Représentation multirésolution et compression d'images : ondelettes et codage scalaire et vectoriel

Az-Eddine Rafaa

THESE

Présentée à l'Université de METZ

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITE EN ELECTRONIQUE

par

Az Eddine RAFAA

REPRESENTATION MULTIRESOLUTION ET COMPRESSION D'IMAGES : ONDELETTES ET CODAGE SCALAIRE ET VECTORIEL

Soutenue le 15 septembre 1994

Président du jury:	M.	A. TOSSER	Professeur à l'ENIM Metz.
Rapporteurs:	M.	A. FAURE	Professeur à l'Université du Havre.
	M.	L. LEGRAND	Maître de Conférence à l'Université de Dijon.
Examinateurs:	M.	M. PAINDAVOINE	Professeur à l'Université de Dijon.
	M.	G. GOVAERT	Professeur à l'Université de Compiègne.
	Mme	M. VALENTIN	Professeur à l'Université de Franche-Comté.
	M.	M. PERRINE	Ingénieur Télécom & Direct. DRN Metz.
	M.	P. BOURCET	Chef de Dept. TS à TDF-C2R Metz.
	Mme	C. TOSSER-ROUSSEY	Maître de Conf. Université de Metz.
Invité:	M	H. LE GUYADER	Directeur de Techex France, Paris.

A mes parents...

REMERCIEMENTS

C'est dans un cadre de collaboration entre le laboratoire Mécatronique Industrielle à l'Université de Metz et le Centre d'étude en Radiodiffusion et Radiocommunications de TéléDiffusion de France (TDF-C2R), que ce travail a été réalisé.

Les travaux rassemblés dans ce mémoire ne sont pas le fait du hasard. Ils sont le fruit des idées qui ont trouvé un terrain fertile et leur a permis de germer en toute sérénité. L'atout majeur de cet environnement est la diversité des personnes qui le forment. Sa force est dans sa confédération. Son caractère humain est sa source d'enthousiasme. Et je garde pour toujours gravée dans ma mémoire cette ambiance de sympathie et de complicité. Je tiens à cet effet à remercier tous ceux, de loin ou de près, qui ont contribué à l'accomplissement de ce travail.

Je tiens tout d'abord à remercier, tout particulièrement, Monsieur le Professeur André Tosser-Roussey, professeur à l'Université de Metz et directeur du laboratoire Mécatronique Industrielle, pour m'avoir accueilli dans son laboratoire et pour la confiance qu'il m'a accordée et qui se manifeste par la direction de ce jury.

Je remercie par la même occasion, Monsieur Patrice Bourcet, chef du département Traitement du Signal de TDF-C2R de m'avoir accueilli et intégré dans son équipe, pour sa vivacité, ses nombreux conseils et documents qu'il m'a prodigués en temps utile et d'avoir accepté de siéger à mon jury.

Je tiens à exprimer ma gratitude et ma sincère reconnaissance à Monsieur Louis Legrand, Maître de Conférence à l'Université de Dijon pour les nombreux conseils et suggestions qui m'ont aidé à améliorer ce mémoire, et aussi pour être rapporteur de cette thèse.

Je remercie également Monsieur Paindavoine, professeur à l'Université de Dijon et Monsieur Gérard Govaert professeur à l'Université de Compiègne pour avoir accepté de juger mon travail, ainsi que Madame Michelle Valentin, professeur à l'Université de Franche-Comté pour sa participation au jury.

Que Monsieur Alain Faure, professeur à l'Université du Havre, soit remercié d'être rapporteur de cette thèse, pour ses remarques et pour l'intérêt qu'il a porté à ce travail.

Je tiens de même à remercier Monsieur Perrine, Directeur de DRN Metz (Direction du Réseau National) de France Télécom, ainsi que Madame Catherine Tosser-Roussey pour l'attention qu'ils ont bien voulu porter à ce travail et pour leur participation à ce jury.

J'ai eu un réel plaisir à avoir travaillé avec Monsieur Arnaud Laprévote, ancien responsable du Laboratoire d'Electronique Numérique et Réseaux de Neurones à TDF-C2R, actuellement à Thomson Consumer Electronics à Strasbourg, et Monsieur Jamal Baïna du laboratoire Codage de l'Image à TDF-C2R. Je les remercie pour leur collaboration, leur bonne humeur et pour les longues discussions passionnantes qu'on a envie d'entretenir et de perpétuer, qui ont dressé des rails solides et mené à termes ce travail et aussi pour l'intérêt qu'ils ont prouvé à l'égard de ce sujet.

Que chaque membre du département Traitement du Signal et particulièrement des Laboratoires Codage de l'Image et du Son, et tous le personnel de TDF-C2R, ainsi que tous les membres du laboratoire Mécatronique Industrielle soient remerciés pour l'excellente ambiance dans laquelle ce travail a été mené.

SOMMAIRE

Deuxième Partie : Aspects et Codage Scalaire

Troisième Partie : Aspects et Codage Vectoriel

INTRODUCTION GENERALE

Une puissance de calcul et des espaces mémoire importants sont une nécessité pour une implémentation souple et efficace d'algorithmes de traitement d'images. Il a fallu alors attendre l'apparition de nouvelles générations de calculateurs, qui est une conséquence des progrès notables de l'informatique, de l'électronique et des concepts nouveaux du traitement du signal, pour assister à un développement général des différentes applications liées au traitement d'images.

Les images numérisées constituent un volume d'information considérable, leur stockage ou leur transmission sont très coûteux. Dans l'état où elles sont, les images nécessitent des supports importants de stockage et des bandes passantes beaucoup plus larges que celles des canaux actuels pour assurer leur transmission.

La solution, nullement contestée dans tous les cas, se trouve dans la compression de cette quantité de données. Les progrès en informatique permettent la conception d'algorithmes complexes pour remplir efficacement cette tâche. Ainsi, sont nées des applications diverses allant des procédés d'archivage d'images scanner, d'empreintes digitales ou d'images satellites, aux services tels que Visiophone, Vidéoconférence et même différents standards et normes (H261, JPEG, MPEG, TVnumérique, TVHD...).

La compression fait appel à des domaines de compétence très variés pour mettre au point des méthodes et des concepts capables de réduire le débit binaire nécessaire à la représentation numérique de l'image, tout en sauvegardant une bonne qualité visuelle.

La première étape consiste à représenter l'image dans un plan où l'information est sous une forme plus dense qui doit conserver le même message. La projection du signal 2D sur des bases de fonctions orthogonales est souvent utilisée. On peut citer l'exemple de la transformée en cosinus discrète (TCD) ou la transformée de Karhunen-Loeve (KL). Mais le recours à des transformations localisées dans le plan temps-fréquence (la représentation multirésolution) permet, d'une part, une approche de la modélisation du système de vision humain et d'autre part, l'application de façon souple des techniques de codage adaptatif.

Pour remplir notre objectif lié à la compression d'images, il est essentiel de disposer de formes de représentation adéquates de l'image. Une décomposition en pyramide dyadique offre une représentation multirésolution, une réduction de la redondance et de bonnes propriétés de localisation dans un plan espace-fréquence.

Le processus complet de codage, dans lequel un ensemble d'outils est mis en oeuvre, peut se présenter sous forme de deux systèmes chaînés: système direct et système inverse. Le système direct est formé de l'opération de décomposition en pyramide après quoi on applique différents types de codages, scalaire et/ou vectoriel. Le système inverse est constitué par l'inverse des opérations précédentes. Ceci fait l'objet d'une étude détaillée dans les trois parties de ce mémoire.

La première partie fait une synthèse et retrace l'historique de la représentation multirésolution de l'image. Nous décrivons la représentation en pyramide Laplacienne, suivie de la décomposition d'un signal en sous-bandes. Par la suite, on dégage la convergence de ces techniques vers un concept mathématique d'analyse multirésolution et d'ondelettes. En outre, nous présentons la mise en oeuvre de la décomposition à l'aide de bancs de filtres orthogonaux ou biorthogonaux.

Dans la deuxième partie, nous établissons quantitativement des critères de sélection de filtres et une étude des caractéristiques statistiques des signaux issus de cette représentation. Le reste est consacré au codage scalaire de ces signaux. Nous développons, d'une part, sur la base de l'approximation de Girsh et Pierce, une méthode d'allocation optimale de bits pour les différents signaux de la pyramide afin de minimiser la distorsion totale de quantification, et d'autre part, un algorithme de quantification scalaire optimisé. En dernier lieu, nous introduisons le codage par zone morte et la comparaison avec la quantification scalaire.

La troisième partie s'articule autour du codage par quantification vectorielle. Nous présentons diverses méthodes de formation du dictionnaire, en particulier la méthode LBG et le réseau de Kohonen. Nous proposons des méthodes d'optimisation de la recherche du vecteur de reproduction dans un dictionnaire.

Un codage hybride autour de la quantification vectorielle (QV), obéissant à des critères psychovisuels, est construit sur la base d'une structure fictive appelée sous-pyramide. Cette structure a permis d'introduire la notion de classification et de réaliser un codage adaptatif au contenu fréquentiel et local de l'image. Elle tire avantageusement profit du caractère de localisation de la transformée.

Nous adaptons notre codeur aux coefficients issus de la TCD, moyennant une représentation sous une forme hiérarchique. Enfin, nous introduisons la sous-pyramide dans le codage de séquence d'images par correspondance des blocs. Nous proposons une méthode de réduction de la charge du calcul des vecteurs déplacements en réalisant cette opération à la résolution inférieure de l'image.

Première Partie :

Temps-Fréquence et Ondelettes.

Première Partie : Introduction

La projection du signal sur des bases de fonctions orthogonales est un moyen efficace pour distribuer son information sur des composantes décorrélées. Plusieurs transformations vérifient cette propriété, la transformée de Karhunen-Loeve (KL), la transformée en cosinus discrète (TCD), la transformée de Fourier discrète (TFD)...

La KL est optimale, mais son implémentation délicate lui fait préférer la TCD, qui est elle même une "version simplifiée" de la TFD. Toutes ces transformées ont pour avantage une bonne localisation fréquentielle mais non temporelle.

La transformée en ondelettes, définie par Y. Meyer et P. Lemarier [Lem86], est bien localisée en fréquence et en temps. Elle permet de même une représentation multirésolution de l'image. Cette notion de multirésolution semble modéliser le processus de vision humaine bas niveau [Mar79]

La première ondelette a été introduite par Grossman et Morlet en 1982 [Gro82]. Son évolution a résulté en une synthèse de plusieurs techniques existantes dans les domaines et mathématique et du traitement du signal, la pyramide Laplacienne [Bur83], les sous-bandes[Cro76a][Est77], les espaces de projection...

Cette transformation, appliquée à l'image, permet une représentation temps-échelle de celle-ci. Par conséquent, elle offre une réorganisation de l'information, une faible entropie, des contours orientés, etc.. L'implémentation est rendue possible avec des algorithmes rapides mis en oeuvre à l'aide de bancs de filtres numériques [Mal89a].

La technique de filtrage par bancs de filtres est une technique connue en traitement du signal. Vetterli et Smith lui ont apporté un soin particulier dans leurs récents travaux [Smi86][Vet86], qui ont précédé de peu les récents développements mathématiques en analyse multirésolution [Dau88] qui définissent les conditions que doivent vérifier les filtres pour engendrer une analyse multirésolution orthogonale.

Dans cette partie du mémoire, nous allons faire une synthèse de l'historique et des techniques qui ont permis de bâtir les fondements pratiques de la représentation en multirésolution. Ainsi, au chapitre I, nous développons la première représentation de l'image en multirésolution, la pyramide Laplacienne, pour situer le principe. Le chapitre II est un panorama sur la décomposition en sous-bandes et les bancs de filtres. Ces deux chapitres présentent les outils mis à la disposition de la transformation en ondelettes développée en chapitre III, pour l'analyse multirésolution orthogonale et la construction d'un algorithme rapide de décomposition en pyramide d'ondelettes. Au chapitre IV, nous allons montrer comment, dans le cadre de la biorthogonalité, construire des filtres courts (peu de coefficients) et symétriques.

I. La Pyramide Laplacienne

I-1. Introduction

L'image est un ensemble de pixels qui présente généralement beaucoup de redondance informationnelle. Pour limiter cette redondance, une solution consiste à lui appliquer une transformation. Le principe même des méthodes de codage par transformation est de représenter un ensemble d'éléments corrélés par un ensemble de coefficients décorrélés. Ainsi, les procédés de codage pour la réduction de débit sont avantagés par l'application de ces transformations. On décrit dans ce chapitre, la pyramide Laplacienne, une introduction à une nouvelle génération de transformation autre que les techniques classiques du type TFR, TCD ou autres. Elle a au moins l'avantage d'être le précurseur de la représentation de l'image en multirésolution ou multiéchelle. Ce chapitre lui sera consacré pour mettre en évidence les avantages d'une telle représentation. Il sera complété dans les chapitres ultérieurs par des fondements en filtrage numérique et une analyse mathématique.

I-2. Pyramides Gaussiennes

I-2-1. Cas 1D

L'image, en forme d'une matrice de points, est considérée comme un signal à deux dimensions. A chaque point est attribué une valeur qui indique son niveau de gris. Pour simplifier, on se limite dans un premier temps aux signaux à une dimension. L'extension en deux dimensions s'effectue sans difficultés.

Dans ce cas, les données d'un signal sont sous forme d'une séquence de nombres notée $s = (s_n)_{n\in Z}$. La séquence d'origine $s^0 = (s_n^0)_{n\in Z}$ est identifiée par l'exposant zéro. L'idée de base de la pyramide Gaussienne est la construction des versions "grossières" successives $(s^0, s^1, \ldots, s^j, \ldots, s^J)$ de la version d'origine s^0 à des résolutions différentes. Le schéma proposé

par Burt et Adelson [Bur83a], pour réaliser cette opération, est défini par l'opération suivante :

$$s_k^1 = \sum_n w(n-2k).s_n^0 \qquad (1.1)$$

où w(n) sont des valeurs de pondération issues d'une fonction Gaussienne. La première version grossière s^1 est construite à partir des échantillons pondérés du signal s^0 avec les valeurs de w, figure (1.1).

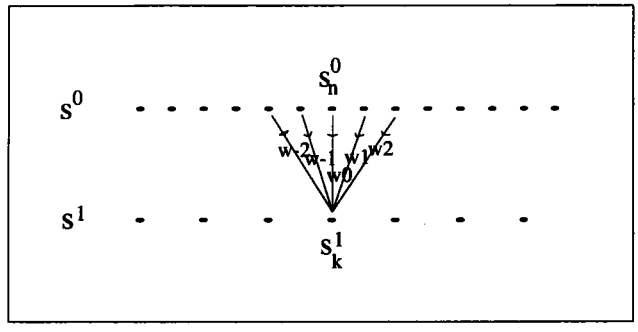


Figure 1.1 : Représentation schématique de l'opération (1.1)

On peut déduire aisément à partir de la relation (1.1) que cette tâche est équivalente à deux opérations. La première est une opération de convolution ou un filtrage, la seconde est une opération de décimation ou sous-échantillonnage. La représentation schématique de ces opérations est donnée par le diagramme de la figure suivante :

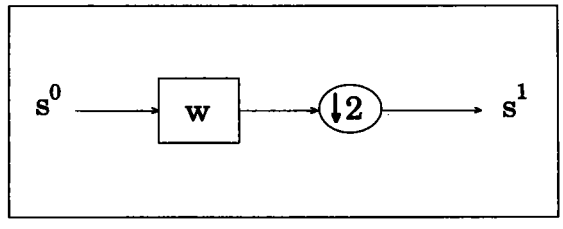


Figure 1.2 : Diagramme relatif à l'opération de convolution de la séquence s^0 avec les coefficients de pondération w suivie d'une opération de décimation

Il est évident que le signal s^1 contient moins d'informations que le signal s^0. En terme d'approximation, on peut dire que la version grossière ou réduite s^1 est une approximation du signal s^0 à une résolution inférieure. Si on note 2^0 la résolution de s^0, la résolution de s^1 est alors 2^{-1}. Il suffit de réitérer les opérations précédentes pour produire les autres versions. Ainsi à partir de la version s^1 on peut déduire s^2. Au bout de J itérations, on obtient les approximations successives $(s^0, s^1, \ldots, s^j, \ldots, s^J)$ de s^0 aux différentes résolutions 2^{-j}, pour $0 \leq j \leq J$.

I-2-2. Extension au cas 2D

Dans le cas d'une image, une configuration simple du passage en 2D s'établit par la réalisation de l'opération précédente en deux temps. Une fois sur les lignes et une fois sur les colonnes. L'extension de la relation (1.1) au cas 2D se fait alors de la façon suivante :

$$s^j(l,k) = \sum_n w(n-2k) \sum_m w(m-2l) s^{j-1}(m,n) \qquad (1.2)$$

Cette opération est représentée par le diagramme de la figure (1.3) suivante:

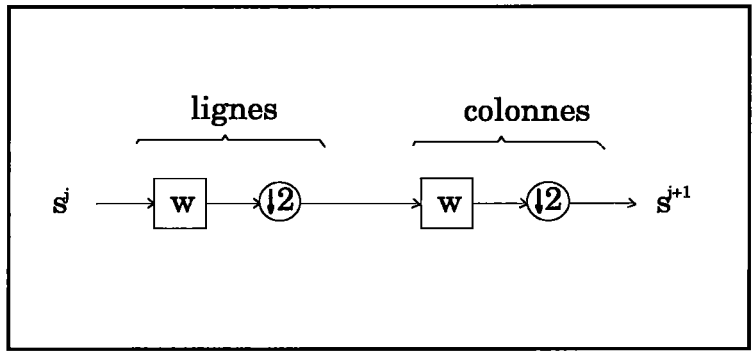


Figure 1.3 : Diagramme relatif aux cas 2D

Si la taille de l'image à la résolution 2^0 est de NxN, alors celle de la version s^1 à la résolution 2^{-1}, est N/2xN/2. Le nombre total d'éléments de la pyramide Gaussienne constituée par la séquence $(s^0, s^1, \ldots, s^j, \ldots, s^J)$, figure (1.4), pour un J assez grand tend vers $\frac{4}{3}N^2$.

La construction de la pyramide Gaussienne est une étape qui permet par la suite l'obtention de la pyramide Laplacienne.

Figure 1.4 : Exemple d'une pyramide Gaussienne.

I-3. Pyramides Laplaciennes

La pyramide Gaussienne affiche un caractère très redondant, puisque à partir d'un niveau de la pyramide on peut déduire tous les niveaux inférieurs. Mais son intérêt apparaît avec la construction de la pyramide Laplacienne. Cette dernière, introduite par Burt [Bur83a], a pour objectif de ne garder de la pyramide Gaussienne que l'information perdue entre deux versions successives.

Comme nous l'avons précisé précédemment, en passant d'une version à une autre plus réduite, les détails fins sont perdus. L'objectif est que la pyramide Laplacienne soit constituée seulement de ces détails. La différence entre les versions successives est un moyen qui va permettre de récupérer ces détails. Ceci ne peut être possible que par une remise à une même échelle des deux versions.

I-3-1. Extraction des détails perdus entre versions successives en 1D

Pour extraire le signal détail d^0, c'est à dire la différence entre la version s^0 à la résolution 2^0 et son approximation s^1 à la résolution 2^{-1}, il faut les mettre à une même dimension. Pour cela, on détermine dans un premier temps un signal $\tilde{s}^0$ qui représente s^1 à la même dimension que s^0. Il est construit par une interpolation de s^1 suivie d'une convolution avec les mêmes coefficients w(n) :

$$\tilde{s}_n^0 = \sum_k w(n-2k).s_k^1 \qquad (1.3)$$

Cette opération est schématisée par la figure (1.4) suivante:

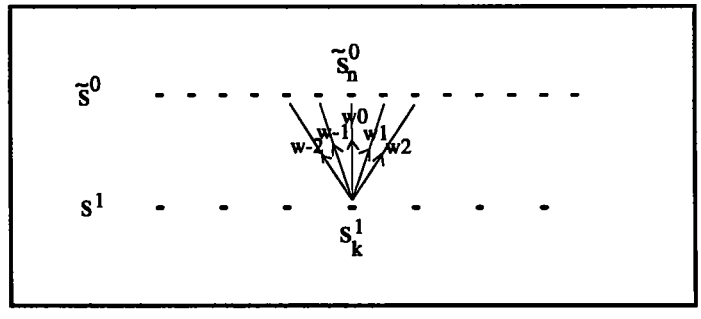


Figure 1.4 : Représentation schématique de l'opération de remise à l'échelle.

On peut généraliser l'équation précédente aux autres versions :

$$\tilde{s}_n^{j-1} = \sum_k w(n-2k)s_k^j \qquad (1.4)$$

Le signal détail d^j est calculé directement par différence entre s^j et la version interpolée $\tilde{s}_n^j$ de la version réduite s^{j+1}.

$$d_n^j = s_n^j - \tilde{s}_n^j \qquad j = 0, ..., J-1 \qquad (1.5)$$

Le schéma de cette décomposition est donné par la figure suivante :

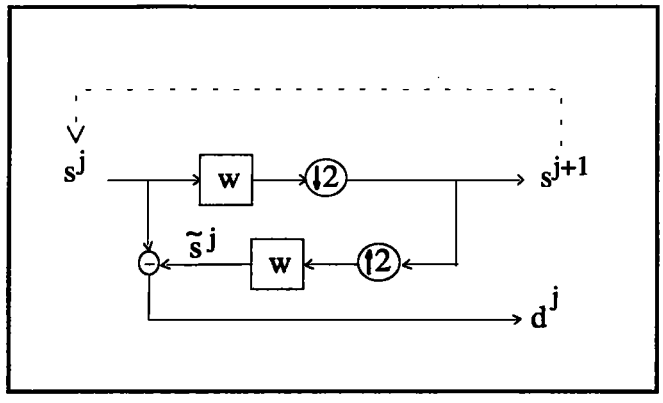


Figure 1.5 : Extraction du signal détails entre versions successives.

I-3-2. Signal détail en 2D

L'extension au cas 2D permet la construction de la pyramide Laplacienne. Cette pyramide est constituée des images de détails perdus entre une image et sa version réduite. Les équations (1.4) et (1.5) deviennent alors :

• interpolation des versions inférieures pour une remise à l'échelle :

$$\tilde{s}^{j-1}(m,n) = \sum_k w(m-2k) \sum_l w(n-2l) s^j(k,l) \qquad (1.6)$$

• calcul des images de détails perdus entre deux versions successives :

$$d^j(m,n) = s^j(m,n) - \tilde{s}^j(m,n) \qquad (1.7)$$

Un exemple d'une pyramide Laplacienne $\left(d^j\right)_{0 \le j \le J}$, déduite de la pyramide Gaussienne précédente, est représenté par la figure (1.6). On constate que d'une part, les deux pyramides sont constituées du même nombre de coefficients, d'autre part, elles ont pour dernier niveau la même version image s^J.

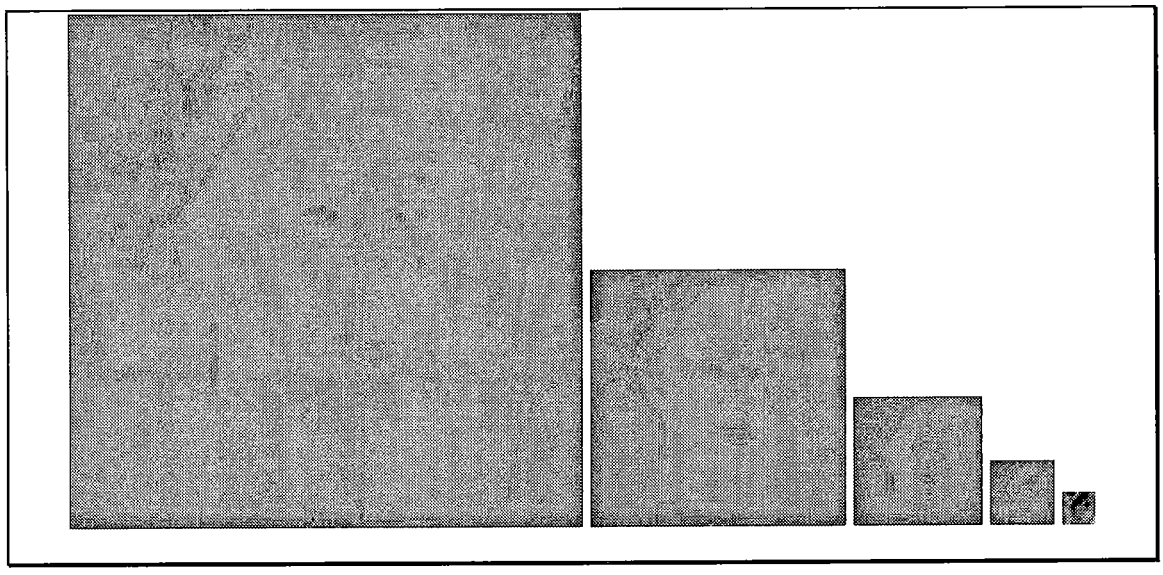


Figure 1.6 : Exemple de représentation d'une pyramide Laplacienne.

I-4. Le noyau générateur

La fonction de pondération w, appelée noyau générateur, utilisée par Burt est choisie sous certaines contraintes. Pour simplifier on a : [Bur83a]

•Seul un nombre limité de coefficients de w est différent de zéro :

$$w(i) = 0 \qquad \text{ici } |i| > 2 \tag{1.8a}$$

•normalisation : $$\sum_i w(i) = 1 \tag{1.8b}$$

•symétrie : $$w(i) = w(-i) \tag{1.8c}$$

•Plus une contrainte supplémentaire, appelée "égale contribution", donnée par l'équation suivante : $$\sum_n w(2n) = \sum_n w(2n+1) \tag{1.9}$$

Ces contraintes sont satisfaites par les noyaux générateurs du type:[bur81]

$w(0) = a$
$w(1) = w(-1) = 1/4$
$w(2) = w(-2) = 1/4 - a/2$

On note au passage que le paramètre a ne peut prendre que des valeurs dans l'intervalle]0.125,0.625[pour des raisons de régularité [Dau90] qui seront développés au chapitre III.

Des exemples de fonction w, pour différentes valeurs du paramètre a, sont représentés par la figure (1.7).

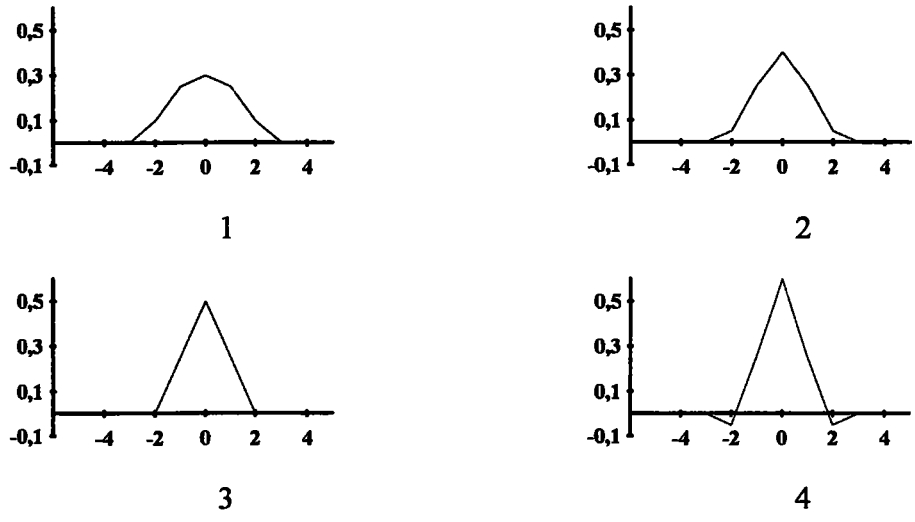


Figure 1.7 : La forme des coefficients de pondérations en fonction du paramètre a. 1) a=0.3 fonction plus large qu'une Gaussienne. 2) a=0.4 fonction presque Gaussienne. 3) a=0.5 fonction triangulaire. 4) pour a= 0.6 fonction trimodale.

Le noyau générateur en 2D appliqué à l'image est choisi, pour des raisons de simplicité, comme un noyau séparable. Cela permet de traiter séparément les lignes et les colonnes. Il est donné par :

$$w'(m,n) = w(n) \,.\, w(m). \qquad (1.10)$$

I-5. Caractéristiques de la pyramide Laplacienne

La pyramide Laplacienne présente une structure de données qui offre plusieurs avantages en traitement d'image. En plus de la simplicité de sa mise en oeuvre, elle permet de réduire la redondance présente dans l'image. Par conséquent, elle réduit le débit nécessaire au codage de l'image d'origine. Cependant, cette structure pyramidale n'est avantageuse que si elle est capable de reconstituer l'image d'origine.

I-5-1. Reconstruction de l'image d'origine.

Un des avantages de la représentation en pyramide Laplacienne est sa capacité à restituer l'information d'origine. Ainsi, à partir de la séquence détail $(d^0, d^1, \ldots, d^j, \ldots, d^J)$ on peut remonter à l'image d'origine.

En partant du niveau le plus bas d^J de la pyramide Laplacienne qui n'est autre que la version réduite s^J, on veut reconstituer la version s^{J-1}. Une fois ce niveau reconstitué, la même opération est réitérée sur cette version pour déduire les versions suivantes.

A partir de l'équation (1.7), la version s^{J-1} est reconstituée à partir de d^{J-1} et s^J.

Puisque $\tilde{s}^{J-1}$ est donnée fonction de s^J, relation (1.6), on peut écrire :

$$s^{J-1}(m,n) = d^{J-1}(m,n) + \tilde{s}^{J-1}(m,n) \qquad (1.11)$$

La reconstruction des autres versions se fait sur le même principe. La figure (1.8) donne la représentation graphique de cette reconstruction dans le cas général.

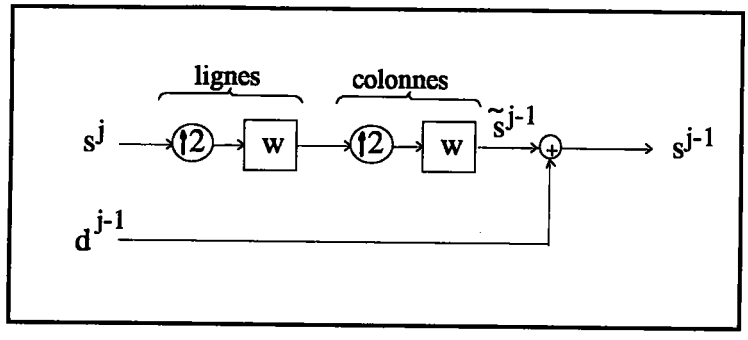


Figure 1.8 : Reconstruction des versions supérieures de la pyramide Gaussienne.

L'opération peut alors être réitérée jusqu'à obtention de l'image originelle :

$$s^0(m,n) = d^0(m,n) + \tilde{s}^0(m,n)$$

avec $$\tilde{s}^0(m,n) = \sum_k w(m-2k) \sum_l w(n-2l) s^1(k,l)$$

I-5-2. Décorrélation de l'information

La nature même d'une image veut qu'elle soit très redondante, tout au moins pour des images usuelles. Sur le plan de la gestion de l'information, ceci présente beaucoup d'inconvénients, particulièrement quand il s'agit de leur stockage ou leur transport. Une conséquence de cette redondance est la grande corrélation statique dans le domaine spatial des éléments qui composent l'image.

Un des avantages de la décomposition en pyramide Laplacienne est qu'elle est composée de versions d'images dont les éléments présentent de faibles corrélations (tableau (1.1)). On peut considérer ces éléments et ces versions comme statistiquement indépendants. Les coefficients de corrélation des versions qui composent la pyramide Gausssienne sont voisins de 1, alors qu'ils ont des valeurs assez faibles dans le cas de la pyramide Laplacienne. La décomposition de l'image en pyramide Laplacienne peut être considérée comme un processus de décorrélation de l'information.

On ajoute aussi, mais sans entrer dans les détails, que les valeurs des coefficients de corrélation sont étroitement liées au choix des coefficients de pondération du noyau générateur w. Les meilleurs résultats sont obtenus pour a=0.6, [Bur81].

Tableau 1.1 : Coefficients de corrélation intra-niveau
pour chacune des pyramides (image "femme au chapeau")

Niveau	P. Laplac.	P. Gauss.
0	-0.27	0.96
1	-0.23	0.93
2	-0.21	0.86
3	-0.19	0.74
4	0.49	0.49

I-5-3. Réduction du débit binaire.

A travers l'aspect des distributions des coefficients de la séquence Laplacienne, on peut constater que la plupart des coefficients sont concentrés autour de la valeur nulle. Cela a pour conséquence, des valeurs d'entropie correspondantes relativement faibles, tableau (1.2). Malgré son excédent de coefficients, l'entropie totale relative à toute la pyramide Laplacienne est plus faible que celle de l'image d'origine. Ceci montre de manière quantitative l'intérêt d'une telle représentation pour la réduction de débit binaire.

Tableau 1.2: Valeurs d'entropie de chaque niveau de la pyramide
Laplacienne et celle de l'image originale (femme)

Niveau	Entropie
0	4.55
1	4.64
2	5.20
3	5.79
4	6.81
Tot.	6.32
Im. orig.	7.52

I-6. Conclusion

La pyramide Laplacienne est une structure de données souple avec des caractéristiques intéressantes pour le traitement d'image. Elle représente une image comme une séquence d'images englobant des activités spécifiques. Cette structure permet une relative décorrélation de l'information contenue dans l'image. Par conséquent, le nombre de bits nécessaires à son codage est réduit.

L'inconvénient majeur de cette structure est l'excédent des coefficients qui constituent la pyramide dans un rapport de 4/3. Cela signifie que la pyramide Laplacienne est constituée de plus d'éléments que l'image d'origine. Or l'objectif recherché est de coder l'image avec le minimum de bits. L'augmentation du nombre d'élément représentant l'image paraît alors contradictoire et pénalise ce procédé.

Néanmoins, la pyramide Laplacienne ouvre la voie à un nouveau procédé de codage d'image. On verra dans les chapitres suivants comment on peut faire usage de nouvelles techniques pour remédier à ces inconvénients.

II. Codage en Sous-Bandes

II-1 Introduction

Depuis son introduction, pour le codage de la parole à débit réduit, par Crochiere et al en 1976 [Cro76a], [Cro76b], le codage en sous bandes s'est avéré une technique particulièrement puissante.

L'idée de base du codage en sous-bandes est la décomposition du signal d'origine en différentes bandes de fréquence complémentaires. L'avantage d'une telle décomposition est la possibilité de coder séparément les bandes avec un codeur approprié. Par conséquent, une distribution de bits adaptative devient envisageable en respectant les particularités statistiques et psycho-acoustiques ou visuelles de chaque bande [Cro81]. Cela permet une distribution adéquate de l'erreur de codage parmi les sous-bandes [Ram86].

L'extension au filtrage en sous-bandes multidimensionnel a été introduit par Vetterli [Vet84]. Il a traité de l'aspect décomposition en sous-bandes des signaux multidimensionnels sans, pour autant, présenter des applications en codage. Les premières applications aux images ont été présentées par Woods et O'Neil [Woo86a,], [Woo86b] avec des approximations et des analyses théoriques.

Le codage d'image en sous-bandes s'est avéré une technique qui peut se substituer efficacement à la pyramide Laplacienne. Son avantage principal est de remédier à l'excédent d'échantillons produit par celle-ci. En contre partie, le problème de la reconstruction devient plus délicat à résoudre. Différentes techniques ont été developpées pour venir à bout de ce problème. Plusieurs sont décrites brièvement dans ce chapitre, après une introduction aux techniques de codage en sous-bandes.

II-2 Principe de codage en sous-bandes

L'idée de base du codage en sous-bandes est la décomposition du signal en différentes bandes de fréquences complémentaires. En pratique, le signal est décomposé en différentes sous-bandes de fréquences par des filtres passe-bande. Le signal résultant est sous échantillonné dans le but de garder constant le nombre de coefficients traités.

Pour la reconstruction, les différentes sous-bandes sont interpolées par une insertion adéquate d'échantillons nuls suivie d'un filtrage passe-bande.

Dans le cas d'un signal à une dimension, le système d'analyse/synthèse en deux sous-bandes (ou deux canaux) d'égale largeur est décrit par le graphe de la figure (2.1). Il est composé de deux modules, un module de décomposition (ou analyse) et un module de reconstruction (ou synthèse).

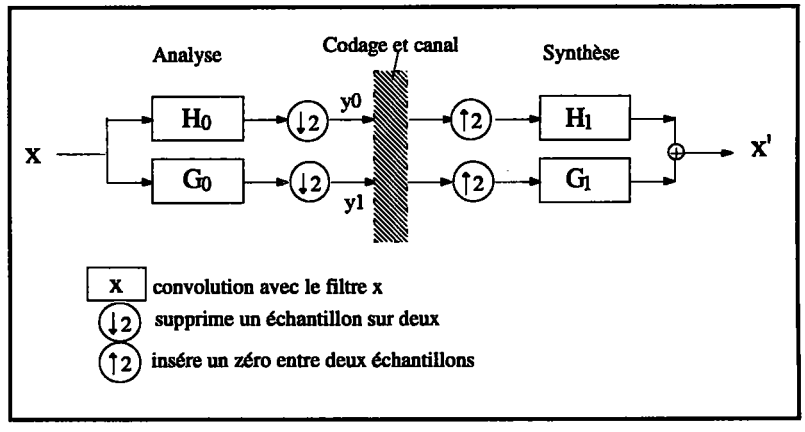


Figure 2.1 : Analyse/synthèse en sous-bandes à deux canaux d'un signal à 1D

2-2-1. Module de décomposition

Dans le cas monodimensionnel, le signal d'entrée x(n) est décomposé en deux bandes y_0 et y_1 par deux filtres H_0 et G_0 respectivement passe bas et passe haut. Ceci est suivi d'une décimation d'ordre 2 pour garder constant le nombre total de coefficients entre la sortie et l'entrée.

Dans le domaine transformé en *z*, on peut écrire y_0 et y_1 sous la forme :

$$y_0(z) = 1/2[\, H_0(z^{1/2})\, X(z^{1/2}) + H_0(-z^{1/2})X(-z^{1/2})\,] \qquad (2.1a)$$

$$y_1(z) = 1/2\,[\, G_0(z^{1/2})\, X(z^{1/2}) + G_0(-z^{1/2})X(-z^{1/2})\,] \qquad (2.1b)$$

2-2-2. Module de reconstruction :

Le module de reconstruction a pour objet la reconstitution du signal. Il est formé à base d'interpolation et d'un banc filtres H_1 et G_1 respectivement passe bas et passe haut. Le signal x', résultant de cette combinaison, est donné par :

$$X'(z) = 1/2[\, H_0(z)\, X(z) + H_0(-z)X(-z)\,]\, H_1(z) + 1/2\,[\, G_0(z)\, X(z) + G_0(-z)X(-z)\,]\, G_1(z) \qquad (2.2)$$

2-2-3. Condition de reconstruction exacte du signal:

Le banc de filtre analyse/synthèse doit être conçu de sorte qu'il permet la reconstruction exacte du signal d'origine, c'est à dire :

$$x'(n) = x(n)$$

La question qu'on se pose en face à un tel système est : comment choisir les filtres H_0, G_0, H_1 et G_1 pour garantir la restitution parfaite du signal x(n) ?

Pour y répondre, il faut établir les conditions que doivent vérifier ces filtres.
Si on regroupe les termes en X(z) et X(-z), le signal en sortie X'(z) peut s'écrire alors :

$$X'(z) = C(z)X(z) + D(z)X(-z) \qquad (2.3a)$$

avec

$$C(z) = 1/2\,[H_0(z).H_1(z) + G_0(z).G_1(z)] \qquad (2.3b)$$

$$D(z) = 1/2[H_0(-z).H_1(z) + G_0(-z).G_1(z)] \qquad (2.3c)$$

La réponse en fréquence (pour $z = e^{j\omega}$) du système linéaire à deux bandes est contenue dans le premier membre C(z) de l'équation (2.3), tandis que le second membre représente le terme de recouvrement ou "aliasing". Pour reconstituer parfaitement le signal d'origine il faut que D(z) soit nul et que C(z) soit une fonction de transfert unitaire.

2-2-4. Contraintes sur les filtres

Le terme de recouvrement peut être annulé si les filtres de synthèse sont liés aux filtres d'analyse par les relations suivantes :

$$H_1(z) = G_0(-z) \qquad (2.4a)$$

$$G_1(z) = -H_0(-z) \qquad (2.4b)$$

Dans ces conditions la fonction de transfert du système global devient :

$$C(z) = 1/2\ [H_0(z).G_0(-z) - H_0(-z).G_0(z)] \qquad (2.5)$$

Pour n'introduire ni distorsion de phase ni distorsion de fréquence sur le signal de sortie il faut que la fonction C(z) ait une phase linéaire et un module constant. C(z) doit s'écrire alors :

$$C(z) = z^{-(N-1)} \qquad (2.6)$$

Ce qui correspond à un retard pur d'ordre N-1 qui correspond au retard de propagation dans le système.

En résumé, avoir un système d'analyse/synthèse à reconstruction parfaite revient à déterminer les filtres H_0 et G_0 capables de vérifier la relation (2.6), les autres filtres sont déduits à partir de la relation (2.4). Plusieurs solutions ont été proposées, les plus utilisées sont décrites dans la suite de ce chapitre.

II-3 Les Filtres Miroirs en Quadrature

Croisier, Esteban et Galand ont été à l'origine des bancs de filtres miroirs en quadrature QMF "Quadrature Mirror Filters" [Croi76], [Est77]. Ce sont des filtres à réponse impulsionnelle finie (RIF), à phase linéaire, qui permettent l'élimination du recouvrement dans l'étage de reconstruction. Ils sont devenus un système fondamental dans la construction de la plupart des structures en arbre des codeurs en sous-bandes.

La solution QMF définit le filtre passe-haut G_0 en fonction du filtre passe-bas H_0 par la relation :

$$G_0(z) = H_0(-z) \qquad (2.7)$$

De cette façon, tous les filtres se déduisent de H_0. Si $h_0(n)$ est la réponse impulsionnelle du filtre H_0, alors celle du filtre G_0 est donnée par :

$$g_0(i) = (-1)^{-(i-1)}\, h_0(N-i-1) \qquad i = 0, \ldots N-1; \qquad (2.8)$$

On dit alors que les filtres H_0 et G_0 sont en miroir.

Si N est l'ordre du filtre H_0 et si H_0 est un filtre à phase linéaire alors il peut s'écrire sous la forme :

$$H_0(z) = |H_0|\, z^{-(N-1)/2} \qquad (2.9)$$

En combinant les équations (2.7) et (2.9), la fonction de transfert s'écrit :

$$C(z) = 1/2\, [\, |H_0(z)|^2 - (-1)^{(N-1)}\, |G_0(z)|^2\,]\, z^{-(N-1)} \qquad (2.10)$$

En combinant les équations (2.10) et (2.6), alors H_0 vérifie la condition de quadrature suivante, pour N impair :

$$|H_0(z)|^2 + |H_0(-z)|^2 = 2 \qquad (2.11)$$

La fonction de transfert $H_0(\omega)$ vérifie alors :

$$|H_0(\omega)|^2 + |H_0(\omega+\pi)|^2 = 2 \qquad (2.12)$$

Si H_0 vérifie l'équation précédente, la fonction de transfert du système correspond alors à un retard pur et on a :

$$x'(n) = x(n-N-1)$$

En conclusion, si les filtres du banc sont reliés par les conditions (2.4) et (2.7) leur confèrant la qualité de filtres miroirs et si le filtre H_0 vérifie la condition de quadrature (2.11) alors le signal décomposé par ce banc de filtres, appelé QMF, est reconstitué parfaitement.

Malheureusement, on ne sait pas construire des filtres à partir de la relation (2.12). Pour ces raisons, les QMF à reconstruction parfaite ne sont pas réalisables. On peut néanmoins minimiser l'erreur de reconstruction en augmentant la taille des filtres, [Joh80].

II-4. Les Filtres Conjugués en Quadratures

Pour les filtres QMF la reconstruction exacte nécessite la détermination du filtre qui doit satisfaire l'équation (2.12). Plusieurs méthodes ([Joh80] [Foo82] [Jai83]) ont été développées pour construire des filtres FIR qui approchent cette condition, où la distorsion peut être minimisée. Il y a pourtant deux cas qui existent pour lesquels cette condition est satisfaite.

- Le filtre idéal passe bas de support infini et dont l'implémentation est impossible.
- Le filtre de Haar qui malheureusement présente une mauvaise sélectivité fréquentielle.

Une nouvelle contrainte a conduit à la création d'une nouvelle classe de filtres

d'analyse/synthèse appelée filtres conjugués en quadrature, CQF [Smi86] de l'anglais "Conjugate Quadrature Filters". La condition de symétrie des coefficients des filtres a été relâchée au profit des filtres FIR à reconstruction exacte, sans distorsion de phase ni de fréquence.

Pour cette nouvelle classe, la condition imposée sur les filtres d'analyse est donnée par :

$$G_0(z) = -H_0(-z).z^{-N} \qquad (2.13)$$

où N+1 est l'ordre du filtre (ou le retard du système). A l'opposé des QMF, H_0 n'est plus supposé à phase linéaire mais plutôt causal et le retard z^{-N} rend aussi G_0 causal.

Les équations (2.13) et (2.4) donnent la fonction de transfert total du système pour N impair.

$$C(z) = 1/2\,[H_0(z).G_0(z^{-1}) + H_0(-z).G_0(-z^{-1})]\,z^{-N} \qquad (2.14)$$

A partir de l'équation (2.6), pour avoir une construction parfaite cela revient à déterminer le filtre produit F_0 qui satisfait l'équation suivante :

$$F_0(z) + F_0(-z) = 2 \qquad (2.15)$$

avec $$F_0(z) = H_0(z).H_0(z^{-1}) \qquad 2.16)$$

A une constante près, F_0 appartient à la classe des filtres antisymétriques par rapport à la demi-fréquence de Nyquist [Smi86]. Cette contrainte est facile à obtenir et permet la construction de filtres à bandes de transition étroite.

Les bancs de filtres CQF génèrent des filtres non symétriques dont les supports sont relativement grands. Malgré leur reconstruction parfaite, leur implémentation est relativement difficile.

II-5. Autre solution

L'inconvénient de l'approche des classes QMF est qu'ils ne permettent pas une reconstruction parfaite. Cependant, la distorsion d'amplitude peut être rendue assez petite en utilisant des filtres relativement longs. Pour des applications telles que le traitement d'image, l'utilisation de filtres de grandes tailles ne suscite pas d'améliorations notables, mais augmente la complexité du calcul. D'autre part, la classe CQF, malgré la reconstruction parfaite, présente

des phases non linéaires. Ceci est rarement exploité en traitement d'images. L'utilisation de filtres à phase linéaire est souhaitable.

Une nouvelle façon de résoudre ce problème a été proposée par Vetterlli [Vet86]. La relation entre les filtres de décomposition n'est pas donnée a priori. Elle consiste à factoriser le filtre produit de la relation (2.5) en des composantes basses fréquences et hautes fréquences ayant des phases linéaires.

Soit P(z) le filtre produit donné par :

$$P(z) = H_0(z).G_0(-z) \qquad (2.17)$$

La condition de reconstruction parfaite s'écrit alors en fonction de P(z) comme suit :

$$P(z) - P(-z) = 2z^{-N} \qquad (2.18)$$

Où N est le retard introduit par la combinaison des bancs de filtres d'analyse/synthèse.

On peut noter que si les filtres $H_0(z)$ et $G_0(z)$ sont à phase linéaire, alors le filtre produit P(z) l'est aussi. De plus, seuls les filtres produits, dont le nombre de coefficients est impair, peuvent satisfaire cette relation.

