ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2012-ENST-056

EDITE - ED 130

Doctorat ParisTech

T H E S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Signal et Images »

présentée et soutenue publiquement par

Manel ABID

le 5 Octobre 2012

Codage/décodage source-canal conjoint des contenus multimédia

Directeur de thèse : **Béatrice PESQUET-POPESCU**
Co-direction de la thèse : **Maria TROCAN**

Jury

M. Fabrice LABEAU	Rapporteur
M. Peter SCHELKENS	Rapporteur
Mme Anissa MOKRAOUI	Examinateur
M. François-Xavier COUDOUX	Examinateur
M. Michel KIEFFER	Examinateur

THÈSE

TELECOM ParisTech
école de l'Institut Télécom - membre de ParisTech

Contents

Remerciements

Je remercie tout d'abord mes directrices de thèse, Béatrice Pesquet-Popescu et Maria Trocan, pour m'avoir fait confiance et pour m'avoir guidée, encouragée, conseillée tout au long de ces trois années. Je tiens de plus à remercier Michel Kieffer pour sa précieuse collaboration au cours de ma thèse.

J'adresse également mes remerciements à tous le membres de mon jury : François-Xavier Coudoux qui m'a fait l'honneur de présider ce jury, Peter Schelkens et Fabrice Labeau qui ont bien voulu accepter la charge de rapporteur et Anissa Mokraoui qui a bien voulu juger ce travail.

Je souhaite de plus remercier toutes les personnes que j'ai pu côtoyer et avec lesquelles j'ai pu collaborer durant mes trois années de thèse au département TSI de Télécom Paris Tech, et plus spécifiquement, tous les membres de l'équipe multimédia.

Enfin, je remercie du fond du coeur mes chers parents qui n'ont jamais cessé de croire en moi pendant toutes mes années d'études et qui m'ont toujours encouragée à aller de l'avant. Je leur dédie cette thèse.

Abstract

The development of multimedia broadcasting and on-demand services for mobile devices such as tablets or smartphones involves the transmission of contents over heterogeneous networks, consisting of mixed wired and wireless channels. For such best-effort networks, the quality of service (packet loss rate, delay and reconstructed signal quality) is not always satisfactory due to time-varying characteristics of the channels. Compressed data packets may be lost due to congestion in the network or corrupted by channel impairments. Moreover, due to the limited bandwidth, the multimedia content has to be highly compressed, which makes the transmitted bit streams extremely sensitive to transmission impairments. Therefore, the demand for efficient compression algorithms, as well as reliable coding techniques, is very important in multimedia transmission systems.

This thesis aims at proposing and implementing efficient joint source-channel coding and decoding schemes in order to enhance the robustness of multimedia contents transmitted over unreliable networks.

In a first time, we propose to identify and exploit the residual redundancy left by wavelet video coders in the compressed bit streams. An efficient joint-source channel decoding scheme is proposed to detect and correct some of the transmission errors occurring during a noisy transmission. This technique is further applied to multiple description video streams transmitted over a mixed architecture consisting of a wired lossy part and a wireless noisy part.

In a second time, we propose to use the structured redundancy deliberately introduced by multirate coding systems, such as oversampled filter banks, in order to perform a robust estimation of the input signals transmitted over noisy channels. Two efficient estimation approaches are proposed and compared. The first one exploits the linear dependencies between the output variables, jointly to the bounded quantization noise, in order to perform a consistent estimation of the source outcome. The second approach uses the belief propagation algorithm to estimate the input signal via a message passing procedure along the graph representing the linear dependencies between the variables. These schemes are then applied to estimate the input of an oversampled filter bank and their performance are compared.

Résumé de la Thèse

Contexte et motivations

Le développement croissant d'applications multimédia, de services à la demande et de terminaux mobiles (tablettes et smartphones) a conduit à un usage intensif d'architectures mixtes, comprenant canaux radio-mobiles et réseaux à pertes de paquets comme Internet. Pour de tels supports de communication, la qualité de service n'est pas toujours garantie à cause des variations des caractéristiques de la source et des conditions du canal. Les paquets de données peuvent être perdus au cours de leur acheminement, suite à des congestions survenant sur la partie Internet du canal de transmission, ou/et corrompus par des erreurs causées par des perturbations sur la partie radio-mobile. Garantir une transmission fiable des contenus multimédia devient alors d'une grande necessité, d'autant plus que les applications conversationnelles ou de type diffusion, largement utilisées de nos jours, ne permettent pas la retransmisssion de l'information perdue ou erronée.

Pour augmenter la robustesse des contenus transmis, les recherches se sont pendant longtemps axées sur l'optimisation séparée du codeur source et du codeur canal, appliquant ainsi le théorème de Shannon [141] qui établit qu'une telle séparation permet au système de communication d'atteindre les performances optimales et de garantir ainsi une transmission fiable avec une probabilité d'erreur aussi petite que l'on veut. Cependant, cette optimisation séparée, suppose que les caractéristiques de la source et du canal sont parfaitement connues, ce qui n'est généralement pas le cas en pratique. De plus, elle fait l'hypothèse que les codeurs source et canal travaillent sur des blocs de données de tailles infiniment longues, ce qui conduit à une complexité élevée des deux codeurs, prohibitive pour les situations de communication pratiques, telles que les applications en temps réel. Dans de telles situations, le codeur canal a une complexité limitée, ce qui ne permet pas de corriger toutes les erreurs de transmission. Par ailleurs, la théorie de Shannon ne fournit pas de méthode de construction de codes source optimaux, capables de corriger les erreurs résiduelles laissées après décodage canal. Ces erreurs peuvent alors fortement dégrader le signal reconstruit par le décodeur source.

Ainsi, les dernières décennies ont connu le développement de solutions alternatives reposant sur des techniques de codage/décodage source-canal conjoint [44, 49]. Ces techniques ont pour objectif de répondre aux contraintes de délai et de complexité imposées en pratique, et d'assurer une transmission robuste vis-à-vis des perturbations inconnues et variables du canal de communnication.

Cette thèse s'inscrit dans ce contexte et a pour but de proposer des schémas de codage/décodage source-canal conjoint, augmentant la robustesse des contenus multimédia transmis sur des canaux radio-mobiles ou mixtes Internet et radio-mobiles, qui sont peu fiables.

Codage/décodage source-canal conjoint

Ces techniques se répartissent essentiellement en deux catégories

1. *Codage source-canal conjoint :* dans cette catégorie, les codeur source et canal sont construits de manière conjointe. Le codeur canal peut être informé des degrés de sensibilité des données compressées qui lui sont délivrées par le codeur source, et ainsi adapter son niveau de protection. On parle alors de schémas de protection inégale vis-à-vis des erreurs [20, 24, 26, 12]. A l'inverse, le codeur source peut être construit pour un canal de transmission donné. Tel est le cas des schémas de codage par descriptions multiples, conçus pour les canaux à pertes de paquets et pour les applications ayant des exigences de délais très réduits [165, 115, 43, 52, 157]. Dans ces schémas, le codeur source génère plusieurs descriptions, qui peuvent être décodées indépendemment les unes des autres, ou ensemble, permettant ainsi une meilleure reconstruction. D'autres schémas de codage source-canal conjoint sont basés sur des transformations redondantes de la source à coder, ce qui revient en quelque sorte à réaliser un codage canal avant le codage source [56, 116, 79]. Les erreurs résiduelles laissées par un décodeur canal classique peuvent alors être détectées et corrigées grâce à cette redondance *structurée* introduite au niveau de la source. On peut citer l'exemple des bancs de filtres suréchantillonnés [151] qui fournissent une représentation redondante du signal placé en entrée, pouvant être exploitée au décodeur pour détecter et corriger les erreurs de transmission et/ou compenser les effacements [56, 131, 54, 79, 86, 130, 2].
2. *Décodage source-canal conjoint :* les techniques appartenant à cette catégorie se basent sur le fait que tous les codeurs source pratiques produisent des trains binaires contenant une certaine redondance [137]. Cette redondance peut provenir de la syntaxe du codeur source [138, 168], de la corrélation résiduelles entre les coefficients après transformation, ou encore de la paquétisation des données [11, 91, 136]. Les techniques de décodage source-canal conjoint cherchent alors à exploiter au mieux cette redondance pour améliorer le décodage [62, 22, 42]. Un exemple de telles méthodes, est le décodage souple des codes à longueurs variables (CLVs) de type Huffman et arithmétiques [76, 122]. Le décodage souple des CLVs se base sur l'estimation statistique de la séquence émise par la source, à partir d'un ensemble d'observations bruitées. Généralement, un critère de vraisemblance ou de probabilité a posteriori est considéré pour optimiser l'estimateur. La recherche de la solution s'effectue alors dans l'ensemble des séquences vérifiant un certain nombre de contraintes, déduites à partir de

la redondance résiduelle présente dans les CLVs. Des algorithmes de décodage tels que l'algorithme de Viterbi [114], sont souvent utilisés pour effectuer cette recherche et éliminer les séquences *non valides*.

Les schémas proposés dans cette thèse appartiennent aux deux catégories citées ci-dessus. Nous nous sommes intéressés dans un premier temps aux techniques de décodage source-canal conjoint des trains binaires délivrés par des codeurs vidéo par ondelettes. La redondance résiduelle laissée par de tels codeurs est identifiée, puis exploitée pour détecter et corriger les erreurs de transmission survenant lors de la transmission sur un canal radio-mobile. Ce schéma est ensuite appliqué dans le cadre d'une diffusion des contenus vidéo à travers un canal hétérogène, où les pertes de paquets sont compensées à l'aide du codage par descriptions multiples.

Nous nous sommes ensuite intéressés aux techniques de codage source-canal conjoint basées sur une transformation redondante. Un schéma d'estimation cohérente de la source, exploitant de manière efficace la redondance introduite, est proposé. Des outils de programmation linéaire et de calculs par intervalles sont mis en oeuvre pour détecter les erreurs de trasmission et estimer la source. Un autre schéma d'estimation, basé sur l'algorithme de propagation de croyances est également introduit. Ces deux schémas sont alors appliqués pour estimer l'entrée de bancs de filtres suréchantillonnés, et leurs performances sont comparées.

Décodage source-canal conjoint des sources vidéo

Dans cette partie, nous présentons le schéma de décodage source-canal conjoint proposé. Ce schéma est présenté dans la Figure 1. Considérons une séquence vidéo formée de N_t trames, regroupées en un vecteur $\mathbf{x}$. Ce vecteur est encodé et un vecteur

$$\mathbf{z} = \mathbf{F}(\mathbf{x}) \tag{1}$$

est obtenu. La fonction $\mathbf{F}(\cdot)$ dénote l'opération de codage vidéo, includant les transformées temporelle et spatiale, ainsi que la quantification et le codage entropique. Le contenu de $\mathbf{z}$ est alors transmis à traver un canal radio-mobile bruité.

Modèle du canal

Le canal désigne la partie de la chaîne de communication située entre la sortie de la fonction $\mathbf{F}(\cdot)$ et l'entrée du décodeur vidéo. Ce canal inclut la modulation, la paquétisation et le canal physique, ainsi que les opérations de démodulation et de dé-paquétisation.

Suppons que le vecteur $\mathbf{z}$ est organisé en N_p paquets de même longueur

$$\mathbf{z} = \left(\left(\mathbf{z}_1\right)^T, \ldots, \left(\mathbf{z}_{N_\mathrm{p}}\right)^T \right)^T. \tag{2}$$

Les mécanismes classiques de détection d'erreurs, mis en place au niveau des couches protocolaires basses, ne permettent pas aux paquets corrompus d'atteindre la couche applicative. L'implémentation de schémas de décodage source-canal conjoint au niveau de la couche applicative, nécessite alors la présence de mécanismes dits trans-couches (*cross-layer*) pour permettre aux différentes couches protocolaires de communniquer entre elles et d'assurer ainsi la remontée d'informations, telles que celles des paquets contenant des erreurs, et les informations souples du canal, des couches basses vers les couches hautes de la pile protocolaire [117, 42].

Tout au long de ce travail, nous supposons qu'un tel schéma trans-couches est implémenté, ainsi que des mécanismes de décodage robuste des entêtes des paquets reçus, pour permettre l'extraction des informations provenant des couches protocolaires basses [100, 101].

A la sortie de ce canal, deux types d'information sont obtenus. D'une part les informations souples sur les paquets transmis $\mathbf{z}_k$, modélisées par le vecteur

$$\mathbf{r} = \left(\left(\mathbf{r}_1\right)^T, \ldots, \left(\mathbf{r}_{N_\mathrm{p}}\right)^T \right)^T . \tag{3}$$

Les information souples contenues dans le vecteur $\mathbf{r}_k$ consistent par exemple, en des probabilités à posteriori ou des vraisemblances sur les données transmises dans le paquet $\mathbf{z}_k$. Ces informations sont fournies par le décodeur canal, au niveau de la couche physique, du côté du récepteur.

D'autre part, les informations sur l'état du canal sont également disponibles et sont modélisées par le vecteur

$$\boldsymbol{\rho} = \left(\rho_1, \ldots, \rho_{N_\mathrm{p}}\right)^T , \tag{4}$$

dont les composantes indiquent si le paquet $\mathbf{z}_k$, a été perdu ($\rho_k = 0$) ou reçu ($\rho_k \geqslant 1$). Le fait que $\rho_k = 0$ peut être déduit par exemple du contenu des entêtes de paquets. Les valeurs de $\rho_k \geqslant 1$ sont supposées être connues au récepteur. Elle indiquent par exemple, le niveau du bruit introduit dans le paquet $\mathbf{z}_k$.

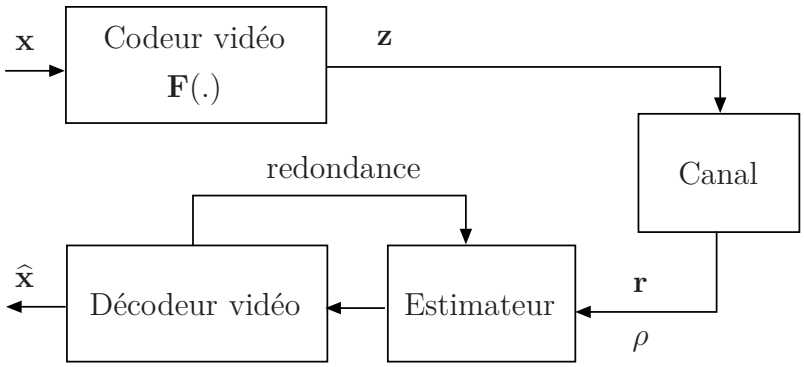


Figure 1.: Schéma de décodage source-canal conjoint.

Schémas d'estimation optimale

Cette section présente deux schémas d'estimation optimale, qui sont indépendants du codeur vidéo considéré. Nous commençons par formuler l'estimée optimale de $\mathbf{x}$, qui est difficile à mettre en oeuvre en pratique. Ensuite, nous décrivons une estimée *contrainte* de $\mathbf{z}$ qui est plus pratique et pour laquelle les informations souples en sortie du canal ainsi que la redondance de la source, sont plus faciles à exploiter.

Estimation optimale de x

Le critère d'estimation que nous considérons est celui du maximum à posteriori (MAP). L'estimée $\widehat{\mathbf{x}}_{\mathrm{MAP}}$ de $\mathbf{x}$ connaissant $\boldsymbol{\rho}$ et $\mathbf{r}$ est

$$\widehat{\mathbf{x}}_{\mathrm{MAP}} = \arg\max_{\mathbf{x}} p\left(\mathbf{x}|\boldsymbol{\rho}, \mathbf{r}\right). \tag{5}$$

Cette estimée peut être écrite de la manière suivante

$$\widehat{\mathbf{x}}_{\mathrm{MAP}} = \arg\max_{\mathbf{x}} p\left(\boldsymbol{\rho}, \mathbf{r}|\mathbf{F}\left(\mathbf{x}\right)\right) p\left(\mathbf{x}\right), \tag{6}$$

en faisant la somme sur tous les $\mathbf{z}$ possibles, puis en utilisant la règle de Bayes. Seul $\mathbf{z} = \mathbf{F}\left(\mathbf{x}\right)$ est alors gardé dans cette somme. L'évaluation de $\widehat{\mathbf{x}}_{\mathrm{MAP}}$ est difficile à mettre en place en pratique. En effet, elle requiert la maximisation d'une fonction discontinue, à cause de la quantification présente dans $\mathbf{F}(\cdot)$, sur un nombre très grand de variables qui sont tous les pixels des N_{t} trames de $\mathbf{x}$.

Estimation contrainte de z

De manière alternative, nous proposons d'estimer d'abord le vecteur $\mathbf{z}$, en utilisant le fait que ce train binaire a été généré à partir d'une séquence vidéo $\mathbf{x}$. Nous obtenons alors une estimation contrainte de $\mathbf{z}$ connaissant $\boldsymbol{\rho}$ et $\mathbf{r}$

$$\widehat{\mathbf{z}}_{\mathrm{MAP}} = \arg\max_{\mathbf{z}\in\mathcal{S}} p\left(\mathbf{z}|\boldsymbol{\rho}, \mathbf{r}\right), \tag{7}$$

où

$$\mathcal{S} = \{\mathbf{z} \mid \exists \mathbf{x},\ \mathbf{z} = \mathbf{F}(\mathbf{x})\} \tag{8}$$

est l'ensemble de tous les trains binaires pouvant être générés à partir des vecteurs $\mathbf{x}$.

En effectuant l'estimation de $\mathbf{z}$ parmi les éléments de l'ensemble $\mathcal{S}$, nous prenons implicitement en compte la redondance laissée par le codeur vidéo. En effet, puisque la séquence $\widehat{\mathbf{z}}_{\mathrm{MAP}}$ aurait pu être générée par le codeur vidéo, elle est nécessairement cohérente avec les contraintes imposées par la redondance résiduelle.

En utilisant la règle de Bayes et en supposant que les conditions du canal de transmission sont indépendantes pour chaque paquet $\mathbf{z}_k$, nous obtenons

$$\widehat{\mathbf{z}}_{\mathrm{MAP}} = \arg\max_{\mathbf{z}\in\mathcal{S}} p(\mathbf{z}) \prod_{k=1}^{N_{\mathrm{p}}} p(\mathbf{r}_k|\rho_k, \mathbf{z}_k)\, p(\rho_k|\mathbf{z}_k)\,. \tag{9}$$

Dans cette thèse, nous avons considéré le codeur vidéo par ondelettes Vidwav [126]. Dans la section suivante, nous présentons brièvement ce codeur et la structure du train binaire qu'il génère. Cela nous aidera à identifier une partie de la redondance résiduelle et nous permettra de construire l'ensemble $\mathcal{S}$. Nous soulignons ici le fait, que le schéma de décodage proposé peut être étendu à d'autres codeurs vidéo par ondelettes, dès que la rédondance résiduelle de ces codeurs est identifiée.

Présentation de Vidwav

Vidwav est un codeur scalable par ondelettes 3D utilisant un filtrage temporel compensé en mouvement (FTCM) [126]. Dans ce travail, le schéma $t+2$D, illustré par la Figure 2 est considéré. Le FTCM est d'abord opéré sur les trames de la séquence vidéo $\mathbf{x}$, placée en entrée. Ensuite, la transformation spatiale par ondelettes est effectuée sur les sous-bandes temporelles générées. Les sous-bandes spatio-temporelles obtenues sont alors divisées en N_{b} blocs 3D $\mathbf{b}_k$, $k = 1, \ldots, N_{\mathrm{b}}$ qui sont ensuite codés, indépendamment les uns des autres, à l'aide du codeur entropique 3D-ESCOT [71].

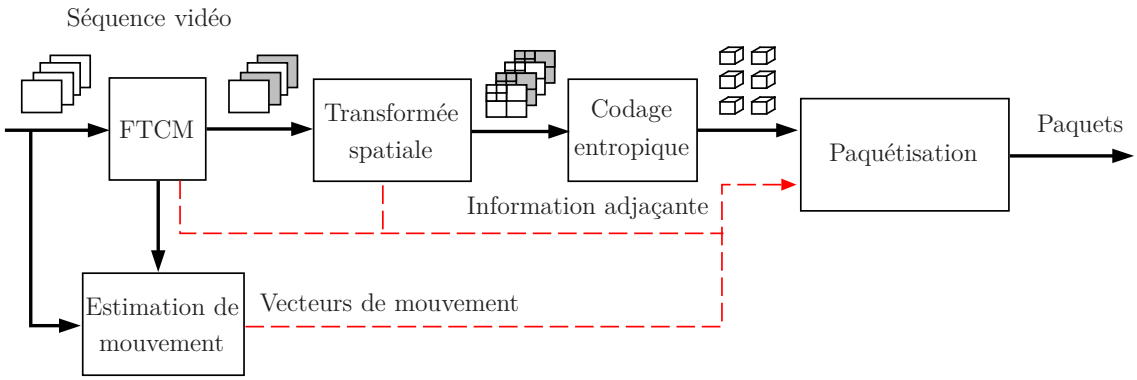


Figure 2.: Schéma en blocs de Viwav.

Codage entropique

Pour chaque bloc 3D $\mathbf{b}_k$, l'algorithme 3D-ESCOT opère un codage arithmétique par plan de bits, avec adaptation de contexte. Un train binaire *emboîté* $\mathbf{p}_k$ est alors généré. Il est formé par différents *segments*, résultant chacun d'une passe de codage par plan de bits [71]. De plus, des informations relatives à $\mathbf{p}_k$ sont obtenues, telles que le nombre de passes de codage n_k considérées dans $\mathbf{b}_k$, la longueur en octects

de $\mathbf{p}_k$, etc. Ces informations sont enregistrées dans l'entête $\mathbf{h}_k$, associée au train binaire $\mathbf{p}_k$. Cette entête contient également d'autres informations, nécessaires pour le décodage du bloc $\mathbf{b}_k$, telles que l'indexe de la sous-bande spatio-temporelle ou encore les vecteurs de mouvement associés à $\mathbf{b}_k$.

Le vecteur $\mathbf{z}_k$ obtenu à la sortie du codeur entropique est alors tel que

$$\mathbf{z}_k = \begin{pmatrix} \mathbf{h}_k \\ \mathbf{p}_k \end{pmatrix}. \tag{10}$$

Le codeur Vidwav produit généralement un train binaire $\mathbf{z}$ emboîté et scalable formé par plusieurs couches (*layers*). Dans la section suivante, nous allons décrire le schéma de décodage conjoint proposé dans le cas où une seule couche est générée. Ce schéma sera ensuite généralisé au cas de plusieurs couches.

Schéma de décodage source-canal conjoint

Dans cette partie, nous allons décrire le schéma de décodage conjoint pour un train binaire formé par une seule couche. Dans un premier temps, nous allons supposer que tous les paquets sont reçus ($\rho_k \geqslant 1$, $k = 1, \ldots, N_\mathrm{p}$), et que certains de ces paquets sont corrompus par des erreurs de transmission. Le but est alors de détecter ces erreurs et de les corriger en évaluant $\widehat{\mathbf{z}}_\mathrm{MAP}$ définie par (9).

Nous faisons de plus l'hypothèse, que chaque paquet de $\mathbf{z}$ défini dans (2), contient le code entropique généré pour un seul bloc 3D $\mathbf{b}_k$ donné. Ainsi $N_\mathrm{p} = N_\mathrm{b}$ et

$$\mathbf{z} = \left((\mathbf{z}_1)^T, \ldots, (\mathbf{z}_k)^T, \ldots, (\mathbf{z}_{N_\mathrm{b}})^T \right)^T. \tag{11}$$

Cette hypothèse a pour but d'alléger les notations et l'expression (11) est aussi générale que (2).

De manière similaire, nous pouvons écrire les informations souples du canal définies par (3) et les informations sur l'état du canal définies par (4) comme suit

$$\mathbf{r} = \left((\mathbf{r}_1)^T, \ldots, (\mathbf{r}_{N_\mathrm{b}})^T \right)^T, \tag{12}$$

$$\boldsymbol{\rho} = (\rho_1, \ldots, \rho_{N_\mathrm{b}})^T. \tag{13}$$

Supposons que tous les vecteurs $\mathbf{z} \in \mathcal{S}$ ont la même probabilité a priori, l'équation (9) devient alors

$$\widehat{\mathbf{z}}_\mathrm{MAP} = \arg\max_{\mathbf{z}\in\mathcal{S}} \prod_{k=1}^{N_\mathrm{b}} p\left(\mathbf{r}_k | \rho_k, \mathbf{z}_k\right) p\left(\rho_k | \mathbf{z}_k\right). \tag{14}$$

Cette hypothèse est valide tant qu'aucun a priori sur $\mathbf{x}$ n'est disponible au décodeur.

Considérons les ensembles

$$\mathcal{S}_k = \left\{ \mathbf{z}_k \mid \exists \mathbf{x},\, \exists \left(\widetilde{\mathbf{z}}_1, \ldots, \widetilde{\mathbf{z}}_{k-1}, \widetilde{\mathbf{z}}_{k+1}, \ldots, \widetilde{\mathbf{z}}_{N_\mathrm{b}}\right), \right.$$
$$\left. \text{tq} \quad \left(\left(\widetilde{\mathbf{z}}_1\right)^T, \ldots, \left(\widetilde{\mathbf{z}}_{k-1}\right)^T, \left(\mathbf{z}_k\right)^T, \left(\widetilde{\mathbf{z}}_{k+1}\right)^T, \ldots, \left(\widetilde{\mathbf{z}}_{N_\mathrm{b}}\right)^T\right)^T = \mathbf{F}\left(\mathbf{x}\right) \right\}, \quad (15)$$

définis pour chaque $k = 1, \ldots, N_\mathrm{b}$. L'ensemble $\mathcal{S}_k$ contient tous les trains binaires $\mathbf{z}_k$ pouvant résulter du codage entropique d'un bloc $\mathbf{b}_k$ qui peut être obtenu à partir d'une certaine séquence vidéo $\mathbf{x}$ à N_t trames. Nous avons alors

$$\mathcal{S} \subset \mathcal{S}_1 \times \cdots \times \mathcal{S}_{N_\mathrm{b}}. \quad (16)$$

En utilisant (14) et (16), nous dérivons une estimée sous-optimale pour chaque train binaire $\mathbf{z}_k$

$$\widehat{\mathbf{z}}_k = \arg \max_{\mathbf{z}_k \in \mathcal{S}_k} p\left(\mathbf{r}_k | \rho_k, \mathbf{z}_k\right) p\left(\rho_k | \mathbf{z}_k\right). \quad (17)$$

L'estimée $\left(\left(\widehat{\mathbf{z}}_1\right)^T, \ldots, \left(\widehat{\mathbf{z}}_{N_\mathrm{b}}\right)^T\right)^T$ est alors une solution sous-optimale de (14) puisque

$$\left(\left(\widehat{\mathbf{z}}_k\right)^T, \ldots, \left(\widehat{\mathbf{z}}_{N_\mathrm{b}}\right)^T\right)^T$$

n'est pas forcément dans $\mathcal{S}$, d'après (16).

