Compression multimodale du signal et de l'image en utilisant un seul codeur

Emre Zeybek

▶ **To cite this version:**

Emre Zeybek. Compression multimodale du signal et de l'image en utilisant un seul codeur. Autre. Université Paris-Est, 2011. Français. NNT : 2011PEST1060 . tel-00665757

HAL Id: tel-00665757

https://theses.hal.science/tel-00665757

Submitted on 2 Feb 2012

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

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

UNIVERSITÉ PARIS-EST

THESE DE DOCTORAT

Ecole Doctorale : MSTIC

par

Emre ZEYBEK

pour obtenir le grade de

DOCTEUR EN TRAITEMENT DU SIGNAL ET DE L'IMAGE

Spécialité : **SCIENCES DE L'INGENIEUR**

Equipe d'accueil: Laboratoire Images Signaux et Systèmes Intelligents (LiSSi, E.A. 3956)

Directeur de la thèse : **Amine Naït-Ali**

Co-encadreur de la thèse : **Régis Fournier**

Sujet de la thèse :

Compression multimodale du signal et de l'image en utilisant un seul codeur

---

Soutenue publiquement le .24 /03/2011 devant la commission d'examen composée de :

Amine Naït-Ali	Professeur	Université Paris Est	Directeur de Thèse
Christian Olivier	Professeur	Université de Poitiers	Rapporteur
Azzedine Beghdadi	Professeur	Université Paris 13	Rapporteur
Jacques Lemoine	Professeur	Université Paris Est	Examinateur
Jean Claude Nunes	Maître de conférences	Université Rennes 1	Examinateur
Régis Fournier	Maître de conférences	Université Paris Est	Examinateur

Résumé

Cette thèse a pour objectif d'étudier et d'analyser une nouvelle stratégie de compression, dont le principe consiste à compresser conjointement des données issues de plusieurs modalités, en utilisant un codeur unique. Cette approche est appelée « Compression Multimodale ». Dans ce contexte, une image et un signal audio peuvent être compressés conjointement et uniquement par un codeur d'image (e.g. un standard), sans la nécessité d'intégrer un codec audio. L'idée de base développée dans cette thèse consiste à insérer les échantillons d'un signal en remplacement de certains pixels de l'image « porteuse » tout en préservant la qualité de l'information après le processus de codage et de décodage. Cette technique ne doit pas être confondue aux techniques de tatouage ou de stéganographie puisqu'il ne s'agit pas de dissimuler une information dans une autre. En Compression Multimodale, l'objectif majeur est, d'une part, l'amélioration des performances de la compression en termes de débit-distorsion et d'autre part, l'optimisation de l'utilisation des ressources matérielles d'un système embarqué donné (e.g. accélération du temps d'encodage/décodage). Tout au long de ce rapport, nous allons étudier et analyser des variantes de la Compression Multimodale dont le noyau consiste à élaborer des fonctions de mélange et de séparation, en amont du codage et de séparation. Une validation est effectuée sur des images et des signaux usuels ainsi que sur des données spécifiques telles que les images et signaux biomédicaux. Ce travail sera conclu par une extension vers la vidéo de la stratégie de la Compression Multimodale.

Mots clés : compression multimodale, JPEG 2000, H. 264, MPEG, décomposition d'ondelette, spline interpolation, quadtree.

Abstract

The objective of this thesis is to study and analyze a new compression strategy, whose principle is to compress the data together from multiple modalities by using a single encoder. This approach is called “Multimodal Compression” during which, an image and an audio signal is compressed together by a single image encoder (e.g. a standard), without the need for an integrating audio codec. The basic idea developed in this thesis is to insert samples of a signal by replacing some pixels of the "carrier’s image” while preserving the quality of information after the process of encoding and decoding. This technique should not be confused with techniques like watermarking or stéganographie, since Multimodal Compression does not conceal any information with another. Two main objectives of Multimodal Compression are to improve the compression performance in terms of rate-distortion and to optimize the use of material resources of a given embedded system (e.g. acceleration of encoding/decoding time). In this report we study and analyze the variations of Multimodal Compression whose core function is to develop mixing and separation prior to coding and separation. Images and common signals as well as specific data such as biomedical images and signals are validated. This work is concluded by discussing the video of the strategy of Multimodal Compression.

Keywords: multimodal compression, JPEG 2000, H. 264, MPEG, wavelet decomposition, spline interpolation, quadtree.

Table des Matières

Introduction

La numérisation des données du type signal, image et vidéo est devenue une procédure technique de plus en plus employée dans les systèmes d'information récents. Cette numérisation a pour objectif de faciliter, entre autres, leur analyse, leur traitement et leur transmission, en adéquation avec des architectures matériels spécifiques. Parmi les applications qui se déploient à grande vitesse, la transmission de l'information à travers des réseaux informatiques, y compris les réseaux de téléphonie mobile, ont connu durant cette dernière décennie, une offre et un besoin sans précédent. Ce besoin ne concerne pas spécialement le multimédia ; le domaine médical, notamment dans le cadre de la télémédecine a connu des applications novatrices et très intéressantes.

Dans un tel contexte (i.e. transmission), la vitesse d'acheminement des données, d'un point A vers un point B est souvent limitée par la bande passante du canal de transmission. Pour cette raison, la compression des données est une solution utilisée, presque d'une manière systématique dans les systèmes de transmission actuels. Il est clair, qu'à ce niveau, il faut distinguer entre la compression sans perte et la compression avec perte, voire la compression progressive (i.e. codage/décodage de la moins bonne qualité jusqu'à la meilleure qualité possible). Dans le domaine du multimédia, la compression sans perte est rarement utilisée en raison du faible taux de compression obtenu.

Dans certaines applications, notamment celles qui nécessitent la transmission des données en utilisant des systèmes embarqués (i.e. téléphone portable), plusieurs codecs distincts doivent être implémentés. Par exemple, les téléphones portables intègrent des codecs d'images, de vidéos et de signaux audio. Chaque codec fonctionne d'une manière indépendante et utilise des ressources particulières en occupant, par exemple, un espace mémoire dans le système. En conséquence, ce mode de fonctionnement a tendance d'augmenter la consommation d'énergie du système de transmission.

Cette thèse a pour objectif d'étudier et d'analyser une nouvelle stratégie de compression dont le principe consiste à compresser conjointement des données issues de plusieurs modalités, en utilisant un codeur unique. Nous

appelons cette stratégie « Compression Multimodale ». Dans ce contexte, une image et un signal audio peuvent être compressés conjointement et uniquement par un codeur d'image (e.g. un standard), sans avoir besoin d'utiliser un codec audio. L'idée de base consiste à insérer les échantillons d'un signal en remplacement de certains pixels de l'image « porteuse » sans pour autant dégrader la qualité de l'information après le processus de codage et de décodage. Cette technique ne doit pas être confondue aux techniques de tatouage ou de stéganographie puisqu'il ne s'agit pas de dissimuler une information dans un autre. En Compression Multimodale, l'objectif majeur est d'une part, l'amélioration des performances de la compression en termes de débit-distorsion et d'autre part, l'optimisation de l'utilisation des ressources matérielles d'un système embarqué donné.

Tout au long de ce rapport, nous allons étudier et analyser des variantes de la Compression Multimodale dont le noyau consiste à élaborer des fonctions : (1) de mélange en amont du codage et (2) de séparation, en amont du décodage.

Ce travail de thèse a été préparé au sein du *Laboratoire Images Signaux et Systèmes Intelligents* (LiSSi, E.A. 3956) de l'Université Paris-Est Créteil (UPEC). Une des deux équipes de ce laboratoire oriente ses activités vers le traitement du signal et de l'image appliqués aux domaines du Génie Biologique et du Génie Médical. Ce travail a été effectué sous la direction du Professeur Amine Naït-ali, directeur adjoint de l'équipe TIS (Traitement des Images et des Signaux). Elle a été co-encadrée par Régis Fournier, Maître de Conférences et membre de cette même équipe.

Ce rapport de thèse s'articule autour de 4 chapitres. Il est structuré comme suit :

Dans le premier chapitre, nous présentons les techniques de base les plus utilisées en compression des données numériques. L'accent est mis sur les différentes techniques de réduction du débit binaire, de redondance spatiale et temporelle.

Le deuxième chapitre porte sur la notion de la Compression Multimodale. Dans ce chapitre, nous présentons dans un premier temps, les techniques principales de dissimulation de l'information, notamment, le tatouage et la stéganographie. L'objectif consiste à mettre en exergue la frontière entre ces approches et la Compression Multimodale.

Plusieurs variantes de cette méthode seront analysées pour compresser conjointement une image et un signal, voire un ensemble de signaux par un standard du type JPEG 2000. Entre autres, on s'intéresse aux techniques d'insertion dans le domaine spatial et fréquentiel (par décomposition en ondelettes), selon des stratégies non supervisées. On s'intéressera également à la stratégie d'insertion supervisée par décomposition en « quadtree » dans le domaine spatial. Cette technique prend en considération la nature même de l'image, rendant ainsi la phase de compression (par un codeur donné), particulièrement efficace.

Le troisième chapitre est consacré à la présentation et à l'analyse des résultats des méthodes décrites dans le chapitre précédent. L'évaluation a été effectuée sur des images et des signaux de caractéristiques différentes y

compris sur des données médicales. Les performances de la Compression Multimodale seront mises en évidence à travers des courbes « débit-distorsion ».

Le quatrième chapitre est consacré aux perspectives de notre travail. Il s'agit en effet d'une extension de la Compression Multimodale pour coder une séquence d'images (i.e. une vidéo) y compris sa bande audio, à travers le standard H.264. Dans ce chapitre, nous décrirons, dans un premier temps, les problèmes liés à la synchronisation des trames de la vidéo par rapport au signal audio. Nous discuterons ensuite l'adaptation de la méthode non-supervisée dans le domaine spatial des séquences d'images. Ce chapitre sera donc une ouverture intéressante mettant en évidence le potentiel de la Compression Multimodale.

Ce rapport se termine par un chapitre que l'on consacre à la conclusion générale dans laquelle nous récapitulerons nos contributions et analyserons les résultats obtenus.

Chapitre 1

Un aperçu de la compression des données

1.1 Introduction

Durant ces dernières années, le problème de la compression a tant retenu l'attention des chercheurs que l'on ne compte plus le nombre de publications dans ce domaine. Par conséquent, plusieurs méthodes et algorithmes ont été intégrés dans des standards de compression ; entre autres, en compression de l'image et de la vidéo.

L'objectif de ce chapitre ne consiste pas à passer en revue toutes les techniques existantes en compression de données, mais néanmoins, on souhaite que le lecteur ait un aperçu des algorithmes et des techniques de compression, les plus récurrents.

Comme il est bien connu, lorsque l'on évoque la compression, cela peut laisser entendre, soit la compression sans perte d'information (on parlera dans ce cas de la compression réversible), soit la compression avec perte d'information (il s'agit de la compression irréversible).

En compression sans perte, l'information décodée est l'image parfaite de l'originale. Son inconvénient est son faible taux de compression, notamment lorsqu'il est question de compression des images ou des signaux. Par ailleurs, en compression avec perte, des taux de compression élevés peuvent être obtenus au détriment d'une dégradation, parfois non-perceptible par l'humain, si l'on contrôle de manière optimale le débit binaire.

Dans ce chapitre, nous présenterons : (1) quelques méthodes utilisées pour la compression sans perte des données (section 1.2), (2) la base de la compression d'images (section 1.3), (3) la base de la compression vidéo (section 1.4). On évoquera tout au long de ce chapitre, les standards les plus utilisés dans les systèmes d'information.

1.2 Techniques de compression sans perte

1.2.1 L'entropie de l'information

L'entropie est la mesure du désordre ou la mesure de l'imprévisibilité. Dans les systèmes informatiques, le degré de l'imprévisibilité d'un message peut être utilisé comme une mesure de l'information véhiculée par le message.

Le concept important dans la définition de l'entropie est celui de la prévisibilité. Un message parfaitement prévisible ne véhicule aucune information. En 1948, Shannon a défini l'information véhiculée par un événement *I(E)*, mesurée en bits ; en terme de probabilité de cette événement *p(E) :*

$$I(E) = \log_2\left(\frac{1}{p(E)}\right) \qquad (1.1)$$

Un événement qui est complètement prédictible ($p=1$) n'a aucune information ($log_2(1)=0$) .

Dans la théorie du codage de source, une source discrète est un dispositif qui fournit aléatoirement des séquences de symboles issus d'un ensemble discret fini. On modélise une source par un ensemble de variables aléatoires dont les valeurs proviennent d'un alphabet de taille fini $\Omega = \{x_0, x_1, ..., x_N\}$. Ici Ω est appelé l'ensemble des symboles de source. De plus, une source est dite sans mémoire si la séquence de symboles générée par la source est une suite de variables indépendantes et identiquement distribuées. Une source de ce type est appelée « une source discrète sans mémoire » (*Discrete Memoryless Source* - DMS *en anglais* [**1**].

A titre d'exemple, il est possible de créer une source DMS qui génère les lettres de 'a' jusqu'à 'd', tel que la lettre 'a' est 7 fois plus probable que les lettres 'b', 'c' et 'd'. Pour une telle source, on peut montrer la probabilité et par conséquent la quantité d'information de chaque symbole possible est parfaitement déterminée (voir Tableau I).

TABLEAU I
L'ENTROPIE ET LA PROBABILITE DES SYMBOLES D'UNE SOURCE DISCRETE SANS MEMOIRE

E	p(E)	I(E)
a	0,7	0,515
b	0,1	3,322
c	0,1	3,322
d	0,1	3,322

L'entropie d'une source d'information telle une DMS, est définie comme la quantité de l'information moyenne transmise par chaque symbole fourni par la source. Donc l'entropie *H(S)* d'une source comportant *n* symboles, de s_1 à s_n, avec une probabilité $p(s_i)$ pour chaque symbole peut être définie comme suit :

$$H(S) = p(s_1)I(s_1) + p(s_2)I(s_2) + ... + p(s_n)I(s_n)$$

$$H(S) = \sum_{i=1}^{n} p(s_i) I(s_i)$$

$$H(S) = \sum_{i=1}^{n} p(s_i) \log_2 \left(\frac{1}{p(s_i)} \right) \qquad (1.2)$$

Selon le théorème de codage de source sans bruit de Shannon [**2**], si l'on souhaite coder une source le plus efficacement possible en n'y introduisant aucune ambigüité, le nombre moyen de bits par symbole utilisé par le code doit être au moins égale à l'entropie de la source [**2**].

1.2.2 Le codage Huffmann

En 1952, Huffmann a mis en évidence une méthode pour construire des codes compacts de longueur variables [**2**].Le codage Huffmann attribue un code de sortie à chaque symbole. Les codes de sortie pouvant être aussi court qu'un bit, ou beaucoup plus longs que les symboles d'entrée en fonction de leurs probabilités. Le nombre de bits optimum pour chaque symbole est $-\log_2 p_i$, où p_i est la probabilité du symbole en question.

Les étapes nécessaires pour générer le code Huffmann pour des symboles dont les probabilités d'occurrence sont connues peuvent être regroupées comme suit :

- Classer tous les symboles dans l'ordre de leur probabilité d'occurrence,
- Fusionner les deux symboles les moins probables pour former un symbole combiné, et reclasser encore les symboles dans l'ordre de probabilité : Cela génère une arborescence dont chaque nœud représente la probabilité cumulative de tous les nœuds en dessous,
- Tracer un chemin vers chaque feuille, en notant la direction sur chaque nœud.

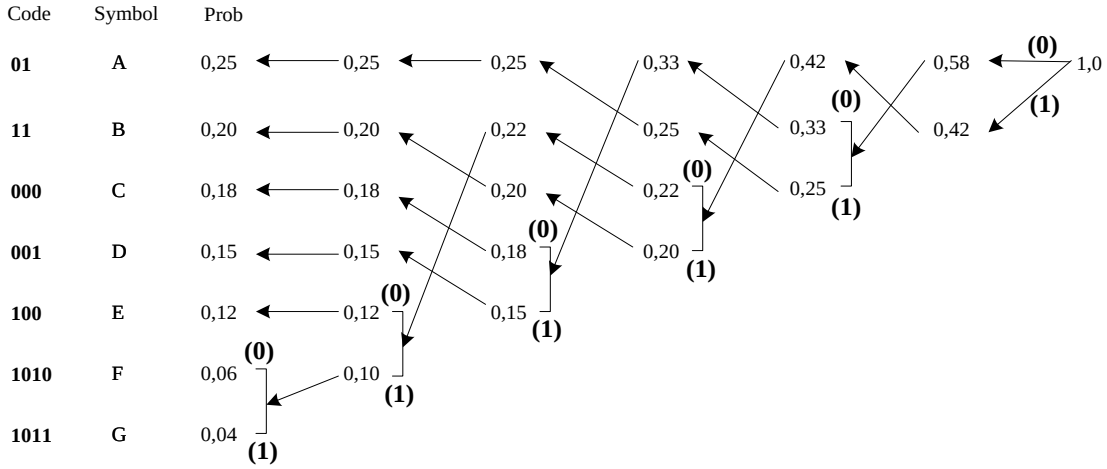


Figure 1-1 Un exemple du code Huffmann pour sept symboles.

La Figure 1-4 montre un exemple du codage Huffmann pour sept symboles (A-G). Les probabilités des symboles dans l'ordre décroissant sont présentées dans la troisième colonne. Dans la colonne suivante, les deux

éléments de plus faible probabilité sont ajoutés afin de produire une probabilité combinée. Cette probabilité combinée est utilisée dans le nouveau classement [**2**].

Cette procédure continue jusqu'à la dernière colonne où la probabilité de « 1 » est atteinte. A partir de la dernière colonne, chaque branche de probabilité de l'arborescence, est marquée d'un « 0 » au dessus, et d'un « 1 » au dessous. Ceci est illustré sur la Figure 1-1 par des chiffres en gras.

Le codage qui correspond au symbole est donc lu sur l'arborescence de la droite vers la gauche en suivant la séquence. Le nombre moyen de bits est calculé:

$$0,25\times 2+0,20\times 2+0,18\times 3+0,15\times 3$$
$$+0,12\times 3+0,06\times 4+0,04\times 4=2,65 \text{ bits}$$

Ce qui est très proche de l'entropie calculée par :

$$-\begin{pmatrix} 0,25\cdot\log_2(0,25)+0,2\cdot\log_2(0,2)+0,18\cdot\log_2(0,18)+0,15\cdot\log(0,15) \\ +0,12\cdot\log_2(0,12)+0,06\cdot\log_2(0,06)+0,04\cdot\log_2(0,04) \end{pmatrix}=2,62 \text{ bits}$$

Cependant, le fait que les mots de code attribués doivent se composer d'un nombre entier de bits, rend le codage Huffmann sous-optimal [**3**]. Par exemple, si la probabilité d'un symbole est 0,33, le nombre optimal de bits pour coder ce symbole est d'environs 1,6 bits. Mais le schéma de codage Huffmann doit assigner soit 1 bit, soit 2 bits pour le code. Dans les deux cas, en moyenne, ceci mène à un codage avec plus de bits par rapport à son entropie.

Pour des symboles de haute probabilité, le codage Huffmann est non optimal. Par exemple, pour un symbole d'une probabilité de 0,9, la taille optimale du code est 0,15 bits, mais le codage Huffmann doit attribuer une valeur intégrale de bits. Cela oblige le choix de 1 bit pour la taille du symbole, qui est 6 fois plus grande que nécessaire.

Il convient de noter que le codage Huffmann des grands nombres de symboles peut finir par une chaine binaire très longue pour des valeurs de faibles occurrences. Cela rend le codage Huffmann inexploitable. Dans un tel cas, un groupe de symboles est représenté par ces probabilités totales. Puis, il est codé par la méthode Huffmann. Cette technique est appelée « le codage Huffmann modifié » (Modified Huffmann Coding). Cette technique est utilisée dans le codage JPEG. Une autre méthode est utilisée dans le H.261 et MPEG : le Huffmann à deux dimensions. Il existe également Huffmann à trois dimensions que l'on utilise dans le H.263 [**4**] [**5**].

1.2.3 Le codage arithmétique

Le codage Huffmann peut être optimal si la probabilité de symbole est une puissance entière de ½, ce qui n'est généralement pas le cas. Le codage arithmétique est une technique de compression qui code les données en créant une chaine de code de valeurs fractionnelles comprises entre 0 et 1. Ce codage encourage une séparation claire entre le modèle de représentation pour les données et le codage de l'information par rapport au modèle. Un

autre avantage du codage arithmétique, est que chaque symbole n'a pas besoin d'être représenté par un nombre entier de bits. Par conséquent, il est possible d'aboutir à un codage plus efficace. Le codage arithmétique atteint la limite théorique de l'entropie pour tout type de source.

Il existe deux types de modélisation pour le codage arithmétique : le modèle fixe et le modèle adaptatif. Pour un contexte donné, la modélisation repose sur le calcul de la distribution de probabilité du prochain symbole à coder. Il doit être possible pour le décodeur de produire de manière exacte la même distribution pour le même contexte. Contrairement au codage Huffmann, le codage arithmétique accueille bien les modèles adaptatifs.

Dans le modèle fixe, le codeur et le décodeur connaissent la probabilité attribuée à chaque symbole. Les modèles fixes sont particulièrement efficaces lorsque les caractéristiques de la source sont proches du modèle et présentent des petites fluctuations.

Dans le modèle adaptatif, les probabilités attribuées aux symboles peuvent changer lorsqu'ils sont traités par le codeur, en fonction de la fréquence d'occurrence de chaque symbole. Le codeur attribue un compteur à chaque symbole. Ces compteurs qui peuvent être initialisé à zéro, sont mis à jour par le codeur lorsqu'il « voit » un symbole à l'entrée. Cela permet au codeur d'approximer les fréquences observées des symboles.

1.2.3.1 Les principes du codage arithmétique

L'idée fondamentale du codage arithmétique, est d'utiliser une échelle sur laquelle les intervalles de nombre réels du codage sont représentés entre 0 et 1. En effet, cela correspond à la fonction de la densité de probabilité cumulative de tous les symboles. L'intervalle nécessaire pour représenter un message devient de plus en plus petit lorsque le message devient plus long. Par conséquent, le nombre de bits utilisé pour représenter cet intervalle augmente.

Exemple : Soit $\Omega = \{\mathbf{a}, \mathbf{e}, \mathbf{i}, \mathbf{o}, \mathbf{u}, \mathbf{!}\}$ forme un alphabet avec les probabilités du chaque symbole indiqués dans le Tableau II .

Une fois que l'on connait la probabilité de chaque symbole, on lui attribue une portion de l'intervalle [0,1) qui correspond à la probabilité de l'apparence du symbole dans la fonction de densité cumulative. Il faut aussi noter qu'un caractère situé sur un intervalle de [« inférieur », « supérieur ») représente toutes les valeurs réelles possibles de la valeur inférieure jusqu`à la valeur supérieure non incluse. Par exemple, le symbole **u** de probabilité 0,1, qui est défini dans l'intervalle cumulatif de [0,8, 0,9) peut prendre n'importe quelle valeur de 0,8 à 0,8999999….

Dans le codage arithmétique, le premier symbole codé construit la partie la plus signifiante du codage. Supposons que l'on souhaite coder le message « **eaii !** », le premier symbole à coder est le «**e** », donc le message final codé doit être un nombre supérieur ou égale à 0,2 mais inférieur à 0,5. Par composition des intervalles, le codage du deuxième caractère 'a' limitera l'intervalle de sortie à [0,2, 0,26). Cela continuera de cette manière jusqu'au codage du dernier caractère. Chaque caractère limitera davantage l'intervalle de sortie. Le Tableau III donne un résumé des intervalles du codage de la chaine de caractère « eaii !».

TABLEAU II
EXEMPLE DE CODAGE ARITHMETIQUES:
LE MODEL FIXE

Symbole	Probabilité	Intervalle
a	0,2	[0,0, 0,2)
e	0,3	[0,2, 0,5)
i	0,1	[0,5, 0,6)
o	0,2	[0,6, 0,8)
u	0,1	[0,8, 0,9)
!	0,1	[0,9, 0,1)

TABLEAU III
LA REPRESENTATION
DU PROCESSUS DE CODAGE ARITHMETIQUE

	Nouveau caractère	Intervalle
Initialement		[0, 1)
Après avoir vu un symbole	e	[0,2, 0,5)
	i	[0,2, 0,26)
	o	[0,23, 0,236)
	u	[0,233, 0,2336)
	!	[0, 23354, 0,2336)

La Figure 1-2 montre une autre représentation du processus de codage arithmétique. Sur cette figure, l'intervalle du codage est élargi à chaque étape et il est marqué sur une échelle qui facilite la visualisation des points d'extrémité. L'intervalle finale [0,23354, 0,2336) représente le message. C'est-à-dire que si l'on transmet un nombre dans cet intervalle, cela se traduira au message « eaii ! ».

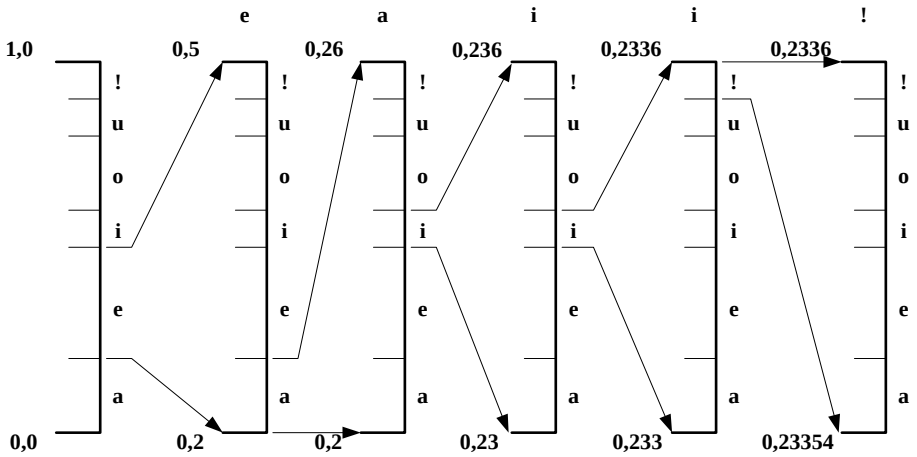


Figure 1-2 La représentation du processus de codage arithmétique avec la mise à l'échelle des intervalles pour chaque étape du message « eaii ! »

Etant donné ce schéma de codage, il est relativement facile à visualiser le décodage du message « eaii ! ». Afin de vérifier cela, on suppose qu'un nombre x=0,23355, dans l'intervalle $0{,}23354 \leq x < 0{,}2336$ est transmis et introduit à l'entrée du décodeur. Le décodeur en utilisant les mêmes intervalles de probabilités que ceux du codeur, exécute une opération similaire. En commençant par l'intervalle [0,1), seul l'intervalle [0,2, 0,5) enveloppe le code transmis, donc le premier symbole doit être « e ». Similairement au processus de codage, les symboles suivants sont maintenant définis dans l'intervalle [0,2, 0,5). Cela revient à redéfinir le code dans l'intervalle [0, 1) et à le compenser par la valeur inférieur puis le mettre en échelle dans son intervalle d'origine. C'est-à-dire, le nouveau code devient (0,233355-0,2) / (0,5-0,2)= 0,11185, ce qui est encadré par l'intervalle [0,0, 0,2) du symbole « a ». Le Tableau IV représente un résumé du processus de décodage.

TABLEAU IV
EXEMPLE DE DECODAGE ARITHMETIQUES:
LE MODEL FIXE

Le nombre codé	Symbole	Intervalle
0,23335	e	[0,2, 0,5)
0,11185	a	[0,0, 0,2)
0,55925	i	[0,5, 0,6)
0,59250	i	[0,5, 0,6)
0,92500	!	[0,9, 1,0)

1.2.4 Le codage prédictif

Le codage prédictif est une méthode simple pour la réduction de redondance. La théorie sous-jacente au codage prédictif consiste à prédire les valeurs des échantillons d'un signal en se basant sur les valeurs *a priori*, et à coder l'erreur de cette prédiction. Cette méthode est appelée « differential pulse code modulation » (DPCM).

La Figure 1-3 illustre le schéma bloc d'un codeur DPCM, où la différence entre les échantillons d'entrée et ceux qui sont prédits, est quantifiée et codée pour la transmission. Au coté du décodeur, le signal d'erreur reçu est ajouté au signal prédit. En l'absence de l'étape de quantification, la compression est sans perte.

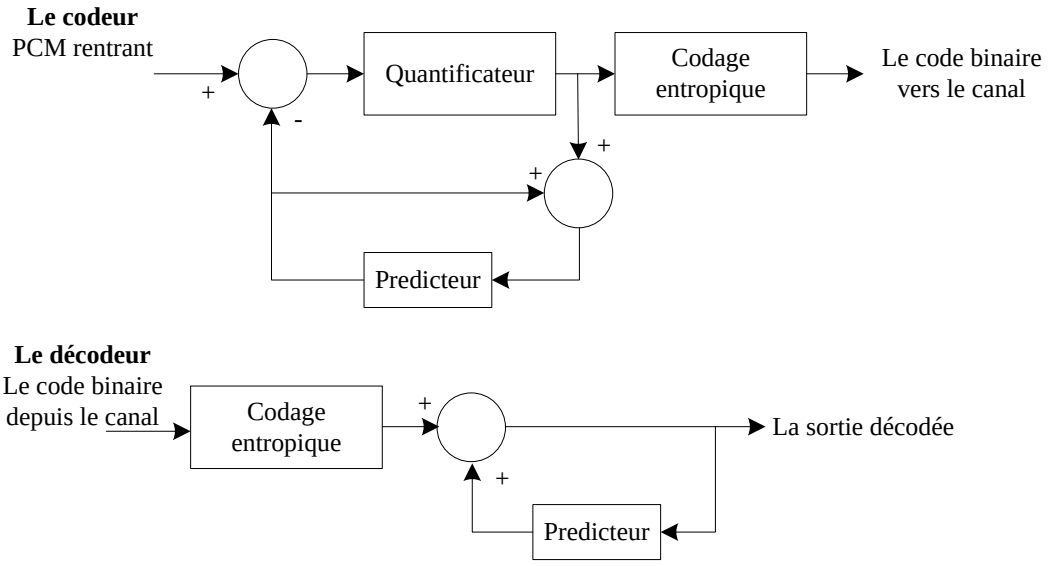


Figure 1-3 Le schéma bloc du codage DPCM

1.3 Compression d'image avec perte

1.3.1 Réduction de la redondance spatiale.

1.3.1.1 La transformation en espace couleur YCbCr

Les images numériques sont généralement représentées sous une forme RGB, mais les composants de cette représentation manifestent une forte corrélation entre eux [**6**]. Certains codeurs d'images (et de vidéo aussi) exige une transformée sur les couleurs RGB avant le codage réel pour représenter l'image dans un domaine qui diminue la corrélation entre les pixels. Cette transformée est appelée « transformée en YCbCr ». La transformée en YCbCr décompose les plans RGB en trois plans orthogonaux, notamment en Luminance (Y) et deux plans de chrominance (Cb & Cr). La Figure 1-4 illustre l'image Lena, décomposée en YCbCr [**6**].

Figure 1-4 Conversion d'une image RGB en YCbCr

Il existe deux types de transformation en YCbCr. Le premier type est appelé « La transformée de couleur réversible » ou RCT[1] et elle est calculée par l'équation suivante :

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 65.481 & 128.533 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \qquad (1.3)$$

L'équation (1.3) convertit un pixel RGB normalisé (défini dans la plage [0,1] $\in \mathbb{R}$) en un pixel YCbCr, où le Y est défini dans plage [16, 235] $\in \mathbb{Z}^+$ et les Cb & Cr sont défini dans [16,240] $\in \mathbb{Z}^+$.

Le deuxième type est appelée « transformée de couleur irréversible » ou ICT[2]. Comme son nom l'indique, les couleurs en RGB obtenues par la transformée inverse ne sont jamais les mêmes que les couleurs de départs. Donc, elle provoque une légère perte sur les valeurs RGB reconstruites. L'équation qui sert à calculer l'ICT est définit comme suit :

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.500 \\ 0.500 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (1.4)$$

1.3.1.2 Le codage par transformation

Le codage par transformation de domaine est principalement utilisé dans la réduction de redondance spatiale sur les images, en « mappant » les pixels dans un domaine de transformée. La puissance de ces techniques réside dans le fait que l'énergie des échantillons d'une image naturelle est concentrée dans la région de basses

[1] RCT-Reversible Colour Transform
[2] ICT-Irreversible Colour Transform.

fréquences ; donc dans une région contenant peu de coefficients. Ces coefficients, par la suite peuvent être quantifiés tout en négligeant les coefficients insignifiants, sans pour autant dégrader la qualité de l'image. Le processus de quantification est pourtant avec perte, dans lequel les valeurs originales ne sont pas préservées.

La transformée en cosinus discrète (DCT) est considérée comme le meilleur choix de transformée pour le codage des images par transformation [**3**], car elle est caractérisée par des vecteurs de base bien définis qui varient doucement. Cela correspond bien aux changements d'intensité de la plupart des images naturelles [**1**]. Un autre aspect important de la DCT, c'est qu'il existe une version rapide pour des applications basées « software » [**7**].

Une DCT en deux dimensions est un processus séparable qui peut être implémenté par deux DCTs unidimensionnelles : une dans la direction des colonnes, l'autre dans la direction des lignes. Pour un bloc de taille $M \times N$ pixels la DCT est définie comme suit :

$$F(u) = \sqrt{\frac{2}{N}} C(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{\pi(2x+1)u}{2N}\right) \qquad u = 0,1,...N\text{-}1 \qquad (1.5)$$

où $C(u) = \sqrt{\frac{1}{2}}$ pour $u = 0$ ou $C(u) = 1$ autrement.

Dans l'équation (1.5) *f(x)* représente l'intensité du *x*-ème pixel, et *F(u)* représente les échantillons de la DCT unidimensionnelle de taille *N*. L'inverse de la transformée est ainsi définie comme suit :

$$f(x) = \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} C(u) F(u) \cos\left(\frac{\pi(2x+1)u}{2N}\right) \qquad x = 0,1,...N\text{-}1 \qquad (1.6)$$

Il est à de noter que le facteur de normalisation $\sqrt{1/N}$ est utilisé pour rendre la transformée orthonormale. Par conséquent, l'énergie dans les deux domaines (i.e. spatial et transformé) est égale. La transformée en deux dimensions est définie comme suit :

$$F(u,v) = \sqrt{\frac{2}{M}} C(v) \sum_{y=0}^{M-1} F(u,y) \cos\left(\frac{\pi(2y+1)v}{2M}\right) \qquad v = 0,1,...M\text{-}1 \qquad (1.7)$$

où *C(v)* est défini de même façon que *C(u)*.

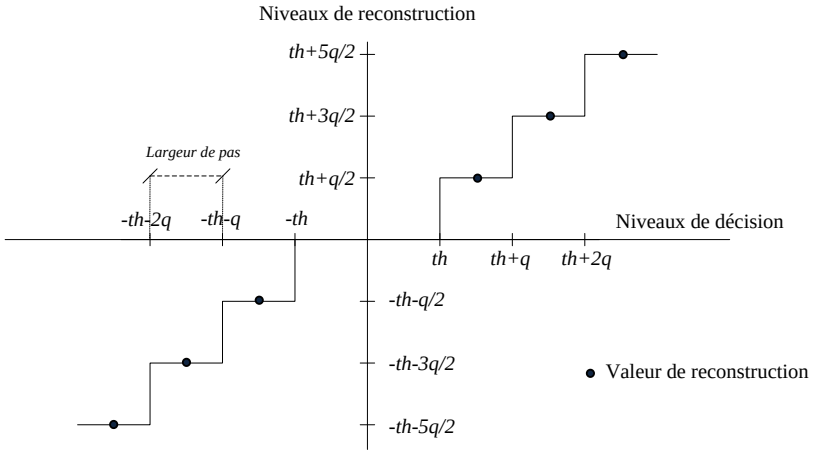


Figure 1-5 Les caractéristiques de la quantification

Ainsi un bloc de $M \times N$ pixels est transformé en un bloc de coefficients de même taille. Le coefficient *F(0,0)* représente la valeur DC[3] du bloc. Le coefficient *F(0,1)* qui correspond à la valeur DC de tous les premiers coefficients unidimensionnels AC[4], représente le premier coefficient AC du bloc dans la direction horizontale. Similairement, le coefficient *F(1,0)* qui correspond à la valeur DC des premiers coefficients unidimensionnels AC, représente le premier coefficient AC du bloc dans la direction verticale, ainsi de suite.

1.3.1.3 Quantification des coefficients de la DCT

En réalité, la transformation des pixels en domaine DCT n'aboutit pas à une compression. Un bloc de taille 64 pixels est transformé en 64 coefficients. En raison de l'orthonormalité de la DCT, l'énergie dans les deux domaines ne change pas, donc il n'y a pas de compression effectuée. Toutefois, la transformation entraine la concentration de l'énergie sur les composantes de basses fréquences. La majorité des coefficients qui en restent, représentent une faible énergie. C'est l'étape de quantification et le codage entropique qui réduisent le débit binaire. De plus, en exploitant les caractéristiques du système visuel humain [**6**] [**8**] qui est moins sensible aux distorsions présentes sur les hautes fréquences, on peut employer une quantification grossière pour les coefficients appartenant aux hautes fréquences.

Le quantificateur utilisé dans tous les codecs standards de l'image et de la vidéo, s'articule autour du « quantificateur à seuil uniforme » (UTQ[5]). Les caractéristiques de ce quantificateur sont montrées à la Figure 1-5. Les deux paramètres importants du UTQ est la valeur du seuil *th* et la taille de pas *q*, et la valeur de reconstruction se trouve aux centroïdes sur les pas (voir la Figure 1-5). Un aspect clé de l'UTQ, c'est que les tailles de pas peuvent être facilement adaptées aux distributions des coefficients AC et DC afin de faciliter le contrôle du débit binaire.

[3] DC – Composant continu
[4] AC – Composant harmonique
[5] UTQ – Uniform Threshold Quantizer

On peut identifier deux autres types d'UTQ dans les standards de codecs d'aujourd'hui, notamment le quantificateur avec zone morte (UTQ-DZ) et le quantificateur sans zone morte (UTQ). Ils sont illustrés sur la Figure 1-6. Le terme « zone morte » couramment se réfère à la région centrale du quantificateur, où les coefficients sont quantifiés à zéro.

L'UTQ est typiquement utilisé dans le codage de la vidéo pour la quantification des coefficients DC intra-trames et l'UTQ-DZ est utilisé pour coder les coefficients AC et DC de la prédiction inter-trames. Le but est d'augmenter davantage le nombre de coefficients non significatifs AC qui seront mis à zéro, donc d'augmenter davantage le taux de compression. Ces deux quantificateurs sont directement liés au quantificateur générique de la Figure 1-5, où le seuil *th* est fixé à 0 dans l'UTQ et à *q/2* dans l'UTQ-DZ pour avoir plus de sortie à valeur nulle (voir Figure 1-6). Ainsi, la taille de la zone morte peut varier de *q* à *2q*.

Dans certaines implémentations (i.e. H.263 et MPEG-4), les niveaux de décision (ou de reconstruction) du l'UTQ-DZ peuvent être décalés de *q/4* ou *q/2*.

Dans la pratique, plutôt que de transmettre les coefficients quantifiés au décodeur, un index de quantification qui représente son rapport à la taille du pas de quantification est transmis.

$$I(u,v) = \left\lfloor \frac{F(u,v)}{q} \right\rfloor \qquad (1.8)$$

La raison pour définir l'indice de quantification est qu'il a une entropie beaucoup plus faible que celle du coefficient quantifié. Au niveau du décodeur ; après l'application d'une étape de quantification inverse, les coefficients sont reconstruits suivant:

$$F^q(u,v) = \left\{ I(u,v) + \tfrac{1}{2} \right\} \times q \qquad (1.9)$$

Pour les codecs standards, le pas de quantification *q* est fixé à 8 pour l'UTQ mais il peut varier entre 2 et 62 pour des tailles paires du pas de quantification dans l'UTQ-DZ. Par conséquent, l'intervalle de définition entier du quantificateur ou le paramètre Q_p qui est le pas de quantification du quantificateur, peut être identifié avec 5 bits. Les quantificateurs uniformes avec ou sans zone morte peuvent être aussi utilisés dans le codage DPCM des pixels.

1.3.2 Le standard JPEG

Le standard JPEG spécifie deux types de codage et de décodage, notamment compression sans perte et avec perte. Le mode de fonctionnement sans perte, est en effet une simple méthode basée sur le codage prédictif.

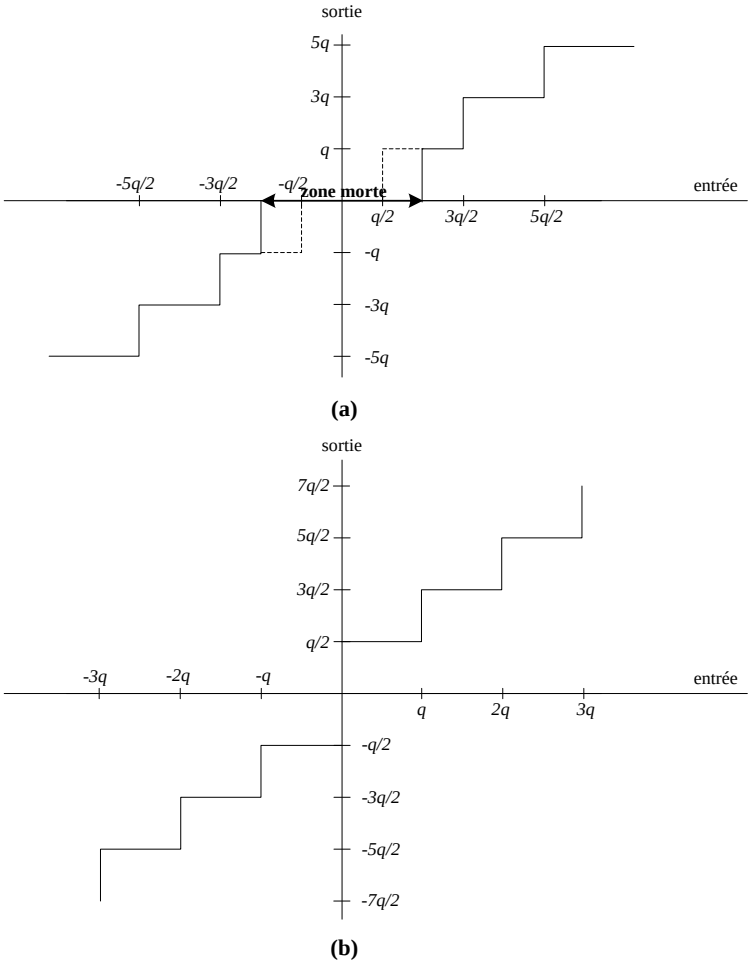


Figure 1-6 Les types de quantificateurs uniformes (a) avec zone morte (b) sans zone morte

1.3.2.1 La compression sans perte avec JPEG Lossless (JPEG – LS)

La compression sans perte dans le standard JPEG est basée sur un simple codage prédictif de la méthode DPCM en utilisant les valeurs des pixels voisins.

