation.*) en utilisant la valeur de coefficient DC de bloc précédent (équation (I.28)), comme représenté sur la figure (I.10(a)).

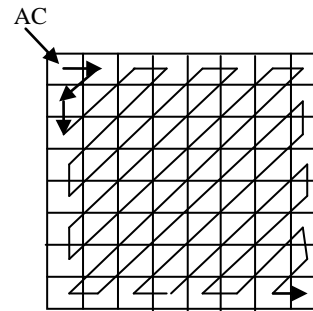
$$DIFF = DC_i - DC_{i-1} \quad (I.28)$$

Le but de ce traitement est d'exploiter la corrélation entre les valeurs DC des blocs adjacents et les coder plus efficacement, car ils contiennent typiquement la plus grande partie d'énergie d'image.

Les 63 coefficients AC commençant à partir du coefficient AC(0,1) sont codés par plage après un balayage de zigzag comme représenté sur la figure (I.10(b)). Le but de l'adoption d'un modèle de balayage de zigzag est de faciliter le codage par entropie en rencontrant d'abord les coefficients les plus susceptibles d'être non-nuls (les coefficients de basses fréquences). C'est dû au fait que, pour la plupart des scènes naturelles, l'énergie d'image se concentre principalement dans quelques coefficients de basses fréquences.



(a) : Codage différentiel des coefficients DC



(b) : Lecture en Zig-Zag

Figure (I.10) : Préparation des coefficients quantifiés pour le codage entropique [29]

- Le deuxième point est de préciser la compression des 63 coefficients AC du bloc. L'algorithme JPEG emploie une combinaison de RLE et Huffman ou codage arithmétique [19]. L'idée est que la séquence des coefficients AC contient juste quelques nombres différents de zéro, avec le nombre de zéros entre eux.

Pour chaque nombre non zéro x , l'encodeur :

- 1- Trouver le nombre Z des zéros consécutifs précédent x ;
- 2- Trouver x dans le tableau (I.2) et préparer son nombre de ligne et de colonne (G et H);
- 3- Le paire (G, Z) est utilisé comme des nombres de ligne et de colonne pour le tableau (I.3);
- 4- Le code Huffman trouvé en cette position dans le tableau (I.3) est correspond à H (où H est écrit comme un nombre de G bits) et le résultat est, finalement, le code émet par le codeur JPEG pour les coefficients AC du x et tous les zéros consécutifs qui lui précèdent.

0	0										0
1	-1	1									10
2	-3	-2	2	3							110
3	-7	-6	5-	-4	4	5	6	7			1110
4	-15	-14	...	-9	-8	8	9	10	...	15	11110
5	-31	-30	-29	...	-17	-16	16	17	...	31	111110
6	-63	-62	-61	...	-33	-32	32	33	...	63	1111110
7	127	126	125	...	-65	-64	64	65	...	127	11111110
⋮				⋮							
14	-16383	-16382	-16381	...	-8193	-8192	8192	8193	...	16383	111111111111110
15	-32767	-32766	-32765	...	-16385	-16384	16384	16385	...	32767	1111111111111110
16	32768										1111111111111111

Tableau (I.2) : Codage de différence des coefficients DC [18]

Le codage des différences de DC est fait en utilisant le tableau (I.2), d'abord voici quelques mots sur ce tableau.

- Chaque ligne a son nombre (du côté gauche), le code pour la ligne (du côté droite), et plusieurs colonnes entre eux et contient plus de nombres que sa précédente mais pas les nombres contenus dans les lignes précédentes.

- La ligne i contient une gamme des nombres entiers $[-(2^i-1), +(2^i-1)]$ à l'exception de la gamme moyenne $[-(2^{i-1}-1), +(2^{i-1}-1)]$.

Les codes de Huffman dans le tableau (I.3) ne sont pas ceux recommandés par le standard JPEG. La norme recommande l'utilisation des tableaux de Huffman (donnés dans l'Annexe) et indique que jusqu'à quatre tableaux d'indicatif de Huffman peuvent être employés par un codec de JPEG.

L'encodeur devrait noter l'indicatif d'EOB (*End of Bloc*) à la position (0, 0), et le code ZRL (*Zero Run Length*) à la position (0, 15). Le premier indique l'extrémité de bloc et le

dernier est l'indicatif émis pour 15 zéros consécutifs quand le nombre de zéros consécutifs dépasse 15. Ces indicatifs sont ceux recommandés pour les coefficients AC de luminance du tableau (I.2).

$G \backslash Z$	0	1	...	15
0	1010			1111111001(ZRL)
1	00	1100	...	111111111110101
2	01	11011	...	111111111110110
3	100	1111001	...	111111111110111
4	1011	111110110	...	111111111111000
5	11010	1111110110	...	111111111111001
⋮	⋮			

Tableau (I.3) : Codage des coefficients AC [18]

I.7.2. Algorithme de décompression JPEG

La décompression reprend d'une manière inverse toutes les étapes de la compression présentées précédemment. Il se fait grâce aux matrices incluses dans le fichier compressé. On notera que la fonction inverse de la DCT est l'IDCT (Inverse Discrete Cosine Transform). La chaîne de décompression est donnée par le schéma suivant:

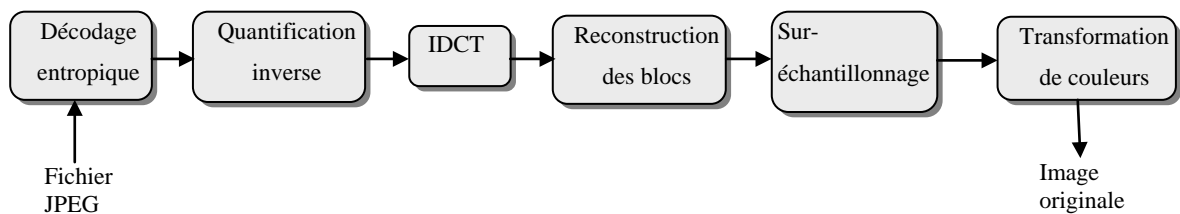


Figure (I.11) : Chaîne de décodage de l'algorithme JPEG [20]

I.7.3. Exemple

Dans cette section, nous donnons un exemple de codage d'un bloc de 8×8 sélectionné à partir de l'image 'Lena.bmp' de taille 256×256.

$$\begin{bmatrix} 138 & 143 & 145 & 144 & 144 & 142 & 143 & 141 \\ 141 & 144 & 143 & 144 & 145 & 142 & 143 & 137 \\ 141 & 144 & 140 & 144 & 142 & 140 & 141 & 137 \\ 145 & 147 & 143 & 139 & 142 & 141 & 140 & 138 \\ 143 & 143 & 145 & 143 & 139 & 138 & 138 & 131 \\ 143 & 145 & 143 & 144 & 137 & 135 & 135 & 135 \\ 148 & 145 & 144 & 139 & 134 & 134 & 132 & 134 \\ 146 & 144 & 142 & 137 & 134 & 133 & 129 & 134 \end{bmatrix}$$

(a) Bloc original

$$\begin{bmatrix} 235.6250 & -1.0333 & -12.0809 & -5.2029 & 2.1250 & -1.6724 & -2.7080 & 1.3238 \\ -22.5904 & -17.4842 & -6.2405 & -3.1574 & -2.8557 & -0.0695 & 0.4342 & -1.1856 \\ -10.9493 & -9.2624 & -1.5758 & 1.5301 & 0.2029 & -0.9419 & -0.5669 & -0.0629 \\ -7.0816 & -1.9072 & 0.2248 & 1.4539 & 0.8963 & -0.0799 & -0.0423 & 0.3315 \\ -0.6250 & -0.8381 & 1.4699 & 1.5563 & -0.1250 & -0.6610 & 0.6088 & 1.2752 \\ 1.7541 & -0.2029 & 1.6205 & -0.3424 & -0.7755 & 1.4759 & 1.0410 & -0.9930 \\ -1.2825 & -0.3600 & -0.3169 & -1.4601 & -0.4900 & 1.7348 & 1.0758 & -0.7613 \\ -2.5999 & 1.5519 & -3.7628 & -1.8448 & 1.8716 & 1.2139 & -0.5679 & -0.4456 \end{bmatrix}$$

(b) Les coefficients DCT

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

(c) Table de quantification

$$\begin{bmatrix} 15 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(d) Coefficients quantifiés et arrondis

$$\begin{bmatrix} 240 & 0 & -10 & 0 & 0 & 0 & 0 & 0 \\ -24 & -12 & 0 & 0 & 0 & 0 & 0 & 0 \\ -14 & -13 & 0 & 0 & 0 & 0 & 0 & 0 \\ -14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(e) Coefficients déquantifiés

$$\begin{bmatrix} 142 & 144 & 147 & 150 & 152 & 153 & 154 & 154 \\ 149 & 150 & 153 & 155 & 156 & 157 & 156 & 156 \\ 157 & 158 & 159 & 161 & 161 & 160 & 159 & 158 \\ 162 & 162 & 163 & 163 & 162 & 160 & 158 & 157 \\ 162 & 162 & 162 & 162 & 161 & 158 & 156 & 154 \\ 160 & 161 & 161 & 161 & 160 & 158 & 156 & 154 \\ 160 & 160 & 161 & 162 & 161 & 160 & 158 & 157 \\ 160 & 161 & 163 & 164 & 164 & 163 & 161 & 160 \end{bmatrix}$$

(f) Bloc restitué

Figure (I.12) : Exemple de DCT et de quantification

- La figure (I.12(a)) est un bloc de 8×8 à 8 bits, extrait à partir de l'image 'Lena 256×256 '. Les petites variations d'un pixel à l'autre indiquent la prédominance de basses fréquences spatiales. Après la soustraction de 128 de chaque pixel, le bloc est transformé par la DCT donnée par l'équation (I.5).
 - La matrice (I.12 (b)) montre les coefficients résultant de la DCT. Excepté quelques uns des coefficients de fréquence plus bas, les amplitudes sont tout à fait petites.
 - La matrice (I.12(c)) est un exemple de la matrice de quantification de la norme JPEG [29] pour les coefficients de la luminance (niveau de gris) incluse dans l'annexe.
 - La matrice (I.12 (d)) montre les coefficients DCT quantifiés et arrondis par la matrice de quantification, comme indiqué par l'équation (I.26).
- Enfin, la matrice dans la figure (I.11(f)) montre les valeurs du bloc restitué, on remarque qu'ils sont semblables aux originaux dans la figure (I.11(a)).

Les éléments de la matrice (I.12(d)) doivent être codés par le code de Huffman avant la transmission au décodeur. Le premier nombre du bloc à coder est le coefficient DC, qui doit être codé par le codage différentiel.

- Les mêmes tableaux (I.2 et I.3) utilisés par le codeur, bien sûr, doivent être utilisés par le décodeur.

I.8. Mesures de performances et de distorsion

Lorsqu'on utilise des méthodes non réversibles pour la compression d'images, il est utile de pouvoir mesurer la perte de qualité de l'image reconstruite par rapport à l'image initiale afin de pouvoir optimiser la compression.

La distorsion est l'erreur introduite par l'opération de compression, due au fait qu'éventuellement l'image reconstruite n'est pas exactement identique à l'image originale.

I.8.1. Taux ou Facteur de compression

Le taux de compression donne une mesure de performance des méthodes de compression des images fixes [30] est donnée par (I.31).

$$\text{CR} = \frac{\text{nombre de bits de l'image originale}}{\text{nombre de bits de l'image reconstruite}} \quad (\text{I.31})$$

CR : taux de compression (Compression Ratio).

Ce taux est une mesure de performance de l'algorithme de compression. Grâce à ce taux de compression on peut comparer les différents algorithmes.

Pour une même méthode de compression et un même CR réalisé sur des images distinctes, la qualité obtenue peut être très variable d'une image à l'autre. Les propriétés statistiques des images originales jouent un rôle prépondérant dans le résultat obtenu. Par exemple, dans le cas d'une image sur-échantillonnée qui est donc très redondante, il est facile d'obtenir des taux élevés. La théorie de l'information donne une limite théorique au CR maximal qu'il est possible d'obtenir sans distorsion pour toute méthode de compression sur une image donnée.

I.8.2. Système visuel humain: VHS (Visual Human System)

L'œil est un outil essentiel pour apprécier la qualité d'une image. Il va permettre à l'utilisateur d'identifier le contenu des images, la netteté de celles-ci et la qualité des contours.

I.8.3. Erreur quadratique moyenne

Pour les méthodes avec pertes, la mesure de distorsion utilisée généralement en compression d'image, pour pouvoir comparer les différentes méthodes de compression, est l'erreur quadratique moyenne MSE (*Mean Square Error*) [30] :

$$\text{MSE} = \frac{1}{M \cdot N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[f(i, j) - \hat{f}(i, j) \right]^2 \quad (\text{I.32})$$

$f(i, j)$: Image originale.

$\hat{f}(i, j)$: Image reconstruite.

$M \cdot N$: Taille de l'image.

Plus l'erreur quadratique moyenne est grande, plus il y a de différence de qualité entre l'image originale et l'image reconstruite. Pour la compression sans perte, comme il n'y a

aucune différence entre l'image reconstruite et l'image originale, l'erreur quadratique moyenne est nulle.

I.8.4. Rapport signal à bruit en pic

La mesure la plus couramment utilisée par la communauté internationale est la mesure du rapport signal à bruit en pic PSNR (*Pic Signal to Noise Ratio*), il se mesure en décibel (dB) et il est défini par [30] :

$$\text{PSNR} = 10 \cdot \log_{10} \frac{(2^R - 1)^2}{\text{MSE}} \quad \text{dB} \quad (\text{I.33})$$

où R représente le nombre de bits désignés pour un pixel. Quelques auteurs définissent le PSNR en fonction de RMSE (*Root Mean Square Error*) comme [18]:

$$\text{PSNR} = 20 \cdot \log_{10} \frac{(2^R - 1)}{\text{RMSE}} \quad \text{dB} \quad (\text{I.34})$$

La valeur typique de PSNR varie entre 20 et 40 [18]. Supposant que les valeurs des pixels varient de [0,255], un RMSE de 25.5 résulte un PSNR de 20, et un RMSE de 2.55 résulte un PSNR de 40. Un RMSE de zéro (c.-à-d., des images identiques) résulte un PSNR infini. Un RMSE de 255 résulte un PSNR de zéro.

I.9. Conclusion

On a vu dans ce chapitre l'intérêt et les différents types de la compression d'image. Ainsi, on a constaté que la DCT exploite les redondances interpixels pour produire une excellente décorrélation. En outre, la DCT compacte l'énergie dans la région des basses fréquences. Par conséquent, une partie du contenu à hautes fréquences peut être négligée sans dégradation significative de qualité. Ces propriétés qui sont explorées dans les normes de codage d'image visuelle comme le JPEG, MPEG, MPEG4 ..., et l'existence des algorithmes rapides rendent l'investigation sur l'amélioration de la DCT un axe très important de plusieurs recherches actuelles. Le chapitre suivant fait l'objet d'une étude des versions entières de cette transformée.



**CHAPITRE
II**

Transformées en cosinus discrète entières

II.1. Introduction

Les signaux numériques image, vidéo et audio sont représentés par des valeurs entières. La DCT à virgule flottante convertit ces valeurs entières à des coefficients réels. Malgré que les algorithmes rapides de la DCT comme [22]-[28], [31], [32], réduisent le nombre d'opérations arithmétiques de manière significative, ils exigent toujours des opérations en virgule flottante; ce qui rend les implémentations hardware et software de ces algorithmes très lentes, nécessitent beaucoup d'espace mémoire et consomment trop d'énergie électrique.

Pour remédier à ces problèmes, les coefficients DCT sont approximés par des nombres entiers, alors les multiplications en virgule flottante sont remplacées par les multiplications en nombres entiers [8], [10]-[13].

Les méthodes d'approximation en nombres entiers permettent de construire et de produire avec souplesse des versions entières de la DCT qui utilisent seulement les additions et les décalages des bits (bit-shift operations). Ces versions entières résultantes sont comparables à la transformée originale en termes de propriétés mathématiques de base comme la linéarité, l'orthogonalité, la symétrie des vecteurs de base et la récurrence et en termes de performances comme le gain et l'efficacité en compression. Ainsi, elles sont beaucoup plus rapides que la DCT conventionnelle et ont, en conséquence, des applications pratiques plus larges comme dans les communications sans fil et par satellites, aussi bien que dans l'internet et les systèmes multimédia.

Ce chapitre présente une étude bibliographique sur les transformées DCT entières, en commençant en premier temps par les critères nécessaires pour les construire et en étudiant ensuite quelques méthodes de développement des DCT entières existantes dans la littérature.

II.2. Critères de développement des versions entières de la DCT

La méthode d'approximation garantit que toutes les propriétés mathématiques (données dans la section I.5) et la structure récursive sont préservées dans la matrice de la transformée entière. Ce fait, permet à n'importe quel algorithme rapide existant d'être appliqué efficacement pour le calcul des DCT entières ICT_N (*Integer Cosine Transform* d'ordre N) [3], [18].

Si on considère \mathbf{T}_N la matrice DCT de taille N avec ces éléments $\{T(i, j)\}, i, j = 0, 1, 2, \dots, N-1$ et son approximation entière $\hat{\mathbf{T}}_N$ avec ces éléments $\hat{T}_N(i, j)$.

L'approximation entière doit satisfaire les propriétés suivantes:

- *La propriété d'orthogonalité:* Les lignes (colonnes) de la matrice transformée entière sont orthogonales entre eux. Cette propriété assure que les ICT_N directes et inverses ont la même structure.
- *La propriété de nombre entier:* Les éléments de la matrice approximée de la DCT sont des nombres entiers. En fait, les éléments sont rapprochés par les nombres raisonnables, c.-à-d. rapports des nombres entiers. La nature unitaire de ces matrices empêche les éléments pour être juste des nombres entiers. Elle élimine le besoin de beaucoup d'opérations arithmétiques en virgule flottante dans le calcul de la transformée, depuis chaque élément entier de la matrice de transformée approximée est représenté par un nombre finis de bits.
- La relation avec les valeurs réelles de la matrice DCT :

$$\begin{aligned}
 \text{(a)} \quad & \text{si } |T_N(i, j)| \geq |T_N(i, k)|, \text{ alors } |\hat{T}_N(i, j)| \geq |\hat{T}_N(i, k)|, \quad i, j, k = 0, 1, \dots, N-1 \\
 \text{(b)} \quad & \text{sign}(T_N(i, j)) = \text{sign}(\hat{T}_N(i, j)), \quad i, j = 0, 1, \dots, N-1
 \end{aligned} \tag{II.1}$$

Beaucoup de méthodes sont proposées dans la littérature pour approximer la DCT réelle d'ordre 8 et 16 à ses versions entières [8], [32], [33], [35], [36]. La section suivante fait l'objet d'une étude de quelques une de ces méthodes.

II.3. Méthode de développement des transformées DCT entières

II.3.1. Méthode basée sur la transformée de Hadamard

Au début, Hein et Ahmed [36] ont prouvé que la DCT peut être calculée via la transformée de Walsh-Hadamard [37] (WHT: *Walsh-Hadamard Transform*) dont les éléments sont des ± 1 seulement, à travers une matrice de conversion qui a une structure diagonale en blocs. Le principe de cette méthode est donné par la suite et ces détails peuvent être trouvés dans [36], [38].

Les transformées DCT et WHT dans la notation de matrice-vecteur (les facteurs de normalisation sont omis) sont données respectivement par [38]:

$$\mathbf{X}_T = \mathbf{T}_N \times \mathbf{x}^t, \quad \mathbf{X}_W = \mathbf{W}_N \times \mathbf{x}^t \tag{II.2}$$

où \mathbf{x} est le vecteur d'entrée, \mathbf{X}_T est le vecteur transformé en domaine DCT, \mathbf{X}_W est le vecteur transformé en domaine WHT et \mathbf{W}_N est la matrice de Hadamard.

L'examination des matrices \mathbf{T}_N et \mathbf{W}_N pour une taille N donnée montre que les deux transformées ont la même structure de symétrie/ antisymétrie [38].

Si on réarrange les lignes de \mathbf{T}_N et de \mathbf{W}_N dans l'ordre des bits inversés (bit-reversal order), alors l'équation (II.2) peut être réécrite d'après [38] comme:

$$\bar{\mathbf{X}}_T = \bar{\mathbf{T}}_N \times \mathbf{x}^t, \quad \bar{\mathbf{X}}_W = \bar{\mathbf{W}}_N \times \mathbf{x}^t \quad (\text{II.3})$$

Avec $\bar{\mathbf{T}}_N$ et $\bar{\mathbf{W}}_N$ sont respectivement la matrice DCT et la matrice Hadamard ré-ordonnées.

Depuis $\bar{\mathbf{W}}_N$ est une matrice orthogonale, en substituant la relation $\bar{\mathbf{W}}_N^t \times \bar{\mathbf{W}}_N = \mathbf{I}_N$ dans l'équation (II.3), on trouve:

$$\bar{\mathbf{X}}_T = \bar{\mathbf{T}}_N \times \bar{\mathbf{W}}_N^t \times \bar{\mathbf{W}}_N \times \mathbf{x}^t = \mathbf{C}_N \times \bar{\mathbf{X}}_W \quad \text{avec} \quad \mathbf{C}_N = \bar{\mathbf{T}}_N \times \bar{\mathbf{W}}_N^t \quad (\text{II.4})$$

où \mathbf{C}_N est la matrice de conversion qui convertit un vecteur du domaine de Hadamard au domaine de la DCT. La matrice de conversion a 2 propriétés essentielles:

- Elle est orthogonale, étant le produit de 2 matrices orthogonales $\bar{\mathbf{T}}_N$ et $\bar{\mathbf{W}}_N^t$.
- Elle a une structure diagonale en blocs tant que les lignes de \mathbf{T}_N et \mathbf{W}_N sont dans l'ordre de bits inversés.

Après, Jones et al. [39] ont prouvé qu'une approximation entière de la DCT peut être réalisée par WHT. Cette DCT approximée est appelée la matrice transformée C (CMT: *C Matrix Transform*), c'était la première tentative pour approximer les valeurs réelles d'une transformée trigonométrique à des nombres entiers.

D'après (II.4), la matrice CMT d'ordre N notée par $\bar{\mathbf{C}}_N^{CMT}$ est donnée par :

$$\bar{\mathbf{C}}_N^{CMT} = \mathbf{C}_N \times \bar{\mathbf{W}}_N, \quad \mathbf{C}_N = \bar{\mathbf{T}}_N \times \bar{\mathbf{W}}_N^t \quad (\text{II.5})$$

où toutes les matrices, avec \mathbf{C}_N , ont leurs lignes dans l'ordre de bits inversés. La matrice $\bar{\mathbf{W}}_N$ nécessite pas d'être approximée parce que tous ces éléments sont des ∓ 1 . A partir de (II.5), on peut voir que la matrice \mathbf{T}_N est approximée indirectement à travers la matrice \mathbf{C}_N

Puisque la matrice de conversion à valeurs réelles \mathbf{C}_8 est orthogonale, $\mathbf{C}_8 \times \mathbf{C}_8^t = \mathbf{I}_8$. La même relation doit se retenir pour la matrice de conversion à nombres entiers $\hat{\mathbf{C}}_8$, c.-à-d $\hat{\mathbf{C}}_8 \times \hat{\mathbf{C}}_8^t = \mathbf{I}_8$. L'équation implique que les éléments du $\hat{\mathbf{C}}_8$ satisfassent l'ensemble d'équations algébriques suivantes:

$$df - eg - jh + ik = 0, \quad (\text{II.11})$$

$$dk + ej - hg - if = 0, \quad (\text{II.12})$$

$$b^2 + c^2 = d^2 + e^2 + h^2 + i^2 = f^2 + g^2 + j^2 + k^2 = a^2 \quad (\text{II.13})$$

Ce qui doit être résolu dans le domaine des nombres entiers. Les équations (II.11) et (II.12) sont des conditions d'orthogonalité qui assurent que les vecteurs de base de $\hat{\mathbf{C}}_8$ sont orthogonaux entre eux. L'équation (II.13) est la condition de normalité, et lui signifie que la norme 2 de chaque vecteur de base de $\hat{\mathbf{C}}_8$ est constante et égale à l'unité. Et en fin, afin de construire les vecteurs de base de $\hat{\mathbf{C}}_8$ ressemblés à ceux de la matrice de conversion en valeurs réelles \mathbf{C}_8 (les magnitudes des éléments de $\hat{\mathbf{C}}_8$ sont approximativement proportionnelles à celles des éléments de \mathbf{C}_8), on doit prendre les contraintes sur les variables $a, b, c, d, e, f, g, h, i, j, k$ en comparant les magnitudes des éléments de la matrice \mathbf{C}_8 et ceux de la matrice $\hat{\mathbf{C}}_8$ on obtient les inégalités suivantes [38] :

$$d > g > j > h > k > f > i > e > 0, \quad (\text{II.14})$$

$$j > c > h, \quad (\text{II.15})$$

$$b > d, \quad (\text{II.16})$$

$$a > b. \quad (\text{II.17})$$

Toutes les solutions des nombres entiers satisfaisant (II.11)-(II.13) sous des contraintes données par les équations (II.14)-(II.17) garantiront que la matrice de conversion approximée $\hat{\mathbf{C}}_8$ est orthonormale et proche de la matrice de conversion originale \mathbf{C}_8 . Mais la recherche exhaustive par ordinateur des solutions de (II.11)-(II.13) sous les contraintes (II.14)-(II.17) pour une représentation sur 6 bits ne donne pas des résultats. Cependant, il faut faire une analyse soigneuse en modifiant certaines contraintes pour trouver des solutions, ces solutions malgré qu'elles portent des améliorations en terme de MSE et de performances, elles exigent beaucoup plus d'opérations arithmétiques [38].

