



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>



NANCY 1
FACULTE DES SCIENCES & TECHNIQUES



UFR Sciences et Techniques Mathématiques Informatique Automatique
École Doctorale IAEM - Lorraine
DFD Automatique

N° attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--

Thèse

présentée pour l'obtention du

Doctorat de l'Université Henri Poincaré, Nancy 1

Spécialité Automatique, Traitement du Signal et Génie Informatique

par

Teddy Voinson

Quantification Vectorielle Algébrique avec Zone Morte : application à la compression d'images à bas débit et au tatouage d'images.

soutenue publiquement le 7 février 2003.

Composition du jury :

Président :	M. Alain Richard	Professeur à l'UHP Nancy 1
Rapporteurs :	Mme Christine Fernandez	Professeur à l'Université de Poitiers
	M. Atilla Baskurt	Professeur à l'Université C. Bernard, Lyon 1
Examineurs :	M. Franck Davoine	Chargé de Recherche au laboratoire Heudyasic
	M. Samuel Nowakowski	Maître de Conférences HdR à l'UHP Nancy 1 (Directeur)
	M. Jean-Marie Moureaux	Maître de Conférences à l'UHP Nancy 1

*A mes parents,
à ma soeur.*

7 juillet 2003

Table des matières

Résumé	xi
Abstract	xiii
Remerciements	xv
Introduction générale	1
1 La compression d'images par transformée en ondelettes	3
1.1 Introduction	3
1.2 Mesures du débit, de la qualité et de la complexité	4
1.2.1 Débit et taux de compression	4
1.2.2 Qualité : calcul de distorsion	5
1.2.3 Complexité : coût calculatoire, coût mémoire	5
1.2.4 Compromis : taux de compression/ qualité/ complexité	6
1.3 La transformée en ondelettes	6
1.3.1 Analyse multirésolution	7
1.3.2 Distribution des coefficients	7
1.4 Méthodes de compression	10
1.4.1 Chaînes de compression avec ou sans perte	10
1.4.2 Méthodes sans pertes	10
1.4.3 Méthodes avec pertes	11
1.4.4 Evolution des méthodes avec pertes	11
1.5 Principes des méthodes	12
1.5.1 Notion de seuillage	12
1.5.2 Approche intra-bande	12
1.5.3 Approche inter-bande	13
1.5.4 Conclusion	13
1.6 Méthode de référence et standard	13
1.6.1 <i>SPIHT</i>	13
1.6.1.1 Approche par "Arbres de zéros"	13
1.6.1.2 Performances	14
1.6.2 <i>JPEG2000</i>	15
1.6.2.1 Introduction	15
1.6.2.2 Description générale	15
1.6.2.3 Performances	17
1.7 Notre schéma de compression	18
1.7.1 Démarche générale	18
1.7.2 Description de notre schéma	18

1.8	Conclusion du chapitre	18
2	La Quantification Vectorielle Algébrique avec Zone Morte	21
2.1	Introduction	21
2.2	La quantification	21
2.3	La Quantification Scalaire	21
2.3.1	Définitions	21
2.3.2	Distorsion de la QS	22
2.3.3	Bruit granulaire et bruit de surcharge	23
2.3.4	Distorsion en fonction du débit : approximation haute résolution	23
2.3.5	La Quantification Scalaire uniforme	24
2.3.6	La Quantification Scalaire non uniforme	24
2.3.7	Conditions d'optimalité : Algorithme de Lloyd-Max	24
2.3.8	Conclusion	25
2.4	La Quantification Scalaire avec Zone Morte	25
2.4.1	Introduction	25
2.4.2	Définitions	26
2.4.3	Optimisation de la Quantification Scalaire avec Zone Morte	26
2.4.4	Conclusion	28
2.5	La Quantification Vectorielle	28
2.5.1	Définitions	28
2.5.2	Différents types de partitionnement	29
2.5.3	Distorsion de la QV	30
2.5.4	Méthodes par apprentissage	30
2.5.4.1	L'algorithme <i>GLA</i>	30
2.5.4.2	L'algorithme <i>LBG</i>	31
2.5.4.3	Autres méthodes	31
2.5.5	Complexité au codage et au décodage	31
2.5.6	Conclusion	32
2.6	La Quantification Vectorielle Algébrique	32
2.6.1	Introduction	32
2.6.2	Réseau régulier de points	32
2.6.3	Troncature du réseau $\mathbb{Z}^n$	33
2.6.4	Normalisation et quantification sur $\mathbb{Z}^n$	33
2.6.5	Algorithme de la QVA	34
2.6.6	Méthodes d'indexage	36
2.6.7	Distorsion de la QVA	36
2.6.8	Compromis : taux de compression/ qualité/ complexité	37
2.6.9	Conclusion	37
2.7	La Quantification Vectorielle Algébrique avec Zone Morte	37
2.7.1	Définitions	37
2.7.2	Principe général	38
2.7.3	Forme de la zone morte	38
2.7.4	Quantification des vecteurs appartenant à la zone morte	38
2.7.5	Algorithme de la QVAZM	40
2.7.6	Compromis : taux de compression / qualité / complexité	40
2.7.7	Conclusion	40
2.8	Modèle du débit de la QVAZM pour une source laplacienne	40
2.8.1	Modèle du débit entropique de la QVA	40
2.8.2	Modèle du débit entropique de la QVAZM	41

2.8.3	Modèle du débit R_e^{ZM}	42
2.8.4	Modèle du débit $R_e^{dictionnaire}$	44
2.8.5	Modèle du débit $R_e^{surcharge}$	45
2.8.6	Simulation de $p_{ZM}(r)$	46
2.8.7	Simulation du modèle $R_{ZM}(\gamma, r_{ZM})$	46
2.9	Résultats sur les différentes sous-bandes	47
2.9.1	Performances	47
2.9.2	Intérêt du modèle	47
2.10	Conclusion du chapitre	50
3	Allocation optimale des débits associée à la QVAZM et au codage <i>Stack-Run</i>	51
3.1	Introduction	51
3.2	Description générale	51
3.3	Allocation optimale de débit associée à la QVA	52
3.3.1	Définitions	52
3.3.2	Hypothèses	53
3.3.2.1	Indépendance des sous-bandes	53
3.3.2.2	Orthogonalité de la décomposition	54
3.3.2.3	Convexité	54
3.3.3	Formalisation du problème	54
3.3.4	Algorithme d'allocation des débits associée à la QVA	55
3.4	Allocation des débits associée à la QVAZM	56
3.4.1	Objectif	56
3.4.2	Problème : une optimisation conjointe	57
3.4.3	Formalisation du problème	57
3.4.4	Description de l'algorithme proposé	59
3.4.5	Conclusion	61
3.5	Codage <i>Stack-Run</i> adapté aux plages d'énergies nulles	61
3.5.1	Adaptation aux plages d'énergies nulles	61
3.5.2	Description du codage <i>Stack-Run</i> adapté	61
3.5.3	Amélioration de la forme des vecteurs	62
3.5.4	Performances et conclusions	63
3.6	Résultats : reconstruction de l'image	63
3.6.1	Rappels	63
3.6.2	Images Naturelles	63
3.6.3	Qualité visuelle	64
3.7	Conclusion du chapitre	68
4	Compression par QVAZM et tatouage d'images	85
4.1	Introduction	85
4.2	Généralités	85
4.2.1	Terminologie	85
4.2.2	Les attaques dans le cas des images	87
4.2.2.1	Les traitements innocents	87
4.2.2.2	Les traitements malveillants	88
4.2.3	Propriétés du tatouage d'images	88
4.2.4	Classification des schémas de tatouage	88
4.3	L'analyse multirésolution pour le tatouage d'images	89
4.3.1	Domaines d'insertions	89
4.3.2	Insertion et détection dans le domaine multirésolution	89

4.3.2.1	L'insertion	89
4.3.2.2	La détection	90
4.3.3	Conclusion	91
4.4	Compression et tatouage d'images conjoints	91
4.4.1	Introduction	91
4.4.2	Motivations	91
4.4.3	QVAZM, insertion et détection	92
4.4.3.1	L'opération d'insertion	92
4.4.3.2	L'opération de détection	93
4.4.4	Langage utilisé	94
4.4.4.1	Propriétés du langage	94
4.4.4.2	Description du langage	95
4.4.5	Algorithme d'insertion	96
4.4.6	Algorithme de détection	96
4.4.7	Conclusion	98
4.5	Robustesse face aux attaques de type bruit additif gaussien	98
4.5.1	Etude scalaire sans zone morte	98
4.5.2	Etude vectorielle avec zone morte	100
4.6	Résultats	103
4.6.1	Imperceptibilité	103
4.6.1.1	Sans zone morte	103
4.6.1.2	Avec zone morte	103
4.6.2	Robustesse face aux attaques	108
4.6.2.1	Robustesse de la QVA sans zone morte	108
4.6.2.1.1	Bruit additif gaussien :	108
4.6.2.1.2	Re-compression <i>JPEG</i> :	108
4.6.2.1.3	Filtre médian :	108
4.6.2.2	Robustesse de la QVA avec zone morte	113
4.6.2.2.1	Bruit additif gaussien :	113
4.6.2.2.2	Re-compression <i>JPEG</i> :	114
4.6.2.2.3	Filtre médian :	114
4.7	Conclusion du chapitre	119
5	Conclusion générale	121
A	Fonction de répartition $F(r)$	123
A.1	Problématique	123
A.2	Démonstration	123
B	Calcul du $Card(r, n_0)$	125
B.1	Aspect théorique	125
B.2	Démonstration	125
	Glossaire	127
	Notations	129
	Bibliographie	131
	Index	137

Table des figures

1	Chaîne de compression simplifiée.	2
1.1	Image artificielle originale de dimension 512×512 pixels.	6
1.2	Visualisation des détails préservés sur les différentes sous-bandes (horizontaux, verticaux, diagonaux) pour une image artificielle.	7
1.3	Image <i>Lena</i>	8
1.4	Histogramme de l'image <i>Lena</i>	8
1.5	(a) Coefficients de la sous-bande V1. (b) Distribution des coefficients de la sous-bande V1 et courbe théorique d'une source laplacienne centrée et d'écart-type $\sigma_{V1} = 2,5$	9
1.6	Décomposition en sous-bandes de l'image naturelle <i>Lena</i>	9
1.7	Principales étapes d'une chaîne de compression avec perte.	10
1.8	Méthodes utilisées par les principaux algorithmes de compression d'images. En gris, notre algorithme. Comparaison du <i>PSNR</i> pour la compression de l'image <i>Lena</i> à un débit de 0,125 bpp ($TC = 64 : 1$).	13
1.9	Structure arborescente de <i>SPIHT</i>	14
1.10	Exemple de compression avec un codage imbriqué.	15
1.11	Diagramme simplifié du standard <i>JPEG2000</i> : les images sont associées au parcours flêché en trait continu (l'image qui montre le découpage en "tuiles" est mise à titre indicatif).	16
1.12	Codage d'un "code-bloc" de coefficients d'ondelettes.	17
1.13	Chaîne de compression d'une image	19
2.1	Distorsion en fonction du débit de la borne de Shannon, $D(R) = 2^{-2R}$	24
2.2	Quantification Scalaire uniforme	24
2.3	Quantificateur Scalaire non uniforme	25
2.4	Quantification Scalaire uniforme avec Zone Morte.	27
2.5	Partitionnement irrégulier (à gauche) et régulier (à droite) sur $\mathbb{Z}^2$. Chaque vecteur X appartenant à une région P_i est quantifié par Y_i	29
2.6	Exemple de réseau de dimension $n = 2$ engendré par la matrice génératrice $G = [a_1 \ a_2]^T$, avec $a_1 = (1, 1)$ et $a_2 = (2, 0)$	33
2.7	Troncature pyramidale (à gauche) et sphérique (à droite) pour une meilleure adaptation à des sources respectivement laplacienne et gaussienne.	34
2.8	Processus de normalisation et de quantification sur $\mathbb{Z}^2$ de la QVA pour un vecteur à l'extérieur du dictionnaire (en haut), et un vecteur à l'intérieur du dictionnaire (en bas).	35
2.9	Dictionnaire pyramidal sur un réseau $\mathbb{Z}^2$ avec zone morte.	39
2.10	Zoom sur le dictionnaire.	39
2.11	Densité de probabilité du rayon $f_{\ \mathbf{x}\ _1}(r)$ d'une source laplacienne de moyenne nulle et d'écart-type $\sigma = 1$, pour des vecteurs de dimension $n = 8$	43

2.12	Fonction de répartition $F_{\ \mathbf{x}\ _1}(r)$ dans le cas d'une source laplacienne de moyenne nulle et d'écart-type $\sigma = 1$, pour des vecteurs de dimension $n = 8$	44
2.13	Description de la loi du rayon de la QVA avec $n = 8$, sans zone morte ($\delta = 0$) et $\gamma = 1$ pour une source laplacienne de moyenne nulle et d'écart-type $\sigma = 1$	46
2.14	Description de la loi du rayon de la QVA avec $n = 8$, avec zone morte ($\delta = 4$) et $\gamma = 1$ pour une source laplacienne de moyenne nulle et d'écart-type $\sigma = 1$	47
2.15	Evolution théorique du débit $R_{ZM}(\gamma, r_{ZM})$ en fonction des couples (γ, r_{ZM}) pour une source laplacienne (moyenne nulle, $\sigma = 1$). L'axe vertical correspond aux différents débits, l'axe de gauche à l'évolution du facteur de projection γ et l'axe de droite à l'évolution du rayon de la zone morte r_{ZM}	48
2.16	Evolution des couples (γ, r_{ZM}) pour une source laplacienne (moyenne nulle, $\sigma = 1$) et pour un débit fixé ($R = 1,0$ bpp). Courbe théorique en continu, et courbe pratique en continu gras.	50
3.1	Chaîne de compression d'une image	52
3.2	Représentation des sous-bandes SB_k d'une transformée en ondelettes sur $J = 3$ niveaux de décomposition. Chaque sous-bande SB_k a son propre débit R_k et sa propre distorsion D_k	53
3.3	Courbes $D_T(R_T)$ et $w(\lambda)$ pour l'allocation optimale des débits.	56
3.4	(a) Distorsion en fonction du débit sur une sous-bande SB_k avec et sans zone morte. (b) Couple optimal (γ_k^*, r_{ZMk}^*) qui permet d'atteindre le débit optimal $R_k^* = \hat{R}_k$ avec la distorsion minimale $D_k^* < \hat{D}_k$	57
3.5	Simulation des courbes de $D_k(\gamma_k^*, r_{ZMk}^*)$ des sous-bandes SB_k de 6 à 11 ($J = 5$) de l'image <i>Lena</i> pour un débit cible $R_C = 0,125$ bpp avec la QVAZM.	58
3.6	Description des courbes $R_k(\gamma_k, r_{ZMk})$ et $D_k(\gamma_k^*, r_{ZMk}^*)$ pour la recherche du couple optimal (γ_k^*, r_{ZMk}^*) d'une sous-bande SB_k	59
3.7	Description de l'algorithme général de l'optimisation des couples (γ_k^*, r_{ZMk}^*) pour chaque sous-bande SB_k	59
3.8	Comparaison des performances des différentes méthodes (<i>PSNR</i> en fonction du débit) : <i>JPEG2000</i> , <i>SPIHT</i> , QVA + arithmétique, QVAZM + <i>Stack-Run</i> (binaire) appliquées à l'image <i>Lena</i>	65
3.9	Zoom sur le chapeau de <i>Lena</i> pour un TC = 64 : 1 (0,125 bpp), images de gauche à droite : originale, <i>SPIHT</i> , <i>JPEG2000</i> , QVAZM.	66
3.10	Zoom sur le chapeau de <i>Lena</i> pour un TC = 103 : 1 (0,0777 bpp), images de gauche à droite : originale, <i>SPIHT</i> , <i>JPEG2000</i> , QVAZM.	66
3.11	Zoom sur les yeux de <i>Lena</i> pour un TC = 64 : 1 (0,125 bpp), images du haut : originale, QVAZM, images du bas : <i>JPEG2000</i> , <i>SPIHT</i>	67
3.12	Zoom sur une zone texturée de l'image <i>Mandrill</i> , TC $\simeq$ 64 : 1 (0,122 bpp), images de gauche à droite : originale, <i>SPIHT</i> , <i>JPEG2000</i> , QVAZM.	67
4.1	Schéma général de la transmission d'un message dissimulé. (a) Insertion d'un message dans le médium, (b) Extraction du message à partir du stégo-médium.	86
4.2	Répartition des trois critères pour différentes approches.	87
4.3	Schéma général d'insertion d'une signature dans une image.	90
4.4	Schéma général de détection d'une signature à partir d'une image tatouée et attaquée.	91
4.5	Schéma de compression et tatouage conjoints (opération d'insertion de la signature).	93
4.6	Schéma de détection de la signature.	94
4.7	Schéma des différents cas de figure lors de la détection de la signature.	97
4.8	Nous cherchons l'écart-type σ maximum du bruit $\mathbf{b}_i$ pour avoir $P(\mathbf{b}_i > \frac{\gamma}{2}) = \epsilon_d$	99
4.9	Visualisation du bruit gaussien sur un réseau $\mathbb{Z}^2$	101

Liste des tableaux

2.1	Comparaison des performances QVA/ QVAZM sur les sous-bandes de l'image <i>Lena</i> .	49
2.2	Comparaison des performances QVA/ QVAZM sur l'image <i>Lena</i> reconstruite.	49
3.1	Exemple de codage <i>Stack-Run</i> adapté	63
3.2	Comparaison des méthodes pour l'image <i>Lena</i> à 0,125 bpp	64
3.3	Comparaison des résultats sur différentes images	64
3.4	Comparaison des résultats sur d'autres images	65
4.1	<i>PSNR</i> de l'image <i>Lena</i> sans zone morte pour différentes longueurs de message. . . .	103
4.2	<i>PSNR</i> de l'image <i>Lena</i> avec zone morte pour différentes longueurs de message. . . .	103
4.3	Caractéristiques des sous-bandes 6, 7 et 8 du niveau $j=3$ sans zone morte	108
4.4	Pourcentage d'erreur de lecture de l'image <i>Lena</i> sans zone morte avec une recompression <i>JPEG</i>	108
4.5	Pourcentage d'erreur de lecture de l'image <i>Lena</i> sans zone morte pour une attaque de type filtre médian.	109
4.6	Caractéristiques des sous-bandes 6, 7 et 8 du niveau $j=3$ avec zone morte.	113
4.7	Pourcentage d'erreur de lecture de l'image <i>Lena</i> avec zone morte avec une recompression <i>JPEG</i>	114
4.8	Pourcentage d'erreur de lecture de l'image <i>Lena</i> avec zone morte pour une attaque de type filtre médian.	114

Résumé

Etant donné les volumes actuels de données numériques, la compression est devenue une phase incontournable des applications mettant en œuvre la transmission ou le stockage d'images numériques. Dans ce contexte, la chaîne de compression basée sur une transformée en ondelettes des données associée à une quantification et un codage entropique fait aujourd'hui référence. Le travail présenté dans ce mémoire concerne les étapes de quantification et de codage de cette chaîne. Généralement la quantification scalaire (QS) est préférée à la quantification vectorielle (QV) pour des raisons de complexité calculatoire, en dépit de la supériorité de la QV démontrée par la théorie de l'Information, en termes de qualité d'image comprimée et de taux de compression.

Nous proposons ici la mise en œuvre d'un quantificateur vectoriel algébrique (QVA) alliant les principaux avantages des approches vectorielles et un faible coût de calcul. L'originalité de la méthode réside dans la définition d'une zone morte vectorielle associée au QVA, de forme identique à celle du dictionnaire, qui permet d'effectuer un seuillage par énergie des vecteurs de la source. En d'autres termes, les blocs de coefficients d'ondelettes jugés non significatifs pour la qualité visuelle sont mis à zéro par l'intermédiaire du seuillage. Ceci permet de coder plus précisément les blocs de coefficients significatifs. Ainsi, à taux de compression constant, il est possible de diminuer la distorsion dans chacune des sous-bandes de la transformée en ondelettes, et donc la distorsion globale de l'image comprimée. En outre, un modèle théorique pour le calcul du débit binaire (i.e. du taux de compression) a été mis au point pour accélérer le réglage du paramètre de forme optimal de la zone morte. Par ailleurs, la redondance engendrée par la mise à zéro de nombreux blocs de coefficients a été exploitée dans l'étape de codage à travers un codeur *Stack-Run*, codeur par plages de zéros dont nous avons modifié la structure pour l'adapter à notre chaîne de compression. Les résultats obtenus ont été comparés avec ceux des algorithmes de référence actuels, à savoir *JPEG2000* et *SPIHT*. Ils sont encourageants et montrent de bonnes performances dans les bas débits, se traduisant par une meilleure préservation de nombreux détails de l'image.

Dans un tout autre registre, le développement des communications et notamment des échanges de données numériques a mis en lumière un problème de protection de la propriété intellectuelle de ce type de données. C'est dans ce contexte que s'est développé le tatouage, qui vise à insérer dans une image, un message invisible et indélébile, à l'aide d'une clé secrète, que seul un utilisateur possédant la clé peut extraire. Une des difficultés majeures du tatouage tient dans la résistance aux attaques, qu'elles viennent de pirates, ou de traitements bienveillants comme la compression. Celle-ci est en effet le dual du tatouage, puisqu'elle vise à retirer toute information invisible dans l'image. C'est dans ce contexte que la QVA avec zone morte vectorielle a été également utilisée à profit dans une application de tatouage et compression conjoints, permettant d'améliorer la robustesse du tatouage face à certaines attaques et notamment celle d'un bruit additif gaussien, comme en témoignent les résultats expérimentaux obtenus.

Mots-clefs : compression d'images, Quantification Vectorielle Algébrique (QVA), transformée en ondelettes, "Zone Morte Vectorielle", codage *Stack-Run*, tatouage d'images.

Abstract

Given the present great amount of numerical data, compression has become an unavoidable step for the transmission or storage applications of numerical images. In this context, the compression chain based on data discrete wavelet transform, associated to quantification and entropy coding, is today a reference. Our work concerns the quantization and coding steps of that chain. Generally we prefer to use scalar quantization rather than vector quantization for computation complexity reasons, in spite of the VQ superiority proved by the information theory, in terms of compression ratio and compressed image quality. We propose to use Lattice Vector Quantization (LVQ), which combines the main advantages of vector approaches and a low complexity.

The originality of our method consists in defining a vectorial dead zone, adapted to codebook shape and associated with LVQ. That dead zone enables us to carry out energy thresholding of source vectors. In other words, blocks of wavelet coefficients that are considered to be “non significant” for the visual quality are reset to zero by use of thresholding. It permits to code with more precision the significant coefficient blocks. Therefore, with a constant compression ratio, it is possible to decrease the distortion in each discrete wavelet transform subband, and thus the global distortion of the compressed image. Furthermore, a theoretical rate (i.e. compression ratio) model has been developed, in order to accelerate the optimal setting of the dead zone shape parameter. Moreover, the redundancy generated by the reset to zero of numerous coefficient blocks has been used with the Stack-Run coder. The structure of the latter, which makes use of runs of zeros, has been modified to be adapted to our compression chain. Our results have been compared with the current reference algorithms, namely *JPEG2000* and *SPIHT*; they are encouraging, especially at low bit rate, as can be shown by a good preservation of visual image details.

In another field, widespread development of the communications, such as the exchange of numerical data, highlights the problem of copyright and intellectual property protection for the use of those data. In this context, there has been a great development of watermarking, which aims to insert an invisible and indelible message in an image, by use of a secret key; thanks to which only the key owner can extract the message. One of the major difficulties of watermarking is to face out attacks, coming from hackers or unintentional processes as compression. Indeed, compression is the dual of watermarking, since it tries to remove all invisible information in the image. It is in this context that Dead Zone Lattice Vector Quantization has been used very relevantly for both watermarking and compression, permitting to improve the watermarking robustness to some attacks such as an additive gaussian noise, as shown by experimental results.

Key-words : images compression, Lattice Vector Quantization (LVQ), discrete wavelet transform (DWT), “Vector Dead Zone”, *Stack-Run* codage, Watermarking.

Remerciements

Mes travaux ont été effectués au sein du Centre de Recherche en Automatique de Nancy (CRAN : UMR 7039), et s'inscrivent dans la thématique « Instrumentation, Robotique, Images, Signal » (Thème IRIS). Je remercie son directeur Monsieur Francis LEPAGE pour son accueil.

Je tiens à remercier Monsieur Jean-Louis CLERC, Professeur de mathématiques à Université Henri Poincaré, Nancy 1, et Directeur de l'Ecole Doctorale IAEM de Lorraine, pour m'avoir permis de suivre cette formation.

Je remercie vivement, Madame Christine Fernandez, Professeur à l'Université de Poitiers, et membre de l'Institut de Recherche en Communications Optiques et Micro-ondes-Signal Image Communications (UMR 6615), pour avoir bien voulu être rapporteur de ma thèse et pour l'intérêt qu'elle porte à mon travail.

Je suis également très reconnaissant envers Monsieur Atilla Baskurt, Professeur à l'Université Claude Bernard, Lyon 1, et membre du Laboratoire d'Informatique Graphique, Image et Modélisation (LIRIS - LIGIM), pour avoir accepté d'être rapporteur de ma thèse et pour l'attention qu'il porte à mon travail.

Je remercie Monsieur Alain Richard, Professeur à l'Université Henri Poincaré, Nancy 1, et responsable du Département de Formation Doctorale en Automatique, pour son soutien et pour l'honneur qu'il m'a fait en présidant le jury.

Je tiens à exprimer toute ma reconnaissance à Monsieur Franck Davoine, Chargé de Recherche au Laboratoire HEUDIASYC – CNRS (UMR 6599) à l'Université de Technologie de Compiègne pour l'honneur qu'il m'a fait de participer à mon jury de thèse.

Ce travail a été effectué sous la direction de Monsieur Samuel Nowakowski, Maître de Conférences (HDR) en détachement de l'Université Henri Poincaré Nancy 1. Je tiens à le remercier pour les conseils qu'il m'a apportés au cours de ces années de thèse.

Je remercie Monsieur Jean-Marie Moureaux, Maître de Conférences à l'Université Henri Poincaré, Nancy 1, et membre du CRAN, en qui j'ai toujours trouvé un interlocuteur attentif et pour la confiance qu'il m'a témoignée. Je le remercie vivement en particulier pour l'aide précieuse qu'il m'a apportée au cours de ces années. Je lui exprime également toute ma gratitude pour la patience, et la rigueur avec laquelle il a relu ce mémoire.

Je remercie également, Monsieur Thierry Cecchin, Professeur à l'Université Henri Poincaré, Nancy 1, et Directeur de l'IUT de Saint-Dié, pour les structures d'accueil dont il m'a fait bénéficier au cours de ma thèse.

J'adresse également tous mes remerciements à tous ceux qui ont contribué à l'étude et à la rédaction de ce mémoire : Ludovic, Virginie, Aline, Mary et tous les autres.

Introduction générale

"Qui n'a jamais rêvé de pouvoir communiquer et voyager à travers l'espace et le temps?". L'Homme n'a cessé d'imaginer de nouveaux moyens pour y parvenir. La nature et la forme de ces moyens ont évolué avec les avancées technologiques et la mise en place de réseaux mondiaux. Aujourd'hui, il est devenu possible par exemple de communiquer par visioconférence avec différentes personnes dans le monde entier en temps réel, et de stocker de grandes quantités d'informations. Depuis une quinzaine d'années en effet, le développement des techniques de numérisation et la mise en place de nouveaux standards de compression ont permis la multiplication des échanges de données de type images, sons et vidéos. Ce phénomène s'accroît encore aujourd'hui avec la généralisation des réseaux haut débit, y compris chez les particuliers (*ADSL*¹, câble, etc.), et les nouveaux protocoles de communication. De la même façon, les progrès technologiques offrent des supports de stockages (graveurs + CD-Rom) rapides et bon marché.

De nombreux travaux de recherche dans le domaine de la compression de données sont à l'origine de cette explosion des communications. Ils ont apporté des standards aux noms bien connus de *JPEG*² (et aujourd'hui *JPEG2000*) ou encore *MPEG*³. Cependant les échanges sans cesse croissants et les applications sans cesse nouvelles posent fatalement de nouveaux problèmes. Le progrès appelant le progrès, les volumes de données caractérisant les images augmentent constamment, notamment grâce (ou à cause) d'une résolution de plus en plus grande. C'est le cas par exemple en imagerie médicale et c'est le prix à payer pour une exploration de plus en plus précise des organes du corps humain. La compression est donc encore une fois incontournable et doit faire face à la diversité et aux contraintes des applications. Parmi les nouveaux problèmes liés à ces applications, figure également la protection de la propriété intellectuelle des documents numériques qui circulent sur les réseaux. Différents travaux concernant le domaine du tatouage⁴ d'images se sont ainsi développés afin de préserver cette propriété intellectuelle. Ces méthodes permettent d'insérer une information indélébile et imperceptible au sein de l'image à l'aide d'une clé secrète, que seul un utilisateur possédant la clé pourra extraire. Là encore se pose le problème de la compression qui doit coexister avec le tatouage, alors que les deux opérations sont duales.

Dans le cadre de cette thèse, nos travaux se sont orientés vers une méthode de compression d'images à bas débit. Nous l'avons également étudiée à travers une application de compression et tatouage conjoints. Nous proposons dans ce mémoire, une chaîne de compression dont les caractéristiques permettent d'atteindre des taux de compression importants tout en préservant une bonne qualité visuelle de l'image.

Nous présentons sur la figure 1 notre chaîne de compression simplifiée. La première étape consiste à effectuer une "décorrélation" des données de l'image à l'aide de la transformée en ondelettes. Ce passage dans un domaine transformé permet d'adapter plus efficacement la méthode de compression en aval. L'étape de Quantification permet ensuite de réduire la quantité d'information à

¹ *Asymetric Digital Subscriber Line (ADSL)* en anglais.

² *Joint Picture Expert Group* en anglais.

³ *Motion Picture Expert Group* en anglais.

⁴ *Watermarking* en anglais.

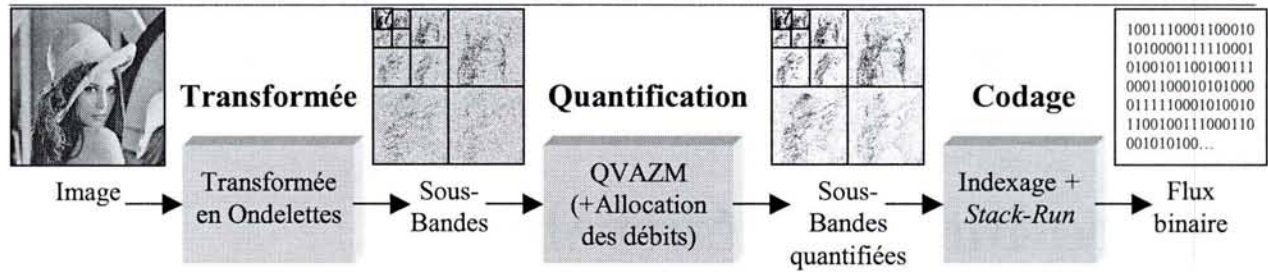


FIG. 1: Chaîne de compression simplifiée.

stocker, au prix cependant d'une certaine dégradation de l'information (compression avec perte). C'est à ce niveau que nous avons amélioré les performances de la Quantification Vectorielle Algébrique classique, avec la création d'une zone morte vectorielle. Cette approche, appelée Quantification Vectorielle Algébrique avec Zone Morte (QVAZM), a été associée à un processus d'allocation optimale des débits. Enfin, l'étape de codage (étape sans perte) consiste à coder efficacement les index des données quantifiées, afin de générer le flux final des données compressées. Compte tenu des redondances engendrées par la création de la zone morte vectorielle, nous avons proposé une version adaptée à nos données du codeur de type Stack-Run.

Ce mémoire s'articule autour de quatre chapitres. Dans le chapitre 1, nous présentons brièvement les outils mathématiques nécessaires à la compression d'image, ainsi que les principales méthodes et standards de compression. Le chapitre 2 est consacré à la description théorique de la Quantification Vectorielle Algébrique avec Zone Morte. Nous présentons en particulier un modèle théorique du débit entropique dans le cas d'une source Laplacienne (modèle bien adapté généralement à la distribution des coefficients d'ondelettes). A travers le chapitre 3, nous exposons l'algorithme d'allocation optimale des débits associée à la Quantification Vectorielle Algébrique avec Zone Morte et au codage Stack-Run. Enfin, le chapitre 4 est consacré à la présentation de la Compression et du Tatouage d'images conjoints, utilisant la QVAZM. Nous y développons en particulier les aspects concernant la robustesse de notre schéma de tatouage, face notamment à des attaques de type bruit additif gaussien.

Chapitre 1

La compression d'images par transformée en ondelettes

1.1 Introduction

Comprimer une image consiste à réduire la quantité d'information qu'elle contient avant une transmission ou un stockage, et ce, "sans dégrader la qualité de l'image" (compression sans pertes) ou "le moins possible" (compression avec pertes). Ces objectifs ne peuvent être remplis qu'en prenant en considération différents types de redondances existant dans l'image (dans le domaine spatial ou fréquentiel). A l'origine, les méthodes de compression traitaient directement la valeur des pixels de l'image. Cependant, le passage dans un domaine transformé au sein de la chaîne de compression a permis d'améliorer de manière considérable le compromis : quantité d'information/ qualité de l'image. En effet, ce passage permet de décorréliser presque totalement l'information originale, et ainsi d'appliquer des méthodes de codage bien adaptées. Différents types de transformées ont été proposés, ne permettant pas toujours de localiser l'énergie du signal à la fois dans le domaine spatial et dans l'espace fréquentiel. Ainsi, de nombreux travaux se sont basés sur une transformée en cosinus discrète¹, utilisée sur des blocs 8 par 8 pixels par l'ancien standard *JPEG*. A l'heure actuelle, l'utilisation de la transformée en ondelettes discrète² permet d'obtenir de bonnes performances en termes de complexité (utilisation de filtres avec peu de coefficients) et de localisation spatiale et fréquentielle de l'énergie des signaux. Cette transformée a d'ailleurs été retenue dans le nouveau standard *JPEG2000* [86][87]. Nous détaillerons ses caractéristiques dans la suite de ce chapitre. Cette décomposition génère un ensemble de sous-bandes qui peuvent être exploitées de différentes manières. Dans un cas, les sous-bandes sont considérées indépendantes, les méthodes associées sont appelées intra-bandes. Elles nécessitent de mettre en place différents algorithmes d'optimisation des débits. Dans l'autre cas, les sous-bandes sont utilisées conjointement. Certaines méthodes exploitent alors les corrélations inter-bandes. Ce type d'approche est à la base des algorithmes *EZW*³ [83] et *SPIHT*⁴ [78], qui sont des algorithmes de référence en terme de compromis taux de compression/ qualité/ complexité. Cependant, ce type d'approche oblige à connaître ou à disposer d'informations contenues dans les différentes sous-bandes.

Nos travaux se sont plutôt orientés vers un schéma de compression intra-bandes qui exploite au maximum les redondances dans chaque sous-bande. Cette approche nous permet d'obtenir un très bon compromis taux de compression/ qualité/ complexité. Nous définissons au paragraphe suivant ces différents critères, puis à travers les autres paragraphes, nous présentons brièvement les principales étapes de la chaîne de compression, à savoir : la transformée en ondelettes, certaines méthodes

¹ *Discrete Cosine Transform (DCT)* en anglais.

² *DWT : Discrete Wavelet Transform* en anglais

³ *Embedded Zerotree Wavelet (EZW)*.

⁴ *Set Partitioning In Hierarchical Trees (SPIHT)*.

de compression dont un algorithme de référence (*SPIHT*) ainsi que le nouveau standard *JPEG2000*. Enfin, nous présentons la démarche et les caractéristiques de notre chaîne de compression d'images.

1.2 Mesures du débit, de la qualité et de la complexité

Une image numérique monochrome I est un signal numérique à deux dimensions [81]. Par simplification, elle peut être représentée par un ensemble fini de pixels $x(i)$:

$$I = \{x(i) \mid i = \{0, \dots, M-1\}\} \quad (1.1)$$

où M correspond au nombre total de pixels de l'image (ou d'échantillons).

Chaque pixel est caractérisé par un niveau de gris (une intensité lumineuse) associé à une valeur numérique. Nous pouvons ainsi considérer la valeur d'un pixel d'une image numérique comme la réalisation d'une variable aléatoire $\mathbf{x}$ dont les valeurs sont prises dans l'alphabet $\{x_0, \dots, x_m, \dots, x_{M-1}\}$ où M correspond au nombre de niveaux de gris que peuvent prendre chaque pixel $x(i)$:

$$x(i) = x_m \quad m \in \{0, \dots, M-1\} \quad (1.2)$$

Le nombre de niveaux de gris M est généralement choisi comme une puissance de deux (représentation en binaire) :

$$M = 2^b \quad (1.3)$$

où b est le nombre de bits de codage.

La majorité des images naturelles monochromes est représentée par $M = 256$ niveaux de gris x_m variant de 0 à 255, codés en binaire sur $b = 8$ bits. Nous définissons ainsi le nombre de bits par pixel $R_O = \log_2(M) = 8$ bpp⁵, qui correspond au débit original de l'image. L'objectif d'un schéma de compression est de réduire ce débit en préservant intégralement ou partiellement les niveaux de gris de l'image.

1.2.1 Débit et taux de compression

Le débit permet d'estimer le nombre de bits moyen utilisé pour coder un élément. Cet élément peut être un pixel ou un groupement de pixels. Suivant les méthodes de codage, le débit n'est pas toujours un entier. L'objectif est d'obtenir un débit cible R_C , tel que $R_C < R_O$ (par exemple, pour une même qualité visuelle de l'image). Pour déterminer le degré de compression obtenu, on définit également : le taux de compression (TC) :

$$TC = \frac{\text{Nombre de bits de l'image originale}}{\text{Nombre de bits de l'image comprimée}} = \frac{R_O}{R_C} \quad (1.4)$$

Lorsque l'on souhaite préserver intégralement la qualité de l'image (compression sans perte), le calcul du débit entropique R_E permet de connaître le débit minimal que l'on peut atteindre sans dégrader l'information. Il est défini par :

$$R_E = H_0 = - \sum_{m=0}^{M-1} p(x_m) \log_2(p(x_m)) \quad (1.5)$$

où $p(x_m)$ correspond à la probabilité d'apparition du niveau de gris x_m dans l'image. L'entropie H_0 (entropie d'ordre zéro) est valide pour une source discrète sans mémoire (suite de réalisations de variables aléatoires indépendantes et identiquement distribuées).

⁵bpp : abréviation de "bits par pixel"

1.2.2 Qualité : calcul de distorsion

Lorsque l'on applique une méthode de compression qui dégrade partiellement les niveaux de gris de l'image (compression avec perte), la qualité de l'image peut être estimée de deux manières :

- par une mesure objective (effectuer un calcul mathématique) ;
- par une mesure subjective (constater la qualité visuelle de l'image).

Les travaux sont basés en particulier sur le premier type de mesure qui permet d'écrire l'ensemble des développements analytiques de l'étude théorique d'une méthode. Une mesure objective évalue la qualité d'une image, en calculant la distorsion (la dégradation) engendrée entre les valeurs $x(i)$ de l'image originale I et les valeurs $y(i)$ de l'image quantifiée I^Q . Dans tout le mémoire, l'erreur sera calculée au sens de la norme l_p avec $p = 2$:

$$\|I - I^Q\|_p = \left(\sum_{i=0}^{M-1} |x(i) - y(i)|^p \right)^{\frac{1}{p}} \quad (1.6)$$

La mesure de la distorsion D consiste à calculer l'erreur quadratique moyenne⁶ entre les deux images :

$$D = \frac{1}{M} \sum_{i=0}^{M-1} (x(i) - y(i))^2 \quad (1.7)$$

La qualité peut être également évaluée par le rapport du signal sur bruit⁷ (unité le décibel : dB) :

$$SNR = 10 \log_{10} \left(\frac{\sum_{i=0}^{M-1} x(i)^2}{\sum_{i=0}^{M-1} (x(i) - y(i))^2} \right) \text{ dB} \quad (1.8)$$

et par le rapport du pic signal sur bruit⁸ (unité le décibel : dB) :

$$PSNR = 10 \log_{10} \left(\frac{x_{\max}^2}{\frac{1}{M} \sum_{i=0}^{M-1} (x(i) - y(i))^2} \right) \text{ dB} \quad (1.9)$$

où $x_{\max}$ correspond à l'intensité maximale de l'image. Dans notre exemple, pour des images codées sur 8 bpp : $x_{\max} = 255$. Ces différentes mesures permettent d'évaluer la qualité globale de l'image. Cependant, ce type de mesure ne permet pas d'apprécier la qualité visuelle de l'image ni même de connaître ou de localiser différents types de dégradations (zones texturées, zones de contours, etc.). Ces mesures constituent néanmoins le moyen le plus efficace pour comparer nos résultats par rapport aux autres méthodes.

1.2.3 Complexité : coût calculatoire, coût mémoire

La complexité tient compte de deux principaux aspects :

- le coût calculatoire : il consiste à évaluer le nombre d'opérations (opérations arithmétiques, tests, conditions, etc.) ;
- le coût mémoire : il correspond à estimer la quantité de mémoire utilisée pour le stockage de données nécessaires (stockage de dictionnaires ou de tables de correspondance, tables de données, etc).

La complexité d'une méthode de compression est principalement liée au type d'applications visées (développement de logiciels, implantation sur processeur spécialisé, etc.). Certaines applications privilègient par exemple une technique de compression rapide (acquisition temps réel,...),

⁶ Mean Squared Error (MSE) en anglais.

⁷ Signal to Noise Ratio (SNR) en anglais.

⁸ Peak Signal to Noise Ratio (PSNR) en anglais.

alors que d'autres méthodes cherchent à avoir une technique de décompression rapide (accès à une base de données,...), les meilleures méthodes associent ces deux aspects.

1.2.4 Compromis : taux de compression/ qualité/ complexité

Le compromis : taux de compression/ qualité/ complexité évalue les performances globales d'un schéma de compression. Nos travaux se sont orientés en particulier vers une optimisation du compromis : taux de compression / qualité en utilisant une faible complexité de calcul liée à notre méthode de quantification. Nous allons maintenant présenter les principales étapes qui constituent la chaîne de compression d'images.

1.3 La transformée en ondelettes

Le passage dans un domaine transformé est la première étape de la chaîne de compression. Elle permet de réduire les redondances existant dans une image naturelle. Ceci correspond à effectuer une décorrélation partielle des données de l'image originale. La transformée en ondelettes discrète est associée à différents filtres qui doivent répondre à certains critères : faible complexité et orthogonalité, dans le cadre d'une application de compression. De nombreux travaux ont montré que le filtre 9-7 [26][1] était le mieux adapté en particulier pour des applications de compression d'image avec perte. Le lecteur pourra consulter différentes références [3][1][14] pour approfondir les aspects théoriques de la transformée en ondelettes dans le cas 2D, ou dans le cas 3D [13]. Nous décrivons, dans le chapitre 3, les caractéristiques de la transformée en ondelettes, que nous avons mise en oeuvre (5 niveaux de résolution et utilisation du filtre 9-7).

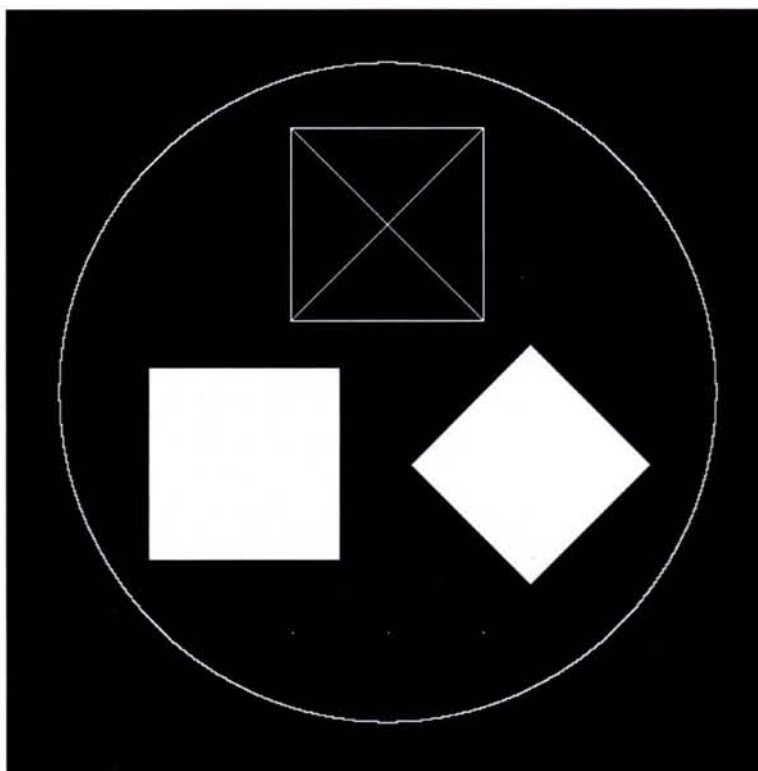


FIG. 1.1: Image artificielle originale de dimension 512×512 pixels.

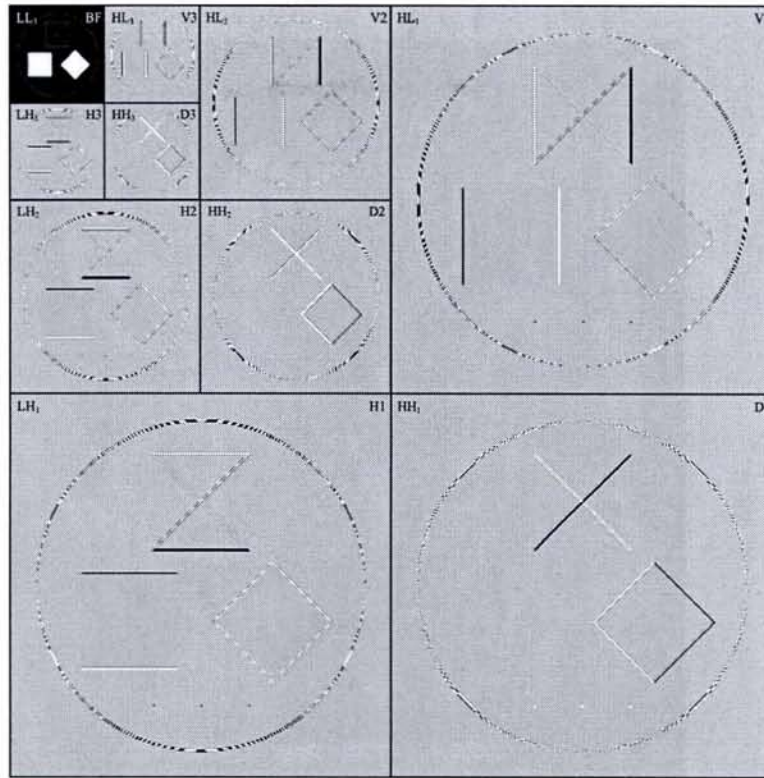


FIG. 1.2: Visualisation des détails préservés sur les différentes sous-bandes (horizontaux, verticaux, diagonaux) pour une image artificielle.

1.3.1 Analyse multirésolution

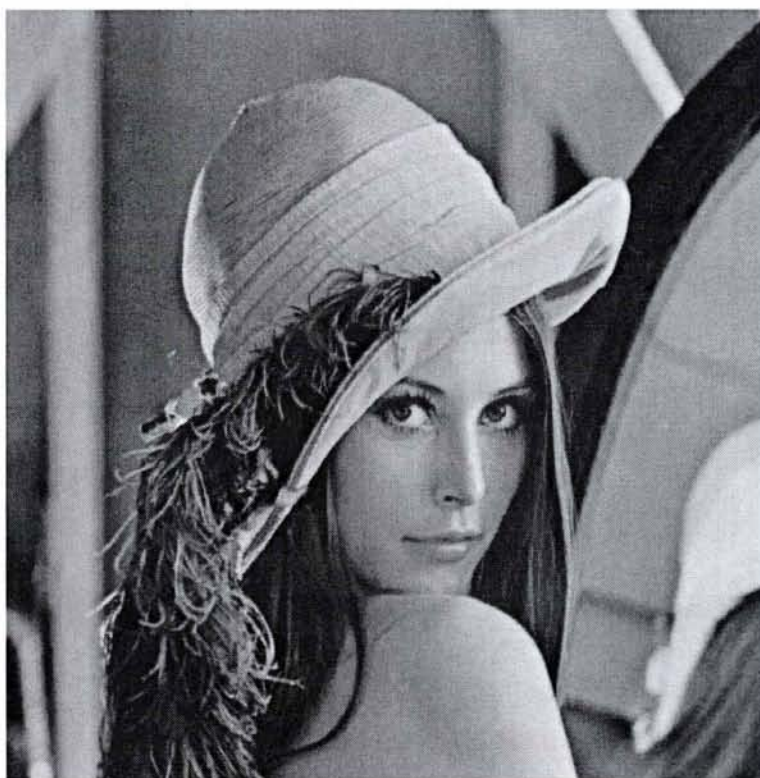
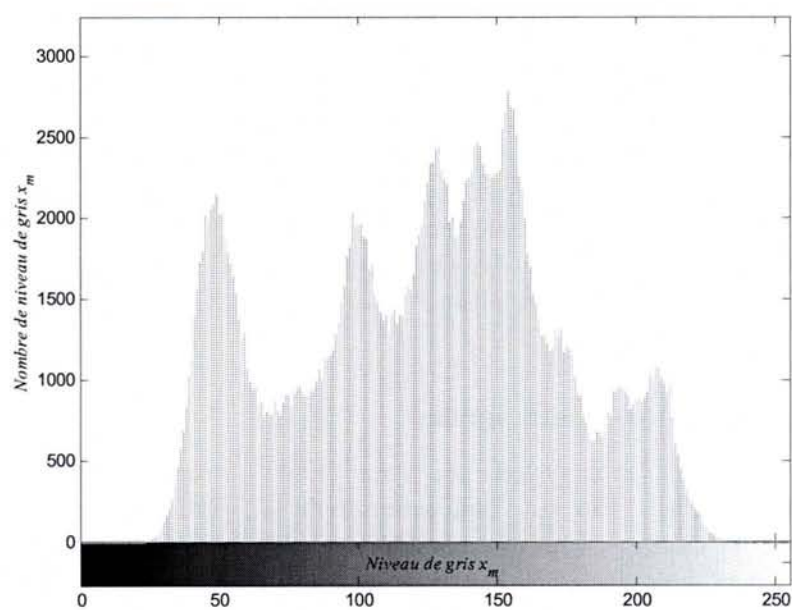
La décomposition en sous-bandes d'une image artificielle est présentée sur la figure 1.2. Nous visualisons clairement les détails qui sont préservés pour chaque sous-bande dans les directions : horizontale, verticale, diagonale (pour des résolutions différentes). Par exemple, nous constatons que les contours du losange sont parfaitement préservés sur toutes les sous-bandes diagonales, alors qu'ils sont presque totalement atténués sur les différentes sous-bandes des autres directions. Ce type de redondances est en particulier exploité par les méthodes utilisant les "arbres de zéros"⁹. Nous rappelons que cette approche est utilisée par l'algorithme *SPIHT* (dépendances inter-bandes), alors que notre approche considère les sous-bandes indépendantes les unes des autres (dépendances intra-bandes). Nous détaillerons cette modélisation, au chapitre 3, lors de la description de l'allocation des débits.

1.3.2 Distribution des coefficients

La distribution des niveaux de gris x_m d'une image naturelle ne correspond pas à un modèle statistique simple comme le montre l'histogramme de la figure 1.4. Par contre, la distribution des coefficients d'ondelettes $x \in \mathbb{R}$ peut être caractérisée par une loi gaussienne généralisée. Cependant, pour faciliter l'étude théorique et les développements analytiques, nous utiliserons comme modèle une loi laplacienne centrée d'écart-type σ de densité de probabilité :

$$f_x(x) = \frac{1}{\sigma\sqrt{2}} \exp\left(-\frac{\sqrt{2}}{\sigma}|x|\right) \quad (1.10)$$

⁹ *Zerotrees* en anglais.

FIG. 1.3: Image *Lena*.FIG. 1.4: Histogramme de l'image *Lena*.

Cette modélisation est très souvent effectuée en compression d'image. Comme nous pouvons le constater sur la figure 1.5(b) la loi laplacienne fournit une bonne approximation de la distribution des coefficients de la sous-bande V1 de la transformée en ondelettes de l'image *Lena* (voir figure 1.6). Ainsi, le modèle d'une source laplacienne centrée caractérisée par un écart-type $\sigma_{V1} = 2,5$ permet d'obtenir une bonne estimation de la distribution des coefficients de cette sous-bande.

