Etude d'un schéma de quantification vectorielle algébrique et arborescente. Application à la compression de séquences d'images numériques

Vincent Ricordel

N° d'ordre : 1570

THÈSE

présentée devant

L'UNIVERSITÉ DE RENNES I

U.F.R. Structure et Propriétés de la Matière

Pour obtenir

Le grade de Docteur de l'Université de Rennes I
Mention : Traitement du Signal et Télécommunications
École Doctorale : Sciences pour l'Ingénieur

par

Vincent RICORDEL

Équipe d'accueil : TEMIS/IRISA
Composante universitaire du Directeur de Thèse : IFSIC/IRISA

Titre de la thèse

Étude de schémas de quantification vectorielle algébrique et arborescente. Application à la compression de séquences d'images numériques.

Soutenue le 2 décembre 1996, devant la commission d'Examen composée de :

M.	René	COLLOREC	Président
MM.	Jean-Pierre	ASSELIN DE BEAUVILLE	Rapporteurs
	Michel	BARLAUD	
Mme	Odile	MACCHI	Examinateurs
MM.	Jean-Paul	GUILLOIS	
	Claude	LABIT	
Mme	Christine	GUILLEMOT	Membre invité

A Marie-Bé.

Remerciements

Je tiens à exprimer tous mes remerciements

A Monsieur René Collorec, Professeur à l'Université de Rennes I, d'avoir accepté de présider le jury de cette thèse.

A Monsieur Michel Barlaud, Professeur à l'Université de Nice-Sophia Antipolis, et à Monsieur Jean-Pierre Asselin de Beauville, Professeur à l'Université de Tours, d'avoir accepté d'être les rapporteurs de ce manuscrit.

A Madame Odile Macchi, Directeur de recherche CNRS, à Monsieur Jean-Paul Guillois, Enseignant-chercheur à l'ENST-Paris, ainsi qu'à Madame Christine Guillemot, Ingénieur au CCETT-Rennes, d'avoir accepté de participer à mon jury et de s'être intéressés à mes travaux.

A Monsieur Claude Labit, Directeur de recherche INRIA, qui m'accueilli dans son équipe et a soutenu mes travaux. Je lui suis très reconnaisssant pour la confiance et l'intérêt qu'il m'a constamment accordés. J'ai ainsi pu apprécié ses qualités humaines, sa bienveillance et la pertinence de ses remarques scientifiques.

Ce travail de recherche a été effectué à l'IRISA (Institut de Recherche en Automatique et Systèmes Aléatoires) à Rennes, au sein du projet TEMIS (Traitement, Exploitation et Modélisation d'Images Séquentielles). Je souhaite aussi remercier chaleureusement l'ensemble des doctorants, et toutes les personnes qui ont contribué de près ou de loin à la réalisation de cette thèse. J'ai ainsi pu travailler dans les meilleures conditions, et ces trois années de recherche ont été passionnantes.

Table des matières

Glossaire des acronymes

ACELP	*Adaptive Code Excited Linear Predictive Coder*
ACVLC	*Arithmetically Computed Variable Length Coding*
BF	Boucle Fermée
BFOS	algorithme de Breiman, Friedman, Olshen et Stone [Breiman et al.84]
BO	Boucle Ouverte
CCIR	Comité Consultatif International pour la Radiodiffusion
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CD-ROM	*Compact Disk - Read Only Memory*
CELP	*Code Excited Linear Predictive Coder*
CIF	*Common Intermediate Format*
CQF	*Conjugate Quadrature Filter*
CTIA	*Cellular Telecommunication Industry Association*
DAB	*Digital Audio Broadcasting*
DCC	*Digital Compact Cassette*
DCT	*Discrete Cosinus Transform*
DPCM	*Differential Pulse Code Modulation*
ECMA	Equation de Contrainte du Mouvement Apparent
EDI	*Extented Definition Interleaved format*
EDP	*Extented Definition Progressive format*
ELT	*Extented Lapped Transform*
EQM	Erreur Quadratique Moyenne
ETSI	*European Telecommunication Standart Institute*
FPLMTS	*Future Public Land Mobile Telecommunication System*
GOP	*Group Of Pictures*
HDI	*High Definition Interleaved format*
HDP	*High Definition Progressive format*
i.i.d	indépendante(s) et identiquement distribuée(s)
INRIA	Institut National de Recherche en Informatique et Automatique
IRISA	Institut de Recherche en Informatique et Signaux Aléatoires
ISO/IEC	*International Standart Organisation /* *International Electronic Commisssion*
JPEG	*Joint Photographic Expert Group*

LAN	*Local Area Networks*
LD-CELP	*Low Delay Code Excited Linear Predictive Coder*
LBG	algorithme de Linde, Buzo et Gray [Linde et al.80]
MIC	Modulation par Impulsions Codées
MICD	Modulation par Impulsions Codées en Différentiel
MICDA	Modulation par Impulsions Codées en Différentiel Adaptatif
MLT	*Modulated Lapped Transform*
MPEG	*Moving Picture Expert Group*
MSDL	*MPEG4 Syntactic Description Language*
NTSC	*National Television Systems Committee*
PAL	*Phase Alternation Line*
PCM	*Pulse Code Modulation*
PSNR	*Peak Signal to Noise Ratio*
QCIF	*Quarter Common Intermediate Format*
QMF	*Quadrature Mirror Filter*
QS	Quantification Scalaire - ou - Quantificateur Scalaire
QV	Quantification Vectorielle - ou - Quantificateur Vectoriel
QVA	Quantification Vectorielle Algébrique - ou - Quantificateur Vectoriel Algébrique
QVAA	Quantification Vectorielle Algébrique et Arborescente - ou - Quantificateur Vectoriel Algébrique et Arborescent
QVAr	Quantification Vectorielle Arborescente - ou - Quantificateur Vectoriel Arborescent
RNIS	Réseaux Numériques à Intégration de Services
RPE-LTP	*Regular Pulse Excitation - Long Term Prediction*
RRP	Réseau(x) Régulier(s) de Points
SECAM	Système Electronique Couleur avec Mémoire
SNHC	*Synthetic Natural Hybrid Coding*
SVH	Système Visuel Humain
TEMIS	Traitement, Exploitation et Modélisation d'Images Séquentielles
TTA	Technique Temporelle Asynchrone
TV3D	Télévision numérique en relief
TVHD	Télévision numérique Haute Définition
UIT-T	Union Internationale de Télécommunications - secteurs des Télécommunications
UMTS	*Universal Mobile Telephone System*
UVLC	*Universal Variable Length Coding*
VSELP	*Vector Sum Excited Linear Predictive Coder*
VLSI	*Very Low System Integration*
VQ	*Vector Quantization*

Introduction

Le domaine d'étude est le codage ou la compression de séquences d'images numériques. En effet, la simple numérisation des images génère un volume considérable d'information qui doit être comprimée à des fins d'archivage ou de transmission. Les recherches visent alors l'élaboration de méthodes efficaces de compression qui, sous une contrainte d'un débit de transmission fixé, assurent la meilleure qualité de reconstruction des images transmises. Les traitements numériques ont notamment pour but de tirer profit des redondances intra-inter-images et des caractéristiques psychovisuelles humaines (élimination de l'information inutile).
Ce domaine d'étude a déjà fait l'objet de recherches intenses qui ont conduit à l'élaboration de recommandations et de standards (H261, MPEG1&2). Les efforts se poursuivent dans le cadre de l'édification de futures normes de compression du signal vidéo animé (MPEG4, TVHD numérique, TV3D) et la conception de nouveaux services de vidéocommunications (télésurveillance, télémanipulation, consultation de bases de données image).

Ce travail a été réalisé au sein de l'équipe TEMIS du laboratoire IRISA-INRIA de Rennes, et fait suite à des études antérieures de schémas de codage de séquences d'images par quantification vectorielle adaptative multiclasse [Maresq86] [Monet et al.90]. Cette recherche entre aussi dans le cadre d'une action nationale concertée et menée par plusieurs laboratoires (I3S-Lassy Sophia [Antonini91], L2S-ESE Gif [Skowronski96], IRISA Rennes) et partiellement financée par le CNET (contrat CNET/GDR No $936B005$), pour comparer plusieurs approches de quantificateurs vectoriels sur diverses sources vectorielles d'information image. Enfin il faut noter la récente et importante activité nationale sur ce thème de recherche avec des travaux menés au laboratoire TIMC/IMAG Grenoble [Davoine95], au CCETT Rennes [Onno96] et à l'IRESTE Nantes [Senane96].

Un schéma de compression-décompression se décompose généralement en trois phases : la première transforme et réorganise les données pour la quantification. Cette seconde phase réalise, de façon effective et irréversible, la compression en représentant le signal par un nombre fini d'états. Ainsi réduite la séquence est transmise ou stockée. Enfin, pour la reconstruire, une dernière étape doit effectuer la transformation inverse de la première. Notre thèse se focalise principalement sur l'étude de la seconde phase du schéma de compression.

Si on se réfère à la théorie introduite par Shannon, de meilleures performances sont réalisables si on quantifie des vecteurs plutôt que des scalaires. La quantification vectorielle consiste donc à répartir dans un espace de dimension fixée un nombre déterminé de vecteurs représentant, ce nombre étant fonction du débit alloué au quantificateur. Ce thème de recherche a toujours été le champ d'investigations pour les approches de compression à très bas débit et la conception de codecs dissymétriques où la partie décodage est souhaitée à très bas coût.
Cependant la théorie qui établit la supériorité de la quantification vectorielle n'est pas constructive car elle ne décrit pas la conception du codeur optimal, de plus elle n'intègre pas les contraintes pratiques de temps de calcul et de taille mémoire. En fait la quantification vectorielle, technique complexe, ne se révèle performante qu'inscrite au sein d'un schéma de codage lui même élaboré où un changement de représentation des symboles de la source à comprimer doit être réalisé. De façon intuitive, il s'agit de prétraiter la source afin de localiser l'information dans une zone compacte de l'espace et d'obtenir le signal le plus stationnaire. Ainsi la quantification est pleinement efficace : les vecteurs de reproduction sont répartis dans cet espace confiné et le dictionnaire constitué de ces représentants demeure valide au cours du temps. Enfin il faut savoir que la spécification de la source vectorielle détermine les caractéristéristiques conceptuelles que doit avoir le quantificateur.
Pour la compression de séquences d'images animées, sachant que les modules de la chaîne de compression-décompression sont interdépendants et contribuent conjointement aux performances globales du codeur, nous optons pour un schéma performant de prétraitement de la source qui comprend : un module d'estimation et de compensation du mouvement, son principe est d'estimer le mouvement entre deux images successives puis de prédire l'image suivante à partir du mouvement calculé et de l'image courante (ce module réduit la redondance inter-images) ; un module de transformation de l'erreur résiduelle entre l'image prédite et celle réelle par une décomposition linéaire sur base orthonormée dans le cadre du codage en sous-bandes ou par représentation en ondelettes (ce module réduit la redondance intra-image). Une modélisation de la distribution statistique de ce type de source vectorielle hybride est généralement faite à l'aide d'une fonction de la famille des gaussiennes généralisées. Cependant malgré le prétraitement, le signal image à quantifier n'est jamais stationnaire. C'est pourquoi nous choisissons une technique d'apprentissage pour concevoir le dictionnaire et ainsi rendre opérationnelle une quantification vectorielle adaptative où une réactualisation des vecteurs représentants est opérée, au cours du temps, à partir de séquences d'apprentissage représentatives de la source. Pour concevoir notre quantificateur, nous n'avons pas retenu une technique classique d'apprentissage du type LBG arborescent car l'encodage et surtout la construction du dictionnaire demeurent trop complexes. La quantification algébrique, qui est rapide, n'est intéressante que si la source est stationnaire et telle que sa statistique autorise une troncature aisée des réseaux. Notre approche vise alors à tirer profit de ces deux techniques de codage : la quantification simple et rapide sur réseaux réguliers de points, la construction par apprentissage d'un dictionnaire arborescent qui autorise toujours une quantification rapide mais également une partition de l'espace adaptée à la distribution de la source et adaptée à un critère débit-distorsion.

Précisément notre contribution dans le cadre de la quantification de l'erreur de prédiction après compensation du mouvement d'un système vidéo couvre les points suivants :

- considérant un réseau algébrique parmi ceux pour lesquels des algorithmes de quantification rapides sont connus (*i.e.* Z^k, D_k, E_8 et Λ_{16}) :
  - nous tronquons ce réseau tel qu'il puisse être emboîté, en le contractant, dans son voronoï ;
  - une hiérarchie de réseaux tronqués est obtenue en ajustant leurs échelles afin qu'un réseau de résolution supérieure s'emboîte dans le réseau de résolution juste inférieure ;
  - un schéma simple de quantification multi-étages en découle.
- un dictionnaire arborescent non-équilibré est construit par apprentissage et l'arbre peut-être découpé ou élagué suivant le critère débit-distorsion introduit par BFOS. L'approche de classification arborescente descendante est retenue car elle demeure adaptée à la construction du dictionnaire lorsque la dimension vectorielle croît ;
- dans ce contexte le réseau algébrique le plus efficace est déterminé. Il s'agit de Z^k qui offre un emboîtage optimal ;
- le quantificateur vectoriel est validé en l'inscrivant au sein de chaînes complètes de compression-décompression. Des solutions aux problèmes spécifiques posés au quantificateur (*e.g.* allocation binaire entre les sous-bandes des images transformées, traitement des vecteurs source dont la norme est supérieure à celle maximale envisagée par le dictionnaire) sont proposées. Deux schémas de codage sont utilisés :
  - un schéma classique mettant en oeuvre les techniques adoptées par les codeurs de la famille MPEG (*i.e.* estimation-compensation du mouvement par blocs, transformée en cosinus discrète) ;
  - un schéma novateur issu des travaux de compression par filtrage basé mouvement où le module de codage s'intéresse à une segmentation des images en régions homogènes au sens du mouvement. Une décomposition par transformée en ondelettes est faite.

Le document de thèse est organisé comme suit :

- dans le premier chapitre nous donnons d'abord un bref aperçu des étapes de la normalisation de la compression des images numériques. Cette partie permet de rappeler les enjeux économiques qui ont entraînés et entraînent encore de gros efforts de développement et de recherche, les évolutions des standards de compression de JPEG à MPEG2, et les enjeux suscités par la mise au point de MPEG4. Dans une autre partie nous développons quelques rappels en compression de séquences d'images numériques afin de situer notre travail dans le contexte général de la théorie de l'information et des approches de codage. Nous nous intéressons plus précisément aux outils de décorrélation inter-intra-images mis en oeuvre avant la quantification ;

- le lecteur averti moins intéressé par le contexte général de notre étude peut directement engager sa lecture au deuxième chapitre entièrement consacré à la quantification vectorielle. Nous en donnons les principes et le formalisme. Nous présentons des résultats obtenus avec le cas particulier de la quantification scalaire et enchaînons par l'analyse théorique qui explicite la supériorité du cas vectoriel. Un premier état de l'art sur les méthodes mises en oeuvre suit, avec la quantification vectorielle optimale trop coûteuse et quelques formes de base qui visent à réduire cette complexité. Nous présentons aussi dans ce chapitre une solution classique au problème de l'allocation binaire optimale entre les sous-bandes des images transformées ;

- nous proposons de développer un quantificateur vectoriel algébrique et arborescent, l'objet des troisième et quatrième chapitres est alors de décrire avec précision et critique la quantification vectorielle algébrique puis celle arborescente. Le troisième chapitre enchaîne une description des réseaux réguliers de points à un état de l'art des méthodes de quantification spécifiques sur ces réseaux. Le quatrième chapitre détaille particulièrement les méthodes et critères introduits pour la construction d'un dictionnaire arborescent non-équilibré ;

- au cinquième chapitre nous explicitons la conception de notre quantificateur. Après avoir analysé les caractéristiques de la source vectorielle, nous introduisons la technique de troncature et d'emboîtage des réseaux, nous décrivons le schéma de quantification multi-étages résultant et la construction du dictionnaire arborescent non-équilibré. Une fois le réseau optimal déterminé, nous achevons ce quantificateur vectoriel algébrique et arborescent en fixant le traitement des vecteurs source marginaux d'énergie trop grande. Nous développons également un algorithme d'allocation binaire visant à accéder à un plus grand nombre de quantificateurs optimaux pour le codage en sous-bandes ;

- jusqu'à présent un cadre expérimental de quantification de sources vectorielles synthétiques était suffisant. Le sixième et dernier chapitre fournit deux schémas complets de codage vidéo intégrant notre quantificateur vectoriel : un manipulant des outils classiques de type MPEG, l'autre basé régions et orienté vers le codage bas débit de scènes visiophoniques. Les outils expérimentaux sont décrits et les résultats analysés ;

- finalement une conclusion générale résumant l'essentiel de nos travaux est faite. Des perspectives de travail pour améliorer nos résultats, et aborder les nouvelles stratégies de codage introduites par la partition des images en régions, sont données.

Chapitre 1

Normalisations et principes de la compression d'images numériques

1.1 Un aperçu des normes de compression des images numériques

1.1.1 Introduction

La compression des signaux de parole, de musique ou d'images représente un enjeu économique important. En effet les récents développements en électronique et en informatique ont fait surgir des possibilités et des besoins de manipulation des signaux numériques à des fins de stockage et de transmission. Les standards sont alors une nécessité afin que les industriels de l'audiovisuel puissent manipuler de la même façon les données. Des groupes d'experts ont été rapidement mis en place par les organismes internationaux pour solliciter les industriels et les chercheurs. Un gros effort de développement et de recherche a donc été fourni dans le domaine du codage spécialement ces dix dernières années et déjà a abouti à toute une série de recommandations et de normes.
Au début des années 70, l'UIT-T (c'est le nouveau nom du CCITT) a d'abord souhaité s'engager dans le processus de normalisation du codage du signal de parole (voir l'annexe A) pour le réseau téléphonique public, car les enjeux économiques étaient à cette époque les plus significatifs dans ce secteur et car ce signal, considéré comme localement stationnaire contrairement à ceux audio et vidéo, a un modèle de production simple sous la forme d'un filtrage autorégressif. Nous citons cet exemple car le standard ultime (le codeur CELP) aboutit à une analyse LPC pour déterminer à un rythme régulier les coefficients du filtre avec utilisation de la quantification vectorielle comme outil de compression.
Ce sont les étapes de la normalisation de la compression des images numériques et surtout de la vidéo qui, naturellement dans le contexte de notre étude, retiennent notre attention. Pour ces signaux complexes ayant des caractéristiques statistiques variées, les organismes internationaux ont séparé les différents besoins relatifs aux systèmes de communication. Ceci se traduit par l'émergence successive de standards différents pour le codage de l'image fixe et de la vidéo. Nous notons une hiérarchisation des normes avec une progression vers

des systèmes de plus en plus sophistiqués (chaque époque offrant des prouesses techniques plus grandes) où le nouveau standard intègre les outils éprouvés de codage du précédent. Pour la vidéo, le domaine applicatif est si vaste (*e.g.* les méthodes peuvent-être utilisées pour la télévision numérique, la visiophonie, la visioconférence, les jeux vidéo ou les stations de travail multimédia) que plusieurs normes sont et seront encore nécessaires en fonction des applications, de leur degré de complexité et de leur taux de compression. Là est l'enjeux de la mise au point de nouveaux outils de compression vidéo tel que la quantification vectorielle, afin qu'elle soit intégrée aux prochains standards qui répondront aux besoins des futurs systèmes de communication.

1.1.2 Compression des images fixes

Le standard JPEG, pour l'archivage et la transmission d'une image fixe couleur de bonne qualité, s'imposait. Les applications sont nombreuses: photos satellites, photos d'agence de presse, tableaux, ...
Nous choisissons de décrire ce standard car les techniques mises en oeuvre seront étendues et adaptées aux normes de codage vidéo. Les étapes que nous pouvons retenir concernant l'édification de JPEG sont:

1980, recommandation pour le **fac-similé** [Ccitt80]: celle-ci est mise en place par l'UIT-T pour la transmission, en environ une minute sur la ligne téléphonique, d'une image noir et blanc en résolution numérique au format A4 de l'ISO. A titre d'exemple le taux de compression est 7 si l'image originale représente 2 Mbits et la ligne autorise 4800 bauds.

1992, norme **JPEG** [Wallace91] [Pennebaker et al.93]: le standard de l'ISO/ UIT-T pour la compression des images fixes multiniveaux est reconnu sous l'appellation ISO/IEC 10918-1 ou norme JPEG. D'importantes contraintes ont été imposées: la qualité des images reconstruites doit être excellente, la complexité opératoire raisonnable, et le nombre d'applications maximal de façon à créer un effet de masse encourageant la conception de circuits VLSI. Pour le parcours séquentiel de l'information image et la structuration du flux de données à transmettre, un balayage de l'image de type "raster scan" est imposé (balayage de gauche vers la droite et de haut en bas), un encodage progressif et hiérarchique est aussi prévu (où un premier encodage fournit une image de qualité médiocre, puis des encodages successifs conduisent à une meilleure résolution). Notons qu'un codage sans pertes numériques (*i.e.* réalisant une procédure de compression réversible) de l'image est possible. Le format des images codées est en principe inférieur à 768x576 pixels pour des débits allant de 8 Mbit/s à 40 Mbit/s.

Nous donnons le principe de l'algorithme en considérant le codage avec pertes d'une image noir et blanc, sachant qu'une image couleur (constituée d'une composante de luminance et de deux de chrominances) est traitée comme un ensemble d'images de ce type. Cependant l'algorithme, tenant compte du système visuel humain moins sensible aux hautes fréquences des chrominances, procède en sous-échantillonnant ces dernières par un facteur

deux avant de les quantifier, et en mettant en oeuvre une matrice de seuillage adaptée :

- une décomposition séquentielle de l'image en bloc de taille 8x8 pixels est réalisée, suivie d'une transformée en cosinus discrète (DCT) bi-dimensionnelle de chaque bloc (soit une décomposition du signal sur une base de 64 fonctions orthogonales) ;

- après seuillage (mise à zéro des coefficients inférieurs à un seuil), les 64 coefficients d'un bloc sont alors quantifiés uniformément (une table de 64 éléments définit les pas de quantification). Un critère perceptif est introduit à ce niveau car les éléments jugés peu significatifs visuellement (ce sont les composantes hautes fréquences, l'énergie d'une image étant concentrée dans les fréquences basses) sont plus grossièrement quantifiés (pas de quantification plus grand). Ce critère perceptif introduit est donc indépendant des caractéritiques de l'image (sa taille, ...) et de l'application. Une table de quantification type est fournie par le standard mais elle n'est pas imposée ;

- le premier coefficient d'un bloc (appelé coefficient "DC") représente la valeur moyenne de ce bloc. Les coefficients DC de blocs voisins sont fortement corrélés, ils sont donc codés par MICD (*i.e.* quantification de la différence d'amplitude de l'échantillon et d'une valeur prédite [Gray84]) ;

- enfin un codage entropique sans distorsion est appliqué afin de tirer profit des caractéristiques statistiques de l'image. Les coefficients quantifiés des blocs sont ordonnés par un balayage en zig-zag plaçant d'abord ceux relatifs aux basses fréquences, puis un codage à longueur variable de code de la suite des symboles non nuls et des plages de zéros est effectué. Ce codage consiste à assigner aux symboles les plus fréquents des mots de code plus longs, les codages entropiques les plus couramment utilisés étant celui de Huffman et celui arithmétique.

1.1.3 Compression de séquences d'images numériques animées

Introduction

Cette description des recommandations et standards pour la compression de séquences d'images numériques animées, est motivée par le fait que nous testerons notre quantificateur vectoriel inscrit dans un schéma de codage dont les outils seront issus de la famille MPEG. Nous distinguons, au niveau du plan, les recommandations pour la compression des signaux vidéo de type visiophone et vidéoconférence (où la caméra est fixe), des normes pour l'archivage et la diffusion (où la caméra peut-être fixe ou mobile). Notre présentation est naturellement chronologique car la complexité des systèmes de codage est allée croissante et il y complémentarité entre eux. Nous développons aussi la description de MPEG4, car ce standard générique illustre comment la voie demeure ouverte pour la conception de nouveaux outils de codage vidéo telle que la quantification vectorielle.

Recommandations pour la compression de signaux vidéo de type visiophone et vidéoconférence

1990, recommandation **H.261** [Liou91] : celle-ci fut définie par l'UIT-T pour la téléphonie visuelle utilisant le RNIS dont le débit d'un canal est 64 kbit/s (56 kbit/s dans la configuration nord américaine) : le visiophone (empruntant un à deux canaux), la visioconférence (empruntant au minimum six canaux pour une qualité raisonnable, le maximum prévu étant de 30 canaux).
Premièrement l'UIT-T a du définir un format pour le signal vidéo qui soit commun aux nombreuses normes existantes (625 ou 512 lignes, ...) : le format CIF et le QCIF dont les caractéristiques sont données au tableau 1.1. Les fréquences des images sont de 7.5, 10, 15 ou 30 images/s.
A titre d'exemple, le débit d'une séquence d'images couleur au format CIF à 30 images/s (pour la visioconférence) est :

$$\left(288 \times 352 \times \frac{3}{2}\right) \times 30 \times 8 \approx 36 \text{ Mbit/s}$$

et celui d'une séquence QCIF à 10 images/s (pour le visiophone) est :

$$\left(144 \times 176 \times \frac{3}{2}\right) \times 10 \times 8 \approx 3 \text{ Mbit/s}$$

Le taux de compression pour transmettre cette dernière dans un canal RNIS est donc presque égal à 50.
La recommandation subdivise les formats CIF et QCIF en une structure hiérarchique avec le bloc (de taille 8x8 pixels pour la luminance et les chrominances), le macrobloc (formé de 4 blocs de luminance, et de 2 blocs de chrominance), le groupe de blocs (3x11 macroblocs) et l'image (3 groupes de blocs pour une QCIF, 12 pour une CIF). L'algorithme de codage pour comprimer la vidéo exploite les redondances spatiales dites intra-image et celles temporelles inter-image, exactement c'est un schéma de codage hybride utilisant la DCT et le MICD avec la compensation de mouvement.
En mode intra, le MICD n'est pas opérationnel. Le schéma est alors très proche de celui JPEG : chaque bloc est transformé et quantifié linéairement. En mode inter, le MICD est opérationnel et une quantification prédictive standard est réalisée :

- d'abord une compensation du mouvement est faite en prédisant l'image courante à partir de celle précédente. La prédiction repose sur l'estimation d'un vecteur de mouvement purement translationnel pour chaque macrobloc de luminance (uniquement), la précision de l'estimation est réalisée au pixel entier. La recommandation H.261 ne spécifie pas la méthode d'estimation à employer, l'information de mouvement doit-être codée et transmise ;

- si la différence entre le macrobloc courant et celui prédit est supérieure à un certain seuil, l'erreur de prédiction est transmise. Elle subit une transformation par une DCT bi-dimensionnelle sur ses blocs et une quantification linéaire.

Les données quantifiées à transmettre sont ensuite multiplexées et un codage entropique est effectué (cinq tables de codes à longueur variable sont disponibles). Un tampon de transmission règle le débit d'information à un niveau constant en contrôlant le pas de quantification (qui croît si le buffer est presque plein et inversement décroît quand le buffer se vide). Notons que pour la quantification des coefficients des blocs transformés, des critères psychovisuels interviennent aussi avec une allocation des bits disponibles aux composantes les plus significatives.
Cette recommandation n'a pas complètement défini les algorithmes de codage afin de permettre les améliorations futures, par contre le format des données transmises est parfaitement figé afin de définir avec certitude la nature du train binaire d'informations à transmettre et à décoder, le codeur est donc défini.
Une application actuelle de H.261 est pour le standard H.320 retenu pour le vidéophone. Cette dernière norme regroupe le standard vidéo (H.261), ceux audio (G.711, G.722 et G.728) et ceux fixant les protocoles relatifs aux réseaux (H.221, H.230 et H.242).

1996 : une future recommandation dénommée **H.263** est en cours de ratification par l'IUT-T. Ce standard pour la vidéo vient prendre place au sein de la norme H.324 conçue pour la vidéoconférence transmise sur réseau local LAN à 28,8 kbit/s. H263, qui sera compatible avec la recommandation H.261, offre en plus : le codage d'une large gamme de formats image (du sous-QCIF 88x72 pixels au 16CIF 1536x1152 pixels), une estimation du mouvement adaptée aux blocs de taille 8x8, des services de codage optionnels (sur demande du récepteur), et une meilleure qualité image dès les très bas débit.

Compression des signaux vidéo pour l'archivage et la diffusion

1988, création du groupe d'experts **MPEG** : ce comité d'étude commun à l'UIT-T et au CCIR est mis en place sous le contrôle de l'ISO/IEC. Il a pour objet l'étude de la compression des sons et des signaux vidéo pour la diffusion numérique avec la TVHD, le multimédia, l'enregistrement sur CD-ROM de films, la publication électronique, ...

1993, norme internationale **MPEG1** ou norme **ISO/IEC 11172** [Isoiec93] [Legall91] : Le codeur décrit par la norme est alors constitué de deux codeurs séparés: un pour le signal vidéo (**MPG-Vidéo**), l'autre pour le signal musical (**MPG-Audio**). "MPEG system" décrit la synchronisation et le multiplexage des deux chaînes de bits respectives à chacun des codeurs, le but étant de transmettre l'ensemble de l'information à un débit d'environ 1,5 Mbit/s (la borne maximale étant de 1,8 Mbit/s).

A titre d'information, nous rappelons pour MPEG-audio (encore appelée MUSICAM) que la qualité nécessaire à l'enregistrement, la transmission et la reproduction des signaux musicaux interdit la mise en oeuvre de modèles simples de représentation. Les applications sont multiples et font désormais appels à des techniques numériques, citons : le DAB, la partie son de la TVHD, l'enregistrement sur DCC, la transmission sur ligne RNIS, ...
Cette famille de codeurs pour la partie audio, autorise différentes fréquences d'échantillonnage (f_e = 32, 44.1 et 48 kHz) et donc différents débits (192, 128 et 64 kbit/s).

Trois couches se distinguent : la première fonctionne à 192 kbit/s pour une qualité de son excellente (transparence par rapport à l'original), la seconde pour un débit plus faible (128 kbit/s) produit une qualité de son identique, la troisième encore à l'étude offrira à 64 kbit/s un son de qualité acceptable. Les méthodes employées sont essentiellement : le codage par transformée et le codage en sous-bandes qui cherchent, avant quantification, à décorréler le signal et donc à concentrer sa puissance sur un nombre réduit de composantes qui pourront être finement quantifiées ; l'utilisation d'un modèle d'audition très élaboré prenant en compte des critères psycho-acoustiques de masquage des fréquences non-audibles.

Trois groupes d'applications sont visées par MPEG1 (MPEG-vidéo) qui se veut une norme de codage générique, indépendante d'une application particulière :

- celles **assymétriques** qui utilisent le décodage d'un train binaire préenregistré. Les délais de codage ne sont donc pas importants et il est possible d'optimiser la qualité des images en réalisant plusieurs passes de codage de façon à ajuster les paramètres (*e.g.* consultation de publications électroniques, vidéos, jeux vidéos) ;
- celles **symétriques** qui nécessitent l'utilisation mixte du codage et du décodage en temps réel dans un cadre de communications interactives ou de productions de séquences (*e.g.* production de publications électroniques, visiophonie) ;
- celles de stockage avec l'enregistrement sur CD-ROM, DAT, disque optique, ...

Nous pouvons aussi citer la transmission sur les réseaux large bande RNIS, asynchrones TTA ou locaux LAN.
MPEG1 a bénéficié des travaux réalisés lors de la conception de la norme JPEG et de la recommandation H.261 pour réaliser un équilibre optimal entre codage intra-image et celui inter-image :

- la redondance spatiale est réduite, comme pour JPEG, par utilisation d'une transformation orthogonale fréquentielle spatiale (DCT) puis quantification dans le domaine transformé ;
- la redondance temporelle est extraite par compensation du mouvement où, pour des blocs de l'image de taille 16x16 pixels, on prédit l'image courante par translation des blocs de l'image à des instants précédents (technique de mise en correspondance de blocs ou "block matching", des techniques basées sur la représentation multirésolution des images sont aussi utilisables). L'estimation du mouvement est faite avec une précision au demi pixel. L'information de mouvement doit-être transmise (un vecteur de translation par bloc).

Mais à la différence de H.261, MPEG1 n'explicite pas un format image. Le débit pour l'image étant de 1,15 Mbit/s le format vidéo typique visé est le YCbCr 4:2:0 (appelé aussi YUV 4:1:1) qui est restreint par rapport au YUV 4:2:2 décrit par l'avis 601 du CCIR [Ccir82] pour la diffusion de la télévision numérique (voir les paramètres au tableau 1.1). Toujours par rapport à H.261, le taux de compression atteint est moins élevé

de façon à assurer une très bonne qualité de restitution des images. Cependant la principale difficulté à résoudre par MPEG réside dans la possibilité de pouvoir accéder aléatoirement aux images de la séquence (*e.g.* afin de démarrer n'importe où la visualisation).
A la structure hiérarchique de blocs introduite par H.261, MPEG a du ajouter un niveau structurel supérieur : le groupe d'images ou GOP (un exemple est montré à la figure 1.1). Un GOP est constitué de 3 types d'images : les images intra (I) (*i.e.* celles codées en mode intra), les images prédites (P) à partir des images I et P précédentes, les images bidirectionnelles (B) interpolées en utilisant les images I ou P qui précèdent et suivent l'image B. Nous pouvons remarquer les conséquences de cette structures en GOP :

- les images B n'étant jamais utilisées comme prédiction, elles peuvent être plus sévèrement codées (*e.g.* les rapports de compression entre les images I:P:B peuvent-être 10:3:1) ;
- une différence d'ordre des images au codage et à l'affichage apparaît. Ainsi pour notre exemple de la figure 1.1, l'ordre de codage est :

mode image	I	P	B	B	P	B	B	P	B	B	I	B	B
numéro image	1	4	2	3	7	5	6	10	8	9	13	11	12

- la prédiction améliorée de MPEG1 qui utilise les images passées et futures se fait au détriment d'une complexité matérielle accrue (contrôle du débit, mémoire au décodeur pour réorganiser les séquences).

Notons que deux types de table de quantification sont disponibles, chacune adaptée à la nature statistique des données à coder : une table de type JPEG pour la quantification des blocs intras, une autre pour les blocs différentiels résultant de la prédiction temporelle ou de l'interpolation (où, si la prédiction est correcte, les hautes fréquences prédominent et peuvent être quantifiée plus grossièrement). Ces deux types de tables déterminent des pas de quantification "uniformes" exceptés autour de zéro, avec pour les blocs non-intras une large zone vide ("dead-zone") et au contraire pour ceux intras des pas de quantification plus fins.

1994, norme **MPEG2** ou norme **ISO/IEC 13818** [Isoiec94] : les applications télévisuelles ont fortement orienté MPEG2. En plus des formats entrelacés, de nouvelles fonctionnalités étaient demandées telles que le codage hiérarchique, la compatibilité entre niveaux de qualité pour un même format image, l'adaptation aux grands formats (TVHD), le masquage d'erreurs, le codage des formats 4:2:2 et 4:4:4.
La vidéo à qualité de reconstruction télévisuelle ("broadcast") nécessite aussi une qualité supérieure à celle obtenue avec MPEG1 (avec lequel MPEG2 est compatible) ; pour obtenir cette qualité, un débit de 4 à 6 Mbit/s est nécessaire. Le spectre d'utilisations visées est plus large avec la transmission par câble, satellite et voie terrestre de la TV numérique (MPEG2 permet la multiplication du nombre de chaînes pour offrir de la vidéo à la demande), la vidéoconférence, le stockage de programmes audiovisuels, la télésurveillance, ...
La principale nouveauté de MPEG2 est donc de coder l'entrelacé en prenant compte

format	fréquence Hz	mode	nb pts Y	nb lig. Y	nb pts Cb Cr	nb lig. Cb Cr	aspect	débit Mbit/s	type codage
QCIF	30	prog.	176	144	88	72	4/3	9	I
CIF	30	prog.	352	288	176	144	4/3	36	I
SIF525	30	prog.	352	240	176	120	4/3	30	I
SIF625	25	prog.	352	288	176	144	4/3	30	I
4.2.0-625	25	entrelacé	720	576	360	288	4/3	124	II
4.2.0-525	30	entrelacé	720	480	360	240	4/3	124	II
4.2.2-625	25	entrelacé	720	576	360	576	4/3	166	III
4.2.2-525	30	entrelacé	720	480	360	480	4/3	166	III
EDI	25	entrelacé	960	576	480	576	16/9	221	III
EDP	50	prog.	960	576	480	576	16/9	442	III
HDI	25	entrelacé	1920	1152	960	1152	16/9	885	III
HDP	50	prog.	1920	1152	960	1152	16/9	1769	III

type codage	norme	application	débit
I	H.261	visiophonie	p x 64 kbit/s
II	MPEG1	1/4 TV	1,5 Mbit/s
III	MPEG2	TV/TVHD	1,5 4 8 10 Mbit/s

TAB. 1.1 – *Paramètres des différents formats image.*
"nb pts Y" *indique le nombre de points de luminance par ligne*
"nb pts Cb Cr'" *celui des points de chrominance*
"nb lig." *précise le nombre de lignes par images*
"aspect" *correspond au rapport de la taille d'écran (horizontal/vertical)*
La recommandation 601 du CCIR [Ccir82] définit deux sous-ensembles de formats image : le premier pour la norme américaine NTSC (taille de l'image de luminance 720×486, pour la chrominance 360×486, 30 images/s), le second pour les normes européennes PAL et SECAM (taille de l'image de luminance 720×576, pour la chrominance 360×576, 25 images/s)

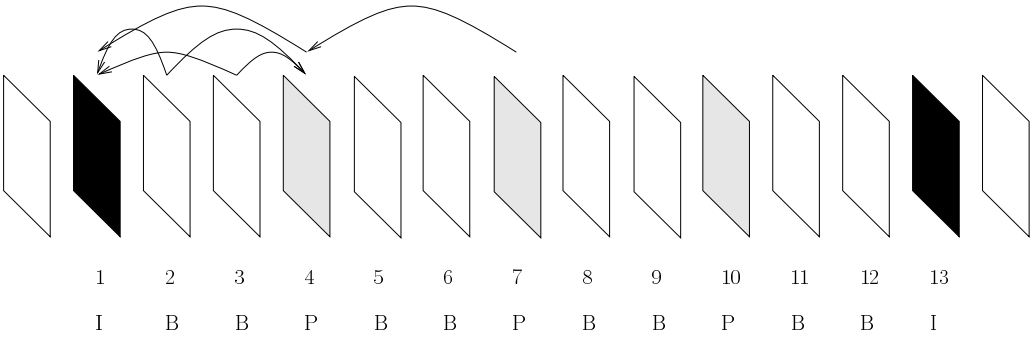


FIG. 1.1 – *Mode de prédiction des images.*
Les deux paramètres importants sont M (la distance en nombre d'images entre deux images prédites P) et N (celle entre deux images intras I, les images B étant celles bidirectionnelles) qui permettent un accès plus ou moins aléatoire aux images. Les paramètres courants sont M = 3 et N = 12 pour un système TV à 25 Hz, ou M = 3 et N = 15 pour un système à 30 Hz.

des déplacements des objets d'une trame à l'autre (le codage de trames par MPEG1 en réunissant celles-ci en images restitue des images de qualité médiocre chargées d'artéfacts, le codage séparé des trames est peu efficace car il ne prend pas en compte la corrélation verticale entre celles-ci). MPEG2 suit alors les principes de codage de MPEG1 mais réalise en plus une prédiction entre images et/ou trames (la meilleure solution étant choisie), et fera la DCT sur des blocs image ou trame. Pour répondre à toutes ces possibilitée de codage MPEG2 a été conçue comme une "boîte à outils à base de DCT" autour d'un concept de "**profiles**" et de **niveaux.** Précisément un "profile" est une technique de codage issue de la boîte à outils MPEG2 assemblée pour répondre à une application, et les niveaux différents qui peuvent exister dans un même "profile" sont caractérisés par un ensemble de paramètres fixant les valeurs maximales de résolution de l'image codée, de fréquence et de débit via la taille du tampon mémoire régulateur. Il existe cinq "profiles" :

- le "Simple profile" n'utilise pas d'images interpolées ;
- le "Main profile" est le schéma de codage classique (utilisation d'images I, P et B) pour le format TV 4:2:0 ;
- le "SNR scalable" permet d'envoyer un train binaire supplémentaire au train principal pour accroître la qualité de l'image TV (le train binaire principal peut-être mieux protégé contre les erreurs que celui supplémentaire) ; ce "profile" convient au schéma de codage à deux niveaux de qualité ;
- le "Spatial scalable" permet de coder la TVHD à partir du codage format TV (pour une codage hiérarchique où le décodeur TV accède à une partie du train binaire correspondant à la TVHD) ;
- le "High profile" pour une meilleure restitution des couleurs, ajoute la possibilité de coder le format 4:2:2 alors que les autres profils ont le format 4:2:0 comme format d'entrée.

On dénombre quatre niveaux (les débits maximaux sont donnés en fonction du "Main profile") : "Low" (format 1/4 TV, 4 Mbit/s), "Main" (format TV, 15 Mbit/s), "High 1440" (HDTV avec 1440 points/ligne, 60 Mbit/s), "High 1920" (HDTV avec 1920 points/ligne, 80 Mbit/s).
Dans ce contexte, un décodeur est à la norme MPEG2 (tel "profile"/niveau) s'il peut au plus décoder un train binaire correspondant à ce "profile"/niveau, et un train binaire sera à la norme MPEG2 (tel "profile"/niveau) si au moins un décodeur de ce "profile"/niveau peut le décoder.
Nous pouvons énumérer des fonctions définies par MPEG2 :

- l'adaptation aux différents formats d'entrée (4:2:0 et 4:2:2) ;
- l'adaptation au besoin de gradation des codeurs dits hiérarchiques (*i.e.* la capacité d'avoir plus d'une résolution d'image et/ou niveau de qualité dans le train binaire des données vidéo) à des fins de compatibilité entre formats pour la diffusion TV (*e.g.* compatibilité MPEG1/MPEG2 ou HDTV/TV). Les données vidéo sont alors

hiérarchisées et transmises sur plusieurs couches qui sont décodées, suivant l'application, en nombre variable (nous verrons au chapitre 4 que la quantification vectorielle arborescente est adaptée au codage hiérarchique de l'information). Deux modes graduels ("scalable mode") se distinguent et peuvent se combiner : la gradation spatiale ("spatial scalability") quand il n'y a pas de changement de fréquence entre les images, la gradation temporelle ("temporal scalability") quand les deux images ont des fréquences image différentes.

Les caractéristiques algorithmiques de MPEG2 sont celles de MPEG1 (le codage classique des images IPB se retrouve dans le "Main profile") auxquelles s'ajoutent d'autres possibilités de codage : soit toutes les images sont codées en intra (le débit résultant est maximal), soit elles sont toutes prédites (le délai de codage-décodage est ainsi réduit) et demeure une image intra périodique ou une réactualisation cyclique des lignes de macroblocs en intra.

1996 : dès 1994 un groupe de travail se met en place pour définir les bases de la future norme **MPEG4** [Pereira96] dont le but est de définir un environnement de représentation vidéo autorisant, avec bien sûr un aspect de compression (ce ne sera pas forcément le seul critère pris en compte) de nouvelles fonctionnalités basées sur le contenu des scènes. A titre d'exemples trois champs d'applications peuvent-être distingués : les services audiovisuels interactifs (*e.g.* manipulation de bases de données et de terminaux interactifs, jeux vidéo, éditions audiovisuelles, téléachat) ; les services de communication audiovisuels avancés qui peuvent utiliser une combinaison de médias (*e.g.* accès universel et communication entre terminaux fixes ou mobiles) ; la télésurveillance et la télémanipulation. Trois idées maîtresses sont retenues :

- le concept central de **contenu** tel que MPG4 permette une description des scènes et qu'il soit possible d'interagir ;
- l'idée d'**intégration** où l'information audiovisuelle apparaît et est traitée de façon variée avec des objets synthétiques ou naturels (codage SNHC), 2D ou 3D ; une information audio mono, stéréo ou multicanaux (son "surround") ; une information vidéo mono, stéréo ou issues de plusieurs capteurs ; divers outils d'analyse et de codage. MPEG4 pourrait alors offrir un environnement standardisé ou une approche globale de traitement de toutes ces informations serait possible ;
- l'idée de **flexibilité** et d'**extensivité** afin d'intégrer aisément les matériels, algorithmes et autres outils futurs. A cette fin MPEG4 développe un langage propre de description syntaxique (ou MSDL).

Le premier choix de MPEG4 s'est fait au niveau des outils de représentation de la scène. Celle-ci est structurée en un ensemble d'objets audiovisuels (à chaque objet est associée une composante objet vidéo et/ou une composante objet audio). Les études actuelles portent sur l'intégration restreinte d'un objet vidéo 2D de forme arbitraire (*i.e.* une région).
Un nouvel environnement de représentation, manipulant une structure de données différentes de celles de MPEG1&2, est aussi défini. Il repose sur la composition de la scène en objets suivant un critère sémantique et non pas suivant des contraintes de codage,

la contribution de chaque objet est indiquée et tous sont initialement disponibles au codeur. A titre d'exemple, aucune restriction n'est faite sur le partionnement de la scène en régions (la segmentation peut-être automatique, semi-automatique ou manuelle ; leur nombre, taille et format sont libres).
L'objet, qui est l'unité spatiotemporelle de base, détermine aussi l'unité de codage portant : l'indice de contribution à la scène, la forme, la position, le taux de compression, les intéractions avec les autres objets et l'information de texture. Après répartition des ressources binaires entre les objets, chacun est codé indépendamment par l'outil le plus adapté.
Pour conclure, il faut distinguer deux types de représentation au codage :

- une pour la représentation de l'information qui est organisée et structurée afin de répondre à une fonctionnalité donnée, elle est indépendante de la technique de codage utilisée mais influence la structure du flot binaire ;

- une pour le codage (le portage) de l'information sémantique (l'accès aux objets) qui, classiquement, vise à restituer la meilleure qualité pour les ressources binaires disponibles.

MPEG4 est donc un standard générique fournissant des outils pour représenter une scène comme la composition de multiples objets, indépendamment de la façon dont ces objets sont générés et de l'application. La séparation entre les outils de définition des objets (qui n'a pas à être standardisée) et ceux de codage, a pour but d'autoriser la détermination libre des objets et l'utilisation d'outils génériques de codage.
En janvier 1996 une première version d'un modèle de vérification définissant complètement un environnement de codage/décodage (*i.e.* une plate-forme expérimentale commune où les nouveaux outils à tester sont intégrés par substitution) ainsi que des spécifications syntaxiques relatives au langage de description ont été produits. C'est dans ce cadre que des propositions ont été soumises par des chercheurs et des industriels pour coder les objets. Or les techniques qui ont le mieux répondu, en terme de qualité et de fonctionnalité, ne sont que des versions modifiées des standards existants (notamment la norme H263). Ce résultat souligne que les nouvelles techniques de codage ne sont pas suffisamment optimalisées pour rivaliser avec les normes actuelles, que la mise au point de nouveaux outils de codage est souhaitable, et qu'un nouveau standard entièrement basé sur des approches régions demeure, pour le moment, prématuré.

1.1.4 Conclusion

Le rappel de l'émergence successive des normes de compression pour les images numériques, souligne les efforts considérables de développement et de recherche fournis en quelques années, afin de répondre aux besoins et aux enjeux économiques de l'électronique et de l'informatique. Les standards audiovisuels annoncent aussi ce que sera la télévision numérique du futur : la réception par satellites, la haute définition, un format cinéma, un son numérique n'ayant rien à envier à celui d'une chaîne Hi-fi et aussi tout un ensemble de logiciels et de systèmes de télécommunication. Il faut remarquer que la recherche de

forts taux de compression avec obtention d'une bonne qualité de restitution, sera toujours réalisée par la non transmission de l'information prédictible, et l'exploitation efficace des propriétés psychovisuelles et psychoacoustiques de masquage du système auditif et visuel humain.
Les nouveaux codeurs vidéo devront aussi répondre aux nouvelles fonctionnalités, et MPEG4 acceptera les outils de compression qui coderont au mieux les objets. La quantification vectorielle, déjà retenue par les ultimes standards de codage du signal de parole, pourrait être le prochain outil de compression choisi pour la future génération de codeurs vidéo. En effet cette technique est supérieure à la quantification scalaire utilisée par les standards actuels. Cette dernière consiste à discrétiser les amplitudes des échantillons indépendamment les uns des autres, par contre la quantification vectorielle regroupe les échantillons en vecteurs avant de les traiter (*i.e.* trouver les vecteurs représentants parmi une collection ou dictionnaire) et permet en plus de prendre en compte la cohérence, la prédiction et la corrélation du signal (nous détaillerons ces propriétés au chapitre suivant). Toutefois des problèmes apparaissent dus à l'explosion de la complexité lorsque la dimension vectorielle croît, des formes particulières de quantificateurs vectoriels sont donc à introduire afin de réduire cette complexité. Parfois les dictionnaires obtenus, contraints structurellement, permettent intrinsèquement de faire face aux différentes extensions méthodologiques des standards. Nous donnons l'exemple de la quantification vectorielle arborescente où le codage est effectué à l'aide d'un arbre de décision (le coût d'un représentant est alors proportionnel à sa hauteur dans l'arbre, mais la qualité de reconstruction est-elle inversement proportionnelle à cette hauteur). Ce quantificateur peut naturellement répondre aux besoins de gradation des codeurs car l'information est hiérarchisée. Enfin pour une caractérisation et une compréhension des scènes, notons que la quantification vectorielle peut aussi être utilisée comme une outil (parmi d'autres) de classification de formes ou de spécification de prototypes de classes [Ghazzali92].

1.2 Quelques rappels en compression de séquences d'images

1.2.1 Préambule

Nous ne développons pas ici l'aspect quantification (ce sera l'objet des chapitres suivants). Le but est de faire quelques rappels afin de situer notre thèse dans le contexte général de la théorie de l'information, et par rapport aux différentes approches en compression d'images (de nombreux ouvrages et articles tutoriels existent [Jain81] [Kunt81] [Jain89] [Rabbani et al.91] [Kunt et al.93]). Ce qui nous intéresse précisément, est de décrire les outils de décorrélation utilisés avant la quantification dans un système de communication d'images. En suivant l'ordre de la chaîne de codage de séquences d'images que nous utiliserons au chapitre 6, nous présentons les techniques de décorrélation inter-image avant celles intra-image. Il est conseillé au lecteur initié, moins intéressé par cette description contextuelle, de poursuivre directement sa lecture au chapitre 2.

1.2.2 Description d'un système de communication d'images

Considérons le schéma d'une chaîne de compression/décompression d'images numériques décrite en figure 1.2; nous supposerons que les opérations de discrétisation des plans image (échantillonnage) et des amplitudes (quantification) ont déjà été réalisées. L'avantage d'une telle représentation est que, en théorie, il est possible de transmettre le signal avec un taux d'erreur aussi faible que l'on veut. L'inconvénient réside au niveau de la masse d'information générée qu'il est nécessaire de comprimer pour la transmettre ou la stocker, c'est le but du **codage de source**. Cette dernière opération se scinde en deux phases [Kunt et al.85] [Tziritas et al.94]: l'analyse et la quantification.
En nous limitant à un contexte de compression où le codeur n'a pas à répondre à une fonctionnalité particulière, l'**analyse** vise à l'extraction des paramètres caractéristiques des images et exploite au maximum les redondances spatiotemporelles entre pixels. Ces paramètres peuvent porter des informations de nature structurelle (*e.g.* blocs de pixels, régions, objets), des informations de mouvement (*e.g.* champs denses de vecteurs vitesse, modèles de ces vecteurs), les paramètres d'un modèle de prédiction, des informations relatives aux contours et aux textures (dans un domaine transformé ou non). C'est le modèle de sélection et d'extraction de ces paramètres qui va définir les différentes méthodes de codage d'images qui sont présentées dans la suite de ce chapitre.
La seconde phase du codage de source est la **quantification** du signal d'innovation (*i.e.* l'information qui n'a pu être prédite lors de l'analyse). Nous ne détaillons pas dès à présent mais dans les chapitres qui suivent, cette opération qui est l'objet de l'étude de cette thèse. Nous soulignons seulement que c'est à ce niveau qu'est effectuée la diminution (avec ou sans pertes) des éléments binaires nécessaires à la représentation de la source, il faut ajouter que la mise en forme du signal lors de l'analyse a conditionné les propriétés structurelles du quantificateur.
En pratique, définir un codeur nécessite de faire des compromis afin de prendre en compte un certain nombre de contraintes car il faut: évaluer la dégradation apportée au signal en fonction du débit alloué (le choix du critère d'évaluation est difficile car il n'existe pas une mesure de distorsion simple qui soit suffisante pour rendre compte de la dégrada-

tion subjective faite), limiter le coût du traitement qui varie en fonction des applications (*e.g.* un délai de reconstruction très rapide est parfois exigé), garantir une robustesse aux erreurs de tranmission. Par rapport à cette dernière contrainte, nous retenons le cadre d'étude où l'optimisation du codage de la source et celle du codage du canal sont réalisées séparément [Proakis89] (l'optimisation conjointe ne sera pas abordée). La transmission de l'information avec un taux d'erreur aussi faible que voulu par le codage du canal est en fait idéaliste car il est nécessaire de fortement protéger l'information (*i.e.* utilisation d'un code correcteur d'erreur).

La première étape du codage de source dit avec pertes ou **irréversible** (c'est le cas que nous étudierons) consiste évidemment à diminuer le nombre d'échantillons à coder/décoder en exploitant [Jayant et al.84] [Netravali et al.88], les redondances statistiques qui traduisent les corrélations spatio-temporelles du signal, ainsi que les propriétés psychovisuelles de masquage spatial et temporel du système visuel humain. En pratique ceci conduit à un codage dit inter-intra-image qui est souvent une combinaison d'un codage inter-image et d'un codage intra-image. Nous nous proposons de décrire, brièvement, ces méthodes de **codage hybride.**

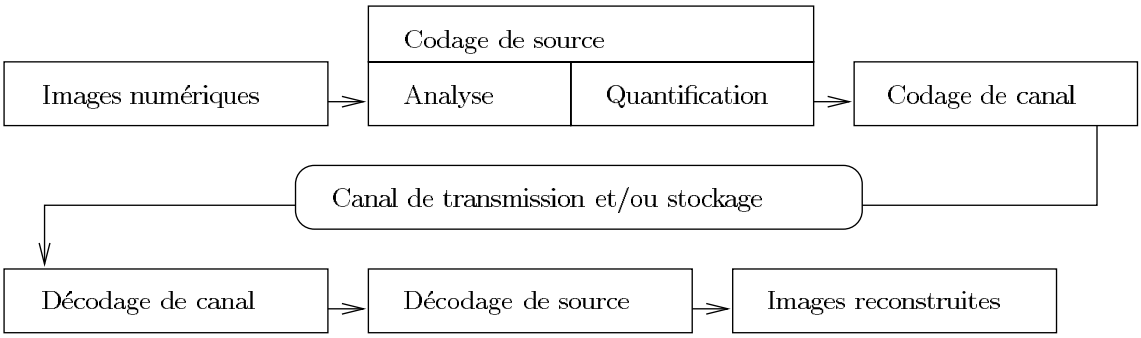


FIG. 1.2 – *Schéma général d'un système de communication.*

1.2.3 Techniques de décorrélation inter-image

Si une scène télévisuelle contient des objets se déplaçant, et si une estimation de ces mouvements peut-être faite, alors il est possible de prédire une image en utilisant les éléments de l'image précédente qui se sont déplacés (*e.g.* un champ dense de vecteurs caractérisant le déplacement de chaque pixel de l'image $t-1$ à t est estimé, puis par compensation, les valeurs des pixels de l'image $t+1$ sont prédites à partir de ce champ des vecteurs déplacement et de l'image t). C'est le principe du **codage par estimation et compensation du mouvement**.
Cette méthode appartient à la grande famille des techniques de décorrélation par prédiction linéaire basée sur le fait que des pixels voisins, dans le domaine spatial ou temporel, sont fortement corrélés (*e.g.* le codage prédictif MICD, ou DPCM en anglais, appartient à cette famille). Alors un pixel peut-être prédit en tenant compte de l'information causale des valeurs de pixels déjà codés (des fonctions de prédiction linéaires ou non, mono ou

bi-dimentionnelles, fixes ou adaptatives ont été utilisées), seule l'erreur de prédiction est quantifiée et transmise. Pour reconstruire les images au décodeur, le signal de prédiction est recalculé et ajouté à l'erreur reçue (voir la figure 1.3).

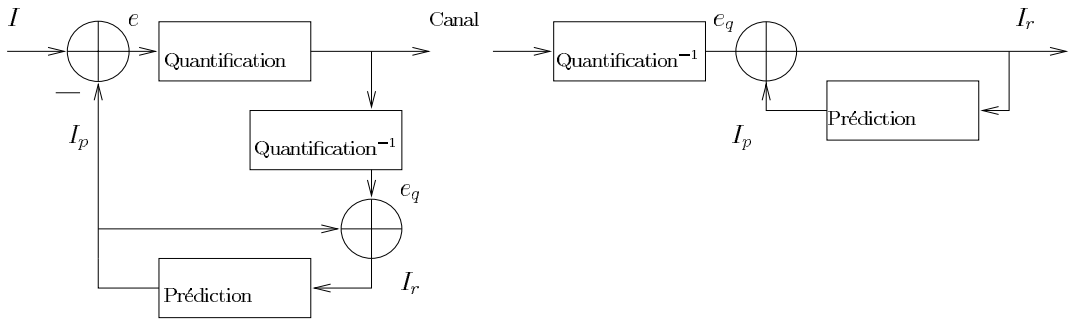


FIG. 1.3 – *Codeur et décodeur prédictifs.*
I est une image originale, I_p celle prédite et I_r celle resconstruite.
e est l'image d'erreurs de prédiction et e_q celle quantifiée.

La mise en évidence et la mesure des mouvements apparents entre deux images successives de la séquence (*i.e.* les mouvements 2D perçus à travers les variations temporelles de l'intensité lumineuse dues aux déplacements 3D des objets dans la scène) sont réalisées par le calcul des vecteurs vitesse instantanée qui créent la notion de champ de vecteurs de vitesse. On peut distinguer trois grandes classes de méthodes d'estimation du champ des déplacements apparents [Netravali et al.88] [Nicolas92] :

- les méthodes par mise en correspondance ("block matching") qui reposent sur une mesure de similarité ou de corrélation entre blocs de deux images successives. Cette méthode rapide est la plus utilisée (elle est à la base des normes MPEG). Cependant elle est dépendante de la résolution de la fenêtre de recherche et peut seulement identifier des mouvements translationnels parallèles à l'image plane ;

- les méthodes différentielles qui reposent sur des contraintes de variations locales de l'intensité. Notons $I(x, y, t)$ la fonction intensité à l'instant t et au pixel de coordonnées (x, y) dans le plan image. Si un motif élémentaire (*i.e.* la projection sur l'image d'un pavé de la surface d'un objet) situé en (x, y), se déplace de (dx, dy) entre deux images successives indicées par t et $t + dt$, alors :

$$I(x + dx, y + dy, t + dt) = I(x, y, t)$$

  Un développement de Taylor au premier ordre autour du point (x, y, t) permet d'aboutir, en négligeant les termes d'ordre supérieur, à : $\vec{w}.\vec{\nabla}I + \frac{\partial I}{\partial t} = 0$ où I est $I(x, y, t)$, $\vec{\nabla}I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$ le gradient spatial de l'intensité et $\vec{w} = (\frac{dx}{dt}, \frac{dy}{dt})$ le vecteur vitesse instantannée. Cette équation de contrainte de mouvement apparent (ECMA) permet de déduire la composante de vitesse mesurable localement et parallèle au gradient spatial de l'intensité : $w^{\perp} = -\frac{\partial I}{\partial t}/\|\vec{\nabla}I\|$. Cependant, le calcul des gradients

difficilement rigoureux et les termes de lissage généralement introduits pour régulariser le champ, posent des problèmes au niveau des discontinuités du mouvement ;

- les méthodes par transformées qui sont utilisées car certaines propriétés de l'évolution spatio-temporelle du signal sont plus évidentes dans le domaine fréquentiel (*e.g.* une translation pure dans l'image modifie le plan de vitesse dans le domaine fréquentiel).

Les deux premières méthodes permettent d'obtenir un champ dense d'estimation de vecteurs vitesse apparente sur toute l'image. Mais les résultats obtenus sont généralement bruités car de nombreux problèmes se posent, entre autres : l'hypothèse de base est de supposer que les variations de la luminance ne sont dues qu'aux mouvements propres des objets (on suppose donc que l'illumination de la scène est constante, qu'il n'existe pas d'effet d'ombrage) ; il est impossible d'estimer correctement le déplacement des zones d'occlusion (les zones découvertes et recouvertes par les objets en mouvement) car il n'existe pas de correspondances d'une image à l'autre ; les estimées de mouvement obtenues ne correspondent à aucun mouvement physique ; les estimateurs locaux sont inefficaces pour estimer de larges amplitudes de mouvements.
Les travaux précédents permettent d'estimer un champ dense de déplacement en réalisant la reconstruction de l'image $t+1$ à partir de l'image t par compensation de mouvement. Mais le volume d'information est considérable (deux paramètres par pixel correspondants aux deux mouvements translationnels). Pour permettre une utilisation efficace de l'information de mouvement dans un schéma de transmission, il est nécessaire de définir un modèle permettant une **description globale** (ou par région) du mouvement [Nicolas92]. A partir de quelques hypothèses (les objets en mouvement peuvent se décomposer en un ensemble de surfaces planes parallèles au plan image, seules les rotations autour de l'axe de profondeur sont permises) les modèles suivants sont utilisés ;

- le modèle de mouvement "constant" (2 paramètres : t_x, t_y),
- le modèle de mouvement linéaire simplifié "semi-linéaire" (4 paramètres : t_x, t_y, k, θ),
- le modèle de mouvement "linéaire" (6 paramètres : $t_x, t_y, k, \theta, h_1, h_2$).

Les vecteurs de paramètres apportent alors une représentation compacte du champ des vitesses de la région et correspondent respectivement à des composants 2D de translation apparente (t_x, t_y), un rapport de divergence (k), un l'angle de rotation (θ), des composants hyperboliques (h_1, h_2). En codage on cherche à minimiser le volume d'information à transmettre et donc à minimiser le nombre de paramètres nécessaire à la reconstruction des images. Le problème étant de trouver le meilleur compromis entre la qualité de la reconstruction et le nombre de paramètres pour une région donnée, le modèle semi-linéaire est alors souvent retenu [GG95] et c'est celui que nous utiliserons au chapitre 6.

Nous notons enfin que deux types de structures de codage avec compensation du mouvement coexistent [Netravali et al.79] [Driessen et al.90] avec :

- **compensation du mouvement en avant** qui repose sur une méthode simple de mise en correspondance de blocs (blocs rectangulaires, mouvement de translation) ou

plus complexe (régions de formes abitraires, modèle de mouvement semi-linéaire et suivi de segmentation temporelle). La figure 1.4 présente le schéma d'un tel codeur où, en plus de l'erreur de prédiction, l'information relative au mouvement (paramètres des vecteurs, éventuelle carte de segmentation) doit-être transmise. Ce codeur est néanmoins préféré (nous le retiendrons pour nos futures expérimentations) car l'estimation du mouvement, qui est réalisée sur les images originales, conserve toute sa cohérence ce qui est primordial à bas débit pour une bonne restitution de la scène ;

- **compensation du mouvement en arrière** où l'estimation du mouvement est faite en utilisant l'information causale connue au codeur et au décodeur (voir la figure 1.4), aucune information supplémentaire en plus de l'erreur de prédiction n'est à transmettre. Cependant à très bas débit, l'estimation du mouvement à partir des images codées-décodées entâchées d'erreurs, risque d'ajuster les régions uniquement en fonction du critère numérique de minimisation de distorsion et de perdre la cohérence physique du mouvement, ce qui est très pénalisant pour le rendu visuel de la scène.

Nous avons présenté la compensation du mouvement en avant ainsi que le modèle de mouvement semi-linéaire, nous les manipulerons au chapitre 6 pour expérimenter notre quantificateur vectoriel au sein de codeurs vidéo orientés vers le codage très bas débit. Cette première phase de codage inter-image a réalisé la décorrélation suivant l'axe temporel de la séquence d'images. Un codage plus efficace est effectué, si une décorrélation spatiale intra-image est faite sur les images d'erreurs de prédiction avant que celles-ci ne soient quantifiées puis transmises. Nous proposons à présent de passer en revue les principales méthodes de codage intra-image qui sont utilisées au sein de codeurs hybrides, sachant que l'objet de notre étude est la conception du quantificateur vectoriel qui achève la compression de ce signal différentiel.

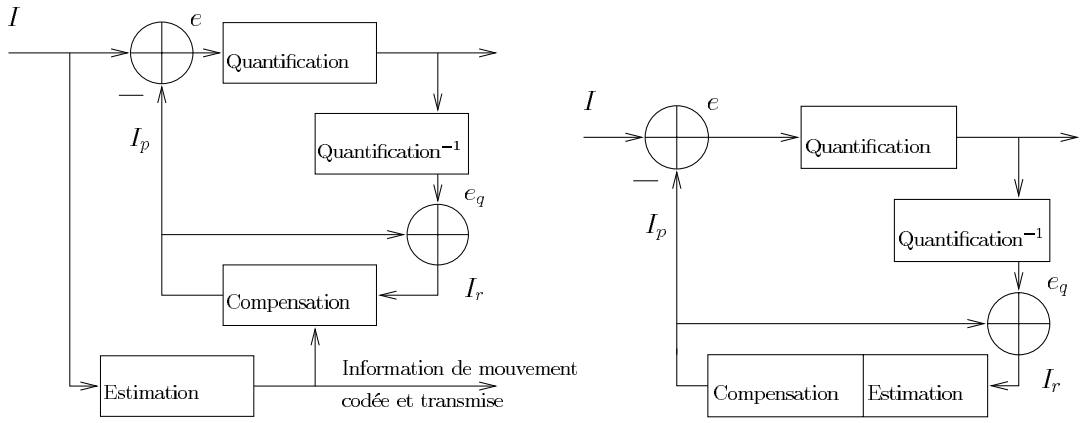


FIG. 1.4 – *Codeur avec compensation du mouvement en avant / codeur avec compensation du mouvement en arrière.*

1.2.4 Techniques de décorrélation intra-image

Codage par transformée

Une technique de codage efficace consiste à répartir judicieusement les ressources binaires disponibles entre différents sous-ensembles hétérogènes de façon à réserver davantage de bits aux plus significatifs [Jayant et al.84] [Shoham et al.88] [Woods91] [Ramchandran et al.93]. Cependant les pixels d'une image présentent, *a priori*, une même distribution et sont dépendants entre-eux. Le codage par transformée est alors largement utilisé de façon à rendre hétérogène les différentes composantes d'un bloc en les décorrélant et en concentrant l'information sur un faible nombre d'éléments.
En pratique pour effectuer ce codage simultané de pixels, l'image est découpée en blocs carrés et à chacun de ces vecteurs une transformée bidimensionnelle est appliquée. Si $I(x,y)$ est un bloc de taille $(l \times m)$ pixels prélevés dans l'image, sa transformation est donnée par :

$$F(u,v) = \sum_{x=0}^{l-1} \sum_{y=0}^{m-1} I(x,y).A(x,y,u,v)$$

Les $(l \times m)$ coefficients F sont codés et transmis. Au récepteur, la transformation inverse est réalisée à l'aide des coefficients décodés $\hat{F}$:

$$I_d(x,y) = \sum_{u=0}^{l-1} \sum_{v=0}^{m-1} \hat{F}(u,v).B(u,v,x,y)$$

A et B sont les noyaux de la transformation (*i.e.* des opérateurs linéaires). Il est avantageux de les choisir orthogonaux (pour une meilleure compression), et séparables (par simplicité) tels que $A(x,y,u,v) = A_1(x,u).A_2(y,v)$ et $B(u,v,x,y) = B_1(u,x).B_2(v,y)$. Nous obtenons deux transformations monodimensionnelles s'appliquant, l'une sur les lignes, l'autre sur les colonnes de l'image. Il faut remarquer qu'en l'absence de codage des coefficients, une reconstruction parfaite serait obtenue car il y a conservation de l'énergie et de l'entropie. La théorie [Rabbani et al.91] montre aussi que l'efficacité de la transformée est liée à la taille du bloc transformé (le codage par transformée n'exploite que la corrélation contenue dans le bloc même).
Le deuxième intérêt de passer dans un domaine transformé apparait car cette technique de codage est liée à la théorie des bancs de filtre à reconstruction parfaite [Vaidyanathan90]. L'application de critères à caractère psychovisuel permet alors d'éliminer les composantes qui n'ont pas de contribution perceptuelle : le système visuel humain étant sensible aux basses fréquences, une quantification fine des coefficients représentant celles-ci est faite, les autres sont alors grossièrement quantifiés.
La transformée optimale est celle de Karhunen-Loeve (TKL) qui produit des coefficients décorrélés et concentre le maximum d'énergie dans les premiers coefficients. Le noyau de la transformation est fonction de la matrice d'autocorrélation du signal, la TKL est donc peu utilisée si ce dernier est non-stationnaire (*e.g.* il faut calculer un noyau pour chacun des blocs d'une image à transformer [Jain79]). Si l'interprétation fréquentielle n'est pas simple, celle en terme de modes propres est par contre aisée.