L'approximation de la DCT par la transformée de Hadamard a été étendu pour une taille de $N=16$ et $N=32$ [42], mais toutes ces approximations sont obtenues par des essais ce qui produit des erreurs.

La méthode décrite ci-dessous est développée pour remédier à ces problèmes, elle est basée sur le principe de la symétrie dyadique de la matrice DCT. Son principe est décrit dans la section suivante et ces détails peuvent être trouvés dans [8], [41], [43]-[45].

II.3.2. Méthode basée sur le principe de la symétrie dyadique

Une méthode qui exploite la symétrie de la matrice DCT pour générer des transformées entières est proposée dans [8], [41], [43]-[45] dont le principe est expliqué ci-dessous:

On cherche la matrice approximée $\hat{\mathbf{T}}_8$ de la matrice DCT sous la forme:

$$\hat{\mathbf{T}}_8 = \mathbf{D}_8 \mathbf{V}_8 \quad (\text{II.18})$$

où \mathbf{D}_8 est une matrice diagonale dont les éléments sont des facteurs de normalisation, et \mathbf{V}_8 est une matrice des nombres entiers.

La procédure pour construire la matrice approximée $\hat{\mathbf{T}}_8$ à partir de la matrice DCT \mathbf{T}_8 inclue les étapes suivantes:

1- Générer la matrice DCT \mathbf{T}_8 dans sa forme simple.

$$\mathbf{T}_8 = \frac{1}{2} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos \frac{\pi}{16} & \cos \frac{3\pi}{16} & \sin \frac{3\pi}{16} & \sin \frac{\pi}{16} & -\sin \frac{\pi}{16} & -\sin \frac{3\pi}{16} & -\cos \frac{3\pi}{16} & -\cos \frac{\pi}{16} \\ \cos \frac{\pi}{8} & \sin \frac{\pi}{8} & -\sin \frac{\pi}{8} & -\cos \frac{\pi}{8} & -\cos \frac{\pi}{8} & -\sin \frac{\pi}{8} & \sin \frac{\pi}{8} & \cos \frac{\pi}{8} \\ \cos \frac{3\pi}{16} & -\sin \frac{\pi}{16} & -\cos \frac{\pi}{16} & -\sin \frac{3\pi}{16} & \sin \frac{3\pi}{16} & \cos \frac{\pi}{16} & \sin \frac{\pi}{16} & -\cos \frac{3\pi}{16} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \sin \frac{3\pi}{16} & -\cos \frac{\pi}{16} & \sin \frac{\pi}{16} & \cos \frac{3\pi}{16} & -\cos \frac{3\pi}{16} & -\sin \frac{\pi}{16} & \cos \frac{\pi}{16} & -\sin \frac{3\pi}{16} \\ \sin \frac{\pi}{8} & -\cos \frac{\pi}{8} & \cos \frac{\pi}{8} & -\sin \frac{\pi}{8} & -\sin \frac{\pi}{8} & \cos \frac{\pi}{8} & -\cos \frac{\pi}{8} & \sin \frac{\pi}{8} \\ \sin \frac{\pi}{16} & -\sin \frac{3\pi}{16} & \cos \frac{3\pi}{16} & -\cos \frac{\pi}{16} & \cos \frac{\pi}{16} & -\cos \frac{3\pi}{16} & \sin \frac{3\pi}{16} & -\sin \frac{\pi}{16} \end{bmatrix} \quad (\text{II.19})$$

2- Remplacer les éléments de la matrice \mathbf{T}_8 par des variables dont les valeurs sont assumées d'être entières. Observant les valeurs absolues des éléments de \mathbf{T}_8 on trouve (II.20).

$$\begin{aligned}
& |t_{00}| = |t_{01}| = |t_{02}| = |t_{03}| = |t_{04}| = |t_{05}| = |t_{06}| = |t_{07}| \\
& = |t_{40}| = |t_{41}| = |t_{42}| = |t_{43}| = |t_{44}| = |t_{45}| = |t_{46}| = |t_{47}| = 1/\sqrt{2} \\
& |t_{10}| = |t_{17}| = |t_{32}| = |t_{35}| = |t_{51}| = |t_{56}| = |t_{73}| = |t_{74}| = \cos \frac{\pi}{16} \\
& |t_{11}| = |t_{16}| = |t_{30}| = |t_{37}| = |t_{53}| = |t_{54}| = |t_{72}| = |t_{75}| = \cos \frac{3\pi}{16} \\
& |t_{12}| = |t_{15}| = |t_{33}| = |t_{34}| = |t_{50}| = |t_{57}| = |t_{71}| = |t_{76}| = \sin \frac{3\pi}{16} \\
& |t_{13}| = |t_{14}| = |t_{31}| = |t_{36}| = |t_{52}| = |t_{55}| = |t_{70}| = |t_{77}| = \sin \frac{\pi}{16} \\
& |t_{20}| = |t_{23}| = |t_{24}| = |t_{27}| = |t_{61}| = |t_{62}| = |t_{65}| = |t_{66}| = \cos \frac{\pi}{8} \\
& |t_{21}| = |t_{22}| = |t_{25}| = |t_{26}| = |t_{60}| = |t_{63}| = |t_{64}| = |t_{67}| = \sin \frac{\pi}{8}
\end{aligned} \tag{II.20}$$

On représente les éléments de même magnitude par une seule variable de l'ensemble $\{a, b, c, d, e, f, g\}$, où $a, b, c, d, e, f, g \in \mathbb{N}$ sont des entiers. Cette propriété de nombres entiers assure que la matrice \mathbf{V}_8 dans (II.18) peut être représentée par un nombre fini de bits.

La matrice approximée $\hat{\mathbf{T}}_8$ peut être exprimée par:

$$\hat{\mathbf{T}}_8 = \mathbf{D}_8 \times \begin{bmatrix} g & g & g & g & g & g & g & g \\ a & b & c & d & -d & -c & -b & -a \\ e & f & -f & -e & -e & -f & f & e \\ b & -d & -a & -c & c & a & d & -b \\ g & -g & -g & g & g & -g & -g & g \\ c & -a & d & b & -b & -d & a & -c \\ f & -e & e & -f & -f & e & -e & f \\ d & -c & b & -a & a & -b & c & -d \end{bmatrix} \tag{II.21}$$

3. Trouver les conditions d'orthogonalité sous lesquelles le $i^{\text{ième}}$ et $j^{\text{ième}}$ vecteurs de base (lignes) \mathbf{v}_i et \mathbf{v}_j de la matrice \mathbf{V}_8 sont orthogonaux, c'est-à-dire le produit scalaire de \mathbf{v}_i et \mathbf{v}_j satisfait la relation suivante :

$$\mathbf{v}_i \mathbf{v}_j^t = \sum_{k=0}^7 \mathbf{v}_{ik} \mathbf{v}_{kj} = 0, \quad \forall i \neq j, \quad i, j = 0, 1, \dots, 7. \tag{II.22}$$

Cette condition assure que la matrice entière \mathbf{V}_8 est orthonormale. La Multiplication de cette matrice avec la matrice diagonale \mathbf{D}_8 dans (II.21) assure que la $2^{\text{ième}}$ norme de chaque vecteur de base \mathbf{v}_i de la matrice \mathbf{V}_8 est égale à l'unité.

$$\|\mathbf{v}_i\|_2 = \sqrt{\sum_{k=0}^7 \mathbf{v}_{ik}^2} = 1, \quad i, k = 0, 1, \dots, 7 \quad (\text{II.23})$$

et assure que $\hat{\mathbf{T}}_8$ est orthogonale. Pour cela, les éléments de la matrice \mathbf{D}_8 doivent être exprimés en fonction de ceux de \mathbf{V}_8 comme :

$$d_{ii} = \|\mathbf{v}_i\|_2^2 \quad (\text{II.24})$$

car

$$\mathbf{V}_8 \mathbf{V}_8^t = [\mathbf{D}_8^{-1}]^2 \mathbf{I}_8 \quad (\text{II.25})$$

L'application de (II.22) sur (II.21) conduit d'après [8] à l'équation (II.26) :

$$a(b - c) - d(b + c) = 0 \quad (\text{II.26})$$

Dû à la symétrie dyadique des vecteurs de base de la DCT, la condition d'orthogonalité (II.26) dévoile que seulement la condition qui relie les variables a, b, c et d assure l'orthogonalité de la transformée $\hat{\mathbf{T}}_8$. Ceci signifie que les variables e, f et g peuvent être considérées indépendamment. D'autre part, les éléments de la matrice diagonale \mathbf{D}_8 sont donnée par :

$$\begin{aligned} d_{00} &= d_{44} = 8g^2, \\ d_{11} &= d_{33} = d_{55} = d_{77} = 2(a^2 + b^2 + c^2 + d^2), \\ d_{22} &= d_{66} = 4(e^2 + f^2). \end{aligned} \quad (\text{II.27})$$

4. Exprimer les contraintes sur les variables a, b, c, d, e, f, g sous forme d'inégalités. Celles-ci assurent que la forme générale de la matrice entière \mathbf{V}_8 résultante soit similaire à celle de la matrice \mathbf{T}_8 de la DCT conventionnelle. En plus de (II.26), nous obtenons trois inégalités:

$$a > b > c > d > 0, \quad (\text{II.28})$$

$$a > e > f > 0, \quad (\text{II.29})$$

$$a > g > 0. \quad (\text{II.30})$$

5- Trouver un ensemble de nombres entiers $\{a, b, c, d, e, f, g\}$ qui satisfont l'équation algébrique (II.26) sous les contraintes (II.28)-(II.30). Pour trouver des solutions en nombres

entiers pour l'ensemble $\{a, b, c, d, e, f, g\}$, une recherche exhaustive par ordinateur doit être exécutée. Généralement l'ensemble d'équations algébriques peut avoir un nombre infini de solutions. Ceci implique qu'on peut trouver un nombre infini de transformations entières.

Plusieurs matrices ICT (Integer Cosine Transform) d'ordre 8 sont données dans [8], en particulier, celles qui ont des performances comparables à celle de la DCT et qui exigent 4 bits pour la présentation de ces éléments qui sont très attractives pour l'implémentation pratique.

Cette méthode [8], [41], [43]-[45] qui est basée sur le principe de la symétrie dyadique est étendue dans [41] pour le cas de $N=16$. L'inconvénient majeur de cette méthode de génération des transformées entières est qu'elle exige une recherche exhaustive qui sera plus complexe lorsque la taille de la DCT augmente, bien que les solutions ne sont pas toujours garanties [8].

II.3.3. Méthode basée sur le principe lifting

T.D.Tran [46]-[48] présente l'implémentation et l'application de beaucoup de familles des transformées DCT rapides avec le principe lifting, appelées BinDCT. Ces transformées sont obtenues à partir des factorisations de Chen et Loeffler [48] en remplaçant chaque rotation en plan de celles-ci par 3 liftings et après chaque coefficient lifting est remplacé par un nombre dyadique rationnel ; c.à.d des nombres fractionnels de la forme $\frac{k}{2^b}$ où k et $b \in \mathbb{N}$ et k est un entier impair [48].

- Principe lifting et la rotation en plan

La figure (II.1) illustre la décomposition d'une rotation en plan à 3 étapes de lifting [48].

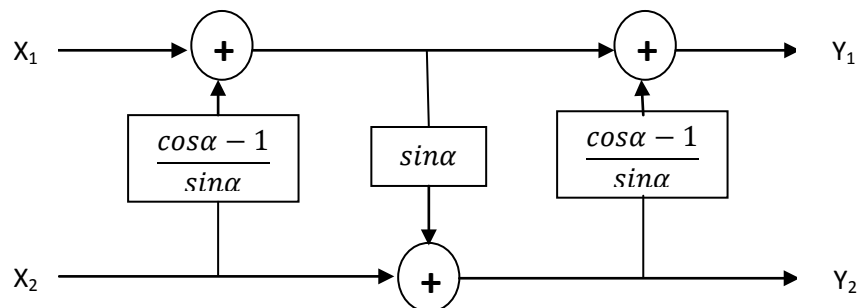


Figure (II.1) : Représentation de la rotation en plan directe par trois étapes lifting

Cette décomposition peut être exprimée sous forme matricielle par l'équation donnée par (II.31).

$$\begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} = \begin{bmatrix} 1 & p \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ u & 1 \end{bmatrix} \begin{bmatrix} 1 & p \\ 0 & 1 \end{bmatrix} \quad (\text{II.31})$$

où $p = (\cos \alpha - 1) / \sin \alpha$ et $u = \sin \alpha$

Chaque étape de lifting est une transformée biorthogonale est son inverse aussi a une structure lifting simple, i.e.,

$$\begin{bmatrix} 1 & x \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -x \\ 0 & 1 \end{bmatrix} \quad (\text{II.32})$$

Alors, l'inverse de la rotation en plan peut être décomposée aussi à trois étapes de lifting comme :

$$\begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -p \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -u & 1 \end{bmatrix} \begin{bmatrix} 1 & -p \\ 0 & 1 \end{bmatrix} \quad (\text{II.33})$$

La figure (II.2) représente cette décomposition inverse:

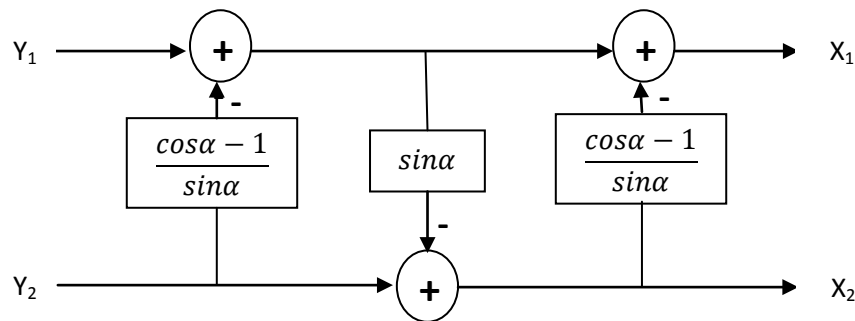


Figure (II.2) : Représentation de la rotation en plan inverse par trois étapes lifting [48]

où les éléments de multiplication représentent les coefficients lifting.

Cette méthode possède non seulement la propriété des nombres entiers mais aussi ces éléments sont des entiers dyadiques rationaux, dont la multiplication par un nombre rationnel dyadique peut être réalisée en utilisant des opérations binaires; le multiplicande est multiplié premièrement par le numérateur k et le résultat est ensuite décalé à droite par b bits.

Plusieurs versions entières avec leurs structures rapides directes et inverses sont proposées dans [46]-[48] n'exigent que des additions et des décalages. Les matrices de la

transformée BinDCT sont données dans [46]-[48], la structure ainsi que les transformées directes et inverses de la version la plus simple (C) sont données dans la figure (II.3) et les équations (II.34) et (II.35) respectivement.

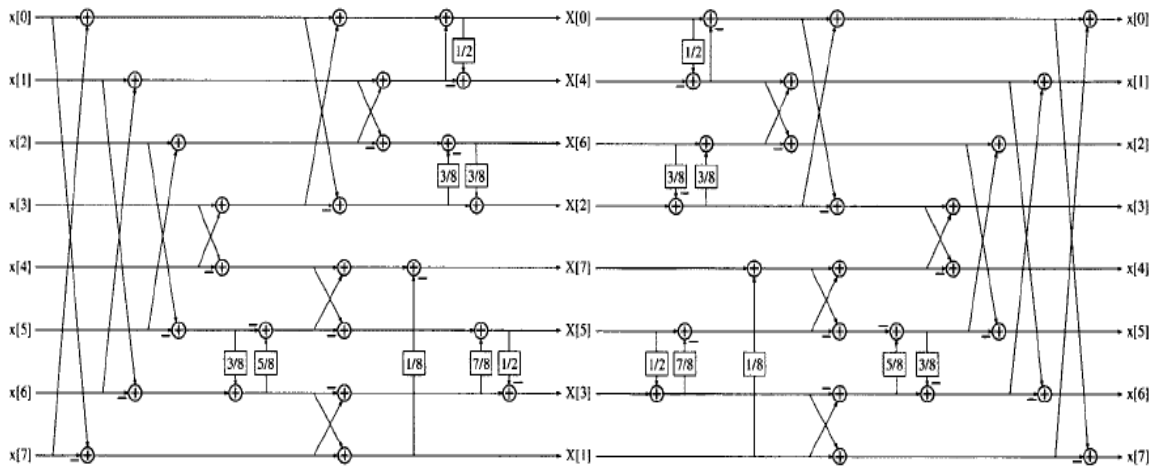


Figure (II.3) : Structure de la Bin DCT version C [46]

$$\text{binDCT} - C = \begin{bmatrix} 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & 3/16 & 0 & 0 & -3/16 & -1/2 & -1/2 \\ 55/128 & 3/16 & -3/16 & -55/128 & -55/128 & -3/16 & 3/16 & 55/128 \\ 9/32 & -1/8 & -19/64 & -1/4 & 1/4 & 19/64 & 1/8 & -9/32 \\ 1/4 & -1/4 & -1/4 & 1/4 & 1/4 & -1/4 & -1/4 & 1/4 \\ 7/16 & -3/4 & 7/32 & 1/2 & -1/2 & -7/32 & 3/4 & -7/16 \\ -3/16 & 1/2 & -1/2 & 3/16 & 3/16 & -1/2 & 1/2 & -3/16 \\ -1/16 & 1/4 & -13/32 & 1/2 & -1/2 & 13/32 & -1/4 & 1/16 \end{bmatrix} \quad (\text{II.34})$$

$$\text{binDCT} - C - \text{inv} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 & 1/4 & 1/4 & 1/4 & 1/4 \\ 1/2 & 13/32 & 1/4 & 1/16 & -1/16 & -1/4 & -13/32 & -1/2 \\ 1/2 & 3/16 & -3/16 & -1/2 & -1/2 & -3/16 & 3/16 & 1/2 \\ 1/2 & -7/32 & -3/4 & -7/16 & 7/16 & 3/4 & 7/32 & -1/2 \\ 1/2 & -1/2 & -1/2 & 1/2 & 1/2 & -1/2 & -1/2 & 1/2 \\ 1/4 & -19/64 & 1/8 & 9/32 & -9/32 & -1/8 & 19/64 & -1/4 \\ -3/16 & 55/128 & -55/128 & 3/16 & 3/16 & -55/128 & 55/128 & -3/16 \\ 0 & 3/16 & -1/2 & 1/3 & -1/2 & 1/2 & -3/16 & 0 \end{bmatrix} \quad (\text{II.35})$$

Le calcul rapide de la version C exige 30 additions et 13 décalages. Malgré que cette transformée entière (version C) est considérée comme la version la plus simple de la famille BinDCT, elle reste toujours complexe et exige un grand nombre d'opérations.

II.3.4. Méthode basée sur les algorithmes à complexité variable

K. Lengwehasatit et all. [11], [49] proposent un algorithme d'approximation DCT basé sur les algorithmes à complexité variable, 5 niveaux d'approximations sont utilisés dont la structure générale est représentée sur la figure (II.4).

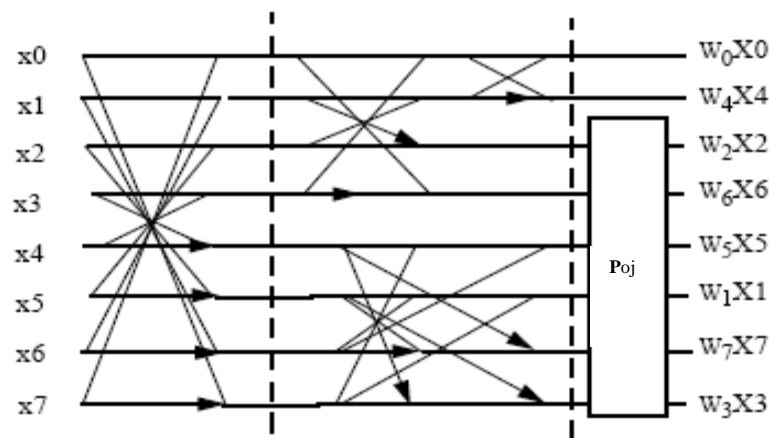


Figure (II.4) : Algorithme de la DCT approximée [49]

Les lignes flèches représentent une multiplication par -1 avant l'addition.

Cette structure est modifiée à partir de la structure de l'algorithme rapide donnée dans [51]. Pour la matrice DCT conventionnelle, la matrice \mathbf{P}_0 contient des facteurs de multiplication non rationnels. D'après la figure (II.4), plusieurs algorithmes peuvent être produits avec différents niveaux de complexité en remplaçant la matrice \mathbf{P}_0 par l'une de plusieurs matrices approximées, notées par \mathbf{P}_{0j} pour $j=1, \dots, J$, où J est le nombre des niveaux de l'approximation. La matrice équivalente en utilisant \mathbf{P}_{0j} est donnée par $\hat{\mathbf{D}}_j$. Ces matrices sont données par les équations suivantes [49]:

$$p_{0j} = \begin{bmatrix} q_1 & 0 \\ 0 & I \end{bmatrix} \quad (\text{II.36})$$

$$p_{o2} = \begin{bmatrix} q_1 & \bar{0}^t & \bar{0}^t & \bar{0}^t & \bar{0}^t \\ \bar{0} & 1 & 0 & 1/8 & -1/8 \\ \bar{0} & -1/8 & 1 & 1/8 & 0 \\ \bar{0} & 0 & -1/8 & 1 & 1/8 \\ \bar{0} & 1/8 & 1/8 & 0 & 1 \end{bmatrix} \quad (\text{II.37})$$

$$p_{o3} = \begin{bmatrix} q_2 & \bar{0}^t & \bar{0}^t & \bar{0}^t & \bar{0}^t \\ \bar{0} & 1 & 1/8 & 1/8 & -1/8 \\ \bar{0} & -1/8 & 1 & 1/8 & -1/8 \\ \bar{0} & -1/8 & -1/8 & 1 & 1/8 \\ \bar{0} & 1/8 & 1/8 & -1/8 & 1 \end{bmatrix} \quad (\text{II.38})$$

$$p_{o4} = \begin{bmatrix} q_3 & \bar{0}^t & \bar{0}^t & \bar{0}^t & \bar{0}^t \\ \bar{0} & 1 & 1/8 & 1/8 & -1/4 \\ \bar{0} & -1/8 & 1 & 1/4 & -1/8 \\ \bar{0} & -1/8 & -1/4 & 1 & 1/8 \\ \bar{0} & 1/4 & 1/8 & -1/8 & 1 \end{bmatrix} \quad (\text{II.39})$$

$$p_{o5} = \begin{bmatrix} q_3 & \bar{0}^t & \bar{0}^t & \bar{0}^t & \bar{0}^t \\ \bar{0} & 1 & 1/8 & 1/8 & -3/16 \\ \bar{0} & -1/8 & 1 & 3/16 & -1/8 \\ \bar{0} & -1/8 & -3/16 & 1 & 1/8 \\ \bar{0} & 3/16 & 1/8 & -1/8 & 1 \end{bmatrix} \quad (\text{II.40})$$

avec :

$$q_2 = \begin{bmatrix} 1 & -3/8 \\ 1/2 & 1 \end{bmatrix}, q_1 = \begin{bmatrix} 1 & -1/2 \\ 1/2 & 1 \end{bmatrix}, q_3 = \begin{bmatrix} 1 & -3/8 \\ 3/8 & 1 \end{bmatrix}, \bar{0} = [0 \ 0] \quad (\text{II.41})$$

Les sorties des transformées approximatives DCT sont $\hat{X}_i = W_i X_i$ pour $i = 0, 1, \dots, 7$, où W_i est un facteur de normalisation qui peut être incorporé dans la phase de quantification.

Le nombre d'opérations exigé pour ces approximations (référé comme # 1,2,3,4 et 5) sont 24 adds+2 shifts, 33 adds+7 shifts, 38 adds+8 shifts, 38 adds+12 shifts et 42 adds+12 shifts, respectivement. La matrice qui exige la plus faible complexité (seulement 24 adds+2 shifts) parmi ces approximations est \hat{D}_1 qui est donnée par (II.42).

$$\hat{\mathbf{D}}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0.5 & -0.5 & -1 & -1 & -0.5 & 0.5 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0.5 & -1 & 1 & -0.5 & -0.5 & 1 & -1 & 0.5 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix} \quad (\text{II.42})$$

Alors, la complexité de la matrice $\hat{\mathbf{D}}_1$ est faible par rapport à celle de la matrice (II.34) de [46]. Cette réduction est réalisée sans affecter les performances.