Pour construire des filtres courts et symétriques, LeGall a proposé une stratégie de résolution de ce problème [LeG88]. Elle consiste à choisir le filtre produit à demi bande P(z) sous la forme :

$$P(z) = a_0 + a_2z^{-2} + \ldots + a_{2p-2}z^{-2p+2} + z^{-2p+1} + a_{2p-2}z^{-2p} + \ldots + a_0z^{-4p+2} \quad (2.19)$$

On peut noter que P(z) à une longueur de 4p-1, mais il n'y a que p coefficients à optimiser et le résultat donne des filtres de longueur 2p. Pour p = 1, on retrouve les filtres de Haar, mais ce type de filtre a une très mauvaise sélectivité fréquentielle.

Pour p = 2, P(z) s'écrit (en imposant un zéro double en -1) :

$$P(z) = 1/4(1 + z^{-1})^2\ 1/4(-1 + 3z^{-1} + 3z^{-2} - z^{-3})(1 + z^{-1})$$

La factorisation génère deux paires de filtres à reconstruction parfaite :

1ère solution : (H_0 et G_0 de même longueur)

$$H_0(z) = 1/4(1 + 3z^{-1} + 3z^{-2} + z^{-3})$$
$$G_0(-z) = 1/4(-1 + 3z^{-1} + 3z^{-2} - z^{-3})$$

2ème solution : (H_0 et G_0 de longueur différente)

$$H_0(z) = 1/8(-1 + 2z^{-1} + 6z^{-2} + 2z^{-3} - z^{-4})$$
$$G_0(-z) = 1/2(1 + 2z^{-1} + z^{-2})$$

Ces exemples ont fourni des filtres très courts à phase linéaire, de plus les coefficients ont une arithmétique très simple permettant une implémentation simple.

II-6. Conclusion

Dans le cadre de la construction de pyramides Laplaciennes, les sous-bandes sont considérées comme un nouvel outil qui peut se substituer à l'approche développée par Burt. Cet outil permet de remédier particulièrement au surnombre des coefficients. Pour fabriquer ces sous-bandes, il faut développer des filtres qui répondent à certaines conditions.

Les différentes approches des techniques de génération de filtres à reconstruction exacte et sans recouvrement présentées dans ce chapitre ont engendré un choix multiple de filtres. A part la complexité d'implémentation et la longueur des filtres, y a-t-il d'autres critères de sélection ?

Meer [Mee82] a choisi le plan fréquentiel pour comparer les différents bancs de filtres. Il comparait les fonctions de transferts des filtres par rapport au filtre idéal. Dans ce contexte, il a défini deux paramètres. Le paramètre η décrivant l'anti-recouvrement (ou antialiasing) et le paramètre μ pour l'indice de filtrage. Les paramètres sont définis comme suit :

$$\eta = \left[\frac{\int_0^{1/4} |H(f)| df}{\int_0^{1/2} |H(f)| df} \right]^2$$

et

$$\mu = \int_0^{1/4} \int_0^{1/4} \left| 1 - |H(f_x) H(f_y)| \right| df_x df_y$$

Les meilleurs filtres sont ceux dont les paramètres η et μ sont proches respectivement de 1 et de 0.

L'analyse multirésolution et la théorie des ondelettes offrent un nouveau moyen pour concevoir des filtres. D'autre part, des critères supplémentaires de comparaison sont définis. Ils sont basés sur des particularités mathématiques telles que le nombre de moments nuls ou la condition de régularité. Ceci sera abordé dans les chapitres suivants après une introduction à la théorie des ondelettes et à l'analyse multirésolution.

III. La Tansformation en Ondelettes

III-1. Introduction

Les ondelettes deviennent un sujet concret et suscitent l'intérêt dans de multiples domaines de la recherche. Plus intéressant encore, chacun les perçoit sous un angle différent, une nouvelle base de représentation des fonctions, une technique de représentation temps-échelle ou tout simplement une nouvelle théorie mathématique. Toutes ces approches sont valables et montrent la versatilité des ondelettes. Elles sont tout au moins un point de convergence entre la théorie de l'analyse multirésolution et les principes de base de la décomposition en sous-bandes. C'est ce point que ce chapitre tentera de mettre en valeur.

III-2. Définition des ondelettes

Les ondelettes proposées par Grossmann et Morlet [Gro84] sont une famille de fonctions générées à partir d'une fonction mère ψ par dilatation d'un facteur échelle a et d'une translation b.

L'intérêt d'un tel ensemble est sa capacité d'analyse d'un signal en mettant au point une cartographie d'activité dans le plan temps fréquence, en plus de la possibilité de synthèse du signal à partir de ces éléments.

3-2-1. Version continue

Les ondelettes sont définies en fonction d'une unique fonction ψ de la façon suivante :

$$\psi_{a,b}(x) = |a|^{-1/2}\, \psi\left((x-b)/a\right) \qquad (3.1)$$

pour (a,b) dans R^2 et $a \neq 0$.

Un signal f peut alors être représenté par les fonctions Uf suivantes :

$$(Uf)(a,b) = \langle \psi_{a,b} , f \rangle = |a|^{-1/2} \int f(x).\psi((x-b)/a)\, dx \qquad (3.2)$$

Si on appelle U "la transformation en ondelettes continue", le signal f est totalement représenté par les (Uf)(a,b) si ψ remplit la condition d'admissibilité suivante :

$$C_\psi = \int |\omega|^{-1} |\Psi(\omega)|^2 \, d\omega < \infty \qquad (3.3)$$

où $\Psi(\omega)$ est la transformée de Fourrier de ψ.

On déduit de cette condition que $\int\psi(x)dx = 0$. Cela veut dire que la fonction $\psi(x)$ admet au moins quelques oscillations. Un des exemples standards d'une telle fonction est l'ondelette de Morlet donnée par :

$$\psi(x) = (2/\sqrt{3})\, \pi^{-1/4}\, (1-x^2)\, e^{-x^2/2}.$$

Un tel schéma de transformation a été introduit comme une alternative à "la transformée de Fourier à fenêtre glissante". Cette dernière avait pour objectif de réaliser une analyse fréquentielle et locale du signal. Elle est définie par la relation suivante:

$$(Gf)(p,q) = \langle g_{p,q}, f \rangle = \int f(x)\, g_{p,q}\, dx \qquad (3.4)$$

où $g_{p,q} = g(x-q)\, e^{ipx}$

La fonction g représente la fenêtre d'analyse. Dans l'approche de Gabor [Gab46], Elle est choisie comme une fonction Gaussienne du type :

$$g(x) = \pi^{-1/4} \exp(-x^2/2)$$

Les deux transformations, en ondelettes et à fenêtre glissante, ont plusieurs caractéristiques en commun :

- Elles sont définies à base de produit scalaire entre f et $\psi_{a,b}/g_{p,q}$,
- Elles procèdent à une analyse locale du contenu fréquentiel du signal,
- Elles ont une formulation identique pour la reconstruction de la fonction f :

$$f(x) = 1/(2\pi a^2 C_\psi) \iint (Uf)(a,b)\, \psi_{a,b}(x)\, da\,db \qquad (3.5)$$

et

$$f(x) = 1/2\pi \iint (Gf)(p,q)\, g_{p,q}(x)\, dp\,dq \qquad (3.6)$$

Comme en pratique les signaux sont discrets, on établit dans la suite les versions discrètes des deux transformations.

3-2-2. Version discrète

Dans le cas de la transformation en ondelettes discrète, a et b prennent respectivement a_0^m et $nb_0a_0^m$ comme valeurs, avec $m,n \in Z$ et $a_0 > 1$, $b_0 > 0$ deux valeurs fixes de R.

De même dans le cas de la transformation de Fourier à fenêtre glissante, $p = mp_0$, $q = nq_0$, avec $m,n \in Z$ et p_0, $q_0 > 0$, valeurs fixes dans R.

Dans ces conditions, $\psi_{a,b}$ et $g_{p,q}$ deviennent respectivement $\psi_{m,n}$ $g_{m,n}$, et on a :

$$\psi_{m,n} = a_0^{-m/2} \psi(a_0^{-m} x - nb_0) \quad (3.7a)$$

$$g_{m,n} = g(x-nq_0) e^{imp_0x}. \quad (3.7b)$$

Dans les deux cas, m correspond aux différents rangs de fréquences à analyser.

Dans le paragraphe précédent on a cité des particularités identiques entre les deux transformées. Mais, le plus intéressant est ce qui les différencie.

Aux fréquences élevées, $g_{m,n}$ présente plusieurs oscillations et inversement peu d'oscillations aux basses fréquences. Ces oscillations sont modulées par la fonction $g(x-nq_0)$. Ce qui fait que la fonction "enveloppe" de toutes les fonctions $g_{m,n}$ est toujours la même, quel que soit le rang de fréquence analysé, figure (3.1a).

Par contre, les fonctions $\psi_{a,b}$ se comportent autrement. Les versions contractées ou dilatées de l'ondelette mère de base possèdent une "enveloppe", plutôt variable, adaptée aux rangs des fréquences analysées, figure (3.1b).

La représentation dans un plan temps-fréquence illustre mieux les différences entre ces deux transformées. La figure (3.2) représente l'emplacement des centres des localisations de fonction $\psi_{m,n}$ et $g_{m,n}$ et leurs fenêtres d'analyse dans le plan temps-fréquences.

Il est clair que pour les fonctions $g_{m,n}$, ces centres sont placés d'une manière régulière pour tous les rangs de fréquences, figure (3.2a). Ce n'est pas le cas des ondelettes pour lesquelles la fenêtre d'analyse est variable. En effet, les résolutions temporelle et fréquentielle changent avec les rangs de fréquences de façon opposée, quand l'une est large l'autre est fine et inversement, figure (3.2b). Bien que la fenêtre d'analyse soit variable, sa surface demeure cependant constante afin de respecter le principe d'incertitude $\Delta t.\Delta f$ = constante.

L'avantage d'une résolution temporelle fine dans les hautes fréquences permet aux ondelettes de bien s'adapter à l'analyse des signaux présentant des singularités de très courtes durée de vie.

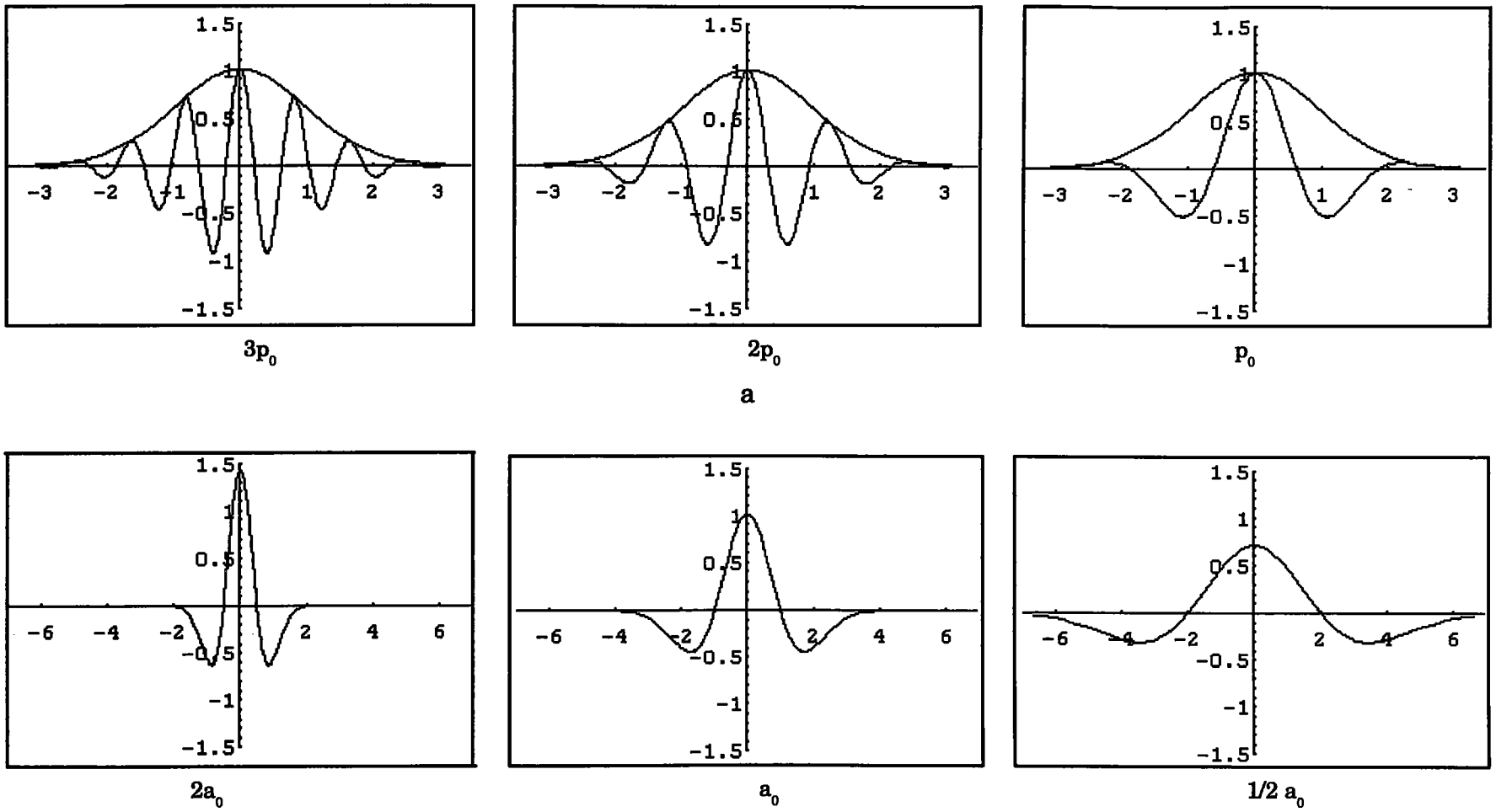


Figure 3.1 : Exemples de fonctions $\psi_{m,n}$ et $g_{m,n}$. On remarque l'enveloppe constante des fonctions $g_{m,n}$ où le nombre d'oscillations varie avec le rang de fréquence alors qu'on garde pour les $\psi_{m,n}$ la même allure de fonction à un facteur près de dilatation

Cependant, la résolution grossière aux basses fréquences découle du fait que des larges intervalles de temps sont nécessaires pour détecter les variations lentes. C'est pourquoi la fonction $\psi_{m,n}$ est bien dilatée (resp. concentrée) pour des grandes (resp. petites) valeurs de m, ce qui lui permet d'extraire les composantes basses (resp. hautes) fréquences du signal.

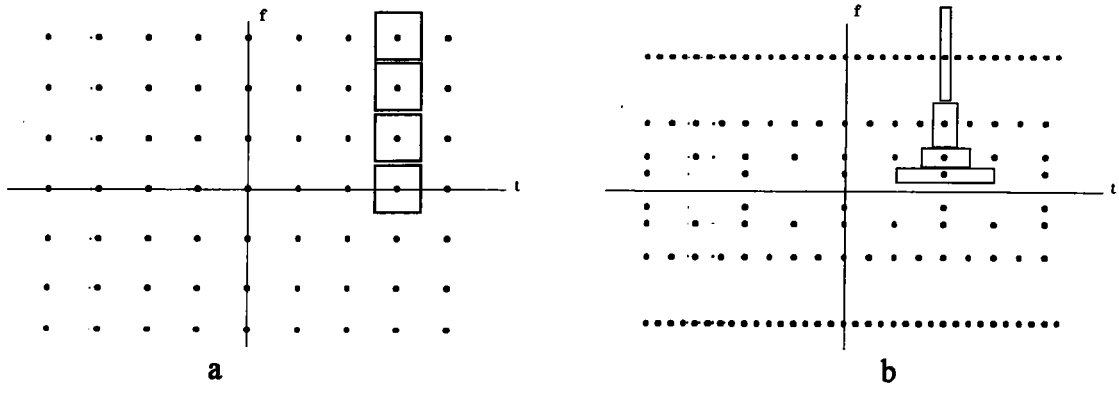


Figure 3.2 : Les centres de localisations et les résolutions temporelles et fréquentielles des fonctions $g_{m,n}$ et $\psi_{m,n}$.

Ceci est loin d'être le cas en transformée de Fourier à fenêtre glissante où la fenêtre d'analyse est toujours la même. Par conséquent, elle a une mauvaise localisation des singularités et une adaptation insuffisante aux variations lentes. Pour de plus amples détails relatifs aux ondelettes généralisées et la comparaison avec les transformées de Fourier à fenêtre glissante voir [Dau86] [Dau90].

3-2-3. Base orthonormale

Il existe un cas particulier très intéressant pour $a_0 = 2$ et $b_0 = 1$. Y. Meyer a montré que dans ce cas là, les fonctions $\psi_{m,n}$ forment une base orthonormale de $L^2(R)$, [Mey86]. $\psi_{m,n}$ s'écrit alors :

$$\psi_{m,n} = 2^{-m/2}\, \psi(2^{-m} x - n) \qquad (m,n) \in Z^2 \qquad (3.8)$$

Les équations de décomposition et de reconstruction deviennent :

$$f(x) = \sum_{m,n} c_{m,n}(f)\, \psi_{m,n} \qquad (3.9)$$

avec $$c_{m,n}(f) = \langle f, \psi_{m,n} \rangle = \int f(x)\, \psi_{m,n}\, dx \qquad (3.10)$$

La base Haar est un exemple classique, elle rejoint ce type de base orthonormale et elle est construite à partir de la fonction suivante :

$h(x) = 1$ pour $x \in [0, 1/2[$

$h(x) = -1$ pour $x \in [1/2, 1[$

$h(x) = 0$ ailleurs.

Dans la suite est développée la définition de l'analyse multirésolution. Par la même occasion, est présentée une méthode de construction de bases d'ondelettes orthonormales.

III-3. Analyse multirésolution

C'est au début du siècle que l'analyse multirésolution est apparue avec les travaux de Haar et Littlewood-Paley sur l'approche multiéchelle. Ce n'est que ces dernières années qu'elle se voit liée à la théorie des ondelettes orthonormales développée par Y. Meyer en 1985.

D'autre part, des outils de filtrage numérique, pour la représentation pyramidale des signaux digitaux, sont apparus avec les travaux de Burt et Adelson (pyramide Laplacienne) et d'Estaban et Galand (sous-bande) dans le cadre du traitement de l'image ou de la parole.

Enfin, on peut considérer que le point de convergence de tous ces travaux se trouve dans l'analyse multirésolution développée en 1987 par S. Mallat et Y. Meyer.

3-3-1. Définition d'une analyse multirésolution

L'idée de base de l'analyse multirésolution est de présenter une fonction f avec une suite d'approximations successives. Chacune d'elles, est une version lissée de f avec une fonction de plus en plus dilatée. Les approximations successives sont de résolutions différentes, d'où le nom d'analyse multirésolution.

On se limite à l'étude d'analyse multirésolution dyadique des signaux f(x) de $L^2(R)$, fonction à énergie finie. Elle consiste en : [Mal89a]

• une famille de sous-espaces $V_m \subset L^2(R)$, $m \in Z$,

$$\ldots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots \qquad (3.11)$$

tel que $\cap V_m = \{0\}$, $\cup V_m = L^2(R)$,

et $f(x) \in V_m \Leftrightarrow f(2x) \in V_{m-1}$

Cette relation précise que les V_m correspondent à différentes échelles.

• il existe une fonction $\phi \in V_0$ appelée "fonction d'échelle", telle que les $\phi_{m,n}$ constituent une base orthonormale de V_m. Elles sont données par :

$$\phi_{m,n}(x) = 2^{-m/2}\, \phi(2^{-m} x - n) \qquad (m,n) \in Z^2 \qquad (3.12)$$

Si on note P_m l'opérateur de projection dans V_m et $P_m(f)$ la fonction projection de f dans V_m, dans ce cas $P_m(f)$ est l'approximation de la fonction f à la résolution 2^{-m} et est définie par

les relations suivantes :

$$P_m(f) = \sum_n c_{m,n}(f)\, \phi_{m,n} \tag{3.13}$$

avec

$$c_{m,n}(f) = \langle f, \phi_{m,n} \rangle = \int f(x)\, \phi_{m,n}(x)\, dx \tag{3.14}$$

3-3-2. Lien avec les ondelettes

Il est clair que les fonctions approximations de f perdent des informations au fur et à mesure que m devient grand. On définit ces informations comme des fonctions détails appartenant au sous-espace W_m complémentaire de V_m dans V_{m+1}. W_m est défini par :

$$W_m \perp V_m \tag{3.15a}$$

$$W_m \oplus V_m = V_{m+1} \tag{3.15b}$$

Si on note Q_m l'opérateur de projection de f dans W_m, alors la différence d'information entre deux approximations successives $P_m(f)$ et $P_{m+1}(f)$ est donnée par :

$$Q_m(f) = P_m(f) - P_{m+1}(f) \tag{3.15c}$$

De même $Q_m(f)$ est donnée par la projection orthogonale de f sur le sous-espace W_m.

On montre alors qu'il existe une fonction $\psi \in W_0$ telle que l'ensemble $\{\psi_{m,n} = 2^{-m/2}\, \psi(2^{-m} x - n)\}_{n \in Z}$ forme une base orthonormale de W_m. La base $\{\psi_{m,n},\ m, n \in Z\}$ n'est autre que la base orthonormale des ondelettes de $L^2(R)$ définie dans le paragraphe précédent.

Puisque $Q_m(f) \in W_m$, on peut écrire alors :

$$Q_m(f) = \sum_n d_{m,n}(f)\, \psi_{m,n} \tag{3.15d}$$

$$d_{m,n}(f) = \langle f, \psi_{m,n} \rangle = \int f(x)\, \psi_{m,n}(x) dx \tag{3.15e}$$

Dans [Dau88], on montre qu'il existe une fonction α_n telle que la fonction d'échelle $\phi(x)$ vérifie la relation suivante :

$$\phi(x) = \sum \alpha_n\, \phi(2x + n) \tag{3.16}$$

Ainsi, on peut déduire la fonction ψ à partir de ϕ par :

$$\psi(x) = \sum (-1)^n \alpha_{n+1}\, \phi(2x + n) \tag{3.17}$$

Cela montre que les ondelettes sont une conséquence directe de l'analyse multirésolution.

La suite de ce chapitre est consacrée à l'implémentation d'un algorithme de

décomposition d'une séquence de données en coefficients d'ondelettes, le lien avec la décomposition en sous-bande et la construction de filtres en rapport avec les fonctions ψ et ϕ.

III-4. Implémentation

Dans son implémentation algorithmique, S. Mallat a exploité les caractéristiques attractives de l'analyse multirésolution pour développer un algorithme de décomposition et reconstruction des signaux 2D images. Il a suivi la même philosophie utilisée par Burt et Adelson dans la construction des pyramides Laplaciennes qui est à l'origine des méthodes de construction multirésolutions. Cette implémentation est plus efficace et permet même sous certaines conditions une sélectivité directionnelle des activités fréquentielles.

3-4-1. Algorithme : cas mono-dimensionnel

En pratique, les signaux sont discretisés. Soit s_n le signal à décomposer en coefficients d'ondelettes. On suppose que le signal s_n est à la résolution de départ, on le note alors s^0_n. La décomposition est réalisée en utilisant une analyse multirésolution. Pour cela, les espaces V_m sont supposés définis et la fonction d'échelle ϕ existe, de sorte que les conditions (3.11) et (3.12) soient satisfaites.

A la séquence s^0_n correspond une fonction f de V_0 telle que :

$$f(x) = \sum_n s^0_n \phi(x-n)$$

On peut construire alors toutes les versions réduites $P_m(f)$ correspondant à f aux différentes résolutions, de même les $Q_m(f)$ correspondant aux détails perdus entre deux résolutions successives.

Etant donné que $f \in V_0$, la relation (3.15) nous permet d'écrire f sous la forme :

$$f = P_1(f) + Q_1(f)$$

A $P_1(f)$ correspond la séquence réduite s^1_n et à $Q_1(f)$ correspond la séquence détail d^1_n. On peut écrire :

$$P_1(f) = \sum_n s^1_n \phi_{1,n}$$

et

$$Q_1(f) = \sum_n d^1_n \psi_{1,n}$$

Les séquences s^1 et d^1 sont respectivement la version lissée de s^0 et le signal détail perdu entre les versions s^0 et s^1. Elles sont déduites à partir de s^0. La séquence s^1 s'écrit :

$$s^1_k = \langle f, \phi_{1,k} \rangle$$

En remplaçant f par sa valeur, on a:

$$s^1_k = \sum_n s^0_n < \phi_{0,n} , \phi_{1,k} > \quad (3.18a)$$

De la même manière on déduit la séquence d^1 :

$$d^1_k = \sum_n s^0_n < \phi_{0,n} , \psi_{1,k} > \quad (3.18b)$$

3-4-2. Relation entre analyse multirésolution et banc de filtre

Les produits scalaires qui apparaissent dans les équations de calcul des séquences s^1 et d^1 peuvent être écrits d'une autre manière. Si on pose h(n) et g(n), deux fonctions telles que :

$$h(n) = < \phi(x/2), \phi(x-n) > = 2^{-1/2} \int \phi(x/2)\phi(x-n)dx \quad (3.19a)$$

et
$$g(n) = < \psi(x/2), \phi(x-n) > = 2^{-1/2} \int \psi(x/2)\phi(x-n)dx \quad (3.19b)$$

Les relations (3.18a) et (3.18b) peuvent s'écrire alors :

$$s^1_k = \sum_n h(n-2k)s^0_n \quad (3.20a)$$

et
$$d^1_k = \sum_n g(n-2k)s^0_n \quad (3.20b)$$

On montre facilement qu'à partir des relations précédentes, on peut généraliser ces relations pour déterminer toutes les autres versions réduites et par conséquent, les versions détails correspondantes de la séquence d'origine s^0_n. Les formulations générales sont données par :

$$s^j_k = \sum_n h(n-2k)s^{j-1}_n \quad (3.21a)$$

et
$$d^j_k = \sum_n g(n-2k)s^{j-1}_n \quad (3.21b)$$

Cet algorithme utilise les mêmes fonctions h(n) et g(n) à tous les étages de décomposition. C'est ici que réside l'importance des analyses multirésolutions, sur le développement d'un algorithme de décomposition rapide d'un signal en coefficients d'ondelettes.

La fonction h(n) n'est autre que la fonction α_n de la relation (3.16) à un facteur $\sqrt{2}$ près. Elle est étroitement liée à la fonction d'échelle ϕ. On peut aussi écrire cette relation de la façon suivante:

$$\Phi(\omega) = H(\omega/2)\, \Phi(\omega/2) \quad (3.22)$$

Ou $\Phi(\omega)$ et $H(\omega)$ sont respectivement les transformées de Fourier de ϕ et h(n). A partir de la relation (3.17) on déduit la relation qui lie g(n) à h(n) : [Mal89b]

$$g(n) = (-1)^{n-1} h(n-1) \quad (3.24)$$

Qu'on peut aussi écrire sous la forme :

$$G(\omega) = e^{-j\omega} H(\omega) \qquad (3.25)$$

Enfin la relation entre ψ et ϕ dans le plan fréquentiel et donnée par :

$$\Psi(\omega) = G(\omega)\, \Phi(\omega). \qquad (3.26)$$

Le fait de définir h et g, simplifie énormément le calcul des coefficients des ondelettes $c_{m,n}(f) = < f, \psi_{m,n} >$ et ceci par un procédé itératif utilisant les mêmes fonctions h et g tout au long du traitement, comme on va le voir. Il faut noter que les fonctions h et g sont fixées par le choix d'une analyse multirésolution.

On utilise les mêmes fonctions à tous les niveaux de la décomposition, par conséquent, la décomposition devient aussi simple que la construction de la pyramide Laplacienne vue précédemment. Les fonctions h et g ne sont autres que les réponses impulsionnelles des filtres respectivement passe bas et passe haut définis en sous-bande. Les relations (3.21) correspondent bien à un filtrage suivi d'une décimation. La figure (3.1) représente le schéma, en une dimension, de décomposition d'une séquence s^0 en une version lissée s^1 et une version détail d^1. Après J itérations on obtient la séquence (d^1, ..., d^j, ... , d^J, s^J) constituée des versions détails successives et la version la plus réduite de s^J.

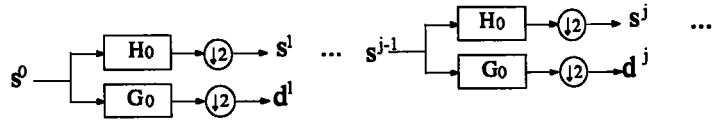


Figure 3.1: Représentation de la décomposition d'un signal en versions lissées et détails.

3-4-3. Comparaison avec la pyramide Laplacienne

La relation (3.21a) correspond à la même décomposition que celle de la pyramide Laplacienne de Burt et Adelson (1-1) où la fonction de pondération w(n) et remplacée par h(n). La différence entre ces deux méthodes est dans l'obtention de la séquence détails d^1.

On montre que h vérifie les mêmes contraintes (1-8) et (1-9) que w. Par ailleurs g vérifie la relation $\sum_n g(n) = 0$ qui est en rapport avec la nature différentielle de g.

Dans les pyramides Laplaciennes le nombre total d'éléments, après J itérations, est supérieur au nombre d'éléments en entrée. Par contre l'algorithme développé ci-dessus garde constant le nombre de coefficients, grâce à l'orthogonalité des fonctions d'analyse. En plus, comme on va le voir, il permet une sélectivité directionnelle, dans le cas bi-dimensionnel, sans coût supplémentaire, un autre avantage sur la pyramide Laplacienne [Mal89b].

3-4-4. Reconstruction

Le problème de la reconstruction est plus simple avec cet algorithme de décomposition. A partir de la relation (3.15) on peut écrire :

$$P_{j-1}(f) = P_j(f) + Q_j(f)$$
$$= \Sigma_k s^j_k \phi_{j,k} + \Sigma_k d^j_k \psi_{j,k}$$

La version s^{j-1} s'écrit :

$$s^{j-1}_n = < P_{j-1}(f) , \phi_{j-1,n} >$$

alors

$$s^{j-1}_n = \Sigma_k s^j_k <\phi_{j,k} , \phi_{j-1,n}> + \Sigma_k d^j_k <\psi_{j,k} \phi_{j-1,n}>$$

$$s^{j-1}_n = \Sigma_n h(n-2k)s^j_k + \Sigma_n g(n-2k)d^j_k \qquad (3.29)$$

On remarque alors que la reconstruction est facilitée par l'analyse multirésolution et l'algorithme de décomposition et utilise aussi les mêmes filtres que la décomposition, figure 3.2. L'opération de reconstruction peut être réitérée en utilisant les mêmes filtres jusqu'à la reconstruction du signal d'origine.

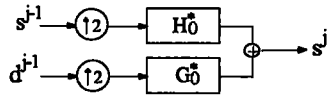


Figure 3.2 : Reconstruction de la version s^j à partir de la version inférieure s^{j-1} et du signal détail d^{j-1} .

Un exemple de décomposition, d'une séquence ligne d'une image, est donné par la figure (3.3).

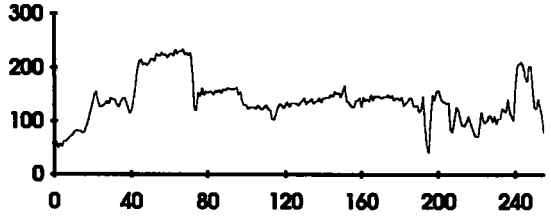
III-5. Exemple de construction d'ondelettes à partir d'une analyse multirésolution

Un exemple d'analyse multirésolution de $L^2(R)$ [Lem88][Bat87], construit à partir des splines polynomiales d'ordre n-1 dans chaque intervalle [k, k+1] $k \in Z$, est donné dans la suite du paragraphe. La fonction d'échelle $\Phi(\omega)$ associée à cette analyse multirésolution est donnée par la relation suivante :

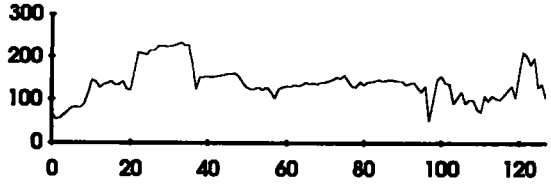
$$\Phi(\omega) = \omega^{-n} (A_{2n}(\omega))^{-1/2}$$

Où la fonction $A_n(\omega)$ est définie par :

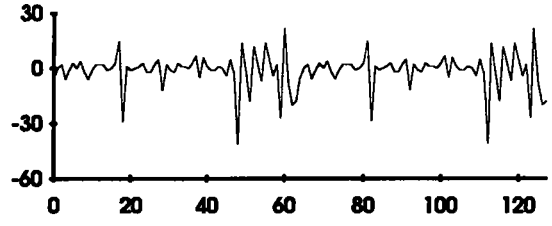
$$A_n(\omega) = \sum_{k=+\infty}^{-\infty} (\omega + 2k\pi)^{-n}$$



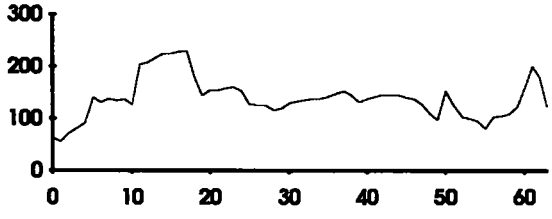
Ligne 128 de l'image Lacornouille correspondant à la résolution 2^0



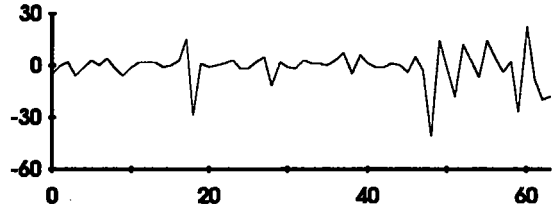
Signal à la résolution 2^{-1}



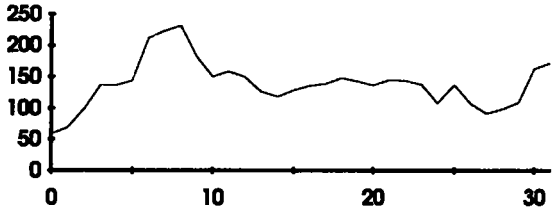
Signal détail résolution 2^{-1}



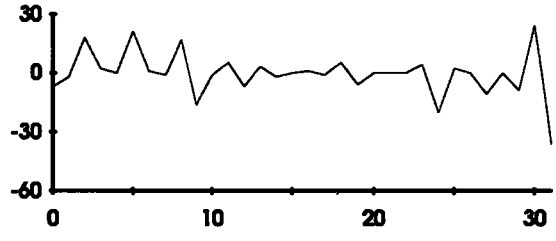
Signal à la résolution 2^{-2}



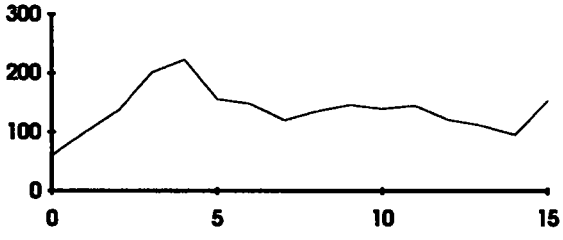
Signal détail résolution 2^{-2}



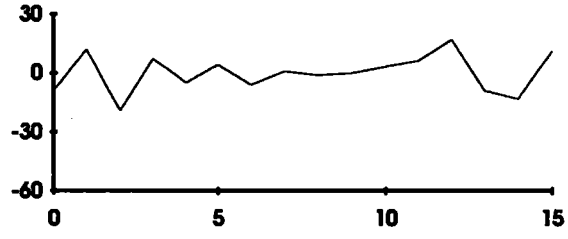
Signal à la résolution 2^{-3}



Signal détail résolution 2^{-3}



Signal à la résolution 2^{-4}



Signal détail résolution 2^{-4}

Figure 3. 3 : Exemple de décomposition en versions réduites et versions détails

Un cas particulier des splines polynomiales est le cas des splines cubiques qui correspond à n = 4. Dans ce cas, on pose $C(\omega) = A_8(\omega)$. La fonction $\Phi(\omega)$ peut s'écrire en posant :

$$C(\omega) = 1/105\ (N_1(\omega) + N_2(\omega))\ (\sin(\omega/2))^{-8}$$

avec

$$N_1(2\omega) = 5+30(\cos(\omega))^2+30(\sin(\omega))^2(\cos(\omega))^2$$
$$N_2(2\omega) = (\sin(\omega))^4(\cos(\omega))^2 + 2/3(\sin(\omega))^2 + 70(\cos(\omega))^4$$

On a alors :

$$\Phi(\omega) = \omega^{-4}\ (C(\omega))^{-1/2}$$

La fonction de transfert H du filtre passe bas correspondant à cette analyse multirésolution est déduite à partir de la relation (3.22), soit :

$$H(\omega) = 2^{-4}\ (C(\omega)/C(2\omega))^{-1/2}$$

La fonction ondelette correspondante est établie à partir de la relation (3.26) et sa transformée de Fourier est donnée par :

$$\Psi(\omega) = e^{-j\omega/2}\ \omega^{-4}(C(\omega/2 +\pi))^{1/2}\ \ (C(\omega)/C(\omega/2))^{-1/2}$$

Enfin La fonction G passe haut peut être déduite de la relation (3.25).

Les graphes correspondant à chacune des fonctions précédentes sont présentés dans la figure suivante :

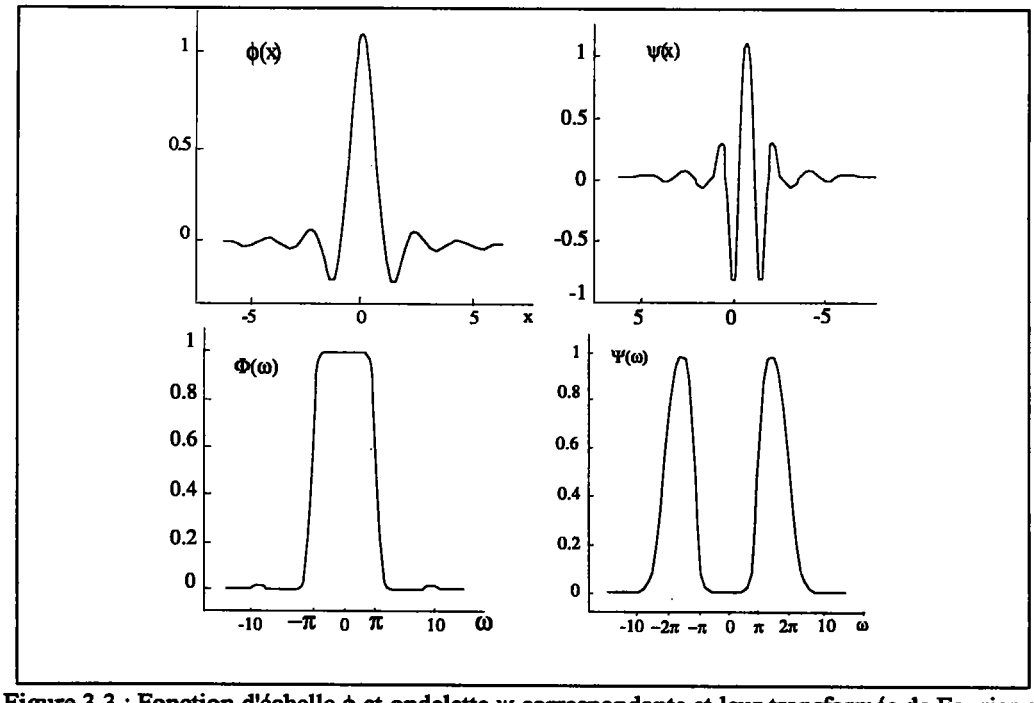


Figure 3.3 : Fonction d'échelle ϕ et ondelette ψ correspondante et leur transformée de Fourier construite à partir d'une analyse multirésolution spline cubique.

Table 3.1 : Coefficients h(n) correspondant à l'analyse multirésolution splines cubique (noté F11)

n	0	±1	±2	±3	±4	±5	±6	±7	±8	±9	±10	±11
h(n)	.542	.307	-.035	-.078	.023	.030	-.012	-.013	.006	.006	-.003	-.002

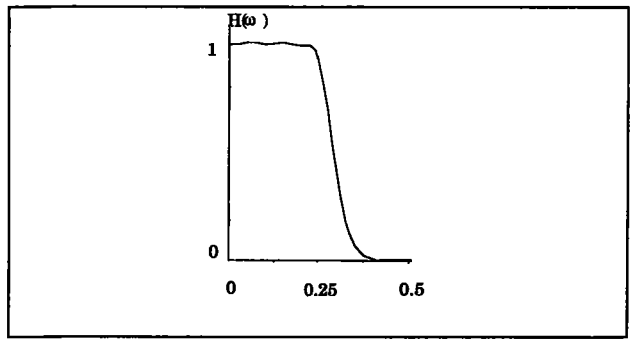


Figure 3.4 : Le filtre H qui découle de l'analyse multirésolution spline cubique

III-6. Construction d'une analyse multirésolution à partir d'un banc de filtres

On a vu au paragraphe précédent comment à partir des relations (3.21) et (3.29) on peut réaliser un algorithme de décomposition et de reconstruction d'un signal en coefficients d'ondelettes. Cependant, pour réaliser cette opération, il suffit d'avoir les filtres h et g. L'analyse est utilisée seulement pour générer les filtres appropriés.

On peut se poser maintenant la question suivante, peut on, à partir des filtres h et g, construire une analyse multirésolution? Ceci est le propos de ce paragraphe.

3-6-1. Condition sur les filtres

Tout d'abord, il faut établir les propriétés des filtres générés par les analyses multirésolutions puis déterminer directement les filtres satisfaisant ces propriétés sans passer pour autant par une analyse multirésolution.

On définit deux opérateurs H et G qui réalisent les opérations représentées par les relations (3.21). Chacun des opérateurs correspond à un filtrage avec respectivement les filtres H et G (dont les réponses impulsionnelles sont respectivement h(n) et g(n)), suivi d'une opération de décimation. On pose alors :

$$\begin{aligned} s^1 &= H\, s^0 \\ d^1 &= G\, s^0 \end{aligned} \qquad (3.30)$$

De même pour la reconstruction et à partir de la relation (3.29) on pose :

$$s^0 = H^* s^1 + G^* d^1 \qquad (3.31)$$

Les opérateurs H^* et G^* sont identiques aux opérateurs précédents, sauf qu'ils sont précédés d'une opération d'interpolation.

La première propriété des filtres est obtenue à partir des relations (3.20) par:

P1. $HH^* + GG = \mathrm{Id}$

La seconde définit l'orthogonalité. A savoir :

P2. $HG = 0$

Une troisième propriété est en rapport avec le caractère passe bas et passe haut des filtres H et G.

P3. $\Sigma h(n) = \sqrt{2}$

$\Sigma g(n) = 0$

Une dernière propriété, en rapport avec la réitération de l'opérateur H, appelée "régularité", est la suivante :

P4. La fonction constante par morceau définie par $(H^*)^k e$ doit converger vers une "bonne" fonction régulière (c'est à dire une fonction qui admet plusieurs moments nuls) quand k tend vers l'infini. k est le nombre d'itérations, e est la fonction impulsionnelle ($e(0) = 1$ pour $n=0$ et nul ailleurs).
La fonction vers laquelle converge $(H^*)^k e$ n'est autre que la fonction d'échelle ϕ définie précédemment.

Cette dernière propriété découle de la relation (3.22) qui lie la fonction d'échelle au filtre passe bas et qui peut s'écrire aussi sous la forme suivante :

$$\Phi(\omega) = \prod_{j=1}^{\infty} H(2^{-j}\omega) \qquad (3.32)$$

Un exemple de la convergence de la fonction $(H^*)^k e$ est donné par la figure (3.5). On remarque que si la condition P4 n'est pas satisfaite, $(H^*)^k e$ converge vers une distribution singulière, figure (3.5b).

On peut conclure enfin que si les filtres H et G remplissent les conditions P1 à P4, alors ils peuvent construire une analyse multirésolution [Dau88].

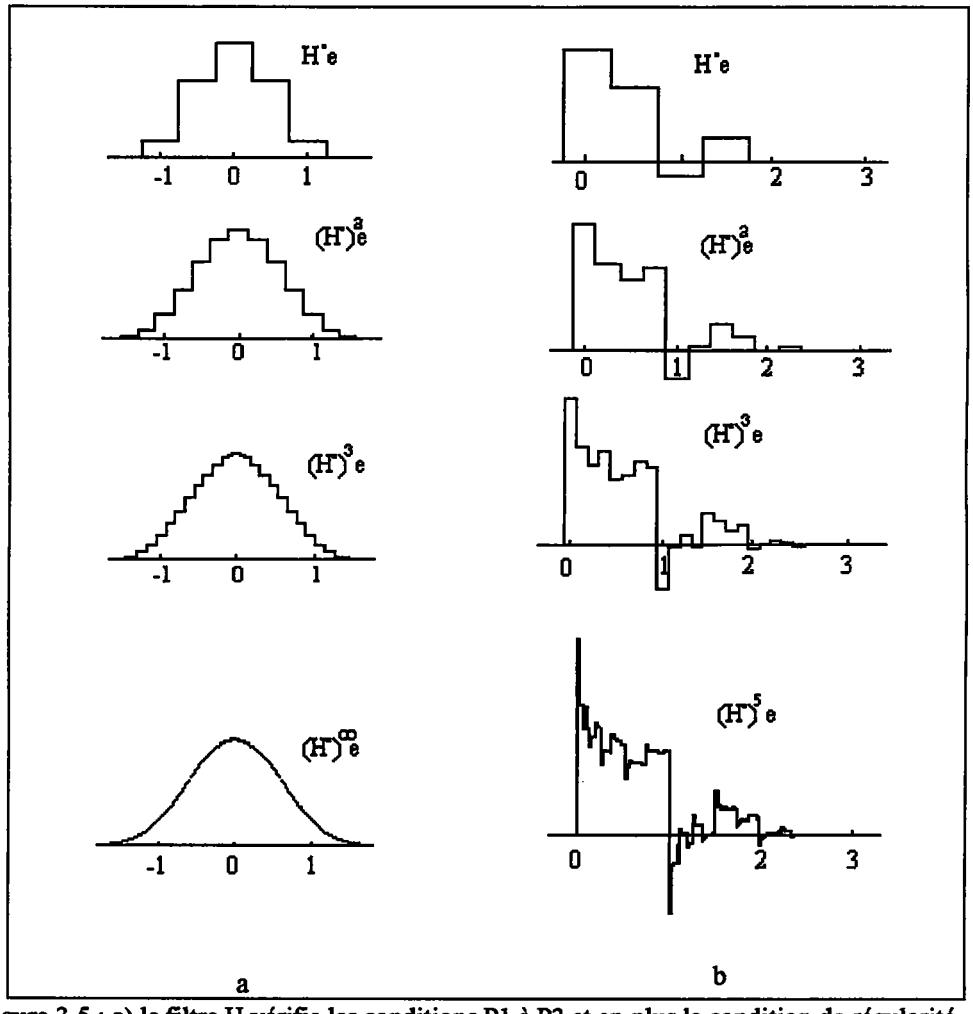


Figure 3.5 : a) le filtre H vérifie les conditions P1 à P3 et en plus la condition de régularité.
b) le filtre H vérifie P1 à P3 mais non P4.

3-6-2. Relation avec les CQF

Ces relations ne sont pas nouvelles. Les conditions P1 à P3 sont similaires aux conditions déclarées par Smith et Barnwell [Smi87] pour construire les CQF "Conjugate Quadrature Filters". Ils sont un cas spécial des QMF "Quadrature Mirror Filters" établis par Esteban et Galand [Est77]. D'ailleurs on peut montrer que les conditions P1 et P2 sont équivalentes à la relation $|H(\omega)|^2 + |H(\omega+\pi)|^2 = 2$ qui est une relation déjà vérifiée par les CQF. Les CQF réalisent une reconstruction exacte sans "aliasing", tout comme les QMF, mais ils sont dépourvus de toute distorsion d'amplitude et de phase, (cf chapitre précédent). Cependant, la condition de régularité P4 n'étant pas imposée, ces filtres ne sont généralement pas équivalents à des bases d'ondelettes orthogonales.