En pratique la transformée de Fourier est utilisée, elle possède une interprétation fréquentielle évidente et des algorithmes de calculs rapides. Cependant c'est la transformée en cosinus discrète (TCD) qui est la plus utilisée car elle permet d'obtenir des coefficients réels, elle réalise aussi une bonne approximation de la TKL (pour des tailles élevées de blocs) et des algorithmes de calculs rapides sont disponibles [Vetterli et al.84] (une autre raison est que les coefficients d'energie faible sont localisés dans les hautes fréquences et sont donc conformes à l'aspect psychovisuel). Les autres transformées parfois utilisées en image sont celles de Haar, de Walsh et de Hadamard [Clarke85].
Pour la quantification plusieurs stratégies sont possibles :

- une quantification **inter-bande** où le bloc transformé est directement quantifié. Un codage par plage est souvent effectué après seuillage, quantification scalaire uniforme et balayage en zig-zag (l'allocation binaire est figée en fonction de critères psychovisuels). Un inconvénient apparait lorsqu'une erreur locale de quantification sur un coefficient est faite, cela entraîne dans tout le bloc un effet de moyenne ;

- une quantification **intra-bande** où, en considérant les blocs transformés de l'image, on constitue des sous-bandes en regroupant les coefficients relatifs à la même activité fréquentielle (voir figure 1.5). Cette configuration est particulièrement intéressante en quantification vectorielle car chaque sous-bande ayant une activité fréquentielle et un poids psychovisuel propres, il est possible de différemment répartir entre elles les bits et de choisir la forme des vecteurs quantifiés de façon à tirer profit de la corrélation inter-bloc. C'est donc cette forme de quantification que nous manipulerons au chapitre 6 pour le codeur de type MPEG.

Une faiblesse du codage par transformée réside dans le traitement et le codage indépendant des blocs sans tenir compte des discontinuités existant à la fréquence bloc (la taille des blocs n'est pas adaptable). Pour atténuer ces problèmes qui se traduisent par des effets de blocs visibles, Malvar a proposé les transformées dites avec recouvrement [Malvar92] (avec les MLT et ELT).

Codage en sous-bandes

Dans un codeur en sous-bandes [Vaidyanathan90], le signal (*i.e.* l'image I de la figure 1.6) est filtré par un banc de M filtres d'analyse réalisant une partition de l'axe fréquentiel, puis sous-échantillonné de façon à ne pas accroître la quantité d'information à coder (le cas idéal étant un sous-échantillonnage critique par un facteur M et sans perte d'information), et chaque sous-bande est codée séparément (une stratégie d'allocation binaire est naturellement appliquée donnant un poids plus important aux composantes basses fréquences). Au récepteur chaque composante est décodée, un sur-échantillonnage et une interpolation par un banc de filtres de synthèse sont réalisés, puis les différents signaux sont additionnés.
Le filtrage est efficace si les sous-bandes sont une représentation spectrale fidèle relativement à l'image (le cas idéal produirait alors des sous-bandes décorrélées entre elles). Il faut aussi que le traitement et la quantification de celles-ci entraînent une distorsion

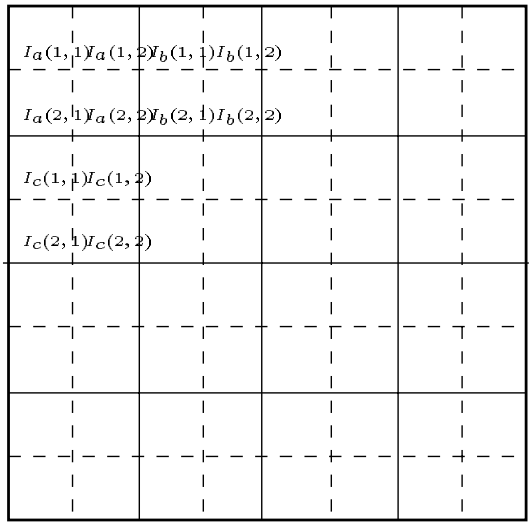


Image originale

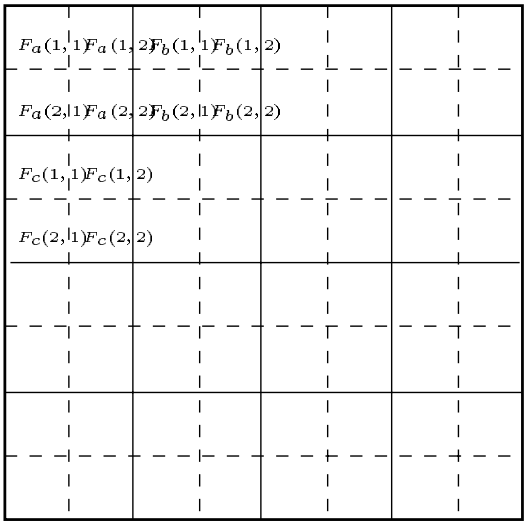


Image transformée

agencée en inter-bande

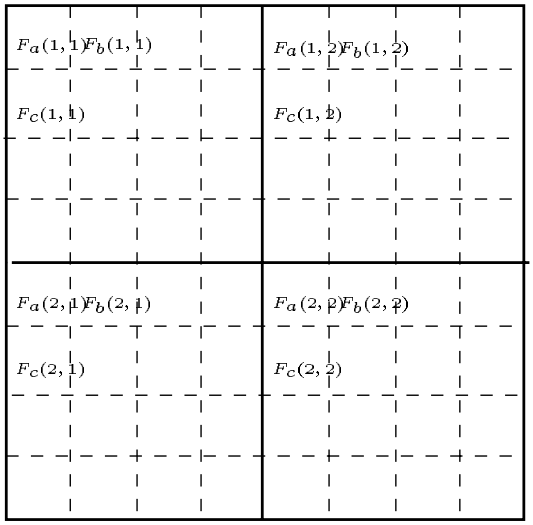


Image transformée

agencée en intra-bande

FIG. 1.5 – *Représentation inter-bande/intra-bande (exemple avec une transformée sur des blocs* 2×2*).*

perçue aussi faible que possible. Enfin la reconstruction de l'image ne doit pas entraîner de distorsion supplémentaire (qu'elle soit presque parfaite en l'absence de quantification). Les délais de calcul doivent aussi être faibles, l'ordre des filtres est donc limité ce qui empêche la construction de filtres proche du passe-bande idéal. Mais il y a possibilité de reconstruction parfaite (en l'absence de quantification), à un délai de reconstruction près, lorsque l'expression des filtres de synthèse en fonction de ceux d'analyse permet l'élimination du recouvrement des spectres. Les filtres d'analyse restent alors à déterminer. Plusieurs études ont été menées [Baaziz91], nous pouvons citer les cas où les filtres d'analyse sont : des QMF [Vetterli84] qui sont non récursifs, à phase linéaire et à supports symétriques, mais il n'y a pas reconstruction parfaite ; des CQF [Smith et al.86] qui assurent une reconstruction parfaite, la phase est quasi linéaire mais leurs supports sont non assymétriques et les calculs plus lourds ; les pseudo-QMF [Vetterli84] qui autorisent une décomposition directe en sous-bandes avec reconstruction parfaite ; les bi-orthogonaux [Antonini et al.92].
Il existe une relation entre bancs de filtres et tranformées orthogonales. Pour les distinguer nous remarquons que le codage par transformée met en oeuvre des fenêtres longues dans le domaine spatial (le domaine image) avec un faible recouvrement (*i.e.* dans le domaine fréquentiel une analyse en bande étroite est faite avec un étalement spectral important) ; inversement pour le codage en sous-bandes, les fenêtres sont courtes avec un fort recouvrement dans le domaine spatial (*i.e.* celles d'analyse dans le domaine fréquentiel sont large bande avec une bonne sélectivité).
Nous ajoutons que plusieurs stratégies de décomposition en sous-bandes sont utilisées telles que : la décomposition parallèle directe en sous-bandes (M grand) ; la découpe dyadique où la bande basse fréquence est décomposée itérativement, une représentation hiérarchique du signal est alors obtenue.

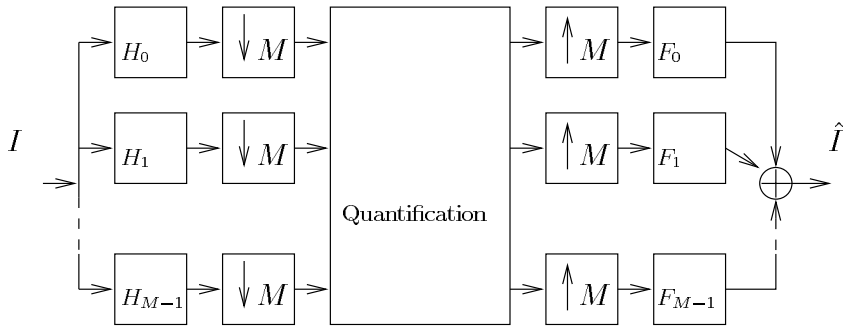


FIG. 1.6 – *Codage en sous-bandes, bancs de filtres avec quantification.*

Codage par décomposition en ondelettes

Nous rappelons que nous utiliserons la décomposition en ondelettes comme outil d'analyse avant la quantification vectorielle au chapitre 6.
La théorie des ondelettes [Meyer90] apporte une base à la spécification des filtres d'analyse/synthèse de la décomposition en sous-bandes et formalise la décomposition d'un signal

sur une base de fonctions orthogonales. La transformée d'un signal continu g est :

$$TO(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} g(x).\psi(\frac{x-b}{a})dx$$

où $\psi(x)$ (l'ondelette mère) est une fonction donnée à support compact définie pour tout $a \in \mathbb{R}^+$ et $b \in \mathbb{R}$. b caractérise l'emplacement de g sur l'axe x, et a caratérise l'échelle. Il y a adaptation de la fenêtre d'analyse car si $a > 1$, nous projettons g sur une version dilatée de ψ (pour l'analyse de phénomènes basse fréquence à variation lente), et si $a < 1$ nous projettons g sur une version contractée de ψ (analyse de phénomènes rapides haute fréquence).
En pratique les paramètres sont discrétisés avec, par exemple, $a = 2^j$ et $b = k.2^j$. Les coefficients d'ondelette sont alors calculés par :

$$c_j(k) = 2^{-j/2} \int_{-\infty}^{+\infty} g(x).\psi(2^{-j}x - k)dx$$

Un choix de fonctions ψ et ϕ avec de bonnes propriétés (sélectivités en fréquence, phase linéaire) conduit à la décomposition du signal sur des fonctions de base et à la reconstruction :

$$g(x) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} c_j(k).\phi(2^{-j}x - k)$$

g est également discrète, nous recherchons donc les réponses impulsionnelles h et f des filtres H (lié à ϕ) et F (lié à ψ) assurant la décomposition et la reconstruction de g.
Il faut aussi introduire les notions d'échelle et de résolution qui conduisent à l'analyse multirésolution du signal [Mallat89] [Daubechies90] [Baaziz91] [Rioul93]. L'**échelle** du signal original est par définition égale à 1. Si un nouveau signal est créé à partir de l'original, l'échelle se définit comme le rapport du nombre de valeurs du nouveau signal sur celui relatif à l'original (l'échelle croît si des valeurs sont ajoutées, elle diminue si elles sont supprimées). La **résolution** est introduite afin de caractériser les différentes versions d'un signal donné à la même échelle. Par convention la résolution du signal original est 1, celle de la version d'un signal obtenu par modification de l'échelle est proportionnelle au nombre d'échantillons du signal original "encore" présents (la résolution d'un signal est toujours inférieure ou égale à son échelle). La condition de biorthogonalité entre h et f est nécessaire et suffisante pour qu'une version d'un signal à une échelle et à une résolution données soit unique, la condition d'orthogonalité est que $\psi = \phi^*$ plus celle de biorthogonalité. L'**analyse multirésolution** du signal correspond à une décomposition du signal suivant des versions à différentes résolutions, la synthèse consiste à reconstruire le signal à partir de ces différentes versions. Considérons le cas simple de changements d'échelle en puissance de 2, si nous passons d'un signal à l'échelle $e = 2^{-i}$ et à la résolution $2.r = 2^{-j}$, à une résolution inférieure r (nous conservons la même échelle), le résidu est alors le signal perdu et nous avons l'égalité :

$$\text{version(e,r)} + \text{résidu(e,r)} = \text{version(e,2.r)}$$

Nous pouvons aussi parler de "signal des détails" au lieu de résidu, et de "tendance" au lieu de version.
A la transformée pyramidale du signal, il faut ajouter la condition d'échantillonnage critique pour que les nombres de valeurs entre le signal original et sa décomposition soient égaux. Cette dernière contrainte lie les fonctions de transferts des filtres d'analyse et de synthèse de façon analogue à celles permettant d'obtenir une reconstruction parfaite. La figure 1.7 présente l'exemple d'une transformée en ondelette du signal sur 2 octaves, nous obtenons simplement 2 filtres itérés en octave (le banc de filtre de la figure 1.5 avec $M = 2$) mais l'interprétation est différente: au lieu de parler de décomposition en sous-bandes, nous parlons de décomposition du signal sur des fonctions de base. La synthèse des filtres d'analyse et de synthèse est réalisée en utilisant des algorithmes d'optimisation.
L'analyse des images est effectuée par extension au cas bidimensionnel [Antonini91] [Gaidon93] à l'aide de filtres monodimensionnels séparables (découpe dyadique) ou bidimensionnels (transformée pyramidale en quinconce où le facteur de résolution est $\sqrt{2}$). La motivation essentielle de l'analyse multirésolution en codage d'image est la possibilité de traiter séparément les sous-images (*i.e.* chaque plage spatio-fréquentielle) en adaptant pour chacune la quantification en fonction de son niveau de résolution, de sa statistique et de son poids spychovisuel. En quantification vectorielle un dictionnaire multirésolution a ainsi été proposé par Antonini [Antonini91], nous reprendrons cette technique pour nos expérimentations.

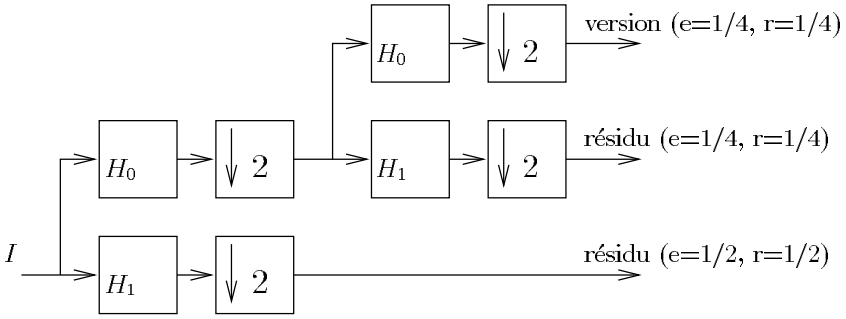


FIG. 1.7 – *Une transformées en ondelettes discrète sur 2 octaves ("e" est l'échelle et "r" la résolution).*

1.2.5 Conclusion

Nous avons montré qu'un codeur source de séquences d'images est un système composé d'un certain nombre d'outils algorithmiques de base, dont il s'agit de modéliser les propriétés afin de les assembler et réaliser le codage le plus efficace possible (en terme de compression pour une restitution satisfaisante). Jusqu'à présent nous nous sommes intéressés, tout en présentant les techniques de décorrélation que nous utiliserons dans nos codeurs expérimentaux du chapitre 6, à la description de l'approche hybride qui combine un codage inter-image (exploitant les redondances temporelles), à un autre codage intra-image des erreurs résiduelles (tirant profit des redondances spatiales subsistantes). Les

deux autres outils de base venant achever la construction du codeur de la source image sont le quantificateur, qui est l'élément fondamental [Moreau95] réalisant effectivement la compression (avec pertes), suivi du codeur entropique.
Les premières étapes de décorrélation de la séquence d'images sont nécessaires pour simplifier ce signal non-stationnaire. Mais il faut remarquer que l'erreur de prédiction demeure elle-même non-stationnaire, sa quantification ne pourra pas être rudimentaire. L'outil le plus performant est alors le quantificateur vectoriel qui n'est pas une simple généralisation du quantificateur scalaire (la supériorité du cas vectoriel sera explicitée au chapitre 2). En pratique il est indispensable de concevoir le quantificateur vectoriel en fonction de la chaîne de compression dans laquelle il prend place (même si celle-ci peut-être considérée comme une simple mise en forme du signal) et en fonction de contraintes contextuelles (*i.e.* temps des calculs, symétrisation du codage). Le dictionnaire de représentants du quantificateur vectoriel, conçu pour le codeur de séquences d'images, doit donc être très structuré avec la mise en oeuvre typique de réseaux réguliers de points ou de formes arborescentes, mais cela implique souvent de surquantifier le signal. Le codage entropique intervient alors. Il s'appuie sur les concepts fondamentaux de la théorie de l'information introduite par Shannon [Shannon48] [Shannon59] [Berger71] [Blahut87] [Gray90] (voir l'annexe B) et est généralement admis comme la base du codage de source. Cependant il ne sera considéré dans le cadre de cette étude, que comme une simple étape de réduction du débit par la mise en place d'un code présentant des mots à longueur variable (*i.e.* la longueur moyenne de code est réduite en attribuant aux vecteurs représentants les plus fréquents les mots de code les plus courts).
Après ces quelques rappels en compression de séquences d'images qui soulignent plus particulièrement l'imbrication et le rôle des outils de décorrélation du système de communication, nous pouvons à présent décrire l'objet même d'étude de notre thèse : le quantificateur vectoriel. Nous avons choisi d'engager cette description par un chapitre récapitulatif des principes et résultats fondamentaux, car le domaine de la quantification vectorielle est particulièrement vaste.

Chapitre 2

Principes généraux de la quantification vectorielle

2.1 Introduction

L'objet de ce chapitre est de rappeler les résultats fondamentaux de la quantification vectorielle. Après les définitions et principes, nous nous intéressons à des fins de comparaison au cas particulier de la quantification scalaire, puis nous établissons la supériorité théorique du cas vectoriel. Cependant cette théorie n'est pas constructive et n'explicite pas la construction du quantificateur vectoriel optimal. Une solution que nous détaillons est proposée par Linde et al. [Linde et al.80] mais elle demeure coûteuse en terme de calculs. Nous décrivons alors quelques premières approches qui visent à réduire cette complexité. Pour compléter cette présentation générale de la quantification vectorielle nous présentons d'autres formes classiques de quantificateurs. Nous donnons également dans ce chapitre des solutions proposées au problème de l'allocation des ressources binaires entre les différentes sous-bandes des images transformées.
Le lecteur familiarisé avec ces résultats établis de la quantification vectorielle peut, dès à présent, s'intéresser aux deux chapitres suivants. En effet nous avons choisi de présenter séparément et en détail les deux quantificateurs vectoriels, aux formes structurées particulièrement adaptées aux contraintes du codage d'images, qui ont le plus orientés notre recherche : la quantification vectorielle algébrique avec la mise en oeuvre de réseaux réguliers de points, et celle arborescente.

2.2 Définitions et principes

Le formalisme introduit dans cette partie sera conservé dans la suite du manuscrit.

La **quantification** consiste en l'approximation d'un signal d'amplitude continue par un signal d'amplitude discrète. La **quantification vectorielle** [Gray84] [Gersho et al.92] consiste alors à représenter tout vecteur $\mathbf{x}$ de dimension k par un autre vecteur $\mathbf{y}_i$ de même dimension mais ce dernier appartenant à un ensemble fini $\mathcal{D}$ de L vecteurs. Les

$\mathbf{y}_i$ sont appelés les **vecteurs représentants**, les vecteurs de reproduction ou les code vecteurs. $\mathcal{D}$ est appelé le **dictionnaire** ou le catalogue des formes.
Il n'y a rien de mystérieux à considérer des espaces de grandes dimensions, pour mieux appréhender les raisonnements il suffit de s'avoir que tout s'organise autour des coordonnées des vecteurs, qu'il n'y a pas lieu de s'imposer une représentation mentale géométrique. Pour l'illustrer nous précisons qu'un **vecteur** $\mathbf{x}$ de l'espace $\mathbb{R}^k$ est simplement une matrice colonne constituée de k nombres réels $x(n)$: $\mathbf{x} = (x(1), x(2), ..., x(k))^T$, et que par exemple, une sphère entièrement caractérisée par son centre $\mathbf{u} = (u(1), u(2), ..., u(n))^T$ (où $()^T$ indique la transposé) et son rayon ρ est constituée des points dont les coordonnées satisfont : $\sum_{n=1}^{k}(x(n) - u(n))^2 = \rho^2$.
Un **quantificateur vectoriel** de **dimension** k et de **taille** L peut-être défini mathématiquement comme une application $\mathcal{Q}$ de $\mathbb{R}^k$ vers $\mathcal{D}$:

$$\begin{array}{ccccc} \mathcal{Q} & : & \mathbb{R}^k & \longmapsto & \mathcal{D} \\ & & \mathbf{x} & & \mathcal{Q}(\mathbf{x}) = \mathbf{y}_i \end{array}$$

avec

$$\mathcal{D} = \left\{ \mathbf{y}_i \in \mathbb{R}^k / i = 1, 2, ..., L \right\}$$

Cette application $\mathcal{Q}$ détermine implicitement une **partition** de l'espace source $\mathbb{R}^k$ en L régions C_i. Ces régions encore appelées classes ou **régions de Voronoï** sont déterminées par :

$$C_i = \left\{ \mathbf{x} \in \mathbb{R}^k / \mathcal{Q}(\mathbf{x}) = \mathbf{y}_i \right\}$$

Les conditions suivantes sont satisfaites :

$$\bigcup_{i=1}^{L} C_i = \mathbb{R}^k \quad \text{et} \quad C_i \bigcap C_j = \emptyset, \quad i \neq j, \quad \forall i, j = 1, 2, ..., L$$

Dans le but d'alléger les notations, l'abréviation **QV** *est désormais utilisée pour désigner selon le contexte, soit un quantificateur vectoriel, soit la quantification vectorielle.*

La **compression** ou **codage** de données vise à diminuer la quantité des éléments binaires nécessaire à la représentation de l'information contenue dans le signal à transmettre ou à archiver. Cette diminution peut autoriser ou non la perte d'information [Netravali et al.88]. Une bonne compression est réalisée si nous réunissons la loi d'encodage $\mathbf{E}$ minimisant le nombre d'éléments binaires à transmettre, et la loi de décodage $\mathbf{D}$ assurant une bonne reconstruction du signal source. Lorsque $\mathbf{D}^{-1} = \mathbf{E}$, il s'agit d'un codage sans pertes d'information dit **réversible** (codage statistique ou entropique, voir l'annexe B), celui-ci exploite les redondances du signal mais ne permet pas des taux de compression élevés. Sinon $\mathbf{D}^{-1} \neq \mathbf{E}$ et il s'agit d'un codage avec pertes d'information dit **irréversible**. Cette dernière méthode où le signal décomprimé est dégradé permet des taux de compression importants ; le but est alors que les erreurs engendrées soient imperceptibles pour l'observateur. C'est dans ce cadre de compression de séquences d'images minimisant une distorsion sous la contrainte d'un débit donné que nous nous plaçons. Sous ce mot distorsion se cache une multitude de définitions possibles, la plus utilisée étant certainement la norme quadratique même si celle-ci s'avère imparfaite d'un point de vue perceptuel.

Quantification vectorielle appliquée au codage

La quantification vectorielle offre la combinaison des opérations d'encodage et de décodage (voir la figure 2.1). En effet, considérons $\mathcal{I}$ l'ensemble des L **index** des vecteurs de reproduction $\mathbf{y}_i$ du dictionnaire : $\mathcal{I} = \{1, 2, \ldots, L\}$, nous avons $\mathcal{Q} = \mathbf{D} \circ \mathbf{E}$ où la loi d'encodage est déterminée par :

$$\begin{array}{rccc} \mathbf{E} & : & \mathbb{R}^k & \longmapsto & \mathcal{I} \\ & & \mathbf{x} & & \mathbf{E}(\mathbf{x}) = i \end{array}$$

et le processus de décodage est défini par :

$$\begin{array}{rccc} \mathbf{D} & : & \mathcal{I} & \longmapsto & \mathcal{D} \\ & & i & & \mathbf{D}(i) = \mathbf{y}_i \end{array}$$

Encodage

La procédure d'encodage consiste, pour tout vecteur $\mathbf{x}$ du signal d'entrée, à rechercher dans le dictionnaire D le code-vecteur $\mathbf{y}_i$ le plus proche de $\mathbf{x}$. Nous introduisons ici la définition générale de la **métrique** L_p, où pour un vecteur $\mathbf{x}$ de dimension k :

$$L_p(\mathbf{x}) = \sum_{n=1}^{k} |x(n)|^p$$

La distance entre deux vecteurs $\mathbf{x}_1$ et $\mathbf{x}_2$ est alors :

$$(L_p(\mathbf{x}_1 - \mathbf{x}_2))^{1/p} = d_p(\mathbf{x}_1, \mathbf{x}_2) = \left(\sum_{n=1}^{k} |x_1(n) - x_2(n)|^p \right)^{1/p}$$

La notion de proximité que nous avons choisie de mettre en place pour mesurer la **distorsion** entre les vecteurs $\mathbf{x}$ et $\mathbf{y}_i$ est la distance euclidienne (c'est le cas particulier de L_p où $p = 2$) :

$$d(\mathbf{x}, \mathbf{y}_i) = d_2(\mathbf{x}, \mathbf{y}_i) = \left(\sum_{n=1}^{k} (x(n) - y_i(n))^2 \right)^{1/2} \tag{2.1}$$

Les régions de Voronoï sont alors données par (voir l'exemple de la figure 2.2) :

$$C_i = \left\{ \mathbf{x} \in \mathbb{R}^k / \mathcal{Q}(\mathbf{x}) = \mathbf{y}_i, \; si \; d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j), \forall j \neq i \right\} \tag{2.2}$$

Chaque région contient un ensemble de vecteurs de $\mathbb{R}^k$ et tous les vecteurs $\mathbf{x}$ qui appartiennent à C_i sont représentés par le même vecteur $\mathbf{y}_i$ du dictionnaire. La compression d'information est réalisée à ce niveau car c'est uniquement l'index relatif au représentant $\mathbf{y}_i$ minimisant le critère de distorsion qui sera transmis ou stocké. Cette opération créant des pertes d'information transforme la QV en une opération irréversible, le signal quantifié

ne pourra plus être restitué identique à l'original.
La quantité d'information requise pour représenter les vecteurs source est donnée par R le **débit binaire** ou résolution, en bit par composante du vecteur ou bit par dimension ou encore **bit par échantillon** ([bpp]). Ce débit peut-être apprécié de deux façons :

- soit il y a contrainte de "débit fixe" et un code à longueur fixe (dit aussi "code naturel") est adopté. La QV permet alors des résolutions fractionnaires (*i.e.* R n'a pas besoin d'être entier, seul $R.k$ l'est), car :

$$R = \frac{1}{k}.\log_2 L \tag{2.3}$$

- soit il y a contrainte d'"entropie fixe". Un code à longueur variable (voir l'annexe B) sera donc construit après quantification afin de réduire encore le débit. R peut-être estimé en calculant l'entropie du dictionnaire $H(\mathcal{D})$ (*e.g.* l'entropie associée aux représentants) et il est bien sûr admis que le débit binaire réel obtenu sera de quelques pour-cent supérieur. Nous accédons aux probabilités d'occurence p_i des code-vecteurs $\mathbf{y}_i$ en modélisant la statistique de la source, ou en procédant à l'aide d'une base d'apprentissage constituée d'un grand nombre d'échantillons représentatifs de la source, alors :

$$R \simeq H(\mathcal{D}) = -\frac{1}{k}\sum_{i=1}^{L} p_i.\log_2(p_i)$$

Décodage

Le décodeur est considéré comme un récepteur voué à la reconstruction du vecteur source, pour cela il dispose d'une réplique du dictionnaire qu'il consulte afin de restituer le code vecteur correspondant à l'index qu'il reçoit. Le décodeur réalise donc l'opération de décompression.

Evaluation des performances du système

Le but du système de codage est de réaliser le meilleur compromis entre le coût du codage et l'erreur faite. Pour simplifier nous pouvons dire qu'il s'agit d'obtenir avec un débit minimal, une distorsion moyenne caractérisant les performances globales du QV, elle aussi minimale. Il importe surtout que la mesure de distorsion mise en oeuvre traduise la dégradation subjective faite au signal. Seule une mesure traduisant les caractéristiques perceptuelles humaines peut apporter un tel résultat [Netravali et al.88] [Watson93]. Précisément la mesure de distorsion doit réunir trois conditions essentielles [GJ et al.76] telles que cette fonction réelle positive soit simple à calculer, utilisable par un algorithme de minimisation et significative pour qu'une grande distorsion implique une mauvaise qualité de la restitution subjective du signal (et inversement). Il est clair que des mesures objectives de qualité subjective sont très difficiles à concevoir et nous nous sommes contentés d'utiliser une erreur quadratique moyenne (EQM). Cette mesure de distorsion moyenne est

la plus répandue et s'interprète comme la puissance de l'erreur de quantification, elle est souvent retenue car elle vérifie les deux premières conditions cependant elle ne caractérise que pauvrement la dégradation qualitative faite au signal. La troisième condition serait obtenue au prix de l'introduction d'une pondération psychovisuelle adaptée au codage des séquences d'images.
Par la suite, seules la décomposition espace-fréquence du signal suivie de la stratégie d'allocation binaire attribuant les bits disponibles aux sous-bandes les plus importantes visuellement, permettront la prise en compte de cet aspect psychovisuel au codage tout en utilisant la distance euclidienne. Cette dernière est même jugée intéressante car l'EQM dans les sous-bandes correspond à l'EQM après reconstruction, ce qui évite de faire la synthèse pour la connaître (mais les erreurs avant et après reconstruction ne sont égales que si les transformées ou décomposition en sous-bandes sont orthogonales).

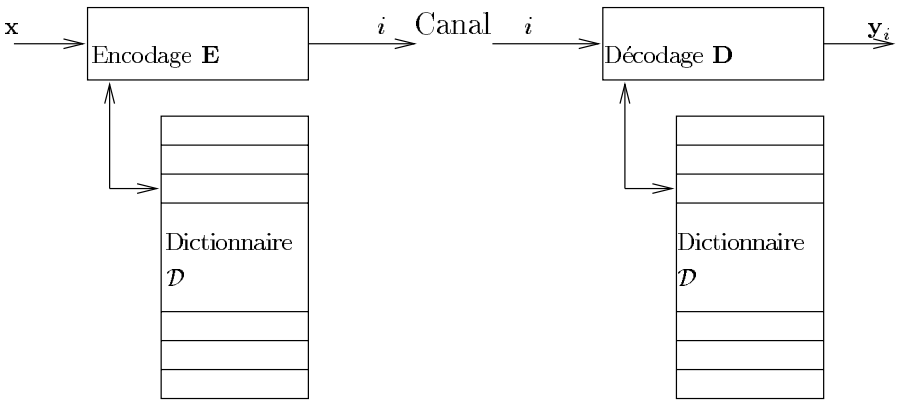


FIG. 2.1 – *Schéma du quantificateur vectoriel.*

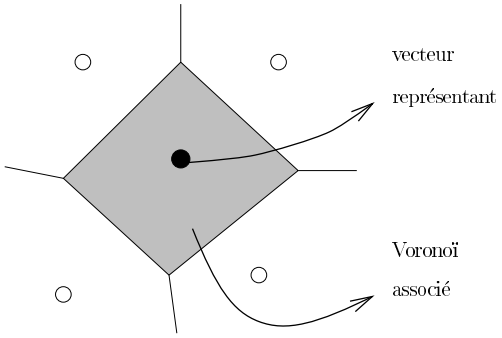


FIG. 2.2 – *Principe de la quantification vectorielle.*

2.3 Quantification scalaire

2.3.1 Introduction

La **quantification scalaire** [Gersho et al.92] est une forme particulière de la QV, celle où la dimension des vecteurs est égale à un. La figure 2.3 illustre la caractéristique en marche d'escalier du plus simple des QS, celui uniforme à débit fixe qui est entièrement déterminé par :

- les $L+1$ **niveaux de décisions** : $d_0, d_1, ..., d_L$, qui partitionnent en L intervalles égaux l'axe des réels $\mathbb{R}$ et déterminent le pas de quantification ;
- les L **valeurs de reproduction** : $y_1, y_2, ..., y_L$, qui sont les centres de masse de chacun des intervalles de décision.

Cet exemple bien connu permet d'introduire les différents bruits ou erreurs de quantification rencontrés :

- le **bruit granulaire** qui se produit ici lorsque la valeur d'entrée x se situe dans l'une des cellules $[d_{i-1}, d_i]$, l'erreur résultante est la différence entre x et $\mathcal{Q}(x)$. Elle peut être majorée par un demi pas de quantification ;
- le **bruit de surcharge** ou de dépassement qui se produit ici lorsque la valeur d'entrée se situe hors de l'intervalle $[d_0, d_L]$. La valeur de reproduction est alors soit y_1 soit y_L, et l'erreur résultante peut-être supérieure à un demi pas de quantification.

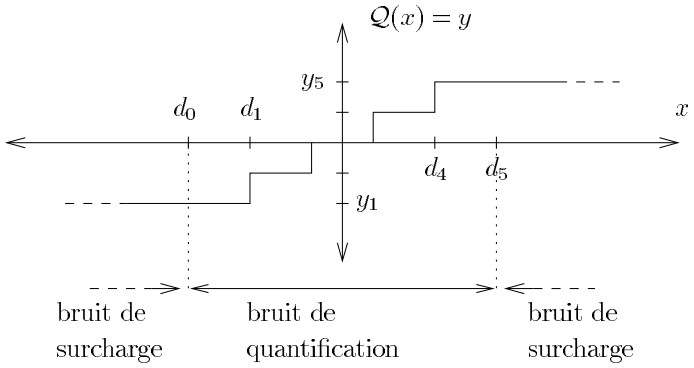


FIG. 2.3 – *Exemple d'un QS uniforme pour* $L = 5$.

2.3.2 Quantification scalaire optimale

Le QS optimal est celui qui minimise, pour une source donnée et sous la contrainte d'un débit maximal fixé ou d'une entropie maximale fixée, l'erreur moyenne de reconstruction due aux bruits de quantification. Les niveaux de reconstruction sont donc répartis en tenant compte de la densité de probabilité de la variable à quantifier.

Contrainte de débit fixe

Dans ce cas nous comprenons intuitivement que les seuils de décision sont plus concentrés dans la zone de l'espace où la densité de probabilité des vecteurs à quantifier est plus élevée. Précisément, en interprétant les échantillons x comme la réalisation d'un processus aléatoire stationnaire centré X de densité de probabilité marginale $p_X(x)$, nous voulons minimiser l'EQM (où E signifie l'espérance mathématique) :

$$\begin{aligned} D &= E\left((X - \mathcal{Q}(x))^2\right) \\ &= \int_{-\infty}^{+\infty} (x - \mathcal{Q}(x))^2 . p_X(x)\, dx \\ &= \sum_{i=1}^{L} \int_{d_{i-1}}^{d_i} (x - y_i)^2 . p_X(x)\, dx \end{aligned}$$

D est aussi la variance σ_Q^2 (car le processus est centré) de l'erreur de quantification. Il s'agit de trouver les partitions $[d_{i-1}, d_i]$ et les représentants y_i minimisant D. Cette minimisation conjointe n'admettant pas de solution simple a une solution unique, et le QS obtenu est connu sous le nom de **quantificateur de Lloyd-Max** [Lloyd82] [Max60]. Il n'existe que deux conditions nécessaires d'optimalité qui ne sont pas suffisantes (sauf dans le cas d'une variable gaussienne), il faut réunir :

- la meilleure partition (qui n'a pas à être connue explicitement), elle vérifie la "règle du plus proche voisin" explicitée par la formule 2.2 avec connaissance de la mesure de distorsion ;
- les meilleurs représentants vérifiant la condition dite "du centroïde" (ou centre de gravité) car la valeur choisie pour un représentant est la valeur moyenne dans l'intervalle considéré.

La partie encodage du quantificateur (la partition) doit alors être optimale étant donnée la partie décodage (les meilleurs représentants) et réciproquement. Le calcul montre que nous obtenons :

$$\begin{aligned} y_i &= \frac{\int_{d_{i-1}}^{d_i} x . p_X(x)\, dx}{\int_{d_{i-1}}^{d_i} p_X(x)\, dx} \quad \text{avec} \quad 1 \leq i \leq L \\ d_i &= \frac{y_i + y_{i+1}}{2} \quad \text{avec} \quad 1 \leq i \leq L-1 \\ d_0 &= -\infty \ \text{et} \ d_L = +\infty \end{aligned}$$

En pratique la densité de probabilité du signal est inconnue. L'algorithme de Lloyd-Max procède donc par apprentissage en utilisant des données empiriques ayant toutes le même poids (la base ou séquence d'apprentissage doit évidemment être suffisamment grande). Nous donnons dans la suite de ce chapitre une description de l'algorithme de Lloyd généralisé aux vecteurs de dimensions supérieures, nous indiquons seulement à ce niveau qu'il

s'agit d'un procédé itératif vérifiant successivement les deux conditions d'optimalité.

Dans le cadre de l'**hypothèse dite haute résolution** qui consiste à admettre que le nombre L de niveaux de quantification est très élevé (*i.e.* la partie de la densité de probabilité du processus contenue dans chaque élément de la partition de Voronoï est approximativement constante), il est possible d'obtenir explicitement l'expression de la puissance de l'erreur de quantification uniquement en fonction de $p_X(x)$ avec :

$$\sigma_Q^2 = \frac{1}{12}\left(\int_{-\infty}^{+\infty}(p_X(x))^{1/3}\,dx\right)^3.2^{-2.R} = \frac{1}{12}.||p_X(x)||_{1/3}.2^{-2.R}$$

Pour une source stationnaire gaussienne centrée de variance σ_X^2 nous obtenons :

$$\sigma_Q^2 = c_1.\sigma_X^2.2^{-2.R} \text{ avec } c_1 = \frac{\sqrt{3}}{2}.\pi \tag{2.4}$$

Ces dernières équations vont nous servir de références pour les comparaisons entre quantificateurs. Nous pouvons ajouter que des études [Gersho et al.92] ont montré qu'un QS non-uniforme est équivalent à un QS uniforme précédé d'une transformation non linéaire et suivi de la transformation inverse.

Contrainte d'entropie fixe

Ce nouveau mode de quantification appelé "quantification avec contrainte entropique" vise à minimiser D sous la contrainte que l'entropie du dictionnaire soit inférieure à R (L est inconnu) :

$$H(\mathcal{D}) \leq R \tag{2.5}$$

En nous plaçant dans l'hypothèse haute-résolution et en considérant le quantificateur de Lloyd-Max sous contrainte entropique, l'analyse théorique montre que le QS optimal est tout simplement un QS uniforme suivit d'un codage entropique [Berger82] [Favardin et al.84], alors :

$$\sigma_Q^2 = \frac{1}{12}.2^{2.h(X)}.2^{-2.R}$$

où $h(X)$ est l'entropie différentielle qui caractérise la quantité d'information que possède une source continue sans mémoire (voir l'annexe B). Dans le cas d'une source stationnaire gaussienne centrée $h(X) = 1/2.\log_2(2.\pi.e.\sigma_X^2)$, donc :

$$\sigma_Q^2 = \frac{\pi.e}{6}.\sigma_X^2.2^{-2.R}$$

Le gain apporté par le quantificateur avec contrainte entropique relativement au QS de Lloyd-Max est de $(\sqrt{3}.\pi/2)/(\pi.e/6) \simeq 1,91$ soit 2.81 dB. Ce résultat est à rapprocher de la valeur asymptotique $\sigma_Q^2 = \sigma_X^2.2^{-2.R}$ donnée par la borne de Shannon (la limite théorique de la fonction débit-distorsion, ce résultat est présenté à l'annexe B). Le QS avec contrainte entropique a une puissance d'erreur de quantification égale à $(\pi.e)/6 \simeq 1.42$ fois cette limite, il est à $(1/2).log_2(\pi.e/6) \simeq 0,25$ bit de la limite théorique.

2.3.3 Quantification scalaire prédictive

L'objet de la quantification scalaire prédictive est de chercher à décorréler le signal, de le simplifier, avant de le quantifier. La figure 1.3 illustre l'exemple d'un tel quantificateur en boucle fermée, mais au lieu des symboles I, I_p, I_r, e et e_q qui représentent des images, nous considérons respectivement x, x_p, x_r, e et e_q qui sont des échantillons indicés par n (*e.g.* un balayage de type "raster-scan" de l'image est effectué).
Nous remarquons que le décodeur simule le codeur dans la boucle de prédiction et que celle-ci est faite de façon causale ("backward") à partir des valeurs reconstruites, ainsi il n'y a pas besoin de transmettre d'autres informations que l'erreur de prédiction. Cependant cette prédiction ne peut s'appuyer sur des échantillons trop dégradés et le débit ne peut-être trop bas. Cette technique est à opposer à celle de prédiction en avant à partir des échantillons originaux qui impliquerait la transmission d'une information supplémentaire au décodeur. En principe il faut distinguer l'erreur de quantification $q(n) = e(n) - e_q(n)$ de l'erreur de reconstruction $\bar{q}(n) = x(n) - x_r(n)$ car ce qui nous intéresse est la puissance de l'erreur de reconstruction, mais la propriété fondamentale du quantificateur prédictif est d'avoir l'égalité $q(n) = \bar{q}(n)$.
Une prédiction linéaire de x en utilisant P échantillons du passé (P est l'ordre de la prédiction), s'exprime sous la forme :

$$x_p(n) = -\sum_{i=1}^{P} a_i.x_r(n-i)$$

C'est une opération de filtrage et il faut déterminer les coefficients a_i minimisant la distorsion moyenne entre x et x_p, nous pouvons choisir de minimiser la puissance de l'erreur de prédiction $E\left((X(n) - x_p(n))^T(X(n) - x_p(n))\right)$ relativement aux a_i (les $\{x(n)\}$ étant supposés des réalisations du processus aléatoire stationnaire $\{X(n)\}$). Il existe tout un développement théorique associé à la prédiction linéaire qui aboutit à l'équation normale ou de Yule-Walker ou de Wiener-Hopf :

$$\Gamma_X(P).\,(a_1, a_2, \ldots, a_P)^T = -\,(\rho_1, \rho_2, \ldots, \rho_P)^T$$

où les ρ_i sont les coefficients normalisés de la fonction d'autocorrélation :

$$\rho_i = \frac{E\left(X(n).X(n-i)\right)}{E(X^2(n))}$$

et $\Gamma_X(P)$ la matrice d'autocorrélation normalisée à l'ordre P du processus aléatoire X. Les paramètres a_i optimaux sont donnés par l'inversion de Γ_X. Pour calculer ce prédicteur optimal à l'ordre P (*i.e.* résoudre le système linéaire à P équations et P inconnues), nous pouvons utiliser les algorithmes de Gauss ou ceux plus rapides de Linvinson ou de Schur [Gersho et al.92].
Si nous considérons une source stationnaire gaussienne centrée et une quantification optimale avec R [bpp] (le code est à longueur fixe), nous obtenons :

$$\sigma_Q^2 = \sigma_{\bar{Q}}^2 = \frac{c_1.\sigma_X^2.2^{-2.R}}{G_p(P)} \tag{2.6}$$

Cette dernière équation montre que la puissance de l'erreur de quantification est réduite de :

$$G_p(P) = \frac{\sigma_X^2}{\sigma_Q^2}$$

C'est le gain de prédiction qui est une fonction croissante de P et tend vers la limite $G_p(\infty)$. Ce gain mesure l'amélioration apportée par la quantification de l'erreur de prédiction plutôt que la quantification directe du signal (si le processus est blanc $G_p(\infty) \to 1$, si le processus est totalement prédictible $G_p(\infty) \to +\infty$), sa valeur asymptotique s'exprime en fonction du déterminant de Γ_X avec :

$$G_p(\infty) = \lim_{P \to +\infty} \frac{1}{(\det \Gamma_X(P))^{1/P}} \tag{2.7}$$

ou de la densité spectrale S_X du processus (voir l'annexe B, l'inverse de cette valeur porte aussi le nom de mesure d'étalement spectral). En fait la puissance de prédiction décroît rapidement puis reste pratiquement constante à partir d'un certain ordre P_0 (*e.g.* pour un processus autorégressif d'ordre P_0) au delà duquel le signal n'est plus corrélé, la prédiction ne peut donc être améliorée. Comme les signaux ne sont jamais stationnaires, il faut aussi actualiser les coefficients a_i à intervalles réguliers.

Pour une source quelconque, le gain de prédiction est toujours calculé comme le rapport de la variance du signal d'entrée sur celle du signal d'erreur et indique le gain théorique maximal possible. En pratique la modélisation d'une image est faite par un champ de Markov à deux dimensions, mais l'hypothèse de stationnarité n'est jamais vérifiée. De plus les coefficients calculés comme précédemment sont optimaux au sens de l'EQM, mais ils ne minimisent pas l'entropie. Aussi les schémas de codage différentiel (*i.e.* les codeurs MICD ou DPCM) tenant compte de la corrélation existante entre les valeurs d'intensité de pixels voisins, s'appuient le plus souvent sur un simple prédicteur fixe comme celui de la figure 2.4 où le calcul de la prédiction du pixel courant $I(2,2)$ est une combinaison linéaire des pixels voisins $I(1,1)$, $I(1,2)$ et $I(2,1)$. Pour le codage en sous-bandes, cette technique est adaptée pour le codage des basses fréquences pour lesquelles les coefficients transformés sont plus fortement corrélés. Il est évident que le gain de prédiction réalisé est loin de celui annoncé par la théorie.

Nous venons de préciser les performances qu'il faut attendre de la QS. L'analyse qui suit va démontrer que quelles que soient les caractéristiques de la source, la QV permet d'obtenir un signal reconstruit de meilleure qualité que dans le cas scalaire.

2.4 Supériorité de la quantification vectorielle sur celle scalaire

2.4.1 Résultats de Zador

La théorie de l'information annonce d'une façon générale que de meilleurs résultats en codage de source sont obtenus si des vecteurs sont quantifiés plutôt que des scalaires

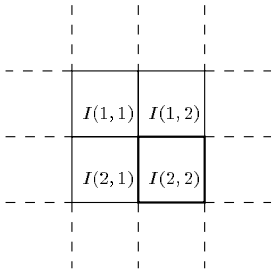


FIG. 2.4 – *Exemple d'un voisinage utilisé en prédiction linéaire.*

(voir l'annexe B). En contre partie une complexification des systèmes ainsi que des délais de calcul plus importants sont nécessaires. Ce gain de la QV sur la QS est notamment du à l'exploitation de la mémoire de la source (*i.e.* des corrélations existantes entre les coordonnées des vecteurs). Zador [Zador82] a théoriquement prouvé la supériorité de la QV sur la QS même si les échantillons de la source sont statistiquement indépendants.

En reprenant les notations de la partie 2.2, nous précisons la définition de l'EQM [par dimension] dans le cas vectoriel où $p_{\mathbf{X}}(\mathbf{x})$ est alors la densité de probabilité conjointe du vecteur $\mathbf{x}$ (un code à longueur fixe est supposé construit) :

$$D = \frac{1}{k} \int_{\mathbb{R}^k} d^2(\mathbf{x}, \mathbf{y}_i).p_{\mathbf{X}}(\mathbf{x}) \; d\mathbf{x} = \frac{1}{k} \sum_{i=1}^{L} \int_{C_i} d^2(\mathbf{x}, \mathbf{y}_i).p_{\mathbf{X}}(\mathbf{x}) \; d\mathbf{x} \tag{2.8}$$

Nous recherchons le quantificateur optimal (*i.e.* les vecteurs représentants $\mathbf{y}_i$) qui pour k, L, et $p_{\mathbf{X}}(\mathbf{x})$ donnés minimisent D :

$$D(k, L, p_{\mathbf{X}}) = \min_{\mathbf{y}_i} D$$

$D(k, L, p_{\mathbf{X}})$ est l'erreur minimale qui peut-être atteinte. Zador a démontré qu'il est possible de réduire D en quantifiant des vecteurs de grandes dimensions. Exactement il a prouvé, en adoptant l'hypothèse haute résolution, que :

$$\lim_{L \to +\infty} L^{\frac{2}{k}}.D(k, L, p_{\mathbf{X}}) = G_k.(\int_{\mathbb{R}^k} p_{\mathbf{X}}(\mathbf{x})^{\frac{k}{k+2}} \; d\mathbf{x})^{\frac{k+2}{k}} = G_k.||p_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}}$$

que nous pouvons réécrire (sachant $L = 2^{R.k}$ avec l'équation 2.3) :

$$D(k, +\infty, p_{\mathbf{X}}) = \lim_{L \to +\infty} D(k, L, p_{\mathbf{X}}) = G_k.||p_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}}.2^{-2.R}. \tag{2.9}$$

La répartition optimale des vecteurs de reproduction est donc celle proportionnelle à $p_{\mathbf{X}}(\mathbf{x})^{\frac{k}{k+2}}$. Le paramètre G_k ne dépend que de la dimension k (il dépend aussi du choix de la métrique), et Zador a montré qu'il décroît lorsque k augmente. Si $k \to +\infty$: $G_k \to \frac{1}{2.\pi.e} \simeq 0.05855$ (soit un gain maximal par rapport à la quantification scalaire de $1,53$ dB).

Interprétation et calcul de G_k

$$G_k = \frac{\lim_{L\to+\infty} L^{\frac{2}{k}}.D(k,L,p_{\mathbf{X}})}{||p_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}}}$$

Nous considérons le cas simple où les $\mathbf{x}$ sont uniformément distribués sur une grande région de $\mathbb{R}^k$ (sur une hyperboule). Alors sur cet espace :

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{\sum_{i=1}^{L}\int_{c_i} d\mathbf{x}}$$

D est minimale si chaque $\mathbf{y}_i$ est au centre de son Voronoï, et en considérant l'hypothèse haute résolution (*i.e.* L très grand, il n'y a pas de problème de bord) :

$$D = \frac{1}{k}.\frac{\sum_{i=1}^{L}\int_{c_i} d^2(\mathbf{x},\mathbf{y_i}).d\mathbf{x}}{\sum_{i=1}^{L}\int_{c_i} d\mathbf{x}} \tag{2.10}$$

Nous calculons :

$$||p_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}} = (\int_{\mathbb{R}^k} p_{\mathbf{X}}(\mathbf{x})^{\frac{k}{k+2}}\ d\mathbf{x})^{\frac{k+2}{k}} = (\sum_{i=1}^{L}\int_{C_i} p_{\mathbf{X}}(\mathbf{x})^{\frac{k}{k+2}}\ d\mathbf{x})^{\frac{k+2}{k}} = (\sum_{i=1}^{L}\int_{c_i} d\mathbf{x})^{\frac{2}{k}}$$

Finalement :

$$G_k = \lim_{L\to+\infty} \frac{D(k,L,p_{\mathbf{X}})}{(\frac{1}{L}.\sum_{i=1}^{L}\int_{C_i} d\mathbf{x})^{\frac{2}{k}}} \tag{2.11}$$

En considérant la quantification optimale d'une source uniforme et des conditions asymptotiques, G_k qui dépend de la forme du Voronoï, s'interprète comme le rapport de l'erreur quadratique moyenne [par dimension] sur un facteur le rendant sans dimension. Zador a donc montré que cette erreur peut être réduite en utilisant k grand.

Gersho [Gersho79] a montré sous certaines conditions (source uniforme, haute résolution) que l'erreur quadratique par dimension est donnée par le moment d'inertie d'ordre 2 du Voronoï. En effet, en considérant l'équation 2.11 et en notant $\mathrm{vol}(C_i) = \int_{C_i} d\mathbf{x}$:

$$G_k = \lim_{L\to+\infty} \frac{1}{k}.\frac{\frac{1}{L}.\sum_{i=1}^{L}\int_{C_i} d^2(\mathbf{x},\mathbf{y}_i)\ d\mathbf{x}}{(\frac{1}{L}.\sum_{i=1}^{L}\mathrm{vol}(C_i))^{1+\frac{2}{k}}}$$

La source considérée étant uniforme, nous adoptons le cas où les vecteurs de reproduction sont les points d'un réseau régulier. Les Voronoï sont alors congrues au même polytope Δ, alors :

$$G_k(\Delta) = \frac{1}{k}.\frac{\int_{\Delta} d^2(\mathbf{x},0)\ d\mathbf{x}}{\mathrm{vol}(\Delta)^{1+\frac{2}{k}}}$$

Nous retrouvons le moment d'inertie normalisée d'ordre 2 de Δ. Il est alors possible de déterminer dans certains cas particuliers la forme optimale du polytope qui minimise

$G_k(\Delta)$ et donc la distorsion sur tout l'espace. Par exemple si $k = 1$, les représentants sont uniformément distribués sur la droite réelle, nous construisons le réseau où les Voronoï sont des intervalles de longueur 1, Δ est l'intervalle $[-1/2, +1/2]$ (c'est le réseau $\mathbf{Z}$ du chapitre 3). Nous obtenons :

$$G_1(\Delta) = \frac{\int_{-1/2}^{+1/2} x^2 \, dx}{\int_{-1/2}^{+1/2} dx} = \frac{1}{12} \simeq 0.08333$$

Ce résultat est classique en quantification : si une variable uniforme est quantifiée scalairement, l'erreur quadratique moyenne est 1/12. Pour $k = 2$, le polytope optimal est hexagonal, et $G_2(\Delta) = \frac{5}{36.\sqrt{3}} \simeq 0.080175$. Lorsque la dimension croît encore, les formes des polytopes se rapprochent de celle sphérique (même si il est impossible de recouvrir un espace à l'aide de sphères sans provoquer des trous ou des recouvrements). En effet, la sphère S (centrée à l'origine et de rayon unitaire) a le moment d'inertie le plus faible, ce dernier représente alors une borne minimale appelée "sphere bound" : $G_k(\Delta) \geq \frac{1}{k+2}.\mathrm{vol}(S)^{-2/k}$ avec $\mathrm{vol}(S) = \frac{\Pi^{k/2}}{(k/2)!}$.
Nous nous contentons de préciser que Conway et Sloane ont précisé, lors de l'étude des réseaux réguliers de points, une borne plus précise appelée "Conway and Sloane conjecture bound". Des valeurs de G_k ont été tabulées [Conway et al.82b] [Conway et al.85] [Conway et al.93].

Puissance de l'erreur de quantification dans le cas d'un processus gaussien

Pour un processus gaussien centré :

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{\left(2.\pi.\sigma_X^2\right)^{k/2}.\sqrt{\det(\Gamma_X(k))}}.\exp\frac{-\mathbf{x}^T.\Gamma_X(k)^{-1}.\mathbf{x}}{2.\sigma_X^2}$$

soit :

$$||p_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}} = 2.\pi.\left(\frac{k+2}{k}\right)^{\frac{k+2}{2}}.\sigma_X^2.\left(\det(\Gamma_X(k))\right)^{1/k} \tag{2.12}$$

En reprenant l'équation 2.9 :

$$\sigma_Q^2 = G_k.2.\pi.\left(\frac{k+2}{k}\right)^{\frac{k+2}{2}}.\sigma_X^2.\left(\det(\Gamma_X(k))\right)^{1/k}.2^{-2.R} = c_k.\sigma_X^2.\left(\det(\Gamma_X(k))\right)^{1/k}.2^{-2.R} \tag{2.13}$$

avec :

$$c_k = G_k.2.\pi.\left(\frac{k+2}{k}\right)^{\frac{k+2}{2}}$$

c_k est donc une fonction décroissante avec k, pour $k = 1$ nous retrouvons le résultat de l'équation 2.4, pour $k \to +\infty$: $c_k = 1$.
L'équation 2.13 est à rapprocher de l'équation 2.6 obtenue avec le QS prédictif (il faut aussi

considérer l'équation 2.9 donnant $G_P(\infty)$), car si nous considérons les cas asymptotiques d'une prédiction d'ordre infini pour le cas scalaire (*i.e.* $P \to +\infty$) et d'une QV à l'aide de vecteurs de taille infinie (*i.e.* $k \to +\infty$), le gain apporté par la QV relativement au QS est égal à $c_1 = 4,35$ dB. Ce premier résultat souligne la supériorité du QV capable de tirer profit de toutes les dépendances existantes au sein du signal.

2.4.2 Gains de la quantification vectorielle sur celle scalaire

Les résultats de Zador établissent de façon générale la supériorité de la QV sur la QS car elle permet d'obtenir un signal mieux reconstruit (dans le cas de sources identiques). A partir de l'expression donnée par Zador (l'équation 2.9), Lookabaugh et Gray [Lookabaugh et al.89] ont alors distingué les différents gains théoriques déterminant cette supériorité de la QV. En définissant le gain QV/QS : $G_V(k)$, comme étant le rapport sous l'hypothèse haute résolution de la distorsion de la QS sur celle de la QV, ils sont parvenus à identifier trois facteurs ou gains indépendants :

$$G_V(k) = \frac{D(1,+\infty,p_X)}{D(k,+\infty,p_{\mathbf{X}})} = \frac{G_1}{G_k}.\frac{||p_X(x)||_{\frac{1}{3}}}{||p_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}}} = \frac{G_1}{G_k}.\frac{||p_X(x)||_{\frac{1}{3}}}{||p^*_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}}}.\frac{||p^*_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}}}{||p_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}}}$$

où $p^*_{\mathbf{X}}(\mathbf{x})$ est la densité de probalité conjointe de la source supposée sans mémoire, c'est donc le produit des densités marginales :

$$p^*_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^{k} p_X(x_i) \tag{2.14}$$

Nous distinguons :

- le gain de partitionnement ou de structure ("space filling advantage") :

$$G_P(k) = \frac{G_1}{G_k}$$

- le gain de forme ou de distribution ("shape advantage") :

$$G_F(k) = \frac{||p_X(x)||_{\frac{1}{3}}}{||p^*_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}}}$$

- le gain en mémoire ("memory advantage") :

$$G_M(k) = \frac{||p^*_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}}}{||p_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}}}$$

Gain de partitionnement

$G_P(k)$ caractérise la façon dont le quantificateur couvre l'espace à l'aide des Voronoï (il est indépendant des caractéristiques de la source). Alors que des Voronoï rectangulaires correspondent nécessairement à k quantificateurs scalaires, le QV autorise un libre choix dans la forme des Voronoï et permet une optimisation du partitionnement de l'espace (il répond au problème d'empilement de sphères dans un espace qui sera présenté au chapitre 3), et donc une diminution de la distorsion moyenne. L'explication réside dans l'interprétation de G_k que nous avons faite précédemment.
Il est important de noter que le gain de structure espéré de la QV sur la QS reste peu significatif (*e.g.* $G_P(10) = 0,75$ dB, $G_P(+\infty) = \frac{1}{2.\pi.e} \simeq 0,05855 = 1,5$ dB).

Gain de forme

Ce gain est directement lié à la forme de la densité de probabilité marginale de la source, car (pour la source sans mémoire) :

$$||p^*_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}} = \left(\int_{\mathbb{R}^k} \left(\prod_{i=1}^{k} p_X(x_i) \right)^{\frac{k}{k+2}} dx_i \right)^{\frac{k+2}{k}} = \left(\int_{\mathbb{R}} \left(p_X(x)^{\frac{k}{k+2}} \, dx \right)^k \right)^{\frac{k+2}{k}}$$

La propriété d'équirépartition développée au chapitre 3, souligne que la distribution de vecteurs équiprobables d'une source sans mémoire se fait approximativement dans un volume déterminé de l'espace. Ce volume est fonction du produit des densités de probabilité marginales et est d'autant plus compact que la source suit une loi "étroite" (*e.g.* c'est un hypercube si la source est uniforme, une hyperboule si la source est gaussienne, et une hyperpyramide si elle est laplacienne).
Afin de minimiser la distorsion la QV peut tenir compte de cette distribution spatiale des vecteurs (alors que le QS ne peut qu'attribuer des niveaux de décisions sur l'axe réel), tout en adaptant le volume des Voronoï à la densité vectorielle (ce qui correspond à l'adaptation des pas de quantification du QS). A titre d'exemple nous indiquons que le gain de forme pour une source uniforme est unitaire (*i.e.* il n'y a aucun avantage par rapport à la QS), pour un processus gaussien centré $G_F(k) = \left(3^{3/2}\right) / \left(\frac{k+2}{k}\right)^{\frac{k+2}{k}}$ donc $G_F(10) = 2{,}4$ dB, $G_F(+\infty) = 2{,}81$ dB.
Ce gain est d'autant plus élevé que la source suit une loi fine, mais il n'enregistre qu'une faible augmentation au-delà de la dimension 8.

Gain en mémoire

Ce gain est du aux dépendances linéaires (*i.e.* les corrélations) et non-linéaires existantes entres les différentes composantes des vecteurs. En effet ces dépendances se traduisent par une distribution des vecteurs dans un volume plus ou moins allongé dans l'espace, ce volume allongé est d'autant plus étroit que la dépendance entre les coordonnées des vecteurs est grande [Moureaux94]. Nous rappelons l'exemple de variables gaussiennes

qui se répartissent dans une hyperboule si elles sont indépendantes, et dans une hyper-ellipse lorsque la corrélation entre les échantillons augmente (ce volume, à un coefficient près, est égal à $(\Gamma_X(k))^{1/k}$).
La QV va donc à nouveau tirer profit de cette distribution plus compacte de la source pour répartir les représentants. Le gain en mémoire croît évidemment lorsque les dépendances entre les coordonnées vectorielles sont plus fortes (il peut être très important même pour une source uniforme). Ceci explique qu'il est judicieux de choisir, lors du découpage de la sous-bande d'une image transformée, une configuration de bloc privilégiant l'axe selon lequel les pixels sont les plus corrélés. Nous comprenons aussi que la QV ne réclame pas que les coordonnées des vecteurs à quantifier soient décorrélées car il s'adapte justement à cette corrélation, par contre il faut chercher à décorréler au maximum les vecteurs entre-eux.

En reprenant le cas d'un processus gaussien (équation 2.12) :

$$||p^*_{\mathbf{X}}(\mathbf{x})||_{\frac{k}{k+2}} = 2.\pi.\left(\frac{k+2}{k}\right)^{\frac{k+2}{2}}.\sigma_X^2$$

donc :

$$G_M(k) = 1/\left(\det\left(\Gamma_X(k)\right)\right)^{1/k}$$

$G_M(k)$ est toujours supérieur à 1 et la QV est donc toujours préférable.
A titre d'exemple, pour un processus autorégressif gaussien du 2^e ordre résultant du filtrage d'un bruit blanc par un filtre de transfert définit dans [Moreau95] :

$$\frac{1}{A(z)} = \frac{1}{1 + a_1.z^{-1} + a_2.z^{-2}}$$

avec $a_1 = -1,27$ et $a_2 = 0,81$, $G_M(+\infty) = 5,75 = 7,6$ dB.

Conclusion

Cette étude souligne que la supériorité de la QV est due à sa capacité de mettre directement à profit les propriétés des composantes des vecteurs source, pour effectuer une meilleure répartition des représentants dans l'espace. Les gains augmentent alors toujours avec la dimension. Cette analyse théorique reposant sur l'hypothèse haute résolution, présente évidemment des bornes asymptotiques que nous ne pouvons qu'approcher. Mais il faut remarquer que pour une source sans mémoire, même dans les conditions idéales, au-delà de la dimension 8 le gain s'amenuise [Lookabaugh et al.89]. La distinction des trois gains de partitionnement, de forme et de mémoire, est aussi très utile pour la conception d'un quantificateur vectoriel, afin de préciser les caractéristiques du système et d'en analyser les performances.
La partie suivante décrit alors le quantificateur vectoriel optimal dont les résultats sont les plus proches de ceux théoriques, mais cela au prix d'une complexité calculatoire rédhibitoire en codage d'images.

2.5 Quantification vectorielle optimale

2.5.1 Introduction

Comme nous l'avons expliqué, quantifier consiste à répartir dans un espace de dimension fixée un nombre déterminé de représentants, ce nombre étant fonction du débit alloué au quantificateur. L'efficacité du quantificateur se mesure alors à la qualité de restitution du signal source qui doit être la plus fidèle possible (*i.e.* avec une erreur de reconstruction minimale). Pour une distribution statistique donnée de la source et un débit fixé, le quantificateur **globalement optimal** est celui qui minimise la distorsion moyenne, un quantificateur **localement optimal** a un dictionnaire qui peut être légèrement perturbé sans que la distorsion moyenne n'augmente.
La théorie qui établit la supériorité de la QV sur la QS n'est pas constructive, elle ne décrit pas la façon de concevoir le dictionnaire globalement optimal. Seules des propriétés suffisantes sont connues qui permettent de construire des dictionnaires localement optimaux.
Le quantificateur (localement) optimal est alors celui réunissant pour un débit fixé [Linde et al.80] [Gersho et al.92] :

- l'encodeur optimal (pour un dictionnaire fixé), celui-ci respecte la "règle du plus proche voisin" que nous avons décrite à l'équation 2.2. Cette règle détermine la partition de l'espace $\mathbb{R}^k$ en Voronoï ;
- le décodeur optimal (pour une partition de $\mathbb{R}^k$ donnée), tel que le vecteur représentant $\mathbf{y}_i$ minimise la distorsion associée au Voronoï C_i, $\mathbf{y}_i$ est donc le centroïde de cette région ;
- une troisième condition est nécessaire : il faut que la probabilité d'avoir un vecteur à coder à la même distance de deux représentants soit nulle, sinon ce vecteur source est affecté à l'un des 2 représentants, la partition optimale de l'espace n'est plus et donc la condition de décodeur optimal est devenue impossible. Si les vecteurs source sont à amplitude continue, cette troisième condition est toujours vérifiée.

2.5.2 Algorithme de Lloyd généralisé

Les trois conditions précédentes conduisent à la conception d'un algorithme qui réalise, à partir d'une séquence d'apprentissage représentative de la statistique de la source à coder, la construction d'un dictionnaire (localement) optimal. Cet algorithme de classification, encore appelé algorithme des **K-moyens** ("K means" [Macqueen67]) est l'extension au cas vectoriel de l'algorithme de Lloyd-Max du cas scalaire. Il s'agit d'un algorithme d'optimisation itératif opérant à partir d'un dictionnaire initial. A chaque itération (dite "itération de Lloyd"), deux opérations distinctes sont appliquées (voir le schéma de la figure 2.5) :

- une classification suivant la règle d'encodage optimal,
- une optimisation suivant la règle de décodage optimal.

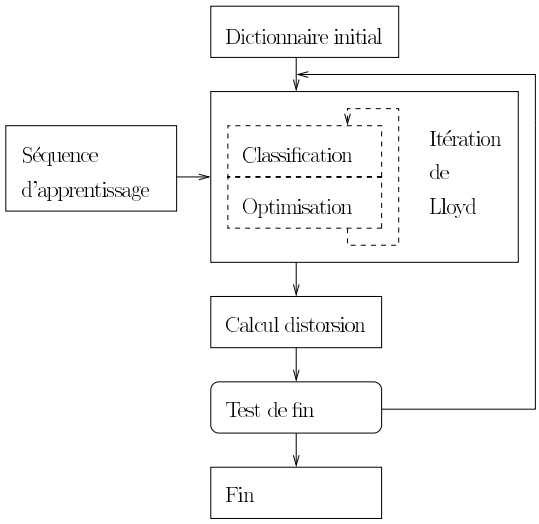


FIG. 2.5 – *Schéma de fonctionnement de l'algorithme de Lloyd.*

Chaque itération de Lloyd, en modifiant localement le dictionnaire, réduit ou laisse inchanger la distorsion moyenne. L'algorithme converge en un nombre fini d'itérations vers le minimum local le plus proche correspondant au dictionnaire initial. Le choix de ce dernier est donc capital car il conditionne les résultats finaux de l'algorithme. Plusieurs méthodes ont été proposées pour le déterminer :

- une initialisation aléatoire : le dictionnaire le plus simple est celui qui contient les L premiers vecteurs de la suite d'apprentissage ou L vecteurs extraits aléatoirement de cette suite. Ces vecteurs peuvent bien sûr ne pas être représentatifs de la suite d'apprentissage, et conduire à des résultats très médiocres ;

- un algorithme à seuil où au lieu de prendre L vecteurs aléatoirement, une distance minimale est fixée entre les éléments du dictionnaire initial. Cette méthode permet d'obtenir une meilleure représentativité que dans le cas précédent mais n'est toujours pas satisfaisante, le seuil étant souvent difficile à déterminer puisque dépendant de la complexité de la séquence d'apprentissage ;

- une méthode des "vecteurs produits", elle nécessite de quantifier scalairement les k composantes des vecteurs de la séquence d'apprentissage sur P_k niveaux (avec $P_1.P_2 \ldots P_k = L$) et d'effectuer un produit cartésien entre les dictionnaires de base pour obtenir les L représentants initiaux. Le traitement des composantes de manière indépendante ne permet pas d'obtenir à coup sûr un dictionnaire optimal, il est même possible d'obtenir des représentants initiaux ne représentant aucun vecteur de la séquence d'apprentissage ;

- une méthode par dichotomie vectorielle qui est référencée comme étant l'**algorithme LBG** [Linde et al.80]. Elle combine à l'itération de Lloyd une technique dite de "splitting" (le schéma de l'algorithme est présenté à la figure 2.6). Celle-ci consiste

à découper chaque vecteur représentant $\mathbf{y}_i$ en 2 nouveaux vecteurs $\mathbf{y}_i + \varepsilon$ et $\mathbf{y}_i - \varepsilon$ (ε étant un vecteur de perturbation de faible énergie), avant d'appliquer au nouveau dictionnaire obtenu les itérations de Lloyd. Le dictionnaire initial est alors le centroïde de la séquence d'apprentissage, puis l'algorithme génère une succession de dictionnaires (à chaque boucle le nombre de vecteurs de reproduction est multiplié par 2).