II.3.5. Méthode basée sur l'application de la fonction *Signe*

Le principe donné dans [10] est très simple, il consiste à appliquer la fonction *signe* sur les éléments de la matrice DCT, la transformée résultante est nommée SignDCT (SDCT), notée par \mathbf{T}_N^{SDCT} , et donnée par :

$$T_N^{SDCT}(i, j) = \frac{1}{\sqrt{N}} \text{sign}\{T_N(i, j)\} \quad (\text{II.43})$$

où « sign » désigne la fonction *signe*, défini par :

$$\text{sign}\{x\} = \begin{cases} +1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ -1 & \text{if } x < 0. \end{cases} \quad (\text{II.44})$$

L'application de la fonction *signe* sur la matrice DCT 8×8 donne la matrice suivante:

$$\mathbf{T}_{SDCT} = \frac{1}{\sqrt{8}} \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & -1 & +1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & -1 & +1 & +1 & -1 & -1 & +1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \end{bmatrix} \quad (\text{II.45})$$

Plusieurs avantages sont immédiatement apparait pour la matrice SignDCT : tous ses éléments sont ± 1 donc aucune multiplication n'est exigée, elle maintient la périodicité de la

DCT et la structure spectrale, par conséquent elle a des caractéristiques de compactage d'énergie et de décorrélation. Ce fait peut être vérifié pour une valeur donnée de N en comparant des spectres de DCT et de SignDCT qui sont très étroits entre eux [10].

Puisque la SignDCT est dérivée de la DCT, n'importe quel algorithme rapide de calcul de DCT peut être adopté pour le calcul efficace de la SignDCT. Mais ce ci est valide pour la SignDCT directe seulement, parce que la matrice \mathbf{T}_{SDCT} n'est pas orthogonale. Cependant, il est inversible et son inverse est donnée par :

$$\mathbf{T}_{SDCT}^t = \frac{1}{\sqrt{8}} \begin{bmatrix} +1 & +2 & +1 & +2 & +1 & 0 & +1 & 0 \\ +1 & +2 & +1 & 0 & -1 & -2 & -1 & 0 \\ +1 & 0 & -1 & -2 & -1 & 0 & +1 & +2 \\ +1 & 0 & -1 & 0 & +1 & +2 & -1 & -2 \\ +1 & 0 & -1 & 0 & +1 & -2 & -1 & +2 \\ +1 & 0 & -1 & +2 & -1 & 0 & +1 & -2 \\ +1 & -2 & +1 & 0 & -1 & +2 & -1 & 0 \\ +1 & -2 & +1 & -2 & +1 & 0 & +1 & 0 \end{bmatrix} \quad (\text{II.46})$$

La matrice SignDCT (II.45) de [10] a une complexité comparable à celle de la matrice $\hat{\mathbf{D}}_1$ (II.42) de [11].

II.3.6. Méthode basée sur l'insertion appropriée des éléments nuls dans la SDCT

La méthode donnée dans [12] et [50] consiste à insérer des éléments nuls dans la matrice SDCT proposée dans [10], dont le but est d'améliorer ses performances en compression d'images.

La première matrice est donnée dans [12], elle a un bon compactage d'énergie en comparaison avec la SDCT et exige moins de nombre d'opérations arithmétiques; son calcul rapide exige seulement 18 additions et 2 décalages, par contre la SDCT exige 24 addition, donc il est claire qu'il y a une réduction de 25% dans le nombre d'opérations arithmétiques.

La matrice de [12] est donnée par l'équation (II.47).

$$\mathbf{C}_1 = \mathbf{D}_1 \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 1/2 & -1/2 & -1 & -1 & -1/2 & 1/2 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1/2 & -1 & 1 & 1/2 & -1/2 & 1 & -1 & 1/2 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (\text{II.47})$$

où \mathbf{D}_1 est une matrice diagonale d'orthogonalisation, donnée par:

$$\mathbf{D}_1 = \text{diag} \left(\frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}} \right) \quad (\text{II.48})$$

La matrice proposée dans [50] exige 21 additions et 3 décalages de bit. Elle est donnée par :

$$\mathbf{C}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & -1 & 0 & 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 0 & 0 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \quad (\text{II.49})$$

Mais, cette matrice n'est pas orthogonale, sa matrice inverse est donnée par:

$$\mathbf{C}_2^{-1} = \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 & 2 & 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 0 & -1 & -2 & -1 & 0 \\ 1 & 1 & -1 & -2 & -1 & 1 & 1 & 0 \\ 1 & 0 & -1 & 0 & 1 & 4 & -1 & -4 \\ 1 & 0 & -1 & 0 & 1 & -4 & -1 & 4 \\ 1 & -1 & -1 & 2 & -1 & -1 & 1 & 0 \\ 1 & -2 & 1 & 0 & -1 & 2 & -1 & 0 \\ 1 & -1 & 1 & -2 & 1 & -1 & 1 & 0 \end{bmatrix} \quad (\text{II.50})$$

La matrice (II.49) présente une réduction de 12,5 % dans le nombre d'opérations par rapport à la SDCT et possède des bonnes performances de compression dans le cas des taux de compression élevés par rapport à la SDCT et à la transformée de [12].

Cette méthode ne garantit pas de trouver des transformées orthogonales, d'ailleurs la deuxième transformée générée par cette méthode n'est pas orthogonale et elle donne des bons résultats juste pour la compression en taux élevé.

II.4. Conclusion

Dans ce chapitre, nous avons donné une étude bibliographique sur les transformées DCTs entières: leurs complexités arithmétiques, leurs critères de développement et quelques méthodes pour leurs constructions. La méthode la plus simple est celle de la SDCT qui conserve les propriétés de la DCT conventionnelle, mais son inconvénient est qu'elle n'est pas orthogonale et son efficacité dans la compression est un peu plus loin de la DCT. Le chapitre suivant est consacré au développement des algorithmes de génération des transformées entières en se basant sur une combinaison de deux algorithmes donnés dans [8] et [50].



**CHAPITRE
III**

Transformées DCT entières proposées

III.1. Introduction

L'intérêt qu'apporte les transformées DCT entières nous a motivé à développer deux transformées entières dédiées à la compression d'images. Le développement de la première transformée est basé sur une mise à zéro approprié des coefficients de la SDCT qui ont une faible signification perceptuelle et une énergie très inférieure à celles des autres coefficients. Par contre, le développement de la deuxième transformée est basé sur une combinaison entre la méthode proposée pour la première transformée et le principe décrit dans [8]. Cette deuxième méthode que nous avons déjà publié dans [16] assure l'orthogonalité des transformées générées. Les deux transformées résultantes ont un bon compromis entre les performances en compression d'images et le nombre d'opérations exigées pour leur calcul.

Dans ce chapitre, nous donnons au début le principe des méthodes proposées avec les matrices résultantes, après nous développons les algorithmes rapides pour leur calcul, une comparaison en terme de complexité avec des transformées existantes est en suite présentée et en se terminant par la structure (data flow graph) de chacune d'elles, ces structures facilitent l'implémentation de ces transformées.

III.2. Transformées entières proposées

Beaucoup d'algorithmes de conversion de la DCT à ses versions entières ont été proposées dans la littérature (vues dans le chapitre précédent), la SDCT figure parmi ces algorithmes la plus simple dont plusieurs avantages sont apparus pour la compression d'images [10] : tous ses éléments sont ∓ 1 , aucune multiplication n'est exigée, elle maintient la périodicité et les propriétés spectrales de la DCT, et par conséquent, elle a des bonnes caractéristiques de décorrelation et de compactage d'énergie [10]. Et puisque la SDCT est dérivée à partir de la DCT, n'importe quel algorithme pour le calcul rapide de la DCT est adopté pour elle mais il est valide seulement pour la SDCT directe; car la SDCT n'est pas orthogonale, à cet égard, nous avons amélioré les performances de la SDCT en remplaçant quelques de ses éléments par des zéros. Ces éléments sont ceux qui ont une faible signification perceptuelle et aussi ont une énergie plus faible que les autres coefficients et le choix est fait par des applications de la SDCT sur différentes images de test standards.

- La transformée proposée résultante est donnée par l'équation (III.1).

$$\mathbf{T}_1 = \mathbf{D}_{\mathbf{T}_1} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (\text{III.1})$$

$$= \mathbf{D}_{\mathbf{T}_1} \times \mathbf{C}_{\text{proposée1}}$$

La matrice d'orthogonalisation de \mathbf{T}_1 est donnée directement par :

$$\mathbf{D}_{\mathbf{T}_1} = \text{diag} \left(\frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{2}, \frac{1}{\sqrt{2}} \right) \quad (\text{III.2})$$

Une observation sur la matrice proposée (III.1) nous permet de voir sa simplicité du coté complexité du calcul, presque la moitié de ces éléments sont nuls (43,75 %), la matrice diagonale est intégrée dans la phase de la quantification.

La propriété d'orthogonalité est peut être vérifiée simplement par l'équation:

$$\mathbf{T}_1 \times \mathbf{T}_1^{-1} = \mathbf{I} \quad (\text{III.3})$$

- La deuxième transformée est le résultat d'une combinaison entre la première méthode donnée ci-dessus et la méthode basée sur la symétrie dyadique donnée dans [8].

La projection de la matrice SDCT donnée par l'équation (II.45) sur la matrice des variables donnée par (II.21) nous a conduit à trouver que tous les variables sont égale à 1.

$$a = b = c = d = e = f = g = 1 \quad (\text{III.4})$$

Mais la matrice SDCT ne satisfait pas la condition d'orthogonalité donnée par l'équation (II.26). Notre méthode hybride consiste à appliquer la méthode de remplacement des éléments appropriés de la matrice SDCT par des éléments nuls toute en respectant la condition

d'orthogonalité discutée dans le chapitre II. Alors, la deuxième matrice proposée est donnée par l'équation (III.5).

$$\begin{aligned} \mathbf{T}_2 &= \mathbf{D}_{\mathbf{T}_2} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix} \\ &= \mathbf{D}_{\mathbf{T}_2} \times \mathbf{C}_{\text{proposée 2}} \end{aligned} \quad (\text{III.5})$$

avec $\mathbf{D}_{\mathbf{T}_2}$ est la matrice diagonale pour rendre la transformée proposée orthogonale donnée d'après l'équation (II. 27) par :

$$\mathbf{D}_{\mathbf{T}_2} = \text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}} \right) \quad (\text{III.6})$$

D'après (III.5), on remarque que tous les éléments sont des ± 1 et 0, donc pas de multiplication exigée. On peut vérifier facilement que la matrice est orthogonale :

$$\mathbf{T}_2 \times \mathbf{T}_2^{-1} = \mathbf{I} \quad (\text{III.7})$$

Cette orthogonalité permet de calculer la transformée directe et inverse par le même algorithme.

\mathbf{T}_2 est une matrice DCT 8×8 entière exige le même nombre d'opérations que celle de la SDCT (\mathbf{T}_{SDCT}) proposée dans [10] mais elle est orthogonale ce qui n'est pas le cas pour la SDCT où 12% des éléments non diagonales de $\mathbf{T}_{SDCT} \times \mathbf{T}_{SDCT}^t$ d'ordre 8 sont non nuls.

III.3. Algorithmes rapides pour les transformées proposées

Dans cette section, nous proposons des algorithmes rapides pour les deux transformées proposées ci-dessus.

III.3.1. Algorithme rapide pour la transformée T_1

L'algorithme est basé sur une factorisation en produit de matrices simples. En commence par la réorganisation de la matrice $C_{proposée1}$ donnée dans l'équation (III.1) pour grouper les lignes symétriques ensemble et les lignes antisymétriques ensemble, cette réorganisation n'est pas strictement nécessaire, son but est seulement de clarifier les étapes de factorisation. Le résultat de cette étape est donné par:

$$C_{proposée1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (III.8)$$

La factorisation en produit de matrices plus simples nous donne:

$$C_{proposée1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.9)$$

Les éléments différents de zéro au coin gauche supérieur de la matrice centrale forment le même modèle de symétrie que les lignes de la matrice dans l'équation (III.8). La factorisation encore une fois de ce coin nous donne l'équation (III.10).

$$\mathbf{C}_{proposée1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (\text{III.10})$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Si on note:

$$\mathbf{G}_3 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (\text{III.11})$$

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.12})$$

$$\mathbf{G}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.13})$$

La matrice $\mathbf{C}_{proposée1}$ peut être écrite comme :

$$\mathbf{C}_{proposée1} = \mathbf{G}_3 \times \mathbf{G}_2 \times \mathbf{G}_1 \quad (\text{III.14})$$

- **Calcul de la complexité de la transformée $C_{proposée 1}$**

La complexité des transformées est calculée pour un vecteur de taille 8×1 , pour la transformation d'une matrice de 8×8 , il suffit de multiplier le nombre d'opérations exigées pour un vecteur par 8,

- La multiplication d'un vecteur de 8 entrées par G_1 exige 8 additions.
- La multiplication d'un vecteur de 8 entrées par G_2 exige 4 additions.
- La multiplication d'un vecteur de 8 entrées par G_3 exige 4 additions.

Alors, la multiplication d'un vecteur de 8 éléments par la matrice transformée $C_{proposée 1}$ exige 16 additions seulement, elle présente donc une réduction supérieure à 33% dans le nombre d'opérations par rapport à la SDCT [10] et à la transformée donnée dans [11], et plus de 11% par rapport à la transformée donnée dans [12].

III.3.2. Algorithme rapide pour la transformée T_2

Si la matrice $C_{proposée 2}$ donnée par l'équation (III.5) est divisée en deux moitiés verticalement, la moitié gauche de chaque ligne est une image symétrique ou antisymétrique de l'autre moitié. On peut réorganiser la matrice pour grouper les lignes symétriques ensemble et les lignes antisymétriques ensemble, on trouve :

$$C_{proposée 2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix} \quad (III.15)$$

ou encore,

$$C_{proposée 2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad (III.16)$$

Dans l'équation (III.16), les éléments différents de zéro au coin gauche supérieur de la matrice centrale forment le même modèle de symétrie que les lignes de la matrice dans l'équation (III.5). On va factoriser encore de la même façon ce coin de matrice (équation (III.17)).

$$\mathbf{C}_{proposée 2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 \end{bmatrix} \quad (\text{III.17})$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

L'équation (III.17) apparaît d'être beaucoup plus compliquée que l'originale dans l'équation (III.5), mais elle exige réellement moins d'étapes de calcul.

Si on note :

$$\mathbf{H}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.18})$$

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.19})$$

$$\mathbf{H}_3 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 \end{bmatrix} \quad (\text{III.20})$$

Alors, la matrice $\mathbf{C}_{\text{proposée 2}}$ est donnée par :

$$\mathbf{C}_{\text{proposée 2}} = \mathbf{H}_3 \times \mathbf{H}_2 \times \mathbf{H}_1 \quad (\text{III.21})$$

- **Calcul de la complexité de la transformée $\mathbf{C}_{\text{proposée 2}}$**

- La multiplication d'un vecteur de 8 entrées par \mathbf{H}_1 exige 8 additions.
- La multiplication d'un vecteur de 8 entrées par \mathbf{H}_2 exige 4 additions.
- La multiplication d'un vecteur de 8 entrées par \mathbf{H}_3 exige 12 additions.

Alors, la multiplication d'un vecteur ligne de taille 8 par $\mathbf{C}_{\text{proposée 2}}$ exige 24 additions. Il est clair donc que la transformée proposée, la SDCT [10] et la transformée dans [11] ont des complexités comparables.

Une comparaison générale en terme de complexité entre les transformées proposées et les autres transformées existantes sont données dans le tableau suivant.

Opérations	DCT [28]	Transformée de [11]	Transformée de [12]	Transformée proposée \mathbf{T}_1	Transformée proposée \mathbf{T}_2	Transformée de [50]	SDCT de [10]
Additions	29	24	18	16	24	21	24
Bit-shifts	0	2	2	0	0	3	0
Multiplications	5	0	0	0	0	0	0

Tableau (III.1) : Comparaison entre les transformées proposées et les autres transformées en terme de complexité

III.4. Structure des matrices proposées

Dans cette section, nous présentons les structures des deux transformées proposées en se basant sur les décompositions développées dans la section précédente.

III.4.1. Structure de $C_{proposée 1}$

Dans les deux figures (III.1) et (III.2), $x[i]$ représente l'entrée et $X[i]$ la sortie, où les flèches désignent une multiplication par (-1) avant l'addition. Ces graphes peuvent être utilisés dans l'implémentation de ces transformées sur les cartes DSP.

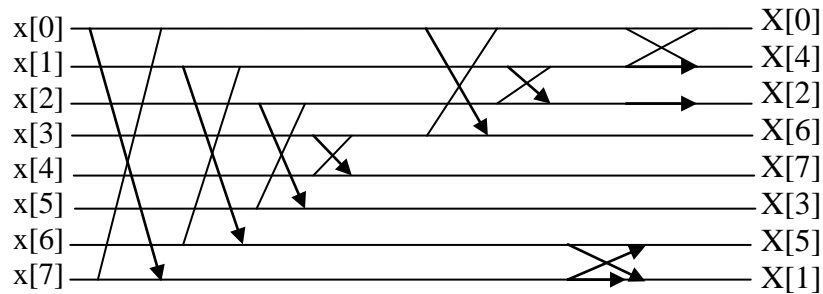


Figure (III.1) : Structure de la transformée proposée $C_{proposée 1}$

III.4.2. Structure de $C_{proposée 2}$

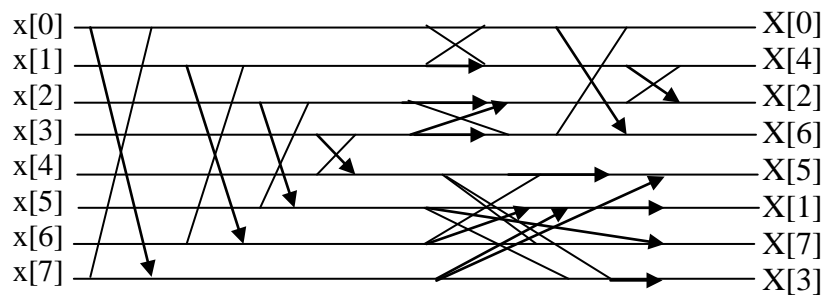


Figure (III.2) : Structure de la transformée proposée $C_{proposée 2}$

III.5. Conclusion

Dans ce chapitre, nous avons proposé deux transformées DCT entières, l'une basée sur la méthode décrit dans [12] et l'autre basée sur une méthode hybride. Les algorithmes et les structures de ces nouvelles transformées sont aussi développés. Nous avons montré que ces transformées exigent généralement des complexités réduites par rapport à celles des transformées existantes. L'étude des performances de ces transformées en compression d'images est donnée dans le chapitre (V). Le chapitre suivant est consacré à la modélisation des coefficients DCT entières proposées et existantes.

A purple square graphic with a slight gradient and a dark border, containing the text 'CHAPITRE IV' in a bold, black, sans-serif font.

**CHAPITRE
IV**

Modélisation des transformées DCT entières

IV.1. Introduction

Le taux de compression, pour une qualité de reconstruction d'image donnée, dépend directement de la matrice de quantification. On peut rappeler que la matrice proposée par la norme JPEG a été déterminée à partir des propriétés de la vision humaine, pour le plus grand nombre d'images [51]. Cette matrice de quantification proposée par JPEG [2] est propre à la DCT conventionnelle, et puisque les DCTs entières, comme [11] et [12], jouent un rôle très important dans le future des communications sans fil et par satellite aussi bien pour les ordinateurs portatifs, il est intéressant donc de développer des matrices de quantification appropriées pour chacune d'elles.

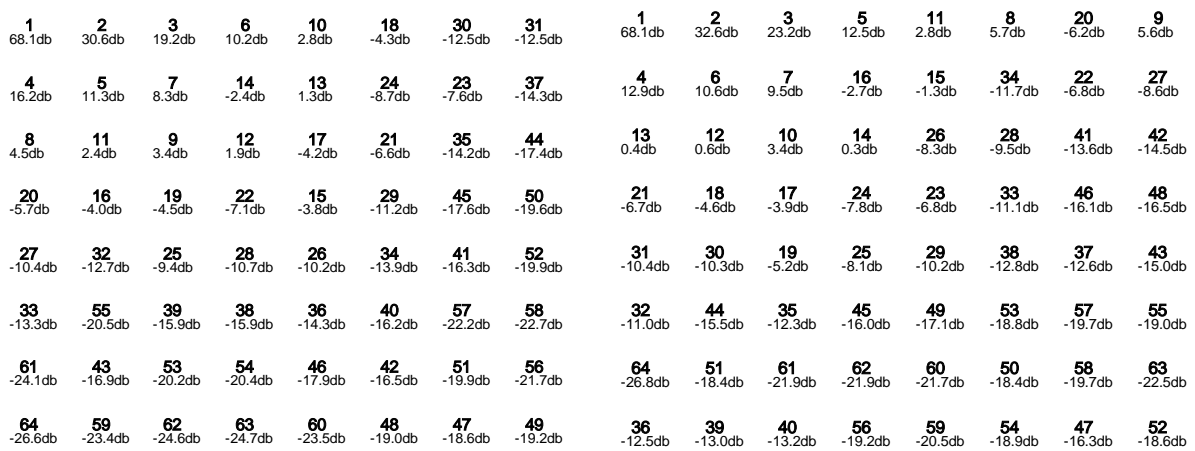
De ce fait, dans ce chapitre nous avons généré des matrices de quantification appropriées pour les transformées proposées dans le chapitre précédent, ainsi que celles proposées dans [11] et [12]. La procédure de génération est basée sur la distribution énergétique des coefficients pour chaque transformée entière donnée et ceci pour un ensemble d'images de test standards. Ensuite une méthode statistique est présentée pour modéliser la distribution des coefficients DCT entières, cette modélisation peut être utilisée pour générer des matrices de quantification optimales appropriées pour chaque transformée entière. Notre méthode est réalisée de façon extensive à l'aide d'une batterie d'images de test standards et des modèles de lois théoriques comme ceux de Cauchy, Laplace et Gauss. Nous montrons que ces coefficients peuvent être modélisés par une distribution de mélange de trois Gaussiennes.

IV.2. Distribution énergétique des coefficients de la DCT conventionnelle et de ses versions entières

La quantification non optimale provoque une dégradation d'image à sa reconstruction. La méthode décrite par JPEG a pour objectif de rendre nuls les coefficients de faible valeur qui ne représentent que peu d'information dans l'image, et de réduire la dynamique des autres. Les dégradations engendrées ne doivent pas être perceptibles à l'œil. Cependant, il est important de remarquer que tous les coefficients ne représentent pas la même quantité d'information. En effet, la réduction de la dynamique, voir la mise à zéro, d'un coefficient correspondant aux hautes fréquences (souvent assimilées au bruit) ne provoquera pas autant de dégradation que celle d'un coefficient correspondant aux basses fréquences. Donc l'énergie d'un coefficient joue un rôle très important pour définir le meilleur pas de quantification

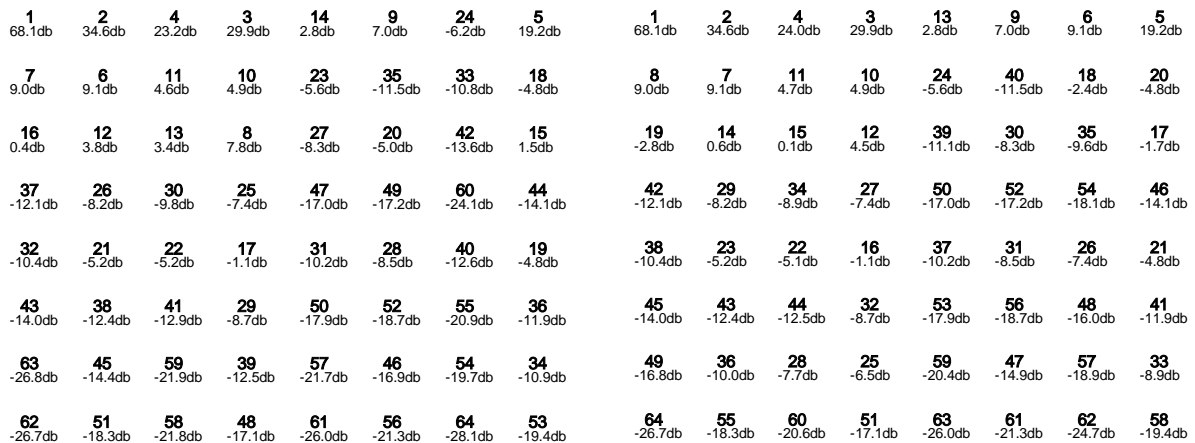
approprié pour chaque coefficient. A cet égard, nous avons proposé des matrices de quantification appropriées pour les transformées DCT entières de [11] et [12] ainsi pour nos transformées proposées dans le chapitre précédent. Le choix de ces transformées est dû au fait qu'elles présentent un bon compromis qualité de compression/complexité.

La figure (IV.1) présente un exemple des distributions énergétique de la DCT conventionnelle, celle des transformées de [11] et [12] ainsi de nos transformées, et ceci pour l'image 'Lena 256×256'.