Tests de syntaxe

Dans cette partie, nous allons décrire deux tests de syntaxe permettant de tenir compte de la redondance résiduelle laissée par Vidwav.

Comme nous l'avons mentionné précédemment, chaque $\mathbf{z}_k$ est formé d'une entête $\mathbf{h}_k$ et d'un train binaire $\mathbf{p}_k$. Après la transmission de $\mathbf{z}_k$, le vecteur reçu

$$\mathbf{r}_k = \left(\left(\mathbf{r}_{k,\mathrm{h}}\right)^T, \left(\mathbf{r}_{k,\mathrm{p}}\right)^T\right)^T$$

contient les informations souples du canal sur $\mathbf{h}_k$ données par le vecteur $\mathbf{r}_{k,\mathrm{h}}$ et sur $\mathbf{p}_k$ données par le vecteur $\mathbf{r}_{k,\mathrm{p}}$. En supposant que les entêtes $\mathbf{h}_k$ ont été protégées par des codes canal puissants et donc qu'elle sont reçues sans erreurs, le problème donné par (17) se limite à l'estimation de $\mathbf{p}_k$ sous la contrainte que $(\mathbf{h}_k, \mathbf{p}) \in \mathcal{S}_k$

$$\widehat{\mathbf{p}}_k = \arg \max_{\mathbf{p} \text{ tq } (\mathbf{h}_k, \mathbf{p}) \in \mathcal{S}_k} p\left(\mathbf{r}_{k,\mathrm{p}} | \rho_k, \mathbf{p}\right). \quad (18)$$

La valeur de ρ_k contient des informations sur les caractéristiques du canal comme par exemple son rapport signal à bruit (SNR), et permet d'évaluer la vraisemblance $p(\mathbf{r}_{k,\mathrm{p}}|\rho_k, \mathbf{p})$. L'espace de recherche de $\widehat{\mathbf{p}}_k$ est l'ensemble de toutes les séquences $\mathbf{p}$ pouvant être générées à partir d'une séquence vidéo $\mathbf{x}$, et qui sont en même temps *cohérentes* avec les informations contenues dans $\mathbf{h}_k$.

Un simple test de syntaxe t^{b} qui peut être alors implémenté au décodeur se base sur le fait que tout train binaire $\mathbf{p}_k$ obtenu à partir d'un bloc 3D donné, résulte d'un nombre n_k connu de passes de codage et a une longueur en bits ℓ_k connue. Les valeurs de n_k et de ℓ_k peuvent être déduites de l'entête $\mathbf{h}_k$, supposée correctement reçue. Le décodeur entropique 3D-ESCOT doit alors consommer exactement ℓ_k bits en décodant n_k passes d'une séquence $\mathbf{p}$ donnée. Si plus ou moins de ℓ_k bits sont consommés, les codeur et décodeur arithmétiques sont désynchronisés, ce qui implique que $\mathbf{p}$ contient des erreurs de transmission.

Le test t^{b} est alors défini de la manière suivante

$$t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p}) = \begin{cases} 1 & \text{si } \Lambda(\mathbf{p}) = \Lambda(\mathbf{h}_k) \\ 0 & \text{sinon} \end{cases}, \tag{19}$$

où $\Lambda(\mathbf{p})$ dénote le nombre de bits consommés lors du décodage de $\mathbf{p}$ et $\Lambda(\mathbf{h}_k)$ indique le nombre de bits à décoder, indiqué par l'entête $\mathbf{h}_k$. Comme $\mathbf{h}_k$ ne contient pas d'erreurs, $\Lambda(\mathbf{h}_k) = \ell_k$.

Pour une séquence $\mathbf{p}$ donnée $t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p}) = 0$ implique alors que $\Lambda(\mathbf{p}) \neq \ell_k$ et donc que $\mathbf{p} \neq \mathbf{p}_k$. Cependant, $t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p}) = 1$ n'implique pas nécessairement que $(\mathbf{h}_k, \mathbf{p}) \in \mathcal{S}_k$ et donc que $\mathbf{p} = \mathbf{p}_k$. En effet, les bits d'une certaine séquence $\mathbf{p} \neq \mathbf{p}_k$ peuvent entraîner la consommation d'exactement ℓ_k bits par le décodeur 3D-ESCOT. En absence de désynchronisation entre codeur et décodeur arithmétiques, le fait que $\mathbf{p} \neq \mathbf{p}_k$ n'est pas détectable par t^{b}.

Dans le train binaire généré par Vidwav, $\Lambda(\mathbf{h}_k)$ est connue en octects [126]. Le test t^{B} qui peut être considéré est alors le suivant

$$t^{\mathrm{B}}(\mathbf{h}_k, \mathbf{p}) = \begin{cases} 1 & \text{si } 8\,\lceil \Lambda(\mathbf{p})/8 \rceil = \Lambda(\mathbf{h}_k) \\ 0 & \text{sinon} \end{cases}, \tag{20}$$

où $\lceil \cdot \rceil$ désigne l'opérateur d'arrondi par valeurs supérieures. La détection d'erreurs par t^{B} est possible lorsque la désynchronisation est suffisante, c'est à dire lorsque plus ou moins d'octects que prévu sont consommés par le décodeur arithmétique. Le test t^{B} est donc moins efficace en termes de détection d'erreurs que le test t^{b}. Nous pouvons écrire $\Lambda(\mathbf{h}_k) = \ell_k + \lambda_k$, où $\lambda_k = 0, \ldots, 7$ est le nombre de bits ajoutés à la fin de $\mathbf{p}_k$. Pour implémenter le test $t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p})$ au niveau du décodeur Vidwav, la valeur λ_k peut être rajoutée à l'entête $\mathbf{h}_k$ lors de l'encodage. Cette information *adjaçante* nécessite seulement trois bits additonnels par train binaire $\mathbf{z}_k$.

Pour une entête $\mathbf{h}_k$ donnée, introduisons les ensembles

$$\mathcal{C}(\mathbf{h}_k) = \{\mathbf{p} \mid (\mathbf{h}_k, \mathbf{p}) \in \mathcal{S}_k\}, \tag{21}$$

$$\mathcal{C}^{\mathrm{b}}(\mathbf{h}_k) = \left\{\mathbf{p} \mid t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p}) = 1\right\} \tag{22}$$

et

$$\mathcal{C}^{\mathrm{B}}(\mathbf{h}_k) = \left\{\mathbf{p} \mid t^{\mathrm{B}}(\mathbf{h}_k, \mathbf{p}) = 1\right\}. \tag{23}$$

Nous obtenons alors

$$\mathcal{C}(\mathbf{h}_k) \subset \mathcal{C}^{\mathrm{b}}(\mathbf{h}_k) \subset \mathcal{C}^{\mathrm{B}}(\mathbf{h}_k). \tag{24}$$

En utilisant (19) ou (20), des solutions sous-optimales de (18) sont obtenues

$$\widehat{\mathbf{p}}_k^{\mathrm{b}} = \arg \max_{\mathbf{p} \in \mathcal{C}^{\mathrm{b}}(\mathbf{h}_k)} p\left(\mathbf{r}_{k,\mathrm{p}} | \rho_k, \mathbf{p}\right) \tag{25}$$

et respectivement

$$\widehat{\mathbf{p}}_k^{\mathrm{B}} = \arg \max_{\mathbf{p} \in \mathcal{C}^{\mathrm{B}}(\mathbf{h}_k)} p\left(\mathbf{r}_{k,\mathrm{p}} | \rho_k, \mathbf{p}\right). \tag{26}$$

Ces estimées nécessitent de considérer tous les éléments de $\mathcal{C}^{\mathrm{b}}(\mathbf{h}_k)$ ou de $\mathcal{C}^{\mathrm{B}}(\mathbf{h}_k)$ lorsque t^{b} ou t^{B} sont utilisés.

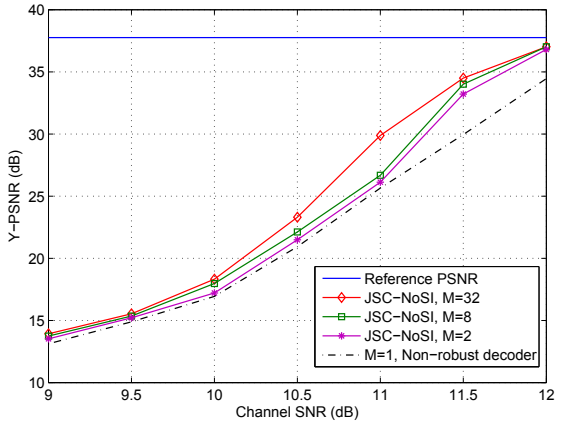
Nous utilisons un algorithme de décodage séquentiel, le $M-$algorithme [3] pour effectuer la recherche dans $\mathcal{C}^{\mathrm{b}}(\mathbf{h}_k)$ ou de $\mathcal{C}^{\mathrm{B}}(\mathbf{h}_k)$. Cet algorithme explore de manière partielle l'arbre binaire représentant toutes les séquences de longueur $\Lambda(\mathbf{h}_k)$, en commençant par la racine qui est de norme nulle. A chaque itération de l'algorithme, les chemins considérés sont étendus aux noeuds suivants et seuls les M chemins maximisant la métrique

$$\mathcal{M}(\mathbf{p}, \mathbf{r}_k) = -\log p(\mathbf{r}_k | \mathbf{p}, \rho_k) \tag{27}$$

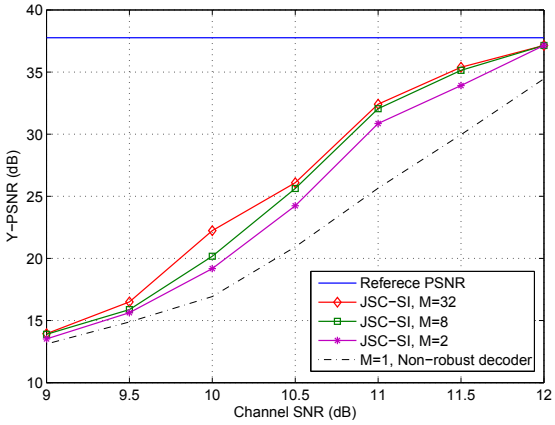
sont gardés à l'itération suivante. Les autres chemins sont éliminés. L'algorithme prend fin lorsque la profondeur $\Lambda(\mathbf{h}_k)$ de l'arbre binaire est atteinte. Le test $t^{\mathrm{B}}(\mathbf{h}_k, \mathbf{p})$ ou $t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p})$ est alors utilisé pour éliminer les séquences n'appartenant pas à $\mathcal{C}^{\mathrm{B}}(\mathbf{h}_k)$ ou $\mathcal{C}^{\mathrm{b}}(\mathbf{h}_k)$, en commençant par la séquence maximisant (27).

Cet algorithme est sous-optimal [3]. Si M n'est pas assez grand, la séquence $\mathbf{p}_k$ transmise peut être éliminée. A la fin de l'algorithme, si aucune des M séquences ne statisfait les tests de cohérence, nous choisissons d'utiliser la première séquence obtenue, correspondant à $M = 1$. En effet si des erreurs se sont produites dans les derniers bits de $\mathbf{p}_k$, cette stratégie conduirait à des artifacts plus modérés.

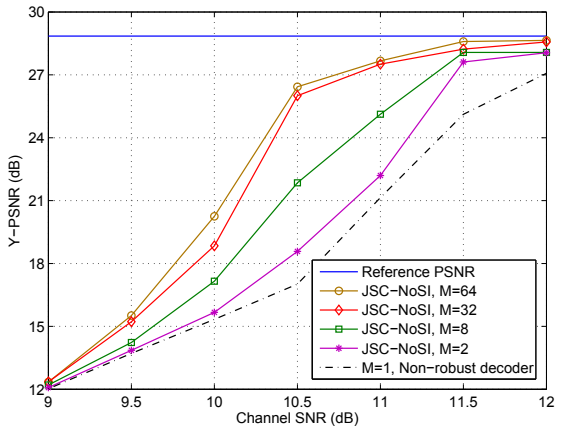
Résultats expérimentaux



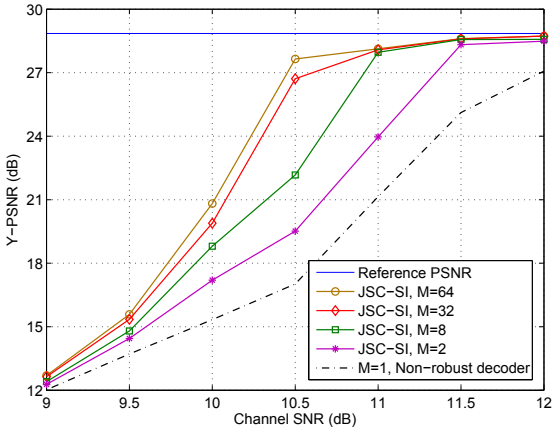
(a) `foreman.qcif`, JSC-NoSI



(b) `foreman.qcif`, JSC-SI



(c) `mobile.cif`, JSC-NoSI

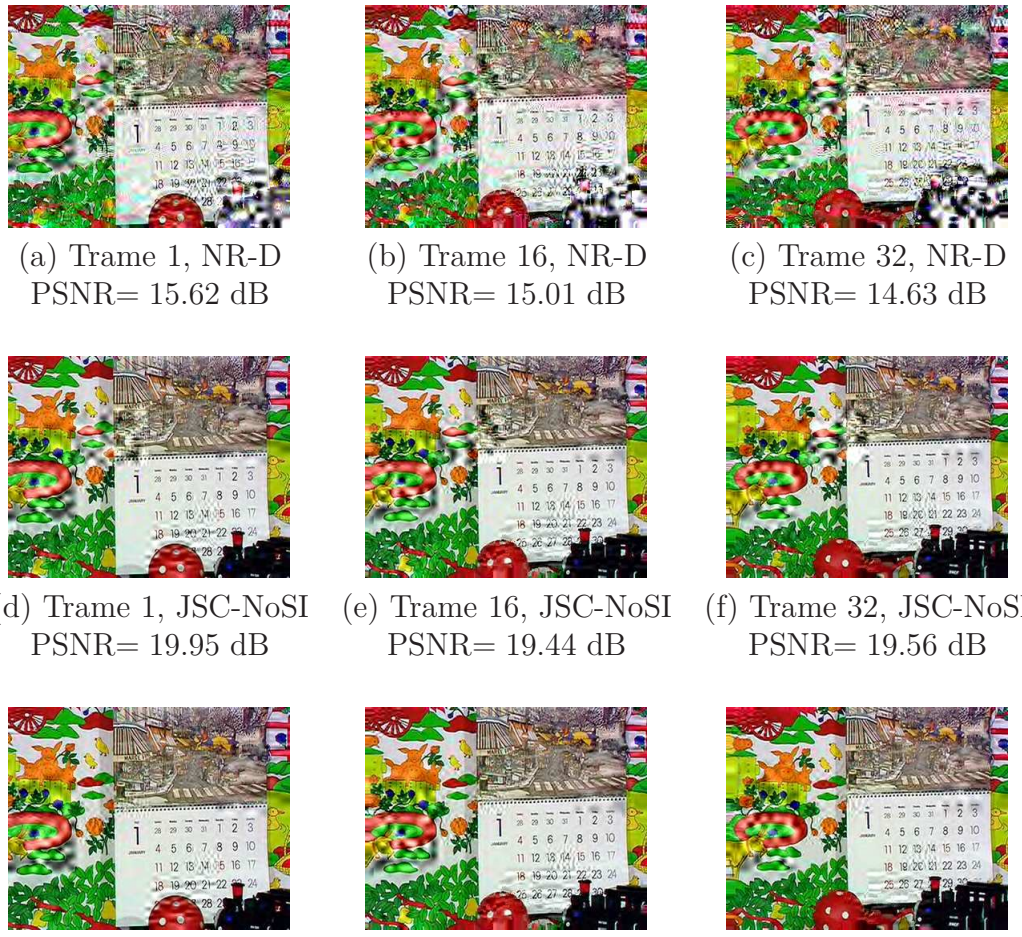


(d) `mobile.cif`, JSC-SI

Figure 3.: Y-PSNR (dB) en fonction du SNR du canal (dB) pour le décodeur non-robuste, le schéma de décodage conjoint sans information additionelle (JSC-NoSI) et le schéma de décodage conjoint avec information additionelle (JSC-SI).

Dans nos simulations, nous avons considéré les deux séquences `foreman.qcif` et `mobile.cif`, codées respectivement à 128 kbps et 768 kbps, avec un débit par trame égal à 15 fps et 30 fps. Le canal de transmission considéré est modélisé par un canal Gaussien AWGN avec un SNR variant entre 9 dB et 12 dB. Les trains binaires générés sont transmis à travers ce canal et seules les séquences $\mathbf{p}_k$, $k = 1, \ldots, N_\mathrm{b}$ sont supposées être corrompues par du bruit. La Figure 3 montre le PSNR moyen de la composante Y, des séquences vidéo reconstruites avec un décodeur classique basé sur les décisions dures du canal et les schémas de décodage conjoint basés sur les tests de cohérence t^B (JSC-NoSI) et t^b (JSC-SI), en fonction du SNR du canal.

Plusieurs valeurs du paramètre M de l'algorithme séquentiel ont été considérées. L'information additionnelle introduite au niveau des entêtes $\mathbf{h}_k$ permet d'augmenter les performances du décodage conjoint. Le gain en PSNR par rapport à un schéma de décodage non-robuste, passe de 4 dB à 6 dB, en considérant le schéma JSC-SI au lieu du schéma JSC-NoSI, pour `foreman.qcif` et pour un SNR canal égal à 11 dB. De même, le gain en termes de SNR canal est plus élevé en considérant le schéma JSC-SI. Par exemple, pour la séquence `mobile.cif` et pour un PSNR égal à 27 dB, le gain en termes de SNR canal passe de 1.3 dB avec le schéma JSC-NoSI à plus de 1.5 dB. La Figure 4 illustre les performances qualitatives données par les schémas de décodage conjoint JSC-NoSI et JSC-SI, par rapport au décodeur classique pour `mobile.cif`.



(a) Trame 1, NR-D PSNR= 15.62 dB
(b) Trame 16, NR-D PSNR= 15.01 dB
(c) Trame 32, NR-D PSNR= 14.63 dB
(d) Trame 1, JSC-NoSI PSNR= 19.95 dB
(e) Trame 16, JSC-NoSI PSNR= 19.44 dB
(f) Trame 32, JSC-NoSI PSNR= 19.56 dB
(g) Trame 1, JSC-SI PSNR= 21.19 dB
(h) Trame 16, JSC-SI PSNR= 20.66 dB
(i) Trame 32, JSC-SI PSNR= 21.44 dB

Figure 4.: Trames reconstruites de la séquence `mobile.cif`, transmise à travers un canal AWGN avec un SNR= 10 dB puis décodée avec un décodeur non-robuste (NR-D), le schéma JSC-NoSI et le schéma JSC-SI avec $M = 64$.

Le schéma de décodage conjoint JSC-SI est ensuite appliqué à un train binaire formé

par plusieurs couches. Dans ce cas, chaque séquence $\mathbf{p}_k$, $k = 1, \ldots, N_\mathrm{b}$ est formée par plusieurs *segments* comme suit

$$\mathbf{p}_k = \left(\left(\mathbf{p}_k^{(1)}\right)^T, \ldots, \left(\mathbf{p}_k^{(N_\mathrm{L})}\right)^T \right)^T, \tag{28}$$

où N_L est le nombre de couches et $\mathbf{p}_k^{(m)}$ est le segment de $\mathbf{p}_k$ inclu dans la couche $m = 1, \ldots, N_\mathrm{L}$. Une entête $\mathbf{h}_k^{(m)}$ est alors associée à ce segment. Cette entête contient le nombre de passes de codage $n_k^{(m)}$ dont résulte $\mathbf{p}_k^{(m)}$, ainsi que sa longueur $\ell_k^{(m)}$ en octects. Pour implémenter le test t^b, nous ajoutons 3 bits supplémentaires à $\mathbf{h}_k^{(m)}$ pour connaitre $\ell_k^{(m)}$ au bit près. Le test t^b est alors appliqué pour estimer chaque segment $\mathbf{p}_k^{(m)}$, en commençant par la première couche. Pour chaque couche $m = 1, \ldots, N_\mathrm{L}$, nous vérifions si $\sum_{i=1}^{m} \ell_k^{(i)}$ bits ont été consommés par le décodeur arithmétique. Si pour une couche donnée $m = 2, \ldots, N_\mathrm{L}$, le $M-$algorithme, ne délivre aucune séquence candidate validant le test t^b, les séquences $p_k^{(i)}$, $i \geqslant m$ courante et suivantes sont considérées comme perdues et la reconstruction du bloc $\mathbf{b}_k$ s'effectue en décodant jusqu'à la couche $m-1$.

La Figure 5 illustre le PSNR de la séquence `foreman.qcif`, codée en 3 couches, avec les débits suivants : 32 kbps (1 couche), 64 kbps (2 couches) et 128 kbps (3 couches), transmise sur un canal AWGN puis décodée en utilisant le décodeur non-robuste et le schéma de décodage conjoint JSC-SI.

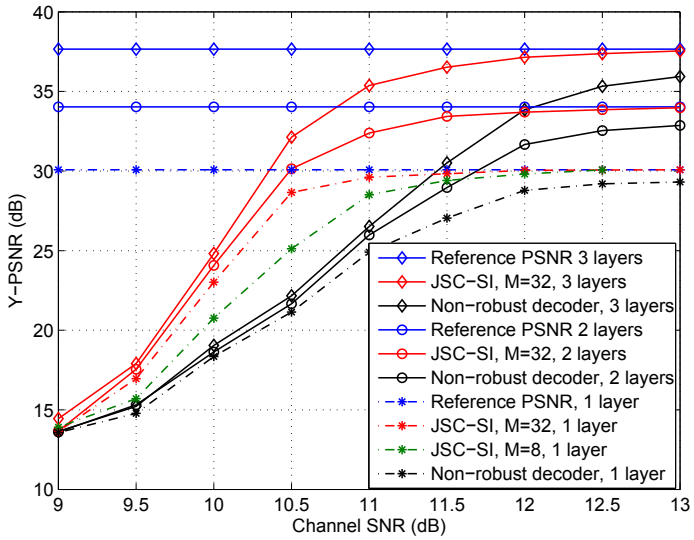


Figure 5.: Y-PSNR (dB) de la séquence `foreman.qcif` en fonction du SNR canal (dB) lorsqu'une, deux et trois couches sont décodées.

Nous effectuons le décodage d'une, de deux puis de trois couches. Nous observons que le gain en termes de PSNR devient moins élevé en décodant la troisième couche et ce à des SNRs faibles. Cela est dû au fait que certaines séquences des première

et deuxième couches sont corrompues et ne sont pas détectées avec le test t^{b}, ou ne sont pas correctement estimées. Certaines séquences de la troisième couche, reçues sans erreurs, peuvent alors être détectées comme corrompues. Le schéma JSC-SI est alors moins efficace pour les couches hautes, dont l'impact est néanmoins plus faible sur le décodage que les chouches plus basses. Notons toutefois, qu'à un même débit de codage égal à 128 kbps, le schéma JSC-SI est plus efficace en considérant un train binaire à 3 couches au lieu d'un train binaire à une couche. Par exemple, pour $M = 32$ et pour un SNR canal égal à 10.5 dB, le gain en PSNR obtenu en considérant un train binaire à 3 couches est d'envrion 10 dB et de 5 dB avec un train binaire à une seule couche.

Application au codage par descriptions multiples

Nous appliquons à présent notre schéma de décodage conjoint, au codage par descriptions multiples. Nous considérons un schéma de transmission formé d'une partie filaire à pertes de paquets, et d'un lien radio-mobile introduisant des erreurs sur les paquets transmis. Ce schéma est illustré par la Figure 6.

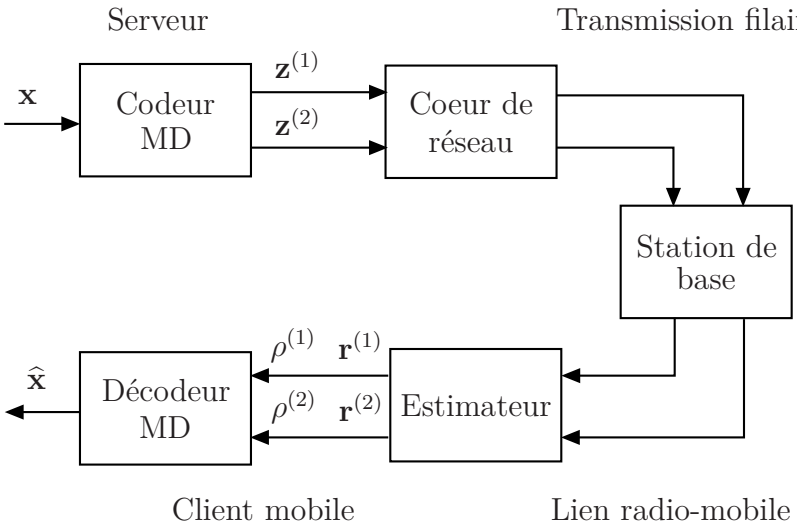


Figure 6.: Transmission vidéo hétérogène des descriptions multiples.