La Figure 1-7 décrit les éléments principaux du codeur d'image JPEG-LS. Dans le codage, l'image source numérisée qui peut être représentée en RGB ou en YCbCr, est présentée à l'entrée du prédicteur.

Pour la représentation en YCbCr, l'image peut être représentée sous n'importe quel format de 4 :4 :4 à 4 :1 :0, quel que soit l'amplitude et la taille des échantillons (i.e. 8 bits/pixels). Le prédicteur est de type simple DPCM (voir la Figure 1-3) où chaque pixel de chaque composante de couleur est codé de manière différentielle. La prédiction d'un pixel d'entrée *x* est effectuée à partir des trois pixels voisins de la même composante de couleur. Ce prédicteur est connu sous le nom *Median Edge Detector* (MED). Ceci est illustré sur la Figure 1-8

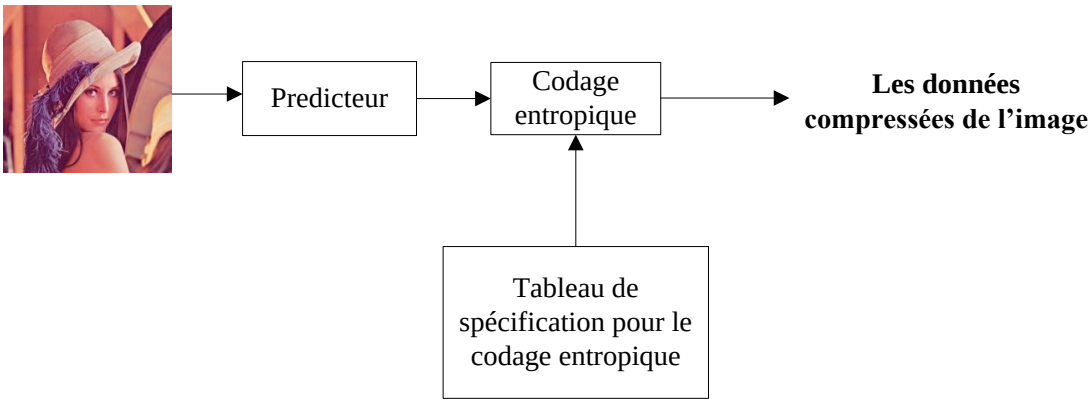


Figure 1-7 Le schéma de bloc du codeur JPEG en mode sans perte (codeur JPEG-LS)

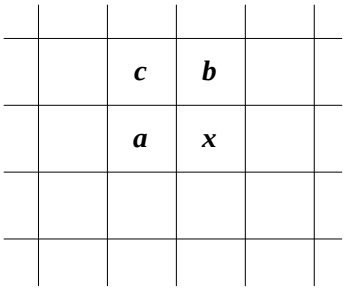


Figure 1-8 La voisinage de la prédiction à trois-échantillons, le modèle utilisé par le prédicteur MED

La prédiction est ensuite soustraite de la valeur réelle du pixel à la position *x*, et la différence est codée en entropie soit par la technique Huffmann soit par le codage arithmétique. Les spécifications du tableau d'entropie déterminent les caractéristiques de la méthode choisie pour le codage entropique.

Le processus de codage peut être légèrement modifié en y introduisant une étape avant le codage sans perte qui réduit par un ou plusieurs bits, la précision des pixels d'entrée. En mode sans perte du standard JPEG, la précision des échantillons est spécifiée entre 2 et 16 bits. Par conséquent, il est possible d'atteindre des taux de compression plus élevés par rapport au mode normal sans perte, mais toutefois cela reste inférieur par rapport à la compression avec perte qui est basée sur la DCT, pour les mêmes *bitrates* et qualité d'image choisis. Il s'agit de compression avec perte. La réduction de la précision des pixels d'entrée par *b* bits ou la quantification des échantillons de différence par un pas de quantificateur de 2^b se compensent.

1.3.2.2 La compression avec perte du JPEG

En plus de la compression sans perte, le JPEG définit trois modes de compression avec perte. Ils sont appelés respectivement « le mode séquentiel baseline », « le mode progressive » et « le mode hiérarchique ». Ces modes sont tous basés sur la DCT pour parvenir à un taux de compression important, tout en préservant la qualité des images reconstruites. La différence principale entre ces modes repose sur la transmission des coefficients DCT.

Dans le mode baseline du JPEG, la valeur du chaque échantillon est incrémentée d'une valeur de $2^{8-1=7} = 128$ avant d'être transformée en DCT, puis l'image est partitionnée dans de blocs non chevauchant de taille 8×8, en suivant un chemin, de droite à gauche et de haut en bas. Chaque bloc est transformé en DCT. Les 64 coefficients sont quantifiés à la qualité désirée. Les coefficients quantifiés sont ensuite codés en entropie puis redirigés vers la sortie. La Figure 1-9 illustre le diagramme de l'algorithme de la compression par JPEG baseline.

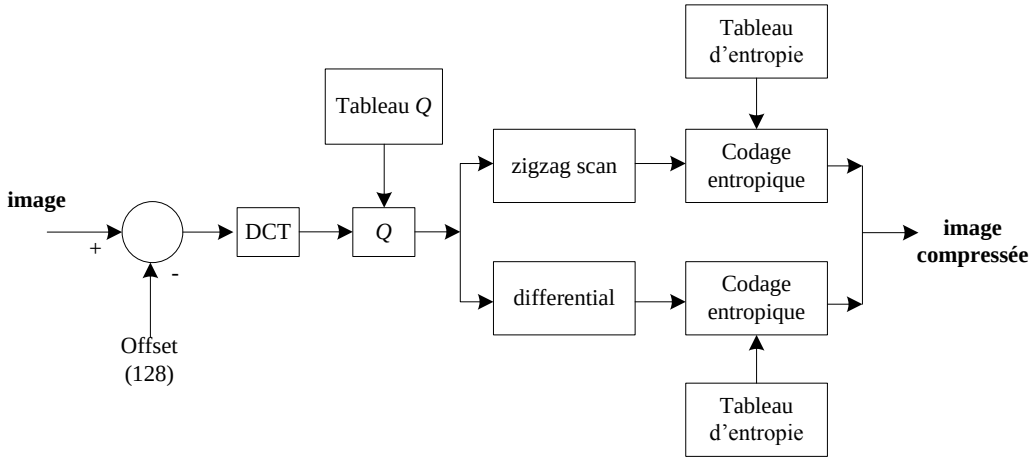


Figure 1-9 Diagramme de bloc du codeur JPEG baseline

Deux exemples de quantification utilisés dans JPEG sont donnés sur le Tableau VI et le Tableau V [**4**]. Un coefficient DCT quantifié $F^q(u,v)$ avec des fréquences spatiales dans la direction horizontale *u* et verticale *v* est donné par:

$$F^q(u,v) = \left\lfloor \frac{F(u,v)}{Q(u,v)} \right\rfloor \qquad (1.10)$$

où $Q(u,v)$ sont des coefficients de luminance et de chrominance donnés dans les Tableau VI et Tableau V et $F(u,v)$ est le coefficient DCT avant quantification. Au niveau du décodeur, les coefficients quantifiés sont reconstruits par quantification inverse :

TABLEAU V
JPEG CHROMINANCE *Q* TABLE

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

TABLEAU VI
JPEG LUMINANCE *Q* TABLE

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

$$F^{Q}(u,v) = F^{q}(u,v) \times Q(u,v) \qquad (1.11)$$

Un facteur de qualité que nous dénotons ici par « *q_JPEG* », est utilisé pour contrôler les éléments de la matrice de quantification *Q(u,v)* [**9**]. Le facteur *q_JPEG* est généralement exprimé en pourcentage sur un intervalle allant de 1 à 100. Les matrices de quantifications présentées sur le Tableau VI et le Tableau V sont utilisées quand *q_JPEG =50*. Pour les autres facteurs de qualité, les éléments de la matrice de quantification *Q(u,v)* , sont multipliés par la facteur de compression α.

$$\alpha = \frac{50}{q\_JPEG} \quad \text{if} \quad 1 \le q\_JPEG \le 50$$
$$\alpha = 2\frac{2 \times q\_JPEG}{100} \quad \text{if} \quad 50 \le q\_JPEG \le 99 \qquad (1.12)$$

En aucun cas, la valeur minimum de la matrice de quantification modifiée ne doit pas être inférieur à $\alpha.Q(u,v) = 1$. Pour la qualité *q_JPEG*=100, la compression est sans perte et toutes les valeurs de *Q(u,v)* sont mises à 1.

Après la quantification, le coefficient DC (le coefficient (0,0) de la transformée) et les 63 coefficients AC sont séparément codés comme le montre la Figure 1-9. Les coefficients DC sont codés en DPCM en utilisant la prédiction du coefficient DC du bloc précédent, i.e. DIFF = DC_i-DC_{i-1}. (Voir Figure 1-10).

L'énergie principale de l'image est concentrée dans les coefficients DC. Les coefficients DC sont codés séparément des coefficients AC pour exploiter davantage ce fait. Les 63 coefficients AC sont codés en RLE suivant un chemin « zigzag » à partir du coefficient AC(0,1) (voir la Figure 1-10).

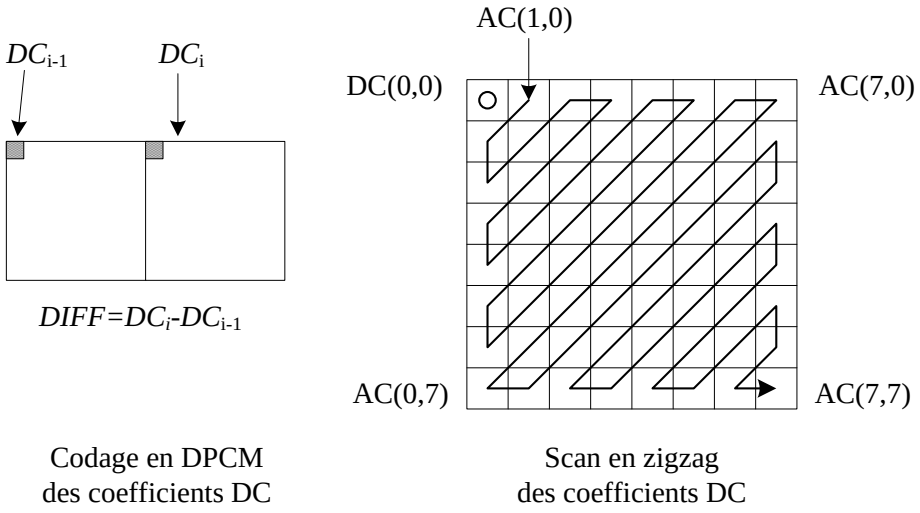


Figure 1-10 La préparation des coefficients DCT pour le codage entropique

Le choix du « pattern » à chemin zigzag permet de faciliter le codage entropique en passant au début par les coefficients les plus probables d'avoir une valeur non nulle. Encore une fois, c'est dû au fait que pour plupart des scènes naturelles, l'énergie de l'image est surtout concentrée dans quelques coefficients des basses fréquences.

1.3.3 Le standard JPEG 2000

Le codeur JPEG2000 suit la même structure générique du codage d'image intra-trame introduit pour le standard JPEG baseline. C'est-à-dire, la décorrélation des pixels dans la trame par une transformée, suivi par la quantification et le codage entropique. Cependant, afin de satisfaire les objectifs de conception du JPEG2000 (ISO 2000), il est nécessaire d'employer une étape de prétraitement sur les pixels et une étape de post-traitement sur les données compressées. La Figure 1-11 illustre un diagramme de bloc du codeur JPEG2000 [**10**].

Figure 1-11 Le diagramme de bloc général du codeur JPEG2000

1.3.3.1 Pré traitement

Les pixels de l'image sont traités *à priori* afin de rendre la réalisation des objectifs de conception du JPEG2000 plus facile. Il existe trois éléments dans l'étape de prétraitement.

- ***Tuilage (Tiling)***
  Le tuilage est l'opération de repartitionner l'image en des blocs non chevauchant. Une tuile est l'unité basic de codage. Toutes les opérations de codage sont appliquées un par un aux tuiles qui sont indépendantes entre elles. Le tuilage est particulièrement important pour réduire l'utilisation de mémoire et de ce fait, il est également possible de traiter et d'accéder à n'importe quelle partie de l'image, d'une manière indépendante.
- ***Décalage de niveau DC***
  Pour chaque tuile, une valeur qui correspond à 2^{B-1}, est soustraite des valeurs des composants RGB où B est le nombre de bits par composant de couleur. Une telle compensation rend certains types de traitement plus facile, comme le débordement numérique, le codage arithmétique. Cette valeur est rajoutée aux composants de couleur au niveau du décodeur.
- ***Transformation de couleur***
  Comme expliqué dans la section 1.3.1.1, il existe une corrélation significative dans la représentation RGB des couleurs. En plus des types de transformation de couleurs montrés dans la section 1.3.1.1, JPEG-2000 définit un troisième type de RCT à utiliser dans le mode sans perte. Les valeurs des pixels RGB doivent être des entiers :

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (1.13)$$

  Dans le standard JPEG2000, les composants de couleurs de la transformée sont référées comme Y, U et V. La transformée décrite par l'équation (1.13) , ne décorrèle pas aussi bien qu'un ICT (1.4), mais elle a la propriété de rétablir les valeurs exactes RGB des pixels originaux.

1.3.3.2 Codage fondamental

Dans cette étape, chaque composant de couleur (YCbCr ou YUV) est codé par le codeur. Les étapes du codage par JPEG sont : la transformation, la quantification et le codage entropique. La Figure 1-12 illustre en détail le principe de ce codeur.

1.3.3.2.1 Transformée en ondelette discrète (DWT[6])

Dans le codeur JPEG2000, la DCT qui est utilisée pour la transformation des pixels dans JPEG, est remplacée par la DWT. La DWT est choisie pour remplir certains besoins de performances prises *a priori* par le comité « JPEG ». A titre d'exemple :

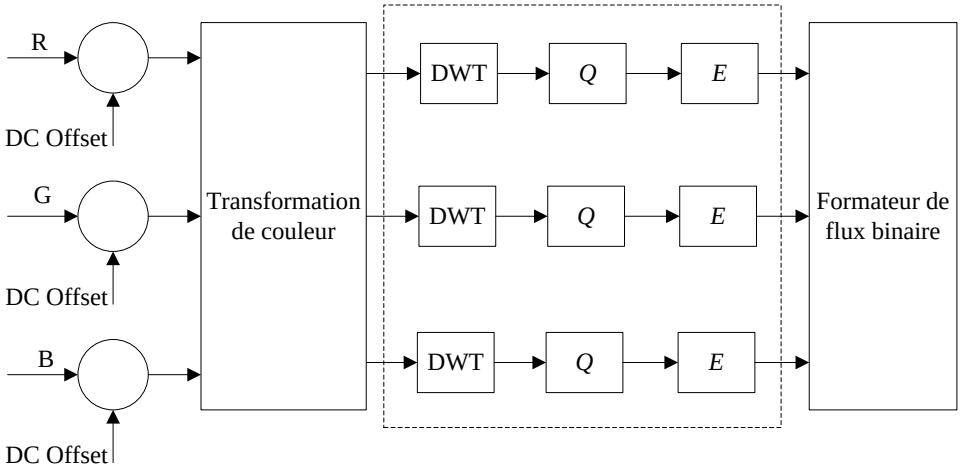


Figure 1-12 Les éléments du codeur de JPEG 2000.

- La représentation multi-résolution de l'image est une propriété intrinsèque de la transformée en ondelettes. Cela fournit également une scalabilité spatiale, sans avoir à sacrifier l'efficacité de la compression,
- Même pour les petites tailles de tuiles choisies, la DWT ne crée pas d'artefacts,
- La DWT exploite une région plus large de l'inter-corrélation des pixels. Elle produit un gain de compression plus élevé pour des faibles taux binaires choisis, car la DCT dans JPEG exploite seulement une région de 8x8 pixels,
- La DWT avec des coefficients entiers peut être utilisée pour la compression sans perte. Il est à signaler, que dans le cas de la DCT, les coefficients de la transformée sont arrondis. Ceci rend impossible la compression sans perte.
- Le standard JPEG2000 recommande deux types de banc de filtres à utiliser dans la compression avec et sans perte. Le standard définit deux types d'ondelette à utiliser pour la compression avec perte et sans perte. Pour le mode sans perte, le standard choisit l'ondelette 5/3 de Le Galle et Tabatai [**11**]. Il y a 5 coefficients pour le filtre passe-bas et 3 coefficients pour le filtre passe-haut. Tous coefficients sont des entiers. Ce qui rend réversible à la transformée. Pour le mode avec, l'ondelette 9/7 de Daubechies à coefficient réels [**12**] [**13**] est utilisée.

Le Tableau VII met en évidence les coefficients des filtres d'analyse passe-bas et passe-haut pour les filtres 9/7 et 5/3 respectivement.

[6] DWT-Discrete Wavelet Transform

1.3.3.2.2 Quantification

Après la transformation en ondelette discrète, les coefficients obtenus sont quantifiés linéairement par un quantificateur à zone morte (la Figure 1-6). Le choix du pas de quantification peut être conduit par l'importance perceptuelle de la bande en question pour le système visuel humain. De la même façon que dans JPEG (Tableau VI), une matrice de pondération des coefficients peut être utilisée.

TABLEAU VII
LES FILTRES D'ANALYSE PASSE-BAS
ET PASSE-HAUT UTILISES DANS JPEG2000

Coefficients	Compression avec perte (9/7)		Compression sans perte (5/3)	
	Passe bas	Passe haut	Passe bas	Passe haut
0	+0,602949	+1,115087	3/4	1
±1	+0,266864	-0,591272	1/4	-1/2
±2	-0,078223	-0,057544	-1/8	
±3	-0,016864	+0,091272		
±4	+0,026729			

1.3.3.2.3 Codage entropique

Les indices des coefficients quantifiés dans chaque sous-bande sont codés en entropie afin de créer un flux binaire compressé. Pour JPEG2000, le comité de JPEG propose le codage « embedded bloc coding with optimized trunctation » ou EBCOT [**10**].

Dans EBCOT, chaque sous-bande d'une tuile de l'image est repartitionnée dans des blocs rectangulaires qui sont appelés « blocs de code». Ces blocs de code sont codés en entropie individuellement. Les détails du codage EBCOT sont expliqués dans [**3**].

1.3.3.3 Post-traitement

Une fois l'image compressée, le flux binaire produit par les blocs de code individuels est retraité pour faciliter certaines fonctionnalités du standard JPEG2000 :

- La région d'intérêt : la capacité de compresser certaines parties d'une image avec un faible taux de compression.
- L'évolutivité : la capacité de décoder une image avec plusieurs niveaux de qualités ou de résolutions depuis le flux binaire.

1.3.4 Conclusion des compressions JPEG (JPEG et JPEG 2000)

1.3.4.1 Artefacts de blocs

La compression par JPEG ne peut fonctionner avec des blocs de 8x8 à la fois. Ceci provoque des artefacts connus sous le nom de « des artefacts de blocage » dans le domaine de la compression d'image. Contrairement à la DCT utilisée dans JPEG, la DWT dans JPEG2000 n'est pas limitée à des blocs de taille fixe.

Bien que JPEG2000 supporte comme option, le découpage de l'image en des tuiles avant la compression, cette fonctionnalité est rarement utilisée pour les images de très grande taille, car les tuiles aussi peuvent provoquer artefacts de blocs remarquables (voir la Figure 1-13).

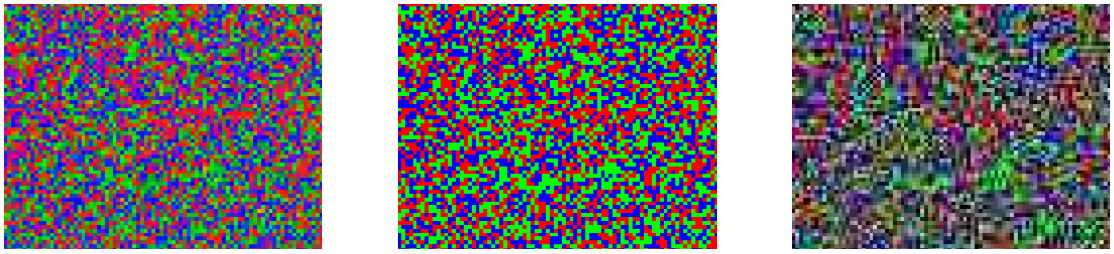
JPEG2000 0.25 bpp (a) | Image originale (b) | JPEG 0.25 bpp (c)

Figure 1-13 Les artefact de blocs sur les images compressées (a) Image compressée par JPEG 2000 (b) Image originale (c) Image compressée par JPEG.

1.3.4.2 Distorsion des couleurs:

Comme nous avons évoqué au §1.3.1.1, le système visuel humain (SVH) est beaucoup plus sensible aux changements dans la luminosité par rapport aux changements. Ce fait il est largement exploité dans la compression par JPEG et JPEG2000. Les codeurs conservent la luminance et défavorisent une grande partie de l'information liée à la couleur. Le processus impliqué est appelé « le sous-échantillonnage de chrominance », pendant lequel les plans Cb et Cr de la décomposition RCT de l'image, sont sous-échantillonnée par un taux plus élevé que le plan Y.

Cette distorsion de la couleur se manifeste sur les images compressées par des couleurs moins brillantes et légèrement délavées [**14**]. Ce problème semble être plus grave dans JPEG que JPEG2000 (voir la Figure 1-14).



JPEG2000 Image originale JPEG

(a) (b) (c)

Figure 1-14 La distorsion de la couleur qui existe dans les codeurs JPEG et JPEG 2000
(a) JPEG2000 (b) Image originale (c) JPEG

1.3.4.3 Artefacts de « ringing »:

La compression par JPEG 2000 et par JPEG fonctionnent dans le domaine spectral. Ces deux standards essayent de représenter une image par une somme des ondes oscillantes. En effet, le domaine spectral est un domaine approprié pour capturer les basses variations sur les pixels de l'image, mais pas pour capturer les variations brusques comme il existe par exemple sur les contours (voir Figure 1-15).

JPEG2000 Image originale JPEG

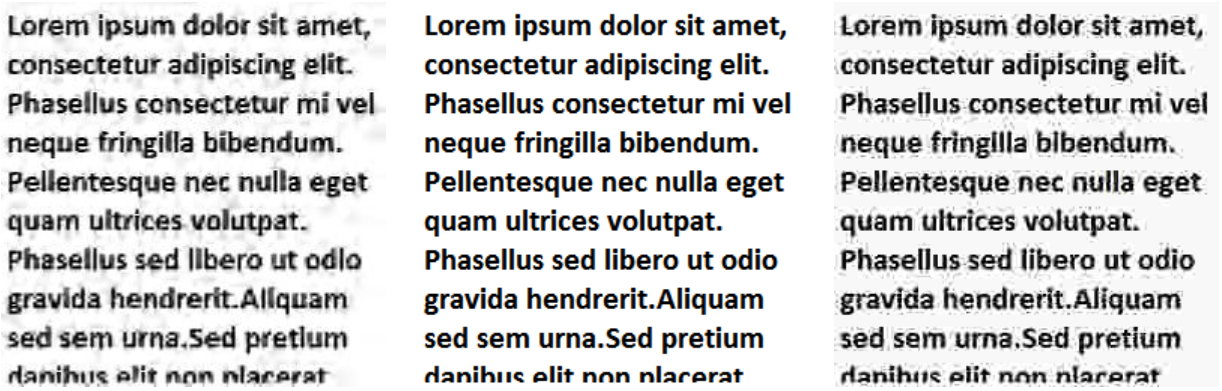
(a) (b) (c)

Figure 1-15 Les artefacts de « ringing » qui se produisent sur les images compressées.
Image (a) compressée par JPEG2000 (b) originale (c) compressée par JPEG

Un bon exemple pour les artefacts de « ringing » résulte de la compression des images avec des arrêts vives. En particulier, du texte, des diagrammes et des dessins au trait. La Figure 1-16, illustre cet effet sur une image d'un texte compressée par JPEG et par JPEG 2000 [**14**].

La Figure 1-16(a) illustre l'effondrement de la capture de bord, qui est efficace autrement dans JPEG2000. L'effet principal lié à ce problème, est l'apparition des « crêtes » courts verticales et horizontales. Comme, il peut être vu sur la Figure 1-16 (c), JPEG souffre du même problème, aggravé par de graves artefacts de blocs.

Les résultats indiquent que JPEG2000 est meilleur que JPEG, mais les qualités de deux standards restent très inférieures par rapport à la compression sans perte (voir la Figure 1-16 (b)).



JPEG2000 (a) | Image originale (b) | JPEG (c)

Figure 1-16 Les artefacts de « ringing » qui se produisent sur les images compressées.
Image (a) compressée par JPEG2000 (b) originale (c) compressée par JPEG

1.3.4.4 Brouillage

Le brouillage signifie un lissage sur l'image que sa version originale. Il est signalé que la compression par JPEG2000 a beaucoup de problèmes pour ce type d'artéfact, surtout aux faibles bitrates. Bien que l'information soit conservée dans la forme, elle provoque une perte de la texture [**14**]. Comme le SVH est très sensible au brouillage, c'est un des cas où le JPEG2000 fonctionne moins bien que le JPEG de base.

JPEG2000 (a) | Image originale (b) | JPEG (c)

Figure 1-17 Exemple de brouillage qui s'apparaissent sur les images compressées aux faibles bitrates par JPEG2000 et JPEG standard
(a) Image compressée par JPEG2000 (b) Image originale (c) Image compressée par JPEG

La JPEG2000 introduit plusieurs nouvelles fonctionnalités comme la région d'intérêt de codage. Son mode progressif semble bien, ainsi que ses caractéristiques pour la résilience d'erreur sont rarement utiles pour la fiabilité de la communication numérique [**15**].

1.4 Compression de la vidéo

Les analyses statistiques [**3**] effectuées sur des signaux de vidéo indiquent qu'il existe une forte corrélation entre les trames successives d'une vidéo. Cette corrélation existe aussi entre les pixels de chaque trame. Théoriquement, la décorrélation de ces signaux peut mener à la compression de la bande passante sans dégrader significativement la qualité de résolution de l'image.

De plus, l'insensibilité du système visuel humain à certaines pertes spatio-temporelles dans l'information peut être exploitée pour diminuer davantage la redondance. Par conséquent, les techniques de compression avec perte peuvent être employées pour diminuer la densité du flux binaire, en maintenant en même temps une qualité acceptable de l'image.

En compression d'images, on exploite seulement la corrélation spatiale et la technique utilisée est appelée le codage *intra-trame*. Cette technique forme la base de la compression JPEG. Si l'on exploite aussi la corrélation temporelle, la technique est appelée le codage *inter-trame*. Le codage prédictif inter-trame est le principe essentiel dans tous les standards de codecs utilisés aujourd'hui, comme H.261, H.263, H.264, MPEG-1,2 et 4 [**5**] [**16**] [**17**] [**18**].

Les standards de la compression vidéo se basent sur ces trois principes de réductions de redondance :

1. La réduction de la redondance spatiale : réduire la redondance entre les pixels dans une image en utilisant une technique de compression telle que « le codage par transformation » (voir la section 1.3.1),
2. La réduction de la redondance temporelle : enlever les similarités présentes entre les trames successives, en codant seulement leurs différences,
3. Le codage entropique : réduire la redondance entre les symboles compressés à la sortie du codeur en utilisant une des techniques de codage entropique (voir la section 1.2.1).

1.4.1 Réduction de la redondance temporelle

La redondance temporelle est réduite en calculant la différence entre des images successives. Cela est appelé le codage inter-trame. Pour les régions statiques de la séquence d'images, cette différence est quasi nulle, par conséquent, elles ne sont pas codées. Les régions qui présentent du changement entre des trames successives (due au changement de luminance ou au mouvement des objets) entrainent une erreur de l'image significative qui doit être codée. L'erreur causée par le déplacement des objets sur une image, peut être réduite si le mouvement de l'objet en question peut être estimé [**16**]. Dans tous les standards du codage de la vidéo [**16**] [**17**] [**19**] [**20**] [**21**], ces trames successives qui sont codées ensemble forment une entité hiérarchique qui s'appelle *Group of Pictures* (GOP).

En d'autres termes, un GOP est une série d'une ou plusieurs trames qui facilitent l'accès aléatoire à la séquence. Dans un GOP, la première image est codée en intra-trame (souvent dénotée par la lettre *I*). Ensuite, les images

successives sont codées en trame *P* ou en trame *B* selon le résultat de l'étape de l'estimation. Les trames *P* sont des trames prédictives et sont codées en inter-trame en utilisant une des trames précédentes (*I* ou *P*) comme repère. Il est à signaler qu'une trame *P* (Prédite), peut, elle-même, être un repère pour les trames qui lui succède [**19**]. Il existe un autre type de trames dans une structure GOP appelé : les trames *B* (Bidirectionnelle) ou les trames prédites bi-directionnellement. Ces trames de type *B* peuvent utiliser des trames déjà codées, ou qui vont l'être, ou une combinaison des deux. Ceci améliore davantage la qualité de la compression [**19**]. La Figure 1-18

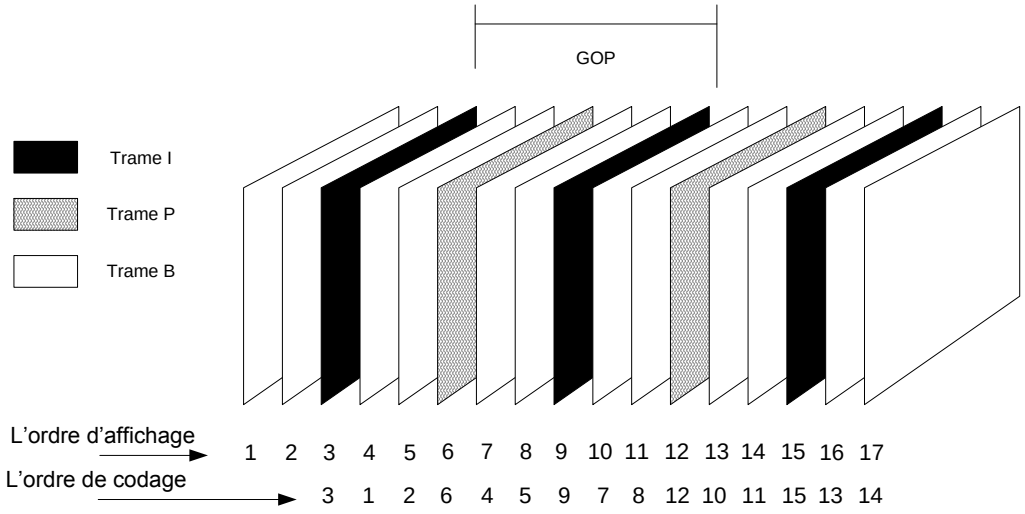


Figure 1-18 Une exemple de GOP. [**3**]

illustre la disposition des trames et de GOP dans une séquence MPEG.

1.4.2 Estimation du mouvement

La première étape, dans la compensation de mouvement, consiste à estimer les objets. La technique la plus utilisées dans les codecs standards de vidéo pour réaliser une estimation du mouvement est le « block matching » (BMA). Le principe de l'algorithme, dans sa version original, consiste à diviser une trame en plusieurs blocs de ($M \times N$) pixels ou N^2 dans le cas de blocs carrés [**22**]. Ensuite, il s'agit de chercher pour chaque bloc le bloc le plus ressemblant dans une image de référence. A l'initialisation, l'algorithme commence la recherche à partir d'un point de départ et choisit un ensemble de points candidats autour du point courant, selon une distance appelée "pas de recherche". Pour un mouvement de déplacement maximal de *w* pixels par trame, le bloc courant de pixels est apparié au bloc correspondant au même coordonnées, mais dans la trame précédente, à l'intérieur de la fenêtre de taille $N+2w$ (voir la Figure 1-19). Le meilleur appariement est obtenu selon le critère utilisé.

Plusieurs types de critère de mesure de similarité peuvent être utilisés comme la fonction de corrélation, l'erreur quadratique moyenne (EQM), l'erreur absolue moyenne (EAM) [**23**] [**24**] [**25**]. Dans le cas de la fonction de corrélation, le meilleur appariement est celui qui maximise cette fonction. Cependant, dans une seconde étape, deux critères doivent être minimisés. Les fonctions d'appariement de type EQM et EAM sont définies par :

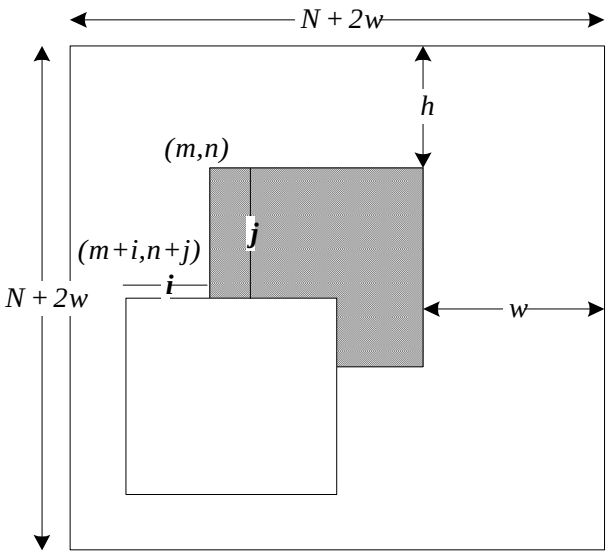


Figure 1-19 La trame au courant et la trame précédente dans une fenêtre de recherche

Pour EQM :

$$M(i,j) = \frac{1}{N^2}\sum_{m=1}^{M}\sum_{n=1}^{N}\left(f(m,n) - g(m+i,n+j)\right)^2 \qquad -w \le i,j \le w \qquad (1.14)$$

et pour EAM :

$$M(i,j) = \frac{1}{N^2}\sum_{m=1}^{M}\sum_{n=1}^{N}\left|f(m,n) - g(m+i,n+j)\right| \qquad -w \le i,j \le w \qquad (1.15)$$

où *f(m,n)* correspond au bloc courant de taille N^2 pixels de coordonnées *(m,n)* et *g(m+i,n+j)* est le bloc correspondant dans la trame précédente au nouvelles coordonnées *(m+i,n+j)*. Le meilleur appariement correspondra à la position *(i=a)* et *(j=b)* et le vecteur de *MV(a,b)* de mouvement représente le déplacement de tout les pixels à l'intérieur du bloc.

La recherche exhaustive du meilleur appariement nécessite $(2w+1)^2$ évaluations du critère de similarité. Afin de réduire cette complexité de calcul, le critère EAM est préféré à EQM dans les codecs vidéo. Cependant, pour chaque bloc de taille N^2 pixels, nous devons encore faire $(2w+1)^2$ évaluations du critère de similarité. Chacun avec environ $2N^2$ additions et soustractions. Ceci est handicapant quant à l'utilisation de l'algorithme BMA dans les logiciels utilisant des codecs. L'étude de la complexité des codeurs vidéo a montré que l'estimation du mouvement est comprise entre, environ, 50% et 70% de la complexité totale du codeur [**3**] [**26**].

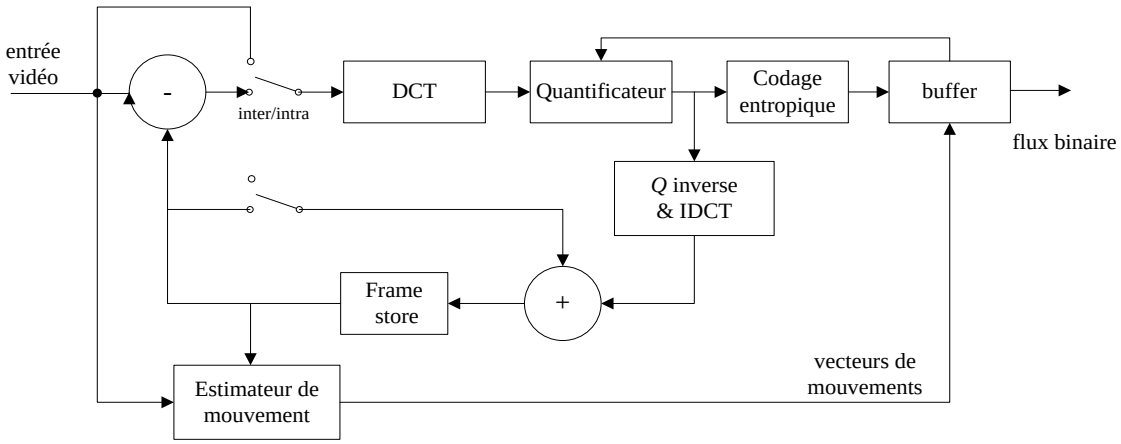


Figure 1-20 Un codeur générique prédictif inter-trame

TABLEAU VIII
LE POURCENTAGE DU TEMPS DE TRAITEMENT NECESSAIRE POUR L'ESTIMATION DU MOUVEMENT DANS LE CODEUR MPEG-1 (GHANBARI 2003)

Category	**DCT Rapide**		**DCT Brute force**	
Type de Trame	Mobile	Claire	Mobile	Claire
P-Trame ME	66.1%	68.4%	53.3%	56.1%
B-Trame ME	58.2%	60.9%	46.2%	48.7%

Bien sur, cette estimation dépend de l'activité dans la scène et si la transformée en cosinus discrète rapide est utilisée pour le calcul des coefficients. Par exemple, le pourcentage du temps de traitement pour calculer le vecteur de mouvement dans le cas des séquences de test « Claire » et « Mobile » en utilisant le codec MPEG-1 est donné dans le Tableau VIII. Il est à signaler que le temps nécessaire pour l'estimation du vecteur de mouvement dans le cas des images B-Trame est supérieur à celui de P-Trame car l'intervalle de recherche dans les P-Trame est plus grand que dans les B-Trame. Par conséquent, le temps total de traitement, dans le cas des P-Trame, est supérieur à celui dans le cas des B-Trame, comme présenté sur le Tableau VIII.

1.4.3 Les principes des codeurs inter-trames de vidéo

La Figure 1-20 montre un codeur générique basé sur l'inter-trame qui est utilisé dans la plus part des codecs standards de vidéo tel que: H.261, H.263, H.264, MPEG-1, MPEG-2 et le MPEG-4 [**16**] [**17**] [**18**] [**21**] [**19**] [**20**]. Dans les paragraphes suivants, le principe de base de chaque élément de ce codec est présenté.

1.4.3.1 Boucle inter-trame

Dans le codage basé sur la prédiction de l'inter-trame, l'erreur de prédiction des pixels dans la trame en court à partir de la trame précédente est codée puis transmise. A niveau du récepteur, après décodage le signal erreur

associé à chaque pixel, est ajouté à la valeur prédite pour reconstruire l'image. Les performances de cette méthode dépendent de la qualité du prédicteur : une meilleure prédiction se traduit par une faible erreur de prédiction, ce qui par conséquent améliore le débit. Si la scène est immobile, la qualité de la prédiction est très bonne, ce qui veut dire que le pixel dans la trame courante est le même que dans la trame précédente. Cependant, dans le cas où un mouvement existe, en supposant que le mouvement de l'image est la conséquence du déplacement de la position de l'objet, alors un pixel déplacé dans la trame précédente suivant un vecteur de mouvement est utilisé.

1.4.3.2 Estimation du mouvement

Attribuer un vecteur de mouvement à chaque pixel est très lourd d'un point de vue calcul. Pour palier à ce problème, la compensation de mouvement de pixels est utilisée, de manière à ce que le vecteur de mouvement au dessus par pixel soit petit. Dans les codecs standards, le mouvement d'un bloc 16x16 pixels, appelés macroblocs (MB), est estimé puis compensé. Il est à signaler que l'estimation est basée sur l'information de luminance des images. Une version modifiée de cette approche est utilisée pour la compensation des blocs de chrominance, en fonction des formats des images.

1.4.3.3 Inter/intra commutateur

Chaque MB est codé soit en inter-trame soit en intra-trame, appelé inter/intra MB. Le choix de l'une des deux techniques dépend de la technique de codage. Par exemple, dans JPEG, tous les MB sont codés en intra-trame car le JPEG est utilisé surtout pour coder des images, qui sont immobiles.

1.4.3.4 Transformée en cosinus discrète (DCT)

Chaque MB est divisé en blocs de luminance et chrominance de taille 8×8 pixels. Chaque bloc est transformé alors via la transformée en cosinus discrète (DCT). Il y a quatre blocs de luminance dans chaque MB mais le nombre de blocs de chrominance dépend du format de l'image.