En conclusion, on peut dire que les filtres H et G qui accomplissent les conditions P1 à P3, c'est à dire les CQF, associées à la condition P4 déterminent automatiquement une base d'ondelettes orthogonale de $L^2(R)$. Cette dernière condition fait qu'au filtre H est associé une fonction d'échelle ϕ. La fonction ϕ admet de plus un support compact. Plus exactement, si h(n) $\neq 0$ pour $0 \leq n \leq N$, alors ϕ a pour support l'intervalle [0, N]. Quant au support de la fonction

ondelette associée, c'est l'intervalle [(1-N)/2, (1+N)/2] [Dau88].

3-6-3. Construction des filtres

Une façon d'assurer que la condition P4 soit satisfaite est que h(n) puisse s'écrire sous la forme :

pour $N \geq 2$

$$\Sigma h(n)e^{jn\omega} = 2^{-N}(1+e^{jn\omega})^{-N}Q(\omega) \qquad (3.33)$$

avec

$$|Q(\omega)|^2 = \sum_{j=0}^{N-1}\left(\frac{N-1+j}{j}\right)\sin^{2j}(\omega/2) + \sin^{2N}(\omega/2)\, R((\cos(\omega)/2) \qquad (3.34)$$

Où R est une fonction impaire en cosinus, [Dau88].

Des exemples de fonction d'échelle ϕ et ondelettes associée ψ, construites à partir des filtres définis par les relations précédentes (pour $R \equiv 0$), sont représentés sur la figure (3.6). On constate que la régularité des fonctions ϕ et ψ augmente avec N, c'est à dire, il existe un $\mu > 0$ tel que ϕ et $\psi \in C^{\mu N}$.

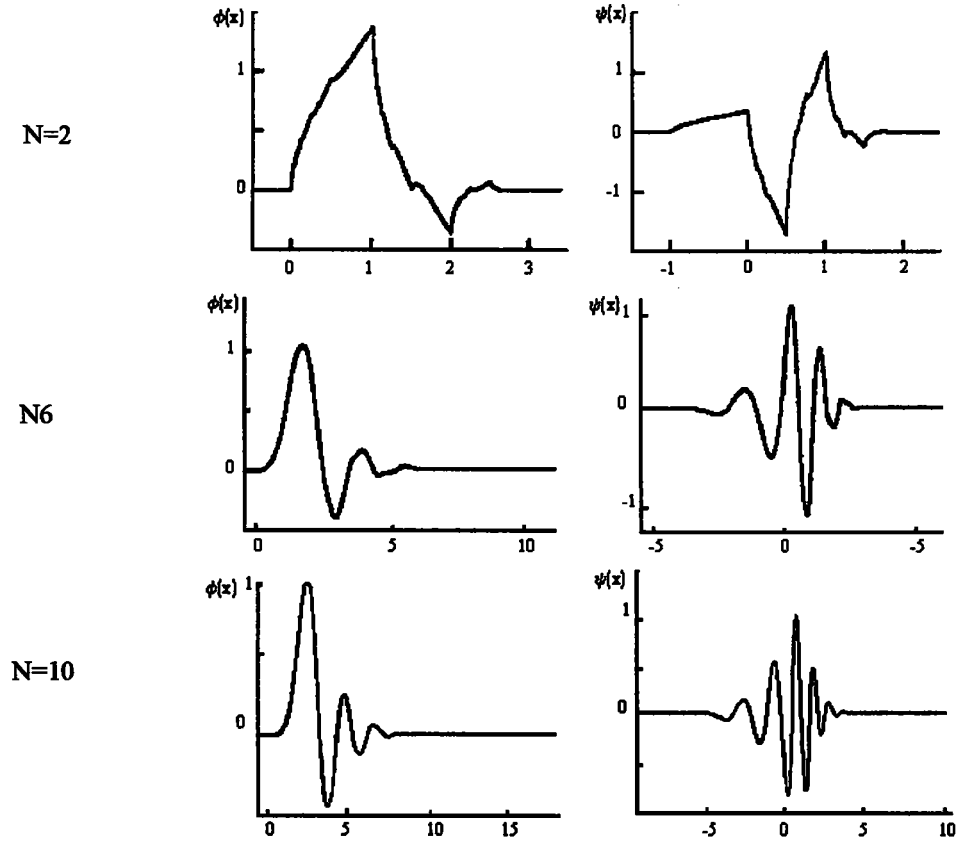


Figure 3.6: Les fonctions ϕ et ψ pour différentes valeurs de N et $R \equiv 0$ dans la relation (3.34)

IV. Les Ondelettes Biorthogonales

IV-1. Introduction

La théorie des bases d'ondelettes orthogonales et l'analyse multirésolution nous ont conduit à la génération des filtres permettant l'analyse en multirésolution et la reconstruction parfaite.

Les filtres construits sont des filtres à réponse impulsionnelle finie, de très courte taille, satisfaisant la condition de régularité qui est une condition qui permet de donner un aperçu du comportement du filtre tout au long des décompositions.

Une contrainte supplémentaire, fortement suggérée pour des applications telles que le codage ou la compression des images, est liée à la sensibilité de l'oeil à toute distorsion de phase. Pour cela, il faut utiliser des filtres à phase linéaire. Si on réussit à construire des filtres symétriques et à réponse impulsionnelle finie, alors cette contrainte sera parfaitement remplie.

Mais, avoir des filtres RIF orthonormaux à reconstruction exacte et à phase linéaire est malheureusement impossible, [Smi86][Bar82]. Seul les filtres correspondant à la base de Haar peuvent répondre à cette exigence sans répondre au critère de la régularité.

On verra dans ce chapitre qu'en relâchant la condition de l'orthogonalité il est possible de construire des filtres à phase linéaire et à reconstruction parfaite. Dans les chapitres précédents, il a toujours été question de schémas de codage où les mêmes filtres h(n) et g(n) sont utilisés pour la décomposition comme pour la synthèse. Nous allons étudier dans ce chapitre un autre schéma de codage dont le banc de filtre permettant une reconstruction exacte admet des filtres de synthèse différents de ceux de la décomposition. Ces types de filtres sont plus flexibles et sont faciles à construire, de plus, ils ont l'avantage d'être symétriques, ce qui n'est pas le cas de ceux vus précédemment.

Des filtres de ce type ont été développés par Vetterli, Smith et Cohen dans [Vet86], [Vet92], [Smi87] et [Coh92]. Les premiers ont développé ce principe dans un schéma de codage en sous-bandes. Quant à Cohen, il a établi les principes mathématiques des analyses multiresolutions où les bases d'ondelettes orthonormales sont devenues bases d'ondelettes biorthogonales. Ce sont des bases duales $\psi_{n,m}$ et $\psi'_{n,m}$, chacune donnée par dilatation et translation des fonctions uniques et non orthogonales ψ et ψ'.

L'analyse multirésolution, liée à des bases d'ondelettes biorthogonales, devient légèrement plus complexe. A savoir que toutes les propositions vues précédemment sont "doublées", particulièrement les cas des espaces emboîtés d'approximations et aussi les fonctions d'échelle ϕ et ϕ'.

IV-2. Filtres et conditions de reconstruction parfaite.

Si aux fonctions ϕ et ψ on fait correspondre les filtres h et g, on peut écrire :

$$s^1_k = \sum_n h(n-2k)s^0_n$$

$$d^1_k = \sum_n g(n-2k)s^0_n$$

Et si leurs fonctions duales ϕ' et ψ' sont liées aux filtres h' et g', l'équation de reconstruction s'écrira alors :

$$s'^0_k = \sum_n \left[h'(2n-k)s^1_n + g'(2n-k)d^1_n\right] \qquad (4.1)$$

Avoir une reconstruction parfaite revient à avoir : $s'^0 = s^0$.

Ceci peut se traduire par les relations suivantes :

$$g(n) = (-1)^{1+n}\, h'(-n)$$
$$g'(n) = (-1)^{1+n}\, h(-n) \qquad (4.2)$$

On note h'^* le conjugué de h', une solution au problème revient à trouver là les filtres h et h' solution de l'équation suivante :

$$h(z)h'^*(z) + h(-z)h'^*(-z) = 2. \qquad (4.3)$$

IV-3. Construction des analyses multirésolutions

En suivant la même philosophie développée au chapitre précédent on établit les relations qui lient les filtres à une analyse multirésolution et répondent à la relation (4.3).

Soient m_0 et m'_0 deux fonctions périodiques liées aux fonctions h et h' par les relations suivantes (transformée de Fourier de $2^{-1/2}h$ et $2^{-1/2}h'$) :

$$\begin{aligned} m_0(\xi) &= 2^{-1/2} \Sigma\, h(n)\, e^{-jn\xi} \\ m'_0(\xi) &= 2^{-1/2} \Sigma\, h'(n)\, e^{-jn\xi} \end{aligned} \qquad (4.4)$$

Comme vu au chapitre précédent, les fonctions ϕ et ϕ' sont liées à h et h' par:

$$\begin{aligned} \Phi(\xi) &= m_0(\xi/2)\, \Phi(\xi/2) \\ \Phi'(\xi) &= m'_0(\xi/2)\, \Phi'(\xi/2) \end{aligned} \qquad (4.5)$$

De même que dans le cas orthogonal, on déduit une relation entre les fonctions ondelettes ψ et ψ' et les fonctions d'échelle correspondantes ϕ et ϕ'.

A partir des relations (4.2) et (4.5)on peut écrire :

$$\begin{aligned} \Psi(\xi) &= e^{-j\xi/2}\, m'^{*}_0(\xi/2+\pi)\, \Phi(\xi/2) \\ \Psi'(\xi) &= e^{-j\xi/2}\, m_0(\xi/2+\pi)\, \Phi'(\xi/2) \end{aligned} \qquad (4.6)$$

Noter que les fonctions ϕ et ψ sont liées par la fonction m'_0 et non par m_0, et inversement pour ϕ' et ψ'. Ceci est une conséquence de la relation (4.2)

Enfin s'il existe un ε tel que :

$$\begin{aligned} |\Phi(\xi)| &< C(1+|\xi|)^{-1/2-\varepsilon} \\ |\Phi'(\xi)| &< C(1+|\xi|)^{-1/2-\varepsilon} \end{aligned} \qquad (4.7)$$

Alors toute fonction f de $L^2(R)$ peut s'écrire sous la forme : [Coh92]

$$\begin{aligned} f &= \sum_{i,j} \langle f, \psi'_{j,k} \rangle \psi_{j,k} \\ &= \sum_{i,j} \langle f, \psi_{j,k} \rangle \psi'_{j,k} \end{aligned}$$

VI-4. Exemples de construction de filtres liés à des bases d'ondelettes biorthogonales

Avoir des filtres à phase linéaire revient à choisir m_0 de sorte qu'elle vérifie $m_0(\xi) = m_0(-\xi)$. A partir des relations (4.3) et (4.4) on peut déduire la relation suivante :

$$m_0(\xi)m'^{*}_0(\xi) + m'_0(\xi+\pi)m'^{*}_0(\xi+\pi) = 1 \qquad (4.8)$$

Si on trouve m_0 et m'_0 vérifiant la relation (4.8) on déduit, par l'intermédiaire des relations (4.5) et (4.6), les fonctions d'échelles et les ondelettes correspondantes.

IV-4-1. Ondelettes biorthogonales associées à la pyramide Laplacienne de Burt

La fonction m_0 est liée dans ce cas au filtre utilisé par Burt (noyau de pondération) pour construire la pyramide Laplacienne, voir chapitre I. m_0 s'écrit alors :

$$m_0(\xi) = -(1/4 - 2a)\, e^{-2j\xi} + 1/4\, e^{-j\xi} + a + 1/4\, e^{-j\xi} + -(1/4 - 2a)\, e^{-2j\xi}$$

Pour différentes valeurs de a, la valeur 0.6 donne la plus petite entropie des coefficients de la pyramide. Pour cette valeur de a, $m_0(\xi)$ s'écrit :

$$m_0(\xi) = (\cos(\xi/2))^2(1+4/5\sin^2(\xi/2))$$

La fonction m'_0 duale de m_0 et qui satisfait l'équation (4.8) est donnée par la relation suivante :

$$m'_0(\xi) = (\cos(\xi/2))^2(1+6/5\sin^2(\xi/2) - 24/30\sin 4(\xi/2))$$

Elle peut aussi s'écrire sous la forme suivante :

$$m'_0(\xi) = 1/280[-3e^{-3j\xi} -15e^{-2j\xi} + 73e^{-j\xi} + 170 + 73e^{j\xi} -15e^{2j\xi} -3e^{3j\xi}]$$

Les coefficients des filtres h et h' sont donnés par le tableau (4.1). Les fonctions d'échelle correspondantes et les ondelettes associées sont représentées par le graphe de la figure (4.1).

Tableau 4.1 : Filtres duaux relatifs à la fonction de pondération de Burt pour a=6. (noté F23)

	0	±1	±2	±3
$1/\sqrt{2}$ hn	0.6	0.25	-0.05	0
$1/280\sqrt{2}$ h'n	170	73	-15	-3

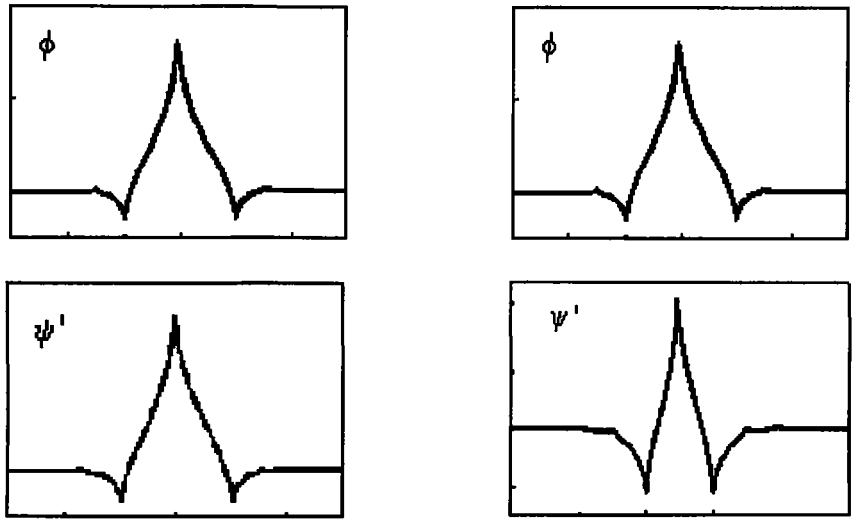


Figure 4.1 : Les fonctions ϕ, ϕ', ψ, et ψ' pour m0 correspondant au filtre Laplacien de Burt.

D'autres exemples peuvent être développés, en particulier des exemples liés aux cas des splines. Pour trouver une solution à l'équation (4.8), on part de la constatation suivante [Coh92], si $\psi \in C^L$ et $\psi' \in C^{L'}$ alors $m_0(\xi)$ et $m'_0(\xi)$ sont divisibles respectivement par $(1+e^{-j\xi})^{L'+1}$ et $(1+e^{-j\xi})^{L+1}$. Les valeurs de L et L' doivent être choisies suffisamment grandes pour que ϕ et ψ soient raisonnablement régulières. Cela revient aussi à ce que les fonctions m_0 et m'_0 soient divisibles par un nombre pair de cos(ξ/2). Dans ce cas m_0 et m'_0 s'écrivent :

$$\begin{aligned} m_0(\xi) &= (\cos(\xi/2)^{2l}\, p_0(\cos(\xi)) \\ m'_0(\xi) &= (\cos(\xi/2)^{2l'}\, p'_0(\cos(\xi)) \end{aligned} \qquad (4.9)$$

avec

$$p_0(\cos\xi)p'_0(\cos\xi)=\sum_{n=0}^{k-1}\left(\frac{k-1+n}{n}\right)\left(\frac{\sin^2\xi}{2}\right)^n+\left(\frac{\sin^2\xi}{2}\right)^k R(\cos\xi) \qquad (4.10)$$

On donne dans les paragraphes suivants deux exemples de construction d'ondelettes biorthogonales liés à ce type de fonctions. La première liée directement aux splines, mais qui engendre des filtres très dissymétriques en longueur. La deuxième remédie à cette dissymétrie par factorisation de la relation (4.10), une approche qui nous rappelle la construction des filtres par l'approche de LeGall décrite au chapitre sous bandes.

IV-4-2. Cas des splines.

Si ϕ est une fonction spline d'ordre N, alors le m_0 correspondant s'écrit :

$$m_0(\xi) = \eta\,(\cos\xi/2)^N = \sum_{n=-\lfloor N/2\rfloor}^{N-\lfloor N/2\rfloor} 2^{-N}\left(\frac{N}{n+\lfloor N/2\rfloor}\right)e^{-jn\xi}$$

avec $\eta = 1$ si N est pair et $\eta = \exp\{j\xi/2\}$ si N impair. Le filtre dual m'_0 correspondant s'écrit alors comme suit :

$$m'_0(\xi) = \eta\,(\cos\xi/2)^{N'} \sum_{n=0}^{k-1}\left(\frac{k-1+n}{n}\right)\left(\sin^2\xi/2\right)^{2n} + \left(\sin^2\xi/2\right)^{2k} R(\cos\xi)$$

avec $N' \geq 1$, $N+N' = 2k$ et R un polynôme impair.

Le tableau suivant donne des exemples de filtres h et h' pour $R \equiv 0$ et N= 2 et pour différentes valeurs de N'.

Tableau 4.2: exemples de filtres h et h' pour $R \equiv 0$, N= 2 et N' = 2, 4, 6 et 8.
(Le banc de filtre pour N= 2 et N' = 4 est noté F21)

		0	±1	±2	±3	±4	±5	±6	±7	±8
N = 2	$2^{-2}\sqrt{2}.h_n$	2	1							
N' = 2	$2^{-3}\sqrt{2}.h'_n$	6	2	-1						
N' = 4	$2^{-7}\sqrt{2}.h'_n$	90	38	-16	-6	3				
N' = 6	$2^{-10}\sqrt{2}.h'_n$	700	324	-123	-78	34	10	-5		
N' = 8	$2^{-15}\sqrt{2}.h'_n$	22050	10718	-3796	-3126	1228	670	-300	-70	35

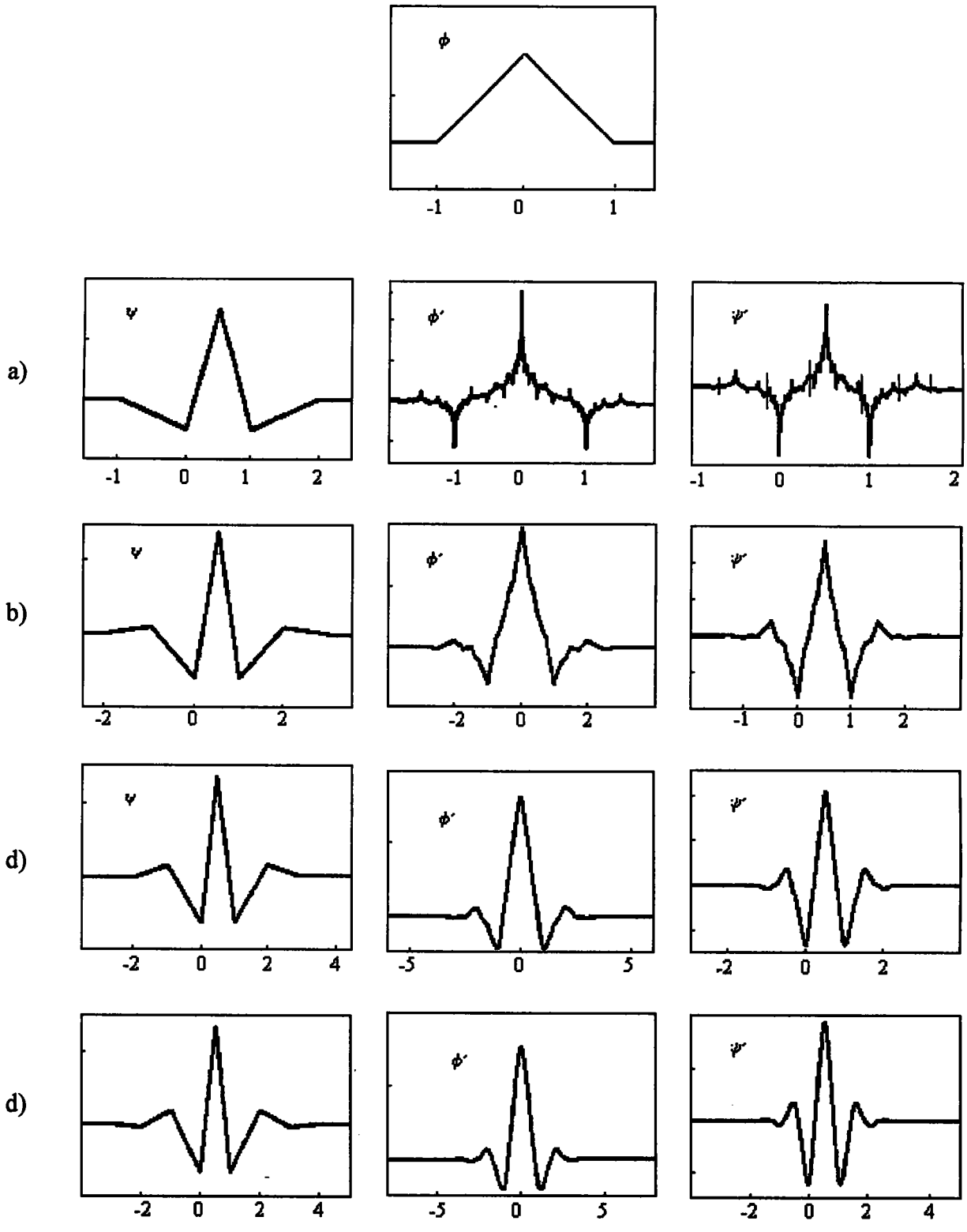


Figure 4.2 : Les fonctions ϕ', ψ, et ψ' pour différentes valeurs de N' générées à partir d'une fonction *B*-spline ϕ d'ordre 2 (N=2). a) N' = 2. b) N' = 4.
c) N' = 6. d) N' = 8.

IV-4-3. Variante des splines

On remarque dans l'exemple précédent que si ψ et ψ' ont la même régularité, ceci produit des filtres avec une grande dissymétrie en longueur. Dans l'exemple suivant, on construit une variante des splines qui remédie à cette dissymétrie .

Pour $R \equiv 0$, on factorise la relation (4.10) comme dans l'approche de Le Gall décrite dans le chapitre 3, de sorte que les longueurs de m_0 et m'_0 soient assez proches. Les exemples suivants sont donnés pour $N = N' = k$ et pour des valeurs de $k = 4$ et $k = 5$.

Tableau 4.3 : Coefficients des filtres pour k =4 et k=5.
(Le banc de filtres pour k=4 est noté F22)

k =4	0	±1	±2	±3	±4
hn	.557543	.295635	-.028771	-.045635	0
h'n	.6602949	.266864	-.078223	-.016864	.026748

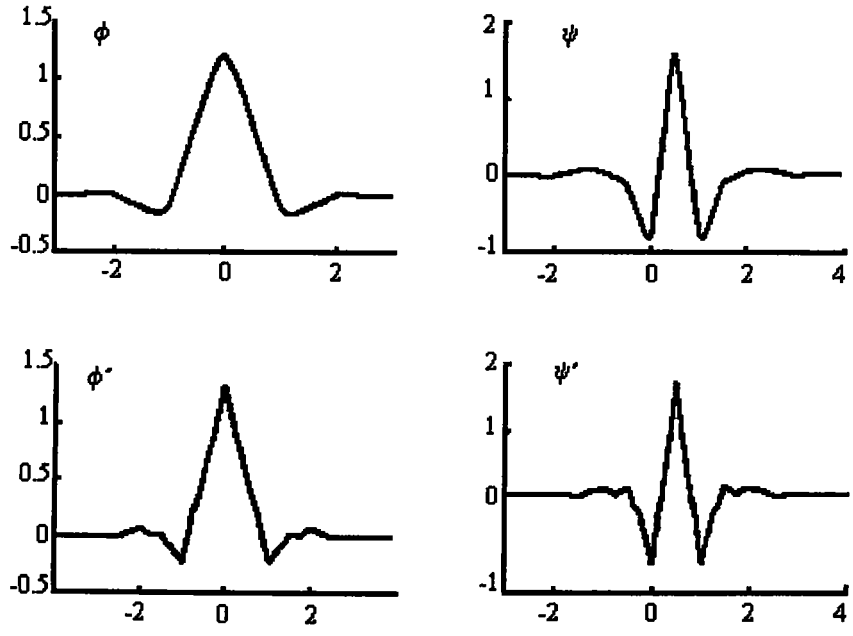
k=5	0	±1	±2	±3	±4	±5
hn	.636046	.337150	-.066117	-.096661	-.00190	.009515
h'n	.520897	.244379	-.038511	.005620	.028063	0

La figure suivante donne le graphe des fonctions ϕ, ϕ', ψ et ψ' relatives aux fonctions m_0 et m'_0 pour k=4 et k = 5.

IV-5. Conclusion

Nous avons présenté, dans cette première partie, une approche unitaire des techniques de la pyramide Laplacienne, du codage en sous-bandes et de la transformation en ondelettes. Les similarités entre les différentes approches ainsi que les différences ont été examinées. Enfin nous avons montré dans le chapitre III comment à partir de l'approche de la décomposition en pyramide Laplacienne et les principes du codage en sous-bande il était possible de remonter à la théorie des ondelettes et son implémentation.

Aprés cette synthèse détaillée, la deuxième partie de ce mémoire est consacrée à l'étude des propriétés statistiques des coefficients issus de la décomposition des images (signal 2D) en pyramide d'ondelettes et leurs codages scalaire.



a

b

Figure 4.3 : Les fonctions ϕ, ϕ', ψ, et ψ' pour une variante de splines avec N=N'
a) k=4 b) k = 5.

Deuxième Partie :

Aspects et Codage Scalaire

Deuxième Partie : Introduction

La transmission d'images numériques, qui se manifeste actuellement par une évolution rapide, provoque un déferlement d'applications, télécopies, visiophone, vidéoconférence, TV numérique, TVHD, etc. Aussi sur le plan de l'archivage, l'image numérique continue sa conquête, CDI, empreintes digitales, images satellites, catalogues (musique, film, produits manufacturés...)...

Mais le facteur capital, qui coopère pour une telle évolution est la compression. En résumé, comprimer une image revient à réduire le coût de son stockage et/ou de sa transmission. Pour des applications très diverses, la compression est alors une nécessité et un moyen pour améliorer le rapport qualité/coût.

Pour comprimer un signal, il existe deux types de méthodes. La première est dite sans perte. Elle est à base de techniques de réduction de la redondance et appliquée généralement aux données. La seconde est dite avec perte, à base de techniques de suppressions d'informations non pertinentes. Souvent les deux techniques peuvent se combiner. Les méthodes avec pertes sont particulièrement utilisées en son et en image. C'est ce type de codage qui sera largement abordé dans le cadre de ce mémoire.

Que faut-il entendre par le terme perte ?

Les techniques de compression avec perte s'occupent d'éliminer deux types d'informations : information redondante et information non pertinente.

- *L'information redondante :*

Elle peut être définie comme "une fonction de message répétitive", donc non nécessaire, au sens où si elle manquait, le message serait encore complet.

En d'autres termes ceci revient aussi à dire qu'une information redondante est une information corrélée. Pour réduire alors la corrélation, plusieurs techniques existent, en particulier des techniques de transformation. Une transformation est la représentation du signal sous une autre base qui produit des coefficients moins corrélés (théorie de Shannon). Une telle transformation doit être inversible pour être capable de réaliser l'opération inverse afin d'obtenir le signal de départ. Nous allons montrer que la transformation en ondelettes, introduite dans la première partie, répond efficacement à ces exigences.

- *L'information non pertinente :*

Une information est non pertinente si sa présence ou son absence n'est pas perçue de façon gênante par l'utilisateur final. Sa perte conduit à une altération irréversible du fait de la perte d'une partie de l'information. C'est une grandeur très subjective, elle reste très difficile à cerner et elle est essentiellement dépendante de l'application visée. Ainsi une information non pertinente en vidéoconférence ne l'est pas nécessairement en imagerie médicale.

Les exemples de pertes d'informations sont nombreux. Nous pouvons citer celui de Musicam qui

est l'algorithme normalisé pour le codage à réduction de débit du signal Audio. Il utilise un procédé qui en présence de deux types d'informations spécifiques, ne retient que la plus pertinente. Le signal résultant a une qualité subjective acceptable. Pour cela, Musicam se base sur des modèles psychoacoustiques quantifiant l'effet de masquage mutuel entre les informations. Une information "masque" une autre quand elle la rend inaudible.

Un autre exemple classique est la numérisation de l'image analogique. Elle subit deux quantifications, une sur les évolutions temporelles et une autre sur l'amplitude du signal. Dans le cas de la quantification temporelle (échantillonnage), il n'est gardé d'un signal continu que des échantillons pris à des intervalles réguliers de temps. Si on respecte le théorème de Shannon, le signal est entièrement reconstitué à partir de ces échantillons. Ceci revient alors à un codage sans perte si on ne tient pas compte des pertes introduites par le pré-filtrage des systèmes de numérisation. La quantification d'amplitude permet de représenter les échantillons précédents du signal vidéo par des termes quantifiés et numérisés. On constate que plus le nombre de niveaux de quantification est faible et plus l'image perd en qualité. Les 256 niveaux de gris (chaque pixel est codé alors sur 8 bits) sont un large compromis qui satisfait la plupart des applications, et la perte introduite est presque imperceptible.

Dans les exemples présentés, la quantification utilisée suit une loi uniforme à pas constant. Dans le cas contraire, il s'agit d'une quantification non uniforme ou à pas variable. Ce procédé sera largement illustré dans les chapitres suivants pour quantifier les coefficients de la pyramide ondelettes. Nous allons chercher à fixer le nombre de pas de quantification et les placer de manière à minimiser la distorsion du signal. En d'autres termes, on vise à avoir le moins de perte possible pour un débit donné. C'est un quantificateur non figé qui s'adapte à la statistique du signal. L'objectif est de construire un codeur qui minimise l'erreur introduite sur le signal. D'autre part, il doit distribuer cette erreur de sorte que la distorsion de l'image reconstruite ne soit pas perçue d'une manière gênante par l'utilisateur. Sachant que normalement, plus le débit est faible et plus l'image reconstruite perd en qualité. Le but alors est de satisfaire le compromis qualité/débit.

Les quantificateurs cités ci-dessus sont de la famille des quantificateurs scalaires et sont considérés comme des opérations de compression avec perte. Ainsi, cette deuxième partie est consacrée à l'aspect codage scalaire des coefficients d'ondelettes. On décrit au chapitre 5 comment décomposer une image en pyramide d'ondelettes, une opération qui réalise une réduction de la redondance spatiale, une première étape vers la compression. Ceci est suivi d'une étude des propriétés statistiques de ces coefficients. Le chapitre 6 traite le problème d'allocation de bits. Ceci permet une distribution optimale des bits aux différentes composantes de la pyramide d'ondelettes en vue d'un codage scalaire optimal de leurs coefficients. Au chapitre 7, il sera développé un quantificateur scalaire optimal. Les chapitres 8 et 9 sont relatifs à une approche algorithmique de l'allocation de bit et à une introduction au codage par zone morte.

V. Image et Pyramide d'Ondelettes

V-1. Introduction

Dans la première partie, il a été présenté la transformée en ondelettes 1D. De nombreuses méthodes d'extension en plusieurs dimensions existent, en particulier le cas bidimensionnel de l'image. On peut distinguer deux catégories de transformation 2D. Les analyses multirésolutions séparables et les analyses multirésolution non séparables. Dans le premier cas, on trouve l'analyse mutirésolution dyadique développée par Mallat [Mal89b]. Dans le second, l'analyse multirésolution en quinconce (ou en $\sqrt{2}$), développée par Feauveau [Fea90], est un exemple. Dans la suite de ce mémoire, la décomposition dyadique séparable est adoptée, puisqu'elle s'adapte aux objectifs de notre schéma de codage.

V-2. Analyse multirésolution dyadique séparable

5-2-1 Définition d'une analyse multirésolution dyadique

Dans cette section, on décrit l'extension du cas monodimensionnel au cas à deux dimensions, concernant en particulier des applications en traitement d'images. Dans ce cas, le signal est une fonction f(x, y) de $L^2(R^2)$ à deux variables et à énergie finie. Les approximations multirésolutions de $L^2(R^2)$ sont des séquences d'espaces emboîtés de $L^2(R^2)$ qui satisfont l'extension en deux dimensions des propriétés (3.11).

Si on appelle $(V^2_m)_{m\in Z}$ une telle approximation multirésolution alors l'approximation d'un signal f(x,y) à la résolution 2^m est égale à sa projection orthogonale sur l'espace V^2_m. De la même manière qu'en monodimension, il existe une fonction $\phi(x,y)$ unique (fonction d'échelle) qui par dilatation et translation forme une base orthonormale de chaque espace V^2_m. Les fonctions de cette base sont données par :

$$\phi_{m,(nx,ny)}(x,y)= 2^{-m/2}\,\phi(\,2^{-m/2}x-n_x,\,2^{-m/2}y-n_y) \qquad m,\, n_x,\, n_y \text{ dans } Z \quad (5.1)$$

5-2-2 Analyse multirésolution dyadique séparable

Pour revenir aux approximations multirésolutions dyadiques séparables chaque espace V^2_m peut être décomposé en un produit tensoriel de deux espaces identiques monodimensionnels V^1_m de $L^2(\boldsymbol{R})$ [Mey88]:

$$V^2_m = V^1_m \otimes V^1_m$$

La séquence $(V^2_m)_{m\in Z}$ forme une approximation multirésolution de $L^2(\boldsymbol{R}^2)$ si et seulement si $(V^1_m)_{m\in Z}$ est une approximation multirésolution de $L^2(\boldsymbol{R})$. Si la fonction d'échelle de $(V^1_m)_{m\in Z}$ est la fonction à une dimension $\phi(x)$ alors la fonction $\phi(x,y)$ fonction d'échelle de V^2_m peut s'écrire :

$$\phi(x,y) = \phi(x)\,\phi(y) \qquad (5.2)$$

et

$$\phi_{m,(nx,ny)}(x,y) = \phi_{m,nx}(x)\phi_{m,ny}(y) \qquad (5.3)$$

De même qu'en 1D, l'approximation du signal f(x,y) à la résolution 2^m est donnée par le produit scalaire suivant :

$$c_m(f) = (< f(x,y),\ \phi_{m,nx}(x)\phi_{m,ny}(y)>) \qquad (5.4)$$

5-2-3. Ondelettes bidimensionnelles associées

Le signal détail à la résolution 2^m est donné par la projection du signal sur l'espace W^2_m, le complément orthogonal de V^2_m dans V^2_{m+1}. Les fonctions, générées par dilatation et translation des trois ondelettes suivantes, forment des bases orthonormales des W^2_m :

$$\begin{aligned}\Psi^v(x,y) &= \phi(x)\,\psi(y)\\ \Psi^h(x,y) &= \psi(y)\,\phi(x)\\ \Psi^d(x,y) &= \psi(x)\,\psi(y)\end{aligned} \qquad (5.5)$$

$\psi(x)$ est l'ondelette associée à la fonction d'échelle $\phi(x)$ dans le cas monodimensionnel. Le signal détail sera composé de trois signaux d^v_m, d^h_m et d^d_m résultant des produits scalaires de f(x,y) avec les trois fonctions suivantes:

$$\begin{aligned}\Psi^v_{m,(nx,ny)}(x,y) &= \phi_{m,nx}(x)\psi_{m,ny}(y)\\ \Psi^h_{m,(nx,ny)}(x,y) &= \psi_{m,nx}(x)\phi_{m,ny}(y)\\ \Psi^d_{m,(nx,ny)}(x,y) &= \psi_{m,nx}(x)\psi_{m,ny}(y)\end{aligned} \qquad (5.6)$$

5-2-4. Implémentation

La transformée en ondelettes bidimensionnelles peut être réalisée avec l'extension

séparable de l'algorithme de décomposition monodimensionel (cf paragraphe III – 4). Toutes les lignes puis toutes les colonnes sont traitées par les filtres utilisés en 1D. A chaque itération, le signal $s_{m+1}(f)$, à la résolution 2^{m+1}, est décomposé en signal version grossière $s_m(f)$ à la résolution 2^m et trois signaux d^v_m, d^h_m et d^d_m. Cet algorithme est illustré par le graphe de la figure (5.1). Les signaux détails d^v_m, d^h_m et d^d_m constituent l'information perdue lors du passage de la version $s_{m+1}(f)$ à la version grossière $s_m(f)$ de résolution respective 2^{m+1} et 2^m.

La décomposition en ondelettes séparables répartit le signal en un ensemble de bandes de fréquences indépendantes. Chacune correspond à une orientation spatiale particulière. Ainsi, les trois signaux d^v_m, d^h_m et d^d_m sont relatifs aux orientations respectives verticale, horizontale et diagonale, figure (5.2).

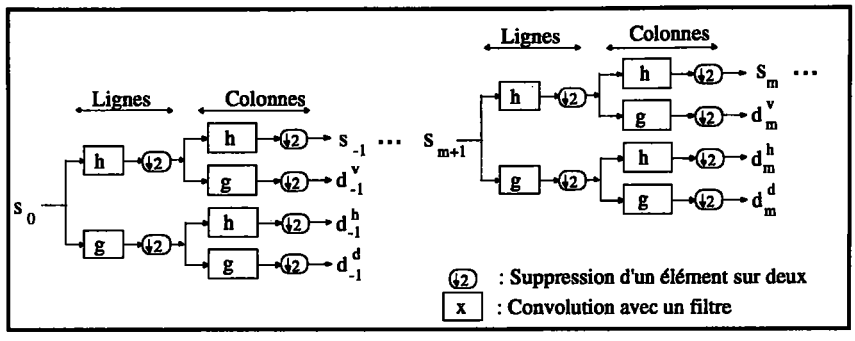


Figure 5.1: Décomposition d'un signal image en ondelettes basée sur l'algorithme de décomposition monodimensionnelle

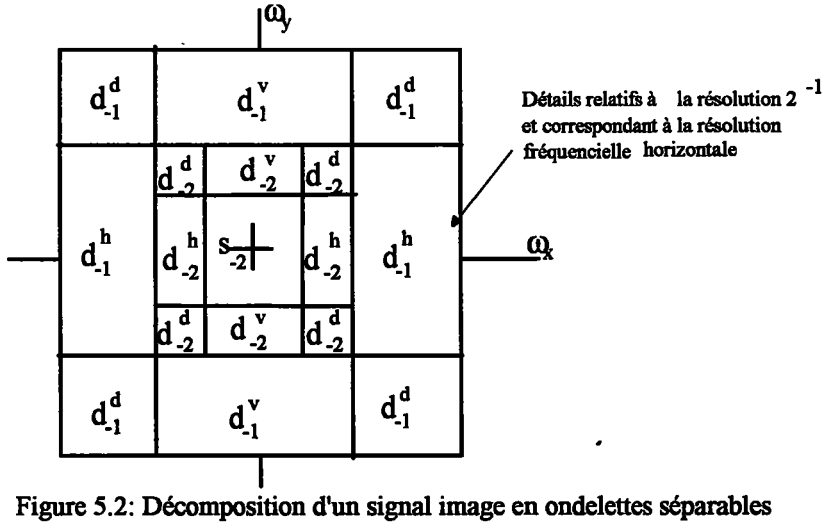


Figure 5.2: Décomposition d'un signal image en ondelettes séparables basée sur l'algorithme de décomposition monodimensionnelle

La figure (5.3) représente une décomposition en pyramide d'ondelettes d'une image en trois niveaux. Chaque niveau est constitué de trois signaux détails correspondants aux orientations horizontales, verticales et diagonales. L'image réduite correspond la résolution 2^{-3} de l'image d'origine.

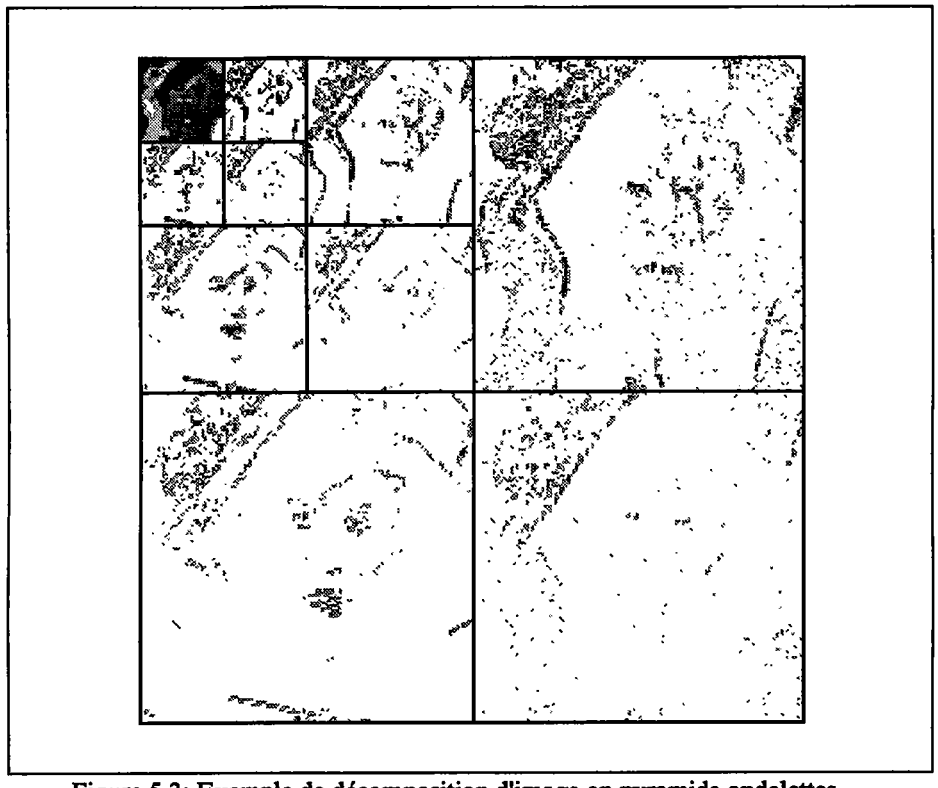


Figure 5.3: Exemple de décomposition d'image en pyramide ondelettes.

5-2-5. Reconstruction

Le schéma de reconstruction utilisé en une dimension peut être étendu au cas bidimensionnel. A chaque étape l'image $c_{m+1}(f)$ à la résolution 2^{m+1} est reconstruite à partir de l'image réduite $c_m(f)$ et des détails d^v_m, d^h_m et d^d_m correspondants. Cet algorithme, qui fait appel aux mêmes filtres qu'en 1D, est illustré par la figure (5.4).

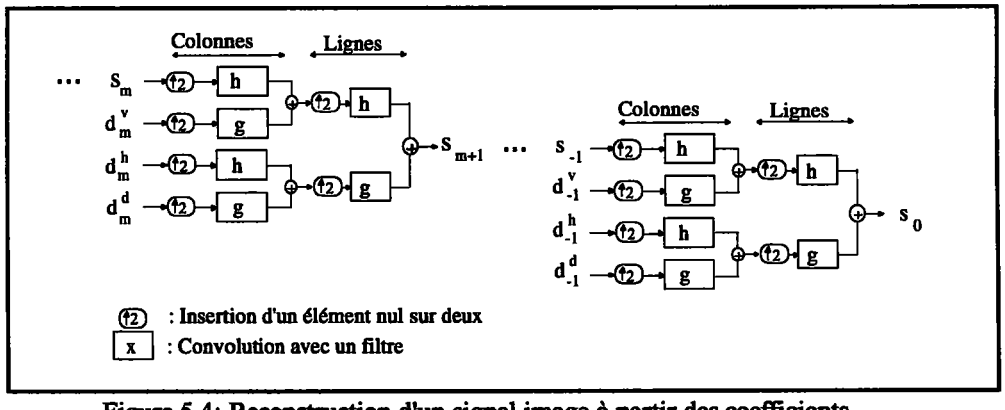


Figure 5.4: Reconstruction d'un signal image à partir des coefficients d'ondelettes

V-3. Caractéristiques d'une pyramide d'ondelettes

Les deux représentations, image en pixels et pyramide d'ondelettes sont équivalente au sens de l'égalité de Parseval [Mal89a]. La pyramide est constituée du même nombre

d'échantillons que l'image en pixel de départ. Et puisque les deux représentations sont équivalentes et constituées du même nombre d'échantillons, quels sont alors les avantages d'une telle décomposition ?

Avant d'aborder ce sujet, il serait aussi intéressant d'ajouter à cette égalité de nombres d'échantillons, l'égalité des formats d'échantillons aussi. De cette façon, la pyramide d'ondelettes occupera un même nombre de bits que l'image en pixels. Nous introduisons dans un premier temps ce qu'on appelle, l'uniformisation des échantillons. Par la suite, des critères de choix des filtres seront établis pour répondre efficacement à la compression.

5-3-1. Uniformisation des échantillons

Uniformiser les données en entrée et en sortie revient simplement à utiliser le même format pour les pixels et les coefficients d'ondelettes. Les images utilisées sont des matrices de pixels numérisés et quantifiés sur 256 niveaux de gris et par conséquent codés sur 8 bits. Les coefficients résultants de la transformation sont des valeurs réelles et nécessitent plus de 8 bits pour les coder. Pour garder un débit binaire constant entre l'entrée et la sortie il faut alors uniformiser les données. Cela revient alors à coder les coefficients en sortie sur 8 bits. Cette opération est équivalente à une quantification scalaire uniforme sur 256 niveaux. Les valeurs réelles des coefficients leur sont alors attribuées les valeurs entières les plus proches. Cette opération d'uniformisation sera appliquée à toutes les images traitées par la suite.

5-3-2. Réduction de la redondance spatiale

L'étude des distributions de chaque niveau d'une image décomposée révèle une caractéristique intéressante de la décomposition en pyramide d'ondelettes. En effet, alors que les distributions des valeurs de pixels des images d'origines ont une forme multimodale, celles des images qui composent la pyramide ont une forme particulière. En l'occurrence, elles sont unimodales. Cependant, l'image de faible résolution retrouve presque la même distribution que l'image d'origine, figure (5.5b). Ce type de fonction de distribution peut être approchée par des modèles mathématiques connus comme on le verra plus loin.

Vu que la distribution de ces coefficients n'est pas uniforme, on procède à des mesures d'entropies pour évaluer l'information contenue dans chaque niveau de la pyramide. La mesure de la valeur de l'entropie d'un signal est définie par :

$$e = -\Sigma p_i \log_2 p_i$$

Où l'indice i décrit l'ensemble des valeurs des coefficients et les p_i représentent la probabilité d'occurrences du niveau i des coefficients.

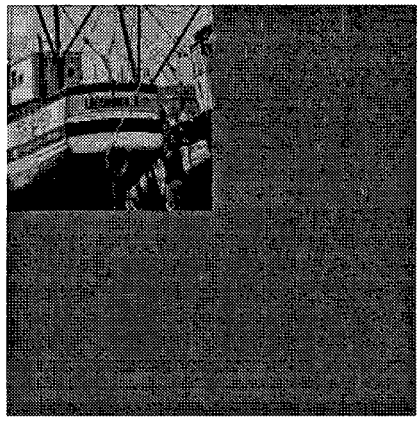
Le tableau suivant donne un exemple de mesure d'entropie des sous images d'une

pyramide d'ondelette.