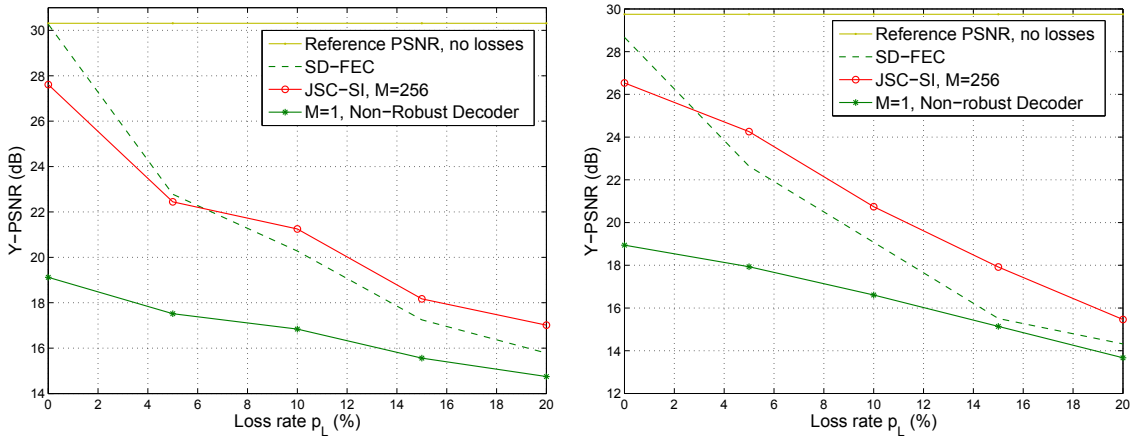
Le schéma de codage par descriptions multiples que nous avons considéré est celui proposé dans [148]. Ce schéma est basé sur une transformation temporelle redondante. La redondance entre les deux descriptions est introduite au dernier niveau temporel en utilisant un banc de filtres suréchantillonnés.

Les paquets perdus dans la sous-bande d'approximation spatio-temporelle de l'une des descriptions sont compensés à l'aide de ceux reçus dans la sous-bande d'approximation spatio-temporelle de l'autre description. Le décodage conjoint source-canal

permet de corriger les erreurs de transmission et d'augmenter ainsi le nombre de paquets corrects qui peuvent être exploités par le décodeur à descriptions multiples.

Nous avons comparé ce schéma de décodage robuste, à un schéma robuste classique dans lequel le train binaire généré est protégé à l'aide d'un code canal Reed-Solomon [132], introduisant le même niveau de redondance. Les résultats obtenus sont illustrés par la Figure 7. Les pertes de paquets sont modélisées par une loi uniforme avec un taux de pertes p_{L} variant entre 0 % et 20 %. En absence de pertes de paquets, le schéma basé sur le code correcteur d'erreurs est plus robuste que le schéma par descriptions multiples avec décodage conjoint. Cependant, pour $p_{\mathrm{L}} \geqslant 5$ % ($p_{\mathrm{L}} \geqslant 3$ %), le schéma par descriptions multiples devient plus efficace. Un gain en termes de PSNR de 1 dB (2 dB) est observé pour un niveau de redondance égal à 1.06 (1.04).

Ces résultats sont en accord avec ceux obtenus par d'autres travaux [144, 13, 143, 4]. Le schéma de codage par descriptions multiples est généralement plus efficace qu'un schéma de codage canal classique lorsque les niveaux de redondance sont assez faibles et que le taux de pertes est élevé.



(a) niveau de redondance égal à 1.06 (b)niveau de redondance égal à 1.04

Figure 7.: Y-PSNR (dB) de la séquence `foreman.qcif` en fonction du taux de pertes de paquets p_{L} pour un SNR canal de 10 dB. Le schéma par descriptions multiples avec décodage conjoint (JSC-SI) et le schéma robuste basé sur un code correcteur d'erreurs (SD-FEC) sont comparés.

Codage source-canal conjoint par transformation redondante

Dans cette partie, nous nous intéressons aux schémas de codage source-canal conjoint, basés sur une transformation redondante, comme par exemple les bancs de filtres

suréchantillonnés [151]. Le but est alors d'exploiter la redondance structurée introduite, pour estimer de manière robuste la source, à partir des sorties bruitées du canal de transmission.

Schéma de transmission

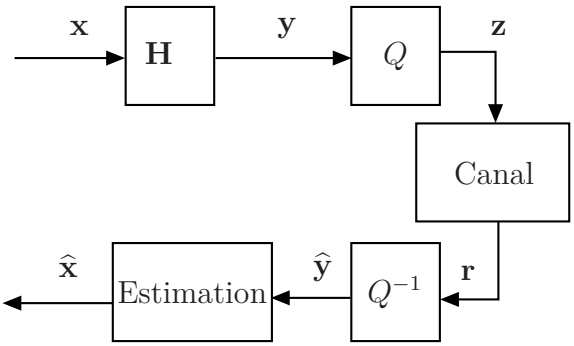


Figure 8.: Schéma de transmission considéré.

Le schéma de transmission considéré est illustré par la Figure 8. Le vecteur $\mathbf{x} \in \mathbb{R}^N$ à transmettre est supposé être une réalisation du vecteur aléatoire $\mathbf{X} = (X_1, \ldots, X_N)^T$. Ce vecteur est transformé par une matrice $\mathbf{H}$ de taille $M \times N$, $M > N$, supposée de rang plein N. Le vecteur $\mathbf{y} = \mathbf{Hx}$ obtenu à la sortie de $\mathbf{H}$ contient plus d'éléments que $\mathbf{x}$. De plus il est supposé être la réalisation du vecteur aléatoire $\mathbf{Y} = (Y_1, \ldots, Y_M)^T$.

Chaque composante y_m, $m = 1, \ldots, M$ de $\mathbf{y}$ est quantifiée à l'aide d'un quantificateur scalaire Q sur ρ bits. Les intervalles de quantification $[s(0)], \ldots, [s(2^\rho - 1)]$ forment alors une partition de $\mathbb{R}$. Après quantification, le vecteur $\mathbf{z}$ formé par les indices de quantification z_m est obtenu. Ce vecteur est une réalisation du vecteur aléatoire $\mathbf{Z} = (Z_1, \ldots, Z_M)^T$ avec

$$Z_m = Q(Y_m) = z \Leftrightarrow Y_m \in [s(z)] = [\underline{s}(z), \overline{s}(z)], \; z \in \{0, \ldots, 2^\rho - 1\}, \quad (29)$$

où pour un indice $z \in \{0, \ldots, 2^\rho - 1\}$ donné, $\underline{s}(z)$ et $\overline{s}(z)$ dénotent respectivement les bornes inférieure et supérieure de l'intervalle $[s(z)]$. Les indices z_m sont ensuite transmis sur un canal bruité.

Le modèle du canal est le même que celui considéré dans la section précédente. Nous supposons que des mécanismes trans-couches sont implémentés pour permettre la remontée des informations souples du canal des couches basses vers les couches hautes de la pile protocolaire. Le canal représente alors tout ce qui se trouve entre la sortie de la quantification Q et l'entrée de la quantification inverse Q^{-1}, incluant le codage entropique, la modulation, la paquétisation réseau, la transmission sur le canal physique, ainsi que la dé-paquétisation robuste, la démodulation et le décodage entropique [100, 42]. De plus, ce canal est supposé être sans mémoire.

A la sortie du canal, les informations souples $\mathbf{r} = \left((\mathbf{r}_1)^T, \ldots, (\mathbf{r}_M)^T\right)^T$ sur les bits transmis sont obtenues. Le vecteur $\mathbf{r}_m \in \mathbb{R}^\rho$ (or $\mathbb{C}^\rho$) est la sortie du canal associée à z_m. L'effet du canal de transmission est alors décrit par la probabilité de transition $p_{\mathbf{R}|Z}(\mathbf{r}|z)$.

Le problème que nous considérons est celui d'estimer l'entrée $\mathbf{x}$, à partir de $\mathbf{r}$

$$\widehat{\mathbf{x}}_{\text{MAP}} = \arg \max_{\mathbf{x} \in \mathbb{R}^N} p_{\mathbf{X}|\mathbf{R}}(\mathbf{x}|\mathbf{r}_{1:M}), \tag{30}$$

où $\mathbf{r}_{1:M} = \left((\mathbf{r}_1)^T, \ldots, (\mathbf{r}_M)^T\right)^T$.

En utilisant la règle de Bayes, le fait que les intervalles de quantifications forment une parition de $\mathbb{R}$ et le fait que le canal considéré est sans mémoire, nous pouvons montrer que (30) peut être écrite de la manière suivante

$$\widehat{\mathbf{x}}_{\text{MAP}} = \arg \max_{\mathbf{x} \in \mathbb{R}^N} p_{\mathbf{X}}(\mathbf{x}) \prod_{m=1}^{M} p_{\mathbf{R}|Z}\left(\mathbf{r}_m | Q\left(\mathbf{h}_m^T \mathbf{x}\right)\right). \tag{31}$$

Pour une observation $\mathbf{r}_{1:M}$ donnée, la fonction

$$f(\mathbf{x}, \mathbf{r}_{1:M}) = p_{\mathbf{X}}(\mathbf{x}) \prod_{m=1}^{M} p_{\mathbf{R}|Z}\left(\mathbf{r}_m | Q\left(\mathbf{h}_m^T \mathbf{x}\right)\right) \tag{32}$$

est continue par morceaux, dû à la quantification. La maximisation de $f(\mathbf{x}, \mathbf{r}_{1:M})$ sur toutes les valeurs possibles de $\mathbf{x} \in \mathbb{R}^N$ n'est pas facile, surtout pour des valeurs élevées de N.

Nous proposons dans ce qui suit deux méthodes d'estimation sous-optimales de (31), mais qui sont moins complexes.

Estimation cohérente

Le but de l'estimation cohérente présentée dans cette partie, est d'exploiter la redondance introduite par $\mathbf{H}$ et le fait que le bruit de quantification est *borné*. Une implémentation sous-optimale de l'estimée MAP définie par (31) est obtenue en effectuant l'estimation dans le sous-espace des vecteurs de quantification *cohérents*, c'est-à-dire ceux pouvant résulter de la quantification d'un vecteur $\mathbf{y}$ pouvant être généré à la sortie de la matrice $\mathbf{H}$.

De manière plus formelle, le principe est de considérer l'ensemble des vecteurs de quantifications $\mathbf{z}$ pour lesquels il existe un vecteur $\mathbf{x} \in \mathbb{R}^N$, satisfaisant $Q(\mathbf{Hx}) = \mathbf{z}$. Cet ensemble

$$\mathcal{Z} = \left\{\mathbf{z} \in \{0, \ldots, 2^\rho - 1\}^M \mid \exists\, \mathbf{x} \in \mathbb{R}^N,\ \mathbf{z} = Q(\mathbf{Hx})\right\} \tag{33}$$

est l'ensemble des vecteurs de quantification cohérents.

L'estimation cohérente que nous proposons est alors la suivante

1. Evaluer

$$\mathbf{z}^{\mathrm{f}}_{1:M} = \arg\max_{\mathbf{z}\in\mathcal{Z}} \prod_{m=1}^{M} p_{\mathbf{R}|Z}\left(\mathbf{r}_m|z_m\right). \tag{34}$$

2. Résoudre

$$\begin{aligned} \widehat{\mathbf{x}}_{\mathrm{MAP}} = \arg\max_{\mathbf{x}}\ & p_{\mathbf{X}}\left(\mathbf{x}\right). \\ \mathrm{tq}\, & \mathbf{H}\mathbf{x} \leqslant \overline{\mathbf{s}}\left(\mathbf{z}^{\mathrm{f}}_{1:M}\right) \\ & -\mathbf{H}\mathbf{x} \leqslant -\underline{\mathbf{s}}\left(\mathbf{z}^{\mathrm{f}}_{1:M}\right) \end{aligned} \tag{35}$$

Pour déterminer $\mathbf{z}^{\mathrm{f}}$ défini dans (34), toutes les valeurs possibles de

$$\mathbf{z}_{1:M} \in \{0,\ldots,2^{\rho}-1\}^{M}$$

peuvent être ordonnées par ordre décroissant de $p_{\mathbf{R}|\mathbf{Z}}\left(\mathbf{r}_{1:M}|\mathbf{z}_{1:M}\right)$, dans une liste $\mathcal{L}_{\mathbf{z}} = \left\{\mathbf{z}^{(1)}_{1:M},\ldots,\mathbf{z}^{(L)}_{1:M}\right\}$, où $L = 2^{\rho M}$. Un test de cohérence peut alors être effectué sur chaque vecteur candidat $\mathbf{z}^{(\ell)}_{1:M}$, en commençant par $\ell = 1$ et en augmentant ℓ jusqu'à ce que $\mathbf{z}^{(\ell)}_{1:M} \in \mathcal{Z}$.

Nous introduisons, dans ce qui suit, deux tests de cohérence différents. Le premier est basé sur de la programmation linéaire alors que le second est basé sur du calcul par intervalles.

Test de cohérence par programmation linéaire

Pour une valeur donnée de $\ell = 1,\ldots,L$, considérons le problème de maximisation suivant

$$\max_{\mathbf{x}\in\mathcal{X}\left(\mathbf{z}^{(\ell)}_{1:M}\right)} \mathbf{a}^{T}\mathbf{x},$$

où

$$\mathcal{X}\left(\mathbf{z}_{1:M}\right) = \left\{\mathbf{x}\in\mathbb{R}^{N} \mid \mathbf{z}_{1:M} = Q\left(\mathbf{H}\mathbf{x}\right)\right\}$$

et $\mathbf{a}$ est un vecteur non nul de taille N. Ce programme est équivalent au programme linéaire suivant

$$\begin{aligned} \max_{\mathbf{x}\in\mathbb{R}^{N}}\ & \mathbf{a}^{T}\mathbf{x}. \\ \mathrm{tq}\ & \mathbf{H}\mathbf{x} \leqslant \overline{\mathbf{s}}\left(\mathbf{z}^{(\ell)}_{1:M}\right) \\ & -\mathbf{H}\mathbf{x} \leqslant -\underline{\mathbf{s}}\left(\mathbf{z}^{(\ell)}_{1:M}\right) \end{aligned} \tag{36}$$

Si une solution $\mathbf{x}$ est trouvée pour (36), alors $\mathcal{X}\left(\mathbf{z}_{1:M}\right) \neq \emptyset$ et $\mathbf{z}^{(\ell)}$ est cohérent.

Ainsi, pour déterminer $\mathbf{z}^{\mathrm{f}}$, nous pouvons résoudre (36) pour différents $\mathbf{z}_{1:M}^{(\ell)}$, en commençant par $\ell = 1$ et en augmentant ℓ jusqu'à ce qu'une solution existe pour un certain ℓ^{f}. Le vecteur $\mathbf{z}^{\mathrm{f}}$ est alors égal à $\mathbf{z}^{(\ell^{\mathrm{f}})}$.

Notons que nous aurions pu résoudre directement (35) avec $\mathbf{z}_{1:M}^{(\ell)}$, en commençant par $\ell = 1$ et en augmentant ℓ jusqu'à ce qu'une solution existe. Il est cependant moins complexe de résoudre (36) pour trouver $\mathbf{z}^{\mathrm{f}}$. Le problème (35) est ensuite résolu avec $\mathbf{z}^{\mathrm{f}}$.

Test de cohérence par calcul par intervalles

Une autre approche pour déterminer si un vecteur de quantification $\mathbf{z}$ appartient à $\mathcal{Z}$, utilise la matrice à détection de parité $\mathbf{P}$, associée à $\mathbf{H}$. En effet, comme $\mathbf{H}$ est supposée de rang plein N, il existe un matrice $\mathbf{P}$ de taille $(M-N) \times M$ et de rang plein $(M-N)$ telle que

$$\mathbf{PHx} = \mathbf{0}, \ \forall \mathbf{x} \in \mathbb{R}^N. \tag{37}$$

Nous avons alors

$$\mathbf{Py} = \mathbf{0} \Longleftrightarrow \exists\ \mathbf{x} \in \mathbb{R}^N \text{ tq } \mathbf{y} = \mathbf{Hx}. \tag{38}$$

En utilisant la définition (33) de l'ensemble $\mathcal{Z}$ et l'équivalance (38)

$$\mathbf{Py} \neq \mathbf{0} \Longleftrightarrow \mathbf{z} = Q(\mathbf{y}) \notin \mathcal{Z} \tag{39}$$

Appliquons à présent l'équivalence (29) à toutes les composantess z_m, $m = 1, \ldots, M$ d'un vecteur $\mathbf{z}$ donné

$$\mathbf{z} = Q(\mathbf{y}) \Longleftrightarrow \mathbf{y} \in [\mathbf{s}(\mathbf{z})]\,, \tag{40}$$

où $[\mathbf{s}\left(\mathbf{z}_{1:M}\right)] = \left(\left[s\left(z_1^{(\ell)}\right)\right], \ldots, \left[s\left(z_M^{(\ell)}\right)\right]\right)^T$ est une *boîte*, c'est-à-dire un vecteur d'intervalles.

Le vecteur d'intervalles $\mathbf{P}\left[\mathbf{s}\left(\mathbf{z}_{1:M}\right)\right]$ pouvant être évalué avec de simples opérations d'additions et de multiplications sur des intervalles [73], est tel que

$$\left\{\mathbf{Py},\ \mathbf{y} \in \left[\mathbf{s}\left(\mathbf{z}_{1:M}\right)\right]\right\} \subset \mathbf{P}\left[\mathbf{s}\left(\mathbf{z}_{1:M}\right)\right]. \tag{41}$$

En utilisant (39) et (41), nous pouvons alors écrire

$$\mathbf{0} \notin \mathbf{P}\left[\mathbf{s}\left(\mathbf{z}\right)\right] \Rightarrow \mathbf{z} \notin \mathcal{Z}. \tag{42}$$

Le test (42) permet de prouver qu'un vecteur de quantification $\mathbf{z}$ n'est pas cohérent, mais il ne permet de prouver qu'un vecteur $\mathbf{z}$ est cohérent, puisque l'inclusion (41) est stricte.

Ce test est donc moins efficace que le test basé sur la programmation linéaire introduit par (36). Il a cependant une complexité entre $O(M)$ et $O(M(M-N))$, ce qui est généralement moins complexe que de résoudre directement (36). Ce test permet d'éliminer rapidement les vecteurs de quantification $\mathbf{z}^{(\ell)}$, $\ell = 1, \ldots, L$ qui ne sont pas cohérents. Les séquences *jugées* cohérentes par ce test peuvent alors être testées à l'aide du programme linéaire (36).

Nous introduisons dans ce qui suit un petit exemple pour mettre en oeuvre les tests de cohérence introduits précédemment, et montrer l'intérêt d'utiliser le test basé sur la matrice de parité avant celui qui est basé sur la programmation linéaire.

Exemple

Dans cet exemple, $N = 2$ et $M = 3$. Considérons un vecteur $\mathbf{x} \in \mathbb{R}^N$, tel que $p_X(x_n) \sim \mathcal{N}(0,1)$, $n = 1, \ldots, N$ et une matrice $\mathbf{H}$ de taille $M \times N$ définie ainsi

$$\mathbf{H} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix}$$

Une matrice à détection de parité associée à $\mathbf{H}$ est la matrice $\mathbf{P} = (1,1,1)$. Le vecteur placé en entrée de $\mathbf{H}$ est $\mathbf{x} = (-0.62, 1.42)^T$, et le vecteur obtenu à sa sortie est $\mathbf{y} = (-0.62, 1.54, -0.92)^T$. Une quantification Q sur 2 bits est considérée ici et les intervalles de quantification sont les suivants

$$\begin{aligned} [s(0)] &=]-\infty, -1.03] \\ [s(1)] &= [-1.03, 0.00] \\ [s(2)] &= [0.00, 1.03] \\ [s(3)] &= [1.03, +\infty[. \end{aligned}$$

Le vecteur de quantification obtenu est alors $\mathbf{z} = (1,3,1)^T$. Ce vecteur est modulé avec une modulation BPSK par exemple, puis transmis sur un canal AWGN avec un SNR égal à 2 dB. L'observation

$$\mathbf{r} = \begin{pmatrix} -0.65 & -0.97 \\ -1.01 & -1.44 \\ 0.07 & -0.92 \end{pmatrix}$$

est obtenue à la sortie de ce canal. Les combinaisons possibles de $\mathbf{z}_{1:3}$ sont ordonnées par ordre décroissant de $p(\mathbf{r}_{1:3}|\mathbf{z}_{1:3})$ et les séquences $\mathbf{z}_{1:3}^{(\ell)}$, $\ell = 1, \ldots, L$ sont obtenues

avec $L = 64$. Dans ce qui suit, nous allons nous limiter aux quatres premières combinaisons de $\mathbf{z}_{1:3}^{(\ell)}$, $\ell = 1, \ldots, 4$. Le Tableau 1 montre les valeurs de $p_{\mathbf{R}|\mathbf{Z}}\left(\mathbf{r}_{1:3}|\mathbf{z}_{1:3}^{(\ell)}\right)$, $\ell = 1, \ldots, 4$. Comme nous pouvons le voir, le vecteur le plus vraisemblable $\mathbf{z}_{1:3}^{(1)} = (3,3,1)^T$, qu'aurait donné un décodeur classique par décisions dures, n'est pas celui qui a été transmis. Ce dernier correspond à la valeur $\ell = 3$.

ℓ	1	2	3	4
$\mathbf{z}_{1:3}^{(\ell)}$	$(3,3,1)^T$	$(3,3,3)^T$	$(1,3,1)^T$	$(1,3,3)^T$
$p_{\mathbf{R}|\mathbf{Z}}\left(\mathbf{r}_{1:3}|\mathbf{z}_{1:3}^{(\ell)}\right)$	0.42	0.33	0.05	0.04

Table 1.: Valeur de $p_{\mathbf{R}|\mathbf{Z}}\left(\mathbf{r}_{1:3}|\mathbf{z}_{1:3}^{(\ell)}\right)$, $\ell = 1, \ldots, 4$.

Nous allons considérer dans un premier temps le test de cohérence basé sur la résolution du programme linéaire (36). Aucune solution n'est trouvée pour ce programme, lorsque $\ell \in \{1, 2, 4\}$. Pour $\ell = 3$, une solution est trouvée. Considérons maintenant le test de cohérence (42), basé sur la matrice de parité $\mathbf{P}$. Les résultats de ce test sont reportés dans le Tableau 2. Avec ce test, seul le vecteur candidat $\mathbf{z}^{(3)}$ est jugé cohérent.

$\mathbf{z}^{(\ell)}$	TP
$\mathbf{z}^{(1)} = (3,3,1)^T$	$\mathbf{0} \notin \mathbf{P}[\mathbf{s}(\mathbf{z}^{(1)})]$
$\mathbf{z}^{(2)} = (3,3,3)^T$	$\mathbf{0} \notin \mathbf{P}[\mathbf{s}(\mathbf{z}^{(2)})]$
$\mathbf{z}^{(3)} = (1,3,1)^T$	$\mathbf{0} \in \mathbf{P}[\mathbf{s}(\mathbf{z}^{(3)})]$
$\mathbf{z}^{(4)} = (1,3,3)^T$	$\mathbf{0} \notin \mathbf{P}[\mathbf{s}(\mathbf{z}^{(4)})]$

Table 2.: Résultat du test de parité (TP) sur les 4 séquences $\mathbf{z}^{(\ell)}$ les plus vraisemblables.

Le Tableau 3 montre les temps d'exécutions t_{QP} et t_{LP} qui sont pris pour résoudre le programme quadratique (35) et le programme linéaire (36) pour les différentes combinaisons $\mathbf{z}_{1:3}^{(\ell)}$, $\ell = 1, \ldots, 4$. Il montre également le temps d'exécution t_{TP}, pris pour effectuer le test de parité (42). Comme nous pouvons le voir, le test de parité prend moins de temps que le test basé sur la résolution de (36). Par exemple, pour prouver que la séquence $\mathbf{z}^{(1)}$ n'est pas cohérente, le test de parité prend 5 fois moins de temps que le test basé sur le programme linéaire.

ℓ	1	2	3	4
$\mathbf{z}_{1:3}^{(\ell)}$	$(3,3,1)^T$	$(3,3,3)^T$	$(1,3,1)^T$	$(1,3,3)^T$
t_{QP}(secondes)	0.0058	0.0066	0.0072	0.0164
t_{LP}(secondes)	0.0024	0.0030	0.0014	0.0022
t_{TP}(secondes)	0.00048	0.00048	0.00047	0.00047

Table 3.: Temps d'exécution t_{QP}, t_{LP} et t_{TP} pris respectivement par (35), (36) et (42), pour $\mathbf{z}^{(\ell)}$, $\ell = 1, \ldots, 4$.

En sélectionnant $\mathbf{z}^{\mathrm{f}} = \mathbf{z}^{(3)} = (1,3,1)^T$ pour résoudre (35), l'estimée $\widehat{\mathbf{x}}_{\mathrm{MAP}} = (-0.51, 0.89)^T$ est obtenue.

Estimation par propagation de croyances

Dans cette partie, une approche différente est considérée pour estimer $\widehat{\mathbf{x}}_{\mathrm{MAP}}$ definie par (30). Cette approche se base sur l'algorithme de propagation des croyances (*belief propagation*) [80, 95, 38] pour calculer les distributions à posteriori $p_{X|\mathbf{R}}(x_n|\mathbf{r}_{1:M})$, $n = 1, \ldots, N$ de chaque composante x_n du signal d'entrée $\mathbf{x} = (x_1, \ldots, x_n)^T$, connaissant l'observation $\mathbf{r}_{1:M}$.

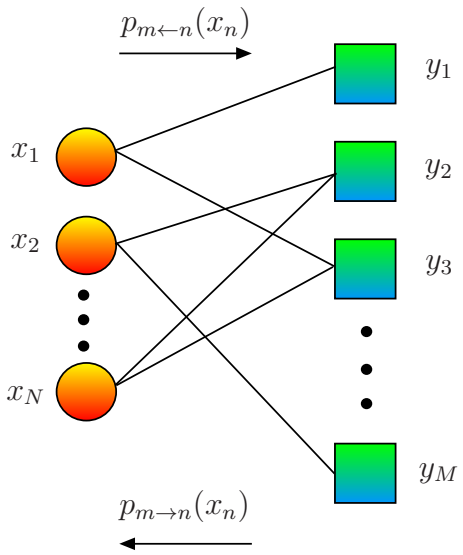


Figure 9.: Graphe représentant les relations entre les différentes variables considérées.