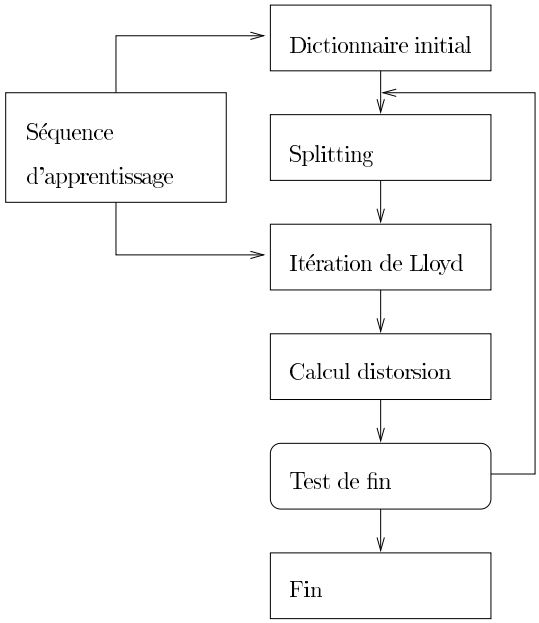


FIG. 2.6 – *Schéma de fonctionnement de l'algorithme de LBG.*

Nous choisissons d'analyser la complexité opératoire liée à l'algorithme LBG qui fournit les meilleurs résultats et est donc le plus utilisé.

Complexité de la construction du dictionnaire avec l'algorithme LBG

Cette complexité est fonction des paramètres du dictionnaire (*i.e.* la dimension k, le nombre de vecteurs représentants L), et des paramètres de la séquence d'apprentissage (*i.e.* le nombre de vecteurs M_v). Lors de la construction du dictionnaire, la phase de classification de l'itération de Lloyd est la plus coûteuse en terme calculatoire. En effet chaque vecteur source doit être comparé à chacun des L représentants, il y a autant de calculs de distorsion dans $\mathbb{R}^k$ et cela à chaque itération. Ainsi si n itérations sont exécutées (en considérant toujours la métrique euclidienne) :

- le nombre de multiplications effectuées est $N_\times = k.L.M_v.n$,
- le nombre d'additions effectuées est $N_+ = (2.k-1).L.M_v.n$,
- le nombre de comparaisons effectuées est $N_c = (L-1).M_v.n$.

La phase d'optimisation (calcul des nouveaux centroïdes des classes) nécessite aussi $k.(M_v - L).n$ additions et $k.L.n$ multiplications. Enfin $k.(L + M_v) + M_v$ emplacements mémoires sont nécessaires pour stocker les vecteurs de la séquence d'apprentissage, leurs classes et les vecteurs du dictionnaire.

Complexité de la recherche au sein du dictionnaire avec l'algorithme LBG

Le dictionnaire obtenu ne possède aucune structure topologique particulière facilitant le codage. Ainsi pour trouver le vecteur représentant correspondant au vecteur de la source à coder, L calculs de distorsion sont nécessaires. Cette procédure de **recherche exhaustive** au sein du dictionnaire a donc une complexité d'ordre $L = 2^{R.k}$, R étant le débit binaire du code à longueur fixe. En détaillant, coder un vecteur nécessite :

- $L.k$ multiplications,
- $(2.k - 1).L$ additions,
- $(L - 1)$ comparaisons.

Or de bonnes performances ne sont atteintes qu'à débit élevé (pour de petites dimensions), ou inversement avec de grandes dimensions (pour des débits faibles).

Ce bilan rédhibitoire a conduit à envisager de nouvelles méthodes de quantification vectorielle permettant un codage avec des coûts calculatoires moindres. Nous proposons dans la partie suivante de passer en revue les principales techniques (qui peuvent parfois être combinées) utilisées afin de simplifier le codage. Nous comprenons que le résultat obtenu sera toujours sous-optimal (le représentant choisit ne sera pas toujours le plus proche voisin du vecteur à coder, cependant il en sera proche en moyenne) : il s'agit alors de faire un compromis.

2.6 Les évolutions de l'algorithme LBG

2.6.1 Préambule

Nous donnons (sans être exhaustif) une description sommaire des nombreux quantificateurs vectoriels conçus, afin d'augmenter la dimension k ou la taille du dictionnaire L sans que la charge de calculs explose, ou afin de gérer les évolutions de la source non-stationnaire (ses nombreux quantificateurs sont exposés en détail dans [Nasrabadi et al.88b] [Gersho et al.92]). L'idée de base étant de structurer le dictionnaire, nous trouverons alors la QV par produit cartésien, la QV multi-étages et celle classifiée. La QV arborescente et celle algébrique, qui appartiennent à cette famille de quantificateurs vectoriels avec contraintes structurelles sur le dictionnaire, seront décrites ultérieurement. Notre but est aussi de dresser une présentation générale de la quantification vectorielle, pour être complet nous nous devons de présenter la QV neuronale, la QV prédictive, la QV avec transformée, celle avec un automate à états finis et enfin celle avec contrainte entropique.

2.6.2 Quantification vectorielle neuronale

Les "réseaux de neurones artificiels" sont des outils de classification de données ayant trouvé des applications en reconnaissance des formes et de la parole. Ils sont aussi utilisés afin de construire le dictionnaire pour la QV d'images [Nasrabadi et al.88a].
Une carte auto-organisatrice de Kohonen [Kohonen89] peut-être représentée comme une couche de neurones (*i.e.* les vecteurs représentant), chacun de ceux-ci étant d'une part relié à l'entrée par l'intermédiaire de poids synaptiques variables, et d'autre part à ses "voisins" sur la grille de Kohonen par l'intermédiaire de poids fixes (en pratique la notion de voisinage est déterminée *a priori*) .
La première couche (de poids variables) sert à calculer les activités de chaque neurone suivant un produit scalaire entre le vecteur entrant (à coder) $\mathbf{x}$ et le vecteur-poids. La deuxième couche (de poids fixes) a un rôle d'inhibition latérale entre les valeurs de neurones, en renforçant l'activité de ceux qui répondent préférentiellement à la présentation de $\mathbf{x}$. Finalement chaque neurone reçoit une contribution due à la première couche de synapses, ainsi qu'une valeur provenant de l'activité de ses voisins topologiques, multipliée par un poids qui décroît en fonction de la distance.
Kohonen [Kohonen89] a proposé un modèle simplifié de cartes auto-organisatrices où il suffit de choisir, à chaque présentation d'un vecteur $\mathbf{x}$ (appartenant à une séquence d'apprentissage), le neurone i^* tel que : $L_2(\mathbf{x}-\mathbf{y}_{i^*}) \leq L_2(\mathbf{x}-\mathbf{y}_i), \forall i \in L$ où L est l'ensemble des indices des neurones. Les poids de ces derniers sont alors modifiés à chaque présentation d'un vecteur $\mathbf{x}$ selon la loi : $\mathbf{y}_i \longleftarrow \mathbf{y}_i + \alpha(t).G(i,\mathbf{x}).(\mathbf{x}-\mathbf{y}_i)$ où $\alpha(t)$ est un facteur d'adaptation décroissant en fonction de la progression dans la séquence d'apprentissage, et $G(i,\mathbf{x})$ une fonction de voisinage. Au départ les poids sont souvent initialisés aléatoirement. Puis au fur et à mesure des présentations des $\mathbf{x}$ il y a auto-organisation du réseau ; des vecteurs proches dans l'espace d'entrée vont converger vers des centroïdes proches dans l'espace de sortie.
En comparaison de l'algorithme LBG, cette approche peut donner des résultats équivalents pour la qualité de restitution d'images [Nasrabadi et al.88a]. Si la construction du dictionnaire par apprentissage n'est plus itérative et ne nécessite plus de partitionner la séquence avant de calculer les centroïdes, elle demeure toujours très coûteuse en terme de calculs.

2.6.3 Quantification vectorielle par produit cartésien

Cette méthode consiste à décomposer un vecteur en plusieurs sous-vecteurs de dimensions éventuellement différentes. Une quantification vectorielle de chacun d'eux est alors réalisée (la complexité calculatoire est proportionnelle à $\sum_i L_i$ où L_i est la taille de chacun des dictionnaires). La difficulté réside dans la technique à adopter pour décomposer le vecteur tel qu'il n'existe plus de dépendances entre les sous-vecteurs [Benazza92].
La QV de type "forme-gain" ("shape-gain VQ" [Sabin et al.84]) est un cas particulier où deux dictionnaires sont conçus : un pour la forme (*i.e.* le contenu spectral des vecteurs), l'autre pour le gain (*i.e.* l'énergie des vecteurs). Cette technique est retenue en codage de parole où le contenu spectral du signal est peu modifié par l'intonation de la voix (qui

l'amplifie). Le principe est le même en image avec le "mean-removed VQ" [Budge et al.85] où le vecteur est décomposé en deux sous codes : un scalaire portant l'information de la valeur moyenne du vecteur (*i.e.* un bloc de l'image), l'autre portant la différence entre le vecteur moyen et celui codé.
Nous rencontrons également dans cette famille de quantificateur la QV "classifiée" ("classified VQ" [Nasrabadi et al.88b]) qui s'appuie sur le partitionnement d'un dictionnaire de taille importante en plusieurs sous-dictionnaires de taille restreinte, et la mise en oeuvre d'une fonction de classification des vecteurs source à l'encodage (à chaque classe correspond un dictionnaire). Une classification suivant la structure contenue dans le bloc de l'image (*i.e.* l'orientation et la localisation des contours) est par exemple proposée dans [Maresq86].

2.6.4 Quantification vectorielle multi-étages

Le quantificateur vectoriel multi-étages ou en cascade [Barnes et al.96], procède par approximations successives : l'erreur d'un premier étage est requantifiée par un second, et le procédé peut-être réitéré. La quantification est alors grossière à la première étape puis devient ensuite de plus en plus fine. Ce QV peut-être reconnu comme un système d'analyse multirésolution du signal. Il demeure particulièrement adapté aux techniques de codage hiérarchiques des images.

2.6.5 Quantification vectorielle prédictive et quantification vectorielle par transformée

Nous avons déjà décrit la QS prédictive qui consiste à décorréler le signal en enlevant la partie prédictible, puis à quantifier le signal résiduel. Il s'agit ici de généraliser la méthode au cas vectoriel. L'extension directe du MICD au cas multidimensionnel existe mais elle demeure efficace que si le signal est fortement corrélé. En effet, la prédiction de blocs entiers a un sens que si la mémoire de la source est suffisante (sinon le système fonctionnant en boucle fermée, les erreurs s'accumuleront). C'est le cas pour le codage de la bande basse fréquence du signal de parole décomposé en sous-bandes.

Pour la compression de séquence d'images, les schémas de codage présentés au chapitre 1 (voir les figures 1.3 et 1.4) sont alors mis en oeuvre. Il y a toujours conservation de la propriété fondamentale d'égalité entre l'erreur de quantification (*i.e.* $q = e - e_q$, en adoptant les notations de ces figures) et l'erreur de reconstruction (*i.e.* $\bar{q} = I - I_r$).
Dans le premier chapitre nous avons également donné le principe du codage hybride où les images d'erreur de prédiction subissent une nouvelle décomposition avant quantification. Il est légitime de s'interroger sur les buts de ces deux étapes de décorrélation avant la quantification vectorielle, alors que celle-ci tire justement profit directement des dépendances existantes entre les composantes des vecteurs. En fait un tel schéma de codage applique précisément les deux règles en compression : la non-transmission de l'information prédictible, et la non-transmission de celle imperceptible par le système visuel humain (SVH). La prédiction, la décomposition de l'erreur résiduelle et sa quantification, réalisent une mise en forme spectrale du bruit de quantification. Ce dernier devient plus important dans les

zones fréquentielles où le signal est le plus énergétique (*i.e.* les hautes fréquences) afin de répondre aux propriétés de masquage du SVH (la connaissance du mode de perception est évidemment nécessaire). Il est même permis de perdre au niveau du rapport signal sur bruit, si il y a à gagner en répartissant le bruit de façon qu'il soit invisible. La difficulté vient qu'il est très délicat de chiffrer le gain apporté par cette mise en forme spectrale du bruit. Ce gain est purement subjectif.
La quantification vectorielle de l'erreur résiduelle décomposée en sous-bandes [Cosman et al.96] offre une grande liberté dans la répartition des bits disponibles (pour un code à longueur fixe la QV permet un débit fractionnaire). Un dictionnaire respectif est construit par sous-bande, et la forme des vecteurs adaptée de façon à exploiter les dépendances entre coefficients transformés (nous retrouvons le principe de la classification des données). Cette chaîne de décorrélation réalise également une mise en forme du signal à quantifier, déjà sa densité de probabilité est devenue unimodale et suit une loi "étroite" (nous verrons au chapitre 5 que c'est une loi gaussienne généralisée). Enfin le signal est rendu plus stationnaire, le dictionnaire conçu demeurera valide au cours du temps.

2.6.6 Quantification vectorielle avec un automate à états finis

Cette approche de quantification vectorielle dite "avec mémoire" est différente, car il y a introduction d'une information provenant des vecteurs précédemment codés dans la minimisation même de la distorsion. L'encodeur et le décodeur sont munis d'une fonction de transition d'états, où l'état du système dépend du dernier symbole canal émis et de l'état précédent. Un dictionnaire est associé à chacun des états. Le problème est alors de trouver la séquence de symboles canal (la seule information transmise) et d'états, de façon à minimiser la distorsion totale. Il faut distinguer deux types de QV avec un automate à états finis : la QV dite "récursive" et celle en treillis.

Quantification vectorielle récursive

Pour les QV récursifs la solution est sous-optimale car choisie à court terme. A chaque instant un QV sans mémoire (*i.e.* suivant la règle du plus proche voisin) sélectionne un vecteur dans le dictionnaire déterminé par la fonction de transition d'état. Celle-ci consiste en un prédicteur du comportement futur de la source (une analogie existe avec la quantification prédictive "en arrière" ou encore avec la QV classifiée). La difficulté réside dans le choix de cette fonction et la détermination des dictionnaires, aucun algorithme optimal n'est connu. La procédure commune est de construire d'abord (avec un *a priori*) les dictionnaires par apprentissage, puis de déterminer la fonction de transition d'états. Celle-ci peut également être conçue à partir de la statistiques des blocs de l'image [Foster et al.85] ou selon la théorie des graphes [Lee et al.94].

Quantification vectorielle en treillis

La QV en treillis ou à décision retardée [Ungerboeck87a] [Ungerboeck87b] optimise les décisions à long terme. Le coût calculatoire est alors élevé, et un délai de codage important est nécessaire. Il faut donc limiter le nombre de vecteurs de la séquence sur laquelle la distorsion doit-être minimisée, et ne pas tester toutes les combinaisons possibles entre les vecteurs en effectuant une recherche par étapes et sur des voisinages. L'évolution temporelle du système est illustrée à l'aide d'un graphe (ou treillis) représentant les transitions possibles entre états. L'algorithme de viterbi [Viterbi et al.74] peut alors être mis en oeuvre afin de trouver dans le graphe orienté le chemin produisant un coût minimal. Cet algorithme repose sur le principe d'optimalité en programmation dynamique, qui permet de faire croître le chemin par des extensions successives. La séquence optimale de dictionnaires, où se situent les vecteurs les plus proches de ceux de la séquence d'entrée, est alors produite. Pour la QV en treillis, la distorsion dépend du nombre total d'échantillons pris en compte pour minimiser la distorsion : $(k.T)$ où k est la dimension vectorielle et T le nombre de vecteurs dans la séquence. Il est donc possible de prendre $k = 1$ et T grand, ce qui diffère de la QV sans mémoire où il faut k grand afin de se rapprocher des limites théoriques.

2.6.7 Quantification vectorielle avec contrainte entropique

Ce QV a pour but de construire le dictionnaire engendrant une distorsion minimale, mais il est aussi contraint à avoir un coût moindre (*i.e.* une entropie minimale). Nous retrouvons alors la formulation introduite à l'équation 2.5.
Dans le cas d'une source stationnaire gaussienne centrée avec mémoire, la borne inférieure de Shannon est :

$$\sigma_Q^2 = \frac{\sigma_X^2 . 2^{-2.R}}{G_p(\infty)}$$

où $G_p(\infty)$ est le gain de prédiction (voir l'équation 2.7 et l'annexe B). Or lorsque la dimension tend vers l'infini (*i.e.* $k \to +\infty$), la puissance de l'erreur du QV de l'équation 2.13 tend vers la borne. Ce résultat confirme, là encore, que la quantification vectorielle est optimale.

L'algorithme décrit dans [Chou et al.89a] est une généralisation de l'algorithme LBG. Il minimise un critère global comportant la distorsion et l'entropie liées au dictionnaire. Ainsi pour une taille fixée de ce dernier et pour différentes entropies, il permet d'atteindre les distorsions les plus faibles. Cet algorithme est optimal mais il est très couteux.
Des études reposant sur l'hypothèse haute résolution [Gersho79] [Newman84] font la conjecture que la construction d'un dictionnaire contraint en entropie conduit à un réseau régulier de points. Cependant si l'hypothèse asymptotique est relâchée (*e.g.* considération de dictionnaires de tailles réduites), d'autres travaux [Sayood et al.84] montrent qu'un QS contraint en entropie peut avoir des performances supérieures à celle d'un QV algébrique.

2.6.8 Conclusion

Cette liste non exhaustive des évolutions de l'algorithme LBG, souligne la diversité des méthodes qui ont été élaborées afin de mettre en oeuvre la quantification vectorielle. Il faut remarquer que de fortes analogies peuvent apparaître entre ces quantificateurs. A titre d'exemples nous faisons remarquer qu'un QV récursif est un QV classifié fonctionnant en boucle fermée, qu'un QV prédictif est un automate ayant un nombre infini d'états, ou encore que des ressemblances demeurent entre la QV "forme-gain" et celle multi-étages, ou entre la QV classifiée et celle arborescente dont nous faisons la description au chapitre 5. Il faut aussi ajouter qu'en pratique ces techniques sont souvent combinées.

La méthodologie la plus récente introduite en QV est certainement la quantification vectorielle algébrique (QVA), où le dictionnaire n'est plus calculé par apprentissage mais défini a priori *comme un ensemble de vecteurs régulièrement distribués dans l'espace. Si les difficultés ne sont plus l'encodage des vecteurs, la construction du dictionnaire et son stockage, elles résident alors dans le choix et la troncature des réseaux, ainsi que dans l'indexage des représentants. Nous avons retenu et adapté la QVA pour concevoir notre quantificateur, la description détaillée de cette technique complexe est faite au prochain chapitre.*

Nous achevons cette présentation générale de la quantification vectorielle en abordant le problème de l'allocation binaire optimale entre les sous-bandes des images transformées. La solution à ce problème est primordiale pour la conception de tous codeurs en sous-bandes.

2.7 Allocation binaire optimale

2.7.1 Formalisation du problème

Nous avons déjà souligné que le codage par transformées ou en sous-bandes autorise une mise en forme spectrale du bruit de quantification, telle que les sous-bandes les plus significatives visuellement soient plus finement quantifiées que les autres. Ce résultat est obtenu par l'intermédiaire d'une distribution inégale des ressources binaires entre les différentes sous-bandes. Il faut reconnaitre que des mesures objectives de la qualité subjective de restitution des images sont difficiles à concevoir même si des travaux de recherche sont engagés en ce sens [Senane96]. Nous adoptons donc, comme dans la plupart des cas, une erreur quadratique moyenne pour mesurer la distorsion. L'objectif est alors de minimiser cette dernière en répartissant judicieusement les bits, et le système s'adapte au SVH en accordant aux basses fréquences plus de débits.
La transformée est supposée orthogonale ainsi les erreurs avant et après reconstruction sont égales, et la distorsion totale D est la somme des distorsions d_j dans chacune des M sous-bandes. R est alors le débit total égal à la somme des r_j débits dans les sous-bandes, et R_d le débit à ne pas dépasser. Nous pouvons formaliser mathématiquement le problème

par :

$$\min D = \min \sum_{j=0}^{M-1} d_j \qquad \text{Sous la contrainte :} \qquad R = \sum_{j=0}^{M-1} r_j \leq R_d \tag{2.15}$$

Pour le résoudre, nous rencontrons deux types d'approches : les méthodes statistiques, et celles par programmation convexe.

2.7.2 Méthodes statistiques

Ces méthodes, utilisées principalement avec la QS, reposent sur une modélisation de la statistique des signaux en sous-bandes afin de déterminer directement les pas des quantificateurs. Les densités de probabilité des signaux sont communément modélisées par des gaussiennes généralisées (voir le chapitre 5). Pour identifier les paramètres du modèle correspondant le mieux à la sous-bande traitée, des tests classiques de "Smirnov-Kolmogorov" ou du "Khi carré" sont alors mis en oeuvre.
Dans [Gersho et al.92] un premier exemple simple, où la statistique des sous-bandes est supposée suivre un loi gaussienne, est donnée. L'hypothèse haute résolution est retenue et la QS est supposée optimale, de telle sorte que la puissance de l'erreur de quantification dans une sous-bande soit donnée par l'équation 2.4. L'allocation optimale de bits conduit à ce que le nombre de pas de quantification d'un QS, soit proportionnel à l'écart-type de la variable aléatoire qu'il quantifie. Cette modélisation et ces hypothèses sont évidemment trop contraignantes en pratique, la solution numérique optimale est même inexploitable car elle peut conduire à ce que le nombre de bits ne soit ni entier, ni positif. Cependant ce résultat a amené l'utilisation pratique d'algorithmes répartissant itérativement les bits dans les différentes sous-bandes, là où ils ont le plus d'effet.
Une autre technique possible à partir des modèles statistiques des données, repose sur la méthode des multiplicateurs de Lagrange. Ceux-ci traduisent les contraintes de 2.15 sous la forme :

$$L(\lambda) = \left\{ \sum_{j=0}^{M-1} d_j(r_j) + \lambda . \left(\sum_{j=0}^{M-1} r_j - M.R_d \right) \right\}$$

Il faut résoudre :

$$\frac{\partial L(\lambda)}{\partial r_j} = 0$$

La solution est alors donnée par :

$$\frac{\partial d_j(r_j)}{\partial r_j} = -\lambda \tag{2.16}$$

Ainsi dans la représentation des courbes débit/distorsion relatives à chacune des sous-bandes, les courbes $d_j(r_j)$ doivent toutes avoir une même pente constante $-\lambda$. Les variations de cette dernière fixent le débit total.
A l'aide de cette méthode, [Levent94] donne une expression de d_j et r_j en fonction du pas de quantification d'un quantificateur scalaire uniforme, mais que dans certains cas particuliers de gaussiennes généralisées et en faisant des hypothèses fortes.

Dans le cas vectoriel la formulation de modèles de densités de probabilité relatifs aux sources réelles est extrêmement complexe. L'expression de l'EQM donnée à l'équation 2.8, montre qu'il faut connaître $P_{\mathbf{X}}(\mathbf{x})$ dans chacun des Voronoï aux formes variables (dans le cas scalaire les domaines d'intégration sont de simples intervalles, et la densité marginale des échantillons suffit). Seule l'hypothèse haute résolution, où les Voronoï sont de si petite taille qu'à l'intérieur la densité est constante, permet de développer cette équation. Cette hypothèse qui s'éloigne fortement des conditions réelles de sources et de débits, est trop restrictive. C'est pourquoi nous ne pouvons retenir une méthode statistique pour résoudre le problème de l'allocation binaire dans le cas de quantificateurs vectoriels.

2.7.3 Méthodes par programmation convexe

Ces méthodes consistent à calculer au préalable un certain nombre de quantificateurs par sous-bande, puis à choisir pour chacune de ces dernières le quantificateur optimal, de façon à minimiser l'équation sous contraintes 2.15. La série de quantificateurs est adaptée à la sous-bande qu'elle code : un intervalle de débit, dont la largeur est fonction du domaine fréquentiel du signal, est attribué à chaque quantificateur (*e.g.* pour les basses fréquences les intervalles sont plus grands).
Ces approches qui ne reposent sur aucune modélisation, ont l'immense avantage d'être utilisables avec n'importe quel type de quantificateurs (QV ou QS). C'est donc celles que nous avons retenues.
Les algorithmes de minimisation sous contrainte ont été introduits par de nombreux auteurs, nous citons [Shoham et al.88] [Ramchandran et al.93]. Nous reformulons ainsi le problème : nous considérons toujours M sous-bandes chacune indicée par j, nous disposons pour chacune de N quantificateurs $q_{j,i}$ (l'indice du quantificateur est i) mais ce nombre de quantificateurs pourrait aussi être variable. Pour le i^e quantificateur et la sous-bande j, $d_{j,i}$ est la distorsion engendrée avec le débit $r_{j,i}$. Toutes les $d_{j,i}(r_{j,i})$ peuvent-être mesurées ou déduites. Alors en choisissant parmi toutes les combinaisons possibles un quantificateur par sous-bande, la distorsion totale et le débit total sont :

$$D = \frac{1}{M}.\sum_{j=0}^{M-1} d_{j,i} \quad \text{et} \quad R = \frac{1}{M}.\sum_{j=0}^{M-1} r_{j,i}$$

L'objectif est toujours de minimiser D sous la contrainte d'un débit $R \leq R_d$; il faut donc trouver la combinaison optimale de quantificateurs $q_{j,i}$. Chaque couple (R, D) possible est symbolisé par un point dans le plan débit/distorsion associé au système, la combinaison totale forme un nuage de points. La théorie de l'information montre qu'il existe vraisemblablement une enveloppe convexe à ce nuage de points (voir l'annexe B). Le problème devient la recherche du point appartenant à cette enveloppe convexe minimisant D sous la contrainte que $R \leq R_d$ [Nguyen95]
Le nombre N^M de combinaisons est très élevé et il n'est pas envisageable de toutes les tester. Pour réduire les calculs, il est tenu compte de la contrainte en recourant (encore)

aux multiplicateurs de Lagrange. Il faut alors résoudre :

$$\min(D+\lambda.R) \Longleftrightarrow \min_{q_{j,i}} \left(\sum_{j=0}^{M-1} d_{j,i} + \lambda. \sum_{j=0}^{M-1} r_{j,i} \right) \Longleftrightarrow \sum_{j=0}^{M-1} \min_{q_{j,i}} (d_{j,i} + \lambda.r_{j,i}) \qquad (2.17)$$

Il y a donc réduction de la complexité car la minimisation de la distorsion est réalisée dans chaque sous-bande (la parallélisation d'un tel algorithme est même possible). Précisément le problème est devenu la recherche, en considérant chaque sous-bande séparément, du quantificateur minimisant la distorsion avec la contrainte en débit. Cette approche locale, qui demeure sous-optimale comparée à celle globale faite en considérant une combinatoire totale entre tous les quantificateurs, est justifiée car la complexité est réduite de façon conséquente.
Nous décrivons de façon succinte la forme générique de l'algorithme mis en oeuvre dans [Shoham et al.88] afin de résoudre la relation 2.17. Cette méthode est aussi proposée par [Ramchandran et al.93] [Demaistre et al.96] dans le cadre de la décomposition hiérarchique du signal en "paquets d'ondelettes", où le but est d'obtenir l'arbre de décomposition en ondelettes achevant la découpe fréquentielle optimale de l'image (la technique avancée est aussi très proche de celle de l'agorithme de BFOS [Breiman et al.84] que nous développons au chapitre 4). Trois étapes se succèdent :

- le calcul séparé pour chaque sous-bande des distorsions $d_{j,i}$ et des débits $r_{j,i}$;

- la recherche des enveloppes convexes relatives à chacun des M nuages de points. Soit la droite d'équation $d_j = -\lambda.r_j + \alpha_j$ définie dans le plan débit/distorsion de la j^e sous-bande. L'enveloppe convexe du nuage est obtenue point par point (*i.e.* pour différentes valeurs de λ) en recherchant dans le plan (d_j, r_j) le point minimisant α_j (c'est le cas particulier présenté à la figure 2.7) ;

- la recherche sur l'enveloppe convexe globale du point le plus proche du débit R_d voulu. Cette enveloppe convexe globale est obtenue, pour chaque λ, par sommation des débits et des distorsions relatives aux sous-bandes. Si nous considérons le nuage de points associé au système global présenté à la figure 2.8, le point recherché de l'enveloppe (*i.e.* C) est celui qui maximise l'ordonnée β du point à l'intersection de la droite $\alpha = D + \lambda.R$ et de la droite $R = R_d$. Une possibilité est de balayer de manière discrète les λ, et de trouver le β maximal (*i.e.* λ_{opt}) à l'aide d'un algorithme du gradient.

La détermination des paramètres λ ainsi que l'optimisation pour trouver λ_{opt}, sont les phases les plus complexes. De plus la solution expérimentale souhaitée peut ne pas appartenir exactement à l'enveloppe convexe. Des méthodes ont donc été proposées afin d'adapter cet algorithme d'allocation binaire, nous présentons au chapitre 5 celle que nous avons retenue pour notre quantificateur.

2.8 Conclusion

La théorie souligne que la quantification vectorielle est toujours plus efficace que celle scalaire, elle est aussi plus délicate à mettre en oeuvre. L'algorithme optimal LBG a un coût calculatoire rédhibitoire pour la compression des images. Depuis quelques années, parmi les nombreuses méthodes introduites afin de réduire ce coût, de nombreux efforts se sont portés sur la quantification vectorielle algébrique (QVA), dont les algorithmes sont rapides et performants. Si cette technique est optimale pour quantifier une source uniforme, elle doit être adaptée au codage des autres types de sources ; c'est pourquoi nous proposons la quantification vectorielle algébrique et arborescente. La première étape, qui est l'objet du prochain chapitre, est de décrire en détail la QVA.

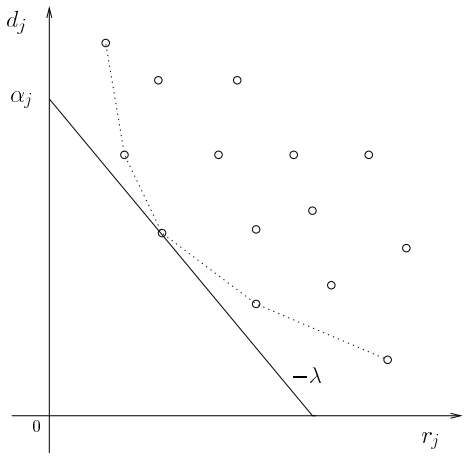


FIG. 2.7 – *Nuage des points débit/distorsion d'une sous-bande j.*

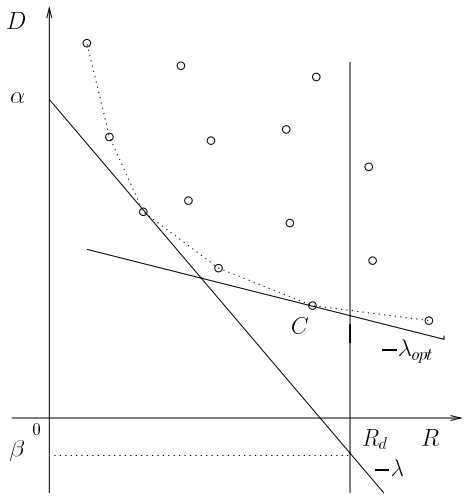


FIG. 2.8 – *Nuage des points débit/distorsion associés au système global.*

Chapitre 3

Quantification vectorielle algébrique

3.1 Introduction

La quantification vectorielle algébrique (**QVA**), introduite afin de réduire la complexité calculatoire inhérente aux techniques de recherches exhaustives au sein du dictionnaire, se conforme à l'idée de structurer fortement ce dernier. Le but est aussi de se dispenser d'une phase d'apprentissage pour construire le dictionnaire. Les représentants sont alors les points régulièrement répartis dans l'espace d'un réseau. L'encodage est simple car l'obtention du vecteur de reproduction est obtenue directement par des calculs sur les coordonnées du vecteur à coder (*i.e.* il n'y a plus de recherche à effectuer au sein du dictionnaire). Enfin la quantification étant indépendante de la taille du réseau et ce dernier étant naturellement connu au codeur et au décodeur, il n'a pas à être transmis et peut-être de grande taille.
Cependant un réseau régulier de points (RRP) ne peut quantifier optimalement qu'une source i.i.d uniforme, et seul le gain de partitionnement est exploité. Un RRP étant infini, il faut également limiter la taille de ce dictionnaire virtuel pour indexer les représentants. La mise en oeuvre de la QVA pour la quantification d'une source non-uniforme est finalement loin d'être triviale. Nous distinguons les étapes suivantes :

- le choix du RRP. Ce dernier conditionne le gain de partitionnement. Seuls sont connus certains réseaux offrant pour une dimension donnée (*i.e.* les dimensions 2, 4, 8, 16 et 24), la minimisation optimale de l'EQM liée à la quantification d'une source uniforme [Conway et al.93]. Mais le critère décisif demeure l'existence, pour certains d'entre eux, d'algorithmes de quantification rapide ;

- le choix de la forme du dictionnaire. Il est nécessaire de tronquer le réseau afin d'obtenir un dictionnaire de taille finie. Le cas idéal est de déterminer la forme de troncature en tenant compte de la distribution la plus probable des vecteurs source, de telle sorte que la QVA tire bénéfice d'une partie du gain de forme. Le nombre de points conservés du réseau est aussi fonction du débit alloué au quantificateur ;

– l'adaptation de la source au dictionnaire. Les algorithmes de quantification rapide sont opérationnels une fois le vecteur source projeté au sein du réseau. Il s'agit donc de normaliser les vecteurs à coder afin qu'ils se situent dans le volume du réseau tronqué. Plusieurs fonctions, linéaires ou non, ont été proposées pour placer avant le quantificateur afin de réaliser cette tâche. En aval du quantificateur, la fonction inverse de celle de projection sera appliquée au représentant précédemment calculé. Il faut également prévoir un traitement particulier concernant les vecteurs source qui, même normalisés, n'appartiennent toujours pas au dictionnaire ;

– l'indexage (*i.e.* l'opération consistant à affecter à chaque code vecteur un indice). Si l'encodage à l'aide du RRP demeure une opération simple, l'ultime étape d'indexage des vecteurs représentants est devenue particulièrement complexe, surtout lorsque le nombre de points dans le réseau tronqué est important. Les index doivent alors être calculés, et de façon générale les techniques visent à réaliser le meilleur compromis entre le coût calculatoire et celui de stockage de tableaux de conversion. Notons que si aucun apprentissage n'a été jusqu'à présent nécessaire, cette étape intervient parfois afin de construire le code entropique associé aux points du réseau (*i.e.* pour évaluer leurs probabilités d'occurence).

La quantification inverse, pour obtenir le représentant final correspondant au vecteur quantifié consiste : à décoder l'index, en déduire le point du réseau puis à renormaliser ce dernier. L'objet de ce chapitre est de détailler les différentes étapes de la réalisation d'un QVA ; nous débutons donc par la description des réseaux. Précisément, après avoir défini les paramètres caractéristiques des RRP qui nous seront nécessaires pour concevoir notre quantificateur, nous ne présentons que les réseaux réguliers retenus pour la quantification et notamment leurs algorithmes de quantification rapide. La description complète de tous les réseaux peut-être trouvée dans [Conway et al.93].

3.2 Réseaux réguliers de points

3.2.1 Définitions

Dans l'espace $\mathbb{R}^k$, nous considérons un empilement régulier de sphères identiques de rayon ρ. Un **réseau régulier** Λ ("lattice" en langage anglo-saxon) est constitué de l'ensemble des centres de ces sphères. Par définition le point 0 est le centre du réseau. Il existe, en plus de 0, k sphères de centres $\mathbf{v}_i = (v_{i_1}, v_{i_2}, \ldots, v_{i_k})^T$ telles que tout point du réseau soit déterminé par la somme (voir la figure 3.1) :

$$\sum_{i=1}^{k} \varepsilon_i . \mathbf{v}_i \quad / \quad \varepsilon_i \in \mathbb{Z}$$

où les $\mathbf{v}_i$ forment la base du réseau, ces vecteurs sont linéairement indépendants. Le parallélotope fondamental (ou Voronoï élémentaire) de Λ est la région formée par :

$$\sum_{i=1}^{k} \theta_i . \mathbf{v}_i \quad / \quad 0 \leq \theta_i \leq 1$$

Elle correspond à un domaine convexe de l'espace de dimension k, délimité par un nombre fini d'hyperplans médiateurs de dimension $(k-1)$. L'espace entier peut-être recouvert en sommant ces régions (qui elles ne se recouvrent pas).

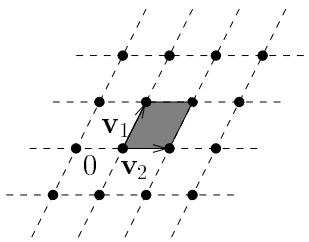


FIG. 3.1 – *Un réseau bidimensionnel et son parallélotope associé.*

La matrice génératrice de Λ est :

$$M = \left(\mathbf{v}_1^T, \ldots, \mathbf{v}_k^T\right)^T \quad \text{où} \quad \mathbf{v}_i^T = (v_{i_1}, \ldots, v_{1_m})$$

S'il est parfois plus facile de décrire un réseau k-dimensionnel à l'aide de vecteurs de m coordonnées ($m \geq k$), nous considérerons cependant avoir $m = k$. Alors un vecteur $\mathbf{y}$ du réseau est déterminé par :

$$\mathbf{y} = M^T.\boldsymbol{\varepsilon} \text{ avec } \boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k)^T \ / \ \varepsilon_i \in \mathbb{Z}$$

Le réseau régulier Λ est redéfini comme l'ensemble des points $\mathbf{y}$ de $\mathbb{R}^k$ tels que :

$$\Lambda = \left\{ \mathbf{y} \in \mathbb{R}^k \ / \ \exists \boldsymbol{\varepsilon} \in \mathbb{Z}^k \, , \mathbf{y} = M^T.\boldsymbol{\varepsilon} = \sum_{i=1}^{k} \varepsilon_i.\mathbf{v}_i \right\}$$

La matrice de Gram du réseau est définie par : $A = M.M^T$. Son déterminant est noté (en considèrant M carrée) : $\det A = (\det M)^2$. Alors le volume d'un parallélotope (et donc celui d'un Voronoï) est :

$$\det M = (\det A)^{1/2}$$

Deux réseaux, Λ_1 et Λ_2, équivalents ou similaires après rotation et/ou réflexion et/ou changement d'échelle, sont notés : $\Lambda_1 \cong \Lambda_2$. Un réseau Λ admet un dual ou réciproque Λ^* tel que : $\Lambda^* = \{\mathbf{y} \in \mathbb{R}^k \ / \ \mathbf{y}^T.\mathbf{u} \in \mathbb{Z} \, , \forall \mathbf{u} \in \Lambda\}$
Les réseaux réguliers de points sont à l'origine des solutions trouvées aux différents problèmes mathématiques d'empilement de sphères dans un espace, de recouvrement de l'espace par des sphères et de recherche du "nombre de contacts". Nous rappelons ces problèmes et leurs solutions afin d'introduire les différents paramètres caractéristiques des RRP. Mais il faut ajouter que ces derniers ont trouvé de nombreux nouveaux champs d'application (*e.g.* théorie des nombres, télécommunications, chimie, mathématique appliquée).

Problème d'empilement de sphères dans un espace

Dans l'espace $\mathbb{R}^k$ (k fixé), l'empilement le plus dense de sphères (où toutes sont équivalentes et ne se recouvrent pas) est recherché. La solution n'est connue, pour $k \geq 3$, que si les centres des sphères appartiennent à un RRP (mais la densité maximale pourrait être celle d'un réseau non régulier). Il faut alors maximiser la proportion d'espace occupée par les sphères ou densité du réseau définie par :

$$\Pi = \frac{\text{volume d'une sphère}}{\text{volume d'une région fondamentale}} = \frac{V_k.\rho^k}{(\det\Lambda)^{1/2}}$$

où V_k est le volume d'une sphère de rayon unité, et $V_k.\rho^k$ celui de la sphère k-dimensionnelle de rayon ρ. Ce dernier définit le **rayon d'empilement** (voir la figure 3.2).

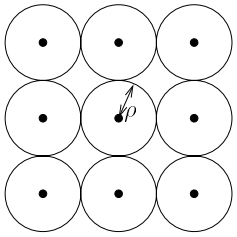


FIG. 3.2 – *Un empilement de sphères dans l'espace bidimensionnel.*

Problème de recouvrement d'espace par des sphères

Le recouvrement le plus économique de $\mathbb{R}^k$ avec des sphères est recherché. Ces sphères toutes identiques qui recouvrent l'espace se recouvrent également entre elles, et ce dernier recouvrement doit être minimal. A nouveau la solution n'est connue, pour $k \geq 3$, que si les centres de ces sphères appartiennent à un RRP. Soit r le rayon des sphères (voir la figure 3.3), il faut minimiser le nombre moyen de sphères contenues dans l'espace ou densité du recouvrement définit par :

$$\Theta = \frac{\text{volume d'une sphère}}{\text{volume d'une région fondamentale}} = \frac{V_k.r^k}{(\det\Lambda)^{1/2}}$$

r est le **rayon de recouvrement** du réseau. Il correspond à la borne supérieure minimale de la distance entre un point $\mathbf{x}$ de l'espace $\mathbb{R}^k$ et le plus proche point du réseau. Si les $\mathbf{y}_i$ sont les points de Λ, et si $d(\mathbf{x}, \mathbf{y}_i)$ est la distance entre les deux vecteurs :

$$r = \sup_{\mathbf{x}\in\mathbb{R}^k} \inf_{\mathbf{y}_i\in\Lambda} d(\mathbf{x}, \mathbf{y}_i)$$

Les points isolés, à la distance r de leurs plus proches voisins dans le réseau, sont des trous (ils se situent aux sommets des intersections entre les hyperplans médiateurs). Le problème de recouvrement apparait le dual de celui de l'empilement, cependant ils sont différents : pour le premier il s'agit de minimiser r, pour le second il faut au contraire maximiser ρ. Alors pour une dimension donnée, les RRP solutions de chacun de ces problèmes sont souvent différents.

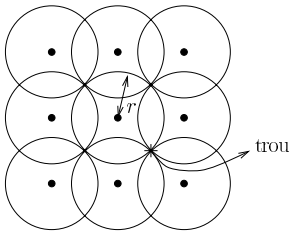


FIG. 3.3 – *Un recouvrement de l'espace bidimensionnel par des sphères.*

Problème du nombre de contacts

Cette fois nous recherchons le nombre maximal τ de sphères (toutes identiques et ne se recouvrant pas), qui peuvent être arrangées entre elles de façon à ce que chacune touche la même sphère centrale. Il s'agit d'un problème d'empilement mais uniquement sur la surface d'une même sphère. Cet aspect local fait que les réseaux trouvés sont en général différents de ceux solutions du problème d'empilement de sphères dans l'espace.
Alors que τ est variable si les réseaux sont irréguliers, il ne l'est plus s'ils sont réguliers. τ n'est en général obtenu que dans ce dernier cas ; plus exactement la solution optimale n'est connue que pour $k = 1,\ 2,\ 3,\ 8$ et 24. Nous faisons remarquer que la détermination du nombre de contacts ("kissing number" en langage anglo-saxon) intervient naturellement dans la construction des meilleurs codes sphériques qui sont associés à des sous-ensembles de points appartenant à la surface d'une sphère. Pour obtenir τ, de nouveaux outils interviennent : les **séries génératrices Thêta** qui indiquent le nombre de points sur la surface d'une sphère.

Séries Thêta

Soit $\mathbf{y}$ un point de Λ, sa norme euclidienne est fonction de ε car :

$$\sum_{i=1}^{k} y_i^2 = \mathbf{y}^T.\mathbf{y} = \varepsilon^T.M.M^T.\varepsilon \ = \varepsilon^T.A.\varepsilon$$

Cette norme détermine la forme quadratique associée à Λ. En théorie des nombres se pose le problème de connaître le nombre de façons d'écrire un entier m comme la somme de k carrés d'entiers (*e.g.* $m = \sum_{i=1}^{k} y_i^2$). Les RRP présentent du fait de leur forme quadratique, une formulation à ce problème : considérant Λ, N_m est le nombre de vecteurs $\mathbf{y}$ de norme m (*i.e.* $\mathbf{y}^T.\mathbf{y} = m$), N_m est encore le nombre de fois que la forme quadratique associée à Λ représente m. Or le calcul de N_m est facilité par l'introduction des séries Thêta associées à Λ :

$$\Theta_\Lambda(z) = \sum_{\mathbf{y}\in\Lambda} q^{\mathbf{y}^T.\mathbf{y}} = \sum_{m=0}^{+\infty} N_m.q^m \quad \text{où} \quad q = e^{i.\pi.z}$$

Les séries Thêta produisent donc le nombre de points qu'il y a à chaque distance de l'origine (le nombre de points sur chaque hypersphère de rayon $\sqrt{m}$, ou encore sur chaque

surface d'énergie m, voir aussi l'exemple de la figure 3.4). Les points étant répartis sur des sphères concentriques, pour connaître le nombre total N_T de points dans la sphère de rayon $\sqrt{m}$, il suffit de faire la somme de ceux sur chacune des sphères de rayons inférieurs : $N_T = \sum_{i=0}^{m} N_i$

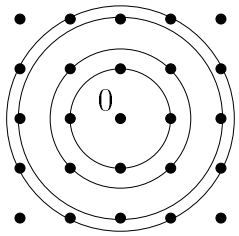


FIG. 3.4 – *Sphères concentriques d'un réseau bidimensionnel.*

La plupart des RRP s'expriment, lors du développement de leur série Thêta, comme des fonctions des séries Thêta de Jacobi dont nous donnons des exemples :

$$\begin{aligned}
\theta_2(z) = &\quad \sum_{m=-\infty}^{+\infty} q^{(m+\frac{1}{2})^2} &&= 2.q^{1/4} + 2.q^{9/4} + 2.q^{25/4} + \ldots \\
\theta_3(z) = &\quad \sum_{m=-\infty}^{+\infty} q^{m^2} &&= 1 + 2.q + 2.q^4 + 2.q^9 + \ldots \\
\theta_4(z) = &\quad \sum_{m=-\infty}^{+\infty} (-q)^{m^2} &&= 1 - 2.q + 2.q^4 - 2.q^9 + \ldots
\end{aligned}$$

Elles s'expriment donc comme un développement en puissance de q donnant le nombre de vecteurs sur les hypersphères successives autour de l'origine. Les séries Thêta des principaux RRP ont été tabulées par Conway et Sloane [Conway et al.93].

Cette première méthode analytique a été étendue pour le dénombrement de points à la surface :

- d'hyperpyramides. Les séries génératrices Nu sont introduites, elles sont l'équivalent des séries Thêta mais pour la norme L_1 (*i.e.* $\nu_\Lambda(z) = \sum_{\mathbf{y}\in\Lambda} z^{\|\mathbf{y}\|_1}$) [Gaidon93] ;
- d'hyperellipses. L'équation d'un point sur une telle surface de demi-axes $a_i.\sqrt{m}$ s'écrit $\sum_{i=1}^{k} \frac{y_i^2}{a_i^2} = m$ (avec $a_i > 0, \forall i$), c'est une norme L_2 pondérée. L'utilisation de séries génératrices Thêta modifiées est donc adoptées [Moureaux94].

Notons aussi que le nombre de points d'un RRP au sein d'un volume de forme complexe, est approximé en effectuant le rapport entre le volume global considéré et celui du Voronoï élémentaire. C'est un calcul approché car des Voronoï incomplets situés proche de la surface, et tels que leurs représentants soient à l'extérieur, sont pris en compte [Antonini91].

Réseaux réguliers meilleurs quantificateurs

Au chapitre 2 nous avons montré que, dans le cadre de l'hypothèse haute résolution, le paramètre G_k de l'équation de Zador (voir l'équation 2.12) s'interprète comme l'erreur quadratique moyenne minimale du quantificateur optimal. Pour une source uniforme le

k	1	2	3	4	5	6	7	8	12	16	24
Meilleur empilement	$\boxed{\mathbb{Z}}$	$\boxed{A_2}$	A_3	D_4	D_5	E_6	E_7	E_8	K_{12}	Λ_{16}	Λ_{24}
Plus grand nombre de contacts	$\boxed{\mathbb{Z}}$ 2	$\boxed{A_2}$ 6	$\boxed{A_3}$ 12	D_4 24	D_5 40	E_6 72	E_7 126	$\boxed{E_8}$ 240	P_{12a} 840	Λ_{16} 4320	$\boxed{\Lambda_{24}}$ 196560
Meilleur recouvrement	$\boxed{\mathbb{Z}}$	$\boxed{A_2}$	A_3^*	A_4^*	A_5^*	A_6^*	A_7^*	A_8^*	A_{12}^*	A_{16}^*	Λ_{24}
Meilleur quantificateur	$\boxed{\mathbb{Z}}$	$\boxed{A_2}$	A_3^*	D_4	D_5^*	E_6^*	E_7^*	E_8	K_{12}	Λ_{16}	Λ_{24}

TAB. 3.1 – *Résultats généraux: les réseaux dont les noms sont encadrés offrent la solution optimale globale (parmi les réseaux réguliers ou non) et ceux à gauche du trait vertical sont optimaux parmi les réseaux réguliers.*
La plupart de ces RRP appartiennent à la famille des réseaux par "strates" noté Λ_k et il existe de nombreuses équivalences. Nous indiquons simplement le nom de certains : $\mathbb{Z}^k$ ($k \geq 1$) est le réseau cubique, A_k ($k \geq 1$) celui "racine" ("root lattice"), A_2 est le réseau hexagonal, A_3 celui cubique à face centrée, Λ_{16} celui de Barnes-Wall et Λ_{24} celui de Leech.

dictionnaire est un RRP ; l'EQM est la même pour tous les Voronoï et les performances du quantificateur sont uniquement liées à la géométrie du polytope fondamental auquel tous les autres sont congrus. G_k devient alors le moment d'inertie normalisée d'ordre 2 du Voronoï élémentaire. Pour une dimension spatiale fixée, le RRP optimal pour la quantification vectorielle est donc celui dont le moment d'inertie est minimal. Cependant, là encore le problème est sans solution globale pour $k \geq 2$.
L'élément déterminant pour l'introduction des RRP, vient de la mise en oeuvre possible d'algorithmes de quantification rapide. Cette fois l'encodage n'est plus effectué par la recherche exhaustive au sein d'un dictionnaire du représentant le plus proche, mais en appliquant directement les calculs sur les composantes du vecteur à encoder. Conway et Sloane ont développé pour $\mathbb{Z}^k$ ($k \geq 1$), A_k ($k \geq 1$), D_k ($k \geq 2$), E_6, E_7, E_8, Λ_{16} et leurs réciproques, des algorithmes de quantification rapide [Conway et al.82a]. Pour notre application nous ne retenons et décrivons que les plus rapides (*i.e.* $\mathbb{Z}^k$, D_k, E_8 et Λ_{16}).
Le tableau 3.1 donne, en se limitant aux dimensions $k \leq 24$, les noms des RRP qui sont solutions aux différents problèmes évoqués.

3.2.2 Réseaux réguliers utilisés pour la quantification vectorielle

Réseau cubique $\mathbb{Z}^k$

Ce réseau dont les Voronoï sont des cubes, est constitué de l'ensemble des points de $\mathbb{R}^k$ de coordonnées entières (voir la figure 3.5) :

$$\mathbb{Z}^k = \{\mathbf{y} = (y_1,\ y_2,\ \ldots,\ y_k)^T \ \ / \ \ y_i \in \mathbb{Z}\}$$

Energie m de la surface de la sphère	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Nombre de points sur les sphères																	
de $\mathbb{Z}$	1	2	0	0	2	0	0	0	0	2	0	0	0	0	0	0	2
de $\mathbb{Z}^2$	1	4	4	0	4	8	0	0	4	4	8	0	0	8	0	0	4
de $\mathbb{Z}^3$	1	6	12	8	6	24	24	0	12	30	24	24	8	24	48	0	6
de $\mathbb{Z}^4$	1	8	24	32	24	48	96	64	24	104	144	96	96	112	192	192	24

TAB. 3.2 – *Nombre de points pour les premières sphères de* $\mathbb{Z}$, $\mathbb{Z}^2$, $\mathbb{Z}^3$, $\mathbb{Z}^4$.

C'est le réseau optimal en dimension 1. Nous donnons ses caractéristiques : $\det M = 1$, $\tau = 2.k$, $\rho = 1/2$, $r = \frac{\sqrt{k}}{2}$, $\Theta_{\mathbb{Z}^k}(z) = (\theta_3(z))^k$ (voir aussi le tableau 3.2).

Quantification rapide dans un réseau cubique $\mathbb{Z}^k$

Soit $\mathbf{x} = (x_1,\ x_2,\ \ldots,\ x_k)^T \in \mathbb{R}^k$ le vecteur à quantifier. Nous désirons lui associer le vecteur de reproduction $\mathbf{y} \in \mathbb{Z}^k$ le plus proche. Soit f la fonction qui, appliquée au réel x_i, nous rend l'entier le plus proche. Dans le cas où x_i est équidistant de deux entiers, f nous rend l'entier ayant la valeur absolue la plus petite (*i.e.* si $x_i = l_i + 0.5$ / $l_i \in \mathbb{Z}$: $f(x_i) = l_i$), ainsi le vecteur représentant d'énergie inférieure est toujours choisi [Bage86]. Nous avons :

$$\mathbf{y} = f(\mathbf{x}) = (f(x_1),\ f(x_2),\ \ldots,\ f(x_k))^T$$

Une quantification scalaire uniforme de pas unité sur chaque coordonnée du vecteur source, est donc effectuée.

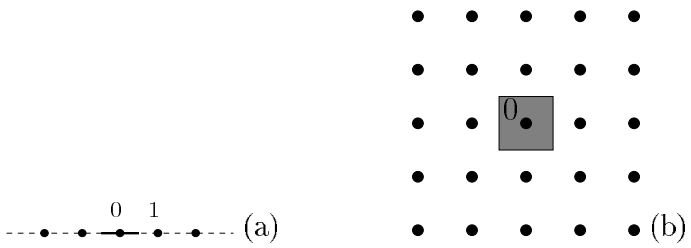


FIG. 3.5 – *Les réseaux* $\mathbb{Z}$ *(a) et* $\mathbb{Z}^2$ *(b).*

Réseau D_k $(k \geq 2)$

D_k est l'ensemble des points de $\mathbb{Z}^k$ dont la somme des coordonnées est paire (voir la figure 3.6) :

$$D_k = \left\{ \mathbf{y} = (y_1,\ y_2,\ \ldots,\ y_k)^T \ \ / \ \ \mathbf{y} \in \mathbb{Z}^k,\ \sum_{i=1}^{k} y_i = 0 \bmod(2) \right\}$$

Energie m de la surface de la sphère	0	2	4	6	8	10	12	14	16	18	20
Nombre de points sur les sphères de D_4	1	24	24	96	24	144	96	192	24	312	144

TAB. 3.3 – *Nombre de points pour les premières sphères de D_4.*

Nous donnons ses caractéristiques : $\det M = 2$, $\tau = 2.k.(k-1)$, $\rho = 1/\sqrt{2}$, $r = 1$ (pour $k = 2$ et 3) et $r = \rho.\sqrt{k/2}$ (pour $k \geq 4$), $\Theta_{D_k}(z) = \frac{1}{2}.\left(\theta_3(z)^k + \theta_4(z)^k\right)$ (voir le tableau 3.3). Ce réseau optimal en dimension 4, sert aussi à construire les réseaux E_8 et Λ_{16}.

Quantification rapide dans un réseau D_k

Soit $\mathbf{x} = (x_1,\ x_2,\ \ldots,\ x_k)^T \in \mathbb{R}^k$ le vecteur de la source à quantifier. Il faut lui associer le vecteur de reproduction $\mathbf{y} \in D^k$ le plus proche. Nous connaissons la fonction f qui associe au réel x_i l'entier le plus proche. Aussi $\delta(x_i) = x_i - f(x_i)$ correspond à l'erreur de quantification ($|\delta(x_i)| \leq 1/2$). Soit w la fonction qui, appliquée à x_i, nous rend le second entier le plus proche ("wrong way") :

$$w(x_i) = f(x_i) + \mathrm{sign}(\delta(x_i)) \quad \text{avec } \mathrm{sign}(z) = \begin{cases} 1 & \text{si } z \geq 0 \\ -1 & \text{si } z < 0 \end{cases}$$

Etant donné $\mathbf{x}$, soit l'entier n ($1 \leq n \leq k$) tel que $n = \arg\max_{1 \leq i \leq k} \delta(x_i)$ (si plusieurs possibilités existent, le plus petit n est choisi), x_n est donc la composante pour laquelle l'erreur de quantification est la plus grande. Soit aussi la fonction g telle que par rapport à $f(\mathbf{x})$, $f(x_n)$ soit remplacée par $w(x_n)$:

$$g(\mathbf{x}) = (f(x_1),\ f(x_2),\ \ldots,\ w(x_n),\ \ldots,\ f(x_k))^T$$

Les deux vecteurs $f(\mathbf{x})$ et $g(\mathbf{x})$ diffèrent par une seule composante, et la somme de leurs coordonnées diffère d'une unité. Le point appartenant à D_k est alors celui dont la somme des composantes est paire. La procédure à suivre pour trouver le point $\mathbf{y} \in D_k$ le plus proche de $\mathbf{x}$ devient :

(1) calcul de $f(\mathbf{x})$, si la somme de ses coordonnées est paire alors $\mathbf{y} = f(\mathbf{x})$,

(2) sinon, calcul de $g(\mathbf{x})$, alors $\mathbf{y} = g(\mathbf{x})$.

Dans le cas (1), il faut effectuer $2.k$ opérations : k arrondis (*i.e.* les $f(x_i)$), $k-1$ sommes (*i.e.* $\sum_{i=1}^{k} f(x_i)$), 1 test de parité. Dans le cas ((1) + (2)) il faut effectuer $3.k+2$ opérations en tout car il faut calculer en plus : k différences (*i.e.* les $\delta(x_i)$), une recherche de maximum et un arrondi (*i.e.* $w(x_i)$). La complexité de cet algorithme est donc de l'ordre de k. Contrairement aux algorithmes de type LBG il n'y a pas de norme à calculer, et le calcul est indépendant de la taille du dictionnaire.

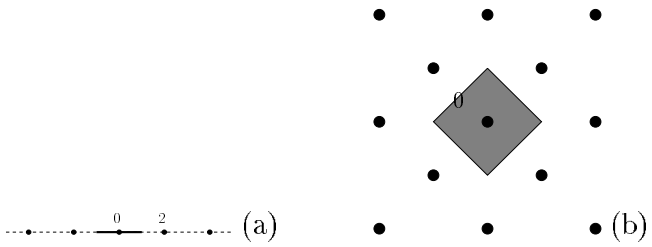


FIG. 3.6 – *Les réseaux D_1 (a) et D_2 (b). Pour ces dimensions D_k et $\mathbb{Z}^k$ sont équivalents à un facteur de dilatation et une rotation près.*

Réseau E_8

Le réseau E_8, encore appelé réseau en "diamant", est défini par la relation :

$$E_8 = D_8 \cup \left[[\frac{\mathbf{1}}{\mathbf{2}}] + D_8 \right] \text{ où } [\frac{\mathbf{1}}{\mathbf{2}}] = \left(\frac{1}{2},\ \frac{1}{2},\ \ldots,\ \frac{1}{2} \right)$$

Il correspond donc à l'union du réseau D_8 avec le réseau décalé (ou 'coset') $[\frac{\mathbf{1}}{\mathbf{2}}] + D_8$, soit encore :

$$E_8 = \left\{ \mathbf{y}' = (y'_1,\ y'_2,\ \ldots,\ y'_k)^T,\ \ \mathbf{y}'' = (y''_1,\ y''_2,\ \ldots,\ y''_k)^T \ \ / \right.$$

$$\left. y'_i \in \mathbb{Z} \text{ et } \sum_{i=1}^{8} y'_i = 0 \bmod(2),\ y''_i \in \left(\mathbb{Z} + \tfrac{1}{2}\right) \text{ et } \sum_{i=1}^{8} y''_i = 0 \bmod(2) \right\}$$

Nous avons : $\det M = 1, \tau = 240, \rho = 1/\sqrt{2}, r = 1$ et $\Theta_{E_8}(z) = \frac{1}{2}.\left(\theta_2(z)^8 + \theta_3(z)^8 + \theta_4(z)^8\right)$ (voir le tableau 3.4).

Quantification rapide dans un réseau décalé $\mathbf{r} + \Lambda$

La procédure Φ permettant de trouver le plus proche voisin du vecteur $\mathbf{x}$ dans le réseau Λ, peut-être étendue afin de déterminer le point le plus proche dans le réseau décalé $\mathbf{r}+\Lambda$ car si $\Phi(\mathbf{x})$ est le point le plus proche de $\mathbf{x}$ dans Λ, $\Phi(\mathbf{x}-\mathbf{r})+\mathbf{r}$ est le point le plus proche dans $\Lambda + \mathbf{r}$. Nous généralisons en considérant une union de réseaux décalés $\mathcal{L} = \bigcup_{i=0}^{l-1} (\mathbf{r}_i + \Lambda)$. $\mathbf{x}$ étant le vecteur à quantifier, la méthode devient :

(1) calcul de chaque $\mathbf{y}_i = \Phi(\mathbf{x} + \mathbf{r}_i)$ $(i = 0,\ 1,\ \ldots,\ l-1)$;

(2) comparaison de chacun des $\mathbf{y}_0,\ \mathbf{y}_1,\ \ldots,\ \mathbf{y}_{l-1}$ avec $\mathbf{x}$, et choix du plus proche au sens de la norme euclidienne (l calculs de distorsion sont donc nécessaires).

Energie m de la surface de la sphère	0	2	4	6	8	10	12	14	16	18	20
Nombre de points sur les sphères de E_8	1	240	2160	6720	17520	30340	60480	82560	140400	181680	272160

TAB. 3.4 – *Nombre de points pour les premières sphères de E_8.*

Quantification rapide dans un réseau E_8

Le réseau E_8 étant l'union des réseaux D_8 et D_8 décalé de $[\frac{\mathbf{1}}{\mathbf{2}}]$, la procédure de quantification du vecteur $\mathbf{x} \in \mathbb{R}^8$ devient :

(1) calcul des vecteurs $f(\mathbf{x})$ et $g(\mathbf{x})$, puis sélection de celui dont la somme des coordonnées est paire. Soit $\mathbf{y}_0$ ce vecteur ;

(2) calcul des vecteurs $f(\mathbf{x} - [\frac{\mathbf{1}}{\mathbf{2}}])$ et $g(\mathbf{x} - [\frac{\mathbf{1}}{\mathbf{2}}])$, puis sélection de celui ayant une somme de coordonnées paire. Ce vecteur auquel est ajouté $[\frac{\mathbf{1}}{\mathbf{2}}]$ est $\mathbf{y}_1$;

(3) calcul des normes $d(\mathbf{y}_0, \mathbf{x})$ et $d(\mathbf{y}_1, \mathbf{x})$. Le vecteur $\mathbf{y}_i$ le plus proche de $\mathbf{x}$ est retenu.

La complexité de quantification est la somme de celles suivantes : complexité de la quantification dans D_8, celle de la quantification dans D_8 décalé (*i.e.* le décalage, la quantification dans D_8 et le décalage inverse) et celle de la recherche du minimum des deux normes. Dans le cas le plus défavorable 129 opérations par vecteur sont nécessaires [Gaidon93]. Cette complexité demeure néanmoins inférieure à celle des algorithmes de type LBG.

Réseau de Barnes-Wall Λ_{16}

Ce réseau est défini par :

$$\Lambda_{16} = \bigcup_{i=0}^{31} (\mathbf{r}_i + 2.D_{16})$$

où les vecteurs de translation $\mathbf{r}_i$ correspondent aux lignes (ou colonnes) d'une matrice de Hadamard de type Sylvester dans laquelle est effectuée le changement $(-1,\ 1) \rightarrow (1,\ 0)$, et aux lignes (ou colonnes) de la matrice complémentée.
Nous indiquons que : $\det M = 16$, $\tau = 4320$, $\rho = 1$, $r = \rho.\sqrt{3}$ (voir aussi le tableau 3.5).

Quantification rapide dans un réseau Λ_{16}

Ce réseau se présente comme l'union de 32 réseaux D_{16} décalés, il faut donc procéder ainsi : recherche du plus proche voisin dans chacun des réseaux décalés, puis choix parmi les 32 représentants possibles de celui le plus proche de $\mathbf{x}$ selon la norme L_2.

Energie m de la surface de la sphère	0	2	4	6	8	10	12	14	16	18
Nombre de points sur les sphères de Λ_{16}	1	0	4320	61440	522720	2211840	8960640	23224320	67154400	135168000

TAB. 3.5 – *Nombre de points pour les premières sphères de* Λ_{16}.

Conclusion

Les réseaux réguliers de points optimaux pour la quantification vectorielle ne sont connus que pour les dimensions 1, 2, 4, 8, 16 et 24. Leur avantage décisif réside dans l'existence pour chacun d'eux, d'un algorithme de quantification rapide où les calculs, indépendants de la taille du dictionnaire, sont opérés directement sur les coordonnées du vecteur à encoder. Néanmoins ces algorithmes se complexifient rapidement dès que la dimension croît.
L'optimalité de ces réseaux est déterminée par rapport à la minimisation de l'EQM liée à la quantification d'une source i.i.d uniforme. Seul le gain de partitionnement est à ce niveau amélioré, et nous avons vu qu'il ne peut-être que faible en pratique. Il est donc nécessaire d'adapter la QVA à la quantification de sources non-uniformes, et de tirer profit des autres gains de la QV. Il est alors procédé à une troncature du réseau en fonction de la répartition spatiale des vecteurs source. C'est cette étape du choix du dictionnaire au sein du RRP que nous allons présenter.

3.3 Choix de la forme du dictionnaire

Le RRP non tronqué définit un ensemble potentiel de représentants en nombre infini. Le débit alloué au quantificateur étant limité, il faut restreindre le nombre de points du réseau formant le dictionnaire (cette remarque évidente si le code construit est à longueur fixe, demeure aussi valide si celui-ci est entropique, à moins que la source ne soit complètement stationnaire et le débit très élevé). Il convient de tirer profit de la liberté offerte par la QV pour déterminer la forme de la région de l'espace à conserver (la QS ne permet de répartir les représentants que sur un intervalle). La base théorique pour définir la forme de troncature du RRP en fonction de la répartition des vecteurs source, est la propriété d'**équirépartition asymptotique** que nous allons expliciter.
Supposons une source sans mémoire dont les échantillons $\{x(n)\}$ obéissent à la loi marginale $p_X(x)$; la densité de probabilité conjointe $p^*_{\mathbf{X}}(\mathbf{x})$ du vecteur $\mathbf{x}$ est alors le produit des marginales. Nous avons les égalités :

$$-\frac{1}{k}.\log_2 p^*_{\mathbf{X}}(\mathbf{x}) = -\frac{1}{k}.\log_2(\prod_{i=1}^{k} p_X(x_i)) = -\frac{1}{k}.\sum_{i=1}^{k}\log_2(p_X(x_i))$$

La loi des grands nombres entraîne la version suivante du théorème de Shannon-McMillan-Breiman [Barron85]:

$$\text{si } k \to +\infty : -\frac{1}{k}.\sum_{i=1}^{k} \log_2 p_X(x_i) \longrightarrow -E(\log_2 p^*_{\mathbf{X}}(\mathbf{x})) = -\int_{-\infty}^{+\infty} p^*_{\mathbf{X}}(\mathbf{x}).\log_2 p^*_{\mathbf{X}}(\mathbf{x})\ d\mathbf{x} = h(X)$$

$h(X)$ est l'entropie différentielle de la source (voir l'annexe B) et la convergence a lieu au sens des probabilités. Finalement :

$$\text{si } k \to +\infty : -\frac{1}{k}.\log_2 p^*_{\mathbf{X}}(\mathbf{x}) \approx h(X) \Longleftrightarrow p^*_{\mathbf{X}}(\mathbf{x}) \approx 2^{-k.h(X)}$$

Nous pouvons interpréter le résultat: si k est grand, la probabilité est concentrée sur les vecteurs pour lesquels $p^*_{\mathbf{X}}(\mathbf{x}) \approx 2^{-k.h(X)}$. *Autrement dit, des vecteurs de grande dimension d'une source sans mémoire ont avec une forte probabilité une densité constante, et ils sont distribués approximativement uniformément dans la région compacte de l'espace où* $p^*_{\mathbf{X}}(\mathbf{x})$ *est égale à* $2^{-k.h(X)}$.