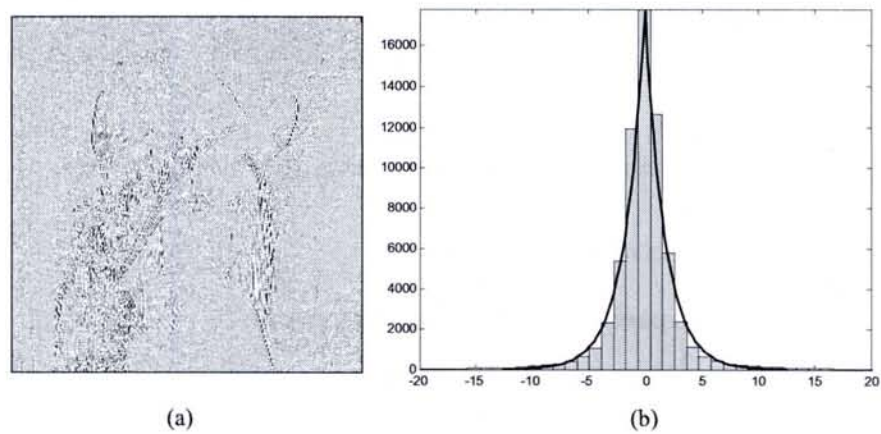


FIG. 1.5: (a) Coefficients de la sous-bande V1. (b) Distribution des coefficients de la sous-bande V1 et courbe théorique d'une source laplacienne centrée et d'écart-type $\sigma_{V1} = 2,5$.



FIG. 1.6: Décomposition en sous-bandes de l'image naturelle *Lena*.

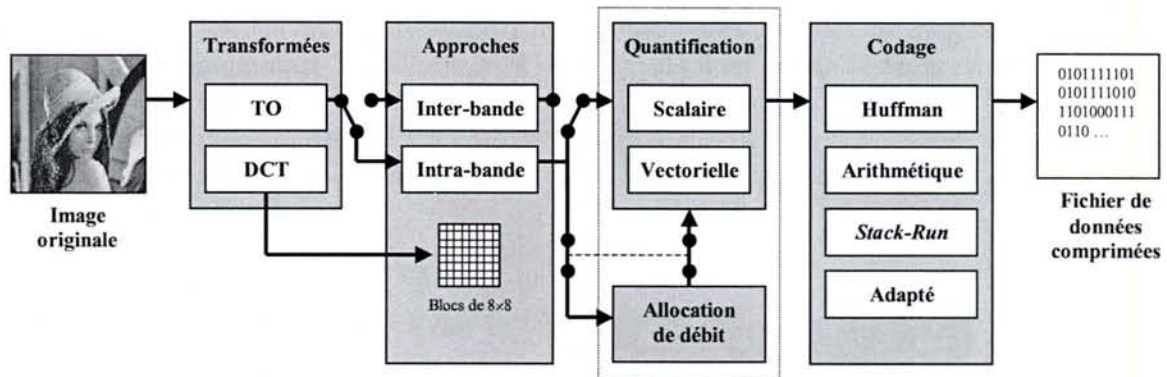


FIG. 1.7: Principales étapes d'une chaîne de compression avec perte.

1.4 Méthodes de compression

1.4.1 Chaînes de compression avec ou sans perte

Nous détaillons à travers ce paragraphe les idées essentielles de chaque méthode. Les méthodes de compression varient suivant les types d'images (naturelles, médicales, satellitaires, etc.) et les applications visées (Internet, stockage, etc.). De plus, les exigences en terme de qualité sont différentes. Il existe deux principales chaînes de compression :

- la chaîne de compression **sans perte** (réversible) : les niveaux de gris (valeurs) de l'image comprimée ne sont tributaires d'aucune modification par rapport aux niveaux de gris de l'image originale (par exemple, pour les applications médicales : aucune erreur de diagnostic ne peut être tolérée). L'avantage de ce type de chaîne est d'avoir une image reconstruite identique, mais l'inconvénient réside dans le faible taux de compression que l'on peut atteindre. En effet, celui-ci est limité par l'entropie de la source (voir formule 1.5) ;
- la chaîne de compression **avec perte** (non réversible) : lors de la phase de quantification, des modifications sont apportées aux valeurs de l'image. L'avantage de ce type d'approche est qu'il est possible d'atteindre des taux de compression importants, mais au détriment de la qualité de l'image reconstruite. Cependant, la majorité des applications grand public s'est orientée vers ce type de compression (appareil photo numérique : images naturelles, transmission d'images sur les différents réseaux, stockage d'images sur CD-Rom, etc.). La figure 1.7 montre les principales étapes d'une chaîne de compression avec perte et présente les différentes méthodes associées à chaque étape.

La phase de quantification est précisément la partie de la chaîne où il y a une perte d'information. Nous présentons, aux paragraphes suivants, les méthodes sans pertes puis avec pertes (Quantification Scalaire et Vectorielle).

1.4.2 Méthodes sans pertes

Les méthodes sans pertes peuvent être employées directement dans une chaîne de compression. Cependant, certaines d'entre elles sont bien souvent utilisées après la phase de quantification d'une chaîne de compression avec perte, lors de la transmission ou du stockage des index. Nous pouvons distinguer :

- les méthodes prédictives [61][45] : celles-ci exploitent la redondance spatiale qui existe entre la valeur courante et les valeurs précédentes ou suivantes ;
- les méthodes par plages [45] : celles-ci exploitent la redondance de plages de valeurs identiques. Pour que ces méthodes soient performantes, il faut que la longueur moyenne des plages soit suffisamment élevée. La méthode la plus connue est le codage *Run-length*, cependant, d'autres

- méthodes permettent d'améliorer les performances de ce codage suivant le type de données à traiter, comme le codage *Stack-Run*. Nous développerons, au chapitre 3, la mise en oeuvre du codage *Stack-Run*, que nous avons adapté pour coder efficacement des plages d'énergie nulle ;
- les codeurs entropiques : ceux-ci tentent de s'approcher le plus possible de l'entropie H_0 de la séquence de valeurs à coder, en affectant un nombre de bits le plus faible possible aux valeurs les plus probables et vice versa. Le codage de Huffman [42] et le codage arithmétique [41] sont les principaux codeurs entropiques utilisés dans le domaine de la compression d'images.

1.4.3 Méthodes avec pertes

Les méthodes de compression d'images avec pertes constituent la majorité des travaux de recherche, en particulier lors de l'étape de quantification. Nous pouvons distinguer :

- les méthodes basées sur la Quantification Scalaire (QS) [39] : elles consistent à traiter les valeurs (de pixels ou de coefficients) de manière individuelle. Différents types de Quantification Scalaire existent, et sont encore utilisés, par exemple pour le standard *JPEG2000*. Nous décrirons plus en détails, dans le chapitre 2, les caractéristiques d'un quantificateur scalaire ;
- les méthodes basées sur la Quantification Vectorielle (QV) [23][27] : elles traitent en même temps un groupement de pixels ou de coefficients, appelés vecteurs. Elles permettent théoriquement d'être toujours plus performante que les méthodes basées sur la Quantification Scalaire. De nombreux travaux se sont articulés autour de la QV par apprentissage. Celles-ci sont les plus lourdes à mettre en oeuvre, car elles consistent à créer au préalable un dictionnaire suivant une séquence d'apprentissage (algorithme Lloyd-Max [58]). Cependant, pour éviter la création d'un dictionnaire et pour accélérer la phase de quantification, nos travaux se sont orientés vers une méthode de Quantification Vectorielle Algébrique (QVA) [65] où le dictionnaire est un réseau régulier de points. Nous la présenterons de manière précise, au chapitre 2.

1.4.4 Evolution des méthodes avec pertes

Les premières méthodes de compression d'images sont apparues voilà une cinquantaine d'années. A l'origine celles-ci traitaient directement les valeurs des pixels de l'image en utilisant principalement la QS associée à différentes approches (optimisation du dictionnaire : algorithme Lloyd-Max, prédiction, etc.) et à différentes techniques de codage (Huffman[42], etc...). Puis, la QV a permis de faire un premier bond dans le domaine de la compression d'images avec une amélioration significative des performances des algorithmes. Deux principales approches peuvent être distinguées : l'une effectue la construction du dictionnaire par apprentissage (algorithme *LBG*[54]) et l'autre définit le dictionnaire sur un réseau régulier de points (QVA).

Mais, c'est avec la prise en compte des dépendances spatiales et l'utilisation de méthodes de décorrélation que la compression d'images a véritablement atteint les performances actuelles. En particulier, en 1974, la Transformation en Cosinus Discrète¹⁰, appliquée sur des blocs de pixels de dimension 8×8 d'une image naturelle, a été utilisée pour le format de compression *JPEG* [69](documentation disponible à l'AFNOR¹¹ sous la référence ISO/IEC IS10918 ou ITU-T T.84, et sur le site <http://www.jpeg.org>). Ce standard est toujours utilisé, mais il devrait être remplacé à l'avenir par le nouveau standard *JPEG2000*[86].

En effet, c'est à la fin des années 80, avec la transformée en ondelettes et sa représentation multirésolution associé au filtre 9-7 [26][1] que l'on a pu obtenir des améliorations considérables en terme de compromis qualité/ taux de compression. De plus cette approche permet de répondre aux nouveaux besoins (transmission progressive, zone d'intérêt, etc.). Différents algorithmes ont utilisé

¹⁰ *Discrete Cosine Transform (DCT)* en anglais.

¹¹ Association française de normalisation, <http://www.afnor.fr>

les propriétés structurelles de la représentation multirésolution, associée à la notion de seuillage (mise à zéro de coefficients ou de vecteurs) : l'approche inter-bandes (*EZW* [83], *SPIHT* [78]) ou l'approche intra-bandes (utilisation d'allocation des débits pour chaque sous-bande).

C'est en sélectionnant les méthodes les plus performantes que nos travaux se sont orientés. D'une part, l'utilisation de la représentation multirésolution nous permet de répondre aux besoins actuels (transmission progressive, etc.) et nous permet de garantir de très bonnes performances en terme de compromis qualité/taux de compression. D'autre part, la transformée en ondelettes associée au filtre 9-7 permet d'utiliser des méthodes de quantification rapide des coefficients d'ondelettes (QVA). De plus, nous avons élaboré la création d'une zone morte vectorielle (utilisation d'un seuillage énergétique) que nous avons associé à une méthode d'allocation des débits et d'optimisation du seuil. Cette approche nous permet d'effectuer un codage efficace des coefficients d'ondelettes significatifs en exploitant les redondances spatiales des plages de vecteurs de coefficients d'ondelettes nuls (utilisation et amélioration d'un codage de type *Stack-Run*).

1.5 Principes des méthodes

De nombreuses méthodes de compression (*JPEG*, *SPIHT*, *JPEG2000*, etc.) [3][61][9], utilisent généralement une transformée (*DCT*, transformée en ondelettes, etc.). Celle-ci permet de décorréler les données de l'image originale, de manière à ce que l'énergie de l'image transformée soit plus compacte. Nous présentons dans la suite de ce paragraphe les principales caractéristiques exploitées ces dernières années en compression d'images par transformée en ondelettes.

1.5.1 Notion de seuillage

L'intérêt d'effectuer un seuillage "brutal" (mise à zéro des valeurs inférieures à un seuil) est d'améliorer par la suite l'étape de quantification. L'étude de la répartition spatiale des coefficients d'ondelettes nous permet de caractériser deux classes disjointes :

- la classe des coefficients significatifs : ceux-ci correspondent aux coefficients d'amplitude importante, et en particulier à la description des contours pour une image naturelle. L'oeil humain est très sensible à ces coefficients qui contribuent le plus à l'énergie du signal ;
- la classe des coefficients non significatifs : ceux-ci correspondent aux coefficients de faible amplitude, et en particulier aux régions dans une image naturelle qui ont des valeurs proches les unes des autres.

La choix de la limite entre ces deux classes est justement le problème du réglage du paramètre de seuil, qui à ce jour, n'a toujours pas de solution systématique. Cependant, différentes méthodes ont été proposées pour trouver un seuil "optimal" associées à un quantificateur et un codeur. De plus, nous pouvons distinguer deux principaux types de seuillage qui permettent d'éliminer ou de prédire les coefficients non significatifs :

- le seuillage par sous-bande qui consiste à mettre à zéro tous les coefficients non significatifs inférieurs au seuil dans chaque sous-bande [30][31] ;
- le seuillage multirésolution qui consiste à tenir compte des seuillages effectués à travers les sous-bandes d'une même direction [83][78].

Ces deux types de seuillage sont très souvent utilisés respectivement dans le cas d'une approche intra-bande ou dans le cas d'une approche inter-bande.

1.5.2 Approche intra-bande

Cette approche considère chaque sous-bande indépendante l'une de l'autre, ainsi elle ne tient pas compte des corrélations entre chaque sous-bande. Il est alors nécessaire de déterminer un débit pour chacune d'elles. Une procédure d'allocation des débits sera présentée en détails au chapitre 3,

Algorithmes	DCT	Transformée en Ondelettes		Allocation de débit	Quantification		Codage				PSNR (dB)
		intra-bande	inter-bande		Scalaire	Vectorielle	Huffman	arithmétique	Stack-Run	adapté	
<i>SPIHT</i>			•							•	31,1
<i>QVAZM</i>		•		•		•			•		30,9
<i>JPEG2000</i>		•		•	•			•			30,8
<i>JPEG</i>	•				•		•				28,2

FIG. 1.8: Méthodes utilisées par les principaux algorithmes de compression d'images. En gris, notre algorithme. Comparaison du *PSNR* pour la compression de l'image *Lena* à un débit de 0,125 bpp ($TC = 64 : 1$).

celle-ci a été retenue pour notre algorithme et elle permet d'atteindre de bonnes performances. Il est ainsi possible d'optimiser le quantificateur pour chaque sous-bandes. Cette approche a également été retenue pour le nouveau standard *JPEG2000* que nous détaillerons dans la suite du chapitre. De plus, en utilisant cette approche il est possible de transmettre progressivement les différentes sous-bandes. Cela permet d'envoyer rapidement un aperçu de l'image et d'affiner la qualité au fur et à mesure que l'on envoie le reste des sous-bandes.

1.5.3 Approche inter-bande

Cette approche tient compte des corrélations entre chaque sous-bande d'une même direction. Celle-ci est généralement associée à une technique de codage utilisant un seuillage multirésolution (*EZW*, *SPIHT*) appelée également approche codage par arbre de zéros. Il n'est pas nécessaire d'utiliser une allocation des débits, cependant, la procédure d'encodage permet de régler le débit cible souhaité. Toutefois, cette approche empêche une optimisation indépendante pour chaque sous-bande, il est alors possible, dans le cas d'une compression à bas débit, que certaines informations de détails soient perdues ou mal représentées au détriment de la qualité visuelle.

1.5.4 Conclusion

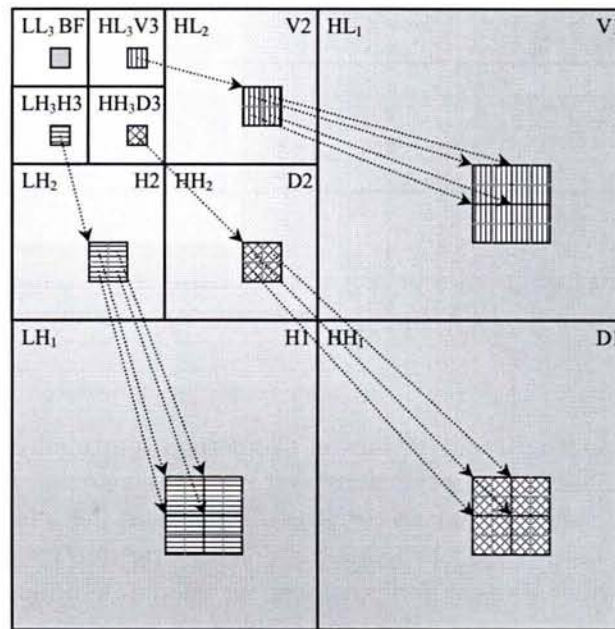
Ces différents aspects montrent que pour une compression à bas débit, et suivant l'application visée, qu'il est préférable d'optimiser la quantification de chaque sous-bande, si l'on souhaite préserver le maximum de détails (au détriment d'une complexité calculatoire légèrement supérieure). Par contre, pour des applications "temps réel", il est préférable d'utiliser une approche inter-bande avec le risque de perdre certains détails. Nous présentons dans le tableau de la figure 1.8 une comparaison des méthodes utilisées par les principaux algorithmes de compression d'images. Nous constatons que notre algorithme se positionne parmi les méthodes de références à base de transformée en ondelettes. Il existe bien d'autres méthodes, cependant nous avons conservé uniquement les plus répandues. Nous pouvons tout de même remarquer que de nombreux travaux se sont orientés vers un codage d'image par transformation fractale [44][36] [28][29] [10][11]. Le principe de ce codage est d'exploiter (dans le domaine spatial) les similarités que l'on peut retrouver dans différentes régions de l'image. Ces méthodes ne seront pas développées dans ce mémoire.

1.6 Méthode de référence et standard

1.6.1 *SPIHT*

1.6.1.1 Approche par "Arbres de zéros"

Un des axes de recherche des méthodes de compression d'images utilisant la transformée en ondelettes consiste à exploiter les redondances entre les différentes sous-bandes de la décomposition

FIG. 1.9: Structure arborescente de *SPIHT*.

(Approche inter-bandes ou par "arbres de zéros"). Cette approche a été initiée par différents travaux [53] et en particulier avec l'algorithme *EZW* [83]. Cet algorithme a ensuite été amélioré pour finalement aboutir à l'algorithme *SPIHT* [78].

Le principe général de cette approche est de structurer les redondances inter-bandes des coefficients non significatifs (considérés comme nuls), sous forme d'un arbre suivant les trois directions : horizontale, verticale et diagonale. La figure 1.9 montre cette structure arborescente. Cette structure permet de savoir où se situent les zones spatiales vides, et donc inversement, où se trouvent les coefficients significatifs à quantifier. Nous pouvons simplement préciser qu'il n'est pas nécessaire d'utiliser que des "arbres de zéros" stricts mais également des arbres tronqués (ou déséquilibrés). Cette organisation des données permet ainsi une bonne compression.

On appelle "carte de signifiante" la structure qui permet de repérer l'information significative ou non limitée par un paramètre de seuil multirésolution. C'est ce seuillage multirésolution qui règle le nombre de coefficients qui seront conservés et ainsi le taux de compression. Cependant, la "carte de signifiante" ne suffit pas, il est ensuite nécessaire d'utiliser une méthode de codage des coefficients qui ont été préservés. Pour cela, il est nécessaire de coder le signe et les bits des coefficients significatifs quantifiés.

Une propriété importante liée à ce type d'approche est que le résultat de la compression (le codage) est imbriqué¹². Ce qui signifie que le résultat du codage de l'image à un taux de compression donné, débute par le résultat du codage de l'image à un taux de compression moins élevé. Cette approche permet une forme de transmission progressive (voir figure 1.10).

1.6.1.2 Performances

De manière générale, l'algorithme *SPIHT* est une référence en terme de compromis qualité (objective)/ complexité. Cependant, visuellement, certains détails ou certaines zones texturées sont généralement moins bien préservées par rapport à certaines méthodes intra-bandes. Cette constatation est une conséquence de la structure en "arbres de zéros" engendrée par le seuillage multirésolution. Nous comparerons, au chapitre 3, notre chaîne de compression à cet algorithme, afin de

¹² *embedded* en anglais.

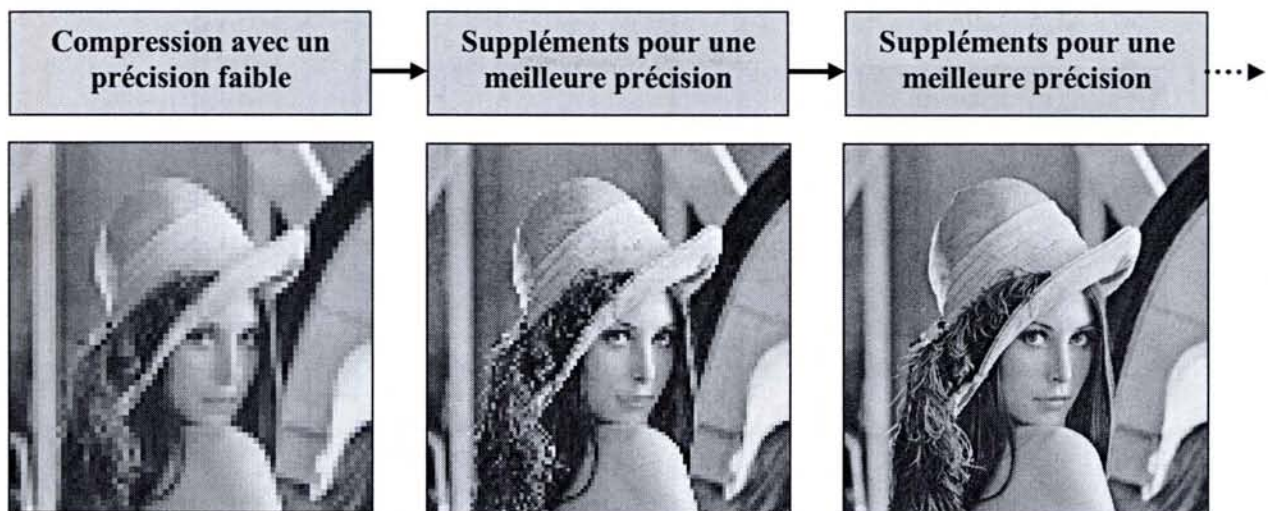


FIG. 1.10: Exemple de compression avec un codage imbriqué.

mettre en évidence ce phénomène.

1.6.2 *JPEG2000*

1.6.2.1 Introduction

Depuis 1996, le processus de normalisation de *JPEG2000* n'a cessé d'évoluer [57][86][87]. Il est actuellement le nouveau standard de compression d'image. Ce standard répond aux besoins et aux problématiques des nouvelles applications (Internet, photos numériques, scanners, imageries : médicales, spatiales, mobiles, etc.). En effet, il doit être en mesure de garantir différents critères :

- compression avec ou sans perte : l'imagerie médicale doit généralement minimiser la dégradation visuelle et parfois ne pas avoir de distorsion numérique du tout ;
- décodage progressif : utiliser par exemple pour la transmission d'images sur le réseau internet, ou lors d'une recherche à travers une base de données (aperçu rapide de l'image) ;
- codage "au fil de l'eau" (codage au fur et à mesure de l'acquisition de l'image à transmettre) : en particulier utilisé pour les applications satellitaires et suivant les contraintes (peu de stockage à bord) liées à l'équipement embarqué (masse, volume, consommation) ;
- faible complexité : actuellement, le développement du marché des appareils photos numériques ou des téléphones mobiles équipés de capteur numérique, nécessitent une implémentation hardware simple avec des coûts mémoire très faibles.

La norme doit donc pouvoir s'adapter aux types d'applications et de plus être performante en terme de compromis taux de compression/ qualité/ complexité. Le lecteur trouvera l'ensemble des travaux et des logiciels liés à *JPEG2000* à l'adresse : <http://www.jpeg.org/JPEG2000.html>.

1.6.2.2 Description générale

La figure 1.11 présente la chaîne de compression simplifiée de *JPEG2000* (pour une image à niveaux de gris, sans tenir compte d'une région d'intérêt, ou de mécanisme de récupération d'erreur dans le cas d'un canal bruité).

L'algorithme *JPEG2000* est appliqué soit sur l'image complète, soit sur des portions d'image appelées "tuiles", soit sur une région d'intérêt de l'image. Il utilise également la transformée en ondelettes avec une approche intra-bande :

- associée au filtre 5-3 [26] dans le cas d'un codage sans perte ;

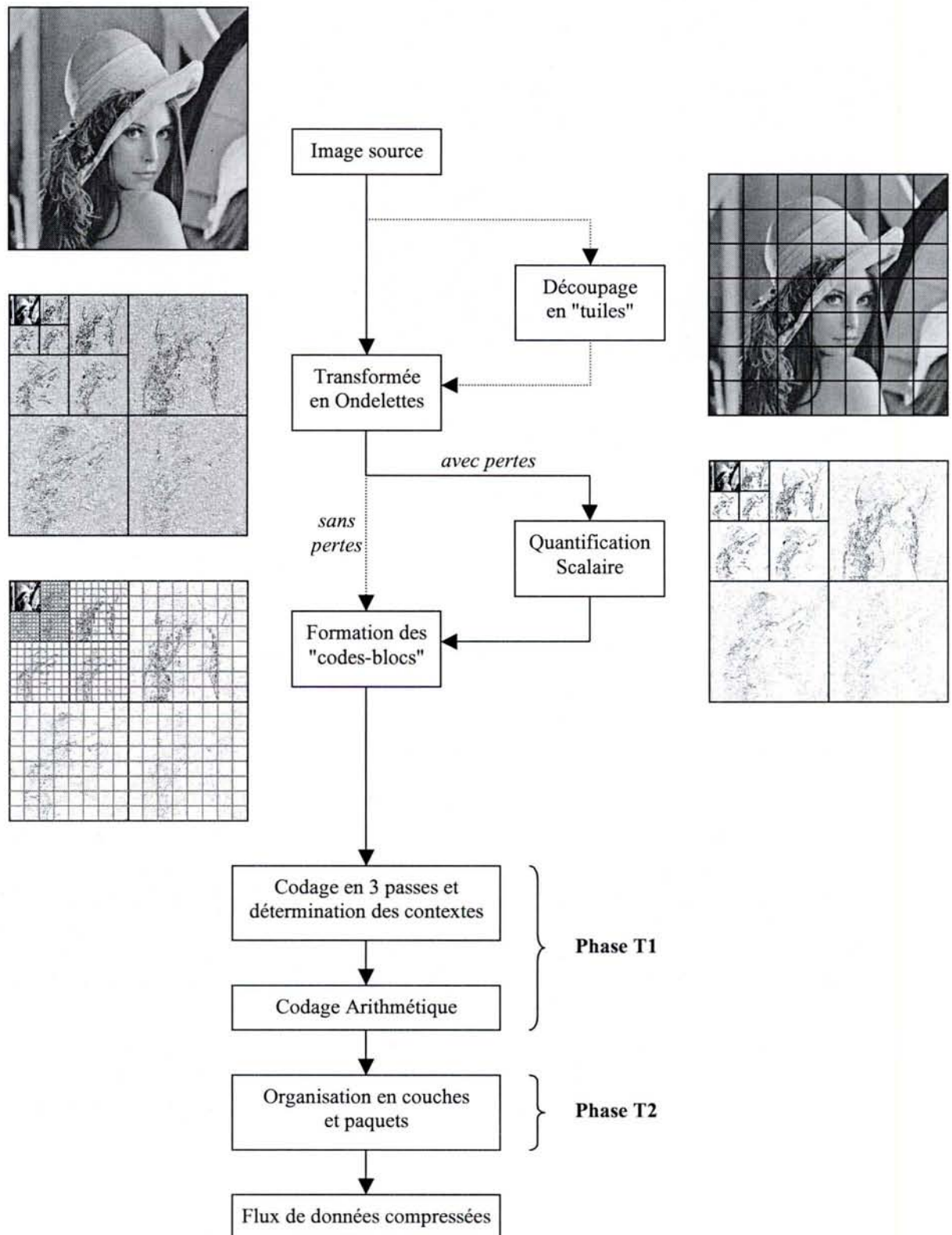


FIG. 1.11: Diagramme simplifié du standard *JPEG2000* : les images sont associées au parcours fléché en trait continu (l'image qui montre le découpage en "tuiles" est mise à titre indicatif).

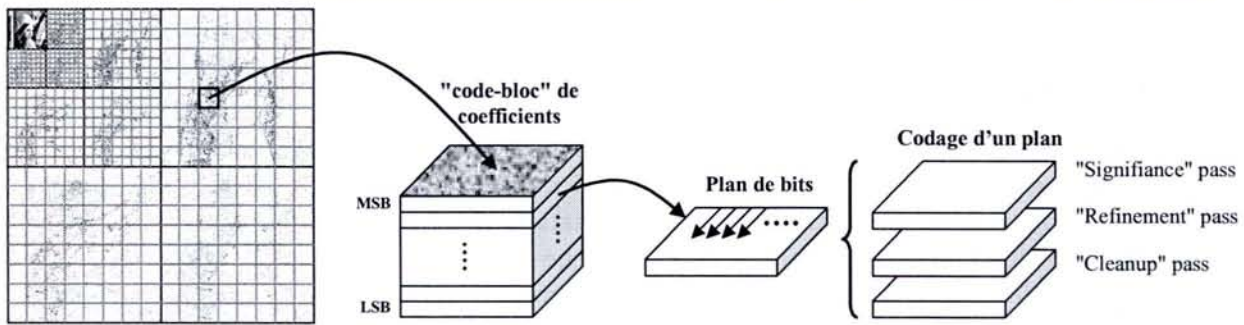


FIG. 1.12: Codage d'un "code-bloc" de coefficients d'ondelettes.

- associée au filtre 9-7 [26] dans le cas d'un codage avec perte.

Dans le cas d'une compression avec perte, il applique une Quantification Scalaire (associé parfois à une zone morte scalaire fixe) qui réduit la précision des coefficients. Cela réduit ainsi la quantité d'information mais au détriment d'une augmentation de la distorsion. Ensuite, des "codes-blocs" de forme rectangulaire sont considérés dans chaque sous-bande de façon indépendante [87] [86]. Le codage (sans perte) de ces "codes-blocs" est réalisé pour répondre à deux types de transmission :

- transmission progressive par résolution (la taille de l'image grandit au fur et à mesure que les données arrivent) ;
- transmission progressive par raffinement de qualité (la qualité de l'image s'améliore au fur et à mesure que les données arrivent), pour cela, il est nécessaire d'utiliser des quantificateurs emboîtables.

Enfin, un codage arithmétique adapté au type de transmission permet de générer le flux de bits de l'image compressée. Nous pouvons également préciser que l'organisation des données ("tuiles", "codes-blocs", paquets de bits,...) contribue à l'allocation de débit des sous-bandes. En effet, nous pouvons distinguer deux phases principales après la formation des "codes-blocs" [3] :

- **phase T1** : celle-ci consiste à ranger en plan de bits les coefficients d'un "code-bloc" : du bit le plus significatif (MSB) au moins significatif (LSB). Chaque plan de bits est ensuite codé en 3 passes : une passe de propagation significative ("Significance" pass), une d'affinage ("Refinement" pass) et une de nettoyage ("Cleanup" pass) [87] (voir figure 1.12). Un codage arithmétique est finalement utilisé pour stocker les différentes séquences de symboles. La contribution de chaque couche (données réorganisées et codées des "codes-blocs") en terme de distorsion et de débit est calculée et stockée ;
- **phase T2** : pour chaque couche, un algorithme d'allocation de débit est mis en oeuvre en tenant compte des informations de chaque "codes-blocs" (pentes débit/ distorsion).

Après cette phase, nous disposons du flux de données compressées qui peut être transmis de différentes manières suivant l'application.

1.6.2.3 Performances

Le standard *JPEG2000* se déploie sur un champ d'applications très large et dispose de nombreuses possibilités de codage et de décodage tout en conservant un bon compromis taux de compression/ qualité/ complexité. Cependant, nous montrerons, au chapitre 3, que notre chaîne de compression offre de meilleures performances en terme de qualité, en particulier à bas débit.

1.7 Notre schéma de compression

1.7.1 Démarche générale

Notre schéma de compression s'inscrit dans le cadre d'une approche intra-bandes : les sous-bandes sont considérées indépendantes les unes des autres afin de les quantifier efficacement. Ce choix nous permet de mieux préserver certains détails et certaines régions texturées et d'obtenir ainsi une meilleure qualité visuelle. Cette approche nécessite cependant de mettre en place une allocation optimale des débits pour chaque sous-bande. Mais une telle optimisation engendre de nombreux calculs. Afin de limiter la complexité, nous avons choisi un algorithme de quantification rapide. Ainsi, nous nous sommes orienté vers la Quantification Vectorielle Algébrique (QVA) qui est une méthode de très faible complexité, avec un seul paramètre de réglage : le facteur de projection γ . Celui-ci permet de régler le taux de compression souhaité, il sera détaillé au chapitre 2. De plus, la QVA permet d'atteindre un compromis taux de compression/ qualité très intéressant.

D'autre part, en observant la répartition spatiale des coefficients d'ondelettes dans chaque sous-bande et en tenant compte du fait qu'il est toujours préférable de mieux coder les coefficients significatifs, nous avons élaboré une zone morte vectorielle, définie par le seuil énergétique r_{ZM} , où certains blocs de coefficients non significatifs sont mis à zéro. Notre problème consiste alors à "optimiser" ce seuil afin d'éliminer les coefficients non significatifs tout en améliorant la qualité pour un débit cible donné. Nous avons donc développé un algorithme recherchant le seuil optimal qui améliore la qualité de l'image tout en préservant les débits optimaux de chaque sous-bande. Enfin, pour accélérer l'optimisation de ce seuil, nous avons fait l'étude détaillée d'un modèle de débit de la Quantification Vectorielle Algébrique avec Zone Morte (QVAZM). Celle-ci sera présentée au chapitre 2.

1.7.2 Description de notre schéma

Nous présentons sur la figure 1.13 le schéma global de notre chaîne de compression d'image. Nous avons utilisé la transformée en ondelettes décomposée sur 5 niveaux de résolution et associée au filtre 9-7[26], bien adapté au cas d'un codage avec perte[1] [3]. Ensuite, la procédure d'allocation des débits (basée sur une optimisation lagrangienne) associée à la Quantification Vectorielle Algébrique avec Zone Morte et au codage *Stack-Run*, permet d'obtenir une bonne répartition des débits dans chaque sous-bande avec le meilleur couple (γ^*, r_{ZM}^*) .

Notre chaîne de compression se situe parmi les meilleures méthodes de compression actuelles en terme de compromis taux de compression/ qualité/ complexité. Nous montrerons en particulier ces bonnes performances pour les bas débits (amélioration de la qualité par rapport au standard *JPEG2000*). Ce schéma sera largement détaillé au chapitre 3, où nous exposerons également l'allocation optimale des débits associée à la QVAZM.

1.8 Conclusion du chapitre

Nous avons présenté à travers ce chapitre les principaux aspects qui ont guidé nos travaux de recherche pour finalement aboutir à notre schéma de compression d'images par Quantification Vectorielle Algébrique avec Zone Morte (QVAZM). Nous présenterons au chapitre 2, les différents éléments théoriques de notre méthode, en particulier, nous définirons les caractéristiques de la QVA, de la QVAZM et nous détaillerons le modèle de débit de la QVAZM. Le chapitre 3 sera consacré à l'étude et la mise en oeuvre d'une allocation optimale de débits associée à la QVAZM et à un codage de type *Stack-Run*, nous y présenterons également nos résultats. Enfin, à travers le chapitre 4, nous montrerons qu'il est possible d'exploiter les caractéristiques de la Quantification Vectorielle Algébrique avec Zone Morte afin d'améliorer la robustesse d'un schéma de tatouage d'images. Nous présenterons une étude sur une image tatouée et modifiée par une attaque de type bruit additif

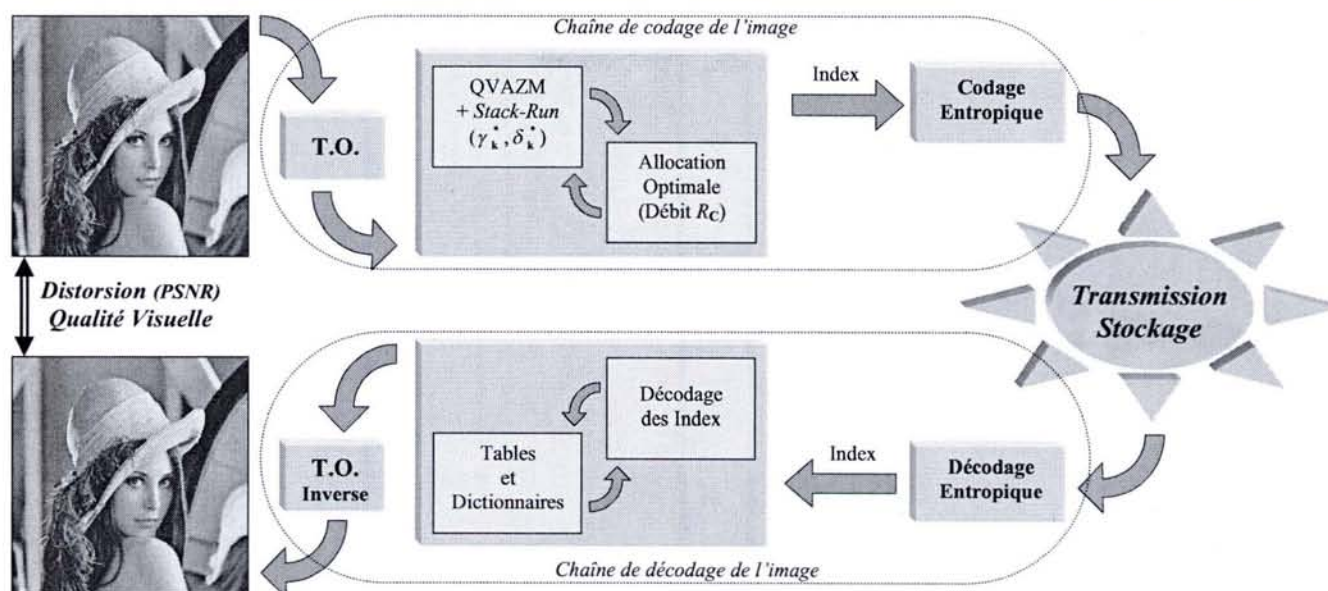


FIG. 1.13: Chaîne de compression d'une image

gaussien. Nous montrerons également la robustesse de cette approche face à des attaques de type recompression ou filtrage spatial (filtre médian).

Chapitre 2

La Quantification Vectorielle Algébrique avec Zone Morte

2.1 Introduction

Ce chapitre est consacré à l'étude théorique de la Quantification Vectorielle Algébrique avec Zone Morte (QVAZM). Cette étude correspond aux fondements de nos travaux. Nous présenterons dans un premier temps une description du processus de quantification dans le cas scalaire et dans le cas vectoriel. Cela nous permettra d'introduire progressivement les notions fondamentales et les différentes notations. Ensuite, nous étudierons plus en détails la Quantification Vectorielle Algébrique (QVA) ainsi que ses principales caractéristiques (en termes de débit, distorsion et complexité). Nous disposerons ainsi de tous les aspects théoriques pour l'étude de la Quantification Vectorielle Algébrique avec Zone Morte. Enfin, nous montrerons l'amélioration que celle-ci apporte, à travers les principaux résultats expérimentaux sur différentes sous-bandes.

2.2 La quantification

La quantification est une opération qui transforme un ensemble de valeurs continues en un ensemble de valeurs discrètes. Il existe deux types de quantification : la quantification scalaire (QS) qui transforme une seule valeur à la fois, et la quantification vectorielle (QV) qui transforme un ensemble de valeurs en même temps. Ces valeurs peuvent correspondre à différents types d'échantillons comme l'intensité des pixels d'une image ou comme les coefficients d'une transformée en ondelettes. Cette discrétisation permet d'obtenir un ensemble fini de valeurs, il est donc possible d'adapter des techniques de compression qui utilisent la redondance spatiale de celle-ci. De manière générale, les travaux de Shannon [81] montrent que la QV a toujours des performances supérieures à celles de la QS en termes de compromis taux de compression/ qualité. Nous présenterons brièvement à travers ce chapitre les performances des différents types de quantification, largement détaillés dans différents ouvrages [38][61][3] et articles [39][23], pour nous consacrer plus particulièrement à la Quantification Vectorielle Algébrique (QVA) [22][35][34] [65][52].

2.3 La Quantification Scalaire

2.3.1 Définitions

La QS consiste à effectuer la quantification d'une seule valeur à la fois. Il existe différents types de QS, nous présentons ici leurs caractéristiques principales afin d'introduire les différentes notations.

Considérons un signal continu $x \in \mathbb{R}$ sur l'intervalle $\mathcal{P} = [A, B]$. Après une discrétisation de ce signal, nous obtenons M échantillons $x(k), k \in \mathbb{N}$. Chaque échantillon $x(k)$ (pixel dans le cas d'une image numérique ou coefficient d'ondelettes dans le cas d'une sous-bande) peut être interprété comme la réalisation d'une variable aléatoire $\mathbf{x}$ dont les valeurs sont prises dans l'alphabet $\{x_0, \dots, x_m, \dots, x_{M-1}\}$ composé de M représentants. Notre objectif est maintenant de réduire ce nombre de représentants. Pour cela, il est possible de découper l'intervalle $\mathcal{P}$ en L partitions P_i , dont les propriétés sont données par :

$$\mathcal{P} = \bigcup_{i=0}^{L-1} P_i \quad \text{et} \quad P_i \cap P_j = \emptyset \quad \forall i \neq j \quad (2.1)$$

telles que $L < M$ et d'associer à chaque partition P_i un représentant y_i . L'ensemble des représentants y_i est appelé le dictionnaire¹, on note $\mathcal{C} = \{y_0, \dots, y_{L-1}\}$. Dans le cadre d'une application numérique, le nombre de représentants L est souvent choisi tel que $L = 2^b$, où b correspond au nombre de bits nécessaires pour coder un représentant.

La procédure d'encodage consiste à décider à quel intervalle P_i appartient x puis à lui associer le numéro $i \in \{0, \dots, L-1\}$ correspondant à l'index du représentant y_i . C'est ce numéro i (ou index) qui sera transmis ou stocké. La procédure de décodage consiste à associer au numéro i le représentant y_i correspondant. Nous pouvons décrire la QS comme une application QS de $\mathcal{P}$ dans un ensemble fini $\mathcal{C}$ contenant L représentants (ou symboles scalaires), cette application est définie par :

$$QS : \begin{cases} \mathcal{P} \longrightarrow \mathcal{C} & \text{avec } \mathcal{C} = \{y_0, \dots, y_{L-1}\} \\ x \longrightarrow QS(x) = y_i & \text{avec } i \in \{0, \dots, L-1\} \end{cases} \quad (2.2)$$

2.3.2 Distorsion de la QS

Nous rappelons que la mesure de la distorsion permet d'évaluer la qualité objective d'un signal de manière mathématique. Elle permet ainsi d'effectuer des développements analytiques afin de trouver des solutions théoriques aux différents problèmes. Le processus de quantification engendre une erreur entre une valeur x de la source et sa valeur quantifiée $QS(x) = y_i$. Cette distorsion est généralement évaluée de manière quadratique :

$$d_{QS}(x, QS(x)) = |x - y_i|^2 \quad (2.3)$$

Afin d'obtenir l'erreur globale pour avoir une estimation de la qualité des données quantifiées, nous calculons l'espérance mathématique E de la distorsion. Elle consiste à effectuer la moyenne statistique, donnée par la relation :

$$D_{QS} = E[d_{QS}(\mathbf{x}, QS(\mathbf{x}))] = \sum_{i=0}^{L-1} \int_{x \in P_i} d_{QS}(x, QS(x)) f_{\mathbf{x}}(x) dx \quad (2.4)$$

où $f_{\mathbf{x}}(x)$ correspond à la densité de probabilité² de la variable aléatoire d'entrée $\mathbf{x}$.

Dans le cas d'un signal discrétisé borné, la distorsion (expérimentale) s'écrit :

$$\widehat{D}_{QS} = \sum_{m=0}^{M-1} d_{QS}(x_m, QS(x_m)) p_{\mathbf{x}}(x_m) \quad (2.5)$$

où $p_{\mathbf{x}}(x_m)$ correspond à la probabilité d'apparition de la valeur x_m du signal discrétisé.

¹ Codebook en anglais

² PDF : Probability Density Function en anglais

Dans le cas d'un signal où toutes les valeurs x peuvent être quantifiées par un quantificateur scalaire dont le dictionnaire $\mathcal{C}$ contient L représentants (ou symboles scalaires) et pour une mesure de distorsion donnée par la relation 2.3, la distorsion d'un quantificateur scalaire peut s'écrire comme l'erreur quadratique moyenne :

$$D_{QS} = E[d_{QS}(\mathbf{x}, QS(\mathbf{x}))] = \sum_{i=0}^{L-1} \int_{x \in P_i} (x - y_i)^2 f_{\mathbf{x}}(x) dx \quad (2.6)$$

Dans le cas d'un signal discrétisé borné, nous pouvons écrire la distorsion (expérimentale) par :

$$\widehat{D}_{QS} = \sum_{i=0}^{L-1} \sum_{\substack{m \\ x_m \in P_i}} (x_m - y_i)^2 p_{\mathbf{x}}(x_m) \quad (2.7)$$

2.3.3 Bruit granulaire et bruit de surcharge

Il existe deux types de distorsion liés aux caractéristiques du quantificateur appelés : le bruit granulaire et le bruit de surcharge. Nous pouvons ainsi écrire la distorsion par la relation :

$$D_{QS} = D_{QS}^{\text{granulaire}} + D_{QS}^{\text{surcharge}} \quad (2.8)$$

Suivant les caractéristiques du quantificateur, le calcul du bruit de surcharge peut être légèrement différent. Cependant, si nous considérons un signal continu x prenant ses valeurs dans l'intervalle $P = [A, B]$ et un quantificateur sur ce même intervalle, le bruit introduit par l'opération de quantification est uniquement un bruit granulaire. Par contre, si nous considérons un signal x prenant ses valeurs dans l'ensemble $\mathbb{R}$ et un quantificateur défini dans l'intervalle $P = [A, B]$, il est nécessaire d'ajouter un bruit de surcharge. Ce bruit correspond à l'erreur entre les valeurs de x qui sont à l'extérieur du quantificateur et les bornes de celui-ci. Ce bruit introduit une distorsion de surcharge définie par :

$$D_{QS}^{\text{surcharge}} = \int_{-\infty}^A (x - \bar{x}_A)^2 f_{\mathbf{x}}(x) dx + \int_B^{+\infty} (x - \bar{x}_B)^2 f_{\mathbf{x}}(x) dx \quad (2.9)$$

où $\bar{x}$ est la valeur x ramenée sur la borne la plus proche du quantificateur ($\bar{x}_A = A$ si $x \in]-\infty, A]$ et $\bar{x}_B = B$ si $x \in [B, +\infty[$).

Afin de simplifier la notation, lorsque le bruit de surcharge est nul ou n'est pas pris en compte, nous conserverons uniquement la notation D_{QS} pour définir la distorsion.

2.3.4 Distorsion en fonction du débit : approximation haute résolution

Les travaux de Bennett [12] et Widrow [93] ont prouvé que sous certaines conditions, il était possible de faire une bonne approximation de la distorsion en fonction du débit $D(R)$. Les conditions sont les suivantes : si le quantificateur dispose d'un nombre important de représentants L (haute résolution, partitions P_i petites), s'il n'y a pas de bruit de surcharge et si la densité de probabilité de la source est "douce", alors il est possible de poser l'hypothèse suivante : sur chaque partition P_i le bruit de quantification du signal est indépendant et uniformément distribué. Bennett a calculé une borne asymptotique maximale correspondant à la distorsion de la formule 2.6. Elle est donnée par la relation :

$$2^{-2R} \leq D(R) \leq \frac{1}{12} 2^{-2R} \left[\int_{-\infty}^{+\infty} f_{\mathbf{x}}(x)^{\frac{1}{3}} dx \right]^3 \quad (2.10)$$

dans le cas de la norme l_2 , où 2^{-2R} correspond à la borne de Shannon. La fonction de cette borne, sur la figure 2.1, montre l'allure générale de la distorsion en fonction du débit $D(R)$.

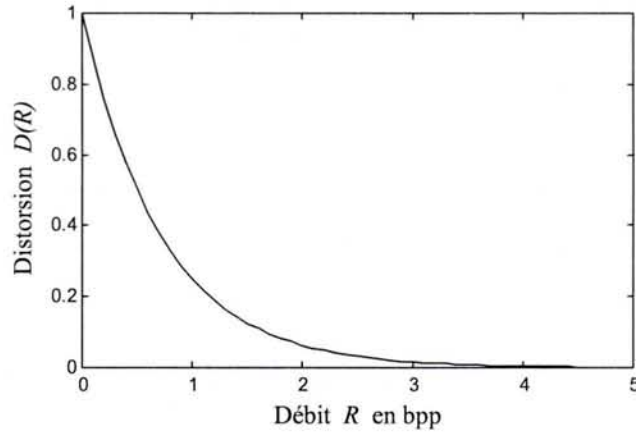


FIG. 2.1: Distorsion en fonction du débit de la borne de Shannon, $D(R) = 2^{-2R}$

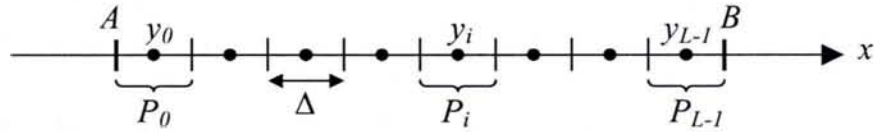


FIG. 2.2: Quantification Scalaire uniforme

2.3.5 La Quantification Scalaire uniforme

La quantification scalaire uniforme consiste à partitionner l'intervalle $\mathcal{P} = [A, B]$ en L partitions distinctes $P_i \in \{P_0, \dots, P_{L-1}\}$ de même longueur $\Delta = \frac{A+B}{L}$. Il suffit ensuite de leur associer un représentant y_i qui correspond à la valeur du centre de la partition P_i . Le partitionnement du QS uniforme est schématisé figure 2.2. Ce type de quantificateur serait bien adapté pour un signal ayant une distribution uniforme : c'est-à-dire telle que la densité de probabilité $f_{\mathbf{x}}(x) = Cte \ \forall x \in [A, B]$, mais ce type de signal est peu courant. Par contre, la simplicité de ce type de quantificateur permet de réduire radicalement la complexité de la quantification. Il est d'ailleurs utilisé en partie dans le nouveau standard de compression *JPEG2000*.

2.3.6 La Quantification Scalaire non uniforme

La QS non uniforme consiste à partitionner l'intervalle $\mathcal{P} = [A, B]$ en L partitions adaptées à la densité de probabilité $f_{\mathbf{x}}(x)$ d'un signal x afin de minimiser la distorsion globale D_{QS} . Il suffit ensuite d'associer le meilleur représentant y_i pour chaque partition P_i . Le partitionnement du QS non uniforme est schématisé figure 2.3. Ce type de quantificateur est bien adapté aux signaux numériques réels. Mais comme on ne connaît pas parfaitement la densité de probabilité $f_{\mathbf{x}}(x)$, la complexité de la recherche du meilleur partitionnement augmente avec le nombre d'échantillons du signal x et le nombre de représentants L . L'algorithme Lloyd-Max [55][58] permet d'obtenir une solution "localement optimale".

2.3.7 Conditions d'optimalité : Algorithme de Lloyd-Max

Les travaux de Lloyd et de Max [55][58] ont permis d'établir les deux conditions nécessaires d'optimalité pour construire un quantificateur adapté à des sources quelconques. Comme nous

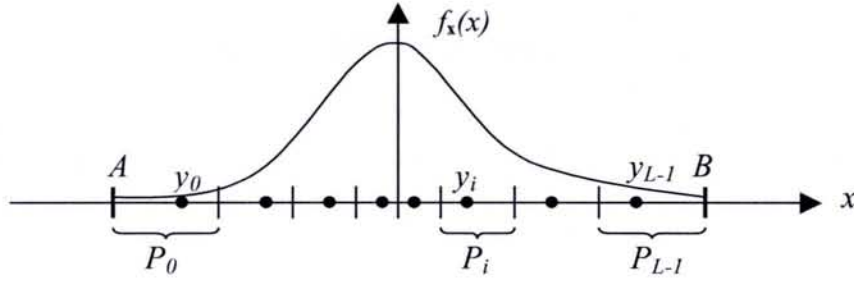


FIG. 2.3: Quantificateur Scalaire non uniforme

l'avons expliqué pour la QS non uniforme, définir un quantificateur "optimal" consiste à trouver les L meilleures partitions P_i du signal et les meilleurs représentants y_i associés (les représentants du dictionnaire $\mathcal{C}$) afin de minimiser la distorsion. Cette optimisation conjointe n'admet pas de solution simple. Cependant, il est possible de définir les deux conditions suivantes :

- connaissant le dictionnaire, il est possible de trouver le meilleur partitionnement. En effet, étant donné un dictionnaire $\mathcal{C} = \{y_0, \dots, y_{L-1}\}$, il est possible de classer les valeurs de x dans chaque partition P_i telle que la règle 2.11 dite du **plus proche voisin** soit vérifiée ;

$$P_i = \{x \mid d(x, y_i) \leq d(x, y_j) \quad \forall i \neq j, \quad j \in \{0, \dots, L-1\}\} \quad (2.11)$$

- connaissant le partitionnement, il est possible de trouver le meilleur dictionnaire. En effet, étant donné un partitionnement $\mathcal{P} = \{P_0, \dots, P_{L-1}\}$, il est possible de trouver le meilleur représentant y_i pour chaque partition P_i en calculant le centre de gravité ou **centroïde** de la partition P_i , formule 2.12.

$$y_i = E[x \mid x \in P_i] = \frac{\int_{x \in P_i} x f_x(x) dx}{\int_{x \in P_i} f_x(x) dx} \quad (2.12)$$

De manière générale, les algorithmes d'optimisation pour la création du dictionnaire sont basés sur ces deux conditions. L'algorithme de Lloyd-Max utilise ces deux conditions afin de construire itérativement un dictionnaire de façon à minimiser la distorsion D_{QS} .

2.3.8 Conclusion

A travers l'étude de la Quantification Scalaire, nous avons défini les principales notions liées à la modélisation d'un signal et au calcul de la distorsion. De plus, nous avons introduit les différentes notations qui permettront de faciliter la compréhension de l'étude de la Quantification Vectorielle. Nous rappelons que la Quantification Scalaire uniforme est une méthode dont le coût de calcul est très faible. Ce qui explique son intégration dans le nouveau standard de compression *JPEG2000*.