Nous utilisons le graphe $\mathcal{G}_{\mathbf{H}}$, associé à la matrice $\mathbf{H}$ et représenté dans la Figure 9. Ce graphe visualise les dépendances linaires entre les variables d'entrée x_n et les variables de sortie y_m. Il est formé par deux types de noeuds : les noeuds dits

variables $n = 1, \ldots, N$ et les noeuds dits *facteurs* $m = 0, \ldots, M$. Ces différents noeuds s'échangent des messages (ou *beliefs*) le long des arêtes

$$\mathcal{E}_{\mathbf{H}} \{(n, m) \in \{1, \ldots, N\} \times \{1, \ldots, M\} \mid \mathbf{H}(m, n) \neq 0\}$$

du graphe $\mathcal{G}_{\mathbf{H}}$.

Soit $\mathcal{N}(n)$ l'ensemble des noeuds facteurs connectés à un noeud variable n. De manière similaire, soit $\mathcal{N}(m)$ l'ensemble des noeuds variables connectés à un noeud facteur m.

Nous allons présenter dans ce qui suit, une implémentation de l'algorithme de propagation de croyances, proposée par Rangan dans [128], qui est bien adaptée au problème d'estimation considéré dans (30). Les messages échangés sont des *croyances* sur les distributions à posteriori de chaque entrée x_n. Ces messages sont mis à jour de manière itérative le long des arêtes $\mathcal{E}_{\mathbf{H}}$.

1. *Initialisation :*
   a) itération $i = 1$
   b) $\forall (n, m) \in \mathcal{E}_{\mathbf{H}}$, initialiser les messages envoyés par n à m

$$p^x_{m \leftarrow n}(i, x_n) = p^x_n(i, x_n) = p_X(x_n). \tag{43}$$

2. *Mélange linéaire :*
   a) supposer que les variables aléatoires X_n sont indépendentes et que $X_n \sim p^x_{m \leftarrow n}(i, x_n)$
   b) $\forall (n, m) \in \mathcal{E}_{\mathbf{H}}$, calculer la distribution $p^y_{m \to n}(i, y_{m \to n})$ de la variable aléatoire

$$Y_{m \to n} = \sum_{\substack{n' \in \mathcal{N}(m) \\ n' \neq n}} \mathbf{H}(m, n') X_{n'}. \tag{44}$$

3. *Mise à jour de la sortie :*
   a) $\forall (n, m) \in \mathcal{E}_{\mathbf{H}}$, calculer la vraisemblance

$$p^u_{m \to n}(i, u_m) = \int_{y_{m \to n}} p_{\mathbf{R}|Y}(\mathbf{r}_m | u_m + y_{m \to n}) p^y_{m \to n}(i, y_{m \to n}) dy_{m \to n}. \tag{45}$$

4. *Mise à jour de l'entrée :*
   a) $\forall (n, m) \in \mathcal{E}_{\mathbf{H}}$, mettre à jour le message envoyé par n à m

$$p^x_{m \leftarrow n}(i + 1, x_n) = \alpha p_X(x_n) \prod_{\substack{m' \in \mathcal{N}(n) \\ m' \neq m}} p^u_{m' \to n}(i, \mathbf{H}(m', n) x_n). \tag{46}$$

   où α est une contante de normalisation, obtenue en imposant que les messages $p^x_{m \leftarrow n}(i + 1, x_n)$ sont de somme égale à 1.

b) $\forall n = 1, \ldots, .N$, mettre à jour les distributions

$$p_n^x(i+1, x_n) = \beta p_X(x_n) \prod_{m \in N(n)} p_{m \to n}^u(i, \mathbf{H}(m,n) x_n). \tag{47}$$

où β est une constante de normalisation, obtenue en imposant que les messages $p_n^x(i+1, x_n)$ sont de somme égale à 1.

5. *Incrémentation :*

a) $i = i + 1$

b) retourner à l'étape 2, jusqu'à ce qu'un nombre suffisant d'itérations soit effectué.

Le message $p_{m \leftarrow n}^x(i, x_n)$ envoyé par le noeud variable n au noeud facteur m, à l'itération i, exprime la distribution à posteriori de x_n, comme *perçue* par les noeuds $y_{m'} \in \mathcal{N}(n) \backslash \{m\}$ à l'itération i. Le message $p_{m \to n}^y(i, y_{m \to n})$ envoyé par le noeud facteur m au noeud variable n à l'itération i, permet d'évaluer la fonction $p_{m \to n}^u(i, u_m)$ mesurant la vraisemblance que l'observation $\mathbf{r}_m$ soit obtenue lorsque $X_n = x_n$.

Après N_i itérations de cet algorithme, l'estimée MAP de $\mathbf{x}$ peut être évaluée de la manière suivante

$$\widehat{\mathbf{x}}_{\mathrm{MAP}} = \left(\widehat{x}_{1,\mathrm{MAP}}, \ldots, \widehat{x}_{N,\mathrm{MAP}}\right)^T, \tag{48}$$

où pour $n = 1, \ldots, N$

$$\widehat{x}_{n,\mathrm{MAP}} = \arg \max_{x_n \in \mathbb{R}} p_n^x(N_\mathrm{i}, x_n).$$

Application aux bancs de filtres suréchantillonnés

Dans cette partie, nous appliquons les schémas d'estimation présentés précédemment aux bancs de filtres suréchantillonnés (BFSs) [151].

Présentation succinte des BFSs

Un BFS est un banc de filtres dont le nombre M de sous-bandes, générées en sortie, est supérieur au facteur d'échantillonnage N. La Figure 10 illustre la représentation polyphase d'un BFS, à M sous-bandes, ayant un facteur d'échantillonnage $N < M$. Soient h_m, $m = 1, \ldots, M$ les M filtres d'analyse de ce BFS. La matrice polyphase $\mathbf{H}(z)$ de taille $M \times N$ qui lui associée est alors définie comme suit

$$\mathbf{H}(z) = \sum_{k=-\infty}^{+\infty} \mathbf{H}_k z^{-k}, \tag{49}$$

où $\mathbf{H}_k$, $k = 0, \ldots, K-1$ est une suite de matrices de taille $M \times N$ pouvant être construites à partir des filtres h_m, $m = 1, \ldots, M$ [85, 86].

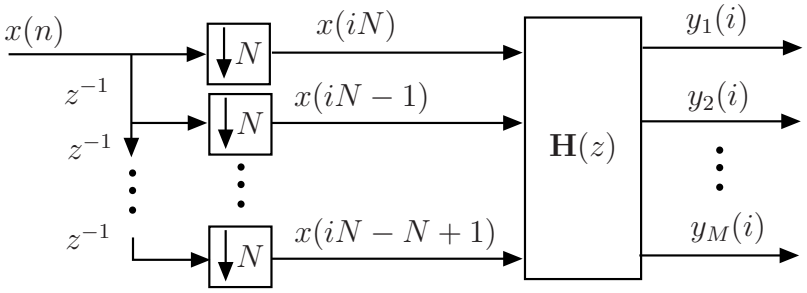


Figure 10.: Représentation polyphase d'un BFS d'analyse, $N < M$.

Il a été montré dans [87, 86], que lorsque $\mathbf{H}(z)$ est de rang plein N et que les filtres h_m sont de réponse impulsionnelle finie, la matrice d'expansion $\mathbf{H}$ associée au BFS, dans le domaine temporel, est de la forme suivante

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \mathbf{0} \\ \mathbf{H}_1 & \mathbf{H}_0 & \mathbf{0} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{H}_{K-1} & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & \ddots & \mathbf{0} & \vdots \\ \vdots & \ddots & \mathbf{H}_{K-1} & \ddots & \mathbf{H}_1 & \mathbf{H}_0 & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{H}_{K-1} & \cdots & \mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix}, \tag{50}$$

où $K \in \mathbb{N}^*$ et que la matrice de parité $\mathbf{P}$ associée à $\mathbf{H}$ est de la forme

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_0 & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \mathbf{0} \\ \mathbf{P}_1 & \mathbf{P}_0 & \mathbf{0} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{P}_{K'-1} & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & \ddots & \mathbf{0} & \vdots \\ \vdots & \ddots & \mathbf{P}_{K'-1} & \ddots & \mathbf{P}_1 & \mathbf{P}_0 & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{P}_{K'-1} & \cdots & \mathbf{P}_1 & \mathbf{P}_0 \end{pmatrix}, \tag{51}$$

où $K' \in \mathbb{N}^*$.

Nous pouvons alors appliquer les schémas d'estimation cohérente et d'estimation par propagation de croyances au cas des BFSs.

Résultats expérimentaux

Nous avons considéré un BFS basé sur les filtres de Haar avec $N = 4$ et $M = 6$. Soit $\mathbf{x} \in \mathbb{R}^{12}$ le vecteur placé en entrée du BFS. Les composantes de $\mathbf{x}$ sont i.i.d selon la loi normale avec une variance $\sigma_x^2 = 1$. La matrice $\mathbf{H}$ considérée est de taille 18×12 et elle est définie comme suit

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_1 & \mathbf{H}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix}, \tag{52}$$

avec

$$\mathbf{H}_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \text{ et } \mathbf{H}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 \end{pmatrix}. \tag{53}$$

La matrice de parité est

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{P}_1 & \mathbf{P}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_1 & \mathbf{P}_0 \end{pmatrix},$$

avec

$$\mathbf{P}_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & -0.5 & 0.5 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \text{ et } \mathbf{P}_1 = \begin{pmatrix} 0 & -1 & 1 & 1 & 0.5 & 0.5 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{54}$$

Chaque composante du vecteur $\mathbf{y} \in \mathbb{R}^{18}$, obtenu en sortie de $\mathbf{H}$, est quantifiée à l'aide d'un quantificateur scalaire Q sur $\rho = 4$ bits, puis modulée et transmise sur un canal AWGN avec un SNR variant entre 0 dB et 14 dB.

Les différentes méthodes de reconstruction que nous comparons sont décrites dans ce qui suit.

Estimation cohérente L'estimation cohérente est appliquée et le vecteur de quantification $\mathbf{z}^{\mathrm{f}}$, qui est le plus vraisemblable parmi les vecteurs de quantification cohérents, est obtenu. Deux méthodes de reconstruction du signal $\mathbf{x}$ sont alors considérées. La première est obtenue en résolvant le programme quadratique (35) avec $\mathbf{z}^{\mathrm{f}}$. Cette estimée sera notée par $\widehat{\mathbf{x}}_{\mathrm{QP}}$. La seconde estimée que nous considérons est celle qui est obtenue en utilisant la méthode de reconstruction par les moindres carrés

$$\widehat{\mathbf{x}}_{\mathrm{LS}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T Q^{-1}\left(\mathbf{z}^{\mathrm{f}}\right).$$

Estimation par propagation de croyances L'estimation MAP basée sur l'algorithme de propagation de croyances est appliquée et l'estimée définie par (48) est obtenue. Cette estimée sera notée par $\widehat{\mathbf{x}}_{\mathrm{BP}}$.

Estimation à partir des décisions dures du canal Cette estimation utilise les décisions dures prises par un décodeur canal classique sur l'observation $\mathbf{r}_{1:M}$. Un vecteur $\widehat{\mathbf{y}}_{\mathrm{HD}}$ de sous-bandes est alors obtenu. Le vecteur d'entrée peut être reconstruit par les moindres carrés

$$\widehat{\mathbf{x}}_{\mathrm{HD}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\widehat{\mathbf{y}}_{\mathrm{HD}}.$$

Reconstruction de référence Il s'agit du schéma de reconstruction de $\mathbf{x}$, en absence de bruit de transmission

$$\widehat{\mathbf{x}}_{\mathrm{REF}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\widehat{\mathbf{y}}_{\mathrm{REF}},$$

avec $\widehat{\mathbf{y}}_{\mathrm{REF}} = Q^{-1}(\mathbf{z})$. Seul le bruit de quantification est présent dans $\widehat{\mathbf{x}}_{\mathrm{REF}}$.

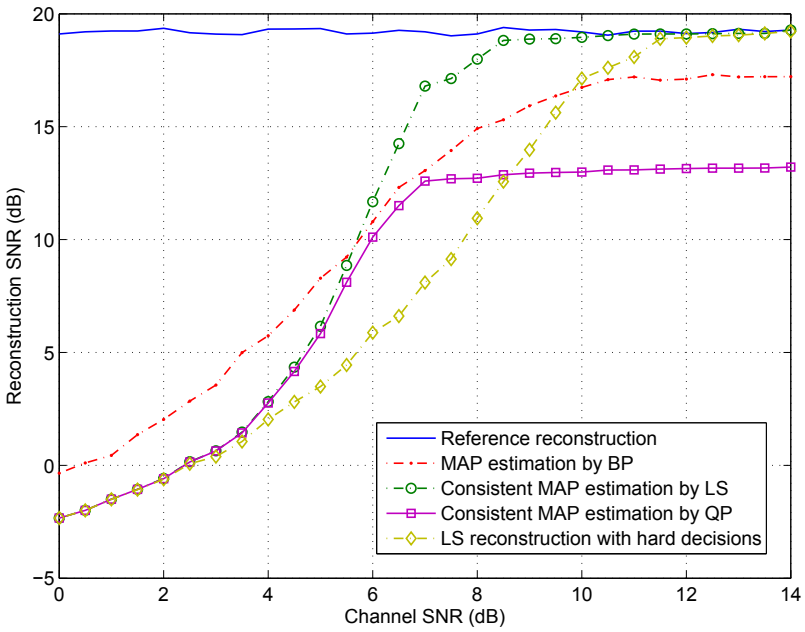


Figure 11.: SNR (dB) des signaux reconstruits en fonction du SNR du canal (dB), pour le banc de filtres suréchantillonnés basé sur les filtres de Haar.

La Figure 11 montre le rapport signal à bruit (SNR) moyen des signaux reconstruits avec les méthodes de reconstruction décrites ci-dessus, en fonction du SNR du canal. Pour un SNR canal inférieur à 8.5 dB, les méthodes d'estimation proposées (estimations cohérente et par propagation de croyances) sont plus efficaces

que l'approche de reconstruction classique basée sur les décisions dures du canal. Par exemple, pour un SNR canal de 4 dB, le gain obtenu en termes de SNR de reconstruction est d'environ 2 dB pour $\widehat{\mathbf{x}}_{\mathrm{QP}}$ ou $\widehat{\mathbf{x}}_{\mathrm{LS}}$. Pour le même niveau de SNR, le gain en SNR de reconstruction est plus important en considérant l'estimation par propagation de croyances et il est environ égal à 4 dB.

Pour un SNR canal inférieur à 4 dB, les reconstructions $\widehat{\mathbf{x}}_{\mathrm{QP}}$ et $\widehat{\mathbf{x}}_{\mathrm{LS}}$, obtenues par estimation cohérente donnent les mêmes performances. Par contre, pour un niveau de SNR canal plus élevé, $\widehat{\mathbf{x}}_{\mathrm{LS}}$ devient plus efficaces que $\widehat{\mathbf{x}}_{\mathrm{QP}}$ et l'écart en SNR de reconstruction entre ces deux estimées se creuse lorsque le SNR du canal devient plus élevé, pour enfin se stabiliser à environ 6 dB. Ce comportenment est expliqué par le fait que pour des bruits de transmission très faibles, le vecteur de quantification le plus vraisemblable $\mathbf{z}^*$ est celui qui a été transmis. La reconstruction par les moindres carrés minimise dans ce cas le bruit de quantification, ce qui est généralement plus efficace que maximiser l'a priori $p_{\mathbf{X}}(\mathbf{x})$ sur l'ensemble $\mathcal{X}(\mathbf{z}^*)$.

L'estimation par propagation de croyances est plus efficace que l'estimation cohérente pour des niveaux faibles de SNR canal. Pour un SNR canal supérieur à 5.5 dB, $\widehat{\mathbf{x}}_{\mathrm{LS}}$ devient plus efficace que $\widehat{\mathbf{x}}_{\mathrm{BP}}$ et pour un SNR canal supérieur à 9.5 dB, $\widehat{\mathbf{x}}_{\mathrm{HD}}$ est également plus efficace que $\widehat{\mathbf{x}}_{\mathrm{BP}}$. Cela pourrait être expliqué par le fait que l'obtention de $\widehat{\mathbf{x}}_{\mathrm{BP}}$ se base sur des distributions de probabilités marginales au lieu de la distribution de probabilité jointe de $\mathbf{x}$. De plus, l'algorithme de propagation de croyances peut ne pas converger en présence de cycles dans le graphe $\mathcal{G}_{\mathbf{H}}$. Ces deux effets n'appraissent pas lorsque l'effet du bruit est très important.

Conclusions et perspectives

Dans cette thèse, nous nous sommes intéressés aux schémas de codage et de décodage conjoint source-canal des contenus multimédia. Nous avons montré comment la redondance laissée par le codeur vidéo pouvait être exploitée pour réaliser un décodage robuste des séquences transmises sur un lien radio-mobile bruité. Grâce au schéma de décodage conjoint proposé, le nombre de paquets corrompus est significativement réduit au prix d'une très légère augmentation du débit. Nous avons appliqué ce schéma de décodage robuste à la transmission par descriptions multiples sur une architecture mixte Internet et radio-mobile. Le décodage source-canal conjoint des paquets reçus permet de corriger les erreurs de transmission et d'augmenter ainsi le nombre de paquets utilisés par le décodeur à descriptions multiples pour compenser les paquets perdus. L'efficacité de ce schéma a été démontrée par rapport à un schéma classique basé sur les décisions dures du canal et sur un code correcteur d'erreurs introduisant le même niveau de redondance.

Une deuxième partie de la thèse a été consacrée à l'étude de schémas de codage source-canal conjoint basés sur une transformation redondante. Deux schémas d'estimation ont été proposés. Dans le premier schéma, nous avons exploité la redondance structurée introduite et le caractère borné du bruit de quantification pour

construire un estimateur cohérent corrigeant les erreurs de transmission. Dans le deuxième schéma d'estimation, nous avons appliqué l'algorithme de propagation de croyances pour évaluer les distributions a posteriori des composantes du signal d'entrée, à partir des sorties bruitées du canal. Nous avons appliqué ces deux schémas aux bancs de filtres suréchantillonnés et constaté leur supériorité par rapport à un schéma de décodage classique basé sur les décisions dures du canal.

A l'issue de cette thèse, plusieurs pistes de recherche peuvent être envisagées. Pour conclure ce résumé, nous proposons d'en décrire quelques unes.

Le schéma de décodage robuste proposé pourrait être amélioré. Comme nous l'avons mentionné, certaines erreurs de transmission sont indétectables par les tests de syntaxe introduits. Afin d'augmenter les taux de détection et de correction d'erreurs, il est possible d'insérer des informations supplémentaires sur les paquets transmis, dans les enêtes correspondantes, comme par exemple le nombre de coefficients non nuls dans une sous-bande spatio-temporelle, le CRC d'un bloc ou d'un ensemble de blocs, etc. Par ailleurs, ces informations supplémentaires pourraient être choisies et adaptées selon le degré d'importance des blocs à protéger. De manière similaire, une optimisation de la complexité du décodage en fonction de la sensitbilité des données décodées, pourrait être envisagée. Le paramètre M de l'algorithme séquentiel peut être adapté à la couche décodée et à la sous-bande spatio-temporelle du bloc courant.

Le schéma de décodage robuste appliqué aux descriptions multiples a montré son efficacité pour de faibles niveaux de redondance. En effet, les schémas classiques de décodage basés sur des codes canal sont généralement plus robustes pour des niveaux de redondance élevés, et lorsque les conditions de transmission sont bien connues. En se plaçant à des niveaux de redondance plus importants, les schémas classiques combinent généralement des codes correcteurs d'erreurs et des codes canal à pertes de paquets, ce qui augmente nettement leurs performances. Il serait alors intéressant de comparer de tels schémas à un schéma de codage par descriptions multiples plus redondant que celui que nous avons considéré, et dans lequel la redondance est introduite par exemple en duplicant la sous-bande d'approximation spatio-temporelle.

Pour faire le lien entre les schémas étudiés pendant cette thèse, une perspective à envisager est d'appliquer notre technique d'estimation cohérente au schéma de décodage robuste par descriptions multiples. En effet, les deux descriptions sont générées à l'aide d'un banc de filtres suréchantillonné. Il est alors possible d'exploiter la redondance structurée introduite par ce banc de filtres et la redondance résiduelle pour améliorer le décodage conjoint des deux descriptions. Certaines erreurs non détectées par les tests de syntaxe utilisés, peuvent alors être détectées et corrigées grâce aux tests de cohérence introduits.

De manière plus générale, il reste un gros travail à faire pour optimiser les algorithmes d'estimation proposés afin de les appliquer à des bancs de filtres à matrice d'analyse plus dense et à un facteur de suréchantillonnage plus faible. En particu-

lier, l'algorithme d'estimation par propagation de croyances pourrait être amélioré en incluant les contraintes de parité dans le graphe représentant les dépendances linéaires entre les variables. De plus, afin de pouvoir appliquer ces deux schémas d'estimation au codage d'images et de vidéo, il semble nécessaire d'étudier leurs comportement et performances sur des bancs de filtres suréchantillonnés ayant un fort gain de codage.

1. Introduction

1.1. Motivation

The development of multimedia broadcasting and on-demand services for mobile devices, such as tablets or smartphones, involves the transmission of contents over heterogeneous networks, consisting of mixed wired and wireless channels. To transmit multimedia contents over such best-effort networks, two main problems have to be solved. First, due to bandwidth scarcity, data generated by multimedia sources have to be efficiently compressed. Second, the compressed bit streams have to be made robust to the unavoidable transmission impairments. Current image and video coders achieve high compression ratio. They are however very sensitive to transmission errors and to packet lossses. Indeed, a single bit error or packet loss, affecting the compressed bit stream, may have a detrimental effect on the reconstructed content.

Conventional communication systems based on Shannon's separation principle handle these issues by optimizing separately the source and the channel coders. Such an optimization assumes that the source and the channel characteristics are perfectly known and that the processed blocks are of infinite lengths. However, these hypotheses cannot be met in practical situations, due to delivery delay, complexity constraints and to unknown and time-varying transmission conditions. As a consequence, alternative coding schemes, namely the joint source-channel coding and decoding techniques have been developed to address these issues. In these schemes, the source and the channel coders are jointly designed to enable the optimization of the communication system resources when there are stringent delay and complexity constraints, and to increase the robustness of multimedia contents transmitted over unreliable networks.

The work conducted during this thesis fits into this category and aims at providing efficient joint-source channel coding and decoding techniques to ensure a robust multimedia transmission. The targeted applications are the real-time and mobile applications, in which the retransmission of damaged or lost data is not allowed.

We start by showing how the residual redundancy left by video coders, and wasted by classical separate source-channel decoders, can be exploited to design a joint

source-channel decoder able to detect and correct the transmission errors. Corrupted packets, which would be dropped in conventional communication schemes, are then corrected. This decoding approach is further applied to multiple description video streams transmitted over a mixed architecture consisting of a wired lossy part and a wireless noisy part. The errors introduced by the wireless transmission are corrected by the joint-source channel decoder whereas the losses occurring during the wired communication are compensated thanks to multiple description coding.

Another direction investigated in this thesis concerns the joint source-channel coding schemes based on a redundant transform, such as oversampled filter banks. Such schemes introduce some structured redundancy in the transmitted bit stream. In the presence of transmission errors, we show how this redundancy may be used to improve the decoding performance by proposing two efficient estimation approaches. The first one exploits the linear dependencies between the output variables, jointly to the bounded quantization noise, to perform a consistent estimation of the source outcome. The second approach uses the belief propagation algorithm to estimate the input signal via a message passing procedure along the graph representing the linear dependencies between the variables. These two schemes are further applied to estimate the input of an oversampled filter bank and their performance are compared.

1.2. Outline of the Thesis

This manuscript is organized as follows. In Chapter 2, we start by motivating the resort to joint source-channel coding and decoding approaches. We overview some relevant existing methods and particularly focus on the approaches close to the ones we propose, such as joint source-channel decoding, multiple description coding and source coding schemes based on redundant transforms.

Our joint-source channel decoding scheme is described in Chapter 3. We start by formulating the estimation problem of a video bit stream transmitted over some noisy channel. Then, we consider a particular wavelet video coder and identify a part of the residual redundancy it leaves. This allows us to build an efficient test detecting the damaged parts of the bit stream as the ones which are not compliant with the redundancy. Combined with a sequential decoding algorithm, we show how this test is able to correct most of the transmission errors, with a manageable complexity.

An application of our joint source-channel decoder to multiple description coding schemes is described in Chapter 4, where we consider a general transmission chain consisting of a packet-loss network followed by a noisy wireless channel. The idea behind the use of multiple description coding is to mitigate the effects of packet losses, while avoiding retransmission and reducing delays. We show how joint source-channel decoding significantly improves the performance of the multiple description decoder, by increasing the number of received error-free packets.

In Chapter 5, we focus on joint source-channel coding schemes based on a redundant linear transform. We address the problem of estimating a signal placed at the input of such a transform, when its output is corrupted by some transmission noise. Then, we describe the consistent estimation scheme we propose and the estimation approach based on belief propagation. We show how these two schemes find an application in oversampled filter banks and in particular, how the consistent estimation technique may be iteratively implemented in this case.

Finally, conclusions and perspectives are drawn in Chapter 6.

The remainder of the manuscript provides a succinct presentation of the sum-product algorithm, from which the belief propagation algorithm we used is derived, in Appendix A. The oversampled filter banks are then briefly introduced in Appendix B. An image denoising technique, based on adaptive lifting schemes and proposed during this thesis, is presented in Appendix C, as it is not directly related to the main topic of this work.

1.3. Summary of the Contributions

This thesis contains several original contributions, which are summarized in the sequel

- An efficient joint source-channel decoding scheme increasing the robustness of video contents transmitted over noisy channels and presenting a manageable complexity.
- An improvement of an existent multiple description coding scheme, thanks to joint source-channel decoding.
- A consistent estimation technique for joint source-channel coding schemes, based on a redundant linear transform and its iterative implementation adapted to oversampled filter banks.
- An application of the belief propagation algorithm for estimating the input of an oversampled filter bank from its noisy subbands.