La propriété d'équirépartition asymptotique décrit des conditions optimales pour la quantification où les vecteurs à coder sont localisés dans une zone compacte de l'espace, et leur distribution quasiment uniforme. Alors le dictionnaire constitué des points du RRP ne recouvrant que cette région où sont distribués les vecteurs à coder, est parfaitement adapté [Gersho79] [Gersho et al.92].

Considérons l'exemple le plus simple d'une source i.i.d gaussienne pour laquelle $h(X) = \log_2 \sqrt{2.\pi.\sigma_X^2.e}$ (voir l'annexe B). Soit $\mathbf{B}$ la région de l'espace telle que :

$$\mathbf{B} = \left\{ \mathbf{x} : p^*_{\mathbf{X}}(\mathbf{x}) = 2^{-2.k.h(X)} \right\}$$

Nous calculons :

$$p^*_{\mathbf{X}}(\mathbf{x}) = 2^{-2.k.h(X)} \Longleftrightarrow (\frac{1}{2.\pi.\sigma_X^2})^{\frac{k}{2}}.e^{-\frac{\sum_{i=1}^{k} x_i^2}{2.\sigma_X^2}} = 2^{-\frac{k}{2}.\log_2(2.\pi.\sigma_X^2.e)} \Longleftrightarrow \frac{1}{k}.\sum_{i=1}^{k} x_i^2 = \sigma_X^2$$

Dans le cas asymptotique les vecteurs équiprobables sont donc répartis sur une hypersphère de rayon σ_X pondéré par $1/k$. Une étude plus poussée montre que quelque soit la dimension, les vecteurs de cette source sont distribués exactement dans une hypercouronne.
Ce dernier résultat conduit à trois façons de délimiter le réseau dans le cas d'une source gaussienne [Antonini91] [Jeong et al.93] [Barlaud94]: la surface d'une hypersphère, le volume d'une hyperboule ou une hypercouronne. Le choix de l'un ou l'autre dictionnaire est conditionné par la possibilité de réaliser le meilleur compromis débit/distorsion (*i.e.* réunir le minimum acceptable de points appartenant au dictionnaire, et avoir un minimum de

vecteurs de la source hors de ce dictionnaire).

Dans le cas d'une source i.i.d laplacienne un raisonnement identique mais adoptant la norme L_1, montre que les vecteurs équiprobables se répartissent sur des hyperpyramides. Celles-ci définissent donc la forme de troncature du RRP [Fisher86] [Jeong et al.93] [Gaidon93] [Barlaud et al.94] [Moureaux94].
Pour des sources (gaussiennes) corrélées, les vecteurs sont distribués au sein d'ellipses orientées dont les axes sont portés par des bissectrices (par rapport au système de coordonnées cartésiennes). Ce résultat conduit à une troncature de forme elliptique du dictionnaire [Fisher89] [Moureaux94], cependant ce dernier ne peut lui-même être orienté dans l'espace. Une rotation (*i.e.* une décorrélation) est donc nécessairement appliquée aux vecteurs de la source avant quantification afin de les amener sur les axes principaux du repère cartésien (la transformation inverse est évidemment réalisée après quantification).
Ces distributions gaussiennes généralisées (corrélées ou non) conduisent intrinsèquement à des dictionnaires de formes symétriques facilitant le dénombrement des représentants (*i.e.* la détermination du nombre et de la position des points sur les surfaces isonormes du RRP tronqué). Cette dernière opération est indispensable pour indexer les vecteurs représentant. Pour chacune de ces distributions, l'adoption d'une norme adaptée fixe la forme du dictionnaire, puis le choix de l'énergie de troncature $\mathcal{E}_t$ établit le nombre de points conservés et par conséquent le débit. Si le code est à longueur fixe ce dernier résultat s'impose (voir l'équation 2.3). Si le code est entropique ce résultat est toujours valable car, lorsque $\mathcal{E}_t$ et donc le nombre de représentants croissent, la source se répartit sur un ensemble de Voronoï de taille réduite et ainsi l'entropie du dictionnaire augmente (dans le cas asymptotique tous les Voronoï deviennent équiprobables).

Nous concluons en soulignant que les techniques actuelles de troncature des RRP offrent peu de liberté pour la QVA de sources complexes (qui ne sont pas simplement modélisables). Il faut ajouter que la QVA sur des ellipses, mieux adaptée au codage des sous-bandes des images, nécessite une étape supplémentaire de décorrélation. Notons enfin que les dimensions vectorielles pratiques des réseaux (i.e. *typiquement D_4 et E_8) sont éloignées de celle asymptotique d'équirépartition. Pour répondre à ces limites nous proposons, dans le cadre de la compression de séquences d'images, une solution originale avec la quantification vectorielle algébrique et arborescente (***QVAA***, voir le chapitre 5).*

3.4 Adaptation de la source au dictionnaire

Dans l'étape précédente de détermination de la forme de troncature et de son énergie $\mathcal{E}_t$, l'énergie de la source n'est pas intervenue. Nous allons montrer qu'il est aussi nécessaire d'adapter les vecteurs à coder au dictionnaire.
Les algorithmes de quantification rapide opérant dans le domaine normalisé du réseau, il est choisi de projeter chaque vecteur source à l'intérieur du dictionnaire. Les vecteurs de reproduction finaux seront obtenus par renormalisation à l'aide de la fonction inverse de celle de projection. Le but est alors de trouver la fonction de normalisation, linéaire ou

non, assurant la minimisation de la distorsion résultant de la quantification.

3.4.1 Fonctions de normalisation

De façon intuitive déterminer la fonction de normalisation de la source consiste à agir sur le "pas de quantification" du QVA. Nous donnons différentes approches proposées pour définir ce facteur de projection.

A haut débit (*i.e.* avec l'hypothèse haute résolution) les approximations possibles permettent à Jeong et Gibson [Jeong et al.93] [Jeong et al.95] de déterminer les fonctions linéaires optimales de projection, ceci pour la QVA avec le réseau cubique de sources i.i.d gaussiennes ou laplaciennes. Mais ces conditions asymptotiques ne considèrent pas les contraintes de bas débits que nous rencontrons.
Pour des débits inférieurs et toujours en considérant le réseau $\mathbb{Z}^k$, Antonini [Antonini et al.95] développe un approche permettant d'obtenir pour différents facteurs de normalisation la distorsion (*i.e.* un bruit granulaire) et le débit. La QVA est alors contrainte en entropie et la source modélisée par une loi i.i.d centrée (l'auteur développe le cas gaussien).
Onno [Onno et al.95] décrit une méthode différente pour la conception d'un QVA avec contrainte entropique. L'idée n'est plus seulement de réduire de façon globale la distorsion pour un débit donné, mais d'introduire ces contraintes pour chaque vecteur à quantifier. En effet la différence du nombre de points sur des hypersphères proches, se traduit par une différence du coût de codage pour leurs vecteurs respectifs. Le quantificateur est alors amené à ne plus forcément choisir dans le RRP le plus proche voisin du vecteur source, mais celui environnant offrant le meilleur compromis débit *vs.* distorsion. Finalement un algorithme d'allocation binaire tenant compte de toutes les contraintes est obtenu. Le processus est relativement coûteux car il doit procéder par itérations successives afin de converger vers le débit souhaité.

Pour différents débits des fonctions non-linéaires de normalisation ont aussi été produites. Elles agissent en adaptant radialement la répartition des vecteurs source au sein du dictionnaire. De manière à mieux appréhender l'action de ces fonctions, il est plus simple de considérer que ce sont les tailles des Voronoï qui sont modifiées ; alors ces derniers sont dilatés là où la densité des points à coder est faible, et inversement contractés dans la région de l'espace où cette densité est forte (il s'agit de la généralisation au cas vectoriel du "companding" scalaire).
Ainsi Kuhlmann et Bucklew [Kuhlmann et al.88] proposent (sans préciser une règle précise pour la découpe de l'espace) d'appliquer sur chaque coordonnée du vecteur source une fonction non-linéaire par morceaux adaptant différents pas de quantification.
Swaszek [Swaszek92] reprend l'idée précédente et conçoit un QVA multi-étages avec le réseau cubique : à chaque étage la quantification devient plus fine dans un hypercube au centre de l'espace. Mais il n'existe aucune stratégie pour définir le nombre d'étages du quantificateur. Notons que cet auteur annonce l'idée d'emboîtage de réseaux que nous exploitons pour la QVAA (voir le chapitre 5).

Pour les hauts débits, Jeong et Gibson [Jeong et al.93] proposent d'exploiter la symétrie des signaux de Gauss et de Laplace, afin d'adapter radialement les vecteurs source. Le développement de la fonction non-linéaire est obtenu pour le réseau cubique et toujours en s'appuyant sur l'hypothèse haute résolution.
Enfin Lebedeff [Lebedeff95] propose pour le codage d'un signal i.i.d gaussien une fonction de normalisation opérant sur la norme des vecteurs, (la QVA est réalisée avec les réseaux D_4 et E_8). L'optimisation du facteur de projection (constant ou non) est obtenue de façon expérimentale : l'algorithme procède par itération en minimisant à chaque étape l'EQM relative à la quantification d'un séquence d'apprentissage. Cette optimisation d'un coût calculatoire élevé est faite hors ligne.

3.4.2 Projection de la source dans une hyperboule

Le développement de cette méthode simple qui est adoptée par la QVAA, permet d'exposer les problèmes typiques rencontrés lors de la mise en oeuvre d'un quantificateur vectoriel algébrique. Cette technique consiste donc à projeter le vecteur à coder à l'intérieur du dictionnaire formé des points du réseau tronqué par l'hypersphère de rayon $\sqrt{\mathcal{E}_t}$ (la généralisation aux troncatures de forme pyramidale ou elliptique se fait par adaptation de la norme).

Soit $\mathcal{SA} = \{\mathbf{x}_j = (x_1, \ldots, x_k)^T \ / \ j = 0, 1, 2, \ldots\}$ la séquence d'apprentissage modélisant la source, et $\mathcal{E}_{max}$ l'énergie maximale probable d'un vecteur à coder, alors :

$$\mathcal{E}_{max} = \max_{\mathcal{E}} \{\mathcal{E}(\mathbf{x}) \ / \ \mathbf{x} \in \mathcal{SA}\}$$

et :

$$\mathcal{E}(\mathbf{x}) = L_2(\mathbf{x}) = \sum_{i=1}^{k} x_i^2$$

Tous les vecteurs source sont alors à l'intérieur de l'hypersphère en appliquant à leurs coordonnées le **facteur d'échelle** F :

$$F = \sqrt{\frac{\mathcal{E}_t}{\mathcal{E}_{max}}}$$

En effet nous avons :

$$\sum_{i=1}^{k} (F.x_i)^2 = F^2 . \sum_{i=1}^{k} x_i^2 = \frac{\mathcal{E}_t}{\mathcal{E}_{max}} . \mathcal{E} \leq \mathcal{E}_t$$

Les vecteurs représentants sont renormalisés simplement par $1/F$. Seul un bruit granulaire est engendré lors de la quantification de la séquence d'apprentissage. Une fois le dictionnaire construit *a priori*, lors du codage de la source réelle, un vecteur normalisé marginal (*i.e.* d'énergie importante) peut demeurer hors du réseau tronqué. Deux conséquences pratiques surviennent car il faut mettre en place :

- un test afin de déterminer si le vecteur à coder appartient ou non au dictionnaire (*i.e.* un test réunissant un calcul de norme et une comparaison avec $\mathcal{E}_t$) ;

– un traitement particulier pour quantifier séparément les vecteurs marginaux. Les méthodes explorées sont décrites ci-après, notons que la distorsion engendrée par cette quantification est le bruit de surcharge :

  . la première approche réside dans le codage du vecteur marginal par le point du réseau le plus proche (qui n'appartient pas au dictionnaire), et transmettre séparément l'information relative à ce représentant. Cette approche qui minimise le bruit est évidemment trop coûteuse ;

  . la seconde approche consiste à recalculer le facteur d'échelle de façon à projeter le vecteur marginal sur la surface de l'hypersphère de troncature (*i.e.* obtenir $F^* = \sqrt{\mathcal{E}^*/\mathcal{E}_t}$ où $\mathcal{E}^*$ est l'énergie du vecteur). Ce vecteur est ensuite quantifié par un point du dictionnaire. Le facteur de projection F^* est aussi transmis avec l'index du représentant afin de renormaliser ce dernier de façon propre [Barlaud et al.94] [Gaidon93] [Moureaux94]. Cette méthode ne dissociant pas la transmission de facteurs d'échelle de celle d'index est particulièrement complexe. Elle devient rapidement coûteuse en terme de débit dès que les vecteurs traités, en principe marginaux, apparaissent plus nombreux ;

  . la façon la plus simple de procéder est certainement de réaliser, comme précédemment, une projection du vecteur marginal sur la surface du dictionnaire (*i.e.* en appliquant F^*). Mais ce vecteur sera traité au décodage comme un vecteur commun et renormalisé par $1/F$. Il n'y a donc pas d'information supplémentaire transmise, le bruit de surcharge créé peut cependant devenir conséquent. La simplicité de cette approche permet un réglage plus aisé des paramètres du quantificateur (*i.e.* F et $\mathcal{E}_t$).

L'exemple simple décrit permet de soulever un dernier problème lié à la normalisation de la source par un facteur d'échelle. En effet un vecteur projeté en surface ou proche de la surface de l'hypersphère d'énergie $\mathcal{E}_t$, peut avoir son plus proche représentant à l'extérieur du dictionnaire (voir la figure 3.7). Les solutions proposées sont :

– la re-projection du vecteur davantage à l'intérieur de l'hyperboule (*i.e.* par un nouveau facteur d'échelle) ;

– la recherche parmi les points appartenant au dictionnaire de celui le plus proche du vecteur projeté à la surface. Cette approche est complexe mais il est montré que le représentant recherché est nécessairement un des points entourant celui le plus proche du vecteur projeté (mais extérieur au dictionnaire).

3.4.3 Conclusion

L'adaptation de la source au dictionnaire est effectuée de façon générale par normalisation par un facteur d'échelle constant ou variable. Les vecteurs source sont ainsi projetés au sein du RRP tronqué, puis ils sont quantifiés par l'algorithme de quantification rapide correspondant. Cependant les fonctions de normalisation proposées sont adaptées aux seules

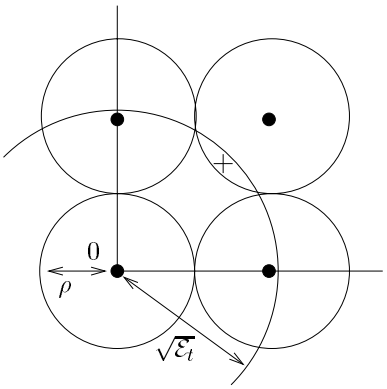


FIG. 3.7 – *Exemple bidimensionnel où le point normalisé à quantifier (la croix) est représenté par un point du réseau extérieur à la sphère d'énergie* $\mathcal{E}_t$.

distributions de sources gaussiennes ou laplaciennes (corrélées ou non). Pour de bas débits une optimisation expérimentale du facteur de projection convient, mais le coût calculatoire requis est élevé. Il est de plus nécessaire de tester tous les vecteurs avant quantification, et les vecteurs marginaux demeurant hors du dictionnaire après projection doivent recevoir un traitement particulier. Nous verrons que la QVAA basée sur une stratégie d'emboîtage de réseaux, est une approche simple et peu coûteuse répondant à ces problèmes.
L'ultime étape est alors l'indexage des points du dictionnaire.

3.5 Indexage des points du dictionnaire

L'indexage consiste à associer à chacun des points du RRP tronqué, un index (ou indice) qui sera codé et transmis au décodeur. Si la QVA apporte une solution à la coûteuse recherche exhaustive au sein du dictionnaire, la difficulté s'est déplacée car l'indexage est devenue une étape complexe. En effet pour un algorithme de type LBG, les indices sont obtenus immédiatement car ils sont stockés avec les représentants dans un tableau, et ce dernier a été parcouru lors de la recherche du plus proche voisin. Avec la QVA le représentant est calculé directement à partir des composantes à coder, et il faut ensuite calculer l'index correspondant (il n'est pas envisageable de parcourir une liste de transcodage).
Les méthodes pour indexer les points du dictionnaire du QVA sont complexes et font l'objet de nombreuses recherches. Nous proposons d'en décrire quelques unes.

3.5.1 Indexage dans la base canonique

C'est l'approche la plus simple qui consiste à indexer les vecteurs par leurs composantes dans la base canonique. Pour les réseaux D_k et E_8, le problème lié au demi-composantes est levé en les multipliant par 2. Cette base indexe tous les points inscrits au sein d'un hypercube, un dictionnaire constitué des points d'un RRP tronqué (*i.e.* de forme sphérique, pyramidale ou elliptique) ne compte qu'une partie de l'ensemble des points de l'hypercube (*e.g.* une hyperboule de rayon 2 dans $\mathbb{Z}^4$ compte 89 points, l'hypercube la contenant a 5^4

points, soit environ 7 fois plus). Cette méthode est donc inadaptée si le dictionnaire est un RRP tronqué du fait du grand nombre de points inutilisés.

3.5.2 Indexage par l'algorithme de Conway et Sloane

Conway et Sloane ont proposé un algorithme d'indexage [Conway et al.83] où les points du RRP pris en compte, sont situés au sein d'une région ayant la forme du Voronoï élémentaire (*i.e.* celui-ci est dilaté). Les index sont calculés dans la base génératrice du réseau afin que les coordonnées des vecteurs soient entières. Excepté dans le cas du réseau cubique, le nombre de points considérés est moins important que précédemment. Mais il demeure un nombre non négligeable de points inutilisés si le dictionnaire est un RRP tronqué.

3.5.3 Indexage basé sur un code produit

Un code produit (ou préfixe) est la concaténation de deux index, chacun désignant une caractéristique particulière du point du réseau. Cette approche convient particulièrement aux dictionnaires de formes symétriques [Fisher86] [Lamblin et al.88] [Moureaux94] constitués d'hypersphères, d'hyperpyramides ou d'hyperellipses concentriques. L'identification d'un point est alors précisée par sa norme et sa position (ou phase) sur la surface d'équiprobabilité.

L'indexage de la norme est simple. Le code, fonction du nombre d'hyperboules dans le dictionnaire, est le plus souvent choisi à longueur fixe (le cas le plus favorable est évidemment celui où ce nombre est une puissance de 2). Un codage entropique de la norme est intéressant lorsque le nombre d'hyperboules concentriques est faible (car leurs probabilités d'apparition seront fortes), il est aussi adapté aux sources suivant une loi gaussienne généralisée où la probabilité d'occurence des vecteurs est forte à l'origine puis décroît rapidement.

L'indexage de la position demeure complexe, car le code est fonction des positions et des nombres variables de points sur les surfaces isonormes (ces nombres ne sont d'ailleurs jamais des puissances de 2). Les approches visent à réaliser le meilleur compromis entre le coût de calcul et celui de stockage de tableaux de transcodage. Nous donnons quelques exemples :

- Lamblin [Lamblin et al.88] décrit un algorithme pouvant être généralisé à tous les réseaux possèdant la propriété suivante : une permutation des coordonnées d'un vecteur donne toujours un autre point du réseau. Cette méthode réalise une partition de l'ensemble des points de la surface en classes, chacune possèdant un "leader". Ce dernier est le vecteur particulier dont la permutation des composantes restitue tous les autres points de la classe (*e.g.* dans D_4 l'hypersphère de norme 4 comptant 24 points, possède 7 leaders). La position d'un vecteur de la classe est alors déterminée par l'ordre des permutations restituant le "leader". L'indexage de la position d'un

point devient la recherche (rapide) de son leader puis celle de l'ordre des permutations. Un compromis efficace entre le coût de calcul et celui de stockage est finalement réalisé ;

- Moureaux [Moureaux94] propose une réduction du coût calculatoire par l'introduction d'un table de correspondance. Celle-ci dont les adresses sont fonction des composantes des vecteurs quantifiés, restitue en sortie le code à transmettre et cela pour les seuls points utilisés lors du codage dans le réseau tronqué. Il faut aussi transmettre au décodeur la table ; cependant la taille de cette dernière devient rapidement imposante (bien qu'elle soit réduite en tenant compte de la symétrie axiale des réseaux). La construction d'un code préfixe à longueur variable est alors proposée [Moureaux et al.94] en tenant compte des nombres respectifs de points sur les surfaces isonormes (au décodeur la connaissance de la norme d'une surface a permis d'en déduire immédiatement le nombre de points, et donc la longueur du code de position qui suit). Cette approche bien adaptée au codage de sources symétriques, reste peu robuste aux erreurs de transmission car les erreurs se propagent ;

- Onno [Onno et al.95] décrit une nouvelle méthode permettant de tenir compte de la statistique du signal quantifié. La notion de "type" est introduite afin de classer les vecteurs suivant leur nombre de composantes indépendantes. Or pour les sous-bandes du signal image, certains "types" sont plus probables que d'autres (ils correspondent aux vecteurs très corrélés). Tous les vecteurs de même type apparaissent par contre équiprobables. Finalement un index de position compte deux codes : un entropique pour le "type", et un autre naturel pour l'ordre du vecteur dans la classe.

La complexité de la QVA ne réside plus dans la recherche du plus proche voisin du vecteur source, mais au niveau de l'indexage des points du dictionnaire. Les algorithmes sophistiqués sont surtout conçus pour l'indexage des points de dictionnaires représentant des sources de Gauss ou de Laplace (corrélées ou non). La complexité d'indexage limite aussi l'utilisation de RRP de hautes dimensions. Nous ajoutons que les dictionnaires comptent généralement un grand nombre de points. Il en résulte une surquantification du signal, et de bas débits ne sont atteints que si un code entropique est construit pour transmettre les index. La modélisation de distributions multidimensionnelles de sources naturelles étant délicate, il faut faire appel à l'apprentissage empirique afin d'accéder aux probabilités d'apparition des représentants.
La QVAA que nous développons, propose dans le cadre de la quantification à bas débits de sources complexes, une solution reposant sur le codage d'un arbre et d'index.

3.6 Conclusion

Les éléments déterminants pour l'introduction de la QVA sont certainement la mise en oeuvre d'algorithmes de quantification rapide, et la non-transmission de l'ensemble des représentants virtuels. Si l'objectif de réduction de la complexité pour la recherche au sein du dictionnaire est remplie, la difficulté s'est en fait transférée au niveau de la limitation de la taille des réseaux et de l'indexage de leurs points. Les solutions proposées sont

adaptées à la quantification de sources symétriques simplement modélisées, car le choix judicieux d'une métrique permet de tronquer le RRP et de dénombrer ses points. Néanmoins ces méthodes demeurent sophistiquées, et les dimensions vectorielles des réseaux sont limitées. Il faut ajouter qu'un test de la norme des vecteurs est aussi mis en place avant quantification, ainsi qu'un traitement particulier pour les points demeurant hors du dictionnaire. Enfin une technique apprentissage est fréquemment retenue pour réaliser le codage entropique des représentants.

Pour répondre à ces limites, nous proposons une approche reposant sur la construction d'une structure de réseaux tronqués et emboîtés. La stratégie de conception du dictionnaire (voir le chapitre 5) conduit à une découpe adaptée de l'espace car fonction de la distribution de la source, et fonction d'un critère débit *vs.* distorsion. Des solutions simples sont produites pour le test des vecteurs à coder ainsi que pour le traitement de ceux hors norme. Enfin l'indexage des représentants est basée sur la structure arborescente de ce dictionnaire.

Avant de développer la conception du QVAA (voir le chapitre 5), nous proposons donc d'étudier la quantification vectorielle arborescente au chapitre 4.

Chapitre 4

Quantification vectorielle arborescente

4.1 Introduction

La quantification vectorielle arborescente (**QVAr**) regroupe les nombreuses approches de quantification où le codage est effectué à l'aide d'un arbre de décision (*e.g.* QV multi-étages, en cascade, par code produit [Gersho et al.92] [Barnes et al.96]). Le but est toujours de réduire la complexité inhérente à la recherche exhaustive au sein d'un dictionnaire de grande taille (cette dernière méthode offrant la quantification optimale). La QVAr offre trois avantages prépondérants :

- la réduction de la **charge de calcul** car la recherche du vecteur de reproduction est effectuée par étapes parmi des ensembles réduits de représentants (la recherche au sein de sous-dictionnaires est accélérée, de plus la séquence d'apprentissage éventuellement nécessaire pour construire un sous-dictionnaire est de taille réduite) ;
- une structure adaptée à une **transmission progressive** de l'information telle que le signal soit reconstruit par une série d'approximations successives [Wang et al.89] [Hwang et al.95] ;
- une structure appropriée pour un codage à **débit variable**.

Concernant ce dernier point, déjà dans le cas le plus simple où le code est à longueur fixe, il suffit de construire un arbre dont les code-vecteurs sont à différentes hauteurs pour obtenir le débit variable. Il est aussi judicieux d'introduire un critère débit-distorsion pour la construction de ce dictionnaire arborescent non-équilibré. L'objet de ce chapitre est donc de décrire les différentes approches possibles pour la conception de tels QVAr.

Le QVAA que nous décrivons au chapitre 5 est précisément un QV en cascade où l'erreur résiduelle d'un étage est aussitôt re-quantifiée par un sous-dictionnaire individuel (voir la figure 4.1). Dans [Kossentini et al.95] [Barnes et al.96] les conditions optimales pour la construction de ce QV sont rappelées, de telle façon que la partition globale (*i.e.* la somme

des partitions partielles obtenues avec les sous-dictionnaires) restitue celle qui serait donnée par une recherche exhaustive dans un seul et même dictionnaire. La conception du QV correspondant est aussi décrite: il ne faut pas construire indépendamment les sous-dictionnaires mais tenir compte pour chaque étage de l'erreur globale faite (*i.e.* pour un étage donné, tenir compte de l'erreur causale due à la quantification des étages précédents, et de celle anti-causale de ceux suivants). A partir d'une séquence d'apprentissage et d'un dictionnaire initial, il s'agit d'actualiser alternativement les sous-dictionnaires (plusieurs stratégies sont possibles [Chan et al.92] [Barnes et al.93] [Kossentini et al.95], mais à chaque fois plusieurs boucles sur l'ensemble des sous-dictionnaires sont requises) en appliquant le paradigme fondamental introduit par l'algorithme de Lloyd généralisé (voir le chapitre 2). Néanmoins le dictionnaire obtenu n'est que localement optimal car les règles d'encodage ne peuvent-être que sous-optimales [Rose et al.96].
Cette étude [Barnes et al.96] montre pourquoi la construction d'un QV en cascade ne peut se faire par application séparée à chaque étage d'un algorithme de Lloyd généralisé, car la succession de ces QV indépendants ne produit qu'une quantification médiocre et inorganisée [Makhoul et al.85] [Gersho et al.92] (*i.e.* un Voronoï à un étage n'est pas la somme des Voronoï des étages inférieurs, et ces derniers sont parfois disjoints, non-convexes ou non-connectés). Cependant l'approche (localement) optimale décrite pour la QV en cascade est beaucoup trop coûteuse pour notre application. Les méthodes que nous explorons sont celles qui imposent une contrainte structurelle supplémentaire sur le dictionnaire (afin d'accélérer la recherche et la construction). Nous retrouvons notamment les techniques de dictionnaires structurés en réseaux [Swaszek92] [Pan et al.95], et c'est parmi celles-ci que vient s'inscrire la QVAA. Avec cette dernière, le dictionnaire est obtenu par emboîtage d'une hiérarchie de RRP tronqués; il en résulte une structure arborescente simple (voir le chapitre 5), et toutes les méthodes que nous décrivons dans ce chapitre 4 sont directement transposables.

4.2 Définitions et principes

Définitions

La première étape est de donner les définitions relatives aux arbres; le formalisme décrit sera conservé jusqu'à la fin du manuscrit.

La figure 4.2 présente l'exemple d'un **arbre** B-aire où $B = 2$ (*i.e.* c'est un arbre binaire), d'un noeud père partent alors B arêtes vers les noeuds fils. Cet arbre est planté (il a une racine) et est équilibré (ses noeuds terminaux sont tous sur la même couche). Précisément un arbre $\mathcal{T}$ est un ensemble de noeuds $\{n_0, n_1, n_2, \ldots\}$, où n_0 est la racine. $\tilde{\mathcal{T}}$ est l'ensemble des noeuds terminaux ou **feuilles** de l'arbre. Un **sous-arbre** $\mathcal{S}$ de $\mathcal{T}$ est un arbre dont la racine $n \in \mathcal{T}$, ses feuilles $\tilde{\mathcal{S}}$ n'appartiennent pas forcément à $\tilde{\mathcal{T}}$ (un noeud unique n est aussi considéré comme un sous-arbre). Si de plus $\tilde{\mathcal{S}} \subset \tilde{\mathcal{T}}$, alors $\mathcal{S}$ est une **branche** de $\mathcal{T}$ (notée $\mathcal{S} = \mathcal{T}_n$). Enfin un sous-arbre élagué $\mathcal{S}$ est un sous-arbre planté en n_0 (noté $\mathcal{S} \preceq \mathcal{T}$).

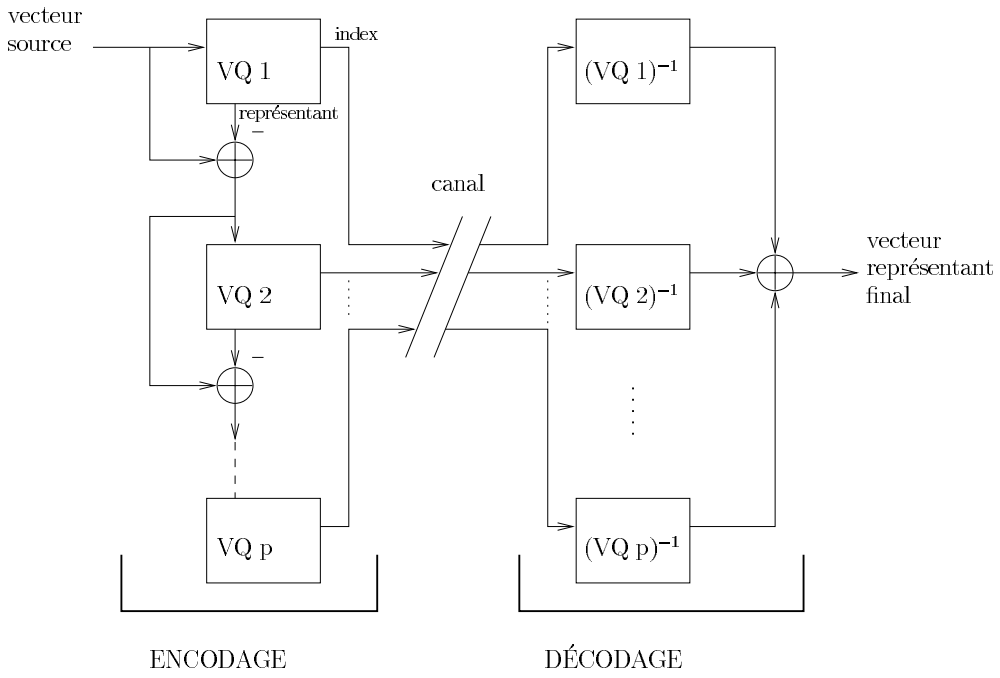


FIG. 4.1 – *Schéma d'un QV en cascade sur p étages.*

Pour la QVAr les représentants sont les L feuilles de l'arbre. Si ce dernier est B-aire et équilibré, sa **hauteur** (ou profondeur) est donnée par :

$$H = \log_B L$$

Le débit binaire (pour une code à longueur fixe) s'exprime [en bpp] :

$$R = \frac{1}{k}.H.\log_2 B$$

Le nombre de noeuds non terminaux est :

$$\sum_{i=0}^{H-1} B^i = \frac{1-B^H}{1-B}$$

Le nombre total de noeuds est donc :

$$\frac{1-B^H}{1-B} + B^H = \frac{1-B^{H+1}}{1-B}$$

Nous distinguons deux grandes méthodes de construction de dictionnaires arborescents :

- une **approche descendante** [Buzo et al.80] [Gersho et al.92] qui consiste à intégrer la construction de l'arbre à celle du dictionnaire. C'est la méthode la plus utilisée (*e.g.* un arbre binaire est réalisé en mémorisant les dictionnaires successifs obtenus avec l'algorithme LBG);

– une **approche ascendante** [Anderberg73] [Jain et al.88] [Benazza92] où l'arbre est construit une fois le dictionnaire obtenu. Ce dernier est réalisé à l'aide d'un algorithme qui *a priori* ne lui garantit aucune structure particulière (*e.g.* à l'aide de l'algorithme de Lloyd généralisé). Une classification hiérarchique ascendante est ensuite édifiée. Elle consiste à former à partir de petites classes homogènes, des classes qui le sont de moins en moins jusqu'à l'obtention d'une unique. Pour cette classification deux fonctions sont définies : une métrique mesurant la distance entre deux vecteurs et une fonction de discrimination évaluant la dissimilarité entre deux classes de vecteurs. Nous ne retiendrons pas l'approche ascendante car bien que plus coûteuse (elle regroupe deux phases distinctes), elle n'apporte pas de résultats supérieurs à ceux obtenus avec la méthode descendante.

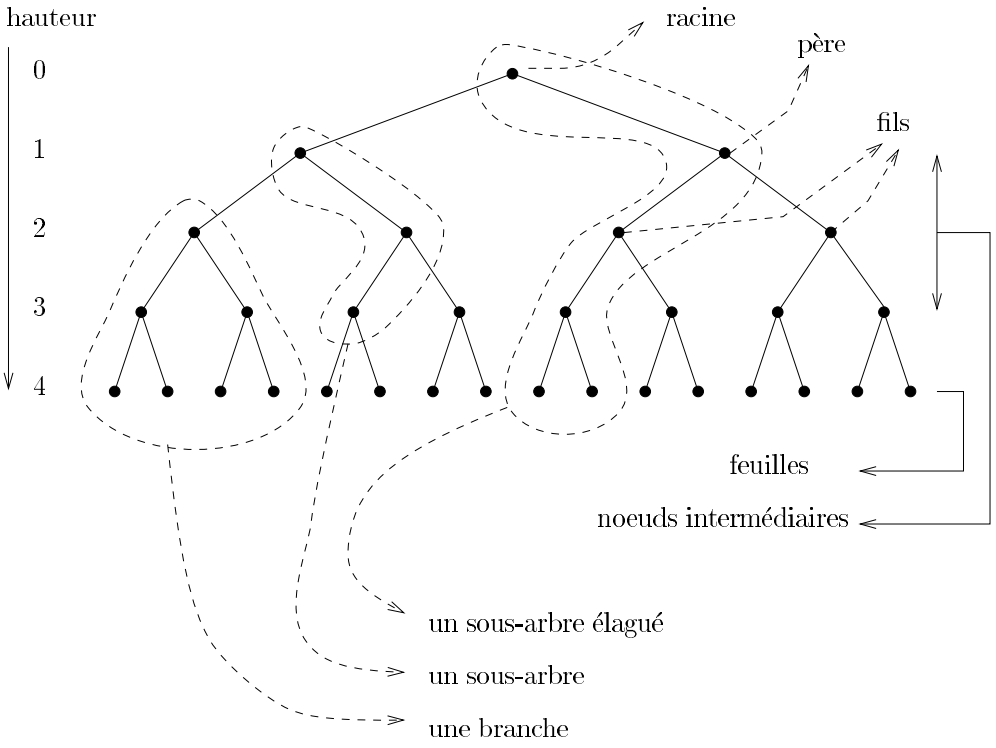


FIG. 4.2 – *Un arbre binaire équilibré.*

Principe du codage

Le codeur pour effectuer sa recherche dispose de l'arborescence et démarre sa recherche à partir de la racine de l'arbre. A chaque étape, le noeud parmi les fils du noeud courant apportant une distorsion minimale pour la reconstruction du vecteur source est sélectionné (dans la plupart des cas un calcul de normes est effectué, suivi du choix de celle minimale). La recherche se poursuit dans le sous-arbre ayant ce noeud comme racine et le processus

est itéré jusqu'à ce qu'une feuille soit atteinte. Le vecteur de reproduction associé au noeud terminal est alors le représentant du vecteur source. Si l'arbre est B-aire et équilibré, la **complexité** de l'algorithme de recherche est donnée par :

$$B.H = B.\log_B L$$

Elle est proportionnelle au logarithme de la taille du dictionnaire. La contrainte est qu'il faut stocker le dictionnaire et l'arbre.

Principe du décodage

Il convient de noter que le décodeur ne dispose généralement pas de l'ensemble des noeuds intermédiaires de l'arbre, mais seulement des feuilles puisque le codeur ne lui transmet que l'indice du représentant associé au vecteur source.
Cependant dans le cas particulier de la reconstruction progressive du signal (*i.e.* pour un codage hiérarchique), le décodeur utilise les noeuds intermédiaires [Wang et al.89] [Gersho et al.92] [Hwang et al.95]. Au lieu d'attendre que le vecteur source ait été complètement spécifié par le codeur après un parcours complet du dictionnaire, la transmission se fait à chaque niveau de l'arbre. Au fur et à mesure que le codeur progresse dans sa recherche, le représentant du vecteur source devient de plus en plus précis mais le coût augmente. Le décodeur interprète les index qu'il reçoit et effectue la reconstruction progressive.

Arbre non-équilibré

Un arbre est non-équilibré si les feuilles ne se situent pas toutes sur la même couche. Pour un code naturel, la longueur du mot de code associé à chaque feuille est proportionnelle à la hauteur de l'arbre. La structure arborescente non-équilibrée est particulièrement adaptée à la construction d'un QV à débit variable, où le codeur affecte plus de bits aux régions riche en information. Si le code construit est entropique, de façon générale les Voronoï associés aux représentants les plus hauts dans l'arbre sont de tailles plus réduites. Ceci détermine la probabilité d'occurence du vecteur de reproduction et donc la longueur du mot de code associé. La construction d'un dictionnaire arborescent non-équilibré permet donc là encore, d'affecter plus de bits aux régions contenant les vecteurs *a priori* plus pertinents.
Nous distinguons deux techniques principales pour la construction d'arborescences non-équilibrées :

- une d'élagage [Breiman et al.84] [Chou et al.89b] [Fiche et al.94] : un arbre équilibré est d'abord construit, cet arbre est ensuite élagué ;

- une de découpage [Makhoul et al.85] [Riskin et al.91] : lors de la construction du dictionnaire par une approche descendante, une application non systématique du découpage des noeuds conduit à concevoir un arbre non-équilibré.

Pour chacune de ces approches, un critère est utilisé pour déterminer la branche à élaguer ou la feuille à découper. Nous donnons quelques exemples :

- Makhoul [Makhoul et al.85] propose de découper le noeud contribuant le plus à la distorsion. Ce critère simple ne tient pas compte du coût de codage et ne fournit pas nécessairement la plus grande baisse de l'EQM ;
- Riskin et Gray [Riskin et al.91] adaptent au codage le critère du retour marginal déjà introduit par Breiman [Breiman et al.84] dans le contexte de la classification de données. Cette fonction de coût permet de découper le noeud offrant le meilleur compromis débit *vs.* distorsion ;
- Zeng [Zeng et al.95] découpe le noeud dont la valeur propre principale de la matrice de covariance relative aux vecteurs source dans le Voronoï, est la plus grande. Le but est de tenir compte de la répartition des points à coder autour du représentant. Une telle approche se justifie si les formes des Voronoï sont variées.

Le critère de découpage que nous adoptons, dans ce contexte de codage de séquences d'images par la QVAA, est le retour marginal. Afin de justifier ce choix, nous proposons de détailler les méthodes de construction de dictionnaires arborescents non-équilibrés que nous mettons en oeuvre.

4.3 Construction d'un dictionnaire arborescent non-équilibré

Cette partie présente les deux principales approches pour la construction d'un dictionnaire arborescent non-équilibré à débit variable. Notre but est de décrire des algorithmes s'appliquant ensuite directement au cas de la QVAA. Dans la littérature, seules les expressions analytiques relatives à des arbres binaires sont complètement développées. Il nous a semblé intéressant d'expliciter les formules pour le cas d'arbres B-aires ($B \geq 2$) que nous rencontrons avec la QVAA.

Par simplification nous utilisons souvent les termes "distorsion" et "débit" au lieu de "distorsion moyenne" et "longueur moyenne des mots du code binaire entropique". C'est en effet le débit entropique associé aux représentants que nous considérons".

Le dictionnaire est construit à l'aide d'une technique d'apprentissage. Il est donc possible de caractériser numériquement en termes de codage débit-distorsion, chacun des noeuds n_i de l'arbre $\mathcal{T} = \{n_0, n_1, n_2, \ldots\}$. Nous définissons :

- $\mathbf{y}_{n_i}$: le représentant associé à n_i ;
- C_{n_i} : le voronoï associé à n_i ;
- $SA = \{\mathbf{x}_i \ / \ i = 0, 1, 2, \ldots, N\}$: la séquence d'apprentissage ;
- $N = \text{card}(SA)$: la taille de la séquence d'apprentissage (*i.e.* son nombre de vecteurs) ;

– $\text{card}(C_{n_i})$: le nombre de vecteurs de la séquence d'apprentissage appartenant à C_{n_i}.

Lors de l'encodage de la séquence d'apprentissage, à chaque noeud n_i de $\mathcal{T}$ est associé :

– une probabilité d'occurence :

$$P(n_i) = \frac{\text{card}(C_{n_i})}{N}$$

– une distorsion moyenne :

$$d(n_i) = \frac{1}{\text{card}(C_{n_i})} . \sum_{\mathbf{x} \in C_{n_i} \ , \ \mathbf{x} \in SA} ||\mathbf{x} - \mathbf{y}_{n_i}||^2$$

– une longueur de mot du code binaire entropique [bits/vecteur] :

$$l(n_i) = -\log_2 P(n_i)$$

Une fois l'arbre construit un codage entropique des feuilles est effectué. Un sous-arbre $\mathcal{S}$ de $\mathcal{T}$ est alors caractérisé par :

– la distorsion moyenne associée à ses feuilles :

$$d(\mathcal{S}) = \sum_{n_u \in \tilde{\mathcal{S}}} P(n_u).d(n_u)$$

– la longueur moyenne des mots du code binaire entropique associée à ses feuilles :

$$l(\mathcal{S}) = \sum_{n_u \in \tilde{\mathcal{S}}} P(n_u).l(n_u)$$

L'arbre entier est lui défini par $d(\mathcal{T}) = D$ et $l(\mathcal{T}) = R$.
Les fonctionnelles $d(\mathcal{S})$ et $l(\mathcal{S})$ sont dites monotones (*e.g.* dans le cas de la construction de l'arbre par une approche descendante, elles sont respectivement croissantes et décroissantes). Elles sont aussi dites linéaires car leur résultat est la somme de valeurs aux feuilles [Breiman et al.84] [Gersho et al.92].
Afin de construire le dictionnaire arborescent non-équilibré nous adoptons donc un critère débit-distorsion pour déterminer les noeuds à découper. Pour simplifier il s'agit de faire un compromis tel que les régions de l'espace où la distorsion est élevée soient découpées, tout en évitant que le coût en terme de débit soit trop important. Pour construire ainsi le dictionnaire, deux stratégies classiques sont adaptées : une d'élagage de l'arbre, l'autre de découpage.

4.3.1 Elagage de l'arbre

Nous présentons l'algorithme de BFOS généralisé (**BFOS**) [Breiman et al.84] [Chou et al.89a] [Gersho et al.92]) où un arbre (en principe équilibré) de grande taille doit d'abord être construit. Une fois cet arbre complet achevé, un processus itératif d'élagage est mis en place : à chaque boucle, suivant le critère débit-distorsion une branche est supprimée. Le stockage en mémoire de l'arbre initial implique que ce dernier ne peut néanmoins être trop grand, une condition s'impose pour contrôler sa taille : que le nombre d'aires de l'arbre soit réduit.

Si $\mathcal{S}_{n_i}$ est une branche de l'arbre complet $\mathcal{T}$, nous déterminons :

- la distorsion associée aux feuilles de $\mathcal{S}_{n_i}$:

$$d(\mathcal{S}_{n_i}) = \sum_{n_u \in \tilde{\mathcal{S}}_{n_i}} P(n_u).d(n_u)$$

- le débit associé aux feuilles de $\mathcal{S}_{n_i}$:

$$l(\mathcal{S}_{n_i}) = \sum_{n_u \in \tilde{\mathcal{S}}_{n_i}} P(n_u).l(n_u)$$

- la hausse de la distorsion si $\mathcal{S}_{n_i}$ est supprimée (hausse définie positive) :

$$\Delta d(\mathcal{S}_{n_i}) = P(n_i).d(n_i) - d(\mathcal{S}_{n_i})$$

- la baisse du débit si $\mathcal{S}_{n_i}$ est supprimée (baisse définie positive) :

$$\Delta l(\mathcal{S}_{n_i}) = l(\mathcal{S}_{n_i}) - P(n_i).l(n_i)$$

Par définition, le **retour marginal** associé à la racine n_i de $\mathcal{S}_{n_i}$ devient :

$$\lambda(n_i) = \frac{\Delta d(\mathcal{S}_{n_i})}{\Delta l(\mathcal{S}_{n_i})}$$

Nous allons à chaque boucle de l'algorithme, supprimer la branche dont la racine indique un **retour marginal minimal** (*i.e.* celle dont la suppression entraîne une hausse de la distorsion minimale et une baisse du débit maximale). L'interprétation à l'aide de la courbe débit-distorsion est simple car la paire (R, D) d'un arbre donné correspond à un point de cette fonction (voir la figure 4.3). A l'arbre complet correspond le point de $D(R)$ le plus bas à droite. A chaque itération un choix doit-être fait parmi les branches à élaguer. A chacun de ces choix possibles correspond un sous-arbre élagué et un nouveau point débit-distorsion. Le point de $D(R)$ désigné par le critère du retour marginal minimal est celui offrant le meilleur compromis débit-distorsion : la courbe tracée circule alors suivant l'enveloppe convexe de tous les points possibles. Chaque $\lambda(n_i)$ peut s'interpréter comme

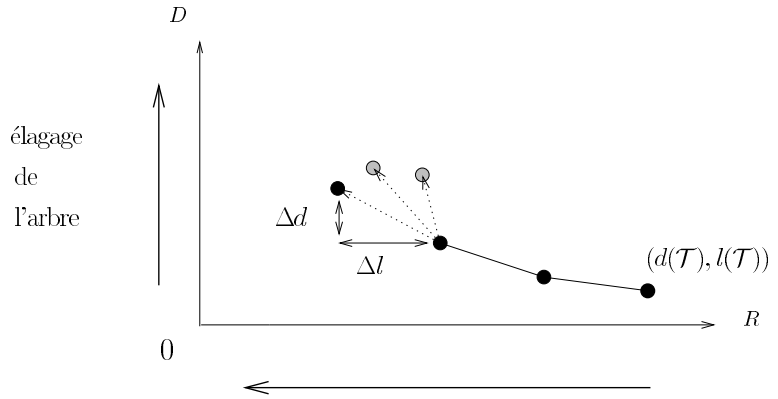


FIG. 4.3 – *Elagage de l'arbre : à une itération donnée de l'algorithme de BFOS, le point choisi est celui dont le retour marginal est minimal.*

une portion possible de la pente de la courbe débit-distorsion.

Cette approche est aussi décrite comme une technique d'optimisation lagrangienne où la fonctionnelle $J(\mathcal{S}) = d(\mathcal{S}) + \lambda.l(\mathcal{S})$ est minimisée (λ est le multiplicateur de Lagrange). En faisant varier λ, tous les points de la courbe débit-distorsion correspondant aux différents sous-arbres élagués possibles $\mathcal{S}$, peuvent-être testés et le plus intéressant est choisi. Remarquons qu'une approche réalisant une compression et une classification conjointes d'images a été avancée [Perlmutter et al.96], elle procède en incorporant au coût lagrangien deux facteurs : un mesurant la distorsion (*i.e.* une métrique), l'autre le risque d'erreur de classification (*i.e.* un risque bayésien).

Après avoir construit l'arbre complet $\mathcal{T}$ et déterminé pour ses noeuds n_i : $P(n_i)$, $d(n_i)$ et $l(n_i)$. Il faut, avant le lancement du processus d'élagage, calculer les retours marginaux associés à chacun des n_i. Une formule de récurrence se met alors en place, en progressant de proche en proche (*i.e.* à partir des feuilles jusqu'à la racine, et des fils vers le père), elle permet de calculer les valeurs de $\Delta d(\mathcal{S}_{n_i})$ et $\Delta l(\mathcal{S}_{n_i})$:

- pour les feuilles :

$$\begin{aligned} \Delta d(\mathcal{S}_{n_i}) &= \Delta l(\mathcal{S}_{n_i}) = 0 \\ \lambda(n_i) &= +\infty \end{aligned}$$

- pour les autres noeuds, si n_i a B fils n_j (la démonstration est à l'annexe C) :

$$\begin{aligned} \Delta d(\mathcal{S}_{n_i}) &= P(n_i).d(n_i) + \sum_B \Delta d(\mathcal{S}_{n_j}) - \sum_B P(n_j).d(n_j) \\ \Delta l(\mathcal{S}_{n_i}) &= \sum_B \Delta l(\mathcal{S}_{n_j}) + \sum_B P(n_j).l(n_j) - P(n_i).l(n_i) \\ \lambda(n_i) &= \frac{\Delta d(\mathcal{S}_{n_i})}{\Delta l(\mathcal{S}_{n_i})} \end{aligned}$$

La distorsion et le débit associés à l'arbre $\mathcal{T}$ sont respectivement :

$$d(\mathcal{T}) = \sum_{n_u \in \tilde{\mathcal{T}}} P(n_u).d(n_u) = P(n_0).d(n_0) - \Delta d(\mathcal{S}_{n_0}) = d(n_0) - \Delta d(\mathcal{S}_{n_0})$$

$$l(\mathcal{T}) = \sum_{n_u \in \tilde{\mathcal{T}}} P(n_u).l(n_u) = P(n_0).l(n_0) + \Delta l(\mathcal{S}_{n_0}) = \Delta l(\mathcal{S}_{n_0})$$

Après chaque élagage, il faut remettre à jour les retours marginaux des ascendants (*i.e.* les pères) de la racine de la branche supprimée. Si nous définissons :

- le sous-arbre élagué produit à l'issue de la boucle j de l'algorithme de BFOS :

$$\mathcal{S}^j \ / \ j = 1,\ 2,\ \ldots \ (\mathcal{S}^0 = \mathcal{T})$$

- la branche plantée en n_i, élaguée lors de cette boucle j :

$$\mathcal{S}^j_{n_i} \ / \ j = 1,\ 2,\ \ldots$$

- les ascendants (jusqu'à la racine n_0) de n_i : les n_k

Alors une fois $\mathcal{S}^j_{n_i}$ élaguée, nous remettons à jour :

- les retours marginaux des n_k en appliquant (la démonstration est à l'annexe C) :

$$\begin{aligned} \Delta d(\mathcal{S}^{j+1}_{n_k}) &= \Delta d(\mathcal{S}^j_{n_k}) - \Delta d(\mathcal{S}^j_{n_i}) \\ \Delta l(\mathcal{S}^{j+1}_{n_k}) &= \Delta l(\mathcal{S}^j_{n_k}) - \Delta l(\mathcal{S}^j_{n_i}) \\ \lambda(n_k) &= \frac{\Delta d(\mathcal{S}^{j+1}_{n_k})}{\Delta l(\mathcal{S}^{j+1}_{n_k})} \end{aligned}$$

- le retour marginal de la nouvelle feuille n_i :

$$\begin{aligned} \Delta d(\mathcal{S}^{j+1}_{n_i}) &= \Delta l(\mathcal{S}^{j+1}_{n_i}) = 0 \\ \lambda(n_i) &= +\infty \end{aligned}$$

La distorsion et le débit du nouveau sous-arbre élagué sont donc (voir la figure 4.4) :

$$d(\mathcal{S}^{j+1}) = d(\mathcal{S}^j) + \Delta d(\mathcal{S}^j_{n_i})$$

$$l(\mathcal{S}^{j+1}) = l(\mathcal{S}^j) - \Delta l(\mathcal{S}^j_{n_i})$$

Ce résultat traduisant la hausse de la distorsion et la baisse du débit à chaque élagage permet d'adapter un critère de fin de construction du dictionnaire : il faut continuer à élaguer tant que la distorsion est jugée acceptable (*i.e.* continuer à élaguer tant que $d(\mathcal{S}^{j+1}) \leq D_{seuil}$), ou tant que le débit demeure supérieur au débit désiré (*i.e.* continuer à élaguer tant que $l(\mathcal{S}^{j+1}) \geq R_{seuil}$). Les vecteurs associés aux feuilles du sous-arbre élagué obtenu, sont les représentants du dictionnaire final.
La figure 4.5 présente un synoptique simplifié de l'algorithme d'élagage (BFOS).

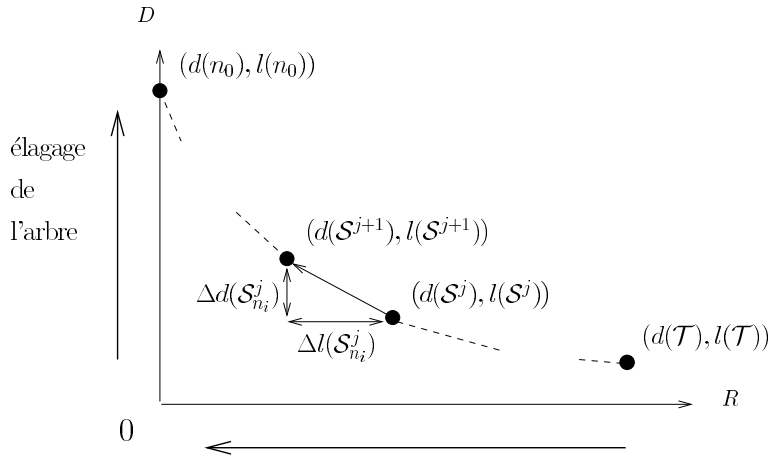


FIG. 4.4 – *Elagage de l'arbre : cas où à l'itération j de l'algorithme, la branche $\mathcal{S}^j_{n_i}$ est élaguée.*

4.3.2 Découpage de l'arbre

Avec cet algorithme [Riskin et al.91] [Gersho et al.92] le dictionnaire arborescent non-équilibré est construit de façon progressive par un processus itératif où, à l'issue de chaque boucle, une seule feuille de l'arbre demeure découpée. A l'initialisation l'arbre n'est (en principe) constitué que de la racine n_0. Puis à chaque boucle de l'algorithme :

- toutes les feuilles de l'arbre sont découpées ;
- seule la branche ainsi créée, présentant le meilleur compromis débit-distorsion est conservée (toutes les autres sont élaguées).

Cette technique est adaptée à la construction du dictionnaire lorsque le nombre d'aires de l'arbre est élevé, car la quantité d'information à stocker est limitée et exactement ajustée à la taille du dictionnaire qui croît progressivement.
Pour développer les calculs nous reprenons les notations déjà introduites. En particulier nous considérons obtenir à l'issue de la boucle j du processus, un sous-arbre élagué $\mathcal{S}^j$ de l'arbre complet final $\mathcal{S}^l = \mathcal{T}$. Ce dernier sera obtenu à l'issue de l itérations ($l \geq j$), tel que le critère de fin de construction du dictionnaire soit atteint.
A l'initialisation nous avons $\mathcal{S}^0 = n_0$. Pour la boucle $j (j \geq 1)$ du processus de découpage, nous considérons :

- les feuilles du sous-arbre élagué $\mathcal{S}^{j-1}$: les n_i ;
- les nouvelles branches obtenues par le découpage des n_i (sachant qu'une seule de ces branches sera maintenue pour $\mathcal{S}^j$) : les $\mathcal{S}^j_{n_i}$;
- les retours marginaux aux n_i déterminés en calculant :

$$\Delta d(\mathcal{S}_{n_i}) \quad = \quad P(n_i).d(n_i) - d(\mathcal{S}_{n_i}),$$

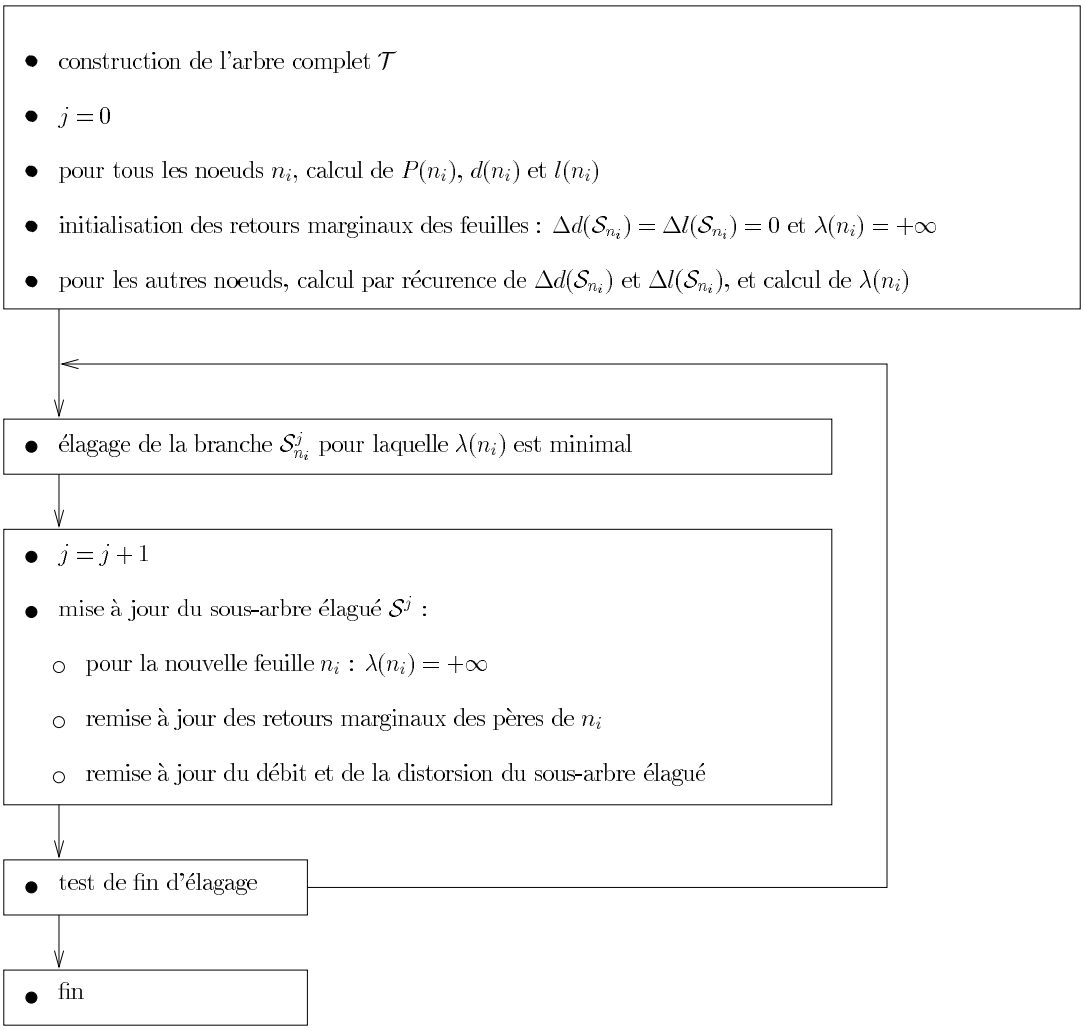


FIG. 4.5 – *Synoptique simplifié de l'algorithme d'élagage (BFOS).*

$$\begin{aligned}\Delta l(\mathcal{S}_{n_i}) &= l(\mathcal{S}_{n_i}) - P(n_i).l(n_i),\\ \lambda(n_i) &= \frac{\Delta d(\mathcal{S}_{n_i})}{\Delta l(\mathcal{S}_{n_i})}.\end{aligned}$$

Nous conservons comme nouvelle branche pour $\mathcal{S}^j$ celle offrant le meilleur compromis débit-distorsion, soit celle dont l'ajout offre une baisse de la distorsion maximale pour une hausse du débit minimale. Il s'agit donc de la branche $\mathcal{S}^j_{n_i}$ dont le **retour marginal** $\lambda(n_i)$ est **maximal.** Remarquons qu'il n'y a pas à calculer tous les retours marginaux aux n_i à chaque itération de l'algorithme, mais uniquement ceux correspondant aux dernières feuilles ajoutées (les autres ont été calculés lors des boucles précédentes et mémorisés).
L'interprétation à l'aide de la courbe débit-distorsion nous est devenue familière (voir la figure 4.6) : à l'initialisation, le sous-arbre élagué $\mathcal{S}^0$ correspond au point de $D(R)$ en haut à gauche (*i.e.* $R = 0$ et D est égale à la variance de la séquence d'apprentissage). A chaque nouvelle itération j de l'algorithme, un choix doit être fait parmi les $\mathcal{S}^j_{n_i}$ de la branche à ne pas élaguer, et à chacun de ces choix possibles correspond un nouveau point de $D(R)$. Au fur et à mesure que l'arbre croît, le point de la courbe $D(R)$ se positionne de plus en plus bas et sur la droite. Le critère du retour marginal maximal permet de désigner le point offrant localement le meilleur compromis baisse de la distorsion *vs.* hausse du débit.

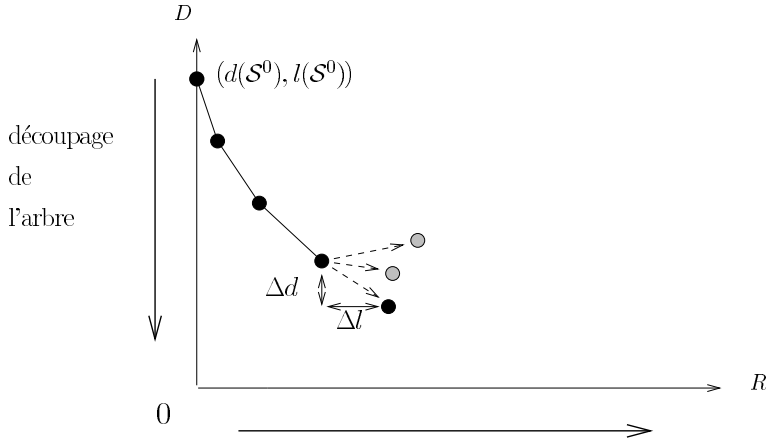


FIG. 4.6 – *Découpage de l'arbre : à une itération donnée de l'algorithme, le point choisi est celui dont le retour marginal est maximal.*

La distorsion et le débit associés au nouveau sous-arbre élagué S^j sont donnés par (la démonstration est à l'annexe D, voir aussi la figure 4.7) :

$$d(\mathcal{S}^j) = d(\mathcal{S}^{j-1}) - \Delta d(\mathcal{S}^j_{n_i})$$

$$l(\mathcal{S}^j) = l(\mathcal{S}^{j-1}) + \Delta l(\mathcal{S}^j_{n_i})$$

Ces paramètres sont les seuls à devoir être remis à jour à chaque itération du processus de découpage, ils traduisent la baisse de la distorsion et la hausse du débit réalisées. L'arbre

croissant au fur et à mesure des découpages, l'arrêt de la construction du dictionnaire est décidé si un débit limite est atteint (*i.e.* arrêt si $R \geq R_{seuil}$) ou si la qualité de la reconstruction de la source est jugée suffisante (*i.e.* arrêt si $D \leq D_{seuil}$).
La figure 4.8 présente un synoptique simplifié de cet algorithme de découpage de l'arbre.

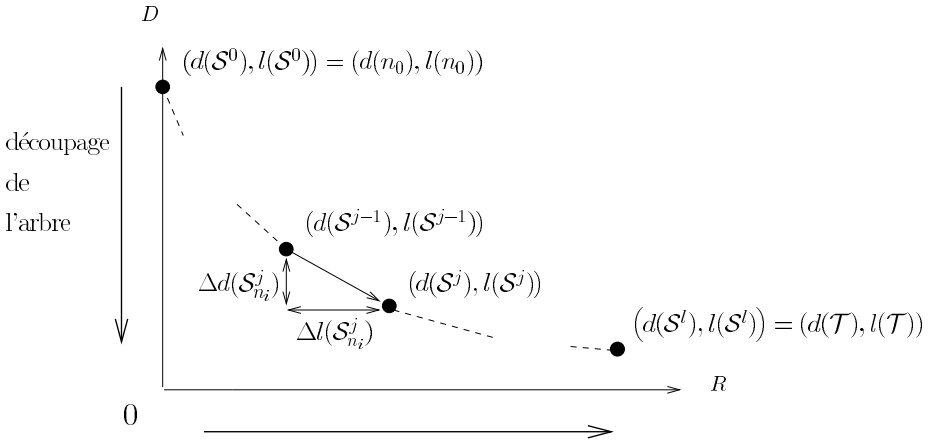


FIG. 4.7 – *Découpage de l'arbre : cas où à l'itération j de l'algorithme, la branche $\mathcal{S}^j_{n_i}$ est ajoutée.*

4.4 Conclusion

De façon générale, l'algorithme de BFOS permet de réaliser un élagage optimal de l'arbre car l'approche est globale (*i.e.* à long terme) et telle que l'ensemble des points débit-distorsion obtenus appartienne à l'enveloppe convexe de $D(R)$. Cependant il faut noter trois limites à cette approche :

- le résultat final est tributaire de la découpe de l'espace réalisée par l'arbre initial (*e.g.* plus l'arbre complet est grand, meilleur sera le résultat) ;
- les points sur la courbe convexe sont éparses, il n'est dons pas toujours possible d'obtenir un débit donné (ou du moins en être très proche). Des méthodes pour obtenir des points intermédiaires ont été proposées, elles visent :
  - soit à générer un arbre s'interposant entre celui avant et après élagage, cependant des artéfacts apparaissent sur les images [Kiang et al.92] ;
  - soit à obtenir des points intermédiaires juste au dessus de la courbe convexe. A titre d'exemples, nous indiquons que Kiang [Kiang et al.92] procède en supprimant uniquement des noeuds n'ayant pas de descendants, ou que Rose [Rose et al.96] modifie la partition finale relative aux feuilles de l'arbre.
    Dans le cadre de l'allocation binaire relative au codage en sous-bandes avec la QVAA (voir le chapitre 5), ce problème de points éparses sur la courbe $D(R)$

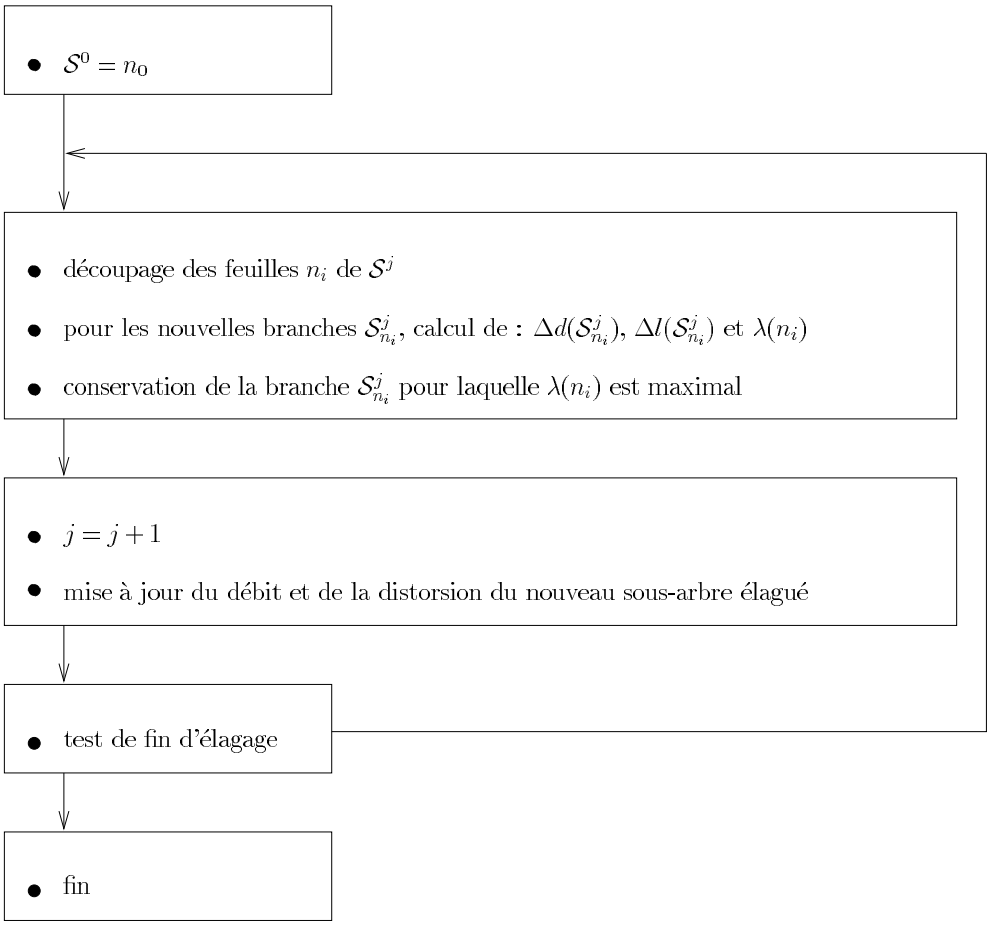


FIG. 4.8 – *Synoptique simplifié de l'algorithme de découpage de l'arbre.*

se pose. Nous développons alors une approche visant aussi à obtenir des points au-dessus de la courbe convexe ;

– la règle d'encodage du plus proche voisin (voir l'équation 2.2) est sous-optimale. En effet celle-ci ne convient que si la QV est sans contrainte. S'il y a contrainte entropique, la règle optimale devient [Chou et al.89a] (les notations sont celles du chapitre 2) :

$$\mathbf{x} \in C_i \quad \text{ssi} \quad d(\mathbf{x}, \mathbf{y}_i) + \lambda.(-\log_2 p_i) \leq d(\mathbf{x}, \mathbf{y}_j) + \lambda.(-\log_2 p_j) \;\; / \;\; \forall j \neq i \qquad (4.1)$$

où $\mathbf{x} \in \mathbb{R}^k$ est le vecteur source, et λ le multiplicateur de Lagrange correspondant au débit choisi. L'équation 4.1 implique que la partition et donc les probabilités des représentants soient fonction du débit. Les algorithmes optimaux de QV contrainte en entropie [Chou et al.89a] [Kossentini et al.93] sont coûteux car ils doivent alors procéder itérativement afin d'ajuster la partition de l'espace et le débit résultant. Pour le BFOS, l'ensemble des partitions possibles est figée dès la construction de l'arbre complet, et la règle d'encodage optimale n'est pas mise en oeuvre (il faudrait un *a priori* sur λ). Rose [Rose et al.96] propose une solution intermédiaire consistant à appliquer un algorithme contraint en entropie sur les feuilles de l'arbre final.