2.4 La Quantification Scalaire avec Zone Morte

2.4.1 Introduction

Le concept de zone morte³ (ZM) a fait l'objet de nombreux travaux, en particulier dans le domaine du débruitage d'image [17][18][19][20] [79][66][90] [94][77]. Ainsi, après une transformée en

³ *Dead Zone* ou *Zero Zone* en anglais

ondelettes, un des objectifs communs à la compression et au débruitage est d'éliminer les coefficients non significatifs, c'est-à-dire les coefficients dont l'amplitude n'est pas suffisamment importante pour apporter une information "significative" lors de la reconstruction. Pour cela une solution consiste à effectuer un seuillage brutal. Ces travaux ont été initiés principalement par Donoho [30][31]. Il a établi un seuil "universel" δ pour les coefficients d'une transformée en ondelettes en J niveaux de décomposition. Ce seuil dépend de l'estimation de la variance du bruit $\hat{\sigma}_J^2$ pour chaque niveau de décomposition j et du nombre total de coefficients d'ondelettes M_c :

$$\delta(j, M) \propto \sqrt{2\hat{\sigma}_J^2 \log(M_c)} \quad (2.13)$$

2.4.2 Définitions

La création d'une zone morte consiste donc à effectuer un seuillage directement sur les valeurs x de la source, ce type de seuillage est également appelé "seuillage dur"⁴ et noté $T_h(x, \delta)$. Celui-ci consiste à mettre à zéro toutes les valeurs au voisinage de zéro limitées par le seuil δ :

$$T_h(x, \delta) = x \cdot \mathbf{1}\{|x| \geq \delta\} \quad (2.14)$$

$$\text{avec } \mathbf{1}\{|x| \geq \delta\} = \begin{cases} 1 & \text{si } |x| \geq \delta \\ 0 & \text{sinon} \end{cases}$$

Cette approche permet d'éliminer les coefficients non significatifs. Il est alors possible, pour un débit donné, de quantifier avec plus de précision les coefficients significatifs. La Quantification Scalaire avec Zone Morte (QSZM) est définie par :

$$QS_{ZM} : \begin{cases} \mathcal{P} \longrightarrow \mathcal{C} & \text{avec } \mathcal{C} = \{y_0, \dots, y_{L-1}\} \\ x \longrightarrow QS_{ZM}(x) = \begin{cases} 0 & , \quad |x| < \delta \\ y_i & , \quad |x| \geq \delta \end{cases} \end{cases} \quad (2.15)$$

2.4.3 Optimisation de la Quantification Scalaire avec Zone Morte

Différents travaux ont intégré le concept de zone morte à la Quantification Scalaire [67][74]. Ces travaux montrent qu'il est possible d'améliorer les performances de la Quantification Scalaire classique.

L'objectif de cette approche est de trouver un intervalle optimal P_{ZM} défini par le seuil δ (quantificateur uniforme avec zone morte [74]). La figure 2.4 montre le découpage des partitions de la Quantification Scalaire uniforme avec Zone Morte (QSZM), c'est-à-dire avec un "seuillage dur" [19][20] [30][31].

Comme le montre la figure 2.4, nous pouvons écrire la distorsion de ce quantificateur par :

$$D_{QS_{ZM}}(\delta) = 2 \int_0^\delta x^2 f_x(x) dx + 2 \sum_{l=1}^{+\infty} \int_{\delta+lq}^{\delta+(l+1)q} \left[x - \left(\delta + \left(l + \frac{1}{2} \right) q \right) \right]^2 f_x(x) dx \quad (2.16)$$

où q correspond au pas de quantification et $f_x(x)$ à la densité de probabilité de la valeur x de la source. Cette formule ne tient pas compte du bruit de surcharge.

La minimisation de cette distorsion $\left(\frac{\partial D_{QS_{ZM}}(\delta)}{\partial \delta} = 0 \right)$ permet de trouver une solution analytique du seuil δ pour une source de distribution laplacienne d'écart-type σ donnée par la formule 2.17.

⁴ *hard-threshold* en anglais

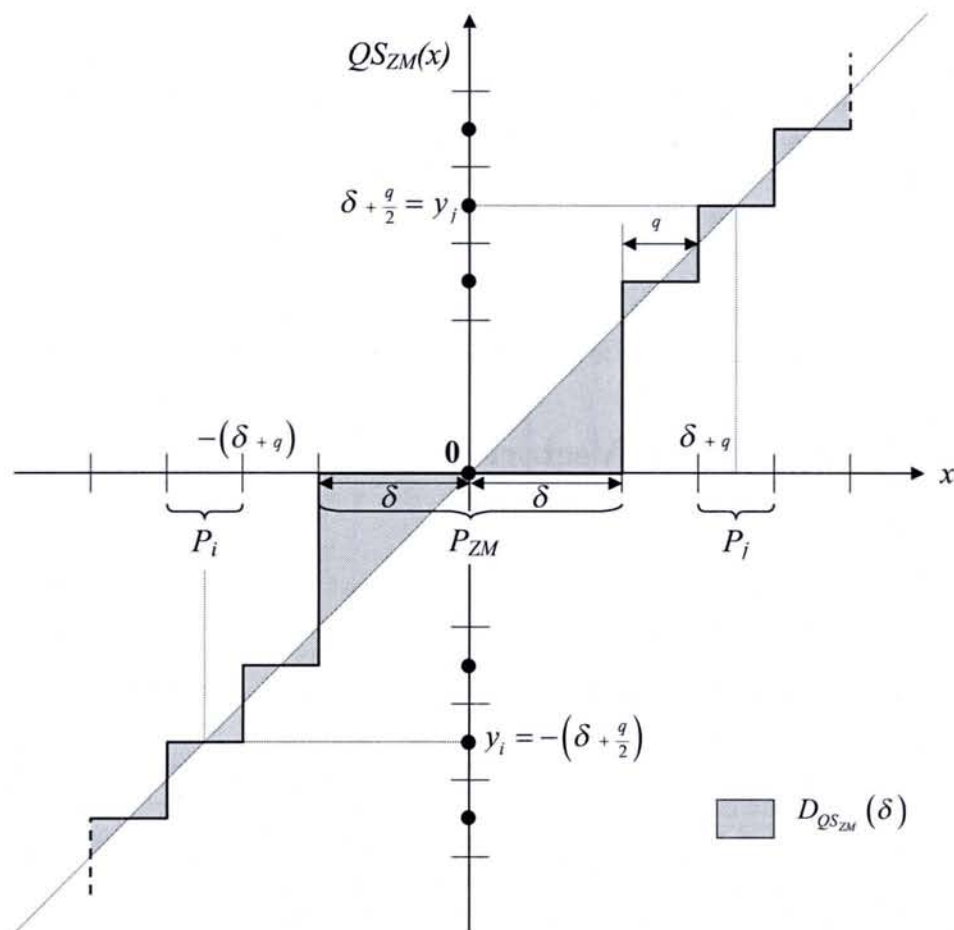


FIG. 2.4: Quantification Scalaire uniforme avec Zone Morte.

Cette formule permet d'obtenir le seuil optimal δ en fonction du pas de quantification q et de l'écart-type σ de la source.

$$\delta^2 = \sigma^2 + \frac{q^2}{4} - \frac{\sigma}{\sqrt{2}} q \times \frac{1 + \exp\left(-\frac{\sqrt{2}}{\sigma} q\right)}{1 - \exp\left(-\frac{\sqrt{2}}{\sigma} q\right)} \quad (2.17)$$

Le gain observé de la Quantification Scalaire uniforme avec Zone Morte par rapport à la Quantification Scalaire uniforme est de 0,8 dB, dans le cas d'une source laplacienne centrée, de variance unité [74].

2.4.4 Conclusion

Cette étude montre l'amélioration que peut apporter le concept de zone morte dans le cas de la Quantification Scalaire uniforme. Le seuillage effectué sur les coefficients d'ondelettes est un seuillage scalaire. Nos travaux se sont portés sur l'étude d'un seuillage vectoriel adapté à la forme du dictionnaire, nous avons ainsi défini une zone morte vectorielle. Dans un premier temps, nous introduirons de manière générale la Quantification Vectorielle. Ensuite, nous présenterons l'intérêt d'utiliser la Quantification Vectorielle Algébrique. Et enfin, nous intégrerons l'étude du concept de zone morte au cas vectoriel. En particulier, nous établirons le modèle théorique du débit de la Quantification Vectorielle Algébrique avec Zone Morte.

2.5 La Quantification Vectorielle

2.5.1 Définitions

De manière générale, la Quantification Vectorielle est une généralisation de la Quantification Scalaire. Plutôt que d'effectuer un processus de quantification sur une seule valeur, la QV consiste à quantifier simultanément un ensemble de valeurs dans un même voisinage spatial, à savoir un vecteur⁵. Nous définissons par n la dimension d'un vecteur, c'est-à-dire le nombre de valeurs qui composent le vecteur.

Considérons un signal $x \in \mathcal{P} = [A, B]$ discrétisé en M échantillons $x(k)$. Chaque échantillon $x(k)$ peut être interprété comme la réalisation d'une variable aléatoire $\mathbf{x}$. La création des vecteurs consiste à regrouper n échantillons. Nous disposons alors de $\frac{M}{n}$ vecteurs. Chaque vecteur $X(k)$ peut être interprété comme la réalisation d'un vecteur aléatoire $\mathbf{X}$ composé des variables aléatoires $\mathbf{x}_1, \dots, \mathbf{x}_n$. La connaissance du modèle de ce signal s'exprime à travers la densité de probabilité conjointe $f_{\mathbf{X}}(X) = f_{\mathbf{x}_1, \dots, \mathbf{x}_n}(x_1, \dots, x_n)$. Ce signal est partiellement connu, cependant en considérant celui-ci comme un ensemble de coefficients d'ondelettes, nous tiendrons compte de l'hypothèse selon laquelle les variables aléatoires $\mathbf{x}_1, \dots, \mathbf{x}_n$ sont indépendantes et identiquement distribuées (i.i.d). Si on note $\mathbf{x}_i$ la variable aléatoire associée à la composante x_i du vecteur X et si on note $f_{\mathbf{x}_i}(x_i)$ la densité de probabilité associée à la variable aléatoire $\mathbf{x}_i$, alors la densité de probabilité conjointe $f_{\mathbf{X}}(X)$ peut être définie comme le produit des densités marginales :

$$f_{\mathbf{X}}(X) = f_{\mathbf{x}_1, \dots, \mathbf{x}_n}(x_1, \dots, x_n) = \prod_{i=1}^n f_{\mathbf{x}_i}(x_i) \quad (2.18)$$

Considérons le vecteur aléatoire $\mathbf{X}$ à réalisation dans l'alphabet $\{X_0, \dots, X_m, \dots, X_{M-1}\}$. Il est possible de réduire le nombre de représentants de cet alphabet en découpant l'espace de dimensions

⁵Un vecteur peut être considéré comme une matrice rectangulaire, ligne ou colonne.

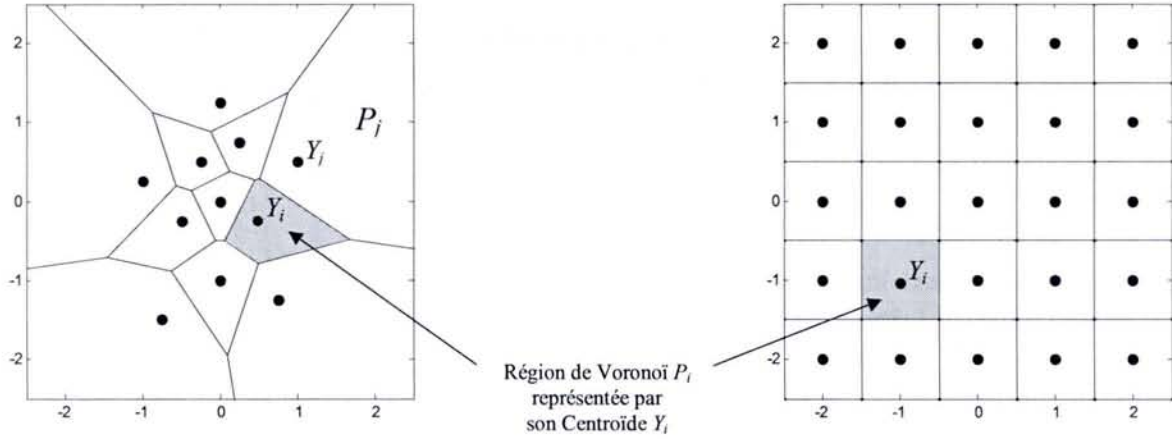


FIG. 2.5: Partitionnement irrégulier (à gauche) et régulier (à droite) sur $\mathbb{Z}^2$. Chaque vecteur X appartenant à une région P_i est quantifié par Y_i .

n en L partitions P_i dont les propriétés sont données par :

$$\mathcal{P}^n = \bigcup_{i=0}^{L-1} P_i \quad \text{et} \quad P_i \cap P_j = \emptyset \quad \forall i \neq j \quad (2.19)$$

telles que $L < M$ et d'associer à chaque partition P_i un vecteur représentant $Y_i = \{y_1, \dots, y_n\}$. Nous appelons l'ensemble des vecteurs représentant Y_i : un dictionnaire $\mathcal{C} = \{Y_0, \dots, Y_{L-1}\}$. Le dictionnaire peut être créé à l'aide de méthodes par apprentissage ou directement défini sur un réseau régulier de points. Nous présenterons ces différentes approches dans les paragraphes suivants.

La procédure d'encodage consiste à décider à quelle partition P_i appartient le vecteur X puis à lui associer le numéro $i \in \{0, \dots, L-1\}$ correspondant au vecteur représentant Y_i . C'est ce numéro i (ou symbole canal) qui sera transmis ou stocké. La procédure de décodage consiste à associer au numéro i le représentant $Y_i = \{y_1, \dots, y_n\}$ correspondant. Nous pouvons décrire la QV comme une application QV de $\mathcal{P}^n$ dans un ensemble fini $\mathcal{C}$ contenant L représentants (ou symboles vectoriels), cette application est définie par :

$$QV : \begin{cases} \mathcal{P}^n \longrightarrow \mathcal{C} = \{Y_0, \dots, Y_{L-1}\} \\ X \longrightarrow QV(X) = Y_i = \{y_1, \dots, y_n\} \end{cases} \quad (2.20)$$

2.5.2 Différents types de partitionnement

Sur la figure 2.5, nous montrons les différents types des partitions P_i (ou régions de Voronoï) pour des vecteurs de dimension $n = 2$. La forme de ces partitions est conditionnée par la position dans l'espace des vecteurs Y_i (ou centroïde de la partition) et par le critère de distorsion utilisé. Nous pouvons distinguer deux types de partitionnement :

- régulier : dans ce cas, le dictionnaire est structuré. Il est alors possible de le construire de manière analytique. Le processus de quantification du vecteur X s'effectue alors directement sur un réseau régulier de points. C'est sur ce type de partitionnement que nos travaux sont basés, en particulier avec la Quantification Vectorielle Algébrique ;
- irrégulier : dans ce cas, le dictionnaire est non structuré. Il doit être construit par des méthodes d'apprentissage à partir de séquences de données. Le processus de quantification du vecteur X doit effectuer une recherche à travers les représentants Y_i du dictionnaire.

2.5.3 Distorsion de la QV

Comme nous l'avons étudié dans le cas scalaire, le processus de quantification engendre une distorsion entre un vecteur X de la source et un vecteur $QV(X) = Y_i$ du dictionnaire[39]. Celle-ci peut être évaluée par l'erreur quadratique :

$$d_{QV}(X, QV(X)) = \|X - Y\|_2^2 = \sum_{i=1}^n (x_i - y_i)^2 \quad i \in \mathbb{N} \quad (2.21)$$

L'espérance mathématique E de la distorsion de la QV normalisée est donnée par la formule 2.22 qui correspond également au bruit granulaire si tous les vecteurs X appartiennent à $\mathcal{P}^n$.

$$D_{QV} = \frac{1}{n} E [d_{QV}(X, QV(X))] = \frac{1}{n} \sum_{i=0}^{L-1} \int_{P_i} \|X - Y_i\|_2^2 f_X(X) dX \quad (2.22)$$

Cependant, pour des dictionnaires non structurés, la formule de la distorsion de la QV devient difficile à calculer. Zador [95] a calculé une borne asymptotique maximale de la distorsion 2.22 en fonction du débit R du quantificateur, en généralisant la formule de Bennett donnée dans la relation 2.10. Dans le cas d'une norme l_2 , la borne est donnée par :

$$D_{QV}(R) \leq G_n(\Lambda) 2^{-2R} \left[\int_{-\infty}^{+\infty} f_X(X)^{\frac{1}{3}} dX \right]^3 \quad (2.23)$$

où les coefficients $G_n(\Lambda)$ correspondent au second moment normalisé du quantificateur (dépend de la dimension n et de la forme du réseau Λ). Dans le cas de la quantification d'un signal uniforme sur un réseau régulier de points, l'erreur quadratique moyenne normalisée est la même pour tous les Voronoï du dictionnaire. Le calcul de $G_n(\Lambda)$ peut ainsi s'effectuer sur le Voronoï d'origine P_0 :

$$G_n(\Lambda) = \frac{1}{n} \frac{\int_{X \in P_0} \|X\|_2^2 dX}{\left(\int_{X \in P_0} dX \right)} \quad (2.24)$$

2.5.4 Méthodes par apprentissage

Comme nous l'avons expliqué, le processus de quantification consiste à associer à un vecteur source X le meilleur représentant Y_i du dictionnaire. Pour cela, il est donc nécessaire de calculer au préalable un dictionnaire $\mathcal{C}$ "optimal". Les méthodes utilisées sont basées sur les deux conditions d'optimalité définies précédemment : la règle du **plus proche voisin** et le calcul du **centroïde** afin de minimiser la distorsion D_{QV} . Nous présentons ici les principales méthodes de construction du dictionnaire par apprentissage basées sur l'algorithme de Lloyd-Max décrit dans l'étude de la Quantification Scalaire.

2.5.4.1 L'algorithme GLA

L'algorithme de Lloyd généralisé⁶ est un algorithme itératif convergent. Il nécessite une séquence d'apprentissage $\mathcal{S} = \{S_1, \dots, S_M\}$ (ou un modèle statistique de la source) et un dictionnaire initial $\mathcal{C} = \{Y_0, \dots, Y_{L-1}\}$. Les vecteurs de $\mathcal{S}$ doivent être des vecteurs représentatifs de la source à quantifier mais différents et tels que le nombre de vecteurs M respecte la relation $16L \leq M$ [38]. L'algorithme GLA à chaque itération fait diminuer la distorsion entre le dictionnaire et la source ou la laisse inchangée. La courbe de la distorsion en fonction des itérations est donc une fonction monotone

⁶ GLA : Generalized Lloyd Algorithm en anglais

décroissante. Le principe de l'algorithme est le même que dans le cas scalaire mais étendu au cas vectoriel.

Algorithme *GLA*

- 1) Choisir un dictionnaire initial $\mathcal{C}_1$. $m = 1$;
- 2) Itération de Lloyd
 - 2.1) Pour un dictionnaire $\mathcal{C}_m = \{Y_0, \dots, Y_{L-1}\}$ donné, trouver le partitionnement optimal $\mathcal{P} = \{P_0, \dots, P_{L-1}\}$ en utilisant la règle du **plus proche voisin** ;
 - 2.2) A partir du partitionnement trouvé en 2.1 et de la méthode de calcul du **centroïde**, trouver le dictionnaire optimal $\mathcal{C}_{m+1}$;
- 3) Calculer la distorsion moyenne D_{QV} obtenue avec le dictionnaire $\mathcal{C}_{m+1}$. Si cette distorsion n'évolue pratiquement plus par rapport à l'itération précédente, alors arrêter l'algorithme, sinon $m + 1 \rightarrow m$ et retourner en 2.

Le problème de cet algorithme est que le minimum obtenu en terme de distorsion est un minimum local. Or, ce minimum est plus ou moins "proche" du minimum global, selon l'initialisation du dictionnaire $\mathcal{C}$. Pour remédier à ce problème, l'algorithme *LBG* augmente à chaque itération la taille du dictionnaire en minimisant la distorsion.

2.5.4.2 L'algorithme *LBG*

L'algorithme *LBG*, du nom de ses concepteurs Linde, Buzo et Gray [54], effectue la création du dictionnaire final de taille L à partir du dictionnaire initial de taille $l = 1$. Son principe est de trouver deux nouveaux représentants pour chaque représentant Y_{l_i} du dictionnaire $\mathcal{C}_l$ en utilisant une méthode de division⁷. Cette méthode consiste à décomposer un représentant Y_{l_i} en deux nouveaux représentants $Y_{l_i} \pm \epsilon_{split}$, où ϵ_{split} est un vecteur de faible énergie par rapport au vecteur Y_{l_i} . Le nombre de représentants du dictionnaire $\mathcal{C}_l$ est de 2^l . L'algorithme *GLA* est alors appliqué sur le nouveau dictionnaire $\mathcal{C}_{l+1}$ de façon à optimiser les 2^{l+1} nouveaux vecteurs.

2.5.4.3 Autres méthodes

D'autres méthodes organisent le dictionnaire sous la forme d'un arbre (généralement binaire). Ces approches permettent d'accélérer la recherche à travers le dictionnaire lors du codage. Ces méthodes sont appelées approches arborescentes descendantes ou ascendantes⁸. Il existe également d'autres approches qui consistent à construire plusieurs dictionnaires spécialisés et à les organiser dans une arborescence, la recherche à travers les différents dictionnaires est guidée par différents critères. Ces méthodes sont appelées Quantification Vectorielle par Classification⁹. Nous proposons au lecteur de consulter l'ouvrage [38] si il souhaite obtenir de plus amples informations sur ces méthodes.

2.5.5 Complexité au codage et au décodage

Toutes les méthodes précédentes ont une phase de création du dictionnaire dont la complexité devient rapidement importante avec le nombre L de représentants et celui des représentants de la séquence d'apprentissage $\mathcal{S}$. Nous rappelons également que la phase de quantification est d'une complexité importante. En effet, la quantification d'un vecteur X par une recherche exhaustive consiste à rechercher le vecteur Y_j le plus proche (en terme de distorsion, formule 2.21) parmi les L

⁷ *splitting* en anglais

⁸ *TSVQ : Tree Search Vector Quantization* en anglais

⁹ *CVQ : Classified Vector Quantization* en anglais

vecteurs du dictionnaire. Nous pouvons donner l'estimation du nombre d'opérations de base pour la quantification d'un vecteur par :

$$N_{op} = (3Ln - L) \text{ multiplications et additions } + (L - 1) \text{ tests} \quad (2.25)$$

Malgré l'amélioration de la recherche exhaustive par une recherche à travers des dictionnaires organisés, la complexité du codage de ces méthodes reste très importante. Par contre, le décodage à l'aide du dictionnaire est très rapide. En effet, celui-ci consiste à lire simplement une table de correspondance¹⁰ qui associe à un index un vecteur.

2.5.6 Conclusion

Les méthodes par apprentissage permettent d'atteindre de bonnes performances en terme de qualité, mais au détriment d'une complexité de calcul très élevée. Une autre approche purement analytique permet de réduire considérablement ce coût calculatoire. Il s'agit essentiellement de la Quantification Vectorielle Algébrique. Cette approche associée au concept de zone morte a été retenue au sein de notre chaîne de compression.

2.6 La Quantification Vectorielle Algébrique

2.6.1 Introduction

La Quantification Vectorielle Algébrique (QVA)¹¹ ou quantification sur un réseau régulier de points a suscité de nombreux travaux durant ces dernières années, [4][22][35][34] [38][46][63][76][65][13]. Elle présente plusieurs avantages. D'une part, il n'y a pas de phase de création d'un dictionnaire. D'autre part, le processus de quantification est nettement plus rapide qu'avec les méthodes par apprentissage. Enfin, il suffit de régler un seul paramètre (un facteur de projection γ) pour atteindre le débit cible ou le taux de compression souhaité. Cependant, il est nécessaire de définir les caractéristiques du quantificateur auparavant. La QVA est ainsi caractérisée par : le type de réseau, le rayon de troncature (limitation du dictionnaire), l'opération de quantification et le type d'indexage. Nous décrirons ces différentes caractéristiques à travers les paragraphes suivants et nous expliquerons les hypothèses simplificatrices faites sur certains algorithmes.

2.6.2 Réseau régulier de points

Un réseau régulier de points Λ dans un espace $\mathbb{R}^n$ est défini par [22] :

$$\Lambda = \left\{ Y \in \mathbb{R}^n \mid \exists U = [u_1, \dots, u_n] \in \mathbb{Z}^n \text{ et } G = \begin{bmatrix} a_1 \\ \dots \\ a_n \end{bmatrix}, a_i \in \mathbb{R}^m, Y = UG = \sum_{i=1}^n u_i a_i \right\} \quad (2.26)$$

où G correspond à la matrice génératrice du réseau : les lignes de G sont des vecteurs linéairement indépendants construits à partir d'un espace $\mathbb{R}^m$ ($m \geq n$) et forment une base pour le réseau.

Nous présentons sur la figure 2.6 la forme d'un réseau de dimension $n = 2$ avec sa matrice génératrice $G = [a_1 \ a_2]^T$. Suivant la dimension n , des réseaux optimaux ont été définis, notamment à travers les travaux de Conway et Sloane [22][21]. Cependant, nous utiliserons le réseau régulier $\mathbb{Z}^n$ pour ses performances en terme de complexité de calcul lors de la phase de quantification, figure 2.5 pour $n = 2$.

¹⁰ LUT : Look Up Table en anglais

¹¹ LVQ : Lattice Vector Quantization en anglais

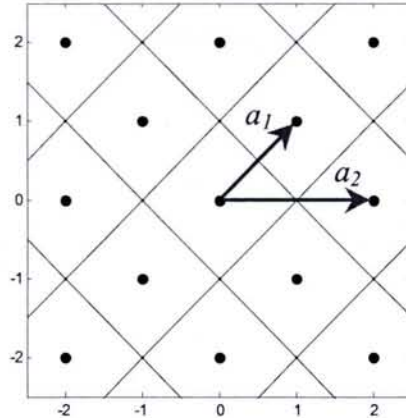


FIG. 2.6: Exemple de réseau de dimension $n = 2$ engendré par la matrice génératrice $G = [a_1 \ a_2]^T$, avec $a_1 = (1, 1)$ et $a_2 = (2, 0)$

2.6.3 Troncature du réseau $\mathbb{Z}^n$

Un réseau régulier de points de dimension n est un hyper-espace infini. Or, pour des applications réelles, nous devons définir un dictionnaire $\mathcal{C}$ fini composé de L représentants Y_i dénombrables. Pour cela, il suffit de limiter le réseau régulier de points [38][22][34][5]. Suivant le type de répartition des vecteurs X de la source (de type laplacienne ou gaussienne), nous adaptons la forme de la troncature. La figure 2.7 présente deux types de troncatures sur un réseau $\mathbb{Z}^2$ de forme pyramidale et sphérique. Cependant, il en existe bien d'autres (troncatures elliptiques [65]).

Pour déterminer la troncature, nous calculons l'énergie maximum du dictionnaire $\mathcal{C}'$ que l'on souhaite conserver, appelée aussi rayon de troncature r_T . Nous utilisons la norme l_1 , relation 2.27, qui correspond à une troncature pyramidale adaptée à une répartition de type laplacienne [35][34], ou la norme l_2 , relation 2.28, qui correspond à une troncature sphérique adaptée à une répartition de type gaussienne.

$$\gamma r_T = \|\bar{X}\|_1 = \sum_{i=1}^n |\bar{x}_i| \quad (2.27)$$

$$(\gamma r_T)^2 = \|\bar{X}\|_2^2 = \sum_{i=1}^n \bar{x}_i^2 \quad (2.28)$$

où $\bar{X}$ représente un vecteur sur la troncature d'énergie maximale γr_T . Nous disposons ainsi d'un dictionnaire $\mathcal{C}$ composé d'un nombre fini de représentants Y_i , il est donc maintenant possible de leur attribuer un index.

2.6.4 Normalisation et quantification sur $\mathbb{Z}^n$

Le processus de normalisation permet de ramener tous les vecteurs X de la source à l'intérieur du dictionnaire $\mathcal{C}'$. Cette projection des vecteurs est effectuée à l'aide du facteur de projection γ . Le dictionnaire $\mathcal{C}$ correspond donc au dictionnaire $\mathcal{C}'$ dilaté par γ . Nous pouvons assimiler γ au pas de quantification q dans le cas scalaire. Comme nous l'avons dit précédemment, le réglage de γ permet d'atteindre le débit ou le taux de compression souhaité. Nous présentons, au chapitre 3, comment l'optimiser sur chacune des sous-bandes d'une décomposition en ondelettes, afin d'atteindre le débit cible total. Les vecteurs ainsi normalisés sont ensuite quantifiés à l'aide d'algorithmes rapides définis par Conway et Sloane [22][21]. Ainsi, la quantification sur le réseau $\mathbb{Z}^n$ consistera simplement

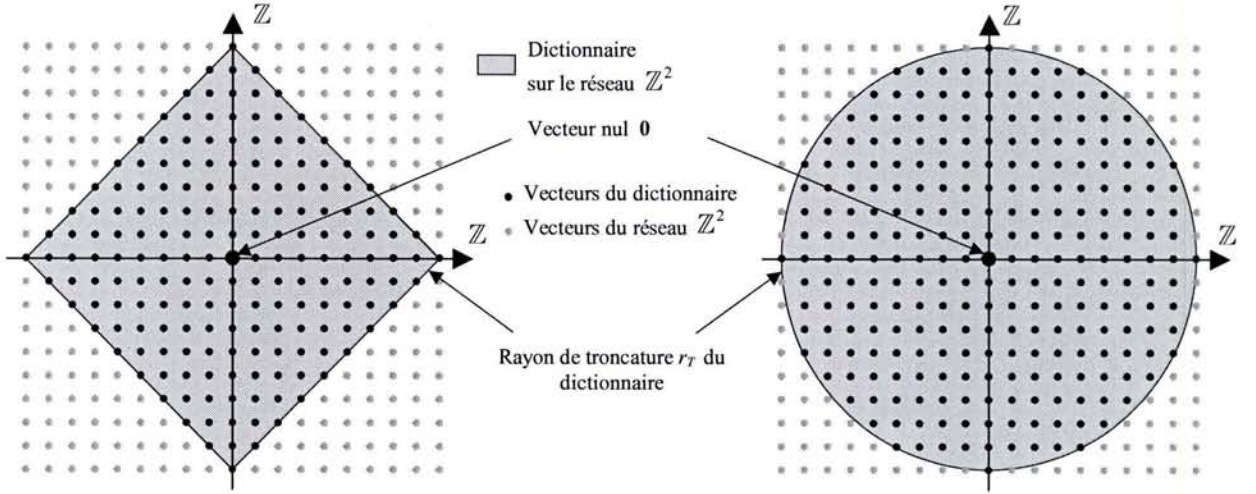


FIG. 2.7: Troncature pyramidale (à gauche) et sphérique (à droite) pour une meilleure adaptation à des sources respectivement laplacienne et gaussienne.

à remplacer chaque composante x'_i du vecteur normalisé X' par son entier le plus proche soit un vecteur Y'_i appartenant au dictionnaire $\mathcal{C}'$. Cependant, pour un dictionnaire $\mathcal{C}'$ donné de rayon de troncature r_T , il peut se produire deux cas de figures [5] :

- soit le facteur de projection γ est suffisamment important pour projeter directement le vecteur X de la source à l'intérieur du dictionnaire $\mathcal{C}'$. Dans ce cas, la quantification du vecteur normalisé X' s'effectuera normalement sur le dictionnaire $\mathcal{C}'$;
- soit le facteur de projection γ n'est pas suffisamment important, il est donc nécessaire de trouver un nouveau facteur de projection γ' qui ré-intègre le vecteur normalisé X' à l'intérieur du dictionnaire $\mathcal{C}'$. La quantification du vecteur normalisé X'' s'effectuera normalement sur le dictionnaire $\mathcal{C}'$.

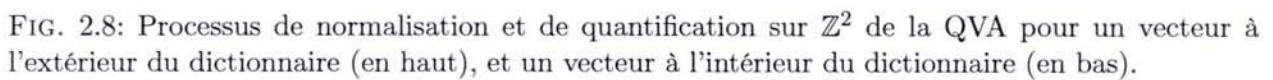
Ce processus de normalisation et de quantification sur $\mathbb{Z}^n$ est représenté sur la figure 2.8, pour une dimension $n = 2$.

2.6.5 Algorithme de la QVA

Nous présentons ici l'algorithme de la QVA pour un vecteur X . Nous définissons la fonction $[.]$ telle que $[X']$ correspond à l'entier le plus proche de chaque composante du vecteur normalisé X' . Le coefficient K_A est un nombre proche de 1 mais inférieur qui permet de ramener un vecteur quantifié Y' à l'intérieur du dictionnaire sur un des représentants Y'_i .

Algorithme de la QVA général

- 1) Normalisation du vecteur X : $X' = \frac{X}{\gamma}$
 - 1.1) Si $\|X'\|_p > r_T$
Alors calcul de $\gamma' = \frac{\|X'\|_p}{r_T}$
 - 1.2) Re-normalisation du vecteur X' : $X'' = \frac{X'}{\gamma'}$
Vecteur normalisé $X' = X''$
- 2) Quantification sur le réseau de X' : $Y' = [X']$
 - 2.1) Si $\|Y'\|_p > r_T$ (uniquement pour un dictionnaire sphérique)
Alors $X' = X' K_A$ avec $K_A = 1 - \epsilon_A$, puis retour en 2)
 - 2.2) Sinon FIN, le vecteur quantifié $QVA(X) = \gamma Y' = Y$



Dans le cadre de notre application, nous considérerons un dictionnaire "infini" (r_T très grand). Les étapes 1.1) et 1.2) ne seront pas intégrées dans l'algorithme final. Néanmoins pour être sûr que tous les vecteurs de la source puissent être quantifiés, une pré-normalisation est effectuée. Cela permet d'assurer une quantification simplifiée de tous les vecteurs. De plus, nous utiliserons un dictionnaire pyramidal ($p = 1$). Nous pouvons donc écrire l'algorithme simplifié suivant :

Algorithme de la QVA simplifié

- 1) Normalisation du vecteur X : $X' = \frac{X}{\gamma}$
- 2) Quantification sur le réseau de X' : $Y' = [X']$
 $\Rightarrow$ vecteur quantifié $QVA(X) = \gamma Y' = Y$

Nous pouvons donner l'estimation du nombre d'opérations de base pour la quantification d'un vecteur par :

$$N_{op} = (2n) \text{ divisions et multiplications} + (n) \text{ valeurs arrondies} \quad (2.29)$$

Le nombre d'opérations dépend uniquement de la dimension des vecteurs, il est donc très inférieur au nombre d'opérations des méthodes par apprentissage.

2.6.6 Méthodes d'indexage

Nous venons de montrer que la QVA permettait d'obtenir très rapidement le meilleur représentant Y_i du dictionnaire structuré sur le réseau. Cependant, l'index i de ce représentant n'est pas connu. Il est donc nécessaire d'associer un index unique i en fonction des caractéristiques du vecteur Y_i . De nombreuses méthodes d'indexage se sont ainsi développées suivant le type de réseau et en tenant compte de la complexité et du coût de stockage [16][35][63].

Plus récemment, une méthode d'indexage très rapide et ne nécessitant pas de stockage a été développée dans le cas de sources gaussienne et laplacienne [56].

Notre étude s'est ainsi basée sur la construction d'un index par code préfixe [63][35][34]. En effet, pour chaque vecteur Y_i , il est possible de calculer son énergie $\mathbf{en} = \|Y_i\|_p$ (norme l_p) et de lui associer sa position $\mathbf{pos}$ sur cette énergie. Cette position est calculée en exploitant la géométrie et la régularité du réseau. Il est possible de déterminer le nombre de positions $N(\mathbf{en})$ pour chaque énergie $\mathbf{en}$ à l'aide de séries génératrices ν [22]. Par exemple, dans le cas d'un réseau $\mathbb{Z}^2$ en considérant un dictionnaire pyramidal, les coefficients du polynôme suivant donnent le nombre de vecteurs par énergie :

$$\nu_{\mathbb{Z}^2}(z) = \left(\frac{1+z}{1-z} \right)^2 = 1 + 4z + 8z^2 + \dots \quad (2.30)$$

Nous présenterons dans la suite du mémoire la manière de stocker en binaire ce couple de valeurs $(\mathbf{en}, \mathbf{pos})$, afin de calculer le débit.

2.6.7 Distorsion de la QVA

Sur un réseau régulier de points, en tenant compte de l'hypothèse haute résolution, il est possible de calculer une approximation de la distorsion de la QVA par échantillon (pixel, coefficient, ...). Comme pour la QS, nous pouvons décomposer en deux le calcul de la distorsion de la QVA par :

$$D_{QVA} = D_{QVA}^{granulaire} + D_{QVA}^{surchage} \quad (2.31)$$

D'après la relation 2.23 établie par Zador[95], la distorsion est donnée par la formule :

$$D_{QVA}^{granulaire} \approx \frac{1}{Vol(P_0)} \int_{P_0} \|X\|_2^2 dX = \frac{1}{|\det \Lambda|} \int_{P_0} \|X\|_2^2 dX \quad (2.32)$$

où P_0 est la partition d'origine (ou le Voronoï d'origine), $Vol(P_0)$ est le volume de la partition P_0 et Λ la matrice génératrice du réseau [22]. La formule 2.32 ne traduit que le bruit granulaire de quantification, Jeong et Gibson ont établi la formule qui englobe également le bruit de surcharge [46].

La formule 2.33 (extension de la relation 2.22) donne le bruit de surcharge des vecteurs X quantifiés en dehors de l'énergie maximum $E_{max} = \gamma r_T$ (rayon de troncature sphérique du dictionnaire projeté par γ), où $\bar{X}$ correspond aux vecteurs projetés sur la surface du dictionnaire $\|\bar{X}\|_2 = \gamma r_T$. La figure 2.8 montre ce phénomène de surcharge, où $\|\bar{X}\|_2 / \gamma = \|X''\|_2 = r_T$.

$$D_{QVA}^{surcharge} = \frac{1}{n} \int_{\|X\|_2 > \gamma r_T} \|X - \bar{X}\|_2^2 f_X(X) dX \quad (2.33)$$

2.6.8 Compromis : taux de compression/ qualité/ complexité

Comme nous l'avons expliqué, la QVA est un processus de quantification de très faible complexité. Elle permet ainsi d'atteindre de très bonnes performances en termes de qualité/ complexité pour différents taux de compression. De plus, il n'y a plus de phase de création de dictionnaire, et certaines méthodes ne nécessitent pas de stockage (pas de tables regroupant le nombre de positions $N(\mathbf{en})$ pour chaque énergie $\mathbf{en}$).

2.6.9 Conclusion

La QVA est donc une méthode qui permet d'obtenir des performances supérieures par rapport aux méthodes par apprentissage. Nos travaux se sont donc focalisés sur cette méthode, et en introduisant le concept de zone morte. Nous disposons maintenant des notions fondamentales pour aborder l'étude théorique de nos travaux concernant la Quantification Vectorielle Algébrique avec Zone Morte.

2.7 La Quantification Vectorielle Algébrique avec Zone Morte

2.7.1 Définitions

L'étude de la QVA montre qu'il est possible d'atteindre de bonnes performances en termes de qualité et de complexité pour un débit fixé. Nous allons focaliser notre étude dans le cas d'un dictionnaire de forme pyramidale [35][34] (norme l_1) car la statistique des coefficients d'ondelettes est souvent modélisée par une loi laplacienne[4]. Cependant, les formules que nous développerons dans la suite du chapitre, pourront être étendues au cas sphérique (norme l_2). L'observation de la répartition spatiale des coefficients d'ondelettes montre qu'il est plus intéressant de conserver les coefficients qui correspondent à la description des contours (détails) plutôt que les coefficients (de faibles valeurs) qui correspondent à des zones homogènes [15][84]. Il est ainsi possible de supprimer ces coefficients afin de mieux quantifier les coefficients significatifs. Pour cela, nous effectuons un seuillage vectoriel δ_v adapté au dictionnaire pyramidal défini par :

$$Tv_h(X, \delta_v) = x.1 \{ \|X\|_1 \geq \delta_v \} \quad (2.34)$$

Cette technique engendre une zone où tous les vecteurs sont mis à zéro, que l'on appelle zone morte (ZM). Cette zone est représentée sur la figure 2.9. Une définition mathématique de la zone morte est donnée par :

$$ZM = \left\{ X \in \mathbb{R}^n \mid \|X\|_1 = \sum_{i=1}^n |x_i| < r_{ZM} \text{ avec } r_{ZM} \in \mathbb{R} \right\} \quad (2.35)$$

dans le cas d'un dictionnaire pyramidal (norme l_1), où X est un vecteur de coefficients d'ondelettes et r_{ZM} le rayon qui correspond au seuil de la zone morte sur le dictionnaire.

Mais, un réglage optimal de ce seuil pose plusieurs problèmes. Différentes études ont proposé des solutions mais principalement dans le cas scalaire. Certains travaux dans le cadre du débruitage d'images ont introduit la notion de seuillage vectoriel [32][85]. D'autres travaux ont introduit un seuillage vectoriel pour limiter des zones de quantification dont les caractéristiques sont différentes [48][49]. Cependant à notre connaissance, aucune étude théorique n'a à ce jour développé le concept de zone morte vectorielle optimisée avec la Quantification Vectorielle Algébrique dans le cadre d'une application de compression d'image.

Nous avons également constaté qu'il n'était pas possible de transposer l'étude théorique de la Quantification Scalaire avec Zone Morte à la Quantification Vectorielle Algébrique avec Zone Morte. En effet, la QSZM n'utilise pas un réseau régulier de points sur $\mathbb{Z}$. Nos travaux se sont donc articulés autour de l'étude théorique de la QVAZM et de l'étude de son modèle de débit entropique. Nous allons maintenant présenter ces différentes études dans le cas pyramidal.

2.7.2 Principe général

La création d'une zone morte consiste donc à effectuer un seuillage sur l'énergie des vecteurs de coefficients d'ondelettes. Tous les vecteurs X dont l'énergie est inférieure au seuil r_{ZM} sont remplacés par le vecteur d'énergie nulle $\mathbf{0}$. Nous définissons également par $\delta = \left\lceil \frac{r_{ZM}}{\gamma} \right\rceil$ la première énergie disponible du dictionnaire sur le réseau $\mathbb{Z}^n$ qui correspond à la projection du seuil r_{ZM} par le facteur γ arrondi à l'entier supérieur le plus proche¹². Nous pouvons définir maintenant l'ensemble des vecteurs du dictionnaire $\mathcal{C}_{ZM}$:

$$\mathcal{C}_{ZM} = \{Y_i \in \gamma\mathbb{Z}^n \mid \gamma\delta \leq \|Y_i\|_1 \leq \gamma r_T\} \cup \{\mathbf{0}\} \quad (2.36)$$

où Y_i est un vecteur quantifié, $\gamma\delta$ le premier rayon du dictionnaire $\mathcal{C}_{ZM}$ et γr_T le rayon de troncature du dictionnaire $\mathcal{C}_{ZM}$.

2.7.3 Forme de la zone morte

Nous avons adapté la forme de la zone morte en considérant le cas d'une répartition laplacienne des coefficients d'ondelettes. La figure 2.9 montre cette forme pour un dictionnaire pyramidal et pour une dimension de vecteur $n = 2$. Nous visualisons parfaitement le seuillage vectoriel adapté au dictionnaire.

2.7.4 Quantification des vecteurs appartenant à la zone morte

La figure 2.10 montre les régions de quantification des vecteurs (dimension $n = 2$) autour du seuil de la zone morte :

- la région de la zone morte : où tous les vecteurs X tels que $\frac{\|X\|_1}{\gamma} < \frac{r_{ZM}}{\gamma}$ sont quantifiés par le vecteur nul $\mathbf{0}$ (région en blanc) ;
- la région de quantification normale : où tous les vecteurs X tels que $\delta \leq \frac{\|X\|_1}{\gamma}$ sont quantifiés en utilisant la quantification vectorielle algébrique classique (région en gris clair) ;
- la région de surcharge : où les vecteurs X tels que $\frac{r_{ZM}}{\gamma} < \frac{\|X\|_1}{\gamma} < \delta$ sont quantifiés sur les vecteurs Y' d'énergie δ (région en gris foncé). Nous appellerons la région de surcharge dans la suite du mémoire : l'association de la région en noir et en gris foncé.

¹² $\lceil \cdot \rceil$ désigne l'entier supérieur le plus proche

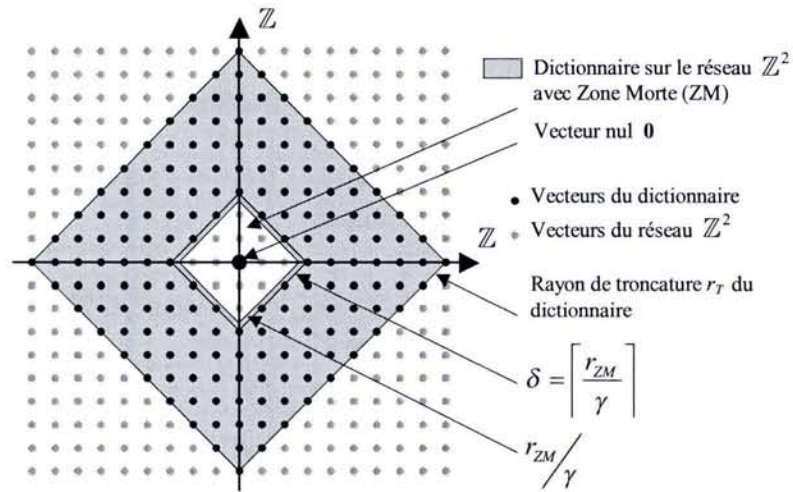
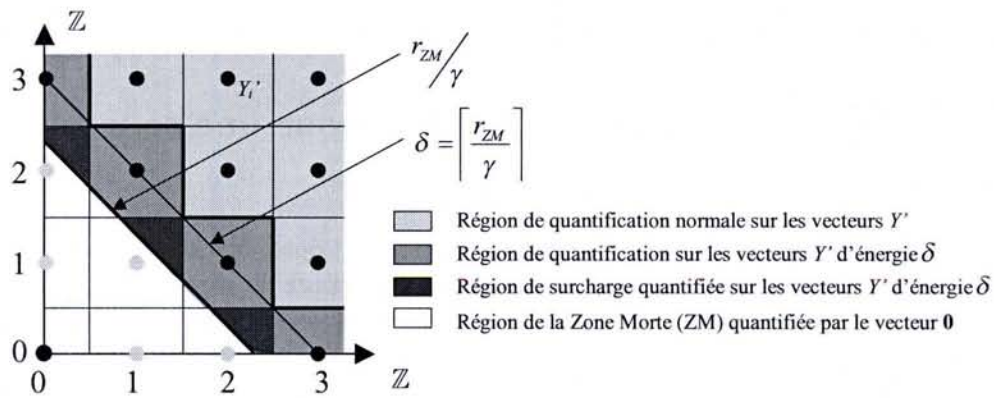
FIG. 2.9: Dictionnaire pyramidal sur un réseau $\mathbb{Z}^2$ avec zone morte.

FIG. 2.10: Zoom sur le dictionnaire.

2.7.5 Algorithme de la QVAZM

Nous présentons ici l'algorithme simplifié de la Quantification Vectorielle Algébrique avec Zone Morte pour le traitement d'un vecteur X . Le coefficient $K_B = 1 + \epsilon_B$ est un nombre proche de 1 qui permet de ramener un vecteur appartenant à la région de surcharge sur un représentant Y' d'énergie δ (région en gris foncé) du dictionnaire.

Algorithme de la QVAZM

- 0) Si $\|X\|_1 < r_{ZM}\gamma$ Alors $X = 0 \Rightarrow Y = 0$, FIN, Sinon 1)
- 1) Normalisation du vecteur X : $X' = \frac{X}{\gamma}$
- 2) Quantification du vecteur X' : $Y' = [X']$
- 3) Cas où Y' n'appartient pas au dictionnaire :
 - 3.1) Si $\|Y'\|_1 < \delta$ avec $\delta = \left\lceil \frac{r_{ZM}}{\gamma} \right\rceil$
 Alors $X' = X'K_B$ avec $K_B = 1 + \epsilon_B$, puis retour en 2)
 - 3.2) Sinon FIN, $\Rightarrow QVA(X) = Y'\gamma = Y$

2.7.6 Compromis : taux de compression / qualité / complexité

Nous montrerons que la QVAZM permet d'obtenir de meilleures performances en terme de qualité que la QVA. De plus, elle n'augmente pas la complexité des calculs pour les vecteurs tels que $\frac{\|X\|_1}{\gamma} > \delta$. Ce sont uniquement certains vecteurs tels que $\frac{r_{ZM}}{\gamma} < \frac{\|X\|_1}{\gamma} < \delta$ qu'il faut intégrer à la zone de surcharge, qui augmenteront la complexité. Le nombre d'opérations supplémentaires dépend du nombre de vecteurs à l'intérieur de la région de surcharge (il est cependant nécessaire d'ajouter 2 tests).

2.7.7 Conclusion

Nous disposons maintenant d'un algorithme de quantification simple et efficace. Nous montrerons que les résultats expérimentaux de la QVAZM améliorent la qualité sur certaines sous-bandes. Notre problème consiste donc maintenant à régler le plus efficacement possible le facteur de projection γ et le seuil de la zone morte r_{ZM} suivant les objectifs à atteindre. Nous présentons une étude du modèle du débit de la QVAZM pour une source laplacienne qui permettra d'améliorer conjointement l'estimation de γ et de r_{ZM} pour atteindre le débit cible d'une sous-bande.

2.8 Modèle du débit de la QVAZM pour une source laplacienne

2.8.1 Modèle du débit entropique de la QVA

L'objectif de cette étude est d'accélérer le processus d'optimisation des débits pour l'ensemble des sous-bandes (voir chapitre 3). Nous développerons l'étude du modèle de débit entropique dans le cas d'une source laplacienne. Dans notre cas, le modèle du débit est décomposé en deux : le nombre de bits b_{en} pour coder l'énergie et le nombre de bits b_{pos} pour coder le nombre de positions $N(r)$ sur un rayon r . Le nombre de bits total pour coder un vecteur sur un rayon r est donné par :

$$b(r) = b_{en}(r) + b_{pos}(r) \quad (2.37)$$

Ce modèle nous permet de connaître le nombre de bits nécessaires pour coder un vecteur. Dans le cas d'un code naturel, le nombre de bits pour coder en binaire l'énergie $en = \|X\|_1 = r$ dépend du rayon maximum r_T (rayon de troncature du dictionnaire). Le nombre de bits est donné par $b_{en}(r) = \lceil \log_2(r_T) \rceil$. Le nombre de bits pour coder le nombre de positions $N(r)$ sur un rayon r est donné par $b_{pos}(r) = \lceil \log_2(N(r)) \rceil$. Par contre, dans le cas d'un code entropique, le nombre de

bits pour coder l'énergie **en** dépend de la probabilité d'apparition du rayon $p_{\|\mathbf{Y}\|_1}(r)$ de l'énergie r des vecteurs $\mathbf{Y}$ du dictionnaire. Le nombre de bits est donné par $b_{\text{en}}(r) = -\log_2(p_{\|\mathbf{Y}\|_1}(r))$. Le nombre de bits $b_{\text{pos}}(r)$ reste le même. Nous pouvons donc écrire le nombre de bits total $b(r)$ pour un rayon r pour un codage entropique par :

$$b(r) = -\log_2(p_{\|\mathbf{Y}\|_1}(r)) + \log_2(N(r)) \quad \text{bits} \quad (2.38)$$

Nous pouvons maintenant écrire le modèle du débit entropique R_e de la QVA pour des vecteurs de dimension n en tenant compte de la probabilité d'apparition du rayon $p_{\|\mathbf{Y}\|_1}(r)$ et du nombre de positions $N(r)$:

$$R_e(\gamma) = \frac{1}{n} \left[\sum_{r=0}^{r_T} p_{\|\mathbf{Y}\|_1}(r) \left(-\log_2(p_{\|\mathbf{Y}\|_1}(r)) + \log_2(N(r)) \right) \right] \quad (2.39)$$

Il apparaît clairement que l'estimation théorique du débit entropique de R_e dépend principalement de l'estimation de la probabilité d'apparition du rayon $p_{\|\mathbf{Y}\|_1}(r)$. Nous présenterons ultérieurement les travaux qui ont permis de donner un modèle mathématique de $p_{\|\mathbf{Y}\|_1}(r)$ [76].

2.8.2 Modèle du débit entropique de la QVAZM

Afin d'établir le modèle du débit entropique de la QVAZM, celui-ci a été décomposé en trois sous-débits correspondant à trois régions du dictionnaire. En observant la figure 2.10, nous avons établi :

- le débit R_e^{ZM} pour la région de la zone morte ;
- le débit R_e^{surchage} pour la région de surcharge associée à la région de quantification sur les vecteurs $\mathbf{Y}'$ d'énergie δ ;
- le débit $R_e^{\text{dictionnaire}}$ pour la région de quantification normale sur le dictionnaire.