1.4.3.5 Quantification

Il existe deux types de quantificateurs: le premier avec une zone morte, pour les coefficients AC et les coefficients DC de l'inter MB; le second sans zone morte est utilisé pour les coefficients DC de l'intra MB. L'intervalle de coefficients quantifiés peut être de -2047 à +2047 [**3**]. Le principe du quantificateur avec une

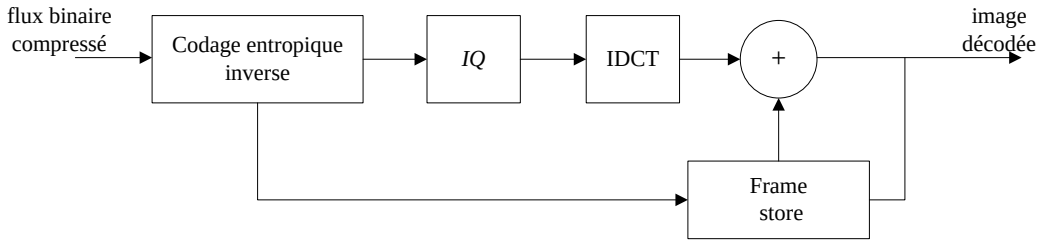


Figure 1-21 Diagramme de bloc d'un décodeur

zone d'élimination est le suivant: si la valeur absolue d'un coefficient est inférieure au pas de quantification *q*, elle est mise à zéros, sinon elle est quantifiée pour générer les indices de quantification.

1.4.3.6 "Déquantification" (quantification inverse, IQ) ou DCT inverse(IDCT)

Pour générer une prédiction pour le codage inter-trame, les coefficients quantifiés de la DCT sont inversement quantifiés puis la DCT inverse est appliquée. Les valeurs ainsi trouvées sont ajoutées à leurs valeurs dans l'image précédente, pour générer un réplica de l'image décodée. L'image est alors utilisée comme une base de prédiction pour l'image suivante de la séquence.

1.4.3.7 Le Tampon (la mémoire tampon)

Le débit généré par un codeur inter-trame est variable. Ceci parce que le débit dépend fortement du mouvement des objets et de leurs détails. Par conséquent, pour transmettre une vidéo avec un débit fixe (par exemple, 2Mo/s), le débit doit être régulé. Pour cela, il suffit d'enregistrer les données codées dans une mémoire tampon et la vider lorsqu'il est nécessaire. Cependant, dans le cas d'une activité intense (i.e. beaucoup de mouvements), le tampon peut être surchargé et une solution à ce problème consiste à mettre en place un retour du tampon vers le quantificateur pour réguler le débit. Comme l'occupation du tampon augmente, le retour force le pas de quantification à augmenter pour réduire le débit. De la même manière, si l'activité dans la scène est faible, le pas de quantification est réduit afin d'améliorer la qualité de l'image.

1.4.3.8 Décodeur

Les données compressées sont décodées par longueur variable (par la méthode de Huffmann ou le codage arithmétique) puis démultiplexées afin de séparer les vecteurs de mouvement et les coefficients DCT. Il est à signaler que les vecteurs de mouvements sont utilisés pour ajuster les coefficients de la DCT. Après quantification inverse, la DCT inverse est appliquée pour convertir ces coefficients en image d'erreur. Cette dernière est ajoutée à la trame précédente pour reconstruire l'image courante. Le schéma de la

Figure 1-21 illustre le principe de cette procédure de décodage.

1.5 Conclusion

Dans ce chapitre, nous avons présenté un panorama succinct des techniques et des standards de compression.

Les techniques présentées pour la réduction de la redondance binaire sont intégrées dans l'ensemble des standards actuels. Dans certains nouveaux codecs, les techniques de codage arithmétique remplacent le codage du Huffmann pour la réduction de l'entropie.

Dans ce chapitre, deux standards de compression d'images ont été présentés, à savoir, le JPEG et le JPEG 2000. Le JPEG est un codec largement utilisé. Il est intégré dans de nombreux systèmes embarqués. Par ailleurs, le JPEG 2000, fondé sur la transformée en ondelette permet d'obtenir de meilleures performances en terme de débit-distorsion. Pour cette raison, le JPEG 2000 sera utilisé dans l'approche de Compression Multimodale que

l’on va présenter dans les prochains chapitres. Quant à la compression vidéo, il a été reporté que les codecs vidéo partagent un système de base commun. Parmi ces codecs, le H.264 ou le MPEG4-10 [**18**] sont des standards très utilisés. Une variante de l’approche développée dans cette thèse intégrera le H.264.

Chapitre 2

Compression Multimodale : concept et variantes

2.1 Introduction

Dans ce chapitre, nous présenterons la base ainsi que des variantes de la Compression Multimodale introduite dans [**27**] et dans [**28**]. Cette technique permet d'encoder conjointement une image et un ensemble de signaux par la simple utilisation d'un codeur unique (i.e. JPEG 2000). Dans ce contexte et afin d'éviter toute confusion avec la stéganographie et le tatouage, la section 2.2 sera consacrée aux techniques de dissimulation de l'information. Après la présentation du concept général de la Compression Multimodale en section 2.3, on s'intéressera à quelques variantes de cette technique à savoir, l'approche spatiale et fréquentielle (section 2.4 et 2.5) en mode non-supervisé, puis une extension de la Compression Multimodale au mode supervisé (section 2.6). Ce chapitre se termine par une conclusion.

2.2 Dissimulation de l'information

La dissimulation de l'information est une libre adaptation de l'expression anglaise « *information hiding* » Elle est couramment utilisée dans la littérature [**29**]. Le concept de dissimulation d'information est très général. Il s'agit simplement du fait de cacher une information dans un support. Toutefois selon les objectifs ou les contraintes, il est possible de distinguer plusieurs types et techniques.

2.2.1 La stéganographie

Le terme stéganographie vient des mots grecs, « steganos » qui veut dire « caché » et « graphien » qui veut dire « écrire ». La stéganographie a pour objectif de cacher un message secret dans un support de données de façon à ce que le destinataire soit l'unique personne autorisée à le déchiffrer.

Dans la terminologie de la stéganographie, un support de données de couverture ou médium désigne le médium vierge dans lequel des informations sont cachées. Une fois les informations insérées, le médium initial devient un stégo-médium.

2.2.2 Le tatouage (ou le watermarking)

Le tatouage a pour but de répondre au problème de la protection des droits d'auteur. Il tente de fournir une solution pour prouver qu'une entité est bien le véritable propriétaire d'un médium. Il s'agit bien de dissimulation, puisque pour y parvenir, on insère un tatouage dans le médium spécifique au propriétaire. Comme celui-ci souhaite protéger son médium et non une version trop dégradée, l'insertion doit minimiser les modifications subies par le medium afin d'être imperceptible. De ce côté, comme ce dernier ressemble à la stéganographie, cela ne présente pas la même chose car un attaquant connait déjà qu'il existe une information cachée dans le stégo-médium. Toutefois cette connaissance ne lui permet pas de supprimer le tatouage ni d'y accéder.

2.2.3 Bilan des techniques de dissimulation de l'information

Les techniques de dissimulation de l'information s'intéressent principalement à la *sécurité* de l'information et non pas à sa compression. La Figure 2-1 résume cette idée. Un message secret peut être embarqué dans une image, en le chiffrant *a priori* par une clé de chiffrement. A la réception, l'image résultante est reçue par le destinataire et s'il connait la clé de déchiffrement, il lui est possible de décoder et visualiser le message secret. L'image résultante peut aussi être interceptée par un tiers, dans la terminologie de la dissimulation de l'information, cette entité est appelée « l'attaquant », de manière générale, il peut s'agir de toute personne qui récupère l'image, ou le stégo-medium à la réception et qui essaie d'atteindre le message secret qu'elle contient. L'objectif fondamental des techniques de dissimulation de l'information est de trouver des méthodes robustes et

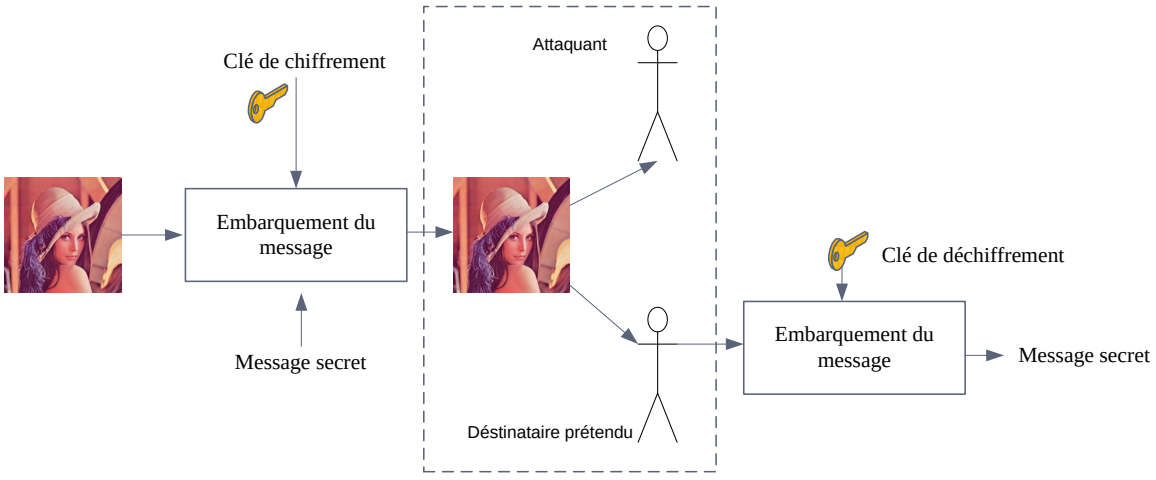


Figure 2-1 Schéma de fonctionnement général des techniques de dissimulation de l'information (la stéganographie et le tatouage)

fiables, pour mieux protéger cette information secrète. La Figure 2-2 illustre un exemple d'application d'une technique de stéganographie.

Il existe quand même dans certaines techniques, l'utilisation de la dissimulation pour compresser un deuxième signal dans l'autre. [**30**] [**31**] [**32**]. Cependant, ces techniques sont limitées en capacité et ne sont pas très robustes à l'utilisation des codeurs avec perte, car comme toutes les techniques de dissimulation de l'information sont basées sur la modification bit à bit des échantillons. Par conséquent, elles sont sujettes à être tronquées brusquement lors de l'étape de la quantification du codeur.

Pour plus d'information, le lecteur peut consulter [**33**] [**29**] [**34**] sur la théorie et les techniques de dissimulation de l'information

L'image d'un arbre (à gauche). La suppression de tous les bits sauf les deux derniers de chaque composant de couleur produit une image entièrement noire. Augmenter la luminosité de cette dernière produit l'image du chat (à droite).

Figure 2-2 Exemple d'application de stéganographie (images prises depuis Wikipédia - http://en.wikipedia.org/wiki/Steganography)

2.3 La Compression Multimodale

D'un point de vue « traitement de signal », nous pouvons visualiser les données numériques comme une série discrète d'échantillons de signal. Les systèmes de compression se basent sur les différentes propriétés du signal en question, telles que ses dimensions, la distribution probabiliste des valeurs d'échantillons, et la plage dynamique du signal. Autrement dit, les systèmes de compression dépendent de la modalité et de la distribution statistique du signal qui est à compresser.

Un système de compression / décompression peut être désigné sous le nom de *codec*. Actuellement, il existe de nombreux systèmes de codec, créés pour fonctionner sur un type de signal spécifique. Par exemple, dans le domaine de la compression d'image, on peut citer des codecs standards, très connus et largement utilisés tels que : JPEG, JPEG-LS, GIF, et PNG. Certains d'entre eux sont adaptés à compression sans perte, dans laquelle

l'image peut être reconstruite d'une façon précise à partir du flux binaire décompressé; et d'autres sont adaptés à la fois à la compression sans perte et à la compression avec perte.

En ce qui concerne la compression du signal, il existe plusieurs codecs dédiés à un certain domaine d'applications spécifiques. En effet, toute série de données peut être considérée comme un signal. Dès lors, les systèmes de compression créés pour ces séries de données particulières reposent fondamentalement sur les propriétés physiques du signal.

Par exemple, un système de compression, développé pour un signal d'électrocardiogramme (ECG), fonctionnera moins bien pour un signal audio. Réciproquement, un système de compression développé pour compresser un signal audio (e.g. MP3, AC3 et AAC), sera moins bien adapté pour la compression d'un ECG ou d'un EEG.

La Compression Multimodale repose sur l'idée qu'un codeur n'est qu'une « boite noire ». C'est-à-dire, il s'agit d'un système qui accepte un flux binaire à son entrée et qui le transforme à un autre flux binaire à la sortie. (Voir la Figure 2-3).

Figure 2-3 Le codeur selon la Compression Multimodale

Comme nous l'avons signalé au début, les systèmes de compression sont généralement conçus en prenant en compte la propriété statistique, la plage dynamique et d'autres propriétés dépendant de la nature du signal à compresser. En fonction de la nature du signal, chaque codeur peut prendre avantage d'une ou de plusieurs techniques de diminution de redondance. Pour un signal unidimensionnel, cela peut s'agir de la diminution du débit binaire par une simple technique telle que le RLE, pour la compression d'une image, il peut s'agir de la diminution de la redondance spatiale, mais aussi de la diminution du débit. Pour la compression d'une séquence d'image ou d'une vidéo, cela peut comprendre toutes les techniques de diminution de la redondance qui ont été introduites au premier chapitre. Nous pouvons dire de manière générale que la compression d'une modalité comprend toutes les techniques de redondance de l'information des niveaux inférieurs (voir la Figure 2-4).

Nous avons vu, au premier chapitre que les systèmes de compression contemporains sont devenus englobant et complexes, afin qu'ils soient adaptés à tous types d'applications. Comme par exemple, le codeur JPEG 2000 est utilisé à la fois dans la compression des images naturelles, des images générées par ordinateur et des images biomédicales. Cela entraine un certain degré de flexibilité dans les caractéristiques d'entrée du codeur qui est assurée par certains paramètres de configuration, comme le pas de quantification, le débit binaire de la sortie etc.

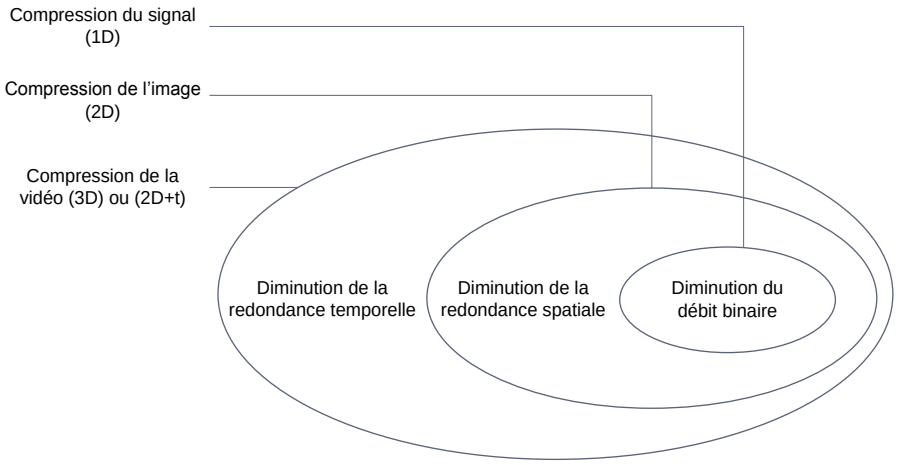


Figure 2-4 Taxonomie des techniques de compression selon la modalité.

La Compression Multimodale exploite cette flexibilité du codeur en choisissant les bons paramètres pour le codeur afin de pouvoir compresser les signaux de différentes modalités ensemble.

La Figure 2-5 illustre l'idée générale de la Compression Multimodale, pour une image et un signal unidimensionnel. Selon la Figure 2-5, par exemple, une image et un signal peuvent être en entrée d'un codeur pour être compressés ensemble.

L'utilisation de la Compression Multimodale a plus de sens quand les signaux sont sémantiquement cohérents; ce qui veut dire quand l'un est le « complémentaire » de l'autre. Comme dans notre précédent exemple, une image et un signal audio sont sémantiquement cohérents, ou une image biomédicale telle qu'une image d'échocardiogramme est sémantiquement cohérente avec un signal d'ECG.

Comme nous allons le voir dans les chapitres suivants, compresser plusieurs signaux de différentes modalités présente les avantages suivants :

1. La Compression Multimodale des signaux de différentes modalités en utilisant un seul codec peut fournir des résultats intéressants en termes de taux de compression. Soit $TC_A(s)$ le taux de compression obtenu suite à une compression du signal s avec le codeur A. Il est possible de formuler l'expression précédente comme suit :

$$TC_M(s_1, s_2) \geq TC_A(s_1) + TC_B(s_2) \qquad (2.1)$$

Dans l'équation (2.1) les codeurs A et B sont des codeurs pour les modalités des signaux s_1 et s_2,

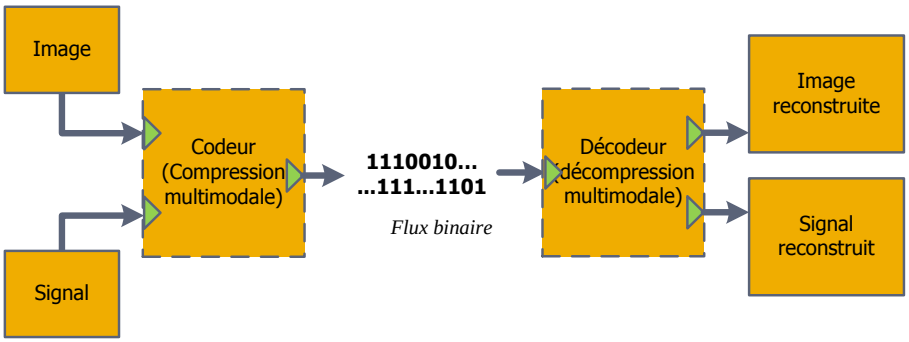


Figure 2-5 L'idée générale de la Compression Multimodale

respectivement et le codeur M peut être un codeur soit de modalité de s_1 ou s_2. A gauche de l'équation le virgule entre s_1 et s_2 indique que ces deux signaux sont compressés ensemble par le codeur M.

2. La Compression Multimodale utilise un unique codec. En termes d'implémentation et d'intégration, il est plus intéressant d'utiliser un seul codec, vis-à-vis, la puissance de calcule et la disponibilité de mémoire dans un système mobile ou embarqué.

Hormis l'augmentation du taux de compression, la Compression Multimodale permet de s'affranchir de certains problèmes liés au stockage ou à la transmission.

En reprenant notre exemple précédent, la compression d'un échocardiogramme et d'un signal ECG séparément nécessite également de stocker les données compressées séparément, ce qui rend plus difficile leur archivage. Avec la Compression Multimodale, comme les deux signaux (image et signal) sont compressés dans le même support, il est plus aisé de les récupérer lorsque le praticien a besoin de les consulter, ce qui est généralement le cas. Le choix fondamental dans la Compression Multimodale, est celui du codeur à utiliser. Nous avons vu au cours de nos tests que le codeur choisi pour la Compression Multimodale doit être un codeur compatible avec la modalité supérieure de l'ensemble. Autrement dit, pour compresser conjointement une image et un signal ECG, nous devons choisir un codeur d'image, car ce type de codeur, intègre des techniques de compression adaptées à la compression des deux modalités (voir la Figure 2-4).

Pour finaliser la mis en place de la Compression Multimodale, nous devons définir un modèle de travail pour le codeur et le décodeur qui les réaliseront (la partie marquée en pointillé sur la Figure 2-5), dans ce contexte, nous pouvons parler de deux modèles de réalisation possible :

1. Altérer le fonctionnement d'un codeur classique (JPEG, JPEG 2000 etc..) pour qu'il prenne en charge plusieurs types de signaux.

Par exemple : Le codeur JPEG qui est initialement conçu pour la compression d'image, peut être modifié pour compresser aussi d'autres types de signaux.

2. Introduire une étape de prétraitement en amont du codeur qui va reconditionner les échantillons des signaux pour pouvoir les adapter le mieux possible aux caractéristiques d'entrée du codeur (reformation du signal). Autrement dit, une fonction de mélange (ou « mélangeur »). Lors du décodage, la fonction inverse est dite fonction de séparation.

La Figure 2-6 illustre la première approche qui consiste en un modèle basé sur l'altération du codeur. Le codeur modifié admet deux entrées ; une pour chaque type de signaux. En fonction de l'entrée, il peut employer un processus de compression propre aux propriétés dynamiques et statistiques de cette dernière.

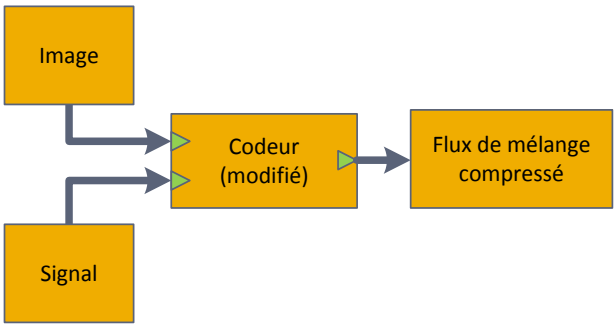


Figure 2-6 Modèle 1 : Altération du codeur pour la compression multimodale.

La deuxième approche telle qu'elle est illustrée sur la Figure 2-7 remplace le processus de Compression Multimodale présenté à la Figure 2-5 par une fonction de mélange. En principe, cette fonction de mélange permet de reformer et/ou reconditionner l'un des signaux pour que ces derniers soient similaires. A ce stade, une des questions à se poser, est : quel type de codeur allons-nous utiliser ? Etant donné deux signaux de différentes modalités, le codeur doit être du type de la modalité supérieure. A titre d'exemple, dans le cas où la fonction de mélange est utilisée pour mélanger un signal unidimensionnel et une image, ce dernier sera reconditionné. Par conséquent le codeur doit être un codeur d'image.

Lors du décodage, la fonction de mélange est complémentée par une fonction de séparation, qui est la réciproque de la fonction de mélange. Après la décompression du flux binaire, le rôle du séparateur est de séparer les signaux qui ont été mélangés lors du codage et d'assurer la reconstruction des échantillons du signal. Dans un tel contexte, idéalement, il serait souhaitable que ces étapes de mélange et de séparation n'introduisent aucune perte sur les signaux reconstruits. Cependant lors de la Compression Multimodale, cela n'est généralement pas le cas. Les fonctions de mélange et de séparation introduisent une perte sur les signaux reconstruits. Dès lors, ces fonctions doivent être choisies de manière à ce que cette perte causée par les étapes de mélange et de séparation soit la plus faible possible.

Un autre avantage de la deuxième approche, (introduire une fonction de mélange et de séparation en amont du codeur), est la possibilité d'évaluer les performances de l'approche avec différents types de codeur, sans altérer le codeur ou le décodeur (voir la Figure 2-7).

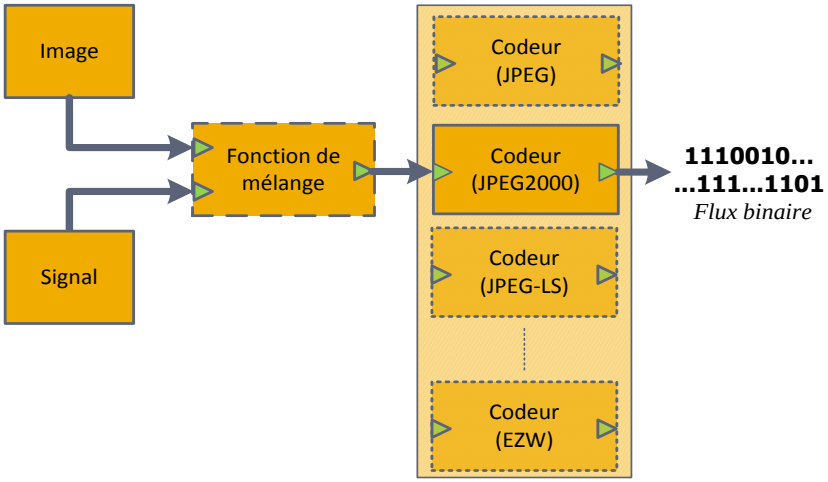


Figure 2-7 Modèle 2 : Mélanger les signaux en amant du codeur (sans modifier le codeur)

Globalement, l'objectif de la Compression Multimodale n'est pas atteindre des taux de compression plus élevés que ceux que l'on peut obtenir en compressant les signaux séparément. Nous allons illustrer dans la suite que pour un certain taux de compression, compresser un mélange de signaux, de différentes modalités avec un unique codeur au lieu de compresser ces signaux séparément, présente un certain avantage.

2.4 Compression Multimodale dans le domaine Fréquentiel (CMF)

2.4.1 Hypothèse

Le codeur JPEG2000 que nous avons étudié dans §1.3.3, en aval de la compression, permet de décomposer l'image en quatre bandes de fréquences. Ce type de décomposition de l'image en quatre bandes de fréquences d'ondelette est appelé *la décomposition dyadique* [**35**] [**36**]. Le premier niveau de décomposition met en évidence quatre bandes fréquentielles, à savoir l'approximation (*LL*), les détails (*HL* et *LH*) et les hautes fréquences (*HH*). Un exemple d'une décomposition en ondelette discrète (*Discret Wavelet Transform*, DWT) d'une image est présenté sur la Figure 2-8. La DWT nous permet d'étudier la distribution fréquentielle d'une image dans le domaine spatial. En d'autres termes, elle nous permet de localiser les composants fréquentiels dans le domaine spatial.

En général, la bande *HH*, renseigne sur la distribution des composants de hautes fréquences sur l'image initiale. Les contours ou les changements brusques de luminance sur l'image initiale, correspondent à des fortes valeurs sur la bande *HH* d'une décomposition DWT. Sur l'exemple de la Figure 2-8, la bande HH (Figure 2-8 (e)),

présente des faibles valeurs autours du zéro, sauf sur certaines régions. Cet aspect est largement exploité par JPEG2000 dans le cas du mode avec pertes.

Lorsqu'une image est codée par JPEG2000, le plus petit nombre de bits possible est alloué pour coder les coefficients appartenant à la bande *HH* car pour un bitrate de compression fixé *a priori*, la majorité des nombres de bits est réservé pour coder les coefficients qui véhiculent le plus d'information, comme les coefficients de la bande LL.

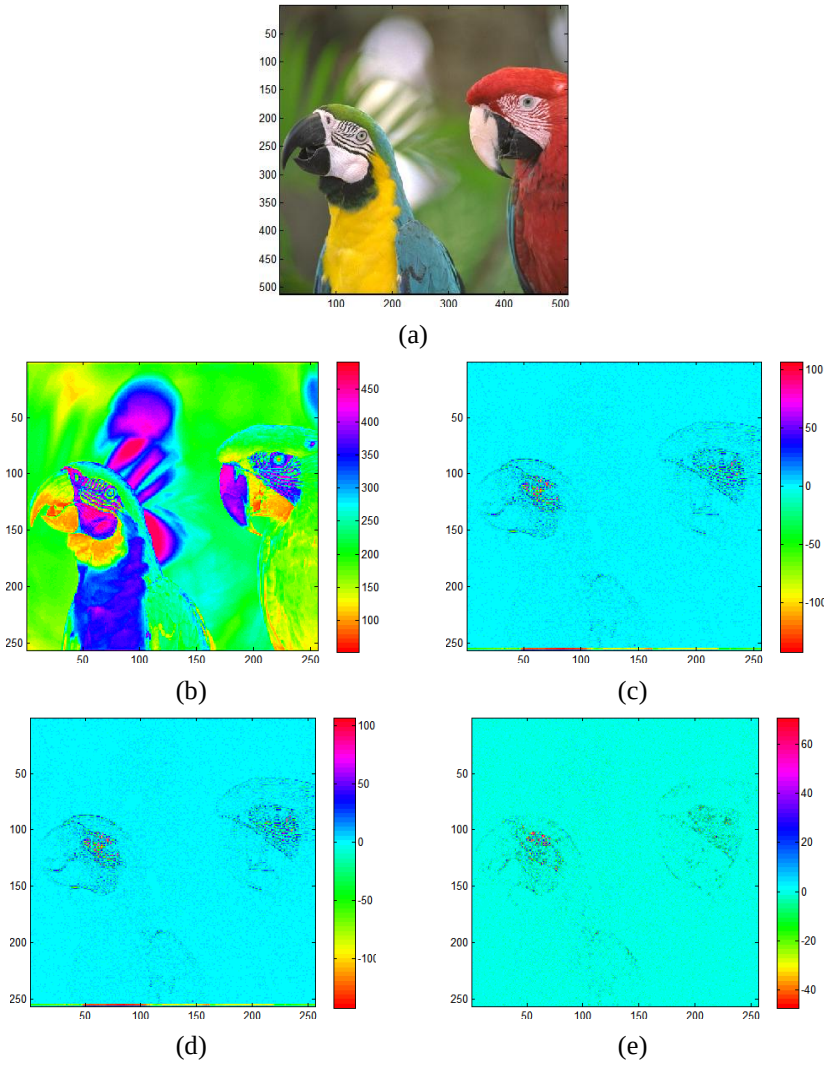
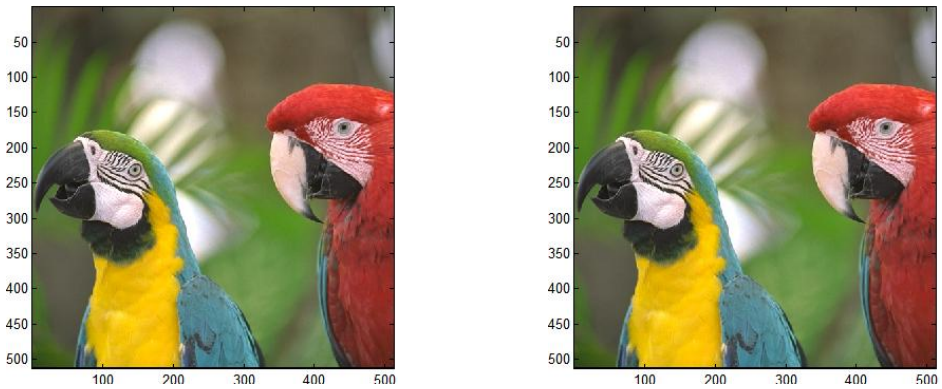


Figure 2-8 Une image(a) et sa décomposition en ondelette de premier niveau. Ici sont présentées respectivement les bandes (b) LL (c) LH (d) HL (e) HH de la décomposition.

Par conséquent, nous pouvons dire qu'une image ayant de faibles variations de luminances, possède de faibles valeurs sur la bande *HH* de sa décomposition en ondelette. En général, les échantillons de la bande *HH* sont d'amplitude tellement faibles, que si nous éliminons toutes les valeurs de la bande *HH* ou si nous remplaçons ses valeurs par des zéros, nous perdrions peu d'information sans dégrader, sensiblement, sa qualité (Figure 2-8)

En observant les images la Figure 2-9, nous pouvons croire que la suppression totale de la bande *HH* provoquerait une perte *imperceptible*. La méthode que nous allons présenter, dans le paragraphe suivant, repose sur cette idée. En d'autres termes, au lieu de supprimer cette bande de fréquences, nous pouvons remplacer aux emplacements de la bande *HH* de l'image originale par les échantillons appartenant à un autre signal



(a) (b)

Figure 2-9 (a) Image initiale (b) Image dont la bande HH de sa décomposition ondelette est remplacée par des valeurs nulles.

2.4.2 Méthode CMF

Dans cette section, nous présentons une méthode de Compression Multimodale (CM) qui fonctionne dans le domaine fréquentiel. La méthode de compression que l'on propose dans cette section est basée sur la décomposition en ondelette [**37**] [**38**].

Dans le cadre de la CM, cette méthode permet de compresser un signal unidimensionnel avec une image en utilisant un unique codeur. Dans ce contexte, nous appelons l'image « l'image porteuse » et le signal « signal embarqué ». Dans la méthode proposée, les étapes du processus de codage consistent à :

1) Sélectionner une région d'insertion de l'image dans le domaine spatial,
2) Effectuer une décomposition en ondelette de premier niveau sur l'image,
3) Insérer des échantillons du signal embarqué:
    a. Reconditionnement des échantillons du signal embarqué,
    b. Insertion des échantillons sur la région dans la partie de la décomposition d'ondelette suivant un modèle d'insertion.
4) Calculer la décomposition inverse du mélange afin d'obtenir une *image de mélange* ;
5) Compresser l'image de mélange avec un unique codeur d'image (e.g. JPEG2000),
6) Stocker ou transmettre vers un récepteur, l'image de mélange avec les paramètres de compression.

2.4.3 Processus de codage

L'utilisateur choisi une région d'insertion rectangulaire dans le domaine spatial. Nous décomposons l'image en question en quatre bandes fréquentielles sur une base d'ondelette, en utilisant un filtre *QMF* comme décrit dans [**39**].

Afin de décomposer l'image en ondelettes, les filtres passe-bas et passe-haut nécessaires sont calculés à partir de l'ondelette « *Daubechies 4* » [**36**]. Ces filtres sont illustrés sur la Figure 2-10.

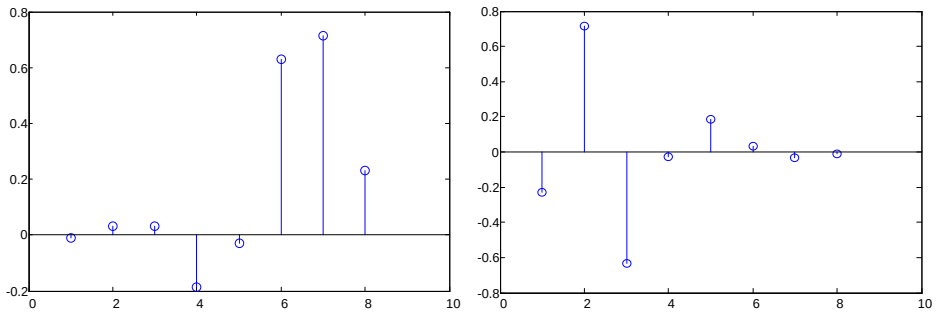


Figure 2-10 Les filtres (a) passe-bas (b) passe-haut utilisés pour la décomposition en ondelette discrète

Ces filtres sont utilisés pour décomposer l'image en quatre bandes de fréquences (*LL, LH, HL, HH)*. Une propriété importante de la décomposition dyadique est la localisation spatio-fréquentielle des pixels: les coefficients d'ondelette appartenant à un pixel (x_i, y_i) sont identifiés sur les emplacements $\left\{ x_i/2, y_i/2 \right\}$ de HH). Cette propriété est appelée l'invariance par translation de la DWT [**35**].

Pour une région d'insertion rectangulaire (x, *y, w, h*) sélectionnée *a priori* par l'utilisateur, dans le domaine spatial, des détails appartenant initialement à la zone *HH* seront remplacés par les échantillons du signal embarqué en suivant un modèle d'insertion qui est aussi défini *a priori* par l'utilisateur.

Cette première étape dans le codage est définie dans la Figure 2-11. Les valeurs de *x, y, w* et *h* doivent être paires. Si ceci n'est pas le cas, elles doivent évidemment être arrondies à la plus proche valeur entière paire. Comme expliqué ci-dessus, afin de localiser les coefficients d'ondelette appartenant à un pixel sur les bandes fréquentielles de la décomposition. Sur la Figure 2-11 BLL, BLH, BHL, BHH marquent respectivement les zones qui correspondent à la région d'insertion choisie dans le domaine spatiale.

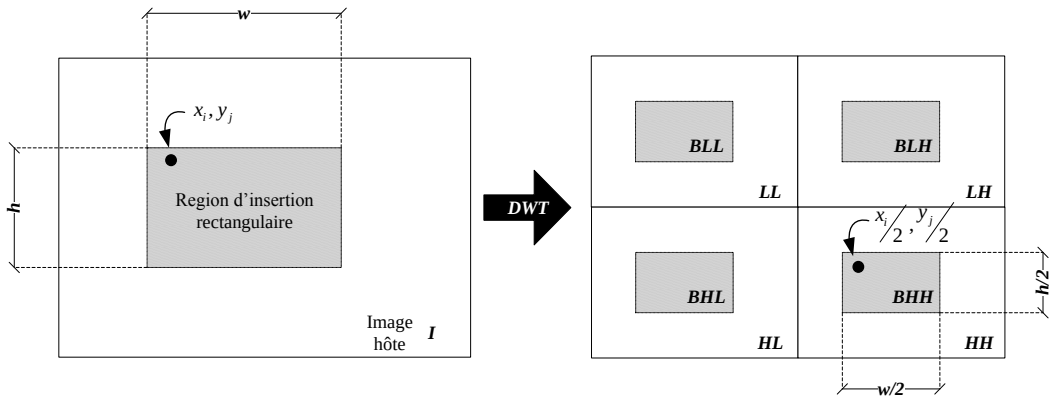


Figure 2-11 Présentation de quatre-bandes de fréquences de la décomposition en ondelette. La région d'insertion sélectionnée correspond aux régions (*BLL*, *BLH*, *BHL* et *BHH*) sur la décomposition dyadique

2.4.3.1 Reconditionnement du signal embarqué

Avant d'insérer les échantillons du signal dans la partie *HH*, le signal embarqué est normalisé si nécessaire et une mise à l'échelle. Après la normalisation doit être effectué une pondération par un paramètre α choisi. La normalisation du signal embarqué est effectuée comme suit :

$$s_i = \frac{s_i}{\max(s_i)} \qquad (2.2)$$

et la pondération du signal embarqué normalisé est donné par l'équation :

$$s' = s.\alpha \qquad (2.3)$$

Cette opération de pondération sur le signal embarqué est nécessaire car dans la décomposition dyadique de l'image porteuse, les coefficients HH s'identifient par des faibles valeurs [**35**]. Lors de la compression de l'image de mélange par le codeur JPEG2000 dans l'étape finale de la méthode, les coefficients appartenant à la zone HH sont défavorisés avec une représentation binaire très faible, qui diminue la précision dans leur représentation binaire. [**10**]. Pour surmonter ces effets, les échantillons sont amplifiés avec une valeur de pondération.

Le choix du paramètre α

Le choix de ce paramètre se fait de façon à ce que le signal ne soit pas dégradé par l'étape de quantification du JPEG2000. Le paramètre α est dépendant de la nature de l'image porteuse et peut changer d'une image à l'autre. Nous reviendrons sur le rôle de ce paramètre sur les résultats dans le chapitre suivant.

2.4.3.2 Insertions des échantillons du signal embarqué

Après la phase de reconditionnement, les échantillons du signal embarqué, sont insérés suivant le modèle d'insertion définit à la Figure 2-12. Dans ce modèle d'insertion, quatre coefficients 2x2 sur la zone BHH définissent un élément d'insertion. Le coefficient, avec les coordonnées (4,4) de cette représentation, c'est-à-dire celui qui est à gauche inférieur, est remplacé par un échantillon du signal embarqué.

$$BHH(m,n) = s_i' \qquad (2.4)$$

où m est l'ensemble des nombres pairs $\{\forall w : [2, w/2] \in \mathbb{Z}^+\}$, n est l'ensemble des nombres pairs $\{\forall h :[2, h/2] \in \mathbb{Z}^+\}$ et s_i' le i-ème échantillon du signal reconditionné. Le choix de ce modèle d'insertion est lié à la reconstruction de l'image initiale dans le processus de décodage et sera expliqué dans les sections suivantes.

Une fois le signal inséré dans la zone *HH*, une transformée en ondelette discrète inverse est calculée. Ceci créera ce que nous appelons une image de mélange.

2.4.3.3 Calcul de l'image de mélange

Il est à noter également que du fait d'introduire de nouvelles valeurs dans la zone *HH*, il n'est pas garanti que les valeurs obtenues (après inversion) restent comprises dans l'espace [0, 255], s'il s'agit évidemment de compresser une image sur 8 bits.

Cet effet se manifeste souvent sous la forme d'une composante continue que nous appelons β, superposée à l'image reconstruite. Pour cette raison afin de conditionner l'image résultante à l'entrée du codeur JPEG-2000, une composante continue β doit être supprimée, puis évidemment rajoutée dans la phase de décodage (voir la section suivante). La valeur de la composante continue est calculée sur l'image de mélange après la transformée

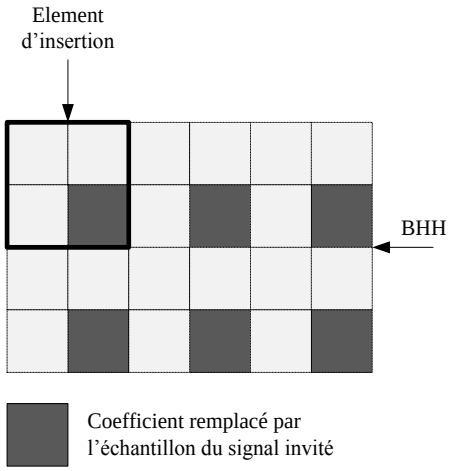


Figure 2-12 Le modèle d'insertion utilisé pour insérer les échantillons du signal embarqué dans la partie HH

en ondelette discrète inverse, de manière suivante ;

$$\beta = \min(I) \qquad (2.5)$$

De plus, les valeurs de l'image inversée doivent être arrondies. L'image obtenue que l'on compresse par JPEG-2000 sera noté I'' tel que ;

$$I'' = round(I' - \beta) \qquad (2.6)$$

où *round* (.) est l'operateur d'arrondissement au plus proche nombre entier.

La Figure 2-13 illustre les étapes comprises dans le processus de codage.