L'approche par découpage est locale, les résultats sont donc sous optimaux comparés à ceux obtenus avec l'algorithme de BFOS [Chou et al.89b]. Balakrishnan [Balakrishnan et al.95] réunit cependant les conditions théoriques telles que la méthode par découpage soit suffisante : il faut que les retours marginaux successifs décroissent. Il décrit également une approche tenant compte de la contrainte entropique : pour chaque nouveau découpage effectué, la règle d'encodage de l'équation 4.1 est mise en oeuvre via une procédure itérative (où à l'initialisation λ est nul, puis successivement mis à jour).

Avec la QVAA, l'arbre est réalisé par emboîtage successifs de réseaux tronqués. Le nombre de points de ce dernier détermine le nombre d'aires de l'arbre (voir le chapitre 5), même limité ce nombre demeure relativement élevé. C'est pourquoi la construction de l'arbre non-équilibré par découpage sera privilégiée. L'introduction des calculs des retours marginaux permet de faire un compromis débit-distorsion, la prise en compte complète de la contrainte entropique n'est pas envisageable car elle conduit à des algorithmes trop coûteux pour notre application.

Chapitre 5

Quantification vectorielle algébrique et arborescente

5.1 Introduction

Nous introduisons dans ce chapitre la quantification vectorielle algébrique et arborescente (**QVAA**). Le cadre d'application choisi est celui de la compression de séquences d'images, et précisément le codage après transformation de l'erreur de prédiction de compensation du mouvement (voir le chapitre 1). Afin de déterminer les caractéristiques de ce nouveau quantificateur, la première étape est d'analyser les spécificités de la source vectorielle. Il faut aussi tenir compte des contraintes contextuelles des temps de calcul (pour la construction du dictionnaire, et l'encodage des vecteurs), et de dissymétrisation du codage (*e.g.* il est autorisé que le codeur soit plus complexe que le décodeur). La QVAA est alors développée afin de tirer profit des deux techniques de quantification précédemment décrites, en conjuguant :

- la quantification rapide à l'aide des RRP, et l'avantage de disposer de dictionnaires prédéfinis ;
- la construction par apprentissage d'un dictionnaire arborescent autorisant une partition de l'espace adaptée à la distribution de la source et à un critère débit-distorsion.

L'objectif de ce chapitre est de présenter les différentes étapes de la conception du QVAA. La description est faite suivant l'ordre selon lequel les problèmes se sont posés, nous justifions ainsi au fur et à mesure nos choix.

5.2 Contexte et spécification de la source vectorielle

Le QVAA sera testé en prenant place au sein d'une chaîne de codage hybride pour la compression à bas débits de séquences d'images animées. Le schéma typique de codage que nous exploitons est illustré à la figure 5.1, les techniques de décorrélation sont présentées

au chapitre 1, nous retrouvons:

- la technique de décorrélation inter-image qui est une estimation-compensation du mouvement en avant. Les erreurs de prédiction sont essentiellement contituées:

  . des bords des objets en mouvement,

  . des zones fortement texturées de ces objets,

  . des zones découvertes et de celles recouvertes par ces objets.

  La distribution statistique des images d'erreurs est devenue unimodale et est communément modélisée par une loi laplacienne [Gaidon93];

- la technique de décorrélation intra-image qui consiste en une transformée du signal résiduel avec configuration en sous-bandes. Précisément nous retenons le cas de coefficients DCT agencés en intra-bande, et de coefficients d'ondelette issus d'une découpe dyadique.
  Nous avons au chapitre 2 justifié cette seconde étape de décorrélation conduisant à une mise en forme spectrale du bruit de quantification et à une classification des données. Un dictionnaire est alors construit par sous-bande.

La modélisation monodimensionnelle de la distribution statistique d'une sous-image est généralement faite à l'aide d'une fonction **gaussienne généralisée** telle que:

$$p_X(x) = a.\exp\left(-|b.x|^{\alpha}\right) \text{ avec } \begin{cases} a &= \frac{b.\alpha}{2.\Gamma(1/\alpha)} \\ b &= \frac{1}{\sigma}.\sqrt{\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}} \end{cases}$$

où σ est l'écart type de la fonction de densité de probabilité à modéliser, et Γ la fonction Gamma définie par:

$$\Gamma(n) = \int_0^{+\infty} e^{-x}.x^{n-1}\,dx$$

Si $\alpha = 2$ nous retrouvons la loi normale, si $\alpha = 1$ c'est celle laplacienne, mais dans notre cas $\alpha < 1$ [Antonini91]. L'étroitesse de cette fonction de densité souligne la forte corrélation qui pourra *a priori* être exploitée par la QV. Mais c'est la distribution multidimensionnelle de la sous-image qui doit-être aussi prise en considération.

Les sous-bandes ont la particularité de préserver la localisation spatiale de l'image résiduelle. Les contours dans des directions privilégiées sont extraits (*e.g.* les directions horizontales, verticales et diagonales) et à différentes échelles si l'analyse est multirésolution. Les sous-images transformées sont ainsi interprétables car, si des pixels voisins de l'image originale sont corrélés, leurs coefficients respectifs ont des amplitudes proches et de même signe. La répartition spatiale des vecteurs corrélés se fait au sein d'une ellipse orientée dont un axe est porté par le vecteur $(1, 1, \ldots, 1)^T$ (par rapport au système de coordonnées cartésiennes) [Fisher89] [Moureaux94].
Dans le cadre du codage prédictif (*e.g.* le signal à transformer est généré par la différence

entre deux images dont l'une compense l'autre), les contours dans l'image différentielle sont caractérisés par des valeurs de pixels de même amplitude mais de signe opposé (voir la figure 5.2). Les coefficients transformés correspondant ont aussi des amplitudes proches et des signes différents. Pour ces sources modélisables par des gaussiennes généralisées, la répartition des vecteurs dans l'espace se fait au sein d'ellipses orientées suivant les axes bissecteurs. La forme des vecteurs prélevés dans la sous-image doit être choisie de façon à exploiter pleinement ces dépendances entre coefficients transformés.

La chaîne hybride de décorrélation conduit à une mise en forme du signal à quantifier, mais ce dernier demeure non-stationnaire [Jayant et al.84] [Antonini et al.92]. C'est pourquoi nous retenons une procédure d'apprentissage afin de construire le dictionnaire à partir d'un ensemble de vecteurs représentatifs de la sous-bande à coder. Nous voulons aussi obtenir une construction rapide de ce dictionnaire de telle sorte que, dès que ce dernier ne sera plus valide, une actualisation des représentants puisse être effectuée à partir d'une séquence d'apprentissage représentative de la statistique courante de la source. Ce QV s'intègre ainsi dans la famille des quantificateurs **adaptatifs** [Goldberg et al.86] [Monet et al.90] (voir la figure 5.3).

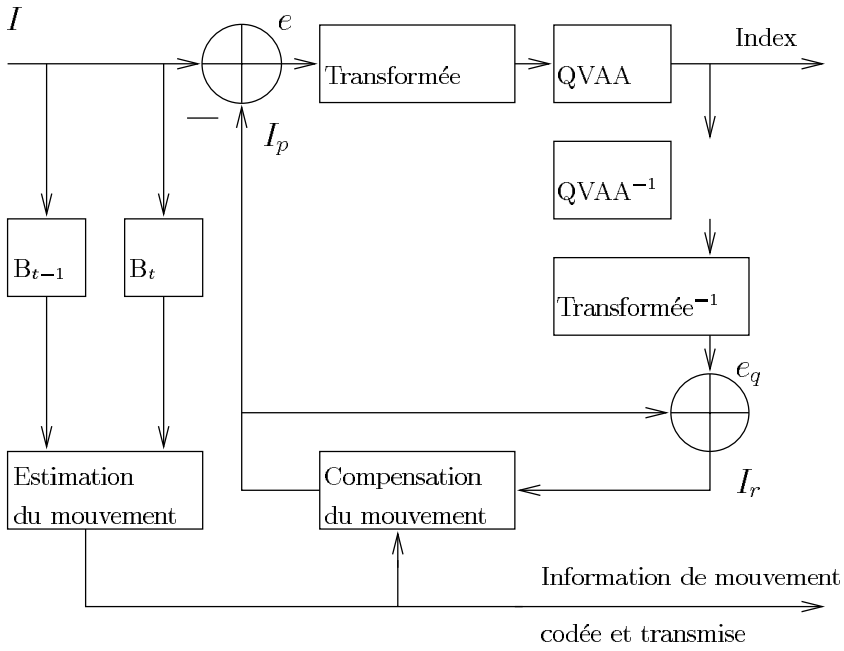


FIG. 5.1 – *Codeur hybride avec compensation du mouvement en avant.*
A l'instant t : I est une image de la séquence, I_p celle prédite et I_r celle reconstruite ; e est l'image d'erreur de prédiction et e_q celle quantifiée ; les B indiquent des mémoires tampon.

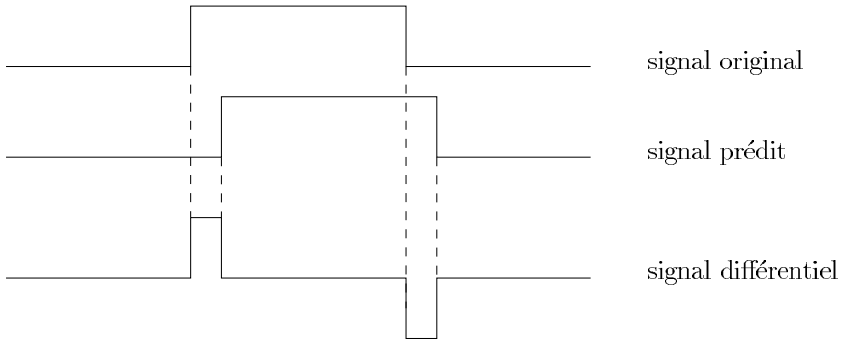


FIG. 5.2 – *Exemple simple montrant que les contours du signal différentiel sont constitués d'éléments de même amplitude mais de signes opposés.*

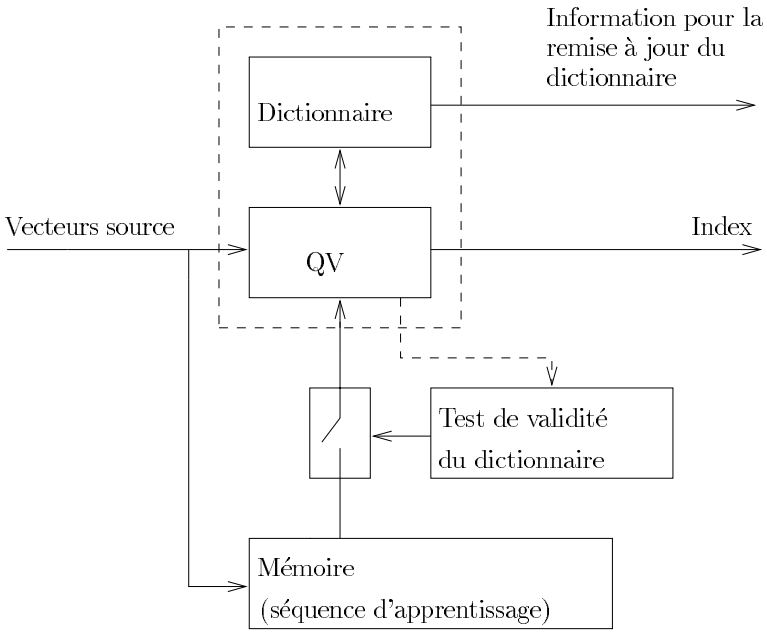


FIG. 5.3 – *Schéma d'un QV adaptatif.*
Le test de validité du dictionnaire peut-être la comparaison d'une EQM à un seuil.

5.3 Mise en oeuvre

5.3.1 Préambule

Pour concevoir ce nouveau QV nous n'avons pas retenu une technique d'apprentissage du type LBG arborescent car l'encodage et surtout la construction du dictionnaire (et son éventuel élagage ou découpage), bien que accélérés, demeurent trop complexes. La QVA qui offre une quantification rapide, n'est adaptée que si la source est stationnaire et telle que sa statistique autorise une troncature aisée du réseau. Ce n'est pas le cas avec le signal des sous-images. Nous proposons alors d'adapter la QVA mais en procédant par emboîtage d'une hiérarchie de RRP tronqués. Les étapes, que nous décrivons, de la construction du QVAA sont [Ricordel et al.95b] [Ricordel et al.95c] [Ricordel et al.95a] :

- le choix du RRP relativement à la dimension vectorielle fixée,
- la troncature du réseau afin qu'il s'emboîte,
- la formation d'une hiérarchie de RRP tronqués s'emboîtant de proche en proche,
- la mise en place d'un schéma de QV multi-étages,
- le découpage ou l'élagage du dictionnaire arborescent résultant.

Nous disposons à ce niveau d'un premier schéma de quantification. Nous l'achevons en précisant le réseau optimal dans ce contexte d'emboîtage, et le traitement particulier des vecteurs hors norme [Ricordel et al.96a] [Ricordel et al.96b]. Pour finir nous abordons le problème lié à l'allocation binaire entre les sous-bandes lorsque le quantificateur est un QVAA [Montrichard et al.96a].

5.3.2 Hiérarchie de réseaux réguliers emboîtés

Pour la dimension vectorielle fixée *a priori*, le RRP mis en oeuvre est choisi parmi $\mathbb{Z}^k$, D_k, E_8 et Λ_{16}. Ce sont les réseaux pour lesquels Conway et Sloane ont déterminé des algorithmes de quantification rapide [Conway et al.82a]. Nous rappelons que D_4, E_8 et Λ_{16} sont aussi les réseaux meilleurs quantificateurs relativement à leur dimension (voir le chapitre 2).
Le réseau adopté (dit réseau "support") est tronqué tel qu'il puisse être emboîté, après un changement d'échelle, dans un de ses Voronoï. Nous précisons que la **résolution** du RRP, égale au nombre de points par unité de volume, est unitaire pour le réseau support ; et que l'**échelle** se définit comme l'inverse de la résolution. L'**emboîtage** est alors l'opération consistant à inclure dans un Voronoï "récepteur" d'un RRP à une résolution donnée, un même réseau tronqué de résolution supérieure. Nous dirons que ce dernier est emboîté dans le Voronoï récepteur. L'emboîtement est l'état de ce qui est emboîté. Nous voulons évidemment que l'**emboîtement** soit **optimal**, tel que les Voronoï du RRP emboîté s'ajustent exactement avec le Voronoï récepteur, soit encore que le volume de ce dernier ne soit rempli que de Voronoï entiers du réseau emboîté. Cette condition est cependant tributaire des propriétés géométriques intrinsèques du RRP support. Notre objectif

devient alors la description d'une méthode générale garantissant, quelque soit le réseau régulier, un **emboîtement sous-optimal** tel que le volume du Voronoï récepteur soit rempli d'un maximum de Voronoï entiers du RRP emboîté.
L'emboîtage consiste précisément à ajuster l'échelle du RRP à emboîter par rapport à celle du RRP auquel appartient le Voronoï récepteur. La figure 5.4 illustre le principe du changement d'échelle d'un réseau (si pour plus de commodités les schémas de ce chapitre présentent des réseaux bidimensionnels, les résultats sont néanmoins généralisables aux dimensions supérieures). Pour raisonner il est plus simple de considérer que l'échelle du réseau à emboîter est fixe (*e.g.* nous considérons le réseau support), et que c'est l'échelle du Voronoï récepteur que nous faisons varier ; ce dernier placé au centre du réseau support est donc dilaté. Dans ce cas :

- si ρ et r sont respectivement les rayons d'empilement et de recouvrement caractéristiques du réseau support (voir le chapitre 3) ;

- les rayons d'empilement et de recouvrement du Voronoï récepteur sont :

$$b.\rho \text{ et } b.r \text{ avec } b \in \mathbb{R} \ / \ b > 1$$

  Ces deux rayons sont aussi spécifiques du réseau dilaté (voir la figure 5.5).

Pour le réseau support les sphères de rayon ρ sont empilées régulièrement les unes contre les autres. Le point de contact entre deux de ces sphères appartient nécessairement à l'hyperplan médiateur entre les deux points du RRP aux centres des sphères. Cet hyperplan est exactement tangent aux deux sphères considérées. Notons aussi que l'empilement n'est pas modifié par un changement d'échelle du réseau.

Un emboîtage sous optimal est donc réalisé si nous dilatons le Voronoï récepteur d'un **facteur d'échelle** *:*

$$b = 2.n + 1 \ / \ n \in \mathbb{N}^*$$

Ainsi un maximum de points de contact entre sphères empilées du réseau support, correspondent à des points de contact entre sphères empilées du réseau dilaté. Ce résultat est recherché car en ces points de contact communs, les hyperplans médiateurs entre points du réseau support sont inclus dans les hyperplans médiateurs entre points du réseau dilaté. Les figures 5.6 et 5.7 présentent les exemples obtenus d'un emboîtement sous-optimal et d'un autre optimal. Le réseau régulier ainsi emboîté est **tronqué** car nous ne conservons que ses Voronoï entièrement ou partiellement à l'intérieur de celui dilaté.

La **hiérarchie** de RRP emboîtés est constituée de la suite de réseaux dont les échelles sont ajustées tels qu'ils puissent s'emboîter de proche en proche. Le facteur d'échelle entre deux réseaux consécutifs de la hiérarchie est donc b (voir la figure 5.8).

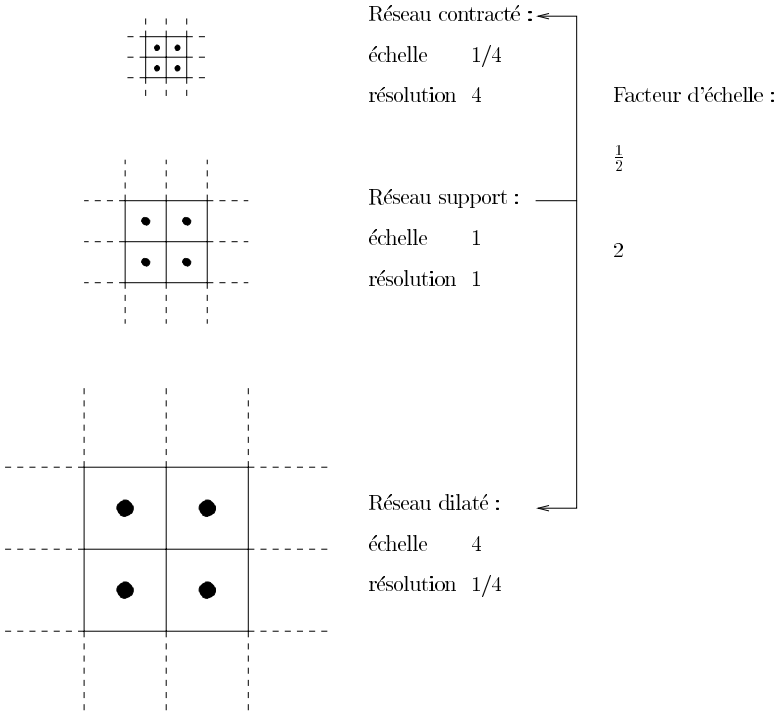


FIG. 5.4 – *Principe du changement d'échelle d'un réseau (exemple bidimensionnel).*

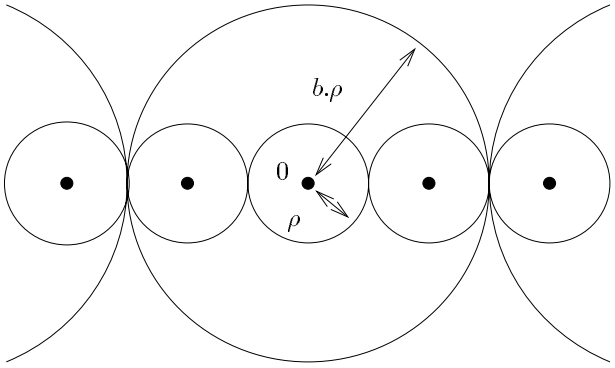


FIG. 5.5 – *Principe de l'emboîtage, le facteur d'échelle est* $b = 3$.

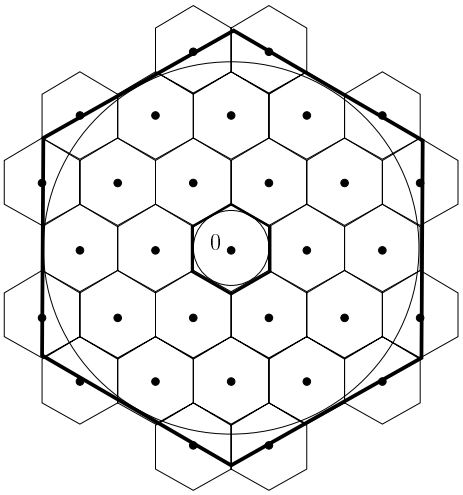


FIG. 5.6 – *Un emboîtement sous-optimal avec le réseau hexagonal ($b = 5$).*

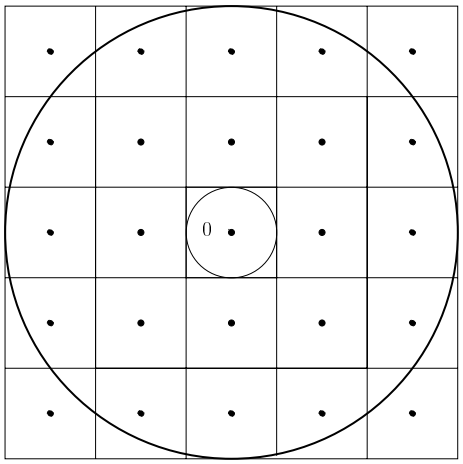


FIG. 5.7 – *Un emboîtement optimal avec le réseau cubique ($b = 5$).*

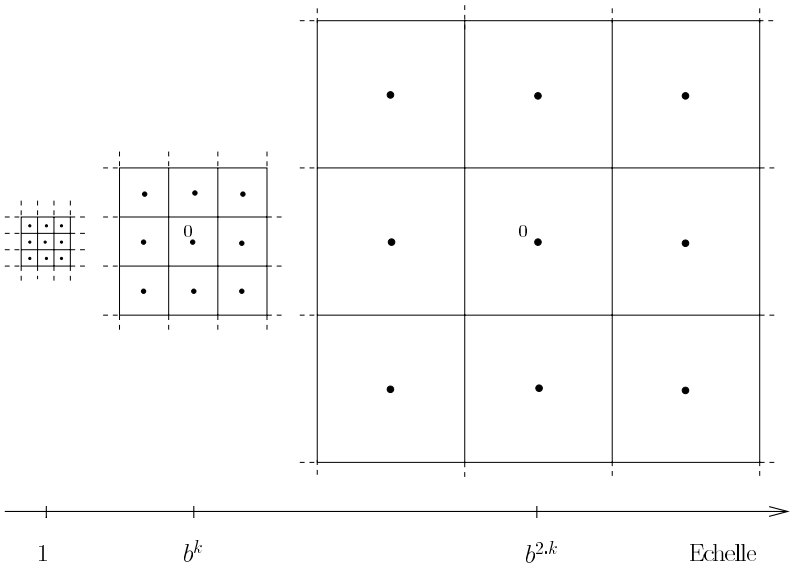


FIG. 5.8 – *Une hiérarchie de réseaux cubiques emboîtés ($b = 3$).*

5.3.3 Schéma du quantificateur

Le principe de la quantification vectorielle par emboîtage de RRP est le suivant :

- le vecteur à coder est projeté un première fois à l'intérieur du réseau tronqué de résolution la plus grossière ;
- ensuite, considérant le Voronoï dans lequel le point à coder se situe, ce dernier peut-être quantifié plus finement en le projetant dans le réseau de la hiérarchie de résolution juste supérieure. Cette opération est aussi décrite comme l'emboîtage d'un réseau tronqué dans le Voronoï ;
- ce processus d'emboîtage peut-être itéré.

Il est évidemment plus simple de ne modifier à chaque étape de la quantification, que l'échelle du vecteur à coder plutôt que de manipuler plusieurs réseaux à différentes échelles. Le schéma de quantification devient alors celui multi-étages de la figure 5.9, où seul l'algorithme de quantification rapide relatif au réseau support est mis en oeuvre à chaque étage. Le **premier facteur** de normalisation (ou d'échelle) F a *a priori* pour but d'inscrire tous les vecteurs de la source dans le RRP tronqué initial. Nous lui donnons pour valeur :

$$F = \frac{b.\rho}{\sqrt{\mathcal{E}_{max}}} \tag{5.1}$$

où $\mathcal{E}_{max}$ est l'énergie maximale d'un vecteur à coder, et ρ le rayon d'empilement du réseau. La source est ainsi, après normalisation par F, inscrite dans l'hyperboule de rayon $b.\rho$ (voir aussi le chapitre 3).
Dès lors, un vecteur à coder est nécessairement inscrit dans un Voronoï du RRP tronqué

le plus grossier de la hiérarchie. Si nous translatons ce Voronoï au centre du réseau, nous retrouvons la cas de la figure 5.5. Le nouveau facteur de normalisation à appliquer au vecteur source ainsi décalé, pour le projeter dans le réseau emboîté suivant, est simplement b.
Nous pouvons compléter la description du schéma de principe du QVAA de la figure 5.9, en précisant que :

- $\mathbf{x}$ est le vecteur source ;
- $\mathbf{y}_j$ est le vecteur représentant à l'étage j d'un RRP tronqué. C'est aussi le vecteur de translation à appliquer ;
- F est le facteur d'échelle utilisé pour projeter $\mathbf{x}$ dans le premier réseau tronqué ;
- b est le facteur de normalisation mis en oeuvre pour projeter chaque vecteur translaté dans le réseau emboîté suivant.

Remarquons que la valeur finale du vecteur de reproduction $\mathbf{y}$ associé au vecteur $\mathbf{x}$ est :

$$\mathbf{y} = \frac{1}{F}.\mathbf{y}_1 + \frac{1}{F.b}.\mathbf{y}_2 + \frac{1}{F.b^2}.\mathbf{y}_3 + \ldots = \frac{1}{F}.\sum_j \frac{\mathbf{y}_j}{b^{j-1}}$$

Ce schéma doit-être distingué de celui d'un QV en cascade classique (comme celui de la figure 4.1) car :

- le nombre d'étages de quantification peut varier pour différents vecteurs source (nous mettrons à profit cette propriété en construisant un arbre de quantification non-équilibré) ;
- seul l'index qui correspond au vecteur représentant de l'étage final sera transmis ;
- une QVA est effectuée à chaque étage, l'emboîtage des RRP tronqués est assuré par une simple normalisation par b des erreurs résiduelles.

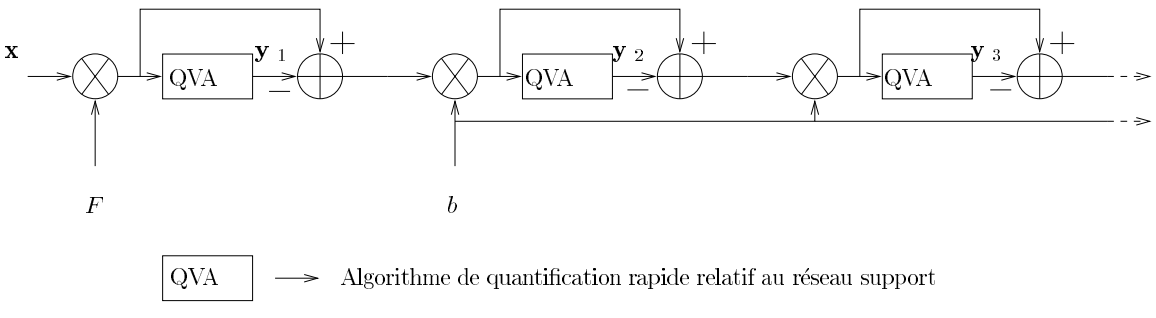


FIG. 5.9 – *Schéma de principe du QVAA.*

5.3.4 Un dictionnaire arborescent

Le dictionnaire construit par emboîtage d'une hiérarchie de RRP a une structure arborescente, où chaque noeud de l'arbre est étiqueté par un point d'un réseau :

- la racine est étiquetée par le point 0 auquel est associé l'espace source en entier ;
- les fils d'un noeud sont les points du réseau emboîté dans le Voronoï associé à ce noeud père ; l'arbre est donc B-aire, B étant égal au nombre de points d'un réseau emboîté ;
- à chaque profondeur de l'arbre correspond une échelle de la hiérarchie : plus la couche est profonde, plus la résolution est fine.

Nous présentons à la figure 5.11 des exemples de construction de dictionnaires. La hiérarchie est constituée de réseaux $\mathbb{Z}^2$ et le facteur d'échelle $b = b_{min} = 3$. Les vecteurs de la source synthétique sont les points blancs, leurs coordonnées indépendantes obéissent respectivement pour chacune des images d'une même ligne, aux lois normale, laplacienne ou gaussienne généralisée avec $\alpha = 0.6$. Cette figure illustre le cas où trois étages de quantification se succèdent. Un vecteur représentant et son Voronoï (tous deux en noir) ne sont représentés que s'ils sont mis en jeu lors de la quantification. Ces premiers résultats illustrent comment les emboîtages successifs des réseaux, permettent d'adapter directement la découpe de l'espace à la répartition des vecteurs source.
Les dictionnaires de ces figures sont des arbres équilibrés car les vecteurs source sont quantifiés à l'aide d'un ensemble de réseaux de même échelle. Il serait plus intéressant de répartir le débit alloué au quantificateur, afin que des zones de l'espace soient plus finement quantifiées que d'autres. L'arbre alors construit doit-être **non-équilibré**, tel que les régions de l'espace correspondant à des feuilles profondes soient finement quantifiées à l'aide de réseaux d'échelles inférieures, et tel que les régions de l'espace correspondant à des feuilles peu profondes soient grossièrement quantifiée à l'aide de réseaux d'échelle grande .
Pour contrôler une telle découpe de l'espace, il est nécessaire qu'elle se fasse progressivement ; le nombre de représentants injectés à chaque nouvel emboîtage doit-être limité [Eriksson94]. Ainsi les zones spatiales à découper sont localisées avec plus de précision, et les bits alloués sont répartis avec justesse. N'oublions pas qu'un codage entropique des feuilles n'est efficace, que si celles ayant une probabilité faible, sont peu nombreuses.

Ceci conduit à fixer le facteur d'échelle entre réseaux successifs de la hiérarchie à une valeur minimale :

$$b = b_{min} = 3$$

Cette nécessité de restreindre le nombre d'aires de l'arbre implique aussi que la dimension vectorielle soit limitée (typiquement les dimensions que nous retenons sont 2, 4, et 8).

Pour construire le dictionnaire arborescent non-équilibré, nous avons présenté au chapitre 4 les deux approches d'élagage ou de découpage. Cette dernière est alors préférable,

car le nombre d'aires de l'arbre du QVAA demeure grand. Le critère débit-distorsion du retour marginal est aussi adopté afin de déterminer les noeuds à découper.

La mise en oeuvre de cet algorithme requiert un **facteur d'apprentissage** suffisant, où ce dernier se définit comme le rapport du nombre de vecteurs d'apprentissage sur celui des représentants (*i.e.* les feuilles de l'arbre). Une borne minimale, que nous adoptons, est proposée dans [Gersho et al.92] avec 150. Le débit considéré étant entropique, cette contrainte sur le facteur d'apprentissage est introduite afin de limiter le nombre de vecteurs de reproduction. Elle implique aussi que la taille de la séquence d'apprentissage soit adaptée à la dimension vectorielle ; pour un débit donné, plus la dimension choisie de l'espace est grande, plus le nombre de représentants mis en jeu sera *a priori* important car le nombre d'aires de l'arbre croît. La taille de la séquence d'apprentissage doit aussi augmenter afin de disposer d'un champ de vecteurs à coder suffisamment dense et représentatif de la statistique de la source.

Nous précisons à ce niveau les solutions que nous avons apportées aux problèmes particuliers liés à l'emboîtage de réseaux. Ces traitements s'ajoutent donc au processus général de découpage de l'arbre décrit au chapitre 4 :

- un premier problème apparaît lorsqu'un vecteur source se retrouve seul dans un Voronoï. Si ce dernier est découpé, la hausse de débit est nulle et donc le retour marginal résultant infini. Ce Voronoï serait donc toujours re-découpé. C'est pourquoi nous imposons un seuil minimal de population pour le découpage d'une cellule (*i.e.* un Voronoï peut-être découpé si typiquement sa population est supérieure à 20 vecteurs) ;

- un second problème se pose lorsque, alors que le critère de population est vérifié, la hausse de débit due au découpage du Voronoï demeure pratiquement nulle. Ceci indique que les points à coder sont regroupés en un amas (voir la figure 5.10), et nous voulons éviter le sur-découpage de cette cellule. Nous ajoutons alors un seuil de distorsion (*i.e.* si $\Delta l \approx 0$, le Voronoï peut-être découpé si sa distorsion est suffisante) ;

- un dernier cas de figure posant problème arrive, lorsque plusieurs Voronoï découpés produisent la même valeur de retour marginal (*e.g.* si les échantillons codés sont des entiers, ou la précision insuffisante). Nous choisissons alors de découper le noeud le plus éloigné du point 0, ce choix est justifié dans l'explication qui suit.

La figure 5.12 montre les exemples des arbres élagués correspondant aux sources synthétiques de la figure 5.11 (les découpages de ces arbres conduisent à des résultats identiques). Elle illustre comment la QVAA adapte la découpe de l'espace, non seulement en fonction de la distribution de la source, mais aussi en fonction du critère débit-distorsion ; pour un débit donné, le QVAA découpe grossièrement la région de l'espace où se concentrent les vecteurs source peu énergétiques, alors la région spatiale moins dense où se situent les vecteurs source énergétiques peut-être découpée plus finement.
Ces sources synthétiques offrent une première modélisation de la distribution statistique

de sources image hybrides, et l'analyse de la figure 5.12 permet de comprendre pourquoi la QVAA est adaptée à leur quantification, car :

- les vecteurs peu énergétiques et ayant la probabilité la plus grande, sont ceux dont les composantes correspondent aux coefficients des régions "homogènes" des sous-images (*i.e.* les régions des sous-images où les erreurs transformées de prédiction de compensation du mouvement ont peu d'amplitude. Ce sont typiquement des régions bruitées non affectées par les objets en mouvement ou, l'intérieur de zones peu texturées en mouvement). Ces vecteurs, pauvres en information visuelle pertinente, peuvent être fortement quantifiés ;

- les vecteurs source énergétiques et de probabilité faible, sont ceux dont les coordonnées correspondent aux coefficients des régions "inhomogènes" des sous-images (*i.e.* ce sont les zones des sous-images, marquées par les objets en mouvement). Pour une bonne restitution des images, ces vecteurs doivent être finement quantifiés.

Cette zone centrale de l'espace où la quantification est plus grossière est la généralisation au cas vectoriel de la "dead zone" déjà introduite avec la quantification scalaire [Woods91]. Cette dernière est mise en oeuvre avec MPEG1&2 pour le codage des images non-intras (voir le chapitre 1).

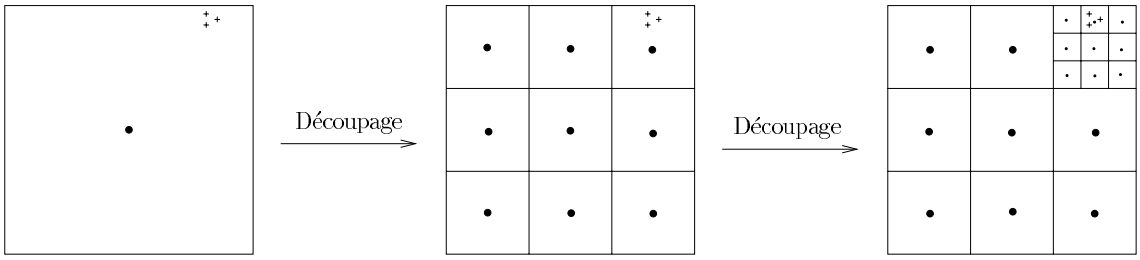


FIG. 5.10 – *Sur-découpage du Voronoï car les vecteurs source (les croix) sont regroupés en amas.*

5.3.5 Dénombrement des points du réseau emboîté

Nous rappelons que le dénombrement et l'indexage des points des réseaux, sont des points difficiles de la QVA. Pour la QVAA, il est aussi nécessaire afin d'indexer les vecteurs représentants, de dénombrer les points du réseau support emboîté et donc tronqué.

Dans le cas du réseau cubique, ce nombre est simplement b^k.

Cependant pour les autres RRP, le dénombrement est moins trivial car la forme des Voronoï est complexe et l'emboîtement sous-optimal. Nous proposons donc de majorer le nombre de points du réseau tronqué, en majorant l'énergie de la sphère les contenant ; alors en utilisant la série Thêta du réseau (voir le chapitre 3), nous obtiendrons une borne maximale.

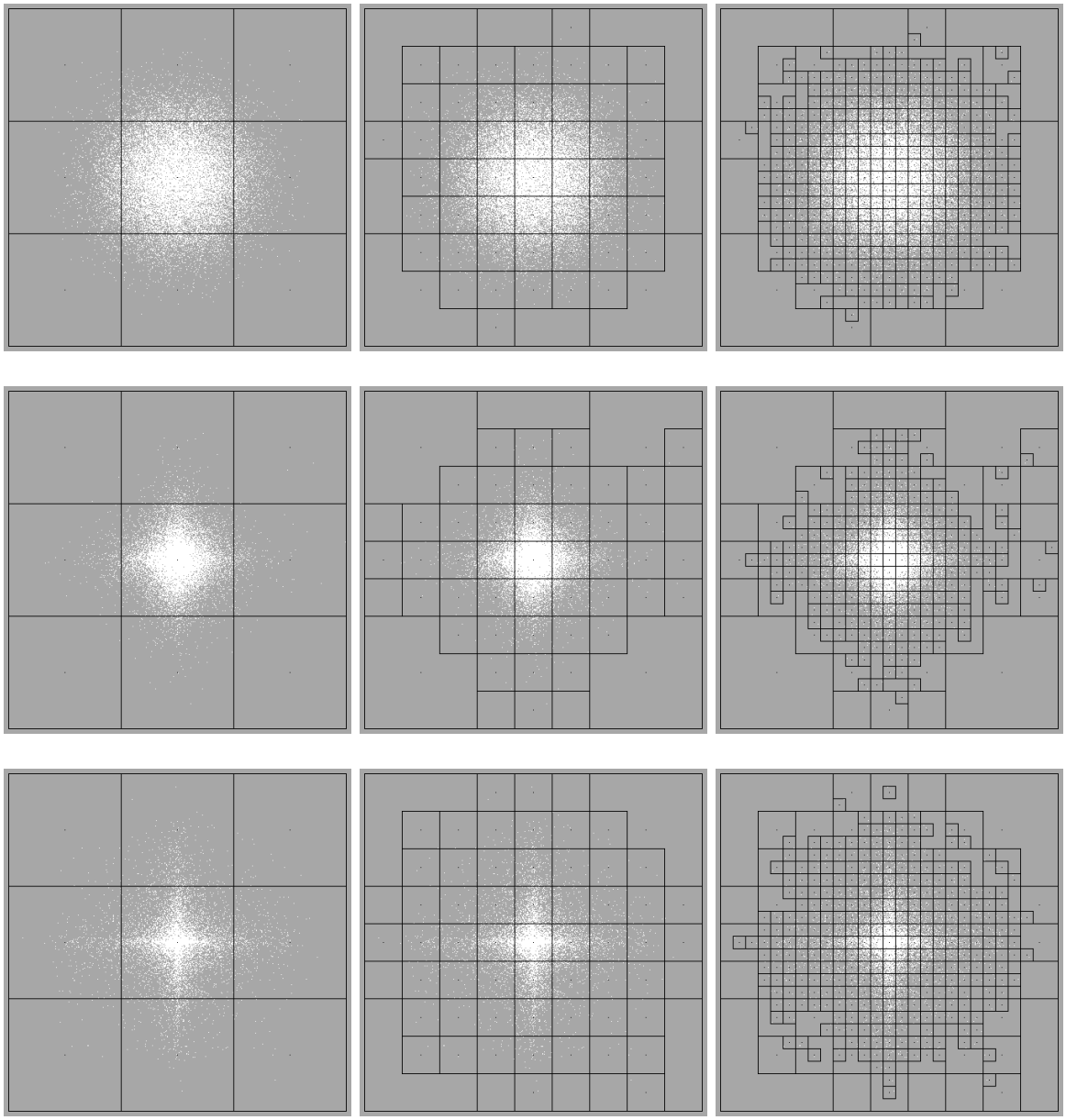


FIG. 5.11 – *Construction de dictionnaires par emboîtage d'une hiérarchie de réseaux* $\mathbb{Z}^2$ *(*$b = b_{min}$*). Les points blancs sont les vecteurs source. Les images sur une même ligne sont relatives aux trois étages de quantification d'une source i.i.d normale (première ligne), laplacienne (seconde ligne) et gaussienne généralisée avec* $\alpha = 0.6$ *(troisième ligne).*

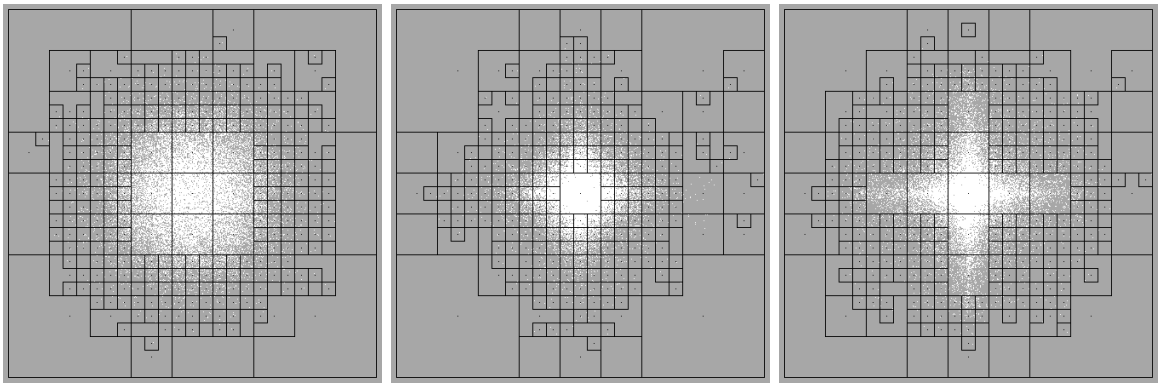


FIG. 5.12 – *[a][b][c] - Emboîtage de réseaux $\mathbb{Z}^2$ et élagage (ou découpage) de l'arbre. Les points blancs sont les vecteurs source, leurs coordonnées i.i.d obéissent respectivement à une loi normale [a], laplacienne [b] et gaussienne généralisée [c] ($\alpha = 0.6$).*

Nous considérons le cas où le vecteur à coder se situe initialement dans le Voronoï au centre du réseau support. Ce vecteur est projeté dans le réseau emboîté suivant en lui appliquant le facteur d'échelle b. Le point projeté peut avoir comme plus haute énergie $(b.r)^2$, il correspond dans ce cas à un trou du réseau dilaté. Son représentant dans le réseau emboîté peut-être à l'extérieur de la sphère de rayon $b.r$, cependant ce représentant ne peut-être à une distance supérieure à r du vecteur qu'il représente (voir la figure 5.13).

Par conséquent le nombre de points du réseau support emboîté est majoré en considérant le nombre de points à l'intérieur et sur la sphère de rayon $(b+1).r$

Cette borne demeure grossière si nous fixons $b = b_{min}$, car le nombre de points à l'intérieur de la sphère est faible comparé à celui des points périphériques. Nous voulons alors affiner la valeur de la borne maximale, et la rapprocher du nombre exact de points du réseau emboîté.

Dans le cas où $b = b_{min}$, nous voulons montrer que le nombre de points du réseau support emboîté est majoré en comptant les points à l'intérieur et sur la sphère de rayon $b_{min}.r$

Nous distinguons trois cas :

1. soit le point à coder, avant projection, est à l'intérieur de la sphère de rayon ρ. Une fois projeté, il est à l'intérieur de la sphère de rayon $3.\rho$ et le résultat est donc vérifié (voir la figure 5.14 [a]).

2. le cas qu'il faut mieux étudier est lorsque le point à coder avant projection est un trou (il est donc sur la sphère de rayon r), projeté ce point est aussi un trou du

réseau dilaté et il est à la distance maximale du centre du réseau emboîté :

- un cas de figure se produit lorsque le rayon de recouvrement r caractéristique du réseau support est maximal, alors les sphères empilées de ce réseau forment entre elles un angle de $\pi/2$ et, suivant un axe, les sphères de rayon r ne se recouvrent pas (elles sont empilées). Dans ce cas, le point à coder projeté est encore un trou du réseau support ; il est donc à la même distance de deux points du réseau emboîté, et celui qui est choisi comme représentant est à l'intérieur de la sphère de rayon $3.r$ (pour $k = 2$, il est possible d'établir la relation : $r = \rho.\sqrt{2}$, voir la figure 5.14 [b]) ;
- un autre cas de figure se produit lorsque r est minimal, les sphères empilées sur la sphère centrale de ce réseau forment entre elles un angle de $\pi/3$. Dans ce cas, le point à coder projeté se situe au centre d'un Voronoï du réseau support et son représentant est sur la sphère de rayon $3.r$ (pour $k = 2$, il est possible d'établir la relation $r = \rho.2/\sqrt{3}$, voir la figure 5.14 [c]) ;
- pour les autres cas de figure le point à coder projeté est donc dans un Voronoï du réseau emboîté et tel que son représentant associé soit à l'intérieur de la sphère de rayon $3.r$. Plus r est grand, plus ce représentant est proche du rayon $3.r$.

3. enfin, le point à coder peut se situer, avant projection, entre les sphères de rayon ρ et r, il demeure toujours inscrit dans le Voronoï central. Ce dernier est un polytope convexe, ses frontières sont des hyperplans, aux sommets des intersections entre ces hyperplans sont les trous. Ce point à coder se situe donc dans un "coin" du Voronoï proche d'un trou. Nous avons montré qu'un trou projeté a son représentant inscrit à l'intérieur de la sphère de rayon $3.r$. Ici, le facteur de projection ayant une valeur petite, ce point à coder projeté a alors une forte probabilité d'être représenté de la même façon que le trou projeté qui lui est proche.

5.3.6 Indexage

La QVAA consiste en une quantification multi-étages où, pour chacun d'eux, le même algorithme de QVA est mis en oeuvre. Une table de correspondance est donc d'abord construite. Elle permet d'obtenir un indice pour chacun des points du RRP tronqué. Cette table constituée hors ligne est connue du codeur et du décodeur. Sa taille est réduite car celle du réseau tronqué est elle-même limitée.
Le dictionnaire une fois construit doit être transmis au décodeur. Sa structure arborescente fait que l'information suffisante pour le caractériser est compacte. En effet, après avoir fixé un ordre de parcours de l'arbre de la racine jusqu'à la dernière feuille (afin de numéroter les noeuds), il suffit pour stocker l'arbre de le re-parcourir en mémorisant pour chaque noeud :

- les numéros des fils,
- celui du père,

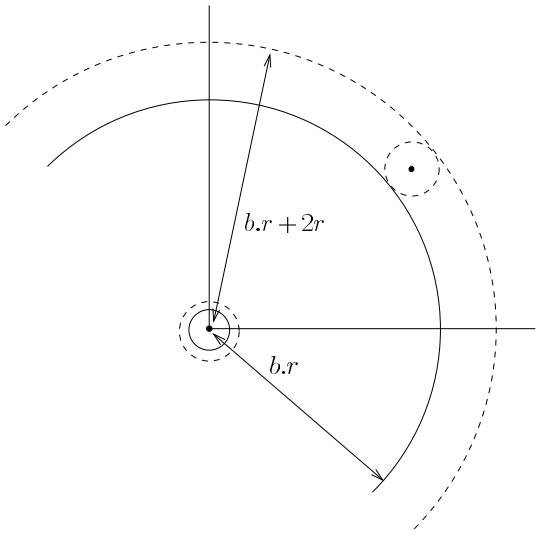


FIG. 5.13 – *Majoration de l'énergie de la sphère contenant les points du réseau tronqué: ces points sont nécessairement inscrits dans la sphère de rayon* $(b+1).r$.

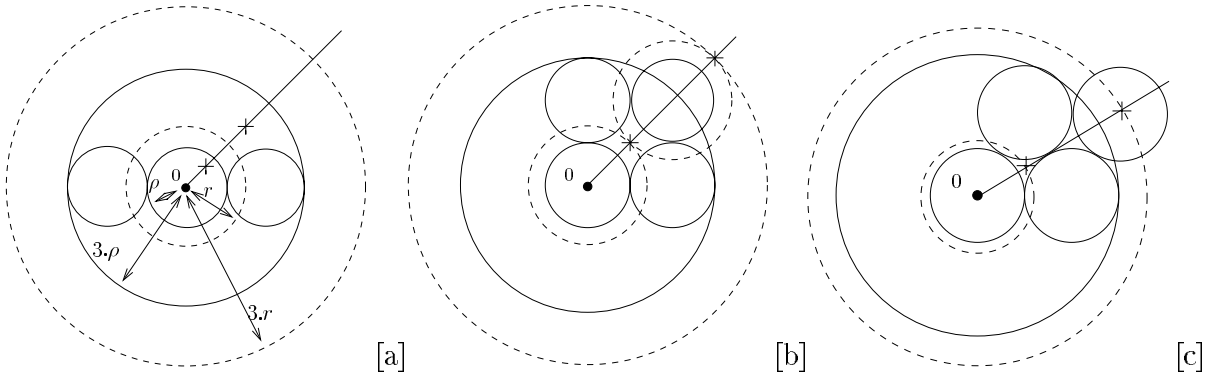


FIG. 5.14 – *Exemples bidimensionnels illustrant que, pour* $b = b_{min} = 3$*, le réseau emboîté est constitué de points à l'intérieur ou sur la sphère de rayon* $3.r$*. Le point à coder, avant et après projection, est symbolisé par une croix. Nous retrouvons respectivement les cas 1 ([a]) et 2 ([b] où* r *est maximal et [c] où* r *est minimal).*

– l'indice du représentant (appartenant à la table de correspondance définie précédemment).

Les feuilles portent en plus un mot de code entropique. Remarquons que le dictionnaire étant construit par apprentissage, nous accédons directement aux probabilités d'occurence des noeuds terminaux afin d'achever ce code.
Dans le contexte de la quantification adaptative, il faut pouvoir réactualiser le dictionnaire. Avec la QVAA nous pourrions procéder en conservant fixe la souche de l'arbre (*i.e.* les noeuds des couches hautes), et en remettant à jour uniquement des branches.

5.3.7 Détermination du réseau optimal

L'étude vise à déterminer parmi $\mathbb{Z}^k$, D_k, E_8 et Λ_{16}, le RRP le plus efficace dans ce contexte d'emboîtage. Des courbes expérimentales débit *vs.* distorsion (*i.e.* entropie du dictionnaire *vs.* distorsion moyenne) sont présentées à la figure 5.15 a. Elles sont obtenues par le codage d'une source synthétique i.i.d normale ($\sigma^2 = 1$) et par découpage de l'arbre (avec le facteur d'échelle $b = b_{min}$). Précisément nous comparons $\mathbb{Z}^4$ à D_4, et $\mathbb{Z}^8$ à E_8. Dans les deux cas, le réseau $\mathbb{Z}^k$ **apparait plus performant** (de plus bas débits sont atteints, et la distorsion est inférieure). Les courbes de la figure 5.15 b obtenues avec Z^k ($k = 1, 2, 4, 8$ et 16), montrent alors que de plus bas débits sont évidemment atteints lorsque la dimension vectorielle croît.

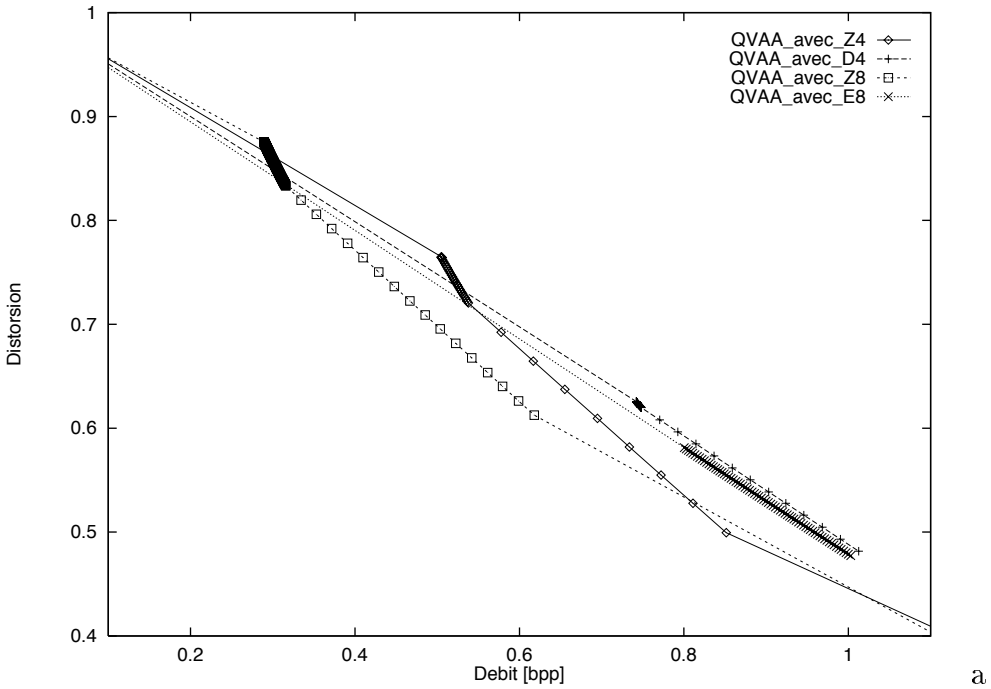
Z^k se distingue car seul ce réseau fournit un emboîtement optimal. Afin d'appréhender l'intérêt de cette propriété, examinons la figure 5.16 qui présente l'exemple d'un emboîtement (optimal) avec $\mathbb{Z}^2$ et celui (sous-optimal) avec le réseau hexagonal (ce dernier est choisi car c'est le RRP meilleur quantificateur relativement à la dimension 2). La source est distribuée uniformément dans les zones grisées. Dans le cas du réseau cubique, les points injectés participent identiquement à la baisse de distorsion moyenne et à la hausse de débit entropique entraînées par le découpage. Dans le cas hexagonal, 6 des 13 points injectés ont une probabilité d'occurence moindre. Ceci se traduit par rapport au cas précédent, par une hausse supérieure du débit entropique et une baisse moindre de la distorsion moyenne (voir l'annexe E).
Retenir le réseau cubique implique de ne pas tirer profit du gain de partitionnement de la QV sur la QS, mais ce gain espéré est peu significatif (voir le chapitre 3).
Un autre avantage de Z^k pour la QVAA est que ce réseau est moins dense que les autres ; le nombre de points injectés à chaque nouvel emboîtage sera limité, ce qui ajoute à la précision du découpage de l'espace vectoriel.

5.3.8 Traitement des vecteurs hors-norme

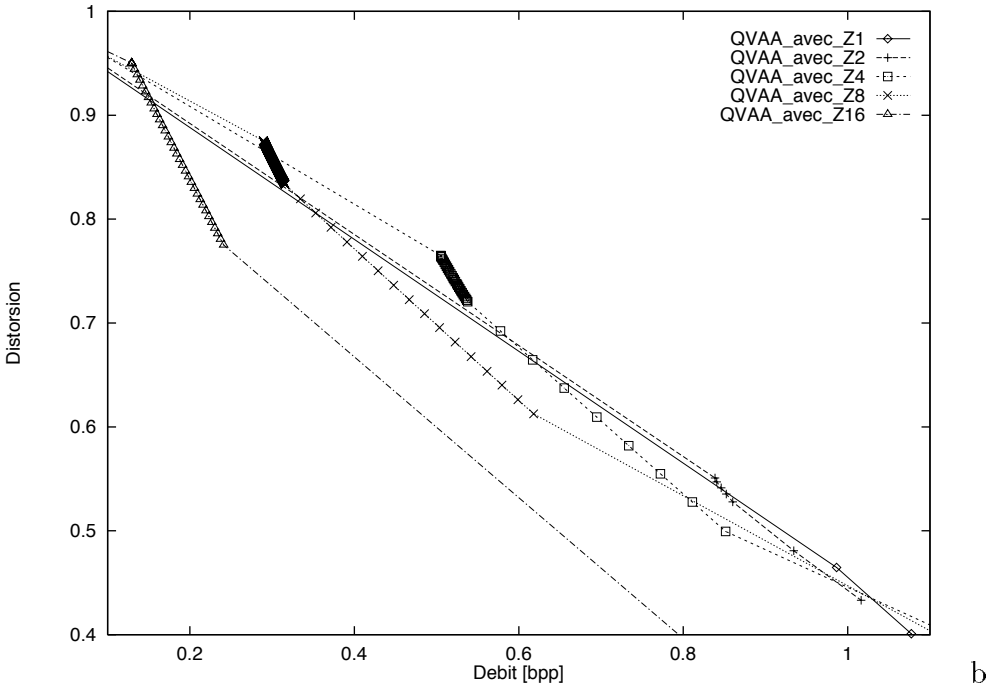
Les étapes précédentes nous ont conduit à adopter pour le QVAA :

– un facteur d'échelle pour l'emboîtage égal à b_{min},

– une approche par découpage de l'arbre,



a



b

FIG. 5.15 – *Courbes débit vs. distorsion (entropie du dictionnaire vs. distorsion) obtenues par découpage du dictionnaire arborescent et mettant en jeu différents réseaux. La source i.i.d obéit à une loi normale* ($\sigma^2 = 1$).

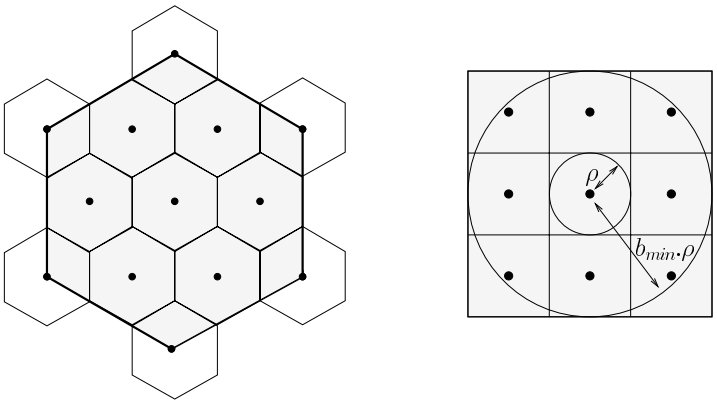


FIG. 5.16 – *Un emboîtement optimal avec le réseau cubique, un autre sous-optimal avec le réseau hexagonal ($b = b_{min}$).*

– le réseau cubique.

Le dictionnaire est alors construit *a priori* à partir d'une séquence d'apprentissage (suffisamment grande) représentative de la statistique de la source. En pratique, les vecteurs ensuite quantifiés peuvent demeurer hors du dictionnaire (*i.e.* s'ils sont trop énergétiques). Il faut donc mettre en place pour achever le QVAA :

– un test de norme à l'entrée du quantificateur,

– un traitement particulier relatif aux vecteurs marginaux hors-norme.

Ce QV autorise alors des solutions peu coûteuses car elles mettent en oeuvre la norme L_∞ des vecteurs.

Détection des vecteurs source hors-norme

L'équation 5.1 du facteur F de normalisation est devenue :

$$F = \frac{b_{min}.\rho}{\sqrt{\mathcal{E}_{max}}} \tag{5.2}$$

Nous rappelons que F est introduit pour projeter le vecteur à coder au sein du premier cube de la hiérarchie. La norme L_∞ des vecteurs $\mathbf{u}$ appartenant à ce dernier est donc telle que (voir aussi la figure 5.16) :

$$L_\infty(\mathbf{u}) = \max_{i=1,\ldots,k} |u_i| \ \leq \ (b_{min} \times \rho)$$

La norme L_∞ avant normalisation d'un vecteur $\mathbf{x}$ pouvant-être quantifié par ce dictionnaire doit-être telle que :

$$L_\infty(\mathbf{x}) = \max_{i=1,\ldots,k} |x_i| \ \leq \ \sqrt{\mathcal{E}_{max}}$$

Traitement des vecteurs source hors-norme

Nous proposons d'adapter la méthode simple décrite au chapitre 3 consistant à projeter le vecteur hors-norme sur le dictionnaire, et à le traiter au décodage comme les autres vecteurs. L'avantage de cette approche est qu'il n'y a pas d'information supplémentaire à transmettre. Avec le QVAA l'opération est encore plus aisée, car pour projeter un vecteur hors-norme sur le premier cube de la hiérarchie, il suffit de faire (dès le test de sa norme) :

$$\text{Si} \quad |x_i|_{i=1,\ldots,k} > \sqrt{\mathcal{E}_{max}} \implies x_i = \text{signe}(x_i).\sqrt{\mathcal{E}_{max}}$$

Ces vecteurs hors-normes sont en principe peu probables ; ainsi le bruit de surcharge créé demeure faible.

5.3.9 Allocation binaire entre sous-bandes

Nous avons formalisé le problème de l'allocation entre les sous-bandes au chapitre 2, et décrit la forme générique de l'algorithme permettant, par programmation convexe, de réaliser une allocation binaire optimale. A titre d'exemple la figure 5.17 montre les courbes débit-distorsion expérimentales obtenues, lors de la construction des dictionnaires des QVAA relatifs à quatre sous-bandes. Nous ne détaillons pas ici le contexte expérimental (ce sera fait au chapitre 6) ; nous indiquons uniquement que le schéma de codage est celui de la figure 5.1 où la transformée est une DCT sur des blocs 2x2.

Lors de la construction des dictionnaires, le découpage des arbres selon le critère du retour marginal maximal a permis d'obtenir directement les enveloppes convexes relatives à chacune des sous-bandes. Chacun des points d'une enveloppe constitue un quantificateur potentiel pour la sous-image. La combinaison de tous ces quantificateurs est représentée par le nuage de points de la figure 5.19.
La mise en oeuvre de l'algorithme de Shoham [Shoham et al.88] produit alors l'**enveloppe convexe globale** du nuage. Ce résultat correspond à une allocation binaire optimale entre les sous-bandes (voir la figure 5.19). Nous décrivons succintement l'algorithme légèrement modifié (car les débits ne sont pas des entiers) que nous mettons en oeuvre. L'hypothèse imposée aux quantificateurs des sous-bandes, d'avoir des débits rangés par ordre croissants, est vérifiée directement par la QVAA.

La méthode générique est celle décrite au chapitre 2 avec la recherche pour différents λ, de celui optimal permettant de trouver le point C de la figure 2.8. Mais l'optimisation se base cette fois sur les valeurs singulières de λ offrant plusieurs solutions à l'équation 2.17 (voir aussi la figure 5.18). Pour une valeur quelconque de λ caractérisant un point de l'enveloppe convexe globale, il est possible de calculer (si elles existent) les deux valeurs singulières λ_{inf} et λ_{sup} les plus proches, telles que $\lambda_{inf} < \lambda < \lambda_{sup}$. Le calcul de λ_{inf} (respectivement λ_{sup}) consiste :

- à rechercher séparément pour chacune des sous-bandes le retour marginal maximal (respectivement minimal), mais parmi l'ensemble restreint des quantificateurs po-

tentiels dont les débits sont supérieurs (respectivement inférieurs) au quantificateur caractérisé par λ ;

– puis à choisir parmi ces retours marginaux le plus grand (respectivement petit).

Un seul quantificateur potentiel (parmi ceux des sous-bandes) a donc été déterminé. Le calcul de λ_{inf} (respectivement λ_{sup}) permet alors de trouver directement le point de l'enveloppe convexe globale qui suit (respectivement précède) celui caractérisé par λ.
Il est possible de restreindre le domaine de recherche des quantificateurs relatifs au calcul de λ_{inf} (respectivement λ_{sup}), si nous connaissons λ_1 et λ_2, les pentes caractéristiques de deux autres points de la courbe convexe globale, et telles que $\lambda_1 < \lambda < \lambda_2$. Ainsi pour trouver λ_{inf} (respectivement λ_{sup}), le domaine de recherche parmi les quantificateurs potentiels est limité à ceux dont les débits sont inférieurs (respectivement supérieurs) aux quantificateurs caractérisés par λ_1 (respectivement λ_2).
Pour trouver le point de l'enveloppe convexe globale de débit juste supérieur ou égal à celui voulu (*i.e.* R_d de l'équation 2.15), le processus est initialisé en fournissant deux premiers points de cette enveloppe : l'un de débit supérieur à celui cible, l'autre de débit inférieur (*e.g.* le premier point est obtenu en faisant la somme relativement aux sous-bandes des distorsions pour les débits minimaux, le second en faisant la somme pour les débits maximaux). A partir du point initial de débit inférieur, par les calculs successifs de valeurs singulières λ_{inf}, l'algorithme converge en trouvant un à un et pour des débits croissants les points de l'enveloppe convexe globale. La recherche stoppe dès que le point de débit juste supérieur (ou égal) à celui cible est atteint. Pour chaque nouveau point trouvé le processus s'accélère, car un domaine de recherche des quantificateurs potentiels a diminué. Evidemment, plus l'intervalle initial sera faible, plus rapide sera le résultat.

La répartition des points du nuage de la figure 5.19 n'est pas homogène (*i.e.* certaines zones sont très denses et d'autres vides) car les courbes débit-distorsion originales sont très discrétisées. Les points de l'enveloppe convexe globale sont donc également rares, et les débits accessibles trop éparses. Nous proposons donc d'**adapter l'algorithme d'allocation binaire** précédent afin d'atteindre des points intermédiaires juste au-dessus de la courbe convexe globale.
Nous voulons donc atteindre les quantificateurs optimaux comme ceux représentés à la figure 5.19. La seule manière sûre de les atteindre tous, serait d'effectuer une recherche exhaustive entre tous les points des courbes débit-distorsion, mais ce n'est pas envisageable. Il faut donc faire une recherche sous-optimale en partant d'un point de l'enveloppe convexe globale. La procédure précédente nous a rendu les deux points de cette enveloppe dont les débits bornent au plus près celui à atteindre. Nous proposons de remettre en oeuvre de façon locale le processus en l'initialisant avec ces deux points. Mais pour ne pas limiter le nombre de solutions pour les nouveaux calculs de λ_{inf}, nous ne restreignons pas le domaine de recherche parmi les quantificateurs potentiels de débits supérieurs. Nous obtenons alors une **portion d'enveloppe convexe** locale entre les deux points initiaux.
La figure 5.20 présente le résultat obtenu (*i.e.* la courbe intitulée "quantificateurs intermédiaires") en effectuant une recherche systématique entre tous les bipoints de l'enveloppe convexe globale. A des fins de comparaison nous avons aussi inscrit le nuage de points

ainsi que l'ensemble des quantificateurs optimaux. Un grand nombre de ces derniers est alors atteint.