(a) Distribution énergétique de la DCT conventionnelle

(b) Distribution énergétique de la DCT entière de [11]



(c) Distribution énergétique de la DCT entière de [12]

(d) Distribution énergétique de la DCT entière proposée T₁

1 68.1db	2 32.6db	3 17.8db	5 12.5db	14 2.8db	7 5.7db	10 4.4db	8 5.6db
4 12.9db	6 10.6db	9 4.7db	17 -2.7db	16 -1.3db	39 -11.7db	23 -7.2db	28 -8.6db
13 3.1db	12 3.4db	15 1.3db	11 3.5db	19 -5.4db	20 -6.3db	25 -7.7db	38 -11.3db
21 -6.7db	18 -4.6db	29 -9.0db	26 -7.8db	22 -6.8db	36 -11.1db	54 -16.9db	49 -16.5db
34 -10.4db	33 -10.3db	30 -9.2db	27 -8.1db	32 -10.2db	42 -12.8db	52 -16.7db	46 -15.0db
35 -11.0db	47 -15.5db	51 -16.5db	48 -16.0db	55 -17.1db	57 -18.8db	64 -23.3db	59 -19.0db
31 -10.2db	24 -7.6db	45 -14.4db	37 -11.2db	44 -13.6db	40 -12.3db	60 -19.2db	50 -16.5db
41 -12.5db	43 -13.0db	53 -16.9db	61 -19.2db	62 -20.5db	58 -18.9db	63 -20.9db	56 -18.6db

(e) Distribution énergétique de la DCT
entière proposée T_2

Figure (IV.1) : Distribution énergétique des coefficients des différentes transformées entières pour l'image « Lena 256×256 »

D'après la figure (IV.1), nous remarquons que la variation de la distribution énergétique pour chaque coefficient d'une transformée entière est différente à celle de la DCT conventionnelle. Cette variation diffère selon la position du coefficient d'une transformée à l'autre. En exploitant cette différence dans la distribution énergétique, nous avons proposé des matrices de quantification appropriées pour chacune de ces transformées entières.

IV.3. Matrices de quantification proposées

Dans cette section, nous allons proposer trois matrices de quantification Q_{p1} , Q_{p2} et Q_{p3} , appropriées, respectivement, pour les transformées entières de [11], [12] et pour nos transformées proposées dans le chapitre précédent. La confection de ces matrices de quantification est basée sur les distributions énergétiques des coefficients de chaque transformée calculées pour un grand nombre d'images standards de test de différentes propriétés statistiques. Alors, le pas de quantification doit être différent selon la position du coefficient dans le bloc: il sera d'autant plus fort que les coefficients seront de faible énergie, donc pauvres en information et vice versa. Par l'implémentation de cette stratégie, nous avons pu aboutir aux matrices de quantification données dans la figure (IV.2).

$$Q_{p1} = \begin{bmatrix} 20 & 16 & 17 & 28 & 22 & 36 & 36 & 31 \\ 18 & 17 & 18 & 19 & 24 & 40 & 23 & 40 \\ 18 & 17 & 18 & 24 & 37 & 50 & 50 & 54 \\ 17 & 18 & 23 & 19 & 45 & 73 & 73 & 58 \\ 18 & 21 & 37 & 56 & 67 & 92 & 101 & 103 \\ 14 & 34 & 45 & 64 & 81 & 104 & 120 & 92 \\ 45 & 50 & 62 & 79 & 103 & 70 & 70 & 101 \\ 41 & 41 & 74 & 59 & 70 & 90 & 100 & 99 \end{bmatrix}$$

(a) Pour la DCT entière de [11]

$$Q_{p2} = \begin{bmatrix} 19 & 16 & 17 & 28 & 22 & 36 & 36 & 31 \\ 18 & 17 & 18 & 19 & 24 & 40 & 23 & 40 \\ 18 & 17 & 16 & 24 & 37 & 50 & 50 & 54 \\ 17 & 18 & 23 & 19 & 30 & 70 & 70 & 58 \\ 18 & 21 & 37 & 54 & 67 & 85 & 101 & 103 \\ 24 & 30 & 44 & 64 & 81 & 104 & 120 & 92 \\ 45 & 50 & 62 & 79 & 103 & 70 & 70 & 101 \\ 41 & 39 & 72 & 59 & 70 & 90 & 100 & 99 \end{bmatrix}$$

(b) Pour la DCT entière de [12]

$$Q_{p3} = \begin{bmatrix} 20 & 15 & 17 & 18 & 22 & 36 & 36 & 31 \\ 18 & 17 & 18 & 19 & 24 & 40 & 23 & 40 \\ 18 & 17 & 18 & 24 & 37 & 50 & 50 & 54 \\ 17 & 18 & 23 & 19 & 45 & 73 & 73 & 58 \\ 18 & 21 & 37 & 56 & 67 & 92 & 101 & 103 \\ 24 & 34 & 45 & 64 & 81 & 104 & 120 & 92 \\ 45 & 50 & 62 & 79 & 103 & 70 & 70 & 101 \\ 41 & 41 & 74 & 59 & 70 & 90 & 100 & 99 \end{bmatrix}$$

(c) Pour nos transformées DCT entières proposées

Figure (IV.2) : Matrices de quantification proposées

Ces matrices de quantification portent une très bonne amélioration en terme du PSNR et de qualité visuelle de l'image reconstruite en les comparant avec la matrice conventionnelle adaptée par JPEG. Les détails de ces résultats seront présentés dans le chapitre suivant.

Malgré que ces nouvelles matrices de quantification présentent un avantage très intéressant par rapport à la matrice conventionnelle, elles restent toujours non optimales. Des matrices de quantification optimales et appropriées pour chaque transformées entières peuvent être déterminées en exploitant la modélisation statistique de la distribution des coefficients que nous allons proposer dans la section suivante.

IV.4. Modélisations statistiques proposées

Jusqu'à présent, la littérature donne beaucoup d'informations différentes sur la modélisation des distributions statistiques pour les coefficients de la DCT conventionnelle [53]-[57]. Dans [53], les auteurs utilisent la modélisation de la distribution gaussienne

généralisée (GGD). Dans [54], une démonstration mathématique, en utilisant le modèle stochastique double des images, a été développée en montrant que la distribution des coefficients de la DCT est Laplacienne. Ce résultat a été exploité dans [55] pour prédire les coefficients nuls après la quantification dans le JPEG. Les auteurs dans [56] ont abouti à des résultats différents en montrant que les coefficients de la DCT suivent un, deux ou trois Gaussiennes pour une batterie de tests d'adéquation plus le test de cramer. Les travaux de [57] et [58] ont abouti aux mêmes résultats que [56].

Ces modélisations diffèrent selon le type d'images étudiées et les approches utilisées. En effet, les coefficients DC (voir chapitre I) suivent une distribution Gaussienne, par contre les coefficients AC (voir chapitre I) peuvent être modélisés par des distributions différentes. D'autres travaux sont données dans [52], [59]-[64]. Cependant, il n'existe pas des modélisations pour le cas des coefficients des DCT entières.

Dans cette section, nous proposons une méthode statistique pour modéliser les coefficients d'une DCT entière donnée, dont le principe est illustré dans la figure (IV.3):

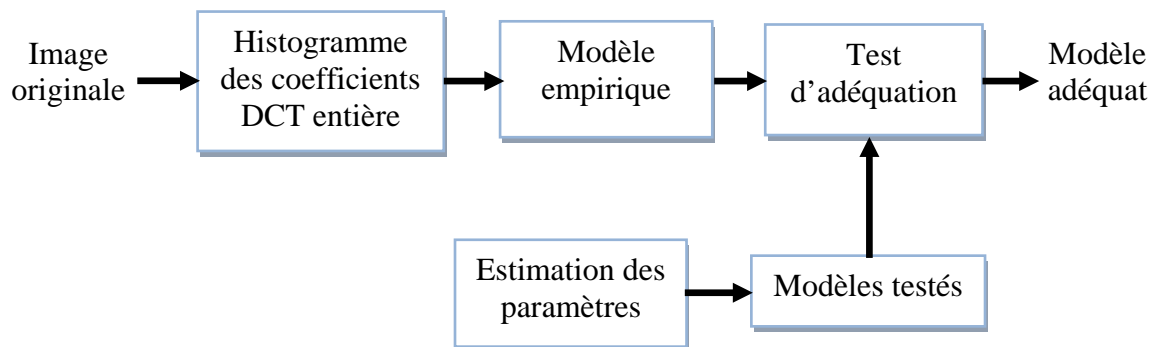


Figure (IV.3) : Principe de la modélisation statistique proposée

Ces étapes de la modélisation statistique sont détaillées dans les sous-sections suivantes :

IV.4.1. Histogramme des coefficients de la DCT entière

D'abord, on divise l'image entière en blocs de taille 8×8 , ensuite, on applique la transformée DCT entière à modéliser sur chacun de ces blocs, après on rassemble les coefficients qui sont dans la même position dans tous les blocs et enfin on génère les

histogrammes correspondants à chaque coefficient. Des exemples des histogrammes des coefficients DCT entière sont donnés dans la section des résultats expérimentaux.

IV.4.2. Modèle empirique

Le modèle empirique peut être représenté par une fonction de répartition empirique. Alors, l'étape suivante est de calculer la fonction de répartition empirique des coefficients pour chaque transformée considérée. Cette fonction est utilisée dans la phase de test d'adéquation, plus précisément dans le test de Kolmogorov-Smirnov (KS test), ce test permet la comparaison entre la distribution empirique et la distribution empirique proposée pour sa modélisation. Elle est calculée pour chaque échantillon généré $X=\{x_1, x_2, \dots, x_M\}$, avec l'ordre statique $x(n)$, $n=1, 2, \dots, M$, à partir de l'expression [58] :

$$\hat{F}_X(x) = \begin{cases} 0 & x < x_{(1)} \\ \frac{n}{M} & x_{(n)} \leq x \leq x_{(n+1)}, \quad n = 1, 2, \dots, M-1 \\ 1 & x \geq x_{(M)} \end{cases} \quad (\text{IV.1})$$

IV.4.3. Modèles testés

Les distributions de Laplace, Cauchy et de Gauss sont généralement les plus utilisées pour des distributions de modélisation des coefficients de la DCT conventionnelle. Par conséquent, elles ont été choisies pour cette étude.

- La fonction de la densité de probabilité de la loi Laplace $L(m, \beta)$ est définie par [55] :

$$f_X(x) = \frac{1}{2\beta} \exp\left(-\frac{|x-m|}{\beta}\right) \quad (\text{IV.2})$$

m est la moyenne et β est un paramètre

- La fonction de la densité de probabilité de la loi Cauchy $C(\alpha, \beta)$ est définie par [55] :

$$f_X(x) = \frac{1}{\pi} \frac{\beta}{\beta^2 + (x-\alpha)^2} \quad \text{avec } -\infty < \alpha < \infty \quad \text{et} \quad -\infty < \beta < \infty \quad (\text{IV.3})$$

α et β sont des paramètres.

- La fonction de la densité de probabilité de la loi Gauss $N(m, \sigma)$ est définie par [55] :

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) \quad (\text{IV.4})$$

m : la moyenne, σ : l'écart type

Des récents travaux [55], [56] et [57] ont montré que la distribution des coefficients DCT des images peut être modélisée par des lois de mélange de Gaussiennes [56]. Il est bien évident qu'une loi mélange possède plus de degrés de liberté qu'une loi simple et qu'elle s'ajustera mieux aux échantillons.

La fonction de la densité de probabilité de la loi mélange de Gaussiennes est définie d'après [56] par:

$$f_X(\sigma_i, p_i, m_i, x) = \sum_{i=1}^k \frac{p_i}{\sigma_i\sqrt{2\pi}} \exp\left(-\frac{(x-m_i)^2}{2\sigma_i^2}\right) \quad (\text{IV.5})$$

avec m_i : la moyenne, σ_i : l'écart type et p_i probabilité à priori de la $i^{\text{ème}}$ Gaussienne.

IV.4.4. Estimation des paramètres des distributions

Afin d'estimer les différents paramètres des distributions des coefficients transformés (moyenne, écart type et probabilité), nous avons utilisé l'algorithme EM (Estimation Maximisation) stochastique [65]. Cet algorithme itératif calcule les paramètres au sens du maximum de vraisemblance.

Exemple : Estimation des paramètres de la loi de gauss (la moyenne et la variance)

La vraisemblance pour une loi de gauss est donnée par :

$$V = \prod_{n=1}^M f_X(x_n) = \prod_{n=1}^M \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_n - m)^2}{2\sigma^2}\right) \quad (\text{IV.6})$$

où x_n sont les coefficients DCT entières.

m, σ^2 sont inconnues et leurs estimations sont les valeurs qui rendent la vraisemblance maximale. Pour trouver le max de V , on procède comme suit :

$$\ln(V) = n \times \ln\left(\frac{1}{\sqrt{2\pi}}\right) + \frac{n}{2} \ln \sigma^2 + \sum_{n=1}^M \left\{ -\frac{(x_n - m)^2}{2\sigma^2} \right\} \quad (\text{IV.7})$$

Comme cette expression dépend de deux paramètres m et σ^2 , il faut annuler les dérivés par rapport à m et σ^2 pour trouver le maximum, alors la moyenne estimée m_{es} et la variance estimée σ_{es}^2 sont données, respectivement par :

$$m_{es} = \frac{\sum_{n=1}^M x_n}{M} \quad (\text{IV.8})$$

$$\sigma_{es}^2 = \frac{\sum_{n=1}^M (x_n - m)^2}{M} \quad (\text{IV.9})$$

Noter bien que la même méthode est appliquée pour les autres modèles de tests.

IV.4.5. Tests d'adéquation

Après avoir estimé les paramètres des différentes distributions théoriques, il est possible de calculer leurs fonctions de répartition pour l'utilisée dans des tests appelés tests d'adéquation ou d'ajustement de lois et ceci pour déterminer la meilleure loi théorique associée à l'échantillon considéré [55].

Ainsi, pour vérifier si un échantillon de $\hat{F}_X(x)$ provient d'une distribution de fonction de répartition connue $F_X(x)$, on pose les hypothèses suivantes :

$H_0 : \hat{F}_X(x) = F_X(x) \Rightarrow$ la loi empirique \hat{F} est assimilable à la loi théorique F .

$H_1 : \hat{F}_X(x) \neq F_X(x) \Rightarrow$ les deux lois ne sont pas comparables.

Plusieurs tests d'adéquation sont présentés dans la littérature [65], le test de Kolmogorov-Smirnov est le premier test d'adéquation utilisé et il fait la base de plusieurs tests non paramétriques. En effet, la majorité des résultats des études existantes sont basés sur ce test d'adéquation uniquement. Les décisions sur les lois étaient prises en considérant que la

loi théorique suivie par le coefficient DCT est celle qui donne le résultat le plus faible à ce test.

Afin d'obtenir des résultats plus fiables, nous avons utilisé deux tests d'ajustement de lois. Le choix de la loi est alors réalisé en considérant qu'une loi empirique suit une loi théorique donnée si les résultats aux tests sont inférieurs à des seuils. Les détails de ces deux tests sont données dans les deux sous-sections suivantes.

IV.4.5.1. Test de Kolmogorov-Smirnov

Ce test compare la fonction de répartition empirique (CDF : Cumulative Distribution Function) avec la fonction de répartition théorique du modèle donné [58], [65] :

$$K_n = \max_{i=1,2,\dots,M} \left| F_X(x_{(i)}) - \hat{F}_X(x_{(i)}) \right| \quad (\text{IV.10})$$

On peut voir que la statistique de KS est une mesure de la distance entre le CDF empirique et le CDF de modèle, il donne pour résultat la plus grande différence existante entre les deux. Si le CDF empirique est adapté pour plusieurs CDFs théoriques, le modèle qui donne la statistique minimum du KS peut être pris pour être le bon modèle adapté pour les coefficients de la transformée considérée.

IV.4.5.2. Test de chi2

Comme le test de KS, le test χ^2 (chi2) compare les fonctions de densité de probabilité. La gamme des données est divisée à k cases A_i disjoint et exhaustive, $i=1,2,\dots,k$.

Soit $E_i = M \times p_i$ la fréquence prévue dans A_i avec $p_i = P(x \in A_i)$, M est le nombre totale de donnés et O_i la fréquence observée dans A_i . Alors, la statistique χ^2 est définie par [58], [65]:

$$V_k = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i} \quad (\text{IV.11})$$

Puisque c'est une mesure de déviation des fréquences empiriques et les fréquences prévues, la fonction de densité de probabilité du modèle qui donne la statistique minimum peut être considérée en tant que le meilleur modèle.

Il est important de préciser que l'utilisation de ces tests d'adéquation doit être réalisée sur des variables aléatoires indépendantes. Une décorrélation complète serait obtenue en utilisant la transformée de Karhunen-Loeve, cependant, l'utilisation de cette transformée est limitée par le fait qu'elle demande une quantité de calculs très importante. On ne peut alors qu'utiliser des transformées orthogonales sous-optimales comme la DCT sur des blocs de faible dimension. Les coefficients obtenus par la DCT sont donc partiellement décorrélés, la corrélation restante n'est pas exploitée. Ainsi, l'hypothèse de base utilisée est que les coefficients suivent des lois statistiques indépendantes. Une telle hypothèse est couramment employée dans la littérature discutant la compression d'image par la DCT [55].

IV.5. Résultats expérimentaux

Dans cette section, nous allons donner les résultats expérimentaux de cette modélisation, pour cela, nous avons considéré des images de test standards naturelles ayant différentes caractéristiques statistiques. Le tableau (IV.1) donne les tailles de ces images de test qui sont représentées sur la figure (IV.4).

Images de test	Taille
Peppers (1)	512×512
Boat (2)	512×512
Bridge (3)	512×512
Goldhill (4)	512×512
Lena (5)	512×512
Baboon (6)	512×512

Tableau (IV.1) : Tailles des images de test standards utilisées

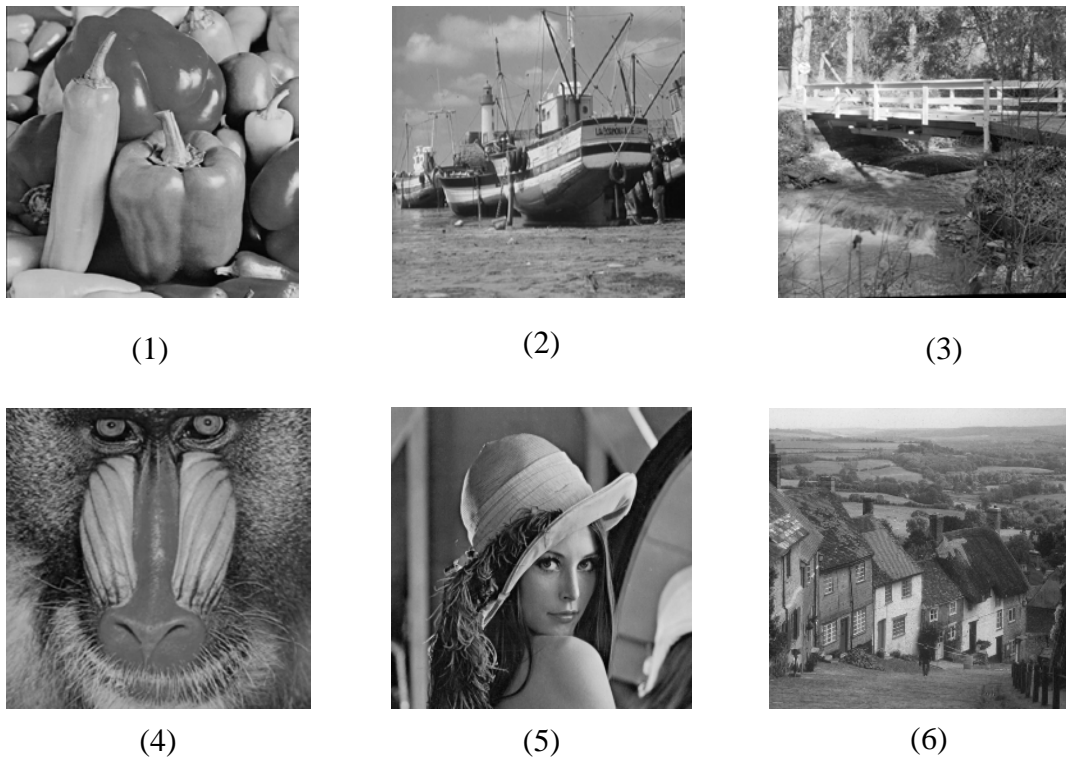


Figure (IV.4) : Images de test standards utilisées

IV.5.1. Modélisation des coefficients de la transformée entière de [11]

La figure (IV.5) montre un exemple des histogrammes des coefficients de la DCT entière de [11]. L'image utilisée est l'image de « Bridge 512×512 ». Le coefficient gauche supérieur est le coefficient DC tandis que les autres sont des coefficients AC.

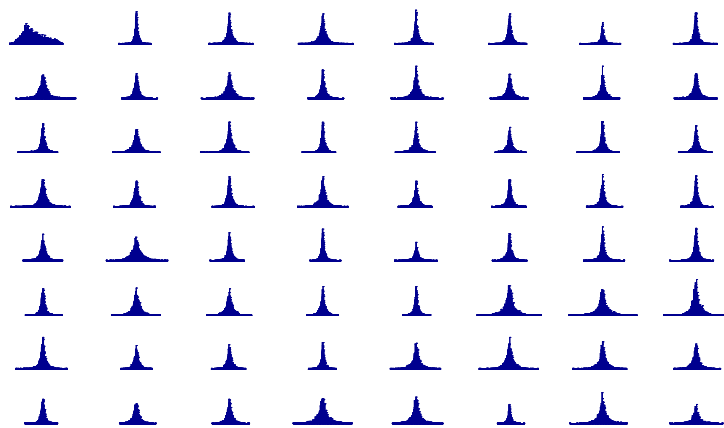


Figure (IV.5) : Histogrammes des coefficients DCT entière de [11] pour l'image « Bridge 512×512 »

Ainsi, pour chaque ensemble des coefficients ayant la même position dans les blocs d'une image, nous avons appliqué l'algorithme de vraisemblance afin d'estimer les paramètres des trois lois possibles: Cauchy, Laplace et Gauss. Ensuite, les valeurs des tests de KS et de χ^2 sont calculées pour les coefficients pris en considération et pour les modèles de distributions proposés dans la section (IV.4.3). La distribution qui nous donne la valeur minimum de KS a été choisie l'un des meilleurs modèles convenables pour modéliser les coefficients DCT entières sous le critère de KS et pareillement pour la valeur de χ^2 . Les valeurs de test de KS et de χ^2 pour les 15 premiers coefficients sont données dans les deux tableaux (IV.2) et (IV.3) et ceci pour la matrice données dans [11] pour les trois images (1), (2) et (3).

Le signe (-) signifie que le coefficient correspondant ne peut pas être modélisé par la distribution considérée.

Position du Coeff.	Image (1)			Image (2)			Image (3)		
	Gauss	Laplace	Cauchy	Gauss	Laplace	Cauchy	Gauss	Laplace	Cauchy
0,0	-	-	-	-	-	-	-	-	-
0,1	-	-	-	-	-	-	-	-	-
1,0	-	-	0.0214	-	-	0.0267	-	0.0243	-
2,0	-	-	0.0208	-	-	-	-	-	-
1,1	-	-	0.0252	-	-	0.0303	-	0.0166	-
0,2	-	-	-	-	-	-	-	-	-
0,3	-	-	-	-	-	-	-	-	-
1,2	-	-	0.0180	-	-	0.0179	-	0.0152	-
2,1	-	-	0.0270	-	-	0.0231	-	0.0137	-
3,0	-	-	-	-	-	0.0289	-	0.0313	-
4,0	-	-	-	-	-	0.0273	-	0.0278	-
3,1	-	-	-	-	-	-	-	0.0238	-
2,2	-	-	0.0274	-	-	0.0229	-	0.0170	-
1,3	-	-	-	-	-	0.0194	-	0.0267	-
0,4	-	-	-	-	-	-	-	-	-

Tableau (IV.2) : Valeurs statistiques de test KS pour les 15 premiers coefficients en ordre de zigzag pour la transformée de [11] pour les lois de Cauchy, Laplace et Gauss

Si on prend par exemple le cas de l'image (1) dans le tableau (VI.2) (test de KS), les 15 premiers coefficients (ré-ordonnés en zigzag) ne peuvent pas être modélisés par la loi de Gauss et la loi de Laplace, seulement 6 coefficients parmi les 15 peuvent être modélisés par une loi

de Cauchy. Si un coefficient avait plusieurs possibilités de modélisation (il peut être modélisé par 2 lois de distribution) la valeur minimale du test est considérée comme la meilleure modélisation pour ce coefficient, dans ce qui suit, elle a été indiquée en gras.