2. Joint Source-Channel Coding and Decoding

2.1. Introduction

In video communication systems, the usage of heterogeneous and best effort networks has become widespread. In such networks, the quality of service (bandwidth, losses and errors rates, delay, *etc.*) required by video communication applications such as on-demand video streaming, is not always guaranteed. Data packets may be lost due to traffic congestion or corrupted by impairments in physical channels.

The issues associated to video transmission are then twofold:

- Data delivered by the video encoders has to be efficiently compressed due to bandwidth scarcity.
- Compressed video bit streams have to be made robust to the unavoidable transmission impairments.

Increasing the robustness of video content is a challenging problem and it raises several issues.

The high compression efficiency reached by many encoders such as H.264/AVC [70] and Vidwav [126], makes the compressed video streams extremely sensitive to errors occurring during transmission. For example, due to spatial and temporal predictions, errors can rapidly propagate in time and space leading to the so called *drift* effect. In a similar way, when using an entropy coder such as CAVLC [70] and CABAC [103], a single bit error is likely to cause a desynchronization between the encoder and the decoder making the following bits useless, even if they were correctly received. This leads to an uncontrolled degradation of the whole video quality.

On the other hand, the characteristics of the transmission channel are generally unknown and time-varying. This makes it difficult to tune efficiently a robust transmission scheme for a given video signal. Furthermore, a reliable transmission scheme over heterogeneous channels has to take into account the hierarchical nature of the delivered video bit streams: scalability and robustness should be jointly considered in order to design an efficient source and channel coding system that allows the

video content to be transmitted and used at different resolutions and/or quality levels with respect to specific terminals and users preferences.

Finally, stringent delay and complexity constraints are often imposed by real-time video applications and have to be taken into account in the design of a communication system.

A number of approaches have been proposed to ensure a robust video transmission:

1. A first group includes the approaches wherein the source and the channel coders work separately, or in *tandem*, in order to make the delivered bit streams more robust to transmission losses and errors. These methods are referred to as the tandem-coding techniques.
2. A second group gathers the methods in which the source and the channel coders are jointly designed to overcome transmission impairments. These are known as the joint source channel coding and decoding techniques.
3. A last group contains the methods which act at the decoder side to conceal the impacts of unrecoverable and/or corrupted data, on the reconstructed signals. These methods are known as error concealment techniques and they can be jointly employed with the approaches in the first two groups in order to increase the decoding quality.

Section 2.2 presents the first group of approaches, namely the tandem coding schemes and points out to its limitations. Then, Section 2.3 describes the principle of joint source channel coding techniques. Two main approaches of this group are reviewed in Sections 2.4 and 2.5 since the contributions of this thesis fit into this category. Finally, some error concealment schemes are presented in Section 2.6.

2.2. Tandem Coding Techniques

The tandem coding techniques rely on Shannon's separation theorem [141] stating that source coder and channel coder can be optimized separately without introducing any performance sacrifice for the overall communication system. In Shannon's theory, the source and channel are assumed to be stationary with known characteristics. Moreover, the blocks are assumed to be of infinite lengths, which makes the delay and the complexity grow to infinity. Under these hypothesis, the source coder can approach the optimal rate-distortion behavior and the channel can transmit the bit stream with a probability of error arbitrary small provided that its coding rate is below its capacity.

Usually in tandem coding techniques, the channel coder is first optimized with respect to the channel capacity to ensure a reliable transmission, then the source coder is optimally designed with the available source coding rate.

In the next section, a brief overview of the channel coding techniques is presented.

2.2.1. Channel Coding

Classical approaches for combating channel/network impairments consist in adding structured redundancy to the compressed bit stream at the transmitter side. This redundancy may be introduced at the physical layer of the protocol stack used to transmit the packetized content, by using bit-level Forward Error Correction codes (FEC) [93, 133], to combat transmission errors. Packet erasure codes, such as Reed-Solomon codes [132], Fountain codes [19], or Raptor codes [142] may be introduced at intermediate layers of the protocol stack to recover lost packets, as is the case, *e.g.*, in Multimedia Broadcast/Multicast Services [74, 107]. In both cases, the redundancy introduced results into an increased final bit rate. At the receiver side, error detection mechanisms, such as Cyclic Redundancy Check (CRC) or checksums are employed to verify the efficiency of the channel decoding at physical layer. If enough error-free packets are received, packet erasure codes may recover the missing packets, otherwise retransmission is requested using Automatic Repeat Request (ARQ) [94] mechanisms, when the application is delay-tolerant.

2.2.2. Limitations of Tandem Systems

As previously mentioned, the separation principle inspired the development of tandem systems. It is assumed that an optimal source encoder removing all the source redundancy is followed by an optimal channel coder that corrects all the errors provided that the coding rate is below the channel capacity. These assumptions can only be met for idealized conditions, *e.g.*, codes of infinite length, infinite delay and complexity, and do not hold in practical situations. For example, channel codes process blocks of finite lengths, for which the probability of error is not equal to zero. The separation principle does not provide any optimal algorithm to design such channel codes, and it does not address either the problem of designing optimal source coders and decoders when the error probability is non-zero.

In practical communication situations, the characteristics of the transmission channels are time-varying and the transmitted packets have different protection needs. Therefore, classical error correcting codes and packet erasure codes may be either oversized, leading to a waste of the available bandwidth, or not strong enough, resulting in residual transmission errors. Adaptation of these codes to the source and the channel conditions may be considered [67], which implies a communication system in which the source and the channel coders are jointly designed. Moreover, the bit streams delivered by the source coder may still contain a certain amount of redundancy and it appears as common sense to use this redundancy as a form of channel coding, at the receiver side. One may even consider to intentionally introduce some redundancy during source coding, to enhance the bit stream robustness to transmission impairments, which may lead to removing or reducing the protection introduced by channel coder.

Based on these observations, several schemes in which source and channel coders are optimized jointly have been proposed. In the next section, a review of such schemes is presented.

2.3. Joint Source-Channel Coding and Decoding Techniques

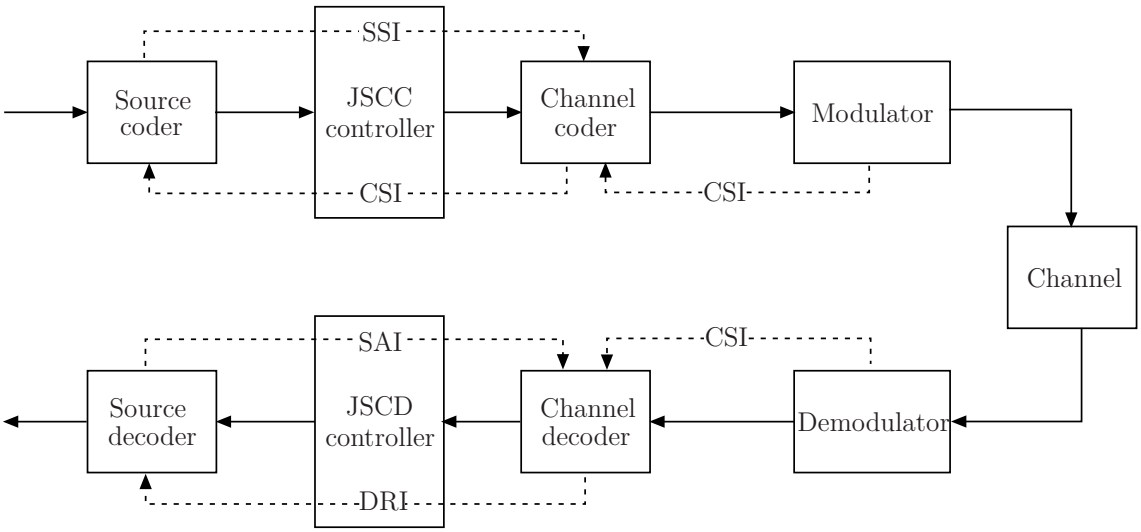


Figure 2.1.: Generic joint source-channel coding/decoding system.

A joint source-channel (JSC) communication system is a system in which the different blocks of the communication chain are jointly designed. For example, in some JSC schemes, the source and channel coders are designed and optimized together to yield a minimal distortion for given channel conditions [20, 24, 21, 169, 96]. In other schemes, the channel coder is designed for a given source coding scheme, or inversely [52].

A generic JSC coding scheme is presented in Figure 2.1. The blocks of this scheme *communicate* together by sending different types of information [117, 84]. At the encoder side, the source significance information (SSI), sent by the source coder provides the channel coder with the protection degrees required by the different parts of the compressed bit stream. The SSI ensures then a *source-controlled* channel coding by adapting the channel protection to the sensitivity level of the considered packet of data. The channel state information (CSI), containing information about the channel transmission conditions, is fed back from the channel encoder to the source encoder. A coordinating module, namely the JSC coding controller uses the SSI and CSI to drive an efficient and robust transmission scheme. For example, a scalable source coding may be coupled with an adaptive channel coder invloving

selective protection and a modulation suitable to the channel conditions [117, 84]. At the receiver side, the decoder reliability information (DRI) sent by the channel decoder to the source decoder allows to evaluate the channel decoder soft decisions. The source *a priori* information (SAI) transmitted from the source decoder to the channel decoder may be taken into account in combination with the DRI to ensure a *JSC-controlled* decoding. It should be noted that the previously-mentioned information are not necessarily (or always) available in all the JSC schemes. In certain JSC schemes, part of this information is optional or absent.

Although the design of JSC communication systems has gained increased popularity during the last years, leading to a large variety of contributions, a commonly recognized classification of these techniques is still lacking. Overviews on JSC systems are found in [44]. A more recent work, [49], presents a comprehensive survey of the existing JSC schemes.

Based on the above description, we categorize the existing JSC schemes into two main classes, according to whether the JSC design takes place at the encoder or at the decoder side:

1. *Joint source-channel coding (JSCC) schemes:* In these schemes, the source encoder and the channel encoder are jointly considered, which is opposite to the separate source and channel coding. The channel coder can be controlled by the source information provided by the source coder, which helps to improve the overall coding efficiency. Similarly, the channel state information may be fed back to the source coder in order to perform a source coding adapted to the channel conditions. The most important JSCC techniques are: JSCC for optimal rate allocation, unequal error protection, multiple description coding, JSCC using optimized quantization, *etc.*
2. *Joint source-channel decoding (JSCD) schemes:* In these schemes, the JSC design takes effect at the decoder side. The decoder reliability information provided by the channel decoder can be used jointly to the source a priori information to improve the decoding of the received packets. Most of the JSCD techniques exploit the redundancy introduced in the packets during the transmission over the layers of the protocol stack. These techniques also use the redundancy left by the source coder in the compressed stream to perform robust decoding.

In this thesis, two different JSC approaches have been considered for increasing the video transmission reliability. In a first time, we considered a JSC decoding technique, based on the use of the redundancy left by the video source coder. This technique allows to detect and correct some of the transmission errors. In a second time, a multiple description coding scheme has been considered to mitigate the effect of packet losses occurring during a lossy transmission.

In Sections 2.3.1 and 2.3.2, we briefly present two JSCC approaches, namely the JSCC for rate allocation and JSCC by unequal error protection, which are different from the methods we considered in this thesis, but which allow us to illustrate how

the source and the channel coders may cooperate to yield a robust transmission scheme. The two next sections, Sections 2.4 and 2.5 will be devoted to review the multiple description coding and JSCD techniques.

2.3.1. JSCC for Rate Allocation

The principal motivation of JSCC is to derive practical communication schemes suitable for realistic transmission situations which are usually subject to some constraints, such as complexity and/or coding rate. The aim of this section is to present the JSCC schemes responding to the second type of constraints, namely the coding rate constraint, which we will refer to as the JSCC for rate allocation between the source and the channel coders.

Unlike the tandem communication systems, in which the redundancy is removed by the source coder and then re-introduced at the channel coder side, the JSCC-RA considers the source and the channel coders as a single coding block consuming a certain bit rate $R_s + R_c$, where R_s and R_c are the rates consumed by the source and the channel coders respectively. A high source coding rate R_s results in a low source distortion D_s, whereas a high channel coding rate R_c implies a strong error protection and correction capabilities and thus a weak channel distortion D_c.

The aim of JSCC for rate allocation is to determine, for a fixed transmission rate budget R_b, the optimal rate distribution (R_s^*, R_c^*) between the source and the channel, which minimizes the overall average end-to-end transmission distortion $D = D_s + D_c$. The basic idea behind this optimization problem is illustrated in Figure 2.2.

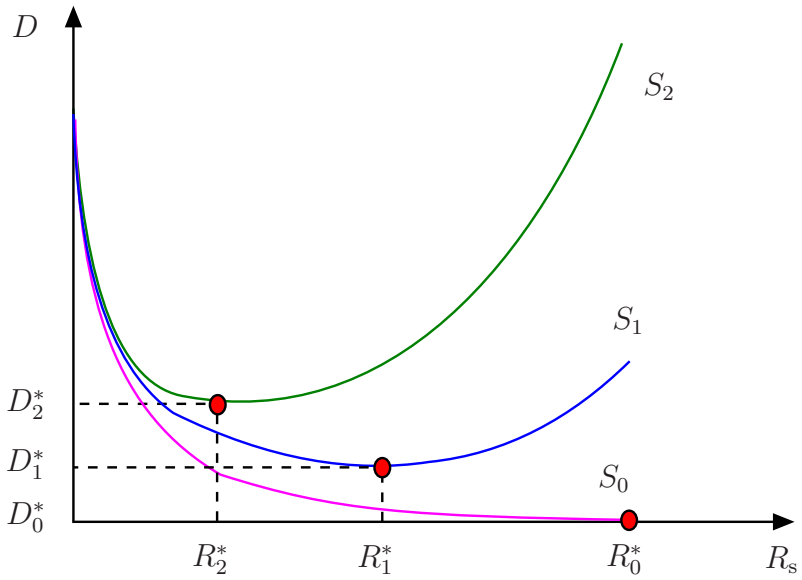


Figure 2.2.: Illustration of a joint source-channel rate allocation problem.

Three different channel states S_0, S_1 and S_2 are considered. Each channel state

S_j, $j = 0, 1, 2$ is described by a channel state information, *e.g.* bit error rate and characterized by a global distortion $D_j(R_s, R_c)$, depending on the rate allocation between the source and the channel coders. Imposing $R_c = R_b - R_s$ makes D_j depends only on R_s, as shown in Figure 2.2. The channel state S_0 represents the case where the channel is error-free ($D_c = 0$) and is illustrated by the lowest curve in Figure 2.2. In this case, increasing the source rate R_s leads to a decreased distortion D_0, as in standard rate-distortion theory, and the lowest distortion is thus obtained by utilizing the largest available bit rate R_b, represented here by the point ($R_0^* = R_b$,$D_0^* = 0$). When the channel is in states S_1 or S_2, this trend does not hold anymore, as the global distortion is given by both source and channel distortions. The more bits are allocated to source coding, the less bits will be left for channel coding, and the higher will be the resulting channel distortion. As illustrated in Figure 2.2, the different channel states S_1 and S_2 result in two different optimal rate allocation strategies indicated by the points (R_1^*,D_1^*) and (R_2^*,D_2^*) respectively. The determination of the optimal rate allocation strategy depends strongly on both source and channel models and parameters.

Rate-distortion (R-D) models for video transmission over noisy channels and lossy networks is a topic of numerous studies [20, 24, 170, 21, 169, 96]. Most of the investigation conducted in this context include the reduction of the effects caused by packet losses and by channel errors. Some of the rate allocation approaches have been proposed for coding mode selection in unreliable environments [127, 34, 170]. Other works investigate the tradeoff between source and channel coding for vector quantization [69, 46, 45]. Recently, in [96], Sabir *et al.* proposed a model that estimates the distortion introduced by quantization and by channel errors in MPEG-4 video streams. This model takes into account the motion estimation and compensation, as well as the transform and entropy coding to evaluate the distortion in the intra- and inter- predicted frames.

A way to perform JSC for rate allocation is by unequal error protection (UEP) of the encoded stream. Several UEP techniques have been proposed for multimedia transmission using R-D optimization frameworks [65, 24, 20].

2.3.2. Unequal Error Protection

The main image and video compression standards, *e.g.,* JPEG-2000 [28], MPEG-2, MPEG-4, H.264/AVC [70] and Vidwav [126] deliver compressed bit streams with unequal bit sensitivities. Some output bits of these source coders have a more important impact on the quality of the reconstructed source, and are thus more sensitive to transmission errors than the other output bits. The principle of UEP is to apply a stronger channel protection to the bits that have a greater impact on the reconstructed signal quality, and weaker channel codes to the less sensitive parts of the compressed bit stream. In theory, a channel encoder mapped to the source encoder may be designed provided that the different bit error sensitivities are

known for each bit position. However the channel coding adaptation to such a fine granularity is complex in practice.

Usually the implementation of practical UEP schemes involves the hierarchical bit stream representation which is achieved through scalability and data partitioning:

- Scalability is the capacity of a compressed bit stream to adapt itself to the transmission conditions as well as to the terminals characteristics. An appropriate partial decoding of the scalable bit stream allows to generate a version of the original signal, with a lower quality and/or a reduced resolution (spatial/temporal) level. For example, recent hybrid video codecs such as H.264/SVC [139] support scalability and generate hierarchical bit streams organized into one base layer, which has the highest protection demand, and several refinement layers with different temporal/spatial resolutions and quality levels, which results in various error sensitivities. Wavelet-based video coders, such as Vidwav [126], offer an inherent spatio-temporal scalability thanks to the multiresolution paradigm.
- Data partitioning consists in a hierarchical representation which separates the more critical parts of the bit stream (such as headers, motion vectors, low frequency DCT and wavelet coefficients, etc.) from the less important parts (such as higher frequency DCT and wavelet coefficients, etc.).

The channel encoder uses the knowledge of the source bit error sensitivities to perform UEP. Particular error correction codes supporting rate compatibility have been proposed to provide different protections according to the sensitivity level of the content to be transmitted. For example, rate compatible convolutional codes [67] and rate compatible Turbo codes [33] can easily obtain different channel code rates through puncturing the same low-rate mother code. In [171], authors proposed an UEP technique for the 3D-SPIHT video coder. The 3D-SPIHT bit stream is first divided into blocks of the same length. Each block is supplemented with a CRC and then encoded using the a rate compatible convolutional code prior to its transmission through the channel. At the decoder side, a treillis decoder chooses the path with the *best* metric satisfying the checksum equations. An automatic repeat request (ARQ) strategy is used when the decoder fails to find a satisfying solution in a reasonable time. Another UEP approach has been proposed in [26] for the MPEG-4 video bit streams transmitted over error-prone wireless networks, where rate compatible convolutional coding is performed with rates chosen according to the perceived importance of bits. In [12], authors introduced a semi-analytical model that estimates the distortion introduced in H.264/AVC bit streams. They studied the effects of error propagation in the next frames of the video sequence, caused by inter prediction as well as the distortion in intra-predicted frames. The proposed model is then used to evaluate the different protection levels in order to implement UEP to the coded video bit stream.

As mentioned in Section 2.3.1, JSCC for rate allocation and unequal error protection may be considered together to jointly derive optimal source and channel coding

rates. In [20], Bystrom and Modestino considered a video transmission scheme in which the bit stream data is first classified according to its priority, then protected by rate compatible convolutional codes before being transmitted over an additive white Gaussian noise (AWGN) channel. They proposed a method which optimally allocates source and channel coding bits under a fixed constraint on the transmission bandwidth. For a given video sequence, rate-distortion curves are constructed and used to determine the optimal rate and distortion values. Despite of its high accuracy, this approach requires a significant amount of computations to construct the rate-distortion curves for each video sequence which makes it infeasible for real-time video applications. Few years later, Bystrom and Stockhammer introduced in [21] a generic method for optimal source and channel rate allocation. To limit the complexity of this problem, they used polynomial models to approximate the rate-distortion surfaces for different video frames. These surfaces are then used with models of the channel bit error rates in order to allocate the source and the channel coding rates between the different frames of a video sequence, with the goal of minimizing the overall distortion under a constraint of the total rate. This scheme was successfully applied to a video sequence compressed with H.263 and channel coded using rate-compatible systematic recursive convolutional codes. In [24], the authors proposed an efficient method for allocating source and channel bits between the subbands of the scalable video coder proposed in [146], such that the overall distortion is minimized given the channel conditions and a total bit rate budget. The spatio-temporal subbands obtained by 3D subband coding are successively refined via layered quantization techniques and coded using a context-based arithmetic coder. An UEP is then applied on the source bits using the rate compatible codes.

As discussed in this section, UEP techniques exploit the hierarchical manner in which the bit streams are usually organized in order to perform channel coding. In particular, when scalable or layered bit streams are considered, the highest level of protection is accorded to the more important base layer. Although this technique is relatively simple and has proven to be efficient, it has the main drawback of being advantageous only if the base layer is received, otherwise the enhancement layers are undecodable even if correctly received. Usually, in such situations, the base layer is retransmitted. However in applications which do not tolerate delay, such a retransmission is not possible.

An important family of JSCC techniques, namely the multiple description coding techniques, has been proposed to overcome such situations, by creating several versions of the signal, each of which being independently decodable. This is the topic of the next section.

2.4. Multiple Description Coding

Multiple description (MD) coding is a JSCC technique which enhances the robustness of a video delivery system. The principle of MD coding consists in represent-

ing the original input signal by M correlated, but independently decodable signals called *descriptions*. These descriptions are assumed to be independently transmitted through different channels. Each transmission channel has a probability of not delivering its description. The receiver is then formed by $2^M - 1$ decoders corresponding to all the decoding situations resulting from the reception of a subset of the transmitted descriptions. When the transmission of some of the descriptions fails, the corresponding *side* decoder should be able to reconstruct the original signal from the available descriptions with an acceptable quality. The reception of all descriptions should allow signal reconstruction with the highest quality by using the *central* decoder. When descriptions are packetized, the robustness of MD coding schemes comes then from the fact that it is unlikely that the same *packet* of the encoded signal is affected in all descriptions.

The amount of redundancy introduced between the descriptions with respect to a single description (SD) coding scheme is an important issue in MD coding, since there is a trade-off between this redundancy and the resulting distortion; the more the descriptions are correlated the better will be the side decoding of a subset of received descriptions but the worse will be the R-D performance of the central decoding when all the descriptions are available. Therefore, a great effort has been spent on analyzing the performance achievable with MD coding schemes the early 1980's [165, 115, 43] and till more recently [157].

2.4.1. Theoretical Performance of MD Coding Schemes

The idea of MD coding was first investigated by the information theory community [165, 115, 43] as a source coding technique. Later, in the second half of the 1990's, with the emergence of best-effort packet-switched networks, *e.g.*, Internet, MD coding has gained more popularity as an effective (*source*) coding strategy which overcomes channel impairments and has been classified as a JSCC technique. Since then, a large variety of MD coders have been designed for speech, image and video contents. A comprehensive review on MD coding theory is provided by Goyal *et al.* in [52].

Figure 2.3 illustrates an MD coding system with $M = 2$ descriptions and 3 decoders. The source, represented by a random vector $\mathbf{X} = (X_1, \ldots, X_N)^T$ is first encoded using an MD encoder which generates two descriptions $\mathbf{X}^{(1)} = \mathbf{F}^{(1)}(\mathbf{X})$ and $\mathbf{X}^{(2)} = \mathbf{F}^{(2)}(\mathbf{X})$. These descriptions are further transmitted through two distinct channels. Let R_i, $i = 1, 2$ be the channel rate used to transmit $\mathbf{X}^{(i)}$. The MD decoder handles 2 different situations:

1. One of the channels fails to deliver its description. The side decoder associated to the other channel is then used to decode the received description. Let $\widetilde{\mathbf{X}}^{(i)} = \left(\widetilde{X}_1^{(i)}, \ldots, \widetilde{X}_N^{(i)}\right)^T$, $i = 1, 2$ denote the reconstruction at the i-th side decoder and D_i the obtained distortion.

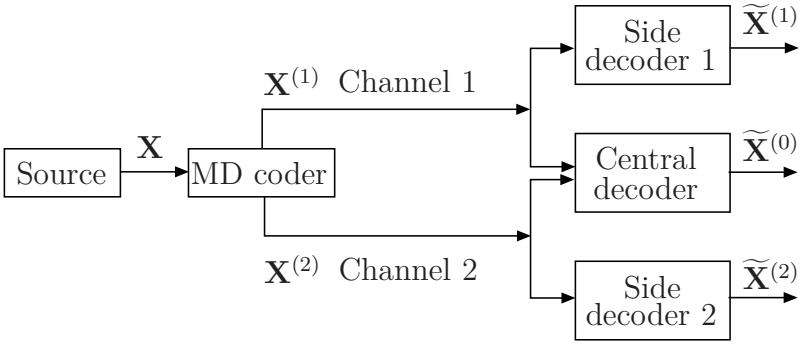


Figure 2.3.: Multiple description scheme with $M = 2$ descriptions.

2. Each channel delivers its own description. The central decoder is then used to decode the two received descriptions and the reconstructed sequence $\widetilde{\mathbf{X}}^{(0)} = \left(\widetilde{X}_1^{(0)}, \ldots, \widetilde{X}_N^{(0)}\right)^T$ is obtained, with an associated distortion D_0.
3. Both channels fail to deliver the descriptions. In this case the video signal is usually reconstructed by its mean or by using error concealment techniques.

For a given distortion measure δ, the distortion at the MD decoder output is given by

$$D_i = \frac{1}{N}\sum_{n=1}^{N} E(\delta(\widetilde{X}_n^{(i)}, X_n)), \;\; i = 0, 1, 2. \tag{2.1}$$

The set of achievable values $(R_1, R_1, D_0, D_1, D_2)$ for a Gaussian random vector $\mathbf{X}$ and a mean squared error (MSE) distortion δ has been obtained by Ozarow in [115].