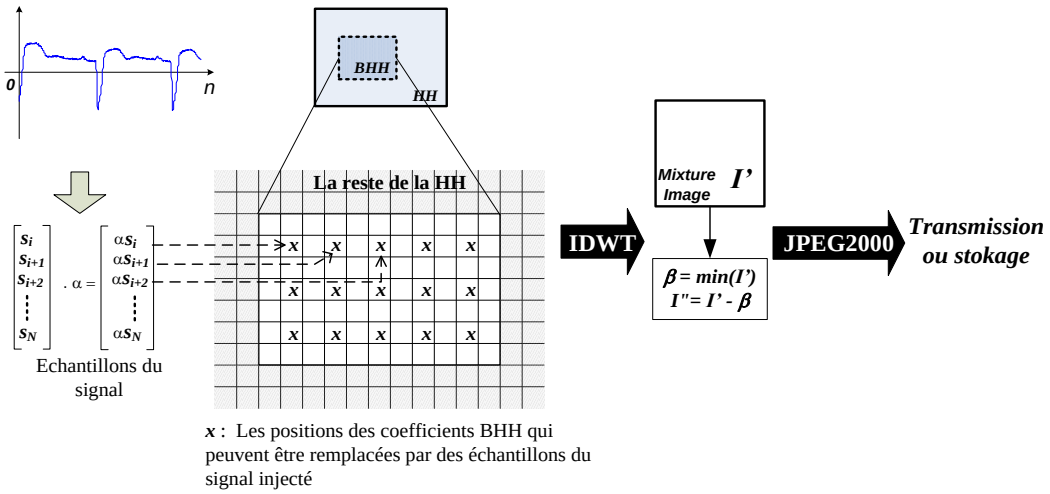


Figure 2-13 Schéma de principe de l'insertion des coefficients du signal embarqué dans les coefficients *HH*.

2.4.4 Procédure de Décodage

Après le codage, l'image de mélange compressée peut être transmise sur un support de transmission ou bien, peut être stockée pour être décompressée plus tard. Au coté du décodeur, le processus de décodage comprend les étapes suivantes :

1) Décompression de l'image de mélange,
2) Décomposition en ondelette de premier niveau de l'image de mélange,
3) Reconstruction de l'image et du signal à partir de l'image de mélange,
    a. Extraction des échantillons du signal embarqué
    b. Prédiction des coefficients HH
4) Calcul de la décomposition en ondelette inverse afin de reconstruire l'image initiale,
5) Reconditionnement des échantillons du signal embarqué.

Le schéma de décodage est illustré à la Figure 2-14. Dans un premier temps, JPEG-2000 décode le mélange et l'image de mélange est récupérée à la sortie. Puis la composante continue β, qui a été supprimée lors du codage, est rajoutée comme suit ;

$$\hat{I}' = \hat{I}'' + \beta \qquad (2.7)$$

où $\hat{I}'$ est l'image de mélange (corrigée par la composante continue) obtenue après décodage JPEG-2000. Afin de récupérer les échantillons du signal embarqué, une décomposition sur une base d'ondelette de l'image est nécessaire.

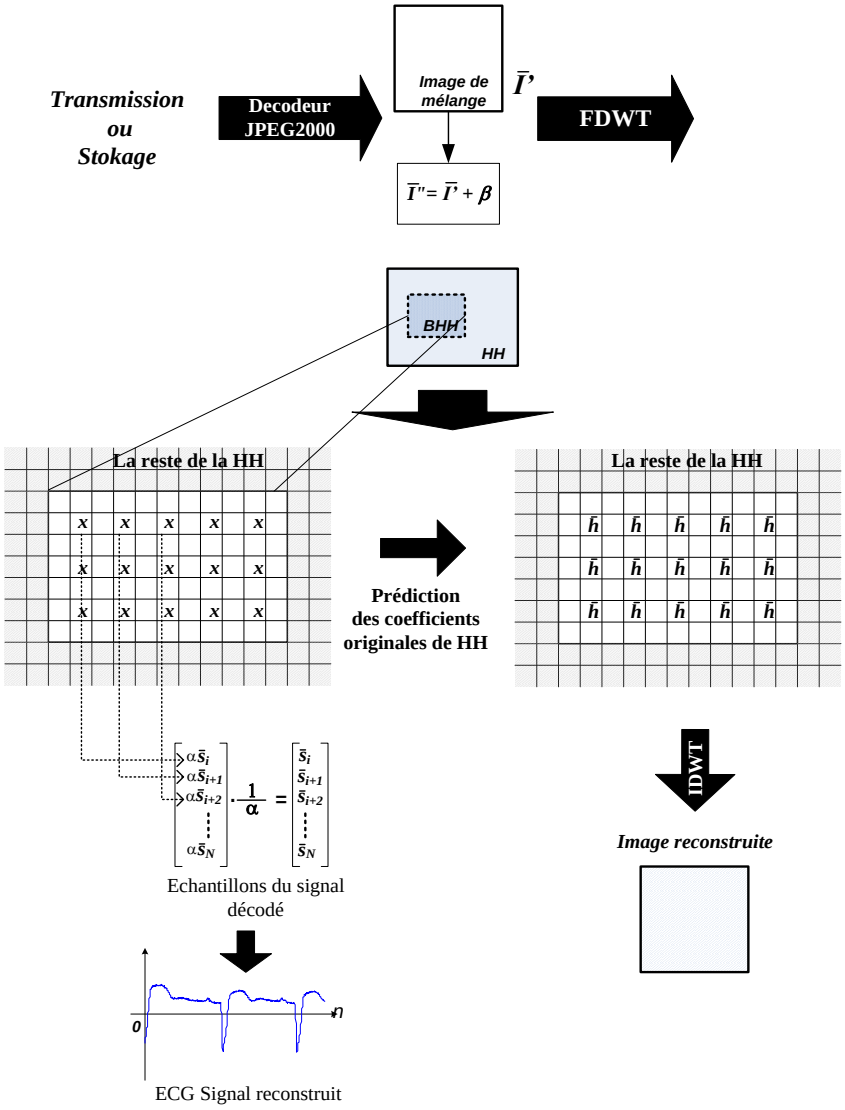


Figure 2-14 Schéma de principe du décodage par la méthode proposée

2.4.4.1 Extraction des échantillons du signal embarqué

L'extraction des échantillons du signal embarqué s'effectue ensuite à partir des colonnes de la zone *HH*, en suivant le même modèle d'insertion que nous avons utilisé lors du codage (voir la Figure 2-12). Les échantillons extraits sont ainsi divisés du paramètre de pondération α qui a été définit par l'utilisateur :

$$\hat{s}_i = \frac{1}{\alpha}.BHH(m,n) \qquad (2.8)$$

2.4.4.2 Prédiction des coefficients HH

Lorsque les échantillons du signal sont extraits de la zone *HH*, l'emplacement de chaque échantillon est remplacé par une valeur nulle. Cet emplacement qui était initialement occupé par une valeur correspondant à un détail de l'image doit être rétabli. Dans ce cas, une phase de prédiction est utilisée afin de déterminer (de manière approximative) les valeurs manquantes de la zone *HH*. (Figure 2-15)

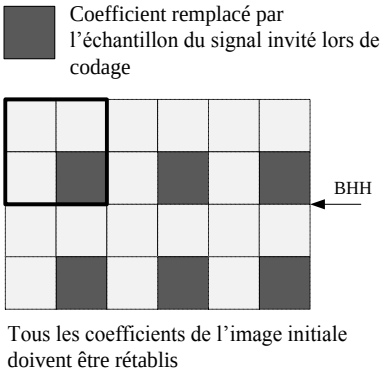


Figure 2-15 La positionnement sur la bande HH des échantillons à rétablir par prédiction

Nous avons adopté le prédicteur (MED) (voir la Figure 1-8) *Median Edge Detector*, habituellement utilisé dans le standard LOCO-I/JPEG-LS (voir la section 1.3.2.1). Ce prédicteur a la particularité d'utiliser trois coefficients voisins à celui que l'on souhaite prédire. Le modèle d'insertion que nous avons décrit à la Figure 2-12, est lié au choix du prédicteur.

Si l'on note par $\hat{h}$, la valeur prédite la règle de prédiction sera donnée comme suit :

$$\hat{h} = \begin{cases} \min(a,b) & si\ c > \max(a,b) \\ \max(a,b) & si\ c < \min(a,b) \\ a+b-c & ailleurs \end{cases} \qquad (2.9)$$

où *a*, *b* et *c* représentent respectivement, le coefficient *HH* gauche, gauche supérieur et haut, en suivant la même notation que celle de l'équation d'insertion de (2.4) telle que :

$$a = BHH(m-1, n-1),$$
$$b = BHH(m, n-1),$$
$$c = BHH(m-1, n)$$

où m est l'ensemble des nombres paires [2, w/2] $\in \mathbb{Z}^+$, n est l'ensemble des nombres paires [2, h/2] $\in \mathbb{Z}^+$.

2.4.5 Capacité de la méthode CMF

La capacité de la méthode CMF dépend de la taille de la région d'insertion et du modèle d'insertion choisi. Avec le modèle d'insertion choisi dans la section 2.4.3.2 et illustré à la Figure 2-12. La capacité d'insertion de la méthode est une fonction de *w* et *h*, exprimée selon l'équation suivante :

$$capa(w,h) = \frac{w \times h}{16} \qquad (2.10)$$

2.5 Compression multimodale dans le domaine spatial

Dans cette section, nous allons présenter deux méthodes de Compression Multimodale qui agissent dans le domaine spatial de l'image. En se basant sur le model décrit dans la section 2.3 Nous exposons deux approches d'insertions des échantillons du signal embarqué dans l'image porteuse. Ces deux variantes introduites par Fournier et al. sont respectivement appelées : l'approche non-supervisée et l'approche supervisée.

Dans le domaine du traitement d'image et de la vidéo, une image codée en RGB[7] peut être décomposée en trois plans d'échantillons. Comme nous l'avons montré dans la section 1.3.1.1 ; ceux sont le plan luma (Y), et les deux plans chromas (U & V) [**39**]. Le système visuel humain est plus sensible à des changements dans le plan Y que les changements dans les planes U et V [**8**].

De ce fait, les techniques d'insertion dans le domaine spatial utilisent le plan Y de la décomposition pour insérer les échantillons du signal embarqué.

Les approches spatiales partagent la plupart des étapes inclues dans les processus de codage et de décodage. En effet, la différence entre les approches se trouve aux niveaux de « la fonction d'insertion » et de « la fonction de séparation ».

2.5.1 Processus de codage

Le processus de codage pour les approches spatiales, de cette section, peut être généralisé comme suit:

1) Sélection d'une région d'insertion sur l'image porteuse,

[7] RGB – Red Green Blue

2) Décomposition de l'image initiale en YCbCr par une transformée RCT (voir la section 1.3.1.1) comme définit dans l'équation(1.3),
3) Conditionnement du signal embarqué,
4) Insertion des échantillons du signal embarqué dans le plan Y de l'image porteuse suivant une fonction de mélange,
5) Calcul de l'image de mélange en utilisant le plan Y modifié (Y') par une transformée RCT inverse,
6) Compression par JPEG2000 de l'image de mélange.

La Figure 2-16 illustre un schéma bloc pour les approches d'insertion dans le domaine spatiale.

2.5.2 Processus de décodage

Les étapes concernées par le processus de décodage dans les approches spatiales sont données comme suit:

1) Décompression de l'image de mélange par JPEG2000 ;
2) Décomposition de l'image mélange en YCbCr ;
3) Extraction et conditionnement inverse des échantillons du signal embarqué à partir du plan Y',
4) Reconstruction du plan Y,
5) Reconstruction de l'image par l'application d'une transformée RCT inverse en utilisant le plan Y reconstruit.

Dans la liste des étapes ci-dessus, la fonction de séparation encapsule les étapes 3 et 4. Le diagramme de bloc pour le processus de décodage est illustré sur la Figure 2-17.

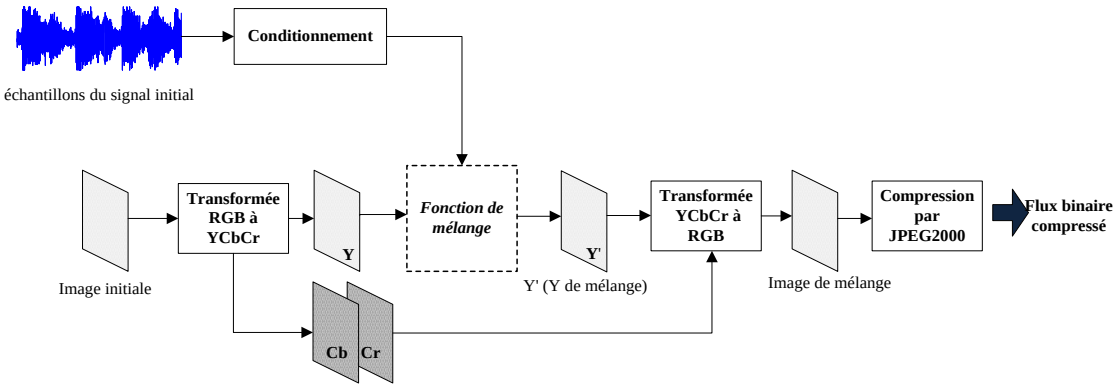


Figure 2-16 Schéma de bloc de compression généralisé pour les techniques de Compression Multimodale dans le domaine spatial

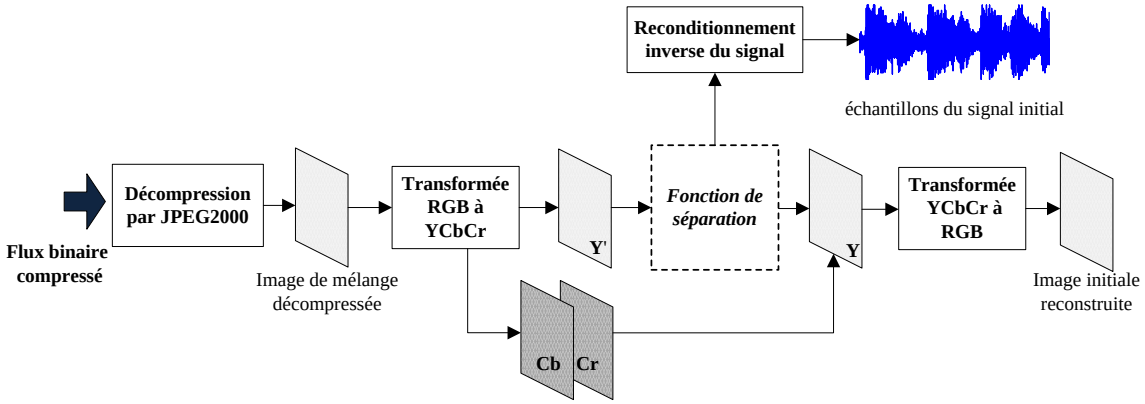


Figure 2-17 Schéma de bloc de décompression généralisé pour les techniques de Compression Multimodale dans le domaine spatial

2.5.3 Conditionnement du signal embarqué lors du codage et du décodage

Les pixels d'une image sont généralement représentés avec 8 bits par pixel (bpp). Mais d'autres signaux peuvent avoir une dynamique qui nécessite une représentation binaire supérieure à 8 bits. En conséquence, le conditionnement des échantillons du signal initial afin de réduire la dynamique du signal est inévitable. Nous utilisons une méthode très simple qui consiste d'abord à normaliser le signal puis à multiplier par la valeur maximale permissible par la représentation binaire. Ce calcul est effectué par l'équation(2.11).

$$s'(t) = \frac{s(t)}{\max\left(s(t)\right)} \times 255 \qquad (2.11)$$

Où $s'(t)$, $s(t)$ et $\max(s(t))$ sont le signal conditionné, le signal initial, et $\max(s(t))$ est la valeur maximale du signal.

Dans le processus de décodage, le conditionnement des échantillons extraits à partir du plan Y', est similaire à celui qui est utilisé dans le codage. Il est calculé comme suit :

$$s_r(t) = \frac{s'(t)}{255} \times \max\left(s(t)\right) \qquad (2.12)$$

Où $s_r(t)$ et $s'(t)$ sont respectivement le signal conditionné et le signal extrait du plan Y.

2.5.4 Approches non-supervisée

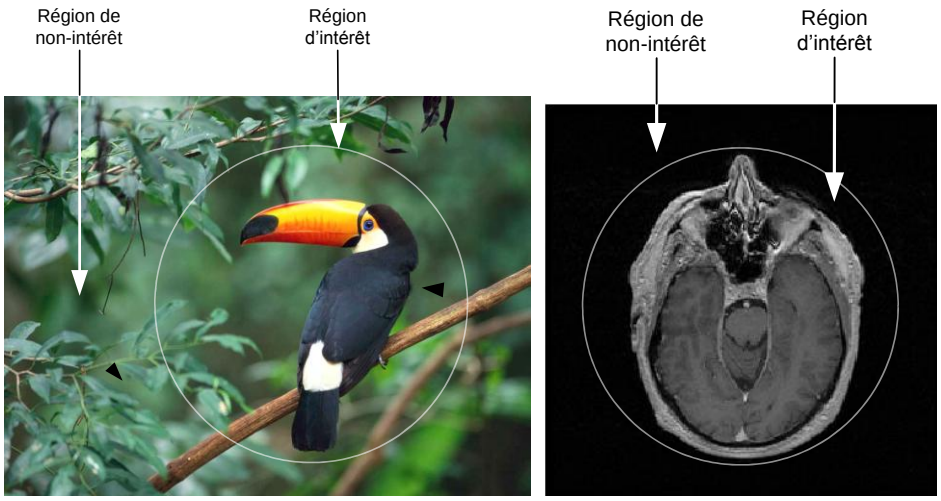
L'idée principale des approches non-supervisée sont basées sur un *a priori*. Elles présument que l'information essentielle, autrement dit la région d'intérêt (ou *ROI – Region of Interest)* sur l'image se trouve sur une région prédéterminée lors de l'acquisition.

Cela peut être illustré de la manière suivante : de l'acquisition d'une scène par un appareil photo, la région d'intérêt est intuitivement cadrée par l'utilisateur, et l'objet d'intérêt se trouve au milieu de l'image. Dans le cas d'une image biomédicale, la région d'intérêt se situe au milieu de l'image, en laissant tout le reste en « noir », ces pixels sont codés par des valeurs nulles (voir la Figure 2-18). Nous appelons cette région en dehors de la ROI, la région de non-intérêt (ou *RONI –Region of Non-Interest*). Dans les approches non-supervisée, il s'agit de l'insertion des échantillons dans la RONI suivant ce *a priori.*

2.5.4.1 Compression Multimodale par insertion spirale sur les contours (CMSS)

Dans cette méthode, les échantillons du signal embarqué sont insérés sur les contours du plan Y suivant un chemin spiral (Voir la Figure 2-20). Suivant ce chemin spiral, un échantillon sur deux est remplacé par un échantillon du signal embarqué. Dans ce cas, cette opération correspond à la quatrième du processus de codage dans le domaine spatial, décrit dans la section 2.5.1.

Lors de décodage, les échantillons du signal sont extraits à partir du plan Y suivant le même modèle spiral qui est utilisé lors de l'insertion, obtenu après la décompression par JPEG2000 et décomposition en YCbCr.



(a) (b)

Figure 2-18 Le positionnement de la région d'intérêt(ROI) et de la région de non-intérêt (a) sur une image naturelle (b) sur une image biomédicale.

Cette opération correspond à l'étape 3 sur la liste des étapes de décodage, donnée dans la section 2.5.2. Après l'extraction des échantillons du signal embarqué, les échantillons originaux du plan Y doivent être reconstruits par une étape d'estimation. Une interpolation spline cubique [**40**] en utilisant les échantillons non-altérés sur le chemin spiral, est utilisé afin d'estimer les valeurs originales de ces échantillons.

2.5.4.1.1 La capacité de la méthode CMSS

La capacité de la méthode CMSS dépend de la taille de l'image et de la taille de la région d'intérêt, définie en pixels.

Soit I est une image de taille $M \times N$ pixels et $s(k)$ est un signal de K échantillons. En l'absence d'une ROI, le nombre d'échantillons qui peut être inséré dans une image, est donné par :

$$K \leq (M \times N)/2 \qquad (2.13)$$

Dans le cas où une ROI existe, la capacité d'insertion est réduite, et définie par la relation suivante :

$$\left((M \times N)/2 - S\right) \qquad (2.14)$$

où S est la taille de la région d'insertion exprimée en pixels.

2.5.4.2 Insertion linéaire sur les contours

Un problème lié à l'approche d'insertion spirale se manifeste lorsqu'on essaye d'implémenter la méthode dans un ordinateur ou dans un système embarqué. Ce problème provient de la représentation binaire des pixels d'une image dans la mémoire, et ce phénomène est appelé « le problème de l'inconsistance du cache » ou « cache

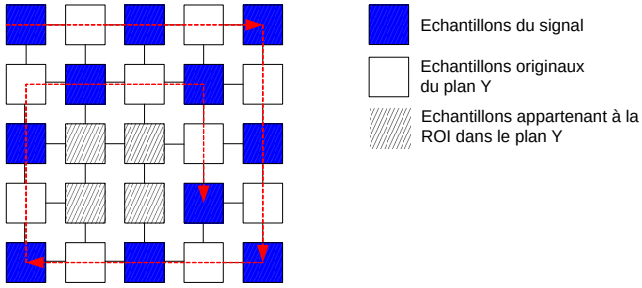


Figure 2-20 Insertion des échantillons du signal invite dans le plan Y suivant un chemin spiral. Aucune insertion n'est effectuée dans la ROI.

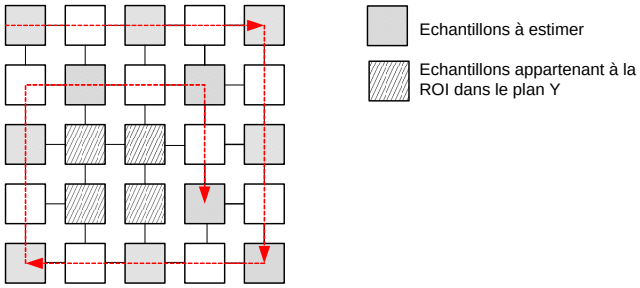


Figure 2-19 Chemin d'extraction spiral utilisé et après l'extraction des échantillons du signal embarqué les emplacements précédemment doivent être estimés afin de reconstruire l'image initiale.

trashing » en anglais. Bien que l'insertion d'un signal sur les contours d'une image en suivant un chemin spiral, soit plus naturelle pour les êtres humains, ceci n'est pas le cas pour les processeurs contemporains que l'on utilise dans les ordinateurs. Les images sont représentées dans la mémoire d'un ordinateur en une unique dimension plutôt qu'en deux. (Voir la Figure 2-22)

Cette manière d'accéder à la mémoire provoque l'inconsistance sur l'unité cache du processeur, et prolonge le temps d'exécution, car au bout de chaque ligne sur les contours, le processeur doit invalider et rafraichir le contenu du cache en chargeant une autre ligne d'une adresse lointaine dans la mémoire. Ceci devient d'autant plus évident quand l'algorithme d'insertion accède aux pixels appartenant aux colonnes verticales (Voir la Figure 2-22c).

Afin de diminuer ce phénomène, nous introduisons une dérivée de la CMSS en suivant un chemin linéaire plutôt que spirale. Dans cette méthode, les échantillons du signal embarqué sont toujours insérés sur les contours mais en suivant un chemin comme illustré sur la Figure 2-21.

Similairement à la CMSS spirale, la CMSS linéaire sur les contours, utilise un chemin d'insertion. En revanche, l'insertion linéaire définit un paramètre appelé le niveau d'insertion, que nous notons par *l*. Ce paramètre limite la largeur et la longueur du chemin d'insertion en direction horizontale et verticale, respectivement. (Voir la Figure 2-21). Les étapes lié à la reconstruction du plan Y après le décodage, restent identique à celles de la m

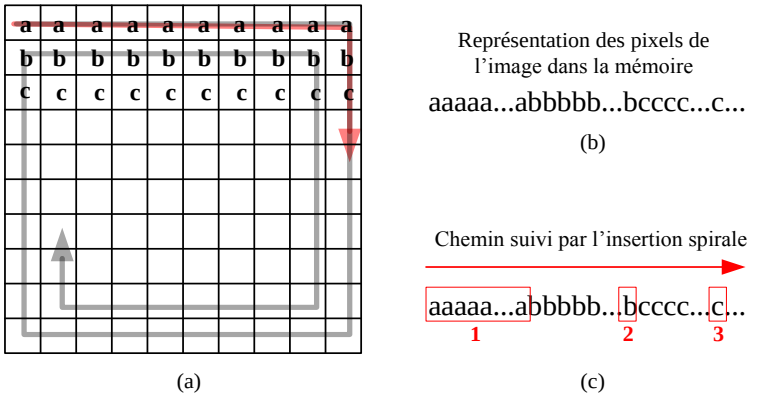


Figure 2-22 Illustration du problème lié à l'insertion spirale (a) L'image porteuse utilisée pour l'insertion(b) Représentation de l'image sur la mémoire (c) Le chemin suivi par le processeur lorsqu'il accède à la mémoire pour modifier les valeurs des pixels.

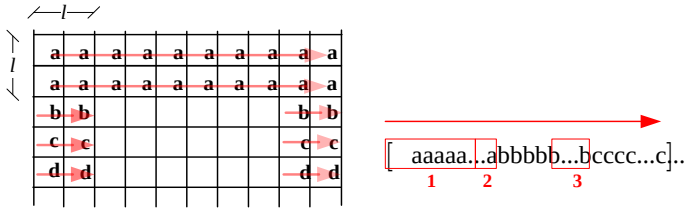


Figure 2-21 Chemin d'insertion utilisé par l'insertion linéaire. Avec cette méthode d'insertion l'accès à la mémoire sur chaque ligne de l'image et ainsi les erreurs de mémoire cache sont diminués.

2.5.4.2.1 La capacité de la méthode CMSS linéaire

Contrairement à ce que nous avons définit pour la capacité dans la CMSS spirale. La capacité de la CMSS linéaire est une fonction de la taille de l'image porteuse et le niveau d'insertion *l* choisi. Soit *I* est une image de taille $M \times N$ pixels, et *l* le niveau d'insertion choisi par l'utilisateur. La capacité peut être définie comme suit :

$$capa(M,N,l) = (l \times M) + \left(\left(N - 2l\right) \times l\right) \qquad (2.15)$$

2.6 Approche supervisée

D'après Shannon, pour la reconstruction parfaite d'un signal échantillonné, la fréquence d'échantillonnage du signal en question doit être égale au moins deux fois sa fréquence maximale, ce qui se traduit par l'expression suivante:

$$f_s \geq 2.f_{\max} \qquad (2.16)$$

où f_s est la fréquence d'échantillonnage et $f_{\max}$ est le constituant de la fréquence maximale.

Comme les images (notamment les *images naturelles*) sont des signaux *non-stationnaires* [**36**] et [**8**], la distribution des fréquences sur les dimensions n'est pas *homogène* [**8**]. De plus, de nos jours, presque tous les dispositifs d'acquisition d'image comme les appareils photographiques numériques, les scanners (sauf ceux utilisant des techniques d'échantillonnage adaptatives) utilisent un pas d'échantillonnage fixe selon la résolution de reconstruction sélectionnée (ou préférée) pendant la capture $\left(f_s >> f_{\max}\right)$. Ceci engendre certaines régions sur-échantillonnées sur l'image obtenue.

Comme indiquée par (2.16) nous pouvons en « toute sécurité » sous-échantillonner ces régions sur-échantillonnées en respectant le théorème de Shannon, puis remplacer certains pixels dans ces régions avec des échantillons appartenant à un autre signal.

Dans l'approche supervisée pour l'insertion spatiale, nous nous intéressons principalement à trouver une région d'insertion sur l'image porteuse identifiée par des basses fréquences (des régions homogènes de l'image). L'approche supervisée repose sur cette idée.

Dans le cadre du processus de codage définit dans la section 2.4.3, notre fonction de mélange consiste à :

1) Identifier les régions homogènes de l'image du plan Y,
2) Sélectionner d'une région d'insertion rectangulaire dans cette région homogène ;
3) Insérer des échantillons du signal embarqué reconditionné suivant le modèle d'insertion.

2.6.1.1 Modèle d'insertion

Le modèle d'insertion utilisée dans l'approche supervisée est illustré sur la Figure 2-23. Sur les lignes impaires du plan Y dans la région d'insertion, les échantillons du signal embarqué sont insérés suivant un pas d'un sur

deux et les valeurs sur les lignes paires sont remplacées par les échantillons du signal embarqués. En effet ce modèle d'insertion est choisi par rapport à l'étape de reconstruction de l'image. Lors du décodage, les valeurs remplacées du plan Y doivent être estimées afin de reconstruire l'image initiale. L'interpolation par splines cubiques utilisée lors de la phase de la reconstruction permet l'estimation des trois échantillons périphériques à partir d'un échantillon du plan Y. Les échantillons qui ne sont pas remplacées par le modèle d'insertion sont marqués en blanc sur la Figure 2-23.

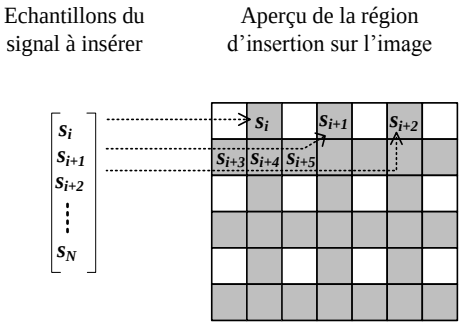


Figure 2-23 Modèle utilisé pour l'insertion des échantillons dans la zone d'insertion. Les lignes impaires contiennent les échantillons du signal, un échantillon sur deux. Les lignes paires ne contiennent que les échantillons du signal à insérer. Ce choix est fait par rapport à l'étape d'interpolation utilisée lors de la reconstruction de l'image au décodeur

2.6.1.2 Identification des régions homogènes de l'image

A fin d'identifier les régions homogènes dans l'image porteuse, une image approximative du plan Y est générée en sous-échantillonnant le plan Y initial par un facteur de deux, puis l'interpolant par les splines cubiques comme décrit par [**40**].

Ensuite, une image d'erreur est calculée entre le plan Y original et sa version interpolée :

$$Y_r = C^3\left(Y_o \downarrow 2\right) \qquad (2.17)$$

$$E_r = Y_r - Y_o \qquad (2.18)$$

Où $C^3(.)$ est un opérateur qui interpole l'image qui lui est passé en argument, $\downarrow$ est l'opérateur de sous échantillonnage, E_r est l'image d'erreur, Y_r et Y_o sont respectivement l'image interpolée et initiale. Sur l'image d'erreur, nous visualisons les régions qui sont bien estimée par l'interpolation. Elles admettent des valeurs nulles ou quasi nulles (Voir la Figure 2-24)

Comme nous l'avons évoqué dans la section précédente, cette étape qui consiste à déterminer une image d'erreur est liée à l'étape de reconstruction de l'image et au modèle d'insertion illustré sur la Figure 2-23. En effet notre problème de trouver une région d'insertion de manière à ce qu'il y ait la moindre dégradation possible sur les images reconstruites peut être reformulé ainsi :

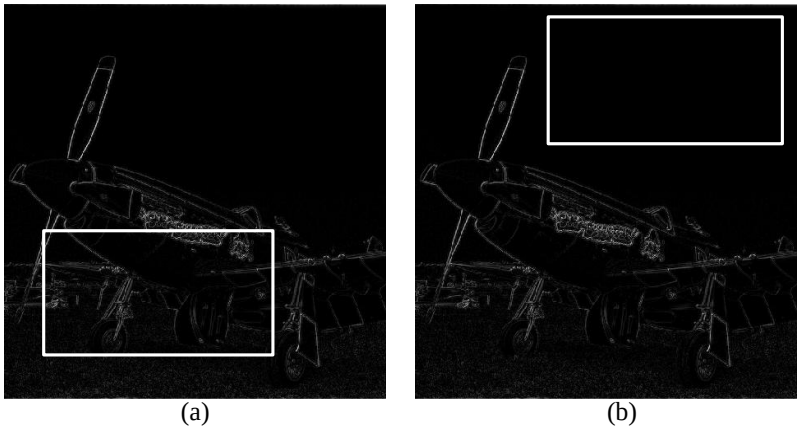
- Trouver la région sur l'image, de manière à ce que l'erreur entre la version sous-échantillonnée puis interpolée de cette région et la version originale soit la plus faible possible

(a) (b) (c)

Figure 2-24 (a) Le plan Y exprimée en niveau de gris de l'image initiale, (b) le plan Y initial sous-échantillonnée puis interpolée et (c) l'image d'erreur de l'interpolation.

Autrement dit, soit Q est une région qui minimise cette erreur et le Y_o^Q la partie sur Y_o qui correspond à cette région. Cela peut être formulé comme suit :

$$\arg\min_{\mathbf{x}\in Q}\left(C^3\left(Y_0^Q \downarrow 2\right) - Y_0^Q\right) \qquad (2.19)$$



(a) (b)

Figure 2-25 Le choix de la région d'insertion pour l'approche supervisée, Nous observons à gauche (a) une mauvaise sélection et à droite (b) une bonne sélection pour la région d'insertion

où $C^3(.)$ est un opérateur qui interpole l'image qui lui est passé en argument, $\downarrow$ est l'opérateur de sous échantillonnage. **X** est un vecteur comprenant les coordonnées $[x_0, y_0, x_1, y_1]$ du rectangle choisi comme la région d'insertion.

2.6.1.3 Fonction de mélange

Après le calcul de l'image d'erreur, l'utilisateur choisit une région d'insertion sur cette image. La région sélectionnée, comme décrit dessus, doit être dans une région identifiée par des valeurs nulles ou quasi-nulles. Sur la Figure 2-25, nous illustrons à gauche une mauvaise région d'insertion, comme les échantillons appartenant au plan Y ne seront pas bien estimés par lors de reconstruction, cette partie de l'image est défavorable. De l'autre coté, à droite sur la même figure, la région sélectionnée sur l'image d'erreur contient des valeurs nulles, c'est un bon choix pour la méthode.

En suite, la fonction d'insertion insère les échantillons du signal embarqué dans cette région suivant le modèle d'insertion. La Figure 2-26 illustre la fonction d'insertion pour l'approche supervisée.

2.6.2 Détection automatique de la région d'insertion pour l'approche supervisée

De manière générale, comme nous l'avons montré par l'équation (2.19) le problème de trouver une région rectangulaire dans une région *Q* fermée est un problème d'optimisation et dans la littérature, il est connu sous le nom de *bin packing* [**41**]. Comme nous utilisons une région rectangulaire alignée avec les axes sur lesquels est définit l'image. Nous nous intéressons plutôt à un cas particulier de ce problème : Trouver le plus grand rectangle avec des axes alignés (LR[8]) dans un polygone qui peut être définie par un nombre limité de *n* vertices, est un problème d'optimisation géométrique qui peut être classé dans la catégorie des problèmes d'inclusion de polygone. [**42**].

Malgré son importance dans la pratique, dans la littérature, la plupart des travaux pour trouver la LR sont limités aux polygones orthogonaux [**43**] [**44**]. Amenta *et al*. ont étendu le problème d'inclusion aux polygones convexes [**45**]. Amenta et al. ont montré que le problème de trouver la LR dans un polygones convexes peut être trouver en temps linéaire en le reformulant comme un problème de programmation convexe. McKenna *et al.* utilise l'approche *divide-and-conquer* pour trouver la LR dans un polygone convexe en temps $O(n \log^5 n)$ [**43**].

Nous utilisons une technique beaucoup plus simple et moins complexe, nommée la technique de décomposition quadtree, afin de fournir une « bonne » approximation de la LR. C'est la technique de décomposition en quadtree.

[8] LR – Largest area axis-parallel Rectangle.

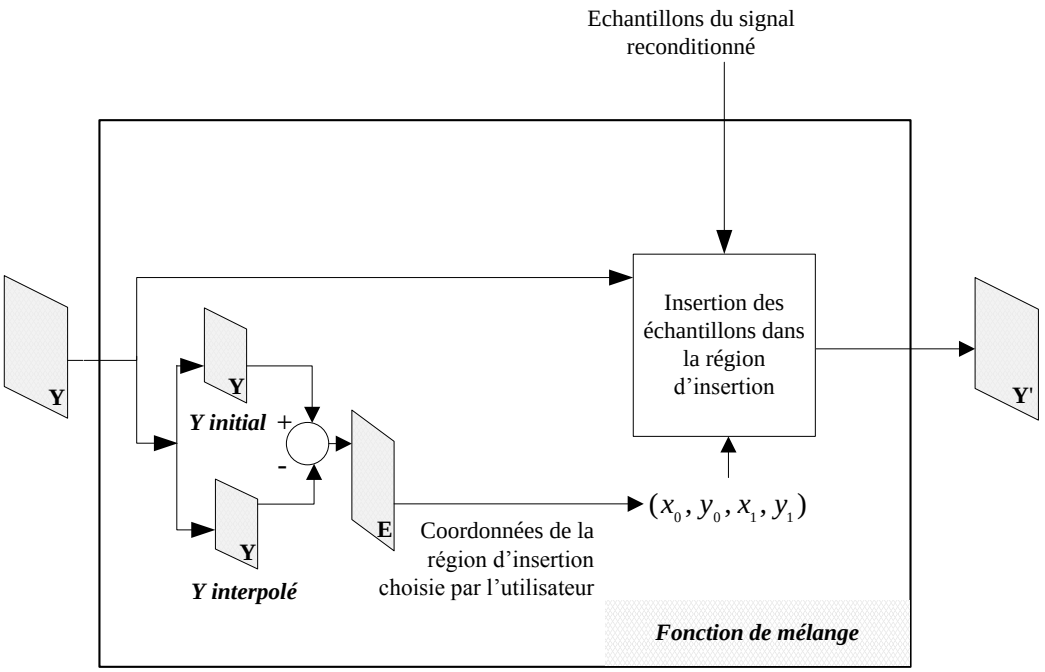


Figure 2-26 Schéma de la fonction de mélange utilisée dans l'approche supervisée. Cela remplace la partie « fonction de mélange » de la Figure 2-16. Dans la fonction de mélange l'utilisateur choisit une région d'insertion sur l'image d'erreur, en suite les échantillons sont insérés dans le plan Y suivant le modèle d'insertion.

2.6.2.1 Détection de la région d'insertion par Quadtree

Dans cette section, nous présentons une méthode de détection pour la sélection automatique de la région d'insertion dans l'approche supervisée. Nous allons appeler cette méthode la CMSQ (Compression Mulitmodale Supervisée Quadtree)

Quadtree est une structure de données en arbre qui partitionne un espace donné à deux dimensions jusqu'à quatre quadrants de régions [**46**]. Dans une *quadtree*, chaque nœud a au plus quatre nœud fils, d'où le nom *quadtree*.

Les quadtree sont souvent utilisés dans les bases de données spatiales et en infographie. Les quadtree sont aussi utilisés pour représenter une image qui est formée de 2^n x 2^n pixels où la racine représente l'espace entier, et chaque sous-nœud représente une sous-région sur le nœud père. Un exemple de cette structure est représenté sur la Figure 2-28.

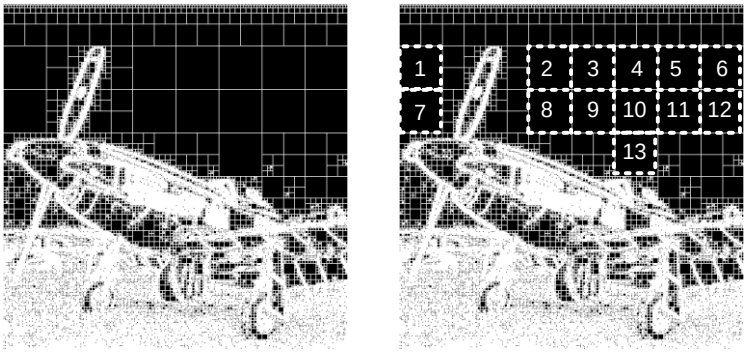
L'algorithme de quadtree décompose un nœud en quatre sous-nœuds si la condition de décomposition est vérifiée. L'algorithme continue à diviser un nœud soit jusqu'à ce qu'il ne reste plus de nœuds à diviser ou bien jusqu'à ce que le critère d'arrêt (l'inverse de la condition de division) soit vrai.

Dans notre méthode, nous utilisons un critère de division comme suit :

$$t \geq 255 \times \left(\max(\Lambda) - \min(\Lambda)\right) \qquad (2.20)$$

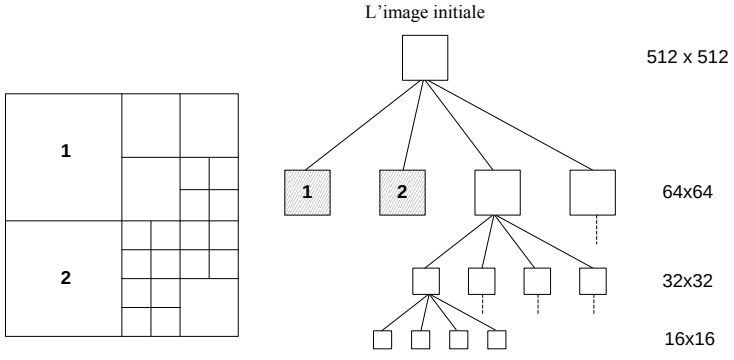
Où t est un paramètre de seuil, max(Λ) et min(Λ) sont la valeur maximum et minimum de pixel dans le nœud Λ. Le paramètre t peut varier d'une image à l'autre et est considéré comme un paramètre qui caractérise l'image en question. Nous allons étudier l'impact du choix de ce paramètre dans le chapitre suivant, dans la partie présentant les résultats.

Afin de pouvoir détecter l'insertion, l'image-erreur est décomposé en sa représentation quadtree, ensuite, seuls les nœuds successifs de la plus grande dimension et qui n'ont pas de nœuds fils directs, sont sélectionnés pour former une région rectangulaire d'insertion, comme dans la Figure 2-28. La Figure 2-27, illustre les régions d'insertion détectées par l'algorithme de détection. Nous constatons que ce résultat donne une bonne



(a) (b)

Figure 2-27 (a) La décomposition en quadtree de l'image d'erreur (b) les nœuds voisins de même taille ont été choisis (64x64 pixels) pour construire la LR_0 comme région d'insertion.



(a) Representation spatiale (b) Arborescense

Figure 2-28 Un quadtree, représentant la repartitionnement d'une image dans des régions de tailles inférieures.

approximation de la région d'insertion qui a été choisie par l'utilisateur sur la Figure 2-25. De plus il est possible de conserver plusieurs zones d'insertions. Voir à la même figure, les deux zones à gauche (1 et 7) et la zone (13).