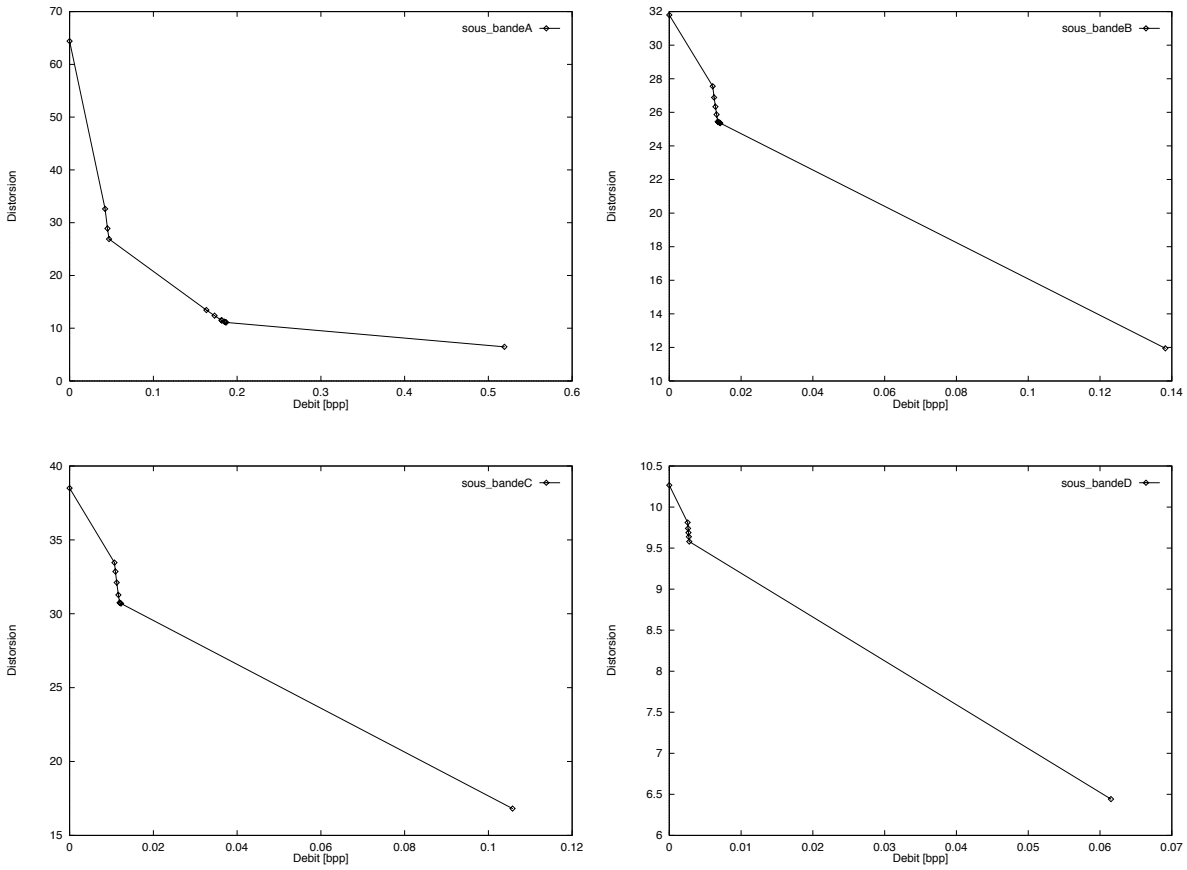


FIG. 5.17 – *Courbes débit-distorsion (entropie/distorsion) relatives à la construction des dictionnaires de quatre sous-bandes.*

5.4 Conclusion

Nous avons décrit les étapes de la construction du QVAA qui consiste en l'emboîtage d'une hiérarchie multi-échelles de RRP. Ce QV est une généralisation au cas vectoriel de l'approche d'emboîtement de QS abondamment utilisée en codage hiérarchique, car elle offre une compatibilité entre les quantificateurs définis par couches et pour différents débits [Taubman et al.94]. Nos choix du facteur d'échelle, de découpage de l'arbre et du réseau $\mathbb{Z}^k$, se justifient en fonction du critère débit-distorsion et par la nécessité de limiter le nombre d'aires de l'arbre. La QVAA offre alors des solutions originales aux problèmes posés par la QVA, avec :

– une découpe de l'espace fonction de la distribution de la source,

- un indexage basé sur la structure arborescente du dictionnaire,
- une détection et un traitement simples pour les vecteurs hors-norme.

Ce QV tire aussi profit des RRP car le dictionnaire à transmettre ne contient pas de représentants, et car la quantification demeure rapide (*e.g.* si h est la hauteur de l'arbre pour quantifier un vecteur, la complexité d'encodage est de l'ordre de $h.k$, et typiquement $h = 3$).
La QVAA est particulièrement adaptée à la quantification de séquences d'images hybrides car le critère débit-distorsion introduit pour le découpage, fait apparaître judicieusement une zone grossièrement quantifiée autour du point 0. Cependant la répartition des points débit-distorsion accessibles est trop discrétisée ; nous avons donc du développer un algorithme d'allocation binaire entre les sous-bandes permettant d'accéder à un plus grand nombre de quantificateurs optimaux.
Nous disposons à ce niveau d'un outil de quantification que nous pouvons tester expérimentalement au sein d'un codeur réel. C'est l'objet du prochain chapitre.

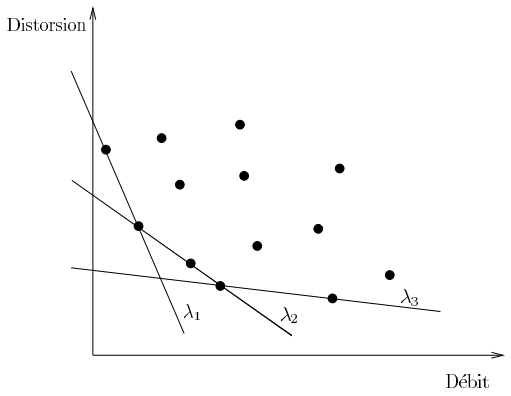


FIG. 5.18 – *Exemples de valeurs singulières de λ : λ_1 et λ_3 offrent deux solutions à l'équation 2.17, λ_2 trois solutions.*

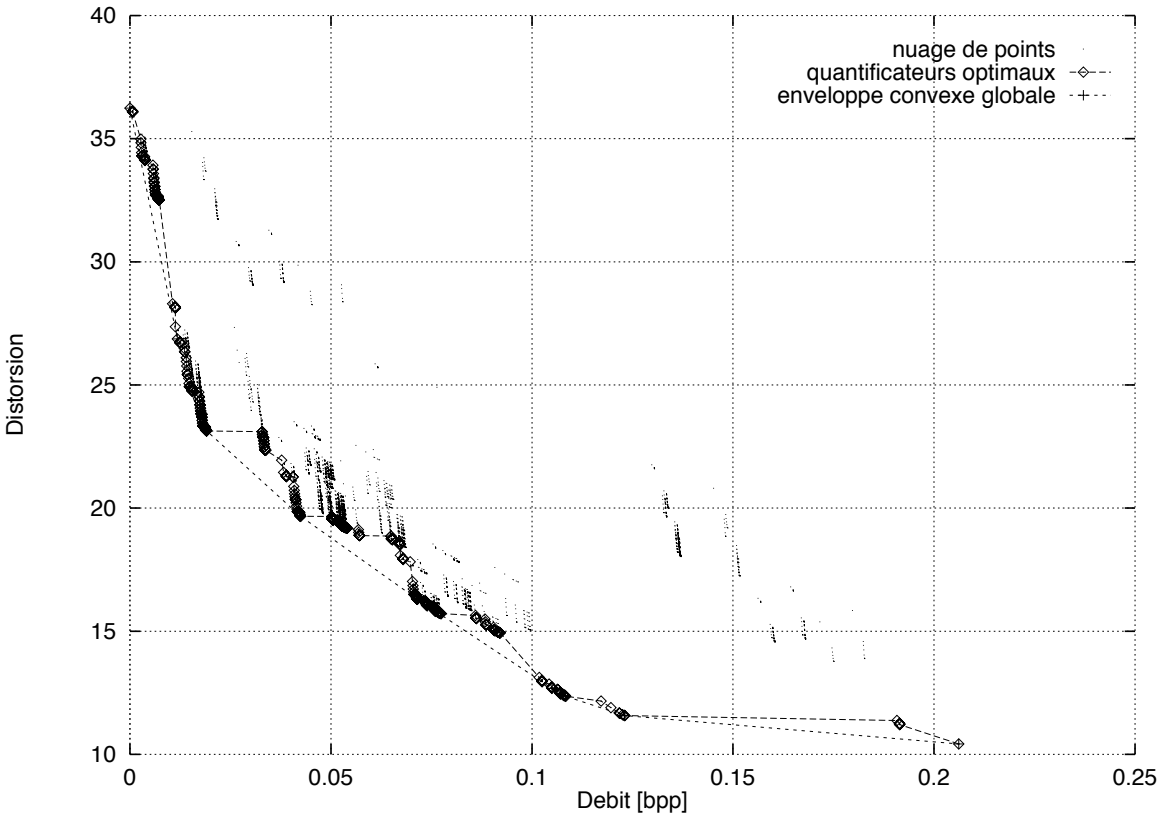


FIG. 5.19 – *Nuage de points relatifs aux courbes débit-distorsion, enveloppe convexe globale et quantificateurs optimaux.*

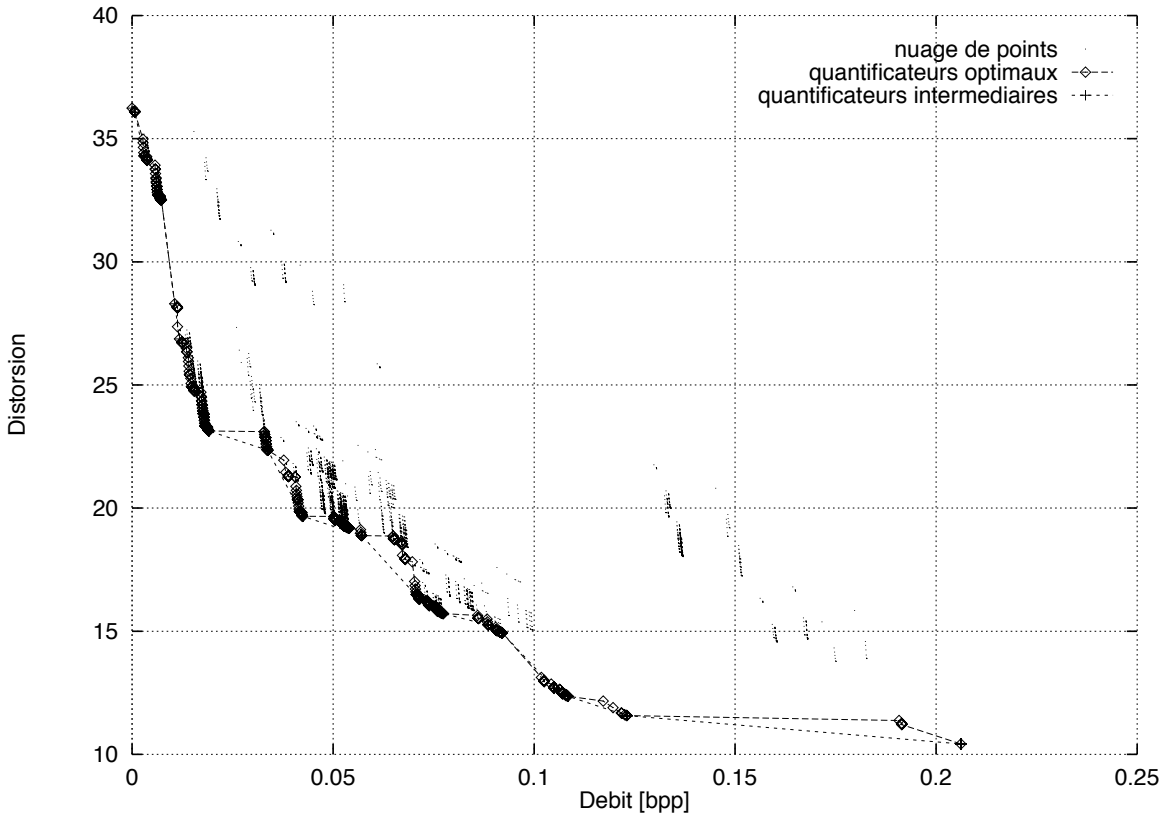


FIG. 5.20 – *Nuage de points relatifs aux courbes débit-distorsion, quantificateurs optimaux et quantificateurs intermédiaires obtenus.*

Chapitre 6

Etude expérimentale et validation de la QVAA

6.1 Introduction

La description complète du QVAA a été faite au chapitre 5. Il s'agit à présent de le tester expérimentalement en l'intégrant au sein de schémas de codage hybride pour la compression de séquences d'images. Afin d'explorer les potentialités du QVAA pour le codage bas débits de scènes visiophoniques, nous mettons en oeuvre deux codeurs :

- le premier emploie des outils classiques de type MPEG [Legall91] [Montrichard et al.96b],
- le second est basé régions.

La figure 5.1 illustre le schéma générique du codeur en sous-bandes. Les différents modules relatifs à la transformée, à l'estimation et à la compensation du mouvement sont donc choisis en fonction des deux types de codeurs. Nous rappelons que les outils de décorrélation ont été présentés au chapitre 1, et que nous n'étudions que la QVAA des images d'erreurs de prédiction transformées.
De façon générale nous distinguons trois phases expérimentales :

- la constitution des séquences d'apprentissage relatives à chacune des sous-bandes, le codeur étant alors en boucle ouverte (BO) ;
- la construction d'un dictionnaire par sous-image à partir de l'ensemble d'apprentissage correspondant ;
- l'encodage de séquences d'images.

Les principaux résultats numériques sont appréciés en calculant :

- le rapport signal sur bruit crête ou "Peak Signal to Noise Ratio". Ce dernier est couramment utilisé pour des images dont l'intensité des pixels est comprise entre 0

et 255, même s'il ne constitue pas une approximation du critère visuel humain. En reprenant les notations introduites à la figure 5.1, si la taille des images est $N_x \times N_y$ et l'indice d'un pixel (x, y), nous avons [en dB] :

$$\begin{aligned} PSNR &= 10.\log_{10} \frac{255^2}{\frac{1}{N_x.N_y}.\sum_{x=1}^{N_x}\sum_{y=1}^{N_y} d\left(e(x,y), e_q(x,y)\right)} \\ &= 10.\log_{10} \frac{255^2}{\frac{1}{N_x.N_y}.d\left(e, e_q\right)} \\ &= G_p + G_q \end{aligned}$$

En effet, afin de considérer séparément l'influence du quantificateur et du prédicteur dans le système fonctionnant en boucle fermée, sont introduits :

. le gain de prédiction [en dB] :

$$G_p = 10.\log_{10} \frac{255^2}{\frac{1}{N_x.N_y}.d(e)}$$

. le gain de quantification [en dB] (la transformée est choisie orthogonale) :

$$G_q = 10.\log_{10} \frac{d(e)}{d(e, e_q)}$$

– les entropies des dictionnaires (voir le chapitre 2).

Les séquences d'images employées sont QCIF, avec les séquences "Salesman" (450 images), "MissAmerica" (108 images) et "Claire" (450 images). Ces séquences sont toutes du même type "tête-épaules" (voir les images de la figure 6.1). Salesman, où le personnage manipule un objet, est alors la plus complexe des trois.
Lorsque des chiffres du coût CPU sont mentionnés, la machine d'expérimentation cible est une SPARC-Station 5.5 (110 Mhz).

6.2 Codage de type MPEG de séquences d'images

Nous qualifions ce schéma de codage de "type MPEG" car les outils retenus comme modules du codeur de la figure 5.1 sont aussi mis en oeuvre par MPEG pour la compression des images prédites. Nous avons pris comme paramètres d'implantation :

– une prédiction du mouvement par mise en correspondance de blocs 16x16 (la fenêtre de recherche est de taille 32x32). L'estimation du mouvement translationnel est faite avec une précision au pixel entier ;

FIG. 6.1 – *Images originales extraites de la base de données avec (de gauche à droite) Salesman, MissAmerica et Claire.*

- une compensation du mouvement en avant. Pour une image donnée, la prédiction est faite par compensation du mouvement de l'image décodée précédente et en utilisant l'information de mouvement correspondante. Pour initialiser le processus, la première image de la séquence n'est pas codée ;
- une transformée DCT des images d'erreurs de prédiction sur des blocs 2x2, suivie d'une configuration en intra-bande.

La taille de la transformée est choisie petite de façon à limiter le nombre des sous-bandes, sinon il serait difficile de constituer des séquences d'apprentissage de tailles suffisantes pour toutes.

6.2.1 Construction du dictionnaire

Le dictionnaire est conçu tel qu'un catalogue de représentants soit construit indépendamment pour chacune des sous-images. La figure 6.2 illustre alors les formes des vecteurs (qui sont choisies de façon à tirer profit de la corrélation inter-bloc), les étiquettes relatives aux sous-bandes et les débits maximaux. Ces derniers sont sélectionnés tels que les sous-bandes *a priori* plus importantes visuellement reçoivent plus de bits, mais aussi de manière à offrir une gamme suffisante de quantificateurs potentiels lors de l'opération d'allocation binaire.

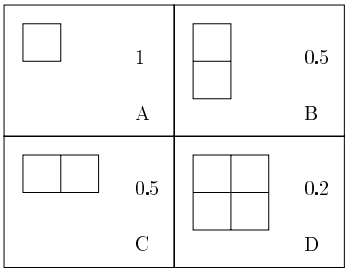


FIG. 6.2 – *Dictionnaire codeur type MPEG : forme des vecteurs et débits maximaux (en [bpp]), le débit global est de 0,55 bpp.*

Le tableau 6.1 rapporte les données obtenues pour la construction des dictionnaires à partir de la séquence Salesman, le facteur de projection F étant celui de l'équation 5.2. Nous remarquons :

- la rapidité de cette construction ;
- que les tailles des dictionnaires croissent très vite avec la dimension vectorielle, ce qui nécessite des séquences d'apprentissage imposantes. N'oublions pas qu'un dictionnaire est jugé représentatif de la source, si le rapport d'apprentissage est de l'ordre de 150 à 200 ;
- l'entropie moyenne associée au dictionnaire global final (*i.e.* 0,5 bpp) est inférieure à 0,55 bpp, car la construction de chaque sous-dictionnaire est stoppée de façon à ne pas dépasser le débit seuil.

Les courbes débit-distorsion relatives aux construction de ces dictionnaires sont représentées à la figure 6.3. Leurs formes sont caractéristiques avec l'alternance :

- de "sauts" importants entre deux points consécutifs de la courbe. Ils indiquent que la zone centrale dense en vecteurs à quantifier a été découpée ; la chute de distorsion et la hausse de débit sont conséquentes ;
- de "chapelets" de points rapprochés. Ils sont générés lorsque les découpages des Voronoï périphériques à celui central, sont ensuite effectués.

Ces courbes nous donnent directement la profondeur probable de l'arbre (*i.e.* pour simplifier, un "saut" supplémentaire implique un nouvel étage de l'arbre de découpage). Chaque courbe donne aussi l'EQM initiale, soit l'énergie de la sous-bande quantifiée (*i.e.* la distorsion pour le débit nul). Ces fonctions traduisent également l'état de la répartition des sources quantifiées ; à titre d'exemple, le premier chapelet de points espacés pour la sous-bande C indique, par rapport à la sous-bande B, des vecteurs source plus éparses autour de l'amas central (voir aussi les images sur la première ligne de la figure 6.14). Nous pouvons ajouter que l'allocation binaire est *a priori* plus intéressante, si les quantificateurs potentiels choisis sur les courbes débit-distorsion, se situent à l'issue d'un chapelet de points (*i.e.* juste avant un nouveau saut). Ainsi les dictionnaires résultants sont plus symétriques (*i.e.* les découpages des Voronoï autour de celui central ont été réalisés).
Si nous fixons le débit pour l'allocation binaire à 0.2 bpp, nous obtenons les nouveaux chiffres du tableau 6.2. La sous-bande C a reçu plus de représentants que celle B car cette source est plus dispersée.

6.2.2 Encodage de séquences

Encodage de Salesman

Nous codons les 200 premières images de la séquence Salesman avec le dictionnaire (pour l'allocation binaire fixée à 0,2 bpp). Les graphiques de la figure 6.4 présentent

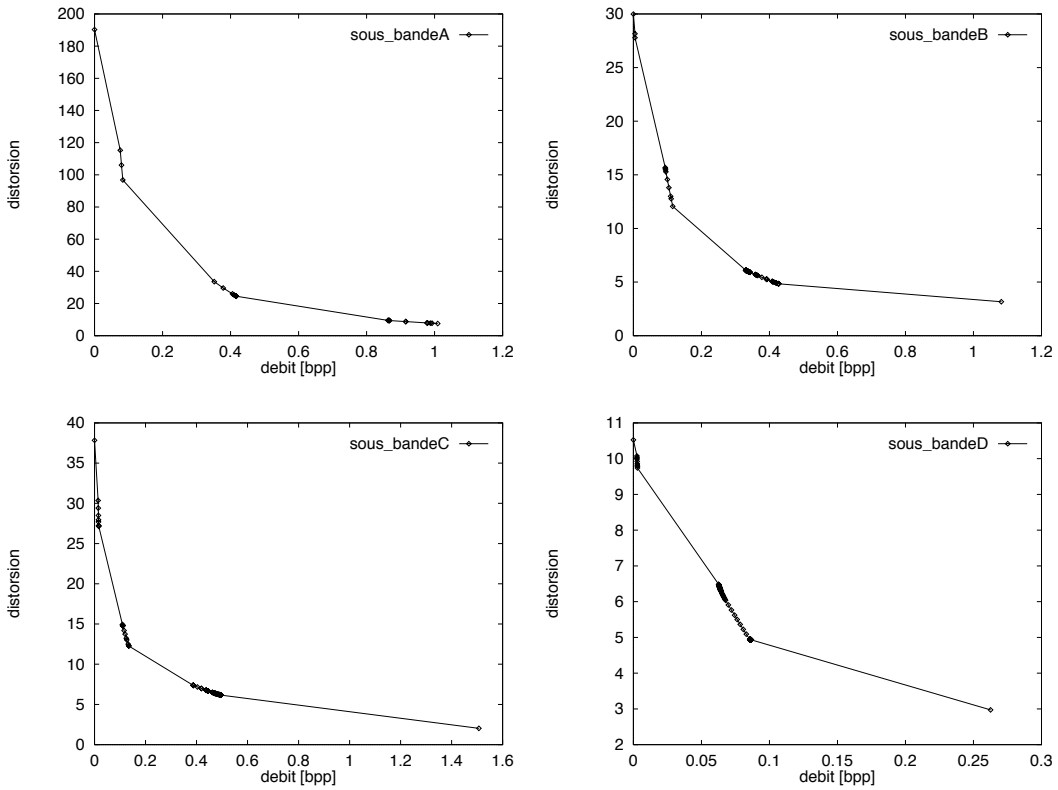


FIG. 6.3 – *Dictionnaire codeur type MPEG: courbes entropie/distorsion relatives à la construction des dictionnaires.*

étiquette sous-bande	construction dictionnaire				
	taille séquence apprentissage	temps cpu [s]	nombre de représentants	entropie [bpp]	rapport apprentissage
A	5 images	6.75	43	0.992	884
B	10 images	13.95	358	0.427	186
C	10 images	14.13	434	0.496	153
D	148 images	48.50	1248	0.087	189

TAB. 6.1 – *Codeur type MPEG, construction du dictionnaire avec Salesman: le débit moyen final est de 0,5 bpp*

étiquette sous-bande	construction dictionnaire	
	nombre de représentants	entropie [bpp]
A	19	0.416
B	64	0.111
C	108	0.134
D	1234	0.086

TAB. 6.2 – *Codeur type MPEG, allocation binaire à 0.2 bpp pour le dictionnaire construit avec Salesman: le débit moyen final est de 0,188 bpp*

respectivement l'évolution du débit et de la distorsion au cours du codage, les résultats numériques sont au tableau 6.3. Nous notons que :

- l'encodage est rapide (environ 2 s/image) ;
- l'entropie a augmenté par rapport à celle réclamée pour le dictionnaire, mais elle demeure dans la limite des 0,2 bpp ;
- les pics d'entropie correspondent logiquement aux pics de G_q, soient aux images de la séquence plus riches en mouvement (*i.e.* il y a plus d'information à quantifier). La sous-bande A est alors celle dont le débit varie le plus, entraînant les fluctuations du débit de l'image. Pour les autres sous-bandes, les entropies demeurent relativement stables ;
- après une période de décroissance correspondant à une stabilisation de la boucle fermée (BF) de codage, le PSNR devient régulier. La qualité de reconstruction est très satisfaisante et de qualité constante, car lorsque la prédiction est moins efficace (*i.e.* chute de G_p), le QVAA compense (*i.e.* hausse de G_q) ;
- les vecteurs source hors-norme sont peu nombreux et appartiennent aux sous-bandes dont les distributions sont déjà plus éparses en BO.

La figure 6.5 représente des images décodées, les images correspondantes des erreurs de prédiction avant (*i.e.* les e) et après (*i.e.* les e_q) quantification, et celles des erreurs de quantification. Ces images soulignent la façon dont opère le QVAA : les zones perturbées par le mouvement sont finement quantifiées, celles bruitées (*e.g.* le fond) sont quantifiées sommairement. Il n'existe pas d'erreurs de quantification marquées sur les contours du personnage.

étiquette sous-bande	encodage séquence	
	nombre de vecteurs hors-norme	entropie [bpp]
A	14	0.314
B	0	0.156
C	1	0.184
D	0	0.151

TAB. 6.3 – *Codeur type MPEG, allocation à 0,2 bpp, encodage de Salesman : le débit moyen final est de 0,201 bpp pour un PSNR de 39,07 dB*

Encodage de MissAmerica et de Claire

Nous utilisons le dictionnaire (avec la même allocation binaire à 0,2 bpp) pour coder la séquence MissAmerica et les 250 premières images de la séquence Claire. Les résultats numériques correspondants sont les tableaux 6.4 et 6.5, ainsi que les graphiques des figures 6.6 et 6.8. Des images extraites des séquences sont affichées aux figures 6.7 et 6.9.

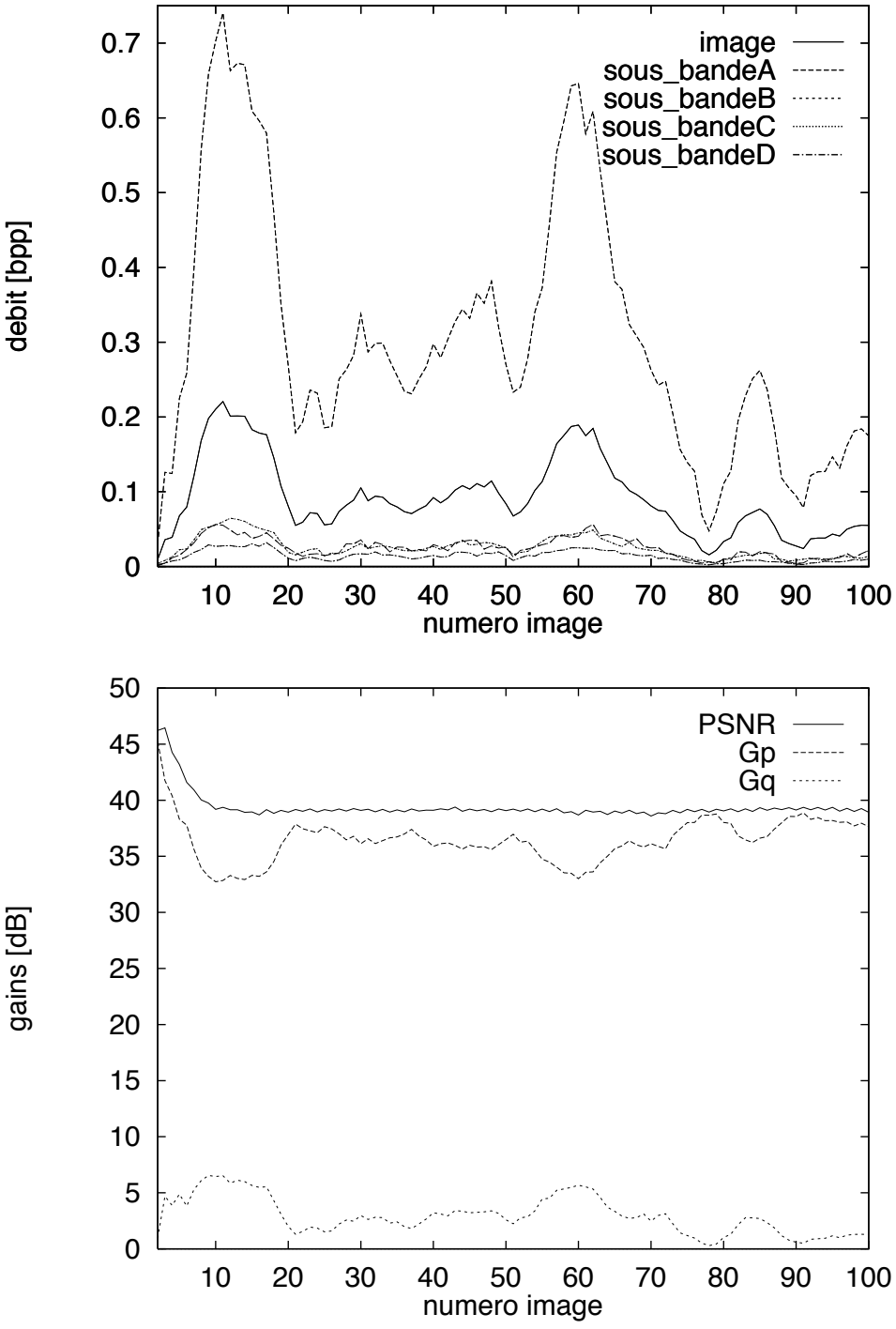


FIG. 6.4 – *Codeur type MPEG, allocation à 0,2 bpp, encodage de Salesman : entropies et gains pour les images 2 à 100.*

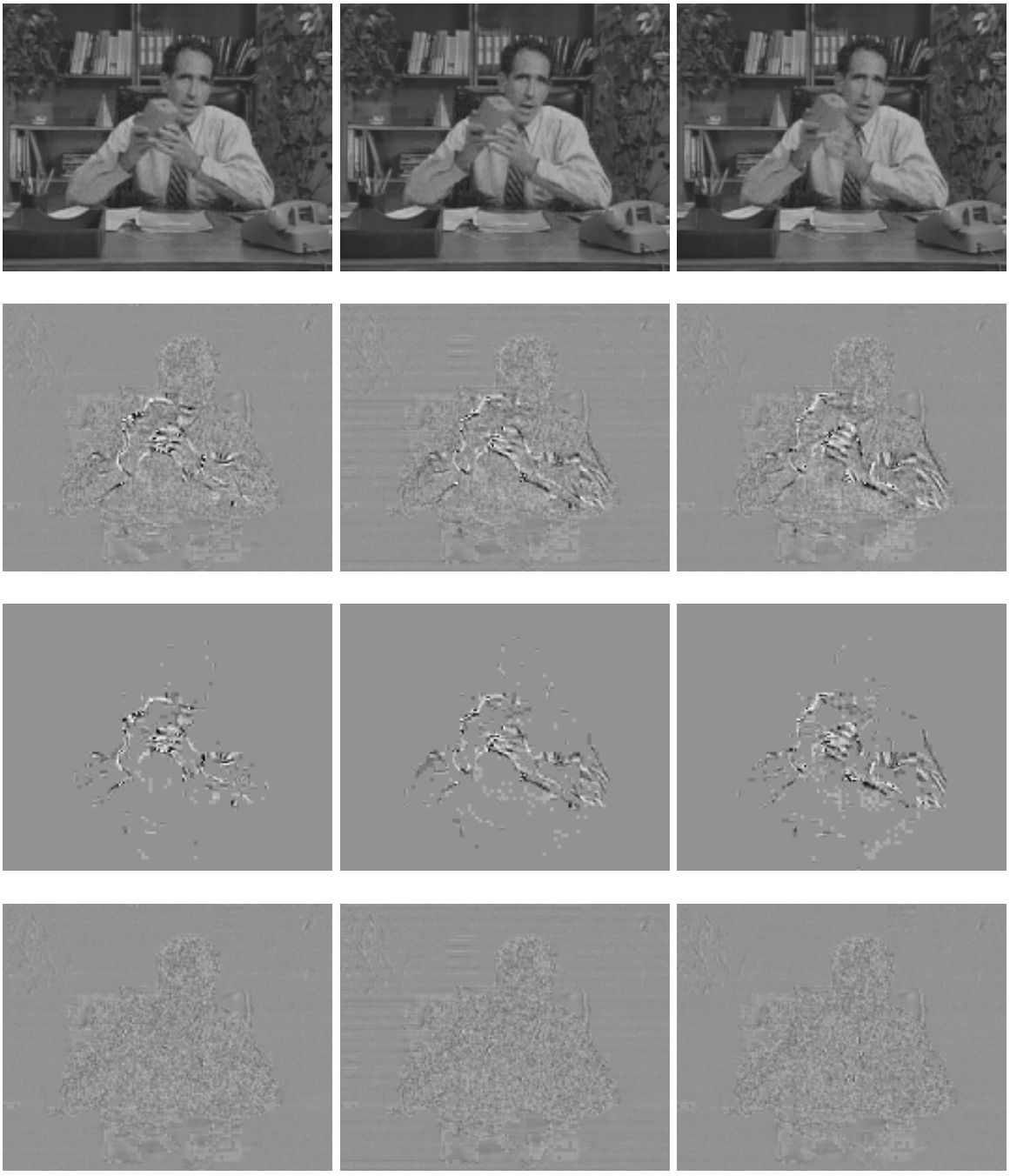


FIG. 6.5 – *Codeur type MPEG, allocation à 0,2 bpp, encodage de Salesman, images extraites des séquences: avec de haut en bas, les images décodées, les erreurs de prédictions avant et après quantification, les erreurs de quantification (les images d'erreurs sont centrées et amplifiées par 3).*

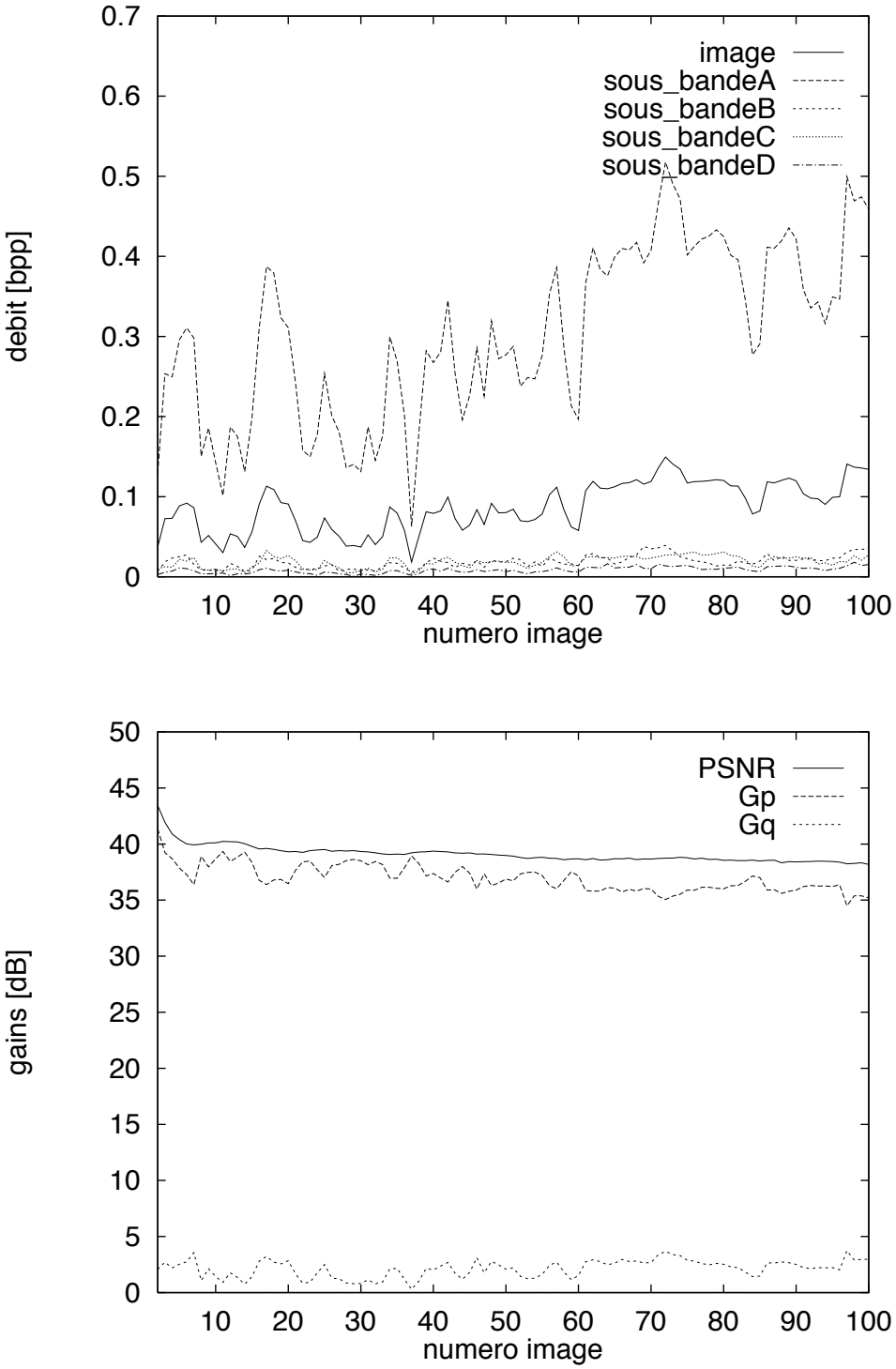


FIG. 6.6 – *Codeur type MPEG, allocation à 0,2 bpp, encodage de MissAmerica : entropies et gains pour les images 2 à 100.*

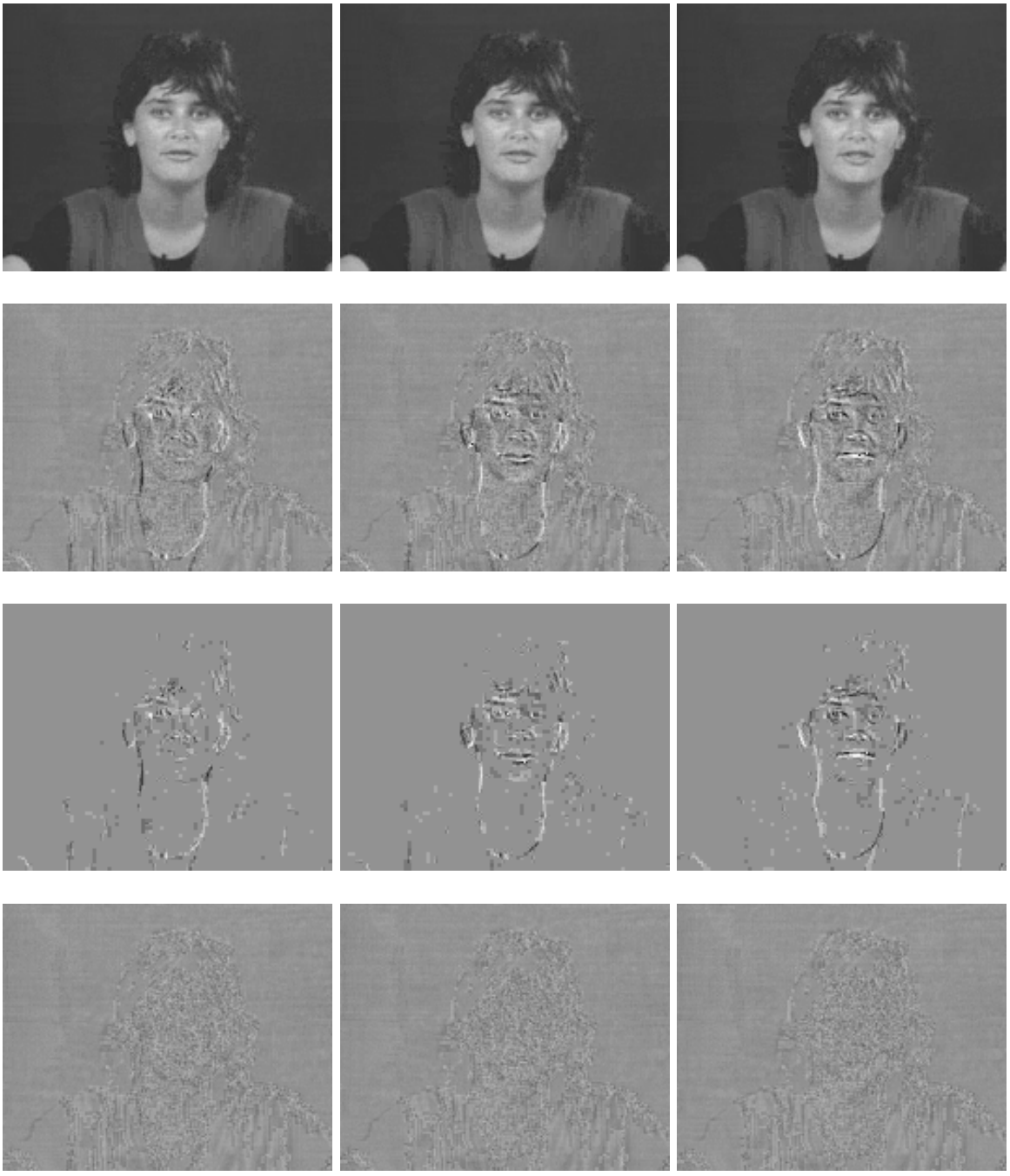


FIG. 6.7 – *Codeur type MPEG, allocation à 0,2 bpp, encodage de MissAmerica, images extraites des séquences: avec de haut en bas, les images décodées, les erreurs de prédictions avant et après quantification, les erreurs de quantification (les images d'erreurs sont centrées et amplifiées par 3).*

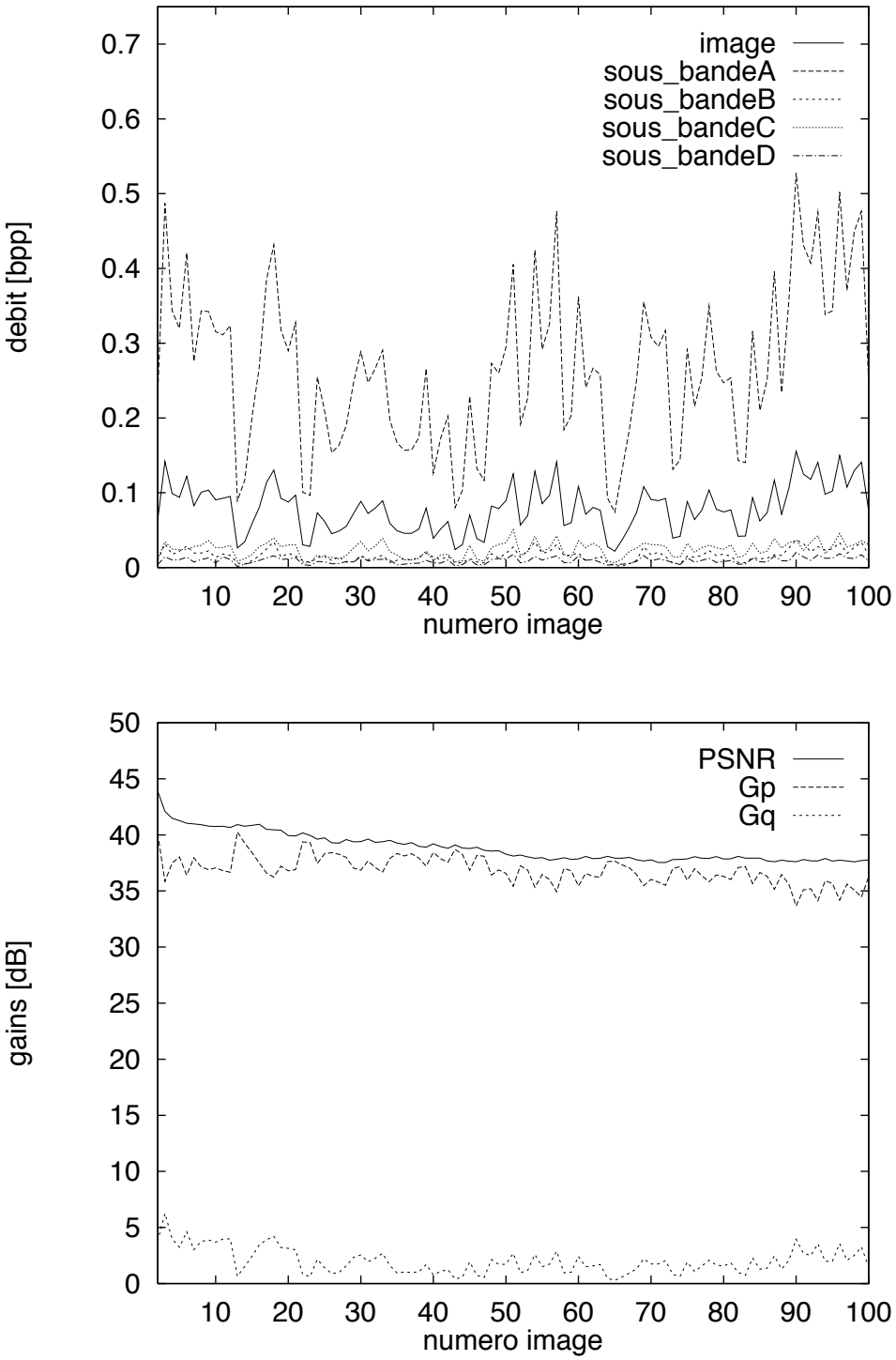


FIG. 6.8 – *Codeur type MPEG, allocation à 0,2 bpp, encodage de Claire : entropies et gains pour les images 2 à 100.*

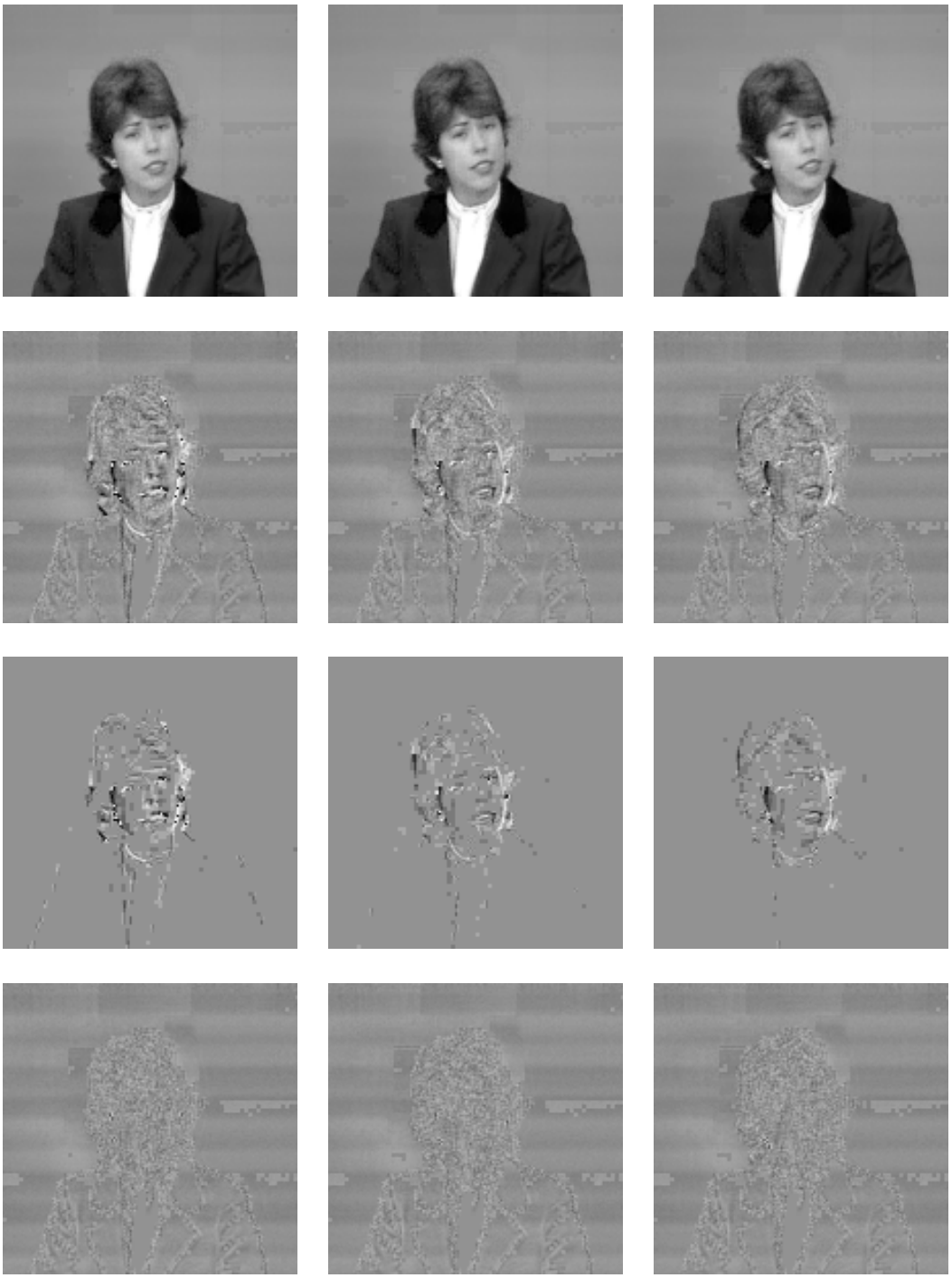


FIG. 6.9 – *Codeur type MPEG, allocation à 0,2 bpp, encodage de Claire, images extraites des séquences : avec de haut en bas, les images décodées, les erreurs de prédictions avant et après quantification, les erreurs de quantification (les images d'erreurs sont centrées et amplifiées par 3).*

étiquette sous-bande	encodage séquence	
	nombre de vecteurs hors-norme	entropie [bpp]
A	8	0.320
B	0	0.106
C	2	0.130
D	0	0.092

TAB. 6.4 – *Codeur type MPEG, allocation à 0,2 bpp, encodage de MissAmerica: le débit moyen final est de 0,162 bpp pour un PSNR de 38.98 dB*

étiquette sous-bande	encodage séquence	
	nombre de vecteurs hors-norme	entropie [bpp]
A	0	0.283
B	0	0.105
C	1	0.138
D	0	0.102

TAB. 6.5 – *Codeur type MPEG, allocation à 0,2 bpp, encodage de Claire: le débit moyen final est de 0,157 bpp pour un PSNR de 38.03 dB*

L'analyse de ces données relatives à l'encodage de ces deux nouvelles séquences viennent confirmer les résultats obtenus avec Salesman. Cependant MissAmerica et Claire sont moins riches en mouvement (*i.e.* il n'y a plus de pics prononcés de G_q), c'est pourquoi le débit requis a chuté et la qualité de restitution s'est encore améliorée.
Les tendances affichées par les courbes semblent indiquer une augmentation progressive du débit et une baisse du PSNR. Nous proposons donc d'examiner le comportement du codage pour de plus longues séquences.

Encodage de longues séquences

Nous codons alors 450 images de la séquence Salesman, et 450 images de la séquence Claire. Les résultats numériques correspondants sont les tableaux 6.6 et 6.7, ainsi que les graphiques des figures 6.10 et 6.11.
Ces données indiquent qu'il n'y a pas eu dérive au cours du temps des résultats obtenus sur les séquences plus courtes. Le système fonctionnant en boucle fermée, le nombre de vecteurs hors-norme a augmenté et concernent toujours les mêmes sous-bandes plus éparses. Les courbes soulignent toujours l'irrégularité du débit de la sous-bande A, perturbatrice du débit global de l'image.
L'ensemble de ces premiers résultats indiquent que la QVAA s'adapte correctement à la structure du codeur hybride de type MPEG, et qu'il est possible de construire un dictionnaire valide pour un type donné de séquences d'images.

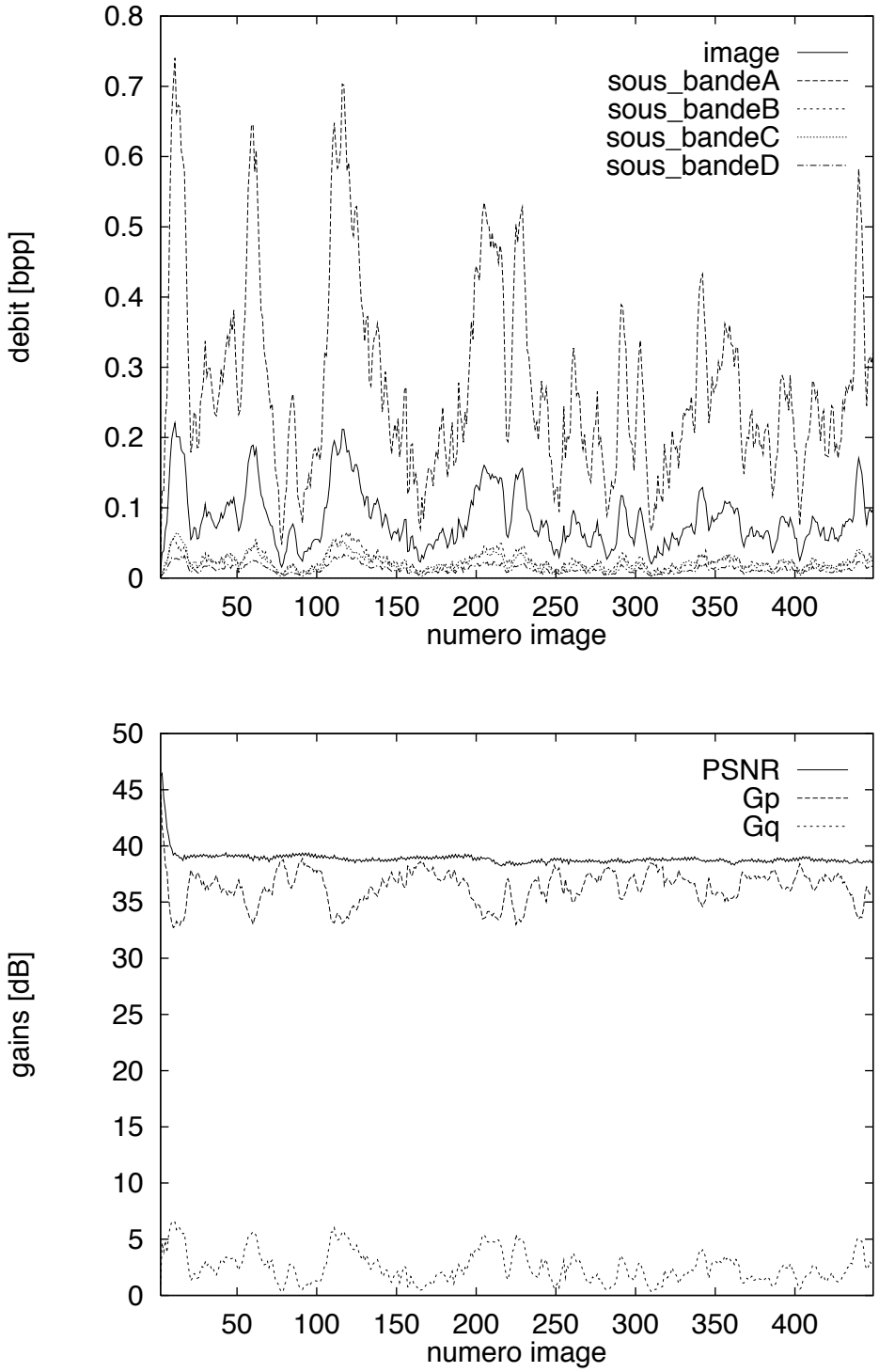


FIG. 6.10 – *Codeur type MPEG, allocation à 0,2 bpp, encodage de 450 images de Salesman : entropies et gains des images.*

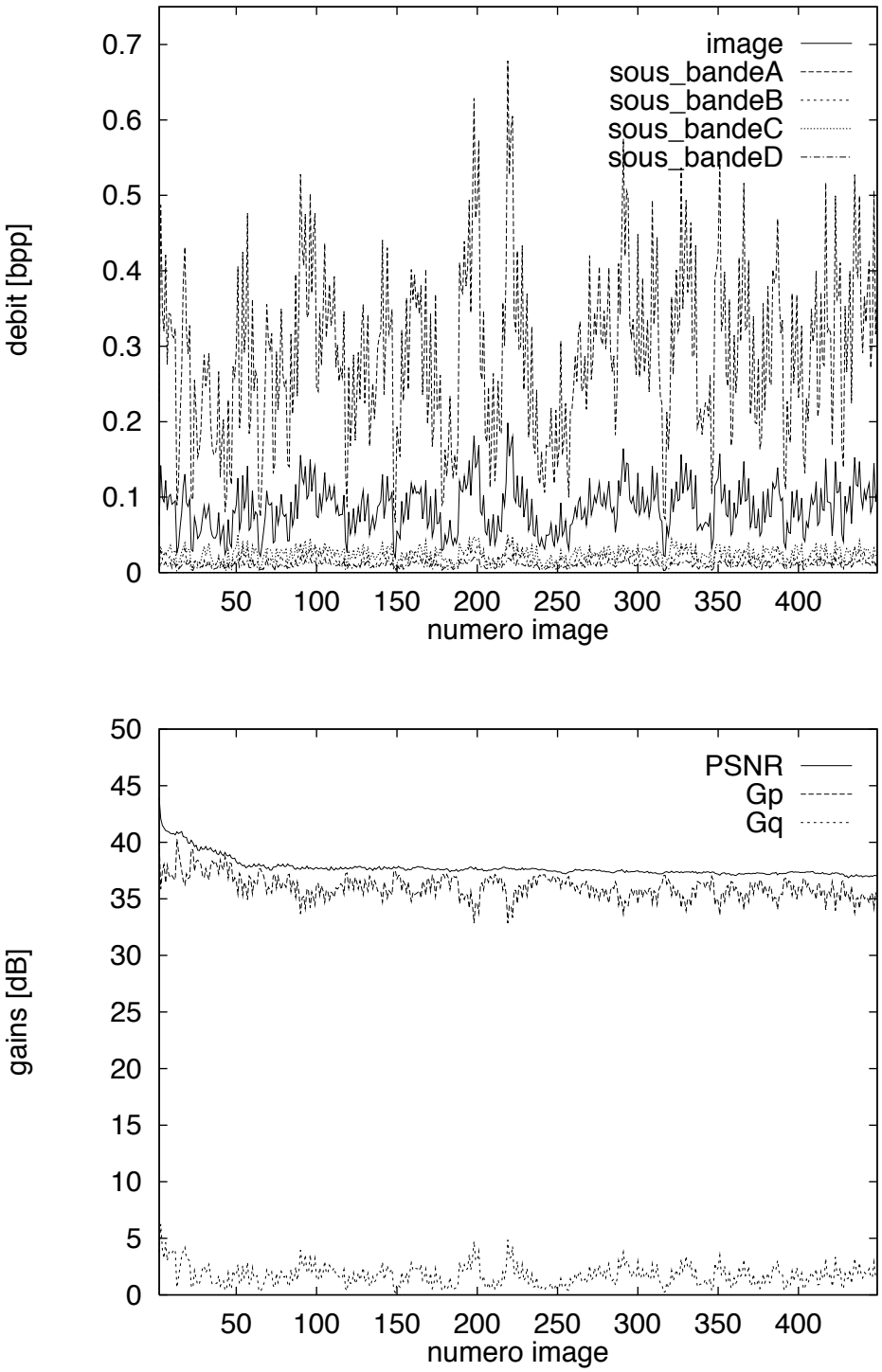


FIG. 6.11 – *Codeur type MPEG, allocation à 0,2 bpp, encodage de 450 images de Claire : entropies et gains des images.*

étiquette sous-bande	encodage séquence	
	nombre de vecteurs hors-norme	entropie [bpp]
A	60	0.287
B	1	0.135
C	1	0.164
D	0	0.144

TAB. 6.6 – *Codeur type MPEG, allocation à 0,2 bpp, encodage de 450 images de Salesman : le débit moyen final est de 0,183 bpp pour un PSNR de 38.84 dB*

étiquette sous-bande	encodage séquence	
	nombre de vecteurs hors-norme	entropie [bpp]
A	2	0.297
B	0	0.110
C	3	0.142
D	0	0.106

TAB. 6.7 – *Codeur type MPEG, allocation à 0,2 bpp, encodage de 450 images de Claire : le débit moyen final est de 0,164 bpp pour un PSNR de 37,69 dB*

6.2.3 Accroissement de la dimension vectorielle

Construction du dictionnaire

Nous voulons juger de l'avantage de mettre en oeuvre des dimensions vectorielles supérieures. Le dictionnaire construit est alors celui de la figure 6.12. Les résultats numériques au tableau 6.8 soulignent combien le nombre de représentants s'amplifie avec la dimension vectorielle, entraînant des difficultés pour réunir des séquences d'apprentissage de tailles suffisantes, et un accroissement notable des temps de calcul. L'ordre précédent pour la prépondérance des sous-bandes est conservé ; en particulier la sous-bande C plus énergétique, a ses points plus dispersés et requiert plus de représentants, de bits et de temps que celle B.

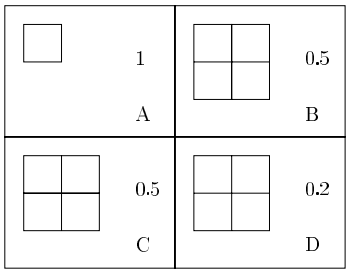


FIG. 6.12 – *Dictionnaire codeur type MPEG : forme des vecteurs et débits maximaux (en [bpp]), le débit global est de 0,55 bpp.*

Le seuil pour l'allocation est fixé à 0,15 bpp soit une valeur inférieure à celle précédente (voir aussi le tableau 6.9). Les rapports d'apprentissage deviennent correctes et indiquent

étiquette sous-bande	construction dictionnaire				
	taille séquence apprentissage	temps cpu [s]	nombre de représentants	entropie [bpp]	rapport apprentissage
A	5 images	6.75	43	0.992	884
B	447 images	212.89	9846	0.256	72
C	447 images	231.37	10132	0.261	70
D	148 images	48.50	1248	0.087	189

TAB. 6.8 – *Codeur type MPEG, dimensions vectorielles accrues, construction du dictionnaire avec Salesman : le débit moyen final est de 0,399 bpp*

étiquette sous-bande	construction dictionnaire		
	nombre de représentants	entropie [bpp]	rapport d'apprentissage
A	19	0.416	2236
B	2097	0.062	338
C	2546	0.068	278
D	62	0.003	3806

TAB. 6.9 – *Codeur type MPEG, dimensions vectorielles accrues, allocation binaire à 0.15 bpp pour le dictionnaire construit avec Salesman : le débit moyen final est de 0,137 bpp*

une bonne représentativité des dictionnaires. Les nombres des représentants des sous-bandes B et C demeurent élevés.

Encodage de séquences

Nous quantifions les séquences précédentes (*i.e.* celles courtes), nous obtenons les résultats du tableaux 6.10, alors :

- pour ces entropies inférieures, les PSNR demeurent élevés (pour une allocation à 0,2 bpp, ils seraient supérieurs à ceux précédents, voir alors le tableau 6.11) ;
- le temps d'encodage a chuté (environ 1 s/image) car il y a moins de vecteurs à quantifier par image.

encodage		
séquence	entropie [bpp]	PSNR [dB]
Salesman	0.154	37.07
MissAmerica	0.131	38.13
Claire	0.124	36.92

TAB. 6.10 – *Codeur type MPEG, dimensions vectorielles accrues, allocation à 0,15 bpp : encodage des séquences.*

L'augmentation des dimensions vectorielles conduit donc logiquement à une amélioration des performances de codage (qui étaient déjà satisfaisantes). Cependant les temps de construction des dictionnaires, et surtout les tailles des séquences d'apprentissage requises deviennent pénalisants.

encodage		
séquence	entropie [bpp]	PSNR [dB]
Salesman	0.244	39.60
MissAmerica	0.233	39.66
Claire	0.197	38.50

TAB. 6.11 – *Codeur type MPEG, dimensions vectorielles accrues, allocation à 0,2 bpp: encodage des séquences.*

6.2.4 Comparaison BO/BF

Le dictionnaire utilisé est celui défini à la figure 6.2. Les courbes de la figure 6.13 illustrent l'écart existant entre la fonction entropie/PSNR obtenue lors de la construction du dictionnaire (*i.e.* en boucle ouverte, BO), et celles obtenues lors de l'encodage des séquences (*i.e.* en boucle fermée, BF). Nous remarquons que les courbes générées en BF sont plus favorables que celles en BO (*i.e.* le PSNR est plus élevé). Afin d'expliquer ce résultat, nous visualisons à la figure 6.14 les répartitions des sources ainsi que le découpage de l'espace pour les sous-bandes B et C des différentes séquences.
Il faut remarquer la différence entre les répartitions statistiques des points de la source en BO, et celles des points de la source en BF. Dans ce dernier cas les erreurs de quantification générées sont réinjectées à l'entrée du codeur ; il en résulte une plus grande amplitude des erreurs de prédiction, et une plus large répartition des points. Ainsi un grand nombre de points à quantifier se situe dans les Voronoï de la couronne autour de la "dead zone" (voir le chapitre 5). Les conséquences sont les suivantes :

- l'entropie augmente car le coût de codage des représentants de cette couronne croît ;
- la distorsion diminue car les vecteurs source hors de la "dead zone" sont mieux quantifiés.

Cette analyse souligne la difficulté inhérente à la constitution de séquences d'apprentissage pour ce codeur fonctionnant en boucle fermée, et il n'existe pas *a priori* de solution simple face à ce problème.
Une seconde remarque s'impose concernant la taille de la "dead zone" autour du point 0 (pour en faire varier les dimensions, il suffit d'intervenir sur le facteur F de projection de la figure 5.9). La détermination de celle-ci intervient dans un contexte de codage psychovisuel où il est choisi de quantifier grossièrement l'information *a priori* invisible pour le SVH. La seule façon pour la fixer serait subjective, en effectuant une série de tests sur un banc d'essai. Mais ceci n'entre pas dans le cadre de notre étude. Nous nous sommes contentés de fixer ce paramètre F en fonction de l'équation 5.2. Les PSNR obtenus au-delà de 0,15 bpp indiquent néanmoins une bonne qualité de restitution des images.

6.2.5 Comparaison à un QS de type MPEG

Le QS de type MPEG, que nous mettons en oeuvre pour la quantification des images prédites, est décrit à l'annexe F. Nous rappelons que ce quantificateur introduit également

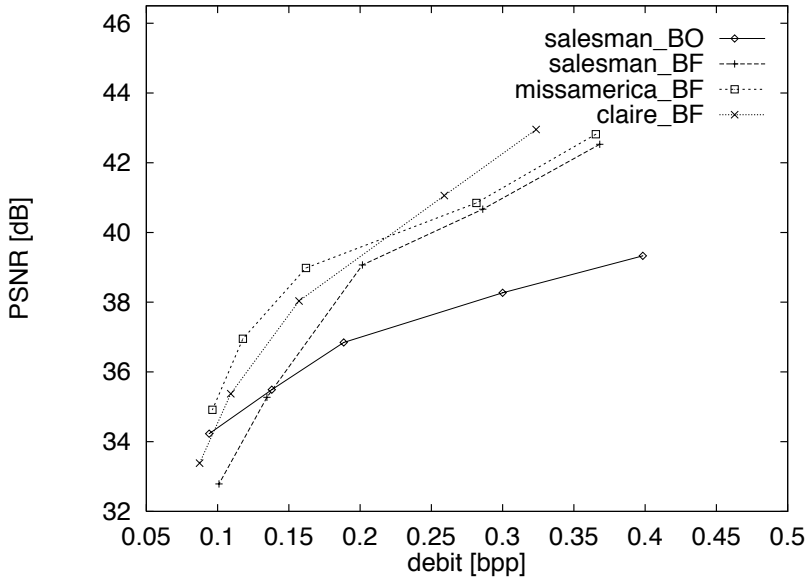


FIG. 6.13 – *Codeur type MPEG, comparaison BO/BF.*

une "dead zone".
Nous traçons à la figure 6.15, les courbes entropies/PSNR obtenues lors du codage des différentes séquences (*i.e.* celles courtes), et pour le dictionnaire défini à la figure 6.2. De façon générale les courbes sont très proches et n'indiquent pas une supériorité manifeste de notre QV. Il faut cependant remarquer que notre schéma de codage effectue une répartition spectrale du bruit de quantification, à laquelle il faut ajouter la formation de cette région vide de représentants autour du 0. La bonne façon de juger les résultats (pour les bas débits), est subjective avec la visualisation des séquences décodées. Enfin notons que les systèmes fonctionnant en boucle fermée, si les séquences à l'entrée des deux codeurs (*i.e.* celui doté de QVAA, et l'autre du QS type MPEG) sont identiques, les sources vectorielles et scalaires à quantifier sont différentes, la comparaison brute de ces courbes devient délicates.