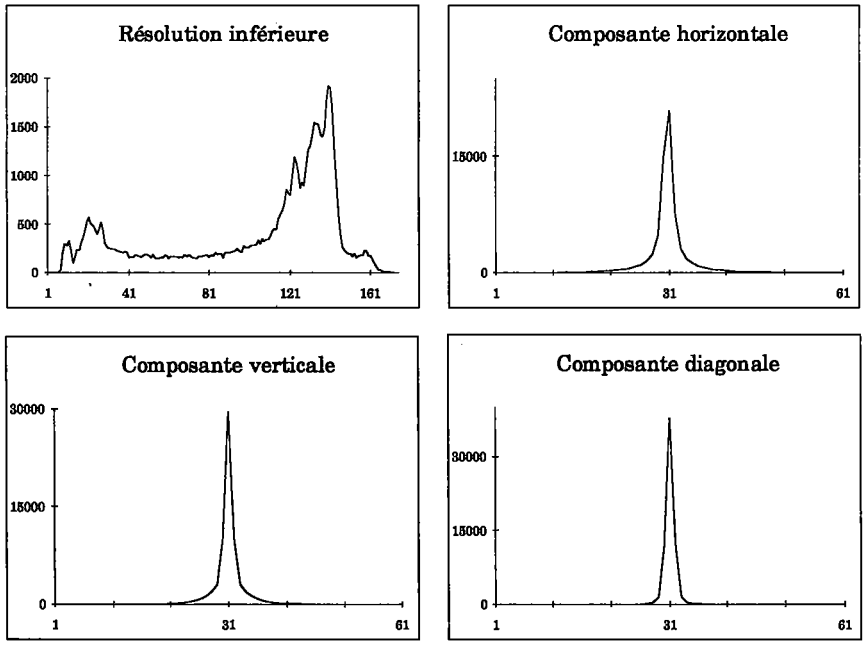
Tableau 5.1 : Entropie des différents signaux qui constituent la pyramide.

Niveau	d1v	d1h	d1d	d2h	d2h	d2d	d3v	d3h	d3d	s-3	Total	Im. Orig.
Entropie	4.15	3.26	3.45	4.07	3.69	3.18	4.60	4.20	3.62	7.29	3.71	7.52

On constate que la représentation en pyramide d'ondelettes réduit d'une façon remarquable l'entropie du signal. Elle améliore nettement la valeur de l'entropie totale par rapport à celle obtenue avec les pyramides Laplaciennes (cf tableau 1.2). Une première étape vers la compression concerne la réduction de la redondance spatiale. Cette étape est bien assurée par une décomposition en pyramide d'ondelettes comme le montre le tableau ci-dessus.



a



b

Figure 5.5b: a) Exemple première décomposition d'image et b)les distributions relatives aux différents signaux.

5-3-3. Effet des filtres sur l'entropie des coefficients

Une étude comparative sur des sous-images des pyramides produites par différents bancs de filtres montre que les valeurs d'entropies sont relatives à chacun de ces derniers. Ainsi, comme l'illustrent les tableaux (5.2) et (5.3), il existe une disparité entre les résultats obtenus par les différents bancs de filtres. On obtient de meilleures performances dans les exemples présentés avec la famille des filtres F11 et F24.

Puisqu'on travaille dans une optique de compression d'image. Les filtres les mieux adaptés sont ceux qui fournissent les meilleures entropies, en occurrence le filtre F11. C'est un banc de filtre QMF qui réalise une décomposition en pyramide d'ondelettes orthogonale. Il est composé de deux filtres de taille 23x23 coefficients. Un autre banc de filtre qui donne un résultat comparable, est celui noté F24. Il réalise une décomposition en pyramide d'ondelettes biorthogonales. Ces filtres sont de taille 9x15 coefficients. La troisième performance est réalisée par le banc de filtre biorthogonal F23, de taille 5x7.

Tableau 5.2: Valeurs d'entropies des différents niveaux de pyramide d'ondelettes en fonction de différents filtres. Femme 256x256

Niveau	F11	F21	F22	F23	F24	F31
d1v	4.10	4.58	4.17	4.15	4.10	4.15
d1h	2.91	3.69	3.23	3.26	3.03	3.26
d1d	3.41	3.94	3.52	3.45	3.43	3.45
d2v	4.10	4.95	4.13	4.07	4.02	4.07
d2h	3.53	4.52	3.69	3.69	3.55	3.69
d2d	3.30	4.43	3.43	3.18	3.22	3.18
d3v	4.70	5.57	4.66	4.60	4.64	4.60
d3h	4.12	5.09	4.18	4.20	4.08	4.20
d3d	3.84	5.09	3.82	3.62	3.73	3.62
s-3	7.32	7.33	7.29	7.29	7.31	7.29
Total	3.60	4.28	3.75	3.71	3.62	3.71

Tableau 5.3: Valeurs d'entropies de différentes niveaux de pyramide ondelette en fonction de différentes filtres. bateau 512x512

Niveau\Filtre	F11	F21	F22	F23	F24	F31
d1v	3.26	3.78	3.39	3.34	3.26	3.34
d1h	2.71	3.40	3.02	3.05	2.84	3.05
d1d	1.70	2.45	1.99	1.96	1.82	1.96
d2v	3.94	4.63	3.87	3.76	3.80	3.76
d2h	3.75	4.51	3.78	3.71	3.69	3.71
d2d	2.76	3.87	2.90	2.72	2.72	2.72
d3v	4.28	5.02	4.07	3.99	4.07	3.99
d3h	5.12	4.22	4.18	4.16	4.18	4.25
d3d	3.34	4.60	3.34	3.18	3.24	3.18
s-3	6.76	6.91	6.70	6.70	6.73	6.70
Total	2.86	3.56	3.05	3.01	2.90	3.01

Il faut noter que le banc de filtre QMF orthogonal a donné la meilleure entropie. Mais en contre partie, il a une taille beaucoup plus grande que ceux des bancs F24 et F23 de la famille

des filtres biorthogonaux. Le rapport débit/coût peut intervenir dans le choix des filtres.

On a présenté l'effet des différents filtres utilisés sur l'entropie des coefficients d'ondelettes, dans la section suivante on mesure l'impact de ces mêmes filtres sur l'image reconstruite. Cette caractéristique est un second critère de choix de filtres.

5-3-4. Impact sur l'image reconstruite

La reconstruction parfaite de l'image est évidemment recherchée. Cependant, elle ne sera pas réalisée vu que les coefficients d'ondelettes ont subi une quantification scalaire. Le fait de quantifier les données en sortie du codeur se traduit par la superposition au signal d'une fonction erreur appelée bruit de quantification. Ce bruit produit une légère distorsion de l'image reconstruite. Plus encore, cette distorsion croit avec le nombre de niveaux de décomposition. Ceci est normal, puisque au moment de la reconstruction, l'erreur de quantification des coefficients d'ondelettes d'un niveau donné se répercute automatiquement sur l'image de résolution supérieure.

Ces répercussions des erreurs entre niveaux sont également liées aux filtres utilisés. La figure (5.6) donne des exemples d'erreurs de reconstruction pour différents filtres et pour différents niveaux de décomposition.

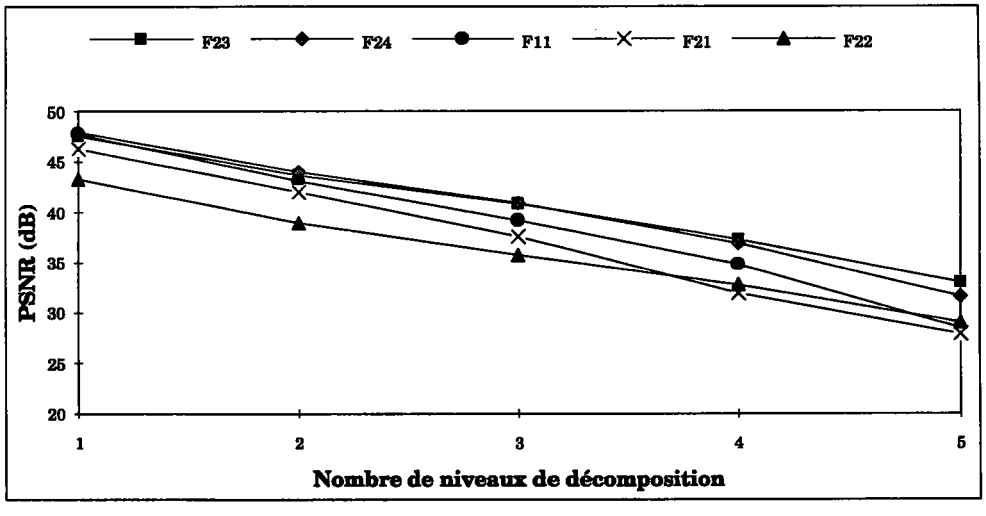


Figure 5.6 : Effets des différents filtres sur l'image reconstruite.

Avoir une meilleure qualité devient l'objectif à atteindre à la reconstruction de l'image. Il faut tenir compte d'un nouveau facteur pour distinguer entre les filtres. C'est le facteur rapport qualité/coût. La qualité d'une image est une grandeur très subjective. Mais on trouve souvent dans la littérature des formules du type rapport signal à bruit, appelé PSNR de l'anglais "*Peak Signal to Noise Ratio*". Il est donné par:

$$\text{PSNR dB} = 20 \log\left[\frac{255^2}{\text{EQM}}\right]$$

Où EQM désigne *l'erreur quadratique moyenne* entre l'image de départ et l'image reconstruite. Cette mesure de PSNR a servi dans un premier temps d'un moyen de comparaison entre les filtres.

Les filtres les mieux adaptés sont ceux qui donnent de meilleurs PSNR. Dans l'exemple ci-dessus, c'est le banc de filtre biorthogonal F24 qui réalise la meilleure performance, suivi du banc F23 de la même famille enfin F11 de la famille des bancs de filtres orthogonaux. Il faut noter que ce dernier avait donné la meilleure entropie. Aussi, on retrouve les mêmes bancs de filtres que ceux qui ont donné les meilleures entropies.

V-4. Critère de sélection de filtres

Des mesures quantitatives et qualitatives ont été effectuées pour évaluer les performances des différents filtres. Seuls les filtres liés aux ondelettes et déclinés en première partie sont utilisés dans la décomposition et la reconstruction de l'image. Ils ont des effets spécifiques sur les distributions des coefficients de la pyramide résultante. Par conséquent, ceci s'est traduit par des répercussions sur les valeurs d'entropies d'une part et sur la qualité de l'image reconstruite d'autre part.

Face à la multitude des filtres issus des ondelettes, nous étions amenés à définir des critères de choix. Ces critères sont basés sur les mesures quantitatives et qualitatives des performances des différents filtres sur les coefficients engendrés par leur utilisation.

En conclusion, les rapports débit/qualité/coût (entropie, PSNR et longueur des filtres) ont été retenus. Il s'est avéré déterminant pour une discrimination des bancs de filtres. C'est un critère qui répond particulièrement à des objectifs de compression. D'un raisonnement pragmatique, tableau (5.4), découle le classement des bancs de filtres biorthogonaux suivant, F24, F23 et F11, par ordre de préférence.

Tableau 5.4 : classification des filtres en fonction de leurs performances.

	F11	F23	F24
Entropie	1	3	2
PSNR	3	1	2
Compléxité	3	1	2
Total	7	5	6
Classement	3	1	2

V-5. Caractérisation des coefficients

L'intérêt d'une décomposition en pyramide d'ondelettes est de pouvoir attribuer un quantificateur particulier à chacun des niveaux de cette pyramide, de telle sorte que le bruit de quantification soit localisé. Cependant, on ne peut obtenir de codage direct simple et optimal que dans le cas ou les signaux à quantifier possèdent des distributions de probabilité connues ce qui n'est pas toujours le cas.

Si on considère un exemple d'image décomposée en pyramide ondelettes, la représentation des distributions réelles des coefficients ondelettes relative à un niveau donné est illustrée par la figure (5.5b).

L'analyse des distributions relatives à une image décomposée, montre qu'elles ont toutes une allure particulière. Celle-ci ressemble à une distribution Laplacienne. L'approximation par une loi Gaussienne généralisée peut être appliquée à ce type de distribution.

5-5-1. Fonction paramétrique : Gaussienne généralisée

La fonction de densité de probabilité, notée *pdf*, d'une distribution Gaussienne généralisée est donnée par la fonction paramétrique suivante :

$$p(x:\sigma,\beta) = \left[\frac{b\eta(\sigma,\beta)}{2\Gamma(1/\beta)}\right] \exp\{-\eta(\sigma,\beta).|x|\}^{\beta}$$

avec $$\eta(\sigma,\beta) = \frac{1}{\sigma}\left[\frac{\Gamma(3/\beta)}{\Gamma(1/\beta)}\right]^{1/2}$$

Son entropie différentielle est donnée par : $$h_s = -\log_2\left[\frac{\beta\eta(\sigma,\beta)}{2\Gamma(1/\beta)}\right] + \frac{1}{\beta\ln(2)}$$

où $\Gamma(.)$ est la fonction gamma et σ^2 la variance du signal.

Le paramètre *b* détermine la décroissance en exponentielle de la *pdf*. Pour une valeur de β égale à *2*, on a une *pdf* Gaussienne. Pour une valeur de β égale à *1* la *pdf* est du type Laplacienne. Pour des valeurs de β de plus en plus petites, la *pdf* prend des allures de plus en plus pointue.

5-5-2. Application à la pyramide d'ondelettes

En général, on ajuste le modèle distribution Gaussienne généralisée par rapport à une distribution réelle donnée d'un niveau de la pyramide pour obtenir le modèle le plus proche. Il faut alors ajuster le paramètre β de façon à approcher au mieux la distribution réelle, au sens d'un certain critère (par exemple le critère des moindres carré). La figure (5.7) donne un exemple de distribution réelle qui correspond au signal y_{1d} pour $d = 1$ avec le modèle le plus

proche donné pour une valeur de $\beta = 0.85$.

On peut dire enfin qu'une décomposition en pyramide d'ondelettes a au moins le mérite de représenter un signal dont les caractéristiques statistiques sont délicates à cerner par des fonctions avec des propriétés statistiques spécifiques. En plus, le modèle de distribution Gaussienne généralisée est un moyen satisfaisant pour donner une bonne approximation des fonctions de probabilités des distributions des signaux issus de cette décomposition.

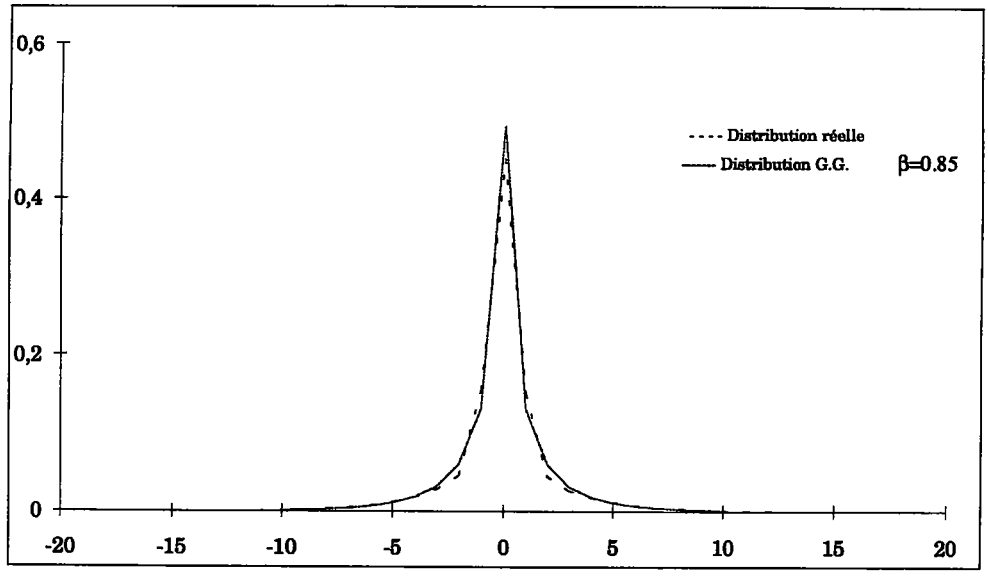


Figure 5.7 : Exemple d'approximation d'une distribution par une Gaussienne généralisée.

V-6. Conclusion

On a montré comment une image peut être décomposée en pyramide d'ondelettes et les avantages qui en découlent sur le plan de la réduction de la redondance spatiale et par conséquent la réduction du débit binaire. Les résultats obtenus sont largement plus performants que ceux de la représentation en pyramide Laplacienne. D'autre part, les distributions étant unimodales, il est possible de les approcher par des lois Gaussiennes généralisées.

Les filtres jouent dans ce cas un rôle très important. Et face à leur diversité, des critères de choix ont été définis. Ils sont essentiellement basés sur les facteurs débit/qualité/coût. En d'autres termes, les filtres doivent satisfaire des exigences telles qu'une bonne entropie à la décomposition, une bonne qualité à la reconstruction et enfin une taille réduite permettant une implémentation simplifiée.

Maintenant que ces données sont cernées, les sections suivantes vont aborder les procédés utilisés pour réaliser l'opération de réduction du débit binaire des pyramides d'ondelettes. Le premier moyen est direct et basé sur le codage scalaire, le second sera construit autour du codage vectoriel.

VI. Pyramide d'Ondelettes et Allocation de Bits

VI-1. Introduction

A ce stade, l'image peut être représentée par une pyramide de sous-images d'ondelettes dont les caractéristiques statistiques sont relativement faciles à définir.

Dans ce chapitre, on introduit d'une part, la quantification scalaire qui permet d'attribuer un certain nombre de bits à chaque sous-image, d'autre part, le problème d'allocation de bits qui en découle. De plus, nous proposons une résolution de ce problème dans le cas du codage scalaire des coefficients de la pyramide d'ondelettes.

VI-2. Quantification scalaire

Une décomposition en pyramide ondelettes engendre plusieurs sous-images. Après la première décomposition, l'image de départ est alors représentée par quatre sous-images. Une sous-image basse résolution (ou basse fréquence) notée y_{10} et trois sous-images détails (ou hautes fréquences) notées y_{1d} (d=1, ..., 3). Où *d* représente les *3* directions privilégiées de la décomposition: horizontale, verticale et diagonale.

Pour coder les signaux ainsi obtenus, on attribue à chaque sous-image un quantificateur scalaire particulier. Soient les signaux y_{1d} et les quantificateurs correspondants Q_{1d} (d=1 à 3). On utilise des quantificateurs de la famille des LCSQ "*Level Constrained Scalar Quantization*", c'est à dire, des quantificateurs scalaires avec des contraintes sur les niveaux de quantification. Un tel quantificateur est basé sur le choix initial du nombre de niveaux de quantification. On note N_{1d} ce nombre attribué au quantificateur Q_{1d} pour coder le signal y_{1d}. Il faut noter qu'à N_{1d} on fait correspondre r_{1d} nombre de bits nécessaires au codage des valeurs à quantifier.

Un quantificateur Q_{1d} appliqué au signal y_{1d}, figure (6.1), génère alors un signal quantifié

noté y'_{1d} et donné par $y'_{1d} = Q_{1d}\{r_{1d}, y_{1d}\}$

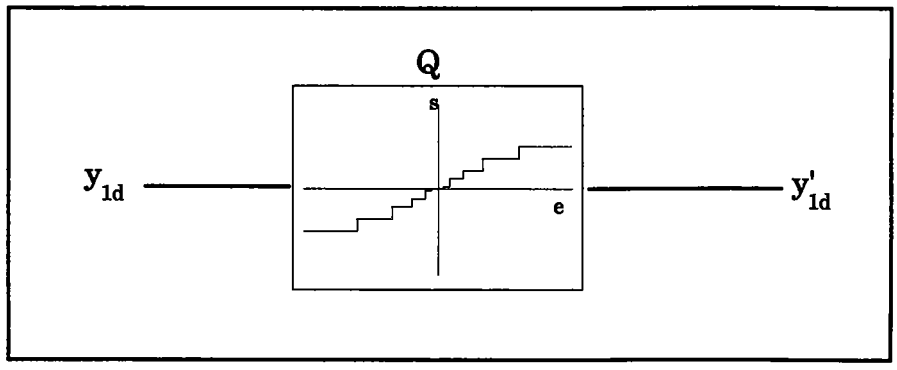


Figure 6.1: Exemple de représentation de quantification scalaire à pas variable

La distorsion introduite se traduit par un bruit e_{1d} de quantification qui est donnée par, $e_{1d} = y_{1d} - y'_{1d}$. Ce bruit manifeste une certaine distorsion du signal d'origine. Si le critère de distorsion est l'erreur quadratique, alors la D_{1d} du signal y_{1d} est donnée par la mesure de l'énergie du bruit de quantification qui peut s'écrire :

$$D_{1d} = E[(e_{1d})^2] = E[(y_{1d} - y'_{1d})^2] \qquad (6.1)$$

On note $D_{1d}(r_{1d})$ la distorsion moyenne produite en quantifiant le signal y_{1d} sur N_{1d} niveaux.

VI-3. Approximation de la distorsion

On a donné ci-dessus une définition de la mesure de la distorsion du signal après quantification. En général, si q(x) est le signal quantifié de x, alors la fonction asymptotique de distorsion, au sens du critère de l'erreur quadratique, est donnée par [Alg66] :

$$D_{SQ}(r) = \frac{1}{12} 2^{-2r} \left[\int [p(x)]^{1/3} dx \right]^{1/3} \qquad (6.2)$$

Où p(x) est la *pdf* de x et r le nombre de bits qui lui sont alloués.

Si on appelle σ la variance de signal x, l'approximation de la variance du bruit du codage est donnée par la performance asymptotique de Gish-Pierce [Gis68] comme suit :

$$D = h\sigma^2 2^{-2r} \qquad (6.3)$$
$$h = \frac{1}{12} \left[\int [p(x)]^{1/3} dx \right]^{1/3}$$

$$h \approx \frac{1}{12} 2^{2h_s} \qquad (6.4)$$

où h_s est l'entropie différentielle définie précédemment (cf paragraphe V-5).

Les y_{1d} sont des signaux de valeur moyenne nulle, de variance $\sigma_{1d}^2 = E[y_{1d}^2]$ et dont la *pdf* est déterminée avec une approximation par des modèles Gaussiens généralisés (voir section précédente). Si on applique les résultats ci-dessus à ces signaux, on peut déduire l'approximation suivante :

$$D_{1d} = h\sigma_{1d}^2 2^{-2r_{1d}}$$

Il est clair qu'à partir de la formule précédente, on peut réduire la distorsion moyenne si on augmente le nombre r_{1d} de bits alloués et vice versa. Un exemple de cette distorsion est illustré par la figure (6.2).

Pour un débit total donné, il faut allouer des bits spécifiquement à chacun des signaux y_{1d} pour d'une part ne pas dépasser le débit total et d'autre part avoir une distorsion optimale. Ceci revient à résoudre le problème d'allocation de bits.

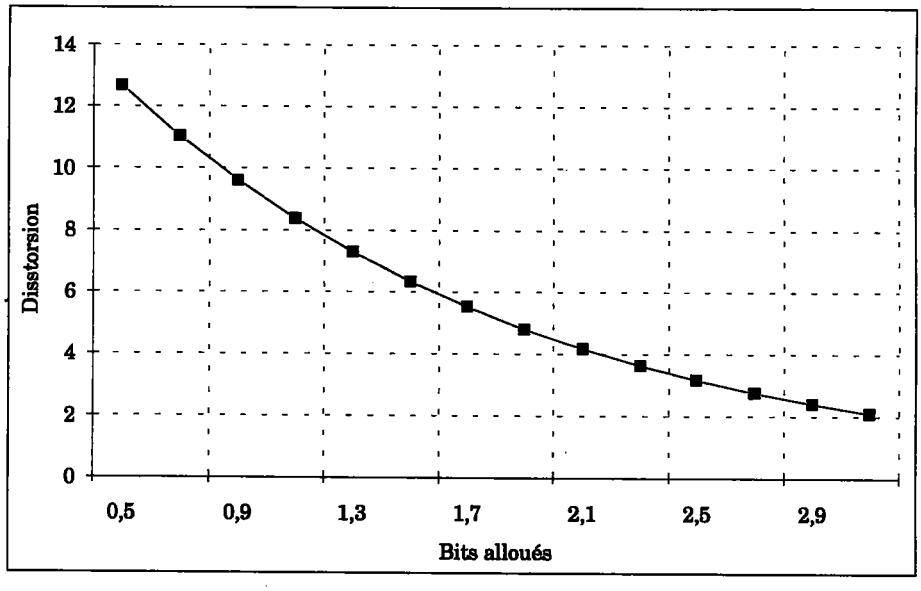


Figure 6.2 : Variation de la fonction distorsion en fonction du nombre de bits alloués.

VI-4. Allocation de bit

Si les distorsions D_{1d} sont relatives aux signaux y_{1d} du premier niveau, on appelle D_1 la distorsion totale produite en codant ces trois signaux. Elle est donnée par :

$$D_1 = \sum_{d=1}^{3} E[(y_{1d} - y'_{1d})^2]$$

On peut écrire ceci en fonction des distorsions relatives D_{1d} :

$$D_1 = \sum_{d=1}^{3} D_{1d}(r_{1d})$$

De même, si les nombres r_{1d} sont relatifs aux signaux y_{1d}, on appelle R_1 le nombre de bit total alloué aux trois signaux. R_1 est donné en fonction des r_{1d} par:

$$R_1 = \sum_{d=1}^{3} r_{1d}$$

Le problème d'allocation de bit est donné par le système suivant :

- minimiser la distorsion totale $D_1(R_1) = \sum_{d=1}^{3} D_{1d}(r_{1d})$
- sous la contrainte $R_1 = \sum_{d=1}^{3} r_{1d}$

Le problème allocation de bit peut être facilement étendu aux autres niveaux de la pyramide à plusieurs décompositions. L'allocation de bit relative au niveau n s'écrit alors :

- minimiser la distorsion $D_n(R_n) = \sum_{d=1}^{3} D_{nd}(r_{nd})$ (6.5a)
- sous la contrainte : $R_n = \sum_{d=1}^{3} r_{nd}$ (6.5b)

La variance du bruit du codage peut être aussi généralisée à ces niveaux par l'approximation asymptotique décrite ci-dessus :

$$D_{nd} = h\sigma_{nd}^2 2^{-2r_{nd}} \tag{6.6}$$

6-4-1. Allocation de bit relative à un niveau

Le problème d'allocation (6.5) peut être résolu en utilisant la méthode des constantes de Lagrange [Woo86b].

Pour un R_n donné, les valeurs de r_{nd} minimisant $D_n(R_n)$ sont données par :

$$r_{nd} = \frac{R_n}{3} + \frac{1}{2}\log_2 \frac{\sigma_{nd}^2}{\rho_n^2} \tag{6.7}$$

On note ρ_n^2 et H_n les moyennes géométriques respectivement des variances σ_{nd}^2 et des h_{nd} (d=1 ... 3):

$$\rho_n^2 = \left(\prod_{d=1}^{3} \sigma_{nd}^2 \right)^{1/3} \tag{6.8}$$

La distorsion partielle optimisée relative à un niveau n s'écrit alors en remplaçant (6.7) dans (6.6) :

$$D_n(R_n) = 3h\rho_n^2 2^{-2\frac{R_n}{3}} \tag{6.9}$$

La relation (6.9) donne la distorsion optimale pour un débit total R_n et la relation (6.7) donne le nombre de bits à attribuer à chacun des signaux y_{nd}.

6-4-2. Allocation globale de bit

6-4-2-1. Problème d'allocation

Le problème précédent d'allocation de bit a été relatif aux signaux détails d'un niveau donné. Il va être étendu à tous les signaux de la pyramide décomposée sur M niveaux.

Dans ce cas, seuls les signaux détails sont à considérer si on suppose nulle la distorsion de l'image basse résolution. La distorsion totale est donnée par :

$$D_T(R_T) = \sum_{n=1}^{M} \frac{1}{2^{2n}} D_n(R_n) = \sum_{n=1}^{M} \frac{1}{2^{2n}} \sum_{d=1}^{3} D_{nd}(r_{nd})$$

Le nombre total de bits alloués à tous les signaux détails de la pyramide est :

$$R_T = \sum_{n=1}^{M} \frac{1}{2^{2n}} R_n = \sum_{n=1}^{M} \frac{1}{2^{2n}} \sum_{d=1}^{3} r_{nd} = \sum_{n=1}^{M} \sum_{d=1}^{3} \frac{1}{2^{2n}} r_{nd}$$

Le nombre de bits total alloués à toute la pyramide à M niveaux, y compris l'image de basse résolution, est donné par $R = R_T + \frac{1}{4^M} R_M$, où R_M est le nombre de bits alloués à cette dernière. Cela revient maintenant à résoudre le problème d'allocation de bit suivant :

- minimiser la distorsion $\quad D_T(R_T) = \sum_{n=1}^{M} \sum_{d=1}^{3} \frac{1}{2^{2n}} D_{nd}(r_{nd})$
- sous la contrainte $\quad R_T = \sum_{n=1}^{M} \sum_{d=1}^{3} \frac{1}{2^{2n}} r_{nd}$

Ce problème peut être résolu de deux façons :

↳ par optimisation directe sur tous les signaux de la pyramide

↳ ou par optimisation sur les niveaux préalablement optimisés.

Les deux propositions sont équivalentes, la deuxième permet de simplifier le problème en se restreignant à l'équation suivante :

- minimiser la distorsion $\quad D_T = \sum_{n=1}^{M} \frac{1}{4^n} D_n \qquad (6.10a)$
- sous la contrainte $\quad R_T = \sum_{n=1}^{M} \frac{1}{4^n} R_n \qquad (6.10b)$

$$\rho^2 = \left[\prod_{n=1}^{M} (\rho_n^2)^{1/4^n}\right]^{1/\xi}$$

Enfin, l'égalité de l'expression (6.14) est vérifiée pour :

$$\rho_n^2 2^{-2\frac{R_n}{3}} = \rho^2 2^{-\frac{2}{3\xi}R_T} \tag{6.16}$$

On déduit alors d'une part la distorsion optimale :

$$D_T = 3\xi h \rho^2 2^{-\frac{2}{3\xi}R_T}$$

D'autre part les nombres de bits R_n qu'il faut allouer à chaque niveau de la pyramide qui correspond à cette distorsion optimale :

$$R_n = \frac{3}{2}\log_2\left[\frac{\rho_n^2}{\rho^2 2^{-\frac{2}{3\xi}R_T}}\right] \tag{6.17a}$$

qu'on peut écrire aussi sous la forme :

$$R_n = \frac{1}{\xi}R_T + \frac{3}{2}\log_2\frac{\rho_n^2}{\rho^2} \tag{6.17b}$$

L'équation (6.17) donne le nombre de bits à allouer à chaque niveau de la pyramide pour un débit total R_T donné, mais surtout, une distorsion totale D_T optimale. Comme la résolution du problème de l'allocation de bits relative à un niveau est déjà effectuée, on peut donner alors le nombre optimal de bits r_{nd} à allouer à chaque composante de la pyramide en fonction de l'allocation de bit totale. En remplaçant (6.17) dans (6.7) on obtient :

$$r_{nd} = \frac{1}{3\xi}R_T + \frac{1}{2}\log_2\frac{\rho_{nd}^2}{\rho^2} \tag{6.18}$$

On obtient ce même résultat en optimisant aussi le problème directement sans passer par l'optimisation partielle par niveau.

6-4-2-3. Application

Le tableau (6.1) donne un exemple d'allocation de bit pour une image décomposée en trois niveaux. Le nombre moyen de bits alloués à tous les signaux détails est égal à 2. On peut aussi distinguer les résultats de l'allocation partielle.

Tableau 6.1 : Exemple d'allocation de bit partielle et totale pour un R_T =2.

	σ^2_{nd}	ρ^2_n	ρ^2	R_T	R_n	r_{nd}	N_{nd}
1h	34,48					2,52	5,76
1d	10,77	14,82			5,75	1,68	3,22
1v	8,76					1,53	2,90
2h	46,77					2,74	6,71
2d	29,49	25,38	17,38	2	6,91	2,41	5,33
2v	11,85					1,75	3,38
3h	90,82					3,22	9,35
3d	56,79	48,85			8,33	2,88	7,39
3v	22,60					2,22	4,66
total						2	
total					2		

VI-5. Conclusion

Introduire une quantification scalaire des différents signaux qui constituent la pyramide et l'approximation de la distorsion par la performance de Gish-Pierce nous ont conduit au problème de l'allocation. La résolution de ce problème par une reformulation des inégalités des moyennes a été proposée. Elle a permis d'attribuer de façon optimale des débits binaires relatifs à chacune des sous-images de la pyramide d'ondelettes.

Les approximations des distorsions produites par les quantificateurs dans cette section sont supposées optimales. Dans la section suivante, on soulève ce problème et on propose un algorithme de quantification conçu sur les principes de bases d'une quantification optimale.

VII. Quantification Scalaire Optimale

VII-1. Introduction

Dans le paragraphe précédent on a montré comment résoudre le problème d'allocation de bit correspondant à chaque sous-image de la pyramide pour générer un débit total optimal pour coder toute la pyramide.

Attribuer un nombre de bits revient aussi à définir un certain nombre de niveaux représentatifs des échantillons du signal. En d'autres termes, le signal est quantifié avec un nombre de niveaux de quantification préétablis.

Si ces niveaux ne sont pas placés de façon régulière, le quantificateur n'est pas uniforme. Et pour un nombre de niveaux donné, il y a de multiples façons de les placer pour quantifier le signal. Notre objectif est de les placer d'une manière optimale. Ceci sera l'objet de cette section.

Le but de la quantification optimale et de minimiser le bruit de quantification suivant un critère donné. Le critère utilisé est la minimisation de la distorsion au sens de la distance des moindres carrés, ou erreur quadratique moyenne.

VII-2. Principe de la quantification optimale

Un quantificateur dont l'erreur quadratique moyenne EQM produite est minimale est un quantificateur optimal suivant ce critère. On veut décrire alors les équations qui gèrent cette optimisation pour développer ce type de quantificateur.

Un quantificateur peut être défini par les variables suivantes :

- p(x) est la fonction densité de probabilité notée *pdf* de x,
- x(i) sont les niveaux de décision,
- q(i) sont les niveaux de sortie.

Si on suppose que la *pdf* du signal *p(x)* est connue, alors il est normal d'attribuer le plus de niveaux de quantification pour les valeurs du signal où la distribution est la plus élevée.

Si N est le nombre de niveaux de quantification, ceci revient à partager la sortie du signal en N intervalles de quantification. Chaque intervalle est délimité par les niveaux de décision x(i).

Les niveaux q(i) déterminent les niveaux en sortie du quantificateur relatif aux intervalles de décision. Ils représentent les valeurs quantifiées du signal en entrée.

VII-3. Quantification Optimale

On va établir les équations élaborées par J. Max et qui gèrent les principes d'un quantificateur optimal. Elles sont basées sur la minimisation de l'EQM.

A toute valeur x dans l'intervalle [x(i), x(i+1)[, correspond une valeur q(i) de quantification. Dans ce cas l'erreur e(i) de quantification est donnée par :

$$e(i) = x(i) - q(i)$$

et l'erreur quadratique moyenne est alors :

$$EQM = \sum_{i=1}^{N} e(i) = \sum_{i=1}^{N} \int_{x(i-1)}^{x(i)} (x - q(i))^2 p(x) dx \qquad (7.1)$$

Minimiser EQM revient à minimiser chaque terme de la sommation par rapport à x(i) d'une part, et par rapport à q(i) d'autre part. Ceci revient à résoudre le système suivant [Max60] :

$$\frac{\partial EQM}{\partial x(i)} = 0 \quad \text{pour } i=1 \ldots N-1; \qquad (7.2a)$$

$$\frac{\partial EQM}{\partial q(i)} = 0 \quad \text{pour } i=1 \ldots N; \qquad (7.2b)$$

Ceci conduit au système d'équation :

$$(x(i) - q(i-1))^2 - (x(i) - q(i))^2 = 0$$

$$\int_{x(i)}^{x(i+1)} (x - q(i))^2 p(x)dx = 0$$

La solution est donnée par :

$$x(i) = 1/2\,[q(i-1) + q(i)] \qquad (7.3a)$$

$$q(i) = \frac{\int_{x(i)}^{x(i+1)} xp(x)dx}{\int_{x(i)}^{x(i+1)} p(x)dx} \qquad (7.3b)$$

A partir de ce dernier système d'équation, on peut déduire d'une part que q(i) est le barycentre de p(x) dans l'intervalle [x(i), x(i+1)[et d'autre part que les niveaux de décision sont les moyennes des valeurs de quantification adjacentes. Ce système donne les équations de base d'une quantification optimale. Les algorithmes suivants se basent sur ces équations pour développer des quantificateurs optimaux.

7-3-1. Algorithme Lloyd-Max

La solution proposée par Lloyd [Llo82] pour élaborer un algorithme de quantification, connu sous le nom "Lloyd-Max algorithm", est la suivante :

étape ❶ : Donner des valeurs initiales à x(i) et q(i) i=1,...,N; k=1.
étape ❷ : Trouver q(k) le centroïde de l'intervalle [x(k-1),x(k)].
étape ❸ : Trouver x(k) la moyenne q(k),q(k+1).
étape ❹ : Si k=N, aller à l'étape 5; sinon mettre k = k+1 et aller à l'étape 2.
étape ❺ : Déterminer c centroïde de [x(N-1), x(∞)[. Si |q(N)-c|<ε, stop. sinon aller à 6.
étape ❻ : Poser q(N) = q(N)-α(q(N)-c) et k=1. Aller à l'étape de 2.

L'inconvénient de cet algorithme est que, d'une part il est mal adapté aux signaux discrets; d'autre part, si on change le nombre de niveaux de quantification il faut réitérer l'ensemble des équations et l'ensemble des étapes de l'algorithme. Mais les équations d'optimisation restent toujours valables. Une alternative à cet algorithme est celle proposée par Nitadori qu'on présente dans la suite.

7-3-2. Algorithme de Nitadori

Le grand avantage de l'algorithme de Nitadori [Nit65] est la simplicité de son implémentation. Il est basé sur un calcul itératif pour générer tous les niveaux de décision x(i) et les niveaux de quantification q(i) d'un signal à distribution Laplacienne connue.

7-3-2-1 Description de l'algorithme de Nitadori

Pour un quantificateur à N niveaux (N= 2L) et pour une distribution à variance unitaire, Nitadori a montré qu'on peut calculer explicitement les seuils de décision x(i) et les valeurs de sortie q(i) du quantificateur optimal. Ces valeurs sont données par le système d'équations suivant :

$$q(i) = \alpha_L + 2\sum_{j=i}^{L-1}\alpha_j \qquad i= 1 \ldots L \qquad (7.4a)$$

$$x(i) = \alpha_L + 2\sum_{j=i+1}^{L-1}\alpha_j + \alpha_{L-1} \qquad i= 1 \ldots L\text{-}1 \qquad (7.4b)$$

Les éléments de la séquence $\{\alpha_i, i=1, 2, \ldots\}$ peuvent être déterminés d'une manière récursive par la relation suivante :

$$(1-\sqrt{2}\alpha_{i+1})e^{\sqrt{2}\alpha_{i+1}} = (1+\sqrt{2}\alpha_i)e^{-\sqrt{2}\alpha_i} \qquad (7.5)$$

avec

$$\alpha_0 = 1/\sqrt{2}$$

7-3-2-2. Exemple

La figure (7.1) donne un exemple d'un signal quantifié avec l'algorithme de Nitadori. On note que la performance de cet algorithme est nettement supérieure à celle d'un codage par quantification scalaire uniforme.

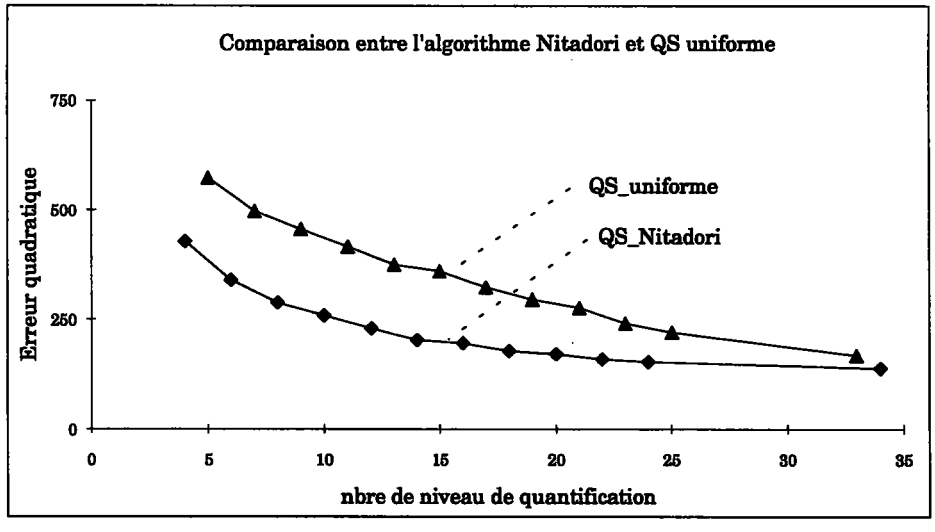


Figure 7.1 : Exemple de performance de l'algorithme de Nitadori et de la quantification scalaire uniforme.

Un inconvénient de cet algorithme est qu'il est spécifique aux seules distributions de type Laplacienne. Ceci n'est pas gênant dans le cadre de nos applications, vu que les signaux traités ont des distributions proches des Laplaciennes. L'algorithme que nous proposons dans la suite, affiche une meilleure adaptation à ces distributions et améliore les performances de l'algorithme de Nitadori. Nous l'avons appelé "algorithme des intervalles glissants".

7-3-3. Algorithme des intervalles glissants

L'exemple précédent donne de bons résultats bien qu'il soit approprié à un type donné de distribution, en l'occurrence les distributions Laplaciennes. Nous proposons dans cette section un algorithme de recherche automatique des intervalles de décision et de leur barycentre. Cette méthode s'adapte parfaitement aux types de distributions autres que Laplaciennes, en particulier les distributions Gaussiennes généralisées étudiées précédemment.

7-3-3-1. Principe

Soit N le nombre de niveaux de quantification (N = 2L+1). Ils sont représentés par [x(i), x(i+1)[les intervalles de décision et q(i) les niveaux de sorties relatifs à ces intervalles.

L'algorithme proposé est basé sur l'interactivité entre les q(i) et les x(i). C'est à dire, si on connaît les q(i) on détermine alors les [x(i), x(i+1)[et vice versa.

Cet algorithme est basé sur un phénomène de glissement de valeurs, une fois les valeurs de q(i) et une fois celles de x(i) connue. Au départ, tous les q(i) sont placés centrés à l'origine. Cela n'est pas un hasard, mais c'est lié au type de distribution traitée, les distributions de la famille des Gaussiennes centrées sur l'origine.

Au début les x(i) sont confondus avec les q(i). On commence par faire glisser, au départ, q(L) de l'extrémité, de sorte qu'ils vérifient la propriété (7.3b), à savoir que q(L) est le centre de gravité de l'intervalle [x(L), x(∞)[. Suite à cette modification, il faut réactualiser la valeur de x(L) suite à la modification de q(L). Elle doit vérifier la propriété (7.3a), c'est à dire que x(L) est la moyenne des valeurs q(L-1) et q(L) , figure (7.2).

A la seconde itération, deux valeurs vont glisser, q(L-1) et q(L) pour s'adapter aux nouveaux intervalles de décision [x(L-1), x(L)[et [x(L), x(∞)[. Ces intervalles ont été modifiés à la première itération suite à la modification de la valeur de x(L). Après cette opération, c'est au tour des valeurs x(L-1) et x(L) d'être réactualisées par rapport aux nouvelles valeurs de q(L-1) et q(L).

Ce sont des opérations en cascades, c'est à dire, un point traité dans une itération fait intervenir un point adjacent à l'itération suivante. Ces opérations sont réitérées jusqu'à ce qu'on atteigne le point q(0) le centre des q(i). q(0) prend comme valeur la moyenne de la distribution qui est égale à zéro dans notre cas.

Les deux opérations principales de cet algorithme seront alors :

- *Intervalles de décision*

Les intervalles [x(i), x(i+1)[sont définis de sorte que les x(i) vérifient la relation suivante:

x(i) est la moyenne des valeurs q(i-1) et q(i)

- *Valeurs de quantification*

Les intervalles [q(i-1), q(i)[sont définis de sorte que tout q(i) vérifient la relation suivante :

$$q(i) = \frac{\sum_{x(i)}^{x(i+1)} xp(x)}{\sum_{x(i)}^{x(i+1)} p(x)}$$

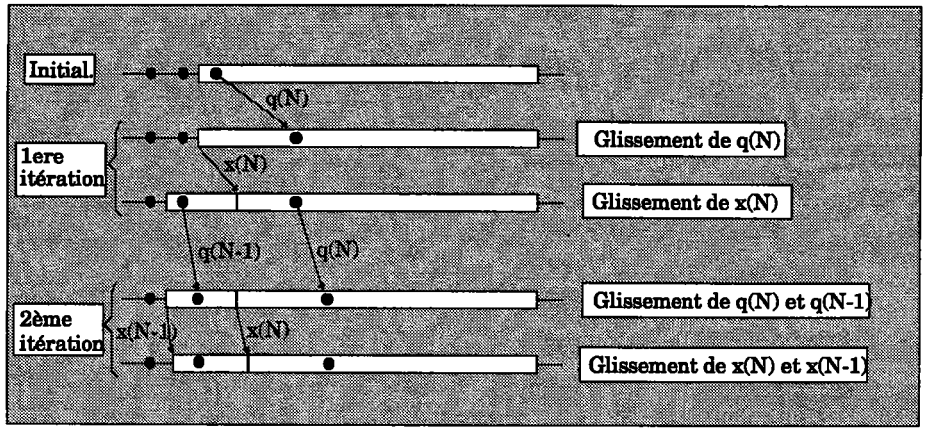


Figure 7.2 : Exemple des premières itérations de l'algorithme des intervalles glissants.

7-3-3-2. Algorithme des intervalles glissants

étape ❶ Initialisation des valeurs de q(i) et x(i) centré sur l'origine, i=0,...,N; et k=L.

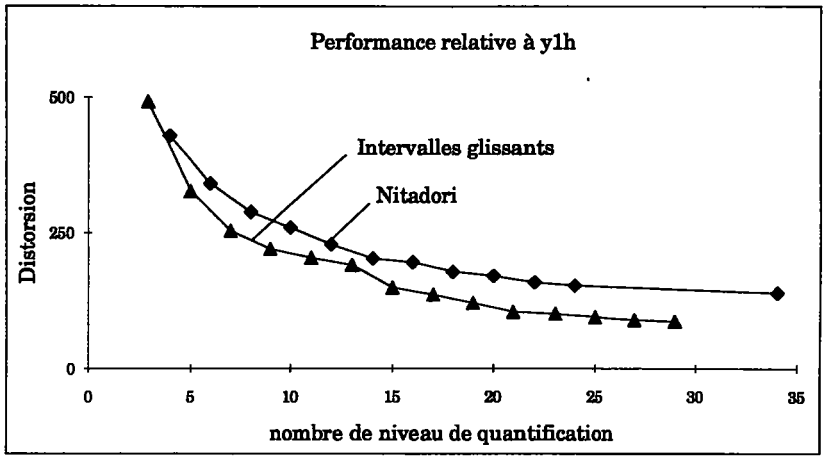
étape ❷ Pour i>k, déterminer q(i), les barycentres des intervalles [x(i), x(i+1)[.

étape ❸ Pour i>k-1, déterminer x(i), les moyennes des valeurs de q(i-1) et q(i).