Theorem 1. [53] Let $X_1, \ldots, X_N$ be a sequence of *i.i.d.* unit variance Gaussian random variables. The achievable set of rates and MSE distortions is the union of the points satisfying

$$D_i \geqslant 2^{-2R_i}, \;\; i = 1, 2 \tag{2.2}$$

$$D_0 \geqslant 2^{-2(R_1+R_2)}\gamma(D_1, D_2, R_1, R_2), \tag{2.3}$$

where

$$\gamma(D_1, D_2, R_1, R_2) = \frac{1}{1 - \left(\sqrt{(1-D_1)(1-D_2)} - \sqrt{D_1 D_2 - 2^{-2(R_1+R_2)}}\right)^2}$$

if $D_1 + D_2 < 1 + 2^{-2(R_1+R_2)}$ and $\gamma(D_1, D_2, R_1, R_2) = 1$ otherwise.

The inequalities in (2.2) represent simply the classical distortion bound for a Gaussian source encoded at a rate R_i, whereas the inequality (2.3) indicates that the central distortion D_0 must exceed the classical distortion bound $2^{-2(R_1+R_2)}$ by a factor of γ.

When the two descriptions are individually *good*, *i.e.*, $D_i = 2^{-2R_i}$, $i = 1, 2$ then $D_1 + D_2 < 1 + 2^{-2(R_1+R_2)}$ and (2.3) becomes

$$D_0 \geqslant D_1 D_2 \frac{1}{1-(1-D_1)(1-D_2)}. \tag{2.4}$$

Since

$$D_1 D_2 \frac{1}{1-(1-D_1)(1-D_2)} = \frac{1}{\frac{1}{D_1}+\frac{1}{D_2}-1} \geqslant \frac{\min(D_1, D_2)}{2},$$

the central distortion is only slightly lower than the lowest of the two side distortions, which means that relatively to the side decoding, the performance of the central decoding is poor. Inversely, assume that the central distortion is as low as possible, *i.e.*, $D_0 = 2^{-2(R_1+R_2)}$. Then $\gamma = 1$ and $D_1 + D_2 \leqslant 1 + 2^{-2(R_1+R_2)}$, which implies that unless the rates R_i, $i = 1, 2$ are very low, the reconstruction of at least one of the side decoders is very poor. Therefore, one can not have at the same time optimal central and side distortions.

Usually the transmission channels are assumed to have the same rates and losses probabilities. Therefore, practical MD schemes generally consider the balanced case, where $D_1 \approx D_2$ and $R_1 \approx R_2$, to design the two descriptions.

2.4.2. Practical MD Coding Schemes

In practical MD coding systems, the main issue is how to create descriptions such that the resulting rates, distortions and complexity satisfy some conditions prescribed by the transmission constraints.

Two main directions have been investigated by the researchers to design MD systems. In the first direction pioneered by Vaishampayan [152], the redundancy is introduced after the signal has been decorrelated, by diversifying the quantization to generate the descriptions, *e.g.*, MD by scalar quantization. The second direction that has been studied is to introduce the redundancy during the signal transformation. A redundant transform is used to generate two correlated descriptions. In this category, two main strategies are found in the literature depending on whether the introduced redundancy is statistical, *e.g.*, MD by correlating transforms or deterministic, *e.g.*, MD by frame expansion. Each of these approaches is reviewed in the sequel.

2.4.2.1. MD by Scalar Quantization

Vaishampayan was the first to provide a practical MD coding framework in which the descriptions are generated by scalar quantization followed by an index assignment

[152]. This technique, known as multiple description scalar quantization (MDSQ), consists in encoding a memoryless stationary zero-mean source $\mathbf{X} = \{X_n\}_{n\in\mathbb{Z}}$ using different scalar quantizers Q_i, $i = 1, 2$ to generate the quantized descriptions $\mathbf{X}^{(i)}$. Each scalar quantizer is defined by a dictionary $\mathcal{A}^{(i)} = \left\{a_k^{(i)}\right\}_{k=1}^{K_i}$ and a partition $\mathcal{P}^{(i)} = \left\{P_k^{(i)}\right\}_{k=1}^{K_i}$ of $\mathbb{R}$

$$\begin{aligned} Q_i : \mathbb{R} &\to \{1, \ldots, K_i\} \\ x &\to Q_i(x) = k \text{ if } x \in P_k^{(i)}. \end{aligned} \tag{2.5}$$

At the decoder side, three inverse quantizers Q_1^{-1}, Q_2^{-1} and Q_0^{-1} are considered, depending on whether only $\mathbf{X}^{(1)}$ or $\mathbf{X}^{(2)}$ or both $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are received. The inverse quantizer $Q_i^{-1} : \{1, \ldots, K_i\} \to \mathcal{A}^{(i)}$, $i = 1, 2$ associates to each quantization index $k \in \{1, \ldots, K_i\}$ the corresponding element of the dictionary $\mathcal{A}^{(i)}$: $Q_i^{-1}(k) = a_k^{(i)}$. Let $\mathcal{I}^{(0)}$ be the subset of $\{1, \ldots, K_1\} \times \{1, \ldots, K_2\}$ containing all the pairs of quantization indexes that may be generated at the output of Q_1 and Q_2. Then, the inverse quantizer $Q_0^{-1} : \mathcal{I}^{(0)} \to \mathcal{D}^{(0)}$ associates to each pair of received quantization indexes $(k_1, k_2) \in \mathcal{I}^{(0)}$ the corresponding element of the dictionary $\mathcal{A}^{(0)} = \left\{a_{k_1k_2}^{(0)},\ (k_1, k_2) \in \mathcal{I}^{(0)}\right\}$ associated to Q_0^{-1} and constructed from the central partition

$$\mathcal{P}^{(0)} = \left\{P_{k_1k_2}^{(0)},\ (k_1, k_2) \in \mathcal{I}^{(0)}\right\}, \tag{2.6}$$

where $P_{k_1k_2}^{(0)} = P_{k_1}^{(1)} \cap P_{k_2}^{(2)}$, $(k_1, k_2) \in \mathcal{I}^{(0)}$. If a received pair $(k_1, k_2) \notin \mathcal{I}^{(0)}$, one knows that a transmission error has occurred. In [152], the MDSQ is designed by minimizing the central distortion D_0 subject to maximal admissible distortions D_i. Further developments of MDSQ include design of entropy-constrained MDSQ [154, 153] and extensions to vector quantization [140, 47, 155].

2.4.2.2. MD by Correlating Transforms

In classical source coding techniques, the transform step aims at decorrelating the input signal. Usually, an orthogonal transform is used, *e.g.* the discrete cosinus transform (DCT) or the wavelet transform, to generate a sparse representation of the original signal. The problems associated to these representations are their high sensitivity to errors and erasures; since the symbols are decorrelated, the knowledge of the correctly received symbols does not bring any information allowing to estimate the lost or corrupted ones.

In Section 2.4.2.1, we have presented an MD approach which uses a scalar quantization producing two quantized symbols instead of one. The MD approach presented in this section is totally different; the descriptions are obtained by the mean of a correlating transform, which introduces some *statistical* redundancy between the independent random variables. This MD approach is referred to as MD with correlating transform (MDCT).

The statistical dependencies between the descriptions are used, at the decoder side, to improve the estimation of the missing coefficients within a description, from the ones received in the other description. MDCT was firstly introduced by Wang, Orchard and Reibman for two variables [162, 164], then generalized by Goyal *et al.* to the multiple variable case [53]. This technique consists in transforming a block of N centered, independent Gaussian random variables into a block of N correlated variables.

Consider the case where $N = 2$ and let X_1 and X_2 be two independent zero-mean Gaussian random variables with variances σ_1^2 and σ_2^2 respectively. These two variables are linearly transformed to get two correlated variables $X^{(1)}$ and $X^{(2)}$

$$\begin{pmatrix} X^{(1)} \\ X^{(2)} \end{pmatrix} = T \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}, \tag{2.7}$$

where T is a linear continuous-valued transform. It was shown in [53] that performing the quantization after T leads to a higher distortion than when quantization and transform are performed in the reverse order. This is due to the fact that T, which intentionally introduces some redundancy between $X^{(1)}$ and $X^{(2)}$, is usually a non-orthogonal transform, and thus the quantization cells obtained after applying T are not square, which is suboptimal. The idea is then to perform first the quantization then apply a discrete-valued transform $\widetilde{T}$ to the obtained quantization indexes. The discrete-valued transform $\widetilde{T}$ is constructed from the linear transform T which is assumed to be invertible of determinant 1. First, T is factorized into a product of triangular matrices then $\widetilde{T}$ is computed by intermediate rounding of these triangular matrix factors. It is then shown that $\widetilde{T}$ is invertible. When both $X^{(1)}$ and $X^{(2)}$ are received, the decoder uses the inverse of $\widetilde{T}$ to obtain $\left(\widehat{X}_1, \widehat{X}_2\right)^T$. The distortion D_0 is then equal to the quantization error. When only one of the descriptions is received, for example $X^{(1)}$, the original vector $(X_1, X_2)^T$ is estimated using the least square estimator: $\left(\widehat{X}_1, \widehat{X}_2\right)^T = E\left[(X_1, X_2)^T | X^{(1)}\right]$. Neglecting the quantization, a linear estimator of $(X_1, X_2)^T$ is obtained as

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = T^{-1} \begin{pmatrix} X^{(1)} \\ \widehat{X}^{(2)} \end{pmatrix}, \tag{2.8}$$

where $\widehat{X}^{(2)} = E\left[X^{(2)} | X^{(1)}\right]$.

For an equal probability of failure on the two channels and assuming that $\sigma_1 \geqslant \sigma_2$, Goyal has shown in [53] that the optimal continuous-valued transform T has the form

$$T = \begin{pmatrix} \alpha & (2\alpha)^{-1} \\ -\alpha & (2\alpha)^{-1} \end{pmatrix}, \tag{2.9}$$

where α varies between $\sqrt{\frac{1}{2}\frac{\sigma_2}{\sigma_1}}$ and $+\infty$. The transforms defined in (2.9) lead to equal rates and distortions $R_1 = R_2$ and $D_1 = D_2$. The parameter α allows to control the

amount of redundancy

$$\rho = \frac{1}{2}\log\frac{\sigma_1^2\alpha^2 + \sigma_2^2/(4\alpha^2)}{\sigma_1\sigma_2} \tag{2.10}$$

introduced between $X^{(1)}$ and $X^{(2)}$, which in turn controls the side distortions D_i, $i = 1, 2$.

2.4.2.3. MD through Frame Expansion

The MDCT exposed in Section 2.4.2.2 introduces a statistical correlation between the descriptions which allows to estimate the missing transform coefficients from the received ones. A similar approach which introduces a deterministic correlation into the transmitted signal was also proposed by Goyal *et al.* [55, 56, 79]. In this case, the input signal $\mathbf{X}$ is expanded via a frame decomposition

$$\mathbf{Y} = \mathcal{F}\mathbf{X}, \tag{2.11}$$

where $\mathcal{F}$ is called a frame operator. The theory of frames was first introduced by Duffin and Schaeffer [41] in 1952 in the context of non-harmonic Fourier series. In the sequel a brief recall on frames is given since the multiple description framework considered in this thesis is built on this type of redundant decompositions. A careful mathematical review of frames is found in [98].

Frames Consider a Hilbert space $\mathbb{H}$ endowed with the inner product $\langle\cdot,\cdot\rangle$ and the norm $\|\cdot\|$. Given a countable index set Γ, a family of vectors $f_k \in \mathbb{H}$, $k \in \Gamma$ is a frame of $\mathbb{H}$ if there are two constants $0 < A < B$ such that for all $f \in \mathbb{H}$

$$A\|f\|^2 \leqslant \sum_{k\in\Gamma} |\langle f, f_k\rangle|^2 \leqslant B\|f\|^2. \tag{2.12}$$

The constants A and B are called the *frame bounds*. If $A = B$, the frame is called *tight*. Given a frame f_k, $k \in \Gamma$ of $\mathbb{H}$, the frame operator $\mathcal{F}$ mapping the Hilbert space $\mathbb{H}$ into the space $\ell^2(\mathbb{Z})$

$$(\mathcal{F}f)_k = \langle f, f_k\rangle, \tag{2.13}$$

admits a pseudo inverse defined as

$$\mathcal{F}^{\#} = (\mathcal{F}^*\mathcal{F})^{-1}\mathcal{F}^*, \tag{2.14}$$

where $\mathcal{F}^*$ is the adjoint operator of $\mathcal{F}$. The vector f can then be recovered in a stable manner from the projections $\langle f, f_k\rangle$, $k \in \Gamma$ as follows

$$f = \sum_{k\in\Gamma} \langle f, f_k\rangle f_k^{\#}, \tag{2.15}$$

where the family of vectors $f_k^{\#}$ defined as

$$f_k^{\#} f = \left(\mathcal{F}^{\#}\mathcal{F}\right)^{-1} f_k \tag{2.16}$$

represents the *dual* frame verifying

$$\frac{1}{B}\left\|f\right\|^2 \leqslant \sum_{k\in\Gamma} \left|\left\langle f, f_k^{\#}\right\rangle\right|^2 \leqslant \frac{1}{A}\left\|f\right\|^2. \tag{2.17}$$

MD via Frame Expansion Multiple descriptions by frame expansion consists in projecting the source on a redundant frame. Each of the resulting coefficient sets forms a description, which is further quantized [56]. The frame approach is similar to error-correcting codes in the sense that both techniques transform a group of N symbols into a group of $N' > N$ symbols by introducing a certain amount of *structured* redundancy, which increases the robustness to channel impairments [54]. The difference between these two schemes is that the redundancy is introduced prior to quantization for the frame approach, whereas in the case of channel coding the redundancy introduction occurs after quantization. The main advantage of the frame approach is the potential quantization noise reduction [57]. In [56] a numerical comparison between the frame approach and traditional channel coding is provided for a Gaussian source. When there are no packet losses, the redundancy introduced by the channel code is useless, whereas the one introduced by the frame approach allows to reduce the quantization noise.

MD by Filter Banks An example of frames of $\ell_2(\mathbb{Z})$ are filter banks. Already popular for audio and image coding applications, filter banks have received a lot of interest since their link to wavelet transforms has been highlighted.

Filter banks frame expansions have been studied to achieve resilience to erasures in [166, 56, 116, 79]. The first application of filter banks to MD was proposed by Yang and Ramchandran in [166], where the descriptions are generated using orthonormal analysis filters. The filtered signals obtained at the analysis stage are decimated by a factor of two, quantized and entropy coded, then transmitted over separate channels. At the reconstruction, the associated synthesis filters are used. The lost symbols within a description are estimated from the received ones by using a linear prediction. The problem of designing optimal filter banks for MD has also been addressed by Dragotti *et al.* in [40]. The difference between the two approaches is given by the place of the quantizer in the transmission chain. The advantage of the approach in [40] is that the quantization cells do not change shape and the quantization error is not increased by the use of non-orthogonal transforms. In [56, 79], the correspondence between oversampled filter banks and frames is shown. For example, oversampled block transforms, like the Discrete Fourier Transforms (DFT) [104] codes, are actually a special class of frames, as shown in [130]. These codes are regarded as JSC block codes which enhance robustness to erasures [79, 130, 102] and may be used for MD coding.

2.4.3. MD for Video Coding and Transmission

The main motivation behind the use of MD video coding in packet-loss networks comes from its ability to provide a minimal quality for video reconstruction without requiring the retransmission of lost packets. Indeed, unlike layered video coding, the descriptions are of an equivalent importance and each received description provides a primary reconstruction quality, even if all the other descriptions are lost. This property makes MD coding particularly interesting for real-time and broadcast applications, where the retransmission of lost data is usually not possible. One may distinguish the MD video coding schemes designed for hybrid video coders, as MPEG and H.264/AVC, from the ones proposed for $t + 2$D wavelet coders. A thorough survey on MD coding for hybrid video coders can be found in [163] and a more recent overview on the MD techniques for wavelet based video coders is in [48].

In [148] a two-description coding scheme is proposed for scalable video coders. The correlation between the descriptions is introduced in the temporal domain by the mean of an oversampled motion-compensated temporal filter. Each description is packetized prior to transmission over a packet-loss channel. Separate packets are created for the motion vectors and for the spatio-temporal subbands. If packets containing motion vectors are lost, or if the packets containing the spatial approximation subband of the temporal approximation subband are lost in one of the descriptions, the entire video sequence is considered as lost and the side decoder associated to the other description is used to reconstruct the input frames. Simulations have shown that, in presence of packet losses, this scheme outperforms both the non-redundant single description (SD) one and the MD-based on temporal splitting of the video sequence into odd and even frames.

Recent works have focused on comparing MD coding schemes to single description coding schemes followed by channel codes, for video transmission over packet-loss networks [144, 13, 143, 4]. In [144], a comparative analysis based on theoretical R-D performance of both techniques is presented. This analysis explicitly takes into account the loss probability introduced by the network as well as the redundancy introduced by the MD coder. In matched channel conditions, the use of an optimized single description scheme combined with a channel code always outperforms MD coding, as predicted by Shannon's separation principle [141]. In [13], a comparison between MD and single descriptions video coding schemes is performed for H.264/AVC coded bit streams. Descriptions are generated via temporal splitting of odd and even frames within the video sequence. The lossy network is described by a two-state Markov model. For similar amount of redundancy and equivalent reconstruction PSNR in the loss-free case, MD coding, which is robust to long burst of erasures, performs better than single description protected with a channel code. In [143], an MDSQ coding scheme is compared to a single description scheme for a scalable wavelet video coder. Unlike channel codes, the redundancy level introduced by the MD coding scheme does not impact a lot the decoding performance. In

[13, 143], it is shown that MD coding performs better at high losses rates. Moreover, hypothetical schemes able to adapt the redundancy level to the receiver channel conditions have been considered. Adaptive channel coding of single description turned out to outperform the adaptive MD, as recently shown in [144].

In this thesis, the considered MD coding scheme is based on the redundant temporal wavelet decomposition proposed in [148]. The obtained temporal descriptions are transmitted over an heterogeneous network formed by a packet-loss wired part, followed by a noisy wireless channel. Most of the above-mentioned works assume that packets are either dropped or arrive error-free at the MD decoder. In this work, *joint protocol-channel* decoding mechanisms [100] are assumed to be implemented to allow erroneous packets to reach the application layer. At the MD decoder side, JSC decoding is used to recover, if possible, the corrupted packets. This approach has already been considered for theoretical MD coding schemes in [64, 89] and in [1] for a $t + 2D$ video coder, where MD is performed by duplicating the wavelet coefficients to form the two descriptions, which are then quantized with different quantizers designed to ensure balanced descriptions [121]. The number of errors due to transmission impairments is significantly reduced thanks to JSC decoding, which is based on the use of the redundancy left by the source encoder. Such JSCD techniques are presented in the next section.

2.5. JSCD Techniques

The JSCD techniques usually rely on the observation that all *practical* source coders produce a (packetized) bit stream containing some redundancy [137]. The source residual redundancy is interpreted as the rate excess left by practical source encoders inside the compressed data. This redundancy can result *e.g.*, from the residual correlation left between the samples after temporal and spatial decorrelation transforms, from the non-uniform distributions of the entropy-coded source symbols or from the structural properties of the bit stream imposed by the coding standard. In addition to this *natural* residual redundancy, one may deliberately insert some *artificial* redundancy into the compressed bit stream, via error resilient source coding tools such as reversible variable length codes, which enhances robustness while keeping efficient compression rates. The JSC decoding techniques based on source redundancy (either natural or artificial) exploit this rate excess using robust channel decoding techniques [156].

The large majority of image and video coding standards adopt variable-length codes (VLCs) as entropy codes due to their high compression ratio and reduced complexity. As a consequence, JSCD techniques based on source residual redundancy have been proposed for VLC codes. A comprehensive survey of these techniques is proposed in [62, 22]. One may distinguish the JSCD techniques exploiting the *natural* residual redundancy present in VLCs [76, 122] from the ones exploiting some artificial redundancy inserted into the coded bit stream.

2.5.1. Soft Decoding of Variable Length Codes

The soft decoding of VLCs is based on estimating the transmitted sequence at the source coder from its noisy measurements. The residual redundancy present in the bit stream is used to reduce the search space. The estimation is then achieved by limiting the search to the sequences satisfying some *constraints* deduced from this redundancy. In [22, 42] are identified three main categories for residual redundancy to be exploited while decoding VLCs:

1. *Redundancy due to the VLC syntax:* This type of redundancy may be used when the considered VLC is incomplete, *i.e.,* the Kraft inequality is strict [138, 168]. In this case, some sequences can not be generated from the entropy coder and are not examined during the estimation process. In [138], authors used a first-order Markov model for the source and incorporated the residual redundancy into the decoding process by considering a symbol-constrained directed graph with the list Viterbi algorithm [5] to decode the VLCs.

2. *Redundancy due to the source coder semantics:* This redundancy comes from the constraints on the bit stream imposed by the source coder when coding the texture information [149, 136, 111, 109, 113]. In [109], Nguyen and Duhamel demonstrated that a significant residual source redundancy is available in the VLC data and may be exploited if appropriate information on the structure and the properties of the compressed image and video streams are taken into account. They proposed a tool to quantify this redundancy as an equivalent of the redundancy introduced by channel coding. Later, in [111], they presented a method to decode the DCT coefficients by taking into account the source constraints, *e.g.* the maximum number of DCT coefficients in a block - *the run constraint* - and the fact that only the last VLC codeword within a block should be labeled as *last* - *the last constraint*.

3. *Redundancy due to the packetization of VLC data:* Prior to transmission over a packet-switched network, the compressed data is packetized. Several information may be deduced and exploited in each received packet to detect and correct transmission errors [11, 91, 136]. For the H.264 video coder, the packetization process is indicated by the standard and each bit stream is efficiently organized into a set of independently decodable network abstraction layer units (NALUs). Robust decoding of each NALU may be performed by using the information extracted from its header, see [136]. In [91], a robust decoding algorithm is considered to process the packets delivered by a H.263+ coder. Firstly the texture blocks, which correspond to variable length-coded DCT coefficients, are localized within each packet. Then, each localized block is decoded by exploiting the constraints imposed by the H.263 coder, *e.g.* the number of DCT coefficients within a block.

These various redundancies may be combined altogether to significantly improve the decoding performance [112, 91, 109, 111, 136]. Recently, in [112], the authors

proposed a new VLC decoder exploiting jointly the VLC syntax and the source semantic constraints, and delivering both maximum likelihood (ML) optimum hard-output and soft-output solutions. Furthermore, the residual redundancy of VLCs may be used jointly to the redundancy introduced by channel codes at physical layers to perform iterative decoding as in [147, 7, 110, 99, 112] which improves the decoding performance. In [99], the authors presented a sequential decoding algorithm which makes a simultaneous usage of the redundancy provided by the CRC at the media acess control (MAC) layer and the inherent source redundancy (syntax and semantic constraints) to achieve robust video decoding in downlink situations.

In this thesis, the redundancy which we exploit comes from the information provided during the packetization of the video bit stream. This information, is necessary to decode the received packets. We will show how this information can also be exploited to detect and correct some of the transmission errors.

2.5.2. Use of Artificial Redundancy

The JSCD techniques presented in Section 2.5.1 rely on the use of the inherent source redundancy. They present the advantage of increasing the robustness of the transmitted data without any modification of the standard and without any increase in the total bit rate. In addition to this inherent source redundancy, some *artificial* redundancy may be introduced at the source coder to further enhance the error resiliency and increase the decoding performance. For example, many works have proposed to design arithmetic coders with error detection and correction capabilities [60, 27, 59, 123, 63]. These approaches are based on the insertion of a forbidden symbol [27] or synchronization markers [63] to enable error detection at the receiver. The correction of the detected errors is then based on sequential decoding [3, 58, 61] using maximum a posteriori (MAP) or ML estimation or ARQ techniques [27, 59, 123]. In [27], a robust JSCD scheme using jointly forbidden symbol and ARQ is proposed. This scheme allows continuous error detection and correction unlike classical ARQ-based techniques which require the reception of the whole packet to enable error detection and the retransmission of its whole content. In [123], Pettijohn *et al.* used both depth-first and breadth-first sequential decoding algorithms, to detect transmission errors. Binary signaling with null-zone soft decoding is employed by testing the presence of an FS in the decoded bit stream.

All the error-resilient coding techniques presented so far require a total or partial modification of the source or/and channel coders and/or decoders. The techniques that will be presented in the next section do not modify the blocks of the communication systems, but conceal the effects of the transmission impairments by appropriately exploiting the received data.

2.6. Error Concealment

Error concealment (EC) is used as a last resort after an error or packet loss has been detected at the decoder side. The effects of lost or corrupted data are *concealed* to limit the signal quality degradation. Good surveys on EC techniques is proposed in [164, 174].

Video error concealment techniques can be devided into two main categories: temporal error concealment and spatial error concealment.

Temporal error concealment (TEC) exploits the inter-frame correlation in video sequences in order to replace the damaged macroblock of the current frame by the spatially corresponding macroblock in the previously decoded frame. Motion compensation is usually integrated into the TEC process to fit the dynamic character of the video sequence and to limit the visual artifacts, especially in the presence of high motion. When the motion vectors are corrupted or lost, they are usually estimated from the motion vectors of adjacent macroblocks [173, 88]. Spatial error concealment (SEC) is performed in the spatial domain by interpolating the pixels of the damaged part of the video frame from the neighboring pixels [172, 78, 51]. Adaptive SEC approaches are further considered to choose for each damaged macroblock the suitable SEC method to be used, which allows to exploit the advantages of the different SEC techniques [160, 161, 135, 9]. Further, hybrid error concealment schemes combining TEC and SEC techniques have been considered [82, 66].

2.7. Conclusion

In this chapter we presented several approaches proposed in the literature for increasing the robustness of video bit streams transmitted over unreliable networks. In particular, we have focused on the class of joint source-channel coding and decoding techniques since the contributions of this thesis fit into this category. We briefly presented the existing approaches in this area and distinguished the ones acting at the coder side, namely the joint source-channel coding techniques, from those implemented at the decoder side, the joint source-channel decoding techniques. This presentation is not exhaustive, our first goal being to give a general panorama of the communication systems designed in a joint-source channel perspective and to highlight the motivation behind the use of such systems in practical situations.

In the next chapters, we will present some of our contributions in this domain, namely a joint source-channel decoding scheme developed to enhance the robustness of bit streams obtained by subband video coders.