En comparant avec l'image de droite de la Figure 2-25(b), il y a bien cohérence de la méthode et en plus de zones d'insertion supplémentaires ont été mis en évidence.

2.7 Conclusion

Nous avons vu dans ce chapitre que la Compression Multimodale est un processus de compression conjoint de différentes séries de données acquises selon différentes modalités, en utilisant un seul codec (e.g. JPEG 2000). Nous avons également vu que la Compression Multimodale peut être appliquée selon des modes différents (e.g. supervisé ou non-supervisé) et dans des domaines différents (e.g. spatial ou fréquentiel). Ces techniques seront évaluées dans le chapitre suivant sur des images et des signaux de types différents y compris sur des données médicales. Les performances seront présentées en termes de courbes débit-distorsion.

Chapitre 3

Evaluation des performances des méthodes de Compression Multimodale

3.1 Introduction

Comme nous l'avons signalé précédemment, ce chapitre fera l'objet d'une évaluation des performances de quelques variantes de la Compression Multimodale. Pour ce faire, nous avons utilisé, entre autres, la base de données de « *Kodak* », téléchargeable sur internet, ainsi que MeDEISA « *Medical Database for the Evaluation of Image and Signal Processing Algorithms* », disponible sur www.medeisa.net. La présentation des résultats sera argumentée par des analyses et des études comparatives objectives. En utilisera ainsi des critères du type « PSNR » pour évaluer la qualité des images et le critère PRD pour évaluer la qualité des signaux.

3.2 Critères d'évaluation

3.2.1 Qualité de l'image reconstruite

La mesure de la qualité des images reconstruites est essentielle afin de fournir des données quantitatives sur la fidélité de la compression effectuée.

Dans la littérature, il existe plusieurs méthodes pour mesurer la qualité de la reconstruction d'image [**47**] [**48**] [**49**] [**50**] et [**51**].

Pour mesurer, la qualité des images reconstruites dans les méthodes de Compression Multimodale que nous avons vues au chapitre précédent, nous utilisons principalement le PSNR (*Peak singal noise ratio*). Le PSNR est le rapport du signal crête à crête à la valeur quadratique moyenne du bruit. Il est défini comme suit :

$$PSNR = 10\log_{10}\left[\frac{d^2}{(1/N)\sum_i\sum_j\left(I_{ref}(i,j)-I_{dec}(i,j)\right)^2}\right] \qquad (3.1)$$

Où, I_{ref} est l'image de référence et I_{dec} l'image comparée. Dans l'équation (3.1) « d » est la valeur maximale permissible par le codage de l'image, dans le cas d'une image codée en 8 bits, ceci est égale à $d = 8$. Dans le domaine de compression d'image, un PSNR d'un niveau de l'ordre de 35dB apparaît être au niveau de qualité acceptable.

3.2.2 Métriques de qualités subjectives

Pour la qualité des images reconstruites, les tests utilisant d'autres métriques de qualité sont aussi effectués.

3.2.2.1 Structural Similarity Index (SSIM)

La SSIM, comme son nom l'indique, est une mesure de similarité entre deux images. Elle mesure la qualité visuelle la similarité de *structure* entre les deux images, plutôt qu'une différence pixel à pixel comme le fait le PSNR. L'hypothèse sous-jacente est que l'œil humain est plus sensible aux changements dans la structure de l'image. [**52**] [**53**]

La métrique SSIM est calculée sur plusieurs fenêtres d'une image. La mesure entre deux fenêtres *x* et *y* de même taille $N \times N$ est calculé comme suit :

$$SSIM(x,y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)}$$

où:

- μ_x est la moyenne de *x*.
- μ_y est la moyenne de *y*.
- σ_x^2 est la moyenne de *x*.
- σ_y^2 est la moyenne de *y*.
- σ_{xy} est la covariance de *x* et *y*.
- $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ sont deux variables destinées à stabiliser la division dans le cas où le dénominateur est très faible.
- L est la valeur maximale permissible par le codage d'un pixel, soit 255 pour des images codées sur 8 bits.

- $k_1 = 0,01$ et $k_1 = 0,03$ par défaut.

Pour l'évaluation de qualité d'une image, la formule précédente est appliquée sur la luminance uniquement. Typiquement, les grandeurs sont calculées sur des fenêtres de taille 8x8.

3.2.2.2 Visual Information Fidelity (VIF)

La VIF est une mesure de qualité qui quantifie l'information présente dans une image de référence, et elle donne un indice quantitatif de la partie recouvrable de cette information sur sa version déformée [**54**].

3.2.2.3 Noise quality measure (NQM)

Dans la NQM, différemment des autres métriques de qualité, l'image reconstruite est considérée comme étant une image subie d'une dégradation fréquentielle suivie par un rajout du bruit. Elle différencie ces deux sources de dégradation, en se basant sur le fait que ces deux sources ont des effets différentes sur le système visuel humain. [**55**]

3.2.2.4 Weighted signal to noise ratio (WSNR)

Par définition ; la WSNR est la SNR calculée entre la puissance du signal pondéré et la puissance du bruit pondéré. Dans le cadre de la mesure de la qualité des images reconstruites, la fonction de pondération est appelée CSF (constrast sensitivity function). La CSF est une représentation linéaire et indépendante de la translation, de du système visuelle humaine (HVS) [**55**].

3.2.3 Qualité du signal reconstruit

Le PRD (pour *Percent Residual Difference*) est une mesure qui exprime le pourcentage d'erreur entre le signal original et le signal reconstruit. Il s'exprime en pourcentage, comme suit :

$$PRD(\%) = \sqrt{\frac{\sum_{k=1}^{K}(x_k - y_k)^2}{\sum_{k=1}^{K}(x_k)^2}} \times 100 \qquad (3.2)$$

Où *x*, *y* et *N* sont respectivement le signal original, le signal ayant subi une transformation et le nombre de points du signal. En général sur les signaux reconstruits une qualité de PRD(%) $\leq$ 20% est acceptable.

Nous utiliserons cette mesure afin de comparer le signal inséré avec sa version extraite après l'étape de compression-décompression.

3.3 Méthodologie d'évaluation de performances des méthodes

Pour chaque méthode que nous avons présentée ; nous avons suivi la procédure de test suivante, pour évaluer les performances :

1) Sensibilité de la méthode : variation des performances en fonction du choix des paramètres.
2) La qualité de la reconstruction de l'image en utilisant la CM est comparée avec la compression de l'image seule. Dans ce contexte :
    a. Pour un bitrate de compression donné, le PSNR de la compression/décompression seule de l'image par JPEG2000 est comparée avec le PSNR de l'image reconstruite avec la CM.
    Lors de la présentation des résultats, nous allons donner les valeurs suivantes :
        - le PSNR-direct : PSNR calculé entre l'image obtenue par la compression /décompression de l'image seule par JPEG2000 et l'image originale,
        - le PSNR-méthode : PSNR calculé entre l'image obtenue par la compression/décompression de l'image par la CM et l'image originale.
    b. Pour le même bitrate, la qualité de la reconstruction du signal en termes de PRD est évaluée.

Comme les méthodes proposées induisent une dégradation à la fois sur les images et les signaux reconstruits, nous devons avoir un bitrate de compression par JPEG2000 où les qualités de reconstruction de l'image et du signal sont acceptables en termes de PSNRs et de PRDs.

3.4 La base de données de tests

Afin d'illustrer les performances des méthodes que nous avons présenté dans le chapitre précédent, nous avons utilisé la base de données des images sans perte de Kodak [**56**]. Toutes ces images sont codées en 8 bits RGB et sont de taille 512x512 pixels. Parmi lesquelles, celles sur la Figure 3-1.

Figure 3-1 Les images de tests utilisées dans les simulations, toutes les images sont de taille 512x512 pixels, et codée en 8bit RGB.

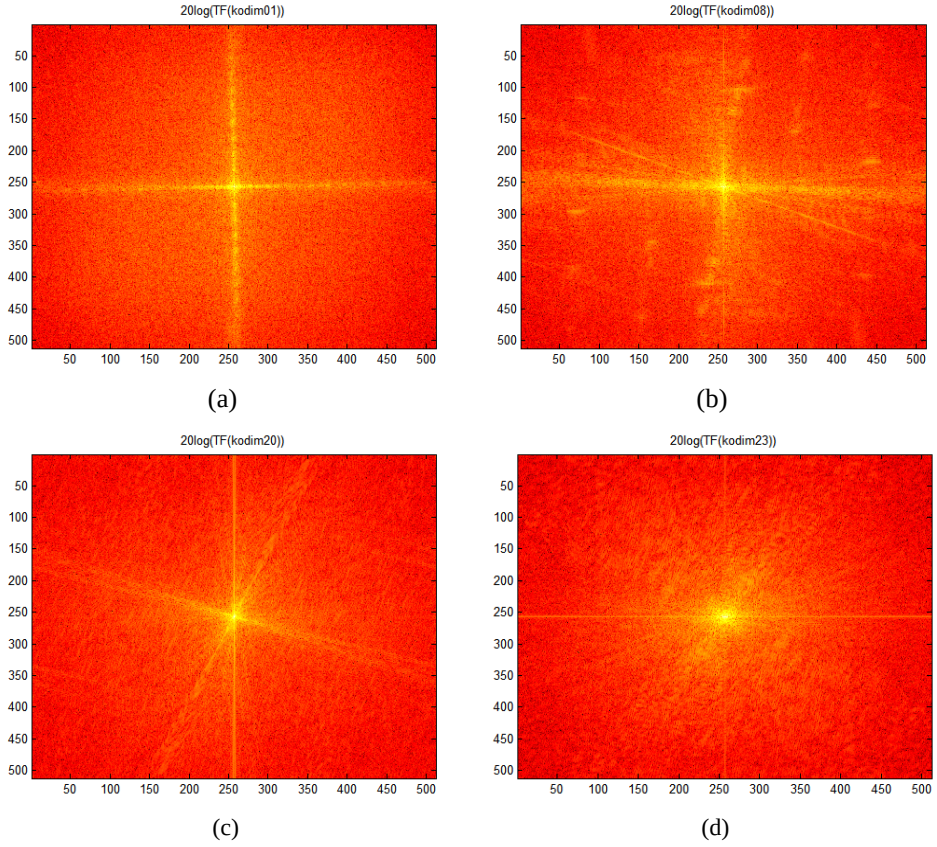
Comme nos méthodes dépendent fortement de la distribution spatio-fréquentielle de l'image, les images de la Figure 3-1 ont des distributions différentes : les images kodim01 et kodim08 sont caractérisées par la présence de hautes fréquences dans pratiquement toute l'image. Tandis que les images kodim20 et kodim23 sont caractérisées par des grandes zones de basses fréquences.

Cet aspect se confirme en observant la Transformée de Fourrier (TF) de ces images et la décomposition en ondelettes de premier niveau sur les figures 3-2(a), 3-2(b), 3-2(c) et 3-2(d), respectivement.

L'image kodim01 (Figure 3-2a) est texturée (les briques), ce qui explique la présence des hautes fréquences dans les directions verticales et horizontale. La TF de l'image kodim08 (Figure 3-2b) est aussi caractérisée par des hautes fréquences dues à la texture présente sur les immeubles. Ceci est confirme via la décomposition en

ondelettes. Nous pouvons ainsi visualiser les plans *HH* de la décomposition en ondelette des images kodim01 et kodim 08 sur la Figure 3-3a et 3-3b, respectivement. L'existence d'une forte concentration de hautes fréquences sur la texture de l'image kodim 01 et kodim08 est affirmée en examinant la Figure 3-3a et 3-3b, respectivement.

Les images kodim20 et kodim23 ont peu de variations de luminance, ce qui est visible sur leurs TFs. Dans les deux TFs, les fréquences principales sont regroupées autour du centre. Sauf sur la TF de la kodim20, nous visualisons une faible présence de hautes fréquences sur l'axe principal horizontal, ceci est due au passage brusque de la région appartenant au ciel, à celle de l'avion (la Figure 3-1). Nous pouvons le confirmer en examinant le plans *HH* de la décomposition en ondelette de l'image kodim20 (Figure 3-3c). Le plan *HH* de la décomposition en ondelette de l'image kodim23 (Figure 3-3d) contient des valeurs presque nulles, sauf dans les emplacements qui correspondent aux contours des têtes des perroquets



(a) (b)

(c) (d)

Figure 3-2 Les Transformées de Fourrier des images de test,(a) kodim01,(b) kodim08, (c) kodim20 et (d) kodim23. Le code couleur montre la puissance des échantillons de fréquences. (De rouge foncé vers le jaune claire dans l'ordre croissant).

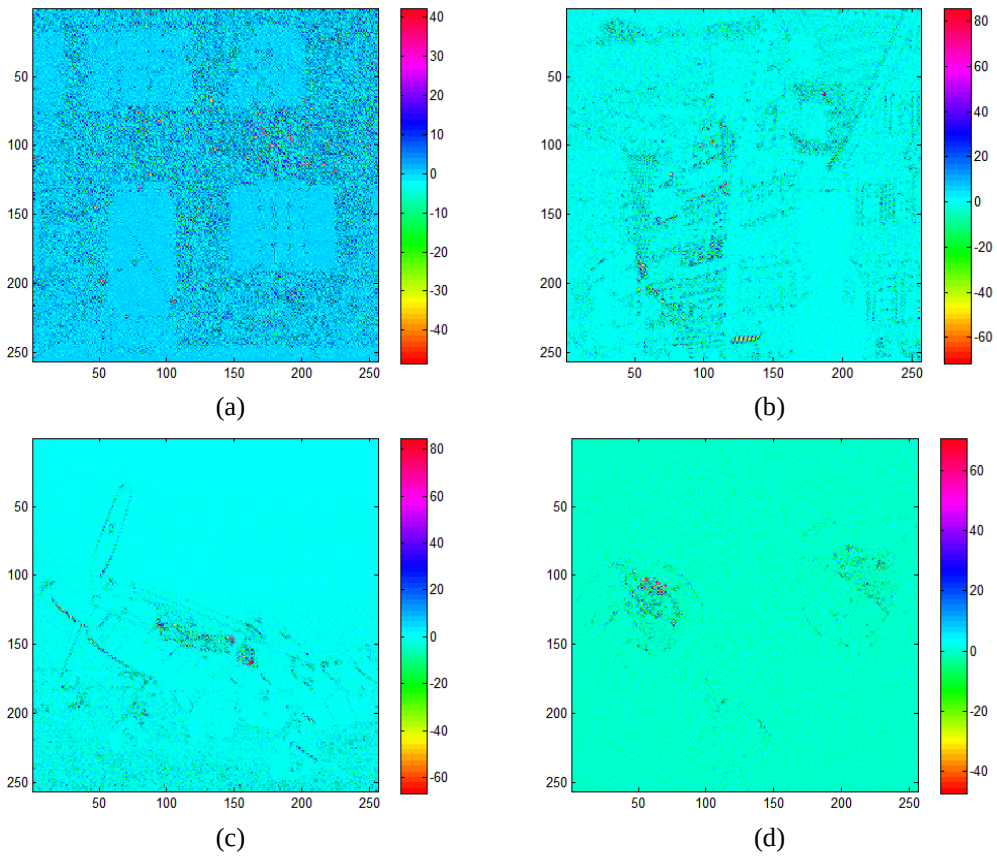
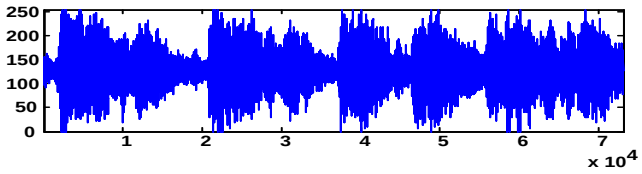


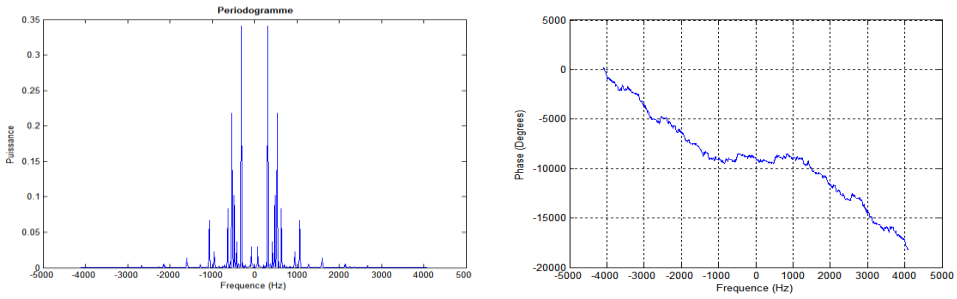
Figure 3-3 Les plans HH de la décomposition en ondelette des images de test présentées sur la Figure 3-1. (a) kodim01 (b) kodim08 (c) kodim20 et (d) kodim23, respectivement.

En ce qui concerne le signal test à insérer. Nous avons effectué plusieurs expériences avec différents types de signaux (bandes de films, signaux médicaux, etc …). Pour illustrer les performances de nos méthodes, nous avons utilisé le signal audio « Handel »[9] (Figure 3-4). C'est un signal à un canal et est généré par *η-law* à un taux d'échantillonnage de 8KHz. De plus, il possède une plage dynamique de 8-bit, normalisée entre [-1, 1]. Les spectres d'amplitude et de phase de Handel sont illustrés sur la Figure 3-4(b) et (c), respectivement. Le spectre d'amplitude du signal Handel montre que l'information essentielle se trouve dans l'intervalle [-1,5 +1,5] KHz.

[9] Handel est de la base donnée Matlab Mathworks ©



(a)



(b) (c)

Figure 3-4 Le signal d'audio "Handel" a été utilisé à travers des tests (a). La densité spectrale du signal "Handel"(b) et la phase de sa TF (c).

3.5 Evaluation des performances de la méthode CMF

Les performances de la méthode de la CM basée sur l'insertion des échantillons d'un signal embarqué dans le domaine fréquentiel (Voir la section 2.4), sont étudiées dans cette section.

La méthode CMF dépend de deux paramètres :

1) Le paramètre α,
2) La région d'insertion.

Comme nous l'avons vu à la section 2.4.3.1, le paramètre α est utilisé pour pondérer les échantillons du signal embarqué normalisé (voir l'équation(2.3)).

Pour le second paramètre, c'est à dire la région d'insertion, nous choisissons la totalité de l'image, pour mieux étudier les effets dégradants de la méthode sur les images reconstruite. Bien entendu, la qualité des images reconstruites sera plus élevée si nous avions choisi une région d'insertion sur une partie de l'image.

Dans le cas où la région d'insertion correspond à la totalité de l'image et d'après l'équation(2.10), la capacité de la méthode est alors limitée à :

$$\frac{w \times h}{16} = \frac{512 \times 512}{16} = 16384$$

où *w* est la largeur de l'image et *h* est la longueur de l'image.

3.5.1 Etude du paramètre α

Un bon choix du paramètre α est important, car comme les échantillons sont insérés dans la partie *HH* de la décomposition dyadique en ondelette, une faible valeur va provoquer la troncature des échantillons lors de codage JPEG2000. En revanche, une valeur de α élevée, provoquera l'augmentation de la composante continue β (voir la section 2.4.3.3) et causera plus de dégradation sur les images reconstruites. Afin de trouver un bon compromis pour le paramètre α, nous avons étudié l'évolution du PSNR et du PNR en fonction de l'évolution des valeurs de α en se fixant un bitrate de compression faible pour le codeur JPEG2000 et sachant que JPEG2000 réservera une faible précision binaire pour les coefficients *HH.* A ce bitrate, l'importance du choix du paramètre α est plus visible. Ceci car les échantillons du signal embarqué insérés dans le plan *HH*, seront davantage dégradés.

Avec la configuration de test décrite ci-dessus, nous avons exécuté la procédure de test sur chacune des images de la Figure 3-1. Sur la Figure 3-5 nous présentons les résultats du test α. Sur ces résultats nous remarquons qu'un choix de α élevé provoque une grande dégradation sur les images reconstruites. En revanche, pour la qualité de reconstruction du signal, le PRD a tendance de diminuer et provoque peu de changement sur le PRD après une certaine valeur de α. Nous observons pour une valeur α = 30 choisie, que le signal a une qualité de reconstruction dans une plage acceptable (PRD $\leq$ 15%). Pour la qualité de l'image reconstruite, une valeur de α fixée à 30 provoque peu de dégradation. Par conséquent, nous avons fixé la valeur de α à 30 pour la suite des expérimentations.

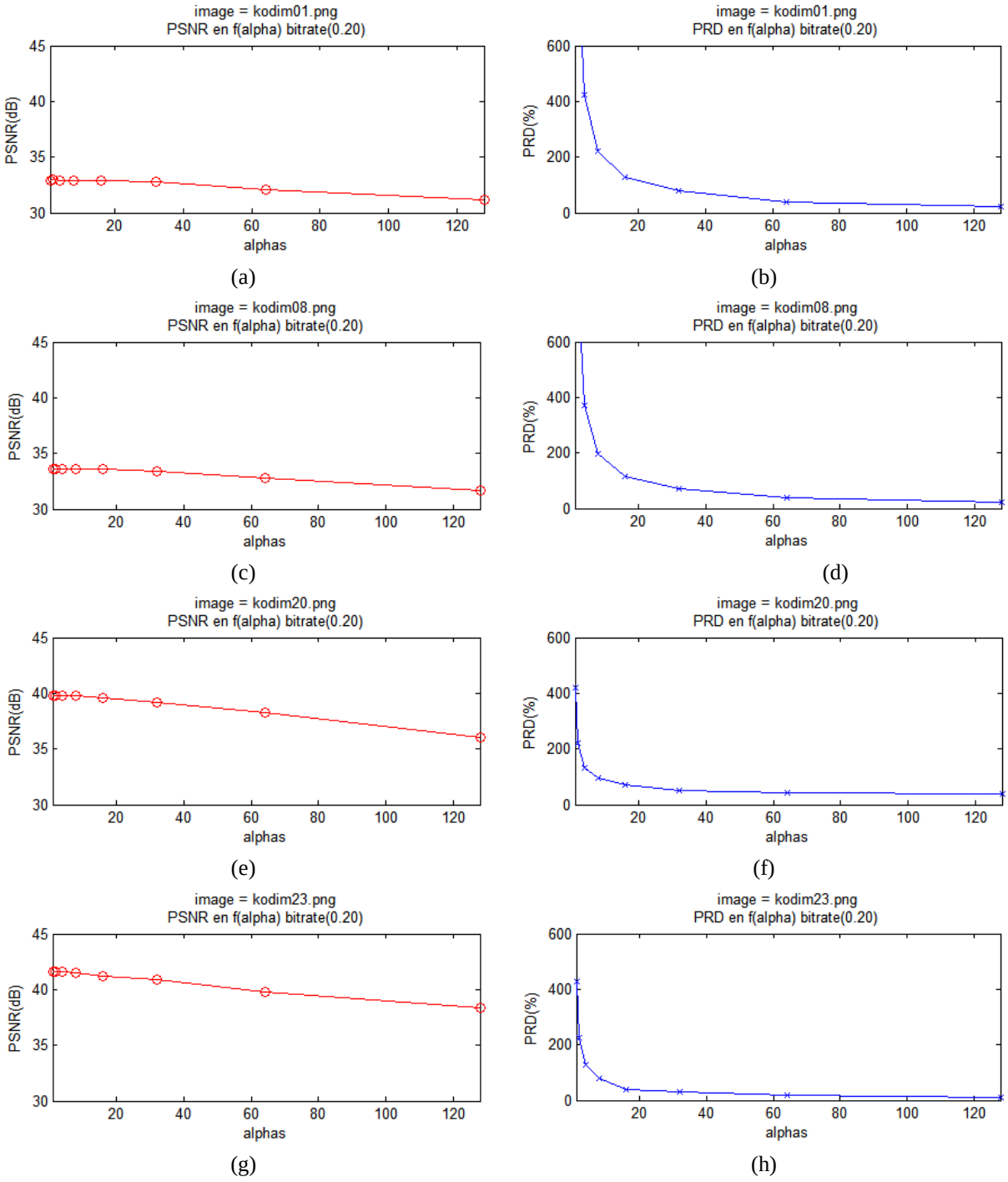


Figure 3-5 Résultats de tests de l'évolution du PSNR et du PRD en fonction du paramètre **α** choisi. (a)-(b) kodim01, (c)-(d)- kodim08, (e)-(f) kodim20, (g)-(h) kodim23.

3.5.2 Performances de la méthode CMF[10]

Sur la Figure 3-6, nous présentons une étude comparative entre le PSNR-direct, le PSNR-méthode et le PRD pour des différents bitrates utilisés dans la compression par JPEG2000. Nous remarquons que pour tous les bitrates, le PSNR-méthode reste dans la plage acceptable (voir la section 3.2). Pour les bitrates supérieurs à 0.5 bpp, nous avons obtenu un signal reconstruit d'une qualité acceptable (PRD $\leq$ ~20%).

Toutefois nous observons une meilleure qualité en matière de PSNR sur les images kodim20 et kodim23 que sur celles de kodim01 et kodim08. A ce stade, nous pouvons dire que pour un bitrate fixé à environ 0.5 bpp, la CM devient avantageuse. En d'autres termes, pour cette valeur de bitrate, nous pouvons compresser le signal dans l'image avec une dégradation acceptable (pour le signale et l'image)

[10] Compression Multimodale dans le domaine Fréquentiel

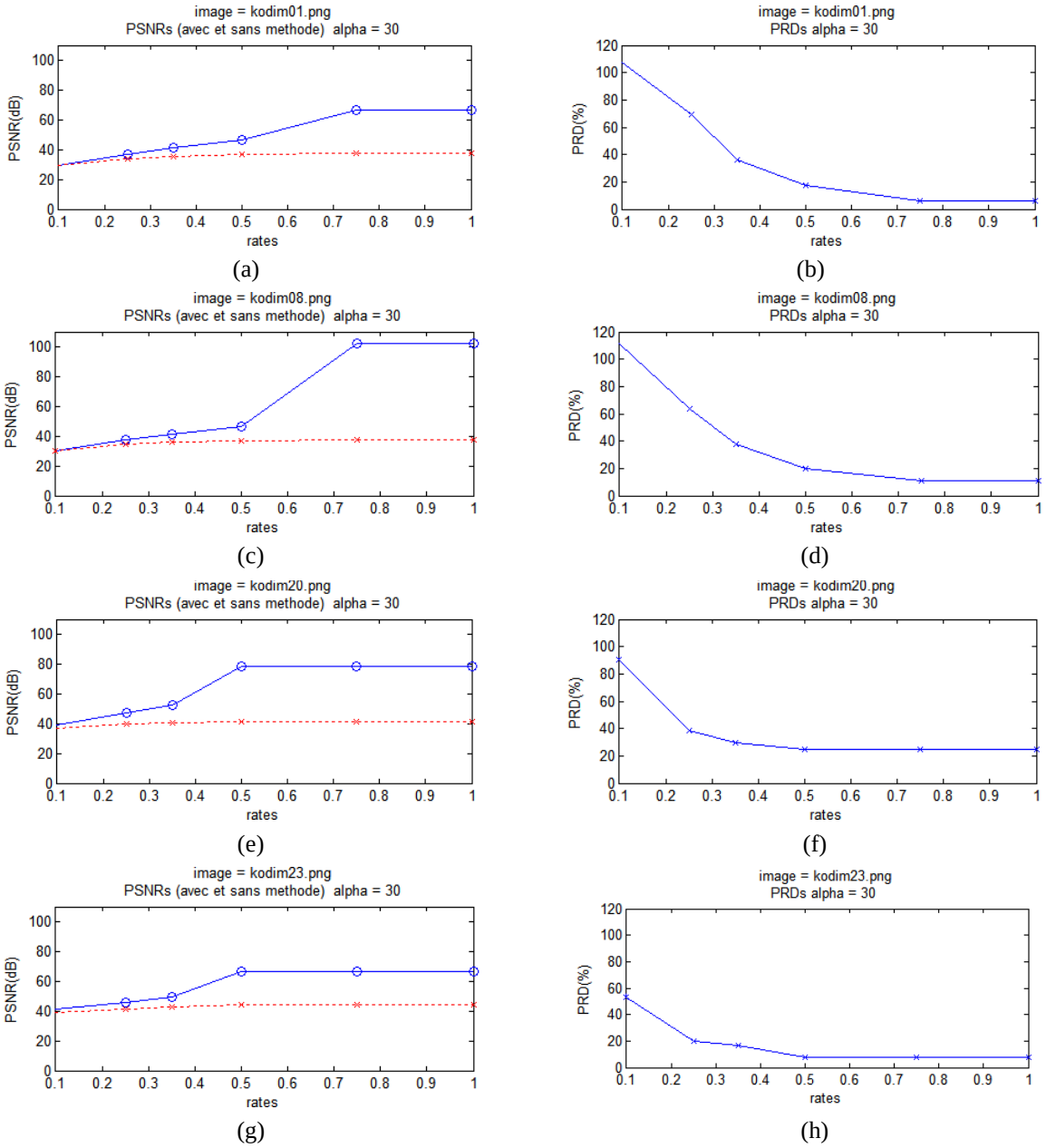
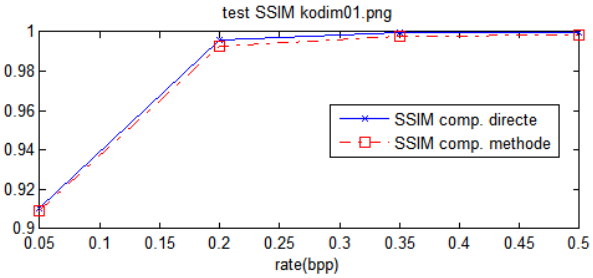
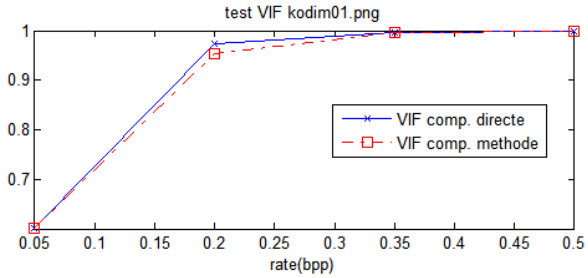


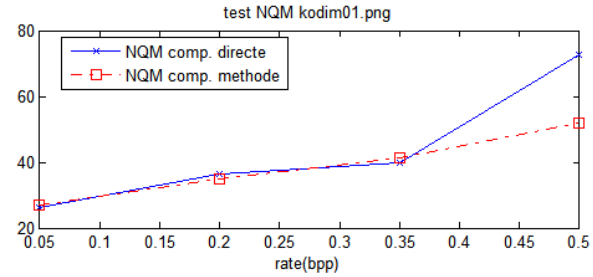
Figure 3-6 Evaluation de la méthode CMF avec la compression seule de l'image par JPEG2000 pour des différents bitrate choisis. (a)-(b) kodim01, (c)-(d)- kodim08, (e)-(f) kodim20, (g)-(h) kodim23.
Sur la Figure (o) dénote le PSNR direct et (x) le PSNR méthode



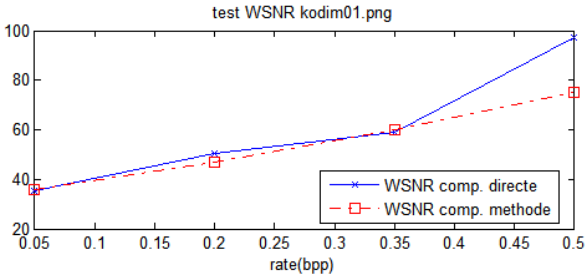
(a)



(b)

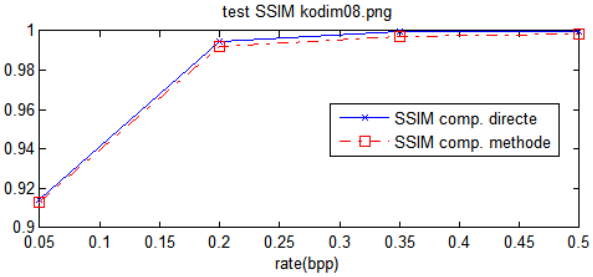


(c)

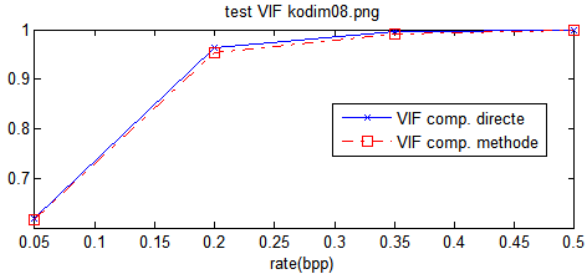


(d)

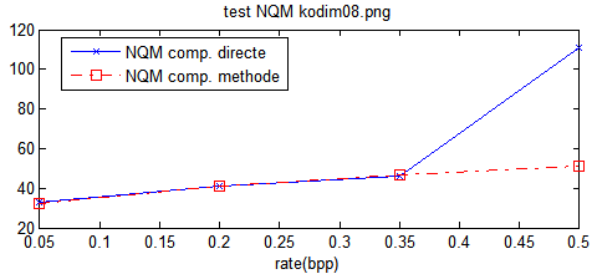
Figure 3-7 Evaluation de la méthode CMF appliquée à kodim01.png pour des différents bitrate choisis, en utilisant les métriques de qualité subjonctive (a) SSIM (b) VIF (c) NQM (d) WSNR



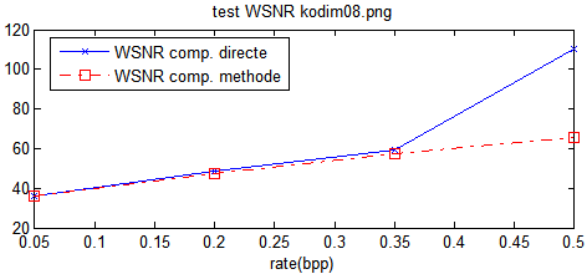
(a)



(b)



(c)



(d)

Figure 3-8 Evaluation de la méthode CMF appliquée à kodim08.png pour des différents bitrate choisis, en utilisant les métriques de qualité subjonctive (a) SSIM (b) VIF (c) NQM (d) WSNR

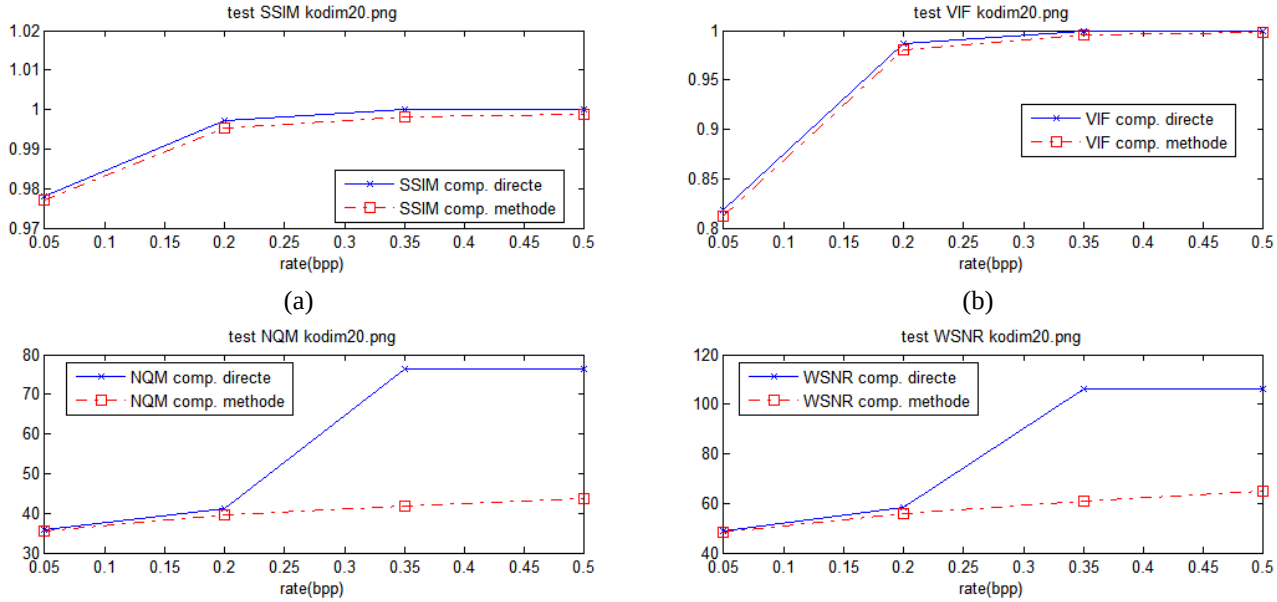


Figure 3-9 Evaluation de la méthode CMF appliquée à kodim20.png pour des différents bitrate choisis, en utilisant les métriques de qualité subjonctive (a) SSIM (b) VIF (c) NQM (d) WSNR

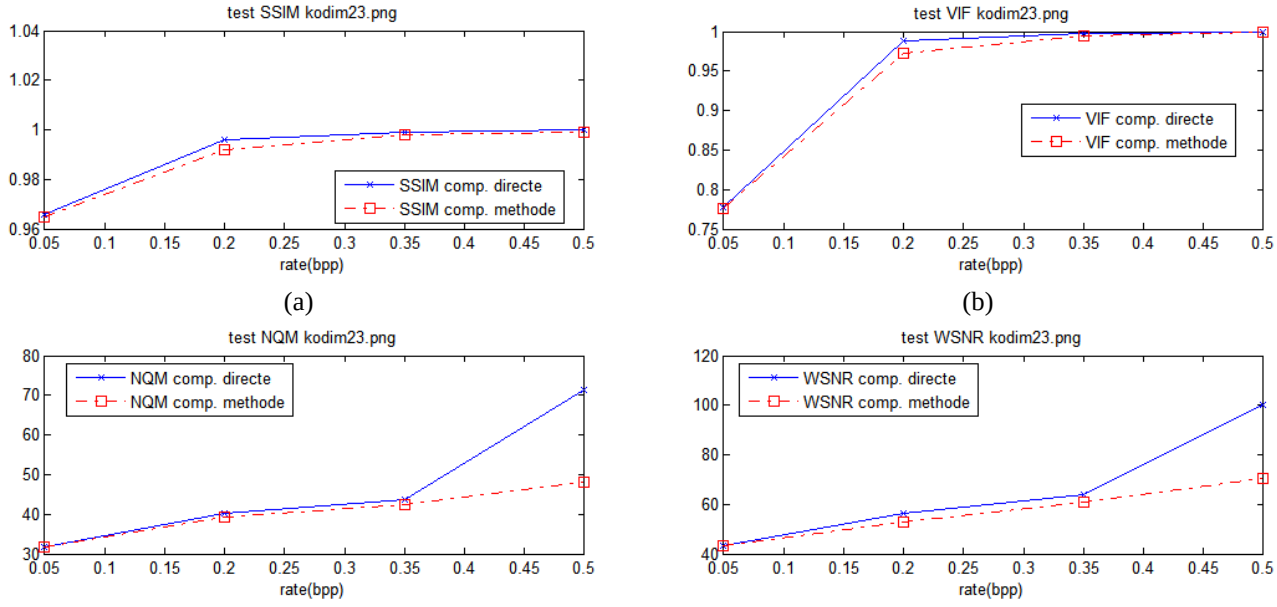


Figure 3-10 Evaluation de la méthode CMF appliquée à kodim23.png pour des différents bitrate choisis, en utilisant les métriques de qualité subjonctive (a) SSIM (b) VIF (c) NQM (d) WSNR

Cela se vérifie à travers l'étude comparative sur les poids de l'image et du signal compressé, présentée, sur les Tableaux IX et X. Le Tableau IX présente les variations du poids des images test compressées avec la CMF en fonction du bitrate. Quant au Tableau X, il présente les variations du poids cumulatif (de chaque image test) en fonction du bitrate où les images tests ont été compressées par JPEG2000 et le signal par le codeur MP3.

Le poids du signal « Handel » compressé avec le codeur MP3 est de 68.8Ko. Pour la capacité que nous avons calculée pour la méthode CMF, le poids du signal réduit est de 15.42Ko. A travers ces résultats, nous pouvons constater que la Compression Multimodale est efficace en termes de taux de compression.

TABLEAU IX
TAILLE IMAGE ET SIGNAL COMPRESSE AVEC
LA CMF POUR DIFFERENTS BITRATES TC (IMAGE, SIGNAL)

Image/bpp	0,05	0,10	0,25	0,35	0,50	0,75	1,00
kodim01,png	12,68	25,60	63,96	89,51	127,62	180,73	180,73
kodim08,png	12,69	25,60	63,75	89,58	127,95	177,87	177,87
kodim15,png	12,80	25,51	63,92	89,51	127,58	141,75	141,75
kodim20,png	12,80	25,55	63,38	89,52	123,71	123,71	123,71

TABLEAU X
TAILLE CUMULATIVE (KO) IMAGE COMPRESSEE SEULE
POUR DIFFERENTS BITRATES ET SIGNAL COMPRESSE SEULE
TC(IMAGE) + TC(SIGNAL)) TC(SIGNAL)= 15,42KO

Image/bpp	0,05	0,10	0,25	0,35	0,50	0,75	1,00
kodim01,png	28,16	40,94	79,09	104,97	143,26	192,74	192,74
kodim08,png	27,99	41,01	79,34	104,90	143,08	189,96	189,96
kodim15,png	28,01	40,96	79,26	104,92	143,04	148,27	148,27
kodim20,png	28,20	41,00	79,22	104,97	123,40	123,40	123,40

3.6 Evaluation des performances de la méthode CMSS

Pour la méthode d'insertion spirale sur les contours (CMSS) de l'image vue dans la section 2.5.4.1, nous allons encore utiliser la totalité de l'image, comme région d'insertion. Dans ce cas, d'après l'équation(2.13), la capacité d'insertion de la méthode est limitée par :

$$\frac{512 \times 512}{2} = 131072$$

La méthode d'insertion spirale ne nécessite l'utilisation d'aucun paramètre. C'est donc une insertion à l'«aveugle» sur l'image. Les résultats obtenus suite à l'exécution de notre procédure de tests, en termes de qualité de reconstruction de l'image et du signal sont à la Figure 3-11.