Nous pouvons ainsi écrire le débit total :

$$R_{ZM}(\gamma, r_{ZM}) = R_e^{ZM} + R_e^{\text{surchage}} + R_e^{\text{dictionnaire}} \quad (2.40)$$

En observant la figure 2.10, et en tenant compte de l'algorithme de la QVAZM, nous avons également défini la probabilité d'apparition du rayon $p_{ZM}(r)$ des vecteurs $\mathbf{Y}'$ du dictionnaire $\mathcal{C}'_{ZM}$ avec zone morte, en fonction de la probabilité d'apparition des rayons discrétisés du réseau $p_{\|\mathbf{Y}\|_1}(r)$, $r \in \mathbb{N}$ et de la fonction de répartition $F_{\|\mathbf{X}\|_1}(r_{ZM})$:

$$p_{ZM}(r) = \begin{cases} F_{\|\mathbf{X}\|_1}(r_{ZM}) & , \quad r = 0 \\ 0 & , \quad 1 \leq r \leq \delta - 1 \\ \sum_{r=0}^{\delta} p_{\|\mathbf{Y}\|_1}(r) - F_{\|\mathbf{X}\|_1}(r_{ZM}) & , \quad r = \delta \\ p_{\|\mathbf{Y}\|_1}(r) & , \quad \delta + 1 \leq r \leq r_T \end{cases} \quad (2.41)$$

$$\text{avec } F_{\|\mathbf{X}\|_1}(r_{ZM}) = f_{\|\mathbf{X}\|_1}(r < r_{ZM}) \quad r \in \mathbb{R} \quad (2.42)$$

Nous pouvons ainsi écrire à l'aide de $F_{\|\mathbf{X}\|_1}(r_{ZM})$, de $p_{\|\mathbf{Y}\|_1}(r)$ et du modèle du débit entropique de la QVA $R_e(\gamma)$, le détail des trois sous-débits de $R_{ZM}(\gamma, r_{ZM})$:

$$\begin{aligned} R_e^{ZM} &= -\frac{1}{n} \left[F_{\|\mathbf{X}\|_1}(r_{ZM}) \log_2 F_{\|\mathbf{X}\|_1}(r_{ZM}) \right] \\ R_e^{surchage} &= -\frac{1}{n} \left[\left(\sum_{r=0}^{\delta} p_{\|\mathbf{Y}\|_1}(r) - F_{\|\mathbf{X}\|_1}(r_{ZM}) \right) \left(\log_2 \left(\sum_{r=0}^{\delta} p_{\|\mathbf{Y}\|_1}(r) - F_{\|\mathbf{X}\|_1}(r_{ZM}) \right) - \log_2 N(\delta) \right) \right] \\ R_e^{dictionnaire} &= -\frac{1}{n} \left[\sum_{r=\delta+1}^{r_T} p_{\|\mathbf{Y}\|_1}(r) \left(\log_2 p_{\|\mathbf{Y}\|_1}(r) - \log_2 N(r) \right) \right] \end{aligned} \quad (2.43)$$

Il est donc finalement possible de donner une formule du débit total $R_{ZM}(\gamma, r_{ZM})$:

$$\begin{aligned} R_{ZM}(\gamma, r_{ZM}) &= -\frac{1}{n} \left[F_{\|\mathbf{X}\|_1}(r_{ZM}) \log_2 F_{\|\mathbf{X}\|_1}(r_{ZM}) \right] \\ &\quad -\frac{1}{n} [p_{ZM}(\delta) (\log_2 p_{ZM}(\delta) - \log_2 N(\delta))] \\ &\quad -\frac{1}{n} \left[\sum_{r=\delta+1}^{r_T} p_{ZM}(r) (\log_2 p_{ZM}(r) - \log_2 N(r)) \right] \end{aligned} \quad (2.44)$$

Nous allons maintenant donner une description précise des trois débits afin d'obtenir le modèle du débit entropique de la QVAZM pour un dictionnaire pyramidal sur un réseau Z^n et pour une source laplacienne.

2.8.3 Modèle du débit R_e^{ZM}

Pour obtenir le modèle du débit R_e^{ZM} , nous allons d'abord définir la densité de probabilité $f_{\|\mathbf{X}\|_1}(r)$, ensuite nous calculerons la fonction de répartition $F_{\|\mathbf{X}\|_1}(r)$ qui nous permettra de calculer $f_{\|\mathbf{X}\|_1}(r < r_{ZM})$. Comme nous l'avons expliqué précédemment (voir chapitre 1), la répartition des coefficients d'ondelettes peut être modélisée par une loi laplacienne [4][46]. Considérons un ensemble de coefficients d'ondelettes x d'une sous-bande, indépendants et identiquement distribués, ceux-ci sont découpés en vecteurs X de dimension n . La densité de probabilité $f_{\mathbf{X}}(X)$ des vecteurs X de la source est définie par :

$$f_{\mathbf{X}}(X) = \left(\frac{\lambda}{2} \right)^n \exp(-\lambda \|\mathbf{X}\|_1) \quad \text{avec} \quad \lambda = \frac{\sqrt{2}}{\sigma} \quad (2.45)$$

où σ correspond à l'écart-type de la distribution d'une des composantes du vecteur.

Ainsi, tous les vecteurs X situés sur une même surface (ou rayon) ont la même densité de probabilité et vérifient :

$$\|\mathbf{X}\|_1 = \text{constante} \quad (2.46)$$

Nous définissons l'énergie d'un vecteur X , comme l'équation d'une hyper-pyramide de probabilité de rayon r :

$$\|\mathbf{X}\|_1 = \sum_{i=1}^n |x_i| = r \quad (2.47)$$

Nous pouvons maintenant écrire la densité de probabilité des rayons $f_{\|\mathbf{X}\|_1}(r)$ (loi du rayon),

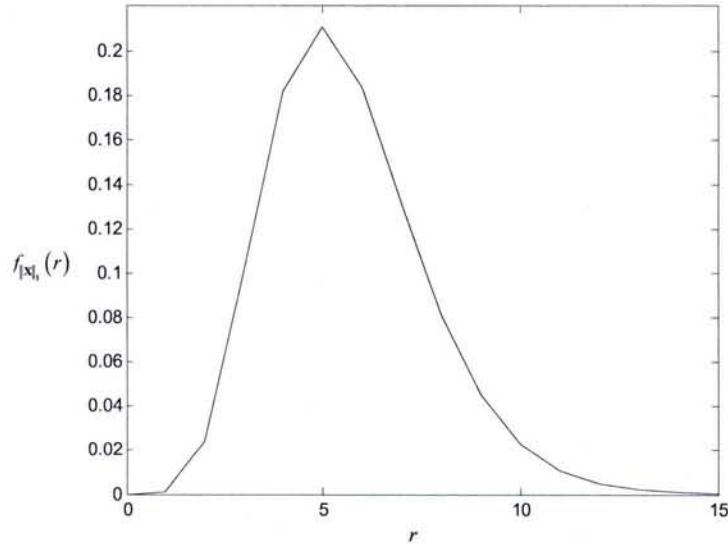


FIG. 2.11: Densité de probabilité du rayon $f_{\|\mathbf{x}\|_1}(r)$ d'une source laplacienne de moyenne nulle et d'écart-type $\sigma = 1$, pour des vecteurs de dimension $n = 8$.

qui correspond à la répartition des hyper pyramides de rayon r [46] :

$$f_{\|\mathbf{x}\|_1}(r) = \frac{\lambda^n}{\Gamma(n)} r^{n-1} \exp(-\lambda r) \quad (2.48)$$

avec $\Gamma(n) = \int_0^\infty u^{n-1} e^{-u} du \quad \forall n \in \mathbb{R}^+ \text{ ou } \Gamma(n) = (n-1)! \quad \forall n \in \mathbb{N}^*$

Nous disposons de la densité de probabilité du rayon $f_{\|\mathbf{x}\|_1}(r)$ pour une répartition laplacienne des vecteurs de coefficients d'ondelettes X (non quantifiés sur le réseau régulier de points). La figure 2.11 donne l'allure de $f_{\|\mathbf{x}\|_1}(r)$ d'une source laplacienne de moyenne nulle et d'écart-type $\sigma = 1$, pour des vecteurs de dimension $n = 8$.

Pour calculer $f_{\|\mathbf{x}\|_1}(r < r_{ZM})$, nous avons calculé la fonction de répartition $F_{\|\mathbf{x}\|_1}(r_{ZM})$ qui est obtenue par une succession d'intégrations par parties de $f_{\|\mathbf{x}\|_1}(r)$. Dans le cas d'une source laplacienne ($\alpha = 1$), $F_{\|\mathbf{x}\|_1}(r_{ZM})$ est donnée par la formule 2.49, et représentée sur la figure 2.12.

$$F_{\|\mathbf{x}\|_1}(r_{ZM}) = 1 - \exp(-\lambda r_{ZM}) \sum_{k=0}^{n-1} \frac{(\lambda r_{ZM})^k}{k!} \quad \text{avec } \lambda = \frac{\sqrt{2}}{\sigma} \quad (2.49)$$

Nous présentons en annexe A la démonstration du calcul de fonction de répartition $F_{\|\mathbf{x}\|_1}(r)$ dans le cas d'une source laplacienne ($\alpha = 1$) et gaussienne ($\alpha = 2$) pour $n \in \mathbb{N}$:

$$F_{\|\mathbf{x}\|_1}(r) = 1 - \exp(-(br)^\alpha) \sum_{k=0}^{\frac{n}{\alpha}-1} \frac{(br)^{\alpha k}}{\Gamma(k+1)} \quad \forall n, k \in \mathbb{N} \quad (2.50)$$

$$\text{avec } \alpha \in \{1, 2\} \quad , \quad b = \frac{1}{\sigma} \sqrt{\frac{\Gamma(\frac{3}{\alpha})}{\Gamma(\frac{1}{\alpha})}} \quad \text{et } \sigma \text{ l'écart-type}$$

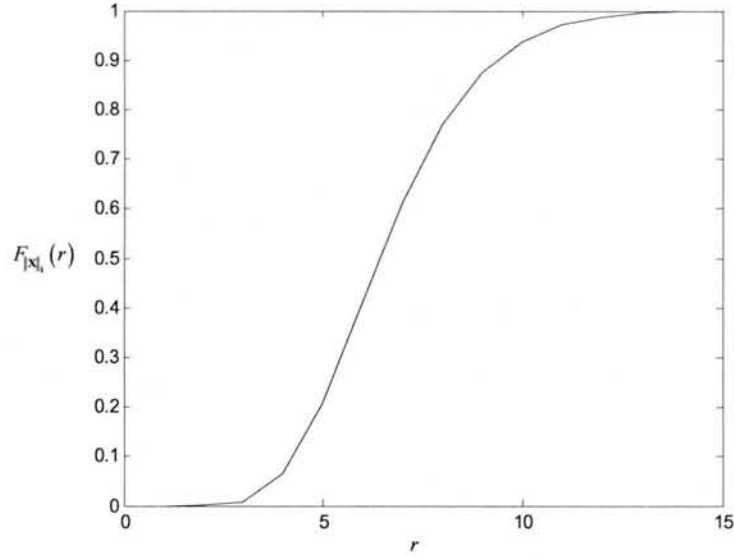


FIG. 2.12: Fonction de répartition $F_{\|\mathbf{x}\|_1}(r)$ dans le cas d'une source laplacienne de moyenne nulle et d'écart-type $\sigma = 1$, pour des vecteurs de dimension $n = 8$.

Il est maintenant possible de donner une description précise du modèle du débit R_e^{ZM} :

$$R_e^{ZM} = -\frac{1}{n} \left[\left(1 - \exp(-\lambda r_{ZM}) \sum_{k=0}^{n-1} \frac{(\lambda r_{ZM})^k}{k!} \right) \log_2 \left(1 - \exp(-\lambda r_{ZM}) \sum_{k=0}^{n-1} \frac{(\lambda r_{ZM})^k}{k!} \right) \right] \quad (2.51)$$

2.8.4 Modèle du débit $R_e^{dictionnaire}$

Pour définir le modèle du débit $R_e^{dictionnaire}$, nous devons établir avec précision la probabilité d'apparition $p_{\|\mathbf{Y}\|_1}(r)$ d'une source laplacienne. Différents travaux [76][68] s'articulent autour de l'évaluation de la loi du rayon sur des réseaux réguliers de points. Ces travaux ont permis d'établir la formule analytique 2.52 valable sur le réseau $\mathbb{Z}^n$.

$$p_{\|\mathbf{Y}\|_p}(\|QV(X)\|_p = \gamma r) = p_{\|\mathbf{Y}\|_p}(r)$$

$$p_{\|\mathbf{Y}\|_p}(r) = \sum_{y_1=-\infty}^{+\infty} \dots \sum_{y_n=-\infty}^{+\infty} \delta \left(\sum_{i=1}^n |y_i|^p - r^p \right) \int_{-\frac{\gamma}{2}}^{+\frac{\gamma}{2}} \dots \int_{-\frac{\gamma}{2}}^{+\frac{\gamma}{2}} f_{\mathbf{X}}(x_1 - y_1\gamma, \dots, x_n - y_n\gamma) dx_1 \dots dx_n \quad (2.52)$$

Où δ correspond à la fonction de Kronecker :

$$\delta(u) = \begin{cases} 1 & \text{si } u = 0 \\ 0 & \text{sinon} \end{cases} \quad \text{Remarque : } \overline{\delta(u)} = \begin{cases} 0 & \text{si } u = 0 \\ 1 & \text{sinon} \end{cases} \quad (2.53)$$

En associant la formule analytique 2.52 à l'étude précédente de la densité de probabilité du rayon $f_{\|\mathbf{x}\|}(r)$ (loi du rayon) d'une distribution laplacienne ($p = 1$), une solution analytique de la $p_{\|\mathbf{Y}\|_1}(r)$ a été proposée [68] :

$$p_{\|\mathbf{Y}\|_1}(r) = p_{\mathbf{Y}}(\|QV(X)\|_1 = \gamma r)$$

$$p_{\|\mathbf{Y}\|_1}(r) = \sum_{n_0=0}^n \text{Card}(r, n_0) \times \left(1 - \exp\left(-\frac{\lambda\gamma}{2}\right)\right)^{n_0} \times \left(\sinh\left(\frac{\lambda\gamma}{2}\right)\right)^{n-n_0} \times \exp(-\lambda\gamma r) \quad (2.54)$$

avec

$$\text{Card}(r, n_0) = \text{cardinal} \left(\left\{ \begin{array}{l} Y = \{y_1, \dots, y_n\} \in \mathbb{Z}^n \mid \sum_{i=1}^n |y_i| = r \\ \text{et } \text{cardinal}(\{y_i \mid y_i = 0\}) = n_0 \end{array} \right\} \right) \quad (2.55)$$

Où n_0 correspond au nombre de composantes nulles du vecteur Y .

L'énumération du nombre de vecteurs $\text{Card}(r, n_0)$ sur chaque rayon r du réseau $\mathbb{Z}^n$ ayant n_0 composantes nulles, formule 2.55, peut être calculée en utilisant une partie des méthodes d'indexage par code préfixe [63]. Nous avons ainsi établi la formule analytique suivante (démonstration en annexe B) :

$$\text{Card}(r, n_0) = 2^{n-n_0} \frac{n!}{n_0!} \sum_{y_{i_1}=1}^{\left\lfloor \frac{r}{n_0} \right\rfloor} \sum_{y_{i_2}=y_{i_1}}^{\left\lfloor \frac{r-y_{i_1}}{n_0-1} \right\rfloor} \dots \sum_{y_{i_{\overline{n_0}}}=y_{i_{(\overline{n_0}-1)}}}^{\left\lfloor r - \sum_{j=1}^{\overline{n_0}-1} y_{i_j} \right\rfloor} \left(\prod_{y_{i_k}=y_{i_1}}^{\left\lfloor r - \sum_{j=1}^{\overline{n_0}-1} y_{i_j} \right\rfloor} \delta(y_{i_k} - y_{i_{(k-1)}}) \times w(y_{i_k})! \right)^{-1} \quad (2.56)$$

$$\text{tel que } \sum_{j=1}^{\overline{n_0}} y_{i_j} = r \quad \text{et} \quad w(y_{i_k}) = \sum_{q=y_{i_1}}^{y_{i_{\overline{n_0}}}} \delta(y_{i_k} - q) \quad (2.57)$$

où $\delta(u)$ est la fonction de Kronecker (relation 2.53), $\overline{n_0}$ correspond au nombre de composantes non nulles du vecteur Y et y_{i_k} est la $i^{\text{ème}}$ composante non nulle.

Finalement, à l'aide des relations 2.54 et 2.56 nous pouvons calculer le modèle du débit $R_e^{\text{dictionnaire}}$ pour une distribution laplacienne :

$$R_e^{\text{dictionnaire}} = -\frac{1}{n} \left[\sum_{r=\delta+1}^{r_T} p_{\|\mathbf{Y}\|_1}(r) \left(\log_2 p_{\|\mathbf{Y}\|_1}(r) - \log_2 N(r) \right) \right] \quad (2.58)$$

où $N(r)$ correspond au nombre de vecteurs sur le rayon r , et $\delta + 1$ est le rayon à partir duquel la quantification des vecteurs se fait normalement par une QVA.

2.8.5 Modèle du débit $R_e^{\text{surcharge}}$

Pour établir le modèle du débit $R_e^{\text{surcharge}}$, nous utilisons les résultats obtenus aux paragraphes précédents, à savoir $f_{\|\mathbf{X}\|_1}(r < r_{ZM})$ et $p_{\|\mathbf{Y}\|_1}(r)$. Nous pouvons ainsi écrire :

$$R_e^{\text{surcharge}} = -\frac{1}{n} \left[\left(\sum_{r=0}^{\delta} p_{\|\mathbf{Y}\|_1}(r) - p_{ZM}(0) \right) \left(\log_2 \left(\sum_{r=0}^{\delta} p_{\|\mathbf{Y}\|_1}(r) - p_{ZM}(0) \right) - \log_2 N(\delta) \right) \right] \quad (2.59)$$

Les formules 2.51, 2.58 et 2.59 nous conduisent donc à la description du modèle du débit entropique de la QVAZM pour une source laplacienne suivante :

$$\begin{aligned} R_{ZM}(\gamma, r_{ZM}) &= -\frac{1}{n} \left[F_{\|\mathbf{X}\|_1}(r_{ZM}) \log_2 F_{\|\mathbf{X}\|_1}(r_{ZM}) \right] \\ &\quad -\frac{1}{n} \left[p_{ZM}(\delta) (\log_2 p_{ZM}(\delta) - \log_2 N(\delta)) \right] \\ &\quad -\frac{1}{n} \left[\sum_{r=\delta+1}^{r_T} p_{ZM}(r) (\log_2 p_{ZM}(r) - \log_2 N(r)) \right] \end{aligned} \quad (2.60)$$

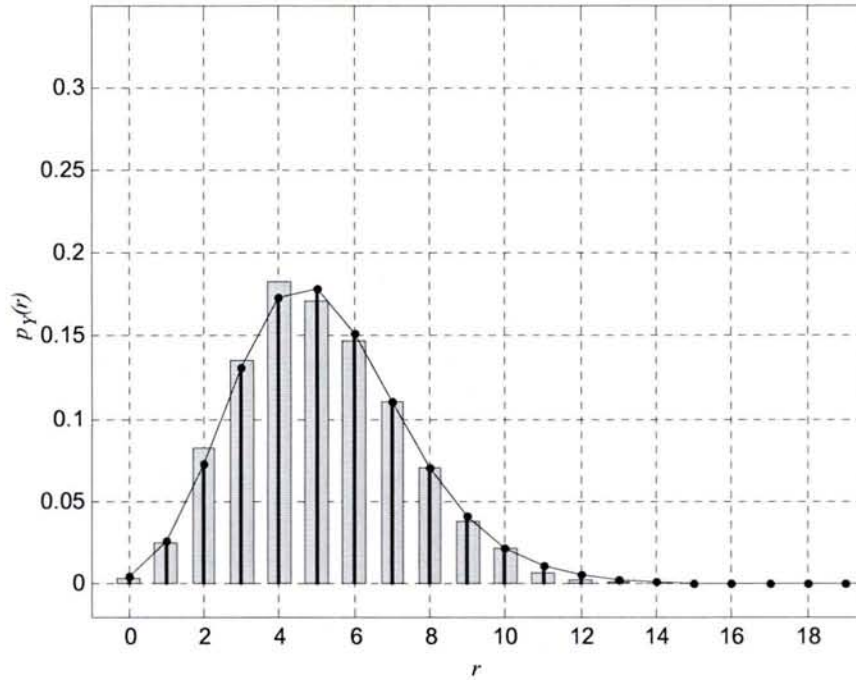


FIG. 2.13: Description de la loi du rayon de la QVA avec $n = 8$, sans zone morte ($\delta = 0$) et $\gamma = 1$ pour une source laplacienne de moyenne nulle et d'écart-type $\sigma = 1$.

2.8.6 Simulation de $p_{ZM}(r)$

La figure 2.13 montre la répartition théorique $p_{\|\mathbf{Y}\|_1}(r)$ des différents rayons d'énergies (loi du rayon) d'une QVA sans zone morte avec $n = 8$ et $\gamma = 1$ pour une source laplacienne de moyenne nulle et d'écart-type $\sigma = 1$. Les pics en gras correspondent à la probabilité d'apparition du rayon théorique de la source laplacienne sans zone morte (le trait continu lie simplement le sommet de ces pics pour visualiser l'allure de la répartition). Le graphique en barres correspond aux relevés expérimentaux. La figure 2.14 montre la répartition $p_{ZM}(r)$ d'une QVA avec une zone morte ($\frac{r_{ZM}}{\gamma} = 3,1$ soit $\delta = 4$) qui tient compte de la forme du dictionnaire $\mathcal{C}_{ZM}$. Le pic au premier rayon δ correspond à la région de surcharge décrite précédemment sur la figure 2.10.

Nous pouvons constater que le modèle théorique de $p_{ZM}(r)$ pour une distribution laplacienne des coefficients d'ondelettes est relativement précis. Cette étude nous permet ensuite de comparer le modèle du débit théorique et pratique.

2.8.7 Simulation du modèle $R_{ZM}(\gamma, r_{ZM})$

La figure 2.15 montre l'évolution théorique du débit $R_{ZM}(\gamma, r_{ZM})$ en fonction des couples (γ, r_{ZM}) pour une source laplacienne (dimension des vecteurs $n = 8$) de moyenne nulle et d'écart-type $\sigma = 1$. Cette figure montre comment évolue le débit $R_{ZM}(\gamma, r_{ZM})$ suivant le facteur de projection γ et suivant le rayon de la zone morte r_{ZM} . La figure 2.16 correspond à une coupe horizontale pour un débit de 1,0 bpp. Nous disposons ainsi de tous les couples (γ, r_{ZM}) qui permettent d'atteindre un débit fixé ($R = 1,0$ bpp). Notre modèle semble être d'autant plus précis que la zone morte est importante.

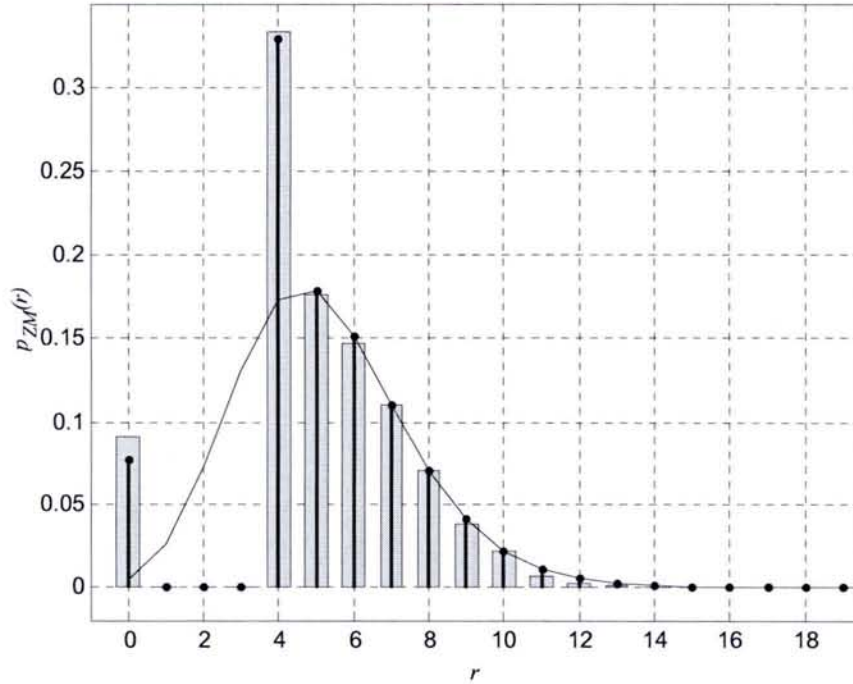


FIG. 2.14: Description de la loi du rayon de la QVA avec $n = 8$, avec zone morte ($\delta = 4$) et $\gamma = 1$ pour une source laplacienne de moyenne nulle et d'écart-type $\sigma = 1$.

2.9 Résultats sur les différentes sous-bandes

2.9.1 Performances

Les résultats que nous présentons dans les tableaux 2.1 et 2.2, sont issus de la quantification de l'image *Lena* pour un débit de 0,125 bpp (TC = 64 : 1). Nous rappelons simplement que notre chaîne de traitement est basée sur une transformée en ondelettes à 5 niveaux de résolution qui utilise le filtre 9-7 [4][1]. De plus, la répartition des débits sur chaque sous-bande a été établie à l'aide d'un algorithme d'optimisation [75] associé à la QVAZM. Nous détaillerons largement cet algorithme dans le chapitre 3. Le tableau 2.1 montre l'amélioration que la QVAZM apporte par rapport à la QVA classique. Nous pouvons constater une diminution de plus de 14 % sur la distorsion de certaines sous-bandes, et une amélioration globale avec une diminution de 7 % sur l'ensemble des sous-bandes de la transformée en ondelettes.

Nous constatons également une amélioration d'environ 0,2 dB entre le *PSNR* de la QVA classique et le *PSNR* de la QVAZM de l'image *Lena* reconstruite (voir tableau 2.2), c'est-à-dire après une transformée en ondelettes inverse.

2.9.2 Intérêt du modèle

Le modèle du débit de la QVAZM pour une source laplacienne (de moyenne nulle et d'écart-type σ) nous permet d'avoir une estimation des couples (γ, r_{ZM}) qui permettent d'atteindre un débit cible pour une sous-bande donnée. Il sera ensuite possible de trouver le couple (γ^*, r_{ZM}^*) "optimal" qui minimise la distorsion de la quantification de la source. Cette approche est le point clé de notre méthode. Cependant, la complexité de notre algorithme est encore importante.

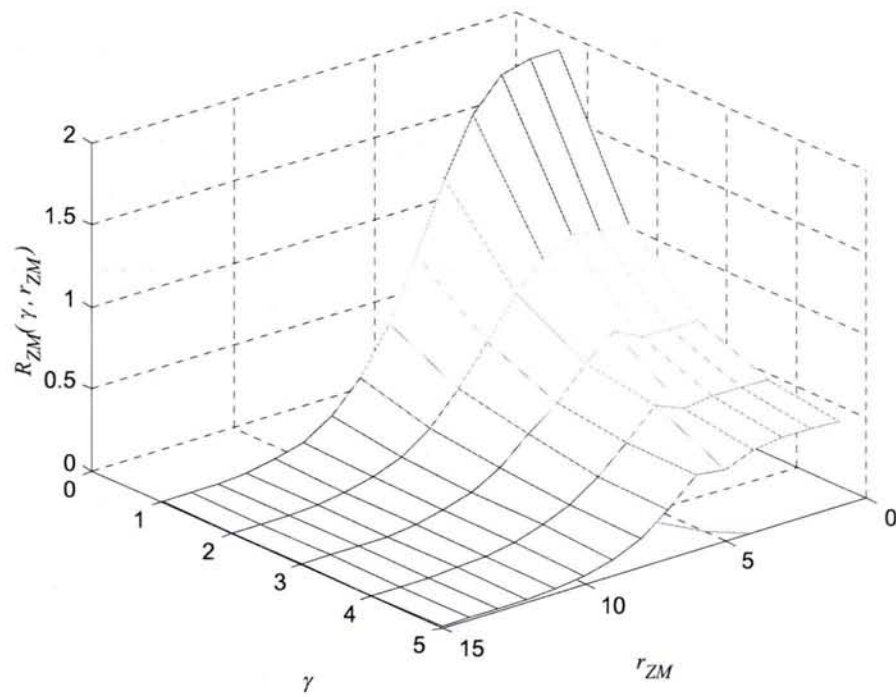


FIG. 2.15: Evolution théorique du débit $R_{ZM}(\gamma, r_{ZM})$ en fonction des couples (γ, r_{ZM}) pour une source laplacienne (moyenne nulle, $\sigma = 1$). L'axe vertical correspond aux différents débits, l'axe de gauche à l'évolution du facteur de projection γ et l'axe de droite à l'évolution du rayon de la zone morte r_{ZM} .

TAB. 2.1: Comparaison des performances QVA/ QVAZM sur les sous-bandes de l'image *Lena*.

Sous bandes	Type	R_e (bpp)	γ	r_{ZM}/γ	$r_{\max}$	D_{SB}	ΔD , Gain
V5 (SB_0)	QVA	4,8813	1,3704	0,0	91	0,1500	-0,0248
	QVAZM	4,8814	1,4219	0,0	87	0,1748	—
H5 (SB_1)	QVA	3,6268	1,4550	0,0	46	0,1892	0,0027
	QVAZM	3,6442	1,4082	1,7295	47	0,1865	1,45 %
D5 (SB_2)	QVA	3,6283	1,4774	0,0	37	0,1885	-0,0110
	QVAZM	3,6252	1,5239	0,0	35	0,1995	—
V4 (SB_3)	QVA	3,1310	3,1981	0,0	35	0,8075	-0,0058
	QVAZM	3,1273	3,2090	0,0	35	0,8133	—
H4 (SB_4)	QVA	1,9957	3,4362	0,0	17	0,7941	0,0264
	QVAZM	1,9941	2,9619	1,6525	19	0,7677	3,44 %
D4 (SB_5)	QVA	1,9923	3,1990	0,0	19	0,6283	0,276
	QVAZM	1,9830	2,7285	1,7295	24	0,6007	4,59 %
V3 (SB_6)	QVA	1,3314	8,3184	0,0	29	3,5678	0,1071
	QVAZM	1,3359	6,7949	3,3820	36	3,4607	3,09 %
H3 (SB_7)	QVA	0,7941	7,9774	0,0	13	2,3599	0,1454
	QVAZM	0,7979	6,0918	2,9311	16	2,2145	6,57 %
D3 (SB_8)	QVA	0,8065	7,6218	0,0	14	2,0137	0,2450
	QVAZM	0,8082	5,7803	2,9787	19	1,7687	13,85 %
V2 (SB_9)	QVA	0,2949	23,5780	0,0	10	10,8316	1,3923
	QVAZM	0,2949	14,5835	3,7295	16	9,4393	14,75 %
H2 (SB_{10})	QVA	0,1376	23,6807	0,0	6	6,3719	0,7164
	QVAZM	0,1376	13,7202	2,9787	9	5,6555	12,67 %
D2 (SB_{11})	QVA	0,0996	22,0549	0,0	5	5,3590	0,6912
	QVAZM	0,1000	11,0087	3,9311	9	4,6678	14,81 %
V1 (SB_{12})	QVA	0,0	105,645	0,0	0	10,7802	0,0
	QVAZM	0,0	105,645	0,0	0	10,7802	0 %
H1 (SB_{13})	QVA	0,0	102,188	0,0	0	4,3281	0,0
	QVAZM	0,0	102,188	0,0	0	4,3281	0 %
D1 (SB_{14})	QVA	0,0	97,846	0,0	0	2,1956	0,0
	QVAZM	0,0	97,846	0,0	0	2,1956	0 %
Toutes les SB de la TO	QVA	0,1252	—	—	—	50,6474	3,3122
	QVAZM	0,1253	—	—	—	47,3352	7,00 %

TAB. 2.2: Comparaison des performances QVA/ QVAZM sur l'image *Lena* reconstruite.

	Type	R_e (bpp)	D	SNR	$PSNR$	$\Delta PSNR$
Image <i>Lena</i>	QVA	0,1252	54,8284	25,0844	30,7407	0,1884
	QVAZM	0,1253	52,5019	25,2727	30,9291	

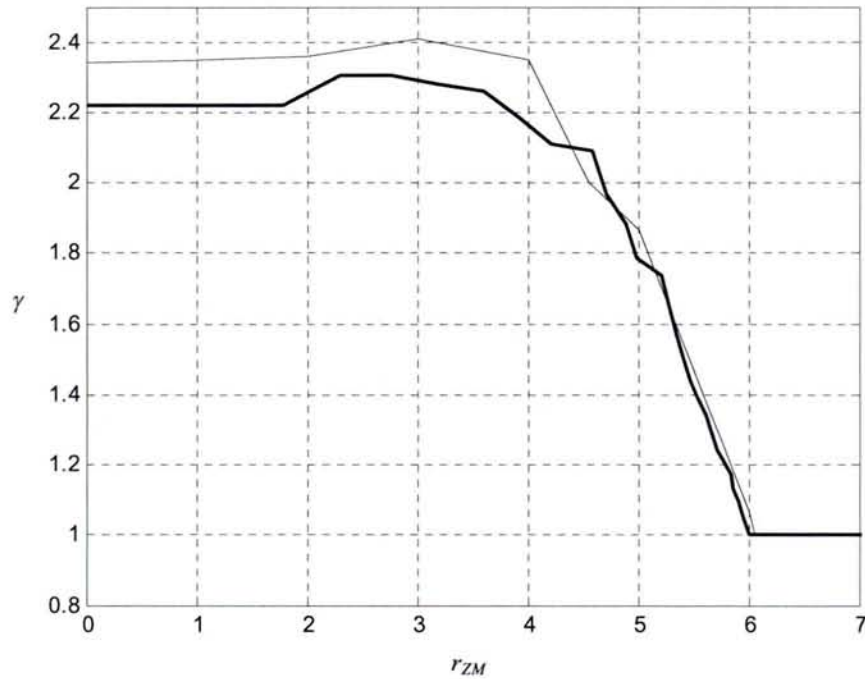


FIG. 2.16: Evolution des couples (γ, r_{ZM}) pour une source laplacienne (moyenne nulle, $\sigma = 1$) et pour un débit fixé ($R = 1,0$ bpp). Courbe théorique en continu, et courbe pratique en continu gras.

2.10 Conclusion du chapitre

Cette étude théorique de la Quantification Vectorielle Algébrique avec Zone Morte (QVAZM) et les principaux résultats qui en découlent, montrent qu'il est possible de diminuer la distorsion sur la plupart des sous-bandes. Nous montrerons à travers le chapitre 3 que cette approche nous permet d'améliorer de manière significative la qualité visuelle d'une image ainsi que le *PSNR* par rapport à *JPEG2000*.

L'intérêt principal de notre approche est de mieux quantifier les coefficients d'ondelettes significatifs. Le fait de créer une zone morte permet d'éliminer les coefficients qui n'apportent pas une information pertinente lors de la reconstruction de l'image. Nous pouvons considérer ces coefficients comme du bruit. C'est d'ailleurs dans ce cadre d'étude que nous montrerons à travers le chapitre 4 la robustesse de notre méthode pour des applications de tatouage d'images, face à différentes attaques (re-compression, filtrage et en particulier bruit additif gaussien).

D'autre part, nous avons établi un modèle du débit entropique par code préfixe pour une source laplacienne. Celui-ci pourra être utilisé pour accélérer la recherche du meilleur couple (γ^*, r_{ZM}^*) qui permet d'atteindre le débit cible de la sous-bande considérée avec une distorsion minimale.

Chapitre 3

Allocation optimale des débits associée à la QVAZM et au codage *Stack-Run*

3.1 Introduction

Ce chapitre est consacré à la présentation d'une méthode d'allocation optimale des débits associée à la Quantification Vectorielle Algébrique avec Zone Morte et au codage *Stack-Run*. Ces fonctions correspondent à une partie de notre chaîne de compression présentée au chapitre 1. Dans un premier temps, nous donnerons une description générale des différents traitements qui composent cette chaîne, à savoir : la transformée en ondelettes, l'allocation optimale des débits avec la QVAZM et le codage *Stack-Run*. Dans un second temps, nous présenterons l'algorithme que nous avons choisi pour déterminer la répartition des débits de chaque sous-bande [74] [75]. Cet algorithme détermine les débits optimaux $\hat{R}_k$ de chaque sous-bande (optimisation conjointe du débit et de la distorsion), avec pour contrainte d'atteindre le débit cible R_C , c'est-à-dire le taux de compression souhaité. Nous précisons que la transformée en ondelettes sur J niveaux de résolution engendre $K = 3J$ sous-bandes notées SB_k où $k \in \{0, 1, \dots, K - 1\}$ et une sous-bande basse fréquence notée SB_{BF} . Chaque sous-bande SB_k est caractérisée par son facteur de projection γ_k qui fixe le débit R_k de la sous-bande. L'objectif de l'allocation optimale des débits $\hat{R}_k$ avec la Quantification Vectorielle Algébrique consiste donc à trouver les facteurs de projection optimaux $\hat{\gamma}_k$ pour chaque sous-bande SB_k (un seul paramètre par sous-bande). Or, comme nous l'avons introduit au chapitre 2, la QVAZM dépend de deux paramètres : le facteur de projection γ et le rayon de la zone morte r_{ZM} . Nous présenterons donc notre algorithme d'optimisation qui permet de trouver le couple optimal (γ_k^*, r_{ZMk}^*) pour chaque sous-bande SB_k , afin d'atteindre le débit optimal $\hat{R}_k$. Enfin, comme nous l'avons montré dans le chapitre 2, la création d'une zone morte engendre des plages importantes de vecteurs nuls. Cette redondance spatiale sera prise en compte au codage en associant un codeur de type *Stack-Run* à la Quantification Vectorielle Algébrique avec Zone Morte. Les résultats expérimentaux seront présentés sur différentes images et comparés aux méthodes de références actuelles telles que *SPIHT* et *JPEG2000*.

3.2 Description générale

Notre chaîne de compression est basée sur trois principaux traitements. Tout d'abord, nous appliquons une transformée en ondelettes discrète en utilisant le filtre biorthogonal 9-7 [26]. Ce filtre a été initialement proposé dans le domaine de la compression d'images il y a une dizaine d'années [1]. De nombreux travaux montrent qu'il est actuellement le plus performant en terme de com-

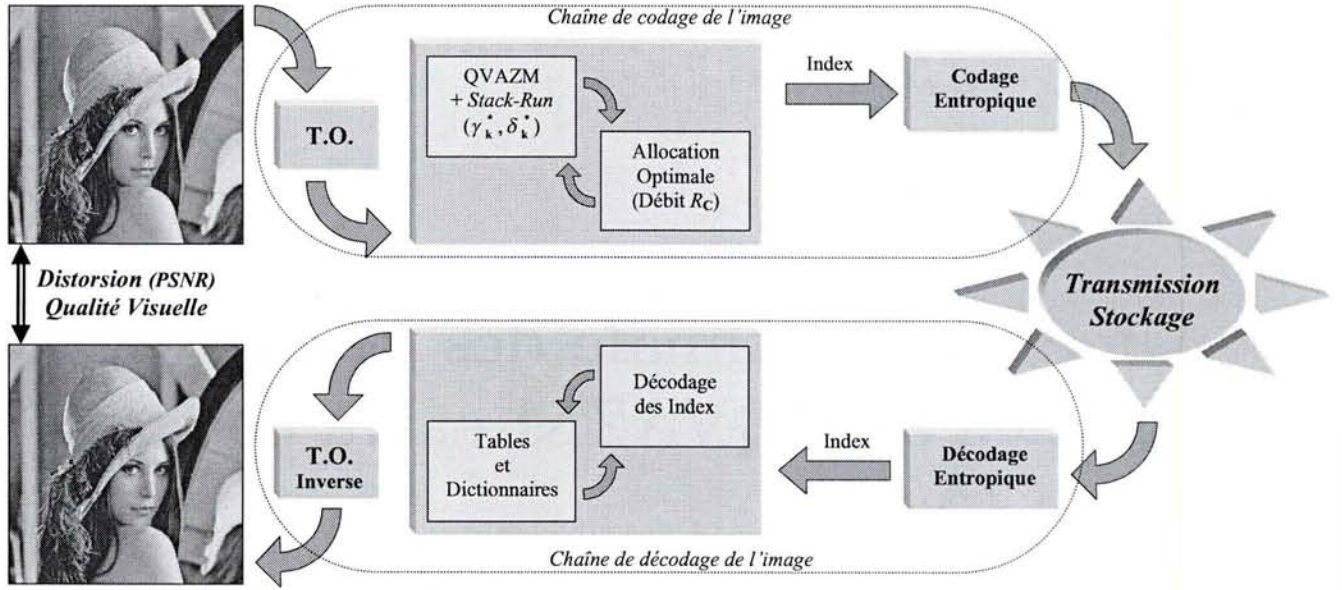


FIG. 3.1: Chaîne de compression d'une image

promis complexité/ qualité [3]. Il a d'ailleurs été choisi dans le nouveau standard de compression *JPEG2000*. Ensuite, nous effectuons notre processus de quantification basé sur la Quantification Vectorielle Algébrique avec Zone Morte associée à une allocation optimale des débits. Nous avons également mis en place un codeur de type *Stack-Run* afin d'améliorer le codage des plages d'énergie nulle. Enfin, l'utilisation d'un codage entropique (en pratique : codage de Huffman ou Arithmétique) des index pourra être envisagé lors la transmission ou le stockage des données finales. Cette chaîne de compression est représentée sur la figure 3.1. Notre travail s'est focalisé en particulier sur l'optimisation du couple (γ_k^*, r_{ZMK}^*) pour atteindre le débit optimal $\hat{R}_k$ de chaque sous-bande SB_k . L'objectif principal est d'obtenir, pour un débit cible R_C donné, la plus faible distorsion entre l'image originale et l'image reconstruite (mesure objective) et également d'obtenir une bonne qualité visuelle (mesure subjective).

3.3 Allocation optimale de débit associée à la QVA

3.3.1 Définitions

De manière générale, l'allocation optimale de débit consiste à répartir le débit cible R_C sur chaque sous-bande SB_k issue de la transformée en ondelettes discrète afin de minimiser la distorsion moyenne totale D_T . Le problème d'optimisation peut s'écrire comme un problème classique de minimisation multidimensionnel avec contrainte d'égalité :

$$\begin{cases} \min_{\{R_k \in \mathbb{R} | k=0, \dots, K-1\}} D_T(R_{BF}, R_0, \dots, R_k, \dots, R_{K-1}) \\ \text{sous la contrainte d'égalité} \\ R_T(R_{BF}, R_0, \dots, R_k, \dots, R_{K-1}) = R_C \end{cases} \quad (3.1)$$

où R_T (i.e. le débit total) correspond à la somme pondérée des débits R_k et du débit de la basse fréquence R_{BF} .

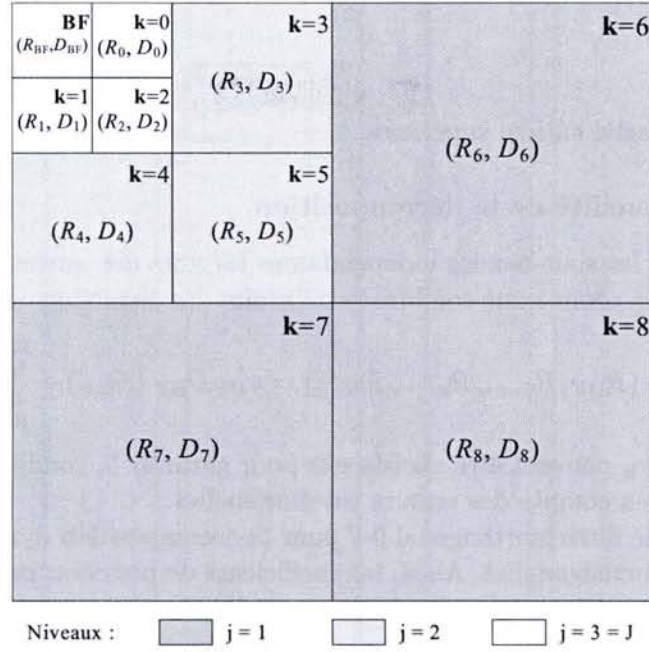


FIG. 3.2: Représentation des sous-bandes SB_k d'une transformée en ondelettes sur $J = 3$ niveaux de décomposition. Chaque sous-bande SB_k a son propre débit R_k et sa propre distorsion D_k .

3.3.2 Hypothèses

Le problème de l'allocation optimale peut être résolu en imposant certaines hypothèses liées à la décomposition de l'image en sous-bandes. En effet, chaque sous-bande SB_k possède ses propres caractéristiques statistiques et ses propres informations fréquentielles et spatiales. Certaines méthodes utilisent la redondance d'information spatiale à travers certaines sous-bandes (Algorithme *EZW*, *SPIHT*[78]). Cependant, afin de rendre plus souple l'utilisation de notre méthode (modification de certaines sous-bandes sans perturber les autres : application au tatouage d'images par exemple), et en tenant compte des contraintes de l'algorithme d'optimisation, nous considérerons les différentes sous-bandes indépendantes les unes des autres tout au long de notre étude. La figure 3.2 montre que chaque sous-bande SB_k a son propre débit R_k et sa propre distorsion D_k .

3.3.2.1 Indépendance des sous-bandes

L'approche allocation de débit par sous-bandes fait l'hypothèse d'un traitement indépendant des sous-bandes, figure 3.2, de sorte que le débit total s'exprime comme la somme pondérée des débits dans chaque sous-bande SB_k et de la sous-bande basse fréquence SB_{BF} :

$$R_T(R_{BF}, R_0, \dots, R_k, \dots, R_{K-1}) = a_{BF}R_{BF} + \sum_{k=0}^{K-1} a_k R_k \quad (3.2)$$

Les coefficients de pondération a_k pour chaque sous-bande SB_k tiennent compte de l'opération de sous-échantillonnage effectuée lors de la décomposition en Ondelettes. Dans le cas d'une décomposition dyadique sur J niveaux avec j l'indice d'un niveau, les a_j sont donnés par la formule :

$$a_j = \frac{1}{2^{2j}} \quad \text{et} \quad a_{BF} = a_J$$

or si chaque niveau j est composé de trois sous-bandes avec un nombre total de sous-bandes $K = 3J$, et avec k l'indice d'une sous-bande défini comme sur la figure 3.2, alors les a_k sont donnés par la

formule :

$$a_k = \frac{1}{2^{\lceil \frac{K-k}{3} \rceil}} \quad (3.3)$$

où $\lceil \cdot \rceil$ désigne la partie entière supérieure.

3.3.2.2 Orthogonalité de la décomposition

En considérant les sous-bandes indépendantes les unes des autres, il est possible d'exprimer la distorsion totale D_T comme une combinaison linéaire des distorsions dans chaque sous-bande SB_k :

$$D_T(R_{BF}, R_0, \dots, R_k, \dots, R_{K-1}) = \pi_{BF} D_{BF}(R_{BF}) + \sum_{k=0}^{K-1} \pi_k D_k(R_k) \quad (3.4)$$

où les coefficients π_k peuvent être choisis soit pour garantir la condition d'orthogonalité soit pour une meilleur prise en compte des erreurs psychovisuelles.

Nous utilisons le filtre biorthogonal 9-7 pour la décomposition dyadique, or ce filtre ne garantit pas la condition d'orthogonalité. Ainsi, les coefficients de pondération π_k , pour chaque sous-bande SB_k , peuvent être choisis pour rendre la décomposition orthogonale. Minimiser la distorsion totale D_T revient alors à minimiser l'erreur quadratique moyenne. Cependant, dans notre cas, il suffit d'obtenir une erreur quadratique moyenne pour chaque sous-bande et donc, afin de simplifier les traitements, les coefficients de pondération π_k ont été réglés à 1.

3.3.2.3 Convexité

Le problème d'optimisation avec contrainte 3.1 peut être résolu avec la dualité lagrangienne. Cet outil est particulièrement adapté pour l'optimisation de fonctions convexes ou localement convexes. Nous présentons au paragraphe suivant la formalisation du problème.

3.3.3 Formalisation du problème

L'allocation globale des débits tient compte des débits R_k et des distorsions D_k des sous-bandes SB_k et également du débit R_{BF} et de la distorsion D_{BF} de la sous-bande basse fréquence SB_{BF} . Cependant, aucun modèle statistique simple ne permet de représenter la distribution des coefficients d'ondelettes de cette sous-bande. Le débit R_{BF} est donc fixé à l'avance afin de coder avec une grande précision la sous-bande des basses fréquences, ce qui implique que la distorsion D_{BF} est également fixée. En effet, cette sous-bande représente les moyennes locales de l'image, une erreur importante sur ces coefficients engendre une erreur importante sur de grandes zones de l'image reconstruite. Le problème de l'allocation de débit (problème 3.1) peut être réécrit en considérant les hypothèses précédentes :

$$\left\{ \begin{array}{l} \min_{\{R_k \in \mathbb{R} | k=0, \dots, K-1\}} \left[\sum_{k=0}^{K-1} \pi_k D_k(R_k) + \pi_{BF} D_{BF}(R_{BF}) \right] \\ \text{sous la contrainte d'égalité} \\ \sum_{k=0}^{K-1} a_k R_k + a_0 R_{BF} = R_C \end{array} \right. \quad (3.5)$$

Ce problème d'optimisation peut être résolu en utilisant les opérateurs de Lagrange et en minimisant le critère J (appelé aussi fonctionnelle) :

$$J_\lambda(R_{BF}, R_0, \dots, R_{K-1}) = \left[\sum_{k=0}^{K-1} \pi_k D_k(R_k) + \pi_{BF} D_{BF}(R_{BF}) \right] + \lambda \left[\left(\sum_{k=0}^{K-1} a_k R_k + a_0 R_{BF} \right) - R_C \right] \quad (3.6)$$

Une solution optimale au problème 3.5 peut être obtenue si nous savons déterminer un point-col de la fonction de Lagrange, c'est-à-dire un couple $(\hat{R}_T, \hat{\lambda})$ vérifiant les conditions de stationnarité données par les relations suivantes :

$$\begin{cases} J_{\hat{\lambda}}(\hat{R}_T) = \min_{R_T} J_{\hat{\lambda}}(R_T) \\ \text{avec} \\ \hat{\lambda}(\hat{R}_T - R_C) = 0 \end{cases} \quad (3.7)$$

Supposons la fonction R_T continue sur l'intervalle admissible des solutions $\mathbf{S}$. Nous pouvons maintenant définir la fonction $w(\lambda)$ pour $\lambda \geq 0$ par la relation 3.8. Cette fonction permet de connaître pour différents λ les débits $\hat{R}_k$ qui minimisent le critère $J_{\lambda}(R_T)$.

$$w(\lambda) = \min_{R_T \in \mathbf{S}} J_{\lambda}(R_T) \quad (3.8)$$

La recherche d'un point-col, s'il existe, peut se faire en résolvant le problème dual du problème initial 3.5 :

$$\begin{cases} \max_{\lambda} w(\lambda) = \max_{\lambda} \left\{ \min_{R_T \in \mathbf{S}} J_{\lambda}(R_T) \right\} \\ \text{avec} \\ \lambda \in \mathbb{R}^{K+} \end{cases} \quad (3.9)$$



Une solution permet de résoudre ce problème en deux étapes [74] :

- la première étape consiste à minimiser le critère Lagrangien par rapport à R_T pour un λ fixé. Nous disposons ainsi de l'allocation optimale des débits $\{\hat{R}_k \in \mathbb{R} \mid k = 0, \dots, K-1\}$. Cette étape garantit que l'on se situe sur l'enveloppe convexe de la fonction Débit-Distorsion $D(R)$, comme le montre la figure 3.3(a) ;
- la seconde étape consiste à trouver le paramètre λ satisfaisant la contrainte d'égalité imposée sur le débit : $\hat{R}_T(\hat{R}_{BF}, \hat{R}_0, \dots, \hat{R}_{K-1}) = R_C$. Nous disposons ainsi du $\hat{\lambda}$ optimal parmi les solutions situées sur l'enveloppe convexe. Cette étape permet d'atteindre le débit cible R_C avec une allocation optimale des débits $\hat{R}_k$. La recherche du $\hat{\lambda}$ optimal est réalisée par une procédure itérative sur la fonction concave $w(\lambda)$ (voir figure 3.3(b)).

3.3.4 Algorithme d'allocation des débits associée à la QVA

L'allocation des débits est uniquement effectuée sur les sous-bandes SB_k hautes fréquences. La sous-bande basse fréquence SB_{BF} est traitée indépendamment des autres sous-bandes SB_k (le débit R_{BF} est fixé par une Quantification Scalaire Uniforme suivie d'un codeur entropique). Les sous-bandes hautes fréquences SB_k sont traitées par la Quantification Vectorielle Algébrique décrite au chapitre 2.