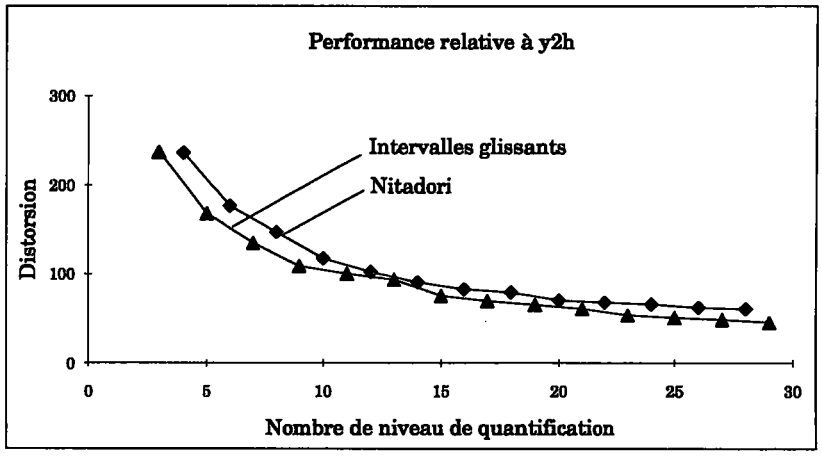
étape ❹ Poser k=k-1, si k=0 stop; sinon aller à l'étape ❷.

La figure (7.3) donne un exemple d'application de cet algorithme à des distributions d'allures différentes, en particulier de plus en plus serrées au centre. On donne au même titre les performances de l'algorithme de Nitadori.

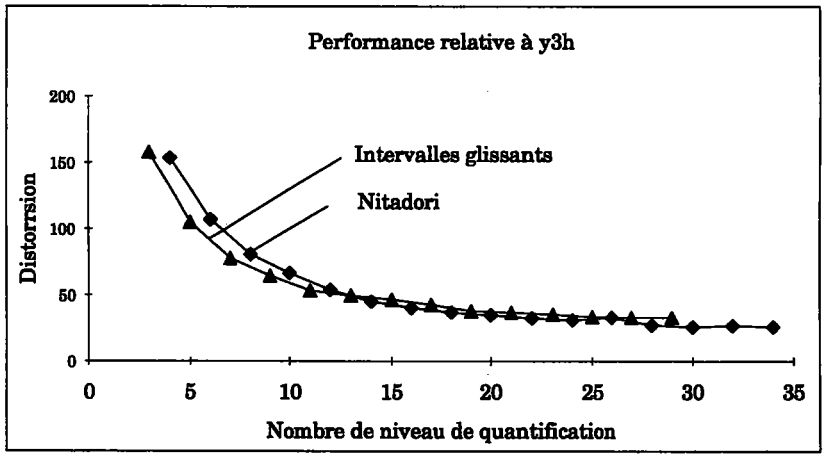
La comparaison des deux résultats montre que les deux algorithmes ont des performances équivalentes pour des distributions larges, figure(7.3a). Mais l'algorithme des intervalles glissants se distingue dans le cas des distributions en pic, figure(7.3b,c).



a



b



c

Figure 7.3 : Comparaison des performances de l'algorithme de Nitadori et d'intervalles glissants, pour différentes distributions relatives aux signaux a)y_{1h}, b)y_{2h} et c)y_{3h}.

VII-4. Application

Dans l'exemple de la figure (7.4), on applique l'algorithme des intervalles glissants pour coder les signaux d'une pyramide d'ondelettes. Cette dernière est issue d'une décomposition d'une image en trois niveaux. Les niveaux de quantification sont donnés par la résolution du problème d'allocation dans le chapitre précédent. L'exemple choisi est l'optimisation donnée par le tableau (6.1).

a b

Figure 7.4 : a) Image "Femme" d'origine. b) Image "Femme" codée avec l'algorithme des intervalles et suivant l'allocation de bit donnée par le tableau (6.1) pour un R_T=2bpp.

VII-5. Conclusion

Après une introduction des principes de base de la QS optimale, on a présenté les algorithmes de codage de Lloyd-Max et Nitadori, suivi d'une comparaison avec l'algorithme proposé, "Algorithme des intervalles glissants". On obtient alors des performances équivalentes pour des distributions larges et de meilleures performances pour des distributions en forme de pic.

Les algorithmes présentés ont tous pour objet de minimiser la distorsion due à la quantification d'un signal sur un nombre donné de niveaux. Ceci a donné une concentration plus forte des niveaux autour de zéro. Dans la suite, on soulève l'importance des contours et on introduit un codage "complémentaire" au précédent, c'est à dire, un codage où on privilégie les coefficients de faible distribution et où on introduit un codage par zone morte. On sera amené à résoudre dans ce cas le problème d'allocation de bit, qui sera traité par une approche algorithmique.

VIII. Approche Algorithmique d'Allocation de Bit

VIII-1. Introduction

Dans les sections précédentes on a traité le problème d'allocation de bit pour générer un débit binaire optimal pour coder la pyramide ondelette. Ce problème peut se résumer en deux volets :

minimisation de la distorsion :$D = \Sigma_i D_i$

sous la contrainte : $R = \Sigma_i r_i$

Les solutions proposées, pour résoudre ce problème, étaient analytiques puisqu'on pouvait approcher les fonctions D_i pour toutes valeurs de r_i dans le cas d'un codage optimal.

Si maintenant la distorsion n'est pas due à une quantification scalaire, mais à tout autre codage, où l'équation qui gère la distorsion n'existe pas, le problème de l'allocation de bit n'admet pas dans ce cas de solution analytique.

Pour cela, nous avons développé une approche algorithmique pour résoudre ce problème. Nous avons abordé dans un premier temps cette approche suite à une quantification scalaire, puis, l'algorithme est étendu à d'autres codages comme le codage par zone morte.

VIII-2. Cas de la quantification scalaire

Il a été montré précédemment que le problème de l'allocation de bits relatifs à toutes les sous-images de la pyramide peut se traiter de deux façons. La première est partielle et commence par une allocation par niveau. La seconde est directe sur tous les signaux. C'est la première méthode qui est retenue pour développer l'approche algorithmique.

Chaque niveau n de la pyramide est constitué par trois sous-images y_{nd} {d=1,...,3 et n=1,...,M} ou d indique les directions horizontale, verticale et diagonale. Si on quantifie ces

signaux en leur attribuant un nombre de bit r_{nd} (ou un nombre de niveau) ils subissent alors une distorsion au sens d'un certain critère. La figure (8.1) montre des exemples de distorsion au sens de l'erreur quadratique en fonction de nombre de bits r_{nd} alloués.

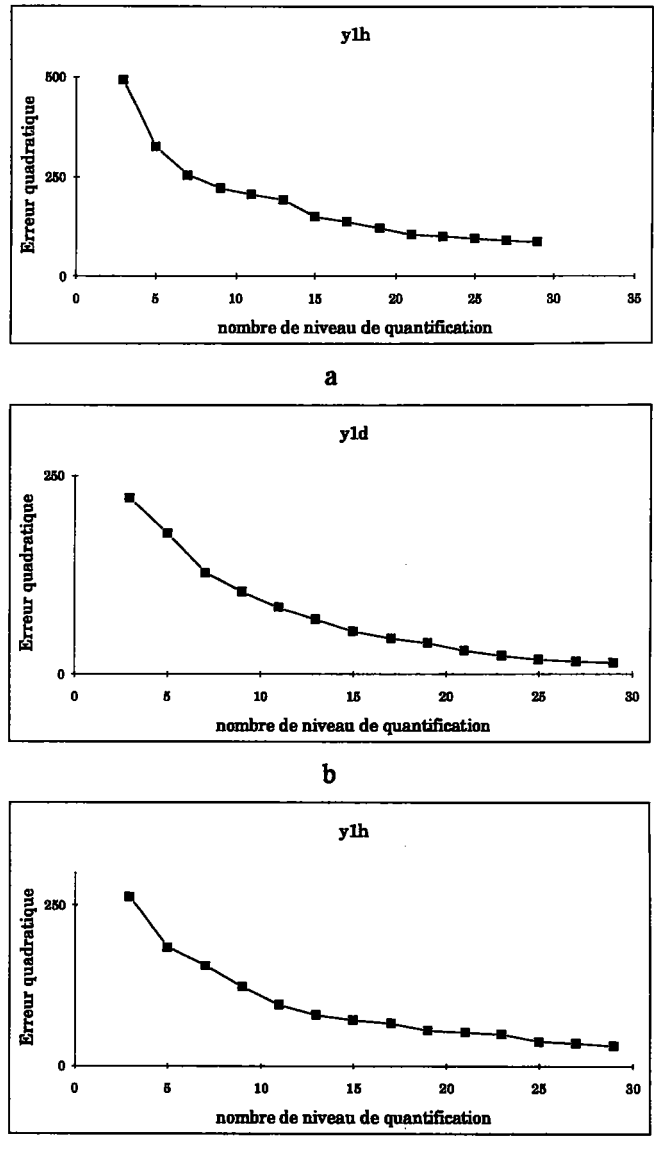


Figure 8.1 : Exemple de distorsion produite sur les différents signaux relatifs au niveau 1 de la pyramide a) y_{lh}, b) y_{lv} et c) y_{ld}.

VIII-3. Allocation de bit par niveau

Comme vu précédemment, la distorsion relative à un niveau n, après quantification des trois signaux $\{y_{nd}; d=1,...,3\}$, est :

$$D_n(R_n) = \sum_{d=1}^{3} D_{nd}(r_{nd})$$

ou r_{nd} est le nombre de bits alloués à y_{nd} et $D_{nd}(r_{nd})$ la distorsion correspondante.

A La distorsion D_n du niveau n on fait correspondre le nombre de bits R_n alloués à ce niveau, $R_n = \sum_{d=1}^{3} r_{nd}$.

La figure (8.2) donne un exemple d'un certain nombre de combinaisons de R_n et les distorsions $D_n(R_n)$ correspondantes. Le résultat obtenu est un nuage de points. Mais ce qu'il faut retenir, c'est l'existence d'une courbe minimale qui n'est autre que la distorsion optimale pour des valeurs de R_n données. Le propos de cette section est d'extraire cette courbe d'une manière peu coûteuse par une approche algorithmique.

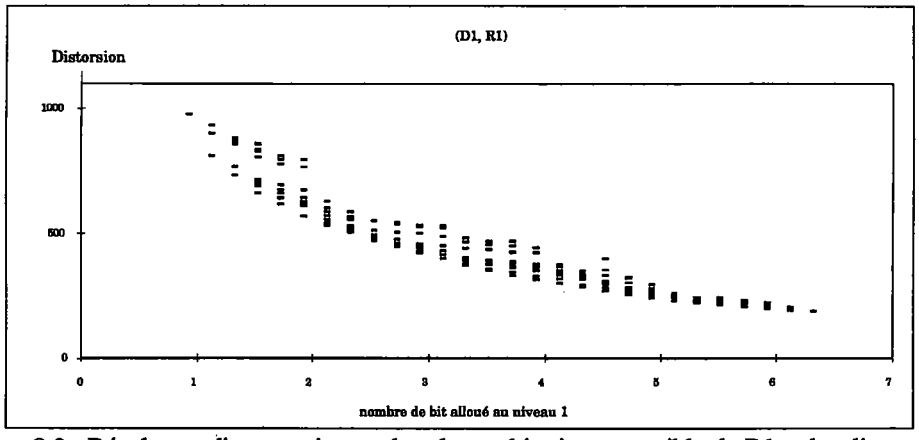


Figure 8.2 : Résultante d'un certain nombre de combinaisons possible de R1 et les distorsions correspondantes. On peut noter l'existence d'une courbe minimale.

8-3-1. Exposé du problème

Si on associe à chaque sous-image y_{nd} un quantificateur Q_{nd}, le problème de l'allocation partielle et optimale relative au niveau n est donné par :

- minimiser la distorsion $D_n(R_n) = \sum_{d=1}^{3} D_{nd}(r_{nd})$ (8.1a)
- sous la contrainte : $R_n = \sum_{d=1}^{3} r_{nd}$ (8.1b)

Pour un nombre R_n de bits donné, on désire déterminer les r_{nd} correspondants sans pour autant tester toutes les combinaisons possibles pour avoir le minimum de la distorsion.

8-3-2. Description de la procédure d'extraction de la courbe minimale

Les trois courbes de distorsion relatives au niveau n sont décrites par les points P_{nd} (figure (8.3a)), chacun de ces points correspond à un couple (D_{nd}, r_{nd}).

La courbe de distorsion optimale est décrite par le point P_n. P_n^c représente le point courant de cette courbe. A ce point correspondent évidemment trois points $\{P_{nd}^c$ d=1,...,3$\}$, figure (8.3).

On pose $P_n^c = (D_n^c, R_n^c)$ ou R_n^c est formé par le triplet $(r_{n1}^c, r_{n2}^c, r_{n3}^c)$ et tel que $R_n^c = r_{n1}^c + r_{n2}^c + r_{n3}^c$.

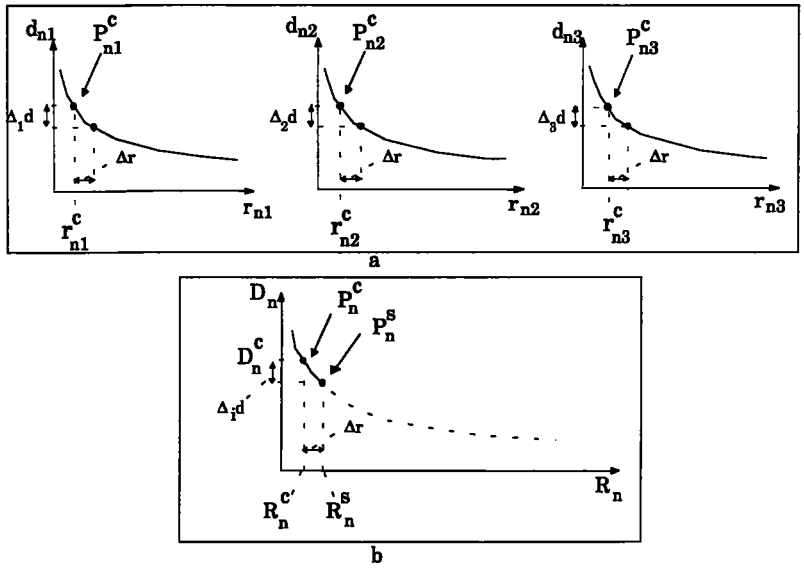


Figure 8.3 : Schéma de scrutation de la courbe optimale. a) les trois courbes relatives au niveau n, b) courbe optimale.

Le point suivant recherché de cette courbe optimale est noté P_n^s. Ce point est déduit à partir du point P_n^c. Si au point P_n^c correspond le couple $(r_n^c, D_n^c(r_n^c))$ au point P_n^s va correspondre un des points P_n^i formés par les couples suivant :

$$\{P_n^i = (D_n^i, \vec{R}_n^i);\ i=1,...,3\}$$

où

$$\vec{R}_n^i = \vec{R}_n^c + \vec{\Delta}_i r$$
$$D_n^i = D_n(\vec{R}_n^c + \vec{\Delta}_i r) = D_n^c + \Delta_i d$$

avec $\Delta_i d = D_n(\vec{R}_n^c + \vec{\Delta}_i r) - D_n^c$ et $\vec{\Delta}_i r$ un des vecteurs d'incrémentation suivants :

$$\Delta_1 \vec{r} = \begin{pmatrix} \Delta r \\ 0 \\ 0 \end{pmatrix};\ \Delta_2 \vec{r} = \begin{pmatrix} 0 \\ \Delta r \\ 0 \end{pmatrix} \text{ et } \Delta_3 \vec{r} = \begin{pmatrix} 0 \\ 0 \\ \Delta r \end{pmatrix}$$

et Δr l'incrément élémentaire.

Pour que P_n^s appartienne à la courbe optimale, il suffit de lui faire correspondre le point P_n^i qui vérifie :

$$P_n^s = \left\{ P_n^i \,/\, \Delta_i d = \max_{j=1..3}(\Delta_j d) \right\}$$

Une fois P_n^s est déterminé, il devient alors le point courant P_n^c. Cette opération est réitérée jusqu'à obtention du point recherché ou de la courbe optimale.

8-3-3. Algorithme

étape ❶ initialisation des P_n^c et $\{P_{nd}^c \; d = 1 \ldots 3\}$ pour un niveau n donné.

étape ❷ déterminer P_n^s tel que :

$$P_n^s = \left\{ P_n^i \,/\, \Delta_i d = \max_{j=1..3}(\Delta_j d) \right\}$$

étape ❸ Si le dernier point est atteint, stop. Sinon $P_n^c = P_n^s$, aller à l'étape ❷

La figure (8.4) donne un exemple d'extraction de la courbe d'allocation optimale relative au premier niveau de la pyramide.

On compare la courbe générée par cet algorithme et celle obtenue par une recherche exhaustive des minimums de toutes les combinaisons possibles. Les deux courbes sont presque identiques, à l'exception de quelques décrochages. Mais la courbe déterminée par approche algorithmique rejoint immédiatement la courbe réelle.

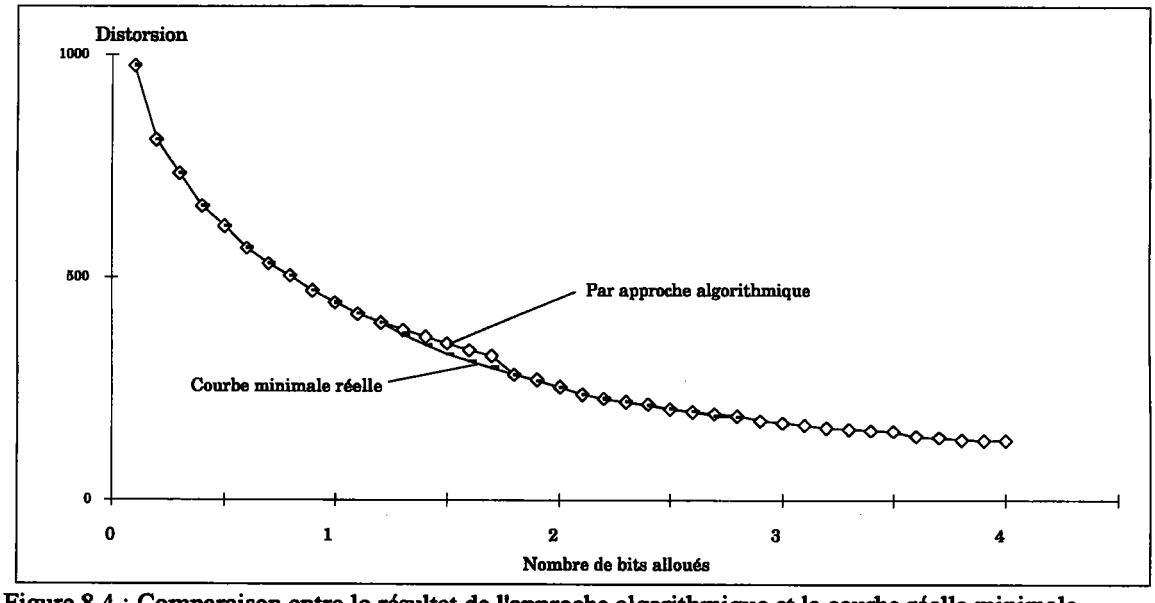


Figure 8.4 : Comparaison entre le résultat de l'approche algorithmique et la courbe réelle minimale.

VIII-4. Allocation totale

Au paragraphe précédent on a déterminé algorithmiquement les courbes optimales relatives à chaque niveau de la pyramide. Le problème de l'allocation totale de bits peut être traité de deux manières équivalentes, soit directement sur tous les signaux de la pyramide, soit partiellement sur chaque niveau de la pyramide.

L'algorithme précédent sera adapté aux courbes optimales d'allocation partielle de bits pour déterminer l'allocation totale de bits optimale. Aussi, il est possible de représenter le nuage de points qu'on obtient en plaçant les points formés par les couples (R_T, D_T), figure (8.5). Nous pouvons aussi remarquer l'existence d'une courbe minimale. Cette courbe est la

courbe d'allocation optimale totale. Ce problème rejoint exactement celui du paragraphe précédent. A part qu'il faut modifier la relation précédente en ajoutant le facteur $1/4^n$ (voir système d'équations ci-dessus)

A partir des courbes optimales par niveau formées par les couples (R_n, D_n), on souhaite obtenir la courbe optimale d'allocation de bit totale vérifiant :

- minimiser la distorsion $D_T(R_T) = \Sigma \frac{1}{4^n} D_n(R_n)$
- sous la contrainte $R_T = \sum_{n=1}^{M} \frac{1}{4^n} R_n$

La figure (8.6) représente la courbe optimale d'allocation de bit totale formée par les couples (R_T, D_T) déterminée par le même algorithme précédent en introduisant le facteur $1/4^n$.

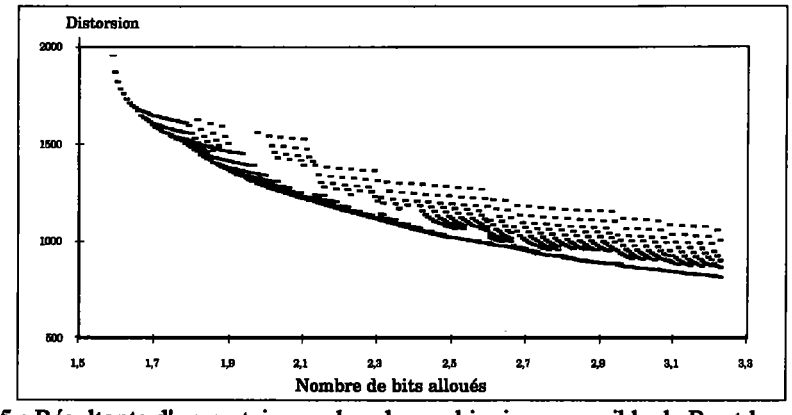


Figure 8.5 : Résultante d'un certain nombre de combinaisons possible de R_T et les distorsions correspondantes. On note toujours l'existence d'une courbe minimale.

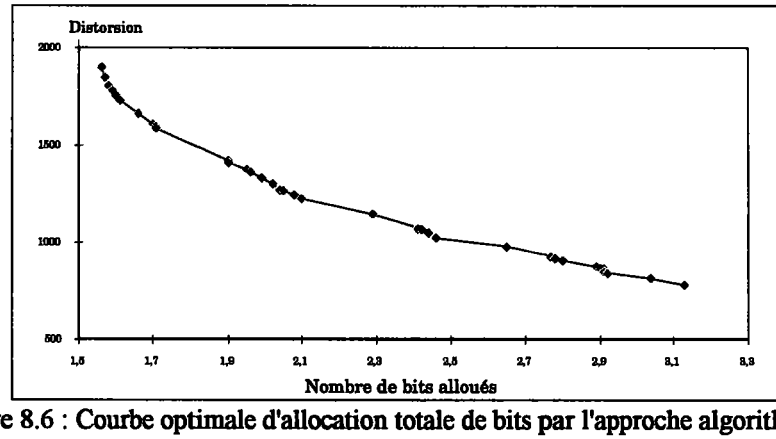


Figure 8.6 : Courbe optimale d'allocation totale de bits par l'approche algorithmique

VIII-5. Conclusion

Cet algorithme permet alors de traiter le problème d'allocation de bit pour des signaux codés avec des techniques dont la distorsion n'est pas approchée d'une manière analytique. On a donné un exemple dans le cas d'un codage scalaire. On verra dans le chapitre suivant l'extension de cette approche algorithmique à d'autre technique de codage, en l'occurrence le codage par zone morte pour lequel l'équation de la distorsion est difficile à établir.

IX. Codage par Zone Morte

IX-1. Introduction

La quantification scalaire décrite précédemment approche au mieux la probabilité de la distribution d'un signal sur un certain nombre de niveaux.

Dans le cas de la QS optimisée, on peut constater que les coefficients sont codés avec des pas et des précisions variables. La précision est plus grande pour les coefficients de petites valeurs qui ont une forte probabilité. Inversement, elle est plus faible pour les coefficients de valeurs importantes mais de faible probabilité.

Or, dans une image, les coefficients de valeurs importantes sont porteurs d'information sur les contours, mais ils ont une probabilité d'occurrence faible. Par conséquent, ils sont codés grossièrement. Un moyen qui permet de les coder avec plus de précision est de négliger ou d'annuler les coefficients de petites valeurs. Ceci permet un codage précis et efficace des coefficients de valeur importante. Pour cela on introduit le principe de "zone-morte".

IX-2. Codage par introduction de zone morte

Les sous-images "haute fréquence" de la pyramide sont constituées de l'information contour qui est plongée dans un bruit de l'image de faible énergie. Ces observations nous ont amené à introduire une zone morte. C'est un intervalle [-z, z] centré sur zéro où toute valeur est mise à zéro. Ceci permet d'éliminer les faibles valeurs équivalentes au bruit de l'image (figure 9.1), et de coder avec plus de précision l'information contour.

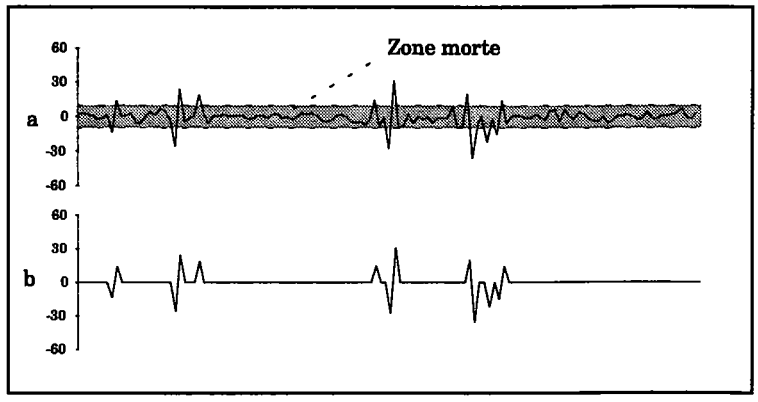


Figure 9.1 : Résultat d'un codage par zone morte.

Comme montré par la figure (9.2), les valeurs de la zone morte sont annulées et les valeurs en dehors de cet intervalle, valeurs actives, sont quantifiées uniformément.

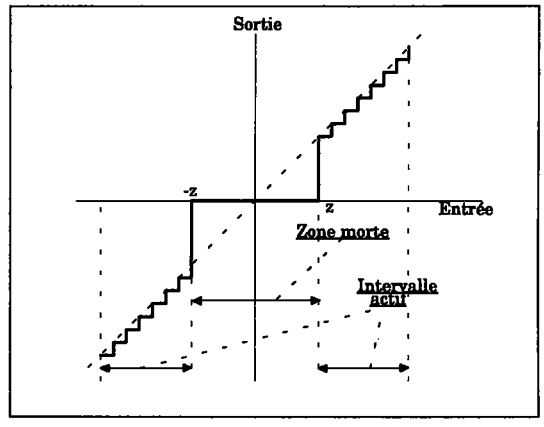


Figure 9.2 : Fonction de transfert d'un codeur par zone morte.

A la suite de cette opération, une distorsion du signal est produite. La variance de cette distorsion peut être déterminée par la variance du signal d'origine dans l'intervalle [-z, z] par $D(z) = \int_{-z}^{+z} x^2 p(x) dx$.

On a vu dans les sections précédentes que pour des sous-images ondelettes une approximation de la *pdf* p(x) est possible avec les fonctions paramétriques Gaussiennes généralisées. Néanmoins, une formulation analytique de D(z), et par conséquent la résolution analytique du problème d'allocation de bit, n'est pas évidente, d'où le recours à une approche algorithmique.

La figure (9.3) donne un exemple de distorsion en fonction de la valeur de l'intervalle de la zone morte des sous-images du premier niveau d'une pyramide d'ondelettes. On note que la distorsion augmente avec la largeur de cet intervalle.

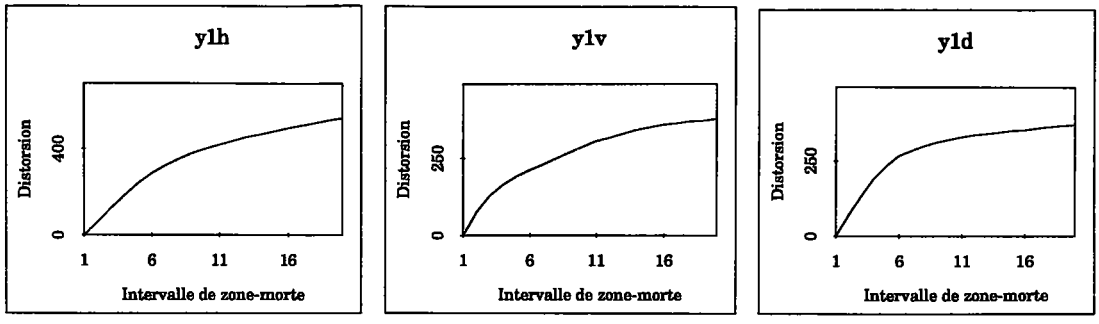


Figure 9.3 : Exemple de distorsion en fonction des différents intervalles de zone morte appliqués du niveau 1 de la pyramide.

IX-3. Allocation de bit pour le codage en zone morte

A défaut d'approximation analytique on est amené à résoudre le problème d'allocation de bit par approche algorithmique. L'algorithme décrit précédemment reste toujours valable. Reste maintenant à définir une relation entre z et le nombre de bits nécessaires pour coder le signal après introduction de zone morte. Le problème n'est pas aussi simple qu'en codage scalaire où la relation est triviale entre le nombre de niveaux de quantification et nombre de bits nécessaires pour coder les échantillons. Une façon d'établir une telle relation est de mesurer l'entropie du signal après introduction de zone morte. Un exemple de relation entre z et le nombre de bits alloués peut être établi par le codage d'Huffman, Figure (9.5).

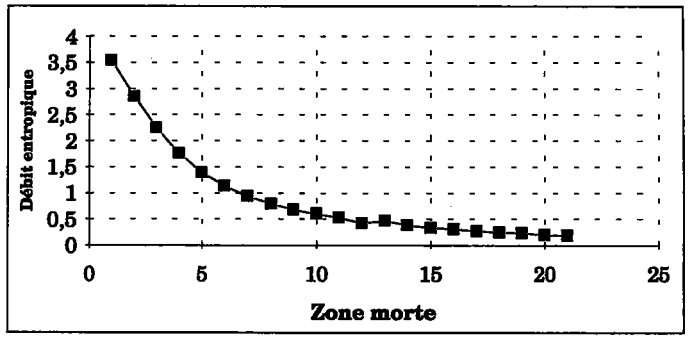


Figure 9.5 : Exemple de relation entre z et le débit binaire entropique

Enfin le problème d'allocation de bit par niveau peut s'écrire :

- minimiser la distorsion $D_n(R_n) = \sum_{d=1}^{3} D_{nd}(b_{nd})$ (1a)
- sous la contrainte : $R_n = \sum_{d=1}^{3} b_{nd}$ (1b)

Où b_{nd} est le nombre de bits alloués au signal y_{nd} (d=1,...,3) du niveau n produisant une distorsion D_{nd}.

Ces relations sont équivalentes à celles que nous avons vues précédemment en QS. La distorsion est définie en fonction du débit binaire entropique, sur lequel, on applique l'approche algorithmique d'allocation de bit. Un exemple de résultat de cet algorithme pour extraire la

courbe optimale de distorsion est illustré par la figure (9.4). Nous pouvons par conséquent déterminer les zones mortes à appliquer à chaque composante de la pyramide de façon à optimiser la distorsion totale. Un exemple de codage est donné dans les paragraphes suivants ainsi qu"une comparaison avec le codage scalaire.

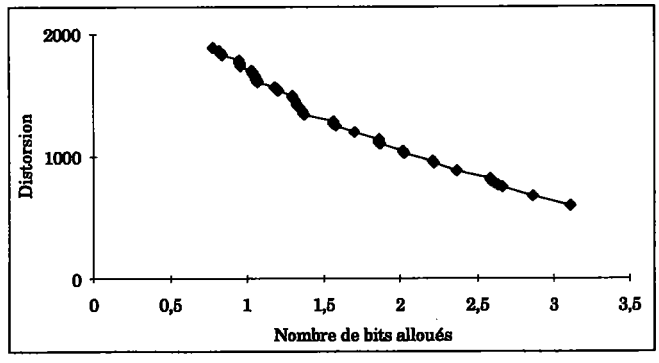


Figure 9.4 : Courbe optimale d'allocation de bits par approche algorithmique pour un codage par zone morte.

IX-4. Comparaison avec le codage scalaire

Pour pouvoir comparer les deux techniques de codage, scalaire et par zone morte, il faut que les valeurs des débits binaires, relatives à chaque codage, soient conformes. Pour cela, on utilise dans les deux cas de codage un débit binaire entropique.

La figure (9.5) donne un exemple de comparaison des deux techniques pour une même image. On peut constater que pour le même débit, le codage par zone morte donne une meilleure performance que celle produite par codage scalaire. La question qu'il faut poser maintenant est : qu'advient-il de l'image reconstruite?

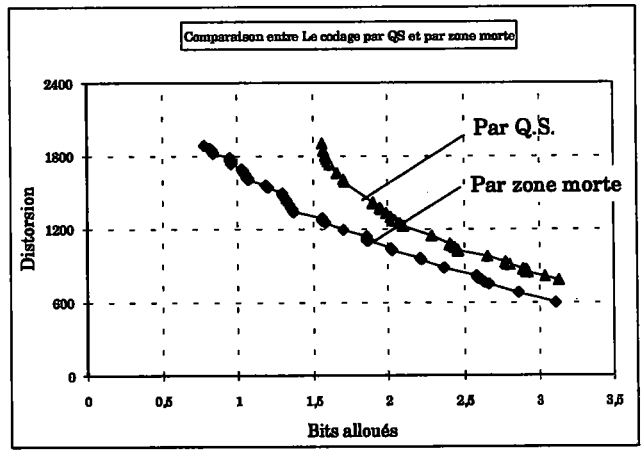


Figure 9.5 : Comparaison entre les deux techniques de codage, par quantification scalaire et par zone morte.

IX-5. Application

La figure (9.6a) donne un exemple d'image codée par un codage avec zone morte. Pour un débit entropique totale de l'ordre 1.2bpp, le PSNR est de 35.50db. L'allocation de bit est réalisée par approche algorithmique. La figure (9.6b) est le résultat obtenu par la même approche pour un codage par quantification scalaire et pour un débit entropique d'environ 1.89bpp. Son PSNR est de 33.20db, inférieur au précédent et confirme les résultats théoriques précédents. Sur le plan subjectif, il est très difficile de départager les deux résultats. Ceci rejoint, ce qui est cité précédemment sur le fait que la qualité de l'image dépend seulement de l'application finale.

a b

Figure 9.6 : Application à l'image des deux techniques de codage, a), par zone morte pour R= 1.2bpp, PSNR = 35.50db, b)par quantification scalaire pour R=1.89bpp. PSNR = 33.20db.

IX-6. Conclusion

On a présenté dans ce chapitre une nouvelle manière de coder les coefficients de la pyramide d'ondelettes, en éliminant une partie des coefficients centrés sur l'origine. Ceci permet un codage plus fin des coefficients de grandes valeurs et de probabilité d'occurrence faible, mais qui représente néanmoins les contours de l'image. Le problème d'allocation de bit a été soulevé et traité par approche algorithmique. En fonction du débit entropique, on obtient des performances en codage par zone morte légèrement supérieures à celle du codage scalaire.

Troisième Partie :

Aspects et Codage Vectoriel

3ème Partie : Introduction

Les nouvelles sources de données, images numériques par exemple, et le besoin grandissant de communication exigent des taux de compression toujours grandissants. Ce qui justifie la recherche de nouvelles techniques encore plus performantes.

Ces dernières années ont vu le développement de plusieurs techniques de QV aussi bien en codage de la parole qu'en codage de l'image. Deux domaines où le problème de réduction du débit de transmission est particulièrement important.

La quantification vectorielle (QV) a trouvé usage il y a seulement quelques années, malgré sa supériorité sur la quantification scalaire, pour des raisons qu'on peut résumer ainsi :

- la simplicité de l'implémentation de la quantification scalaire d'où un certain nombre d'avantage
- les performances de la quantification scalaire ont été jusqu'alors suffisantes.

Contrairement à la quantification scalaire, la quantification vectorielle profite de la corrélation entre les coefficients. Les performances restent toujours supérieures même pour des données patiellement décorrélées. L'intérêt de la quantification vectorielle et visible quand il s'agit de codage des images.

Il paraît évident de substitutier la QV aux techniques classiques non optimales au sens ou elles ne profitent pas de la corrélation entre les coefficients.

Les vecteurs dans ces cas sont formés par un certain nombre des coefficients, de préférence connexe pour avoir un maximum de corrélation en lignes et aussi en colonnes. Ces coefficients forment souvent un bloc dans l'image et la dimension des vecteurs est définie alors en fonction de la taille du bloc.

Mais l'inconvénient majeur de la QV est qu'elle utilise, pour son implémentation, un certain nombre de vecteurs nécessaire au codage. Ces vecteurs sont rassemblés dans ce qu'on appelle un dictionnaire. La fabrication de ce dictionnaire est le principal handicap de l'implémentation des QV.

La famille des quantificateurs scalaires étudiée précédemment est considérée comme une opération de compression avec perte. Il en sera de même pour la famille des quantificateurs vectoriels. La QV consiste alors à quantifier non pas des éléments disparates, mais un groupe d'éléments qui vont former un vecteur. Le terme vecteur ici est très subjectif et les similitudes

entre les opérations utilisées dans ce cadre et celle de l'algèbre linéaire sont vraisemblablement à l'origine de cette utilisation abusive. La QV permet d'obtenir des performances supérieures à celles obtenues par la quantification scalaire. En outre elle est capable de descendre en dessous de la limite de 1 bit/élément.

La quantification vectorielle est l'objet des chapitres suivants. Nous présentons une mise en application d'un codage par quantification vectorielle à des coefficients transformés. L'objet de cette étude n'est pas de faire une liste exhaustive des techniques de QV. Cependant, une description détaillée des QV que nous avons mis en oeuvre sera déclinée. Dans un premier temps, les principes de la quantification vectorielle et leurs applications pour le codage seront exposés au chapitre X. Par la suite, des techniques existantes de formation de dictionnaire seront décrites. Au chapitre XI, une amélioration des performances de recherche des vecteurs dans un dictionnaire sera proposée. Enfin, Au chapitre XII, la quantification vectorielle conjuguée à d'autres techniques de codage sera appliquée à la pyramide d'ondelettes. Mais pour cela, une répartition des coefficients sera nécessaire. Ainsi, nous présenterons la structure de la sous-pyramide, son codage hybride et la notion de classification appliquée à cette structure. Pour finir, au chapitre XIII, l'estimation par correspondance des blocs est introduite pour une extension au codage de séquences d'image. Il y sera aussi question de l'optimisation de la recherche des vecteurs déplacements.

X. Quantification Vectorielle

X-1. Introduction

La QV est l'extension de la QS à un espace à plusieurs dimensions. Elle est classée parmi les techniques de compression avec perte. Elle affiche des performances supérieures à celles de la QS. Cette dernière avait pour avantage la simplicité et ses performances ont été suffisantes jusqu'alors. Les exigences en taux de compression grandissant, pour des applications multiples dans un besoin croissant en communication, la QV apparaît comme une solution nécessaire. De nombreux travaux sur la QV ont vu le jour ces derrières années dans un but de codage, en particulier des images numériques[Nas88].

X-2. Quantification vectorielle

Un quantificateur vectoriel Q de dimension K et de taille N peut être défini comme une application d'un espace vectoriel E de dimension K dans un espace F de même dimension et de taille finie, $F \subset E$. A chaque vecteur X de E, $X = (x_1, x_2, \ldots, x_K)$, on fait correspondre parmi les N vecteurs de F un vecteur Y_i, $Y_i = (y_{i1}, y_{i2}, \ldots, y_{iK})$.

$$Q : \quad E \longrightarrow \hat{A} \subset E$$
$$X \longrightarrow Q(X) = Y_i$$

L'espace vectoriel $\hat{A}$ étant fini, l'ensemble de tous les vecteurs de reproduction $\hat{A} = \{ Y_i ; i = 1 \ldots N\}$, est appelé dictionnaire ou "Codebook" en anglais.

Si le dictionnaire $\hat{A}$ est donné, on peut faire une partition de l'espace E. Une partition de l'espace E est l'ensemble S défini comme suit :

$$S = \{ S_i / i = 1 \ldots N\}$$

avec : $S_i = \{ X \in E / Q(X) = Y_i\}$

de telle sorte que tous les éléments de S_i sont représentés par un seul vecteur de reproduction Y_i du dictionnaire.

Le quantificateur vectoriel Q sera alors complètement défini par la connaissance des N vecteurs Y_i du dictionnaire $\hat{A}$ et la partition S de E. On peut poser alors $Q = \{\hat{A}, S\}$.

L'opération de quantification est la résultante de deux opérations, figure (10.1), une liée au codeur, l'autre liée au décodeur :

•Le codeur effectue les opérations suivantes :

- ➲ il reçoit le vecteur X en entrée,
- ➲ il détermine le vecteur de reproduction Y_i, parmi les N vecteurs du dictionnaire, pour lequel $Q(x) = Y_i$,
- ➲ enfin, il transmet l'indice dans le dictionnaire du vecteur de reproduction.

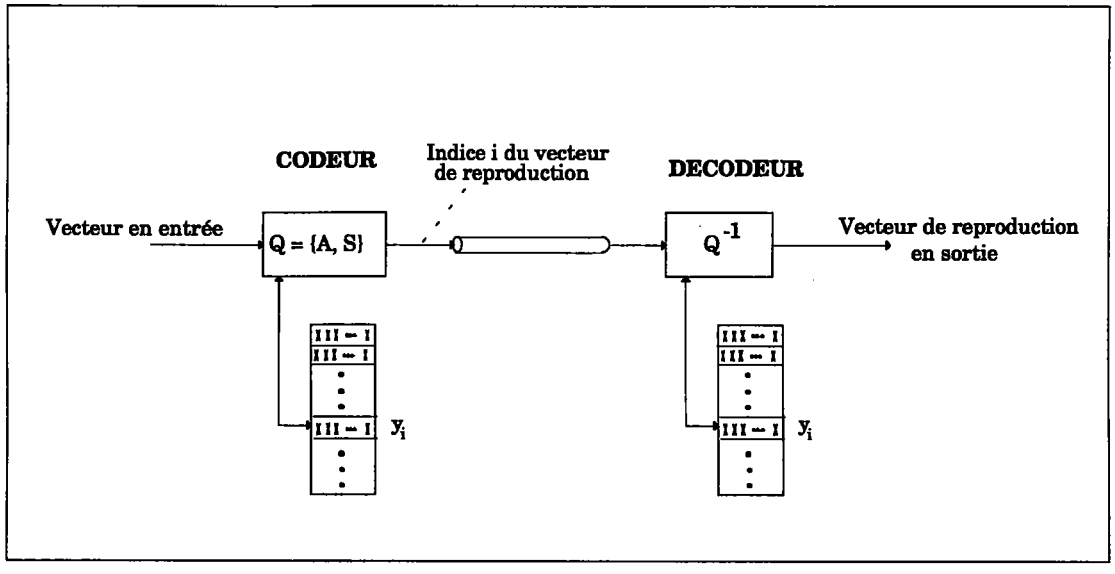


Figure 10.1: Synoptique d'un quantificateur vectoriel.

• le décodeur ayant le même dictionnaire que le codeur :

- ➲ reçoit les indices qui proviennent du codeur,
- ➲ fournit en sortie le vecteur du dictionnaire correspondant à l'indice reçu.

Un vecteur X en entrée est représenté par un vecteur Y_i du dictionnaire. Y_i est dit le plus proche voisin au sens d'un certain critère ou distance d donnée. Le vecteur Y_i vérifie alors la relation suivante :

$$d(X, Y_i) \leq d(X, Y_j) \quad j = 1 \ldots N.$$

Il existe plusieurs distances. La plus utilisée est la distance au sens des moindres carrés. Elle donne en même temps la distorsion ou l'erreur quadratique entre un vecteur X et son vecteur de reproduction Y_i. Elle est donnée par l'expression suivante :

$$d(X, Y_j) = \sum_{i=1}^{K} \left[X(i) - Y_j(i) \right]^2 \qquad (10.1)$$

Le débit binaire ou taux de compression réalisé par l'opération de QV, est lié d'une part à la dimension K des vecteurs et d'autre part à la taille N du dictionnaire. Le nombre moyen de bits R par échantillon nécessaires pour représenter un vecteur du dictionnaire est donné par $R = \frac{\log_2 N}{K}$. Il faut noter que c'est un codage à débit fixe; c'est à dire, que l'on transmet toujours un nombre constant de bits par vecteur en entrée. Pour une dimension donnée des vecteurs, c'est la taille du dictionnaire qui fixe le débit moyen.

Le maillon le plus important de la QV est le dictionnaire. Il existe différentes techniques pour le construire. Deux sont exposées dans les paragraphes suivants. L'une basée sur la méthode LBG, l'autre sur les réseaux de Kohonen.

X-3. Formation de dictionnaire

10-3-1. Méthode LBG

Le but de l'algorithme LBG est de générer un dictionnaire. Pour cela, il faut une séquence de vecteurs qui vont servir de séquence d'apprentissage, appelée aussi séquence d'entraînement, de l'anglais "training set". Cette séquence est formée d'un ensemble de vecteurs représentatifs de l'ensemble des vecteurs à coder.

La procédure de génération de dictionnaire est capitale puisqu'elle est censée fournir le dictionnaire le mieux représentatif de la séquence d'apprentissage au sens d'un certain critère.

L'algorithme, développé par Linde Bruzo et Gray [Lin80], connue sous l'appellation LBG, est fréquemment utilisé pour générer ces dictionnaires. Le critère de ressemblance utilisé est le critère du plus proche voisin. Il est défini par une distance d qui peut être l'erreur quadratique :

$$d(X, Q(X)) = \sum_{i=1}^{k-1} \left| \mathbf{x}_i - Q(\mathbf{x}_i) \right|^2$$

10-3-1-1. Critère de performance d'un dictionnaire

Pour définir la performance d'un dictionnaire on mesure la distorsion totale donnée par $D(Q\{\hat{A}, S\})$, où $\hat{A}$ est le dictionnaire utilisé par le quantificateur Q et S la partition de la séquence d'apprentissage. Elle mesure la distorsion produite en représentant tous les vecteurs d'entrée par les vecteurs de reproduction du dictionnaire. Cette distorsion est donnée par :

$$D\{\hat{A}, S\}) = E[d(X, Q(X))] \quad (10.2)$$

X est un vecteur aléatoire muni d'une pdf $f_x = p\{X_j \leq x_j; j = 1 \ldots k-1\}$

Soit un quantificateur Q donné et $\hat{A}$ son dictionnaire, $\hat{A} = \{ Y_j; j=1 \ldots N\}$, une partition $S(\hat{A})$ de la séquence d'apprentissage associée à ce dictionnaire est donnée par :

$$S(\hat{A}) = \{ S_i / i = 1 \ldots N\} \quad (10.3a)$$
$$S_i = \{ X/ Q(X) = Y_i\} \quad (10.3b)$$

La partition optimale au sens de la distorsion totale est construite en attribuant à chaque vecteur d'entrée X un vecteur Y_i du dictionnaire qui minimise la distorsion $d(X, Y_j)$. La partition optimale, $P(\hat{A}) = \{P_j; j = 1 \ldots N\}$, est définie alors par :

$$P_j = \{ X \in S / d(x, Y_i) \leq d(x,Y_j); j = 1 \ldots N\} \quad (10.4)$$

La performance du dictionnaire, liée à la distorsion totale, est donnée par :

$$\begin{aligned} D(Q) = D(\{\hat{A}, S\}) &= E[d(X, Q(X))] \\ &= E(X/Y_j) / X \in S_j) \Pr(X \in S_j) \end{aligned} \quad (10.5)$$

Par conséquence, $D(\{\hat{A}, P(\hat{A})\}) = E(\min_{Y_j \in \hat{A}} d(X,Y_j))$.