Position du Coeff.	Image (1)			Image (2)			Image (3)		
	Gauss	Laplace	Cauchy	Gauss	Laplace	Cauchy	Gauss	Laplace	Cauchy
0,0	-	-	-	-	-	-	-	-	-
0,1	-	-	-	-	-	-	-	-	-
1,0	-	-	-	-	-	-	-	-	-
2,0	-	-	-	-	-	-	-	-	-
1,1	-	-	-	-	-	-	-	3.0156	-
0,2	-	-	-	-	-	-	-	-	-
0,3	-	-	-	-	-	-	-	-	-
1,2	-	-	-	-	-	-	-	4.0305	-
2,1	-	-	-	-	-	-	-	1.7775	-
3,0	-	-	-	-	-	-	-	-	-
4,0	-	63.7	-	-	-	-	-	-	-
3,1	-	-	-	-	-	-	-	5.7374	-
2,2	-	-	-	-	-	-	-	4.4121	-
1,3	-	-	-	-	-	-	-	-	-
0,4	-	-	-	-	-	-	-	-	-

Tableau (IV.3) : Valeurs statistiques de test χ^2 pour les 15 premiers coefficients en ordre de zigzag pour la transformées de [11] de Cauchy, Laplace et Gauss

Maintenant, pour le cas de la transformée de [11], d'après les deux tableaux (VI.2) et (VI.3), nous pouvons conclure que les coefficients de la matrice DCT entière dans [11] ne peuvent pas être modélisés ni par la loi de Cauchy ni par la loi de Laplace et ni par une seule loi de Gauss. Les statistiques totales consolidées pour chaque image considérée, et le résultat totale de la somme des 3 images considérées, de test KS et de test χ^2 pour la transformée de [11] pour les trois lois considérés : Cauchy, Laplace et Gauss sont résumés dans les deux tableaux suivants :

Image de test	AC			DC		
	Gauss	Laplace	Cauchy	Gauss	Laplace	Cauchy
(1)	06	01	07	00	00	00
(2)	09	00	11	00	00	00
(3)	00	14	00	01	00	00
Toutes les images	15	15	18	01	00	00

Tableau (IV.4) : Résultat total de KS test pour la transformée de [11] de Cauchy, Laplace et Gauss

Image de test	AC			DC		
	Gauss	Laplace	Cauchy	Gauss	Laplace	Cauchy
(1)	01	02	00	00	00	00
(2)	00	01	00	00	00	00
(3)	00	20	00	00	00	00
Toutes les images	01	23	00	00	00	00

Tableau (IV.5) : Résultat total de χ^2 test pour la transformée de [11] de Cauchy, Laplace et Gauss

Ayant remarqué que les coefficients DCT ne suivaient ni la loi de Laplace, ni la loi de Cauchy, nous nous sommes focalisés le reste de cette étude, pour les autres transformées, sur la loi mélange de Gaussiennes. Ainsi, pour chaque coefficient, nous avons appliqué l'algorithme EM afin d'estimer les paramètres des trois lois possibles : une, deux ou trois Gaussiennes notées par (1G, 2G, 3G) respectivement, (nous nous sommes limités à trois Gaussiennes afin d'avoir moins de calculs à réaliser). Puis, pour chacune de ces lois, nous avons calculé les valeurs des tests d'adéquation de KS et de χ^2 . Dans ce qui suit les tableaux montrent les résultats expérimentaux pour cette étude pour quelques images de test considérées.

Position de Coeff.	Image (3)			Image (4)			Image (5)			Image (6)		
	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G
0,0	-	0.0275	0.0241	0.0271	-	0.0216	-	-	0.0257	-	-	-
0,1	-	0.0171	0.0172	-	0.0193	0.0276	-	0.0280	-	-	-	-
1,0	-	-	-	-	-	-	-	-	-	-	-	-
2,0	-	-	-	-	-	-	-	-	-	-	-	-
1,1	-	0.0135	0.0149	-	0.0197	0.0168	-	-	0.0153	-	-	-
0,2	-	-	0.0211	-	0.0186	0.0143	-	0.0226	0.0139	0.0284	0.0219	0.0215
0,3	-	0.0196	0.0211	-	0.0227	0.0200	-	0.0256	0.0219	-	0.0362	-
1,2	-	0.0128	0.0167	-	0.0194	0.0294	-	0.0303	0.0191	0.0293	0.0295	0.0286
2,1	-	0.0196	0.0303	-	0.0299	-	-	0.0305	0.0123	0.0317	0.0289	-
3,0	-	-	-	-	-	-	-	-	-	-	-	-
4,0	-	-	-	-	-	-	-	-	-	-	-	-
3,1	-	-	-	-	0.0223	0.0301	-	-	0.0219	-	-	-
2,2	-	0.0191	0.0301	-	0.0128	0.0251	-	0.0242	0.0114	0.0273	0.0236	0.244
1,3	-	0.0246	0.0250	-	0.0237	0.0219	-	-	-	-	-	-
0,4	-	0.0196	0.0280	-	0.0237	0.0272	-	-	-	0.0399	0.0316	0.0393

Tableau (IV.6) : Valeurs statistiques de test KS pour les 15 premiers coefficients en ordre de zigzag pour la transformée de [11]

Position du Coeff.	Image (3)			Image (4)			Image (5)			Image (6)		
	1G	2 G	3G	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G
0,0	-	-	28.4648	-	-	8.8294	-	-	113.3197	-	-	1.8412
0,1	-	-	23.5533	-	-	34.0098	-	86.5544	25.5597	-	-	12.4261
1,0	-	17.7438	5.6068	-	-	5.0599	-	-	15.1113	-	4.5338	4.2226
2,0	-	-	29.1189	-	-	29.2534	-	56.4309	21.3203	-	-	2.9812
1,1	-	-	16.8919	-	-	6.8604	-	65.0969	16.4646	11.555	-	9.9693
0,2	-	-	7.9189	-	-	12.1186	-	33.9889	6.7728	-	2.6719	1.3139
0,3	-	-	9.1534	-	-	27.6061	-	49.6371	3.2467	-	-	2.6967
1,2	-	-	17.0958	-	-	18.6976	-	26.2301	4.2264	10.7751	-	3.0191
2,1	-	17.5589	13.5087	-	-	65.1307	-	43.7826	16.1461	-	-	8.7991
3,0	-	18.8314	5.4813	-	-	18.5968	-	53.2861	3.1850	-	-	3.7992
4,0	-	19.8153	14.6234	-	-	32.5330	-	68.6578	5.1455	-	-	5.1111
3,1	-	-	86.6108	-	24.6555	25.8682	-	49.5729	10.4448	10.1121	-	7.1915
2,2	-	6.9746	15.1406	-	28.0449	12.7120	-	60.1991	11.0309	7.6944	4.9101	2.0837
1,3	-	12.1208	4.8698	-	-	1.5578	-	14.5274	14.5020	-	-	5.2538
0,4	-	-	26.6161	-	14.1743	2.4818	-	10.1044	1.4180	-	-	3.9274

Tableau (IV.7) : Valeurs statistiques de test χ^2 pour les 15 premiers coefficients en ordre de zigzag pour la transformée de [11]

D'après les deux tableaux (IV.6) et (IV.7), il apparaît que pour toutes les images considérées, les coefficients peuvent être modélisés par une loi somme de 2 et 3 Gaussiennes, mais avec des valeurs de tests d'adéquation différentes. La loi qui présente des faibles valeurs de tests d'adéquation est considérée comme la meilleure modélisation adéquate pour la transformée. Alors, d'après les résultats dans les deux tableaux (IV.6) et (IV.7), malgré que pour certains coefficients la distribution d'une somme de trois Gaussiennes n'est pas la bonne modélisation, elle reste la meilleur pour la majorité des coefficients. Il apparaît alors que les coefficients de la transformée de [11] ne suivent ni la loi de Cauchy, ni la loi de Laplace, mais des lois de mélange de 3 Gaussiennes. Ces résultats sont confirmés en comparant les densités de probabilité des coefficients avec celles des modèles de test proposés (Figure (IV.6)).

Il est clair, d'après la figure (IV.6), que la loi de mélange de deux et trois Gaussiennes (Figures (IV.6(d)) et (IV.6(e))) sont les plus adaptées pour la modélisation du coefficient (2,1) que les autres modèles proposés.

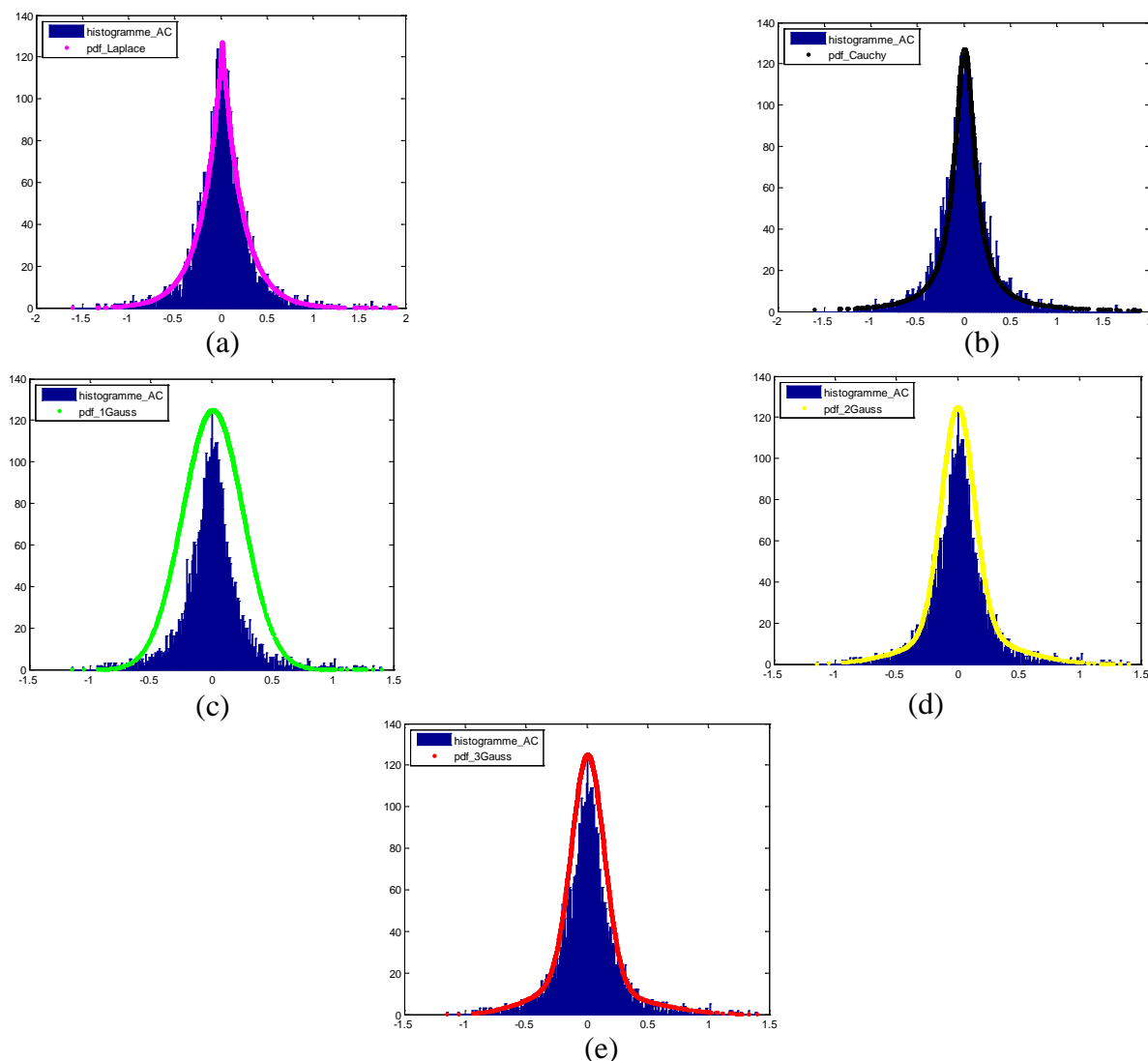


Figure (IV.6) : Approximation de la loi empirique du coefficient (2,1) de la transformée de [11] de l'image « Bridge 512×512 » par les différentes lois théoriques

Ces résultats sont confirmés dans les deux tableaux (IV.6) et (IV.7), si en se basant sur le test de KS, la loi de mélange de deux Gaussiennes figure la plus adaptée pour la modélisation de ce coefficient et si en se basant sur le test de χ^2 , la loi de mélange de trois Gaussiennes présente la meilleure modélisation, alors nous allons prendre le cas de mélange de deux Gaussiennes comme la loi la plus adaptée et ceci dû à sa faible complexité.

Nous avons fait la même étude pour les autres transformées entières et nous donnons dans les sous-sections suivantes seulement les résumés des résultats correspondants.

IV.5.2. Modélisation des coefficients de la transformée de [12]

La figure (IV.7) montre un exemple des histogrammes des coefficients de la DCT entière donnée dans [12]. L'image utilisée est l'image de « Bridge 512×512 ».

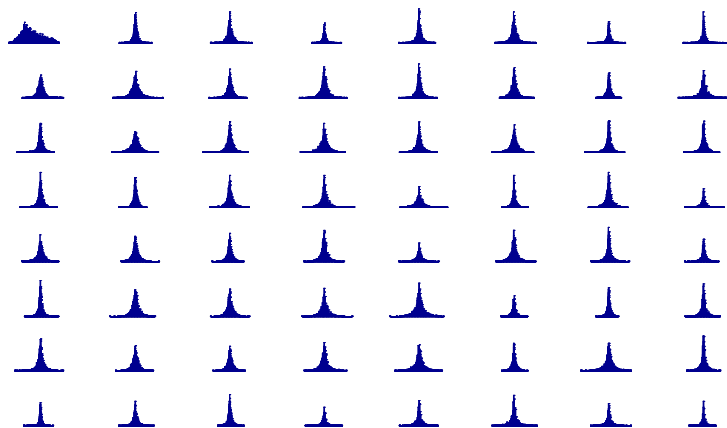


Figure (IV.7) : Histogrammes des coefficients DCT entière de [12] pour l'image « Bridge 512×512 »

La figure (IV.7) montre que les histogrammes de la transformée de [12] sont légèrement différents à ceux de la transformée de [11], ceci se conduit à une même modélisation.

Les résultats des tests de KS et chi2 pour les 15 premiers coefficients sont données dans les tableaux (IV.8) et (IV.9) pour 4 images de test considérées.

Position du Coeff.	Image (3)			Image (4)			Image (5)			Image (6)		
	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G
0,0	-	0.0275	0.0249	0.0276	-	0.0217	-	-	0.0279	-	-	-
0,1	-	0.0239	0.0226	-	0.0257	-	-	-	-	-	-	-
1,0	-	-	-	-	-	-	-	-	-	-	-	-
2,0	-	-	-	-	-	-	-	-	-	-	-	-
1,1	-	-	-	-	-	-	-	-	-	-	-	-
0,2	-	0.0259	0.0232	-	0.0492	0.0110	-	0.0271	0.0188	-	0.0284	0.0275
0,3	-	-	-	-	-	-	-	-	-	-	-	-
1,2	-	-	-	-	-	0.0260	-	-	-	-	-	-
2,1	-	0.0209	-	-	-	0.0278	-	-	0.0232	-	-	-
3,0	-	-	-	-	-	-	-	-	-	-	-	-
4,0	-	-	-	-	-	-	-	-	-	-	-	-
3,1	-	-	-	-	-	-	-	-	-	-	-	-
2,2	-	0.0179	0.0301	-	0.0126	0.0244	-	0.0275	0.0140	0.0236	0.0211	0.0208
1,3	-	-	-	-	-	-	-	-	-	-	-	-
0,4	-	0.0242	-	-	-	-	-	-	-	-	-	0.0388

Tableau (IV.8) : Valeurs statistiques de test KS pour les 15 premiers coefficients en ordre de zigzag pour la transformée de [12]

Position du Coeff.	Image (3)			Image (4)			Image (5)			Image (6)		
	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G
0,0	-	-	28.4648	-	-	8.8294	-	-	113.3197	-	-	1.8068
0,1	-	-	13.4284	-	-	38.5251	-	74.9050	45.4526	-	4.0295	2.6561
1,0	-	-	32.2169	-	-	6.2705	-	-	9.4829	-	3.4372	1.9971
2,0	-	-	27.9881	-	-	29.2534	-	56.4309	21.3203	-	-	2.9014
1,1	-	-	5.6834	-	-	24.7212	-	39.3856	21.0471	-	-	9.0114
0,2	-	-	7.9189	-	-	12.1186	-	33.9889	6.7728	-	3.1927	1.8120
0,3	-	-	16.3607	-	-	17.0342	-	-	2.5545	-	-	7.0668
1,2	-	-	20.8920	-	-	6.8002	-	23.8861	6.1580	-	-	0.9654
2,1	-	18.5875	11.6704	-	-	4.3713	-	45.9980	5.0657	-	15.5384	2.9340
3,0	-	21.7801	33.3365	-	-	10.6571	-	63.8901	4.3501	-	-	5.1256
4,0	-	-	14.6234	-	-	32.5334	-	69.0825	5.0961	-	-	4.8204
3,1	-	6.9764	13.5523	-	-	30.5385	-	86.4758	3.5167	8.8802	-	6.6778
2,2	-	-	15.1406	-	28.0440	12.7120	-	60.1001	11.0309	7.6274	4.9523	2.0613
1,3	-	-	46.7127	-	-	7.7296	-	-	18.2104	-	-	27.4104
0,4	-	-	26.6161	-	13.9906	2.4189	-	11.8094	1.0417	-	-	3.9274

Tableau (IV.9) : Valeurs statistiques de test χ^2 pour les 15 premiers coefficients en ordre de zigzag pour la transformée de [12]

Dans le cas de la transformée de [12], le test de KS ne nous donne pas des bons résultats mais en se basant sur le test de chi2, nous voyons que la loi de mélange de trois Gaussiennes est adaptée pour toutes types d'images considérées et pour tous les coefficients considérés.

Un exemple de comparaison entre la densité de probabilité empirique et théorique pour le coefficient (2,1) est représenté sur la figure (IV.8).

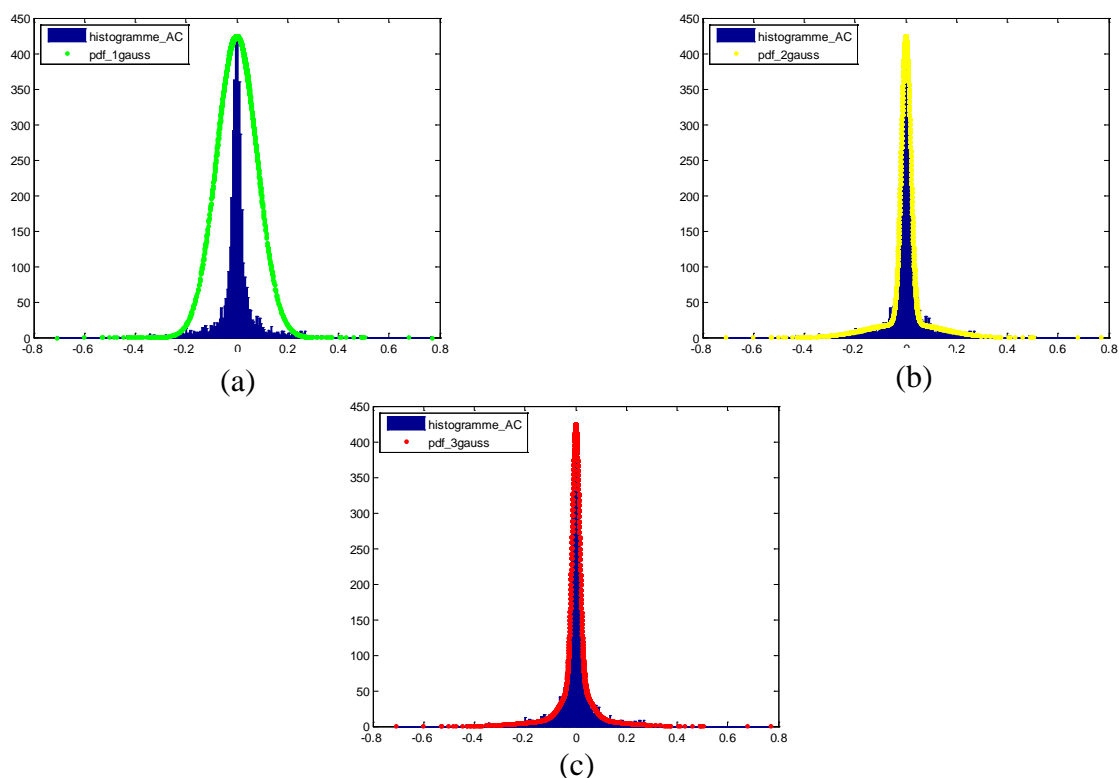


Figure (IV.8) : Approximation de la loi empirique du coefficient (2,1) de la transformée de [12] de l'image « Lena 512×512 » par différentes lois théoriques

D'après la figure (IV.8), il est clair que la loi de mélange de trois Gaussiennes est la distribution la plus adaptée pour modéliser la distribution du coefficient (2,1) de l'image (Lena 512×512), et ceci peut être vérifié dans les deux tableaux (IV.8) et (IV.9).

IV.5.3. Modélisation des coefficients de la transformée entière proposée T_1

La figure (IV.9) montre un exemple des histogrammes des coefficients de la DCT entière T_1 proposée dans le chapitre précédent. L'image utilisée est l'image de « Bridge 512×512 ».

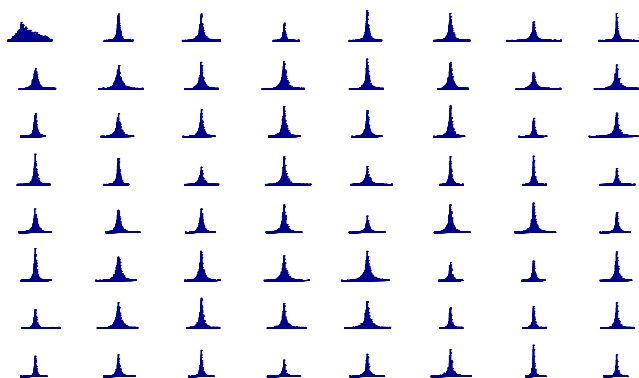


Figure (IV.9) : Histogrammes des coefficients DCT entière proposée T_1 pour l'image «Bridge 512×512 »

Les résultats expérimentaux pour les deux tests sont donnés dans les deux tableaux suivantes et ce ci pour les 15 premiers coefficients ordrés en zigzag pour les 4 images de test (3), (4), (5) et (6).

Position du Coeff.	Image (3)			Image (4)			Image (5)			Image (6)		
	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G
0,0	-	0.0285	0.0250	0.0249	-	0.0191	-	-	0.0271	-	-	-
0,1	-	-	0.0307	-	0.0275	-	-	-	-	-	-	-
1,0	-	-	-	-	-	-	-	-	-	-	-	-
2,0	-	-	-	-	-	-	-	-	-	-	-	-
1,1	-	-	-	-	-	-	-	-	-	-	-	-
0,2	-	-	-	-	-	-	-	-	-	-	-	-
0,3	-	-	-	-	-	-	-	-	-	-	-	-
1,2	-	-	-	-	-	-	-	-	-	-	-	-
2,1	-	-	-	-	-	-	-	-	-	-	-	-
3,0	-	-	-	-	-	-	-	-	-	-	-	-
4,0	-	-	-	-	-	-	-	-	-	-	-	-
3,1	-	-	-	-	-	-	-	-	-	-	-	-
2,2	-	-	-	-	-	-	-	-	-	-	-	-
1,3	-	-	-	-	-	-	-	-	-	-	-	-
0,4	-	-	-	-	-	-	-	-	-	-	-	-

Tableau (IV.10) : Valeurs statistiques de test KS pour les 15 premiers coefficients en ordre de zigzag pour la transformée proposée T_1

D'après ce tableau, nous voyons que les trois lois proposées ne présentent pas une bonne modélisation, et ceci en se basant sur le test d'adéquation KS. Des différents résultats sont obtenus en utilisant le test d'adéquation Chi2. Ces résultats sont résumée dans le tableau (IV.11).

Position du Coeff.	Image (3)			Image (4)			Image (5)			Image (6)		
	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G
0,0	-	-	28.4648	-	-	8.8294	-	-	113.3197	-	-	1.8412
0,1	-	-	3.4284	-	-	38.9616	-	74.9050	45.4526	-	-	2.6561
1,0	-	-	32.2169	-	-	6.2705	-	-	9.4829	-	3.4372	1.9971
2,0	-	-	27.1833	-	-	24.9430	-	66.1826	19.3422	8.5072	-	5.1290
1,1	-	-	5.6834	-	-	24.7212	-	39.3856	21.0471	-	-	9.0114
0,2	-	-	17.1518	-	-	18.9359	-	34.5395	10.4618	-	-	3.5815
0,3	-	-	16.3607	-	-	17.0342	-	64.7360	2.5545	-	-	6.7748
1,2	-	-	7.9108	-	-	3.5136	-	28.9299	8.5418	12.8983	-	5.8475
2,1	-	-	27.4339	-	11.1656	7.8787	-	42.4818	7.0668	10.4291	-	5.7985
3,0	-	21.7801	33.3365	-	-	10.6571	-	63.8901	4.3501	-	-	5.1256
4,0	-	19.8153	14.6234	-	-	32.5334	-	69.0825	5.0961	-	-	5.9150
3,1	-	18.2560	13.5523	-	-	30.5385	-	86.4758	3.5167	-	-	6.6778
2,2	-	-	20.6033	-	-	9.5014	-	-	8.1700	8.8890	-	8.4193
1,3	-	-	46.7127	-	-	7.7296	-	51.7802	18.2104	-	-	27.4104
0,4	-	-	26.6161	-	13.9906	2.4189	-	-	1.0417	-	-	3.9274

Tableau (IV.11) : Valeurs statistiques de test χ^2 pour les 15 premiers coefficients en ordre de zigzag pour la transformée proposée T_1

La figure (IV.10) montre un exemple de comparaison entre l'histogramme du coefficient (2,1) de la DCT entière proposée T_1 dans le chapitre précédent appliquée sur l'image de « Lena 512×512 » et les densités de probabilité des modèles de test utilisés.