Dans la première étape, compte tenu du choix du quantificateur (QVA), caractérisé par son paramètre de réglage : le facteur de projection γ , l'algorithme consiste à minimiser le critère $J_{\lambda}(\gamma)$ où $\gamma = \{\gamma_0, \dots, \gamma_k, \dots, \gamma_{K-1}\}$ correspond au vecteur des facteurs de projection associés aux différentes sous-bandes SB_k . L'objectif de l'allocation revient maintenant à déterminer le facteur de projection optimal $\hat{\gamma}$, à λ fixé, minimisant la fonctionnelle $J_{\lambda}(\gamma)$ (formule 3.10). La procédure de minimisation du problème utilise une méthode itérative basée sur le gradient conjugué (avec la variante de Fletcher & Reeves [37]).

$$J_{\lambda}(\gamma_0, \dots, \gamma_{K-1}) = \left[\sum_{k=0}^{K-1} \pi_k D_k(\gamma_k) + \pi_{BF} D_{BF} \right] + \lambda \left[\left(\sum_{k=0}^{K-1} a_k R_k(\gamma_k) + a_0 R_{BF} \right) - R_C \right] \quad (3.10)$$

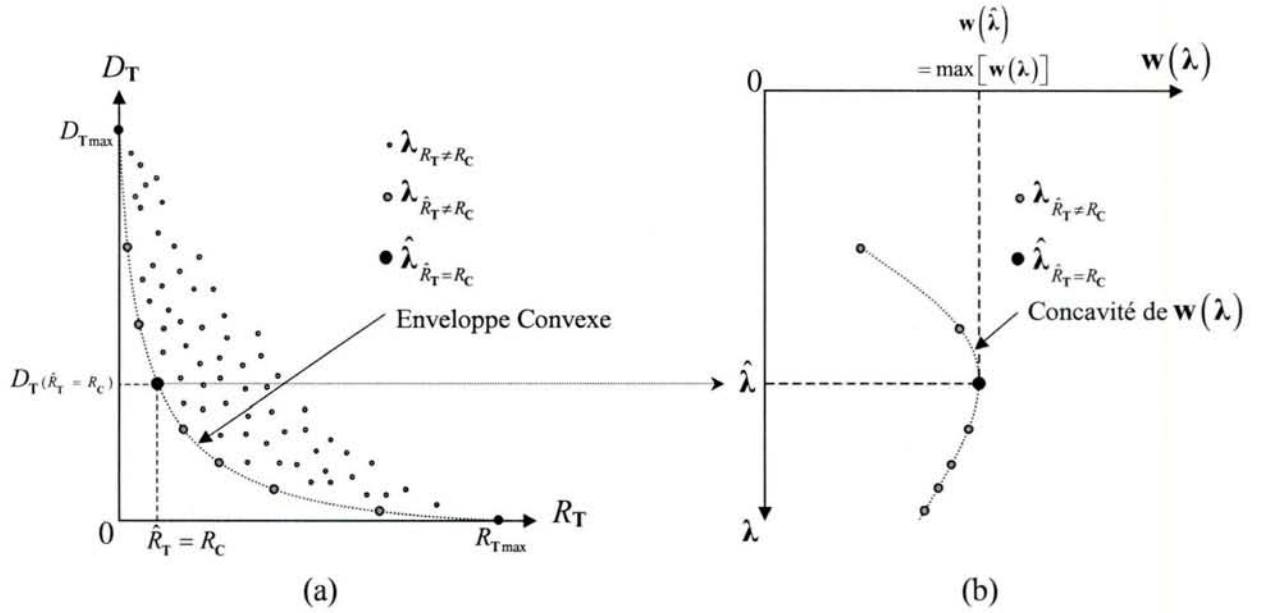


FIG. 3.3: Courbes $D_T(R_T)$ et $w(\lambda)$ pour l'allocation optimale des débits.

Dans la seconde étape, la procédure de recherche du $\hat{\lambda}$ optimal pour atteindre le débit R_C utilise une méthode itérative qui engendre une suite de $\{\lambda_j\}$ convergeant vers $\hat{\lambda}$ optimal. Cette étape ne pose pas de réelles difficultés en raison de la propriété de la concavité de $w(\lambda)$. La formule 3.11 donne la procédure de calcul des λ_{j+1} , où θ_j est la direction de déplacement souvent établie à partir du gradient de J au point λ_j et κ le pas du gradient choisi a priori. Afin de réduire le coût calculatoire, la contrainte d'égalité sur le débit ($R_T - R_C$) a été utilisée pour le calcul du gradient θ_j .

$$\lambda_{j+1} = \lambda_j + \kappa \theta_j = \lambda_j + \kappa (R_T - R_C) \quad (3.11)$$

Nous disposons à présent d'un algorithme d'allocation optimale des débits pour les sous-bandes hautes fréquences SB_k qui dépend d'un seul paramètre, le vecteur des facteurs de projection $\gamma = \{\gamma_0, \dots, \gamma_k, \dots, \gamma_{K-1}\}$. Il est donc possible de trouver le vecteur optimal $\hat{\gamma} = \{\hat{\gamma}_0, \dots, \hat{\gamma}_k, \dots, \hat{\gamma}_{K-1}\}$ pour atteindre le débit cible R_C .

3.4 Allocation des débits associée à la QVAZM

3.4.1 Objectif

L'algorithme d'allocation des débits permet d'obtenir la répartition optimale des débits à travers les sous-bandes k (c'est-à-dire les $\hat{R}_k$), afin d'atteindre le débit cible R_C . Les débits optimaux $\hat{R}_k$ ont été calculés à l'aide de la QVA en utilisant les facteurs de projections optimaux $\hat{\gamma}_k$. Notre étude s'est ensuite orientée vers une allocation des débits associée à la Quantification Vectorielle Algébrique avec Zone Morte. L'objectif de cette approche consiste à optimiser de manière conjointe les facteurs de projection γ_k et les rayons des zones mortes r_{ZMk} de façon à trouver le couple optimal (γ_k^*, r_{ZMk}^*) pour chaque sous-bande SB_k . Ce problème d'optimisation conjointe est particulièrement complexe car il dépend de deux paramètres : le facteur de projection γ_k et le rayon de la zone morte r_{ZMk} , qui font varier à la fois le débit R_k et la distorsion D_k .

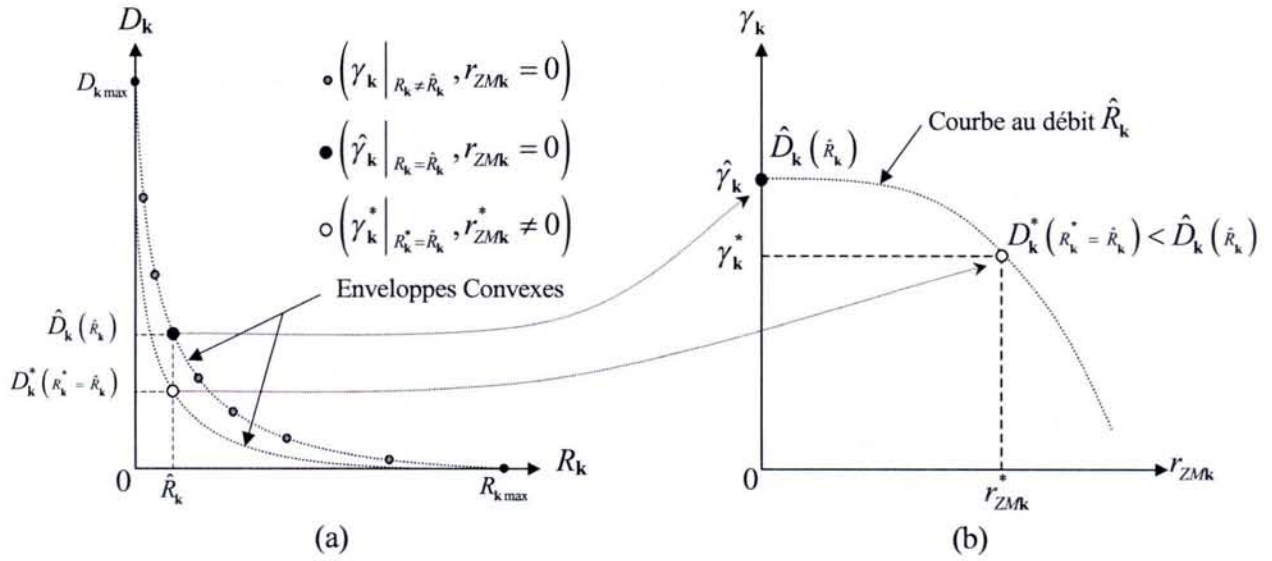


FIG. 3.4: (a) Distorsion en fonction du débit sur une sous-bande SB_k avec et sans zone morte. (b) Couple optimal (γ_k^*, r_{ZMk}^*) qui permet d'atteindre le débit optimal $R_k^* = \hat{R}_k$ avec la distorsion minimale $D_k^* < \hat{D}_k$.

3.4.2 Problème : une optimisation conjointe

Les deux paramètres γ_k et r_{ZMk} sont indépendants et permettent tous deux de modifier le débit et la distorsion. Cependant nous avons choisi de considérer le facteur de projection γ_k comme le paramètre prioritaire. En effet, celui-ci permet de fixer le débit $\hat{R}_k$ à atteindre pour la sous-bande considérée lors de la phase d'allocation des débits. Ainsi, le rayon de la zone morte r_{ZMk} sera plutôt considéré comme un paramètre d'ajustement afin de diminuer la distorsion et en préservant le débit $\hat{R}_k$ à atteindre. En tenant compte de ces hypothèses, nous avons intégré une procédure d'optimisation conjointe des paramètres γ_k^* et r_{ZMk}^* en conservant la répartition des débits optimaux $\hat{R}_k$ du programme d'allocation sans zone morte décrit au §3.3. Cette méthode permet de converger vers une solution unique optimale, c'est-à-dire une solution sur l'enveloppe convexe de la courbe Débit-Distorsion avec zone morte. La figure 3.4(a) montre l'objectif de notre approche qui consiste à trouver pour chaque sous-bande SB_k , le couple optimal (γ_k^*, r_{ZMk}^*) qui permet d'atteindre le débit optimal $R_k^* = \hat{R}_k$ avec une distorsion minimale $D_k^* < \hat{D}_k$. Comme le montre la figure 3.4(b), lorsque le débit optimal R_k^* est égal à $\hat{R}_k$, il est important de noter que $\gamma_k^* \neq \hat{\gamma}_k$ si $r_{ZMk} \neq 0$ et que $\gamma_k^* = \hat{\gamma}_k$ si $r_{ZMk} = 0$.

3.4.3 Formalisation du problème

Nous avons montré au chapitre 2 qu'il était possible d'obtenir la courbe théorique, dans une sous-bande SB_k , des différents couples (γ_k, r_{ZMk}) pour atteindre un débit R_k donné (voir figure 2.16 du chapitre 2, illustré également sur la figure 3.4(b)). De plus, nous pouvons constater en observant la figure 3.5 qu'il est possible en introduisant une zone morte, de réduire la distorsion $\hat{D}_k$, pour le débit $\hat{R}_k$ donné. Nous rappelons que le facteur de projection γ_k^* est toujours calculé pour atteindre le débit fixé $R_k^* = \hat{R}_k$ quelle que soit la zone morte $r_{ZMk} \neq 0$. Nous pouvons également remarquer que la courbe $D_k(\gamma_k^*, r_{ZMk})$ est localement convexe dans chaque intervalle $[\delta_k, \delta_k + 1]$ avec $\delta_k = \left\lceil \frac{r_{ZMk}}{\gamma_k^*} \right\rceil \in \mathbb{N}$. Il est donc possible de trouver dans chaque intervalle un seuil minimal

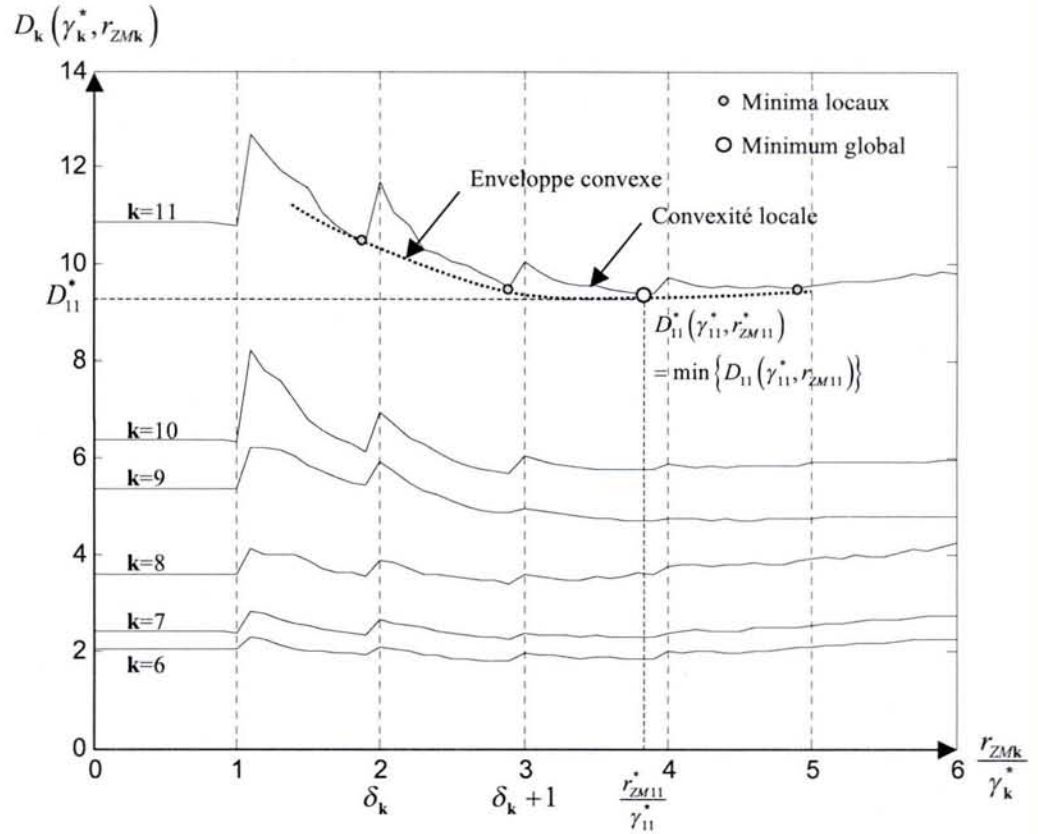


FIG. 3.5: Simulation des courbes de $D_k(\gamma_k^*, r_{ZMk})$ des sous-bandes SB_k de 6 à 11 ($J = 5$) de l'image *Lena* pour un débit cible $R_C = 0,125$ bpp avec la QVAZM.

$\frac{r_{ZMk}}{\gamma_k^*} \in \mathbb{R}$. En tenant compte uniquement des seuils minimaux de la courbe $D_k(\gamma_k^*, r_{ZMk})$, nous pouvons considérer que ces minima locaux sont sur une enveloppe convexe. Il est donc possible de trouver un unique seuil minimal $\frac{r_{ZMk}^*}{\gamma_k^*} \in \mathbb{R}$ qui nous permet d'obtenir le couple optimal (γ_k^*, r_{ZMk}^*) .

Notre problème général consiste alors à trouver parmi les couples (γ_k^*, r_{ZMk}) , le couple optimal (γ_k^*, r_{ZMk}^*) qui permet d'atteindre la distorsion minimale D_k^* .

$$\begin{cases} (\gamma_k^*, r_{ZMk}^*) = \min_{D_k(\gamma_k^*, r_{ZMk})} (\gamma_k^*, r_{ZMk}) \\ \text{avec } (\gamma_k^*, r_{ZMk}) = \left\{ (\gamma_k, r_{ZMk}) \mid R_k^*(\gamma_k, r_{ZMk}) = \hat{R}_k \quad \forall \quad r_{ZMk} \in \mathbb{R} \right\} \end{cases} \quad (3.12)$$

Notre algorithme de recherche du couple optimal (γ_k^*, r_{ZMk}^*) , pour chaque sous-bande SB_k , est réalisé en deux étapes imbriquées :

- la première étape consiste à rechercher le facteur de projection γ_k^* pour un rayon r_{ZMk} donné. Cette étape nous assure la condition sur le débit optimal à atteindre $R_k^* = \hat{R}_k$ de la sous-bande SB_k . La figure 3.6(a) montre l'allure des courbes $R_k(\gamma_k, r_{ZMk})$ pour différents rayons r_{ZMk} .
- la seconde étape consiste à trouver le rayon optimal r_{ZMk}^* pour atteindre la distorsion minimale D_k^* . Cette étape qui imbrique la première étape, permet d'obtenir le couple optimal (γ_k^*, r_{ZMk}^*) pour un débit $\hat{R}_k$ donné. La figure 3.6(b) montre l'allure de l'enveloppe convexe de la courbe $D_k(\gamma_k^*, r_{ZMk})$, représentée également sur la figure 3.5, où chaque facteur de projection γ_k^* a été calculé à la première étape afin d'assurer $R_k^* = \hat{R}_k$.

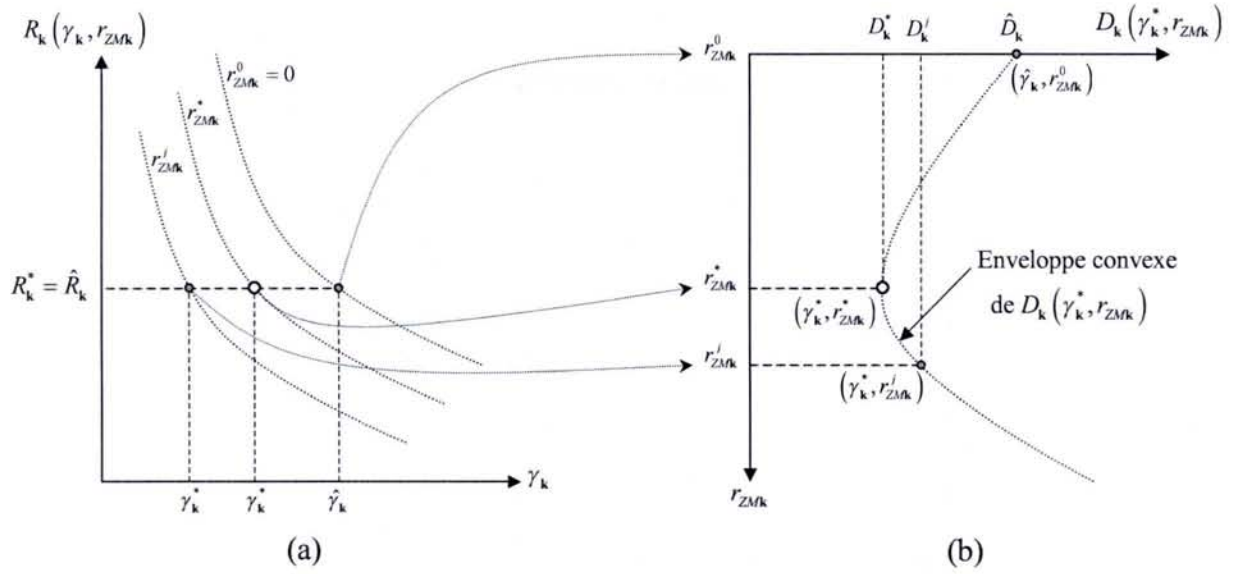


FIG. 3.6: Description des courbes $R_k(\gamma_k, r_{ZMk})$ et $D_k(\gamma_k^*, r_{ZMk})$ pour la recherche du couple optimal (γ_k^*, r_{ZMk}^*) d'une sous-bande SB_k .

3.4.4 Description de l'algorithme proposé

La recherche du couple optimal (γ_k^*, r_{ZMk}^*) est effectuée sur les sous-bandes hautes fréquences SB_k . La sous-bande basse fréquence SB_{BF} est traitée de manière indépendante par une Quantification Scalaire Uniforme. Les sous-bandes SB_k sont traitées par la Quantification Vectorielle Algébrique avec Zone Morte décrite au chapitre 2. La figure 3.7 présente l'algorithme général de l'optimisation du couple (γ_k^*, r_{ZMk}^*) pour chaque sous-bande SB_k . Nous présentons ensuite la description détaillée des algorithmes utilisés.

Dans la première étape, la procédure de recherche du facteur de projection γ_k^* pour un rayon r_{ZMk} donné est effectuée par un algorithme de dichotomie. En effet, la fonction $R_k(\gamma_k, r_{ZMk})$ est décroissante, il est donc possible de converger vers le débit optimal $\hat{R}_k$ pour obtenir le facteur de projection γ_k^* . Les bornes pour l'initialisation de l'algorithme de dichotomie ont été choisies en

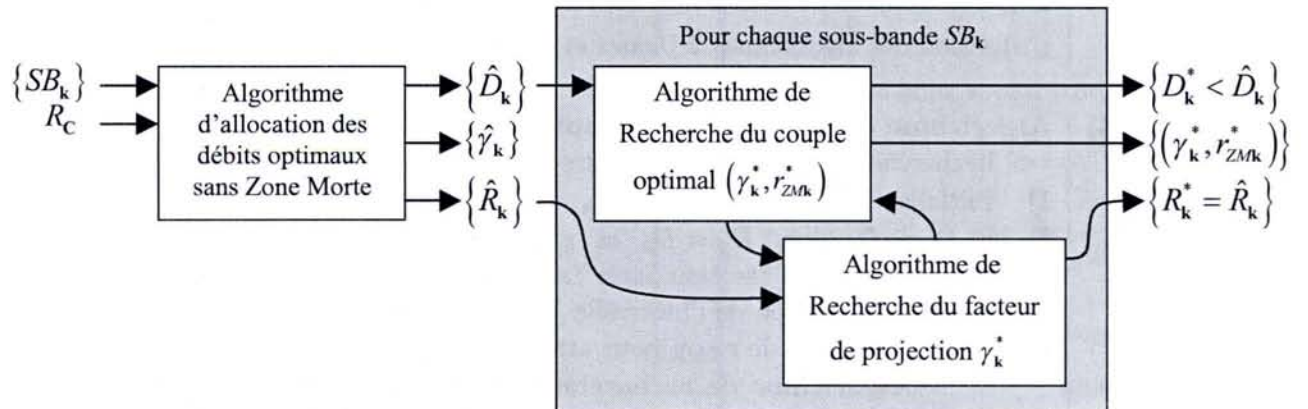


FIG. 3.7: Description de l'algorithme général de l'optimisation des couples (γ_k^*, r_{ZMk}^*) pour chaque sous-bande SB_k .

fonction de la sous-bande SB_k traitée. Le facteur de projection minimal $\gamma_k^{\min}$ (borne minimale) est initialisé à 0, 1 pour toutes les sous-bandes SB_k . Par contre, le facteur de projection maximal $\gamma_k^{\max}$ (borne maximale) dépend de la sous-bande SB_k traitée, il est défini par la formule 3.13 où κ_γ est un coefficient de pondération fixé a priori ($\kappa_\gamma \simeq 10$).

$$\begin{cases} \gamma_k^{\min} = 0, 1 \\ \gamma_k^{\max} = \kappa_\gamma \cdot k \end{cases} \quad (3.13)$$

A l'aide de l'étude théorique du modèle du débit présentée au chapitre 2, il est possible de réduire cet intervalle de recherche. En effet, en connaissant les caractéristiques de la sous-bande (écart-type, moyenne, dimension des vecteurs), nous pouvons déterminer la plage de recherche de γ_k pour une zone morte r_{ZMk} donnée. Cette approche permettra simplement d'accélérer la procédure de recherche du couple optimal (γ_k^*, r_{ZMk}^*) et donc d'accélérer le processus d'optimisation.

Algorithme de recherche du facteur de projection γ_k^*

Recherche de γ_k^* pour r_{ZMk} donné, initialisation $[\gamma_k^{\min}, \gamma_k^{\max}]$

- 1) Initialisation de $\gamma_k = \frac{\gamma_k^{\min} + \gamma_k^{\max}}{2}$
- 2) QVAZM avec $\gamma_k \Rightarrow \{R_k, D_k\}$
- 3) Tant que $\left| \frac{R_k - \hat{R}_k}{\hat{R}_k} \right| < \epsilon_R$
 Algorithme de Dichotomie
 - 3.1) Si $R_k > \hat{R}_k$, alors $\gamma_k^{\min} = \gamma_k$, sinon $\gamma_k^{\max} = \gamma_k$
 - 3.2) Calculer le nouveau facteur de projection $\gamma_k = \frac{\gamma_k^{\max} + \gamma_k^{\min}}{2}$, Retour en 2)
- 4) Sinon γ_k^* trouvé $\Rightarrow \{R_k^*, D_k^*\}$

Dans la seconde étape, compte tenu de la convexité globale de $D_k(\gamma_k^*, r_{ZMk})$, nous avons mis en place une procédure de recherche pour trouver la distorsion minimum D_k^* parmi les quelques minima locaux dans chaque intervalle $[\delta_k, \delta_k + 1]$ avec $\delta_k = \left\lceil \frac{r_{ZMk}}{\gamma_k^*} \right\rceil$ (recherche de type exhaustive, mais sur moins d'une dizaine de valeurs). Pour rechercher ces minima locaux, compte tenu de la convexité locale de $D_k(\gamma_k^*, r_{ZMk})$, nous avons adapté l'algorithme de la "section dorée" [72]. Cet algorithme permet de trouver le minimum local de $D_k(\gamma_k^*, r_{ZMk})$ dans chaque intervalle $[\delta_k, \delta_k + 1]$ et donc le rayon de la zone morte r_{ZMk} optimal associé. Nous décrivons ci-dessous le détail de l'algorithme général issu de la figure 3.7.

Algorithme général de recherche du couple optimal (γ_k^*, r_{ZMk}^*)

- A) Algorithme d'allocation optimale des débits $\hat{R}_k$ pour atteindre R_C
 Utilisation des algorithmes d'Uzawa et de Fletcher & Reeves $\Rightarrow \{\hat{\gamma}_k, \hat{R}_k, \hat{D}_k\}$
 Pour chaque sous-bande SB_k :
 - B) Algorithme de recherche du couple optimal (γ_k^*, r_{ZMk}^*)
 $\Rightarrow$ Recherche de r_{ZMk}^* pour atteindre le minimum global de $D_k(\gamma_k^*, r_{ZMk})$:
 - 1) Initialisation $\delta_k = 0$, $D_k = \hat{D}_k$ et $r_{ZMk} = 0$
 - 2) Si $D_k \leq \hat{D}_k$ alors $D_k^* = D_k$ et $r_{ZMk}^* = r_{ZMk}$
 - 3) Algorithme de la "section dorée" sur chaque intervalle
 - 3.1) Initialisation de l'intervalle $[\delta_k, \delta_k + 1]$
 - 3.2) Recherche de r_{ZMk} pour atteindre le minimum local de $D_k(\gamma_k^*, r_{ZMk})$
 Algorithme de recherche du facteur de projection γ_k^* pour $R_k^* = \hat{R}_k$
 - 3.3) r_{ZMk} trouvé pour une distorsion D_k minimum dans l'intervalle $[\delta_k, \delta_k + 1]$
 - 4) Si $\delta_k < \delta_k^{\max}$ alors $\delta_k = \delta_k + 1$ Retour en 2)
 Sinon couple optimal trouvé $\Rightarrow (\gamma_k^*, r_{ZMk}^*)$

3.4.5 Conclusion

L'algorithme que nous avons proposé et mis en oeuvre nous permet d'obtenir pour chaque sous-bande SB_k , une distorsion plus faible $D_k^* < \hat{D}_k$ pour un même débit $R_k^* = \hat{R}_k$. Nous disposons donc d'un algorithme qui détermine le couple optimal (γ_k^*, r_{ZMk}^*) pour atteindre le débit optimal $\hat{R}_k$. L'étude théorique du modèle du débit développée au chapitre 2 permettra ultérieurement de diminuer la complexité de la recherche en conservant les performances en termes de qualité. Dans le cas idéal, il suffira de balayer les couples théoriques (γ_k, r_{ZMk}) qui permettent d'atteindre le débit $\hat{R}_k$ et de trouver celui qui obtient la distorsion minimale. D'autre part, l'étude d'un modèle de distorsion de la QVAZM calculerait directement le couple optimal (γ_k^*, r_{ZMk}^*) . Ces travaux sont en cours de développement et nous permettent d'envisager une diminution importante de la complexité de l'algorithme.

3.5 Codage *Stack-Run* adapté aux plages d'énergies nulles

Comme nous l'avons expliqué au chapitre 2, la création d'une zone morte engendre une augmentation du nombre de vecteurs nuls. Nous proposons d'intégrer un codage bien adapté à ce type de redondances spatiales. En effet, pour l'instant, nous utilisons lors du calcul du débit des énergies un codage entropique. Nous rappelons que le codage des index des énergies doit être sans perte. En pratique, il est nécessaire d'utiliser des codeurs de type Huffman [42] ou arithmétique [41]. Le codeur *Runlength* [43] est bien adapté au codage de plages de valeurs identiques. Mais par rapport à notre application, la complexité de ce codeur est trop importante. Nous avons donc utilisé le principe du codage *Stack-Run*, largement détaillé dans différents travaux [88][74], qui est tout aussi efficace avec une complexité réduite. En effet, ce codage présente quatre avantages décisifs décrits ci-dessous :

- un codage adaptatif : le codage s'adapte directement aux valeurs à coder quel que soit le nombre de bits utilisés par les valeurs ;
- un codage instantané : le codage est calculé directement au cours de la lecture des valeurs ;
- un codage performant : les performances du codage *Stack-Run* sont comparables à celles du codage *Runlength* ;
- un codage automatique des plages de zéros : le codage détermine automatiquement la longueur des plages de zéros.

Compte tenu du nombre important de vecteurs nuls suite à la Quantification Vectorielle Algébrique avec Zone Morte, nous avons donc intégré et adapté le codage *Stack-Run* aux plages d'énergies nulles afin d'optimiser encore les performances de notre chaîne de compression dans le cadre d'une application à bas débits.

3.5.1 Adaptation aux plages d'énergies nulles

A l'origine le codage *Stack-Run* permet de coder efficacement des plages de zéros et des entiers signés [74]. Pour nos besoins, nous avons adapté ce codage afin de coder efficacement les plages d'énergies qui correspondent uniquement à des valeurs positives ou nulles, c'est-à-dire à des entiers non signés. Nous décrivons au paragraphe suivant notre technique de codage à travers un exemple.

3.5.2 Description du codage *Stack-Run* adapté

De manière générale, nous définissons deux classes d'énergies :

- la classe d'énergies nulles : celles-ci sont souvent regroupées et forment des plages de zéros (*run*) ;
- la classe d'énergies non nulles : celles-ci sont plutôt ponctuelles et correspondent à des entiers positifs (*level*).

Les énergies peuvent ainsi être traduites sous la forme de deux trames : la trame des *runs* et la trame des *levels*.

Nous rappelons que la création d'une zone morte ne tient pas compte des premières énergies du réseau n'appartenant pas au dictionnaire, il ne reste donc que l'ensemble des énergies suivant :

$$\{\|Y'\|_1\} = \{0\} \cup \{\delta_k^*, \delta_k^* + 1, \dots, r_T\}$$

Or, connaissant la valeur de la zone morte $\delta_k^* = \left\lceil \frac{r_{ZMk}}{\gamma_k^*} \right\rceil$ sur le réseau $\mathbb{Z}^n$, il est possible de soustraire la valeur $(\delta_k^* - 1)$ pour chaque énergie $\|Y'\|_1$. Afin d'optimiser le codage, nous avons donc stocké les valeurs des énergies telles que :

$$(\|Y'\|_1)_{\text{Stockée}} = \|Y'\|_1 - (\delta_k^* - 1)$$

Ce recalage nous permet de coder plus efficacement les énergies et donc de diminuer le débit. Nous disposons ainsi des énergies stockées suivantes :

$$\{(\|Y'\|_1)_{\text{Stockée}}\} = \{0, 1, 2, \dots, r_T - (\delta_k^* + 1)\}$$

Nous allons à présent donner notre procédure de codage en prenant l'exemple d'une trame d'énergies, et en donnant la description des lignes du tableau 3.1.

Soit la trame d'énergies de la ligne 1. Il est possible de calculer la valeur décimale des *runs* et des *levels* (ligne 2). Ensuite, nous les écrivons en binaire du bit de poids faible au bit de poids fort (ligne 3). Nous introduisons également entre deux *levels* successifs un *run* nul pour éviter des erreurs de codage.

Afin de distinguer les deux types de trame, nous attribuons des symboles différents pour coder les différents bits des *runs* des *levels*. Nous définissons un alphabet de quatre symboles $\{0, 1, -, +\}$ où les symboles "0" et "1" sont affectés aux bits des *levels* et représentent les chiffres binaires 0 et 1. Quant aux symboles "-" et "+", ils sont affectés aux bits des *runs* et représentent les chiffres binaires 0 et 1 (ligne 4).

Nous constatons que le bit de poids fort (pour un *run* ou un *level*) est toujours égal à 1. Il est donc possible d'améliorer le code en effectuant une limitation du nombre de symboles c'est-à-dire en supprimant le bit de poids fort. Mais afin de préserver les *runs* d'un seul zéro ou des *levels* de valeur 1, nous conserverons le bit de poids fort lorsque les *runs* ou les *levels* ont une longueur en $2^k - 1 \quad \forall k \in \mathbb{N}$ (ligne 5).

Il est maintenant nécessaire de mettre en place un moyen pour effectuer le passage d'un *run* à un *level* et vice versa. Pour cela, nous avons remplacé le dernier symbole d'un *run* ou d'un *level* par les symboles respectifs d'un *level* ou d'un *run* (ligne 6). Nous pouvons enfin écrire de manière distincte la trame des *runs* t_{runs} (ligne 7 avec son codage binaire naturel en dessous), et la trame des *levels* t_{levels} (ligne 8 avec son codage binaire naturel en dessous).

Compte tenu de la probabilité d'apparition des quatre symboles dans chacune des trames, ces dernières pourraient être codées encore plus efficacement avec un codage arithmétique (adaptatif). Sinon, dans le cas d'un codage binaire des trames, la procédure présente l'intérêt d'être d'une très faible complexité et sans stockage de table de décodage.

3.5.3 Amélioration de la forme des vecteurs

Le codage *Stack-Run* est d'autant plus efficace que les plages d'énergies nulles sont importantes. Pour augmenter la taille des plages de vecteurs nuls, nous avons modifié la forme de ceux-ci afin de les rendre "localement compacts" (i.e carré ou rectangulaire). Nous présentons dans le tableau 3.2 (lignes 5 à 8) l'amélioration apportée par les vecteurs de formes rectangulaires en terme de compromis qualité/ débit. D'autre part, le sens de lecture des vecteurs à travers les sous-bandes pourrait encore améliorer le codage en augmentant la longueur de certaines plages de zéros.

	Trames	run				level		run			level		level		run		
1)	Energies	0000000000000000				4		000000000000000			2		4		0000000000		
2)	Entiers	15				4		13			2		4		9		
3)	Binaires	1111				001		1011			01 0		001		1001		
4)	Symboles	+	+	+	+	0	01	+	-	++	01	-	0	01	+	-	-+
5)	Limitation	+	+	+	+	0	0	+	-	+	0	-	0	0	+	-	-
6)	Passage	+	+	+	1	0	-	+	-	1	-	0	0	-	+	-	0
7)	t_{runs}	+	+	+	1			+	-	1		0			+	-	0
	en binaire	11	11	11	01			11	10	01		00			11	10	00
8)	t_{levels}					0	-				-		0	-			
	en binaire					00	10				10		00	10			

TAB. 3.1: Exemple de codage *Stack-Run* adapté

3.5.4 Performances et conclusions

Les performances obtenues avec ce codage *Stack-Run* (binaire) de complexité très faible, montrent qu'il est possible de réduire de presque 3,5% le débit global tout en préservant le même *PSNR* obtenu avec un codage arithmétique (lignes 12 et 13 du tableau 3.2). De plus, si nous comparons les lignes 12 et 6 du tableau 3.2, nous pouvons constater qu'il est possible avec notre codage *Stack-Run* (binaire), d'atteindre les mêmes performances qu'avec un codage entropique. Le tableau 3.2 présente les résultats expérimentaux du codage *Stack-Run* par rapport aux autres méthodes.

3.6 Résultats : reconstruction de l'image

3.6.1 Rappels

Dans un premier temps, nous effectuons sur l'image originale une transformée en ondelettes discrète avec $\mathbf{J} = 5$ niveaux de décomposition, en utilisant le filtre biorthogonal 9-7. Ensuite, nous utilisons l'algorithme d'allocation des débits suivi de notre algorithme de recherche du couple optimal (γ_k^*, r_{ZMk}^*) . La forme des vecteurs est choisie en fonction du type de codage (arithmétique ou *Stack-Run*). Le codeur de type Huffman n'a pas été mis en oeuvre (légèrement moins performant que le codeur arithmétique). Le type de débit sera précisé en fonction des algorithmes utilisés (entropique, arithmétique, *Stack-Run* entropique, *Stack-Run* binaire).

3.6.2 Images Naturelles

Nous effectuons dans un premier temps la comparaison des différents types de quantification utilisés : *SPIHT* (<http://www.cipr.rpi.edu/research/SPIHT>), *JPEG2000* (<http://jj2000.epfl.ch>), QVA, QVAZM, QVAZM *Stack-Run*. Le tableau 3.2 présente les résultats avec l'image *Lena* pour un débit entropique de 0,125 bpp.

Nous constatons que pour un même débit, la QVAZM (vecteur rectangulaire), avec un codage de type : *Stack-Run* (binaire), permet d'atteindre un meilleur *PSNR* qu'avec le standard de compression *JPEG2000* (lignes 2 et 12 du tableau 3.2). De plus, on constate une amélioration de la qualité visuelle en particulier sur les zones texturées (par exemple sur le chapeau de l'image *Lena*). Nous pouvons également remarquer que pour un même *PSNR* = 30,90 dB (Image 3.7) le débit avec notre codage de type *Stack-Run* (binaire), est plus faible qu'avec un codage classique de type arithmétique (lignes 12 et 13 du tableau 3.2).

Nous avons effectué également une comparaison de la QVA avec un codage de type arithmétique et de la QVAZM avec un codage de type *Stack-Run* (binaire), avec les méthodes *SPIHT* et

	Méthodes	Types de codage	Débit exact (bpp)	PSNR (dB)
1)	<i>SPIHT</i>		0,125	31,08
2)	<i>JPEG2000</i>		0,125	30,82
3)	QS	entropique	0,1255	30,32
4)	QSZM	entropique	0,1256	30,58
5)	QVA	entropique	0,1252	30,74
6)	QVAZM	entropique	0,1253	30,93
7)	QVA (vec. rec.)	entropique	0,1251	30,79
8)	QVAZM (vec. rec.)	entropique	0,1258	31,06
9)	QVA (vec. rec.)	<i>Stack-Run</i> (entropique)	0,1224	30,78
10)	QVAZM (vec. rec.)	<i>Stack-Run</i> (entropique)	0,1228	30,83
11)	QVA (vec. rec.)	arithmétique	0,1250	30,50
12)	QVAZM (vec. rec.)	<i>Stack-Run</i> (binaire)	0,1252	30,90
13)	QVAZM (vec. rec.)	arithmétique	0,1295	30,90

TAB. 3.2: Comparaison des méthodes pour l'image *Lena* à 0,125 bpp

	IMAGES :	<i>Lena</i>			<i>GoldHill</i>			<i>Peppers</i>		
		N°	Débit (bpp)	PSNR (dB)	N°	Débit (bpp)	PSNR (dB)	N°	Débit (bpp)	PSNR (dB)
1)	<i>SPIHT</i>	3.2	0,0777	29,23	3.9	0,08	27,38	3.16	0,08	28,80
2)	<i>JPEG2000</i>	3.3	0,0777	28,32	3.10	0,08	26,96	3.17	0,08	27,87
3)	QVAZM	3.4	0,0777	28,91	3.11	0,08	27,16	3.18	0,079	28,45
4)	<i>SPIHT</i>	3.5	0,125	31,08	3.12	0,125	28,48	3.19	0,125	30,65
5)	<i>JPEG2000</i>	3.6	0,125	30,82	3.13	0,125	28,22	3.20	0,125	30,19
6)	QVAZM	3.7	0,1252	30,90	3.14	0,125	28,33	3.21	0,122	30,17

TAB. 3.3: Comparaison des résultats sur différentes images

JPEG2000, pour différents débits. La figure 3.8 présente les différentes courbes pour chacune des méthodes.

Nous pouvons remarquer les bonnes performances de notre approche pour les faibles débits. Ces bonnes performances s'expliquent par l'intégration de la zone morte vectorielle sur différentes sous-bandes SB_k . Cela engendre des plages importantes d'énergie nulle, ce qui permet d'effectuer un codage plus efficace. Nous pouvons remarquer, par exemple, un gain de presque 0,6 dB pour un débit de 0,077 bpp (TC = 103 : 1) entre la QVAZM + *Stack-Run* (binaire) et *JPEG2000*. Nous rappelons que *JPEG2000* est basé sur une Quantification Scalaire (plus simple à mettre en oeuvre mais moins performante). Nous rappelons également que notre approche traite chaque sous-bande de manière indépendante, alors que la méthode *SPIHT* utilise une technique inter-bande, ce qui explique que nous sommes légèrement inférieurs à *SPIHT* en terme de PSNR (entre 0,2 dB et 0,3 dB de moins), cependant, notre approche nous permet de mieux préserver les zones texturées et certains détails de l'image. Nous constatons finalement que la QVA (+codage arithmétique) sans zone morte est moins performante (−0,4 dB) que la QVAZM. Nous présentons également les résultats de notre méthode dans les tableaux 3.3 et 3.4, pour différents débits, et différentes images.

3.6.3 Qualité visuelle

D'un point de vue subjectif, nous pouvons constater une amélioration de la qualité visuelle de notre approche, en particulier sur les zones texturées. Par exemple : si nous observons le chapeau de l'image *Lena* pour un TC = 64 : 1 (figure 3.9 : zoom de l'IMAGE 3.5, 3.6 et 3.7) ou pour un TC = 103 : 1 (figure 3.10 : zoom de l'IMAGE 3.2, 3.3 et 3.4), nous pouvons constater que certains détails

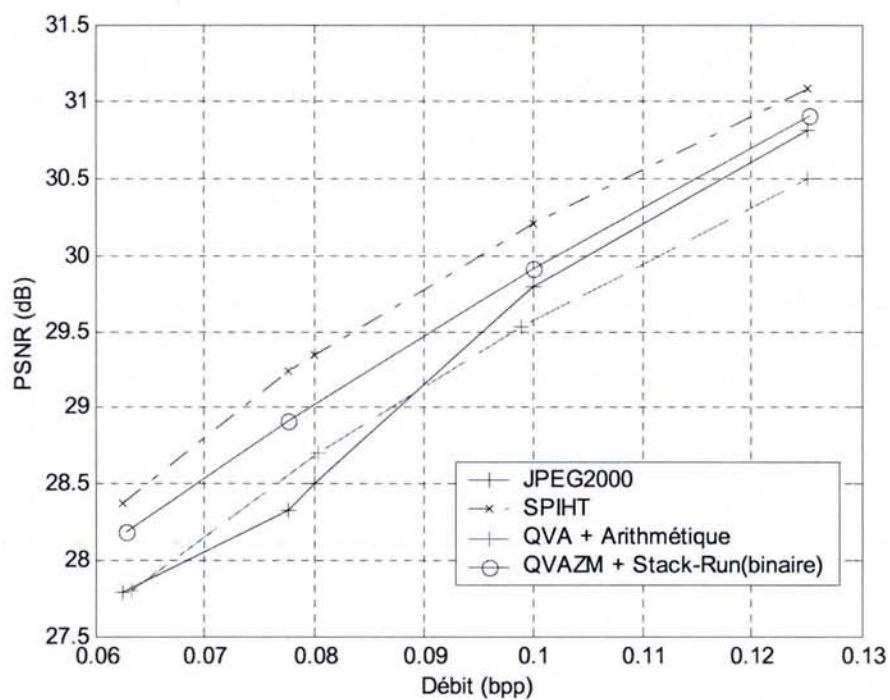


FIG. 3.8: Comparaison des performances des différentes méthodes ($PSNR$ en fonction du débit) : $JPEG2000$, $SPIHT$, QVA + arithmétique, $QVAZM$ + $Stack-Run$ (binaire) appliquées à l'image *Lena*.

	IMAGES :	<i>Boat</i>			<i>Mandrill</i>		
		N°	Débit (bpp)	$PSNR$ (dB)	N°	Débit (bpp)	$PSNR$ (dB)
7)	<i>SPIHT</i>	3.23	0,125	28,16	3.27	0,122	21,69
8)	<i>JPEG2000</i>	3.24	0,125	27,97	3.28	0,122	21,54
9)	<i>QVAZM</i>	3.25	0,125	27,96	3.29	0,122	21,68

TAB. 3.4: Comparaison des résultats sur d'autres images

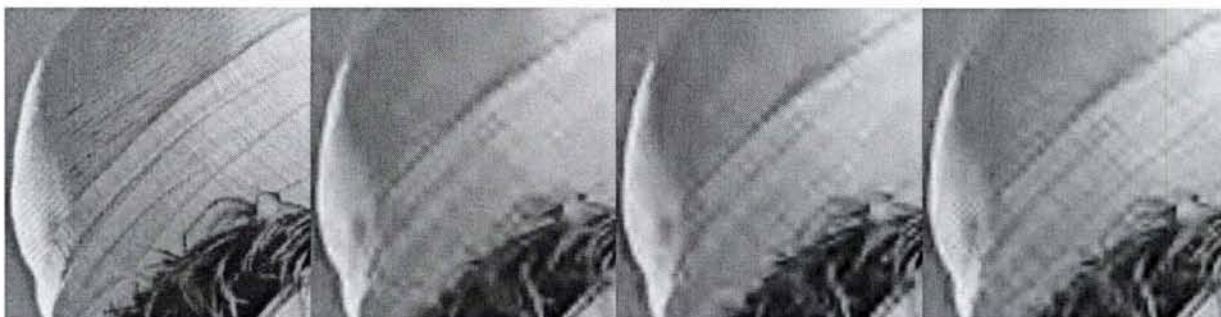


FIG. 3.9: Zoom sur le chapeau de *Lena* pour un $TC = 64 : 1$ (0,125 bpp), images de gauche à droite : originale, *SPIHT*, *JPEG2000*, *QVAZM*.



FIG. 3.10: Zoom sur le chapeau de *Lena* pour un $TC = 103 : 1$ (0,0777 bpp), images de gauche à droite : originale, *SPIHT*, *JPEG2000*, *QVAZM*.

sont mieux préservés avec notre méthode que celle de *SPIHT* ou de *JPEG2000*. Cette constatation est également valable pour des zones texturées (figure 3.12 : zoom de l'IMAGE 3.27, 3.28 et 3.29). Nous pouvons également observer l'amélioration de la qualité visuelle sur le zoom effectué sur les yeux de *Lena* pour un $TC = 64 : 1$ (zoom de l'IMAGE 3.5, 3.6 et 3.7), en particulier sur l'oeil de gauche. Les différentes images que nous présentons à la fin du chapitre permettent d'apprécier cette qualité visuelle.

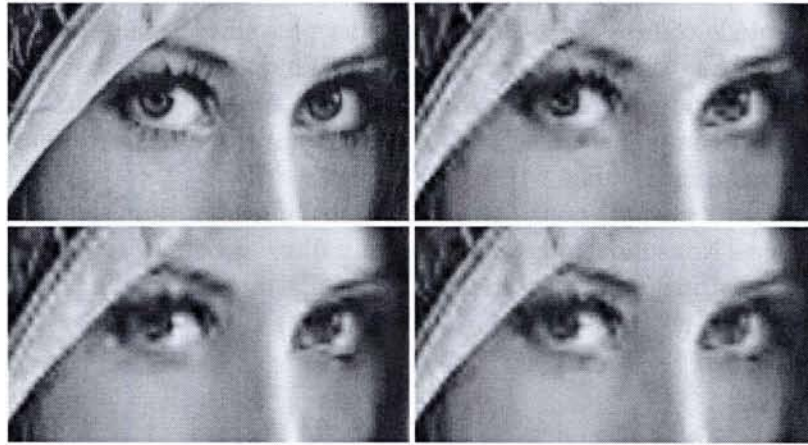


FIG. 3.11: Zoom sur les yeux de *Lena* pour un $TC = 64 : 1$ (0,125 bpp), images du haut : originale, QVAZM, images du bas : *JPEG2000*, *SPIHT*.

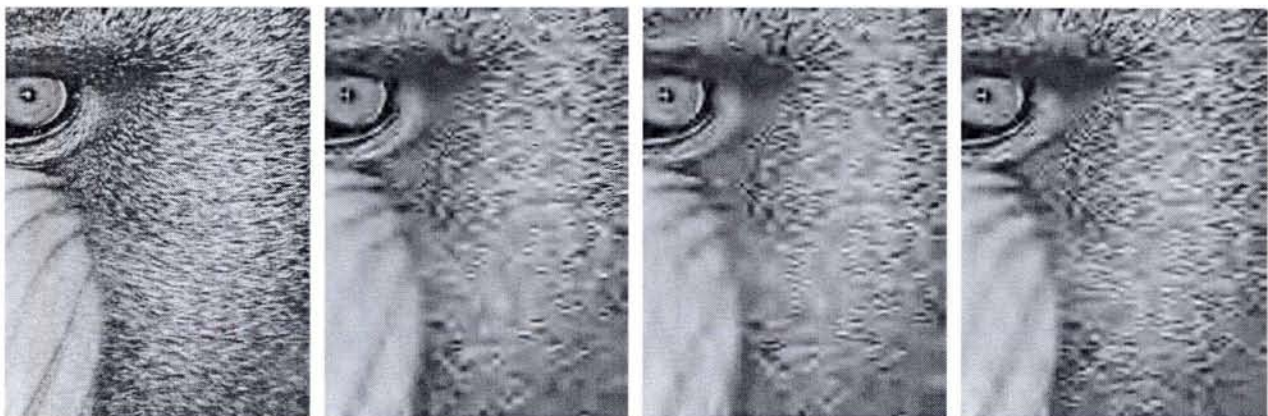


FIG. 3.12: Zoom sur une zone texturée de l'image *Mandrill*, $TC \simeq 64 : 1$ (0,122 bpp), images de gauche à droite : originale, *SPIHT*, *JPEG2000*, QVAZM.

3.7 Conclusion du chapitre

Nous avons présenté à travers ce chapitre l'étude et les performances de notre chaîne de compression. Nous avons montré que notre algorithme de recherche du couple optimal $(\gamma_{\mathbf{k}}^*, r_{ZM\mathbf{k}}^*)$ permettait de réduire la distorsion $D_{\mathbf{k}}$ de chaque sous-bande $SB_{\mathbf{k}}$ pour un débit optimal $\hat{R}_{\mathbf{k}}$ donné. De plus, nous avons adapté le codage *Stack-Run* pour coder plus efficacement les plages d'énergie nulle. Les résultats montrent que les performances, en terme de qualité objective et subjective (visuelle) de la QVAZM associée au codage *Stack-Run*, sont meilleures à bas débits. Nous avons constaté en particulier un gain compris entre 0,3 dB et 0,6 dB par rapport au standard *JPEG2000* pour des taux de compression inférieurs à 64 : 1 (0,125 bpp).

Dans le chapitre suivant, nous présentons la compression d'images avec QVAZM associée au tatouage d'images. Nous montrons que la création d'une zone morte vectorielle permet d'améliorer la robustesse du tatouage face à différents types d'attaques (en particulier, celles de type bruit additif gaussien).



IMAGE 3.1 : Image *Lena* originale.



IMAGE 3.2 : Image *Lena* SPIHT, TC=103 :1, PSNR=29,23dB.



IMAGE 3.3 : Image *Lena* JPEG2000, TC=103 :1, PSNR=28,32dB.



IMAGE 3.4 : Image *Lena* QVAZM + *Stack-Run* (binaire), TC=103 :1, PSNR=28,91dB.



IMAGE 3.5 : Image *Lena* SPIHT, TC=64 :1, PSNR=31,08dB.



IMAGE 3.6 : Image *Lena* JPEG2000, TC=64 :1, PSNR=30,82dB.



IMAGE 3.7 : Image *Lena* QVAZM + *Stack-Run* (binaire), TC=64 :1, PSNR=30,91dB.



IMAGE 3.8 : Image *Goldhill* originale.



IMAGE 3.9 : Image *Goldhill* *SPIHT*, TC=100 :1, PSNR=27,38dB.



IMAGE 3.10 : Image *Goldhill* *JPEG2000*, TC=100 :1, PSNR=26,96dB.



IMAGE 3.11 : Image *Goldhill* QVAZM + *Stack-Run* (binaire), TC=100 :1, PSNR=27,16dB.



IMAGE 3.12 : Image *Goldhill* SPIHT, TC=64 :1, PSNR=28,48dB.



IMAGE 3.13 : Image *Goldhill JPEG2000*, TC=64 :1, PSNR=28,22dB.