D'où pour n'importe quelle autre partition on a :

$$D(\{\hat{A}, S\}) \geq D(\{\hat{A}, P(\hat{A})\}) \quad (10.6)$$

Ce qui revient à dire que la partition $P(\hat{A})$, construite pour un dictionnaire donné est la meilleure partition possible au sens du critère d'erreur quadratique moyenne.

Inversement, soit $S = \{S_i / i = 1 \ldots N\}$ une partition décrivant un quantificateur. On peut dans ce cas déduire le dictionnaire optimal relatif à cette partition. Ce dictionnaire, noté $\hat{X}(S)$, vérifie alors :

$$E(d(X, \hat{X}(S)) / X \in S) = \min_u E[d(X, u) / X \in S] \quad (10.7)$$

Dans le cas où le critère de distorsion de quantification est l'erreur quadratique moyenne, le dictionnaire $\hat{X}(S)$ est formé alors par l'ensemble des vecteurs $\hat{X}(S_j)$ centroïdes ou centre de gravité de chaque partition S_j.

On déduit que pour une partition donnée $S = \{ S_j; j = 1 \ldots N\}$, aucun dictionnaire autre que $\hat{A} = \{Y_j; j = 1 \ldots N\}$ ne peut donner une performance meilleure que celle réalisée par le

dictionnaire $\hat{X}(S) = \{ \hat{X}(S_j); j = 1 \ldots N\}$. Ceci revient au fait qu'une telle partition donne la distorsion optimale puisque :

$$D(\{\hat{A}, S\}) = \sum_{j=1}^{N} E\left[d(X, Y_j) / X \in S_j\right] P(X \in S_j)$$
$$\geq \sum_{j=1}^{N} \min_{u} E[\, d(X, u) / X \in S] \, P(X \in S_j) \qquad (10.8)$$

En résumé, on constate que d'un côté, à partir d'un dictionnaire donné, on peut déterminer la partition optimale qui minimise l'erreur de distorsion. D'un autre côté et à partir d'une partition donnée, en occurrence la partition optimale définie ci-dessus, on peut déterminer le dictionnaire optimal qui minimise cette même erreur. Ceci traduit les deux procédures principales de l'algorithme LBG. Deux optimisations qu'on peut résumer par :

- partition optimale : Cette partition est synonyme de groupement des vecteurs de la séquence d'entraînement en ensembles de vecteurs. Elle est relative à un dictionnaire donné. Tous les vecteurs d'un même ensemble admettent pour vecteur de reproduction un même vecteur dans ce dictionnaire de façon que la relation (10.4) soit vérifiée.

- Dictionnaire optimal : relatif à la partition précédente. Déterminer le dictionnaire optimal revient à former un dictionnaire à partir des centres de gravité des vecteurs de chaque classe de cette partition.

10-3-1-2. Algorithme LBG pour une distribution connue

Ces deux étapes sont les procédures principales de l'algorithme LBG, qui à partir d'un dictionnaire donné, dictionnaire initial, engendre la meilleure partition au sens d'un certain critère. Aussi, à partir de cette partition on détermine le meilleur dictionnaire formé par les centroïdes des ensembles qui constituent cette partition. Ainsi à chaque itération ce processus fait que la distorsion quadratique moyenne doit être réduite ou laissée inchangée.

Le quantificateur optimal est celui qui minimise la distorsion pour une séquence aléatoire de vecteur $X = (x_1, x_2, \ldots, x_K)$ ou X est de probabilité Pr(X). L'algorithme LBG est donc le suivant :

étape ❶ On se donne à l'initialisation :

- un dictionnaire initial $\hat{A}_0$ de taille N et de dimension K.
- un ε, performance du dictionnaire.
- $m = 0$
- $D_{-1} = \infty$

étape ❷ pour $\hat{A}_m = \{Y_j; j = 1 \ldots N\}$; on cherche la meilleure partition possible $P(\hat{A}_m) = \{ P_j; j = 1 \ldots N\}$ obtenue en faisant correspondre chaque X au vecteur Y_i de $\hat{A}_m$ minimisant $d(X, Y_j)$. C'est la règle dite du plus proche voisin :

$$X \in P_j \Rightarrow d(X, Y_j) \leq d(X, Y_i) \qquad i = 1 \ldots N$$

étape ❸ La distorsion moyenne résultante D_m :

$$D_m = D(\{\hat{A}, P(\hat{A})\}) = E(\min_{Y_j \in \hat{A}} d(X, Y_j))$$

étape ❹ La performance est donnée par $(D_{m-1} - D_m)/D_m$ .Si cette performance est $\leq \varepsilon$ alors fin de la procédure et le quantificateur est décrit par le dictionnaire $\hat{A}_m$ et la partition $P(\hat{A}_m)$. Si ce n'est pas le cas, aller à l'étape étape ❺.

étape ❺ A partir de la partition $P(\hat{A}_m)$ on calcule les $\hat{X}(\hat{A}_m) = \{ \hat{X}(P_j); j = 1 \ldots N\}$ les centroïdes de l'ensemble $P(\hat{A}_m)$. Le nouveau dictionnaire devient $\hat{A}_{m+1} = \hat{x}\left[P(\hat{A}_m)\right]$. Poser m = m+1 et aller à l'étape ❷.

Au bout d'un certain nombre d'itérations, le rapport $(D_{m-1} - D_m)/D_m$ qui relate la performance du système, diminue. On arrête l'itération si cette dernière est nulle ou inférieure à une certaine valeur ε. Deux cas peuvent se présenter. le système s'arrête au bout d'un certain nombre d'itération, donc on peut dire que le point de convergence a été atteint. Sinon, la distorsion D_m tend vers une valeur limite D_∞ pour un dictionnaire limite $\hat{A}_\infty$, le couple $\{\hat{A}_\infty, P(\hat{A}_\infty)\}$ est appelé le point de convergence ou le point fixe du quantificateur et $\hat{A}_\infty$ est le dictionnaire optimal pour la distribution Pr(X).

10-3-1-3. Algorithme LBG pour une distribution non connue

Si maintenant on dispose d'une source à quantifier, mais dont les propriétés statistiques ne sont pas définies. Il faut avoir dans ce cas une séquence T d'entraînement représentative de la source à coder, $T = \{ X_k; k=0, \ldots, L\}$, L le nombre de vecteurs de la séquence. Dans ce cas la distorsion moyenne du quantificateur peut s'écrire :

$$D(Q) = E[\, d(X, Q(X))\,] = \frac{1}{L}\sum_{j=1}^{L} d(X_j, Q(X_j)) \qquad (10.9)$$

Le quantificateur optimal est celui qui minimise cette distorsion.

La mesure de distance erreur quadratique est donnée par :

$$d(X, Y) = \sum_i |x_i - y_i|^2$$

L'algorithme qui permet de former un dictionnaire relatif à de telles sources et à base du LBG devient :

étape ❶ $\hat{A}_0$ est un dictionnaire initial formé de N vecteurs de dimension K. Soit T = { X_k,; k=0, ..., L} la séquence d'apprentissage L est le nombre de vecteurs.

on se donne un ε, performance du dictionnaire.

On pose $m = 0$

$D_{-1} = \infty$

étape ❷ pour un dictionnaire $\hat{A}_m = \{Y_j; j = 1 \ldots N\}$; on détermine $P(\hat{A}_m) = \{P_j; j = 1 \ldots N\}$, obtenue en appliquant la règle dite du plus proche voisin, qui au vecteur X fait correspondre un vecteur Y_i de $\hat{A}_m$ minimisant $d(X, Y_j)$:

$$X \in P_j \Rightarrow d(X, Y_j) \leq d(X, Y_i) i = 1 \ldots N$$
$$d(X, Y) = \Sigma i \, |x_i - y_i|^2$$

étape ❸ La distorsion moyenne résultante D_m est donnée par :

$$D_m = D(\{\hat{A}, P(\hat{A})\}) = \frac{1}{L}\sum_{j=1}^{L} \min_{y \in \hat{A}_m}(d(x_j, y))$$

étape ❹ mesure de la performance donnée par $(D_{m-1} - D_m)/D_m$. Si cette performance est ≤ ε alors fin de la procédure. Le quantificateur est décrit par le dictionnaire $\hat{A}_m$ et la partition $P(\hat{A}_m)$. Si ce n'est pas le cas aller à l'étape suivante.

étape ❺ A partir de la partition $P(\hat{A}_m)$, on détermine les $\hat{X}(\hat{A}_m) = \{ \hat{X}(P_j); j = 1 \ldots N\}$, les centroïdes de l'ensemble $P(\hat{A}_m)$ donnés par :

$$\hat{x}(P_j) = \frac{1}{\|P_j\|} \sum_{k/x_k \in P_j} x_k$$

$\|P_j\|$ représente le nombre de vecteurs de T qui appartiennent à la partition P_j.

Le nouveau dictionnaire devient $\hat{A}_{m+1} = \hat{x}\left[P(\hat{A}_m)\right]$.

Poser m = m+1 et aller à l'étape ❷.

Pour une performance et un dictionnaire initial donnés, l'algorithme précédent converge, au bout de m itérations, vers un couple $\{\hat{A}_m, P(\hat{A}_m)\}$ qui représente le quantificateur de la séquence T. Si T est suffisamment grande pour être bien représentative de la source à coder, on peut considérer que le quantificateur Q ainsi établi est sous optimal. Si L tend vers l'infini le quantificateur Q tend vers le quantificateur optimal.

L'algorithme LBG a besoin d'un dictionnaire initial pour amorcer son processus. Il dépend étroitement de ce choix. Un choix judicieux de ce dictionnaire s'impose. Plusieurs méthodes ont été développées. Dans le cadre de ces travaux, nous avons retenu la méthode dite "splitting" décrite dans le paragraphe suivant.

10-3-1-4. Choix du dictionnaire initial

Une façon des plus simples pour choisir un dictionnaire initial est de prendre les N premiers vecteurs de la séquence d'entraînement. Mais ce choix n'est pas très satisfaisant. Aussi on peut les prendre d'une façon aléatoire parmi les L vecteurs de la séquence d'entraînement, mais là aussi les vecteurs peuvent ne pas être bien répartis dans l'espace à K dimensions.

Pour remédier à ces inconvénients, un choix convenable est la méthode qu'on appelle "splitting".

C'est une technique qui utilise l'algorithme décrit précédemment. En partant d'un dictionnaire vide au départ, le premier élément qui va constituer ce dictionnaire est alors le centroïde de toute séquence d'entraînement.

A l'itération suivante, l'élément unique du dictionnaire est remplacé par deux vecteurs issus de ce même vecteur en introduisant un vecteur de perturbation $\vec{\varepsilon}$. C'est ce qu'on appelle "splitting" et qu'on peut représenter de la façon suivante:

$$\vec{Y} \longrightarrow (\vec{Y} + \vec{\varepsilon}, \vec{Y} - \vec{\varepsilon}) \qquad (10.11)$$

On lance l'algorithme LBG avec ce dernier dictionnaire comme dictionnaire initial, et on produit alors un dictionnaire optimal à deux éléments.

Si on réitère ces opérations, figure (10.2), la taille du dictionnaire se voit doubler à chaque stade. Après M itérations, le dictionnaire est formé de 2^M vecteurs. Ce processus est arrêté quand le dictionnaire atteint la taille désirée en puissance de deux.

L'algorithme de "splitting" est le suivant :

étape ❶ soit $\hat{X}(T)$ le centroïde de toute la séquence d'entraînement. $\hat{A}_0(1)$ le dictionnaire initial formé d'un seul vecteur de sorte $\hat{A}_0(1) = \{\hat{X}(T)\}$.Soit $\vec{\varepsilon}$ est le vecteur de perturbation.$(Y_i + \vec{\varepsilon})$ et $(Y_i - \vec{\varepsilon})$ et N la taille du dictionnaire final.

étape ❷ A partir du dictionnaire $\hat{A}_0(M) = \{Y_i; i = 1, \ldots, M\}$ chaque vecteur P_i du dictionnaire est décomposé en deux vecteurs $(Y_i + \vec{\varepsilon})$ et $(Y_i - \vec{\varepsilon})$. Soit $\hat{A}_0^*(M)$ le dictionnaire formé de :

$$\hat{A}_0^*(M) = \{Y_i + \vec{\varepsilon}, Y_i - \vec{\varepsilon}; i = 1, \ldots, M\}.$$

étape ❸ $\hat{A}_0(2M) = \hat{A}_0^*(M)$;

Lancer l'algorithme LBG avec dictionnaire initial $\hat{A}_0(2M)$.

M = 2M. Si M = N, arrêter le processus. Sinon, aller à l'étape ❸.

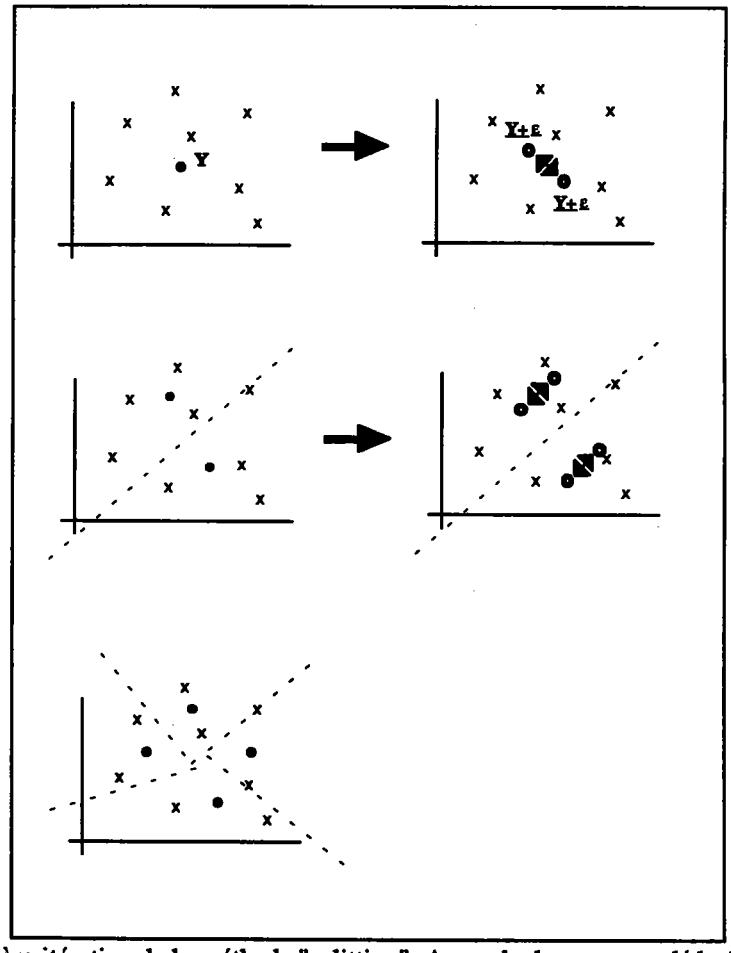


Figure 2 : Première itération de la méthode "splitting". A gauche les vecteurs dédoublés, à droite, aprés passage de l'algorithme LBG utilisant ces derniers vecteurs comme dictionnaire initial.

10-3-2. Réseaux de neurones

10-3-2-1. Introduction

Les réseaux de neurones peuvent être définis comme un ensemble d'éléments non linéaires opérant en parallèle et interconnectés par des poids qui s'adaptent dans le temps et en fonction des entrées présentées.

Ce type de processus est typiquement réalisé par le cerveau. Les éléments présentés sont captés à partir d'entrées sensorielles, tactiles, visuelles ou acoustiques. En général, le système nerveux, au cours des premières étapes de son développement, s'organise. Une fois cette organisation établie, elle conserve suivant les cas plus ou moins de plasticité.

Le but alors des réseaux de neurones artificiels est de pouvoir reproduire certaines fonctions typiques et élémentaires du cerveau.

Les réseaux de neurones fonctionnent en général en deux phases, phase d'apprentissage et phase d'usage :

- la phase d'apprentissage est la phase d'organisation du réseau.
- la phase d'usage est en rapport avec l'application qu'on désire réaliser avec le réseau. On peut distinguer les utilisations suivantes :
  - ⇨ identification de la classe d'un échantillon donné
  - ⇨ mémoire associative
  - ⇨ ou quantificateur vectoriel

Pour résumer on peut dire qu'un réseau effectue une classification des échantillons qu'on lui présente. Le réseau doit s'organiser de sorte que si on présente un échantillon en entrée, à priori inconnu, le réseau réagit en activant plus ou moins les sorties en fonction de son organisation interne et de l'élément présenté. En d'autres termes, le système tend à attribuer une réaction particulière à une excitation inconnue qui survient à son entrée (on peut qualifier un réseau de neurones de système d'identification).

Plusieurs solutions ont été proposées dans ce domaine, voir [Rum87][Lip87]. Sans faire une étude exhaustive de tous les types de réseaux existant, nous nous limitons à une catégorie de réseaux appelé réseaux de Kohonen. Ce réseau semble être le plus approprié à la réalisation d'une opération de quantification vectorielle [Nas88].

A l'origine, Kohonen a mis au point un algorithme qui permet de former, de façon

autonome, une cartographie des propriétés des données "*self organization feature maps*". Le réseau n'est pas destiné à modéliser fidèlement le processus à l'oeuvre dans le cerveau, mais tente d'en reproduire les traits essentiels tout en restant raisonnablement simple du point de vue des calculs.

En effet, il réalise une projection de l'espace d'entrée sur un ensemble fini de vecteurs de sortie, suivant une procédure proche de celle développée précédemment par l'algorithme LBG. Cette méthode conduit alors à un quantificateur vectoriel qualifié comme un quantificateur localement optimal.

Une des différences majeure entre les deux techniques est le problème du dictionnaire initial qui reste un problème délicat pour la méthode LBG, alors que les valeurs de convergences des vecteurs du réseau de Kohonen ne dépendent pas des valeurs initiales [Koh82a][Koh82b][Koh88].

10-3-2-2. Principe de la formation du réseau de Kohonen

Le réseau de Kohonen est un réseau formé de NxK coefficients ou liens. Le réseau active un seul parmi les N noeuds pour répondre à une entrée présentée sous forme d'un vecteur à K éléments.

Chaque point i en entrée (i = 1 ... K) est relié aux noeuds j du réseau (j = 1, ..., N). A ces liens sont attribués des coefficients ou des poids notés μ_{ij}, figure (10.3).

On peut en déduire que l'espace d'entrée est de dimension K et l'espace de sortie est un espace fini à N vecteurs de même dimension. Un vecteur d'entrée est donné par $X= (x_i; i = 1, ..., K)$ et chaque sortie du réseau est un vecteur Y_j défini par tous les poids des liens qui le relie aux entrées. Il est de la forme suivante:

$$Y_j = (\mu_{ij}; i = 1, ..., K); \qquad (10.12)$$

Les N sorties sont disposées d'une façon ordonnée de manière à définir un voisinage physique ou topologique de chaque noeud. Les noeuds de sortie sont disposés en général en ligne et/ou en colonne, figure (10.3).

Pour représenter d'une manière optimale une séquence d'apprentissage présentée à l'entrée du réseau, à la différence de LBG, le réseau de Kohonen ne fait pas de partition. Chaque fois qu'un vecteur est présenté à l'entrée (à noter le côté séquentiel des entrées) le réseau s'organise. Pour cette raison, la séquence d'apprentissage présentée au réseau doit être suffisamment longue.

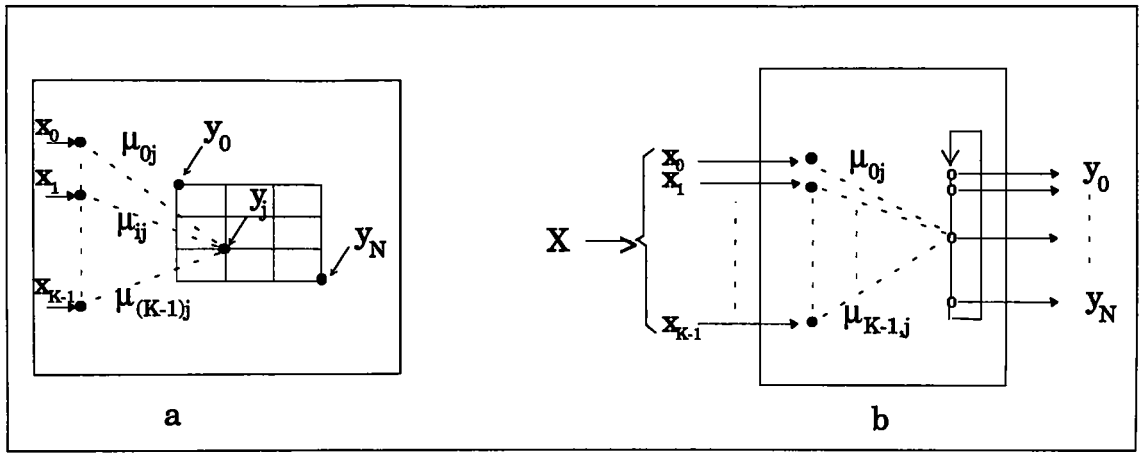


Figure 10.3 : Connections dans un réseau de Kohonen. t a) Reseau à deux dimension, b) réseau à une dimenssion.

A la fin de l'apprentissage, les vecteurs donnent, au sens d'un certain critère qu'il faut définir, une représentation des différentes classes qui forment la séquence.

Contrairement à la méthode LBG ou la fin de l'algorithme est liée à la convergence de la performance définie par la distorsion totale, la fin de l'organisation du réseau de Kohonen est relative au passage de tous les éléments de la séquence d'apprentissage dans le réseau, sans aucune mesure de distorsion.

10-3-2-3. Algorithme

L'algorithme d'évolution du système est le suivant :

étape ❶ Initialiser, à des petites valeurs, les poids de connexion du réseau μ_{ij}; (i = 1, ..., K; j = 1, ..., K).
le dictionnaire est formé par les vecteurs $Y_j = (\mu_{ij}; i = 1,..., K)$.
soit T = {x(n); n= 1, ..., N} l'ensemble des vecteurs d'apprentissage et n l'indice du vecteur.
n= 0.

étape ❷ Présenter le vecteur x(n) à l'entrée.

étape ❸ Déterminer le noeud j* tel que Y_{j*} soit le plus proche voisin, $j \in [1,..., N]$, au sens d'un certain critère (par exemple ici l'erreur quadratique moyenne) :
$d(x_n, Y_{j*}(n)) \leq \min_{Y_j \in A(n)} (d(x(n), Y_{j*}(n))\ j \in [1, ..., N]$.

étape ❹ déterminer V(j*, n) et α_n.
⇨V(j*, n) définit le voisinage topologique du noeud j* à l'itération n.
⇨α_n est une fonction décroissante (fonction gain).

étape ❺ le noeud sélectionné et son voisinage sont modifiés par :

$$\mu_{ij}(n+1) = \mu_{ij}(n) + \alpha_n V(j^*, n)[x_i(n) - \mu_{ij}(n)] \quad (10.13)$$

$i = 1, ..., N.$

$j \in V(j^*, n)$

étape ❻ poser n = n+1;

si n< L taille de la séquence d'apprentissage; aller à étape ❷

sinon fin de l'algorithme. Le dictionnaire est formé par les noeud $Y_j(n)$ j = 1,..., N avec $Y_j = (\mu_{ij}; i = 1,..., K)$.

10-3-2-4. Paramètres du réseau

Le voisinage topologique, noté V(j*, n), contrôle l'influence du vecteur incident sur le voisinage du noeud j* sélectionné. Des simulations numériques ont montré que les meilleurs résultats en auto-organisation sont obtenus avec des fonctions de voisinage variables de sorte que ce voisinage soit assez large au début et se rétrécisse au cours du temps [Koh88].

D'autre part la fonction de gain $\{\alpha_n; n = 0, ..., L\}$ qui vérifie $0< \alpha_n < 1$; est obligatoirement une fonction décroissante dans le temps. Il est possible de supposer que α_n doit satisfaire une condition similaire aux cas des approximations stochastiques [Koh88] à savoir :

$$\Sigma\alpha_n = \infty ; \qquad \Sigma\alpha_n^2 < \infty$$

Figure 10.4 : Exemple d'évolution des fonctions V(j*, n) de voisinage et α_n de gain dans le temps.

Il existe plusieurs approches pour définir la fonction voisinage V(j*, n) et la fonction gain α_n. On en cite les deux suivantes :

• Approche de Ritter et Schulter :

$$\mu_{ij}(n+1) = \mu_{ij}(n) + \alpha_n . h_n^{j,j^*}[x_i(n) - \mu_{ij}(n)] \quad i \in [1,N] \text{ et } j \in [1,N]. \quad (10.14)$$

avec $\alpha(n) = \alpha_i \left(\frac{\alpha_f}{\alpha_i}\right)^{\frac{n}{L}}$

$$h^{j,j^*}(n) = \exp - \frac{\|j - j^*\|}{2\sigma_n^2}$$

avec $$\sigma(n) = \sigma_i \left(\frac{\sigma_f}{\sigma_i} \right)^{\frac{n}{L}}$$

α_n est la fonction gain, h_n^{j,j^*} est une fonction variable dans le temps et définissant un voisinage topologique du noeud actif.

• Approche de Nasrabadi :

$$\mu_{ij}(n+1) = \mu_{ij}(n) + \alpha_n NE_{j^*}(n)[x_i(n) - \mu_{ij}(n)] \quad i \in [1,N] \text{ et } j \in [1,N]. \qquad (10.15)$$

où α_n est la fonction gain, elle est définie par :

$$\alpha_n = A\, e^{(-n/T_1)}$$

Où A est une constante déterminant le gain maximal d'influence, et T_1 une constante déterminant le taux de décroissance de α_n après chaque présentation d'un vecteur en entrée.

Le voisinage topologique $NE_{j^*}(n)$ contrôle l'influence sur le voisinage de j* du vecteur incident. Il est décroissant dans le temps et définie par :

$$NE_{j^*}(n) = A_1 + A_2\, e^{(-n/T_2)}$$

où A_1 et A_2 sont des constantes déterminant les limites du voisinage et T_2 une constante déterminant le taux de rétrécissement de ce voisinage autour du noeud actif.

X-4. Conclusion

La quantification vectorielle a fait l'objet de ce chapitre. Cependant, elle ne peut être traitée sans tenir compte du problème de la formation de dictionnaires. Deux techniques de formation de dictionnaire, une à base de l'algorithme LBG et l'autre à base de réseaux neurones, en l'occurrence le réseau de Kohonen, ont été présentées.

Nous avons présenté l'état de l'art en quantification vectorielle et en technique de constitution de dictionnaires. Ces méthodes ayant déjà fait l'objet de nombreuses recherches, nous les avons appliquées dans leurs versions classiques. Par contre, nous allons aborder le problème de la recherche d'un vecteur dans un dictionnaire. Nous soulevons, dans le chapitre suivant, la problématique de cette recherche en proposant des solutions originales pour y remédier.

XI. Recherche Optimisée dans un Dictionnaire

XI-1. Introduction

Une procédure majeure mais répétitive dans le cadre de la quantification vectorielle est la recherche du plus proche vecteur dans le dictionnaire. C'est à dire, la recherche du vecteur de reproduction qui va représenter le vecteur en entrée. Cette recherche s'effectue dans la phase de formation du dictionnaire et dans la phase de codage d'une séquence à l'entrée du quantificateur.

Cette procédure est très consommatrice en temps et il faut la rendre la moins onéreuse possible. On propose dans ce chapitre deux techniques de scrutation dans le dictionnaire permettant de réduire et d'optimiser le temps de recherche.

XI-2. Recherche optimisée

11-2-1. Description du problème

La procédure classique consiste, pour chaque vecteur en entrée, à déterminer le vecteur le plus proche dans le dictionnaire, au sens d'un critère donné. En d'autres termes il faut comparer le vecteur en entrée avec tous les vecteurs du dictionnaire pour fournir l'indice du vecteur vérifiant cette contrainte (le critère de l'erreur quadratique par exemple). Le vecteur sélectionné est le vecteur de reproduction.

Cette façon de procéder est la plus directe pour réaliser l'opération de recherche. Elle est qualifiée de recherche totale ou de "full search" en anglais. Dans ce cas tous les vecteurs du dictionnaire sont testés. Le nombre de tests et par conséquent, le temps augmente linéairement avec la taille des vecteurs et celle du dictionnaire.

Ainsi, pour une séquence donnée, comprenant S vecteurs de dimension K et pour un dictionnaire constitué de N vecteurs, de même dimension que les vecteurs en entrée, on effectue, pour chaque vecteur de la séquence, N comparaisons pour obtenir l'indice du vecteur de reproduction et SxN comparaisons pour coder toute la séquence.

Pour donner un ordre de grandeur, nous allons évaluer le nombre d'opérations nécessaires pour effectuer la recherche. En utilisant la distance erreur quadratique définie par :

$$d(X, Y_j) = \sum_{i=1}^{K} (x_i - y_{ij})^2$$

On déduit le coût des opérations suivantes :

calcul d'une distance : $c = (K)_{mult} + (K-1)_{.add}$

comparaison avec tous les vecteurs du dictionnaire: $v = N.c$

coût de toute la séquence : $s = S.v = S.N.c$

Pour une image de taille 128x128, si les vecteurs sont formés par des blocs de taille 4x4, soit une dimension de K=16, la séquence à coder est formée de 1024 vecteurs. Si le dictionnaire a une taille de 256 vecteurs par exemple, alors on a 262·144 comparaisons de vecteurs qui vont coûter à peu prés de 4 millions de multiplications et autant d'additions, et ceci pour une seule image.

La formation d'un dictionnaire nécessite généralement des séquences d'entraînement constituées à partir de plusieurs images. Si on suppose alors qu'on utilise 5 images pour former un dictionnaire, on aura à effectuer 20 millions de multiplications et 20 millions d'additions à chaque itération de l'algorithme.

L'objectif de cette section est de proposer des méthodes d'optimisation de cette recherche. Ces méthodes consistent à limiter la recherche de ce vecteur dans des portions du dictionnaire où le vecteur de reproduction peut résider.

Ceci ne peut être fait que si les vecteurs du dictionnaire sont ordonnés. Or il n'existe aucune relation d'ordonnancement entre les vecteurs générés par la méthode LBG (Le cas de l'algorithme de Kohonen est différent puisqu'on a vu dans le chapitre précédent qu'il existe une notion de voisinage topologique).

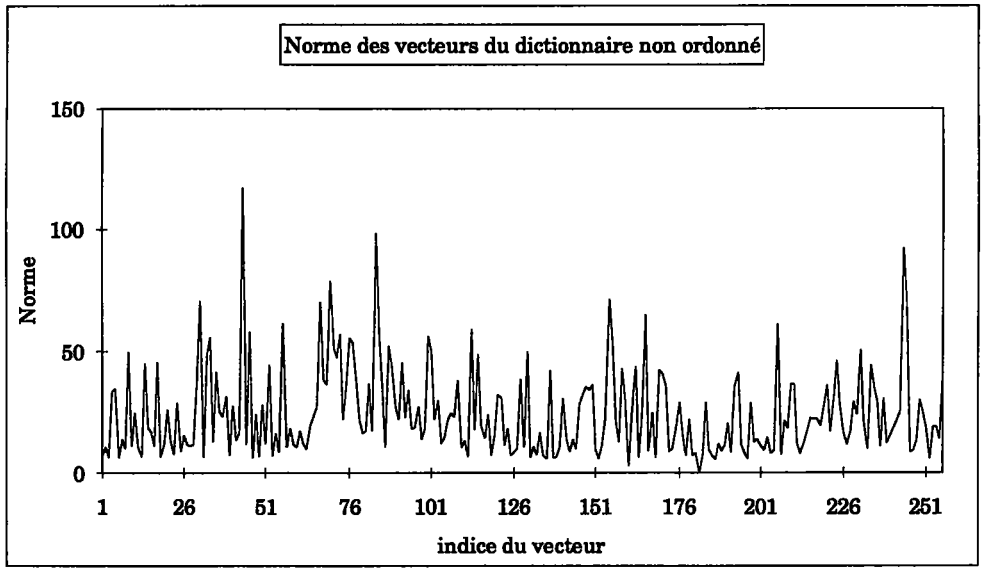
11-2-2. Dictionnaire ordonné

Pour ordonner un dictionnaire il faut définir une grandeur qui va caractériser ses vecteurs. Le moyen le plus simple pour attribuer une grandeur à un vecteur est de mesurer sa

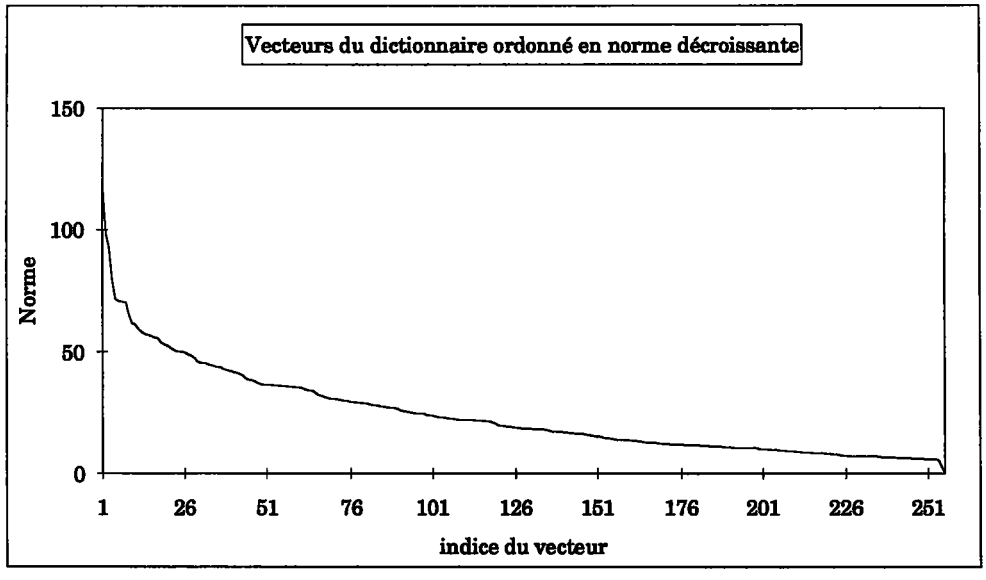
norme. A chaque vecteur q_i du dictionnaire est associée sa norme $\|q_i\|$. La norme euclidienne est donnée par :

$$\|q_i\| = \left(\sum_{i=1}^{K} (q_{ij})^2 \right)^{1/2}$$

Ordonner les vecteurs d'un dictionnaire c'est les ranger suivant leur norme, du plus petit au plus grand ou inversement, figure (11.1).



a



b

Figure 11.1 : Arrangement d'un dictionnaire par norme: a) vecteur avant arrangement, b) vecteurs après arrangement.

11-2-3. Intervalles de reproduction

Maintenant que le dictionnaire est ordonné par norme, les questions suivantes se posent :

-comment définir une correspondance entre un vecteur en entrée et le vecteur de reproduction dans le dictionnaire?

-et puisqu'on a introduit la notion de norme, qu'elle relation peut lier la norme du vecteur en entrée et la norme de son vecteur de reproduction ?

Afin de situer le problème, nous allons effectuer des mesures de différence entre les normes des vecteurs incidents et leurs vecteurs de reproduction. Ceci permet de savoir comment se comporte cette différence vis-à-vis des normes des vecteurs en entrée. Sachant que ce sont les seules grandeurs dont on dispose. La figure (11.2) illustre un exemple de ces différences de normes en fonction des normes des vecteurs en entrée.

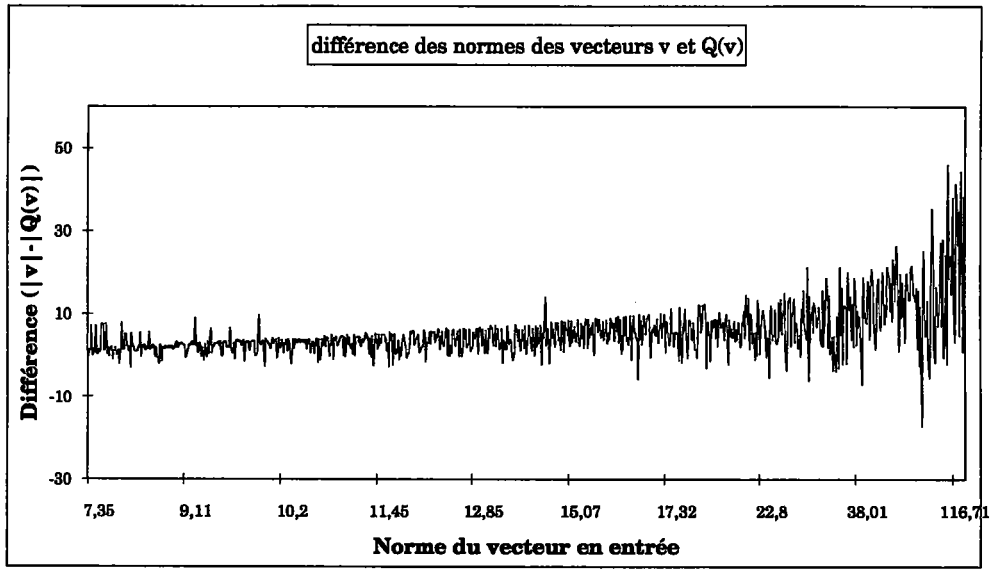


figure 11.2 : Exemple de différence entre la norme du vecteur en entrée et celle de son vecteur de reproduction, en fonction de la norme des vecteurs en entrée.

La première constatation est que ces différences admettent une enveloppe croissante avec la norme des vecteurs en entrée. Ce qui nous permet de déduire que cette différence est plus ou moins liée à la norme du vecteur en entrée. C'est une relation d'ordonnancement. En d'autres termes cette différence n'excède pas un certain niveau si la norme du vecteur en entrée (ou la norme du vecteur de reproduction) n'excède pas elle non plus une certaine valeur.

On peut en conclure que pour une norme du vecteur en entrée, la différence avec la norme du vecteur de reproduction peut être estimée statistiquement. Donc on peut définir un intervalle de normes où le vecteur de reproduction a une grande probabilité d'exister.

Grâce à ces constatations, dans la suite de ce chapitre, deux techniques de recherche

optimisées vont être développées. Dans l'une de ces méthodes, les intervalles sont fixes et dans l'autre, ils varient dynamiquement. Ces intervalles, appelés intervalles de reproduction, permettent de limiter des portions du dictionnaire où le vecteur de reproduction sera recherché.

XI-3. Intervalle fixe des normes

C'est une technique qui découle des constatations faites ci-dessus sur la distribution des valeurs de différence des normes des vecteurs en entrée et celles des vecteurs de reproduction. Nous avons pu constater que les intervalles de reproduction sont liés aux normes des vecteurs en entrée.

Le principe est alors le suivant :

A tout vecteur v en entrée de norme $\|v\|$, on définit un intervalle de recherche du vecteur de reproduction sur l'axe des normes des vecteurs du dictionnaire. Cet intervalle est proportionnel à $\|v\|$ et est aussi centré sur cette même valeur, voir figure (11.3).

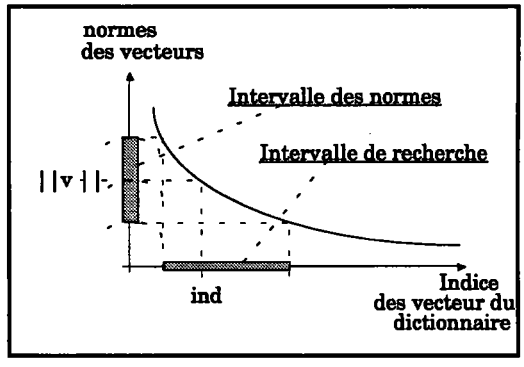


Figure 11.3 : Correspondance entre l'intervalle des normes et l'intervalle de recherche

Le "rayon" de cet intervalle sera alors proportionnel à la norme du vecteur en entrée. Si on appelle R ce "rayon", on peut l'écrire sous la forme suivante :

$$R = \alpha \|v\| \qquad \text{pour un } \alpha \text{ vérifiant : } 0 \leq \alpha \leq 1.$$

Seuls les vecteurs q du dictionnaire dont la norme appartient à l'intervalle de recherche $[\|v\|-R, \|v\|+R[$ sont pris en considération dans la détermination du vecteur de reproduction.

Pour différentes valeurs de α on peut dresser la courbe des taux d'aboutissements. Ces taux correspondent à la détermination des mêmes vecteurs de reproduction que ceux trouvés si la méthode "*full search*" est utilisée. La figure (10.4) donne un exemple de mesure du compromis entre la valeur de α, facteur de proportionnalité, et le taux d'aboutissement de l'opération.

On peut noter que pour des valeurs de $\alpha > 0.6$, on peut considérer que tous les vecteurs de reproduction sont atteints pour des taux de tests d'environ 38% à 48% de comparaisons, équivalent à un gain de 62% à 52%, une performance notable puisque plus de la moitié des comparaisons est évitée. Pour $\alpha = 0.5$, il n y a que 30% de comparaisons pour seulement 10% de vecteurs non atteints, mais qui peuvent être éventuellement approchés par d'autres vecteurs du dictionnaire.

Enfin et pour conclure, le gain est souvent de plus de 50%, dans les pires des cas, on affecte à α la valeur 1. Dans l'exemple de la figure 4, 100% de vecteurs sont atteints pour un gain de comparaisons de 52%.

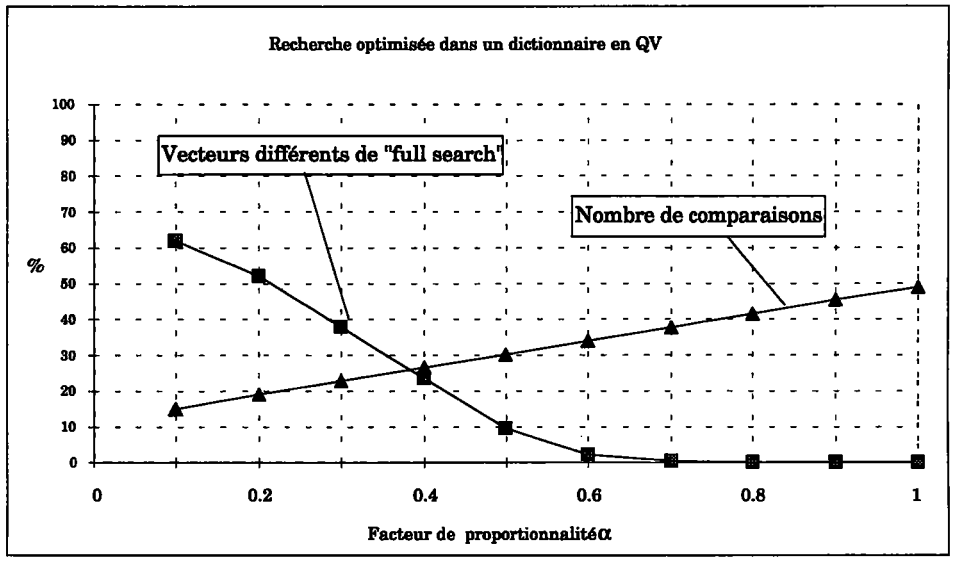


Figure 11.4 : Exemple de taux d'aboutissement pour différentes valeurs de α.

XI-4. Intervalle dynamique de recherche

11-4-1. Introduction

Dans le paragraphe précédent l'intervalle des normes des vecteurs candidats aux tests de comparaisons était fixé au départ à une valeur proportionnelle à la norme du vecteur en entrée. Cette valeur sera maintenue jusqu'à ce qu'on atteigne le vecteur de reproduction. Cette technique découle des constatations sur les statistiques des différences des normes entre les vecteurs en entrée et les vecteurs de reproduction du dictionnaire.

Pour que cette procédure soit bien optimisée, c'est à dire, pour réduire encore le nombre de tests, il faut l'adapter à la statistique de la séquence à coder mais qu'on ignore le plus souvent.

Dans ce paragraphe, le rayon de l'intervalle des normes des vecteurs du dictionnaire candidats aux tests, pour déterminer le vecteur de reproduction, est cette fois dynamique. Il est initialisé au départ mais susceptible d'être modifié au cours des tests.

11-4-2. Principe

Comme au paragraphe précédent, le vecteur en entrée est placé, en fonction de sa norme, sur l'intervalle des normes des vecteurs du dictionnaire préalablement arrangé.

Le rayon du premier intervalle de recherche est déterminé par la distance entre le vecteur en entrée et un vecteur du dictionnaire dont la norme est la plus proche. Ce rayon permet déjà de limiter le nombre de vecteurs à tester. Puisque tout vecteur dont la norme est en dehors de cet intervalle aura automatiquement, par rapport au vecteur d'entrée, une distance supérieure à ce rayon qui est une distance définie par un autre vecteur du dictionnaire. Ceci est illustré par le graphe de la figure (11.5).

Au sens de la distance euclidienne, on peut considérer que le vecteur qui détermine le rayon est le plus proche du vecteur en entrée par rapport aux vecteurs en dehors de cet intervalle. De cette façon on élimine d'office un certain nombre de vecteurs.

Dans un premier temps, seuls les vecteurs du dictionnaire, dont la norme appartient à cet intervalle, sont éventuellement sujets aux tests de comparaisons. On verra par la suite comment cet intervalle peut diminuer. Cela permet en même temps de réduire encore le nombre de vecteur dans l'intervalle et par conséquent le nombre de comparaison.

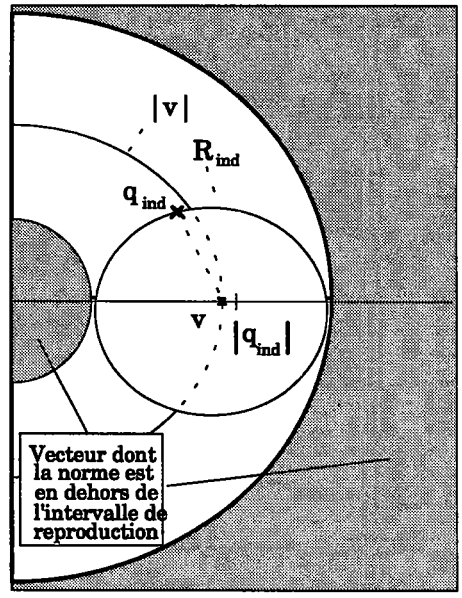


Figure 11.5 : Intervalle initial de recherche défini autour du vecteur v, par la norme $|v-q_{ind}|$ où q_{ind} est le vecteur du dictionnaire dont la norme est la plus proche de celle de v.

11-4-3. Intervalle dynamique de recherche

Pour tout vecteur $\vec{q}_i$ du dictionnaire on peut définir un domaine qui a pour rayon R_i, la norme du vecteur différence entre celui-ci et le vecteur en entrée, et centré sur ce dernier. La valeur du rayon est donnée par :

$$R_i = \|\vec{v} - \vec{q}_i\|$$

Si on appelle $\vec{q}_{ind}$ le vecteur de dictionnaire dont la norme est la plus proche de la norme du vecteur en entrée $\vec{v}$, le rayon de l'intervalle de recherche initial a pour valeur R_{ind} donnée par :

$$R_{ind} = \|\vec{v} - \vec{q}_{ind}\|$$

Sur l'axe des normes des vecteurs du dictionnaire, on définit l'intervalle de recherche de départ I_{ind}. Cet intervalle est déterminé par le rayon R_{ind} défini ci-dessus et centré sur la valeur de la norme $\|\vec{v}\|$ du vecteur en entrée. L'intervalle est défini par :

$$I_{ind} = [\|\vec{v}\| - R_{ind}, \|\vec{v}\| + R_{ind}]$$

Seuls les vecteurs $\vec{q}_i$ du dictionnaire dont la norme appartient à l'intervalle I_{ind} sont testés. Par conséquent, tout vecteur ne vérifiant pas cette condition ne sera pas testé, puisque $R_i > R_{ind}$ comme illustré par la figure (11.5) et vérifié par la relation :

$$\text{Si } \|q_i\| \geq \|v\| + R_{ind} \text{ ou si } \|q_i\| \leq \|v\| - R_{ind} \text{ alors } R_i \geq R_{ind}$$

Cela veut dire que ces vecteurs ne sont pas plus près, au sens de la distance euclidienne, du vecteur d'entrée que le vecteur Q_{ind}.