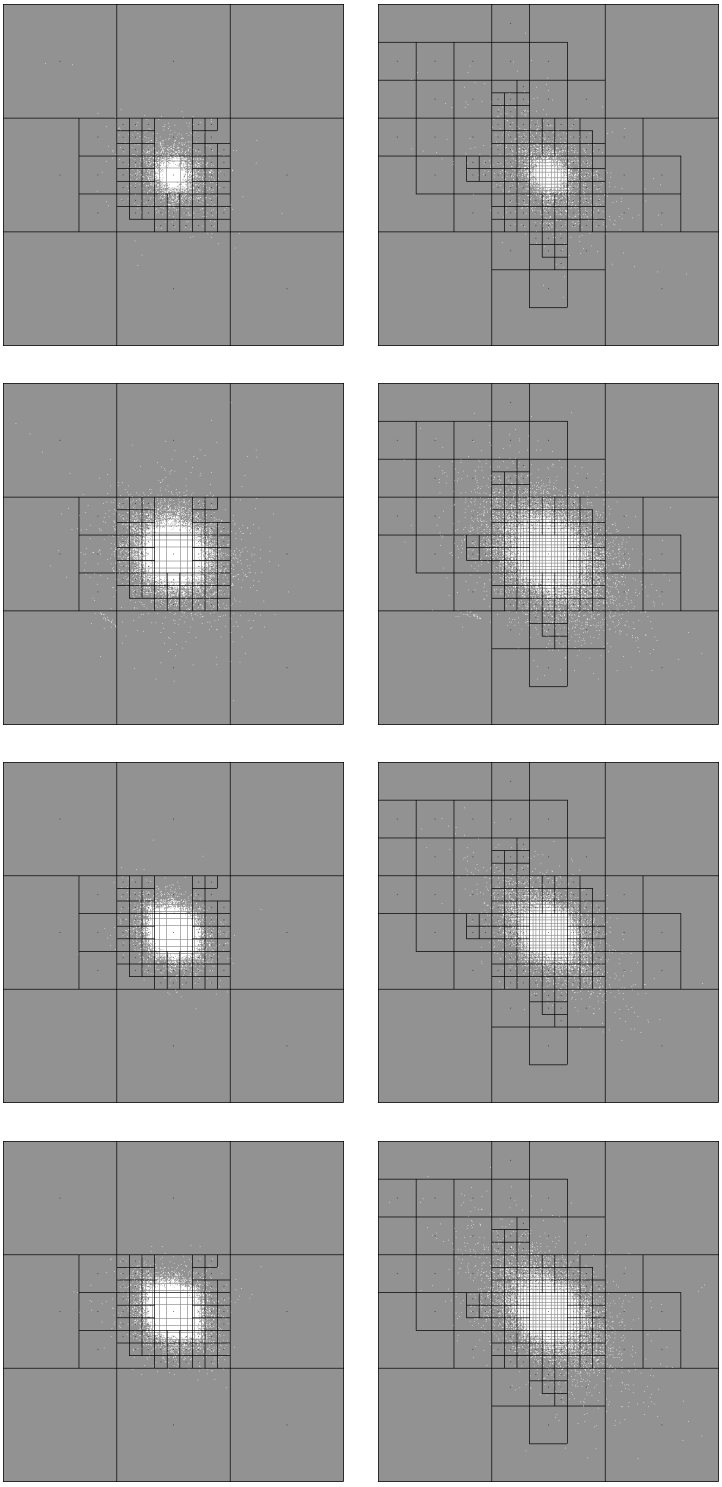


FIG. 6.14 – *Codeur type MPEG, comparaison BO/BF : visualisation des sous-bandes B (images de la première colonne) et C (celles de la seconde colonne), pour (de haut en bas) Salesman en BO, Salesman en BF, MissAmerica en BF et Claire en BF. Les vecteurs source sont les points blancs, les Voronoï et les représentants utilisés sont en noir.*

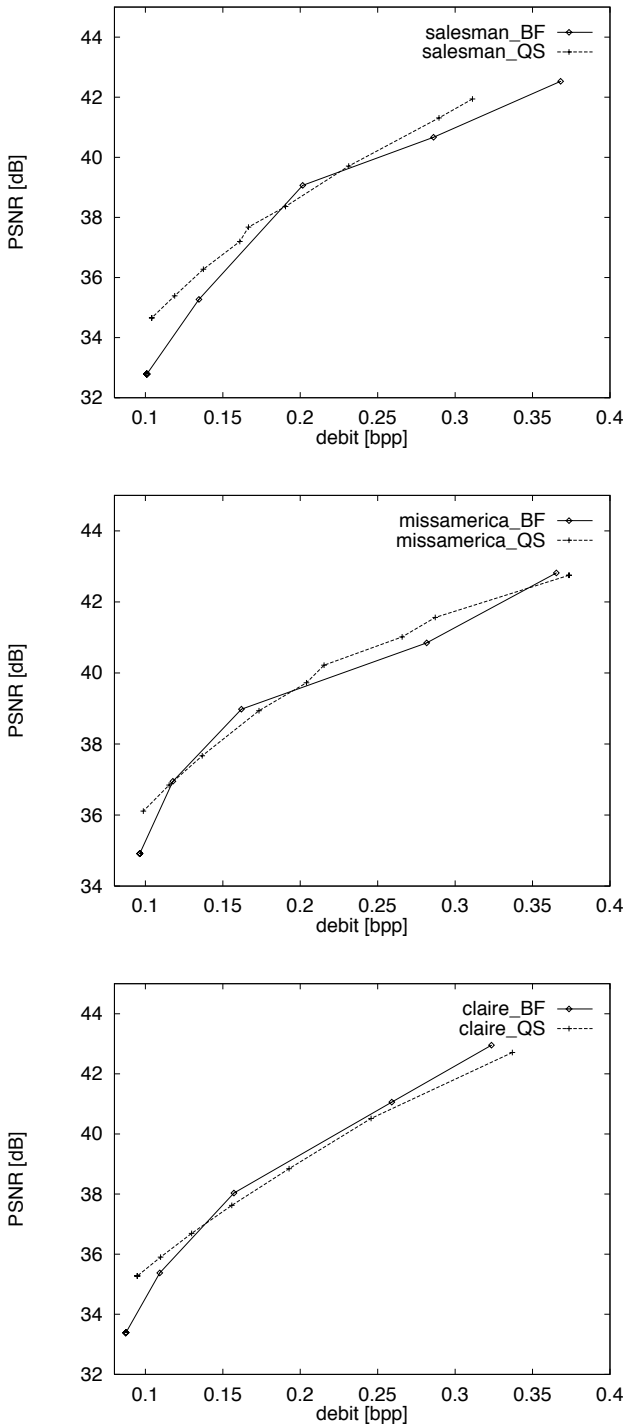


FIG. 6.15 – *Codeur type MPEG, comparaison au QS type MPEG, courbes obtenues en BF pour les différentes séquences.*

6.3 Codage basé régions de séquences d'images

6.3.1 Construction du dictionnaire

Le QVAA est cette fois inscrit au sein d'un nouveau schéma de codage basé régions, et orienté vers la compression bas débits de scènes visiophoniques. L'unité de base de codage n'est donc plus un bloc rectangulaire, mais une région de forme arbitraire résultant de la projection d'un objet 3D sur le plan image [Li et al.94] [Tziritas et al.94] [GG95]. Nous décrivons brièvement les modules du codeur de la figure 5.1, qui sont tels que :

- l'estimation du mouvement est réalisée en deux étapes :
  - une première estimation du mouvement est obtenue par une technique multirésolution [Nzomigni95] effectuant une segmentation en régions homogènes des images au sens du mouvement ;
  - les informations résultantes de mouvement (*i.e.* les cartes de segmentation, et les paramètres du modèle de mouvement affine associés à chaque région) sont ensuite traitées par un algorithme de minimisation basé sur un critère MDL "Minimum Description Length" [Pateux et al.96]. Ce processus est mis en oeuvre afin de réduire le coût de codage des segments, les régions résultantes ont alors des formes polygonales (voir la figure 6.17). A titre d'exemple, le coût moyen de codage est seulement de 400 bits par carte de segmentation (pour 800 points de contrôle) avec MissAmerica, et de 375 bits (700 points de contrôle) pour celle de Salesman ;
- la transformée appliquée aux images d'erreur de prédiction est une décomposition dyadique en ondelettes. Précisément nous réalisons une décomposition multirésolution sur deux niveaux, en utilisant les ondelettes orthonormales et à support compact de Daubechies [Daubechies88] ;
- le dictionnaire multirésolution [Antonini91] construit est celui de la figure 6.16. Le facteur d'échelle F demeure celui de l'équation 5.2.

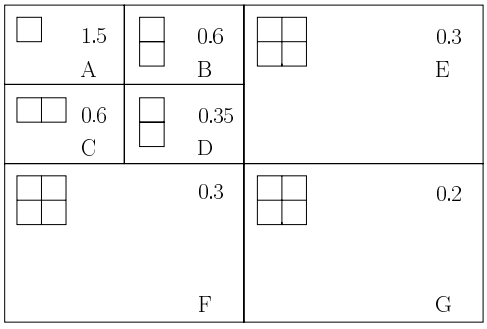


FIG. 6.16 – *Codeur basé régions, dictionnaire : forme des vecteurs et débits maximaux (en [bpp]), le débit global est de 0,391 bpp.*

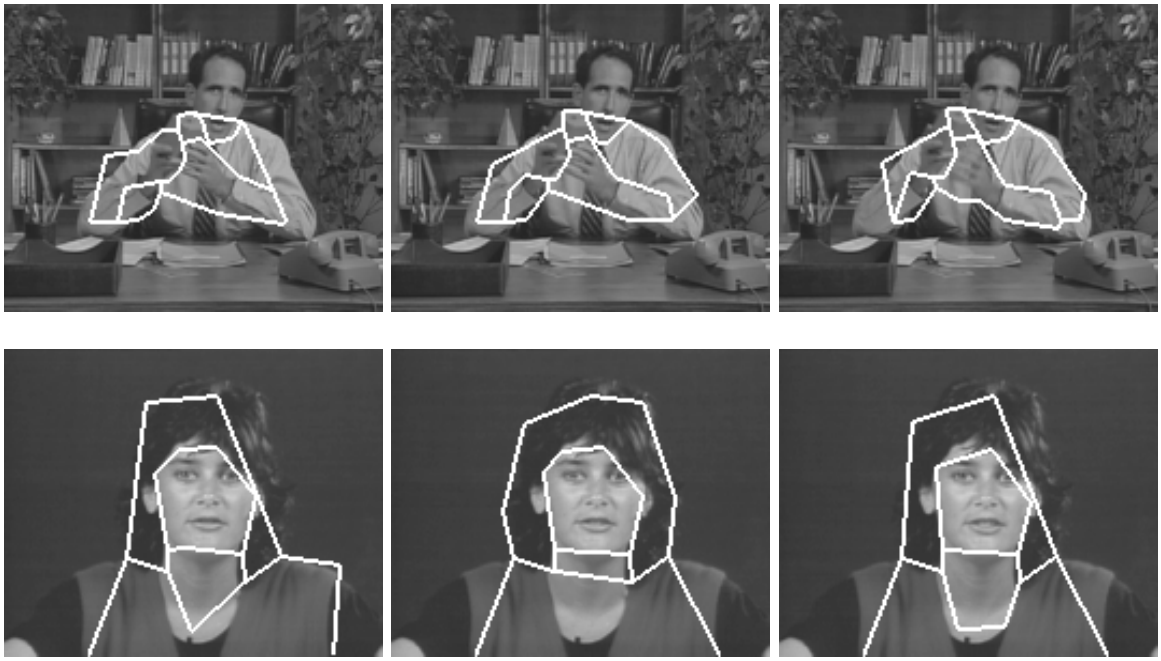


FIG. 6.17 – *Codeur basé régions : images segmentées extraites des séquences.*

Les dictionnaires sont construits à partir des images de la séquence Salesman. Le seuil pour l'allocation binaire est fixé à 0,2 bpp. Les tableaux 6.12 et 6.13 rapportent les résultats numériques. Ces chiffres, de façon globale, appellent les mêmes remarques que celles faites lors de la construction des dictionnaires du codeur de type MPEG, nous soulignons que :

- malgré les tailles imposantes des séquences d'apprentissage requises dès que les dimensions vectorielles augmentent, les temps de construction demeurent raisonnables ;

- la taille mémoire pour stocker les dictionnaires est modérée (les tailles cumulées des dictionnaires représentent 1,6 fois celle d'une image).

Une explication s'impose néanmoins concernant les dictionnaires des sous-bandes B et C. Lors de la construction du dictionnaire global, les sous-bandes dont les sources sont plus dispersées ont logiquement reçu plus de représentants (*e.g.* la sous-bande B par rapport à celle C, ou la sous-bande F par rapport à celle F). Cependant après allocation binaire, la sous-bande B obtient finalement moins de code-vecteurs que celle C (voir aussi les images des espaces source et de leur découpage à la figure 6.19). Les courbes débit/distorsion (voir la figure 6.18) traduisent cette dissemblance entre les distributions de ces sources. La construction du dictionnaire de la sous-bande C est stoppée au sein d'un chapelet de points, ces derniers étant relatifs à un niveau supplémentaire de découpage de l'arbre (par rapport au dictionnaire de la sous-bande B). Finalement le débit alloué étant bas, le découpage coûteux de la sous-bande B ne pourra qu'être grossier.

étiquette sous-bande	construction dictionnaire					
	taille séquence apprentissage	temps cpu [s]	nombre de représentants	entropie [bpp]	rapport apprentissage	taille dictionnaire [octets]
A	5 images	2.27	39	1.476	244	458
B	20 images	3.18	110	0.353	151	964
C	20 images	4,18	76	0.593	219	648
D	20 images	3.08	49	0.336	339	438
E	150 images	82.12	2202	0.218	143	22113
F	150 images	57.22	1500	0.110	157	14877
G	100 images	23.46	145	0.011	1081	1368

TAB. 6.12 – *Codeur basé régions, construction du dictionnaire avec Salesman: le débit moyen final est de 0,257 bpp*

étiquette sous-bande	construction dictionnaire	
	nombre de représentants	entropie [bpp]
A	17	0.532
B	21	0.045
C	42	0.118
D	20	0.033
E	2185	0.218
F	1492	0.110
G	145	0.011

TAB. 6.13 – *Codeur basé régions, allocation binaire à 0,2 bpp pour le dictionnaire construit avec Salesman: le débit moyen final est de 0,175 bpp*

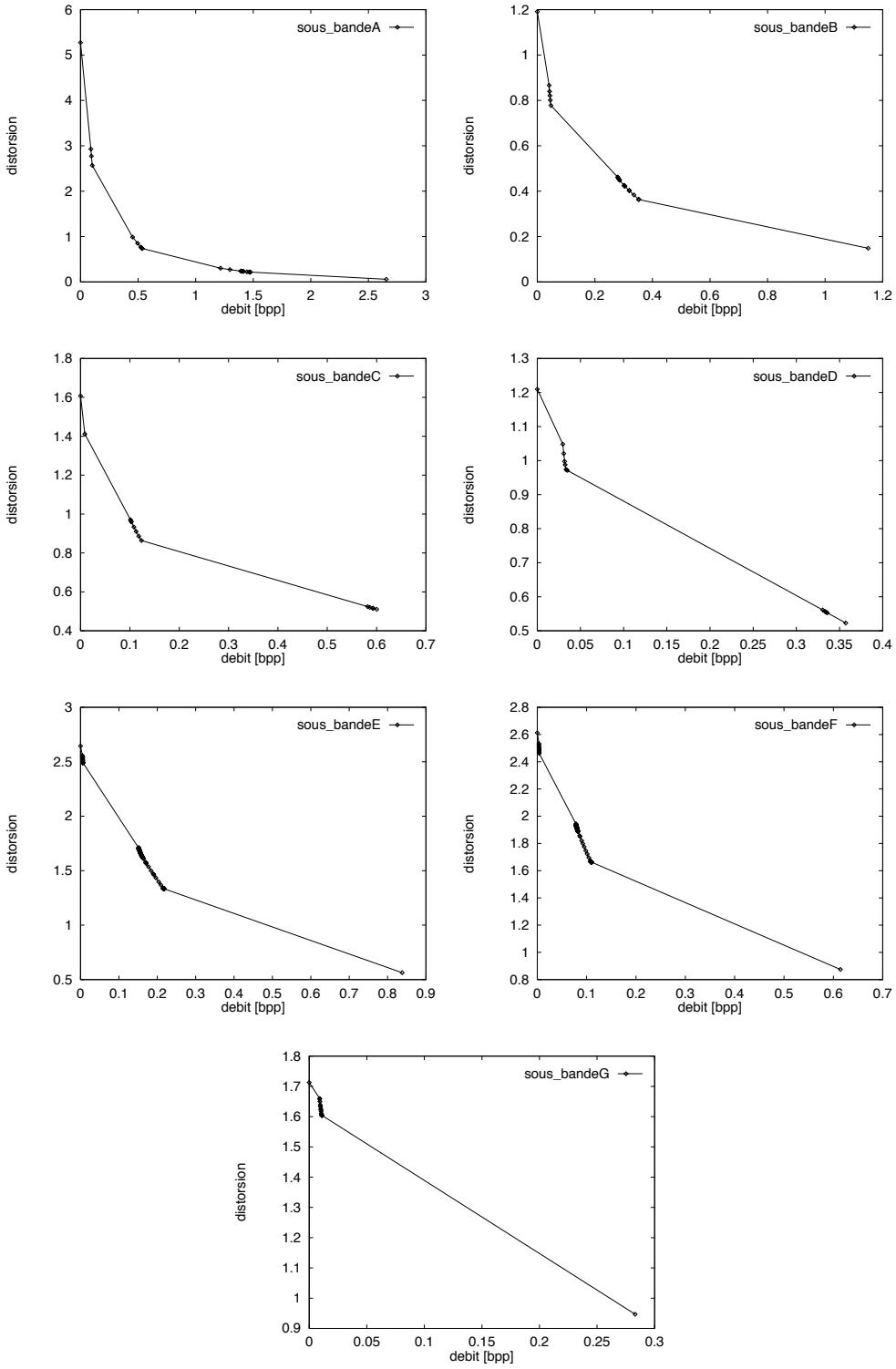


FIG. 6.18 – *Dictionnaire codeur basé régions: courbes entropie/distorsion relatives à la construction des dictionnaires.*

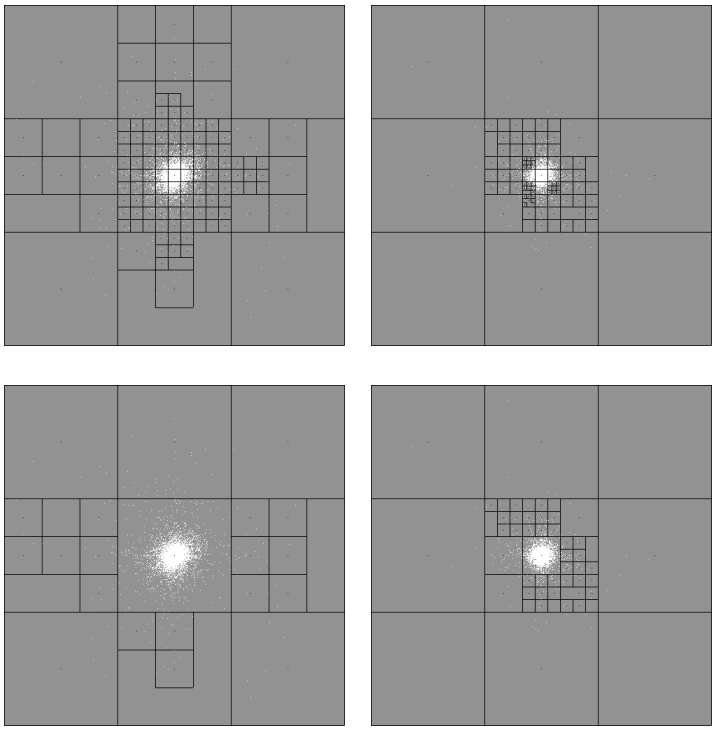


FIG. 6.19 – *Codeur basé régions : visualisation des sources des sous-bandes B (images de la première colonne) et C (images de la seconde colonne), et de leurs dictionnaires avant puis après allocation binaire.*

6.3.2 Encodage de Salesman

Nous codons les 200 premières images de la séquence Salesman, les résultats sont le tableau 6.14 et les graphiques des figures 6.20 et 6.21. Ces chiffres s'expliquent sachant que le dictionnaire a été construit avec les images même de la séquence, et que cette dernière demeure complexe. C'est pourquoi presque la totalité du potentiel de représentants est mobilisée lors de l'encodage des images. Le tableau indique à nouveau que les vecteurs hors norme appartiennent logiquement aux sous-bandes dont la source est déjà en boucle ouverte plus dispersée. Les courbes soulignent :

- un temps d'encodage court ;
- une qualité de reconstruction satisfaisante et constante ;
- qu'une hausse de G_q (*i.e.* plus d'information à coder car il y a plus de mouvement) correspond à une hausse du débit et du nombre de représentants mobilisés ;
- les sous-bandes disposant des dictionnaires plus grands (*i.e.* la sous-bande C par rapport à celle B, et la sous-bande E par rapport à celle F) sont les plus consommatrices en bits et en représentants. Elles marquent aussi le plus de variations :
- l'entropie pour une image (ou la séquence entière) demeure néanmoins proche de la limite imposée.

étiquette sous-bande	encodage séquence		
	nombre de représentants utilisés	nombre de vecteurs hors-norme	entropie [bpp]
A	17	24	0.433
B	21	9	0.131
C	42	0	0.255
D	20	0	0.113
E	1708	1	0.428
F	1208	0	0.255
G	138	0	0.035

TAB. 6.14 – *Codeur basé régions, encodage de Salesman, le débit moyen final est de 0,238 bpp pour un PSNR de 33.86 dB*

6.3.3 Encodage de MissAmerica

Nous codons dans des conditions identiques (même dictionnaire et même allocation) la séquence MissAmerica. Les résultats sont le tableau 6.15 et les graphiques des figures 6.22 et 6.23.
Cette séquence moins riche en mouvement et dont les images n'ont pas servi à construire le dictionnaire, mobilise lors de sa quantification beaucoup moins de représentants. Les conséquences sont que :

- le temps d'encodage a chuté ;

– les courbes relatives aux nombres de représentants par sous-bandes, ainsi que celles liées aux débits sont plus régulières et leurs valeurs moindres.

Il faut remarquer que la qualité de reconstruction de MissAmerica est également supérieure à celle de Salesman. Pour l'expliquer nous visualisons les sources des sous-bandes B, C et D en boucle fermée (voir la figure 6.24), nous notons que :

– le décalage par rapport aux sources en BO est manifeste, et souligne à nouveau l'impossibilité d'obtenir pour ce codeur hybride des séquences d'apprentissage entièrement représentatives des sources en BF ;

– les débits alloués aux dictionnaires étant très bas, les découpages des espaces sont grossiers (les "dead zones" sont de grandes tailles). ;

– la répartition de la source en BF de Salesman est très large (il y a beaucoup d'information) et différente de celle de la source en BO, le coût des représentants de la couronne va aussi croître. Cette distribution très dispersée souligne aussi que les erreurs de quantification introduites successivement dans la boucle de codage ont tendance à s'accumuler ;

– la répartition de la source en BF de missam est plus compacte et mieux modélisée par celle en BO. Les erreurs de quantification introduites dans la boucle de codage ont une influence moins perturbatrice.

étiquette sous-bande	encodage séquence		
	nombre de représentants utilisés	nombre de vecteurs hors-norme	entropie [bpp]
A	16	0	0.246
B	15	0	0.025
C	42	0	0.089
D	12	0	0.011
E	603	0	0.099
F	576	0	0.063
G	29	0	0.001

TAB. 6.15 – *Codeur basé régions, encodage de MissAmerica, le débit moyen final est de 0.064, bpp pour un PSNR de 39.38 dB*

6.4 Conclusion

Le QVAA inscrit au sein de ces deux schémas de codage hybride nous a montré ses potentialités pour la quantification à bas débits de scènes visiophoniques. Les impératifs de rapidité de construction des dictionnaires et d'encodage des images sont satisfaits. Nous avons aussi montré qu'un dictionnaire peut-être achevé pour un type de séquence donnée. Cependant le dictionnaire du QVAA est construit par apprentissage, et la mise en oeuvre de codeurs fonctionnant en boucle fermée, pose une difficulté quant à la constitution de

séquences d'apprentissage représentatives des futures sources à quantifier. Enfin notons que si l'introduction d'une zone grossièrement quantifiée autour du point 0 est judicieuse pour la quantification de sources hybrides, la taille de cette "dead zone" devrait être déterminée via des tests subjectifs.

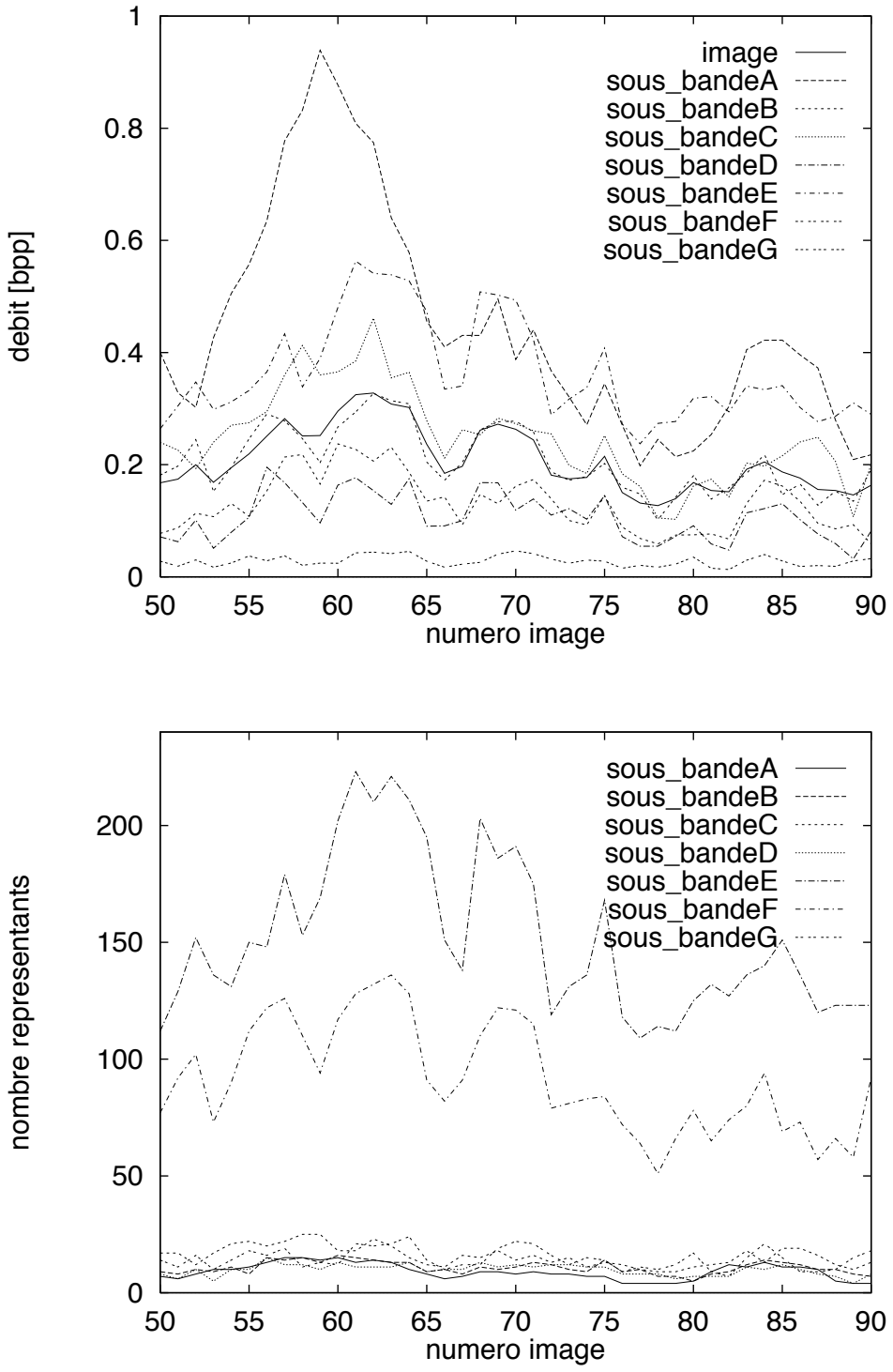


FIG. 6.20 – *Codeur basé régions, encodage de Salesman : entropie et nombre de représentants pour les images 50 à 90.*

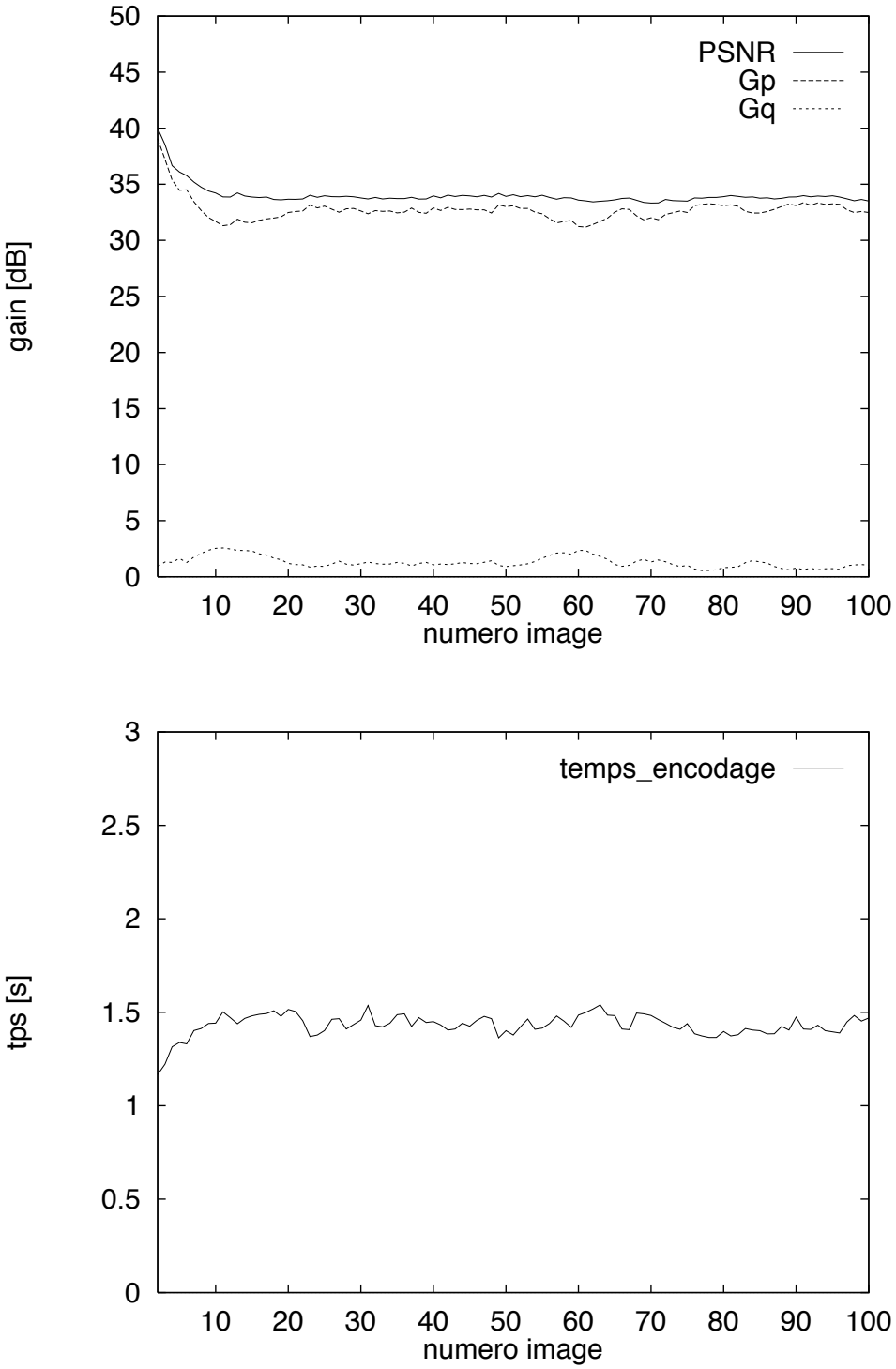


FIG. 6.21 – *Codeur basé régions, encodage de Salesman : gains et temps d'encodage pour les images 2 à 100.*

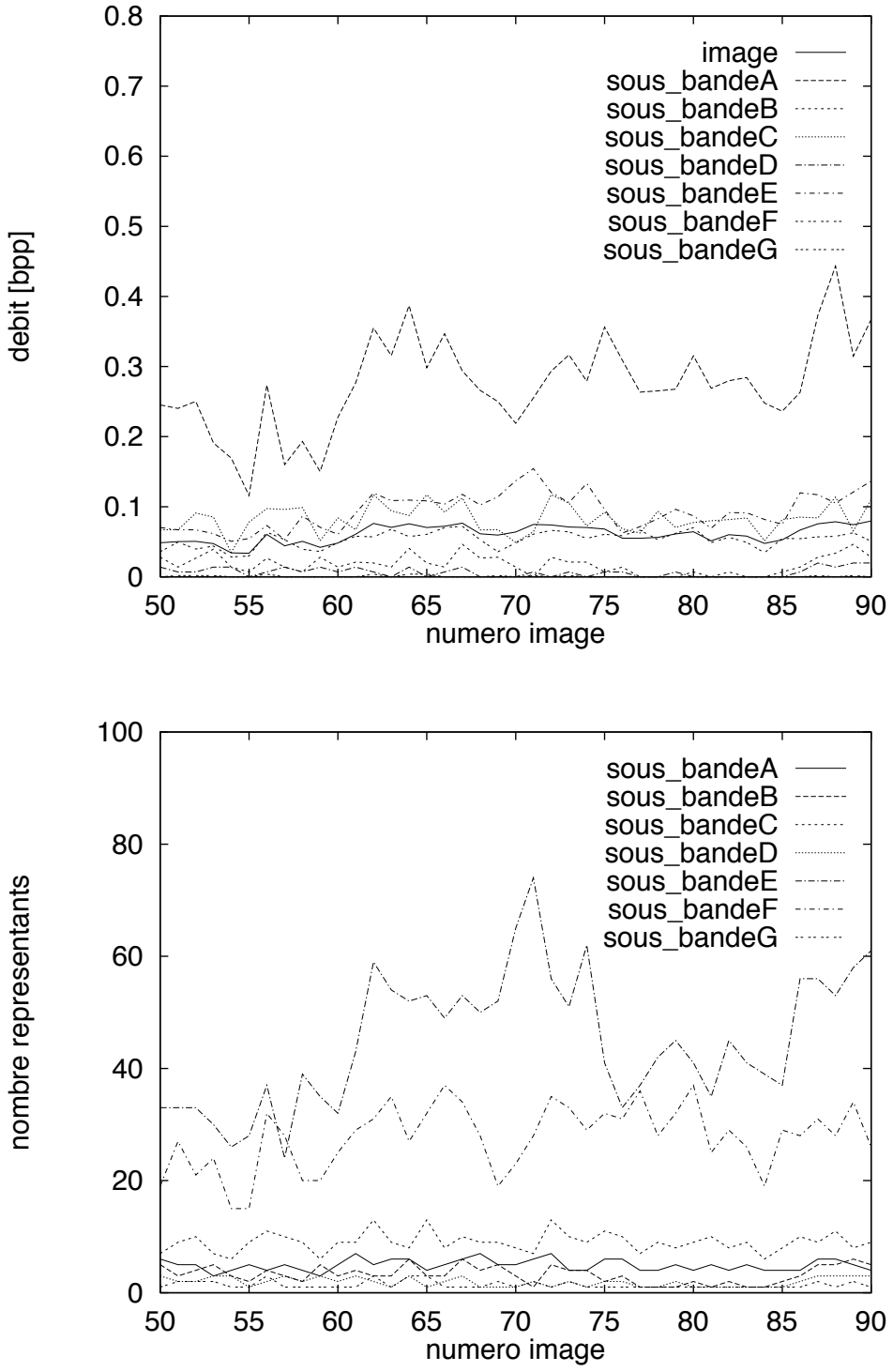


FIG. 6.22 – *Codeur basé régions, encodage de MissAmerica : entropie et nombre de représentants pour les images 50 à 90.*

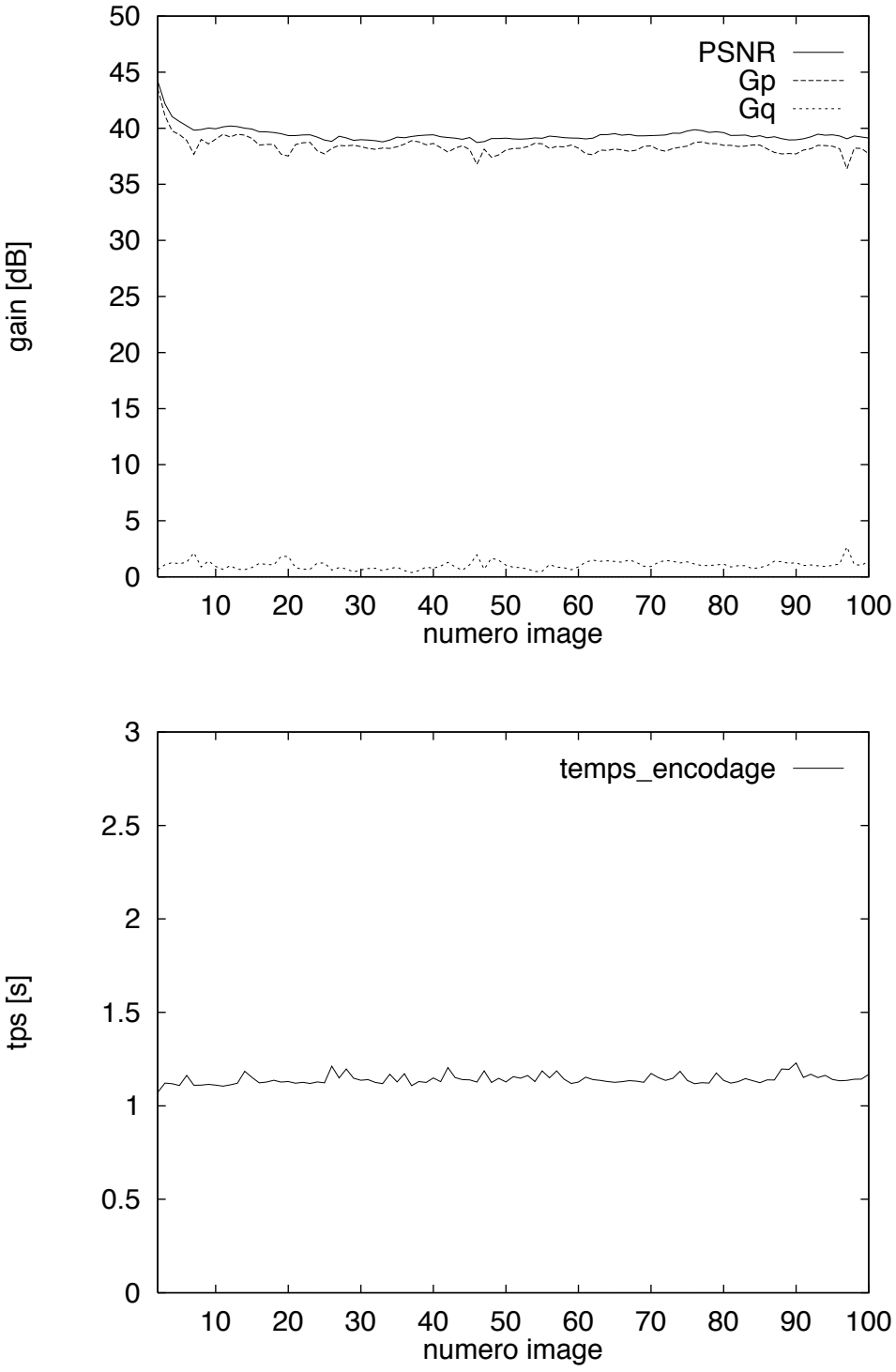


FIG. 6.23 – *Codeur basé régions, encodage de MissAmerica: gains et temps d'encodage pour les images 2 à 100.*

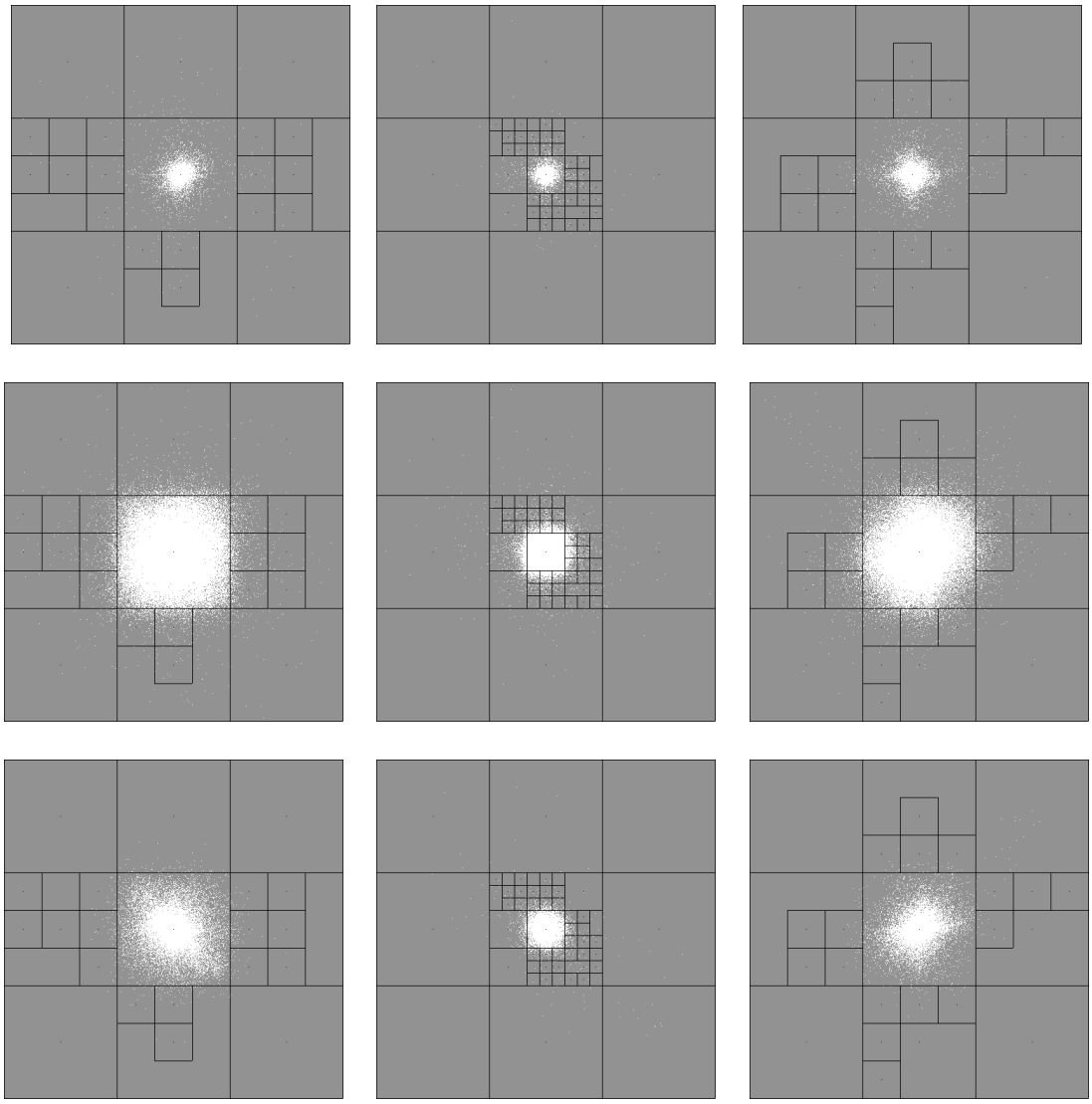


FIG. 6.24 – *Codeur basé régions, comparaison BO/BF : visualisation des sous-bandes B (images de la première colonne), C (seconde colonne) et D (troisième colonne), pour (de haut en bas) Salesman en BO, Salesman en BF, MissAmerica en BF.*

Conclusion et perspectives

Cette étude d'un schéma de quantification vectorielle algébrique et arborescente, s'inscrit parmi les travaux de recherche engagés pour le développement de nouveaux outils de codage vidéo. Les enjeux sont suscités notamment par la mise au point du futur standard international MPEG4 qui acceptera les outils codant au mieux les objets. La QV déjà retenue pour les ultimes normes de codage du signal de la parole, devrait-être l'un des outils efficaces de compression pour les futures générations de codeur vidéo.
L'utilisation de techniques de codage en sous-bandes est incontournable lors du choix d'une méthode de décorrélation pour la compression d'images. C'est pour un codeur de ce type que nous avons donc orienté notre QV, et notre premier travail a été d'étudier les approches de décorrélations inter-intra-images. Après avoir situé notre étude dans le contexte général de la théorie de l'information et des approches de codage, nous avons présenté le vaste spectre des outils de la QV. L'analyse théorique qui y est associée, est précieuse pour la conception d'un quantificateur, afin de préciser les caractéristiques du système et d'en analyser les performances. Si la QV optimale (dont les résultats sont les plus proches de ceux théoriques) a été définie, l'état de l'art souligne que les recherches se sont orientées vers la définition d'approches moins coûteuses. La méthodologie la plus récente est certainement la QVA qui offre une quantification rapide et un dictionnaire prédéfini. Mais cette technique est surtout développée pour la quantification de sources simplement modélisables. Pour l'adapter au codage de séquences d'images, nous proposons de procéder par emboîtages successifs de réseaux tronqués. Cette solution originale permet alors de combiner deux techniques de codage déjà éprouvées séparément :

- la quantification rapide sur réseaux algébriques ;
- la construction par apprentissage d'un dictionnaire arborescent permettant une partition de l'espace adaptée à la distribution de la source et à un critère débit-distorsion.

Au coeur du travail de doctorat est la conception de ce QVAA, avec :

- la technique de troncature et d'emboîtage des réseaux,
- le schéma de quantification multi-étages résultant,
- la construction du dictionnaire arborescent non-équilibré,
- la détermination du réseau algébrique optimal,

- la détection et le traitement des vecteurs hors-norme,
- une méthode propre d'allocation binaire pour le codage en sous-bandes.

La QVAA offre ainsi des solutions aux problèmes posés par la QVA, avec :

- une découpe de l'espace fonction de la distribution de la source,
- un indexage basé sur la structure arborescente du dictionnaire,
- une détection et le traitement simples des vecteurs hors-norme.

Ce nouveau quantificateur est une généralisation au cas vectoriel de l'approche d'emboîtement de QS abondamment cité en codage hiérarchique. Ce QVAA est aussi particulièrement adapté à la quantification de sources image différentielles ou hybrides, car il y a création d'une zone grossièrement quantifiée autour de l'origine.
Les tests sont effectués en inscrivant le QVAA au sein de deux schémas de codage vidéo ; l'un classique de type MPEG, l'autre novateur basé régions. Les résultats expérimentaux soulignent que le QV s'adapte correctement à la structure hybride de ces codeurs, et que les impératifs de rapidité de construction du dictionnaire et d'encodage sont remplis. Les débits atteints sont alors bas pour une qualité satisfaisante de restitution des images.

Les perspectives et les directions à suivre pour améliorer nos résultats portent notamment sur :

- la mise en place de tests subjectifs afin de déterminer les valeurs correctes des paramètres à caractères psychovisuels ;
- la constitution de séquences d'apprentissage plus représentatives des sources hybrides à quantifier.

Il serait aussi intéressant de juger les aptitudes de ce QV pour des codages où sa structure arborescente serait mise à profit. Nous trouvons :

- le codage hiérarchique avec la transmission progressive des images par approximations successives ;
- le codage adaptatif où le dictionnaire serait composé de deux parties :
  - d'une "souche" permanente constituée hors-ligne relativement à une séquence d'apprentissage de grande taille, et représentative d'une large gamme de séquences d'images ;
  - de branches actualisées pour chaque nouvelle séquence à coder (la séquence d'apprentissage serait plus réduite).

Enfin, l'un des axes de recherche au coeur des travaux pour la norme MPEG4 étant dans l'amélioration des standards actuels (*i.e.* H261, MPEG1&2) avec le codage dit basé régions, la QV doit s'adapter à la quantification de régions de formes arbitraires (*e.g.* avec

la découpe adaptative des régions en vecteurs, l'allocation binaire entre régions, l'injection de critères psychovisuels, ...). Cette méthode de codage basé régions doit également permettre d'appréhender le problème de la compression à bas débit de manière sélective au sens d'une qualité de reconstruction inhomogène spatialement. Nous pensons que le QVAA développé dans cette thèse, peut constituer (par hybridation avec d'autres approches de prédiction, de transformée, d'allocation binaire, ...) un maillon intéressant pour la constitution d'une "boîte-à-outils" génériques pour la compression de signaux vidéo.

Annexe A

Compression du signal de parole

L'UIT-T s'est premièrement engagée dans le processus de normalisation du codage du signal de parole pour le réseau téléphonique public. Nous rappelons brièvement les standards de compression de ce signal, car celui ultime aboutit à la mise en oeuvre de la quantification vectorielle comme outil de compression.

A.1 Compression du signal de parole dans la bande téléphonique

Nous pouvons résumer les diverses étapes de la standardisation par l'UIT-T du signal de parole [Moreau95] pour le réseau téléphonique public ([300 – 3300 Hz], fréquence d'échantillonnage $f_e = 8$ kHz) :
1972, norme **G.711** : codage MIC à 64 kbit/s (compression de type non-linéaire, amplitude de l'échantillon sur 8 bits).
1984, norme **G.721** : codage MICDA à 32 kbit/s.
1991, norme **G.728** : codage LD-CELP à 16 kbit/s, techniques de modélisation (modèle de production de type source-filtre) et de **quantification vectorielle**.
1994 : tests comparatifs pour l'élaboration d'une norme pour un codage à 8 kbit/s avec un codeur ACELP.

A.2 Communication entre mobiles

Ce secteur en plein développement vise à économiser la largeur de bande de façon à permettre la communication d'un grand nombre d'utilisateurs sur le canal radio. Trois générations de codeurs se sont succédées : une première avec des techniques d'accès multiples par division du temps (codeur de source RPE-LTP à 13 kbit/s en Europe et codeur de source VSELP en Amérique du Nord à 8 kbit/s), une seconde génération avec le codeur "1/2 GSM" par l'ETSI et celui "Half Rate" par le CTIA, la troisième avec des techniques d'accès multiples par division de code UMTS en Europe et FPLMTS en Amérique du Nord, division par 10 des débits). Une nouvelle génération de codeurs pour une téléphonie mobile par satellites en orbite basse à un débit encore plus faible est attendue.

1989, norme européenne **GSM.**
1990, norme américaine **IS54.**
1992, ouverture du service **Itinéris** par France Télécom.

A.3 Communication du signal de parole en bande élargie

Le signal de parole restituée en bande élargie ($[50 - 7000$ Hz], fréquence d'échantillonnage $f_e = 16$ kHz) est plus net et intelligible que dans le cas de la bande téléphonique. Les applications visées sont le visiophone, la téléphonie sur haut-parleurs, les conférences audio-visuelles, ...

1986, norme **G.722** [Mermelstein88] : pour la transmission simultanée sur un même canal, codage en deux sous-bandes avec pour chacune un codeur type norme G.721. Un débit réduit à 56 kbit/s est aussi possible.
1996 : codeurs à 32 kbit/s et même 16 kbit/s avec, soit des codeurs CELP, soit des codeurs type "signal musical" modifiés.

Annexe B

Eléments de la théorie de l'information pour le codage de la source

Nous indiquons les références des travaux de Shannon dont sont issus la théorie de l'information [Shannon48] [Shannon59], ainsi que celles d'ouvrages relatifs à cette théorie [Berger71] [Blahut87] [Gray90].

B.1 Entropie

Considérons une source à temps discret et ergodique $\{x(n)\}, n = 0, \pm 1, \pm 2, ...$ Le flux des **symboles** $x(n)$ forme une séquence aléatoire $\{X(n)\}, n = 0, \pm 1, \pm 2, ...$ dont les réalisations x_i appartiennent à l'**alphabet** de la source $\mathcal{A} = \{x_i / i = 1, 2, ..., K\}$. Si K est fini la source est dite à **amplitude discrète,** si K est infini la source est à **amplitude continue.** La source est **sans mémoire** si les échantillons successifs sont statistiquement indépendants.

B.1.1 Source à amplitude discrète et sans mémoire

Soit p_i la probabilité d'occurence du symbole x_i : $p_i = Pr\{X(n) = x_i\}$. Une appréciation numérique de l'information propre de l'évènement est alors donnée par : $I(x_i) = -\log_2 p_i$ [en bit/symbole] et l'information propre moyenne ou **entropie** (ou entropie d'ordre 0), qui représente la limite fondamentale pour représenter sans distorsion la source d'information, est [en bpp] :

$$H(X) = E(I(X)) = -\sum_{i=1}^{K} p_i . \log_2 p_i$$

Nous avons : $0 \leq H(X) \leq \log_2 K$, si $H(X) = 0$. La source est totalement **prédictible**, et si $H(X) = \log_2 K$ la source est **non prédictible** (les symboles sont équiprobables).

$\log_2 K$ mesure la **capacité** de l'alphabet. Les **redondances** de la source sont appréciées en calculant la différence : $(\log_2 K - H(X))$

B.1.2 Source à amplitude discrète avec mémoire

Il existe alors des dépendances statistiques entre les échantillons successifs de la source. Soit $\mathbf{x} = \mathbf{x}(n) = (x(n), x(n+1), ..., x(n+k-1))^T$, un vecteur constitué de k échantillons successifs de la source. Ce vecteur est caractérisé par sa probabilité conjointe $p_{\mathbf{X}}(\mathbf{x})$ qui est indépendante du temps si nous considérons une source stationnaire, $\mathbf{x}$ est une réalisation du vecteur aléatoire $\mathbf{X} = (X(n), X(n+1), ..., X(n+k-1))^T$. Soit l'**entropie conjointe** des vecteurs aléatoires (ou entropie d'ordre k) [en bpp] :

$$H_k(\mathbf{X}) = \frac{1}{k}.E(-\log_2 p_{\mathbf{X}}(\mathbf{X})) = -\frac{1}{k}.\sum\sum_{\mathbf{x}}...\sum p_{\mathbf{X}}(\mathbf{x}).\log_2 p_{\mathbf{X}}(\mathbf{x})$$

et :

$$H(X) = \lim_{k\to\infty} H_k(\mathbf{X})$$

Nous considérons aussi l'**entropie conditionnelle** d'ordre k d'un symbole à l'instant n étant donnés les $(k-1)$ symboles précédents : $H(X(n)/X(n-1), X(n-2), ..., X(n-k+1))$. Il est démontré que :

$$H(X(n)/X(n-1), X(n-2), ..., X(n-k+1)) \leq H_k(\mathbf{X})$$

et que :

$$\lim_{k\to\infty} H(X(n)/X(n-1), X(n-2), ..., X(n-k+1)) = H(X)$$

Les deux fonctions $H_k(\mathbf{X})$ et $H(X(n)/X(n-1), X(n-2), ..., X(n-k+1))$ sont décroissantes avec k. De plus, pour deux sources ayant deux alphabets identiques et même probabilité pour les symboles, il est prouvé que :

$$(H(X) \text{ / source avec mémoire}) \leq (H(X) \text{ / source sans mémoire})$$

Ces résultats annoncent l'intérêt de la quantification vectorielle car pour rejoindre le débit entropique il faut regrouper les échantillons de la source pour en exploiter la mémoire. Il est aussi préférable d'affecter un code à un symbole en tenant compte de sa probabilité d'apparition et en connaissant les valeurs de ses voisins, plutôt que de coder des vecteurs en fonction de leurs probabilités d'occurence.

B.1.3 Codage sans perte d'une source à amplitude discrète

Nous rappelons que le dictionnaire d'un quantificateur vectoriel généré pour atteindre une distorsion moyenne donnée, est un exemple de source à amplitude discrète.

La théorie annonce qu'un codage sans perte d'information avec un débit binaire égal à l'entropie est réalisable pour une source à amplitude discrète, cependant elle n'explique comment construire le code.

En reprenant les notations précédentes, le codage des $\{x(n)\}$ consiste en une application de l'alphabet $\mathcal{A}$ de la source vers celui $\mathcal{A}_c$ du code (*e.g.* l'alphabet du code est communément choisi binaire $\mathcal{A}_c = \{a^1, a^2\}$). Un mot de code c_i est une suite finie d'éléments de $\mathcal{A}_c$: $c_i = \left[a^{i(1)} \dots a^{i(l_i)}\right]$, et l_i est la longueur de ce mot. Le **code** $C = \{c_1 \dots c_K\}$ est donc l'ensemble des c_i. L'entropie est toujours inférieure ou égale à la longueur moyenne des mots de code $\overline{l}$:

$$H(\mathcal{D}) \leq \sum_{i=1}^{K} p_i.l_i = \overline{l}$$

L'entropie précise donc la longueur moyenne des mots du code binaire le plus économique pour transmettre l'information, c'est le débit binaire moyen minimum qui souvent ne peut qu'être approché en construisant un code entropique. Les algorithmes de construction de tels codes procèdent en formant des mots à longueur variable. Pour présenter de façon intuitive ces codes entropiques, nous pouvons expliquer que les statistiques du signal à coder sont exploitées en affectant les mots de code les plus courts aux symboles de la source les plus fréquents. Nous présentons, sans les détailler, les codeurs entropiques utilisés en codage d'images :

- l'algorithme de Huffman [Huffman52] est le plus utilisé et il offre dans la plupart des cas des résultats suffisants. Parmi les codes vérifiant la condition du préfixe (*i.e.* aucun mot de code ne doit-être le préfixe d'un autre mot), il est optimal car il fournit une longueur moyenne de mots de code inférieure. Cependant l'entropie de la source est atteinte que dans le cas où les symboles ont des probabilités d'occurence suivant une loi de probabilité en puissance négative de deux. Pour approcher ce résultat il est souvent nécessaire de regrouper les échantillons et de considérer les fréquences d'apparition de vecteurs, les procédures de codage deviennent alors complexes. Un autre problème est lié au fait que la construction du code repose sur un processus itératif impliquant la génération d'un arbre binaire, ce dernier étant fonction des probabilités des symboles à transmettre. Il est alors nécessaire de transmettre cet arbre et des mémoires de transcodage sont mobilisées au décodeur ;

- la famille de codeurs dérivés de celui de Huffman mais qui sont tels qu'il n'est pas nécessaire de transmettre l'arbre ayant engendré le code. L'arborescence s'appuie alors sur des paramètres dont l'adaptation est réalisée en fonction de la statistique du signal à coder. Nous donnons l'exemple des "Arithmetically Computed Variable Length Coding" (ACVLC) qui consistent à générer des codes à l'aide de plusieurs champs de longueurs différentes et où chacun ne peut coder qu'un certain nombre de symboles. Si un champ est de longueur insuffisante pour coder un symbole, le champ suivant de longueur supérieure est ajoutée. Cette technique est souvent présentée comme intermédiaire entre le code à longueur fixe et celui à longueur variable ;

- le codage arithmétique [Howard et al.94] qui offre des performances supérieure à l'algorithme de Huffman et qui tend vers la limite théorique. Sa supériorité repose sur le fait qu'il consiste à attribuer un code à longueur variable à une suite d'évènements dont le nombre peut varier (il n'y a pas alors de table de correspondance entre la

source d'éléments à coder et les mots de code). La construction des mots de code est réalisée à partir de la probabilité du symbole courant et de celles cumulées des symboles précédents. Sa mise en oeuvre demeure relativement complexe, et un délai de décodage est nécessaire car il faut lire la séquence entièrement avant de la décoder ;

- le codage par plage ou "run length coding" qui intervient lorsque un symbole a une probabilité d'apparaître plusieurs fois de suite élévée, il est alors intéressant de coder ce symbole en spécifiant le nombre de ses répétitions. En pratique, pour mettre en oeuvre ce codage, il est intéressant de procéder à un réarrangement des données de la source pour aboutir à une représentation compacte par des vecteurs composés d'éléments identiques. Les "Universal Variable Length Coding" (UVLC) [Delogne et al.91] sont un exemple où l'on cherche à coder des plages de "0" de plans de bits représentants l'information. Le codage par plage est aussi largement exploité par les méthodes de codage entropique précédentes (Huffman, ACVLC, ...).

B.1.4 Source à amplitude continue et sans mémoire

Soit $\{x(n)\}$ une source stationnaire et sans mémoire, nous considérons également qu'elle est centrée. Soit $p_X(x)$ sa fonction de densité de probabilité, $\sigma_X^2 = E(X^2(n))$ sa variance et $R_X = \sigma_x^2.\delta(u)$ sa fonction d'autocorrélation ($\delta(u)$ étant le symbole de Kronecker). Ce signal a une **entropie absolue** infinie (en effet $p_i = 0$ donc $H(X) = +\infty$), c'est pourquoi l'**entropie différentielle** est introduite (elle peut-être positive, négative ou nulle en fonction de l'amplitude de la source) :

$$h(X) = E(-\log_2 p_X(X)) = -\int_{-\infty}^{+\infty} p_X(x).\log_2 p_X(x)\ dx$$

Il est démontré que l'entropie différentielle est maximale si la source suit une loi gaussienne $\mathcal{N}(0, \sigma_X^2)$, dans ce cas :

$$p_X(x) = (2.\pi.\sigma_X^2)^{-1}.\exp(\frac{-x^2}{2.\sigma_X^2}) \quad \text{et} \quad h(X)_G = \log_2 \sqrt{2.\pi.e.\sigma_X^2}$$

Il est aussi pratique de définir la **puissance entropique** qui traduit la répartition de l'information par unité de variance du signal :

$$\mathcal{P} = \frac{1}{2.\pi.e}.2^{2.h(X)}$$

Pour une source gaussienne $\mathcal{P} = \sigma_X^2$, pour une non-gaussienne $\mathcal{P} < \sigma_X^2$.

B.1.5 Source à amplitude continue avec mémoire

Nous considérons cette fois un vecteur $\mathbf{x}$ constitué de k échantillons successifs de la source, $\mathbf{x}$ est décrit par sa fonction de densité de probabilité conjointe $p_{\mathbf{X}}(\mathbf{x})$ et l'entropie différentielle est définie par :

$$h(X) = \lim_{k \to \infty} h_k(\mathbf{X})$$

avec :

$$h_k(\mathbf{X}) = \frac{1}{k}.E(-\log_2 p_{\mathbf{X}}(\mathbf{X})) = -\frac{1}{k}.\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}...\int_{-\infty}^{+\infty} p_{\mathbf{X}}(\mathbf{x}).\log_2 p_{\mathbf{X}}(\mathbf{x})\ d\mathbf{x}$$

De façon générale :

$$(h(X)\ /\ \text{source avec mémoire}) < (h(X)\ /\ \text{source sans mémoire}) \leq \frac{1}{2}.\log_2(2.\pi.e.\sigma_X^2)$$

La borne supérieure est atteinte dans le cas d'une source gaussienne, les entropies inférieures à cette valeur sont dues aux redondances de la source (sa densité de probabilité n'est pas gaussienne et/ou la source présente une mémoire alors son spectre de puissance n'est pas "plat" ou "blanc"). La puissance entropique d'une source gaussienne "colorée" (*i.e.* avec mémoire) est : $\mathcal{P} = \gamma_X^2.\sigma_X^2$, où γ_X^2 est la **mesure d'étalement du spectre.** Nous avons $0 \leq \gamma_X^2 \leq 1$, si $\gamma_X^2 = 1$ nous retrouvons le cas d'une source gaussienne sans mémoire, si $\gamma_X^2 < 1$ la source présente une mémoire (le signal est plus ou moins prédictible). L'inégalité suivante est alors prouvée :

$$(\mathcal{P}\ /\ \text{source avec mémoire}) < (\mathcal{P}\ /\ \text{source sans mémoire}) \leq \sigma_X^2$$

B.2 Fonction débit-distorsion

B.2.1 Introduction

En pratique la source à coder est le plus souvent à amplitude continue. Il est souhaité que le système codeur-décodeur assure la transmission de l'information avec un débit R [en bpp] adapté au canal pour une erreur moyenne de reconstruction D' minimale. En faisant varier R une courbe $D'(R)$ est obtenue. La théorie de l'information annonce qu'il existe une **fonction débit-distorsion** $D(R)$ qui fournit une borne aux performances du système de codage. Cette fonction $D(R)$ indique la distorsion minimale théorique pour le codeur avec le débit R : $D(R) \leq D'(R)$
Dans le cas d'une source à amplitude discrète (pour laquelle une transmission sans erreur est possible), la courbe $R'(D)$ est souvent utilisée. Là encore, la théorie fournit la **courbe distorsion-débit** $R(D)$ qui est l'inverse de la fonction $D(R)$ et telle que : $R'(D) \geq R(D)$

B.2.2 Source à amplitude discrète

Un codage entropique est donc réalisable. Le débit minimum pour transmettre, sans perte, l'information est : $\min\{R\} = R(0) = H(X)$. $R'(0) = H(X)$ est approchée à l'aide d'un code entropique ou à longueur variable. Si la source est avec mémoire la calcul du code est plus aisé. La figure B.1 donne un exemple d'une telle courbe $D(R)$ qui est donc une fonction monotone décroissante avec R. La source étant de variance finie, alors $D(0) = \sigma_X^2$ est finie. En effet nous pouvons considérer que, pour $R = 0$ le codeur n'émet que des 0, les erreurs de reconstruction au décodeur sont alors égales aux échantillons des vecteurs source et la variance de l'erreur équivaut à celle de la source.

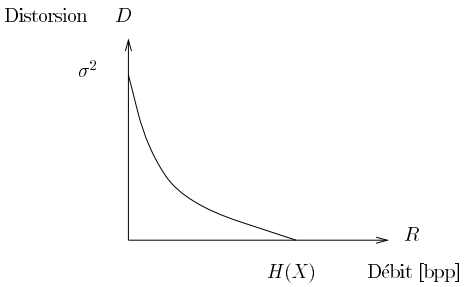


FIG. B.1 – *Exemple de la courbe débit-distorsion d'une source à amplitude discrète.*

B.2.3 Source à amplitude continue

Distorsion

Nous considérons $\{x(n)\}$, $n = 0, \pm 1, \pm 2, ...$ une source stationnaire, à amplitude continue. Soit $\mathbf{x}$ un vecteur de k échantillons de la source, $\mathbf{x}$ est une réalisation du vecteur aléatoire $\mathbf{X}$, ce dernier est caractérisé par sa fonction de densité de probabilité conjointe $p_{\mathbf{X}}(\mathbf{x})$. Les $\mathbf{y}$ sont eux les vecteurs de reproduction qui arrivent au récepteur, $\mathbf{y}$ est donc une réalisation du vecteur aléatoire $\mathbf{Y}$. Nous parlons de modèle à amplitude continue car les échantillons de ces vecteurs appartiennent à la droite réelle. En général le vecteur de reproduction $\mathbf{y}$, correspondant au $\mathbf{x}$ émis, en est différent ($\mathbf{y} \neq \mathbf{x}$), nous choisissons donc d'apprécier la **distorsion moyenne par symbole** à l'aide d'une mesure d'erreur quadratique moyenne :

$$E[d(\mathbf{X},\mathbf{Y})] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} ... \int_{-\infty}^{+\infty} d^2(\mathbf{x},\mathbf{y}).p_{\mathbf{XY}}(\mathbf{x},\mathbf{y})\, d\mathbf{x}.d\mathbf{y}$$

où $d(\mathbf{x},\mathbf{y})$ est la distance euclidienne définit à l'équation 2.1
Nous avons $p_{\mathbf{XY}}(\mathbf{x},\mathbf{y}) = p_{\mathbf{Y/X}}(\mathbf{y/x}).p_{\mathbf{Y}}(\mathbf{x})$. La distorsion moyenne, sujette à une contrainte sur le débit, dépend donc de la statistique de la source $p_{\mathbf{X}}(\mathbf{x})$ et de la probabilité des transitions $p_{\mathbf{Y/X}}(\mathbf{y/x})$. Pour la minimiser il faut choisir une modélisation appropriée entre la source et les vecteurs de reconstruction.