3. JSC Decoding of $t + 2D$ Video Streams

In practical situations with stringent delay and complexity constraints, communication systems based on JSC coding and decoding are an efficient alternative to the classical communication schemes, which are only optimal under idealized conditions. In particular, the JSC decoding (JSCD) techniques are attractive since they rely on the use of the redundancy left by the source encoder and do not introduce any increase in the total bit rate (in the case when no artificial redundancy is inserted). Moreover, these schemes are implemented at the decoder side and usually introduce a manageable complexity.

Current video compression coders and, more particularly, the wavelet-based ones are very efficient, leaving a very small amount of redundancy and making thus the compressed bit streams extremely sensitive to transmission errors, fact that motivates even more the use of the residual redundancy in order to detect and correct the transmission errors.

This chapter presents an efficient JSCD scheme suited for subband video coders and is organized as follows: first, the conceptual JSC decoding scheme is presented in Section 3.1. Then, the considered wavelet video coder is briefly presented in Section 3.2. The aim is to describe the bit streams delivered by this coder and to identify the redundancy that may be exploited at the decoder side in order to perform JSCD. The proposed JSC decoding scheme is described in the case of a bit stream formed by one layer in Section 3.3, then it is extended to the multi-layer case in Section 3.4.

3.1. Conceptual JSC Video Decoder

The considered JSC video decoding scheme is presented in Figure 3.1. Assume that the input video sequence is formed by N_{f} frames. These frames are gathered into one vector $\mathbf{x}$, which is assumed to be the realization of some random vector $\mathbf{X}$. The input vector $\mathbf{x}$ is encoded and a vector

$$\mathbf{z} = \mathbf{F}(\mathbf{x}) \tag{3.1}$$

is generated, where $\mathbf{F}(\cdot)$ denotes the video coding function including the temporal and spatial decorrelation steps, as well as the quantization and entropy coding. The content of $\mathbf{z}$ is transmitted through some noisy channel. Here, the term channel refers to the part of the transmission scheme that is situated between the output of the coding function $\mathbf{F}(\cdot)$ and the input of the source decoder. This channel includes the modulation, the packetization and the physical channel parts, as well as the corresponding demodulation and depacketization operations. At the output of this channel, the channel state information and the decoder reliability information, described in Chapter 2, are obtained.

The aim of the JSC decoder is to evaluate an estimate $\widehat{\mathbf{x}}$ of the original input vector $\mathbf{x}$ by using the information obtained at the channel's output, as well as the information on the redundancy left by the video coder, which is fed back to the estimator, as illustrated in Figure 3.1.

The channel model is introduced in Section 3.1.1, then the maximum *a posteriori* (MAP) estimation problem is formulated in Section 3.1.2.

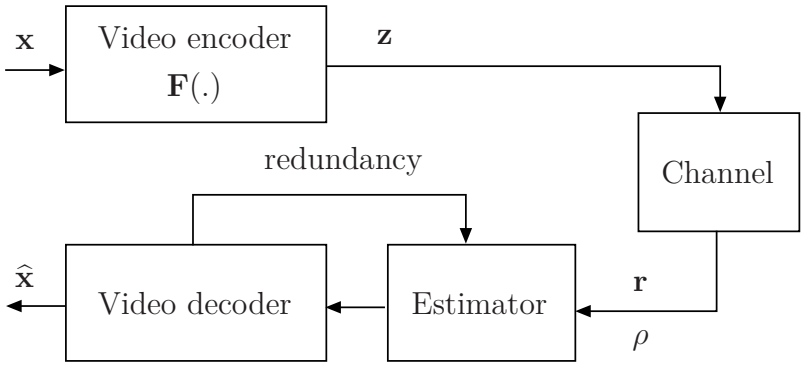


Figure 3.1.: Joint source-channel decoding transmission scheme.

3.1.1. Channel Model

Usually, the transmission of multimedia contents requires the use of some packetization process, *e.g.*, RTP (Real Time Protocol [83]), to ensure the jitter compensation and a correct playback of the data packets at the receiver. Assume that $\mathbf{z}$ is packetized into N_p packets of similar length

$$\mathbf{z} = \left((\mathbf{z}_1)^T, \ldots, \left(\mathbf{z}_{N_\mathrm{p}}\right)^T \right)^T, \tag{3.2}$$

where $\mathbf{z}_k$, $k = 1, \ldots, N_\mathrm{p}$ denotes the content of the $k-$th packet. Classical error detection mechanisms (CRCs or checksums) at lower protocol layers do not allow corrupted packets to reach the upper application layer. Implementing JSCD techniques at the APL layer needs the use of permeable protocol layers at the receiver

side [75, 42, 101]. Such mechanisms require robust header decoding techniques [101] and transmission of the bit soft information or reliability measures (coming from the channel decoders at physical layer) to the upper protocol layers, as detailed in [117]. Reliable recovery of the various headers involved in the protocol stack may be ensured by employing *joint protocol-channel* decoding techniques at the receiver side [101, 42]. Provided that such mechanisms are implemented, the whole network packetization process, the physical channel, the robust depacketization, as well as the channel decoding operation, are modeled by a channel representing everything between the source coder and the source decoder. The channel state information and the decoder reliability information may then be obtained at the output of this channel.

Let the reliability information be modeled by the vector

$$\mathbf{r} = \left((\mathbf{r}_1)^T, \ldots, \left(\mathbf{r}_{N_\mathrm{p}}\right)^T \right)^T, \tag{3.3}$$

where the vector $\mathbf{r}_k$, $k = 1, \ldots, N_\mathrm{p}$ contains soft information on the transmitted packet $\mathbf{z}_k$. This soft information consists, *e.g.*, in likelihood ratios, or *a posteriori* probabilities on the transmitted data. They are provided by the channel decoders at the physical layer on receiver side. Moreover, let the channel state information be modeled by the vector

$$\boldsymbol{\rho} = \left(\rho_1 \ldots, \rho_{N_\mathrm{p}} \right)^T, \tag{3.4}$$

whose components indicate whether a packet $\mathbf{z}_k$, $k = 1, \ldots, N_\mathrm{p}$ has been received ($\rho_k \geqslant 1$) or not ($\rho_k = 0$). The value $\rho_k = 0$ may be deduced, *e.g.*, from the content of successive RTP packet headers. The values of $\rho_k \geqslant 1$ indicate the noise/fade level and are assumed to be known by the receiver.

Along this chapter, all the packets are assumed to be received ($\rho_k \geqslant 1$, $k = 1, \ldots, N_\mathrm{p}$), but some of them may be corrupted by the transmission noise.

3.1.2. Optimal Estimation Schemes

This section presents two optimal estimation schemes, which are more or less independent of the considered video coder. First, the estimation of $\mathbf{x}$ is formulated in Section 3.1.2.1. Then, a constrained estimation of $\mathbf{z}$, which is more practical to handle, and for which it is easier to exploit the channel information and the source residual redundancy, is described in Section 3.1.2.2.

3.1.2.1. MAP Estimation of $\mathbf{x}$

The MAP estimate $\widehat{\mathbf{x}}_\mathrm{MAP}$ of $\mathbf{x}$ knowing $\boldsymbol{\rho}$ and $\mathbf{r}$ is

$$\widehat{\mathbf{x}}_\mathrm{MAP} = \arg\max_{\mathbf{x}} p\left(\mathbf{x}|\boldsymbol{\rho}, \mathbf{r}\right). \tag{3.5}$$

Marginalizing (3.5) with respect to $\mathbf{z}$, one gets

$$\begin{aligned}\widehat{\mathbf{x}}_{\text{MAP}} &= \arg\max_{\mathbf{x}} \sum_{\mathbf{z}} p\left(\mathbf{x},\mathbf{z}|\boldsymbol{\rho},\mathbf{r}\right) \\ &= \arg\max_{\mathbf{x}} \sum_{\mathbf{z}} p\left(\boldsymbol{\rho},\mathbf{r}|\mathbf{x},\mathbf{z}\right) p\left(\mathbf{z}|\mathbf{x}\right) p\left(\mathbf{x}\right). \end{aligned} \tag{3.6}$$

Since $\mathbf{z}$ is deduced from $\mathbf{x}$, in (3.6) one has

$$p\left(\mathbf{z}|\mathbf{x}\right) = \begin{cases} 1 & \text{if } \mathbf{z}{=}\mathbf{F}\left(\mathbf{x}\right) \\ 0 & \text{otherwise.} \end{cases}$$

Consequently, (3.6) becomes

$$\widehat{\mathbf{x}}_{\text{MAP}} = \arg\max_{\mathbf{x}} p\left(\boldsymbol{\rho},\mathbf{r}|\mathbf{F}\left(\mathbf{x}\right)\right) p\left(\mathbf{x}\right), \tag{3.7}$$

where the fact that the knowledge of $\mathbf{x}$ does not bring any additional information on $\boldsymbol{\rho}$ and $\mathbf{r}$ once $\mathbf{F}\left(\mathbf{x}\right)$ is known has been taken into account.

Even if $p\left(\boldsymbol{\rho},\mathbf{r}|\mathbf{F}\left(\mathbf{x}\right)\right)$ in (3.7) is quite easy to evaluate for a given $\mathbf{x}$, the evaluation of $\widehat{\mathbf{x}}_{\text{MAP}}$ requires the maximization of a discontinuous function (due to the quantization present in $\mathbf{F}(\cdot)$), over a prohibitively large number of variables (*i.e.*, the components of $\mathbf{x}$, which are all the pixels within the considered video sequence frames).

3.1.2.2. Constrained Estimation of the Entropy Coded Bit Stream z

Alternatively, one may try to estimate $\mathbf{z}$ in a first time, accounting for the fact that this bit stream has been generated from a video sequence $\mathbf{x}$. The resulting *constrained* MAP estimator of $\mathbf{z}$, knowing $\boldsymbol{\rho}$ and $\mathbf{r}$, is

$$\widehat{\mathbf{z}}_{\text{MAP}} = \arg\max_{\mathbf{z}\in\mathcal{S}} p\left(\mathbf{z}|\boldsymbol{\rho},\mathbf{r}\right), \tag{3.8}$$

where

$$\mathcal{S} = \left\{\mathbf{z} \text{ st } \exists\mathbf{x},\ \mathbf{z} = \mathbf{F}\left(\mathbf{x}\right)\right\} \tag{3.9}$$

is the set of all the bit streams that can be generated from all the vectors $\mathbf{x}$. We point out here that performing the estimation among the elements of $\mathcal{S}$ takes into account the information fed back from the source coder to the estimator, and implies that $\widehat{\mathbf{z}}_{\text{MAP}}$ is a valid sequence, since it could have been generated by the source encoder.

Using Bayes' rule, (3.8) becomes

$$\widehat{\mathbf{z}}_{\text{MAP}} = \arg\max_{\mathbf{z}\in\mathcal{S}} p\left(\boldsymbol{\rho},\mathbf{r}|\mathbf{z}\right) p\left(\mathbf{z}\right). \tag{3.10}$$

In (3.10) one has

$$p\left(\boldsymbol{\rho},\mathbf{r}|\mathbf{z}\right) = p\left(\mathbf{r}|\boldsymbol{\rho},\mathbf{z}\right) p\left(\boldsymbol{\rho}|\mathbf{z}\right). \tag{3.11}$$

Assuming that the congestion conditions for each packet of $\mathbf{z}$ are independent, one gets

$$p(\boldsymbol{\rho}|\mathbf{z}) = \prod_{k=1}^{N_\mathrm{p}} p(\rho_k|\mathbf{z}_k). \tag{3.12}$$

Usually $p(\rho_k|\mathbf{z}_k)$, $k = 1, \ldots, N_\mathrm{p}$ only depends on the length of the packet $\mathbf{z}_k$ and not on its content. Assuming now that each packet $\mathbf{z}_k$ of $\mathbf{z}$ encounters independent transmission channel realizations, one gets

$$p(\mathbf{r}|\boldsymbol{\rho}, \mathbf{z}) = \prod_{k=1}^{N_\mathrm{p}} p(\mathbf{r}_k|\rho_k, \mathbf{z}_k). \tag{3.13}$$

Since we assumed that $\rho_k \geqslant 1$, the evaluation of $p(\mathbf{r}_k|\rho_k, \mathbf{z}_k)$ depends on the values of $\mathbf{z}_k$, ρ_k and on the channel model. Combining (3.10)–(3.13) one gets

$$\widehat{\mathbf{z}}_\mathrm{MAP} = \arg\max_{\mathbf{z}\in\mathcal{S}} p(\mathbf{z}) \prod_{k=1}^{N_\mathrm{p}} p(\mathbf{r}_k|\rho_k, \mathbf{z}_k)\, p(\rho_k|\mathbf{z}_k). \tag{3.14}$$

In (3.14), determining whether $\mathbf{z} \in \mathcal{S}$ depends on the function $\mathbf{F}(.)$ used to perform the video encoding.

In this work, the subband video coder Vidwav [126] has been considered. In the next section, a brief overview of this coder is presented, as well as a description of the bit stream it delivers. This will help to identify a part of the residual redundancy which will be used to build the set $\mathcal{S}$.

3.2. Overview of Vidwav Video Coder

Vidwav is a 3D scalable wavelet-based video coder using motion-compensated temporal filtering (MCTF) based on the Barbell filter [126]. It allows both spatial and in-band MCTF ($t + 2\mathrm{D}$ or $2\mathrm{D} + t + 2\mathrm{D}$). In this work we consider only the $t + 2\mathrm{D}$ scheme, illustrated in Figure 3.2. First, the MCTF is performed on the frames of the input video sequence $\mathbf{x}$, then the spatial wavelet transform is applied on the resulting temporal subbands. The obtained spatio-temporal subbands are then divided into 3D blocks which are entropy encoded in an independent manner.

Consider an input video sequence $\mathbf{x}$ of N_f frames $X_1, \ldots, X_{N_\mathrm{f}}$, represented by the vectors $\mathbf{x}_n$, $n = 1, \ldots, N_\mathrm{f}$ of size $N_\mathrm{r}N_\mathrm{c}$, where N_r and N_c denote the number of rows and columns in each frame X_n. Then

$$\mathbf{x} = \left((\mathbf{x}_1)^T, \ldots (\mathbf{x}_n)^T, \ldots, (\mathbf{x}_{N_\mathrm{f}})^T\right)^T. \tag{3.15}$$

This input sequence is temporally decorrelated by performing MCTF with J decomposition levels to get N_f temporal subband frames $\mathbf{y}_n$ of size $N_\mathrm{r}N_\mathrm{c}$, and which are gathered in the vector $\mathbf{y}$

$$\mathbf{y} = T_J T_{J-1} \dots T_1 \mathbf{x} = \left((\mathbf{y}_1)^T, \dots (\mathbf{y}_n)^T, \dots, (\mathbf{y}_{N_\mathrm{f}})^T \right)^T, \tag{3.16}$$

where T_j, $j = 1 \dots, J$ is an $N \times N$ matrix, with $N = N_\mathrm{f}N_\mathrm{r}N_\mathrm{c}$, representing the MCTF at level j, $j = 1, \dots, J$. The vectors $\mathbf{y}_n$, $n = 1, \dots, N_\mathrm{f}$ are further spatially transformed

$$\mathbf{s}_n = S_1 \mathbf{y}_n, \tag{3.17}$$

where S_1 is the $N_\mathrm{r}N_\mathrm{c} \times N_\mathrm{r}N_\mathrm{c}$ spatial transform matrix. The vector of spatio-temporal subbands is then obtained as follows

$$\mathbf{s} = S_{N_\mathrm{f}} \mathbf{y} = \left((\mathbf{s}_1)^T, \dots, (\mathbf{s}_{N_\mathrm{f}})^T \right)^T, \tag{3.18}$$

where S_{N_f} is a $N \times N$ block diagonal matrix defined as

$$S_{N_\mathrm{f}} = \begin{pmatrix} S_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & S_1 \end{pmatrix} \tag{3.19}$$

and representing the spatial transform operation of the entire vector of temporal subbands $\mathbf{y}$.

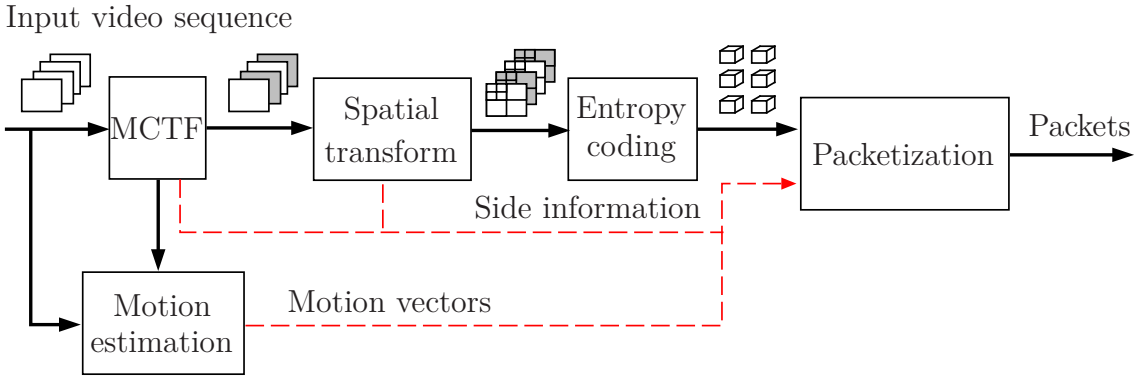


Figure 3.2.: Vidwav encoder modules.

The vector $\mathbf{s}$ defined in (3.18) is further divided into several 3D blocks. Each 3D block is obtained by gathering several temporally-successive 2D blocks belonging to the same spatio-temporal subband of $\mathbf{s}$. An example of a 3D block formation is illustrated in Figure 3.3 for a given temporal subband, where the number of spatial

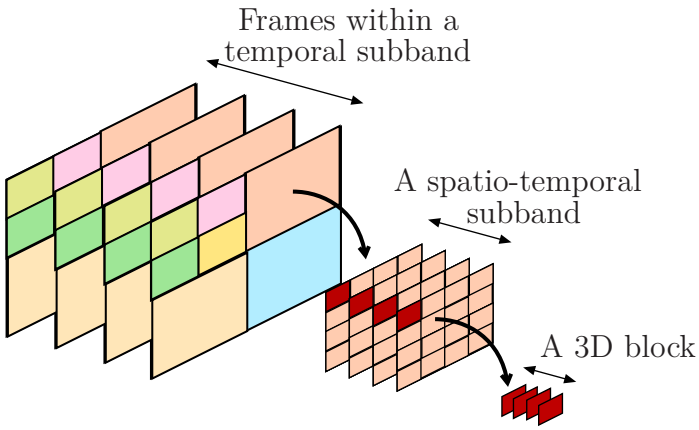


Figure 3.3.: 3D block formation process within a spatio-temporal subband.

decompositions is equal to 2. Let N_b be the total number of obtained 3D blocks. The vector of spatio-temporal subbands $\mathbf{s}$ is then reorganized as follows

$$\mathbf{s} = \left((\mathbf{b}_1)^T, \ldots, (\mathbf{b}_k)^T, \ldots, (\mathbf{b}_{N_\mathrm{b}})^T \right)^T, \tag{3.20}$$

where the vectors $\mathbf{b}_k$, $k = 1, \ldots, N_\mathrm{b}$ represent the content of the N_b 3D blocks.

The aim of the temporal and spatial transforms presented so far is to decorrelate the input video signal $\mathbf{x}$. A very small amount of redundancy is left in the resulting vector of spatio-temporal subbands $\mathbf{s}$. The 3D blocks of $\mathbf{s}$ are further entropy encoded.

In the next section, the entropy coding operation is presented and the information obtained after this step is described. It is important to detail such information, since a part of it will be considered as a residual redundancy, which may be exploited in order to perform JSC decoding.

3.2.1. Entropy Coding

The obtained 3D blocks $\mathbf{b}_k$, $k = 1, \ldots, N_\mathrm{b}$ are independently entropy coded using the 3D Embedded Subband Coding with Optimal Truncation (3D-ESCOT) algorithm [71]. For each 3D block, bitplane coding and context-based arithmetic coding are used. The coding algorithm consists of three basic, *primitive* operations:

1. *Zero Coding*: indicates whether a non-significant sample becomes significant in the current bitplane. The coding is then performed using the context given by the sample's neighbors.
2. *Sign Coding*: when a sample becomes significant its sign is encoded using an adaptive context-based arithmetic coder.
3. *Magnitude Refinement*: it is used to code the new information of an already significant sample. This operation has three contexts depending on whether

the magnitude refinement has been used or not for the current sample and on the significance state of its neighbors.

For each bitplane, the 3D-ESCOT algorithm uses, in turn, three distinct coding passes:

1. *Significance propagation pass*: this pass encodes the neighbors of samples already found significant (either at the current bitplane or at previously coded bitplanes). For each processed sample, the zero coding primitive is applied. Moreover, if the sample becomes significant at the current bitplane, then the sign coding primitive is used to code its sign.
2. *Magnitude refinement pass*: in this pass, only the significant samples are considered and the symbols of these samples in the current bitplane are coded using the magnitude refinement primitive.
3. *Cleanup pass*: in this pass, the samples that are not yet coded in the previous two passes are coded using the zero coding primitive.

Finally, rate distortion (R-D) optimization is used to decide how many bytes are allocated for each coding block. After all these operations, the following information is obtained for each 3D block $\mathbf{b}_k$:

1. the number of bitplanes d_k within $\mathbf{b}_k$ and the number of coding passes[1] $n_k = 3d_k - 2$.
2. the coded bit stream $\mathbf{p}_k$ of $\mathbf{b}_k$, which is an embedded bit stream formed by n_k segments, each one corresponding to the output bit stream of one coding pass.
3. the length in bytes of each bit stream segment and its spatio-temporal subband.
4. the R-D slope information at the end of each coding pass.

Let $\mathbf{E}(.)$ denotes the 3D-ESCOT entropy coding function. The bit stream

$$\mathbf{z}_k = \mathbf{E}(\mathbf{b}_k),\ k = 1, \ldots, N_\mathrm{b} \tag{3.21}$$

is the output associated to the 3D block $\mathbf{b}_k$. This bit stream is formed by the embedded payload $\mathbf{p}_k$ corresponding the entropy-coded spatio-temporal wavelet coefficients of the block $\mathbf{b}_k$, and by a header $\mathbf{h}_k$ containing the motion information associated to $\mathbf{b}_k$ and other metadata necessary for decoding, *e.g.*, the number of coding passes used to code the block, its length in bytes, its spatio-temporal index, its R-D slope information, *etc.* Then

$$\mathbf{z}_k = \begin{pmatrix} \mathbf{h}_k \\ \mathbf{p}_k \end{pmatrix}, \tag{3.22}$$

and the vector

$$\mathbf{z} = \left((\mathbf{z}_1)^T, \ldots, (\mathbf{z}_k)^T, \ldots, (\mathbf{z}_{N_\mathrm{b}})^T \right)^T \tag{3.23}$$

[1]For the first bitplane, the significance propagation pass is not performed.

represents the compressed bit stream delivered by the Vidwav encoder for the input video sequence $\mathbf{x}$. Finally, the video coding function $\mathbf{F}(\cdot)$ defined in (3.1) can be written as follows

$$\mathbf{F}(\mathbf{x}) = \mathbf{E}(S_{N_\mathrm{f}} T_J \dots T_1 \mathbf{x}). \tag{3.24}$$

The Vidwav encoder generally produces a fully embedded and scalable output by generating bit streams formed by several layers, each layer corresponding to a scalabiliy degree (*i.e.,* temporal, spatial and/or SNR or quality scalability).

In Section 3.3, the proposed joint source-channel decoding scheme is described for single layer bit streams and the obtained simulation results are presented. This decoding scheme is further extended to multiple layers bit streams in Section 3.4.

3.3. JSCD of Single Layer Bit Streams

In this section, the entropy-coded bit stream $\mathbf{z}$ is assumed to be formed by one layer. The aim is to provide an implementable counterpart of the optimal estimation schemes presented is Section 3.1.2, which is adapted to the Vidwav coder presented in Section 3.2 and to the structure of the bit stream it delivers. The estimation schemes presented here are based on JSCD by exploiting the redundancy left by the Vidwav encoder.

To simplify the notations, and without loss of generality, each packet of $\mathbf{z}$ defined in (3.2) is assumed to contain one entropy-coded bit stream $\mathbf{z}_k$ of some 3D block $\mathbf{b}_k$. Therefore, $N_\mathrm{p} = N_\mathrm{b}$ and

$$\mathbf{z} = \left((\mathbf{z}_1)^T, \dots, (\mathbf{z}_{N_\mathrm{b}})^T \right)^T. \tag{3.25}$$

The decoder soft information $\mathbf{r}$ and the channel state information ρ, defined in (3.3) and (3.4) are then rewritten as follows

$$\mathbf{r} = \left((\mathbf{r}_1)^T, \dots, (\mathbf{r}_{N_\mathrm{b}})^T \right)^T, \tag{3.26}$$

$$\boldsymbol{\rho} = (\rho_1, \dots, \rho_{N_\mathrm{b}})^T. \tag{3.27}$$

Assume now that all the vectors $\mathbf{z} \in \mathcal{S}$ are *a priori* equally probable, so that (3.14) becomes

$$\widehat{\mathbf{z}}_{\mathrm{MAP}} = \arg \max_{\mathbf{z} \in \mathcal{S}} \prod_{k=1}^{N_\mathrm{b}} p\left(\mathbf{r}_k | \rho_k, \mathbf{z}_k\right) p\left(\rho_k | \mathbf{z}_k\right). \tag{3.28}$$

This assumption is reasonable as long as no informative *a priori* probabilities on $\mathbf{x}$ are available, where $\mathbf{x}$ represents the uncoded video sequence. If such *a priori*

knowledge was available, deducing $p(\mathbf{z})$ would be far from being trivial and (3.28) would be suboptimal compared to (3.14).