IMAGE 3.14 : Image *Goldhill QVAZM + Stack-Run* (binaire), TC=64 :1, PSNR=28,33dB

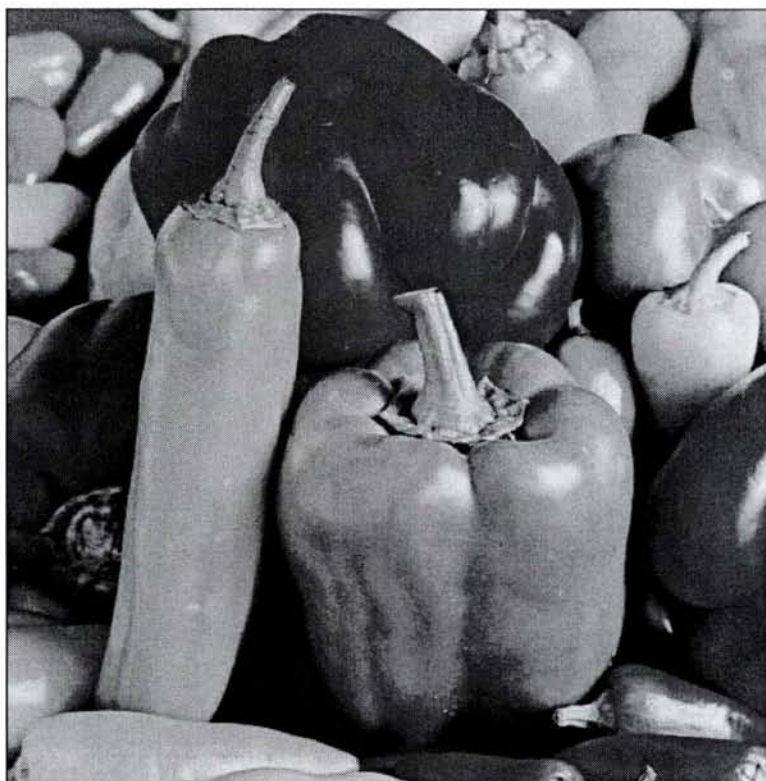


IMAGE 3.15 : Image *Peppers* originale.

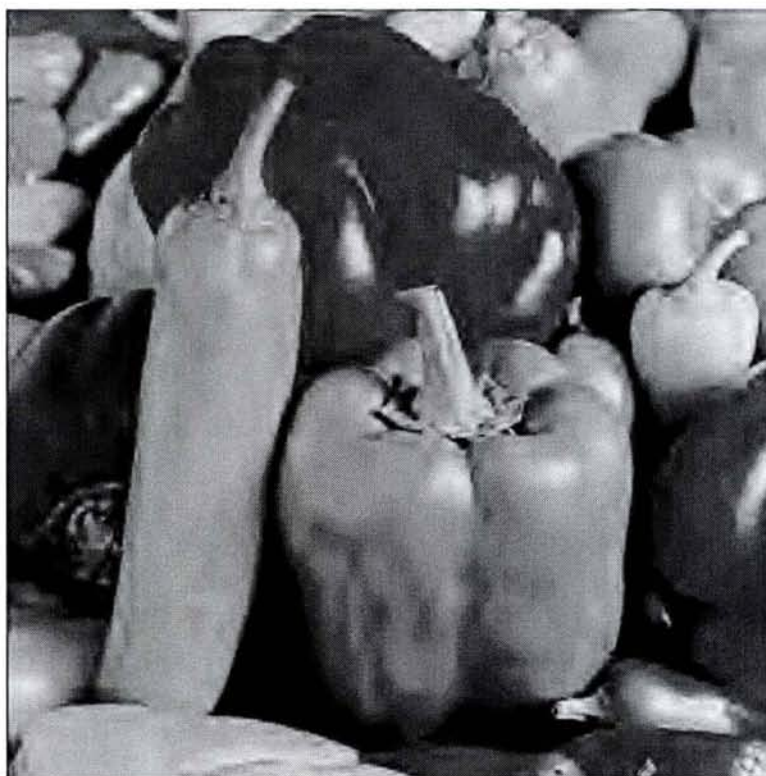


IMAGE 3.16 : Image *Peppers* SPIHT, TC=100 :1, PSNR=28,80dB.

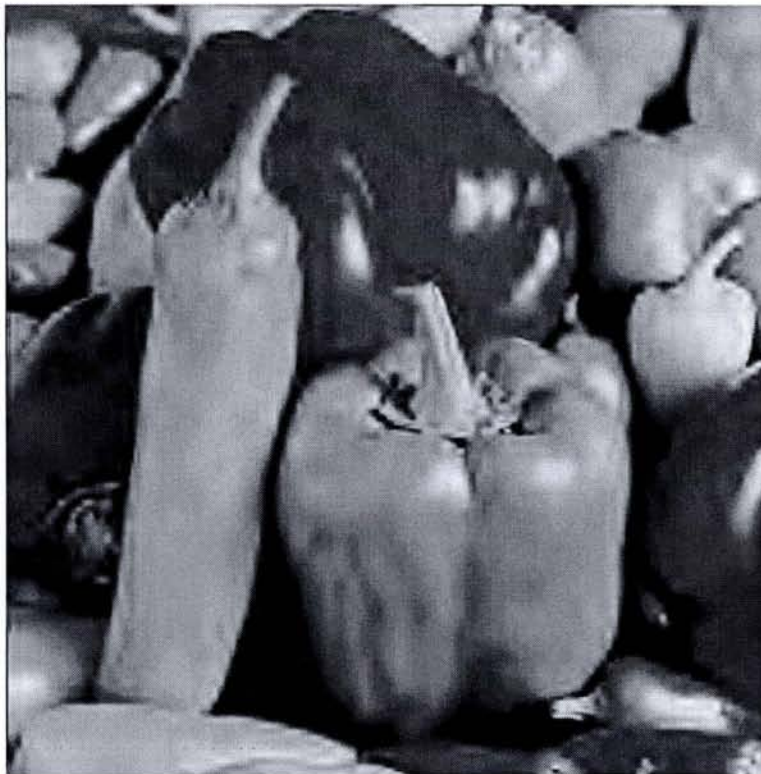


IMAGE 3.17 : Image *Peppers JPEG2000*, TC=100 :1, PSNR=27,87dB.

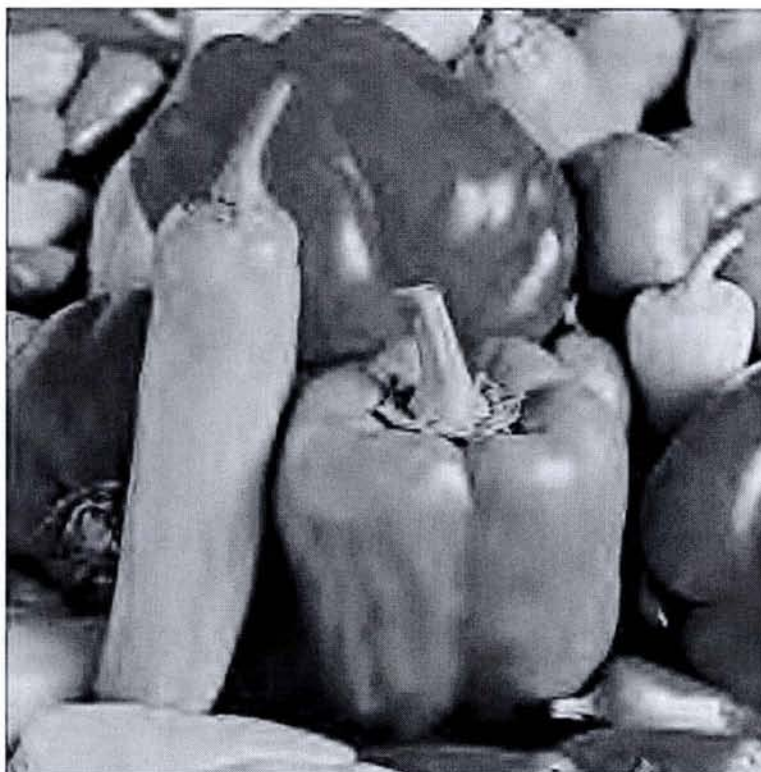


IMAGE 3.18 : Image *Peppers QVAZM + Stack-Run* (binaire), TC=100 :1 (0,0792bpp), PSNR=28,46dB.

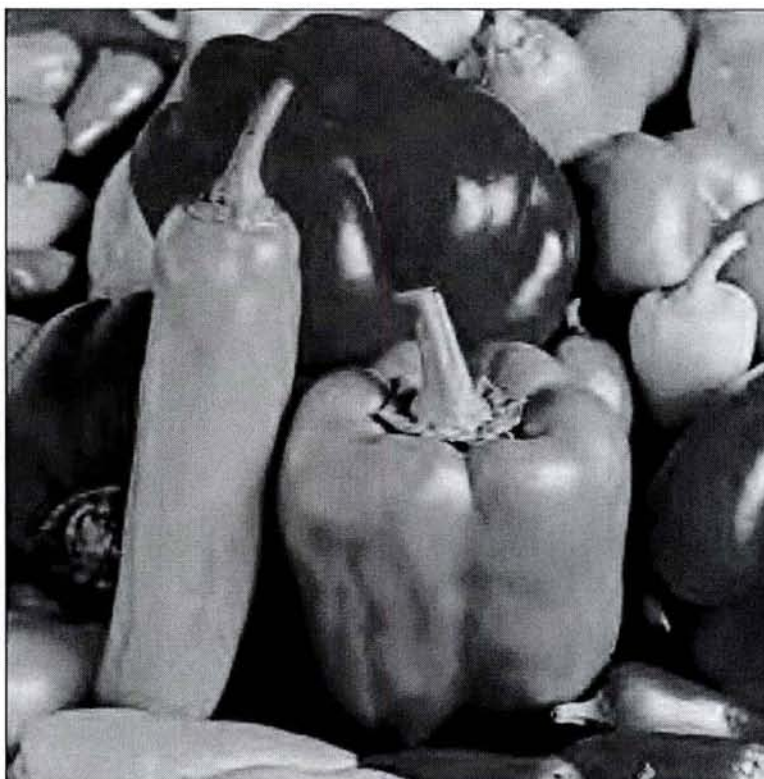


IMAGE 3.19 : Image *Peppers SPIHT*, TC=64 :1, PSNR=30,65dB.

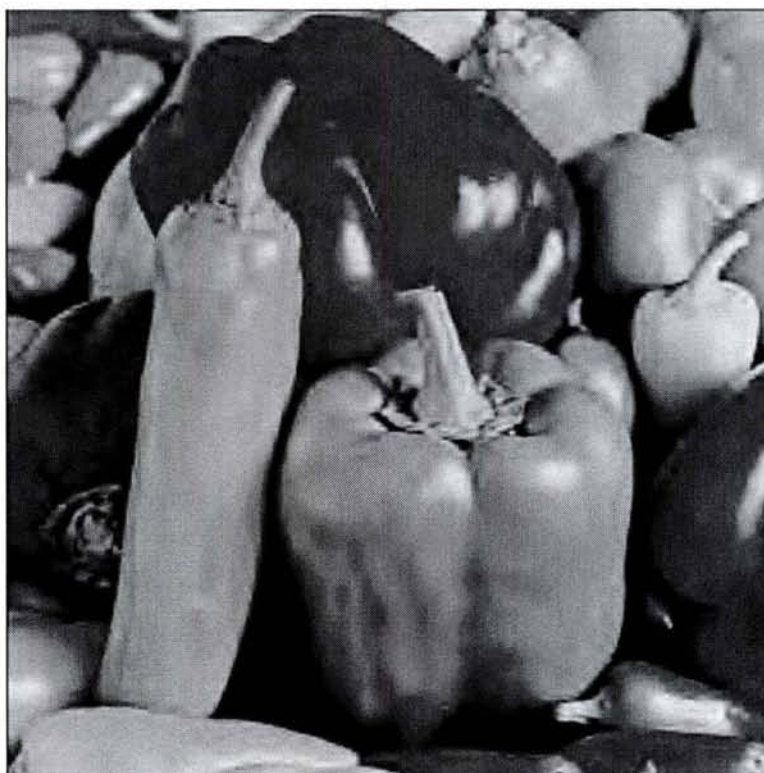


IMAGE 3.20 : Image *Peppers JPEG2000*, TC=64 :1, PSNR=30,19dB.

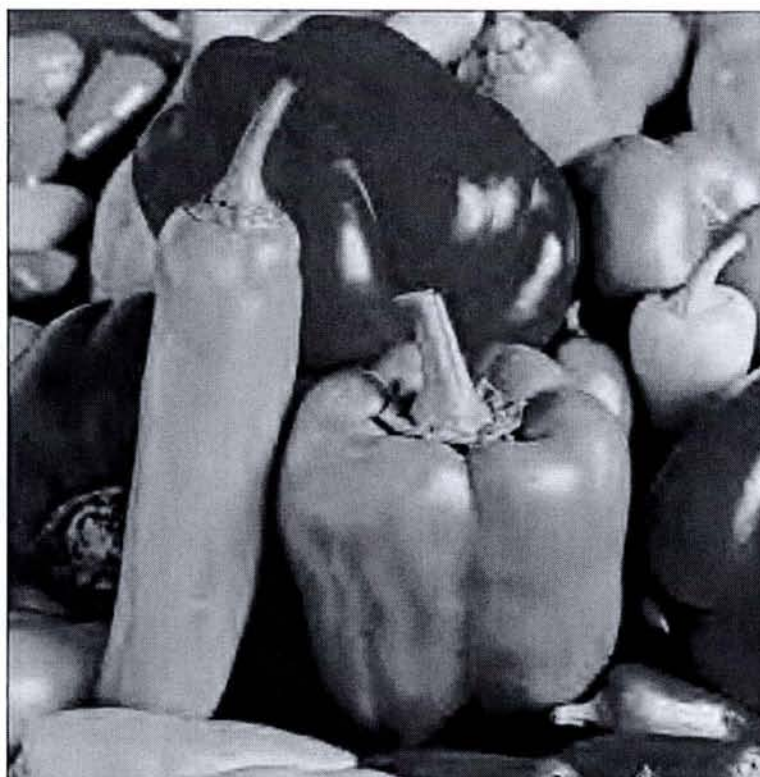


IMAGE 3.21 : Image *Peppers* QVAZM + *Stack-Run* (binaire), $TC \approx 64 : 1$ (0,122 bpp),
PSNR=30,17dB



IMAGE 3.22 : Image *Boat* originale.



IMAGE 3.23 : Image *Boat SPIHT*, TC=64 :1, PSNR=28,16dB.



IMAGE 3.24 : Image *Boat JPEG2000*, TC=64 :1, PSNR=27,97dB.



IMAGE 3.25 : Image *Boat* QVAZM + *Stack-Run* (binaire), TC=64 :1, PSNR=27,96dB

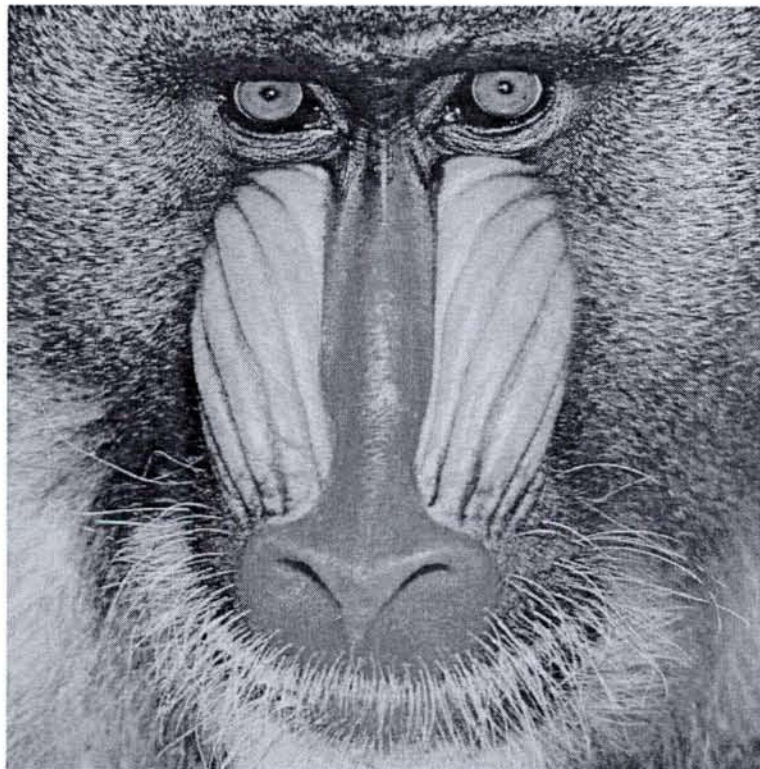


IMAGE 3.26 : Image *Mandrill* originale.

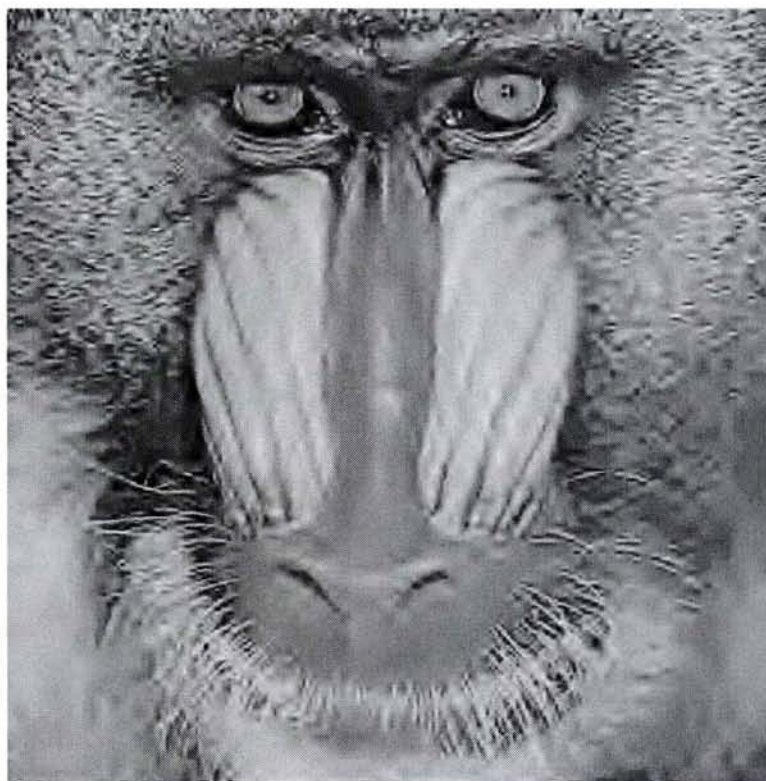


IMAGE 3.27 : Image *Mandrill SPIHT*, $TC \simeq 64 : 1$ (0,122 bpp), PSNR=21,69dB.

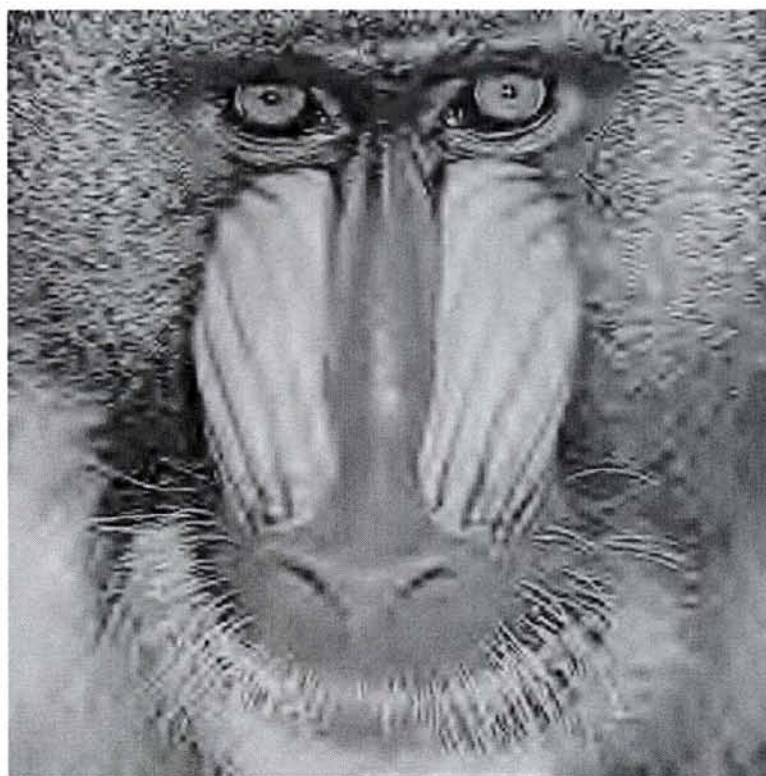


IMAGE 3.28 : Image *Mandrill JPEG2000*, $TC \simeq 64 : 1$ (0,122 bpp), PSNR=21,54dB.

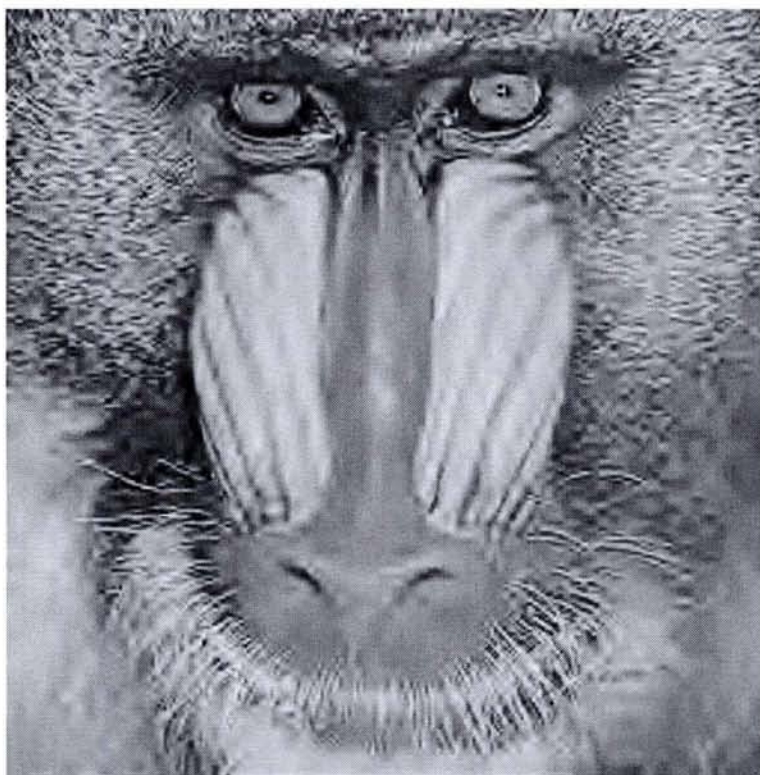


IMAGE 3.29 : Image *Mandrill* QVAZM + *Stack-Run* (binaire), $TC \simeq 64 : 1$ (0,122 bpp),
PSNR=21,68dB

Chapitre 4

Compression par QVAZM et tatouage d'images

4.1 Introduction

L'acquisition, l'intégrité et la protection de l'information et des données constituent aujourd'hui un enjeu majeur à l'échelle mondiale. Avec l'évolution des nouvelles technologies de communications (réseau Internet, transmissions satellite, réseaux téléphoniques) et le développement des techniques de compression numérique, la propagation de l'information et des données numérisées n'a cessé d'augmenter. Il est ainsi devenu possible d'acquérir "simplement" via le réseau Internet bon nombre des productions photographiques, musicales, et cinématographiques. Ce problème de "piratage" (copie de productions sans acquittement de droits d'auteurs) est à l'origine des travaux sur le tatouage¹. C'est une technique récente qui a pour objectif d'intégrer, au sein même des données numériques, une information indélébile et non perceptible. Celle-ci est susceptible d'identifier l'auteur de la production, mais elle peut également renfermer un contenu sémantique (accélération de la recherche dans des bases de données [33]). D'autres fonctionnalités liées au tatouage seront présentées dans la première partie de ce chapitre. Plutôt que de séparer le processus de compression de celui du tatouage, nous avons orienté nos travaux vers une fusion de ces deux techniques. Cette approche semble relativement naturelle dans le sens où les futurs standards de compression (*JPEG2000*, *MPEG7*) ont pour objectif d'optimiser le processus de compression mais également de pouvoir extraire le contenu sémantique des données.

Nos travaux se sont portés en particulier sur le tatouage d'images à niveaux de gris. Nous présenterons dans un premier temps les notions fondamentales du tatouage d'images. Nous introduirons dans un second temps l'intérêt et les propriétés essentielles des techniques de tatouage d'images. Nous développerons ensuite notre schéma de tatouage associé à la Quantification Vectorielle Algébrique avec Zone Morte. Enfin, nous montrerons la robustesse de notre approche face à certaines attaques (en particulier pour un bruit additif gaussien), ainsi que les principaux résultats expérimentaux obtenus.

4.2 Généralités

4.2.1 Terminologie

Dans un contexte général, la dissimulation d'information consiste simplement à cacher une information dans un support. Ce support est appelé médium (dans notre cas : une image), mais une fois que la signature (une marque ou un message) est insérée de manière indélébile, le support

¹ *Watermarking* en anglais

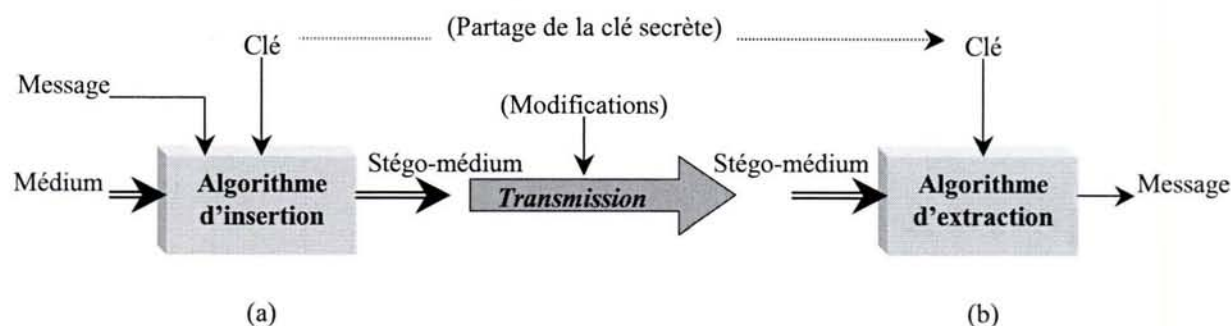


FIG. 4.1: Schéma général de la transmission d'un message dissimulé. (a) Insertion d'un message dans le médium, (b) Extraction du message à partir du stégo-médium.

modifié est appelé stégo-médium (dans notre cas : une image tatouée). Le processus complet de la dissimulation d'information repose sur deux opérations :

- la dissimulation, appelée également l'insertion, consiste à incruster la signature dans l'image ;
- l'extraction, appelée également la détection, consiste à récupérer la signature de l'image tatouée.

A ces deux processus s'ajoutent certaines notions liées à la **cryptographie**. Shannon est à l'origine de cette discipline [82], il a introduit les notions de base d'un système de communication chiffré. En fait, on distingue deux types d'algorithmes de chiffrement :

- à clé secrète (ou symétrique) : celui-ci consiste à utiliser une information secrète (une clé) lors de l'insertion et de la détection du message. Il est donc nécessaire que les deux protagonistes disposent de la clé (échange préalable) ;
- à clé publique (ou asymétrique) : celui-ci évite aux deux interlocuteurs de partager un secret (la clé). En fait, chacun dispose d'un couple de clés : l'une publique et l'autre privée (secrète). Par exemple, si les clés publiques sont à la disposition de tous (dans un annuaire) alors pour envoyer un message à "Bob²", "Alice" le chiffre à l'aide de la clé publique de "Bob". Seul ce dernier est en mesure de déchiffrer le message reçu, à l'aide de sa clé privée (secrète).

Nous présentons sur la figure 4.1, le schéma général de la transmission d'un message dissimulé.

Il existe différentes catégories qui utilisent le concept de dissimulation d'information dont les caractéristiques varient suivant les objectifs recherchés :

- la **stéganographie** : cette approche consiste à dissimuler un message à travers un médium de telle sorte qu'il ne soit jamais intercepté par un processus de surveillance. Le message est indépendant du médium ;
- le **tatouage** : cette approche permet d'insérer une marque ou un message dans un médium afin d'en identifier l'auteur (protection des droits d'auteurs) ou même le contenu sémantique ("indexation" pour une base de données) ;
- le **fingerprinting** : cette approche dépose une empreinte propre à l'utilisateur au sein du médium, il est alors possible de retrouver la provenance d'une copie illicite (traçabilité de copie illégale).

Chacune de ces catégories peut être caractérisée en fonction de trois principaux critères :

- l'imperceptibilité : ce critère évalue les performances à rendre le message invisible sur l'image tatouée. C'est le critère fondamental de la dissimulation d'information quel que soit le type d'approche. Nous rappelons que le message ou la marque est indélébile sur l'image tatouée ;
- la robustesse : ce critère permet d'évaluer l'aptitude à préserver la signature face aux modifications (ou attaques) de l'image tatouée. Il est fondamental dans la majorité des approches.

² "Alice" et "Bob" sont les prénoms utilisés plutôt que les lettres A et B.

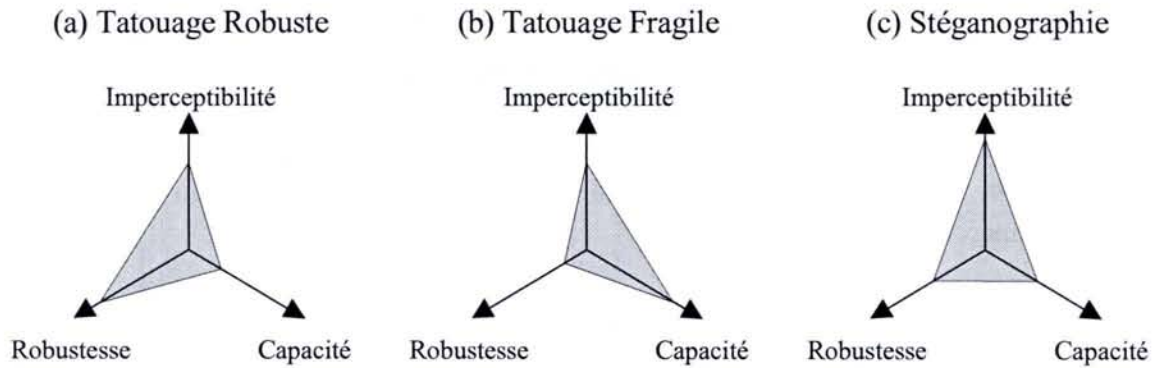


FIG. 4.2: Répartition des trois critères pour différentes approches.

Cependant dans le cadre de l'authentification de documents (carte d'identité, etc.), la marque ou le message doit disparaître si des modifications sont apportées à l'image tatouée : on parle alors de tatouage fragile³ ;

- la capacité : ce critère correspond à la longueur du message (quantité de bits) à insérer dans l'image tatouée. Il peut être exprimé par unité d'accès (par exemple : le nombre de bits par pixel dans une image, ou le nombre de bits par seconde pour la musique). Cette capacité est différente suivant le type d'approche.

Notre objectif est donc de trouver le meilleur compromis imperceptibilité/ robustesse/ capacité suivant le type d'approche. La figure 4.2 montre une représentation possible de ces trois critères pour différentes approches.

4.2.2 Les attaques dans le cas des images

Pour une capacité du message fixée, les deux principales contraintes du tatouage sont l'imperceptibilité du message incrusté et sa robustesse face aux attaques. En particulier, dans le cas de la protection des droits d'auteurs, le tatouage doit être très robuste face à différentes attaques dont l'objectif est de détruire le message tout en préservant la qualité de l'image tatouée. Nous présentons, dans les deux paragraphes suivants, les traitements qui effectuent des modifications volontairement ou non sur l'image tatouée en utilisant des attaques locales ou globales.

4.2.2.1 Les traitements innocents

Les traitements innocents constituent la majorité des attaques. Ils peuvent être regroupés de la manière suivante :

- la compression : ce traitement consiste à réduire le coût de stockage de l'image tatouée sans dégradation visuelle. Cela entraîne l'élimination de certaines composantes de l'image ne contenant pas d'informations significatives pour le Système Visuel Humain (SVH). Dans le cadre de nos travaux, nous nous sommes inspirés de cette constatation. Nous avons ainsi amélioré la robustesse de notre algorithme de tatouage en intégrant le message sur les composantes significatives ;
- les opérations de lissage et de rehaussement de contraste : ce sont des traitements classiques liés à l'amélioration de la qualité visuelle d'une image. Soit ils préservent uniquement les basses fréquences (effet de débruitage) soit ils accentuent les hautes fréquences (amélioration

³ fragile watermarking en anglais.

des contours) en utilisant différentes méthodes (égalisation d'histogramme, répartition des niveaux de gris,...) ;

- les modifications géométriques : dans le cas d'une image, ces traitements consistent à redimensionner, à faire pivoter, ou à découper une partie de l'image ;
- les conversions analogique/ numérique : ce traitement concerne la re-numérisation d'une image tatouée. Par exemple, lorsque l'on numérise une image, imprimée sur papier, à l'aide d'un scanner.

4.2.2.2 Les traitements malveillants

Ce sont ces traitements qui demandent la plus grande robustesse. En effet, les modifications engendrées sont de véritables attaques qui tentent de rendre la signature illisible et donc inexploitable. Ces traitements entrent dans le cadre du "piratage". Nous pouvons distinguer plusieurs types d'attaques :

- celles qui effacent la signature : ce sont par exemple les attaques de type débruitage [92] ;
- celles qui désynchronisent le message : en particulier les attaques de type "jittering" [70], "mosaïque" ou "StirMark" [71] ;
- celles qui exploitent les failles d'un schéma de tatouage : "L'attaque d'IBM".

4.2.3 Propriétés du tatouage d'images

De nombreux travaux ont été développés dans le domaine du tatouage d'images. Nous invitons le lecteur à consulter le site www.watermarkingworld.org au sein duquel différents ouvrages sont référencés [50]. Comme nous l'avons expliqué, la dissimulation d'une information dans un médium consiste, dans notre cas, à incruster une marque ou un message dans une image suivant le type d'application visée :

- Copyright (droit d'auteur) : le message incrusté contient des informations attestant de l'identité du propriétaire de l'image. Ce type de tatouage doit être robuste face aux traitements malveillants (ou attaques) de l'image ;
- Authentification : la signature joue un rôle de "mouchard", elle doit disparaître si des modifications ont été apportées à l'image ;
- Indexation : le message contient des informations sur le contenu sémantique de l'image. Les informations permettront par exemple d'accélérer la recherche ou la classification d'une image dans une base de données. Ce type de tatouage doit être surtout robuste face aux traitements innocents ;
- Méta-documents : l'information insérée est secondaire et indépendante de l'image, la transmission s'effectue ainsi sur un seul canal. Ce type de tatouage rejoint la stéganographie.

4.2.4 Classification des schémas de tatouage

Les différents schémas de tatouage varient suivant le domaine d'insertion utilisé (spatial, fréquentiel, multirésolution) et suivant le type d'attaque supportée (traitements innocents ou malveillants). Une synthèse de ceux-ci a été proposée suivant deux classes [6] :

- les schémas additifs : cette classe de schéma assimile l'image à un canal bruité où la signature, considérée comme le signal à transmettre (séquence binaire) va être ajoutée (modulée) sur ce canal bruité (l'image). Ce principe de transmission d'information sur un canal bruité est appelé l'étalement de spectre [73] ;
- les schémas substitutifs : cette classe de schémas consiste à insérer la signature en remplaçant certaines composantes d'une image (ou d'une sous-bande) par d'autres valeurs caractérisant la signature (la séquence binaire). C'est ce type de schéma que nous avons intégré.

4.3 L'analyse multirésolution pour le tatouage d'images

4.3.1 Domaines d'insertions

Pour répondre aux différents types d'applications, les schémas de tatouage d'images ont évolué avec l'émergence des nouvelles applications. L'objectif étant toujours de trouver le meilleur compromis imperceptibilité/ robustesse/ capacité suivant les applications. Des travaux se sont ainsi développés dans les trois domaines suivants :

- le domaine **spatial** : le message est directement inséré sur les valeurs des pixels de l'image. Le coût calculatoire est ainsi plus faible, il est alors possible d'envisager des applications "temps-réel". De plus, face aux attaques géométriques, les schémas de tatouage dans ce domaine sont assez robustes car le message n'est pas dégradé mais simplement déplacé [7][8] ;
- le domaine **fréquentiel** : ce domaine (transformée *DCT*) est bien adapté pour un tatouage robuste (d'image ou de vidéo) face aux attaques utilisant les anciens standards de compression (*JPEG*, *MPEG2*). Par contre, c'est un domaine peu robuste aux attaques géométriques [6] ;
- le domaine **multirésolution** : c'est le domaine qui est le mieux adapté face aux différents types d'attaques. La décomposition en sous-bandes de la transformée en ondelettes permet d'isoler les "fréquences" adaptées au tatouage. De plus, ce domaine est celui qui est utilisé par les nouveaux standards de compression (*JPEG2000*, *MPEG4*) [86][87][59].

4.3.2 Insertion et détection dans le domaine multirésolution

Le schéma que nous proposons a été mis en place dans le domaine multirésolution. Nos travaux se sont donc focalisés sur la compression et le tatouage d'images dans ce domaine. Dans la suite du mémoire, nous présenterons l'intérêt d'utiliser le principe de compression et de tatouage d'images conjoints. Comme nous l'avons introduit au début du chapitre, un schéma de tatouage d'images se décompose en deux opérations : l'insertion et la détection de la signature.

4.3.2.1 L'insertion

La figure 4.3 présente le schéma général de l'insertion d'une signature dans une image. Dans un premier temps, une transformée en ondelettes $T(I)$ est appliquée à l'image originale I ($\mathbf{J}$ niveaux de résolution engendrent $\mathbf{K} = 3\mathbf{J}$ sous-bandes $SB_{\mathbf{k}}$ où $\mathbf{k} = \{0, 1, \dots, \mathbf{K} - 1\}$). On obtient ainsi l'ensemble des sous-bandes $SB_{\mathbf{k}}$:

$$T(I) = \{SB_{\mathbf{BF}}, SB_{\mathbf{k}}\} \quad \forall \quad \mathbf{k} = \{0, 1, \dots, \mathbf{K} - 1\} \quad (4.1)$$

La signature w va donc être insérée dans quelques unes des sous-bandes $SB_{\mathbf{k}}$. Elle est composée d'un nombre M_w de bits, nombre plus ou moins important selon l'application :

$$w = \{b_1, \dots, b_{M_w}\} \quad b_j \in \{0, 1\} \quad (4.2)$$

Nous effectuons ensuite l'insertion $W(SB_{\mathbf{k}}, w, K)$ de la signature w à l'aide de la clé K sur certaines sous-bandes $SB_{\mathbf{k}}$. Nous disposons alors de quelques sous-bandes tatouées $SB_{\mathbf{k}}^W$:

$$W(\{SB_{\mathbf{k}}\}, w, K) = \{SB_{\mathbf{k}}^W\} \quad (4.3)$$

Dans le cas par exemple d'une décomposition sur $\mathbf{J} = 5$ niveaux de résolution, si nous tatouons les sous-bandes $SB_{\mathbf{k}}$ du niveau intermédiaire $\mathbf{j} = 3$ alors :

$$W(\{SB_{\mathbf{k}}\}, w, K) = \{SB_{\mathbf{k}}^W\} \quad \text{avec } \mathbf{k} \in \{6, 7, 8\} \quad (4.4)$$

Afin de simplifier les notations, les sous-bandes $\mathbf{k}$ qui se sont pas tatouées seront d'indice $\bar{\mathbf{k}}$, d'où :

$$\{SB_{\mathbf{k}}\} = \{SB_{\bar{\mathbf{k}}}\} \cup \{SB_{\mathbf{k}}^W\}$$

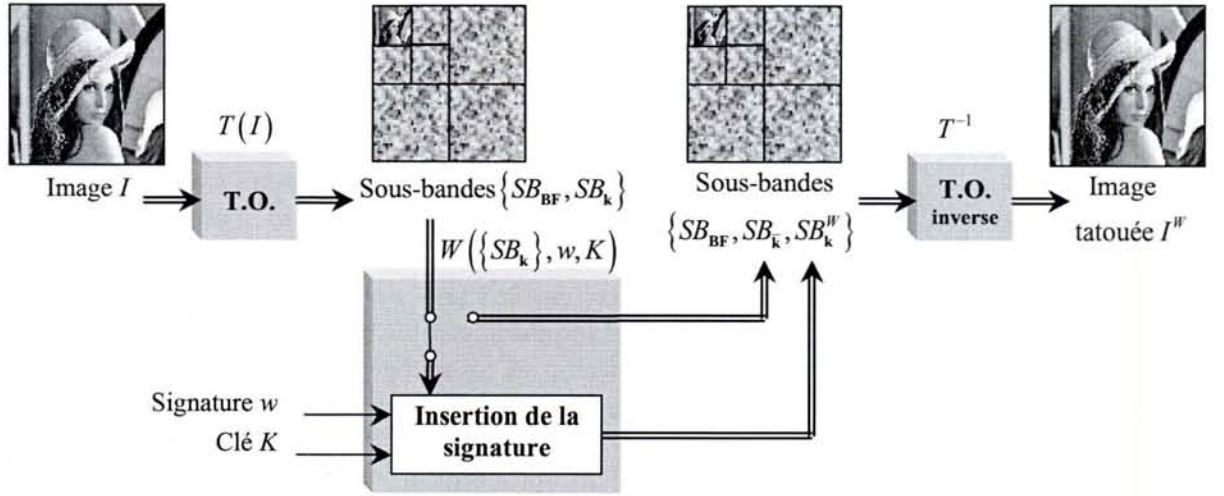


FIG. 4.3: Schéma général d'insertion d'une signature dans une image.

Nous rappelons que les modifications apportées aux sous-bandes lors de l'insertion sont irréversibles (signature indélébile). Il suffit ensuite d'appliquer une transformée en ondelettes inverse T^{-1} de l'ensemble des sous-bandes $\{SB_{BF}, SB_{\bar{k}}, SB_k^W\}$ pour obtenir l'image tatouée I^W :

$$T^{-1}(\{SB_{BF}, SB_{\bar{k}}, SB_k^W\}) = I^W \quad (4.5)$$

4.3.2.2 La détection

La figure 4.4 présente le schéma général de la détection d'une signature à partir d'une image tatouée I^W altérée par les modifications d'une attaque $A(I^W)$ (traitements innocents ou malveillants). Nous présentons en particulier sur cette figure le cas où des modifications sont effectuées dans le domaine spatial (par exemple dans le cas d'un bruit additif), cependant certaines attaques interviennent directement dans le domaine multirésolution. Nous disposons alors d'une nouvelle image attaquée $\tilde{I}^W$:

$$A(I^W) = \tilde{I}^W \quad (4.6)$$

Nous effectuons ensuite la transformée en ondelettes $T(\tilde{I}^W)$ de l'image attaquée $\tilde{I}^W$ afin d'obtenir l'ensemble des sous-bandes non tatouées et attaquées $\widetilde{SB_{\bar{k}}}$ et des sous-bandes tatouées et attaquées $\widetilde{SB_k^W}$:

$$T(\tilde{I}^W) = \{\widetilde{SB_{BF}}, \widetilde{SB_{\bar{k}}}, \widetilde{SB_k^W}\} \quad (4.7)$$

Finalement, il ne reste plus qu'à effectuer l'extraction (détection : sélection puis lecture) $Ext(\widetilde{SB_k^W}, K)$ de la signature w d'origine à l'aide de la clé K des sous-bandes $\widetilde{SB_k^W}$:

$$Ext(\widetilde{SB_k^W}, K) = \tilde{w} \quad (4.8)$$

Cependant, suivant le type d'attaque et la robustesse du schéma de tatouage, il est possible que la signature détectée $\tilde{w}$ ne corresponde plus à la signature d'origine :

$$\tilde{w} \stackrel{?}{=} w \quad (4.9)$$

C'est tout l'objet de l'étude de la robustesse d'une méthode de tatouage.

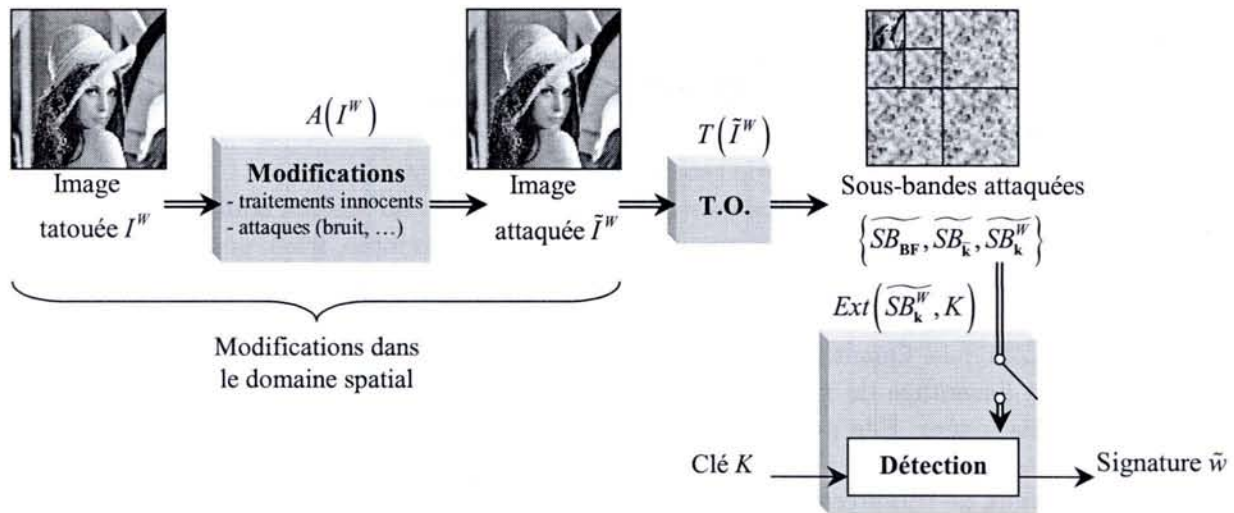


FIG. 4.4: Schéma général de détection d'une signature à partir d'une image tatouée et attaquée.

4.3.3 Conclusion

Cette synthèse a permis de dégager les notions fondamentales et les caractéristiques d'un schéma de tatouage en particulier dans le domaine multirésolution. Nous allons présenter maintenant notre approche basée sur une technique de compression et de tatouage d'images conjoints.

4.4 Compression et tatouage d'images conjoints

4.4.1 Introduction

Notre approche est basée sur un schéma substitutif dans le domaine multirésolution où les coefficients d'ondelettes ont subi une Quantification Vectorielle Algébrique avec Zone Morte (voir chapitre 2 et 3). L'image obtenue est donc une image comprimée et tatouée. En d'autres termes, nous appliquons notre technique de tatouage directement sur les données quantifiées. Différents travaux ont montré l'intérêt d'utiliser ce type de schéma [51][80] [60][25][24]. L'objectif commun de la compression et du tatouage d'images est de minimiser l'impact visuel des modifications apportées à l'image originale. Cependant, les modifications apportées ont des objectifs totalement opposés. La compression d'images vise à supprimer les données imperceptibles afin de réduire la quantité d'information, alors que le tatouage consiste à ajouter une information imperceptible (intégration d'une signature).

À travers cette partie, nous montrerons les avantages de cette approche conjointe et ses performances en terme de robustesse. Nous présenterons ensuite les étapes de tatouage d'images (insertion, détection) que nous avons mises en oeuvre. Puis, nous décrirons le langage de substitution qui a été utilisé. Enfin, nous donnerons le détail des algorithmes développés.

4.4.2 Motivations

Les justifications de notre schéma de tatouage sont basées sur trois aspects :

- le domaine multirésolution : c'est le domaine le mieux adapté au système visuel humain, dans le sens où celui-ci est plus ou moins sensible à certaines gammes fréquentielles sélectionnées. De plus, c'est le domaine utilisé par les nouveaux standards de compression (*JPEG2000*, *MPEG4*). Ainsi, pour une transformée en ondelettes sur $J = 5$ niveaux de décomposition, nous avons défini le niveau de décomposition intermédiaire $j = 3$ comme le meilleur niveau

d'insertion d'une signature. En effet, les coefficients de ce niveau sont généralement bien préservés lors de la phase de quantification, car ils contiennent des informations significatives de l'image, cela permet de mieux préserver la signature face à différentes attaques. De plus, la modification de certains coefficients de ce niveau n'engendre pas de dégradations trop importantes lors de la phase de reconstruction de l'image ;

- un schéma substitutif : ce type de schéma, contrairement au schéma additif, permet d'utiliser au maximum la capacité du support d'insertion. Par exemple, il est possible d'insérer 1 bit d'information b_j (issu du message w) pour chaque vecteur (non nul) d'une sous-bande. Pour cela, il est nécessaire d'utiliser un langage, celui-ci sera détaillé dans la suite du chapitre ;
- la QVAZM : la Quantification Vectorielle Algébrique avec Zone Morte d'une sous-bande présente l'avantage de ne conserver que les coefficients d'ondelettes significatifs, les autres étant mis à zéro. Elle permet ainsi d'améliorer la robustesse de la signature insérée face à différents type d'attaques, comme en particulier les attaques de type bruit additif. Cependant, plus le taux de compression est élevé, moins il y a de vecteurs susceptibles d'être tatoués. La capacité d'insertion de bits diminue donc avec le taux de compression. Il est donc nécessaire de tenir compte du compromis robustesse/ capacité suivant l'application visée.

4.4.3 QVAZM, insertion et détection

4.4.3.1 L'opération d'insertion

La figure 4.5 présente notre schéma général de tatouage d'images (sélectionner des vecteurs puis insérer les bits de la signature). Celui-ci est composé de deux phases :

1. la phase de Quantification : celle-ci est réalisée par la Quantification Vectorielle Algébrique avec Zone Morte (voir chapitre 2 et 3). La figure 4.5(a) présente la phase de quantification $QV(X)$ des vecteurs X d'une sous-bande :

$$QV : \begin{cases} \mathcal{P}^n \longrightarrow \mathcal{C} = \{Y_0, \dots, Y_{L-1}\} \\ X \longrightarrow QV(X) = Y_i = \{y_1, \dots, y_n\} \end{cases} \quad (4.10)$$

2. la phase de tatouage : celle-ci est réalisée par un schéma de substitution. Elle permet d'insérer la signature w à travers différentes sous-bandes SB_k sur certains vecteurs quantifiés Y_i . Afin de simplifier les schémas, nous présenterons les figures dans le cas d'une seule sous-bande tatouée.