Information mutuelle

Le calcul de la courbe $D(R)$ repose sur le concept d'**information mutuelle** [par symbole] $I(X;Y)$ qui est la mesure capable de décrire le flux d'information entre le codeur et le décodeur, exactement elle apprécie la quantité moyenne d'information qu'implique la "réception" des vecteurs par rapport à ceux émis.

$$I(X;Y) = \lim_{k\to\infty} I_k(\mathbf{X};\mathbf{Y})$$

avec :

$$I_k(\mathbf{X};\mathbf{Y}) = \frac{1}{k}.\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} ... \int_{-\infty}^{+\infty} p_{\mathbf{XY}}(\mathbf{x},\mathbf{y}).\log_2\left(p_{\mathbf{Y/X}}(\mathbf{y/x}).p_{\mathbf{X}}(\mathbf{x})\right)\, d\mathbf{x}.d\mathbf{y}$$

$$= \frac{1}{k}.\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\ldots\int_{-\infty}^{+\infty} p_{\mathbf{XY}}(\mathbf{x},\mathbf{y}).\log_2\left(p_{\mathbf{X/Y}}(\mathbf{x/y}).p_{\mathbf{Y}}(\mathbf{y})\right)\, d\mathbf{x}.d\mathbf{y}$$

$I(X;Y)$ dépend donc également de la statistique de la source et de celle des transitions. L'adjectif "mutuel" vient de l'égalité: $I(X;Y) = I(Y;X)$. L'information mutuelle calcule le débit minimal R nécessaire pour avoir une fidélité de reconstruction D. En effet pour une densité $p_{\mathbf{X}}(\mathbf{x})$ donnée considérons S, l'ensemble de tous les schémas de codage ayant une fonction de densité de probabilité de transition pour laquelle l'information mutuelle par symbole est inférieure à un débit donné:

$$S = \left\{p_{\mathbf{Y/X}}(\mathbf{y/x}) : I_k(\mathbf{X};\mathbf{Y}) \leq R\right\}$$

Chacun de ces schémas réalise une erreur moyenne $E[d(\mathbf{X},\mathbf{Y})]$. Nous recherchons alors celui assurant le minimum de distorsion:

$$D_k(R) = \min_{p_{\mathbf{Y/X}}(\mathbf{y/x})\in S} E[d(\mathbf{X},\mathbf{Y})]$$

Le codeur réalisant $D_k(R)$ est optimal, il assure la distorsion moyenne minimale sous une contrainte de débit inférieur à R. Nous remarquons qu'à ce stade $D_k(R)$ est une fonction monotone décroissante avec R. La théorie montre que ce schéma de codage optimal doit être tel que ses vecteurs source soient statistiquement indépendants. Alors la **fonction débit-distorsion** est définie par:

$$D(R) = \lim_{k\to\infty} D_k(R)$$

La fonction $D(R)$ fournit pour à un débit donné R, une borne minimale à la distorsion de tous codeurs (un exemple d'une telle fonction est donné à la figure B.2). En pratique aucun schéma de codage ne peut atteindre une telle performance. Cependant ces équations annoncent que de meilleurs résultats seront obtenus en utilisant des quantificateurs vectoriels.
La courbe inverse de $D(R)$ est $R(D)$. Elle correspond au débit minimal nécessaire pour que le signal reconstruit au récepteur ait une distorsion D. Dans le cas d'une source à amplitude continue $R(0) = +\infty$ car, pour $D = 0$, $p_{\mathbf{Y/X}}(\mathbf{y/x}) = (1 \text{ si } \mathbf{y} = \mathbf{x}\ ;\ 0 \text{ sinon})$ et $H(X) = H(Y) = I(X;Y) = +\infty$. Dans ce dernier cas il n'y a donc pas codage, le cas étudié est plutôt $D > 0$ et $I(X;Y) < +\infty$. Le codage de la source est donc une **opération irréversible** causant une distorsion D pour un débit R.

Source sans mémoire: cas d'une source gaussienne

Nous considérons la source obéissant à la loi normale $\mathcal{N}(0,\ \sigma_X^2)$ alors:

$$R(D)_G = \max\left\{0\ ,\ \frac{1}{2}.\log_2\frac{\sigma_X^2}{D}\right\} = \begin{cases} \frac{1}{2}.\log_2\frac{\sigma_X^2}{D} & \text{si } 0 \leq D \leq \sigma_X^2 \\ 0 & \text{si } D \geq \sigma_X^2 \end{cases}$$

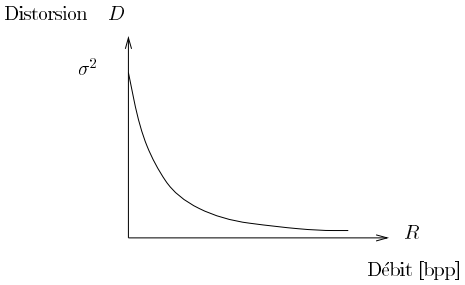


FIG. B.2 – *Exemple de la courbe débit-distorsion d'une source à amplitude continue.*

et

$$D(R)_G = 2^{-2.R}.\sigma_X^2$$

Comme dans le cas de la source à amplitude discrète, il est évident que pour avoir $D = \sigma_X^2$ il n'y a pas d'information à transmettre (*e.g.* tous les vecteurs de reproduction ont leurs composantes nulles). Nous remarquons que l'erreur quadratique moyenne est réduite d'un facteur quatre pour chaque bit ajouté à la transmission, ou encore que le rapport signal à bruit [en DB] est 6.02 fois le débit.

Il est intéressant de modéliser les liens entrée/sortie du système global de codage afin d'en connaître la configuration optimale et d'analyser les effets de la quantification.

Ainsi le système réalisant $R(D)_G$ peut être décomposé en :

$$p_{\mathbf{Y}/\mathbf{X}}(y/x) = \frac{1}{\sqrt{2.\pi.\beta.D}}.\exp\left(\frac{-(y-\beta.x)^2}{2.\beta.D}\right) \quad \text{avec} \quad \beta = 1 - \frac{D}{\sigma_X^2}$$

La sortie obéit à une loi normale $\mathcal{N}(\beta.x,\ \beta.D)$. Les erreurs de quantification suivent donc également une loi gaussienne et elles sont indépendantes vis à vis des variables à l'entrée. Enfin :

$$R(D)_G = \frac{1}{2}.\left(\log_2 \sigma_X^2 - \log_2 D\right)$$

Le débit nécessaire pour reproduire la source avec la distorsion D est la différence en entropie entre la source et le bruit de quantification qui sont deux variables aléatoires normales de variances σ_X^2 et D.

Source sans mémoire : cas d'une source non-gaussienne

Il n'existe pas de fonction débit-distorsion explicite mais des bornes de la forme :

$${}^{L}D(R) \leq D(R) \leq D(R)_G$$

La borne supérieure est la fonction $D(R)_G$ correspondant à la source gaussienne sans mémoire. La borne inférieure correspond à la **borne de shannon :**

$${}^{L}D(R) = \frac{1}{2.\pi.e}.2^{-2.(R-h(X))} = 2^{(-2.R.\mathcal{P})} \quad \text{ou} \quad {}^{L}R(D) = h(X) - \frac{1}{2}.\log_2\left(2.\pi.e.D\right)$$

La puissance entropique $\mathcal{P}$ correspond pour une source non-gaussienne à la variance de la gaussienne qui aurait la même entropie différentielle. Pour une gaussienne $\mathcal{P} = \sigma_X^2$ et on retouve $D(R) = {}^{L}D(R)$. En pratique, pour une large classe de distributions et à haut débit, ${}^{L}D(R)$ tend vers la fonction débit-distorsion du système $D(R)$. Pour des débits inférieurs (de 1 à 3 bits/échantillon), ${}^{L}D(R)$ est une borne trop optimiste, $D(R)$ est alors calculée numériquement via l'algorithme de blahut [Blahut72].

Source avec mémoire : cas d'une source gaussienne

La théorie annonce qu'une plus grande compression est possible pour les sources non-gaussiennes avec mémoire. Cependant pour atteindre un tel résultat (traduit par la courbe $D(R)$), le système de codage nécessite plus d'information sur la source que celle uniquement fournit par son spectre de puissance ($S_X(e^{j.w})$) et sa fonction d'autocorrélation.

La présentation théorique qui suit est introduite afin de mieux apréhender les problèmes liés à la quantification (la localisation des erreurs introduites par le codage).

La fonction $D(R)$ d'une source gaussienne colorée (c.a.d avec mémoire, par opposition à la source sans mémoire dont le spectre de puissance constant est dit blanc) est donnée sous une forme paramétrique (le paramètre est ϕ) :

$$D(\phi)_G = \frac{1}{2.\pi}.\int_{+\pi}^{-\pi} \min\left\{\phi, S_X(e^{j.w})\right\}\ dw$$

$$R(\phi)_G = \frac{1}{2.\pi}.\int_{+\pi}^{-\pi} \max\left\{0, \frac{1}{2}.\log_2 \frac{S_X(e^{j.w})}{\phi}\right\}\ dw$$

Ces équations peuvent être interprétées de la façon suivante, l'axe des fréquence est divisé en deux ensembles A et B (voir la figure B.3) :

- $w \in A$ si $S_X(e^{j.w}) \geq \phi \Longleftrightarrow \frac{S_X(e^{j.w})}{\phi} \geq 1$. Donc $R(\phi)_G > 0$, l'information est transmise et la zone est **passe bande** ;
- $w \in B$ si $S_X(e^{j.w}) < \phi \Longleftrightarrow \frac{S_X(e^{j.w})}{\phi} < 1$. Donc $R(\phi)_G = 0$, la contribution spectrale n'est pas transmise et c'est une zone **stoppe bande.**

Si nous considérons $S_R(e^{j.w})$, la puissance spectrale de l'erreur de reconstruction $r(n) = x(n) - y(n)$:

$$S_R(e^{j.w}) = \min\left\{\phi, S_X(e^{j.w})\right\} = \left\{\phi = \text{cste si } w \in A,\ S_X(e^{j.w}) \text{ si } w \in B\right\}$$

Alors :

$$D(\phi)_G = \frac{1}{2.\pi}.\int_{+\pi}^{-\pi} S_R(e^{j.w})\ dw \quad \text{et} \quad R(\phi)_G = \frac{1}{2.\pi}.\int_{+\pi}^{-\pi} \frac{1}{2}.\log_2 \frac{S_X(e^{j.w})}{S_R(e^{j.w})}\ dw$$

Dans les zones B, la puissance spectrale de l'erreur est égale à celle de la source (l'information relative à ces zones n'est pas transmise). Dans les zones A, la puissance spectrale

de l'erreur est constante, l'information relative à ces zones est transmise et ce que l'on désire alors est un rapport signal sur bruit le plus grand possible.

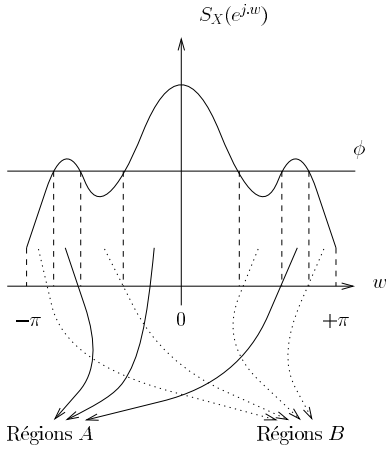


FIG. B.3 – *Schéma de principe (nous reconnaissons les zones A et B).*

En pratique, les sources continues ont un spectre de puissance strictement décroissant et nous pourrions croire que la quantification est parfaite lorsque son action peut être modélisée par un filtre passe-bas idéal ($H(e^{j.w}) = 1$ pour $w \in A$) suivit de l'addition d'un bruit blanc ayant une puissance spectrale constante (égale à ϕ) dans cette même région. En fait, le cas idéal où en effet le spectre de l'erreur des zones A est constante et égale à ϕ, est modélisé par la combinaison d'un filtre passe-bas non idéal :

$$H(e^{j.w},\ \phi) = \min\left\{0,\ 1 - \frac{\phi}{S_X(e^{j.w})}\right\}$$

et d'un bruit $b(n)$ additif, non blanc, indépendant de la variable à l'entrée et tel que :

$$S_B(e^{j.w}) = \min\left\{0,\ \phi.(1 - \frac{\phi}{S_X(e^{j.w})})\right\}$$

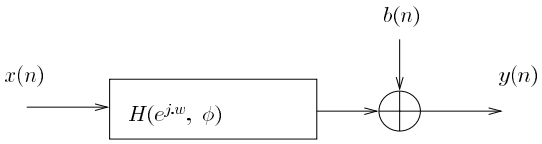


FIG. B.4 – *Modélisation de la quantification idéale.*

Les erreurs $r(n) = x(n) - y(n)$ ont donc une puissance :

$$S_R(e^{j.w},\ \phi) \quad = \quad S_X(e^{j.w}).|1 - H(e^{j.w},\ \phi)|^2 + S_B(e^{j.w},\ \phi)$$

$$= \frac{\phi^2}{S_X(e^{j.w})} + \phi.(1 - \frac{\phi}{S_X(e^{j.w})}) \quad / \ w \in A$$

$$= S_R^1(e^{j.w}) + S_R^2(e^{j.w}) \quad / \ w \in A$$

Il y a 2 termes : S_R^1 qui est du au filtre passe bande des zones A et S_R^2 du à la contribution du bruit additif non blanc dans les mêmes zones.

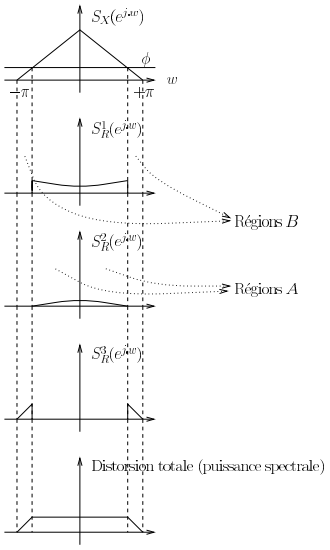


FIG. B.5 – *Les bruits de quantification.*

Le troisième terme S_R^3 qui apparaît dans la figure B.5, est du à la partie stoppe bande des régions B. Dans la littérature S_R^1 et S_R^2 sont les **bruits de codage**, S_R^3 correspond au **bruit de surcharge.**

Cas de petites distorsions

Nous parlons de "petites distorsions" lorsque $\phi \leq \min_w \{S_X(e^{j.w})\}$ (c'est le cas avec l'hypothèse haute résolution). Alors une forme simple de la fonction débit-distorsion est obtenue avec : $D(R)_G = \gamma_X^2.2^{-2.R}.\sigma_X^2$ où γ_X^2 est la mesure d'étalement du spectre S_X :

$$\gamma_X^2 = \frac{\exp\left(\frac{1}{2}.\int_{-\pi}^{+\pi} \log_e S_X(e^{j.w})\ dw\right)}{\sigma_X^2}$$

Pour un débit donné, nous avons :

$$(D(R)_G/\text{ source avec mémoire }) = \gamma_X^2.(D(R)_G/\text{ source sans mémoire })$$

En exploitant la mémoire la distorsion peut donc être réduite d'un facteur γ_X^2 dans la zone de "petites distorsions".

Fonction débit-distorsion en considérant des vecteurs de taille k

Nous avons déjà introduit $D_k(R)$, la fonction débit-distorsion correspondant à une source constituée de blocs de taille k, ces blocs étant indépendants statistiquement entre eux. Dans le cas d'une telle source gaussienne nous avons toujours : $D(R) = \lim_{k\to+\infty} D_k(R)$, et il est montré que :

$$D_k(R) = \frac{1}{k}\sum_{u=0}^{k-1} \min\{\phi,\ \lambda_u\} \quad \text{et} \quad R_k(\phi) = \frac{1}{k}\sum_{u=0}^{k-1} \max\left\{0,\ \frac{1}{2}.\log_2\frac{\lambda_u}{\phi}\right\}$$

λ_u étant la u^e valeur propre de la matrice d'autocorrélation Γ_X d'ordre k du processus $\{X(n)\}$. Nous remarquons que R_k apparaît comme la moyenne de k débits $R_u = \max\left\{0,\ \frac{1}{2}.\log_2\frac{\lambda_u}{\phi}\right\}$ (où chaque R_u résulte du codage de sources gaussiennes sans mémoire de variance λ_u) et que D_k apparaît aussi comme la moyenne de k distorsions optimales $D_u = \min\{\phi,\ \lambda_u\}$. Toutes les variables aléatoires dont les variances sont supérieures au paramètre ϕ contribuent de la même façon à la distorsion globale du système. Ces variables qui n'apportent aucune information n'ont donc pas besoin d'être transmise, alors : $R_u = 0$ pour $\phi \geq \lambda_u$

Fonction débit-distorsion en considérant des vecteurs de taille k et de petites distorsions

C'est la cas si $\phi \leq \min_{u=0,\ 1,\ ...,\ k-1}\{\lambda_u\}$. Alors $D_k = \phi$ et $D_k(R) = 2^{-2.R}.(\prod_{u=0}^{k-1}\lambda_u)^{\frac{1}{k}}$
Finalement nous obtenons :

$$D_k(R) \leq (D(R)_G/\text{ sans mémoire}) = 2^{-2.R}.\sigma_X^2$$

car $\sigma_X^2 = \frac{1}{k}.\sum_{u=0}^{k-1}\lambda_u \geq (\prod_{u=0}^{k-1}\lambda_u)^{\frac{1}{k}}$, il y a égalité si et seulement si les λ_u sont tous égaux (la source est alors blanche).

Les sources avec mémoire peuvent donc être transmises avec des distorsions inférieures aux sources sans mémoire.

En utilisant le résultat connu (où $\det(\Gamma_X(k))$ est le déterminant de la matrice d'autocorrélation à l'ordre k) : $\det(\Gamma_X(k)) = \prod_{u=0}^{k-1}\lambda_u$. Nous pouvons définir la puissance entropique par $\mathcal{P}_k = (\det(\Gamma_X(k)))^{1/k}$. Nous obtenons alors :

$$D_k(R) = 2^{-2.R}.\mathcal{P}_k = {}^{L}D(R)$$

La fonction débit-distorsion est donc égale à la borne de Shannon pour les petites distorsions.

Les théorèmes précédents sont aussi vrais si l'on considère des vecteurs successifs qui ne sont pas indépendants, alors il faut prendre k grand.

Source avec mémoire : cas d'une source non gaussienne

Nous retrouvons :

$$^{L}D(R) \leq D(R) \leq D(R)_{G}$$

avec $^{L}D(R)$ la borne de Shannon. Là encore, en considérant un second moment fixé et la métrique euclidienne, une source gaussienne est moins compressible qu'une non gaussienne.

Annexe C

Expression des formules pour l'élagage

Nous développons les formules intervenant pour le calcul et la mise à jour des retours marginaux dans le cadre de l'algorithme d'élagage du chapitre 4. Ces expressions anlytiques s'appliquent au cas général d'arbres B-aires ($B \geq 2$).

C.1 Formules de récurrence pour le calcul initial des retours marginaux

Nous rappelons que l'arbre $\mathcal{T}$ complet a été construit, et que pour chacun de ses noeuds n_i : $P(n_i)$, $d(n_i)$ et $l(n_i)$ ont été obtenus. Les formules de récurrences, développées en exploitant la propriété de linéarité des fonctionnelles $d(\mathcal{S})$ et $l(\mathcal{S})$, permettent de calculer les retours marginaux avant de lancer le processus d'élagage.

Si n_i a B fils n_j, nous avons :

$$\begin{aligned}
\Delta d(\mathcal{S}_{n_i}) &= P(n_i).d(n_i) - d(\mathcal{S}_{n_i}) \\
&= P(n_i).d(n_i) - \sum_{n_u \in \tilde{\mathcal{S}}_{n_i}} P(n_u).d(n_u) \\
&= P(n_i).d(n_i) - \sum_{B} \sum_{n_u \in \tilde{\mathcal{S}}_{n_j}} P(n_u).d(n_u)
\end{aligned}$$

or :

$$\Delta d(\mathcal{S}_{n_j}) = P(n_j).d(n_j) - \sum_{n_u \in \tilde{\mathcal{S}}_{n_j}} P(n_u).d(n_u) \Longleftrightarrow \sum_{n_u \in \tilde{\mathcal{S}}_{n_j}} P(n_u).d(n_u) = P(n_j).d(n_j) - \Delta d(\mathcal{S}_{n_j})$$

donc :

$$\Delta d(\mathcal{S}_{n_i}) = P(n_i).d(n_i) + \sum_{B} \Delta d(\mathcal{S}_{n_j}) - \sum_{B} P(n_j).d(n_j) \qquad \text{(C.Q.F.D)}$$

En procédant de la même façon :

$$\begin{aligned}\Delta l(\mathcal{S}_{n_i}) &= l(\mathcal{S}_{n_i}) - P(n_i).l(n_i) \\ &= \sum_{n_u \in \tilde{\mathcal{S}}_{n_i}} P(n_u).l(n_u) - P(n_i).l(n_i) \\ &= \sum_{B} \sum_{n_u \in \tilde{\mathcal{S}}_{n_j}} P(n_u).l(n_u) - P(n_i).l(n_i)\end{aligned}$$

et :

$$\Delta l(\mathcal{S}_{n_j}) = \sum_{n_u \in \tilde{\mathcal{S}}_{n_j}} P(n_u).l(n_u) - P(n_j).l(n_j) \Longleftrightarrow \sum_{n_u \in \tilde{\mathcal{S}}_{n_j}} P(n_u).l(n_u) = \Delta l(\mathcal{S}_{n_j}) + P(n_j).l(n_j)$$

donc :

$$\Delta l(\mathcal{S}_{n_i}) = \sum_{B} \Delta l(\mathcal{S}_{n_j}) + \sum_{B} P(n_j).l(n_j) - P(n_i).l(n_i) \qquad \text{(C.Q.F.D)}$$

C.2 Formules pour la mise à jour des retours marginaux après chaque élagage

La branche $\mathcal{S}^j_{n_i}$ plantée en n_i ayant été élaguée lors de la boucle j de l'algorithme, il faut remettre à jour les retours marginaux des ascendants n_k de n_i. Si $\mathcal{S}^{j+1}_{n_k}$ est un sous-arbre élagué du sous-arbre $\mathcal{S}^j_{n_k}$:

$$\begin{aligned}d(\mathcal{S}^j_{n_k}) &= \sum_{n_u \in \tilde{\mathcal{S}}^j_{n_k}} P(n_u).d(n_u) \\ &= \sum_{n_u \in \tilde{\mathcal{S}}^j_{n_i}} P(n_u).d(n_u) + \sum_{n_u \notin \tilde{\mathcal{S}}^j_{n_i},\ n_u \in \tilde{\mathcal{S}}^j_{n_k}} P(n_u).d(n_u) \\ &= \sum_{n_u \in \tilde{\mathcal{S}}^j_{n_i}} P(n_u).d(n_u) + \sum_{n_u \in \tilde{\mathcal{S}}^{j+1}_{n_k}} P(n_u).d(n_u) - P(n_i).d(n_i)\end{aligned}$$

donc :

$$\begin{aligned}d(\mathcal{S}^j_{n_k}) &= d(\mathcal{S}^{j+1}_{n_k}) + d(\mathcal{S}^j_{n_i}) - P(n_i).d(n_i) \\ &= d(\mathcal{S}^{j+1}_{n_k}) - \Delta d(\mathcal{S}^j_{n_i})\end{aligned}$$

soit encore :

$$P(n_k).d(n_k) - \Delta d(\mathcal{S}^{j+1}_{n_k}) = P(n_k).d(n_k) - \Delta d(\mathcal{S}^j_{n_k}) + \Delta d(\mathcal{S}^j_{n_i}) \Longleftrightarrow \Delta d(\mathcal{S}^{j+1}_{n_k}) = \Delta d(\mathcal{S}^j_{n_k}) - \Delta d(\mathcal{S}^j_{n_i})$$

(C.Q.F.D)

En procédant de la même façon :

$$\begin{aligned}
l(\mathcal{S}^j_{n_k}) &= \sum_{n_u \in \tilde{\mathcal{S}}^j_{n_k}} P(n_u).l(n_u) \\
&= \sum_{n_u \in \tilde{\mathcal{S}}^j_{n_i}} P(n_u).l(n_u) + \sum_{n_u \notin \tilde{\mathcal{S}}^j_{n_i},\ n_u \in \tilde{\mathcal{S}}^j_{n_k}} P(n_u).l(n_u) \\
&= \sum_{n_u \in \tilde{\mathcal{S}}^j_{n_i}} P(n_u).l(n_u) + \sum_{n_u \in \tilde{\mathcal{S}}^{j+1}_{n_k}} P(n_u).l(n_u) - P(n_i).l(n_i) \\
&= l(\mathcal{S}^{j+1}_{n_k}) + \sum_{n_u \in \tilde{\mathcal{S}}^j_{n_i}} P(n_u).l(n_u) - P(n_i).l(n_i) \\
&= l(\mathcal{S}^{j+1}_{n_k}) + l(\mathcal{S}^j_{n_i}) - P(n_i).l(n_i) \\
&= l(\mathcal{S}^{j+1}_{n_k}) + \Delta l(\mathcal{S}^j_{n_i})
\end{aligned}$$

soit encore :

$$P(n_k).l(n_k)+\Delta d(\mathcal{S}^{j+1}_{n_k}) = P(n_k).l(n_k)+\Delta d(\mathcal{S}^j_{n_k})-\Delta d(\mathcal{S}^j_{n_i}) \Longleftrightarrow \Delta l(\mathcal{S}^{j+1}_{n_k}) = \Delta l(\mathcal{S}^j_{n_k})-\Delta l(\mathcal{S}^j_{n_i})$$

(C.Q.F.D)

Annexe D

Expression des formules pour le découpage

Nous développons les formules intervenant pour la mise à jour de la distorsion et du débit du sous-arbre B-aires élagué $\mathcal{S}^j$. A l'issue de la boucle j ($j \geq 1$) du processus de découpage, seule la branche $\mathcal{S}^j_{n_i}$ a été conservée. Nous avons :

$$\begin{aligned} d(\mathcal{S}^j) &= \sum_{n_u \in \tilde{\mathcal{S}}^j} P(n_u).d(n_u) \\ &= \sum_{n_u \in \tilde{\mathcal{S}}^{j-1}} P(n_u).d(n_u) + \sum_{n_u \in \tilde{\mathcal{S}}^j_{n_i}} P(n_u).d(n_u) - P(n_i).d(n_i) \\ &= d(\mathcal{S}^{j-1}) - \Delta d(\mathcal{S}^j_{n_i}) \end{aligned}$$

$$\begin{aligned} l(\mathcal{S}^j) &= \sum_{n_u \in \tilde{\mathcal{S}}^j} P(n_u).l(n_u) \\ &= \sum_{n_u \in \tilde{\mathcal{S}}^{j-1}} P(n_u).l(n_u) + \sum_{n_u \in \tilde{\mathcal{S}}^j_{n_i}} P(n_u).l(n_u) - P(n_i).l(n_i) \\ &= l(\mathcal{S}^{j-1}) + \Delta l(\mathcal{S}^j_{n_i}) \end{aligned}$$

(C.Q.F.D)

Annexe E

Calcul de retours marginaux et emboîtage

Nous illustrons simplement l'avantage de $\mathbb{Z}^k$ (dont l'emboîtage est optimal) sur les autres RRP (dont les emboîtages sont sous-optimaux) dans le cadre de la QVAA. Nous calculons alors les retours marginaux dans le cas du découpage du réseau cubique $\mathbb{Z}^2$, et celui du découpage du réseau hexagonal A_2. Ce dernier est choisi car c'est le réseau meilleur quantificateur pour cette dimension (voir le chapitre 3). La source est uniformément distribuée dans les régions grisées des figures E.1 et E.2. Afin de comparer les résultats de hausse de débit et de baisse de distorsion après découpage, ces zones grisées sont considérées de même aire.

Cas du réseau cubique $\mathbb{Z}^2$

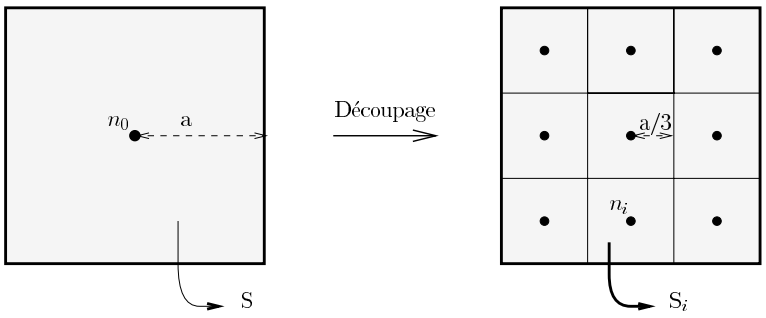


FIG. E.1 – *Découpage du réseau cubique.*

La densité de probabilité uniforme s'exprime :

$$p(x,y) = \frac{1}{S} \quad \text{si } (x,y) \in S \ ; \ \ 0 \text{ sinon}$$

Nous choisissons S = 4 (*i.e.* $a = 1$). Nous avons pour apprécier la distorsion :

$$\int_{-a}^{a}\int_{-a}^{a}(x^2+y^2)dx.dy = \frac{8}{3}.a^4 = f(a)$$

Nous calculons pour les noeuds de la figure E.1 (les définitions sont donnés au chapitre 4) :

$$\begin{array}{ll} P(n_0) & = 1 \\ l(n_0) & = 0 \\ d(n_0) & = f(1) = 8/3 \end{array}$$

$$\begin{array}{ll} P(n_i) & = 1/9 \\ l(n_i) & = \log_2 9 \\ d(n_i) & = 9.f(1/3) = 72/243 \end{array}$$

La distorsion moyenne et le débit moyen associés aux n_i sont :

$$\begin{array}{lll} d(\mathcal{S}) & = 9.P(n_i).d(n_i) & = 72/243 \\ l(\mathcal{S}) & = 9.P(n_i).l(n_i) & = \log_2 9 \end{array}$$

Si nous découpons le noeud n_0, la hausse de débit et la baisse de distorsion sont donc :

$$\begin{array}{llll} \Delta l(\mathcal{S}) & = l(\mathcal{S}) - l(n_0) & = \log_2 9 & \approx 3.17 \\ \Delta d(\mathcal{S}) & = d(n_0) - d(\mathcal{S}) & = 8/3 - 72/243 & \approx 2,37 \end{array}$$

Cas du réseau hexagonal A_2

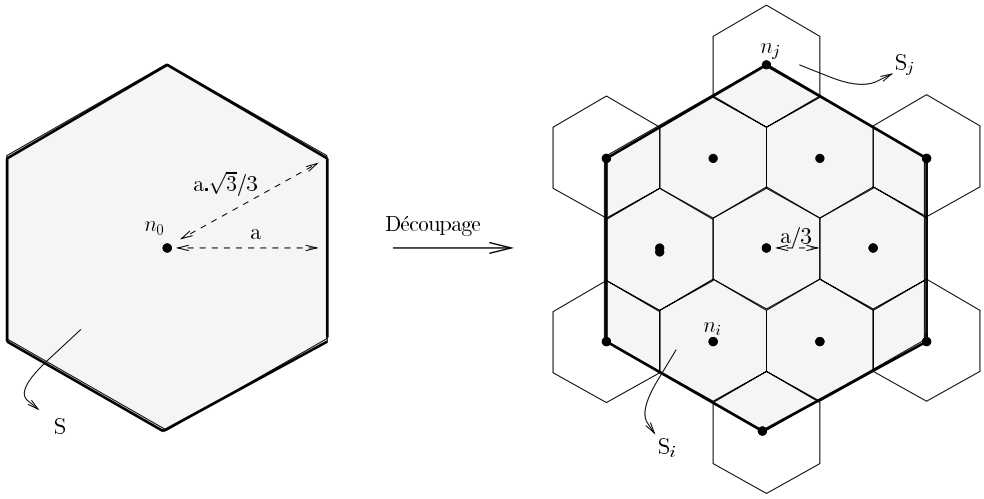


FIG. E.2 – *Découpage du réseau hexagonal.*

Les deux surfaces S (pour le carré et l'hexagone) doivent-être égales alors $a = (\frac{\sqrt{3}}{6}.S)^{1/2}$ avec $S = 4$

Nous avons pour apprécier la densité de probabilité et la distorsion :

$$\int_0^a \int_0^{x.\sqrt{3}/3} dx.dy = a^2.\frac{\sqrt{3}}{6} = g(a)$$

$$\int_0^a \int_0^{x.\sqrt{3}/3} (x^2+y^2)dx.dy = a^4.\frac{5.\sqrt{3}}{54} = h(a)$$

Nous calculons pour le noeud n_0 :

$$\begin{array}{ll} P(n_0) & = 1 \\ l(n_0) & = 0 \\ d(n_0) & = 12.h(a) \approx 2,567 \end{array}$$

Pour les 7 Voronoï entiers après le découpage (*i.e.* les n_i) :

$$\begin{array}{lll} P(n_i) & = g(a/3).12/4 & = 1/9 \\ l(n_i) & = \log_2 9 & \\ d(n_i) & = 12.9.h(a/3) & \approx 0,285 \end{array}$$

Pour les 6 Voronoï non-entiers après le découpage (*i.e.* les n_j) :

$$\begin{array}{lll} P(n_j) & = P(n_i)/3 & = 1/27 \\ l(n_i) & = \log_2 27 & \\ d(n_i) & = 27.4.h(a/3) & \approx 0,283 \end{array}$$

La distorsion moyenne et le débit moyen associés aux nouveaux noeuds sont :

$$\begin{array}{lll} d(\mathcal{S}) & = 7.P(n_i).d(n_i) + 6.P(n_j).d(n_j) & \approx 0,285 \\ l(\mathcal{S}) & = 7.P(n_i).l(n_i) + 6.P(n_j).l(n_j) & \approx 3,522 \end{array}$$

Si nous découpons le noeud n_0, la hausse de débit et la baisse de distorsion sont donc :

$$\begin{array}{lll} \Delta l(\mathcal{S}) & = l(\mathcal{S}) & \approx 3.52 \\ \Delta d(\mathcal{S}) & = d(n_0) - d(\mathcal{S}) & \approx 2,28 \end{array}$$

Annexe F

Description du QS de type MPEG pour des images prédites

Nous décrivons le QS de type MPEG pour la quantification des images prédites, que nous mettons en oeuvre à des fins comparatives au chapitre 6.

Si (u, v) indice la position du coefficient $F(u, v)$ dans le bloc à coder, $C(u, v)$ est le coefficient correspondant quantifié, $Q(u, v) = 16$ et Q_f (un paramètre introduit afin de contrôler le débit) définissent le pas du quantificateur. La quantification est la suivante (où Ent indique la partie entière) :

$$A(u, v) = Ent\left(\frac{F(u, v) \times 16 \pm Q(u, v)/2}{Q(u, v)}\right)$$

et

- si Q_f est impair ou nul :

$$C(u, v) = Ent\left(\frac{A(u, v)}{2 \times Q_f}\right)$$

- si Q_f est pair :

$$C(u, v) = Ent\left(\frac{A(u, v) \pm 1}{2 \times Q_f}\right)$$

  où $\pm$ est positif si $A(u, v)$ est strictement positif, et négatif si $A(u, v)$ est strictement négatif.

La quantification inverse est alors :

$$\hat{F}(u, v) = \frac{(2 \times C(u, v) \pm 1) \times Q_f \times Q(u, v)}{16}$$

où $\pm$ est positif si $C(u, v)$ est strictement positif, et négatif pour $C(u, v)$ strictement négatif. Si $C(u, v) = 0$ alors $F(u, v) = 0$.

Il y a création d'une "dead zone" autour du point 0. A titre d'exemple, la figure F.1 illustre les caractéristiques en marche d'escalier du QS pour deux valeurs de Q_f.

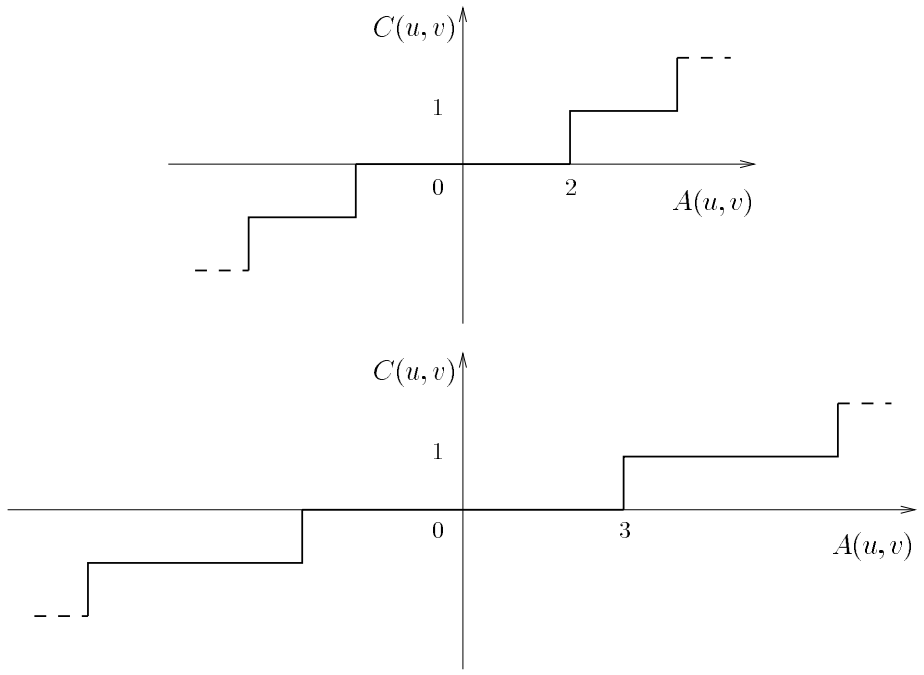


FIG. F.1 – *Caractéristiques de deux QS, pour $Q_f = 1$ (en haut), et $Q_f = 2$ (en bas).*

Bibliographie

[Anderberg73] Anderberg (M.R.). – *Cluster Analysis for applications.* – New York, Academic Press, 1973.

[Antonini et al.92] Antonini (M.), Barlaud (M.), Mathieu (P.) et Daubechies (I.). – Image coding using wavelet transform. *IEEE Transactions on Image Processing*, vol. 2, avril 1992.

[Antonini et al.95] Antonini (M.), Raffy (P.) et Barlaud (M.). – Toward entropy constrained lattice vector quantization. *Proc. of International Conference on Image Processing ICIP.* – octobre 1995.

[Antonini91] Antonini (M.). – *Transformée en ondelettes et compression numérique des images.* – Thèse, Université de Nice-Sophia Antipolis, septembre 1991.

[Baaziz91] Baaziz (N.). – *Approches d'estimation et de compensation de mouvements multirésolutions pour le codage de séquences d'images.* – Thèse, Université de Rennes I, 1991.

[Bage86] Bage (M.J.). – Lattice quantizers : Entropy reduction by proper tie-handling. *IEEE Transactions on Information Theory*, vol. 32, n° 2, mars 1986, pp. 328–330.

[Balakrishnan et al.95] Balakrishnan (M.), Pearlman (W.A.) et Lu (L.). – Variable-rate tree-structured vector quantizers. *IEEE Transactions on Information Theory*, vol. 41, n° 4, juillet 1995, pp. 917–930.

[Barlaud et al.94] Barlaud (M.), Solé (P.), Gaidon (T.), Antonini (M.) et Mathieu (P.). – Pyramidal lattice vector quantization for multiscale image coding. *IEEE Transactions on Image Processing*, vol. 3, n° 4, juillet 1994, pp. 367–381.

[Barlaud94] Barlaud (M). – *Wavelets in image communication.* – New York, Elsevier, 1994.

[Barnes et al.93] Barnes (C.F.) et Frost (R.L.). – Vector quantizers with direct sum codebooks. *IEEE Transactions on Information Theory*, vol. 39, n° 2, mars 1993, pp. 565–580.

[Barnes et al.96] Barnes (C.F.), Rizvi (S.A.) et Nasser (N.M.). – Advances in residual vector quantization : A review. *IEEE Transactions on Image Processing*, vol. 5, n° 2, février 1996, pp. 226–262.

[Barron85] Barron (A.R.). – The strong ergodic theorem for densities : generalized shannon-mcmillan-breiman theorem. *Ann. Probab.*, vol. 13, 1985, pp. 1292–1303.

[Benazza92] Benazza (A.). – *Quantification Vectorielle en Codage d'Images.* – Thèse, Université de Paris XI Orsay, 1992.

[Berger71] Berger (T.). – *Rate Distortion Theory : a mathematical basis for data compression.* – Englewood Cliffs, New Jersey, Prentice Hall International Editions, 1971.

[Berger82] Berger (T.). – Minimum entropy quantizers and permutation codes. *IEEE Transactions on Information Theory*, vol. 28, mars 1982.

[Blahut72] Blahut (R.E.). – Computation of channel capacity and rate distortion functions. *IEEE Transactions on Information Theory*, juillet 1972, pp. 460–473.

[Blahut87] Blahut (R.E.). – *Principles and practise of information theory.* – Addison-Wesley, 1987.

[Breiman et al.84] Breiman (L.), Friedman (J.H.), Olshen (R.A) et Stone (C.J.). – *Classification and regression Trees.* – Wadsworth, Belmont, California, 1984, *The Wadsworth Statistics/Probality Series.*

[Budge et al.85] Budge (S.E.) et Baker (R.L.). – Compression of color digital images using vector quantization in product codes. *IEEE Transactions on Acoust. Speech Signal Processing*, avril 1985.

[Buzo et al.80] Buzo (A.), Gray Jr. (A.H.) et Markel (J.D.). – Speech coding based upon vector quantization. *IEEE Transactions on Acoust. Speech Signal Processing*, pp. 562–574. – octobre 1980.

[Ccir82] CCIR. – 601 CCIR Recommendation - *Encoding parameters of digital television for studios.* – Rapport technique, ITU Suisse, 1982.

[Ccitt80] CCITT. – Normalisation des télécopieurs du groupe 3 pour la transmission de documents, fascicule VII.3, recommandation T.4, 1980.

[Chan et al.92] Chan (W.Y.), Gupta (S.) et Gersho (A.). – Enhanced multistage vector quantization by joint codebook design. *IEEE Transactions*

on Information Theory, vol. 40, n° 11, novembre 1992, pp. 1693–1697.

[Chou et al.89a] Chou (P.A.), Lookabaugh (T.) et Gray (R.M.). – Entropy-constrained vector quantization. *IEEE Transactions on Acoust. Speech Signal Processing*, vol. 37, n° 1, janvier 1989, pp. 31–42.

[Chou et al.89b] Chou (P.A.), Lookabaugh (T.) et Gray (R.M.). – Optimal pruning with applications to tree-structured source coding and modeling. *IEEE Transactions on Information Theory*, vol. 35, n° 2, mars 1989, pp. 299–314.

[Clarke85] Clarke (R.J.). – *Transform coding of images*. – New York, Academic Press, 1985.

[Conway et al.82a] Conway (J.H.) et N.J.A. (Sloane). – Fast quantizing and decoding algorithms for lattice quantizers and codes. *IEEE Transactions on Information Theory*, vol. 28, n° 2, mars 1982, pp. 227–232.

[Conway et al.82b] Conway (J.H.) et N.J.A. (Sloane). – Voronoï regions of lattices, second moments of polytopes, and quantization. *IEEE Transactions on Information Theory*, vol. 28, n° 2, mars 1982, pp. 211–226.

[Conway et al.83] Conway (J.H.) et N.J.A. (Sloane). – A fast encoding method for lattice codes and quantizers. *IEEE Transactions on Information Theory*, vol. 29, n° 6, novembre 1983, pp. 820–824.

[Conway et al.85] Conway (J.H.) et N.J.A. (Sloane). – A lower bound on the average error of vectors quantizers. *IEEE Transactions on Information Theory*, vol. 31, n° 1, janvier 1985, pp. 106–109.

[Conway et al.93] Conway (J.H.) et N.J.A. (Sloane). – *Sphere Packings, Lattices and Groups, 2nd edition*. – New York, Springer-Verlag, 1993, *A series of Comprehensive Studies in Mathematics*.

[Cosman et al.96] Cosman (P.C.), Gray (R.M.) et Vetterli (M.). – Vector quantization of image subbands : A survey. *IEEE Transactions on Image Processing*, vol. 5, n° 2, février 1996, pp. 202–225.

[Daubechies88] Daubechies (I.). – Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math*, vol. 41, 1988, pp. 909–996.

[Daubechies90] Daubechies (I.). – The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, vol. 36, n° 5, septembre 1990, pp. 961–1005.

[Davoine95] Davoine (F.). – *Compression d'images par fractales basée sur la triangulation de Delaunay*. – Thèse, Institut national polytechnique de Grenoble, décembre 1995.

[Delogne et al.91] Delogne (P.) et Macq (B.). – Universal variable length coding for an integrated approach to image coding. *Annales des télécommunications*, vol. 46, n° 7-8, 1991.

[Demaistre et al.96] Demaistre (N.) et Labit (C.). – Progressive image transmission using wavelet packets. *Proc. of International Conference on Image Processing ICIP*. Lauzanne, Suisse. – septembre 1996.

[Driessen et al.90] Driessen (J.N.), Belfor (R.A.F.) et Biemond (J.). – Backward predictive motion compensated image sequence coding. *Proc. of European Signal Processing Conference EUSIPCO*, pp. 757–760. – septembre 1990.

[Eriksson94] Eriksson (T.). – Mutistage vector quantization with dynamic bit allocation. *Proc. of European Signal Processing Conference EUSIPCO*. Edinburgh, Scotland, pp. 383–386. – septembre 1994.

[Favardin et al.84] Favardin (N.) et Modestino (J.W.). – Optimum quantizer performance for a class of non-gaussian memoryless sources. *IEEE Transactions on Information Theory*, vol. 30, mai 1984.

[Fiche et al.94] Fiche (P.), Ricordel (V.) et Labit (C.). – *Etude d'algorithmes de quantification vectorielle arborescente pour la compression d'images fixes*. – Rapport technique n° 2241, INRIA, janvier 1994.

[Fisher86] Fisher (T.R.). – A pyramid vector quantizer. *IEEE Transactions on Information Theory*, vol. 32, n° 4, juillet 1986, pp. 568–583.

[Fisher89] Fisher (T.R.). – Geometric source coding and vector quantization. *IEEE Transactions on Information Theory*, vol. 35, n° 1, janvier 1989, pp. 137–145.

[Foster et al.85] Foster (J.), Gray (R.M.) et M. (Ostendorf Dunahm). – Finite-state vector quantization for waveform coding. *IEEE Transactions on Information Theory*, vol. 31, mai 1985.

[Gaidon93] Gaidon (T.). – *Quantification vectorielle algébrique et ondelettes pour la compression de séquences d'images*. – Thèse, Université de Nice-Sophia Antipolis, décembre 1993.

[Gersho et al.92] Gersho (A.) et Gray (R.M.). – *Vector Quantization and Signal Compression*. – Boston, Kluwer Academic Publishers, 1992.

[Gersho79] Gersho (A.). – Asymptotically optimal block quantization. *IEEE Transactions on Information Theory*, vol. 25, n° 4, juillet 1979, pp. 373–380.

[GG95] Garcia Garduno (V.). – *Une approche de compression orientée-objets par suivi de segmentation basée mouvement pour le codage de séquences d'images numériques.* – Thèse, Université de Rennes I, 1995.

[Ghazzali92] Ghazzali (N.). – *Comparaison et réduction d'arbres de classification, en relation avec des problèmes de quantification en imagerie numérique.* – Thèse, Université de Rennes I, 1992.

[GJ et al.76] Gray Jr. (A.H.) et Markel (J.D.). – Distance measures for speech processing. *IEEE Transactions on Acoust. Speech Signal Processing*, vol. 24, n° 5, octobre 1976.

[Goldberg et al.86] Goldberg (M.), Boucher (P.R.) et Schlien (S.). – Image compression using adaptative vector quantization. *IEEE Transactions on Communications*, vol. 34, n° 2, février 1986, pp. 180–187.

[Gray84] Gray (R.M.). – Vector quantization. *IEEE ASSP Magazine*, avril 1984, pp. 4–29.

[Gray90] Gray (R.M). – *Source coding theory.* – Kluwer Academic Publishers, 1990.

[Howard et al.94] Howard (P.G.) et Vitter (J.S.). – Arithmetic coding for data compression. *Proc. of the IEEE*, vol. 82, n° 6, 1994, pp. 857–865.

[Huffman52] Huffman (D.A.). – A method for the construction of minimum-redundancy codes. *Proc. of the IRE*, pp. 1098–1101. – 1952.

[Hwang et al.95] Hwang (W.) et Derin (H.). – Multiresolution multiresource progressive image transmission. *IEEE Transactions on Image Processing*, vol. 4, n° 8, août 1995, pp. 1128–1140.

[Isoiec93] ISO/IEC. – Norme ISO/IEC 11172, codage de l'image animée et du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s, 1993.

[Isoiec94] ISO/IEC. – Recommendation ISO/IEC 1388, coding of moving picture and associated audio, mars 1994.

[Jain et al.88] Jain (A.K.) et Dubes (R.C.). – *Algorithms for clustering data.* – Englewood Cliffs, New Jersey, Prentice-Hall, 1988.

[Jain79] Jain (A.). – A fast karhunen-loeve transform for finite discrete images. *Proc. of Nat. Electronics Conference.* – octobre 1979.

[Jain81] Jain (A.K.). – Image data compression : A review. *Proc. of the IEEE*, vol. 69, n° 3, mars 1981, pp. 349–389.

[Jain89] Jain (A.K.). – *Fundamentals of Digital Image Processing.* – Englewood Cliffs, New Jersey, Prentice Hall International Editions, 1989.

[Jayant et al.84] Jayant (N.S.) et P. (Noll). – *Digital Coding of Waveforms - Principles and Applications to Speech and Video.* – Englewood Cliffs, New Jersey, Prentice Hall International Editions, 1984.

[Jeong et al.93] Jeong (D.G.) et Gibson (J.D.). – Uniform and piecewise uniform lattice vector quantization for memoryless gaussian and laplacian sources. *IEEE Transactions on Information Theory*, vol. 39, n° 3, mai 1993.

[Jeong et al.95] Jeong (D.G.) et Gibson (J.D.). – Image coding with uniform and piecewise-uniform vector quantizers. *IEEE Transactions on Image Processing*, vol. 4, n° 2, février 1995, pp. 140–146.

[Kiang et al.92] Kiang (S.Z.), Baker (R.L.), Sullivan (G.L.) et Chiu (C.Y.). – Recursive optimal pruning with applications to tree structured vector quantizers. *IEEE Transactions on Image Processing*, vol. 1, avril 1992, pp. 162–169.

[Kohonen89] Kohonen (T.). – *Self-organization and Associative Memory, 3rd edition.* – New York, Springer-Verlag, 1989.

[Kossentini et al.93] Kossentini (K.), Smith (M.) et Barnes (C.). – Entropy-constrained residual vector quantization. *Proc. of International Conference on Acoustics, Speech, and Signal Processing ICASSP.* Minneapolis, USA, pp. 598–601. – avril 1993.

[Kossentini et al.95] Kossentini (F.), Smith (M.J.T.) et Barnes (C.F.). – Necessary conditions for the optimality of variable-rate residual vector quantizers. *IEEE Transactions on Information Theory*, vol. 41, n° 6, novembre 1995, pp. 1903–1914.

[Kuhlmann et al.88] Kuhlmann (F.) et Bucklew (J.A.). – Piecewise uniform vector quantizer. *IEEE Transactions on Information Theory*, vol. 34, n° 5, septembre 1988.

[Kunt et al.85] Kunt (M.), Ikonomopoulos (A.) et Kocher (M.). – Second-generation image-coding techniques. *Proc. of the IEEE*, vol. 73, n° 4, avril 1985, pp. 549–574.

[Kunt et al.93] Kunt (M.), Goesta (G.) et Kocher (M.). – *Traitement de l'information: 2: traitement numériques des images.* – Presses polytechniques et universitaires romandes, 1993.

[Kunt81] Kunt (M.). – *Traitements Numériques des signaux.* – Dunod, 1981.

[Lamblin et al.88] Lamblin (C.) et Adoul (J.P.). – Algorithme de quantification vectorielle sphérique à partir du réseau de gosset d'ordre 8. *Ann. Télécommun.*, vol. 43, n° 3-4, 1988, pp. 176–186.

[Lebedeff95] Lebedeff (D.). – *Etude de la quantification vectorielle des données brutes issues d'un radar à synthèse d'ouverture.* – Thèse, Université de Nice-Sophia Antipolis, décembre 1995.

[Lee et al.94] Lee (W.) et C. (Chan.). – Dynamic finite-state VQ of colour images using stochastic learning. *Signal Processing: Image Communications*, vol. 1, 1994.

[Legall91] Legall (D.). – MPEG : a video compression standart for multimedia applications. *Communications of the ACM*, vol. 34, n° 4, avril 1991, pp. 46–58.

[Levent94] Levent (F.). – *Transmission vidéo numérique en multirésolution. Optimisation des paramètres de codage et de modulation.* – Thèse, Université de Valanciennes et du Hainaut-Cambresis, 1994.

[Li et al.94] Li (H.), Lundmark (A.) et Forchheimer (R.). – Image sequence coding at very low birates : A review. *IEEE Transactions on Image Processing*, vol. 3, n° 5, septembre 1994, pp. 589–609.

[Linde et al.80] Linde (Y.), Buzo (A.) et Gray (R.M.). – An algorithm for vector quantizer design. *IEEE Transactions on Communications*, vol. 28, 1980, pp. 84–95.

[Liou91] Liou (M.). – Overview of the px64 kbit/s video coding standart. *Communications of the ACM*, vol. 34, n° 4, avril 1991, pp. 59–63.

[Lloyd82] Lloyd (S.P.). – Least squares quantization in PCM. *IEEE Transactions on Information Theory*, vol. 28, n° 2, mars 1982, pp. 129–137.

[Lookabaugh et al.89] Lookabaugh (T.D.) et Gray (R.M.). – High-resolution quantization theory and the vector quantizer advantage. *IEEE Transactions on Information Theory*, vol. 35, n° 5, septembre 1989, pp. 1020–1033.

[Macqueen67] MacQueen (J.). – Some methods for classification and analysis of multivariate observations. *Proc. of the Fifth Berkeley Symposium on Math. Stat. and Prob.*, pp. 281–296. – 1967.

[Makhoul et al.85] Makhoul (J.), Roucos (S.) et Gish (H.). – Vector quantization in speech coding. *Proc. of the IEEE*, vol. 73, n° 11, novembre 1985, pp. 1551–1588.

[Mallat89] Mallat (S.). – A theory for multiresolution signal decomposition : the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, n° 7, juillet 1989, pp. 674–693.

[Malvar92] Malvar (H.S.). – *Signal processing with lapped transforms.* – Artech House, 1992.

[Maresq86] Maresq (J.P.). – *Etude de schémas de quantification vectorielle adaptative multiclasses. Application au codage de séquences d'images télévisuelles.* – Thèse, Université de Rennes I, décembre 1986.

[Max60] Max (J.). – Quantizing for minimum distorsion. *IEEE Transactions on Information Theory*, vol. 6, mars 1960, pp. 7–12.

[Mermelstein88] Mermelstein (P.). – G.722, a new CCITT standart for digital transmission of wideband audio signals. *IEEE Communication Magazine*, vol. 26, n° 1, janvier 1988.

[Meyer90] Meyer (Y.). – *Ondelettes et opérateurs.* – Paris, Hermann, 1990.

[Monet et al.90] Monet (P.) et Labit (C.). – Codebook replenishment in classified pruned tree-structured vector quantization of image sequences. *Proc. of International Conference on Acoustics, Speech, and Signal Processing ICASSP*, pp. 2285–2288. – 1990.

[Montrichard et al.96a] Montrichard (M.), Ricordel (V.) et Labit (C.). – *Compression vidéo type MPEG et quantification vectorielle algébrique et arborescente : allocation binaire et analyse du codage.* – Rapport technique, IRISA, septembre 1996. Rapport de DEA.

[Montrichard et al.96b] Montrichard (M.), Ricordel (V.) et Labit (C.). – *Compression vidéo type MPEG et quantification vectorielle.* – Rapport technique, IRISA, juin 1996. Rapport d'ingénieur.

[Moreau95] Moreau (N.). – *Techniques de compression des signaux.* – Paris, Masson, 1995, *Collection technique et scientifique des télécommunications.*

[Moureaux et al.94] Moureaux (J.M.), Antonini (M.) et Barlaud (M.). – Lattice vector quantization of image using a product code form and a new labelling method. *Proc. of Visual Communications and Image Processing VCIP*, pp. 422–433. – septembre 1994.

[Moureaux94] Moureaux (J.M.). – *Quantification vectorielle algébrique pour la compression d'images. Application à l'imagerie radar à synthèse d'ouverture (SAR).* – Thèse, Université de Nice-Sophia Antipolis, décembre 1994.

[Nasrabadi et al.88a] Nasrabadi (N.M.) et Feng (Y.). – Vector quantization of images based upon the kohonen self-organizing feature maps. *Proc. of International Neural Network Conference*, 1988, pp. 101–108.

[Nasrabadi et al.88b] Nasrabadi (N.M.) et King (R.A.). – Image coding using vector quantization : a review. *IEEE Transactions on Communications*, vol. 36, n° 18, août 1988.

[Netravali et al.79] Netravali (A.N.) et Robbins (J.D). – Motion compensated television coding part 1. *Bell System Technical Journal*, vol. 58, n° 3, 1979, pp. 629–668.

[Netravali et al.88] Netravali (A.N.) et Haskell (B.G.). – *Digital pictures : Representation and Compression.* – New York, Plenum Press, 1988.

[Newman84] Newman (D.J.). – The hexagon theorem. *IEEE Transactions on Information Theory*, mars 1984.

[Nguyen95] Nguyen (E.). – *Compression sélective et focalisation visuelle : application au codage hybride de séquences d'images.* – Thèse, Université de Rennes I, 1995.

[Nicolas92] Nicolas (H). – *Hiérarchie de modèles de mouvement et méthodes d'estimation associées. Application au codage de séquences d'images.* – Thèse, Université de Rennes I, 1992.

[Nzomigni95] Nzomigni (V.). – *Compression sans pertes de séquences d'images biomédicales.* – Thèse, Université de Rennes I, décembre 1995.

[Onno et al.95] Onno (P.) et Guillemot (C.). – Quantification vectorielle algébrique contrainte en débit : Un nouvel algorithme de quantification au sens débit-distorsion. *Proc. of Colloque GRETSI.* Juan-les-Pins, France. – septembre 1995.

[Onno96] Onno (P.). – *Bancs de filtres et quantification vectorielle sur réseau : étude conjointe pour la compression d'images.* – Thèse, Université de Rennes I, mars 1996.

[Pan et al.95] Pan (J.) et Fisher (T.R.). – Two-stage vector quantization-lattice vector quantization. *IEEE Transactions on Information Theory*, vol. 41, n° 1, janvier 1995, pp. 155–163.

[Pateux et al.96] Pateux (S.) et Labit (C.). – *Codage efficace de cartes de segmentation pour la compression orientée-régions de séquences d'images.* – Rapport technique, IRISA, à paraitre en décembre 1996.

[Pennebaker et al.93] Pennebaker (W.B.) et Mitchell (J.L.). – *JPEG : still image data compression standart.* – New York, Van Nostrand Reinhold, 1993.

[Pereira96] Pereira (F.). – MPEG4 : a new challenge for the representation of audio-visual information. *Proc. of international Picture Coding Symposium PCS.* Melbourne, Australia. – mars 1996.

[Perlmutter et al.96] Perlmutter (K.O.), Perlmutter (S.M.), Gray (R.M.), Olshen (R.A.) et Oehler (K.L.). – Bayes risk weighted vector quantization with posterior estimation for image compression and classification. *IEEE Transactions on Image Processing*, vol. 5, n° 2, février 1996, pp. 347–360.

[Proakis89] Proakis (J.G.). – *Digital communications.* – New York, McGraw-Hill, 1989.

[Rabbani et al.91] Rabbani (M.) et Jones (P.W.). – *Digital image compression techniques.* – Bellingham : SPIE, 1991.

[Ramchandran et al.93] Ramchandran (K.) et Vettrerli (M.). – Best wavelet packet bases in a rate-distorsion sense. *IEEE Transactions on Image Processing*, vol. 2, n° 2, avril 1993.

[Ricordel et al.95a] Ricordel (V.) et Labit (C). – Quantification vectorielle par emboîtage de réseaux réguliers de points. *Proc. of Colloque GRETSI.* Juan-les-Pins, France. – septembre 1995.

[Ricordel et al.95b] Ricordel (V.) et Labit (C). – Vector quantization by hierarchical packing of embedded truncated lattices. *Proc. of Visual Communications and Image Processing VCIP.* Taiwan, Chine. – mai 1995.

[Ricordel et al.95c] Ricordel (V.) et Labit (C). – Vector quantization by packing of embedded truncated lattices. *Proc. of International Conference on Image Processing ICIP.* Washington DC, USA, pp. 292–295. – octobre 1995.

[Ricordel et al.96a] Ricordel (V.) et Labit (C). – Quantification vectorielle algébrique et arborescente. *Actes des journées Compression et Représentation des Signaux Audiovisuels CORESA.* Grenoble, France, pp. 135–142. – février 1996.

[Ricordel et al.96b] Ricordel (V.) et Labit (C). – Tree-structured lattice vector quantization. *Proc. of European Signal Processing Conference EUSIPCO.* Trieste, Italie, pp. 731–734. – septembre 1996.

[Rioul93] Rioul (O.). – A discrete-time multiresolution theory. *IEEE Transactions on Signal Processing*, vol. 41, n° 8, août 1993, pp. 2591–2606.

[Riskin et al.91] Riskin (E.A.) et Gray (R.M). – A greedy tree growing algorithm for the design of variable rate vector quantizers. *IEEE Transactions on Signal Processing*, vol. 39, n° 11, novembre 1991, pp. 2500–2507.

[Rose et al.96] Rose (K.), Miller (D.) et Gersho (A.). – Entropy-constrained tree-structured vector quantizer design. *IEEE Transactions on Image Processing*, vol. 5, n° 2, février 1996, pp. 393–398.

[Sabin et al.84] Sabin (M.J.) et Gray (R.M.). – Product code vector quantizers for waveform and voice coding. *IEEE Transactions on Acoust. Speech Signal Processing*, vol. 32, juin 1984.

[Sayood et al.84] Sayood (K.), Gibson (J.D.) et Frost (M.C.). – An algorithm for uniform vector quantizer design. *IEEE Transactions on Information Theory*, vol. 30, 1984.

[Senane96] Sénane (H.). – *Représentation d'images en sous-bandes visuelles. Application au codage d'images de télévision sans défauts visibles.* – Thèse, Université de Nantes, 1996.

[Shannon48] Shannon (C.E.). – A mathematical theory of communication. *Bell System Technical Journal*, 1948, pp. 379–423, 623–656.

[Shannon59] Shannon (C.E.). – Coding theorems for a discrete source with a fidelity criterion. *IRE National Convention Record, part 4*, 1959, pp. 142–163.

[Shoham et al.88] Shoham (Y.) et Gersho (A.). – Efficient bit allocation for an arbitrary set of quantizers. *IEEE Transactions on Acoust. Speech Signal Processing*, vol. 36, n° 9, septembre 1988, pp. 1445–1453.

[Skowronski96] Skowronski (J.). – *Analyse et compression d'images - Modélisation vectorielle, ,décomposition en sous-bandes et aspects de quantification.* – Thèse, Université de Paris XI Orsay, 1996.

[Smith et al.86] Smith (M.J.) et Barnwell (T.P.). – Exact reconstruction techniques for tree structured subband coders. *IEEE Transactions on Acoust. Speech Signal Processing*, 1986.

[Swaszek92] Swaszek (P.F.). – Unrestricted multistage vector quantizers. *IEEE Transactions on Information Theory*, vol. 38, n° 3, mai 1992, pp. 1169–1174.

[Taubman et al.94] Taubman (D.) et Zakhor (A.). – Multirate 3-d subband coding of video. *IEEE Transactions on Image Processing*, vol. 3, n° 5, septembre 1994, pp. 572–588.

[Tziritas et al.94] Tziritas (G.) et Labit (C.). – *Motion analysis for image sequence coding.* – Londres, Elsevier Science Publishers, juillet 1994, *Advances in image communication*, volume 4.

[Ungerboeck87a] Ungerboeck (G.). – Trellis-coded modulation with redundant signal sets - part i : Introduction. *IEEE Communication Magazine*, vol. 25, n° 2, février 1987.

[Ungerboeck87b] Ungerboeck (G.). – Trellis-coded modulation with redundant signal sets - part ii : State of the art. *IEEE Communication Magazine*, vol. 25, n° 2, février 1987.

[Vaidyanathan90] Vaidyanathan (P.P.). – Multirate digital filters, filter banks, polyphase networks and applications : a tutorial. *Proc. of the IEEE*, janvier 1990.

[Vetterli et al.84] Vetterli (M.) et H.J. (Nussbaumer). – Simple FFT and DCT algorithms with reduce number of operations. *Signal Processing*, vol. 6, n° 4, août 1984, pp. 267–278.

[Vetterli84] Vetterli (M.). – Multi-dimensional subband coding : some theory and algorithmes. *Signal Processing*, vol. 21, n° 4, avril 1984, pp. 267–278.

[Viterbi et al.74] Viterbi (A.J.) et Omura (J.K.). – Trellis encoding of memoryless discrete-time sources with a fidelity criterion. *IEEE Transactions on Information Theory*, vol. 20, 1974.

[Wallace91] Wallace (G.K.). – The JPEG still picture compression standart. *Communications of the ACM*, vol. 34, n° 4, avril 1991, pp. 30–45.

[Wang et al.89] Wang (L.) et Goldberg (M.). – Progressive image transmission using vector quantization on image in pyramid form. *IEEE Transactions on Communications*, vol. 37, décembre 1989, pp. 1339–1349.

[Watson93] Watson (A.B. (ed.)). – *Digital images and human vision.* – Cambridge : MIT, 1993.

[Woods91] Woods (J.W.). – *Subband image coding.* – Boston, Kluwer, 1991.

[Zador82] Zador (P.). – Asymptotic quantization error of continuous signals and their quantization dimension. *IEEE Transactions on Information Theory*, vol. 28, mars 1982, pp. 139–149.

[Zeng et al.95] Zeng (W.J.), Huang (Y.F.) et Huang (S.C.). – Two greedy tree algorithms for designing variable rate vector quantizers. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, n° 3, juin 1995, pp. 236–242.

Tree-structured lattice vector quantization for the compression of digital image sequences

Abstract :
The thesis deals with the design of a new vector quantizer (VQ) which takes place in an hybrid coding scheme for the compression of digital image sequences.
Such a source (vectors are composed from transformed motion-compensated prediction errors) is always a nonstationary signal. A training procedure from representative image sequences is used for designing the VQ codebook, it permits to fit the spatiotemporal source distribution. But the codebook design and the corresponding encoding-decoding algorithm have to be very fast in order to achieve a temporal updating of the reproduction vectors. The computation complexity of usual classification method presents a limitation to their applicability. A lattice VQ, for which encoding is fastest, is adapted only for a signal whose distribution permits to truncate the lattice.
To overcome these drawbacks, we proposed a new lattice VQ based on the hierarchical packing of embedded lattices. The aim of our approach is to unify both efficient coding methods : a fast lattice encoding-decoding ; and an unbalanced tree-structured codebook design according to a distorsion vs. rate tradeoff.
Finally we analyse experimental results with our VQ taking place in a MPEG-based coder, and a region-based coding scheme.

Key-words :
Vector quantization, image sequence compression, very low bit rate coding, lattice VQ, tree-structured VQ, subband coding, bit allocation.

Etude de schémas de quantification vectorielle algébrique et arborescente. Application à la compression de séquences d'images numériques

Résumé :
Ce travail de thèse vise à concevoir un nouveau schéma de quantification vectorielle (QV) devant prendre place au sein d'une chaîne de codage hybride pour la compression de séquences d'images. Le but est de contribuer à l'élaboration de futures normes de compression du signal vidéo (MPEG4) et à la conception de nouveaux services de vidéocommunications.
La nature non-stationnaire du signal à coder (des vecteurs d'erreurs de prédiction de compensation du mouvement transformées) conduit à retenir une technique d'apprentissage pour la construction du dictionnaire. Si la condition d'opérations d'encodage-décodage rapides est remplie, une QV adaptative est opérationnelle où, quand cela est nécessaire (changement de plan vidéo), l'actualisation des vecteurs représentants est effectuée en utilisant une séquence d'apprentissage issue de la source courante. Le coût calculatoire des techniques classiques d'apprentissage les rend inadaptées. La QV algébrique, rapide, n'est appropriée que si la statistique de la source autorise une troncature aisée des réseaux.
Notre approche vise alors à tirer profit de deux techniques de codage : une quantification rapide sur réseaux algébriques, la construction d'un dictionnaire arborescent non-équilibré apportant une partition de l'espace adaptée à la distribution de la source et selon un compromis débit-distorsion. Précisément, la technique utilisée consiste en un emboîtement de réseaux tronqués de même nature et aboutit à un schéma de QV multi-étages. Pour une cellule de Voronoï d'un réseau à une résolution donnée, sur un critère local débit-distorsion, il est décidé ou non de descendre au réseau plus fin.
Notre recherche se conclue par l'expérimentation de ce quantificateur vectoriel inscrit au sein de deux types de codeurs : un classique (assemblage d'outils algorithmiques de la famille MPEG), l'autre novateur (codeur orienté régions).

Mots-clés :
Quantification vectorielle, compression de séquences d'images, codage très bas débit, réseaux algébriques, quantification vectorielle arborescente, codage en sous-bandes, allocation binaire.

(An abstract is inside)