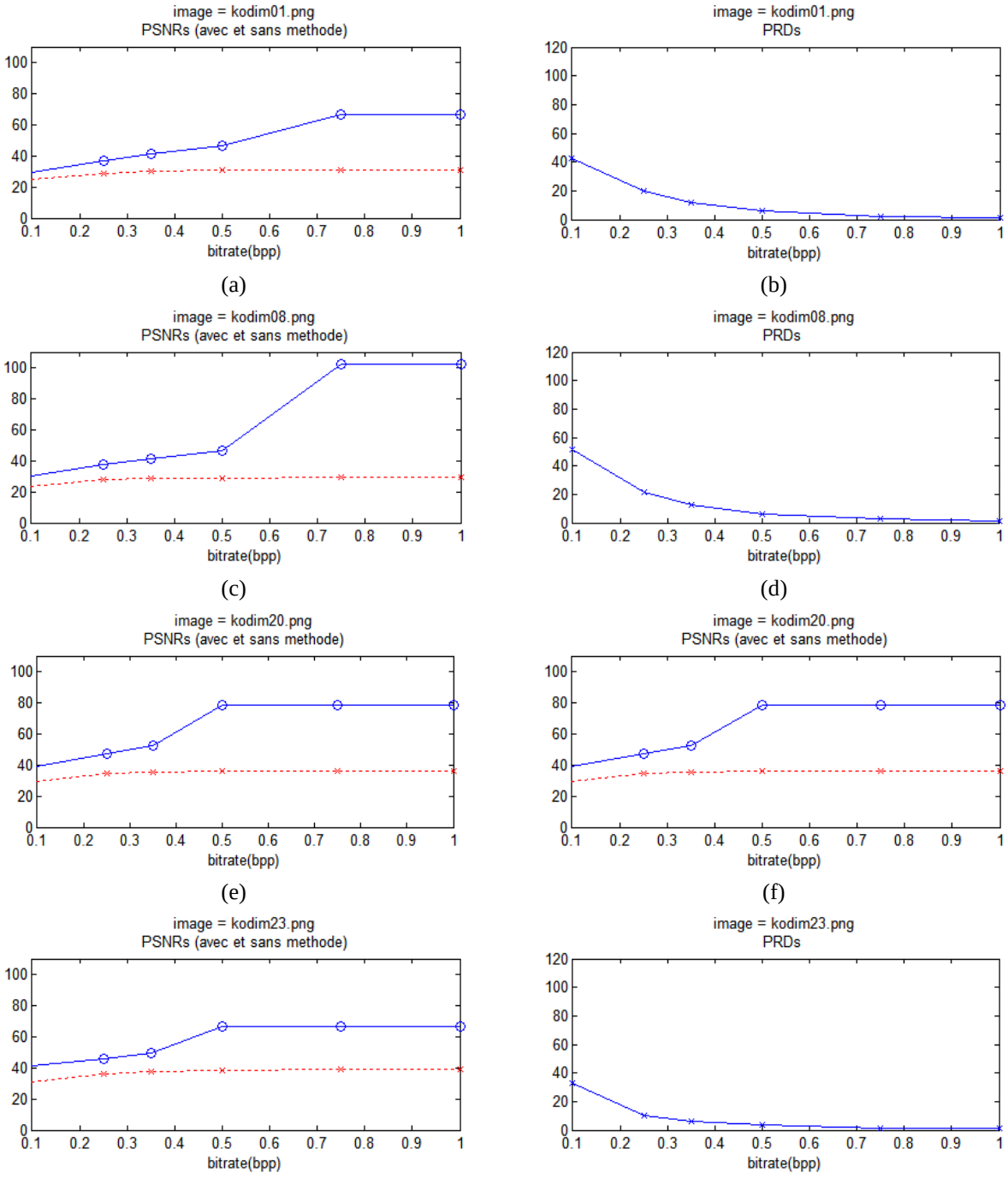
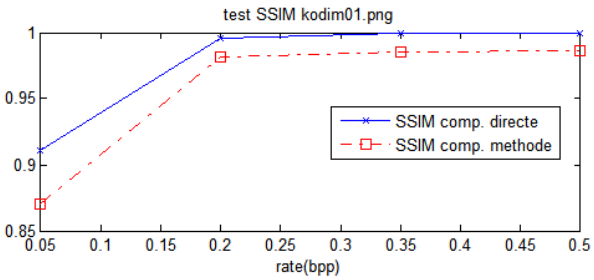
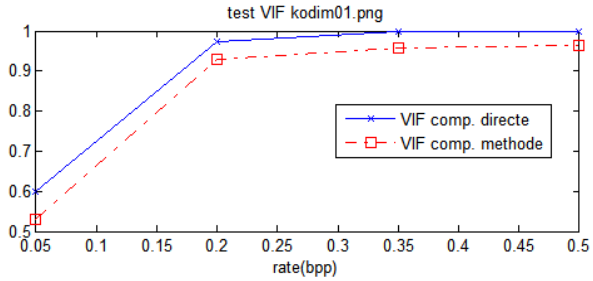


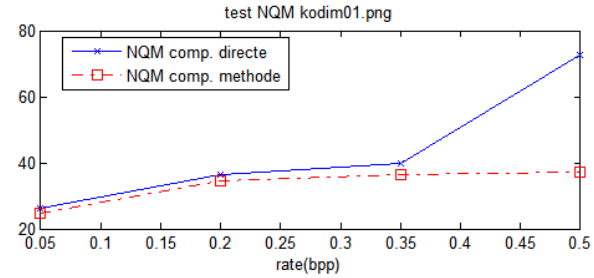
Figure 3-11 Evaluation de la méthode CMSS avec la compression seule de l'image par JPEG2000 pour des différents bitrate choisis. (a)-(b) kodim01, (c)-(d)- kodim08, (e)-(f) kodim20, (g)-(h) kodim23. Sur la Figure (o) dénote le PSNR direct et (x) le PSNR méthode



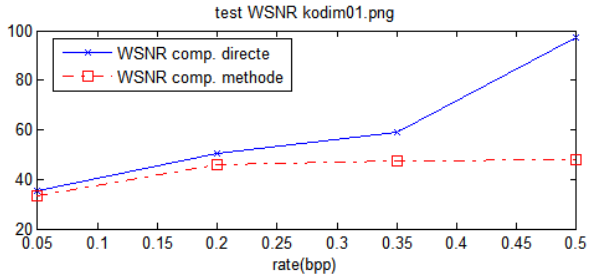
(a)



(b)

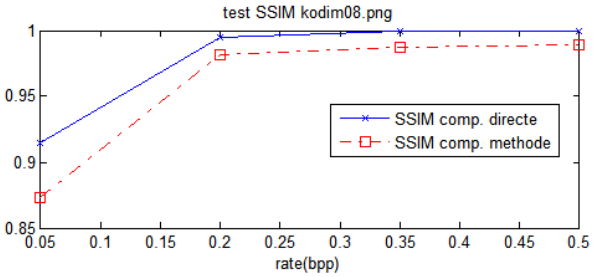


(c)

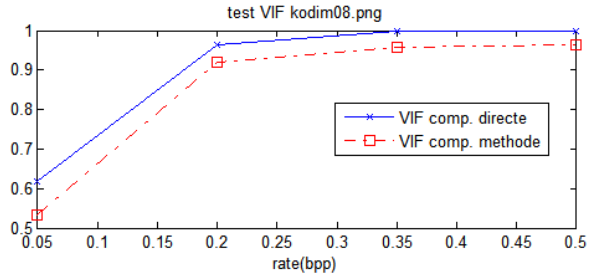


(d)

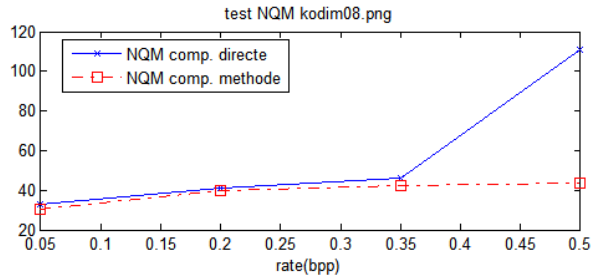
Figure 3-12 Evaluation de la méthode CMSS appliquée à kodim01.png pour des différents bitrate choisis, en utilisant les métriques de qualité subjonctive (a) SSIM (b) VIF (c) NQM (d) WSNR



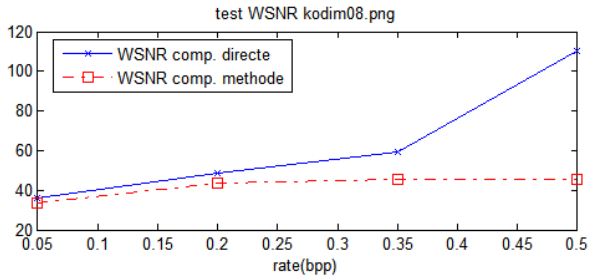
(a)



(b)

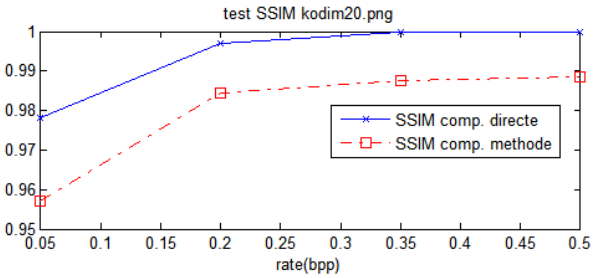


(c)

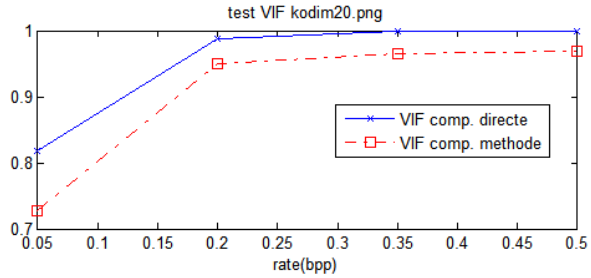


(d)

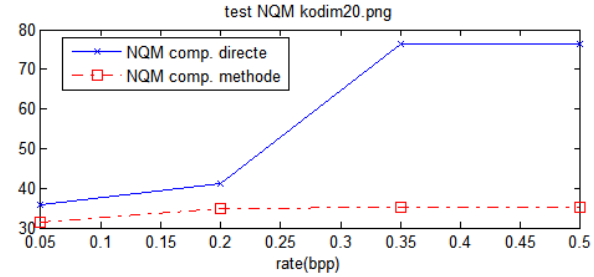
Figure 3-13 Evaluation de la méthode CMSS appliquée à kodim08.png pour des différents bitrate choisis, en utilisant les métriques de qualité subjonctive (a) SSIM (b) VIF (c) NQM (d) WSNR



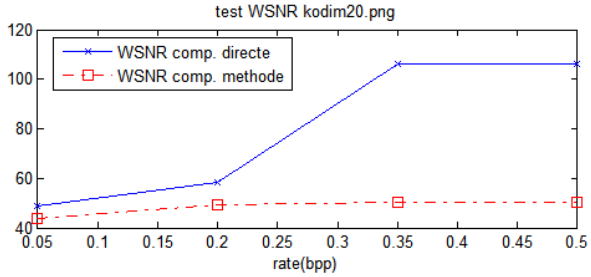
(a)



(b)

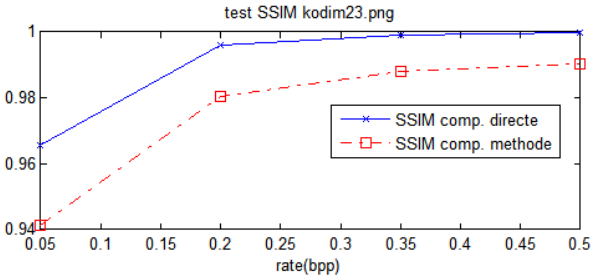


(c)

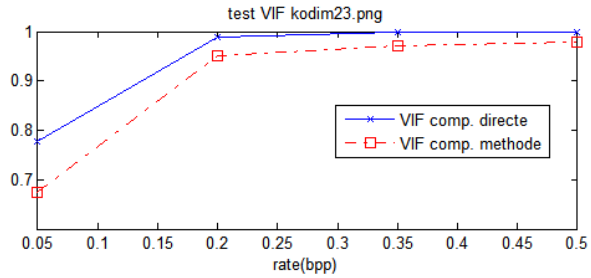


(d)

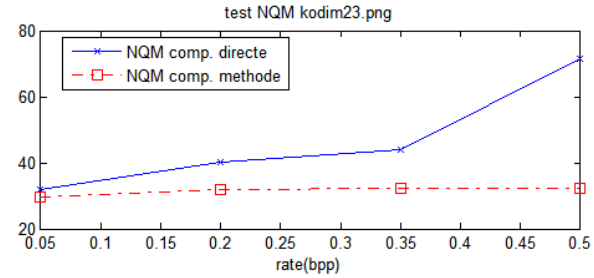
Figure 3-14 Evaluation de la méthode CMSS appliquée à kodim20.png pour des différents bitrate choisis, en utilisant les métriques de qualité subjonctive (a) SSIM (b) VIF (c) NQM (d) WSNR



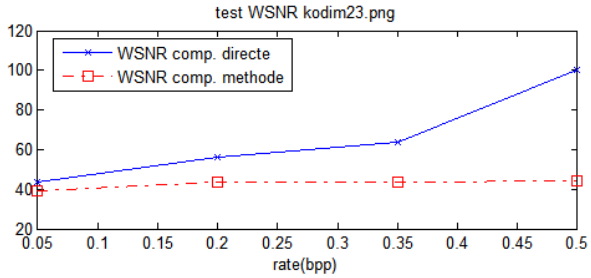
(a)



(b)



(c)



(d)

Figure 3-15 Evaluation de la méthode CMSS appliquée à kodim23.png pour des différents bitrate choisis, en utilisant les métriques de qualité subjonctive (a) SSIM (b) VIF (c) NQM (d) WSNR

Une première remarque au niveau des résultats par rapport à ceux de l'approche fréquentiel que nous avons vus à la section précédente : En utilisant la méthode d'insertion spirale, la qualité de reconstruction désirable (PRD ≤ 20%) pour le signal est obtenue pour des bitrates relativement faibles (i.e. 0.25 bpp), par rapport à ce que nous avons obtenu dans le domaine fréquentiel avec la CMF. En revanche, pour la qualité de reconstruction de l'image, les résultats obtenus pour le même bitrate, sont moins bons que ceux que nous avons obtenus par la technique d'insertion dans le domaine fréquentielle. Sur le TABLEAU XI, nous présentons le poids cumulatif de l'image compressée seule avec JPEG2000 et celui du signal compressé seul avec le codeur MP3. D'autre part, sur le Tableau XII, les poids des fichiers résultants par la Compression Multimodale de l'image et du signal. La quantité finale de l'information à transmettre ou à stocker est beaucoup plus faible avec la Compression Multimodale.

TABLEAU XI
TAILLE IMAGE ET SIGNAL COMPRESSE AVEC
LA CMF POUR DIFFERENTS BITRATES TC (IMAGE, SIGNAL)

Image/bpp	0,05	0,10	0,25	0,35	0,50	0,75	1,00
kodim01,png	81,54	94,32	132,47	158,35	196,64	246,12	246,12
kodim08,png	81,37	94,39	132,72	158,28	196,46	243,34	243,34
kodim15,png	81,39	94,34	132,64	158,30	196,42	201,65	201,65
kodim20,png	81,58	94,38	132,60	158,35	176,78	176,78	176,78

TABLEAU XII
TAILLE (KO) IMAGE ET SIGNAL COMPRESSE
AVEC LA METHODE CMSS POUR DIFFERENTS BITRATES
TC (IMAGE, SIGNAL)

Image/bpp	0,05	0,10	0,25	0,35	0,50	0,75	1,00
kodim01,png	12,63	25,56	63,92	89,37	127,64	191,98	200,53
kodim08,png	12,75	25,58	63,93	89,55	127,92	191,48	205,03
kodim15,png	12,70	25,56	63,91	89,49	127,79	176,36	176,36
kodim20,png	12,70	25,58	63,77	89,53	127,85	165,79	165,79

3.7 Evaluation des performances de la méthode CMSQ

Dans cette section, nous analysons les performances de la méthode CMSQ (voir 2.6.2.1). Comme nous l'avons défini dans la section 2.6.2.1, les performances de la détection par quadtree dépendent d'un paramètre *t* qui est utilisé comme un seuil pour le critère de division de la décomposition de *quadtree*.

D'après la procédure qui est employée pour la détection de la région d'insertion utilisant la décomposition par quadtree (« sélection des nœuds de plus grande taille et qui n'ont pas des nœuds fils – voir la section 2.6.2.1), ce

paramètre *t* a un effet sur la taille de région d'insertion, donc la capacité des échantillons qui peuvent être insérés.

La Figure 3-16 présente les résultats d'une étude de sensibilité de la méthode CMSQ par rapport au paramètre *t* pour différents bitrates fixés pour le codage JPEG2000. Sur la Figure 3-16 pour chaque valeur du paramètre *t,* nous présentons:

- la taille ou le nombre de blocs, qui ont été détectés par l'algorithme,
- la capacité d'insertion avec la région composée à partir des blocs détectés,
- la qualité de l'image reconstruite avec et sans la méthode CMSQ,
- la qualité du signal reconstruit.

Les notations PSNR1 et PSNR2 indiquent la qualité de reconstruction de l'image avec et sans la méthode CMSQ, respectivement. En effet, pour chaque image de test, le PSNR1 reste identique, mais nous le répétons sur chaque ligne du tableau, pour faciliter la comparaison avec le PSNR2.

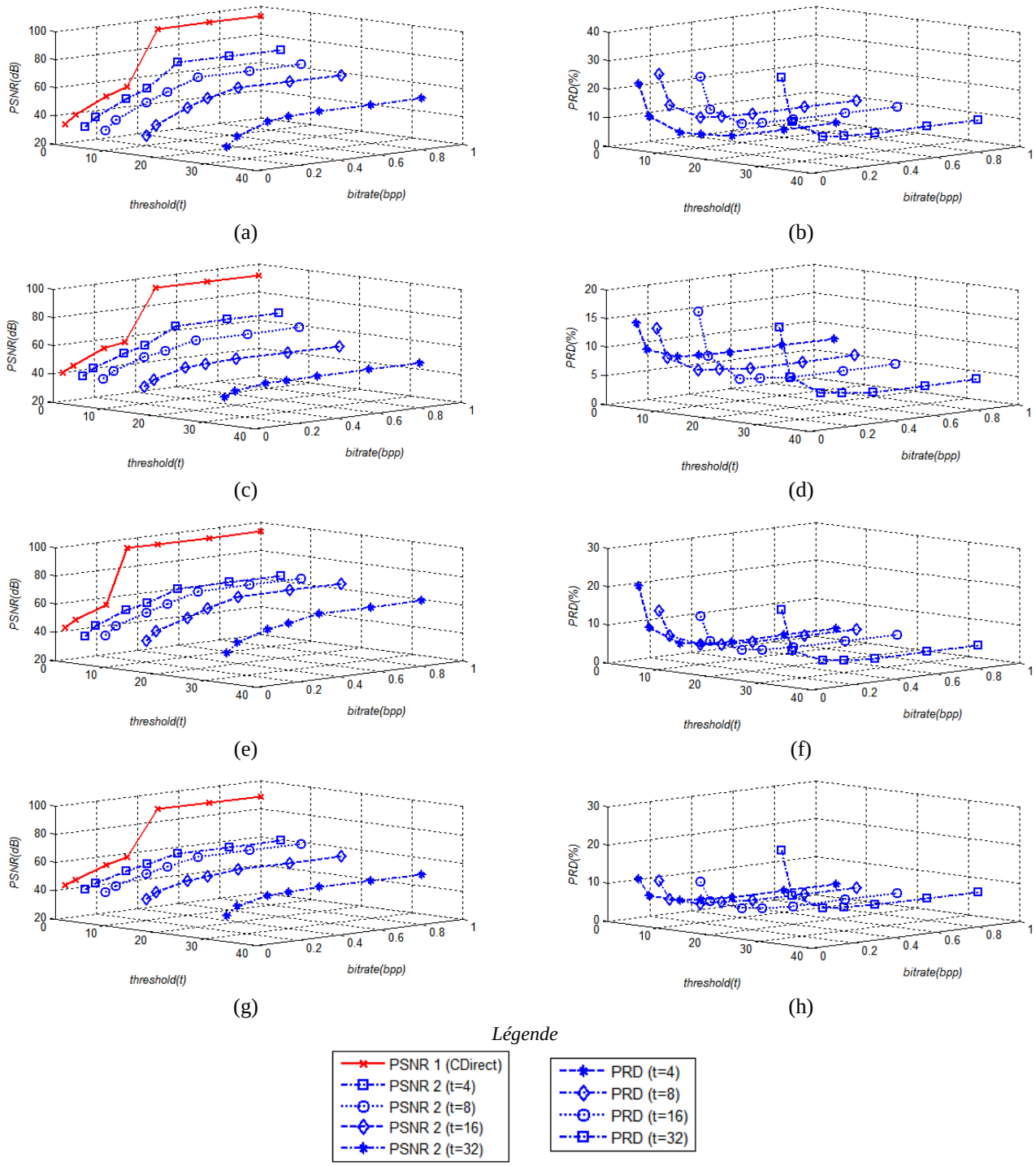
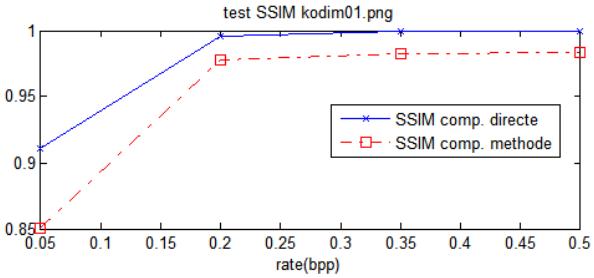
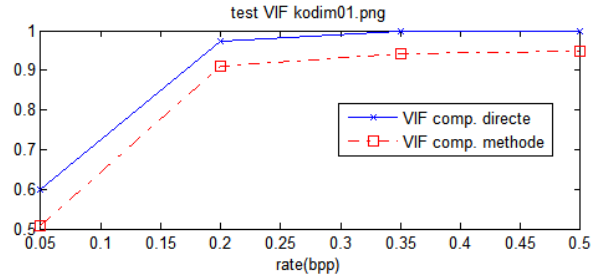


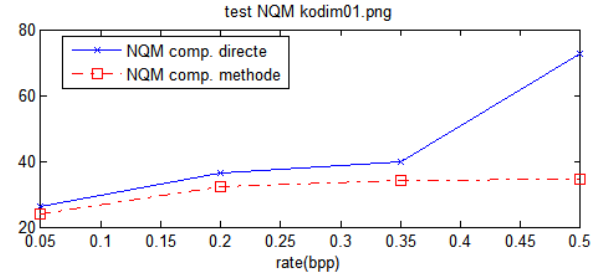
Figure 3-16 Evaluation de la méthode CMSQ avec la compression seule de l'image par JPEG2000 pour des différents bitrate choisis. (a)-(b) kodim01, (c)-(d)- kodim08, (e)-(f) kodim20, (g)-(h) kodim23. Sur la Figure (o) dénote le PSNR direct et (x) le PSNR méthode



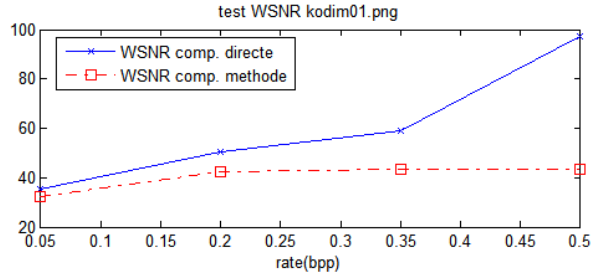
(a)



(b)

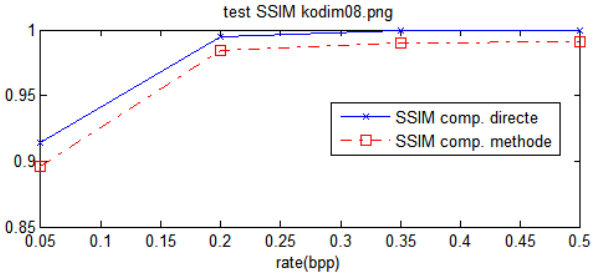


(c)

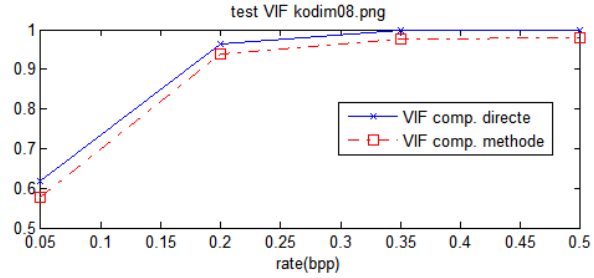


(d)

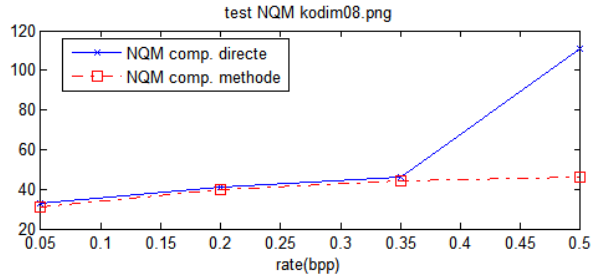
Figure 3-17 Evaluation de la méthode CMSQ (à t= 40) appliquée à kodim01.png pour des différents bitrate choisis, en utilisant les métriques de qualité subjonctive (a) SSIM (b) VIF (c) NQM (d) WSNR



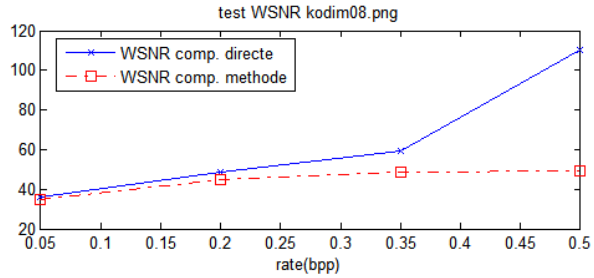
(a)



(b)



(c)



(d)

Figure 3-18 Evaluation de la méthode CMSQ (à t= 40) appliquée à kodim08.png pour des différents bitrate choisis, en utilisant les métriques de qualité subjonctive (a) SSIM (b) VIF (c) NQM (d) WSNR

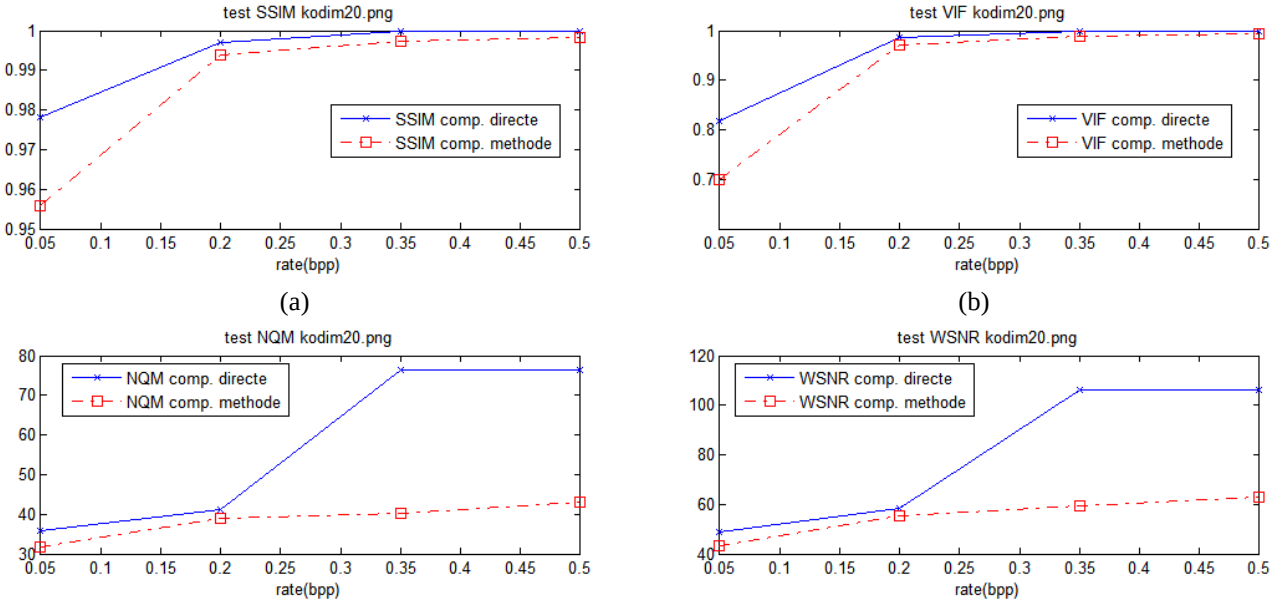


Figure 3-19 Evaluation de la méthode CMSQ (à t= 40) appliquée à kodim20.png pour des différents bitrate choisis, en utilisant les métriques de qualité subjonctive (a) SSIM (b) VIF (c) NQM (d) WSNR

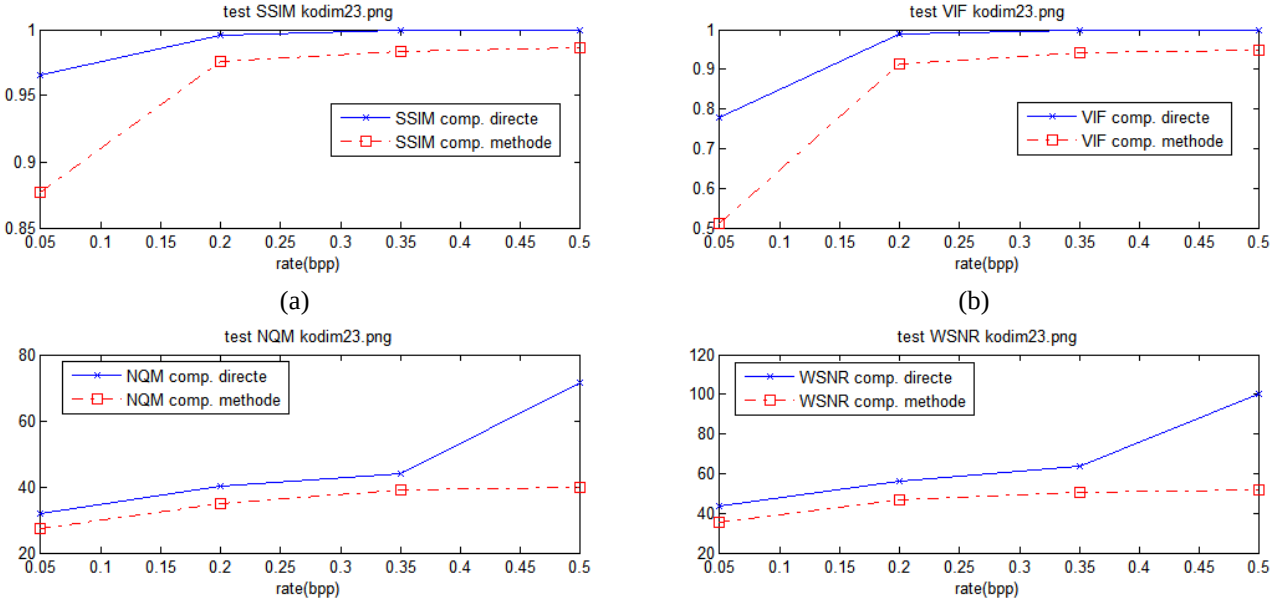
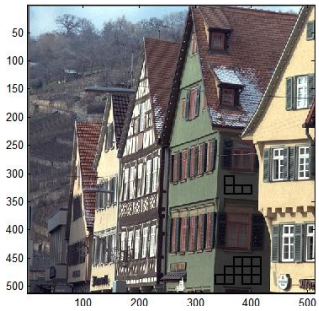


Figure 3-20 Evaluation de la méthode CMSQ (à t= 40) appliquée à kodim23.png pour des différents bitrate choisis, en utilisant les métriques de qualité subjonctive (a) SSIM (b) VIF (c) NQM (d) WSNR

A partir des résultats de la Figure 3-16 , nous observons que pour les quatre images test, la valeur de *t*=4, permet d'avoir de bons résultats de reconstructions de l'image et du signal (PSNR2 ≈ 67dB et PRD ≈ 0%).

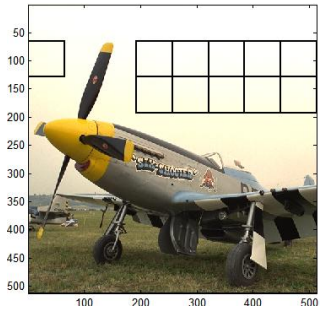
Ceci grâce à l'algorithme de détection « quadtree » (§2.6.2.1) pour la région d'insertion. La Figure 3-21 illustre les régions d'insertion détectées sur les images qui avait été sélectionnées par la technique basée sur la décomposition en « quadtree ».

(a)

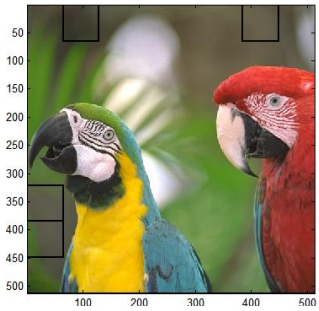


(b)

Figure 3-21 Les régions d'insertion détectées par la technique de détection « quadtree » pour *t = 4.* (a) kodim01, (b) kodim08, (c) kodim20, (d) kodim23.



(d)



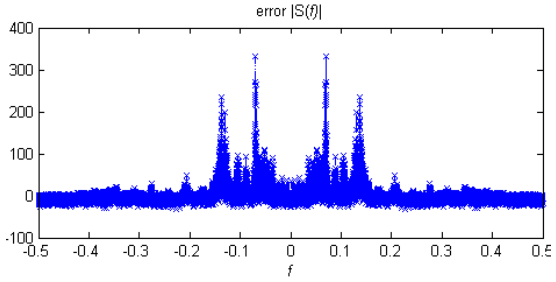
(d)

Pour les images kodim01 et kodim08 (Figure 3-21a et Figure 3-21b, respectivement), avec un seuil *t=4*, l'algorithme détecte une région d'insertion plus petite que celles détectées pour les images kodim20 et kodim23 (Figure 3-21c et, Figure 3-21d respectivement). Ceci est dû au fait que les images kodim01 et kodim08, contiennent plus de composants de hautes fréquences par rapports aux images kodim20 et kodim23. Par conséquent, les images kodim01 et kodim08 ont peu de capacité pour l'insertion d'un autre signal par rapport aux images kodim20 et kodim23. Il est à signaler que nous pouvons augmenter les tailles des régions d'insertions pour kodim01 et kodim08, ainsi que leurs capacités d'accueillir plus d'échantillons en choisissant un *t* supérieur.

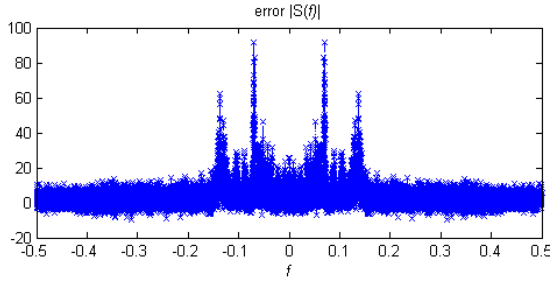
3.8 Analyse de la qualité du signal reconstruit

L'impact de l'insertion du signal audio dans l'image, pourrait être analysé en examinant les spectres d'énergie avant et après la compression. Afin d'analyser cet impact, nous faisons un simple test, en comparant la transformée de Fourrier du signal compressé à un bitrate choisi avec celle du signal initial. La Figure 3-22 illustre pour chacune des méthodes proposées, l'erreur entre la TF du signal initiale et celle du signal compressé par JPEG2000 à un bitrate 0.05 bpp. Nous avons ainsi effectué le même test sur différentes débits binaires choisis pour le codeur JPEG2000, notamment à 0.20 bpp (voir la Figure 3-23), à 0.35 bpp (voir la Figure 3-24) et finalement à 0.50 bpp (voir la Figure 3-25).

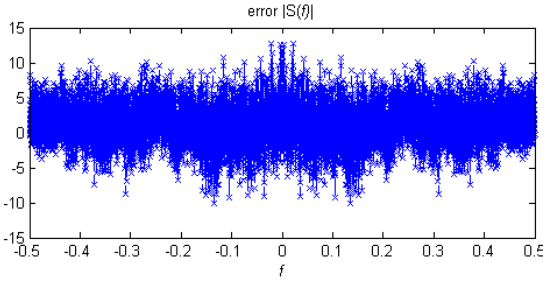
Une première remarque qui s'applique à toutes les méthodes de CM, est qu'une augmentation du débit binaire réduit le bruit introduit sur le signal reconstruit. La présence du bruit est beaucoup plus accentuée sur le spectre du CMF par rapports aux autres méthodes de CM. Ceci peut être visualisé en comparant l'erreur spectrale du signal reconstruit avec chaque méthode sur les figures 3-22, 3-23, 3-24 et 3-25. Nous pouvons ainsi en constater que la méthode qui conserve la qualité du signal au mieux possible, est la méthode CMSS.



(a) CMF

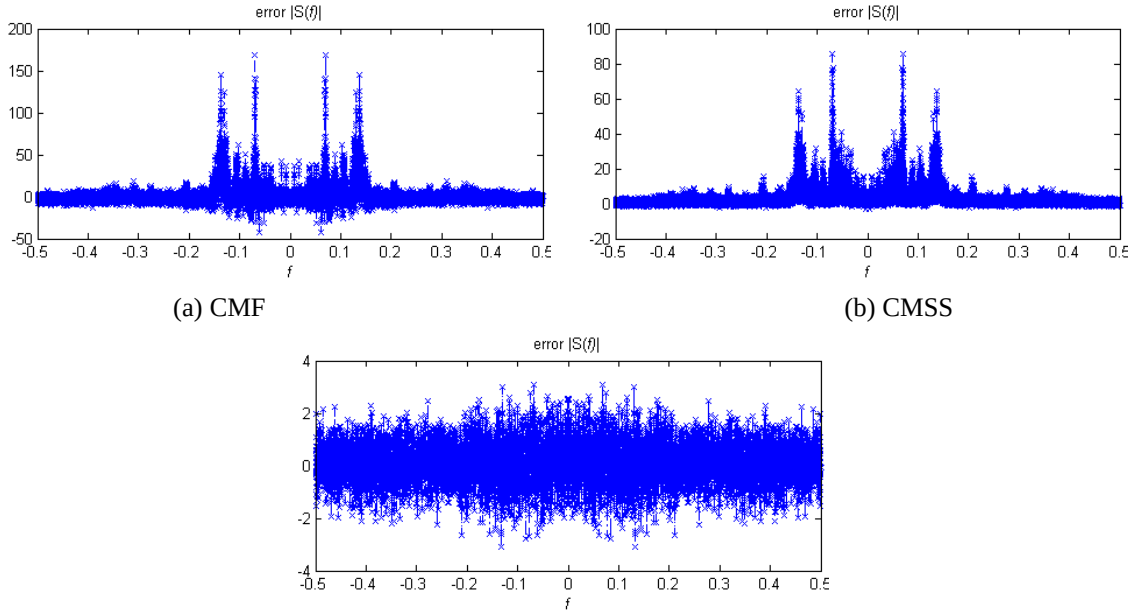


(b) CMSS



(c) CMSQ

Figure 3-22 L'erreur entre la TF du signal reconstruit compressé à **0.05 bpp** par JPEG2000 et la TF du signal initiale.

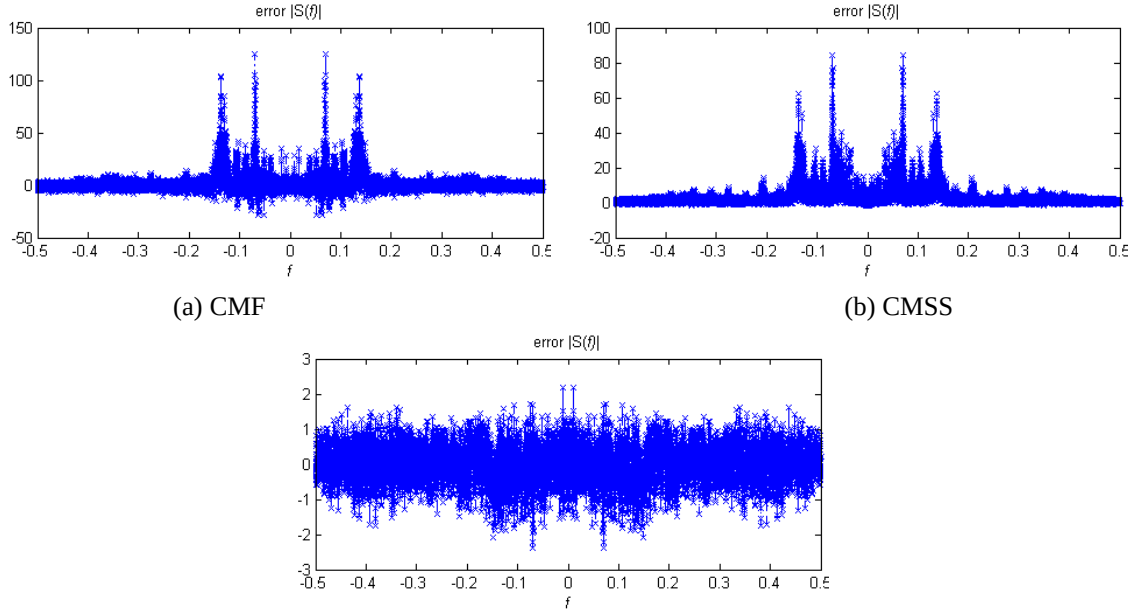


(a) CMF (b) CMSS

(c) CMSQ

Figure 3-23 L'erreur entre la TF du signal reconstruit compressé à **0.20 bpp** par JPEG2000 et la TF du signal initiale

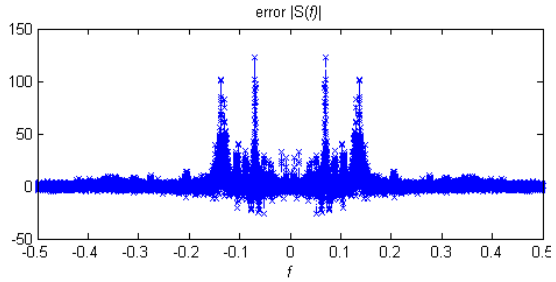
.