On remarque d'après cette figure que la loi de somme de trois Gaussiennes présente la meilleure approximation de la distribution empirique du coefficient (2,1) de la transformée proposée T_1 . Ce résultat peut être confirmé en comparant les statistiques des tests de KS et de Chi2 des tableaux (IV.10) et (IV.12) pour ce coefficient de l'image «Lena 512×512». Dont il apparaît que la statistique de Chi2 pour la modélisation par une somme de trois Gaussiennes est inférieure à celle d'une somme de deux Gaussiennes.

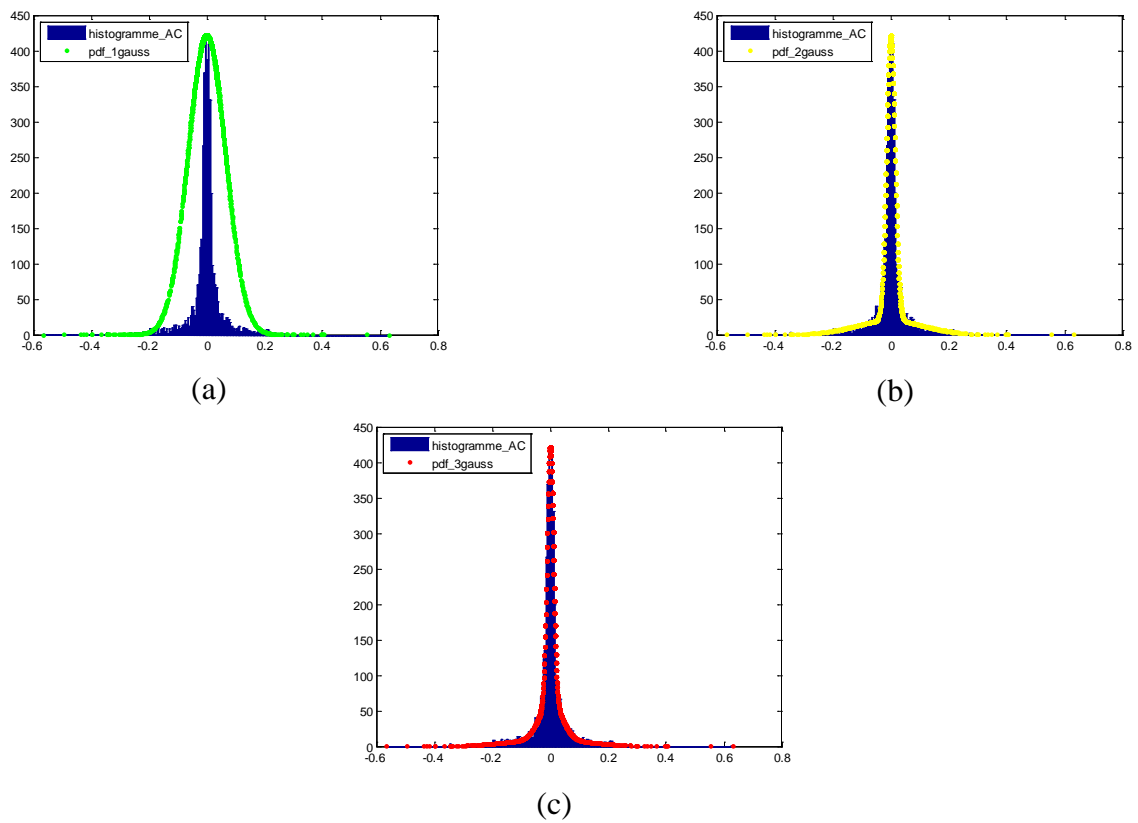


Figure (IV.10) : Approximation de la loi empirique du coefficient (2,1) de la transformée proposée T_1 de l'image « Lena 512×512 » par différentes lois théoriques

IV.5.4. Modélisation des coefficients de la transformée entière proposée T_2

Comme pour tous les autres transformées nous avons présenté les histogrammes des coefficients de la transformée DCT proposée, T_2 . Ils sont illustrés sur la figure (IV.11).

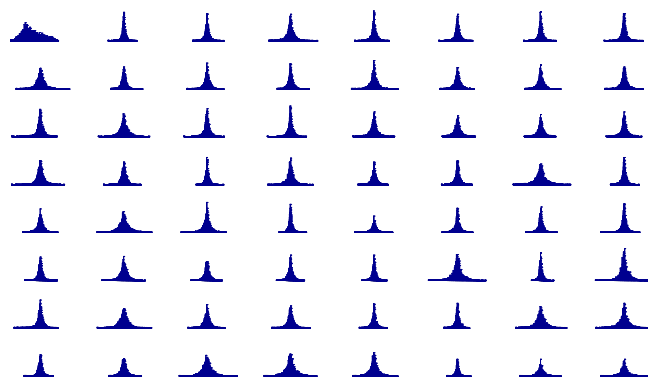


Figure (IV.11) : Histogrammes des coefficients DCT entière proposée T_2 pour l'image « Bridge 512×512 »

Les valeurs des deux tests d'adéquation pour les 15 premiers coefficients DCT sont données dans les tableaux (IV.12) et (IV.13).

Position du Coeff.	Image (3)			Image (4)			Image (5)			Image (6)		
	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G
0,0	-	0.0249	0.0211	0.0241	-	0.0185	-	-	0.0244	-	-	-
0,1	-	0.0187	0.0176	-	0.0180	0.0238	-	0.0290	-	-	-	-
1,0	-	-	-	-	-	-	-	-	-	-	-	-
2,0	-	-	-	-	-	-	-	-	-	-	-	-
1,1	-	0.0132	0.0168	-	0.0202	0.0176	-	-	0.0121	-	-	-
0,2	-	0.0246	0.0249	-	0.0201	0.0159	-	0.0267	0.0175	-	-	-
0,3	-	0.0167	0.0192	-	0.0259	0.0215	-	0.0274	0.0255	0.0483	-	-
1,2	-	0.0139	-	-	0.0186	0.0218	-	0.0296	0.0254	-	-	-
2,1	-	0.0201	-	-	-	-	-	0.0307	0.0175	0.0334	0.0329	0.0284
3,0	-	-	-	-	-	-	-	-	-	-	-	-
4,0	-	-	-	-	-	-	-	-	-	-	-	-
3,1	-	-	-	-	0.0258	-	-	-	0.0312	0.0367	0.0361	0.0361
2,2	-	-	-	-	0.0147	0.0257	-	0.0251	0.0196	0.0322	0.0297	0.0302
1,3	-	0.0218	0.0260	-	0.0245	0.0232	-	0.0303	0.0290	0.0526	-	0.0542
0,4	-	0.0193	0.0256	-	0.0259	0.0289	-	-	-	-	-	0.0293

Tableau (IV.12) : Valeurs statistiques de test KS pour les 15 premiers coefficients en ordre de zigzag pour la transformée proposée T_2

Position du Coeff.	Image (3)			Image (4)			Image (5)			Image (6)		
	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G	1G	2 G	3 G
0,0	-	-	5.6068	-	-	8.8294	-	-	113.3197	-	-	1.8412
0,1	-	-	16.8919	-	-	34.0092	-	86.5544	25.5597	-	-	12.4261
1,0	-	17.7438	20.5950	-	-	5.0599	-	-	15.1113	-	4.5338	4.2226
2,0	-	31.7474	5.3526	-	-	27.2682	-	63.7392	18.4980	-	-	3.5680
1,1	-	-	3.8204	-	-	6.5214	-	65.0969	16.4646	11.555	-	9.9693
0,2	-	-	1.6346	-	-	22.0492	-	36.1665	5.7828	-	5.6665	4.4029
0,3	-	-	4.8698	-	-	27.6061	-	48.2553	3.2845	-	-	3.3989
1,2	-	-	31.1731	-	-	13.3658	-	32.7513	5.6356	-	-	4.4522
2,1	-	18.9305	86.8869	-	-	68.0094	-	49.2778	9.5262	-	-	7.9870
3,0	-	18.2910	14.6234	-	-	18.5968	-	53.2861	3.1850	-	-	4.4950
4,0	-	19.8153	3.1079	-	-	32.5334	-	68.6578	5.1455	-	-	3.5356
3,1	-	-	41.3905	-	24.6555	25.8682	-	49.5729	10.4448	7.4655	5.9195	6.0286
2,2	-	-	2.0415	-	33.0735	20.9864	-	53.5567	10.2612	-	-	2.7158
1,3	-	12.1208	22.4325	-	-	1.5578	-	14.3896	14.4374	-	-	8.1213
0,4	-	-	19.9492	-	15.3779	2.9562	-	11.4980	1.0641	-	-	3.9274

Tableau (IV.13) : Valeurs statistiques de test χ^2 pour les 15 premiers coefficients en ordre de zigzag pour la transformée proposée T_2

Un exemple d'approximation de la loi empirique par différentes lois théoriques est représenté sur la figure (IV.12) pour l'image « Lena 512×512 ».

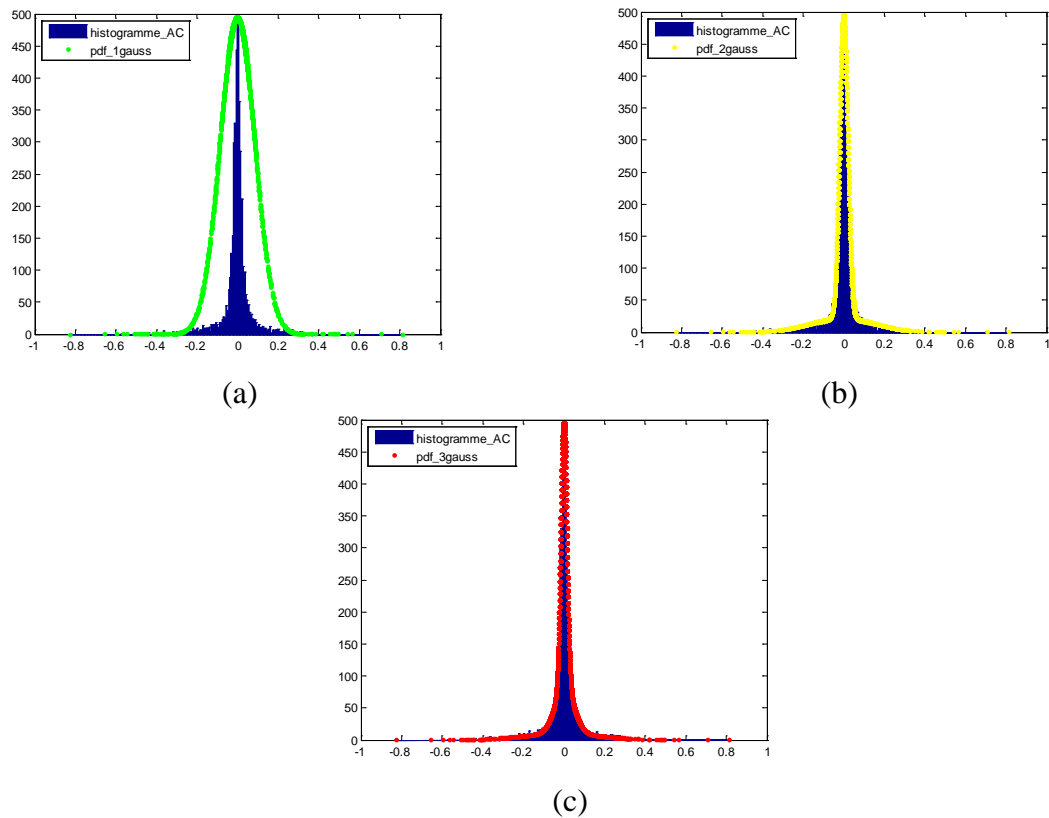


Figure (IV.12) : Approximation de la loi empirique du coefficient (2,1) de la transformée proposée T_2 de l'image « Lena 512×512 » par différentes lois théoriques

D'après les deux tableaux (IV.12) et (IV.13), la plus part des coefficients considérés pour toutes les images sous test peuvent être modélisés par un mélange de deux et trois Gaussiennes, mais si en comparant les valeurs des tests, on trouve que le mélange de trois Gaussiennes adapte mieux l'approximation.

IV. 6. Résultats globaux

Dans les sous-sections précédentes, nous avons trouvé que toutes les transformées peuvent être modélisées par une somme deux ou trois Gaussiennes, et ceci pour les coefficients pris en considération. Dans cette sous-section, nous allons donner les résultats consolidés pour tous les coefficients d'une transformée donnée.

Dans le tableau (IV.14), le nombre des coefficients pour la meilleure modélisation par la loi considérée est donné à droite et à gauche et donne aussi le nombre total des coefficients

qui peuvent être modélisés par cette loi. La meilleure modélisation en se basant sur le test de KS ou Chi2 est mise en gras.

Transformée		Image (3)		Image (4)		Image (5)		Image (6)	
		Test KS	Test χ^2	Test KS	Test χ^2	Test KS	Test χ^2	Test KS	Test χ^2
Transformée de [11]	1G	00/00	00/00	05/16	01/30	00/00	00/00	00/01	00/00
	2G	35/43	04/15	13/20	02/13	03/15	05/42	23/31	10/17
	3G	15/43	60/64	03/14	61/64	17/18	59/64	09/24	54/64
Transformée de [12]	1G	00/00	00/00	04/09	00/26	00/00	00/00	00/00	00/00
	2G	11/18	09/16	03/10	00/14	00/04	05/42	08/12	09/16
	3G	08/13	55/64	06/11	64/64	06/06	59/64	07/14	55/64
Transformée proposée T_1	1G	00/00	00/00	00/00	01/17	00/00	00/00	00/01	01/17
	2G	00/01	09/18	00/00	01/06	00/00	03/51	01/01	01/06
	3G	02/02	55/64	00/00	62/64	01/01	61/64	01/01	62/64
Transformée proposée T_2	1G	00/00	00/00	08/15	01/25	00/00	00/00	00/01	00/00
	2G	25/33	06/13	06/16	02/12	01/08	04/46	19/25	08/16
	3G	13/25	58/64	08/13	61/64	10/10	60/64	11/24	56/64

Tableau (IV.14) : Résultats consolidé pour les transformées étudiées

Les résultats globaux considérés pour les 4 images (3), (4), (5) et (6) sont résumés dans le tableau (IV.15).

Transformée		Test KS	Test χ^2
Transformée de [11]	1G	05/17	01/30
	2G	74/109	21/87
	3G	44/99	234/256
Transformée de [12]	1G	04/09	00/00
	2G	22/44	23/88
	3G	27/44	233/256
Transformée proposée T_1	1G	00/01	02/34
	2G	01/02	28/81
	3G	04/04	240/256
Transformée proposée T_2	1G	08/15	01/25
	2G	51/82	20/87
	3G	42/72	235/256

Tableau (IV.15) : Résultats globaux de la modélisation des transformées entières étudiées

- En se basant sur le test de KS, pour la transformée de [11], 44 de 99 coefficients peuvent être modélisés approximativement par une somme de 3 Gaussiennes contre 74 de 109

coefficient qui peuvent être modélisés par une somme de 2 Gaussiennes, mais en se basant sur le test de χ^2 , 234 de 256 coefficients peuvent être modélisés par la somme de 3 Gaussiennes contre 21 de 87 qui peuvent être modélisés par une somme de 2 Gaussiennes. Nous voyons donc que la somme de trois Gaussiennes est la modélisation adéquate pour la majorité des coefficients de la transformée de [11].

- Le même résultat peut être observé dans le même tableau pour les autres transformées dont 27 de 44 coefficients peuvent être modélisés par une somme de trois Gaussiennes en se basant sur le test de KS pour la transformée de [12] et 233 de 256 dans le cas du test χ^2 .
- Pour nos transformées proposées, il est clair d'après le tableau que la distribution d'une somme de 3 Gaussiennes présente la meilleure modélisation par 240 parmi 256 coefficients pour \mathbf{T}_1 et 235 coefficients pour \mathbf{T}_2 en se basant sur le test de χ^2 .

Ainsi, on peut conclure que pour la majorité des coefficients, la distribution d'un mélange de trois Gaussiennes présente la meilleure approximation pour les coefficients des DCT entières étudiées.

IV.7. Conclusion

Dans ce chapitre, en se basant sur les distributions énergétiques des coefficients de chaque transformée, nous avons aussi développé des nouvelles matrices de quantification propres aux transformées entières proposées et existantes. Ainsi, nous avons utilisé deux tests d'adéquation pour déterminer la distribution adéquate pour la modélisation des coefficients des transformées entières dédiés à la compression d'images de [11] et [12] ainsi pour nos transformées \mathbf{T}_1 et \mathbf{T}_2 proposées dans le chapitre précédent. Les résultats obtenus montrent qu'une seule distribution ne peut pas être utilisée pour modéliser tous les coefficients pour toutes les images. Néanmoins, la distribution de la majorité des coefficients DCT entières étudiées peuvent être modélisés par une somme de 3 Gaussiennes dans la plus part des cas. Ce résultat permet une construction optimale des tables de quantification appropriées pour chacune de ces transformées.

Dans le chapitre suivant, nous allons donner les résultats expérimentaux de l'implémentation des transformées DCT entières proposées et existantes dans le standard JPEG, ainsi nous montrons l'amélioration des performances de la compression réalisée par les matrices de quantification proposées.



**CHAPITRE
V**

Résultats expérimentaux et discussion

V.1. Introduction

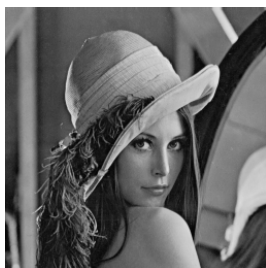
Dans ce chapitre, nous avons adopté la technique JPEG de compression et de décompression des images fixes afin d'étudier et comparer les capacités et les performances des transformées DCT entières proposées dans le chapitre (III) et celles qui existent dans la littérature [28], [10], [12], [50], [11].

Il y a plusieurs raisons de ce choix:

- Le JPEG est la première norme internationale de compression établie pour les images numériques monochromes et couleurs.
- Le JPEG a identifié en tant que la technique de codage, basée sur la transformation, le plus populaire, simple, et efficace qui rapporte une solution satisfaisante à la plupart des problèmes de codage pratiques d'images.
- En outre, la norme JPEG a joué un rôle considérable dans le développement d'autres normes internationales de codage vidéo. Du point de vue méthodologique, la norme JPEG permet d'illustrer la capacité de compression des transformées entières proposées.

V.2. Images de test utilisées

Six images de test naturelles aux niveaux de gris (256×256, 8 bits/pixel) sont utilisées pour montrer l'efficacité des transformées ainsi que les matrices de quantification proposées: Lena, Cameraman, Goldhill, Baboon, Peppers et Bridge qui sont données dans la figure (V.1).



Lena 256×256



Cameraman 256×256



Goldhill 256×256



Baboon 256×256



Peppers 256×256



Bridge 256×256

Figure (V.1) : Images de test naturelles utilisées

Les histogrammes correspondants aux images de test utilisées sont représentés sur la figure (V.2).

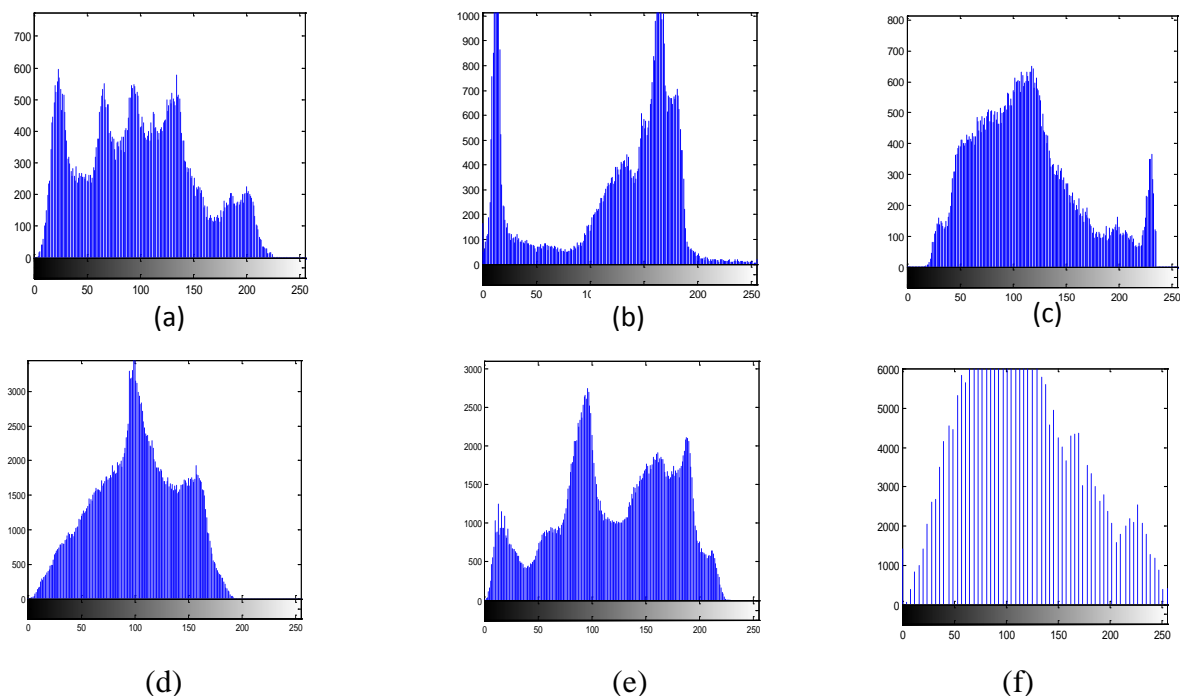


Figure (V.2) : Histogrammes correspondants aux images de test utilisées

(a) Lena, (b) Cameraman, (c) Goldhill, (d) Baboon, (e) Peppers, (f) Bridge

Le choix de ces images dépend de la richesse de l'échelle de gris. Tous les calculs sont exécutés avec **MATLAB 7.8 (R2009a)**.

V.3. Résultats expérimentaux

Pour pouvoir évaluer rapidement les performances des transformées DCT entières proposées, les opérations de codage et de décodage ne sont pas mises en œuvre dans cette section. La figure (V.3) décrit la chaîne de compression réduite utilisée.

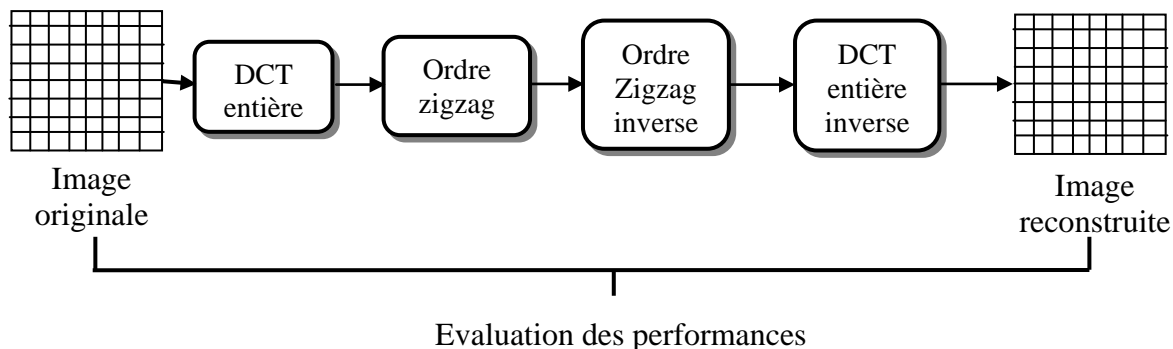


Figure (V.3) : Chaîne de compression et de décompression réduite utilisée

Tout d'abord, l'image est divisée en blocs de taille 8×8 , ensuite la transformée entière considérée est appliquée sur ces blocs indépendamment, si on considère \mathbf{X} d'être un bloc d'image de test, sa transformée directe est inverse sont données par :

$$\mathbf{F} = \mathbf{T}\mathbf{X}\mathbf{T}' = \mathbf{D}(\mathbf{C}\mathbf{X}\mathbf{C}')\mathbf{D} \quad (\text{V.1})$$

$$\mathbf{X} = \mathbf{T}'\mathbf{F}\mathbf{T} = \mathbf{C}'(\mathbf{D}\mathbf{F}\mathbf{D})\mathbf{C} \quad (\text{V.2})$$

\mathbf{T} : Matrice de la transformée considérée.

\mathbf{D} : Matrice diagonale correspondante à la transformée.