Si par contre un vecteur du dictionnaire, dont la norme appartient effectivement à cet intervalle de recherche, est testé alors deux cas peuvent se produire :

- ou bien R_i est supérieur à R_{ind}
- ou bien R_i est inférieur à R_{ind}

Dans le premier cas, le vecteur n'est pas plus proche du vecteur d'entrée que le vecteur $\vec{q}_{ind}$, donc ce n'est pas le vecteur de reproduction. Dans le second cas, le vecteur $\vec{q}_i$ est plus proche du vecteur en entrée que le vecteur $\vec{q}_{ind}$. Par conséquent, la nouvelle valeur du rayon R_{ind} devient R_i, ainsi l'intervalle de recherche est modifié par cette nouvelle affectation et le nouveau vecteur $\vec{q}_{ind}$ devient $\vec{q}_i$ et ind prend i comme nouvelle valeur, figure (11.6).

De cette façon, l'intervalle de recherche diminue au fur et mesure qu'un vecteur testé,

appartenant à l'intervalle de recherche courant, fournit une valeur de rayon inférieure à celle du rayon de l'intervalle courant.

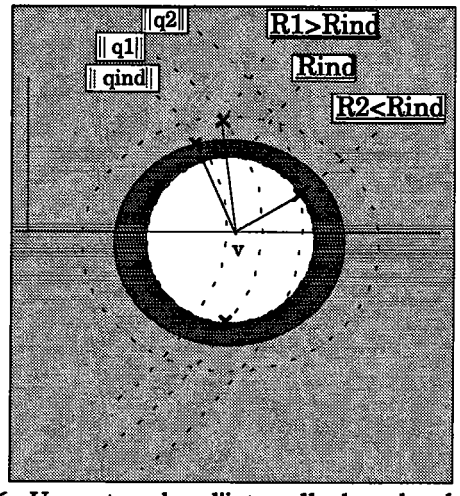


Figure 11.6 : Un vecteur dans l'intervalle de recherche et dont le rayon est inférieur au rayon courant permet de réduire l'intervalle de recherche.

La procédure décrite ci-dessus est réitérée jusqu'à ce que tous les vecteurs du dictionnaire appartenant au dernier intervalle de recherche soient testés.

11-4-4. Algorithme

L'algorithme d'implémentation de la procédure de recherche du vecteur du dictionnaire le plus proche au sens du critère des moindres carrés par la méthode de l'intervalle dynamique est le suivant :

étape ❶ ordonner les vecteurs du dictionnaire par norme décroissante (ou croissante),

étape ❷ déterminer la norme $\|V\|$ du vecteur V en entrée, déterminer l'indice *ind* du vecteur dans le dictionnaire arrangé ayant la norme la plus proche de celle de V,

étape ❸ calculer la valeur du rayon de l'intervalle de recherche :

$$R_{ind} = \|\vec{v} - \vec{q}_{ind}\|$$

l'intervalle des norme de recherche est :

$$I = \left[\|\vec{v}\| - R_{ind}, \|\vec{v}\| + R_{ind}\right],$$

étape ❹ s'il y a un autre vecteur q_i tel que $|q_i| \in I$, si oui, alors aller à l'étape 4. Sinon Stop, le vecteur de reproduction est le vecteur q_{ind},

étape ❹ calculer la valeur de R_i :

$$R_i = \|\vec{v} - \vec{q}_i\|$$

si $R_i < R_{ind}$ alors *ind* = *i* et aller à ❸.
Sinon, aller à ❹.

Un exemple d'application de cet algorithme pour une quantification vectorielle avec recherche optimisée est réalisé sur deux séquences y1 et y2, formées chacune de 1024 vecteurs de dimension respectivement 16 et 4. Les dictionnaires utilisés, pour quantifier ces séquences sont de taille 256. Avec cet algorithme, tous les vecteurs de reproduction atteints sont les mêmes qu'en "full search". On compare seulement le gain en comparaisons entre les deux méthodes. Le pourcentage de comparaison pour 100% de vecteurs atteints est donné par le tableau (11.1). Ou on peut noter que les gains de comparaisons vont de 48% à même 80%.

Tableau 11.1 : Exemple de résultat de l'algorithme des intervalles dynamiques.

séquence	Taux de recherche	Gain
y1	42,5 %	57,5%
y2	19.3 %	80,7%

XI-5. Conclusion

Deux techniques pour réduire la charge de recherche d'un vecteur de reproduction dans un dictionnaire ont été définies dans ce chapitre. La première, à intervalles fixes, est basée sur la distribution des différences des normes entre les vecteurs en entrée et leurs vecteurs de reproduction. La seconde, à intervalle dynamique, est basée sur l'adaptation de l'intervalle de recherche au cours des tests sur les vecteurs de cet intervalle. Toutes les applications par ces deux techniques de recherches optimisées ont donné des gains supérieurs à 50%.

Après l'exposé des techniques de quantification vectorielle, de formation de dictionnaire et de recherche optimisée, il faut voir comment intégrer ceci dans un schéma de codage des coefficients de la pyramide d'ondelettes, qui sera le propos du chapitre suivant.

XII. Réorganisation de la Pyramide et Codage Hybride des Coefficients d'Ondelettes

XII-1. Introduction

Pour avoir des taux de compression importants pour coder une pyramide d'ondelettes, sans pour autant dégrader la qualité, des techniques de codage autres que la quantification scalaire ont été adaptées.

A l'introduction de cette troisième partie, les avantages de la QV sur la QS ont été abordés. Pour utiliser la QV, il faut arranger les coefficients en vecteurs. En d'autres termes, il faut organiser la pyramide pour avoir un usage adéquat d'autres techniques de codage.

Cette organisation de la pyramide découle de la création d'un lien fictif entre l'espace pixel et l'espace transformé. Etablir ce lien revient à définir une correspondance entre une région de l'image ou un bloc et une entité élémentaire qui lui est équivalente et qui émane de la pyramide. Cette entité regroupe un certain nombre de vecteurs sur lesquels sera appliqué un codage hybride construit autour de la quantification vectorielle.

XII-2. Principe de correspondance pixels-coefficients

L'approche prise ici est celle où l'image est formée d'un bloc de taille 8x8. Si celui-ci est décomposé en une pyramide à trois niveaux, la décomposition fictive et imaginaire, illustrée par la figure (12.1), sera obtenue. Cette décomposition n'est pas réalisable en général, car les filtres peuvent avoir des tailles plus grandes que celle du bloc (Le filtre de Haar, par contre, est apte à réaliser cette opération). Cette décomposition fictive a pour objet d'établir seulement un lien entre un bloc pixel et son correspondant en plan transformée. L'exemple de la figure (12.1) est relatif à une telle décomposition en trois niveaux.

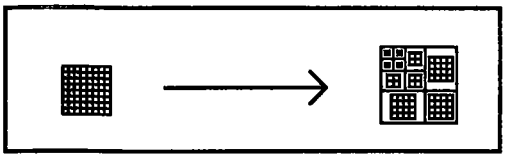


Figure 12.1 : Décomposition fictive sur trois niveaux d'un bloc élémentaire de taille 8x8.

La pyramide élémentaire obtenue est constituée par les éléments suivants, figure (12.2):

Niveau 4: constitué d'un seul coefficient qui représente le terme basse fréquence du bloc, il sera noté DC.

Niveau 3: constitué de trois éléments séparés qui sont les coefficients ondelettes relatifs à la troisième décomposition. Ils décrivent les activités du bloc à ce niveau de résolution. Ils seront notés AC1_d (où d= H, V ou D est relatif à une direction : horizontale, verticale ou diagonale).

Niveau2 : constitué de trois vecteurs de dimension 2x2 relatif chacun à une direction spécifique. Ces éléments sont les coefficients ondelettes relatifs à la deuxième décomposition et décrivent les activités du bloc à ce niveau de résolution. Ils seront notés AC2_d (d=H, V ou D).

Niveau 1: constitué de trois vecteurs de dimension 4x4, relatifs chacun à une direction spécifique. Ces éléments sont les coefficients ondelettes relatifs à la première décomposition et décrivent les activités du bloc à ce niveau de résolution. Ils seront notés AC3_d (d=H, V ou D).

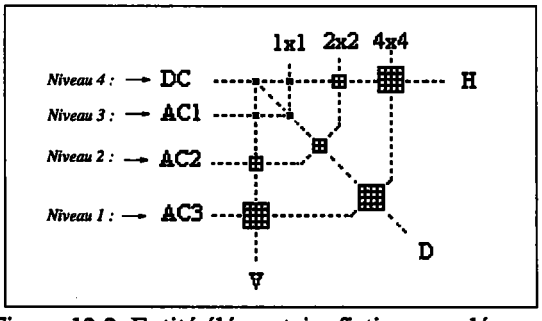


Figure 12.2 :Entité élémentaire fictive, appelée sous pyramide, relative à un bloc de taille 8x8

On va considérer cette représentation comme la plus petite entité pyramidale à trois niveaux de décomposition (M=3) relative à un bloc de pixel de taille 8x8, figure (12.2).

XII-3. Réorganisation de la pyramide

Le principe précédent est considéré, c'est à dire à chaque bloc pixel de taille 8x8 est attribuée l'entité élémentaire correspondante. Ces entités sont appelées *sous-pyramides.* Si une image de taille plus grande est décomposée en pyramide d'ondelettes sur trois niveaux (M = 3), on procède à l'extraction de toutes les sous-pyramides. La pyramide d'ondelettes est alors réarrangée sous une forme illustrée par la figure (12.3), où on peut noter aisément la notion de correspondance plan pixel et plan transformé dans un cas plus général.

L'établissement d'une telle correspondance entre un bloc de taille 8x8 de l'image et l'entité correspondante, ou sous-pyramide, extraite de la pyramide mère offre plusieurs avantages. D'une part, l'aspect vectoriel des coefficients ondelettes est introduit, d'autre part, cette correspondance permet une description locale de l'image.

Dans la suite, les avantages d'une telle correspondance seront développés, en particulier, sa contribution sur les plans codage et classification.

XII-4. Codage hybride des sous-pyramides d'une pyramide d'ondelettes

Le codage de la pyramide ondelette qu'on propose découle de la formation des sous-pyramides ondelettes décrite précédemment. Sachant que cette sous-pyramide (ou entité élémentaire) est formée de quatre niveaux. Seul le premier niveau est singulier, les autres sont formés de trois éléments séparés correspondant aux directions analysées. Les éléments des niveaux 1 et 2 sont formés de vecteurs de dimension 1x1, les niveaux 3 et 4 sont formés respectivement de vecteurs de taille 2x2 et 4x4.

Dans les chapitres précédents, relatifs au codage scalaire des coefficients de la pyramide, on avait remarqué que les niveaux de la pyramide correspondant aux signaux de basses fréquences supportent moins les erreurs de quantification que les signaux de hautes fréquences. L'oeil ne perçoit pas de la même manière le bruit de quantification introduit sur les différentes bandes de fréquence qui composent le signal.

La technique de codage proposée agit directement sur les vecteurs formés par les éléments de la sous-pyramide. Elle permet alors de coder la pyramide en tenant compte de la sensibilité de l'oeil aux erreurs. Le codage est présenté de la façon suivante :

Niveau 4: constitué de trois ensembles de coefficients de taille 4x4. Ces ensembles sont formés de vecteurs et sont codés alors par quantification vectorielle. Ces vecteurs ont une dimension 16.

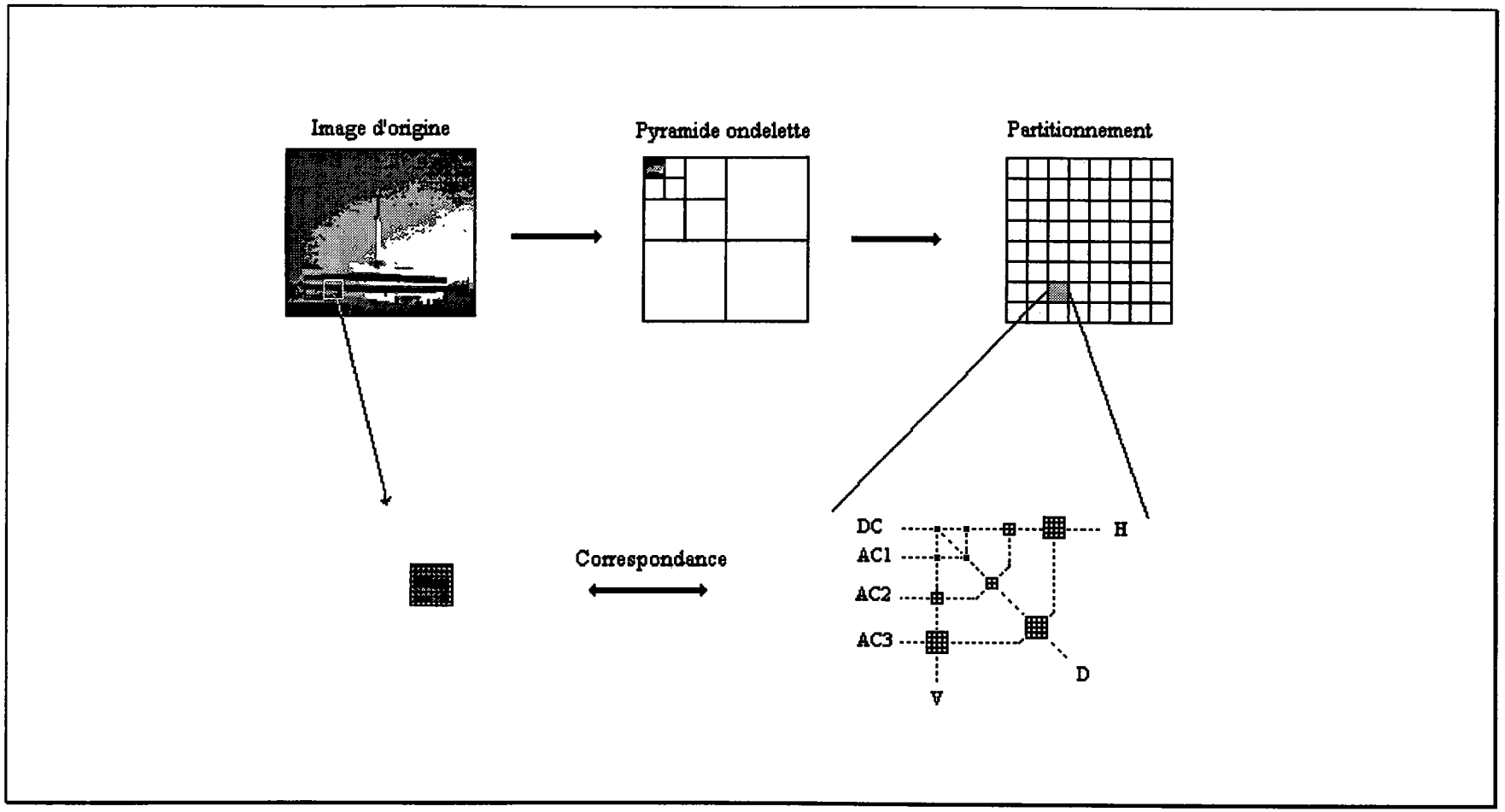


Figure 12.3 :Décomposition et de partitionnement d'une pyramide d'ondelette et principe de correspondance

Niveau 3: constitué de trois ensembles de coefficients de taille 2x2. Ces ensembles sont formés de vecteurs et sont codés aussi par quantification vectorielle. Ces vecteurs ont une dimension 4.

Niveau 2: constitué de vecteurs de dimension 1x1. Ce niveau lui est appliqué une quantification scalaire comme décrite en deuxième partie.

Niveau 1: formé d'un seul coefficient ou vecteur de dimension 1. Il est codé d'une manière prédictive par rapport aux sous-pyramides voisines. Ce choix de codage est établi par rapport à l'importance capitale de ces coefficients sur la qualité de l'image reconstruite.

La quantification vectorielle est utilisée pour coder les vecteurs des niveaux 3 et 4. Sachant que ces niveaux sont formés chacun de trois vecteurs. Chacun d'eux correspond à une direction privilégiée de la décomposition dyadique en pyramide d'ondelettes : horizontale, verticale et diagonale. Une quantification vectorielle nécessite un dictionnaire. Puisqu'il y a trois catégories de vecteurs, nous avons opté pour des dictionnaires séparés, relatifs chacun à une direction spécifique. Par conséquent il en faut trois par niveau.

Le deuxième niveau sera codé par quantification scalaire. Cette dernière est du même type que celle largement traitée en deuxième partie. Cette quantification sera du type LCSQ de l'anglais "*level constrained scalar quantization*", pour un débit binaire donné il faut définir le quantificateur optimal. A ce niveau, il faut aussi gérer le problème d'allocation de bit développé dans la seconde partie.

Enfin le coefficient DC du premier niveau de la pyramide est codé par prédiction, par rapport aux coefficients voisins. Ce codage est du type DPCM "*différential pulse coding modulation*" [Llo82].

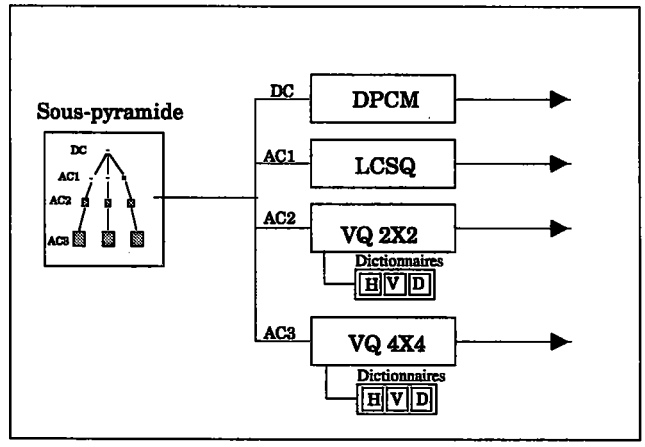


Figure 12.4 : Composition du codeur pour coder les différents éléments constituant les sous-pyramides.

Le dispositif de codage est formé alors par quatre étages de codeurs. Il est illustré par le schéma de la figure (12.4). On trouve au premier étage le codeur DPCM, au deuxième le codeur LCSQ et les quantificateurs vectoriels ainsi que leurs dictionnaires aux troisième et quatrième étage.

XII-5. Classification des sous-pyramides

La correspondance entre blocs pixel et sous-pyramides permet d'un côté un codage cohérent utilisant différentes techniques adaptées aux différents niveaux. D'un autre côté, il est possible de tirer profit du fait qu'une décomposition en ondelettes permet de localiser la singularité d'un signal. Dans le cas d'une image, ces singularités correspondent aux contours. La correspondance établie précédemment devient un outil de description locale de l'image.

Si on considère les activités élémentaires des blocs, c'est à dire, des blocs du type vertical, horizontal..., on peut faire une classification de ces derniers par l'intermédiaire des sous-pyramides correspondantes. Cette classification est fonction des activités des éléments qui composent la sous-pyramide. Des seuils d'activités sont nécessaires pour classer un bloc. Un vecteur d'une sous-pyramide est pris en considération si son activité est supérieure au seuil correspondant, figure (12.5). L'activité d'un vecteur est mesurée par l'énergie des coefficients qui le composent (équivalente au module du vecteur).

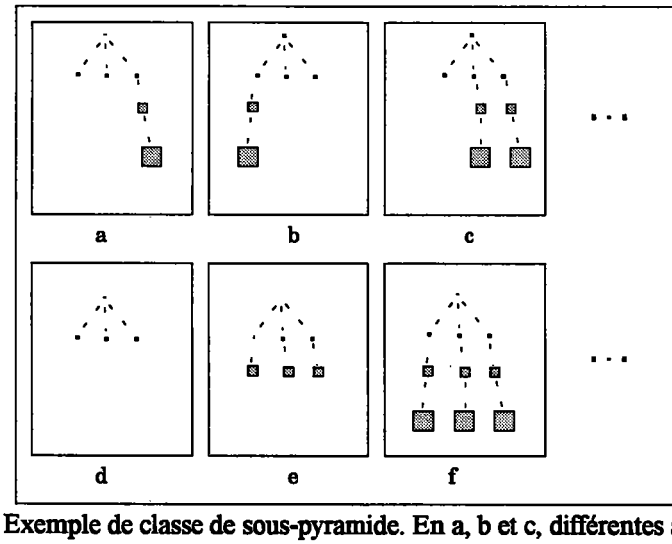


Figure 12.5 : Exemple de classe de sous-pyramide. En a, b et c, différentes activités fréquentielles. En d, f et e différentes activités directionnelles

On définit deux type d'activités d'un bloc :

• activités directionnelles : le bloc a des vecteurs prépondérants dans une direction donnée (exemple, bloc de transition vertical). Dans ce cas, seuls les vecteurs correspondant aux activités verticales sont pris en compte dans le codage. L'intérêt de cette opération est de réduire le nombre de vecteurs nécessaires pour coder un bloc.

Parmi les classes relatives à des activités directionnelles on trouve des blocs d'activité verticale, horizontale, presque verticale, presque horizontale..., figure (12.5a, b et c).

• activités fréquentielles : Elles décrivent l'activité fréquentielle des blocs. Ce qui permet leur classification en tant que bloc à haute, moyenne ou basse fréquence, figure (12.5d, e et f).

En combinant les deux types d'activités, nous avons élaboré une partition à 32 classes suivant leurs activités et directionnelles et fréquentielles. Un exemple de cette classification appliquée à des blocs d'une image est donné par le tableau (12.1) :

Tableau 12.1 :Exemple de classification des blocs d'une image en fonction de leurs activités

Activ. fréq \ Activ. direct.	H	V	H&V	V&D	Autres
AC3_ACTIVITE	42	25	24	74	94
AC2_ACTIVITE	11	53	3	0	81
AC1_ACTIVITE	35	85	0	0	217
Total en %	9,1	17	3	8	40,8

Le type de classe et les éléments sélectionnés sont éventuellement codés en CLV "code à longueur variable". La structure globale du codeur final [Raf94] est illustrée par le schéma de la figure (12.6).

XII-6. Application

Des expérimentations psychovisuelles ont montré que la sensibilité visuelle humaine à une grille sinusoïdale dépend d'une part de la fréquence spatiale et d'autre part de l'orientation spatiale. A propos de l'orientation spatiale, des résultats obtenus par Campell et Kulikovski [Cam65] montrent que le système visuel humain a un maximum de sensibilité pour une orientation verticale ou horizontale. Entre les deux, la sensibilité décroît uniformément pour atteindre un minimum pour une orientation diagonale du signal (une inclinaison de 45° de la grille).

Puisqu'une décomposition en pyramide d'ondelettes fournit déjà des vecteurs d'orientation, alors il faut tenir compte dans le codage de cette particularité de la vision humaine.

Par conséquent, les vecteurs diagonaux seront codés avec moins de précision que les vecteurs des autres orientations et du même niveau. Dans le codage présenté ci-dessus seuls les vecteurs diagonaux des niveaux 3 et 4 (vecteurs AC2_D et AC3_D) vont subir cette modification. Le vecteur AC3_D sera carrément négligé et le vecteur AC2_D aura un

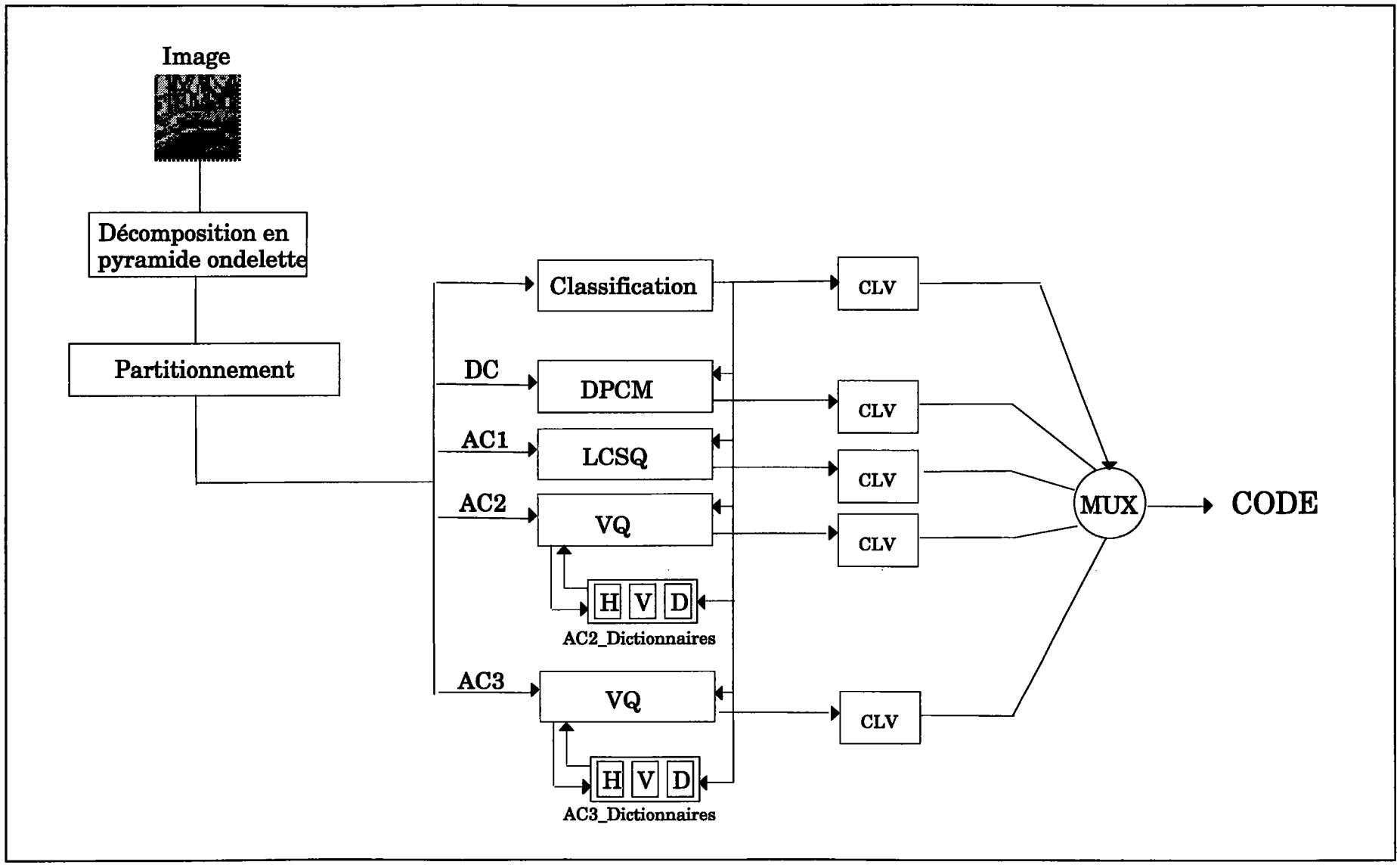


Figure 12.6 : Synoptique du codeur globale

dictionnaire plus petit que les deux autres composantes.

Avoir des dictionnaires multiples pour différents vecteurs constituant la sous-pyramide a permis de ne pas se focaliser sur la recherche du dictionnaire "universel" puisqu'il n'existe pas. Mais l'ensemble des dictionnaires que nous avons construit a été utilisé pour coder des images qui n'ont pas contribué à la construction de ces dictionnaires. Ce qui confère à cet ensemble un certain degré "d'universalité".

Pour la formation des dictionnaires on s'est servi de quatre images, figure (12.7a). Tous les dictionnaires ont une taille de 256, sauf celui des vecteurs AC2_D dont la taille est de 16 vecteurs.

Les coefficients du niveau 2 sont codés par un quantificateur scalaire. Comme décrit dans la deuxième partie, il faut fixer le nombre de bits pour ce niveau et résoudre le problème de l'allocation des bits aux différentes composantes. Pour les applications ultérieures le nombre de bits affecté à ce niveau est de 12.

Si on n'utilise pas de codage entropique ni de classification, le débit binaire est fixe à l'issu de la compression. Dans le cas contraire, le débit est variable. Ceci est décrit en détail dans les paragraphes suivants.

12-6-1. Codeur à débit fixe

Si on inhibe le codage entropique et la classification, le codeur de la figure (12.6) aura un débit fixe. La raison en est que tous les vecteurs de la sous-pyramide sont codés avec un nombre fixe de bit. Une sous-pyramide nécessite 56 bits pour être codée comme décrit par le tableau (12.2). Ce qui donne un débit moyen de 0.875 bpp (Bits Par Pixel).

Tableau 12.2 : allocation de bits aux composantes d'une sous-pyramide

	V	D	H	Niveau
AC1	8	0	8	16
AC2	8	4	8	20
AC3	12			12
DC	8			8
totale				56bits

Un premier exemple de codage, à débit fixe, est appliqué aux quatre images qui ont servi à la formation de la séquence d'entraînement pour générer les dictionnaires, figure (12.7a). La figure (12.7b) donne le résultat des images à la reconstruction. Les images ne présentent presque pas de dégradation notable.

a

b

Figure 12.7.- : en a) les images (256x256) qui ont sérvi à former les dictionnaires de coefficients ondelettes, en b) les résultats du codage pour des débits fixe de 0.875bpp (sans classification ni codage entropique).

12-6-2. Codeur à débit variable

Les images ne contiennent pas toutes les mêmes structures. Pour cela, il faut faire de sorte que le codeur s'adapte d'une part à la statistique des coefficients de la pyramide et d'autre part aux activités locales de l'image. Le codeur entropique et la classification remplissent parfaitement ces rôles. Les seuils utilisés dans la classification sont aussi un moyen pour régler le débit binaire total. Mais plus les seuils sont grands et plus la qualité est faible. Dans l'exemple de la figure (12.8), les seuils choisis sont de 16, 8 et 4 pour respectivement les niveaux 1, 2 et 3. Les débits binaires moyens obtenus sont compris entre 0.26 et 0.29 bpp, pour un PSNR voisin de 30db. Il faut noter que dans les exemples de la figure (12.8), les images codées ont des tailles 512x512. Mais elles utilisent les mêmes dictionnaires que précédemment. Cependant, ces images n'ont pas fait partie de la séquence d'entraînement pour construire les dictionnaires. La qualité des images codées justifie la validité des dictionnaires pour une large panoplie d'images.

XII-7. Adaptation de la TCD au codeur

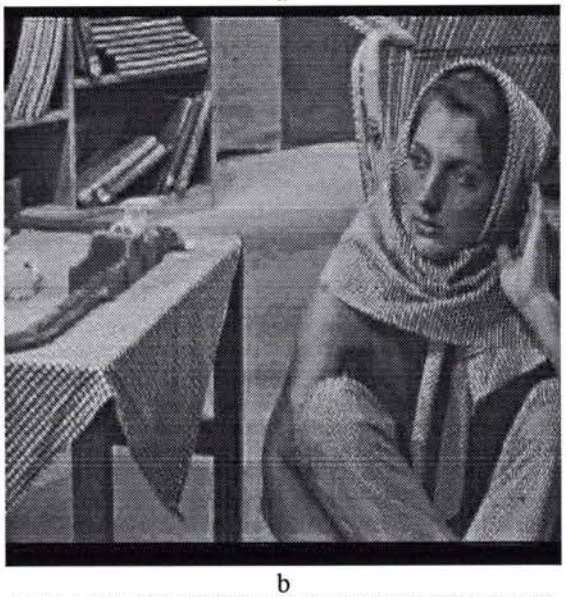
Le codeur décrit ci-dessus a une architecture telle qu'il s'adapte aisément à d'autres transformations, en particulier la TCD, [Raf93a]. Dans le cas de la TCD, les fonctions de bases procèdent à une analyse fréquentielle relative à un bloc, donc "semi-local". Si on regroupe ces fonctions de façon à faire une analyse fréquentielle hiérarchique, figure (12.9), on peut prétendre à une représentation multirésolution. Ainsi, les coefficients TCD sont regroupés pour former une configuration équivalente à la sous-pyramide d'ondelettes utilisée précédemment. La répartition des coefficients de la TCD 2D et les activités correspondantes sont illustrées par le schéma de la figure (12.10).

Le codeur est par conséquent parfaitement adapté à cette configuration. Le seul élément à mettre à jour est le contenu des dictionnaires. Dans ce cas, les mêmes quatre images que précédemment vont servir à générer les dictionnaires propres aux coefficients de la TCD.

Les résultats de ce type de codage appliqué aux images sont donnés en figure (12.11). D'autres exemples de codage pour différentes images sont donnés par la figure (12.12) et montrent la validité des dictionnaires.

Pour des taux de compression élevés des artefacts peuvent apparaître. A ces artefacts s'ajoute l'effet de bloc qui est lié au caractère local de la TCD.

a



b

c

Figure 12.8 : Exemples d'images (512x512) codées en utilisant les dictionnaires précédents et la classification, a)"La cornouaille" à 0.27bpp, PSNR=31.97db. b)"Barbara" 0.29bpp, PSNR=30.15db. c) "Fille" 0.26bpp, PSNR=32.45db.

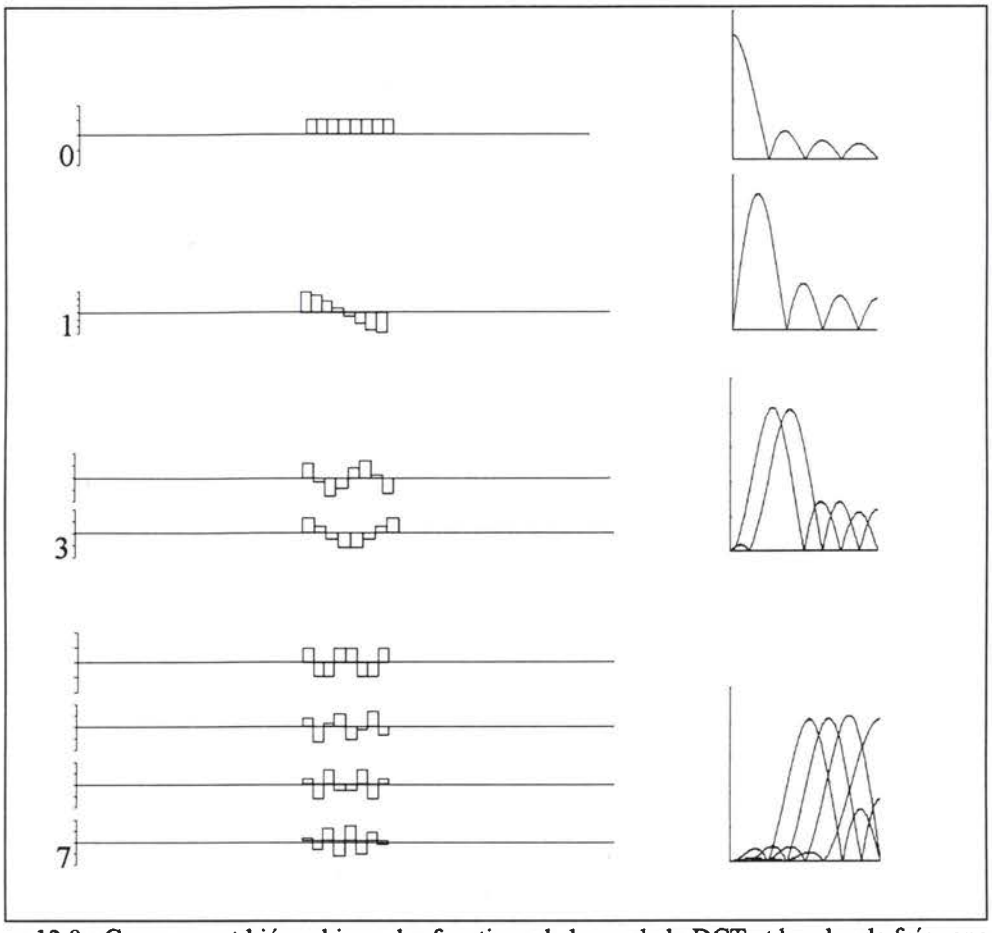


Figure 12.9 : Groupement hiérarchique des fonctions de bases de la DCT et bandes de fréquence analysées.

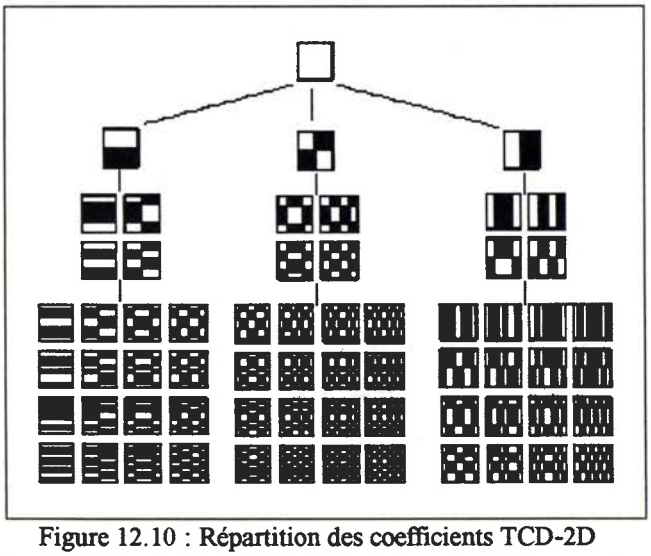


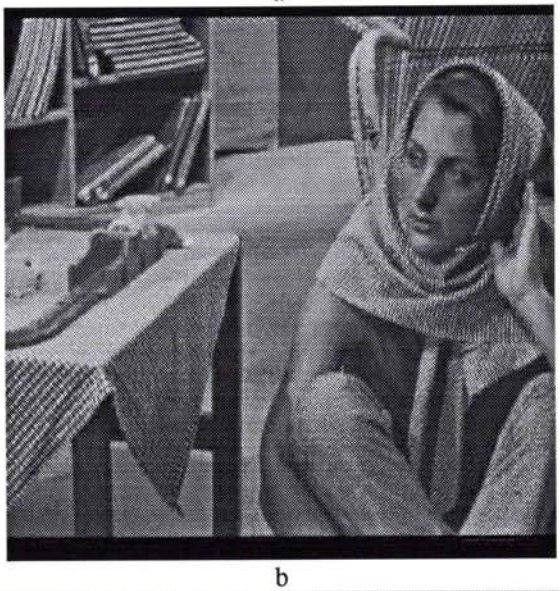
Figure 12.10 : Répartition des coefficients TCD-2D
en une structure équivalente à la sous-pyramide ondelettes.

a

b

Figure 12.11 : en a) les images qui ont sérvi à former les dictionnaires de coefficients TCD, en b) les résultats du codage par adaptation de la TCD au codeur pour des débits de l'ordre de 0.5bpp (sans classification).

a



b

c

Figure 12.12 : Exemples d'images (512x512) codées sans classification en utilisant les dictionnaires TCD précédents, a)"La cornouaille" à 0.46bpp, PSNR=33db. b)"Barbara" 0.52bpp, PSNR=32db. c) "Fille" 0.5bpp, PSNR=33db.

XII-8. Conclusion

Le codeur, décrit dans cette partie, est essentiellement basé sur la configuration des sous-pyramides. Elles sont extraites d'une pyramide mère à trois niveaux de décomposition. Elles sont constituées de composantes de natures et de tailles différentes. Leur mise au point a permis d'une part, d'établir un codage adéquat relatif à chaque composante et d'autre part, de tenir compte des particularités de la vision humaine.

Avec la notion de correspondance plan pixel et plan transformé, il était possible d'introduire la classification des sous-pyramides. Cette classification est basée essentiellement sur les activités fréquentielles et directionnelles des sous-pyramides. Ainsi, un codage adaptatif au contenu réel de l'image a pu avoir lieu. Il est possible alors de fixer le débit à une valeur limite, et en dessous, la classification adapte le codage au contenu local de l'image. Des débits d'environ 0.26 bpp ont été obtenus pour des PSNR voisins de 30db.

Enfin, des transformations, qui peuvent être dotées d'une représentation multi-résolution, sont aisément adaptables à notre codeur. C'est le cas de la TCD. La seule modification introduite est la mise à jour du dictionnaire relatif à cette nouvelle famille de coefficients. Les résultats obtenus sont équivalents, à part l'effet de bloc, dans le cas de la TCD, qui apparaît pour des taux de compression élevés. C'est le handicap de l'utilisation de la TCD et qui donne un avantage certain à la pyramide ondelettes.

Pour élargir la validité de notre procédé, des extensions du système proposé aux images couleurs et aux séquences d'images sont abordées dans le chapitre suivant.

XIII. Extension aux Images Couleurs et aux Séquences d'Images

XIII-1. Introduction

Ce dernier chapitre décrit des extensions possibles de notre codage, en particulier, aux images couleurs et aux séquences d'images. Dans le cas des images couleurs, nous allons considérer deux formats. Le premier est lié au format 4:2:2 de la recommandation CCIR 601. Le second est lié aux images avec palettes arrangées développées à TDF_C2R. Dans le cas de séquences d'images, deux types de codage seront considérés. Le premier est un codage direct sans estimation de mouvement mais dans lequel nous introduisons un nouveau procédé d'interpolation. Le second est un codage avec estimation de mouvement. Cette estimation sera basée sur la méthode dite estimation par correspondance de bloc dans laquelle nous incluons une technique d'optimisation de recherche de vecteur de déplacement.

XIII-2. Codage d'images couleurs

La perception et la représentation des couleurs sont principalement tridimensionnelles et c'est dans un tel domaine que les couleurs sont perçues et évaluées [Lim77].

L'extension de l'image monochrome à l'image couleur se fait donc par l'addition de deux composantes chrominances. Ensemble, les trois composantes forment une image reconnue par un observateur humain comme une image en couleur.

L'objet de ce paragraphe est d'étendre les techniques de codage vues dans cette partie aux images couleurs et de présenter des exemples de codage en ondelettes des images couleurs. Deux représentations d'images couleurs seront traitées, images couleurs en *YIQ* et images palettisées.

13-2-1. Codage des composantes images couleurs

Les trois couleurs principales pour représenter une image couleur sont les composantes *RGB*. Cependant, il est largement admis que le domaine *RGB* n'est pas très efficace et il est préférable de travailler dans le domaine *YIQ* (standard NTSC). Ce choix est justifié par le fait que les composantes *RGB* sont plus corrélées que les composantes *YIQ*. Le passage d'un domaine à l'autre se fait par transformation linéaire (13.1). Le standard CCIR (Comité Consultatif International de la Radio) de télévision numérique a fixé l'allocation des bits de la luminance et la chrominance à 4:2:2. Ce qui revient à coder les chrominances avec deux fois moins d'échantillons lignes que la luminance.

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.597 & -0.277 & 0.321 \\ 0.213 & -.523 & -0.309 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \qquad (13.1)$$

La figure (13.1a) donne un exemple d'images couleurs 512x512. La luminance Y a une taille de 512x512, les deux chrominances I et Q ont une taille de 256x512. La première compression est liée à l'élimination de la bande haute fréquence colonne des composantes de chrominance par les mêmes filtres de décomposition en ondelettes. Les composantes résultantes à la suite de cette opération sont I' et Q' de taille 256x256. Par conséquent, nous avons appliqué notre codage aux trois composantes Y et $I'Q'$ de tailles respectivement 512x512 et 256x256.

Codée avec un taux de 0.27 bpp pour la luminance et 0.25 bpp pour les chrominances I' et Q', l'image reconstruite, en l'occurrence "bateau", figure (13.1b), a un débit moyen de 0.395 bpp. Ce qui donne des taux de compression de 60 et de 40 par rapport aux images respectivement RGB (24 bits par pixel) et *YIQ* (16 bits par pixel). La qualité reste néanmoins acceptable.

13-2-2. Images couleurs palettisées et organisées

Le principe d'une image palettisée est de coder chaque pixel par une couleur existant dans une palette. Une palette est un ensemble de couleurs (par exemple 256 couleurs) les plus représentatives de l'ensemble de départ. Ainsi, la valeur de chaque pixel est une adresse d'une couleur dans la palette. Si la taille de la palette est de 256 couleurs, les adresses varient entre 0 et 255. Par conséquent, l'image est codée avec 256 couleurs seulement.

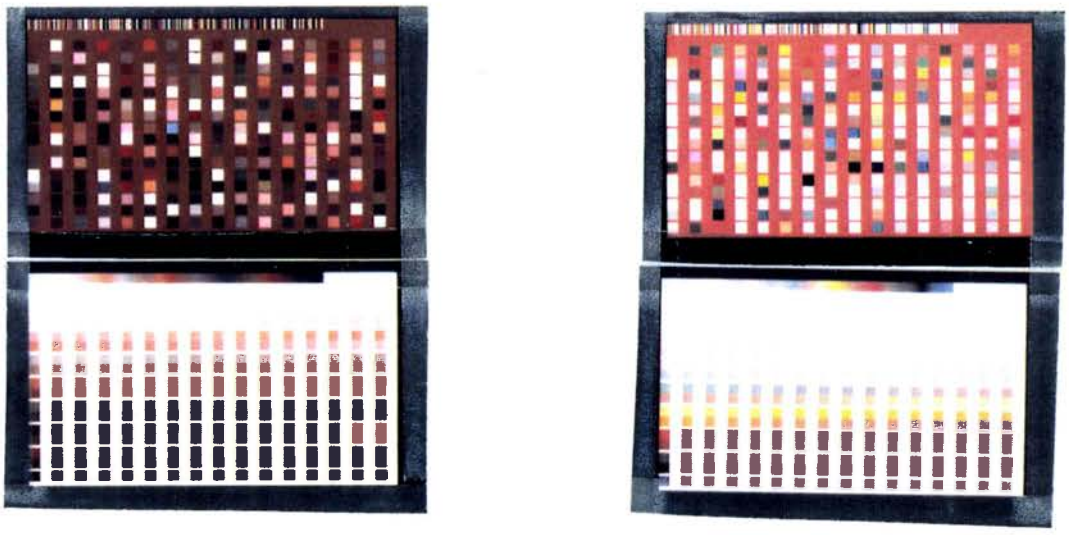
Coder directement ces images donne un résultat méconnaissable. L'idée de base était de réorganiser la palette pour rendre possible cette opération. En effet, un procédé a été développé à TDF-C2R [Lap94], à base des réseaux de neurones et qui génère des palettes organisées, figure (13.2b).

Il est possible à ce stade de coder ce type d'image de manière identique à une image en

a b

Figure 13.1 : Application du système de codage aux images couleurs. a) Images originales, b) Images codées.

a



b

Figure 13.2 : Application au codage d'images palettisées. a) Images palettisées aprés décodage pour un débit de 0.6 bpp. b) Palettes correpondantes avant et aprés organisation.

niveau de gris. La figure (13.2a) donne un exemple de codage des images "clown" et "enfant" par ces méthodes. Cependant, ces images restent néanmoins difficiles à coder. Des qualités acceptables ont été obtenues seulement pour des taux supérieurs à 0.6 bpp sur l'image monoplan palettisée. Le nombre de bits total pour coder une image 512x512 palettisée reste encore supérieur à celui nécessaire pour coder une image en *YIQ*.

Ce qu'il faut retenir c'est la possibilité maintenant de coder directement des images palettisées. Du point de vue de la qualité, on ne peut substituer au codage d'images couleurs en 3 plans, le codage d'images palettisées pour le moment et il faut encore améliorer le système de réorganisation de la palette pour atteindre cet objectif.

XIII-3. Codage séquences d'images

Les images de télévision sont formées de lignes et de trames. La corrélation intra-image et inter-image a permis le développement des différents types de prédicteurs qui opèrent respectivement dans une, deux ou trois dimensions (2 dimensions spatiales et une dimension temporelle).

La recommandation 601 du CCIR a défini le format vidéo numérique du système de télévision, 525-lignes et 625-lignes. Ce standard a pour objet de faciliter le développement d'équipements ayant des caractéristiques communes permettant l'échange international des programmes. Ainsi, le CCIR Specialist Goup (SGXV) a proposé un nouveau format appelé CIF (Common Intermediate Format). Le nombre d'échantillons par ligne active est 360 pour les luminances et 180 pour chaque composante chrominance. Le nombre de lignes par image active est de 288 pour la luminance et 144 pour les deux chrominances.