Nous allons détailler les différentes étapes de l'insertion de la signature $w = \{b_1, \dots, b_{M_w}\}$ avec $b_j \in \{0, 1\}$ de la figure 4.5(b). Dans un premier temps, nous effectuons une sélection $S(\{Y_i\}, K)$ à l'aide de la clé K parmi l'ensemble des vecteurs $\{Y_i\}$ susceptibles d'être tatoués, l'ensemble de ces vecteurs est noté Y_s :

$$S(\{Y_i\}, K) = Y_s \quad (4.11)$$

Les vecteurs qui n'ont pas été sélectionnés sont notés $Y_{\bar{s}}$. Nous pouvons écrire les propriétés suivantes :

$$\begin{aligned} \{Y_i\} &= Y_s \cup Y_{\bar{s}} \\ Y_s \cap Y_{\bar{s}} &= \emptyset \\ \text{Cardinal}(Y_s) &= M_w \end{aligned} \quad (4.12)$$

Ensuite, sur chaque vecteur sélectionné Y_{s_j} , nous effectuons l'insertion d'un bit b_j de la signature :

$$W_b(Y_{s_j}, b_j) = Y_{s_j}^W \quad (4.13)$$

L'étape d'insertion utilise un langage qui modifie les composantes du vecteur Y_{s_j} en fonction du bit b_j à insérer. Appliquée à tous les vecteurs sélectionnés, l'étape d'insertion conduit à l'ensemble

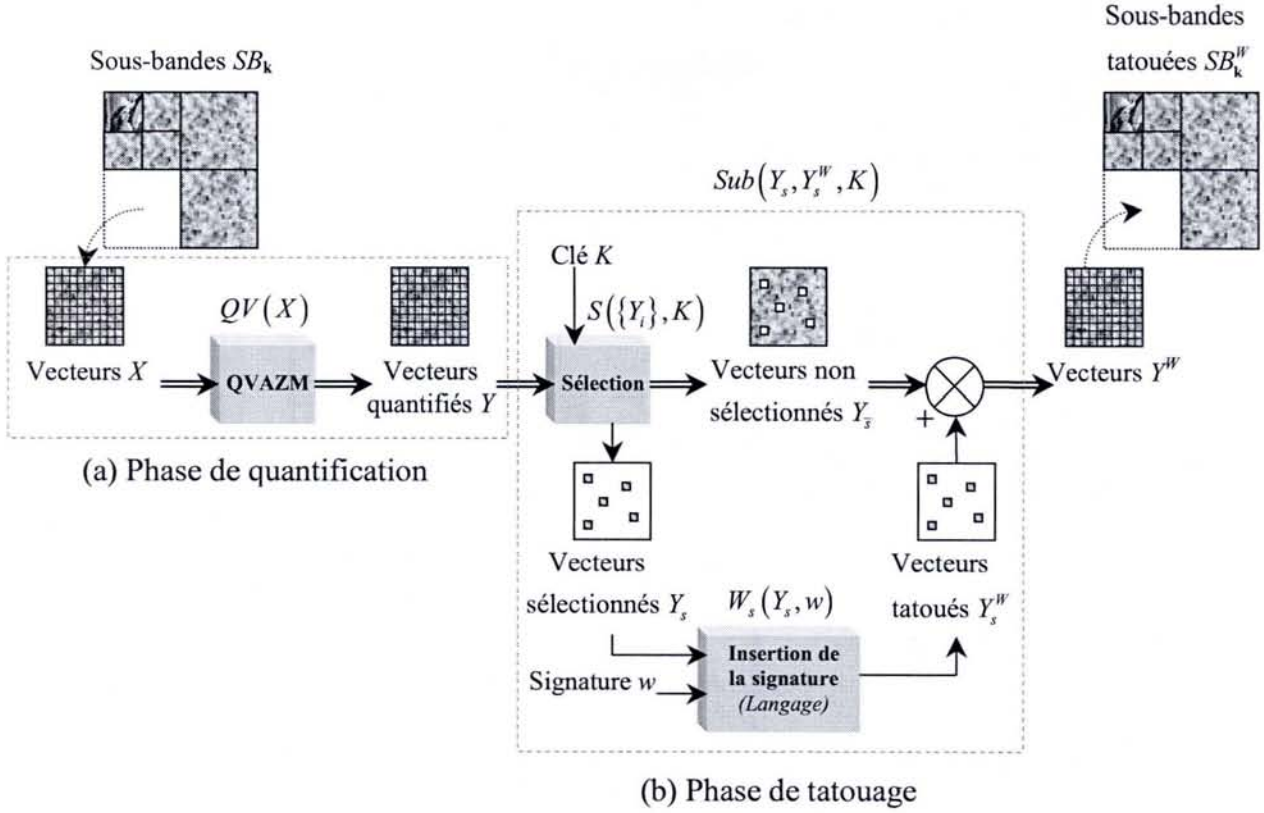


FIG. 4.5: Schéma de compression et tatouage conjoints (opération d'insertion de la signature).

des vecteurs tatoués Y_s^W :

$$W_s(Y_s, w) = Y_s^W \quad | \quad Y_s^W = \{Y_{s_j}^W\} \quad (4.14)$$

Il ne reste plus qu'à réintégrer les vecteurs non sélectionnés $Y_{\bar{s}}$ avec les vecteurs sélectionnés et tatoués Y_s^W pour obtenir l'ensemble des vecteurs Y^W de la sous-bande tatouée SB_k^W :

$$Y_{\bar{s}} \cup Y_s^W = \{Y_i^W\} \quad (4.15)$$

Ces différentes étapes réalisent la substitution $Sub(Y_s, Y_s^W, K)$ des vecteurs Y_s avec les vecteurs tatoués Y_s^W afin d'intégrer la signature. Il suffit ensuite de renouveler ces différentes étapes sur d'autres sous-bandes :

$$Sub(Y_s, Y_s^W, K) = \{Y_i^W\}$$

4.4.3.2 L'opération de détection

La figure 4.6 présente le schéma d'extraction de la signature (retrouver les vecteurs sélectionnés + lecture des bits). Nous appliquons dans un premier temps la transformée en ondelettes $T(\tilde{I}^W)$ de l'image tatouée et attaquée $\tilde{I}^W$, nous obtenons alors l'ensemble des sous-bandes :

$$T(\tilde{I}^W) = \{\widetilde{SB_{BF}}, \widetilde{SB_k}, \widetilde{SB_k^W}\} \quad (4.16)$$

Ensuite, à partir d'une sous-bande $\widetilde{SB_k^W}$ composée de l'ensemble des vecteurs $\{\tilde{Y}_i^W\}$, nous effectuons une requantification afin de s'affranchir des effets de l'attaque :

$$Q(\tilde{Y}_i^W) = \tilde{Y}_i^{Wq} \quad (4.17)$$

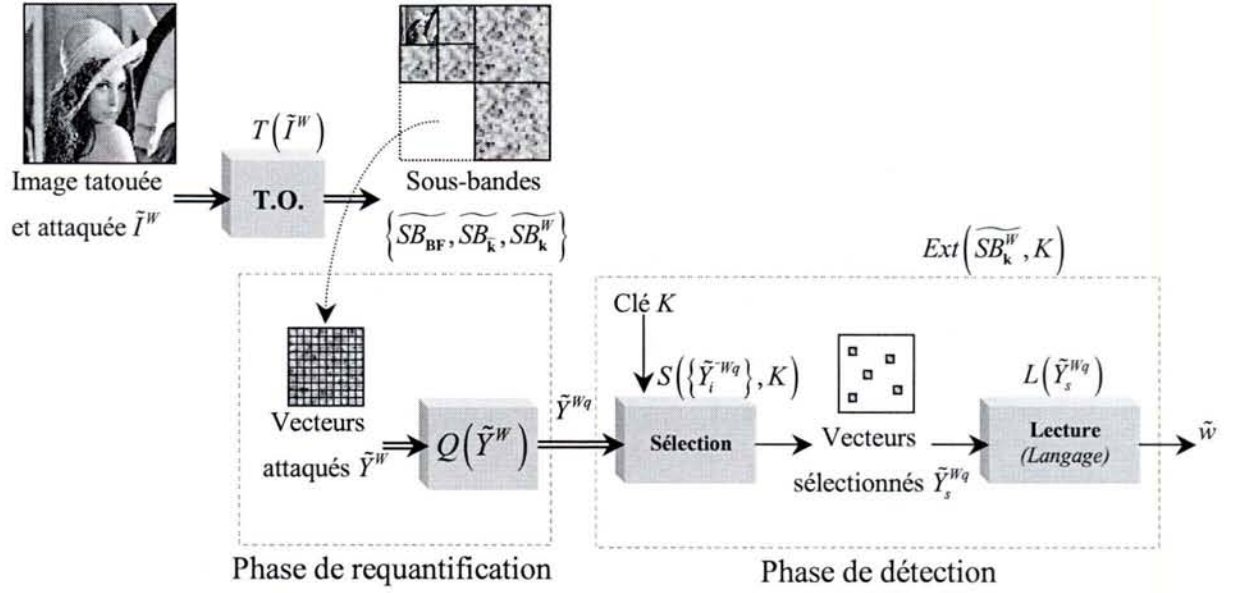


FIG. 4.6: Schéma de détection de la signature.

et la sélection des vecteurs tatoués $\tilde{Y}_s^{Wq}$ à l'aide de la clé K :

$$S\left(\left\{\tilde{Y}_i^{Wq}\right\}, K\right) = \tilde{Y}_s^{Wq} \quad | \quad \tilde{Y}_s^{Wq} = \left\{\tilde{Y}_{s_j}^{Wq}\right\} \quad \forall \quad j = \{1, \dots, M_w\} \quad (4.18)$$

Il ne reste plus qu'à effectuer la lecture de la signature incrustée dans les vecteurs $\tilde{Y}_s^W$ (en fonction du langage utilisé) :

$$L\left(\tilde{Y}_s^{Wq}\right) = \tilde{w} \quad (4.19)$$

Ces différentes étapes constituent la phase d'extraction (détection : sélection puis lecture) $Ext\left(\widetilde{SB}_k^w, K\right)$. Cependant, cette phase n'est pas toujours fiable. Elle dépend de l'importance et du type de l'attaque. On peut alors se poser la question : "La signature extraite $\tilde{w}$ est-elle identique à la signature d'origine w ?" :

$$\tilde{w} \stackrel{?}{=} w \quad (4.20)$$

La réponse à cette question permet d'apprécier la robustesse du schéma de tatouage.

4.4.4 Langage utilisé

4.4.4.1 Propriétés du langage

Comme nous l'avons précisé, il est nécessaire de mettre en place un langage. Celui-ci doit permettre d'insérer de manière invisible et indélébile un message (dans notre cas : 1 bit b_j par vecteur sélectionné Y_{s_j}). C'est le même langage qui est utilisé lors de l'insertion et lors de la lecture de la signature. La lecture est effectuée sur le vecteur requantifié, sélectionné et attaqué $\tilde{Y}_{s_j}^{Wq}$, afin d'alléger la notation, nous définissons $Y_j^W = \tilde{Y}_{s_j}^{Wq}$.

Nous rappelons la description du dictionnaire $\mathcal{C}_{ZM}$ lors d'une Quantification Vectorielle Algébrique avec Zone Morte (défini au chapitre 2) :

$$\mathcal{C}_{ZM} = \{Y \in \gamma\mathbb{Z}^n \mid \gamma\delta \leq \|Y\|_1 \leq \gamma r_T\} \cup \{0\} \quad (4.21)$$

où γ est le facteur de projection, δ le seuil de la zone morte et r_T le rayon de troncature du dictionnaire.

Le langage utilisé doit permettre d'insérer la signature (i.e. le message composé d'un ensemble de bits b_j) :

$$w = \{b_1, \dots, b_{M_w}\} \quad b_j \in \{0, 1\} \quad (4.22)$$

Afin de construire notre langage, il est nécessaire de :

- définir deux classes distinctes du dictionnaire : une classe associée au bit 0 et une classe associée au bit 1. Le dictionnaire est donc partagé en deux classes (dont le vecteur nul est exclu car impossible à tatouer) :

$$\begin{aligned} \text{Classe 0 : } & \mathcal{C}_{ZM}^*(0) \\ \text{Classe 1 : } & \mathcal{C}_{ZM}^*(1) \end{aligned} \quad (4.23)$$

Nous donnons également les propriétés suivantes :

$$\begin{aligned} \mathcal{C}_{ZM}^*(0) \cap \mathcal{C}_{ZM}^*(1) &= \emptyset \\ \mathcal{C}_{ZM}^*(0) \cup \mathcal{C}_{ZM}^*(1) &= \mathcal{C}_{ZM}^* \\ \mathcal{C}_{ZM}^* &= \mathcal{C}_{ZM} - \{0\} \end{aligned} \quad (4.24)$$

- établir une transformation de marquage W_b : c'est cette transformation qui va modifier le vecteur quantifié Y_j en un vecteur tatoué Y_j^W appartenant soit à la classe 0 (bit 0 à insérer) soit à la classe 1 (bit 1 à insérer). Les propriétés de cette transformation ont été définies de la manière suivante : d'une part, elle doit être isonorme (i.e. conserver la même énergie) :

$$\|Y_j^W\|_1 = \|Y_j\|_1 \quad (4.25)$$

Cela minimise l'impact visuel des modifications, et permet également de préserver son rayon sur l'isosurface initiale (avantage lié aux techniques d'indexage). D'autre part, la transformation doit être non équivoque, c'est-à-dire que les classes de vecteurs doivent être strictement disjointes afin d'éviter toute ambiguïté lors de la phase de lecture (voir les propriétés 4.24).

$$\begin{aligned} W_b : \mathcal{C}_{ZM}^* \times \{0, 1\} &\rightarrow \mathcal{C}_{ZM}^*(0) \cup \mathcal{C}_{ZM}^*(1) \\ W_b(Y_j, b_j) &\mapsto Y_j^W \in \mathcal{C}_{ZM}^*(b_j) \end{aligned} \quad (4.26)$$

4.4.4.2 Description du langage

En fonction des propriétés et des contraintes définies précédemment, un langage a été élaboré par Moureaux et Guillemot [64][40] afin de caractériser un vecteur portant l'information d'un bit à 0 (Classe 0) ou un vecteur portant l'information d'un bit à 1 (Classe 1). Les deux classes du dictionnaire ont été établies de la manière suivante :

- **Classe 0** : les deux premières composantes du vecteur Y_j , de distance minimale (non nulle) sont rangées dans l'ordre croissant.
- **Classe 1** : les deux premières composantes du vecteur Y_j , de distance minimale (non nulle) sont rangées dans l'ordre décroissant.

Ce langage consiste donc à trouver dans un premier temps les deux premières composantes de distance minimale (non nulle) $d_{\min}(y_j, y_k)$ d'un vecteur $Y_j = (y_1, \dots, y_n) \in \mathcal{C}_{ZM}^*$. Nous pouvons écrire :

$$\begin{aligned} \min_{Y_j = (y_1, \dots, y_n)} [d(y_j, y_k)] &= d(y_j, y_k) \\ \forall Y = (y_1, \dots, y_n) \in \mathcal{C}_{ZM}^* & \mid \exists(j, k), 0 < d(y_j, y_k) \leq d(y_l, y_m), j < k, \forall l \neq j, \forall m \neq k \end{aligned} \quad (4.27)$$

où $d(y_j, y_k)$ correspond au calcul de la valeur absolue de la différence entre deux composantes d'un vecteur :

$$d(y_j, y_k) = |y_j - y_k| \quad (4.28)$$

La définition mathématique simplifiée des deux classes du dictionnaire est ainsi donnée par :

$$\begin{aligned} \text{Classe 0} : C_{ZM}^*(0) &= \left\{ Y = (y_1, \dots, y_n) \in C_{ZM}^* \mid \exists(j, k), \min_{Y=(y_1, \dots, y_n)} [d(y_j, y_k)] \text{ et } y_j < y_k \right\} \\ \text{Classe 1} : C_{ZM}^*(1) &= \left\{ Y = (y_1, \dots, y_n) \in C_{ZM}^* \mid \exists(j, k), \min_{Y=(y_1, \dots, y_n)} [d(y_j, y_k)] \text{ et } y_j > y_k \right\} \end{aligned}$$

Par exemple, le vecteur $Y_j = (1, 8, 3, 5)$ appartient à la classe 0 : $C_{ZM}^*(0)$ car les deux composantes (1, 3) sont les premières composantes de distance minimale $d(1, 3) = |1 - 3| = 2$ et ces deux composantes sont rangées dans l'ordre croissant $1 < 3$.

Nous pouvons également remarquer que le vecteur nul ainsi que tous les vecteurs dont les composantes sont identiques ne peuvent être tatoués (il ne seront donc jamais sélectionnés). Nous disposons maintenant de toutes les propriétés et de toutes les définitions du schéma de tatouage utilisé. Nous détaillons aux paragraphes suivants les algorithmes des deux opérations que nous avons présentées : l'insertion et la détection.

4.4.5 Algorithme d'insertion

Nous détaillons à travers ce paragraphe, l'algorithme pour l'opération d'insertion. Nous rappelons que pour une transformée en ondelettes sur $J = 5$ niveaux de décomposition, nous avons défini le niveau de décomposition intermédiaire $j = 3$ comme le meilleur niveau d'insertion de la signature $w = \{b_1, \dots, b_{M_w}\} \forall b_l \in \{0, 1\}$. Nous rappelons également que compte tenu du schéma proposé, la signature w est insérée sur les vecteurs Y_s quantifiés par QVAZM et sélectionnés par une clé K .

Afin de garantir la convergence de l'algorithme d'insertion pour n'importe quel vecteur sélectionné, il est nécessaire d'ajouter une étape de relecture du vecteur tatoué $Y_{s_l}^W$, afin de s'assurer que le bit b_l est correctement inséré. Nous avons ainsi établi un algorithme de complexité réduite, permettant de tatouer tous les vecteurs sélectionnés. Celui-ci est donné dans le cas du traitement d'un vecteur sélectionné l noté Y_{s_l} et du bit b_l à insérer.

Algorithme d'insertion :

- 1) $j = 1$
- 2) $Y_{s_l}^W = Y_{s_l}$
- 3) Trouver k ($j < k$) tel que la distance $d(y_j, y_k)$ soit minimale et non nulle
- 4) Permuter y_j avec y_k dans le vecteur $Y_{s_l}^W$ suivant le bit b_l à insérer :
 - $y_j < y_k$ si $b_l = 0$ (Classe 0)
 - $y_j > y_k$ si $b_l = 1$ (Classe 1)
- 5) Lire le bit inséré b_l^W du vecteur tatoué $Y_{s_l}^W$:
 - 5.1) Trouver la classe du dictionnaire $C_{ZM}^*(b_l^W)$ auquel $Y_{s_l}^W$ appartient
 - 5.2) Si $b_l^W \neq b_l$ alors $j = j + 1$ et retour en 2)
 Sinon fin => vecteur tatoué $Y_{s_l}^W$ par le bit b_l

4.4.6 Algorithme de détection

Nous proposons ici la description de l'algorithme pour l'opération de détection. Nous rappelons que celle-ci consiste à extraire la signature w . La détection se fait soit à partir du fichier de l'image compressée (attaquée ou non), soit à partir de l'image tatouée (attaquée $\tilde{I}^W$ ou non I^W). Dans le premier cas de figure, il suffit d'effectuer uniquement le décodage des index, on dispose alors des sous-bandes tatouées SB_k^W (attaquées ou non). Dans le second cas de figure, il suffit d'appliquer la transformée en ondelettes à l'image, on dispose alors des sous-bandes tatouées SB_k^W (attaquées ou non).

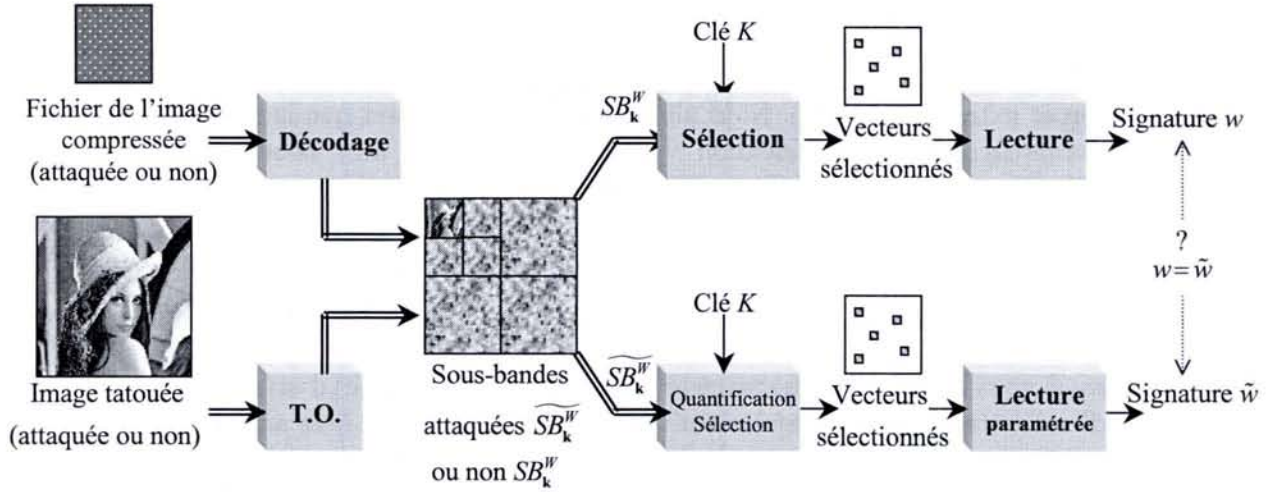


FIG. 4.7: Schéma des différents cas de figure lors de la détection de la signature.

Lors de la procédure de détection, soit on dispose des sous-bandes tatouées SB_k^W non attaquées (cas de figure idéal), soit on dispose des sous-bandes tatouées et attaquées $\widetilde{SB_k^W}$ (cas de figure réel). Dans le premier cas, l'algorithme de détection se ramène simplement à l'étape 5 de l'algorithme d'insertion. On effectue donc simplement la lecture du bit inséré b_i . Par contre, lorsque l'image est attaquée (modifiée), il est nécessaire de mettre en oeuvre une technique de détection plus efficace. C'est sur ce cas de figure que nos travaux se sont portés.

Nous nous positionnons dans le cas où nous disposons des sous-bandes tatouées attaquées $\widetilde{SB_k^W}$. Il est donc indispensable d'améliorer les deux étapes de la détection :

- la sélection : en effet, une attaque peut rendre non nulle l'énergie d'un vecteur alors qu'il avait été écarté de la procédure de tatouage (rappel : un vecteur nul ne peut pas être tatoué). La signature est donc faussée par l'ajout d'un nouveau bit ;
- la lecture : tout simplement, un vecteur dont les composantes ont été modifiées ne permettra peut-être plus de retrouver le bit b_i incrusté. La signature sera différente donc illisible.

Pour pallier ces problèmes lors de la détection, nous effectuons une requantification avant la phase de sélection et de lecture. Soit un vecteur l sélectionné tatoué et attaqué $\tilde{Y}_{sl}^{Wq} = (\tilde{y}_{s1}^{Wq}, \dots, \tilde{y}_{sn}^{Wq})$, pour des raisons de simplification d'écriture on pose $\tilde{y}_{si}^{Wq} = \tilde{y}_i$, d'où $\tilde{Y}_{sl}^{Wq} = (\tilde{y}_1, \dots, \tilde{y}_n)$. Nous pouvons écrire l'algorithme d'extraction suivant :

Algorithme d'extraction :

- 1) Requantification des vecteurs $\{\tilde{Y}^W\}$ pour s'affranchir des effets de l'attaque : $Q(\tilde{Y}^W) = \tilde{Y}^{Wq}$
- 2) Sélection des vecteurs parmi $\{\tilde{Y}^{Wq}\}$ à l'aide de la clé K : $S(\{\tilde{Y}^{Wq}\}, K) = \tilde{Y}_s^{Wq}$
- 3) Lecture du bit inséré $\tilde{b}_l^{Wq}$ d'un vecteur l sélectionné tatoué et attaqué $\tilde{Y}_{sl}^{Wq}$:
Trouver les composantes $(\tilde{y}_j^q, \tilde{y}_k^q)$ telles que :

$$d_{\min} = \min_{\tilde{Y}_{sl}^{Wq} = (\tilde{y}_1^q, \dots, \tilde{y}_n^q)} [d(\tilde{y}_j^q, \tilde{y}_k^q)] \quad \forall j < k \text{ et } d_{\min} \neq 0$$
- 4) Trouver la classe à laquelle appartient le vecteur $\tilde{Y}_{sl}^{Wq}$:
 Si $\tilde{y}_j^q < \tilde{y}_k^q$ Alors Bit $\tilde{b}_l = 0$ (Classe 0)
 Si $\tilde{y}_j^q > \tilde{y}_k^q$ Alors Bit $\tilde{b}_l = 1$ (Classe 1)

4.4.7 Conclusion

Nous disposons d'un schéma de tatouage original qui utilise un langage d'une complexité réduite (insertion et extraction rapide). Nous allons maintenant montrer que le fait d'effectuer conjointement Compression par Quantification Vectorielle Algébrique avec Zone Morte, et tatouage par substitution, nous permet d'améliorer la robustesse, face à différents types d'attaques (bruit additif gaussien, recompression, etc).

4.5 Robustesse face aux attaques de type bruit additif gaussien

Comme nous l'avons expliqué précédemment, les différents types d'attaques modifient les coefficients d'ondelettes et peuvent également éliminer les informations liées aux caractéristiques de l'algorithme de compression. Par exemple, lors d'un changement de format de stockage de type *RAW* : on ne dispose plus du nombre de niveaux de décompositions $\mathbf{J}$, et pour chaque sous-bande : du facteur de projection $\gamma_{\mathbf{k}}$, de la zone morte $\delta_{\mathbf{k}}$ et des caractéristiques du vecteur (taille n et forme). Dans ce cas, il est nécessaire de mettre en place des techniques d'identification de paramètres. Cependant, nous considérerons dans la suite de notre étude que ces paramètres sont connus, afin d'étudier plus précisément l'extraction de la signature d'une image attaquée.

Notre étude se porte sur l'évaluation de la robustesse du schéma de tatouage proposé face à des attaques de type bruit additif gaussien. Nous avons cherché à répondre à la question : "Quel est l'écart-type du bruit maximum acceptable pour une détection suffisamment fiable?". Nous présentons cette étude dans le cas d'une QVA sans zone morte puis avec zone morte.

Afin de simplifier les explications et les notations, notre étude s'est focalisée sur un vecteur tatoué et attaqué. Nous supposons dorénavant connaître le nombre de niveaux de décompositions $\mathbf{J}$ et pour une sous-bande $\widetilde{SB_{\mathbf{k}}^W}$: le facteur de projection γ , la zone morte δ et la taille n des vecteurs (ainsi que leurs formes). Nous rappelons que le vecteur de coefficients d'ondelettes quantifié Y_j d'une sous-bande $SB_{\mathbf{k}}$ appartient à un dictionnaire $\mathcal{C}$ régulier et fini, $Y_j = (y'_1\gamma, \dots, y'_n\gamma)$ où les composantes y'_i sont des entiers ($y'_i \in \mathbb{Z}$). Lorsque ce vecteur est tatoué (donc sélectionné : contient un bit b_i de la signature w) et attaqué, il est représenté par :

$$\tilde{Y}_{s_j}^W = (\tilde{y}_{s_1}^W, \dots, \tilde{y}_{s_n}^W) \quad (4.29)$$

Considérons la réalisation $\mathbf{b}_i$ d'un bruit gaussien de moyenne nulle ($m = 0$) et d'écart-type σ (remarque : $\mathbf{b}_i$ et une réalisation alors que b_j est un bit à insérer). On modélise l'attaque par $\mathbf{b}_i$ sur chaque composante du vecteur, il est donc possible d'écrire :

$$\tilde{Y}_{s_j}^W = (y_{s_1}'^W \gamma + \mathbf{b}_1, \dots, y_{s_n}'^W \gamma + \mathbf{b}_n) \quad (4.30)$$

pour des raisons de simplification d'écriture on pose $y_{s_i}'^W = \mathbf{y}'_i$, d'où :

$$\tilde{Y}_{s_j}^W = (\mathbf{y}'_1 \gamma + \mathbf{b}_1, \dots, \mathbf{y}'_n \gamma + \mathbf{b}_n) \quad (4.31)$$

Maintenant que nous avons modélisé les vecteurs attaqués, nous pouvons faire l'étude de l'écart-type du bruit maximum acceptable pour une détection suffisamment fiable avec ou sans zone morte. L'objectif étant de replacer les coefficients modifiés à leur valeur d'origine. Nous détaillons l'étude sans zone morte en utilisant une Quantification Scalaire.

4.5.1 Etude scalaire sans zone morte

Connaissant le facteur de projection γ , pour retrouver la valeur initiale d'une composante $y_{s_i}^W = \mathbf{y}'_i \gamma$ d'un vecteur non attaqué, à partir de la composante attaquée $\tilde{y}_{s_i}^W = \mathbf{y}'_i \gamma + \mathbf{b}_i$, il suffit

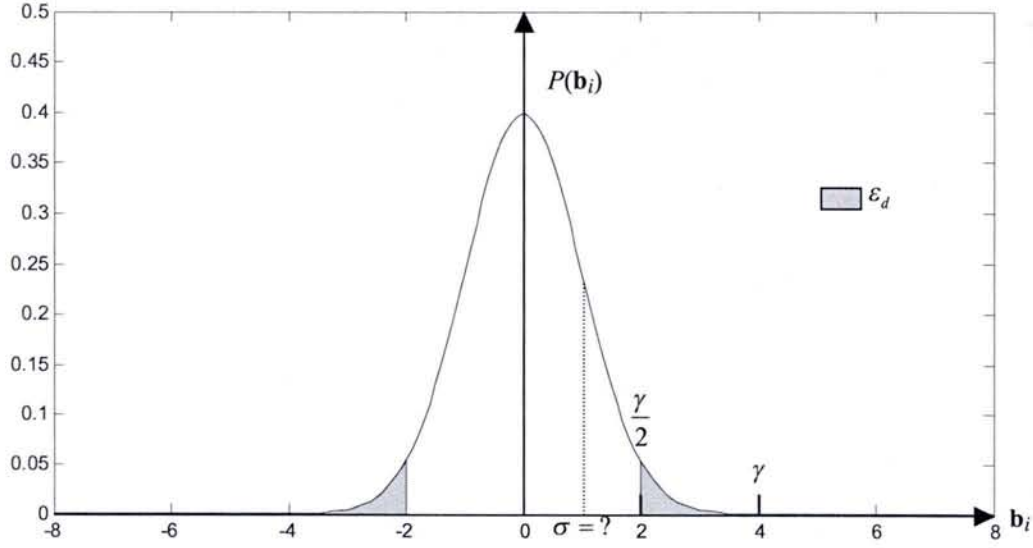


FIG. 4.8: Nous cherchons l'écart-type σ maximum du bruit $\mathbf{b}_i$ pour avoir $P(|\mathbf{b}_i| > \frac{\gamma}{2}) = \epsilon_d$.

de déterminer l'entier $\mathbf{y}'_i$. Pour cela on effectue une Quantification Scalaire (fonction $[\cdot]$) d'une composante $\hat{y}_{s_i}^W$ projetée par γ , soit :

$$\begin{aligned} \hat{y}_{s_i}^W &= \mathbf{y}'_i \gamma + \mathbf{b}_i \text{ on peut écrire } \frac{\hat{y}_{s_i}^W}{\gamma} = \mathbf{y}'_i + \frac{\mathbf{b}_i}{\gamma} \\ \text{QS : } \left[\frac{\hat{y}_{s_i}^W}{\gamma} \right] &= \left[\mathbf{y}'_i + \frac{\mathbf{b}_i}{\gamma} \right] \text{ or } \mathbf{y}'_i \in \mathbb{Z} \\ \text{d'où } \left[\frac{\hat{y}_{s_i}^W}{\gamma} \right] &= \mathbf{y}'_i + \left[\frac{\mathbf{b}_i}{\gamma} \right] \end{aligned} \quad (4.32)$$

Il est donc possible de retrouver la valeur initiale de la composante $y_{s_i}^W$ (avec une erreur de détection ϵ_d) si :

$$\left[\frac{\mathbf{b}_i}{\gamma} \right] = 0 \iff \left| \frac{\mathbf{b}_i}{\gamma} \right| < \frac{1}{2} \text{ soit } |\mathbf{b}_i| < \frac{\gamma}{2} \quad (4.33)$$

La figure 4.8 présente graphiquement le problème qui consiste à trouver l'écart-type σ du bruit additif gaussien en considérant une erreur acceptable de détection ϵ_d .

La réalisation $\mathbf{b}_i$ du bruit gaussien suit la statistique d'une loi normale $\mathcal{N}_{(0,1)}$, on note $x_{\mathbf{b}}$ la réalisation d'une variable aléatoire $\mathbf{x}_{\mathbf{b}}$ centrée réduite (de moyenne nulle et de variance unité) de ce bruit :

$$x_{\mathbf{b}} = \frac{\mathbf{b}_i - m}{\sigma} = \frac{\mathbf{b}_i}{\sigma} \quad (4.34)$$

On peut donc écrire :

$$P\left(|\mathbf{b}_i| > \frac{\gamma}{2}\right) = f_{\mathbf{x}_{\mathbf{b}}} \left(|x_{\mathbf{b}} \sigma| > \frac{\gamma}{2}\right) = f_{\mathbf{x}_{\mathbf{b}}} \left(|x_{\mathbf{b}}| > \frac{\gamma}{2\sigma}\right) \quad (4.35)$$

Du fait de la symétrie de la densité de la loi normale :

$$\begin{aligned} f_{\mathbf{x}_{\mathbf{b}}} \left(|x_{\mathbf{b}}| > \frac{\gamma}{2\sigma}\right) &= f_{\mathbf{x}_{\mathbf{b}}} \left(-x_{\mathbf{b}} > \frac{\gamma}{2\sigma}\right) + f_{\mathbf{x}_{\mathbf{b}}} \left(x_{\mathbf{b}} > \frac{\gamma}{2\sigma}\right) \\ \text{d'où } f_{\mathbf{x}_{\mathbf{b}}} \left(|x_{\mathbf{b}}| > \frac{\gamma}{2\sigma}\right) &= 2f_{\mathbf{x}_{\mathbf{b}}} \left(x_{\mathbf{b}} < -\frac{\gamma}{2\sigma}\right) \end{aligned} \quad (4.36)$$

Or si on veut retrouver la valeur quantifiée initiale d'un coefficient d'ondelettes avec une erreur maximum ϵ_d , il faut que :

$$P\left(|b_i| > \frac{\gamma}{2}\right) = \epsilon_d \iff f_{\mathbf{x}_b}\left(x_b < -\frac{\gamma}{2\sigma}\right) = \frac{\epsilon_d}{2} \quad (4.37)$$

On obtient ainsi une estimation de l'écart-type σ maximum du bruit en consultant la table standard de la loi normale $\mathcal{N}_{(0,1)}$.

Nous pouvons maintenant généraliser ce résultat pour l'ensemble des composantes i du vecteur de dimension n , en considérant chaque variable aléatoire indépendante et identiquement distribuée (i.i.d.) :

$$P\left(\bigcup_{i=1}^n \left(|b_i| > \frac{\gamma}{2}\right)\right) = \sum_{i=1}^n P\left(|b_i| > \frac{\gamma}{2}\right) = nP\left(|b_i| > \frac{\gamma}{2}\right) \quad (4.38)$$

d'où

$$P\left(\bigcup_{i=1}^n \left(|b_i| > \frac{\gamma}{2}\right)\right) = \epsilon_d \iff f_{\mathbf{x}_b}\left(x_b < -\frac{\gamma}{2\sigma}\right) = \frac{\epsilon_d}{2n} \quad (4.39)$$

Nous pouvons maintenant supposer que cette erreur de détection peut être faite sur chacun des vecteurs tatoués lors de la lecture de la signature w . Nous pouvons ainsi définir la probabilité de l'erreur de lecture de la signature w par :

$$P(\tilde{w} \neq w) = M_w P\left(\bigcup_{i=1}^n \left(|b_i| > \frac{\gamma}{2}\right)\right) \quad (4.40)$$

où M_w est le nombre de bits de la signature w , d'où :

$$P(\tilde{w} \neq w) = \epsilon_w \iff f_{\mathbf{x}_b}\left(x_b < -\frac{\gamma}{2\sigma}\right) = \frac{\epsilon_w}{2nM_w} \quad (4.41)$$

Pour illustrer l'équation 4.41, si l'on prend $\epsilon_w = 0,05$, $n = 8$ et $M_w = 60$, la valeur correspondante à $\frac{\epsilon_w}{2nM_w} = 0,000833$, de la loi normale $\mathcal{N}_{(0,1)}$, est $-3,8807$, alors l'écart-type σ du bruit doit être inférieur ou égal à la valeur σ_{ϵ_w} qui vérifie :

$$-3,8807 = -\frac{\gamma}{2\sigma_{\epsilon_w}} \quad (4.42)$$

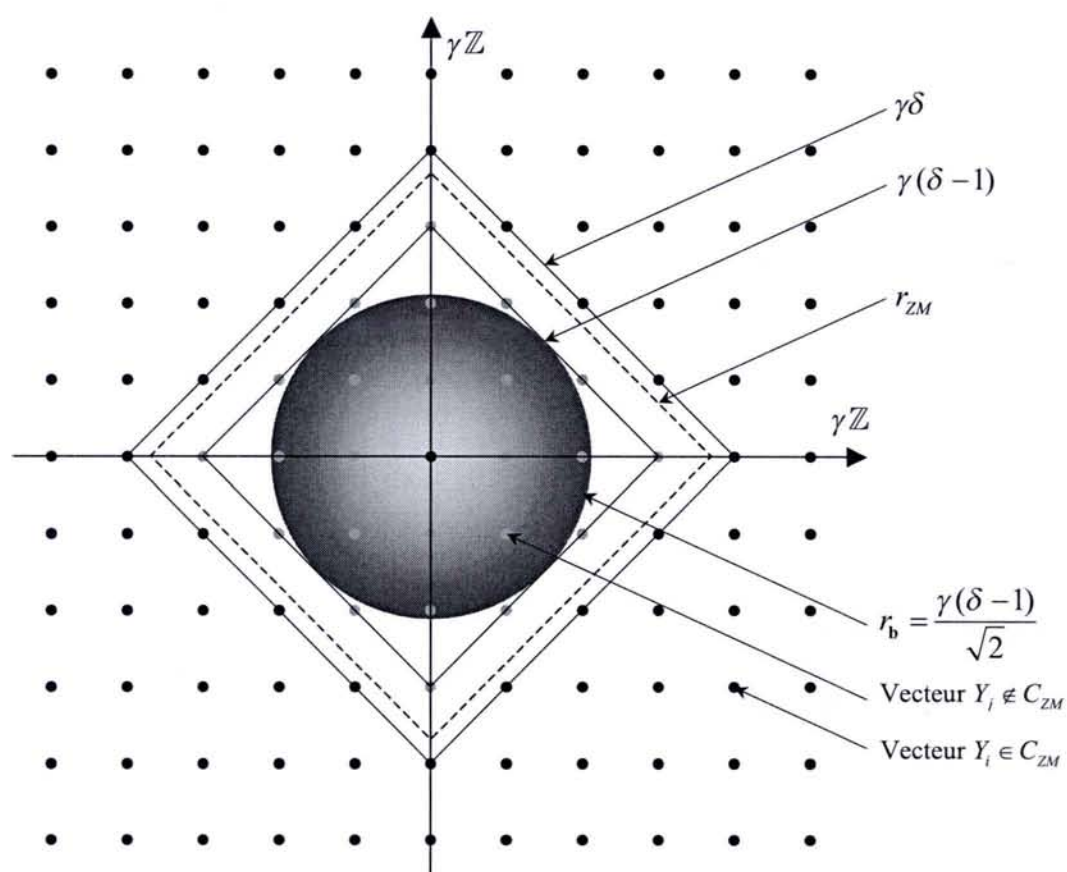
Enfin si $\gamma = 7$, il est donc possible de retrouver la signature avec un risque d'erreur de 5% pour un bruit gaussien d'écart-type maximum $\sigma_{\epsilon_w} = 0,902$.

4.5.2 Etude vectorielle avec zone morte

L'intérêt de la zone morte vectorielle est d'augmenter la zone de quantification autour du vecteur nul. Ainsi, il est également possible d'augmenter l'écart-type du bruit tout en préservant la même robustesse. En particulier, la création de cette zone morte vectorielle permet d'améliorer la sélection des vecteurs tatoués. On note $\mathbf{b}$ la réalisation d'un vecteur aléatoire composé des réalisations $\mathbf{b}_i$ de variables aléatoires indépendantes et identiquement distribuées :

$$\mathbf{b} = \{\mathbf{b}_1, \dots, \mathbf{b}_i, \dots, \mathbf{b}_n\} \quad (4.43)$$

Nous allons maintenant chercher à estimer l'écart-type maximum σ_{ϵ_v} d'un bruit gaussien pour avoir un risque d'erreur de détection ϵ_v en considérant ce bruit par rapport à la norme des vecteurs. La figure 4.9 montre la région où le bruit gaussien est considéré comme acceptable (i.e. dont la

FIG. 4.9: Visualisation du bruit gaussien sur un réseau $\mathbb{Z}^2$.

requantification le maintient inévitablement dans la zone morte) avec un risque d'erreur de détection ϵ_v . Notre objectif est de connaître σ_{ϵ_v} pour avoir :

$$P(\|\mathbf{b}\|_2 > r_b) \leq \epsilon_v \quad \text{avec} \quad r_b = \frac{\gamma(\delta - 1)}{\sqrt{2}} \quad (4.44)$$

Dans le chapitre 2, nous avons établi la fonction de répartition pour une source laplacienne ($\alpha = 1$). Or, ici, nous cherchons à connaître la fonction de répartition d'une source gaussienne ($\alpha = 2$) (modèle du bruit gaussien). La forme générale de la densité de probabilité du rayon (loi du rayon) est donnée par la formule [74] :

$$f_{\mathbf{X}}(r) = \frac{\alpha b^n}{\Gamma(\frac{n}{\alpha})} r^{n-1} \exp(-(br)^\alpha) \quad \text{avec} \quad b = \frac{1}{\sigma} \sqrt{\frac{\Gamma(\frac{3}{\alpha})}{\Gamma(\frac{1}{\alpha})}} \quad (4.45)$$

Dans notre cas $\alpha = 2$ (bruit gaussien), d'où [46] :

$$f_{\mathbf{X}}(r) = \left(2^{\frac{n}{2}-1} \Gamma\left(\frac{n}{2}\right) \sigma^n\right)^{-1} r^{n-1} \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (4.46)$$

où $r = \|\mathbf{b}\|_2$ correspond au rayon du bruit gaussien. Donc, si l'on cherche à connaître $P(\|\mathbf{b}\|_2 > r_b)$, il suffit de calculer dans un premier temps $P(\|\mathbf{b}\|_2 \leq r_b)$. En effet, par définition $P(\|\mathbf{b}\|_2 > r_b) = 1 - P(\|\mathbf{b}\|_2 \leq r_b)$.

Nous présentons en annexe la démonstration du calcul de la fonction de répartition $F(r)$ dans le cas d'une source laplacienne ($\alpha = 1$) et gaussienne ($\alpha = 2$) pour $n \in \mathbb{N}$:

$$F(r) = 1 - \exp(-(br)^\alpha) \sum_{k=0}^{\frac{n}{\alpha}-1} \frac{(br)^{\alpha k}}{\Gamma(k+1)} \quad \forall n, k \in \mathbb{N} \quad (4.47)$$

$$\text{avec } \alpha \in \{1, 2\} \quad , \quad b = \frac{1}{\sigma} \sqrt{\frac{\Gamma(\frac{3}{\alpha})}{\Gamma(\frac{1}{\alpha})}} \quad \text{et } \sigma \text{ l'écart-type} \quad (4.48)$$

Il est donc possible d'écrire $P(\|\mathbf{b}\|_2 \leq r_b)$ pour $\alpha = 2$ à l'aide de la formule 4.47, d'où :

$$P(\|\mathbf{b}\|_2 \leq r_b) = F(r_b) = 1 - \exp\left(-\frac{r_b^2}{2\sigma^2}\right) \sum_{k=0}^{\frac{n}{2}-1} \frac{\left(\frac{r_b^2}{2\sigma^2}\right)^k}{\Gamma(k+1)} \quad \forall n, k \in \mathbb{N} \quad (4.49)$$

Il est donc maintenant possible de calculer les valeurs de $P(\|\mathbf{b}\|_2 \leq r_b)$, et donc les valeurs de $P(\|\mathbf{b}\|_2 > r_b)$ pour différents σ , qui peuvent être pré-calculées et stockées dans une table. Nous pourrions ainsi trouver rapidement l'écart-type maximum σ_{ϵ_v} pour une erreur de détection ϵ_v acceptable. Nous disposons ainsi d'une technique d'évaluation de l'écart-type maximum acceptable σ_{ϵ_v} d'un bruit gaussien, pour avoir une erreur de détection ϵ_v telle que :

$$P(\|\mathbf{b}\|_2 > r_b) = \frac{\epsilon_v}{2nM_w} \quad (4.50)$$

Pour illustrer l'équation 4.50, si l'on prend $\epsilon_v = 0,05$, $n = 8$ et $M_w = 60$, alors $\frac{\epsilon_v}{2nM_w} = 0,9999479$. Enfin, si $\gamma = 7$ et $r_b = 3$, alors l'écart-type σ du bruit doit être inférieur ou égal à la valeur $\sigma_{\epsilon_v} = 1,6180$. Il est donc possible de retrouver la signature avec un risque d'erreur de 5% pour un bruit gaussien d'écart-type maximum $\sigma_{\epsilon_v} = 1,6180$. Nous constatons que celui-ci est plus important que dans le cas de l'étude sans zone morte. Ainsi, pour un même écart-type, la robustesse du tatouage est meilleure avec l'utilisation d'une zone morte vectorielle.

M_w	0 bits	30 bits	60 bits	90 bits	150 bits
$PSNR$ (dB)	30,5000	30,4397	30,3982	30,3567	30,2768
$\Delta PSNR$ (dB)	0	0,06	0,1	0,14	0,22
IMAGE N° :	4.1	-	4.2	4.3	4.4

TAB. 4.1: $PSNR$ de l'image *Lena* sans zone morte pour différentes longueurs de message.

M_w	0 bits	30 bits	60 bits	90 bits	120 bits
$PSNR$ (dB)	30,9044	30,8729	30,8172	30,7860	30,7572
$\Delta PSNR$ (dB)	0	0,03	0,09	0,12	0,15
IMAGE N° :	4.5	-	4.6	4.7	4.8

TAB. 4.2: $PSNR$ de l'image *Lena* avec zone morte pour différentes longueurs de message.

4.6 Résultats

Les simulations ont été effectuées sur l'image *Lena* pour un débit cible de 0,125 bpp (i.e. taux de compression de 64 : 1). Nous avons utilisé une transformée en ondelettes discrète sur $\mathbf{J} = 5$ niveaux de décomposition. Nous avons choisi d'insérer la signature au sein des sous-bandes du niveau de décomposition $\mathbf{j} = 3$. Nous comparerons à chaque fois le cas sans zone morte et avec zone morte. Le nombre de bits M_w de la signature w est réparti à chaque fois sur les 3 sous-bandes du niveau $\mathbf{j} = 3$ (si $M_w = 60$ bits, alors $M_{w\mathbf{k}} = M_w/3 = 20$ bits par sous-bandes $\mathbf{k}$).

4.6.1 Imperceptibilité

Nous présentons à travers ces simulations, les performances de notre schéma de tatouage en terme d'imperceptibilité (évaluation subjective) par rapport à la capacité du message. De plus nous évaluerons la dégradation de la qualité en terme de $PSNR$ (évaluation objective).

4.6.1.1 Sans zone morte

Le tableau 4.1 présente les simulations sans zone morte sur l'image *Lena* pour différentes longueurs M_w du message inséré. Nous pouvons constater une diminution de seulement $\Delta PSNR = 0,1$ dB dans le cas d'un message $M_w = 60$ bits par rapport à l'image non tatouée, et aucune dégradation visuelle en comparant les images 4.1 avec 4.2. Par contre, à partir d'un message de $M_w = 150$ bits, nous visualisons une faible dégradation visuelle en particulier sur les contours.

4.6.1.2 Avec zone morte

Le tableau 4.2 présente les simulations avec zone morte sur l'image *Lena* pour différentes longueurs M_w du message inséré. Nous pouvons constater que la dégradation de la qualité ($\Delta PSNR$) est légèrement plus faible que l'approche sans zone morte. Nous observons par exemple, avec un message de 120 bits avec zone morte, la même dégradation (0,15 dB) que pour un message de seulement 90 bits sans zone morte. De plus, la comparaison des différentes images 4.6, 4.7 et 4.8 avec l'image non tatouée 4.5, nous permet de constater le faible impact visuel de l'insertion d'une signature à l'aide de notre schéma de compression et de tatouage conjoints.



IMAGE 4.1 : Image *Lena* à 0,125 bpp par QVA sans ZM sans signature.



IMAGE 4.2 : Image *Lena* à 0,125 bpp par QVA sans ZM avec $M_w = 60$ bits.



IMAGE 4.3 : Image *Lena* à 0,125 bpp par QVA sans ZM avec $M_w = 90$ bits.



IMAGE 4.4 : Image *Lena* à 0,125 bpp par QVA sans ZM avec $M_w = 150$ bits.



IMAGE 4.5 : Image *Lena* à 0,125 bpp par QVA avec ZM sans signature.



IMAGE 4.6 : Image *Lena* à 0,125 bpp par QVA avec ZM avec $M_w = 60$ bits.



IMAGE 4.7 : Image *Lena* à 0,125 bpp par QVA avec ZM avec $M_w = 90$ bits.



IMAGE 4.8 : Image *Lena* à 0,125 bpp par QVA avec ZM avec $M_w = 120$ bits.

Ecart-type du bruit gaussien sur l'image : $\sigma_{Image} =$						7	8	9	10
Sous-bande	n	γ	δ	ϵ_w	σ_{ϵ_w}	σ_{SB_k}	σ_{SB_k}	σ_{SB_k}	σ_{SB_k}
SB_6	8	8,50	0,0	0,05	1,1790	0,8828	0,9840	1,1496	1,2681
SB_7	8	7,48	0,0	0,05	1,0375	0,8810	1,0017	1,1310	1,2593
SB_8	8	7,44	0,0	0,05	1,0320	0,9101	1,0305	1,2019	1,2848
% d'erreur	-	-	-	-	-	0%	0%	1,66%	3,33%
IMAGE N° :	-	-	-	-	-	-	4.10	4.11	-

TAB. 4.3: Caractéristiques des sous-bandes 6, 7 et 8 du niveau $j=3$ sans zone morte

Qualité <i>JPEG</i>	15	20	supérieure
% erreur	8,33%	0%	0%
IMAGE N° :	4.13	4.14	-

TAB. 4.4: Pourcentage d'erreur de lecture de l'image *Lena* sans zone morte avec une recompression *JPEG*.

4.6.2 Robustesse face aux attaques

Nous avons appliqué à l'image quantifiée et tatouée trois principaux types d'attaques issues en partie du programme "*Stirmark*" [70] (bruit additif gaussien, re-compression *JPEG*, filtre médian), dans le cas sans zone morte puis avec zone morte. La longueur M_w du message w est choisie à 60 bits.

4.6.2.1 Robustesse de la QVA sans zone morte

4.6.2.1.1 Bruit additif gaussien : A l'aide de l'équation 4.41, et en tenant compte des paramètres de compression de l'image *Lena* à 0,125 bpp, il est possible de trouver l'écart-type maximum théorique σ_{ϵ_w} pour chacune des sous-bandes (voir tableau 4.3). Afin de comparer et de valider l'étude théorique, nous avons relevé l'estimation de l'écart-type pratique σ_{SB_k} du bruit engendré sur chacune des sous-bandes SB_k . Nous pouvons constater que : tant que le bruit gaussien d'écart-type maximum $\sigma_{\epsilon_w} \leq \sigma_{SB_k}$ pour chacune des sous-bandes, alors le pourcentage d'erreur reste nul. Sinon on observe un pourcentage d'erreur de lecture. Nous présentons à travers les images 4.10 et 4.11, la dégradation visuelle que ce type d'attaque engendre.

4.6.2.1.2 Re-compression *JPEG* : Nous présentons dans le tableau 4.4 les résultats de la robustesse face à une attaque de type recompression *JPEG* pour différentes qualités sur l'image *Lena* (pour une longueur de message $M_w = 60$ bits). Nous constatons que : même avec une dégradation visuelle importante avec une compression *JPEG* de qualité 20, la signature est parfaitement reconstruite. L'observation de l'image 4.14 montre la dégradation visuelle.

4.6.2.1.3 Filtre médian : Le filtre médian est une attaque à laquelle il est difficile de faire face. Cependant, le message n'a pas été modifié malgré une attaque de type filtre médian de forme 3×3 . Nous présentons dans le tableau 4.5 les résultats de la robustesse face à une attaque de ce type (pour une longueur de message $M_w = 60$ bits). Nous pouvons visualiser la dégradation en comparant l'image 4.15 avec l'image 4.16.

filtre médian	3×3
% erreur	0%
IMAGE N° :	4.16

TAB. 4.5: Pourcentage d'erreur de lecture de l'image *Lena* sans zone morte pour une attaque de type filtre médian.



IMAGE 4.9 : Image *Lena* à 0,125 bpp par QVA sans ZM avec $M_w = 60$ bits non attaquée.



IMAGE 4.10 : Image *Lena* à 0,125 bpp par QVA sans ZM avec $M_w = 60$ bits, avec un bruit gaussien additif d'écart-type $\sigma_{Image} = 8$.



IMAGE 4.11 : Image *Lena* à 0,125 bpp par QVA sans ZM avec $M_w = 60$ bits, avec un bruit gaussien additif d'écart-type $\sigma_{Image} = 9$.



IMAGE 4.12 : Image *Lena* à 0,125 bpp par QVA sans ZM avec $M_w = 60$ bits non attaquée.



IMAGE 4.13 : Image *Lena* à 0,125 bpp par QVA sans ZM avec $M_w = 60$ bits, attaquée par recompression *JPEG* de qualité 15.



IMAGE 4.14 : Image *Lena* à 0,125 bpp par QVA sans ZM avec $M_w = 60$ bits, attaquée par recompression *JPEG* de qualité 20.



IMAGE 4.15 : Image *Lena* à 0,125 bpp par QVA sans ZM avec $M_w = 60$ bits non attaquée.

Ecart-type du bruit gaussien sur l'image : $\sigma_{Image} =$						9	10	11
Sous-bande	n	γ	δ	ϵ_w	σ_{ϵ_w}	σ_{SB_k}	σ_{SB_k}	σ_{SB_k}
SB_6	8	6,6531	3,6909	0,05	2,3930	2,3036	2,3593	2,4092
SB_7	8	5,1717	3,6909	0,05	1,8600	1,9024	1,9973	2,0717
SB_8	8	6,1577	2,7294	0,05	1,4760	1,8145	1,9376	2,01641
% d'erreur	-	-	-	-	-	0%	0%	1,66%
IMAGE N° :	-	-	-	-	-	4.18	4.19	4.20

TAB. 4.6: Caractéristiques des sous-bandes 6, 7 et 8 du niveau $j=3$ avec zone morte.IMAGE 4.16 : Image *Lena* à 0,125 bpp par QVA sans ZM avec $M_w = 60$ bits, attaquée par un filtre médian de forme 3×3 .

4.6.2.2 Robustesse de la QVA avec zone morte

4.6.2.2.1 Bruit additif gaussien : A l'aide de l'équation 4.49, et en tenant compte des paramètres de compression de l'image *Lena* à 0,125 bpp, il est possible de trouver l'écart-type maximum théorique σ_{ϵ_w} pour chacune des sous-bandes (voir tableau 4.6). Tant que le bruit gaussien d'écart-type maximum $\sigma_{\epsilon_w} \leq \sigma_{SB_k}$ pour chacune des sous-bandes, alors le pourcentage d'erreur reste nul. Nous pouvons constater une nette amélioration de la robustesse de notre approche avec zone morte. En effet, l'extraction de la signature s'effectue sans erreur, malgré un bruit gaussien d'écart-type $\sigma_{Image} = 10$ ajouté à l'image tatouée. Nous rappelons que sans zone morte, le bruit gaussien maximum ajouté à l'image tatouée (sans erreur de lecture) était d'écart-type $\sigma_{Image} = 8$. Nous présentons la dégradation visuelle de ce type d'attaques sur les images 4.18, 4.19 et 4.20.