\mathbf{F} : Bloc transformé.

Après l'application de la transformée les coefficients sont ré-ordonnés en zigzag, de 1 à 30 coefficients sont pris pour restaurer l'image.

- Mesure de la qualité d'images reconstruites

La qualité visuelle des images reconstruites est évaluée qualitativement et quantitativement avec les critères présentés dans le chapitre I, en addition de ces critères nous avons choisi d'utiliser la norme de pourcentage d'erreur d'énergie (PEEN : Percentage Energy Error Norm) donnée par l'équation (V.3), qui est une mesure de la distorsion (ou bien la distance relative) entre l'image originale et l'image reconstruite après le processus de décompression.

$$\text{PEEN} = \sqrt{\frac{\sum_{m=0}^{M-1} \sum_{n=0}^{M-1} (f(m,n) - \hat{f}(m,n))^2}{\sum_{m=0}^{M-1} \sum_{n=0}^{M-1} f^2(m,n)}} \times 100 \quad (\text{V.3})$$

$f(m,n)$: Pixel de l'image originale.

$\hat{f}(m,n)$: Pixel de l'image reconstruite.

V.3.1. Résultats obtenus pour la transformée proposée \mathbf{T}_1

Nous présentons dans cette sous-section les résultats de simulation pour la transformée proposée \mathbf{T}_1 , cette dernière apporte une réduction plus que 33% dans le nombre d'opérations

(comme nous avons vu dans le chapitre III) par rapport à celle de [11]. Une comparaison en terme du PSNR entre notre transformée proposée T_1 et quelques transformées entières existantes [11], [10] et [12] en fonction du nombre de coefficients retenus pour la reconstruction de l'image est montrée sur la figure (V.4).

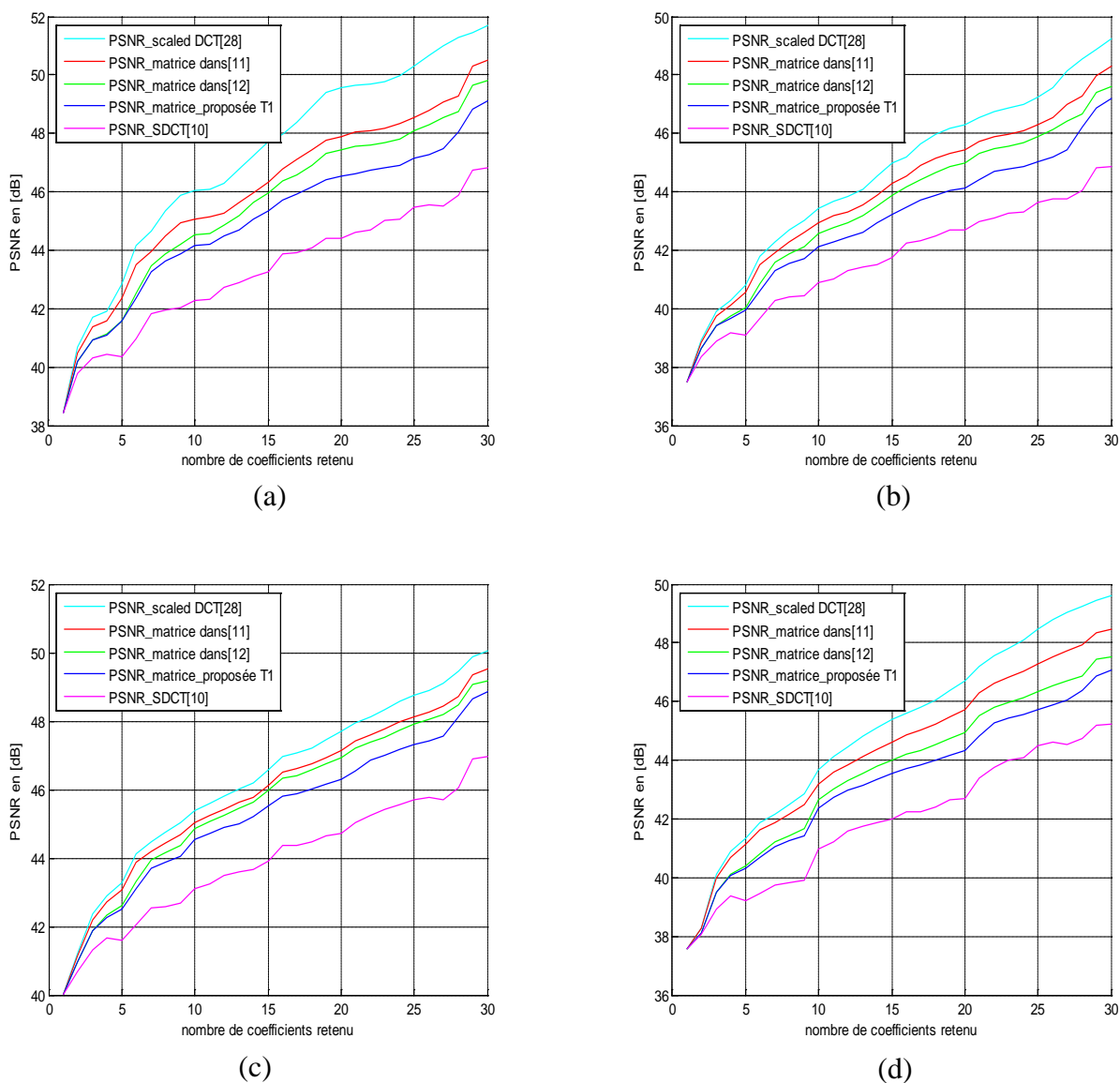


Figure (V.4) : Comparaison en terme de PSNR en utilisant : Scaled DCT [28], transformée de [11], transformée de [12], notre transformée proposée T_1 et SDCT [10] pour les images : (a) Lena, (b) Cameraman, (c) Goldhill et (d) Bridge

Une comparaison objective en terme de PSNR et PEEN est faite dont ses résultats sont résumés dans les deux tableaux (V.1) et (V.2).

Images de test	Scaled DCT [28]	Transformée de [11]	Transformée de [12]	Notre transformée proposée T_1	SDCT [10]
Lena	46.0276	45.0705	44.5247	44.1729	42.2891
Cameraman	43.4152	42.9463	42.5616	42.1152	40.8935
Goldhill	45.3711	45.0239	44.8716	44.5295	43.1187
Bridge	43.6674	43.1674	42.6583	42.3760	40.9745

Tableau (V.1): Comparaison objective en terme de PSNR (en dB) entre la transformée entière proposée T_1 et les autres transformées entières existantes

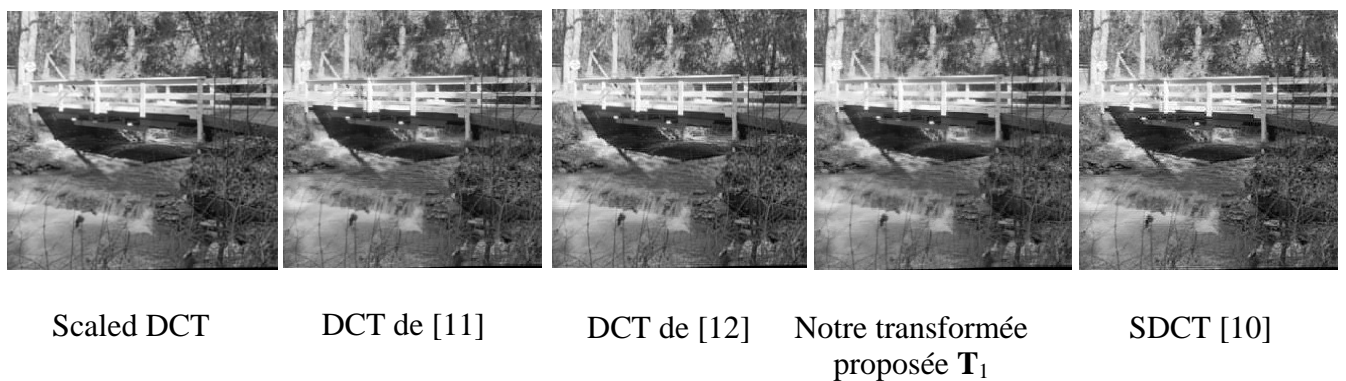
Images de test	Scaled [28]	Transformée de [11]	Transformée de [12]	Notre transformée proposée T_1	SDCT [10]
Lena	9.0904	10.1494	10.8075	11.2543	13.9801
Cameraman	10.2274	10.7947	11.2836	11.8787	13.6726
Goldhill	8.9559	9.3211	9.4859	9.8670	11.6072
Bridge	10.6120	11.2409	11.9193	12.3132	14.4692

Tableau (V.2): Comparaison en terme de PEEN entre la transformée entière proposée T_1 et les autres transformées entières existantes

L'image originale et les images reconstruites en utilisant ces différentes transformées sont illustrées dans la figure (V.4), pour l'image « Lena » et « Bridge ». 30 sur les 64 coefficients des transformées dans chaque bloc ont été utilisés pour la reconstruction de l'image, tous les autres coefficients ont été mis à zéro.



(a)



(b)

Figure (V.5) : Images reconstruites en utilisant : Scaled DCT [28], la transformée de [11], transformée dans [12], notre transformée proposée T_1 et SDCT [10] : (a) Lena, (b) Bridge

Il est clair d'après ces images que la qualité visuelle des images reconstruites en utilisant la matrice proposée T_1 est proche ou légèrement inférieure à celle obtenue en utilisant la matrice DCT conventionnelle ou les autres transformées entières. En plus, un gain considérable dans le nombre d'opérations peut être obtenu en utilisant la transformée proposée T_1 comparée à la DCT conventionnelle et les autres transformées existantes [11], [12] et [10] (voir chapitre III).

V.3.2. Résultats obtenus pour la transformée proposée T_2

La figure (V.6) compare les valeurs du PSNR pour la transformée proposée T_2 et les autres transformées de [28], [10] et [50] en fonction du nombre de coefficients DCT utilisés

pour la reconstruction de l'image pour les quatre images de test, où la DCT conventionnelle est considéré comme le repère de comparaison.

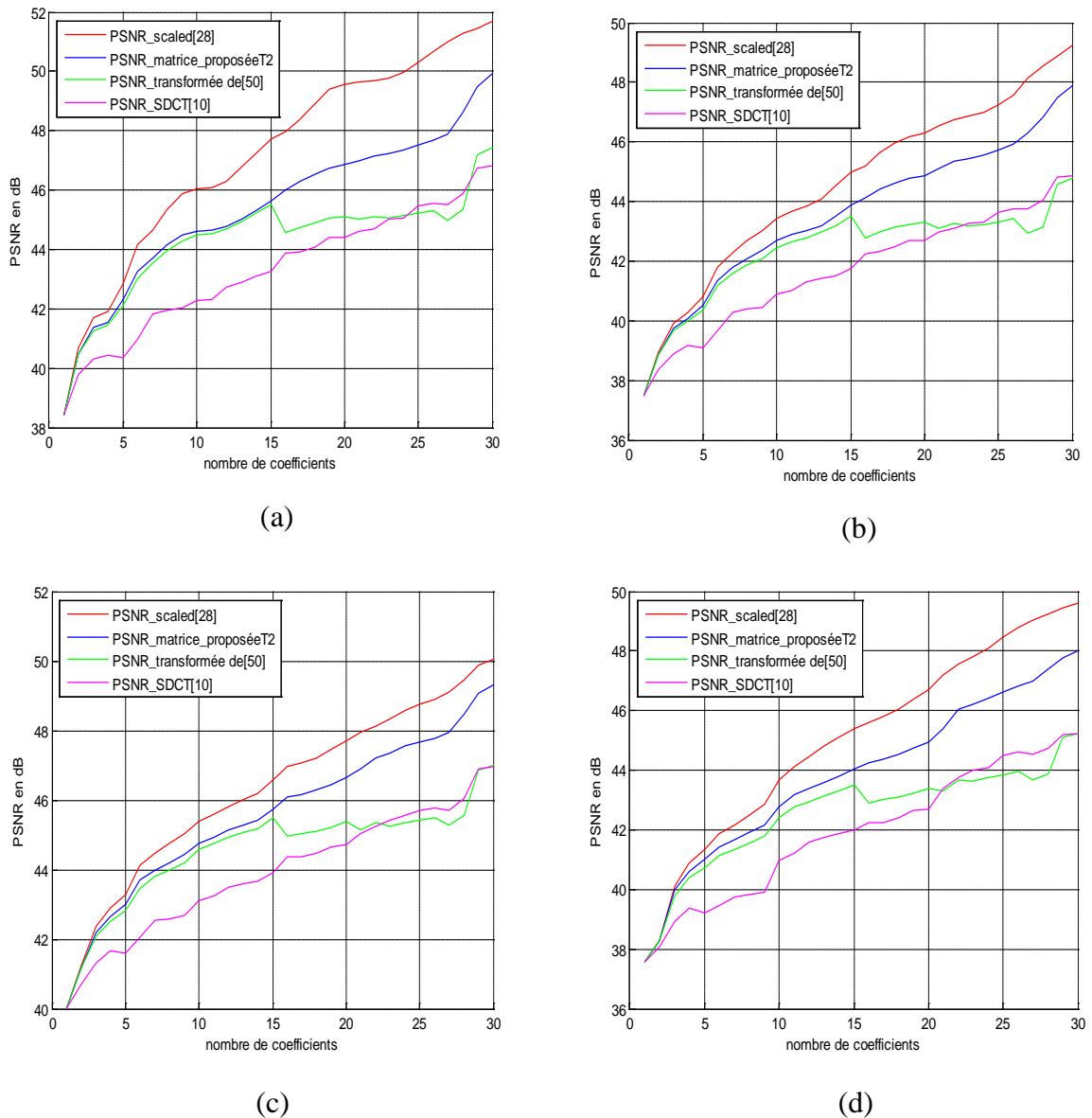


Figure (V.6) : Comparaison en terme PSNR en utilisant : Scaled DCT [28], transformée de [12], notre transformée proposée T_2 et SDCT [10] pour les images : (a) Lena, (b) Cameraman, (c) Goldhill et (d) Bridge.

D'après la figure (V.6), si nous considérons seulement les valeurs du PSNR pour un nombre de coefficients retenus inférieur à quinze (taux de compression élevé), alors nous voyons que notre transformée T_2 est meilleure que la SDCT et apporte quelques améliorations en dB au-dessus de la transformée de [50], mais si en utilise 30 coefficients (taux de

compression bas), alors, notre matrice propose T_2 apparait la meilleure par rapport à toutes les autres transformées entières. Les tableaux (V.3) et (V.4) montrent une comparaison objective des résultats de la compression en terme de PSNR (dB) et de PEEN, respectivement.

Images de test	Scaled DCT [28]	Transformée T_2	Transformée de [50]	SDCT [10]
Cameraman	43.4152	42.6804	42.4615	40.8935
Lena	46,0276	44,6014	44,4899	42,2891
Goldhill	45.3711	44.7462	44.5641	43.1187
Bridge	42.7828	42.7828	42.4136	40.9745

Tableau (V.3): Comparaison objective en terme de PSNR (en dB) entre la transformée entière proposée T_2 et les autres transformées entières existantes.

Images de test	Scaled DCT [28]	Transformée T_2	Transformée de [50]	SDCT [10]
Cameraman	10.2274	11.1303	11.4144	13.6726
Lena	9.0904	10.7126	10.8510	13.9801
Goldhill	8.9559	9.6239	9.8278	11.6072
Bridge	5.3561	6.4263	8.8759	8.8841

Tableau (V.4): Comparaison objective en terme de PEEN entre la transformée entière proposée T_2 et les autres transformées entières existantes.

D'après les deux tableaux (V.3) et (V.4), il apparait que la transformée proposée T_2 porte une amélioration dans les performances de compression pour tous les types d'images considérées malgré qu'elle a le même nombre d'opérations (24 opérations) que les autres [50] et [10].

L'image originale et les images reconstruites en utilisant les différentes matrices DCT présentées précédemment sont illustrées dans la figure (V.7) pour l'image « Lena » et « Bridge ». 30 coefficients DCT sur 64 dans chaque bloc ont été utilisés pour la reconstruction des images. Tous les autres coefficients ont été mis à zéro.

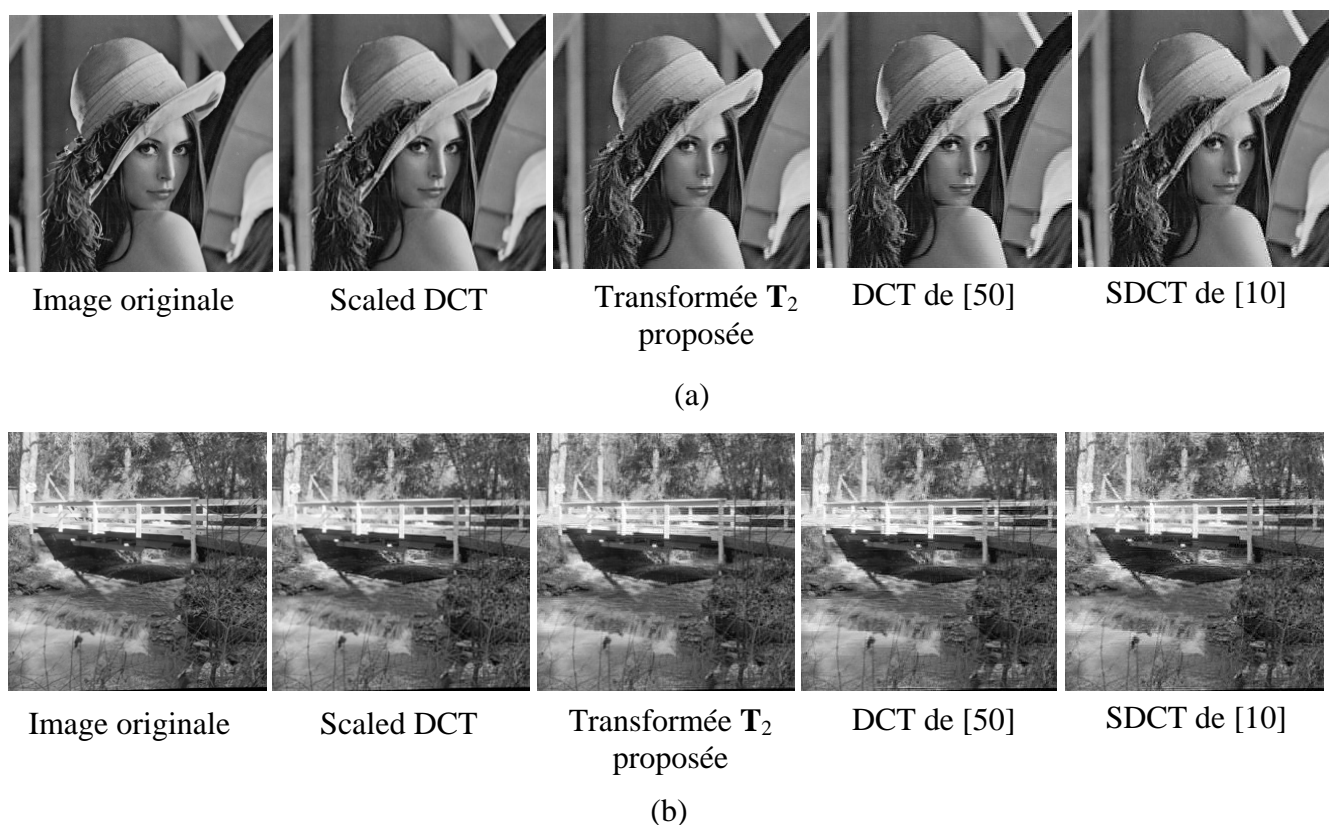


Figure (V.7) : Images reconstruites en utilisant Scaled DCT [28], matrice proposée T_2 , matrice DCT entière de [50] et SDCT [10] : (a) Lena, (b) Bridge.

L'amélioration en terme qualité visuelle des images reconstruites en utilisant la transformée entière proposée T_2 est vérifiée d'après les figures représentées ci-dessus, dont la qualité visuelle des images reconstruites en utilisant cette dernière est proche ou légèrement inférieure à celle obtenue en utilisant la matrice DCT conventionnelle avec une réduction remarquable dans le nombre d'opérations exigé. par contre elle porte une amélioration visuelle en la comparant avec celle les autres transformées en conservant le même nombre d'opérations.

V.3.3. Implémentation des transformées proposées dans la norme JPEG

Nous avons implémenté dans la norme JPEG les transformées entières proposées ainsi que les autres transformées existantes dans la littérature [28], [10], [12], [50], [11]. Cette implémentation nous a donné les résultats qui sont présentés dans le tableau (V.5) en terme de PSNR et le taux de compression CR. Ceci pour les 4 images de test « Lena, Goldhill, Cameraman et Bridge » qui sont de taille 256×256.

Transformées	DCT [28]		Transformée dans [11]		Transformée dans [12]		Transformée proposée T_1		Transformée proposée T_2		Transformée dans [50]		SDCT [10]	
	PSNR	CR	PSNR	CR	PSNR	CR	PSNR	CR	PSNR	CR	PSNR	CR	PSNR	CR
Images de test														
Lena	31.0618	8.3210	29.7654	8.3934	29.0023	8.2622	28.1803	8.3358	28.9996	8.3571	26.8492	8.3316	26.3371	8.2831
Goldhill	30.2941	7.1157	29.5062	7.0834	29.1052	7.1049	28.5454	7.1608	29.2704	7.0773	26.6783	7.1593	26.6294	7.1204
Cameraman	29.9442	8.0442	29.8952	8.0709	29.1984	8.0176	28.9314	8.0373	30.0068	8.0610	26.4844	8.0078	25.9778	8.0551
Bridge	29.7483	5.7558	28.6823	5.7629	27.8053	5.7488	27.3405	5.7077	28.4715	5.6575	25.1022	5.6919	25.5639	5.7197

Tableau (V.5) : Comparaison entre les différentes transformées en terme de PSNR et le CR

En conservant le taux de compression pour toutes les transformées, nous remarquons que la transformée proposée T_1 présente une réduction dans le PSNR pour toutes les images considérées, cette réduction est justifiée par sa réduction en complexité, où elle présente une réduction plus de 33% par rapport à celle de [11] et plus de 11% par rapport à celle de [12]. Pour la transformée T_2 , malgré qu'elle conserve la même complexité par rapport à celle des transformées [50] et [10], elle présente une bonne amélioration en terme du PSNR.

V.3.4. Implémentation des matrices de quantification proposées dans JPEG

Dans cette section, nous allons présenter les résultats expérimentaux obtenus de l'application de nos matrices de quantification proposées sur la norme standard de compression d'image JPEG. Pour chaque transformée, en conservant le même taux de compression, nous avons déterminé le PSNR en utilisant la matrice de quantification conventionnelle de JPEG, Q_{JPEG} , et la matrice de quantification proposée pour chaque transformée.

Les résultats obtenus pour les 6 images de test sont résumées dans le tableau (V.6).

Les résultats montrent bien que l'utilisation des matrices de quantification appropriée pour chaque transformée fournit une bonne amélioration en terme de PSNR pour toutes images de test considérées et une légère amélioration dans le CR pour quelques images. Ces résultats sont confirmés par les images des figures (V.6)-(V.9).

Images de test	DCT conventionnelle		Transformée de [11]				Transformée de [12]				Transformée proposée T_1				Transformée proposée T_2			
	Q_{JPEG}		Q_{JPEG}		Q_{p1}		Q_{JPEG}		Q_{p2}		Q_{JPEG}		Q_{p3}		Q_{JPEG}		Q_{p3}	
	PSNR	CR	PSNR	CR	PSNR	CR	PSNR	CR	PSNR	CR	PSNR	CR	PSNR	CR	PSNR	CR	PSNR	CR
Lena	32.9432	7.9092	31.7815	7.3586	32.3692	7.5329	31.5774	6.8111	32.0565	6.9689	30.9123	6.7521	31.3398	6.8267	31.2138	7.3159	31.9852	7.3603
Goldhill	30.5962	6.8869	30.2185	6.7466	30.6421	6.7203	30.1231	6.3028	30.4501	6.3863	29.9123	6.3100	30.1018	6.3652	29.9724	6.7189	30.5021	6.6225
Bridge	30.0802	5.4759	29.4615	5.3403	30.0164	5.3412	29.0349	4.9679	29.5700	5.0443	28.5241	4.7916	28.9856	5.0389	29.0562	5.3212	29.7450	5.2429
Baboon	29.0433	5.6712	28.5544	5.5823	29.1746	5.5548	28.0976	5.1072	28.7587	5.1563	27.6672	5.1232	28.2378	5.1409	28.2522	5.5577	28.9499	5.4659
Peppers	36.8913	10.4858	34.9277	9.7350	35.3043	10.2240	34.7144	9.1352	34.9328	9.6376	33.9532	9.1531	34.2139	9.5729	34.2690	9.7495	34.8761	10.0331
Cameraman	31.3360	7.0529	31.5337	7.3818	32.7164	7.7741	31.0300	7.3094	32.0319	7.6134	30.5998	7.3769	31.3038	7.6740	31.0916	7.5624	32.0931	7.6992

Tableau (V.6) : Résultats comparatives entre les différentes matrices de quantification proposées et la matrice conventionnelle de JPEG pour les différentes transformées proposées et existantes



(a)



(b)

Figure (V.8) : Image « Lena » restituée en utilisant la transformée de [11](a) Matrice de quantification JPEG Q_{JPEG} (b) Matrice de quantification proposée appropriée Q_{p1} 

(a)



(b)

Figure (V.9) : Image « Lena » restituée en utilisant la transformée dans [12](a) Matrice de quantification JPEG Q_{JPEG} (b) Matrice de quantification proposée appropriée Q_{p2}



(a)



(b)

Figure (V.10) : Image « Lena » restituée en utilisant la transformée proposée T_1 (a) Matrice de quantification JPEG Q_{JPEG} (b) Matrice de quantification proposée Q_{p3} 

(a)



(b)

Figure (11) : Image « Lena » restituée en utilisant la transformée proposée T_2 (a) Matrice de quantification JPEG Q_{JPEG} (b) Matrice de quantification proposée Q_{p3}

Les figures montrent bien l'amélioration qu'apportent les matrices de quantification sur les performances des transformées correspondantes.