Pour coder ces séquences d'images, nous proposons dans ce paragraphe deux types de codages. Le premier est basé sur le seul codage intra-image. Le second fait intervenir en plus du codage intra-image l'estimation de mouvement.

13-3-1. Codage de séquence d'images sans estimation de mouvement

a) codage direct

C'est le cas le plus direct pour coder des séquences d'images. Les images sont codées séparément les unes à la suite des autres et sans estimation de mouvement. La figure (13.3) donne les quatre premières images de la séquence "salesman". Chacune des images de la séquence est codée directement avec le codeur décrit dans le chapitre précédent. La redondance temporelle n'est pas prise en considération

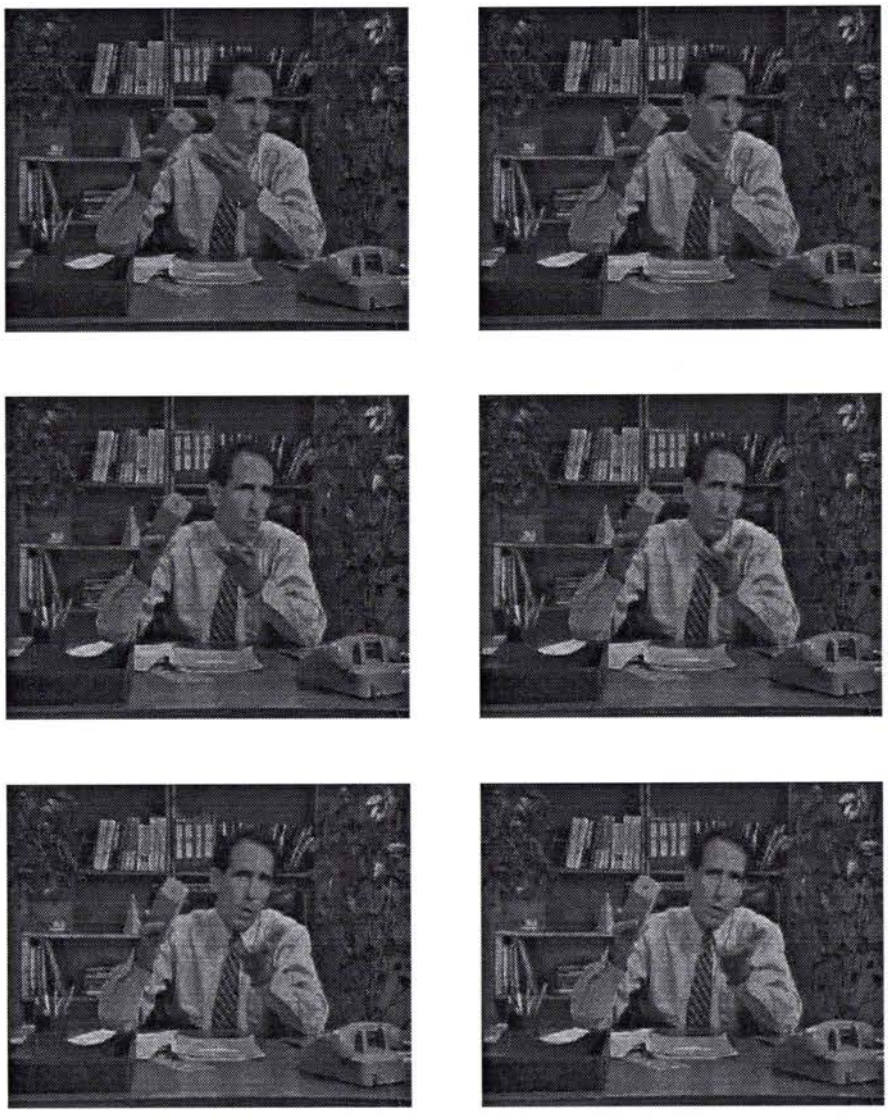


Figure 13.3 : Résultats du codage direct de séquence d'images (CIF) avec un débit moyen de 0.27bpp et un PSNR voisin de 30db.

b) Codage par interpolation

Ce cas ressemble beaucoup au codage décrit ci-dessus. Cependant, nous avons décidé de ne coder qu'une image sur 2 (ou 3...). A la reconstruction, les images qui manquent sont déduites par interpolation. Le procédé utilisé ici pour réaliser l'opération d'interpolation est à base de TCD, un procédé qui a fait l'objet d'un dépôt de brevet [Raf93b].

La figure (13.4) donne un exemple de résultat du codage, les images paires sont codées directement par notre schéma de codage décrit au chapitre précédent, alors que les images impaires sont déduites par le procédé d'interpolation.

13-3-2. Codage de séquence d'images avec estimation de mouvement

13-3-2-1. Introduction

Une séquence d'images n'est autre que la fonction image à deux dimensions, espace (x, y), auxquelles vient s'ajouter une nouvelle dimension, le temps. En général, une séquence est un ensemble d'images prises à des intervalles de temps assez proches, de l'ordre de la ms. Ceci peut aller de 2 à 50, voire même 120, images dans la seconde. Souvent, on peut choisir entre différentes cadences (7, 30 ou 50 images par seconde comme c'est le cas en MPEG). De toute façon, à ces cadences et pour deux images successives on peut considérer qu'il y a peu de changement. Partant de cette considération, il est possible de considérer qu'il existe une forte redondance inter-images. Tous les codeurs seront basés sur cette redondance d'ordre temporel qui s'ajoute à une redondance spatiale longuement abordée dans les chapitres précédents. Le but sera alors de tirer profit de toutes ces redondances pour réduire le débit binaire nécessaire au codage de telles séquences d'images.

Au premier abord, l'idée d'utiliser un codage par prédiction semble des plus envisageables. Ce qui était utilisé pour coder des données de type son ou autre. Cela revient simplement à la généralisation de la technique de prédiction à un signal à deux dimensions. Il découle de cette approche une technique de prédiction multicanaux, où chaque canal correspond à un pixel $P_{ij}(t)$ situé à la position de coordonnées (i, j). Ce pixel sera prédit à partir des valeurs du même pixel $P_{ij}(t-n)$ aux instants précédents.

Très vite, cette technique s'est avérée moins compétitive que d'autres méthodes. Ceci est dû à ce que le codage ne tient pas compte de la corrélation spatiale existante entre les pixels voisins. D'où découle l'idée de coder ensemble les groupes de pixels d'un espace ayant subi les mêmes transformations dans le temps. Les techniques qui mettent en évidence une telle corrélation spatio-temporelle s'avèrent fructueuses. Tout le problème est ramené à ce qu'on peut qualifier ou appeler, *estimation de mouvement*. On s'est intéressé à la technique dite *estimation de mouvement par correspondance de bloc,* connu sous le terme anglais "*block*

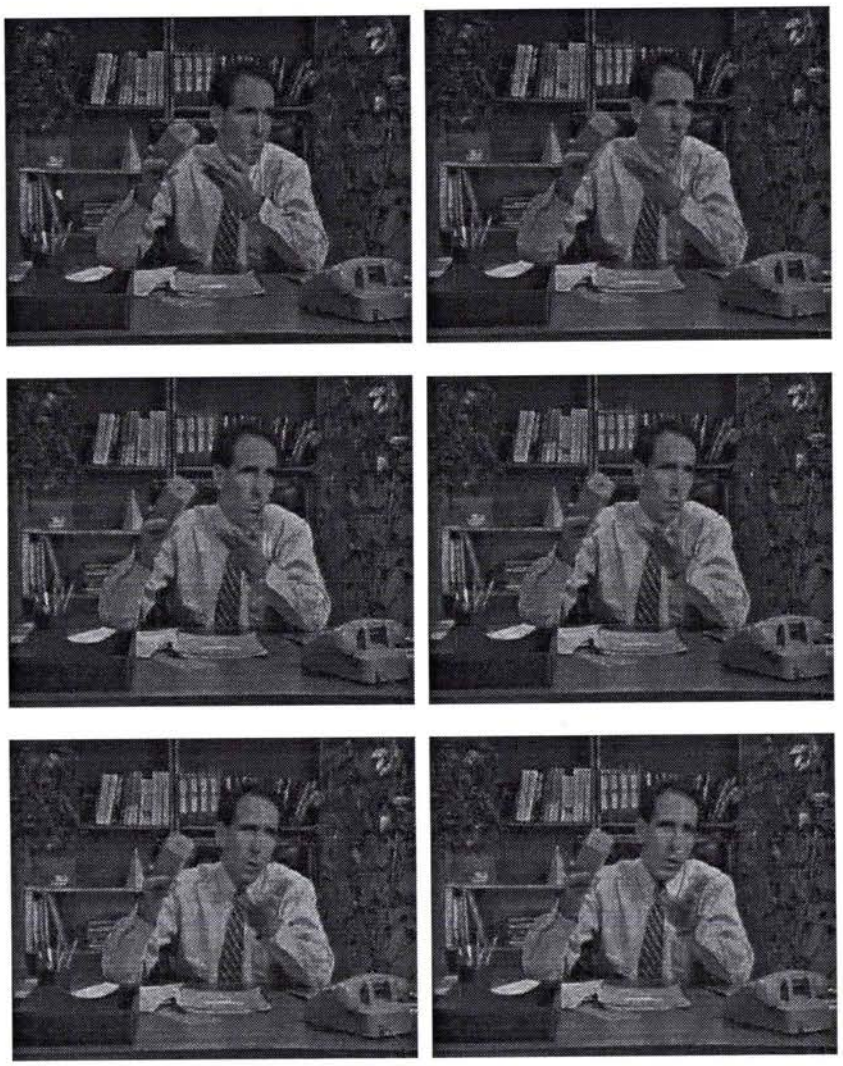


Figure 13.4 : Codage "direct plus interpolation" de séquence d'images. Codage direct des images paires (à droite) et interpolation des images impaires (à gauche) par un nouveau procédé d'interpolation à base de la TCD, (voir [Raf93b]). Le débit moyen est de 0.14bpp.

matching". Cette technique sera abordée dans cette section pour réaliser un codeur de séquence d'images.

13-3-2-2. Estimation de mouvement par correspondance de bloc

A la base, cette technique peut être considérée comme une "prédiction par zones" dans l'image courante à partir d'une ou des plusieurs images antérieures. Plus précisément, elle tente de détecter des déplacements. Elle est en mesure d'affecter des vecteurs déplacements $\vec{D}$ à des zones dans une image.

Si les vecteurs déplacement sont connus, il est aisé de construire l'image courante I(t) à l'instant t à partir de l'image précédente I(t-1) à l'instant t-1. Pour cela, il faut affecter à chaque zone de l'image I(t-1) son vecteur déplacement. Pour la simplicité, il sera question de bloc de taille NxN pour définir les zones d'image. Ainsi à chaque bloc de taille NxN de l'image courante correspond un déplacement qui nous ramène à un bloc de même taille dans l'image précédente.

Considérons un exemple très simple, où tout le plan image a subi un même déplacement. En affectant ce même déplacement à tous les blocs de l'image à l'instant (t-1), on est en mesure de construire l'image à l'instant t.

Le cas décrit ci-dessus est un cas rare. Ce qu'il faut retenir de l'estimation de mouvement par correspondance de blocs, c'est le fait de chercher un bloc dans l'image précédente qui ressemble au mieux à un bloc de l'image courante.

13-3-2-2-a. Vecteur déplacement

Les déplacements sont du type $\vec{D}\begin{pmatrix}\alpha\\ \beta\end{pmatrix}$. Le bloc $B_{ij}(t)$ sera construit à partir du bloc $B_{i-\alpha,j-\beta}(t-1)$, (i, j) indique la position du bloc courant et (i-a, j-b) la position du bloc déplacé de l'image à l'instant t-1. Le problème revient à définir le couple (a, b) lié à chaque bloc de l'image courante.

Il semble a priori raisonnable de définir une zone de recherche. Pour la simple raison que pour des laps de temps très brefs, les déplacements sont locaux (sauf cas particuliers). Pour réaliser l'opération "*block matching*", on affecte d'abord à chaque bloc courant un indice de position formé par les coordonnées (i, j) du premier pixel (haut à gauche) et une taille, carré en général, NxN. Le bloc qui va lui correspondre sera scruté dans une zone de recherche centrée sur le bloc d'indice (i, j) de l'image précédente, voir figure (13.5).

Le déplacement est calculé en déterminant dans la zone de recherche de I(t-1) le bloc Bi',j'(t-1) le plus proche du bloc Bi,j(t) courant de l'image I(t) au sens d'un certain critère. Si ce critère est la distance erreur quadratique moyenne. On peut écrire :

$$d_{i,j}(a,b) = 1/NN' \sum_{n,m} \left[I_t(i+n, j+m) - I_{t-1}(i+n+a, j+m+b) \right]^2$$

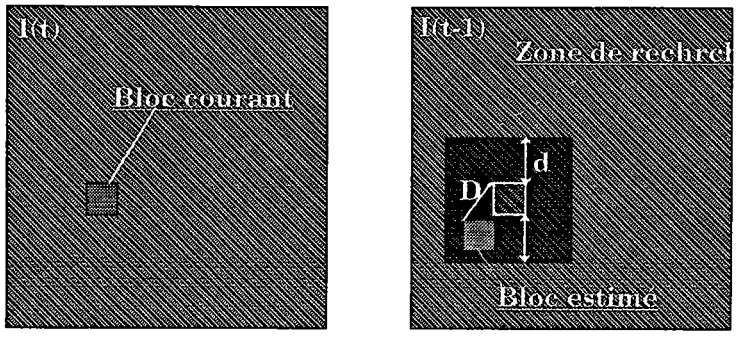


Figure 13.5 : Estimation du mouvement par correspondance des blocs.

Le vecteur déplacement (a,b) affecté au bloc Bi,j(t) est celui qui vérifie :

$$d_{i,j}(\alpha,\beta) = \min_{a,b} \{ d_{i,j}(a,b) \}$$

13-3-2-3. Caractérisation des blocs

Dans une séquence d'images, les blocs ne subissent pas les mêmes effets. La procédure de codage se charge de définir des classes pour différencier ces blocs.

D'abord l'image courante I(t) sera découpée en blocs de taille NxN. Ces blocs seront partagés en deux catégories. La première catégorie sera celles des *blocs non déplacés*, la seconde sera celles des *blocs déplacés*.

Un procédé permettra de savoir si effectivement un bloc a subi un déplacement ou pas. Ainsi, seuls ceux ayant subi un réel déplacement, seront sujets à une mesure de déplacement.

Ces derniers seront aussi partagés en deux catégories. La première est formée par les blocs déplacés sans distorsion notable, *blocs bien estimés*. La seconde catégorie sera formée par les blocs déplacés avec distorsion, *blocs mal estimés*.

Ainsi, les blocs bien estimés seront codés avec simplement un vecteur déplacement, tandis que les blocs mal estimés seront corrigés intégralement.

Ainsi la structure d'un tel système de codage est donnée par la figure (13.6) et résumée comme suit :

Etape 1 : le découpage de l'image en blocs et leur classification en de catégories blocs déplacés ou pas.

Etape 2 : estimation du déplacement par correspondance de bloc en mesurant les vecteurs déplacements relatifs aux blocs déplacés.

Etape 3 : classement des blocs déplacés en deux classes en fonction de la ressemblance, blocs déplacés bien estimés et blocs déplacés mals estimés.

Etape 4 : attribution des codes correspondant à chaque bloc en fonction de son appartenance à une classe.

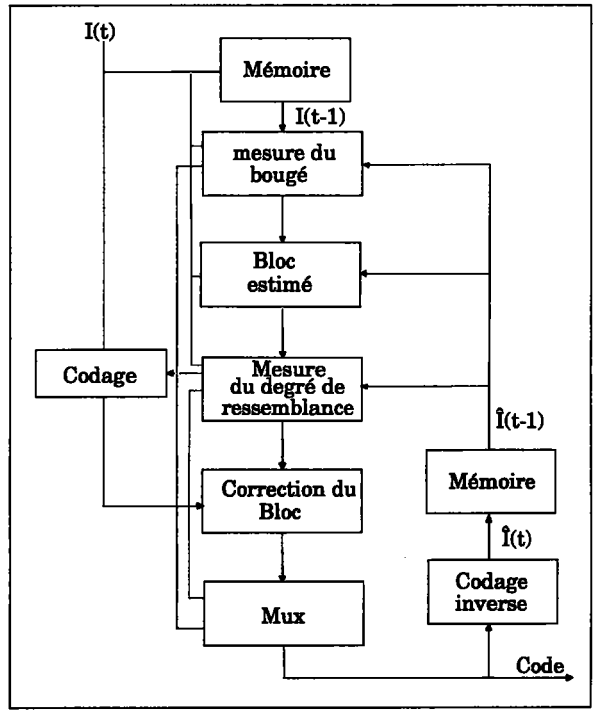


Figure 13.6 : Synoptique du codeur de séquence d'image.

13-3-2-4. Optimisation de la recherche du bloc le plus proche

L'opération de recherche du bloc estimé (le bloc le plus proche au sens d'un critère) est très coûteuse en temps. Par exemple, si le bloc est de taille 8x8, la zone de recherche de taille (2d+8), alors le nombre de comparaisons dans la zone de recherche est de :

$$(2dx2d)x(64.mult + 63.add) = 4d^2x127 \text{ opérations}$$

Si d=10 et pour une image de taille 360x288, figure (13.8), le coût total des comparaisons est de plus de 52 millions d'opérations.

Pour optimiser cette recherche, la technique suivante a été développée :

L'opération de recherche du bloc estimé ne se fait plus à la première résolution mais à la résolution inférieure, figure (13.7). Les blocs comparés sont alors de plus petite taille et la charge de calcul est réduite. La détermination du bloc recherché à ce niveau permet de localiser l'endroit du bloc effectif au niveau supérieur. Seule une recherche plus fine et moins coûteuse est faite à cette première résolution pour déterminer l'emplacement exact du bloc recherché à un pixel prés.

Au niveau inférieur des résolutions figure (13.7), l'opération totale de recherche ne coûte plus que 5 millions d'opérations au lieu de 52 millions. Ce qui donne un rapport de 6.25%

d'opérations. Un exemple de détermination des vecteurs déplacements par cette technique, appliquée à la séquence "Salesman", est donné par les figures (13.9). La figure (13.9a) donne le résultat de recherche à la résolution inférieure. Les figures (13.9 b et c) donnent les résultats des deux étapes suivantes, à savoir, passage à la résolution supérieure et dédoublement des vecteurs déplacement et la recherche locale pour affiner la précision des vecteurs déplacements de l'ordre du pixel. Cette dernière opération fait intervenir huit comparaisons supplémentaires. Le résultat final est quasi similaire à celui d'une recherche classique, comme le montre la figure (13.10).

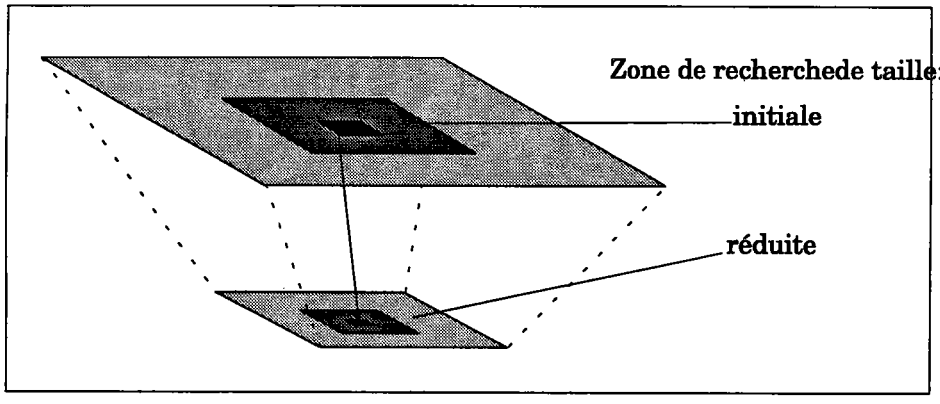
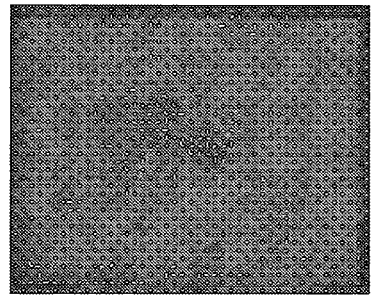


Figure 13.7 : Schéma simplifié de recherche du vecteur déplacement en résolution inférieure.

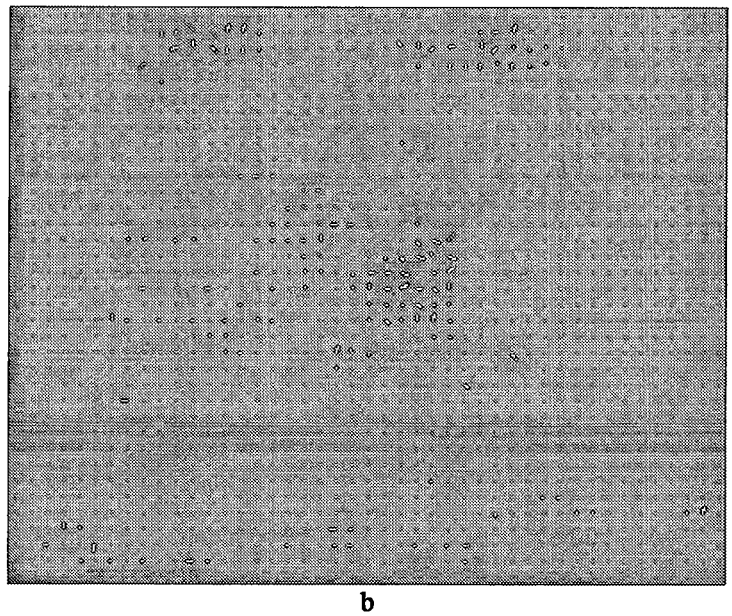
XIII-4. Conclusion

Ce chapitre nous a permis de montrer la souplesse de notre algorithme de codage d'images. Dans un premier temps, le codage qui était à la base dédié aux images monochromes a été étendu aux images couleurs. Deux façons de présenter les images couleurs ont été testées. Le premier codage est appliqué directement aux composantes brutes de l'image. Pour des taux de compression de 0.133 bpp, la qualité subjective de l'image est correcte. Le second est appliqué aux images palettisées. La palette est ordonnée par le procédé décrit dans [Lap94]. L'intérêt réside dans le fait que le codage et le décodage de ces images sont identiques au codage d'images monochromes.

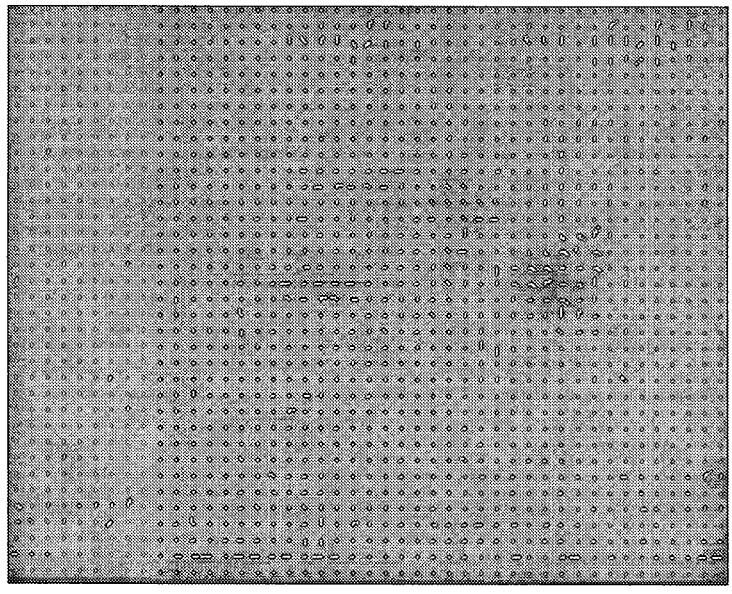
La seconde partie de ce chapitre est liée à l'extension de notre codage aux séquences d'images. Trois techniques ont été abordées pour coder ces séquences, codage direct, codage par interpolation et par estimation. La recherche des vecteurs déplacements a été optimisée dans le cadre du codage par estimation. Le codage par interpolation nous a permis d'introduire notre nouvelle technique d'interpolation d'images, un procédé à base de la TCD.



a

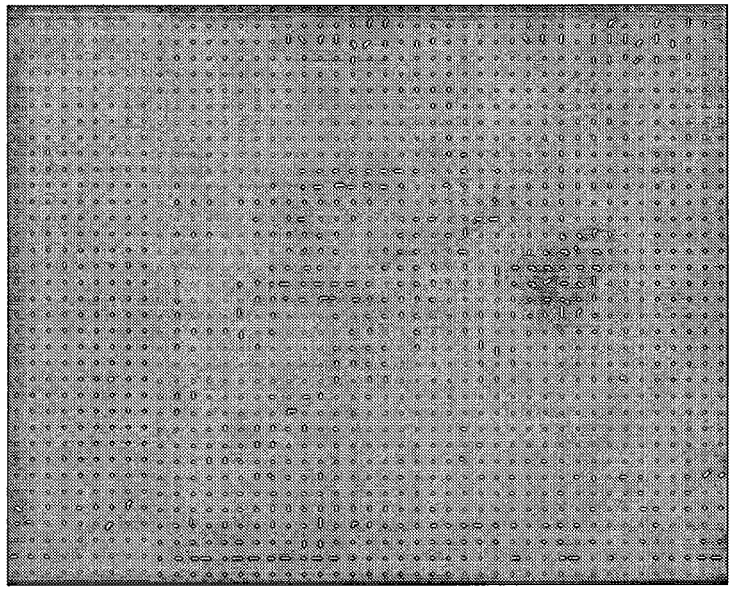


b

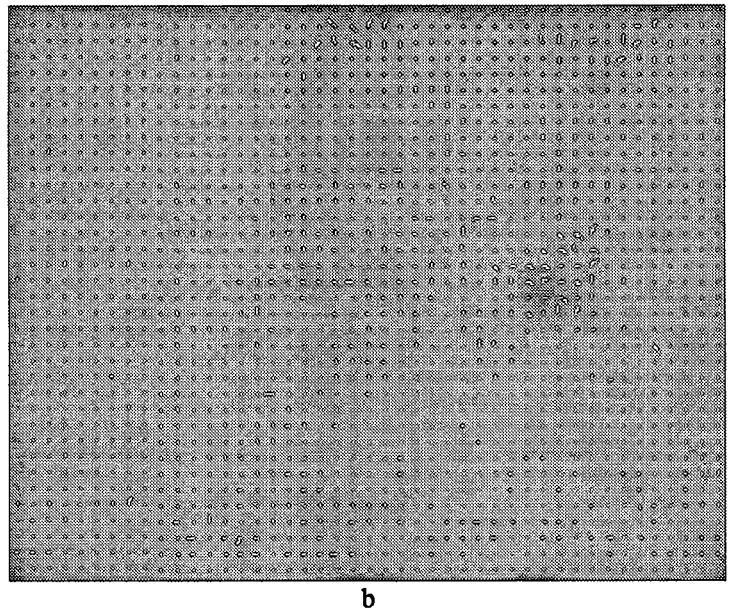


c

Figure 13.9 : Vecteur deplacement, a) en résolution inférieure, b) positionnement des vecteurs dans la résolution supérieure, c) précision sur les vecteurs déplacement.



a



b

Figure 13.10 : Comparaison des deux méthodes : en a) méthode classique, en b) par passage à la résolution inférieure.

CONCLUSION GENERALE

Nous avons dressé une synthèse de l'historique de la représentation en multirésolution et mis en évidence l'importance de la décomposition en ondelettes tant sur le plan mathématique que sur le plan du filtrage numérique. La décomposition en ondelettes est apparue comme un carrefour où se rejoignent plusieurs techniques de codage (la pyramide Laplacienne, les sous-bandes...).

La représentation de l'image en multirésolution a permis d'approcher les processus mis en oeuvre par le système visuel humain et a facilité la réalisation d'un codeur adaptatif. Ce codeur a la particularité de distribuer les erreurs de codage selon la sensibilité psychovisuelle aux différents signaux des différents niveaux. Pour la décomposition, nous avons pu établir des critères de sélection des bancs de filtres permettant une décomposition adaptée à la compression.

Deux catégories de codeurs ont été développées, la première catégorie est directe et ne nécessite aucune information préalable ("memoryless"). Elle est à base de quantification scalaire. Nous avons développé une méthode d'allocation de bits sur le principe d'approximation de Girsh et Pierce de l'erreur de quantification, afin d'allouer des bits aux différents signaux qui constituent la pyramide et un algorithme appelé "algorithme des intervalles glissants" pour un codage scalaire optimal des coefficients d'ondelettes.

Toujours dans le cadre scalaire, nous avons introduit le codage par zone morte aux signaux de la pyramide. Pour des besoins d'allocation de bits et faute d'équation analytique d'approximation de l'erreur de codage nous avons été amenés à réaliser cette allocation par approche algorithmique.

La deuxième catégorie est à base de codage vectoriel permettant des taux de compression plus élevés, mais nécessitant des dictionnaires préalablement établis. Nous avons décrit les principes du codage vectoriel et les méthodes de construction de ces dictionnaires. Pour réduire le temps de recherche des vecteurs de reproduction dans un dictionnaire, nous avons introduit des techniques d'optimisation de cette recherche basée sur l'ordonnancement des vecteurs dans le dictionnaire.

Un codeur hybride a été construit autour de la quantification vectorielle pour coder les structures fictives appelées sous-pyramides. Cette structure a permis un codage vectoriel adéquat et obéissant à des critères psychovisuels. Après la notion de sous-pyramide, nous avons introduit la classification, une technique qui permet un codage adaptatif des images en fonction du contenu fréquentiel.

Nous avons adapté les coefficients issus de la TCD à notre codeur moyennant leur présentation sous forme hiérarchique. Les résultats sont presque aussi performants qu'avec la

structure précédente.

Enfin, nous avons étendu notre système de codage aux images couleurs et aux séquences d'images. Dans le cas d'images couleurs, des débits d'environ 0.133 bpp ont été obtenus pour des qualités acceptables. Dans le cas de séquences d'images, deux types de codage ont été considérés. Le premier est un codage direct des images paires de la séquence avec la restitution des images impaires au décodage par un nouveau procédé d'interpolation. Le second est un codage par estimation de mouvement par correspondance de bloc où nous avons optimisé la recherche des vecteurs déplacements.

Néanmoins, plusieurs points restent à améliorer pour augmenter d'avantage les performances de notre système. Ils portent sur les deux composantes essentielles du codeur, à savoir l'analyse/synthèse et le codage scalaire/vectoriel.

En ce qui concerne l'analyse/synthèse, nous avons établi un critère de sélection des bancs de filtres. Puisque le codage doit obéir à des critères psychovisuels, nous avons pensé de même adapter les bancs de filtres à cette notion. La voie à suivre est liée à la longueur des filtres. Que ce soit la longueur des filtres du même banc (filtres plus courts pour les détails que pour les versions réduites), ou la longueur des filtres relatifs aux différents niveaux de décomposition (filtres plus courts pour les premiers niveaux de décomposition que les derniers qui sont toujours privilégiés par l'allocation optimale de bits).

Il s'est avéré que le codage par zone morte est une technique très intéressante. En fait, il faudrait lui consacrer un soin tout particulier. Notamment, si on introduit un codage par plage, on pourra améliorer les performances de manière remarquable. Car nous avons constaté effectivement des zones entières recouvertes de coefficients nuls.

Dans notre système de codage vectoriel, la multiplication des dictionnaires a permis de s'affranchir du problème "une image - un dictionnaire", puisque le même ensemble de dictionnaires a été utilisé pour toutes les images présentées dans ce mémoire. L'optimisation des dictionnaires permettrait certainement d'améliorer et la qualité de l'image et le taux de compression.

Enfin, le codeur se prêterait très bien à une réalisation matérielle adoptant une architecture parallèle, parallélisme concernant la transformée en ondelettes et parallélisme relatif au traitement des sous-pyramides. Les dix composantes de la sous-pyramide pourraient être codées en une seule fois.

Si l'on ajoute à cette machine de codage d'images fixes le module de codage de séquence d'images qui inclut la technique optimisée de recherche des vecteurs déplacements, un codeur vidéo présentant un excellent compromis entre la qualité de restitution, le taux de compression et la complexité d'implémentation, est obtenu.

Annexe :

On veut établir la relation suivante : $$\frac{1}{\sum_{i=1}^{n}\lambda_i}\sum_{i=1}^{n}\lambda_i x_i \le \left[\prod_{i=1}^{n}(x_i)^{\lambda_i}\right]^{1/\sum_{i=1}^{n}\lambda_i}$$

D'abord, on peut établir que si f est une application convexe de l'intervalle $I \subset R$ dans R, alors : pour tout $(x_1,x_2,\ldots,x_n) \in I^n$ et pour tout $(\alpha_1,\alpha_2,\ldots,\alpha_n) \in [0,1]^n$ tels que $\sum_{i=1}^{n}\alpha_i = 1$, on a

$$f\left(\sum_{i=1}^{n}\alpha_i x_i\right) \le \sum_{i=1}^{n}\alpha_i f(x_i)$$

Soit $(\lambda_1,\lambda_2,\ldots,\lambda_n) \in R^n$;

alors $\left(\frac{1}{\sum_{i=1}^{n}\lambda_i}\lambda_1, \frac{1}{\sum_{i=1}^{n}\lambda_i}\lambda_2, \ldots, \frac{1}{\sum_{i=1}^{n}\lambda_i}\lambda_n\right) \in [0,1]^n$ et $\sum_{j=1}^{n}\frac{1}{\sum_{i=1}^{n}\lambda_i}\lambda_1 = 1$

et $$f\left(\frac{1}{\sum_{i=1}^{n}\lambda_i}\sum_{i=1}^{n}\lambda_i x_i\right) \le \frac{1}{\sum_{i=1}^{n}\lambda_i}\sum_{i=1}^{n}\lambda_i f(x_i)$$

Pour f = ln, on a alors :

$$\ln\left(\frac{1}{\sum_{i=1}^{n}\lambda_i}\sum_{i=1}^{n}\lambda_i x_i\right) \le \frac{1}{\sum_{i=1}^{n}\lambda_i}\sum_{i=1}^{n}\lambda_i \ln(x_i)$$

$$\le \frac{1}{\sum_{i=1}^{n}\lambda_i}\sum_{i=1}^{n}\ln(x_i)^{\lambda_i}$$

$$\le \frac{1}{\sum_{i=1}^{n}\lambda_i}\ln\prod_{i=1}^{n}(x_i)^{\lambda_i}$$

Enfin on a: $$\ln\left(\frac{1}{\sum_{i=1}^{n}\lambda_i}\sum_{i=1}^{n}\lambda_i x_i\right) \le \ln\left[\prod_{i=1}^{n}(x_i)^{\lambda_i}\right]^{1/\sum_{i=1}^{n}\lambda_i}$$

Qui donne : $$\boxed{\frac{1}{\sum_{i=1}^{n}\lambda_i}\sum_{i=1}^{n}\lambda_i x_i \le \left[\prod_{i=1}^{n}(x_i)^{\lambda_i}\right]^{1/\sum_{i=1}^{n}\lambda_i}}$$ C.Q.F.D

Bibiographies

[Alg66] V. Algazi, "Useful, Approximations to optimum quantization", IEEE Trans. *Commun. Techol.*, vol COM-14, pp 297-301, 1966.

[Bar82] T. Barnwell, "Subband Coder design of quadrature filters and optimum ADPCM Coders", *IEEE Trans. Acoust. Speech*, Signal Processing, vol ASSP-30 pp 751-765, oct. 82.

[Bat87] G. Battle, " A block spin construction of ondelettes, Part 1 : Lemarier functions," Commun. Math. Phys., vol. 110, pp 601-615, 1987.

[Bur81] P. Burt, Fast Filter Transforms for Image Processing" *Computer Graphics and Image Proc.*, n°16, pp 20-51, 1981.

[Bur83a] P. Burt & E. Adelson "A multirésolution System with Application , Image Mosaics"*ACM trans. on Graphics*, vol. 2, n° 4, october 1983, pp.217-236.

[Bur83b] P.Burt & E. Adelson, "The Laplacien pyramid as a compact image Code", *IEEE Trans. Comm. 31*, 1983, pp. 482-540.

[Coh92] A. Cohen, I. Daubechies, J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets,", Comm. on Pure and Applied Math. Vol. XLV, pp. 485-560, 1992.

[Cam65] F. Compbell & J. Kulikowski, "Orientation Selectivity of the human Visual System", *J. Physiol*, vol. 181, pp. 576-593, 1965.

[Cro76a] R.E. Crochiere, S. A. Webber, and J. L. Flanagan, "Digital coding of speech insub-bands coding," *Bell Syst. Tech. J.*, vol. 55, pp. 1069-1085, Oct. 1976.

[Cro76b] R.E. Crochiere, , "Digital coding of speech in sub-bands coding," *Bell Syst. Tech. J.*, vol. 55, pp. 1069-1085, Oct. 1976.

[Cro81] R.E. Crochiere, "Digital signal processor: Sub-band coding," *Bell Syst. Tech. J.*, vol. 60, pp. 1633-1653, Sept 1981.

[Croi76a] A. Croisier, P. Esteban and C. Galand, "Perfect Channel Splitting by use of interpolation/Decimation/Tree Decomposition Techniques", *Proc. of the Int. Conf. on Informtion Science and System*, Patras, Greece, pp. 443-446, August 1976.

[Dau86] I. Daubechies, A. Grossman, and Y. Meyer, "Painless non-orthogonal expansions," J. Math. Phys. 27, 1986, pp. 1271-1283.

[Dau88] I. Daubechies, "Orthonormal bases of compactly supported wavelets", Comm. Pure Appl. Math. 41, 1988, pp. 909-996.

[Dau90] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," IEEE Trans. on Information Theory, vol. 36, no. 5, pp. 961-1005, sept 1990

[Est77] D. Esteban and C. Galand, "Application of quadrature mirror filters to split band voice coding schemes," in *Proc. 1977IEEE Int. Conf. Acoust., Speech, Signal Processing,* Hartford, CT, May 1977, pp. 191-195.

[Fea90] J.C. Feauveau "Analyse Multiresolution pour les images avec un facteur de resolution $\sqrt{2}$, *Traitement du Signal,* vol. 7 n°2 1990 pp; 117-128.

[Foo82] S. W. Fool and L. F. Turner, "Design of nonrecursive quadrature mirror filters," *Proc. IEEE,* vol. 129, part G, pp. 61-67, June 1982.

[Gab46] D. Gabor, "Theory of communication," *J. IEE* (London), 93, (1946), 429-457.

[Gis68] H. Gish, and J. Pierce, "Asymptot cally Efficient Quantizatif", *IEEE trans. Theory,* IT - 14, pp. 676-683, sept 68.

[Gro84] A. Grossmann, and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape," *SIAMP J. Math. Anal.* 15 1984, pp. 723-736.

[Jai83] V. K. Jain and R. E. Crochiere, "A novel approch to the design of analysis/synthesis filter banks," in *Proc. 1983IEEE Int. Conf. Acoust. Speech, Signal Processing,* Boston, MA, Apr. 1983, pp. 228-231.

[Joh80] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *Proc. 1980IEEE Int. Conf. Acoust. Speech, Signal Processing,* Denover, CO, Apr. 1980, pp. 291.

[Lap94] A. Laprevot Brevet TDF-C2R

[LeG88] D. LeGall and A. Tabatabai, "Subband coding of digital images using symmetric short kernel filters and arithmétic coding techniques," *In International Conference on Acoustic, Speech and Signal Processing,* pp. 761-764, 1988.

[Lem86] P. G. Lemarie and Y. Meyer, "Ondelettes et bases Hilbertiennes", *Rev. Iberoamerica,* vol. 2, pp. 1-18, 1986. "

[Lem88] P. G. Lemarie. "Ondelettes a localisation exponentielles," J. Math. Pures and Appl. No.67, 1988, pp. 227-236.

[Lim77] J.O. Limb, C.B. Rubinstein and J. E. Thompson, "Digital coding of color video signals-A review", IEEE Trans. on Communications, vol. COM-25, pp.1349-1385, November 1977.

[Lin80] Y Linde, A. Buzo, R.M. Gray, "An Algorithm for vector Quantizer Design", *IEEE Trans. on Comm.*, vol; COM-28, n°1, pp 84-95, jan. 1980.

[Lip87] R.P. Lippmann, "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, APR. 1987.

[Llo82] S. Lloyd, " Least Square Quantization in PCM", *IEEE TRans Inf. theory*, IT-28 pp. 129-137, mars 82.

[Mar82a] D. Marr, Vision Nex-York : *Freeman*, , 1982.

[Mar82b] D. Marr and T. Poggio, "Atheory of human sterio vision", *Proc Royal Soc.* London, vol. B 204, pp. 30001-328, 1979.

[Mal89a] S. Mallat "A theory for Multiresolution Signal Decomposition : the wawelet Representation", *IEEE Trans. on Patern Analysis and Machine Intelligence.* Vol. II, n° 7 July 1989 pp. 674-693.

[Mal89b] S. Mallat "Multiresolution Approximation and Wavelet Ortho normal bases of L^2 (R)" T*rans. of the American Math. Society*. Vol. 315 n°1. Sep 89 pp 69-87.

[Max60] J. Max, "Quantizing for minim distorsion" , *RE Trans inform. TH.*vol. IT-60 mars 60 pp. 7-12..

[Mee87] P. Meer, E. S. Baugher, A. Rosenfeld, "Frequency domain analysis and synthesis of image pyramid generating kernels," IEEE Trans. on Pattern Analysis and Machine Intelligence. Vol. PAMI-9, No 4, July 1987.

[Mey86] Y. Meyer, "Principe d'incertitude, bases hilbertiennes et algèbre d'opérateur," *Bourbaki seminar*, no. 662, 1985-1986.

[Mey88] Y. Meyer, "Ondelettes et opérateurs" Herman, 1988

[Nas88a] N.M. Nasrabadi, R.A. King, "Image Coding Vector Quantization : A review", *IEEE Trans on Comm.* COM-36: 99957-971, August 1988.

[Nas88b] N.M. Nasrabadi and Feug, "Vector Quantization of Image Based Upon the Kohonen Self Orgnizing Festure Maps", *IEEE International Conference on Neural Network*, San Diego California, july 24-27, 1988, pp 101-107.

[Nit65] K. Nitadori "Statistical Analysis of DPCM", *Electron Com Japan*, vol 48, février 61965 pp 17-26.

[Koh82a] T. Kohonen , "Self Organized Formation of Topologically Correct Feature Maps", Biol. *Cybern.*, n°43, pp 59-89, 1982.

[Koh82b] T. Kohonen, "Analysis of simple self-organizing Process", *Biol. Cybern*, n°44, pp 135-140, 1982.

[Koh88] T. Kohonen "Self Organization and Associative Memory", *Springer-Verlag* 1988.

[Raf93a] A.Rafaa, J. Baïna and A. Tosser "Adapted Coding in Transform Domain with Scalar or Vector Quantization", *Eight Workshop on Image and Multimensional Signal Processing, IEEE Signal Processing Society Canne*, sept 1993.

[Raf93b] A. Rafaa et D. LeGoff , "Dispositif de sur-échantillonage réversible et de sous-échantionnage d'échantillons numérisés d'un signal", Brevet déposé à TDF-C2R le 25 juin 1993. N° d'enregistrement national 93 12811.

[Raf93c] A. Rafaa et J. Baina, "Procédé et Dispositif de Focalisation Automatique d'un objectif de Caméra", Brevet déposé à TDF-C2R le 27 Octobre 1993. N° d'enregistrement national 93 07777

[Raf94] A. Rafaa, P. Bourcet, A. Tosser "Codage des coefficients d'ondelettes d'images", colloque TOM, INSA Lyon, mars 94.

[Ram86] T. A. Ramstadt, "Considerations on quantization and dynamic bit-allocation in sub_band coders," *in Proc. ICASSP 86*, Tokyo, Japan, pp. 841-844.

[Rum87] D.E. Rumelhart, J. Mc Clelland, and the PDP Research Group, "Parallel Distributed Processing", Exploration in the Microstructure of Cognition vol. 1 : Foundation, MIT Press, Cambridge, Massachussets, 1987.

[Smi86] M. J. T. Smith and T.P. Barnwell, "Exact reconstruction techniques for tree-structured subband coders," in *IEEE Trans. on Acoust. Speech, Signal Processing*, vol. ASSP-34, N° 3 June 1986.

[Smi87] M. Smith and D. Barnwell, "A New Filter Bank Theory for time frequency Representation", *IEEE Trans. ASSP 35*, 1987, pp 314-326.

[Vet84] M.Vetterli, "Multi-dimensional sub-band coding: Some théory and algorithms," *Sig. Processing*, vol. 6, pp. 97-112, Apr. 1984.

[Vet86] M.Vetterli, "Filter Banks allowing perfect reconstruction," *Signal Processing*, vol. 10, No. 3, pp. 219-244, App 1986.

[Vet92] M. Vetterli and C. Herley, "Wavelets and filter bancks: Theory and design," *IEEE Trans.* ASSP, Sept. 92.

[Vet93] M. Vetterli and C. Harley, "Wavelet and filter bancks : theory and design", *IEEE Trans., on Signal. Proc.*, vol. 41 n°8, august 1993 pp 2536-2556.

[Woo86a] J. W. Woods and S. D. O'Neil, "Sub-band coding of images," presented at Proc. ICASSP, Tokyo, Japan, Apr. 1986.

[Woo86b] J. W. Woods and S. D. O'Neil, "Sub-band coding of images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1278-1288, oct. 1986.

Résumé

La pyramide d'ondelettes est une nouvelle forme de représentation de l'image. Son codage offre pa conséquent une nouvelle méthode de compression des échantillons numériques de l'image. Cette technique es en mesure de supplanter les méthodes traditionnelles et bien établies des transformées blocs qui ont domin l'état de l'art de ces dernières années.

Dans un but de transparence, ce mémoire est organisé en trois parties. La première fait une synthès théorique ainsi que "l'historique" de la représentation en multirésolution qui aboutit à la représentation en pyramide d'ondelettes de l'image. La deuxième partie englobe l'aspect scalaire des différentes composantes de la pyramide: caractéristiques statistiques, codage par quantification scalaire ou par zone morte, associé à une allocation de bits optimale par voie analytique ou algorithmique.

Enfin, la troisième partie décrit l'aspect vectoriel du codage. La pyramide est dans ce cas répartie en entités élémentaires, sous forme d'arbre d'éléments vectoriels auxquelles un codage hybride à base de la Q.S. et la Q.V. leur a été adapté. Cette structure a permis en outre la description de l'activité fréquentielle et directionnelle locale dans l'image puis l'introduction de la notion de classification des ces activités. Pour étendre la validité du codeur élaboré, des extensions aux images couleurs et aux séquences d'images ont aussi été présentées.

Abstract

Wavelet pyramidal is a new form of image representation. Its coding scheme allows a new method of image data compression. This technical may well supplant the well established block-transform methods which have been state of art for these last years.

In a suitable purpose, this work is achieved in three parts. The first part provides theorical and historical synthesis of multiresolution representation which converge into Wavelets pyramidal representation of images. The second part investigate scalar point of view of different pyramid components : statistical characteristics, coding scheme with scalar quantization or with dead-zone, related to optimal bit allocation whether analytical or algorithmic.

Finally, the third part deals with vector coding scheme. The pyramid is then dispatched into elementary entities which are performed on a hierarchical vector tree as which we applied an hybrid coding based on QS and Q.V. This structure allows local directivity and frequency activity description in image, and then an introduction to the notion of activity classification. To extend the validity of the elaborated encoder, extensions to colours images and to images sequences was presented too.