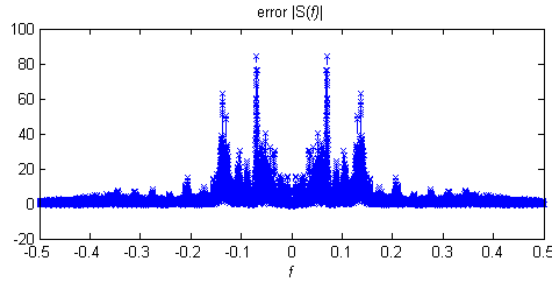
(a) CMF (b) CMSS

(c) CMSQ

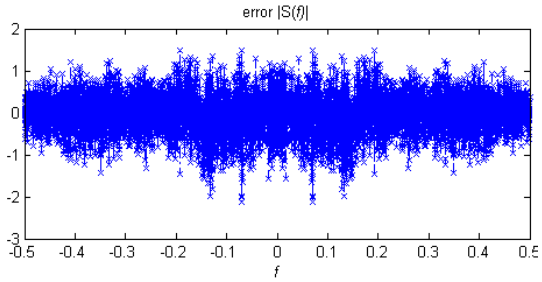
Figure 3-24 L'erreur entre la TF du signal reconstruit compressé à **0.35 bpp** par JPEG2000 et la TF du signal initiale.



(a) CMF



(b)CMSS



(c) CMSQ

Figure 3-25 L'erreur entre la TF du signal reconstruit compressé à **0.50 bpp** par JPEG2000 et la TF du signal initiale.

3.9 Bilan des résultats

Sur le Tableau XIII, nous présentons pour chaque image de test, les résultats comparatifs entre les différentes méthodes de Compression Multimodale que nous avons étudiées. Sur le tableau la notation PSNR2(*k*) et PRD(*k*) exprime le PSNR et le PRD calculé sur l'image et le signal reconstruit suite à l'utilisation de la *k*-ième méthode.

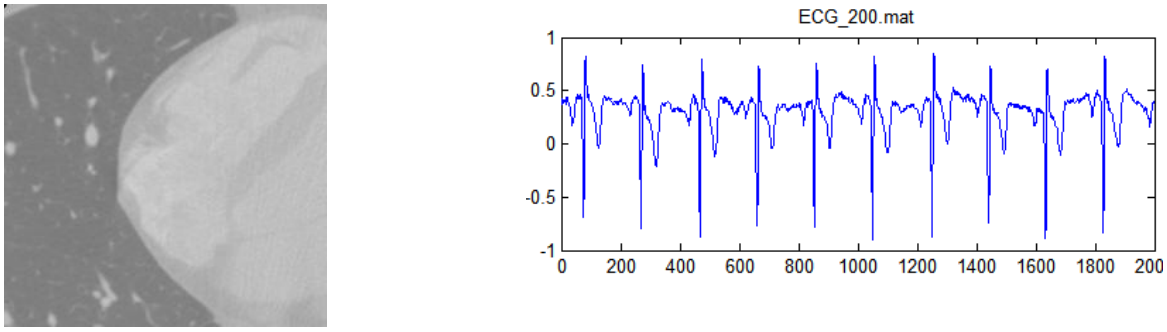
TABLEAU XIII
TABLEAU DE COMPARAISON DES METHODES DE COMPRESSION MULTIMODALE

image	rate	0,05	0,1	0,25	0,35	0,5	0,75	1
kodim01.png	PSNR2(CMF)	26,11	29,21	34,13	35,65	36,63	37,34	37,34
	PRD(CMF)	106,44	107,65	68,93	36,41	17,68	5,86	5,86
	PSNR2(CMSS)	22,08	24,77	28,84	29,96	30,57	30,77	30,78
	PRD(CMSS)	64,81	42,63	19,74	11,77	5,99	2,17	1,58
	PSNR2(CMSQ)	32,97	39,15	49,57	55,14	70,62	70,62	70,62
	PRD(CMSQ)	22,44	10,58	3,48	2,06	0	0	0
kodim08.png	PSNR2(CMF)	25,9	29,75	34,53	35,8	36,64	37,28	37,28
	PRD(CMF)	116,63	111,18	63,93	37,42	19,94	10,63	10,63
	PSNR2(CMSS)	20,38	23,44	27,56	28,42	28,85	29	29,02
	PRD(CMSS)	77,01	51,82	21,51	12,82	6,37	2,73	1,58
	PSNR2(CMSQ)	33,74	39,37	48,41	54,5	64,02	64,02	64,02
	PRD(CMSQ)	20,17	9,68	2,94	1,4	1,32	1,32	1,32
kodim20.png	PSNR2(CMF)	33,79	37,13	39,63	40,19	40,94	40,94	40,94
	PRD(CMF)	104,17	90,77	38,4	29,52	24,76	24,76	24,76
	PSNR2(CMSS)	25,57	29,67	34,57	35,63	36,12	36,31	36,31
	PRD(CMSS)	65,45	37,31	12,67	6,89	3,7	1,51	1,51
	PSNR2(CMSQ)	37,94	44,66	53,08	55,97	63,75	63,75	63,75
	PRD(CMSQ)	20,65	9,27	4,03	3,25	2,83	2,83	2,83
kodim23.png	PSNR2(CMF)	37,4	39,18	41,35	42,45	44,07	44,07	44,07
	PRD(CMF)	98,46	53,37	19,91	16,69	7,54	7,54	7,54
	PSNR2(CMSS)	26,17	30,86	36,38	37,74	38,58	39,03	39,03
	PRD(CMSS)	55,9	32,77	10,19	6	3,77	1,58	1,58
	PSNR2(CMSQ)	41,82	45	51,48	54,19	59,03	59,03	59,03
	PRD(CMSQ)	11,46	6,8	4,5	4,13	3,54	3,54	3,54

3.10 Application biomédicale de la Compression Multimodale

Nous avons étudié les méthodes de CM que nous avons proposées aussi dans le cadre d'une application biomédicale. D'une part, nous avons utilisé la base de données de « MIT-BIH arrhythmia database » pour les signaux [**57**] et d'autre part, la base de données MeDEISA (Medical Database for the Evaluation of Image and Signal Processing Algorithms) [**58**] contenant des images médicales non compressées y compris les images IRM.

Dans cette section nous allons présenter les résultats d'une application biomédicale de la CM. Dans cette application, nous avons inséré un signal ECG ayant une dérivation sur une image IRM en utilisant les méthodes proposées. La Figure 3-26 (a) et (b) présentent respectivement l'image IRM et le signal ECG qui sont utilisés dans le test.



(a) (b)

Figure 3-26 (a) L'image IRM : L'enregistrement IM112 sur MeDEISA
(b) Le signal ECG : L'enregistrement # 200 sur MIT-BIH arrhythmia databaase.

Les Figures 3-27, 3-29 et 3-21 illustrent respectivement, la qualité de reconstruction de l'image en fonction du débit binaire pour chaque méthode et les Figures 3-28, 3-29 et 3-30 illustrent la différence entre la TF du signal reconstruit et celle du signal initiale.

De ces résultats, nous pouvons constater que quelque soit la méthode choisie, un débit binaire plus élevé provoque une augmentation à la fois dans la qualité de reconstruction du signal et de l'image. Toutefois, la CMF semble plus sensible aux changements de débit que les autres méthodes. La CMSS et la CMSQ exposent des meilleures qualités devant faibles débits binaires.

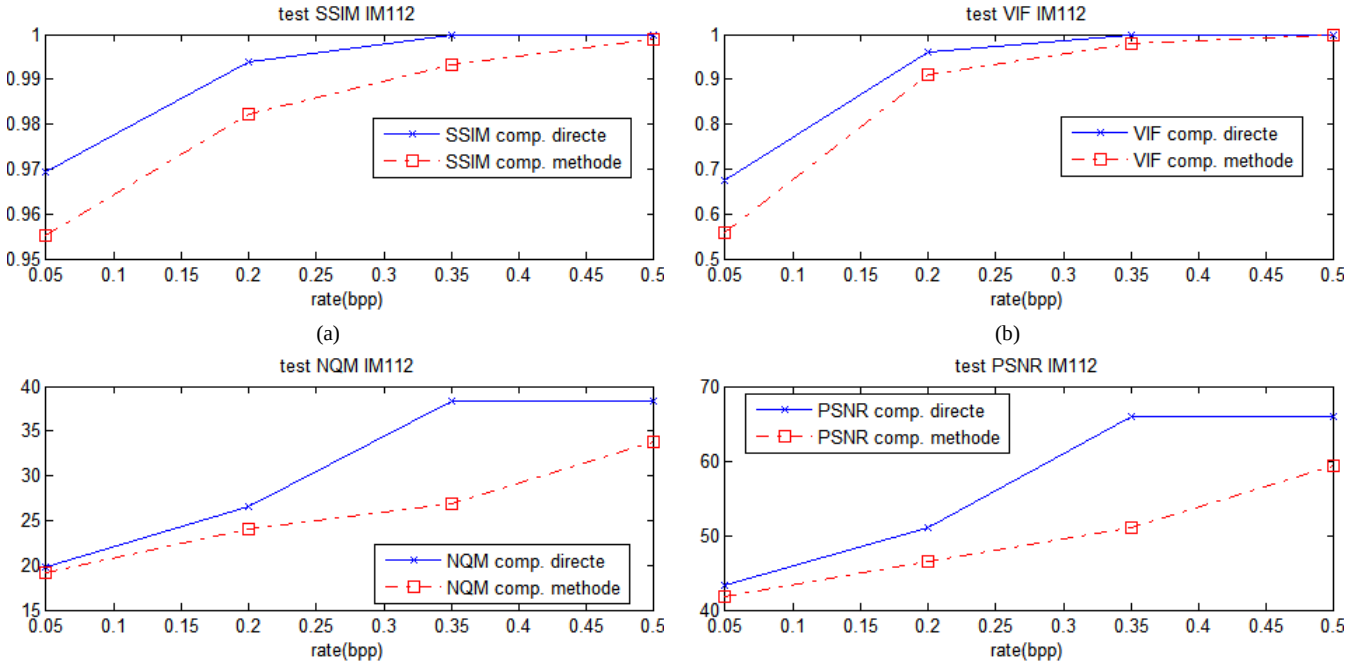


Figure 3-27 Evaluation de la qualité de reconstruction d'image avec la méthode CMF appliquée à des données biomédicale
(a) SSIM (b) VIF (c) NQM (d) PSNR

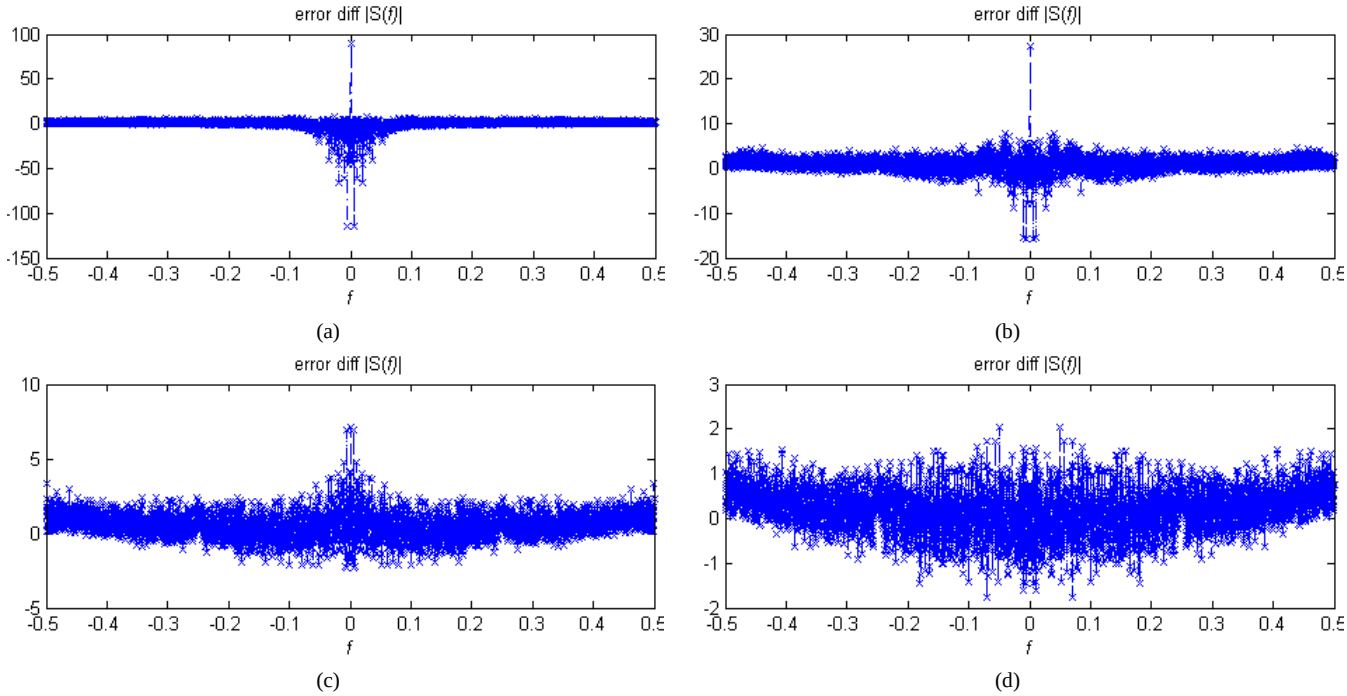


Figure 3-28 La différence entre la TF du signal reconstruit et la TF du signal initiale pour différentes débits binaires choisis pour JPEG 2000 en utilisant la méthode **CMF** pour la CM
(a) 0.05 bpp (b) 0.20 bpp (c) 0.35 bpp (d) 0.50 bpp

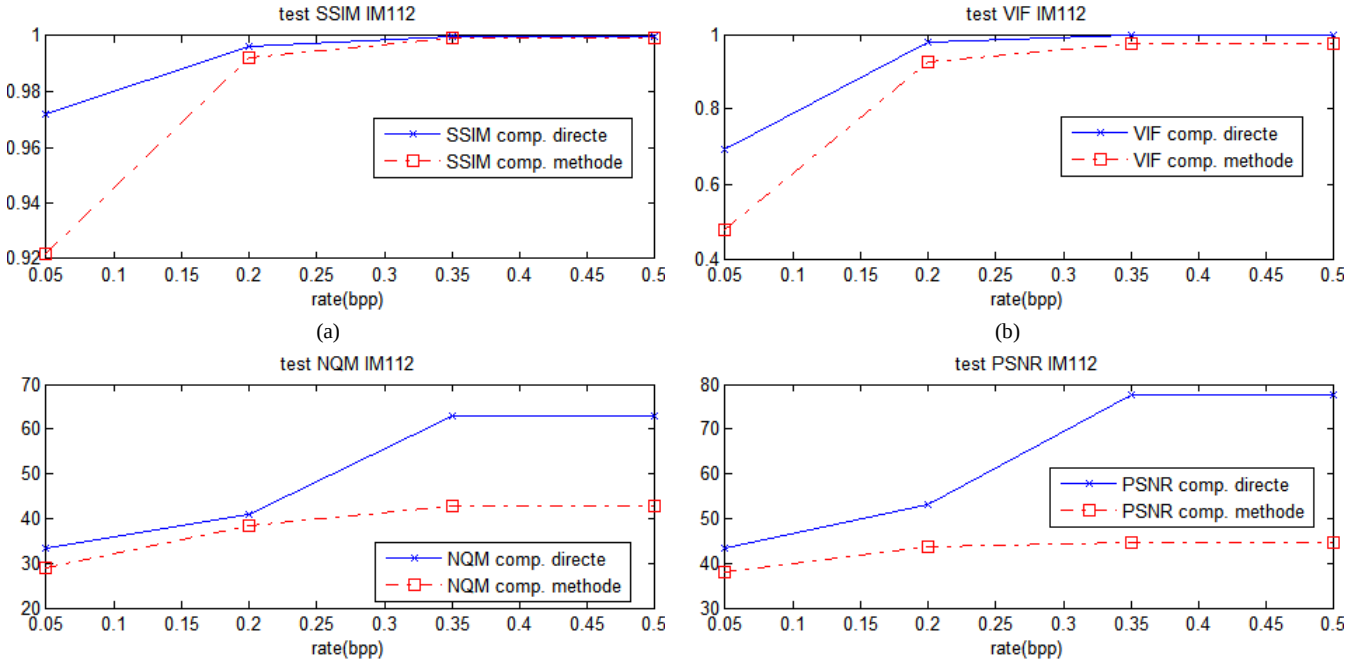


Figure 3-29 Evaluation de la qualité de reconstruction d'image avec la méthode CMSS appliquée à des données biomédicale
(a) SSIM (b) VIF (c) NQM (d) PSNR

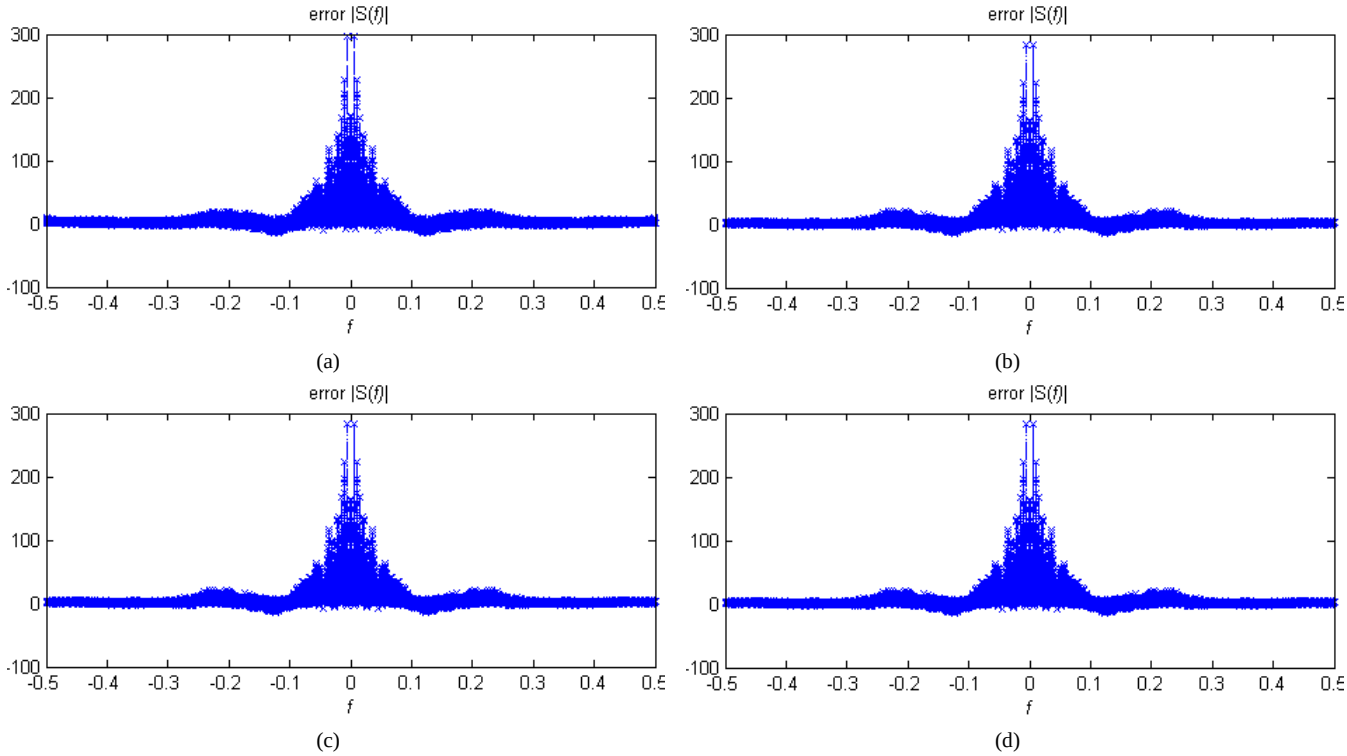


Figure 3-30 La différence entre la TF du signal reconstruit et la TF du signal initiale pour différentes débits binaires choisis pour JPEG 2000 en utilisant la méthode **CMSS** pour la CM
(a) 0.05 bpp (b) 0.20 bpp (c) 0.35 bpp (d) 0.50 bpp

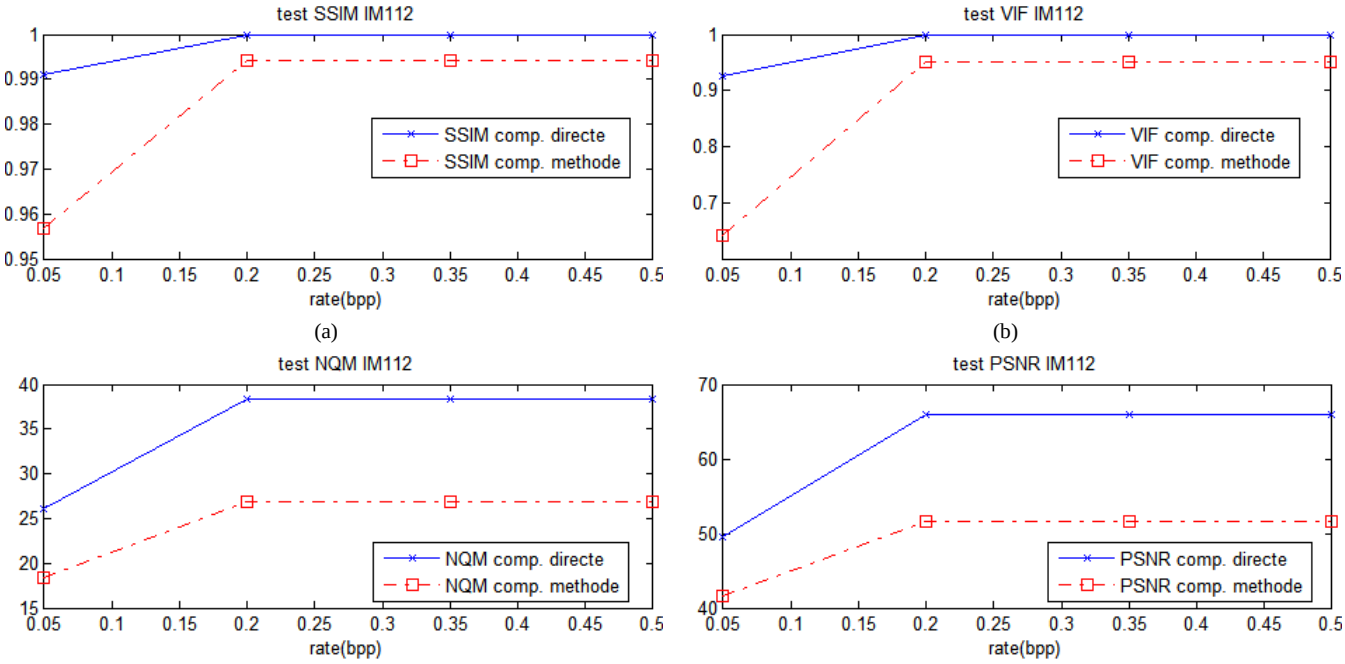


Figure 3-31 Evaluation de la qualité de reconstruction d'image avec la méthode CMSQ appliquée à des données biomédicale
(a) SSIM (b) VIF (c) NQM (d) PSNR

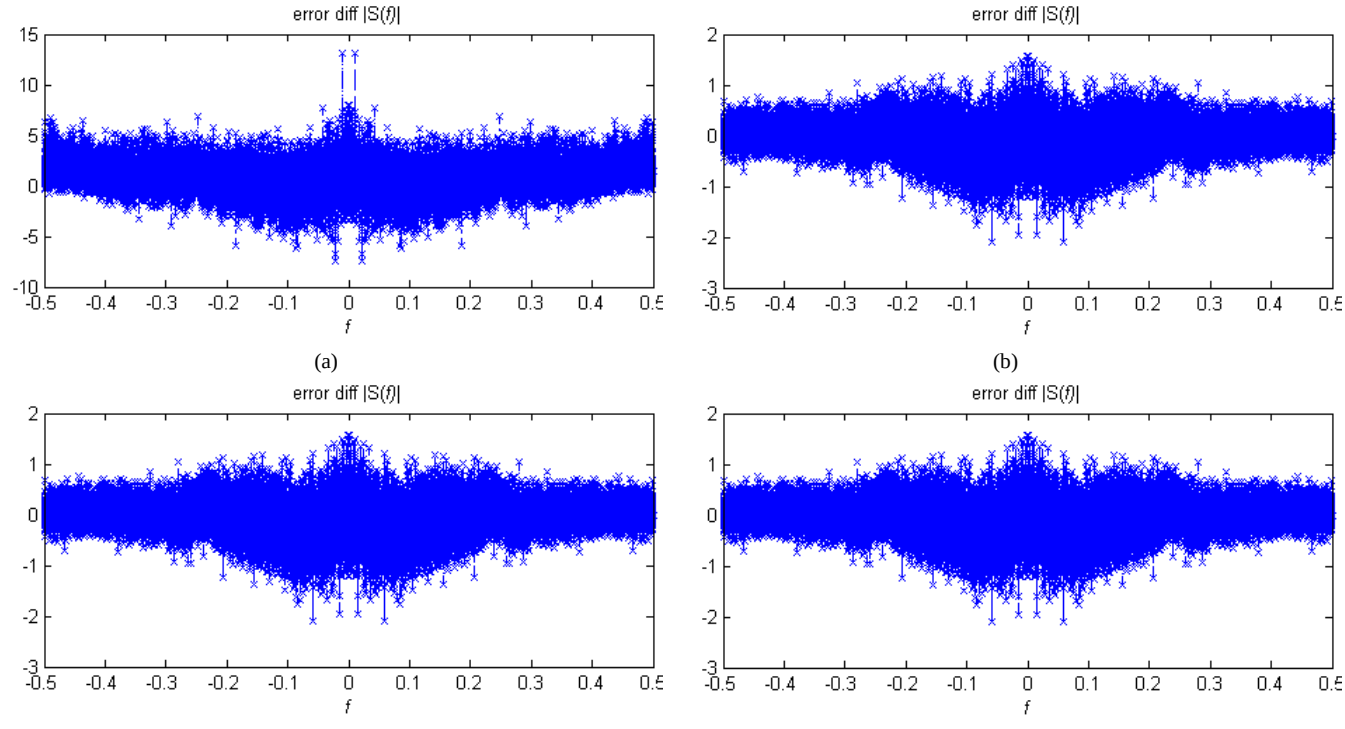


Figure 3-32 La différence entre la TF du signal reconstruit et la TF du signal initiale pour différentes débits binaires choisis pour JPEG 2000 en utilisant la méthode **CMSQ** pour la CM
(a) 0.05 bpp (b) 0.20 bpp (c) 0.35 bpp (d) 0.50 bpp

3.11 Conclusion

Nous constatons que la méthode CMF favorise une bonne reconstruction de l'image face à la qualité de reconstruction du signal. Cela ne nous surprend pas, en effet, car seules les composants hautes fréquences sont utilisées dans cette méthode, les échantillons appartenant au signal sont susceptibles d'être tronqués par le codeur JPEG2000 pour de faibles valeurs de « bitrates » choisies.

Parmi les méthodes spatiales, la méthode CMSS offre un bon compromis entre les qualités de l'image et du signal reconstruit. Mais, nous observons que les meilleurs résultats sont obtenus avec la méthode CMSQ. Cependant, contrairement aux méthodes précédentes, la méthode CMSQ n'utilise pas la totalité de l'image pour région d'insertion.

Chapitre 4
Extension de l'idée de Compression Multimodale à la vidéo

Dans cette partie, nous présentons une méthode de Compression Multimodale pour la Compression Multimodale d'une séquence vidéo (CMV) avec sa bande sonore [**59**] [**60**]. La méthode est basée sur la méthode d'insertion non-supervisée pour la Compression Multimodale d'une image avec un autre signal, (voir la section 2.5.4).

4.1 Problématique

Dans le standard MPEG, les données vidéo et audio sont représentées par une suite de paquets dans un fichier appelé « containeur de média». Les données sont compressées par différents codeurs propres à chaque type de donnée. Cela peut être le codeur H.264 pour la vidéo et le codeur MP3 ou le codeur AAC [**61**] pour l'audio. Quelque soit le type de compression choisie par l'utilisateur, à la fin de l'étape de compression, les flux compressés sont multiplexés afin de ne former qu'un seul flux composé des paquets entrelacés. La synchronisation entre deux flux est assurée par l'utilisation des *timestamps* qui représentent l'instant de présentation de la donnée courante par rapport au début de la lecture [**62**] [**63**]

Lors du décodage, ces derniers flux composés de paquets entrelacés sont d'abord démultiplexés dans des flux séparés, ensuite décodés par un décodeur approprié, puis ils sont redirigés vers un périphérique de lecture afin d'être présentés à l'utilisateur.

Chaque flux compare son *timestamp* actuel avec l'horloge du système pour assurer la synchronisation entre la lecture de la bande d'audio et la présentation des trames de vidéo. En cas de retard de l'un des flux, le

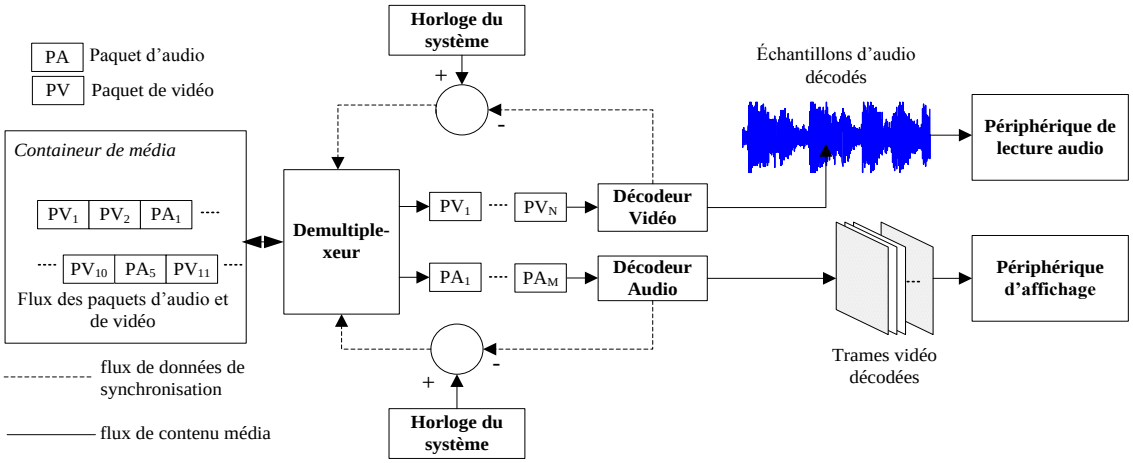


Figure 4-1 Schéma de principe du décodage d'un flux audio-vidéo dans le standard MPEG

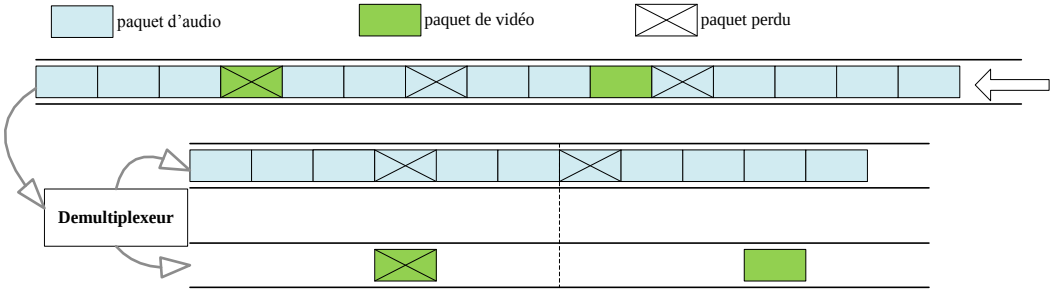


Figure 4-2 Perte de paquets d'audio et de vidéo dans la transmission MPEG

démultiplexeur peut choisir de retarder l'affichage de l'un des flux ou d'ignorer un certain nombre de paquets jusqu'à ce que la synchronisation soit effective. [**64**] (Voir la Figure 4-1).

Le mécanisme de synchronisation par *timestamp* est simple à implémenter et peu couteux en terme de complexité. Pourtant l'un des problèmes principaux de cette approche est qu'elle souffre de perte des paquets lors d'une transmission éventuelle [**65**] [**66**].

Ainsi dans un tel cas, la synchronisation entre le flux vidéo et audio ne sera pas effective. (Voir la Figure 4-2). Un second point à considérer est que l'approche basée sur le *timestamp* nécessite d'implémenter des étapes de démultiplexage du flux d'entrée, puis d'utiliser deux décodeurs pour les reconstruire. Ceci implique l'utilisation de davantage de ressources et de temps de traitement. Ces deux problèmes peuvent être résolus dans le contexte de la compression multimodale.

4.2 Compression multimodale audio et vidéo

L'idée principale de la méthode proposée est d'insérer les échantillons du signal audio dans le domaine spatial en suivant une approche linéaire non-supervisée comme suggéré dans la section 2.5.4.

Les schémas bloc des étapes de codage et décodage avec l'algorithme proposé sont donnés respectivement à la Figure 4-3et la Figure 4-4. Dans cette approche, conformément à notre model de compression en amont du codeur, les échantillons du signal audio sont insérés dans la trame vidéo par une fonction de mélange afin d'obtenir une trame de mélange. Les trames de mélange successives sont ensuite compressées par un codeur H.264/AVC au débit binaire choisi par l'utilisateur. Les paquets de données compressées obtenues à la sortie du codeur peuvent ensuite être redirigés vers un support de stockage ou vers un canal de transmission.

Dès leur réception, le système décompresse les paquets pour obtenir les trames de mélange, et la fonction de séparation en extrait les échantillons audio, et reconstruit les trames vidéo. Suite à cette étape, les échantillons audio peuvent être lus en même temps que l'affichage de la trame vidéo correspondante. Cette méthode, comme il n'existe plus de paquet audio dans le flux initial, supprime la nécessité de l'étape de démultiplexage, ainsi le

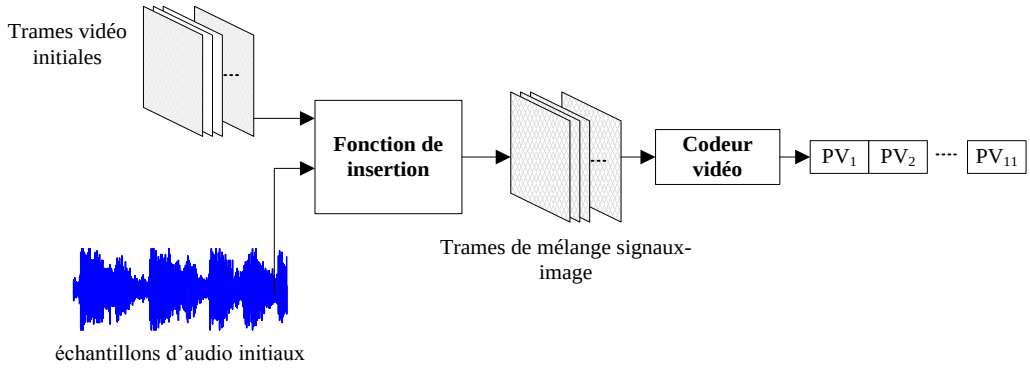


Figure 4-3 Schéma de bloc de Compression Multimodale audio-vidéo

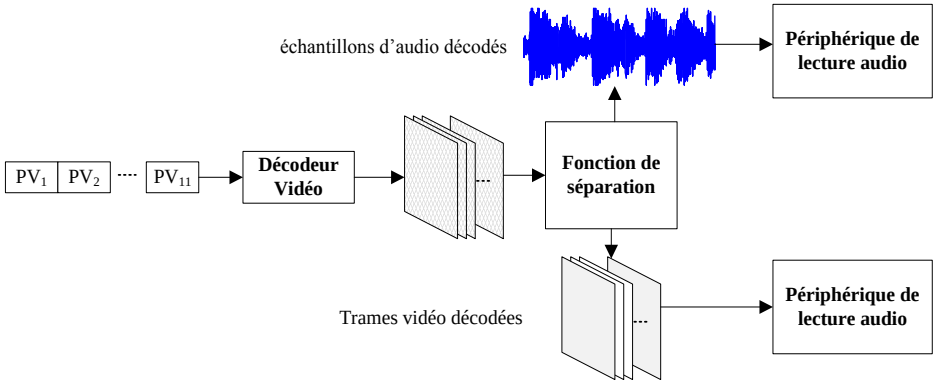


Figure 4-4 Schéma de bloc de Décompression Multimodale audio-vidéo

système n'est plus dépendant des *timestamps* qui sont utilisés explicitement lors de la synchronisation entre l'audio et la vidéo.

Dans le cas de perte de paquets lors d'une transmission, la trame vidéo et le signal audio seront perdus ensemble. Ainsi, cela ne provoquera aucun effet négatif en termes de synchronisation.

Les avantages de la méthode proposée sont les suivants :

1. Un seul codeur/décodeur est utilisé pour compresser/décompresser le flux de donnée. Ce faisant, le nombre de ressources utilisé et le temps de traitement sont diminués.
2. Aucun canal n'est utilisé pour la transmission de l'audio, donc la bande passante est économisée.
3. La lecture des flux est garantie.
4. La synchronisation entre les données de vidéo et d'audio est assurée. Les pertes des paquets lors de transmission n'ont aucun effet sur la synchronisation.
5. Les taches complexes de multiplexage/démultiplexage et synchronisation du système MPEG sont évitées.

4.3 Méthodologie de Compression Multimodale pour la vidéo

Dans la méthode proposée, les échantillons du signal embarqué doivent être insérés sans pour autant dégrader la qualité visuelle de l'image.

4.3.1 Positionnement temporelle des échantillons invités sur les trames de vidéo

Dans la structure de GOP (Group of Pictures) qui est définie par les standards MPEG-x et H.26x, il peut y avoir 3 types de trames : I-Trame, P-Trame et B-Trame, dont chacune est compressée par des différentes modes de compression. Notre méthode utilise l'ensemble de ces trois types de trames, mais il est attendu que les échantillons insérés, dans les I-Trames, ont une meilleure chance de survivre l'étape de quantification du codeur. Dans le cas où une meilleure qualité de reconstruction est désirée, les I-Trames seules peuvent être utilisées.

4.3.2 Positionnement spatiale des échantillons invités dans le domaine YCbCr

Comme nous l'avons dit avant, l'insertion des échantillons du signal embarqué se fait dans le plan Y de la décomposition en YCbCr, de la trame RGB.

4.4 Description de la méthode

4.4.1 Procédure de codage

La Figure 4-5 Illustre le schéma de bloc de la procédure de codage. La méthode d'insertion se compose des principales étapes suivantes :

1. Conditionnement des échantillons du signal embarqué,
2. Insertion des échantillons dans les trames,
3. Compression par H.264/AVC.

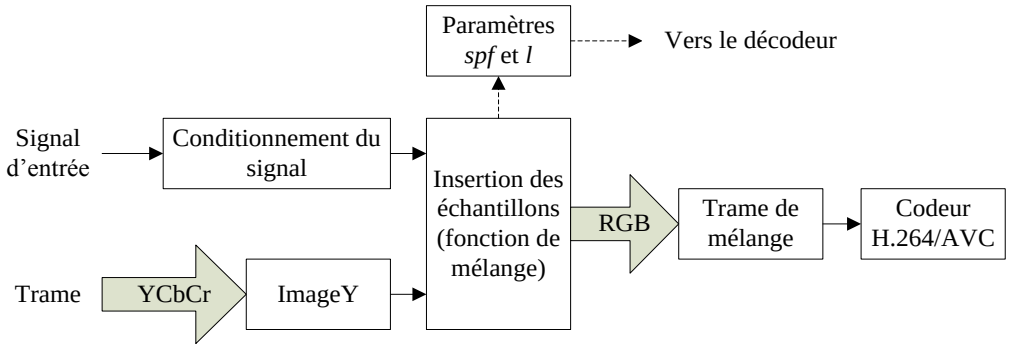


Figure 4-5 Schéma de block résumant la procédure de codage

4.4.1.1 Conditionnement des échantillons du signal embarqué

L'étape du « conditionnement » des échantillons consiste à redimensionner la plage dynamique des échantillons et à les remettre en ordre si le signal audio se compose de plusieurs canaux. Le redimensionnement des échantillons est nécessaire car la plupart des signaux audio sont représentés par des échantillons codés sur 16-bit. Le signal est conditionné de la même façon que celle vue dans la section 2.5.1 par l'équation (2.11)

Dans la seconde partie, s'il existe plusieurs canaux sonores accompagnant la vidéo, un seul canal est construit à partir des échantillons de chaque canal, comme indiqué sur la Figure 4-6.

4.4.1.2 Insertion des échantillons

Après l'étape de conditionnement, les échantillons sont prêts à être insérés dans une trame vidéo. La première étape consiste à calculer le nombre d'échantillons à insérer dans une trame de la séquence de vidéo ainsi que le niveau d'insertion nécessaire à la mise en œuvre de l'insertion linéaire (voir section 2.5.4.2). Nous notons ces deux paramètres *spf* pour le nombre d'échantillons par trame et *l* pour le niveau d'insertion.