V.4. Conclusion

Dans ce chapitre, nous avons donné les résultats expérimentaux de nos transformées ainsi que nos matrices de quantification implémentées dans le standard JPEG. D'après la comparaison avec les transformées entières existantes, nos transformées nouvelles \mathbf{T}_1 et \mathbf{T}_2 proposées dans ce travail offrent différents avantages: elles sont rapides, ont des structures simples pour l'implémentation, et présentent des performances acceptables dans la compression d'images. Comme nous avons montré aussi que les matrices de quantification proposées améliorent les performances en terme de PSNR et de qualité visuelle d'images restituées par les transformées correspondantes.

Conclusion générale et perspectives

Conclusion générale

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 de ce travail de recherche, nous avons vu l'importance primordiale qui joue les transformées DCT entières dans les systèmes de compression d'images, ceci nous amène à développer deux nouvelles transformées entières dédiées à cette application. Nos deux transformées proposées ont une très bonne performance en compression d'images, ceci est mesurée qualitativement par une comparaison entre les images restituées en utilisant nos transformées, la DCT conventionnelle et quelques transformées entières existantes, et quantitativement par les deux critères PSNR et PEEN où elles présentent une petite dégradation tout en conservant la qualité visuelle des images reconstruites. Noté aussi que l'une des transformées proposées apporte une réduction importante en terme de nombre d'opérations exigées par rapport à celles existantes.

Dans la deuxième partie de ce travail, nous avons amélioré les performances du standard JPEG basé sur les transformées DCT entières. Cette amélioration a été réalisée grâce aux nouvelles matrices de quantification appropriées que nous avons développé. L'augmentation des PSNR obtenus en utilisant ces nouvelles matrices de quantification est de plus de 0.5 dB en les comparant avec les résultats de l'utilisation de la matrice de quantification conventionnelle de JPEG.

Nous avons introduit aussi dans ce travail une modélisation statistique des coefficients des transformées entières proposées et existantes. Les résultats expérimentaux montrent que ces coefficients peuvent être modélisés efficacement par une somme de 2 ou 3 Gaussiennes.

Perspectives

Nous suggérons en perspectives le développement des matrices de quantifications optimales pour chaque transformée entière en exploitant cette nouvelle modélisation statistique.

Dû au succès des transformées et des matrices de quantification proposées dans ce modeste travail dans le cas de la compression des images fixes, leur extension au cas des signaux vidéos devrait également fournir de bons résultats.

Nous avons limité notre développement au cas des transformées de taille 8×8 , qui est le cas le plus utilisé dans la compression des images fixes. Notre approche pourra être exploitée pour le développement de nouvelles transformées entières de tailles supérieures.

ANNEXE

I. Matrices de quantifstandication adoptées par le standard JPEG

$$T_{Lum} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \quad T_{Chr} = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

(a) Luminance

(b) Chrominance

Tableau (1) : Matrices de quantification recommandées du JPEG [18]

II. Codes Huffman adoptés par la norme JPEG

Z	R				
	1 6	2 7	3 8	4 9	5 A
0	00 1111000	01 11111000	100 1111110110	1011 111111110000010	11010 1111111110000011
1	1100 111111110000100	11011 111111110000101	11110001 111111110000110	111110110 111111110000111	1111110110 111111110001000
2	11100 111111110001010	11111001 111111110001011	1111110111 111111110001100	111111110100 111111110001101	111111110001001 111111110001110
3	111010 1111111110010001	111110111 1111111110010010	111111110101 1111111110010011	111111110001111 1111111110010100	1111111110010000 1111111110010101
4	111011 1111111110011001	1111111000 1111111110011010	1111111110010110 1111111110011011	1111111110010111 1111111110011100	1111111110011000 1111111110011101
5	1111010 1111111110100001	11111110111 1111111110100010	1111111110011110 1111111110100011	1111111110011111 1111111110100100	1111111110100000 1111111110100101
6	1111011 1111111110101001	111111110110 1111111110101010	1111111110100110 1111111110101011	1111111110100111 1111111110101100	1111111110101000 1111111110101101
7	11111010 1111111110110001	111111110111 1111111110110010	1111111110101110 1111111110110011	1111111110101111 1111111110110100	1111111110110000 1111111110110101
8	111111000 1111111110111001	111111111000000 1111111110111010	1111111110110110 1111111110111011	1111111110110111 1111111110111100	1111111110111000 1111111110111101
9	111111001 1111111111000010	1111111110111110 1111111111000011	1111111110111111 1111111111000100	1111111111000000 1111111111000101	1111111111000001 1111111111000110
A	111111010 1111111111001011	111111111000111 1111111111001100	1111111111001000 1111111111001101	1111111111001001 1111111111001110	1111111111001010 1111111111001111
B	1111111001 1111111111010100	1111111111010000 1111111111010101	1111111111010001 1111111111010110	1111111111010010 1111111111010111	1111111111010011 1111111111011000
C	1111111010 1111111111011101	1111111111011001 1111111111011110	1111111111011010 1111111111011111	1111111111011011 1111111111000000	1111111111011100 1111111111000001
D	11111111000 1111111111100110	1111111111100010 1111111111100111	1111111111100011 1111111111101000	1111111111100100 1111111111101001	1111111111100101 1111111111101010
E	1111111111101011 1111111111110000	1111111111101100 1111111111110001	1111111111101101 1111111111110010	1111111111101110 1111111111110011	1111111111101111 1111111111110100
F	11111111001 111111111111001	1111111111110101 111111111111010	1111111111110110 111111111111011	1111111111110111 111111111111101	1111111111111000 111111111111110

Tableau (2) : Codes Huffman recommandés pour les coefficients AC de chrominance [18]

Z	R				
	1 6	2 7	3 8	4 9	5 A
0	01 111000	100 1111000	1010 111110100	11000 1111110110	11001 11111110100
1	1011 11111110101	111001 111111110001000	11110110 111111110001001	111110101 111111110001010	11111110110 111111110001011
2	11010 111111110001100	11110111 1111111110001101	1111110111 1111111110001110	111111110110 1111111110001111	11111111000010 1111111110010000
3	11011 1111111110010010	11111000 1111111110010011	1111111000 1111111110010100	111111110111 1111111110010101	111111110010001 1111111110010110
4	111010 1111111110011010	111110110 1111111110011011	111111110010111 1111111110011100	1111111110011000 1111111110011101	111111110011001 1111111110011110
5	111011 1111111110100010	1111111001 1111111110100011	111111110011111 1111111110100100	1111111110100000 1111111110100101	1111111110100001 1111111110100110
6	1111001 1111111110101010	11111110111 1111111110101011	1111111110100111 1111111110101100	1111111110101000 1111111110101101	1111111110101001 1111111110101110
7	1111010 1111111110110010	11111111000 1111111110110011	1111111110101111 1111111110110100	1111111110110000 1111111110110101	1111111110110001 1111111110110110
8	11111001 1111111110111011	1111111110110111 1111111110111100	1111111110111000 1111111110111101	1111111110111001 1111111110111110	1111111110111010 1111111110111111
9	111110111 1111111111000100	1111111111000000 1111111111000101	1111111111000001 1111111111000110	1111111111000010 1111111111000111	1111111111000011 1111111111001000
A	111111000 1111111111001101	1111111111001001 1111111111001110	1111111111001010 1111111111001111	1111111111001011 1111111111010000	1111111111001100 1111111111010001
B	111111001 1111111111010110	1111111111010010 1111111111010111	1111111111010011 1111111111011000	1111111111010100 1111111111011001	1111111111010101 1111111111011010
C	111111010 1111111111011111	1111111111011011 1111111111100000	1111111111011100 1111111111100001	1111111111011101 1111111111100010	1111111111011110 1111111111100011
D	11111111001 1111111111101000	1111111111100100 1111111111101001	1111111111100101 1111111111101010	1111111111100110 1111111111101011	1111111111100111 1111111111101100
E	11111111100000 1111111111110001	1111111111101101 1111111111110010	1111111111101110 1111111111110011	1111111111101111 1111111111110100	1111111111100000 1111111111110101
F	111111111000011 1111111111111010	1111111111010110 1111111111111011	1111111111110111 1111111111111100	1111111111111000 1111111111111101	1111111111111001 1111111111111110

Tableau (3) : Codes Huffman recommandés pour les coefficients AC de luminance [18]

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [1] M. Nelson, *The data compression book*. Cambridge, 2nd edition, IDG Books Worldwide, Inc, ISBN: 1558514341.
- [2] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, Vol. 38, No. 1, February 1992.
- [3] J. Miano, *Compressed image file formats JPEG, PNG, GIF, XBM, BMP, Your guide to graphics files on the Web*. USA : CM Press, ISBN 0-201-60443-4, 1999.
- [4] R. L. Queiroz, "Processing JPEG-compressed images and documents," *IEEE Transactions On Image Processing*, Vol. 7, No. 12, pp. 1661-1672, December 1998.
- [5] N. Ahmed, T. Natarajan et K. R. Rao, "Discret cosine transform," *IEEE Transactions On Computers*, pp. 90-93, January 1974.
- [6] K. R. Rao et P.C. Yip, *The transform and data compression handbook*. Boca Raton, CRC Press LLC, 2001.
- [7] T. Acharya, P. S. Tsai, *JPEG2000 standard for image compression : Concepts, Algorithms and VLSI architectures*. USA : John Wiley & Sons, Inc, 2005.
- [8] W. K. Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," *IEE Proceedings*, Vol. 136, Pt. I, No. 4, pp. 276-282, August 1989.
- [9] W. K. Cham et Y. T. Chan, "An order-16 integer cosine transform," *IEEE Transactions on Signal Processing*, Vol. 39, No. 5, pp. 1205-1208, May 1991.
- [10] T. I. Haweel, "A new square wave transform based on the DCT," *Signal Processing 81*, pp. 2309–2319, 2001.
- [11] K. Lengwehasatit et A. Ortega, " Scalable variable complexity approximate forward DCT," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, No. 11, pp. 1236-1248, November 2004.
- [12] S. Bouguezel, M. O. Ahmad et M. N. S. Swamy, "Low-complexity 8×8 transform for image compression," *Electronics Letters*, Vol. 44, No. 21, 9th October 2008.
- [13] M. Iwahashi et M. Ohnishi, "Analytical Evaluation of Integer DCT," *Intemational Symposium on Communications and Information Technologies 2004 (ISCIT 2004)*, Sapporo, Japan, pp. 1065-1068, October 26-29, 2004.

- [14] Y. Chen et P.Hao, "Integer reversible transformation to make JPEG lossless," *ICSP'04 Proceedings*, pp. 835-838.
- [15] Y. Yusong et W. Chunmei, "Invertible integer DCT applied on progressive until lossless image compression," *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis (ISPAO3)*, pp. 1018-1023, 2003.
- [16] N. Brahimi et S. Bouguezel, "A fast orthogonal 8×8 integer DCT for image compression," *International Conference on Applied Informatics (ICAI09)*, November 15-17, 2009, Bordj Bou-Arriridj.
- [17] A. Czihó, "Quantification vectorielle et compression d'image: Application à l'imagerie médicale," Thèse Doctorat, Université de Rennes 1, France, 1999.
- [18] D. Salomon, *Data compression: the complete reference*. London, 4^{ième} édition, Springer Verlag London Limited, 2007.
- [19] T. Brahimi, "Compression d'image sans perte par les transformées en ondelettes réversibles," Mémoire de Magistère, Université de Jijel, Jijel, 2003.
- [20] P. Bas, "Compression d'image fixes et de séquences vidéo," Cours ENSERG/INPG, Laboratoire des Images et des Signaux de Grenoble, France.
- [21] S. A. Khayam, "The discrete cosine transform (DCT): theory and application," Department of Electrical & Computer Engineering, Michigan State University, March, 2003.
- [22] W. H. Chen, C. H. Smith et S. C. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Transactions on Communications*, Vol. Com-25, No. 9, pp. 1004-1009, September 1977.
- [23] Z.Wang, "Fast algorithms for the discrete W transform and for the discrete fourier transform," *IEEE Transactions On Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, No. 4, pp. 803-816, August 1984.
- [24] B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, No. 6, pp. 1243-1245, December 1984.
- [25] M. Vetterli et H. J. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations," *Signal Processing (North Holland)*, Vol. 6, No 4, pp. 267-278, Aug 1984.
- [26] N. Suehiro et M. Hatori, "Fast algorithms for the DFT and other sinusoidal transforms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No, 3, pp. 642-644, June 1986.

- [27] C. Loeffler, A. Lieenberg et G. S. Moschytz, "Practical fast 1-d DCT algorithms with 11 multiplications," *IEEE*, pp. 988-991, 1989.
- [28] E. Feig et E. Linzer, "Scaled DCT's on input sizes that are composite," *IEEE Transactions on Signal Processing*, Vol. 43, No. 1, pp.43-50, January 1995.
- [29] International Telecommunication Union, "Information technology–digital compression and coding of continuous-tone still images–requirements and guidelines," The International Telegraph and Telephone Consultative committee (CCITT), Recommendation T.81.
- [30] P.Beaurepaire, "Compression d'images appliquée aux angiographies cardiaques : aspects algorithmiques, evaluation de la qualité diagnostique," thèse doctorat, d'ordre 07 ISAL0107, Lyon, France, 1997.
- [31] H. R. Wu et Z. Man, "Comments on fast algorithms and implementation of 2-D discrete cosine transform," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 2, pp. 128-129, April 1998.
- [32] J. Makhoui, "A fast cosine transform in one and two dimensions," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, No. 1, pp. 27-34, February 1980.
- [33] C. Mayer et M.Wien, "Method for carrying out integer approximation of transform coefficients, and coder and decoder," Feb, 15, 2005.
- [34] A. Aggoun et I. Jalloh, "Two dimensional DCT/IDCT architecture," *IEE Proc.-Comput.Digit.Tech.*, Vol. 150, No.1, pp. 2-10, January 2003.
- [35] L. Cheng, H. Xu et Y. Luo, "Integer discrete cosine transform and its fast Algorithm," *Electronics Letters*, Vol. 37 No. 1, pp.64-65, 4th January 2000.
- [36] D. Hein et N. Ahmed, "On a real-time walsh-hadamard/cosine transform image processor," *IEEE Transactions on Electromagnetic Compatibility*, Vol. EMC-20, No. 3, pp. 453-457, August 1978.
- [37] W. K. Pratt, J. Kane et H.C. Andrews, "Hadamard Transform Image Coding," *Proceedings of The IEEE*, Vol. 57, No. 1, pp. 58-67, January 1969.
- [38] V. Britanak, P. C.Yip et K. R. Rao, *Discrete cosine and sine transforms: general properties, fast algorithms and integer approximations*. Academic press, Elsevier, 1st edition 2007.
- [39] H. W. Jones, D. H. Hein et S. C. Knauer, "The Karhunen- Loève, discret cosine and related transforms obtained via the Hadamard transform," *Proceedings of the International Telemetry Conference*, Los Angeles, pp. 87-98, November 1978.

- [40] W. K. Cham et Y. T. Chan, "Integer discrete cosine transform," *Proceeding of the International Symposium on Signal Processing, Theories, Implementation, and Applications (ISSPA '87)*, Brisbane, Australia, pp.674-676, August 1987.
- [41] W. K. Cham et Y. T. Chan, "An order-16 integer cosine transform," *IEEE Transactions on Signal Processing*, Vol. 39, No. 5, pp.1205-1208, May 1991.
- [42] H. S. Kwak, R. Srinivasan et K. R. RAO, "C-matrix transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-31, No. 5, pp. 1304-1307, October 1983.
- [43] W. Cham, C. Choy et W. Lam, "A 2-D integer cosine transform chip set and its application," *IEEE Transactions on Consumer Electronics*, Vol. 38, No. 2, pp. 43-47, May 1992.
- [44] W. Cham et F. Wu, "On compatibility of order 8 integer cosine transforms and the discrete cosine transform," *IEEE Region 10 Conference on Computer and Communication Systems*, Hong Kong, pp. 447-449, September 1990.
- [45] T. J. Pang, C. O. Choy, C. Chan et W. Cham, "A self-timed ICT chip for image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, No. 6, pp. 856-860, September 1999.
- [46] T. D. Tran, "The BinDCT : Fast multiplierless approximation of the DCT," *IEEE Signal Processing Letters*, Vol. 7, No. 6, pp. 141-144, June 2000.
- [47] T. D. Tran, "A fast multiplierless block transform for image and video compression," *IEEE*, pp. 822-826, 1999.
- [48] J. Liang et T.D. Tran, "Fast multiplierless approximation of the DCT with the lifting scheme," *IEEE Transaction on Signal Processing*, pp. 1-12, February 2001.
- [49] K. Lengwehasatit et A. Ortega, "DCT computation based on variable complexity fast approximations," *ICIP'98*, October 1998.
- [50] S. Bouguezel. M. O.Ahmed et M.N.S. Swamy, "A multiplication-free transform for image compression," *In Proc. IEEE Int. Conf. Signals, Circuits and Systems, SCS'08*, Hammamet, Tunisia, Nov. 2008.
- [51] O. Kanoun, S. Mezghani et M. S. Bouhleb, "Compression adaptative des images médicales modalité "Thorax"," *3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications*, SETIT 2005, March 27-31, 2005, Tunisia.
- [52] S. Nadarajah, "Gaussian DCT coefficient models," *Acta Appl Math*, 2009, pp. 455-472.

- [53] R. Krupinski et J. Purczynski, "Modeling the distribution of DCT coefficients for JPEG reconstruction," *Signal Processing: Image Communication* 22, pp. 439-447, 2007.
- [54] E. Y. Lam et J. W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Transactions on Image Processing*, Vol. 9, No. 10, pp. 1661-1666, October 2000.
- [55] J. Li, M. Gabbouj, J. Takala et H. Chen, "Laplacian modeling of DCT coefficients for real-time encoding," *IEEE ICME*, pp. 797-800, 2008.
- [56] R. Grisel et F. Astrade, "Compression d'images par modélisation des coefficients TCD en lois mélange et quantification adaptative dans l'environnement JPEG," *Traitement du Signal*, Vol. 14, No. 3, pp. 301-315, 1997.
- [57] T. Eude, R. Grkel, H. Cheriji et R. Debrie, "On the distribution of the DCT coefficients," *IEEE*, pp. 365-368, 1994.
- [58] M. Bhaskaranand et J. D. Gibson, "Distributions of 3D DCT coefficients for video," *IEEE, ICASSP*, pp.793-796, 2009.
- [59] H. Lee, Y. Kim, A. H. Rowberg et E. A. Riskin, "Statistical distributions of DCT coefficients and their application to an interframe compression algorithm for 3-D medical images," *IEEE Transactions on Medical Imaging*, Vol. 12, No. 3, pp. 478-485, September 1993.
- [60] Y. Altunbasak et N. Kamaci, "An Analysis of the DCT coefficient distribution with the H.264 video coder," *ICASSP 2009, IEEE*, pp. 177-180.
- [61] S. R. Smoot et L. A. Rowe, "Study of DCT coefficient distribution," In *Proceedings of the SPIE Symposium on Electronic Imaging*, Vol. 2657, 1996.
- [62] T. Eude, H. Cherd et R. Grisel, "Statistical distribution of DCT coefficients and Their application to an adaptative compression algorithm," *TENCON '94, IEEE Region 10's Ninth Annual International Conference. Theme: Frontiers of Computer Technoloyg*, Vol.1, pp. 427-430, 1994.
- [63] F. Miiller, "Distribution shape of two-dimensional DCT coefficients of natural images," *Electronics Letters*, Vol. 29, No. 22, pp. 1935- 1936, 28th October 1993.
- [64] I. Ming Pao et Ming-Ting Sun, "Modeling DCT Coefficients for Fast Video Encoding," *IEEE Transactions on Circuits a Systems for Video Technology*, Vol. 9, No.4, pp. 608-616, June 1999.
- [65] P. H. Kvam et B. Vidakovic, *Nonparametric Statistics with Applications to Science and Engineering*. Canada : A John Wiley & Sons, Inc., 2007.

Résumé

Ce travail considère la compression des images fixes aux niveaux de gris par les DCT entières. Nous effectuons en premier lieu une étude bibliographique sur les techniques de compression d'images ainsi que sur cette catégorie de transformées et leurs méthodes de développement. Par une exploitation efficace de ces méthodes, nous développons deux nouvelles transformées entières et afin de les rendre attractives pour les applications en temps réel, des algorithmes rapides pour leurs calculs et des structures simples pour leurs implémentations sont aussi développés. Pour améliorer d'avantage les performances de ces transformées entières en termes de qualité de compression, nous proposons des nouvelles matrices de quantification appropriées pour les transformées proposées ainsi que pour quelques transformées existantes. Ces matrices de quantification présentent une amélioration supérieure à 0.5 dB en termes de PSNR par rapport aux résultats obtenus par la matrice de quantification adoptée par JPEG. Ce travail introduit aussi une modélisation statistique des coefficients des transformées entières proposées et existantes. Cette modélisation peut être exploitée pour développer des matrices de quantifications optimales pour chaque transformée entière. Les résultats expérimentaux obtenus en implémentant les transformées entières proposées et existantes dans le standard de compression JPEG montrent clairement l'efficacité et les performances des transformées et des matrices de quantification proposées.

Mots clés : compression d'images fixes, JPEG, DCT entières, matrice de quantification, modélisation statistique.

Abstract

This work considers the compression of gray scale still images using integer discrete cosine transforms. We firstly give the state of the art of the image compression techniques as well as of this category of transforms and their designing methods. By an efficient exploitation of these methods, we develop two novel integer transforms and in order to make them more attractive for real-time applications, we also develop fast algorithms for their computation and simple structures for their implementation. In order to further improve these integer transforms in terms of compression quality, we propose new quantization matrices more appropriate for the proposed transforms as well as for some existing transforms. These quantization matrices improve the PSNR by more than 0.5 dB compared to that obtained using the quantization matrix adopted by JPEG. This work introduces also a statistical modeling of the integer transform coefficients. This modeling can be explored to develop optimal quantization matrices for each integer transform. The experimental results obtained by implementing the proposed and existing integer transforms in the baseline standard JPEG show clearly the efficiency and performance of the proposed integer transforms and quantization matrices.

Keywords : Still image compression, JPEG, integer DCT, quantization matrix, statistical modeling.

ملخص

يتناول هذا العمل ضغط الصور الثابتة ذات التدرج الرمادي باستعمال تحويلات DCT ذات عناصر صحيحة. في المقام الأول قمنا بدراسة بيبلوغرافية لأساليب ضغط الصور وكذلك لهذا النوع من التحويلات وطرق تطويرها. بالاستغلال الفعال لهذه الطرق قمنا بتصميم تحويلين جديدين بعناصر صحيحة ولجعلهما ملائمة للتطبيقات الجد سريعة قمنا بانشاء خوارزميات سريعة لإجراء العمليات الحسابية وكذلك هياكل بسيطة لتنفيذها. لتحسين نتائج هذ التحويلات الصحيحة من حيث جودة الضغط، اقترحنا مصفوفات تكميم جديدة مخصصة للتحويلين المقترحين وكذلك لبعض التحويلات الموجودة. مصفوفات التكميم هذه تظهر تحسينا بنسبة 0.5 ديسيبل من حيث PSNR بالنسبة للنتائج المتحصل عليها باستخدام مصفوفة التكميم المعتمدة من طرف JPEG. هذا العمل يتناول أيضا نموذج إحصائي لمعاملات التحويلين المقترحين والتحويلات الصحيحة الموجودة. هذا النموذج الإحصائي يمكن استغلاله لتطوير مصفوفات التكميم المثالي لكل تحويل DCT صحيح. النتائج التجريبية المتحصل عليها من خلال تنفيذ التحويلات المقترحة والموجودة باستعمال نموذج الضغط JPEG تبين بوضوح فعالية وكفاءة التحويلين ومصفوفات التكميم المقترحة.

كلمات مفتاحية : ضغط الصور الثابتة، JPEG، DCT الصحيحة، مصفوفة التكميم، النموذج الإحصائي.