Qualité <i>JPEG</i>	15	20	supérieure
% erreur	6,66%	0%	0%
IMAGE N° :	4.22	4.23	-

TAB. 4.7: Pourcentage d'erreur de lecture de l'image *Lena* avec zone morte avec une recompression *JPEG*.

filtre médian	3×3
% erreur	0%
IMAGE N° :	4.25

TAB. 4.8: Pourcentage d'erreur de lecture de l'image *Lena* avec zone morte pour une attaque de type filtre médian.

4.6.2.2.2 Re-compression *JPEG* : Nous présentons dans le tableau 4.7 les résultats de la robustesse face à une attaque de type re-compression *JPEG* pour différentes qualités sur l'image *Lena* (pour une longueur de message $M_w = 60$ bits). Nous pouvons constater, pour une qualité *JPEG* de 15, un plus faible pourcentage d'erreur avec zone morte (6,66%) que sans zone morte (8,88%). La dégradation visuelle peut être comparée entre l'image 4.21 et les images attaquées 4.22 et 4.23.

4.6.2.2.3 Filtre médian : Notre tatouage reste robuste malgré une attaque de type filtrage médian de forme 3×3 . Nous présentons dans le tableau 4.8 les résultats de la robustesse face à une attaque de ce type sur l'image *Lena* (pour une longueur de message $M_w = 60$ bits). Nous pouvons visualiser la dégradation visuelle sur l'image 4.25.



IMAGE 4.17 : Image *Lena* à 0,125 bpp par QVA avec ZM avec $M_w = 60$ bits non attaquée.



IMAGE 4.18 : Image *Lena* à 0,125 bpp par QVA avec ZM avec $M_w = 60$ bits, avec un bruit gaussien additif d'écart-type $\sigma_{Image} = 9$.



IMAGE 4.19 : Image *Lena* à 0,125 bpp par QVA avec ZM avec $M_w = 60$ bits, avec un bruit gaussien additif d'écart-type $\sigma_{Image} = 10$.



IMAGE 4.20 : Image *Lena* à 0,125 bpp par QVA avec ZM avec $M_w = 60$ bits, avec un bruit gaussien additif d'écart-type $\sigma_{Image} = 11$.



IMAGE 4.21 : Image *Lena* à 0,125 bpp par QVA avec ZM avec $M_w = 60$ bits non attaquée.



IMAGE 4.22 : Image *Lena* à 0,125 bpp par QVA avec ZM avec $M_w = 60$ bits, attaquée par recompression *JPEG* de qualité 15.



IMAGE 4.23 : Image *Lena* à 0,125 bpp par QVA avec ZM avec $M_w = 60$ bits, attaquée par recompression *JPEG* de qualité 20.



IMAGE 4.24 : Image *Lena* à 0,125 bpp par QVA avec ZM avec $M_w = 60$ bits non attaquée.



IMAGE 4.25 : Image *Lena* à 0,125 bpp par QVA avec ZM avec $M_w = 60$ bits, attaquée par un filtre médian de forme 3×3 .

4.7 Conclusion du chapitre

Compte tenu du développement des moyens de transmission et de stockage actuels, la duplication et l'échange d'informations ne cessent d'augmenter. Afin de protéger la propriété intellectuelle de ces données, des solutions ont été envisagées. C'est dans ce cadre que nous avons orienté nos travaux. Et cela, afin de répondre à deux besoins fondamentaux. Le premier consiste à comprimer les données afin de réduire la quantité d'information à transmettre ou à stocker. Le second concerne la protection des données en leur ajoutant une information (*copyright*). Pour répondre à ces deux aspects (pourtant opposés), nous avons proposé un schéma de compression et de tatouage d'images conjoints.

Ce schéma est composé de deux méthodes : la Quantification Vectorielle Algébrique avec Zone Morte, et un schéma de tatouage par substitution dans le domaine multirésolution. Ces deux méthodes associées ont permis d'obtenir des résultats encourageants quant à l'imperceptibilité et la robustesse de la marque. En particulier, Nous avons montré à travers une étude théorique les limites et les bonnes performances de cette approche dans le cas d'une attaque de type bruit additif gaussien. De plus, d'autres attaques de type recompression ou filtrage (médian) ont été envisagées. Celles-ci nous ont également permis de valider les performances de notre méthode et ces limites.

Chapitre 5

Conclusion générale

Avec la numérisation des supports de communication (images, sons, vidéos) et l'émergence des moyens de communication (réseaux), la compression des données est devenue indispensable. Celle-ci permet de transmettre ou de stocker cette quantité de données multimédia toujours plus importante, compte tenu des limites technologiques et des enjeux économiques. Les méthodes de compression ont fait l'objet de nombreux travaux de recherche, qui ont abouti à différents standards. Actuellement, la majorité des images sont compressées avec le standard *JPEG*. Cependant, les nouveaux besoins liés aux nombreuses applications ont amené la communauté scientifique à élaborer un nouveau standard : *JPEG2000*. A travers ce mémoire, nous avons élaboré et mis en oeuvre une méthode de compression d'images permettant d'améliorer les performances à bas débits, en terme de compromis taux de compression/ qualité en conservant une faible complexité. D'autre part, avec le développement des réseaux, les échanges de productions artistiques (photos, musiques, films) deviennent difficilement contrôlables. Il est donc nécessaire de mettre en place des méthodes permettant d'identifier les auteurs de ces productions (*Copyright*). Nous avons ainsi adapté une approche hybride de compression et de tatouage d'images, permettant d'insérer une signature au sein même des données compressées.

Nos travaux se sont déployés autour d'un schéma de compression d'images, utilisant une transformée en ondelettes, associée à une Quantification Vectorielle Algébrique. L'originalité de notre méthode réside dans la définition d'une Zone Morte Vectorielle afin d'améliorer la qualité de l'image quantifiée, et d'améliorer la robustesse de la signature d'une image tatouée face à différentes attaques.

Dans un premier temps, nous avons fait l'étude théorique et présenté l'intérêt que la Quantification Vectorielle Algébrique avec Zone Morte (QVAZM) pouvait apporter en terme de compromis TC/ Qualité. En effet, il est toujours préférable de quantifier les coefficients d'ondelettes significatifs (correspondant aux contours, ou aux zones texturées de l'image) plutôt que les coefficients de faible amplitude (correspondant à des zones homogènes de l'image). La création d'une Zone Morte engendre une redondance importante de valeurs nulles. Cette caractéristique a été exploitée avec l'adaptation d'un codeur de type *Stack-Run*. Notre approche de type intra-bandes nous a permis de mettre en place une méthode de répartition des débits binaires R_k dans chaque sous-bande SB_k de la transformée, en optimisant les deux paramètres de notre méthode de compression, à savoir le facteur de projection γ_k pour la QVA, et le seuil r_{ZMk} de la Zone Morte. Nous avons ainsi mis en oeuvre un algorithme qui permet de trouver le couple (γ_k^*, r_{ZMk}^*) optimal pour atteindre le débit cible R_k fixé pour une sous-bande SB_k . Les résultats expérimentaux ont montré un gain allant jusqu'à 0,6 dB par rapport au standard *JPEG2000* pour des très bas débits (sur l'image *Lena*). De plus, les résultats visuels montrent que certains détails sont mieux préservés (en particulier sur les zones texturées). Afin de réduire la complexité de l'algorithme, nous avons fait l'étude théorique du

modèle de débit de la QVAZM. Cette étude est basée sur un modèle de débit entropique par code préfixe pour une source de type Laplacienne (généralement utilisée pour estimer la statistique des coefficients d'ondelettes). Celui-ci, associé à un modèle de distorsion permet d'accélérer la recherche du meilleur couple $(\gamma_{\mathbf{k}}^*, r_{ZM\mathbf{k}}^*)$.

Dans un second temps, nous avons proposé l'amélioration d'un schéma de compression et de tatouage d'images conjoints. Nous avons utilisé une approche qui consiste à coupler ces deux processus malgré leurs oppositions en terme d'objectif final (compression : supprimer l'information imperceptible, tatouage : insérer une information imperceptible). La signature est insérée directement sur les vecteurs quantifiés de certaines sous-bandes. Nous avons montré que l'utilisation de la Quantification Vectorielle Algébrique avec Zone Morte permet d'améliorer la robustesse du tatouage face à des attaques de type bruit additif Gaussien.

Dans le futur, le modèle de débit que nous avons établi et l'étude du modèle de distorsion qui est en cours, devraient permettre de réduire de manière significative la complexité de notre algorithme de recherche du couple $(\gamma_{\mathbf{k}}^*, r_{ZM\mathbf{k}}^*)$. D'autre part, la Zone Morte apparaît naturellement adaptée à des applications de compression d'images et de débruitage conjoints. Cette approche nous laisse donc envisager une perspective particulièrement intéressante.

Annexe A

Fonction de répartition $F(r)$

A.1 Problématique

Nous présentons à travers cette annexe le calcul de la fonction de répartition des vecteurs $F(r)$ d'une source gaussienne et laplacienne en fonction de leur module. Celle-ci est utilisée en particulier lors du calcul du modèle de débit de la QVAZM (Chapitre 2) et lors du calcul de l'écart-type maximum du bruit gaussien additif pour une erreur de détection acceptable (Chapitre 4).

Nous rappelons que dans le cas d'une source de statistique gaussienne généralisée i.i.d., de paramètre de forme α , l'expression de la distribution d'un vecteur de dimension n est donnée par :

$$f_X(x_1, \dots, x_n) = a \exp \left(- \sum_{i=1}^n |bx_i|^\alpha \right)$$

où les paramètres a et b ont pour expression :

$$a = \left(\frac{b\alpha}{2\Gamma(\frac{1}{\alpha})} \right)^n \quad \text{et} \quad b = \frac{1}{\alpha} \sqrt{\frac{\Gamma(\frac{3}{\alpha})}{\Gamma(\frac{1}{\alpha})}}$$

Les vecteurs X ayant la même probabilité d'apparition sont sur la même isosurface et vérifient :

$$\sum_{i=1}^n |bx_i|^\alpha = r^\alpha$$

A partir de ces expressions a été établie [74] la loi de répartition des vecteurs (loi du rayon) d'une source gaussienne généralisée en fonction de leur module :

$$f(r) = K_n r^{n-1} \exp(-(br)^\alpha) \quad (\text{A.1})$$

avec

$$K_n = \frac{\alpha b^n}{\Gamma(\frac{n}{\alpha})} \quad \text{et} \quad b = \frac{1}{\alpha} \sqrt{\frac{\Gamma(\frac{3}{\alpha})}{\Gamma(\frac{1}{\alpha})}}$$

Nous allons maintenant poursuivre cette étude pour calculer la Fonction de répartition des vecteurs $F(r)$ d'une source gaussienne et laplacienne en fonction de leur module.

A.2 Démonstration

A partir de la formule A.1, pour calculer la Fonction de répartition des vecteurs $F(r)$, il suffit de calculer :

$$F(r) = \int_0^{r_{ZM}} f(r) dr$$

soit

$$\int_0^{r_{ZM}} f(r) dr = K_n \int_0^{r_{ZM}} r^{n-1} \exp(-(br)^\alpha) dr$$

En effectuant le changement de variable $u = (br)^\alpha$, on a :

$$r = \frac{u^{\frac{1}{\alpha}}}{b} \quad \text{d'où} \quad dr = \frac{u^{\left(\frac{1}{\alpha}-1\right)}}{\alpha b} du$$

d'où

$$\int_0^{r_{ZM}} f(r) dr = \frac{K_n}{\alpha b^n} \int_0^{(br_{ZM})^\alpha} u^{\left(\frac{n}{\alpha}-1\right)} \exp(-u) du$$

On pose $m = \frac{n}{\alpha}$ et $U = (br_{ZM})^\alpha$ et on cherche à calculer l'intégrale :

$$I = \int_0^U u^{(m-1)} \exp(-u) du$$

Par une succession de $k = m - 1$ intégrations par partie :

$$I = - \sum_{k=1}^{m-1} \left[\frac{(m-1)!}{(m-k)!} u^{m-k} \exp(-u) \right]_0^U + \int_0^U (m-1)! \exp(-u) du$$

or rappelons que $\Gamma(\beta) = (\beta-1)! \quad \forall \beta \in \mathbb{N}$:

$$I = - \sum_{k=1}^{m-1} \frac{\Gamma(m)}{\Gamma(m-k+1)} U^{m-k} \exp(-U) + \Gamma(m) (-\exp(-U) + 1)$$

d'où :

$$\int_0^{r_{ZM}} f(r) dr = 1 - \left(\sum_{k=1}^{m-1} \frac{(br_{ZM})^{\alpha(m-k)}}{\Gamma(m-k+1)} + 1 \right) \exp(-(br_{ZM})^\alpha)$$

On pose $K = m - k$, d'où :

$$\int_0^{r_{ZM}} f(r) dr = 1 - \left(\sum_{K=0}^{m-1} \frac{(br_{ZM})^{\alpha K}}{\Gamma(K+1)} \right) \exp(-(br_{ZM})^\alpha)$$

La fonction de répartition des vecteurs $F(r)$ d'une source gaussienne et laplacienne en fonction de leur module est donc donnée par :

$$F(r) = 1 - \left(\sum_{k=0}^{\frac{n}{\alpha}-1} \frac{(br)^\alpha}{\Gamma(k+1)} \right) \exp(-(br)^\alpha) \quad \forall n, k \in \mathbb{N} \text{ et } \alpha \in \{1, 2\}$$

avec

$$b = \frac{1}{\alpha} \sqrt{\frac{\Gamma\left(\frac{3}{\alpha}\right)}{\Gamma\left(\frac{1}{\alpha}\right)}} \quad \text{et} \quad \Gamma(\beta) = (\beta-1)! \quad \forall \beta \in \mathbb{N}$$

Annexe B

Calcul du $Card(r, n_0)$

B.1 Aspect théorique

Nous rappelons que le calcul du $Card(r, n_0)$ intervient dans l'expression analytique du calcul du modèle de débit de la QVAZM. Notre objectif est de déterminer le nombre de vecteurs Y ayant n_0 composantes nulles sur chaque rayon r .

$$Card(r, n_0) = \text{cardinal} \left(\left\{ \begin{array}{l} Y = \{y_1, \dots, y_n\} \in \mathbb{Z}^n \mid \sum_{i=1}^n |y_i| = r \\ \text{et } \text{cardinal}(\{y_i \mid y_i = 0\}) = n_0 \end{array} \right\} \right) \quad (\text{B.1})$$

L'énumération du nombre de vecteurs $Card(r, n_0)$ sur chaque rayon d'énergie r du réseau $\mathbb{Z}^n$ ayant n_0 composantes nulles, formule B.1, peut être calculée en utilisant une partie des méthodes d'indexage par code préfixe [63]. En effet, il est possible de déterminer l'index d'un vecteur à l'aide d'un vecteur *leader*, et du nombre de permutations et de symétries.

Soit un vecteur *leader* Y^l ayant n composantes dont n_0 composantes nulles, on note également $\overline{n_0} = n - n_0$ le nombre de composantes non nulles :

$$Y^l = \left\{ \underbrace{0, \dots, 0}_{n_0}, \underbrace{y_{i_1}^l, \dots, y_{i_{\overline{n_0}}}^l}_{\overline{n_0}} \right\}$$

Le nombre d'orbite, qu'il est possible de trouver à partir d'un vecteur *leader*, est donné par :

$$\theta(Y^l) = 2^{n-n_0} \frac{n!}{w_1! \dots w_q!}$$

où q est le nombre de valeurs différentes prises par les composantes du vecteur Y^l et w_i le poids de la valeur d'une composante (i.e. le nombre de fois où l'on a une même valeur y_i^l).

B.2 Démonstration

Nous cherchons donc le nombre de vecteurs de rayon r ayant n_0 composantes nulles noté $Card(r, n_0)$. Il faut donc calculer le nombre d'orbites qu'il est possible de trouver à partir de tous les vecteurs *leader* de rayon r (le poids w_1 est le nombre de composantes nulles n_0) :

$$Card(r, n_0) = \sum_j 2^{n-n_0} \frac{n!}{n_0! \prod_{q=2} w_{q_j} (y_{i_j}^l)!}$$

où j décrit tous les vecteurs *leader* de rayon r ayant n_0 composantes nulles.

Maintenant que nous savons dénombrer les orbites d'un vecteur *leader* ayant n_0 composantes nulles, notre problème consiste simplement à balayer tous ces *leader* en totalisant toutes les orbites et en vérifiant que :

$$\sum_{j=1}^{\overline{n_0}} y_{i_j} = r$$

Pour cela, nous avons besoin de générer les vecteurs *leader* ayant n_0 composantes nulles. Or, nous rappelons que les propriétés d'un vecteur *leader* sont que ses composantes y_i^l sont classées dans l'ordre lexicographique. Il est donc possible de déterminer les domaines de variation pour chacune de ses composantes :

$$\begin{aligned} y_{i_1}^l &: \left[1; \left\lfloor \frac{r}{\overline{n_0}} \right\rfloor \right] \\ y_{i_2}^l &: \left[y_{i_1}^l; \left\lfloor \frac{r-y_{i_1}^l}{\overline{n_0}-1} \right\rfloor \right] \\ &\dots \\ y_{i_k}^l &: \left[y_{i_{k-1}}^l; \left\lfloor \frac{r-\sum_{j=1}^{k-1} y_{i_j}^l}{\overline{n_0}-(k-1)} \right\rfloor \right] \\ &\dots \\ y_{i_{\overline{n_0}}}^l &: \left[y_{i_{(\overline{n_0}-1)}}^l; \left\lfloor r - \sum_{j=1}^{\overline{n_0}-1} y_{i_j}^l \right\rfloor \right] \end{aligned}$$

Nous avons donc mis en oeuvre une méthode qui nous a permis d'établir la formule analytique suivante ($y_{i_j} = y_{i_j}^l$ afin d'alléger la notation) :

$$Card(r, n_0) = 2^{n-n_0} \frac{n!}{n_0!} \sum_{y_{i_1}=1}^{\left\lfloor \frac{r}{\overline{n_0}} \right\rfloor} \sum_{y_{i_2}=y_{i_1}}^{\left\lfloor \frac{r-y_{i_1}}{\overline{n_0}-1} \right\rfloor} \dots \sum_{y_{i_{\overline{n_0}}}=y_{i_{(\overline{n_0}-1)}}}^{\left\lfloor r-\sum_{j=1}^{\overline{n_0}-1} y_{i_j} \right\rfloor} \left(\prod_{y_{i_k}=y_{i_1}}^{\left\lfloor r-\sum_{j=1}^{\overline{n_0}-1} y_{i_j} \right\rfloor} \delta(y_{i_k} - y_{i_{(k-1)}}) \times w(y_{i_k})! \right)^{-1} \quad (B.2)$$

$$\text{tel que } \sum_{j=1}^{\overline{n_0}} y_{i_j} = r \quad \text{et} \quad w(y_{i_k}) = \sum_{q=y_{i_1}}^{y_{i_{\overline{n_0}}}} \delta(y_{i_k} - q)$$

Où $\delta(u)$ est la fonction de Kronecker (relation 2.53), $\overline{n_0}$ correspond au nombre de composantes non nulles du vecteur Y et y_{i_k} est la $i_k^{\text{ème}}$ composante non nulle.

Glossaire

<i>DCT</i>	<i>Discrete Cosine Transform</i> , transformée en cosinus discrète
<i>DWT</i>	<i>Discrete Wavelet Transform</i> , transformée en ondelettes discrète
<i>GLA</i>	<i>Generalized Lloyd Algorithm</i> , algorithme de Lloyd généralisé
<i>JPEG</i>	<i>Joint Photographic Expert Group</i> , standard de compression d'image fixe
<i>JPEG2000</i>	Nouveau standard de compression d'image fixe
<i>LBG</i>	Algorithme de Linde, Buzo et Gray
<i>LZ</i>	Algorithme de Lempel-Ziv
<i>MPEG</i>	<i>Moving Pictures Expert Group</i> , standard de codage vidéo
<i>MSE</i>	<i>Mean Square Error</i> , erreur quadratique moyenne
<i>PDF</i>	<i>Probability Density Function</i> , densité de probabilité
<i>PSNR</i>	<i>Peak Signal to Noise Ratio</i> , rapport crête signal sur bruit
<i>QS</i>	Quantification Scalaire
<i>QV</i>	Quantification Vectorielle
<i>QVA</i>	Quantification Vectorielle Algébrique
<i>QVAZM</i>	Quantification Vectorielle Algébrique avec Zone Morte
<i>SNR</i>	<i>Signal to Noise Ratio</i> , rapport signal sur bruit
<i>SPIHT</i>	<i>Set Partitioning In Hierarchical Trees</i> , algorithme <i>SPIHT</i>
<i>TO</i>	Transformée en ondelettes
<i>VA</i>	Variable Aléatoire
<i>ZM</i>	Zone Morte

Notations

$\lceil \cdot \rceil$	Entier supérieur le plus proche
$\lfloor \cdot \rfloor$	Entier inférieur le plus proche
$\text{arr}(\cdot)$	Arrondi à l'entier le plus proche
$\ \cdot\ _{\mathbf{p}}$	Norme $l_{\mathbf{p}}$ (Si $\mathbf{p} = 2$: la norme l_2 correspond à l'erreur quadratique)
$\mathbf{0}$	Vecteur d'énergie nulle
γ	Facteur de projection de la QVA
$\Gamma(n)$	Fonction $\Gamma(n) = \int_0^{\infty} u^{n-1} e^{-u} du \quad \forall n \in \mathbb{R}^+$ ou $\Gamma(n) = (n-1)! \quad \forall n \in \mathbb{N}^*$
λ	Multiplicateur de Lagrange
λ	$\lambda = \frac{\sqrt{2}}{\sigma}$
$\delta(u)$	Fonction de Kronecker : $\delta(u) = \begin{cases} 1 & \text{si } u = 0 \\ 0 & \text{sinon} \end{cases}$ et $\overline{\delta(u)} = \begin{cases} 0 & \text{si } u = 0 \\ 1 & \text{sinon} \end{cases}$
δ	Seuil de la Zone Morte sur le dictionnaire : $\delta = \left\lceil \frac{r_{ZM}}{\gamma} \right\rceil$
σ	Ecart-type
b	Nombre de bits
b_{en}	Nombre de bits d'énergie
b_{pos}	Nombre de bits de position
b_j	$j^{\text{ème}}$ bit de la signature w
$\mathbf{b}_i$	$i^{\text{ème}}$ réalisation d'une variable aléatoire correspondant au bruit additif gaussien
$\mathcal{C}$	Dictionnaire $\mathcal{C} = \{Y_i \mid i = \{0, \dots, L-1\}\} \quad \forall Y_i \in \mathbb{R}^n$
$\mathcal{C}'$	Dictionnaire $\mathcal{C}' = \{Y'_i \mid i = \{0, \dots, L-1\}\} \quad \forall Y'_i \in \mathbb{Z}^n$
$\mathcal{C}_{ZM}$	Dictionnaire avec Zone Morte
$\mathcal{C}_{ZM}^*$	Dictionnaire avec Zone Morte sans le vecteur nul $\mathbf{0}$
$\text{Card}(r, n_0)$	Nombre de vecteurs sur chaque rayon r du réseau ayant n_0 composante nulle
D	Distorsion
D_{QS}	Distorsion d'un Quantificateur Scalaire
D_{QV}	Distorsion d'un Quantificateur Vectoriel
D_{QVA}	Distorsion d'un Quantificateur Vectoriel Algébrique
$f_{\ \mathbf{x}\ _1}(r)$	Densité de probabilité du rayon r de norme l_1 (loi du rayon)
$f_{\mathbf{X}}(X)$	Densité de probabilité du vecteur aléatoire $\mathbf{X}$
$f_{\mathbf{x}}(x)$	Densité de probabilité de la variable aléatoire $\mathbf{x}$
$F_{\ \mathbf{x}\ _1}(r)$	Fonction de répartition du rayon r de norme l_1
I	Image composée de $\mathbf{M}$ échantillons
I_Q	Image quantifiée
$\mathbf{J}$	Nombre de niveaux de la transformée en ondelettes
$\mathbf{j}$	$j^{\text{ème}}$ niveau de la transformée en ondelettes
$\mathbf{K}$	Nombre de sous-bandes de la transformée en ondelettes
$\mathbf{k}$	$k^{\text{ème}}$ sous-bandes de la transformée en ondelettes
K	Clé

L	Nombre de représentants d'un dictionnaire
$N(r)$	Nombre de vecteurs sur le rayon r
n	Dimension d'un vecteur
n_0	Nombre de composantes nulles dans un vecteur
$\overline{n_0}$	Nombre de composantes non nulles dans un vecteur
M	Nombre d'échantillons total
M_v	Nombre de vecteurs total
M_c	Nombre de coefficients d'ondelettes total
M	Nombre de représentants (de l'alphabet d'une variable aléatoire)
M_w	Nombre de bits b_j du message w
$p(x_m)$	Probabilité d'apparition de x_m
$p_{\ Y\ _1}(r)$	Probabilité d'apparition du rayon r de norme l_1
$p_{ZM}(r)$	Probabilité d'apparition du rayon r de norme l_1 avec zone morte
$QS(x)$	Quantification Scalaire de la valeur x
$QV(X)$	Quantification Vectorielle du vecteur X
$QVA(X)$	Quantification Vectorielle Algébrique du vecteur X
R	Débit
r	Rayon ou énergie d'un vecteur
r_T	Rayon de Troncature d'un dictionnaire
r_{ZM}	Seuil de la Zone Morte sur les énergies des vecteurs
SB_k	Sous-bande d'indice k
SB_{BF}	Sous-bande basse fréquence
w	Signature, message composé de bits b_j
X	Vecteur
$\mathbf{X}$	Vecteur aléatoire $\mathbf{X}$
$\mathbf{x}$	Variable aléatoire $\mathbf{x}$
x_m	$m^{ème}$ représentant de la VA $\mathbf{x}$
$x(k)$	Echantillon k d'une source
x_i	$i^{ème}$ Composantes d'un vecteur
Y	Vecteur quantifié
Y_s	Ensemble de vecteurs quantifiés sélectionnés
$Y_{\bar{s}}$	Ensemble de vecteurs quantifiés non sélectionnés
Y_{s_j}	Vecteur j quantifié sélectionné
$\mathbf{y}$	Variable aléatoire $\mathbf{y}$
y_i	$i^{ème}$ composante d'un vecteur Y , $i^{ème}$ représentant de la VA $\mathbf{y}$

Bibliographie

- [1] M. Antonini, M. Barlaud, P. Mathieu, et L. Daubechies, "*Image coding using wavelet transform*", IEEE Transactions Image Processing, vol. 1, p. 205-220, Avril. 1992.
- [2] M. Antonini, P. Raffy, et M. Barlaud, "*Toward entropy constrained lattice vector quantization*", in Proc. Int., Conf. Image processing, Washington, DC, Octobre 1995, p. 121-124.
- [3] M. Barlaud et C. Labit, "*Compression et codage des images et des vidéos (traité ic2)*", Hermes Sciences Publication, ISBN : 2-7462-0328-6, Décembre 2001.
- [4] M. Barlaud, P.Solé, T. Gaidon, M. Antonini, et P. Mathieu, "*Pyramidal lattice vector quantization for multiscale image coding*", IEEE Transactions Image Processing, vol. 3, p. 367-381, Juillet 1994.
- [5] M. Barlaud, "*Wavelets in Image Communication*", Advances in image communication : Elsevier, vol. 5, p. 255, 1994.
- [6] P. Bas, "*Méthodes de tatouages d'images fondées sur le contenu*", Thèse de l'Institut National Polytechnique de Grenoble soutenue le 5 octobre 2000, France.
- [7] P. Bas, J. Chassery, et F. Davoine, "*Using the Fractal Code to Watermark Images*", Proc. IEEE Int. Conf. on Image Processing, vol. I, p. 469-473, Octobre 1998.
- [8] P. Bas, J. Chassery et B. Macq, "*Geometrically Invariant Watermarking Using Feature Points*", IEEE Transactions on Image Processing , Septembre 2002.
- [9] A. Baskurt, M. Khamadja, O. Baudin, F. Dupont, R. Prost, D. Revel, and R. Goutte, "*Adaptive image compression scheme. Application to medical images and diagnostic quality assessment*", Journal Européen des Systèmes Automatisés JESA, vol. 31, n° 7, p. 1155-1171, Dec. 1997.
- [10] K. Belloulata, A. M. Baskurt, H. Benoit-Cattin, and R. Prost. "*Fractal coding of subbands with an oriented partition.*" Signal Processing : Image Communication, 12(3) :243-252, 1998.
- [11] K. Belloulata, F. Davoine, A. Baskurt, and J. M. Chassery, "*Compression hybride fractale/sous-bande accélérée*", In Proc. of 16e Colloque sur le Traitement du signal et des images, GRETSI-97, Grenoble, p. 287-290, Sep. 1997.
- [12] W.R. Bennett, "*Spectra of quantized signals*", Bell Syst. Tech.J., vol. 27, p. 446-472, Juillet 1948.
- [13] H. Benoit-Cattin, A. Baskurt, and R. Prost, "*3D medical image coding using separable 3D wavelet decomposition and lattice vector quantization*", Signal Processing, vol. 59, p. 139-153, Juin 1997.
- [14] P. Carré et C. Fernandez-Maloigne, "*Undecimated wavelet shrinkage estimate of 1d and 2d spectra*". Dans IEEE International Conference on Acoustics, Speech, and Signal Processing. Hilton Hotel & Convention Center, Istanbul, Turkey. 5-9 Juin 2000.

- [15] P. Carré et C. Fernandez-Maloigne, "Use of the information in the wavelet transform maxima for image denoising", *Image Processing and Computer Graphics*, 18(13), p. 1055-1065, 2000.
- [16] F. Chen, Z. Gao, et J. Villasenor, "Lattice vector quantization of generalized Gaussian sources", *IEEE Transactions Informations Theory*, vol. 43, p. 92-103, Janvier 1997.
- [17] S. G. Chang, B. Yu, and M. Vetterli, "Bridging compression to wavelet thresholding as a denoising method," in *Proc. Conf. Information Sciences Systems*, Baltimore, MD, p. 568-573, Mars 1997.
- [18] S. G. Chang, B. Yu, and M. Vetterli, "Image denoising via lossy compression and wavelet thresh-olding," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Santa Barbara, CA, p. 604-607, Nov. 1997.
- [19] S. G. Chang, B. Yu, and M. Vetterli, "Spatially adaptive wavelet thresholding with context modeling for image denoising," *IEEE Transactions Image Processing*, vol. 9, p. 1522-1531, Sept. 2000.
- [20] S. G. Chang, B. Yu, and M. Vetterli, "Adaptative wavelet thresholding for image denoising and compression", *IEEE Transactions Image Processing*, vol. 9, p. 1532-1546, Sept. 2000.
- [21] J.H. Conway et N.J.A. Sloane , "A Fast Encoding Method for Lattice Codes and Quantizers", *IEEE Transactions Informations Theory*, vol. IT-29,6, p.820-824, 1983.
- [22] J.H. Conway et N.J.A. Sloane , "Sphere packing, lattices and groups", Springer Verlag, 1988.
- [23] P.C. Cosman, R.M. Gray and M. Vetterli, "Vector quantization of image subbands : a survey", *IEEE Transactions on Image Processing*, 5(2), p. 202-225, Février 1996.
- [24] J.J. Cox, M.L. Miller, "The first 50 years of electronic watermarking", Published in the *Journal of Applied Signal Processing (IEEE)*, 2, 126-132, Avril 2002.
- [25] I.J. Cox, M.L. Miller and J.A. Bloom, "Digital watermarking", Morgan Kaufmann Publishers, Academic Press, 2002.
- [26] I. Daubechies, *Ten Lectures on Wavelets*, Vol. 61 of *Proc. CBMS-NSF Regional Conference Series in Applied Mathematics*. Philadelphia, PA :SIAM, 1992.
- [27] F. Davoine, "Compression d'images par fractales basée sur la triangulation de Delaunay", Thèse préparée au sein du laboratoire TIMC-IMAG, Spécialité : Signal Image Parole, Grenoble, 20 décembre 1995.
- [28] F. Davoine, M. Antonini, J.-M. Chassery, M. Barlaud, "Fractal image compression based on Delaunay triangulation and vector quantization" in *IEEE Transactions on Image Processing*, 5(2), p. 338-346, Février 1996.
- [29] F. Davoine, G. Robert, J.-M. Chassery, "How to improve pixel-based fractal image coding with adaptive partitions ?" in *Fractals in Engineering* (Eds : J.-L. Vehe, E. Lutton and C. Tricot), Springer-Verlag, p. 292-307, Juin 1997.
- [30] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, vol. 81, p. 425-455, 1994.
- [31] D. L. Donoho and I. M. Johnstone, "Wavelet shrinkage : Asymptopia ?," *J. R. Stat. Soc. B*, ser. B, vol. 57, no. 2, p. 301-369, 1995.

- [32] T.R. Downie and B.W. Silverman, "*The discrete multiple wavelet transform and thresholding methods*", IEEE Transactions Signal Processing, vol. 46, p. 2558-2561, 1998.
- [33] C. Fernandez-Maloigne, N. Richard et E. Laizé. "*Content-based image indexing with color texture descriptors*". Dans SCI'2001, Artificial Intelligence And Cognitive Science Within Virtual Environments. Orlando, Florida, USA. 22-25 juillet 2001.
- [34] T.R. Fisher, "*A pyramid vector quantizer*", IEEE Transactions Informations Theory, vol. 32, p.568-583, Juillet 1986.
- [35] T.R. Fisher et J. Pan, "*Enumeration encoding and decoding algorithm for pyramid cubic lattice and trellis codes*", IEEE Transactions Informations Theory, vol. 41, p. 2056-2061, Nov. 1995.
- [36] Y. Fisher, "*Fractal image compression : Theory and Application*", Springer Verlag, 1995.
- [37] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. Computer J., 7 :149-154, 1964.
- [38] A. Gersho et R. M. Gray, "*Vector Quantization and Signal Compression*", Norwell, MA : Kluwer, 1992.
- [39] R.M. Gray and D.L. Neuhoff, "*Quantization*", IEEE Trans. on Information Theory 44 (6), p. 2325-2383, Octobre 1998.
- [40] L. Guillemot, T. Voinson et J.M. Moureaux, "*Lattice Vector Quantization with Dead Zone for an efficient combined Compression and Watermarking Algorithm*", European Signal Processing Conference EUSIPCO 2002, Toulouse (France), 3-6 septembre 2002.
- [41] P.G. Howard et J.S. Vitter, "*Arithmetic coding for data compression*", Proceedings of the IEEE, vol.82(6), p. 857-865, 1994.
- [42] D.A. Huffman, "*A method for the construction of minimum-redundancy codes*", Proc. of the IRE, vol.40, p. 1098-1101, septembre 1952.
- [43] R. Hunter et A.H. Robinson, "*Runlength*", Proceedings of the IEEE, vol.68, p. 854-867, 1980.
- [44] A.E. Jacquin, "*Fractal image coding : a review*", Proceedings of the IEEE, vol.81, n°10, p. 1451-1465, octobre 1993.
- [45] A.K. Jain, "*Fundamentals of Digital Image Processing*", Prentice Hall Information and System Sciences Series, 1989.
- [46] D.G. Jeong et J.D. Gibson, "*Uniform and piecewise uniform lattice vector quantization for memoryless Gaussian and Laplacian sources*", IEEE Transactions Informations Theory, vol.39, p. 786-804, Mai 1993.
- [47] I. M. Johnstone and B. W. Silverman, "Wavelet threshold estimators for data with correlated noise," J. R. Statist. Soc., ser. B, vol. 59, 1997.
- [48] S. Kasaei et M. Deriche, "*Fingerprint compression using piecewise-uniform pyramid lattice vector quantization*", IEEE Conference on Acoustics, Speech and Signal Processing, Munich, Germany, p. 3117-3120, Avril 1997.
- [49] S. Kasaei, M. Deriche et B. Boashash, "*Fingerprint compression using wavelet packet transform and pyramid lattice vector quantization*", IEICE Special Section on Digital Signal Processing E80-A(8) p. 1446-1451, Août 1997.

- [50] Stefan Katzenbeisser and Fabien A. P. Petitcolas "*Information hiding techniques for steganography and digital watermarking*", Contributions de : Fabrizio Marongui Buonaiuti, Scott Craver, Jean-Luc Dugelay, Frank Hartung, Neil F. Johnson, Martin Kutter, Jong-Hyeon Lee, Stanley Lai, Adrian Perrig and Stéphane Roche. Artech House Books, ISBN 1-58053-035-4, Décembre 1999.
- [51] D. Kundur et D. Hatzinakos, "*Digital Watermarking using Multiresolution Wavelet Decomposition*", Proc. IEEE Int. Conf. On Acoustics, Speech and Signal Processing, Seattle, Washington, vol. 5, p. 2969-2972, Mai 1998.
- [52] D. Lebedeff, "*Etude de la quantification vectorielle des données brutes issues d'un radar à synthèse d'ouverture*", Thèse de l'Université de Nice - Sophia Antipolis, Décembre 1995.
- [53] A. S. Lewis et G. Knowles, "*Image compression using the 2-D wavelet transform*", IEEE Transactions on Image Processing, volume 1, numéro 2, p 244-250, avril 1992.
- [54] Y. Linde, A. Buzo, et R.M. Gray, "*An algorithm for vector quantizer design*", IEEE Transactions Communications, vol. COM-28, p. 84-95, Janvier 1980.
- [55] S.P. Lloyd, "*Least Squares Quantization in PCM*", IEEE Transactions on Information Theory, vol. IT-28, p. 129-137, Mars 1982.
- [56] P. Loyer, J.M. Moureaux et M. Antonini, "*Lattice codebook enumeration for generalized Gaussian sources*" IEEE Transactions on Information Theory, Février 2003.
- [57] M.W. Marcellin, M.J. Gormish, A. Bilgin, and M. P. Boliek, "*An overview of JPEG-2000*", in Proc. DCC 2000, Snowbird, UT, p. 523-541, Mars 2000.
- [58] J. Max, "*Quantizing for minimum distortion*", IEEE Transactions on Information Theory, p.7-12, Mars 1960.
- [59] P. Meerwald, "*Quantization Watermarking in the JPEG2000 Coding Pipeline*", International Federation for Information Processing, Conference on Communications and Multimedia Security, Darmstadt, Germany, Mai 21 - 22, 2001
- [60] P. Meerwald et A. Uhl, "*A Survey of Wavelet-Domain Watermarking Algorithms*", SPIE Symposium, Electronic Imaging, Conference on Security and Watermarking of Multimedia Contents, San Jose, CA, USA, Janvier 20 - 26, 2001
- [61] N. Moreau, "*Technique de compression des signaux*", Masson, Paris et CNET-ENST, ISBN : 2-225-84720-7, 1994.
- [62] J.M. Moureaux, M. Antonini, et M. Barlaud, "*Counting lattice points on ellipsoids : Application to image coding*", Electron. Lett., vol. 31, p. 1224-1225, Juillet 1995.
- [63] J.M. Moureaux, P. Loyer et M. Antonini, "*Low-Complexity Indexing Method for Zn and Dn lattice quantizers*", IEEE Transactions on Communications, vol. 46, 12, p. 1602-1609, Décembre 1998.
- [64] J.M. Moureaux et L. Guillemot, "*Image Compression and Watermarking using Lattice Vector Quantization*", SPIE Conference on Visualization and Data Analysis, vol. 4675-66, 2002
- [65] J.M. Moureaux, "*Quantification vectorielle algébrique pour la compression d'images. Application à l'imagerie radar à synthèse d'ouverture (SAR)*", Thèse de l'Université de Nice - Sophia Antipolis, Décembre 1994.

- [66] B. K. Natarajan, "Filtering random noise from deterministic signals via data compression," IEEE Transactions Signal Processing, vol. 43, p. 2595-2605, Nov. 1995.
- [67] C. Parisot, M. Antonini et M. Barlaud, "Optimal Nearly Uniform Scalar Quantizer Design for Wavelet Coding", Proc. SPIE : Visual Communication and Image Processing, vol. 4671, p. 1185-1193, 2002.
- [68] C. Parisot et M. Antonini, "Compression d'images satellites haute résolution par quantification vectorielle algébrique et allocation de débits optimale", I3S Laboratory internal report n°9819, Université de Nice Sophia Antipolis (France), Octobre 1998.
- [69] W.B. Pennebaker and J.L. Mitchell, "JPEG : Still Image Data Compression Standard". Van Nostrand Reinhold, Kluwer Academic Publishers, ISBN : 0442012721, 1993.
- [70] F. Petitcolas, R. Anderson et M. Kuhn, "Attacks on copyright marking systems" In David Aucsmith, Ed., Second Workshop on Information Hiding, in vol. 1525 of Lecture Notes in Computer Science , p. 218-238, Avril 1998.
- [71] F. Petitcolas, et R. Anderson, "On the limits of steganography", IEEE Transactions on Selected Areas in Communications, vol. 16(4), p. 474-481, Mai 1998.
- [72] W.H. Press, S.A. Teukolsky, W.T. Vetterling, et B.P. Flannery, "Numerical Recipes in C", The Art of Scientific Computing. Cambridge University Press, seconde édition, 1992.
- [73] J.G.G. Proakis et M. Salehi, "Communication Systems Engineering", 2ème édition, Prentice Hall Professional Technical Reference, ISBN : 0130617938, 801 pages, Septembre 2001
- [74] P. Raffy, "Modélisation, optimisation et mise en œuvre de quantificateurs bas débits pour la compression d'images utilisant une transformée en ondelettes", Thèse de l'Université de Nice - Sophia Antipolis, Décembre 1997.
- [75] P. Raffy, M. Antonini et M. Barlaud, "Optimal subband bit allocation procedure for very low bit rate image coding", Electron. Lett., vol. 34, p. 647-648, Avril. 1998.
- [76] P. Raffy, M. Antonini et M. Barlaud, "Distortion-Rate Models for Entropy-Coded Lattice Vector Quantization", IEEE Transactions Image Proc., vol. 9, NO. 12, p. 2006-2017, Décembre 2000.
- [77] F. Ruggeri and B. Vidakovic, "A Bayesian decision theoretic approach to wavelet thresholding," Statist. Sinica, vol. 9, no. 1, p. 183-197, 1999.
- [78] A. Said et W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees", IEEE Transactions Circuits Syst. Video Technol., vol. 6, p. 243-250, Juin 1996.
- [79] N. Saito, "Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion," in Wavelets in Geophysics, E. Foufoula-Georgiou and P. Kumar, Eds. New York : Academic, p. 299-324, 1994.
- [80] A. Sequeira, D. Kundur, "Communication and Information Theory in Watermarking : A Survey", Multimedia Systems and Applications IV, A. G. Tescher, B. Vasudev, and V. M. Bove, eds., Proc. SPIE (vol. 4518), p. 216-227, Denver, Colorado, Août 2001.
- [81] C.E. Shannon, "A mathematical theory of communication", Bell Syst. Tech. Journal, vol. 27, p. 379-423, 623-656, 1948.

- [82] C.E.Shannon, "*Communication theory of security systems*", Bell Syst. Tech. Journal, vol. 28, p. 656-715, 1949.
- [83] J. Shapiro, "*Embedded image coding using zerotrees of wavelet coefficients*," IEEE Transactions Signal Processing, vol. 41, p. 3445-3462, Dec. 1993.
- [84] A. Stoica, P. Carré et C. Fernandez-Maloigne, "*Color images de-noising by wavelet maxima representation and regions segmentation*", CGIP'2002, Poitiers, France, Avril 2002.
- [85] V. Strela and A.T. Walden, "*Signal and image denoising via wavelet thresholding : orthogonal and biorthogonal, scalar and multiple wavelet transforms*", Imperial College, Statistics Section, Technical Report TR-98-01, 1998.
- [86] David Taubman, Erik Ordentlich, Marcelo Weinberger et Gadiel Seroussi, "*Embedded block coding in JPEG 2000*", Signal Processing : Image Communication 17 (1) p. 49-72, 2002.
- [87] D. Taubman, "*High performance scalable image compression with EBCOT*" IEEE Transactions on Image Processing, vol. 9, no. 7, p. 1158-70, Juillet 2000.
- [88] M.J. Tsai J. Villasenor et F. Chen, "*Stack-Run image coding*", IEEE Transactions Circuits Syst. Video Technol., vol. 6, p. 519-521, Oct. 1996.
- [89] M. Vetterli and J. Kovacevic', *Wavelets and Subband Coding*, Englewood Cliffs, NJ : Prentice-Hall, 1995.
- [90] B. Vidakovic, "*Nonlinear wavelet shrinkage with Bayes rules and Bayes factors*," J. Amer. Statist. Assoc., vol. 93, no. 441, p. 173-179, 1998.
- [91] T. Voinson, L Guillemot et J.M. Moureaux, "*Image compression using Lattice Vector Quantization with code book shape adapted thresholding*", IEEE 2002 International Conference on Image Processing, Rochester, New York, Septembre 2002.
- [92] Sviatoslav Voloshynovskiy, Shelby Pereira, Alexander Herrigel, Nazanin Baumgärtner et Thierry Pun, "*Generalized watermark attack based on watermark estimation and perceptual remodulation*", In Ping Wah Wong and Edward J. Delp eds., IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000 : Security and Watermarking of Multimedia Content II, Vol. 3971 of SPIE Proceedings, San Jose, California USA, 23-28 January 2000. (Paper EI 3971-34)
- [93] B. Widrow, "*Statistical analysis of amplitude quantized sampled data systems*", Transactions Amer. Inst. Elec. Eng., Pt. II : Applications and Industry, vol. 79, p. 555-568, 1960.
- [94] Y. Yoo, A. Ortega, and B. Yu, "*Image subband coding using context-based classification and adaptive quantization*," IEEE Transactions Image Pro-cessing, vol. 8, p. 1702-1715, Dec. 1999.
- [95] P. L. Zador, "*Asymptotic quantization error of continuous signals and the quantization dimension*", IEEE Transactions InformationsTheory, vol. 28, p. 139-148, Mars 1982.



Index

- Algorithme, 3
 - d'insertion, 96
 - de détection, 96
 - de la QVAZM, 40
 - de Lloyd-Max, 24
 - GLA, 30
 - LBG (*Linde, Buzo et Gray*), 31
 - QVA, 34
- Allocation optimale, 52
- Bruit
 - de surcharge, 23
 - granulaire, 23
- Codage
 - Stack-Run*, 61
- Débit, 4
 - de la QVAZM (source Laplacienne), 40
- Distorsion, 5
 - de la Quantification Scalaire (QS), 22
 - de la Quantification Vectorielle (QV), 30
 - de la QVA, 36
- Forme de la zone morte, 38
- Imperceptibilité, 103
- Indépendance des sous-bandes, 53
- Indexage, 36
- Langage, 94
- Méthodes, 3
 - avec pertes, 11
 - JPEG2000, 15
 - par apprentissage, 30
 - sans pertes, 10
 - SPIHT, 13
- Normalisation, 33
- Partitionnement, 29
- Quantification, 21
 - Scalaire (QS), 21
 - Scalaire avec zone morte, 25
 - Scalaire non uniforme, 24
 - Scalaire uniforme, 24
 - Vectorielle (QV), 28
 - Vectorielle Algébrique (QVA), 32
 - de la QVAZM, 37
- Réseau régulier de points, 32
- Résultats, 63
- Robustesse, 98
 - de la QVA avec zone morte, 113
 - de la QVA sans zone morte, 108
- tatouage, 85
 - détection, 90, 93
 - insertion, 89, 92
 - multirésolution, 89
 - propriétés, 88
- traitements innocents, 87
- traitements malveillants, 88
- transformée en ondelettes, 6
- Troncature, 33

Résumé : Etant donné les volumes actuels de données numériques, la compression est devenue une phase incontournable des applications mettant en œuvre la transmission ou le stockage d'images numériques. Dans ce contexte, la chaîne de compression basée sur une transformée en ondelettes des données associée à une quantification et un codage entropique fait aujourd'hui référence. Le travail présenté dans ce mémoire concerne les étapes de quantification et de codage de cette chaîne. Généralement la quantification scalaire (QS) est préférée à la quantification vectorielle (QV) pour des raisons de complexité calculatoire, en dépit de la supériorité de la QV démontrée par la théorie de l'Information, en termes de qualité d'image comprimée et de taux de compression. Nous proposons ici la mise en œuvre d'un quantificateur vectoriel algébrique (QVA) alliant les principaux avantages des approches vectorielles et un faible coût de calcul. L'originalité de la méthode réside dans la définition d'une zone morte vectorielle associée au QVA, de forme identique à celle du dictionnaire, qui permet d'effectuer un seuillage par énergie des vecteurs de la source. En d'autres termes, les blocs de coefficients d'ondelettes jugés non significatifs pour la qualité visuelle sont mis à zéro par l'intermédiaire du seuillage. Ceci permet de coder plus précisément les blocs de coefficients significatifs. Ainsi, à taux de compression constant, il est possible de diminuer la distorsion dans chacune des sous-bandes de la transformée en ondelettes, et donc la distorsion globale de l'image comprimée. En outre, un modèle théorique pour le calcul du débit binaire (i.e. du taux de compression) a été mis au point pour accélérer le réglage du paramètre de forme optimal de la zone morte. Par ailleurs, la redondance engendrée par la mise à zéro de nombreux blocs de coefficients a été exploitée dans l'étape de codage à travers un codeur Stack-Run, codeur par plages de zéros dont nous avons modifié la structure pour l'adapter à notre chaîne de compression. Les résultats obtenus ont été comparés avec ceux des algorithmes de référence actuels, à savoir *JPEG2000* et *SPIHT*. Ils sont encourageants et montrent de bonnes performances dans les bas débits, se traduisant par une meilleure préservation de nombreux détails de l'image. Dans un tout autre registre, le développement des communications et notamment des échanges de données numériques a mis en lumière un problème de protection de la propriété intellectuelle de ce type de données. C'est dans ce contexte que s'est développé le tatouage, qui vise à insérer dans une image, un message invisible et indélébile, à l'aide d'une clé secrète, que seul un utilisateur possédant la clé peut extraire. Une des difficultés majeures du tatouage tient dans la résistance aux attaques, qu'elles viennent de pirates, ou de traitements bienveillants comme la compression. Celle-ci est en effet le dual du tatouage, puisqu'elle vise à retirer toute information invisible dans l'image. C'est dans ce contexte que la QVA avec zone morte vectorielle a été également utilisée à profit dans une application de tatouage et compression conjoints, permettant d'améliorer la robustesse du tatouage face à certaines attaques et notamment celle d'un bruit additif gaussien, comme en témoignent les résultats expérimentaux obtenus.

Mots-clefs : compression d'images, Quantification Vectorielle Algébrique (QVA), transformée en ondelettes, "Zone Morte Vectorielle", codage Stack-Run, tatouage d'images.

Abstract : Given the present great amount of numerical data, compression has become an unavoidable step for the transmission or storage applications of numerical images. In this context, the compression chain based on data discrete wavelet transform, associated to quantification and entropy coding, is today a reference. Our work concerns the quantization and coding steps of that chain. Generally we prefer to use scalar quantization rather than vector quantization for computation complexity reasons, in spite of the VQ superiority proved by the information theory, in terms of compression ratio and compressed image quality. We propose to use Lattice Vector Quantization (LVQ), which combines the main advantages of vector approaches and a low complexity. The originality of our method consists in defining a vectorial dead zone, adapted to codebook shape and associated with LVQ. That dead zone enables us to carry out energy thresholding of source vectors. In other words, blocks of wavelet coefficients that are considered to be "non significant" for the visual quality are reset to zero by use of thresholding. It permits to code with more precision the significant coefficient blocks. Therefore, with a constant compression ratio, it is possible to decrease the distortion in each discrete wavelet transform subband, and thus the global distortion of the compressed image. Furthermore, a theoretical rate (i.e. compression ratio) model has been developed, in order to accelerate the optimal setting of the dead zone shape parameter. Moreover, the redundancy generated by the reset to zero of numerous coefficient blocks has been used with the Stack-Run coder. The structure of the latter, which makes use of runs of zeros, has been modified to be adapted to our compression chain. Our results have been compared with the current reference algorithms, namely *JPEG2000* and *SPIHT*; they are encouraging, especially at low bit rate, as can be shown by a good preservation of visual image details. In another field, widespread development of the communications, such as the exchange of numerical data, highlights the problem of copyright and intellectual property protection for the use of those data. In this context, there has been a great development of watermarking, which aims to insert an invisible and indelible message in an image, by use of a secret key; thanks to which only the key owner can extract the message. One of the major difficulties of watermarking is to face out attacks, coming from hackers or unintentional processes as compression. Indeed, compression is the dual of watermarking, since it tries to remove all invisible information in the image. It is in this context that Dead Zone Lattice Vector Quantization has been used very relevantly for both watermarking and compression, permitting to improve the watermarking robustness to some attacks such as an additive gaussian noise, as shown by experimental results.

Key-words : images compression, Lattice Vector Quantization (LVQ), discrete wavelet transform (DWT), "Vector Dead Zone", Stack-Run coding, Watermarking.