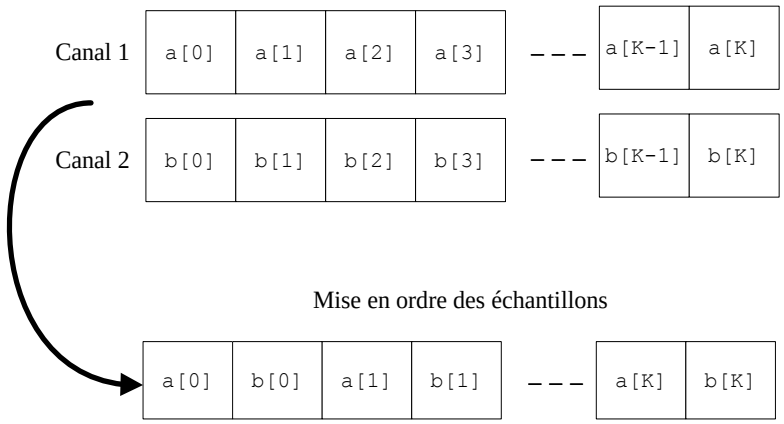


Figure 4-6 Mise en ordre des échantillons d'un signal contenant plusieurs canaux

Les étapes sont les suivantes :

1. Calcul du *spf* et du niveau d'insertion l ,
2. Transformation de chaque trame en représentation YCbCr,
3. Insertion des échantillons sur le plan Y suivant le modèle linéaire abordé au paragraphe 2.5.4.2 (sur le contour),
4. Transformation inverse en RGB,
5. Compression de la trame de mélange par H.264/AVC.

L'idée est de découper le signal audio en parties de taille correspondante à une trame de la séquence. (voir la Figure 4-7).

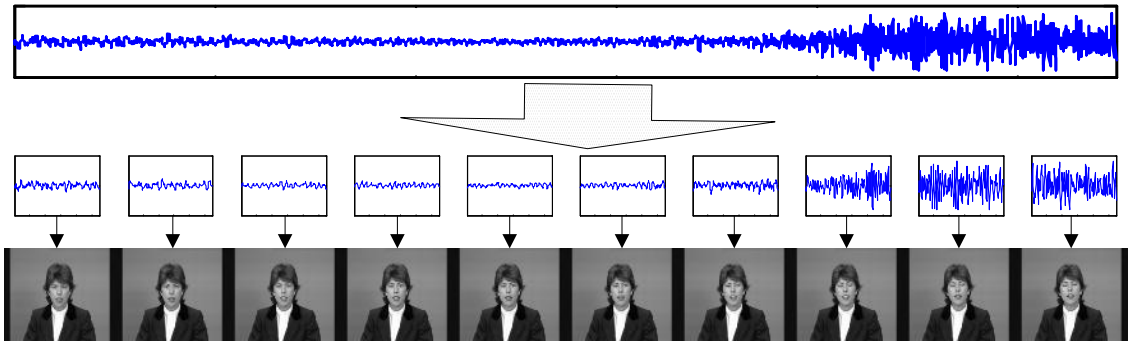


Figure 4-7 La repartitionnement du signal audio et l'insertion dans les trames vidéo

Pour une séquence de vidéo de K trames à *fps* trames par seconde dont la taille d'une trame est $M \times N$ pixels, et pour un signal d'audio à *c* canaux, les paramètres *spf* et *l* sont calculés par :

$$spf = \left\lceil \frac{f_s \times c}{fps} \right\rceil \qquad (3.3)$$

Où f_s est la fréquence d'échantillonnage du signal d'audio. $\lceil . \rceil$ est l'operateur « plafond » qui donne le nombre entier immédiatement supérieur à son argument.

l est calculé par le pseudo code suivant :

Algorithme 5.1

```
cpf ← 0
l ← 1
while (cpf < spf)
      cpf ← cpf + (M + N -((l-1)*4+2));
      l ← l+1
end
l ← (l-1)
```

Les paramètres *spf* et *l* sont calculés une fois pour toute au début puis utilisés lors de toute la procédure de codage. Ces deux paramètres sont également transmis au décodeur pour garantir la reconstruction des trames de vidéo et du signal audio. Suivant l'étape de calcul des paramètres *spf* et *l*, les échantillons $s'(k)$ redimensionnés sont insérés sur le plan Y de la décomposition YCbCr de l'image par l'approche d'insertion linéaire (voir la Figure 4-8).

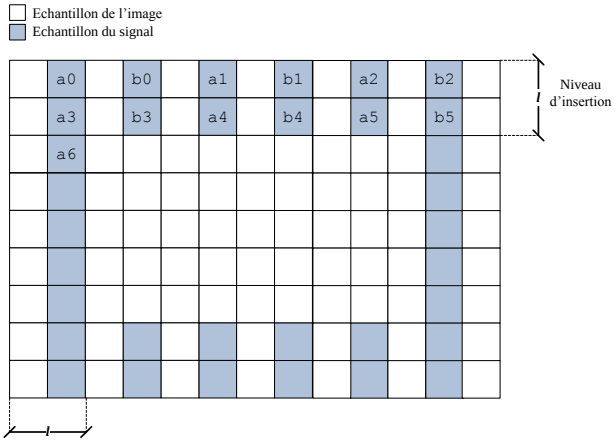


Figure 4-8 L'insertion des échantillons du signal sur le plan Y de la trame

4.4.1.3 Compression par H.264/AVC

Les trames de mélange sont ensuite compressées par le codeur H.264/AVC. Dans cette étape, le codeur en plus de les compresser, transforme les trames en des paquets MPEG. Ces paquets sont transmis ensuite vers un support de transmission ou de stockage. La Figure 4-5 illustre de la procédure de codage avec la méthode proposée.

4.4.2 Procédure de décodage

4.4.2.1 Décompression par H.264/AVC

La Figure 4-9, illustre la vue générale de la procédure de décodage. Dès leur réception, le codeur décompresse les paquets MPEG et obtient les trames à l'étape précédente. A ce stade, les trames décompressées contiennent des images de la séquence initiale et les échantillons du signal. Nous avons besoin de les séparer inversement à ce que nous avons fait lors de l'étape de l'insertion. Pour cela, les trames décompressées sont transformées en YCbCr et le plan Y est mis en entrée de la fonction de séparation.

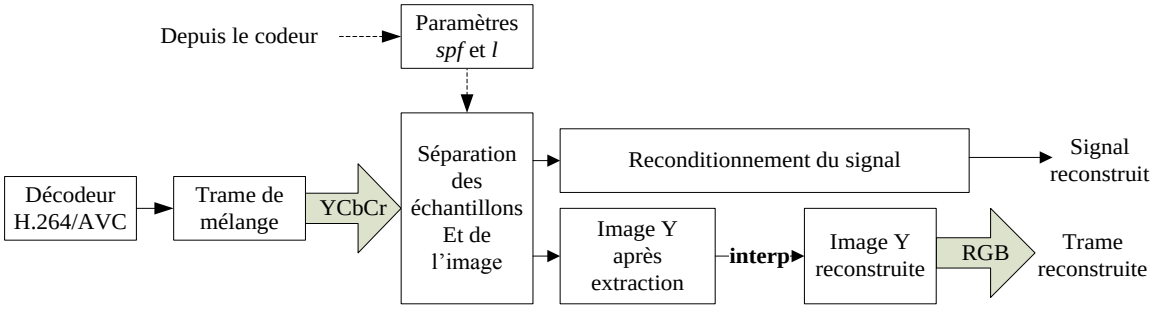


Figure 4-9 Schéma de block résumant la procédure de décodage

4.4.3 Extraction des échantillons du signal

Au niveau de la fonction de séparation, à l'aide du modèle d'insertion utilisé lors du codage, les échantillons du signal embarqué sont extraits un par un avant d'être conditionnés. Cette deuxième étape de conditionnement est pratiquement l'inverse de celle que nous avons vu lors du codage. Elle consiste à rétablir la plage dynamique du signal à son état initial, puis à remettre les échantillons en ordre dans le cas de la présence de plusieurs canaux dans le signal.

Le rétablissement de la plage dynamique du signal est effectué par l'opération suivante :

$$s_r(k) = \frac{s'(k)}{255} \times 2^B \qquad (3.4)$$

4.4.4 Reconstruction des trames de la séquence de vidéo

Une fois le signal extrait, les emplacements occupés par les échantillons du signal sont reconstruits par une approche prédictive. Les pixels sur l'image Y sont reconstruits suivant le modèle illustré sur la Figure 4-8.

4.5 Résultats préliminaires

4.5.1 Capacité d'insertion

Pour une séquence de vidéo à 30 trames par seconde, la capacité d'une trame, à intégrer les échantillons du signal audio « Handel » est donnée par l'équation. (3.3) :

$$spf = \left\lceil \frac{1 \times 8192}{30} \right\rceil = 274$$

Par l'application de l'algorithme 5.1, le niveau de l'insertion est calculé *l=1*.

4.5.2 Qualité des trames reconstruites

Nous avons inséré les échantillons du signal « Handel » dans deux séquences vidéo, à savoir, les séquences « claire » et « foreman ». Les résultats en termes de PSNRs des trames reconstruites après l'extraction des échantillons et la reconstruction sont comparées *par rapport à* la qualité de reconstruction par compression directe via H.264/AVC. Ces résultats sont respectivement présentés pour les séquences « claire » et « foreman » sur les Figures 4-11 et 4- 4-12

Lors de la compression, nous avons opté pour une structure de GOP [**18**] de type «IBPB » dans le codeur H264. Les pics périodiques sur les figures correspondent aux I-Trames sur chaque GOP. Comme attendu, l'information sur les trames I est moins dégradée par rapport aux autres trames dans le même GOP. La qualité de la reconstruction peut être augmentée en choisissant un GOP différent. Les Figures 4-13 et 4-14, présentent certaines trames appartenant à chacune des séquences. L'objectif est de s'intéresser maintenant à la qualité de reconstruction de notre méthode.

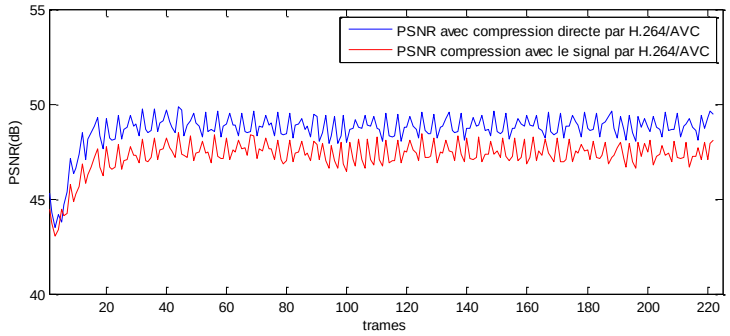


Figure 4-10 Résultats de la qualité de reconstructions des trames de la séquence « claire ».

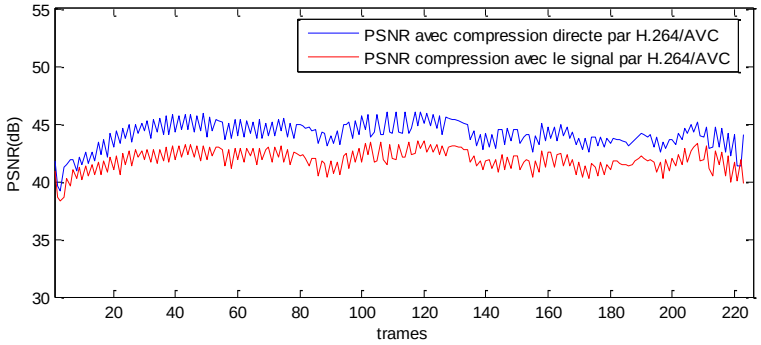
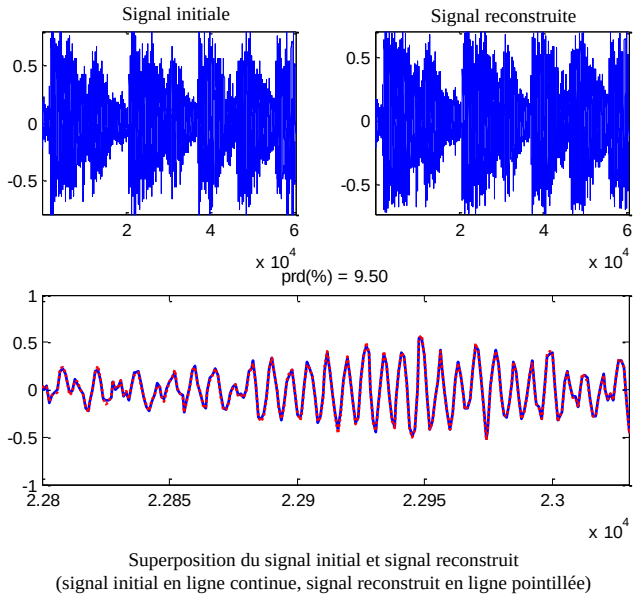


Figure 4-11 Résultats de la qualité de reconstructions des trames de la séquence « foreman ».

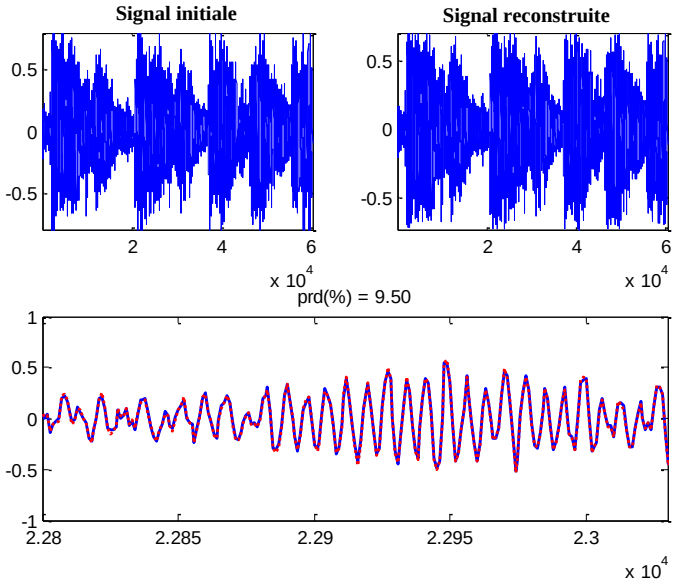
4.5.3 Qualité du signal audio reconstruit

Les résultats expérimentaux, présentées dans la §2.5, justifient que les échantillons insérés sur les trames vidéo peuvent être faiblement transférés sur un support de transmission. Car une bonne qualité de reconstruction des trames entraine également une bonne qualité de reconstruction du signal. Les 4-15 et 4-16 montrent le signal initial et le signal reconstruit à partir des trames des séquences « claire » et « foreman » respectivement. Sur les deux figures, nous observons que les versions initiale et reconstruite du signal, sont quasi-identiques, à part certains échantillons légèrement erronés.



Superposition du signal initial et signal reconstruit
(signal initial en ligne continue, signal reconstruit en ligne pointillée)

Figure 4-12 Echantillons du signal reconstruits à partir des trames de la séquence « claire »



Superposition du signal initial et signal reconstruit
(signal initial en ligne continue, signal reconstruit en ligne pointillée)

Figure 4-13 Echantillons du signal reconstruits à partir des trames de la séquence « foreman »

4.5.4 Vitesse de codage

Nous avons effectué des tests pour mesurer la vitesse d'exécution de codage avec la méthode proposée et la comparer avec celle du H.264 (seul).Les tests ont été effectués sous Matlab sur un ordinateur Intel Centrino avec 2Go de mémoire. Nous avons utilisé la libraire *ffmpeg* pour le codage H.264. Comme il est montré sur le Tableau XIV, la vitesse de codage de la CMV est comparable à celle du H.264.

TABLEAU XIV
LA VITESSE DE CODAGE POUR LES TRAMES « CLAIRE » ET « FORMAN » AVEC H.264 ET AVEC LA METHODE PROPOSEE

Séquence (266 trames)	**H.264 (seconds)**	**CMV (seconds)**
Claire	23.61	25.82
Forman	23.16	25.87

4.5.5 Expérimentations en temps réels

Nous avons développé un logiciel pour pouvoir valider le bon fonctionnement de la CMV dans le cas réel. Ce logiciel étant écrit entièrement en langage C peut être trouvé sur [**60**] avec des vidéos de test qui sont compressées avec notre méthode.

4.5.6 Conclusion

Dans cette section, nous avons présenté une nouvelle approche pour compresser conjointement une séquence vidéo avec sa bande audio. Cette méthode de compression peut être très facilement adaptée à des applications du type Video-over-IP. La méthode proposée est robuste à un codec vidéo avec perte comme H.264/AVC. Avec la méthode proposée, la tâche complexe de démultiplexer et de synchroniser les paquets MPEG est évitée.

Un autre intérêt réside dans le fait que le processus de synchronisation entre la bande sonore et les trames vidéo est automatique. Mais cette synchronisation automatique engendre un certain nombre de problème. Nous avons vu que pour une application réelle, cela rend des fonctions comme « avancer » et « rembobiner » plus difficile à implémenter quand il s'agit d'implémenter la méthode sur un baladeur, car de telles fonctions impose un « flush » [**19**] sur la trame courante et la recherche d'une autre dans le flux MPEG. Le problème est qu'une trame vidéo peut être composée de plusieurs paquets MPEG, donc un délai d'attente est obligatoire, tant que tous les paquets sont décodés par le codeur et séparés par la fonction de séparation.

Conclusions et Perspectives

Dans cette thèse, nous avons abordé la problématique de la compression conjointe de l'image et du signal en utilisant un seul codeur (e.g. JPEG 2000). Il s'agit du concept de la « Compression Multimodale ».

Nous avons étudié en suite deux types de réalisation possible pour la Compression Multimodale, notamment « le modèle basé sur la modification d'un codeur existant », ou « le modèle consistant à introduire une étape en amont du codeur ». Le modèle qui consiste à introduire une étape de traitement a été retenu. Ce modèle est plus facile à intégrer dans un système de codec existant car il s'agit de reconditionner les échantillons du signal embarqué avant de les insérer dans une image porteuse, de manière à ce qu'ils provoquent la moindre dégradation possible sur les images reconstruites.

Dans ce contexte, nous avons introduits deux approches différentes, à savoir, l'approche d'insertion des échantillons dans le domaine fréquentiel ainsi que l'approche d'insertion dans le domaine spatial. Dans le domaine fréquentiel, les échantillons du signal suivant une étape de reconditionnement ont été insérés, dans la partie *HH* de la décomposition en ondelette de l'image. La méthode est dépendante d'un paramètre α qui doit être transmis au codeur lors du codage et du décodage. Dans l'étude que nous avons réalisée au chapitre 3, nous avons constaté qu'un paramètre $\alpha=30$ donne des résultats satisfaisants.

La deuxième approche consistait à insérer les échantillons du signal dans le plan Y de la décomposition en YCbCr de l'image, dans le domaine spatial. Dans les approches spatiales, nous avons utilisés quatre méthodes, la première méthode consistait à insérer les échantillons sur les contours du plan Y, est appelée la méthode d'insertion spirale. Pour la deuxième approche (étant une dérivée de la première) il s'agit d'insérer les échantillons sur les contours suivant un modèle linéaire. Nous avons montré que le modèle linéaire est plus adapté à une intégration dans un ordinateur, étant donné la gestion de la mémoire. La troisième méthode dans l'approche spatiale est basée sur le fait qu'une image possède des régions hétérogènes au sens de la composition

fréquentielle. Nous avons montré que dans cette méthode les régions homogènes de l'image peuvent être utilisées pour accueillir les échantillons d'un signal embarqué. Cette méthode est appelée « méthode supervisée », car l'utilisateur est responsable du choix de la région homogène sur une image de référence dans laquelle les échantillons sont insérés. La quatrième méthode qui est une dérivée de la troisième, consiste à détecter la région d'insertion de façon quasi-optimale par une méthode de décomposition d'image appelée « la décomposition en quadtree ».

Au chapitre 3, les résultats obtenus sur les images et les échantillons du signal montrent que (pour un bas débit, fixé *a priori*) compresser un mélange par un codeur unique est plus intéressant en terme de performance (débit binaire – distorsion) que de compresser séparément l'image par JPEG2000 et le signal par un autre codeur spécifique.

Les méthodes de Compression Multimodale que nous avons étudiées dans cette thèse, sont orientées à la compression d'une modalité 2D (l'image) avec une modalité 1D (le signal). Mais l'idée peut être étendue très facilement à la compression vidéo. C'est-à-dire, la compression d'une séquence vidéo avec un signal audio. Ainsi une modalité 2D+t (la vidéo) et le signal (1D).

Dans le contexte de la Compression Multimodale, l'idée générale est de compresser une séquence avec son signal complémentaire, dans ce cas, il s'agit d'une séquence d'image avec sa bande audio. La même idée peut être utilisée dans d'autre contexte comme par exemple, dans le cadre d'une application biomédicale, cela pourrait être une séquence d'échocardiogramme avec un signal d'ECG.

Annexes

A.1. Interpolation Spline

En analyse numérique (et dans son application algorithmique discrète pour le calcul numérique), l'interpolation est une opération mathématique permettant de construire une courbe à partir de la donnée d'un nombre fini de points, ou une fonction à partir de la donnée d'un nombre fini de valeur. La solution du problème d'interpolation passe par les points prescrits, et suivant le type d'interpolation, il lui est demandé de vérifier des propriétés supplémentaires.

L'essence de l'interpolation est de représenter une fonction quelconque en continue comme une somme discrète des fonctions de base pondérées et décalées. L'approche traditionnelle à l'interpolation impose que ces fonctions satisfasse la propriété d'interpolation, et de nombreux chercheurs à mis un effort signifiant afin de les optimiser sous des contraintes spécifiques [**67**] [**68**].

Une question importante est le choix adéquat de ces fonctions de base(ou fonctions de synthèse). Dans l'interpolation au sens traditionnel les échantillons f_k donnent les coefficients d'interpolations pour les fonctions de synthèse.

$$f(x) = \sum_{k \in \mathbb{Z}^q} f_k \varphi_{\text{int}}(x-k) \qquad (5.1)$$

L'Eq. (5.1) indique le cas où les fonctions de synthèse sont utilisés comme interpolant.(i.e. ils disparaissent à tout points entiers autres que l'origine) $\varphi_{\text{int}}(k) = 0, \quad \forall k \neq 0, k \in \mathbb{Z}^q$ .Evaluation de *f(x)* sur les points entiers donne :

$$f(k_0) = \sum_{k \in \mathbb{Z}^q} f_k \varphi_{\text{int}}(x-k_0) \qquad (5.2)$$

Ce qui revient la même chose qu'une convolution discrète :

$$f(k_0) = (f * \varphi_{\text{int}})(k_0) \qquad (5.3)$$

Un exemple classique pour la φ_{int} pour l'interpolation au sens traditionnel est la fonction *sinc* , dans un tel cas toute fonction synthétisée sont intrinsèquement limitée en bande.

Comme une approche alternative, il existe aussi le model d'interpolation généralisée, elle est de forme :

$$f(x) = \sum_{k \in \mathbb{Z}^q} c_k \varphi(x-k) \qquad (5.4)$$

L'interpolation au sens traditionnel est un cas spécial de l'interpolation généralisée quand on choisit $c_k = f_k$ et $\varphi = \varphi_{\text{int}}$ .La différence entre la formulation classique (5.2) et la formulation généralisée (5.4) est l'introduction des coefficients c_k au lieu des échantillons du signal f_k . Cela permet nouveaux possibilités, dans le sens que l'interpolation peut être effectuée dans deux étapes. Premièrement la détermination de c_k à partir des échantillons f_k , et la deuxième la détermination des valeurs désirées $f(x)$ à partir des coefficients c_k . L'avantage de cette séparation est de permettre un choix excellent pour les fonctions de base, qui représentent des propriétés meilleures que celles qui sont disponible dans le cas limité de l'interpolation traditionnelle [**40**]. On peut aussi observer que l'évaluation de $f(x)$ dans le model généralisé nous donne également une convolution discrète :

$$f(k_0) = \sum_{k \in \mathbb{Z}^q} c_k \varphi(x-k_0) = \sum_{k \in \mathbb{Z}^q} c_k p_{k_0-k} \qquad (5.5)$$

En élargissant l'eq. (5.5) pour toute *k*, et en limitant la φ sur un support fini, les coefficients c_k peuvent être trouvés avec la méthode classique, avec la forme matricielle :

$$\mathbf{c = P^{-1} f} \qquad (5.6)$$

Une autre stratégie pour résoudre les coefficients c_k survient en reconnaissant que l'Eq. (5.5) est une convolution discrète :

$$f(k_0) = (c * p)(k_0) \Leftrightarrow c_{k_0} = (p^{-1} * f)(k_0) \qquad (5.7)$$

où p^{-1} satisfait la condition : $(p * p^{-1})(k_0) = \delta_k$ .Cette approche de considérer l'interpolation comme un filtrage numérique est beaucoup plus efficace pour évaluer. Un exemple de l'approximation d'un signal quelconque par l'interpolation *spline* avec le modèle de filtrage numérique est donné sur la Figure A-1

Les *B-splines* est une famille de fonctions de synthèse [**40**] (Figure A-2). Ce sont des polynômes de dégrée *n*. Un *B-spline* de dégrée *n* est dénoté comme $\beta^n(x)$. Les *B-splines* ont des propriétés très intéressantes, une étude approfondie sur leurs propriétés peut être trouvées dans [**40**]. Dans le cadre de trouver une approximation

optimale afin d'obtenir l'image interpolée, on s'intéresse plutôt à l'interpolation par des *B-splines cubique* [**40**]. Ils sont définis comme suit :

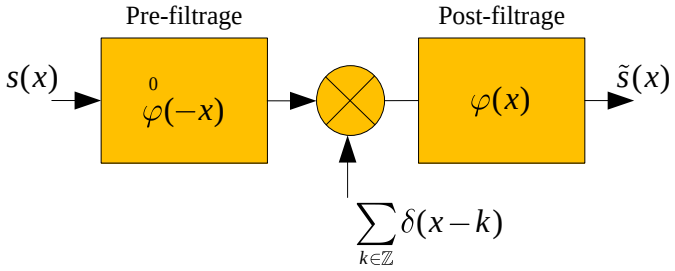


Figure A-1 L'approximation d'un signal s(x) quelconque avec l'interpolation par spline des moindres carrés. L'approximation de s(x) au sens de minimum des moindres carrés est atteinte par une projection orthogonale. Ceci conduit à la pre-filtrage du signal avant l'échantillonnage avec le filtre $\varphi^{\circ}(-x)$, ce qui est le dual du $\varphi(x)$.

$$\beta^3 = \beta^0 * \beta^0 * \beta^0 * \beta^0 = \begin{cases} \frac{2}{3} - \frac{1}{2}|x|^2 (2-|x|) & ,0 \le |x| < 1 \\ \frac{1}{2}(2-|x|)^3 & ,1 \le |x| < 2 \\ 0 & ,2 \le |x| \end{cases} \qquad (5.8)$$

Les coefficients B-splines cubique que l'on utilise afin d'obtenir l'image-interpolée I_{re} se calcule comme suit. Pour la simplicité, on présente les calculs pour un signal 1-D, mais les calculs seraient de même pour une image. En substituant p par β^3 dans l'eq. (5.7) . Nous avons :

$$f(k_0) = (c * \beta^3)(k_0) \Leftrightarrow c_{k_0} = \left((\beta^3)^{-1} * f\right)(k_0) \qquad (5.9)$$

Et en utilisant la propriété de transformée en Z :

$$F(z) = C(z)B^3(z) \Leftrightarrow C(z) = (B^3)^{-1}(z)F(z) \qquad (5.10)$$

Où $B^3(z)$ est la transformée en Z de $\beta^3(x)$ **.**Soit $b_m^n(x)$ $b_m^n(x)$ est le noyau du B-spline discret ;

$$b_m^n(x) = \beta^n(x/m)\Big|_{x=k} \qquad (5.11)$$

En échantillonnant le B-spline cubique sur points entiers, nous obtenons :

$$B_1^3(z) = \sum_{k\in\mathbb{Z}} b_m^3(k) z^{-k}\Big|_{m=1} = (z + 4 + z^{-1})/6 \qquad (5.12)$$

Par conséquent, le filtre obtenu serait :

$$(B_1^3)^{-1}(z) = \frac{6}{z+4+z^{-1}} = 6\left(\frac{1}{1-z_1 z^{-1}}\right)\left(\frac{-z_1}{1-z_1 z}\right) \qquad (5.13)$$

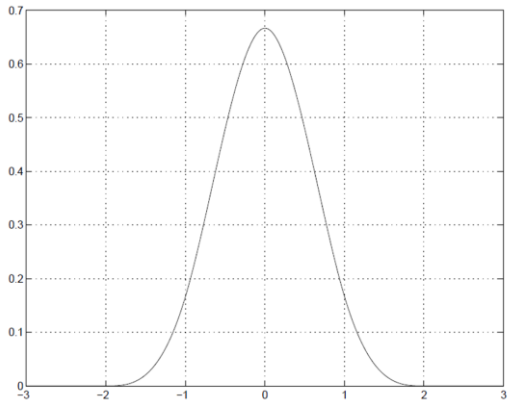


Figure A-2 La fonction de synthèse B-spline cubique.

Bibliographie

[1] A. K. Jain, *Fundemantals of digital image coding*.: Prentice Hall, 1989.

[2] David A. Huffmann, "A methode for construction of minimum redundancy codes," in *Proceedings of the I.R.E*, 1954, pp. 1098-1101.

[3] Mohammed Ghanbari, *Standard Codecs:Image Compression to Advanced Video Coding*.: IEE Telecommunication Series, 2003.

[4] W. B. Pennebaker and J. L. Mitchell, *JPEG: still image compression standard*. New York: Van Nostrand Reinhold, 1993.

[5] ITU, "Draft ITU-T Recommendation H.263, video coding for low bit rate communication," September 1997.

[6] R Lukac and N, P Kostantinos, *Color Image Processing: Methods and Applications*.: CRC Press, 2006.

[7] W. Chen, C. Smith, and S. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Transactions on Communication*, pp. 1004-1009, 1979.

[8] M Kunt, *Traitement numérique des signaux*.: Presses Polytechniques et Universitaires Romandes, 1999.

[9] Independent JPEG Group. (1998, March) The sixth public release of the Independent JPEG Group's free JPEG Software,C source code of JPEG encoder release 6b.

[10] M. W. Marcellin, M. J. Gormish, A. Bilgin, and M. P. Boliek, "An overview of JPEG-2000," in *Data Compression Conference Proceedings*, March 2000, pp. 523-541.

[11] D Le Gall and A Tabatabai, "Subband coding of images using symmmetric short kernel filters and arithmetic coding tehcniques," in *International conference on Acoustic, speech and signal processing*, 1988, pp. 761-764.

[12] I. Daubechies, *Ten Lectures on Wavelets*.: SIAM, 1992.

[13] I Daubechies, "The wavelet transform, time frequency localization and signal analysis," vol. 36, no. 5, pp.

961-1005, 1990.

[14] F. Ebrahimi, M. Chamik, and S. Winkler, "JPEG vs. JPEG2000: An objective comparison of image encoding quality," in *Applications of digital image processing. Conference No27*, Denver, 2004, pp. 300-308.

[15] I. Moccagatta, S. Soudagar, J. Liang, and H. Chen, "Error-resilient coding in JPEG-2000 and MPEG-4," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 899 - 914, Jun 2000.

[16] H.261, ITU-T Recommendation H.261, video codec for audiovisual services at p x 64 kbits/s , 1990.

[17] H.263, H.263 ITU-T Recommendation H.263, video coding for low bit rate communication, September 1997.

[18] T Wiegand, G,J Sullivan, G Bjøntegaard, and A Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. on circuits and systems for video technology*, vol. 13, no. 7, pp. 560-576, JULY 2003.

[19] MPEG-1, Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mb/s, November 1991.

[20] MPEG-2, Generic coding of moving pictures and associated audio information, November 1994.

[21] MPEG-4, Testing and evalutaion procedures document, July 1995.

[22] T Ishiguro and K Iinuma, "Television bandwitdth compression transmission by motion-compensated interframe coding," *IEEE Communication Magazine*, no. 10, pp. 24-30, 1982.

[23] S Kappangantula and K,R RAO, "Motion compensated predicitive coding," in *Proceeding of international technical symposium, SPIE*, San Diego, 1983.

[24] H,C Bergmann, "Displacement estimation based on the correlation of image segments," in *IRE conference on the Electronic image processing*, York, 1982.

[25] J,R Jain and K Jain A, "Displacement measurement and its application in interframe image coding," *IEEE Trans on Commun.*, no. 29, pp. 1799-1808, 1981.

[26] T Shanableh and M Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Trans. on Multimedia*, vol. 2, no. 2, pp. 101-110, 2002.

[27] A. Nait-Ali, "Advanced Biomedical Signal Processing," , 2007.

[28] A. Naït-Ali, E, H Zeybek, and X. Drouot, "Introduction to Multimodal Compression of Biomedical data," in *Advanced Biosignal Processing*.: Springer, 2009, pp. 353-375.

[29] I Cox, M Miller, J Bloom, Fridrich J, and Kalker T, *Digital Watermarking and Steganography 2nd Ed.*: Morgan Kaufmann, 2007.

[30] Chun-Hsiang Huang, Shang-Chih Chuang, Yen-Lin Huang, and Ja-Ling Wu, "Unseen Visible Watermarking: A Novel Methodology for Auxiliary Information Delivery via Visual Contents," in *Information Forensics and Security, IEEE Transactions on* , 2009 , pp. 193 - 206.

[31] R. Lancini, F. Mapelli, and S. Tubaro, "Embedding indexing information in audio signal using watermarking technique," in *Video/Image Processing and Multimedia Communications 4th EURASIP-IEEE Region 8 International Symposium on VIPromCom* , 2002 , pp. 257 - 261.

[32] K. Ramani, E.V. Prasad, S. Varadarajan, and A. Subramanyam, "A Robust Watermarking Scheme for Information Hiding," in *Advanced Computing and Communications, 2008. ADCOM 2008. 16th International Conference on* , Chennai , 2008, pp. 58 - 64.

[33] F,Y Shih, *Digital Watermarking and Steganography: Fundamentals and Techniques*.: CRC Press, 2007.

[34] N,F. Johnson, Z Duric, and S Jajodia, *Information Hiding:Steganography and Watermaking - Attacks and Countermeasures*.: Springer, 2000.

[35] D,F Walnut, *An introduction to wavelet analysis*.: Birkhäuser, 2002.

[36] S. Mallat, *A Wavelet Tour of Signal Processing*.: Academic Press, 1999.

[37] Emre H. Zeybek, Amine Naït-Ali, Christian Olivier, and A. Ouled-Zaid, "A Novel Scheme for joint Multi-channel ECG-ultrasound image compression," in *Proc. of Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, Lyon, 2007.

[38] E H Zeybek, A Naït-Ali, C Olivier, and A Ouled Zaid, "Compression conjointe image échographique-signaux ECG multivoies par JPEG2000," in *CORESA 2007 Actes du colloque*, Montpellier, 2007, pp. 38-43.

[39] P. N. Topiwala, *Wavelet Image and Video Compression*.: Kluwer Academic Publishers, 1998.

[40] P. Thévenaz, T. Blu, and M. Unser, "Interpolation Revisited," *IEEE Transactions on Medical Imaging*, vol. 19, no. 7, pp. 739-758, July 2000.

[41] Vijav, V Vazirani, *Approximation Algorithms*. Berlin: Springer, 2003.

[42] J. S. Chang and C. K. Yap, "A polynomial solution for the potato-peeling problem," *Discrete Comput. Geom. 1*, pp. 155-182, 1986.

[43] M. McKenna, J. O'Rourke, and S. Suri, "Finding the largest rectangle in an orthogonal polygon," in *Proceedings on 23rd Allerton Conference on Communication, Control and Computing*, 1985, pp. 486-495.

[44] D. Wood and C. K. Yap, "The orthogonal convex skull problem," *Discrete Comput. Geometry 3*, pp. 349-365, 1988.

[45] N. Amenta, "Bounded boxes, Hausdorff distance, and a new proof of an interesting Helly-type theorem," in *Proc. 10th ACM Symp. on Computational Geometry*, 1994, pp. 340-347.

[46] F. Raphael and J. L. Bentley, "Quad Trees: A Data Structure for Retrieval on Composite Keys," *Acta Informatica 4*, 1974.

[47] A Chetouani, G. Mostafaoui, and A. Beghdadi, "A New Free Reference Image Quality Index Based on Perceptual Blur Estimation," in *Advances in Multimedia Information Processing.*: SpringerBerlin / Heidelberg, 2009, pp. 1185-1196.

[48] A. Chetouani and A. Beghdadi, "Image Quality Assessment based on Distortion Identification," in *Proceedings of SPIE Volume: 7867*, San Francisco, 2011, pp. 23 - 27.

[49] A. Chetouani, A. Beghdadi, and M. Deriche, "Statistical Modeling of Image Degradation Based on Quality Metrics," *ICPR* , pp. 714-717, 2010.

[50] A. Chetouani, M. Deriche, and A. Beghdadi, "Classification of image degradation using multiple image quality metrics and lineardiscriminant analysis," in *EUSIPCO*, Aalborg, 2010.

[51] A. Lahoulou, E. Viennet, and A. Beghdadi, "Selecting Low-level Features for Image Quality Assessment by Statistical Methods," *CIT. Journal of computing and information technology*, vol. 18, no. 2, pp. 183-189, 2010.

[52] A. C. Bovik, H. R. Sheikh and E. P. Simoncelli Z. Wang, ""Image quality assessment: From error visibility to structural similarity"," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, 2004 April.

[53] A. Loza, L. Mihaylova, N. Canagarajah, and D. Bull, "Structural Similarity-Based Object Tracking in Video Sequences," in *Proceedings of the 9th International Conference on Information Fusion*, Florence , 2006 , pp. 1 - 6.

[54] H.R. Sheikh and A.C. Bovik, "Image information and visual quality," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430 - 444, January 2006.

[55] N. Damera-Venkata, T. D. Kite, W. S. Geisler, B. L. Evans, and A. C. Bovik, "Image Quality Assessment Based on a Degradation Model," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 636-650, April 2000.

[56] R. Franzen. Kodak Lossless True Color Image Suite. [Online]. http://r0k.us/graphics/kodak/

[57] G.,B. Moody, R.,G. Mark, and A.,L. Goldberger, "PhysioNet: a Web-based resource for the study of physiologic signals," *IEEE Eng in Medicine and Biology* , vol. 20, no. 3, pp. 70-75, May-June 2001.

[58] Amine Naït-Ali, Christine Cavaro-Menard, and Emre Zeybek. (2007) MeDEISA. [Online]. http://www.medeisa.net

[59] E. H. Zeybek, A. Naït-Ali, and R Fournier, "A novel multimodal scheme for video and audio compression using H.264/AVC," *IEEE Trans on Multimedia (soumis)*, 2010.

[60] E. H. Zeybek, A Naït-Ali, and R Fournier, MAVC Player a multimodal audio video player, December 2009.

[61] K Brandenburg, "MP3 and AAC explained," in *International Conference on High Quality Audio Coding*, 1999, pp. 1-12.

[62] T.D. C. Little and A. Ghafoor, "Syncronization and Storage Models for Multimedia Objects," *IEEE Journal on Selected Areas in Communication*, vol. 8, no. 3, pp. 413-427, April 1990.

[63] B. K. Schmidt, J.D Northcutt, and M.D Larn, "A Method and Apparatus for Measuring Media Synchronization," in *Proc. of the 5th International Workshop on Network and Operating System Support for Digital Audio and Video*, 1995, pp. 130-141.

[64] M. Yang, N. Bourbakis, Z. Chen, and M. Trifas, "An efficient audio-video synchronization methodology," in *in Proc. of IEEE International Conference on Multimedia and Expo*, Beijing , 2007, pp. 767-770.

[65] S. G. Ally and A. Youssef, "Real-Time Motion-based Frame Estimation in Video Lossy Transmission," in *Proc. of IEEE Symposium on Applications and Internet*, 2001, pp. 139-147.

[66] M Yang and N Bourbakis, "A Prototyping Tool for Analysis and Modeling of Video Transmission Trace over IP Networks," in *Proc of IEEE Int. Workshop on Rapid System Prototyping*, Crete, 2006.

[67] C. R. Appledorn, "A new approach to the interpolation of sampled data," *IEEE Transactions on Medical*

Imaging, vol. 15, no. 3, p. 369–376, June 1996.

[68] N. A. Dodgson, "Quadratic interpolation for image resampling," *IEEE Transactions on Image Processing*, vol. 6, no. 9, p. 1322–1326, Sep. 1997.

[69] ISO, Visual evaluation of JPEG-2000 color image compression performance, March 2000.

[70] (Editor) Nait-Ali A, "Advanced Biomedical Signal Processing," , 2007.

[71] A. Naït-Ali, E H Zeybek, and X. Drouot, "Introduction to Multimodal Compression of Biomedical data ," in *Advanced Biosignal Processing*.: Springer, 2009, pp. 353-375.

[72] G. Strang and T. Nguyen, *Wavelets and Filter Banks*.: Wellesley-Cambridge Press, 1996.

[73] P Vaidyanathan, *Multirate Systems and Filter Banks*.: Prentice-Hall, 1993.

[74] K. Daniels, V. Milenkovic, and D. Roth, "Finding the largest area axis-parallel rectangle in a polygon," *Computational Geometry 7*, pp. 125-148, 1997.

[75] B Chazelle, R L Drysdale III, and D T Lee, "Computing the largest empty rectangle," *SIAM Journal of Comp.*, pp. 300-315, 1986.

[76] Video Trace Library. [Online]. http://trace.eas.asu.edu/yuv/index.html

[77] A. Beghdadi and R. Iordache, "Image quality Assessment using the Joint Space/ Spatial-Frequency Space," *EURASIP Journal on Applied Signal Processing*, p. 8, 2006.