Consider the sets

$$\mathcal{S}_k = \left\{ \mathbf{z}_k \mid \exists \mathbf{x}, \exists (\widetilde{\mathbf{z}}_1, \ldots, \widetilde{\mathbf{z}}_{k-1}, \widetilde{\mathbf{z}}_{k+1}, \ldots, \widetilde{\mathbf{z}}_{N_\mathrm{b}}), \right.$$
$$\left. \text{st} \quad \left((\widetilde{\mathbf{z}}_1)^T, \ldots, (\widetilde{\mathbf{z}}_{k-1})^T, (\mathbf{z}_k)^T, (\widetilde{\mathbf{z}}_{k+1})^T, \ldots, (\widetilde{\mathbf{z}}_{N_\mathrm{b}})^T\right)^T = \mathbf{E}\left(S_{N_f} T_J \ldots T_1 \mathbf{x}\right) \right\}, \tag{3.29}$$

defined for each $k = 1, \ldots, N_\mathrm{b}$. The set $\mathcal{S}_k$ contains all the possible bit streams that can result from the entropy coding of a 3D block $\mathbf{b}_k$ obtained for a video sequence $\mathbf{x}$ of N_f frames.

Remark 1. Consider the set $\mathcal{S}$ introduced in (3.9) which contains all the possible entropy-coded sequences $\mathbf{z}$ that can be generated from all the video sequences $\mathbf{x}$. One has

$$\mathcal{S} \subset \mathcal{S}_1 \times \cdots \times \mathcal{S}_{N_\mathrm{b}}. \tag{3.30}$$

In general, $\mathcal{S} \neq \mathcal{S}_1 \times \cdots \times \mathcal{S}_{N_\mathrm{b}}$. Consider $(\mathbf{z}_1, \ldots, \mathbf{z}_{N_\mathrm{b}}) \in \mathcal{S}_1 \times \cdots \times \mathcal{S}_{N_\mathrm{b}}$ and let $\mathcal{X}_k(\mathbf{z}_k)$ be the set of the vectors $\mathbf{x}$ such as the $k-$th entropy-coded sequence of $\mathbf{E}\left(S_{N_f} T_J \ldots T_1 \mathbf{x}\right)$ is equal to $\mathbf{z}_k$. One knows that $\mathcal{X}_k(\mathbf{z}_k) \neq \emptyset$, since $\mathbf{z}_k \in \mathcal{S}_k$. If $\bigcap_{k=1}^{N_\mathrm{b}} \mathcal{X}_k(\mathbf{z}_k) \neq \emptyset$ then $(\mathbf{z}_1, \ldots, \mathbf{z}_{N_\mathrm{b}}) \in \mathcal{S}$. Now assume that for all $(\mathbf{z}_1, \ldots, \mathbf{z}_{N_\mathrm{b}}) \in \mathcal{S}_1 \times \cdots \times \mathcal{S}_{N_\mathrm{b}}$, $\bigcap_{k=1}^{N_\mathrm{b}} \mathcal{X}_k(\mathbf{z}_k) \neq \emptyset$, this means that the sequences $\mathbf{z}_1, \ldots, \mathbf{z}_{N_\mathrm{b}}$ are uncorrelated, which is not usually the case since these sequences are obtained from the entropy coding of the same video sequence, and thus present some correlation.

Using (3.28)–(3.30), one may derive a suboptimal estimator for each $\mathbf{z}_k$ as follows

$$\widehat{\mathbf{z}}_k = \arg \max_{\mathbf{z}_k \in \mathcal{S}_k} p(\mathbf{r}_k | \rho_k, \mathbf{z}_k)\, p(\rho_k | \mathbf{z}_k). \tag{3.31}$$

The estimator $\left((\widehat{\mathbf{z}}_1)^T, \ldots, (\widehat{\mathbf{z}}_{N_\mathrm{b}})^T\right)^T$ is a suboptimal solution of (3.28) since

$$\left((\widehat{\mathbf{z}}_k)^T, \ldots, (\widehat{\mathbf{z}}_{N_\mathrm{b}})^T\right)^T$$

is not necessarily in $\mathcal{S}$, see (3.30).

As described in Section 3.2.1, each entropy-coded bit stream $\mathbf{z}_k$ is formed by a header $\mathbf{h}_k$ and a payload $\mathbf{p}_k$. After the transmission of $\mathbf{z}_k$, the received vector

$$\mathbf{r}_k = \left((\mathbf{r}_{k,\mathrm{h}})^T, (\mathbf{r}_{k,\mathrm{p}})^T\right)^T$$

contains soft information on $\mathbf{h}_k$ given by $\mathbf{r}_{k,\mathrm{h}}$, and on $\mathbf{p}_k$ given by $\mathbf{r}_{k,\mathrm{p}}$.

Assuming that the headers $\mathbf{h}_k$ have been received without errors, the estimation problem in (3.31) is reduced to the constrained estimation of $\mathbf{p}_k$ from $\mathbf{r}_k$, $\mathbf{h}_k$ and ρ_k

$$\widehat{\mathbf{p}}_k = \arg \max_{\mathbf{p} \text{ st } (\mathbf{h}_k, \mathbf{p}) \in \mathcal{S}_k} p\left(\mathbf{r}_{k,\mathrm{p}} | \rho_k, \mathbf{p}\right). \tag{3.32}$$

In (3.32), the term $p\left(\rho_k | \mathbf{z}_k\right)$ of (3.31) is removed. As mentionned in Section 3.1.2.2, $p\left(\rho_k | \mathbf{z}_k\right)$ usually depends on the length of $\mathbf{z}_k$, which is stored in $\mathbf{h}_k$. Since $\mathbf{h}_k$ is assumed to be error-free, $p\left(\rho_k | \mathbf{z}_k\right)$ is known and does not depend on $\mathbf{z}_k$.

The value of ρ_k gives information on the channel characteristics, *e.g.*, signal to noise ratio (SNR), bit error probability, level of fading and allows to evaluate the likelihood $p(\mathbf{r}_{k,\mathrm{p}} | \rho_k, \mathbf{p})$ in (3.32).

The search space for $\widehat{\mathbf{p}}_k$ is the set of all the payloads $\mathbf{p}$ that may be generated from some video sequence $\mathbf{x}$ and which are *compliant* with the information stored in $\mathbf{h}_k$.

In the next section, two tests are introduced to determine whether a given payload $\mathbf{p}$ is syntax-compliant *i.e.*, whether $(\mathbf{h}_k, \mathbf{p}) \in \mathcal{S}_k$. These tests are then used to perform JSCD using the $M-$algorithm [3] to obtain approximate versions of (3.32).

3.3.1. Syntax Compliance Tests

Recall that the payload $\mathbf{p}_k$ represents the output of the 3D-ESCOT coder for a given 3D block $\mathbf{b}_k$. It results from a known number of coding passes n_k and has a known length ℓ_k in bits. As mentioned in Section 3.2.1, n_k and ℓ_k are stored in the header $\mathbf{h}_k$ at the encoder side. The 3D-ESCOT decoder has then to process exactly ℓ_k bits when decoding n_k passes for a given stream $\mathbf{p}$. If more or less than ℓ_k bits are processed, the arithmetic encoder and the arithmetic decoder are desynchronized, which implies that $\mathbf{p}$ contains transmission errors.

More formally, let $\Lambda(\mathbf{p})$ be the number of bits processed when decoding n_k passes for a given payload $\mathbf{p}$ and let $\Lambda(\mathbf{h}_k)$ be the number of bits that should be processed in $\mathbf{p}$. The value of $\Lambda(\mathbf{h}_k)$ is extracted from $\mathbf{h}_k$. Consider the following compliance test

$$t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p}) = \begin{cases} 1 & \text{if } \Lambda(\mathbf{p}) = \Lambda(\mathbf{h}_k) \\ 0 & \text{otherwise.} \end{cases} \tag{3.33}$$

Assuming that the headers $\mathbf{h}_k$ are error-free, $\Lambda(\mathbf{h}_k) = \ell_k$ and n_k are known at the decoder. Therefore, for a given sequence $\mathbf{p}$, $t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p}) = 0$ implies that $\mathbf{p} \neq \mathbf{p}_k$.

For a given header bit stream $\mathbf{h}_k$, consider the sets

$$\mathcal{C}^{\mathrm{b}}(\mathbf{h}_k) = \left\{\mathbf{p} \text{ st } t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p}) = 1\right\} \tag{3.34}$$

and

$$\mathcal{C}(\mathbf{h}_k) = \left\{\mathbf{p} \text{ st } (\mathbf{h}_k, \mathbf{p}) \in \mathcal{S}_k\right\}. \tag{3.35}$$

Each element $\mathbf{p} \in \mathcal{C}(\mathbf{h}_k)$ is compliant with $\mathbf{h}_k$, *i.e.*, the pair $(\mathbf{h}_k, \mathbf{p})$ can be obtained from some $\mathbf{x}$, which implies that $\Lambda(\mathbf{p}) = \Lambda(\mathbf{h}_k)$ and thus $\mathbf{p} \in \mathcal{C}^{\mathrm{b}}(\mathbf{h}_k)$. Therefore

$$\mathcal{C}(\mathbf{h}_k) \subset \mathcal{C}^{\mathrm{b}}(\mathbf{h}_k) \tag{3.36}$$

and $t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p}) = 0$ implies that $\mathbf{p} \notin \mathcal{C}(\mathbf{h}_k)$. The test $t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p})$ allows thus to prove that a payload $\mathbf{p}$ is not compliant with a given header $\mathbf{h}_k$, but it is unable to prove that some $\mathbf{p}$ is compliant with a given header $\mathbf{h}_k$. The bits contained in some $\mathbf{p} \neq \mathbf{p}_k$ may lead to the processing of exactly ℓ_k bits by the 3D-ESCOT decoder. In absence of desynchronisation between the encoder and decoder, the fact that $\mathbf{p} \neq \mathbf{p}_k$ is not detectable by t^{b}.

In the bit stream generated by the Vidwav coder, as specified in [126], the data is byte-aligned and $\Lambda(\mathbf{h}_k)$ is only known in bytes. Without any modification of the Vidwav specifications, (3.33) becomes

$$t^{\mathrm{B}}(\mathbf{h}_k, \mathbf{p}) = \begin{cases} 1 & \text{if } 8\lceil \Lambda(\mathbf{p})/8 \rceil = \Lambda(\mathbf{h}_k) \\ 0 & \text{otherwise.} \end{cases} \tag{3.37}$$

where $\lceil \cdot \rceil$ means the upward rounding. Error detection with t^{B} is then only possible when the desynchronization is sufficient. For a given header $\mathbf{h}_k$ consider the set

$$\mathcal{C}^{\mathrm{B}}(\mathbf{h}_k) = \left\{ \mathbf{p} \text{ st } t^{\mathrm{B}}(\mathbf{h}_k, \mathbf{p}) = 1 \right\}. \tag{3.38}$$

One has

$$\mathcal{C}(\mathbf{h}_k) \subset \mathcal{C}^{\mathrm{b}}(\mathbf{h}_k) \subset \mathcal{C}^{\mathrm{B}}(\mathbf{h}_k) \tag{3.39}$$

and t^{B} is thus less efficient[2] in terms of error detection than t^{b}.

One may write $\Lambda(\mathbf{h}_k) = \ell_k + \lambda_k$, where $\lambda_k = 0, \ldots, 7$ is the number of bits used to pad $\mathbf{p}_k$. In order to implement $t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p})$ at the Vidwav decoder, one has to supplement $\mathbf{h}_k$ with the content of λ_k at the encoder side. This side information requires only three additional bits per entropy-coded 3D block $\mathbf{z}_k$.

Using (3.33) or (3.37), one gets the approximate versions of (3.32)

$$\widehat{\mathbf{p}}_k^{\mathrm{b}} = \arg \max_{\mathbf{p} \in \mathcal{C}^{\mathrm{b}}(\mathbf{h}_k)} p\left(\mathbf{r}_{k,\mathrm{p}} | \rho_k, \mathbf{p}\right) \tag{3.40}$$

and respectively

$$\widehat{\mathbf{p}}_k^{\mathrm{B}} = \arg \max_{\mathbf{p} \in \mathcal{C}^{\mathrm{B}}(\mathbf{h}_k)} p\left(\mathbf{r}_{k,\mathrm{p}} | \rho_k, \mathbf{p}\right), \tag{3.41}$$

which require to consider all the sequences in $\mathcal{C}^{\mathrm{b}}(\mathbf{h}_k)$ or in $\mathcal{C}^{\mathrm{B}}(\mathbf{h}_k)$, when t^{b} or t^{B} are used. These two sets are not well-structured, *i.e.*, their content cannot be easily

[2]Note however that in this case, the packet length information, stored in the header, is shorter.

generated by a finite-state automaton with a reasonable number of states as is the case for Huffman-based variable-length codes [42]. Thus, efficient decoders derived from the BCJR algorithm [6] may not be employed in this case.

Approximate versions of (3.40) and (3.41), with manageable complexity, may be obtained using sequential decoding algorithms, such as the M-algorithm [3] described in the next section.

3.3.2. Sequential Estimation

All the vectors $\mathbf{p}$ of ℓ bits may be represented by a binary tree $\mathcal{T}$ of 2^ℓ leaves. A given node at depth d in $\mathcal{T}$ represents a subsequence $\mathbf{p}_{1:d}$ of d bits, which is represented by the path of length d, starting from the root of $\mathcal{T}$ and leading to the considered node. For a given $\mathbf{h}_k$, only a subset of leaves in $\mathcal{T}$ corresponds to sequences belonging to $\mathcal{C}^{\mathrm{B}}(\mathbf{h}_k)$ or $\mathcal{C}^{\mathrm{b}}(\mathbf{h}_k)$. The aim of the $M-$algorithm is to partly explore $\mathcal{T}$. At iteration d, the M sequences of length d maximizing the metric

$$\mathcal{M}(\mathbf{p}_{1:d}, \mathbf{r}_k) = -\log p(\mathbf{r}_k|\mathbf{p}_{1:d}, \rho_k) \tag{3.42}$$

are kept in a list $\mathcal{L}$ to be used at iteration $d+1$. When $d = \Lambda(\mathbf{h}_k)$, the test $t^{\mathrm{B}}(\mathbf{h}_k, \mathbf{p})$ or $t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p})$ is used to eliminate the sequences in $\mathcal{L}$ which do not belong to $\mathcal{C}^{\mathrm{B}}(\mathbf{h}_k)$ or $\mathcal{C}^{\mathrm{b}}(\mathbf{h}_k)$, starting from the sequence maximizing (3.42).

The steps of the $M-$ algorithm are:

1. Initialization: $\mathcal{L} = \emptyset$, corresponding to the root of $\mathcal{T}$ to which the null metric is assigned.
2. Extend all paths in $\mathcal{L}$ to the following nodes in $\mathcal{T}$.
3. Among the extended paths, keep at most M paths with the largest metric according to (3.42).
4. Go to Step 2 until all the paths in $\mathcal{L}$ reach the depth $\Lambda(\mathbf{h}_k)$.
5. Select the path $\mathbf{p}$ in $\mathcal{L}$ with the largest metric satisfying $t^{\mathrm{B}}(\mathbf{h}_k, \mathbf{p}) = 1$ or $t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p}) = 1$.

This algorithm is suboptimal [3]: if M is not large enough, the transmitted vector $\mathbf{p}_k$ may be discarded at Step 3.

At Step 5, if no vector in $\mathcal{L}$ satisfies one of the compliance tests, $\mathbf{p}_k$ could be considered as dropped and error concealment methods may be used to alleviate the missing blocks effect at the spatial and temporal inverse transform levels. Alternatively, one may use the first element in the list $\mathcal{L}$, corresponding to $M = 1$, even if it does not satisfy $t^{\mathrm{B}}(\mathbf{h}_k, \mathbf{p}) = 1$ or $t^{\mathrm{b}}(\mathbf{h}_k, \mathbf{p}) = 1$; this translates into considering that errors have occurred only at the end of $\mathbf{p}_k$, leading thus to moderate decoding artifacts. In this work this latter strategy is adopted.

In the next section, the simulation results obtained by using the syntax compliance tests t^{b} and t^{B} are presented.

3.3.3. Simulation Results

For the experiments presented in this section, two video sequences are considered: `foreman.qcif` and `mobile.cif`. The following coding parameters are used:

- Haar temporal filtering with $J = 3$ decomposition levels.
- 9×7 spatial filtering with 3 decomposition levels.
- $N_\mathrm{f} = 32$ frames.
- frame rate equal to 15 fps for `foreman.qcif` and to 30 fps for `mobile.qcif`.
- coding rate equal to 128 kbps for `foreman.qcif` and to 768 kbps for `mobile.cif`.

As described in Section 3.2.1, entropy coding is performed on N_b 3D blocks $\mathbf{b}_k$. The term *block size* is further used to refer to the width and the height of these blocks, the depth being fixed to 4 pixels. Two typical block sizes are considered: 22×18 and 11×9 pixels.

The vector $\mathbf{z}$ of the entropy-coded 3D blocks is organized into packets of approximately the same size, with a maximal length of 1000 bytes. Each packet consists of an integer number of sequences $\mathbf{p}_k$. The headers $\mathbf{h}_k$, corresponding to these sequences, are gathered at the beginning of the packet and are assumed to be protected using a strong FEC code, to ensure their good reception at the decoder side, see Figure 3.4. This assumption is reasonable, as headers represent less than 10% of the total bit stream, as presented in Table 3.1.

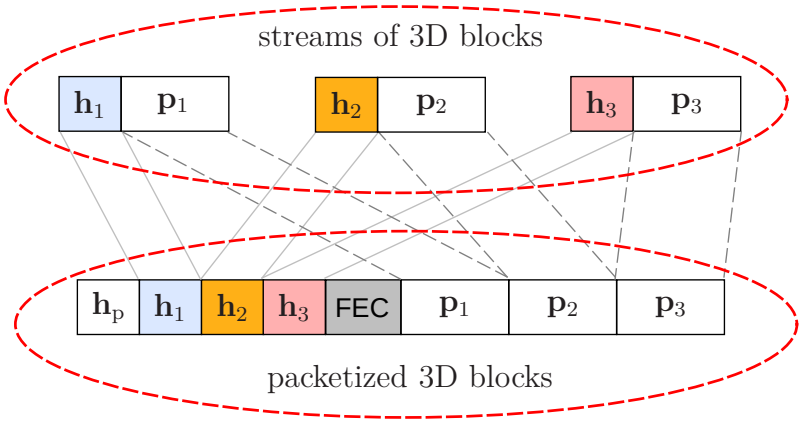


Figure 3.4.: Packetization of the 3D blocks: $\mathbf{h}_k$ are the 3D blocks headers, $\mathbf{p}_k$ are the 3D blocks payloads and $\mathbf{h}_\mathrm{p}$ is the packet header.

The channel presented in Section 3.1.1 is modeled by a BPSK modulation and an additive white Gaussian noise (AWGN) channel with a signal to noise ratio (SNR) going from 9 dB to 12 dB. The generated bit streams are transmitted through this channel and only the payloads $\mathbf{p}_k$, $k = 1, \ldots, N_\mathrm{b}$ are assumed to be corrupted by noise.

	N_b	Headers size	Payloads size
`foreman.qcif`, block size 22×18	273	2712	32257
`foreman.qcif`, block size 11×9	654	3723	31201
`mobile.cif`, block size 22×18	526	10485	94326
`mobile.cif`, block size 11×9	1455	12955	91918

Table 3.1.: Number of 3D blocks, N_b, and headers and payloads size in bytes within the compressed bit streams, as function of the block size.

The JSC decoding is performed using the $M-$algorithm described in Section 3.3.2 for different values of M in order to estimate $\mathbf{p}_k$, $k = 1, , \ldots, N_b$. The JSCD scheme based on the test t^b defined in (3.33) is referred to as JSC-SI, since it uses the side information inserted into the headers at the encoder side. The JSCD scheme based on the test t^B defined in (3.37) does not use the side information and it is referred to as JSC-NoSI. As discussed in Section 3.3.1, the side information used to perform the test t^b consists of 3 additional bits per sequence $\mathbf{p}_k$. The size of this information is then equal to $3N_b$ bits. From Table 3.1, one sees that this supplementary information represents less than 1 % of the total size of the bit stream.

Both JSC-NoSI and JSC-SI schemes are compared to the standard non-robust decoder, which also benefits from the noiseless headers assumption. This decoder corresponds to $M = 1$ and is based on hard decisions of the channel decoder.

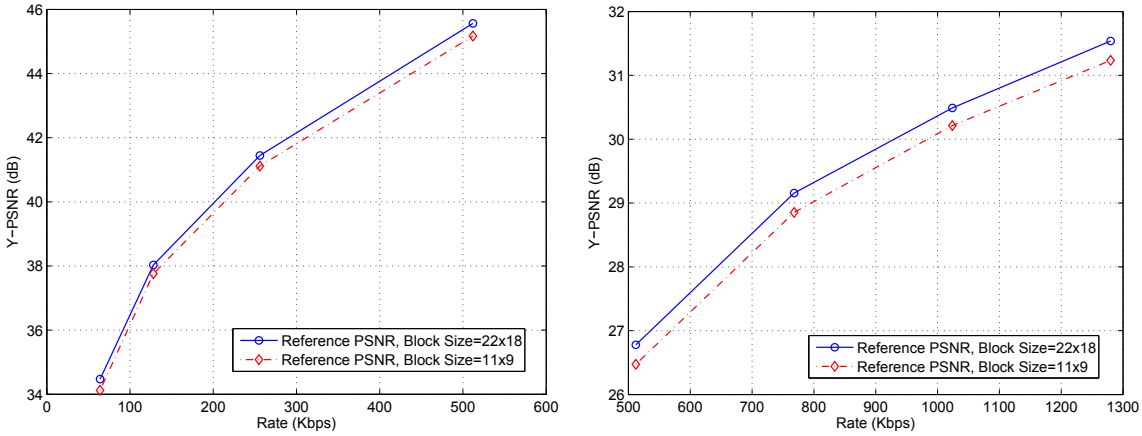
All the presented results are averaged over 100 noise realizations for both `foreman.qcif` and `mobile.cif` sequences.

3.3.3.1. Influence of the 3D Block Size on JSCD Performance

The aim of this part is to compare the performance of JSCD as function of the block sizes. As previously mentioned, for these simulations we consider block sizes of 22×18 and 11×9 pixels. Decreasing the block size results into increasing the number of blocks in the bit stream and thus the frequency at which the arithmetic encoder is reinitialized, but it reduces the encoder efficiency, as illustrated in Figure 3.5.

Block size	22×18				11×9			
Channel SNR (dB)	9	10	11	12	9	10	11	12
Initial PEB	52.20	32.71	14.41	3.79	42.19	21.87	7.24	1.98
PEB, $M = 2$	44.44	24.70	8.24	1.95	32.93	13.61	3.36	0.86
PEB, $M = 8$	36.32	17.52	5.80	1.77	24.26	7.42	2.03	0.77
PEB, $M = 32$	32.11	14.11	5.62	1.72	20.09	5.21	1.95	0.71

Table 3.2.: Percentage of erroneous blocks (PEB) as function of M, the block size and the channel SNR for the JSC-NoSI scheme, `foreman.qcif` sequence.



(a) `foreman.qcif` (b) `mobile.cif`

Figure 3.5.: Y-PSNR (dB) as function of the coding rate (kbps) for block sizes of 22×18 and 11×9 pixels.

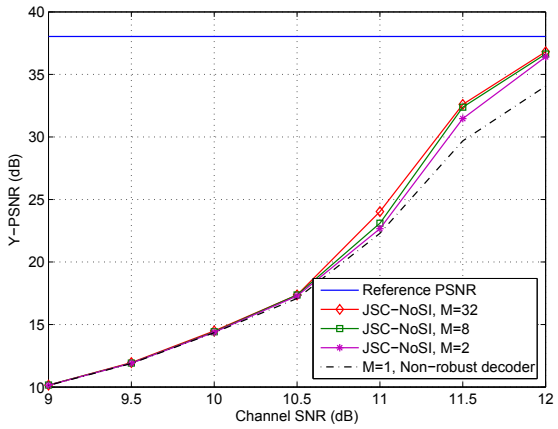
Block Size	22×18				11×9			
Channel SNR (dB)	9	10	11	12	9	10	11	12
Initial PEB	65.92	46.91	21.96	5.94	55.74	28.84	9.83	1.86
PEB, $M = 2$	58.94	37.98	12.93	2.76	45.48	17.90	4.67	1.00
PEB, $M = 8$	52.04	28.85	7.27	1.85	33.78	8.92	2.63	0.69
PEB, $M = 32$	47.15	24.14	4.47	1.57	28.20	6.08	1.62	0.57
PEB, $M = 64$	45.67	21.77	3.90	1.43	26.41	5.43	1.41	0.52

Table 3.3.: Percentage of erroneous blocks (PEB) as function of M, the block size and the channel SNR for the JSC-NoSI scheme, `mobile.cif` sequence.

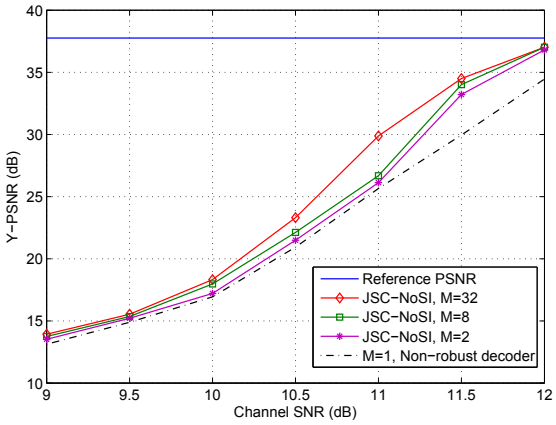
Tables 3.2 and 3.3 present the percentage of erroneous blocks (PEB) present in the compressed bit stream as function of M, the channel SNR and the block size, for the `foreman.qcif` and `mobile.cif` sequences respectively. Here only the JSC-NoSI scheme is considered. The initial PEB, *i.e.*, before correction, is larger when considering larger blocks, which is explained by the fact that these blocks have a higher probability to be corrupted than blocks with smaller size. This probability is equal to $1 - (1 - \eta)^{\ell}$, where η is the bit error probability and ℓ is the length in bits of the entropy-coded 3D block.

Moreover, the PEB decreases when M increases for both block sizes. Relatively to the initial PEB, this decrease is more significant for blocks of size 11×9, which means that the $M-$algorithm presented in Section 3.3.2 is more efficient for sequences with smaller size. This is not surprising since the number of candidates examined by the $M-$algorithm is relatively more important when considering smaller sequences.

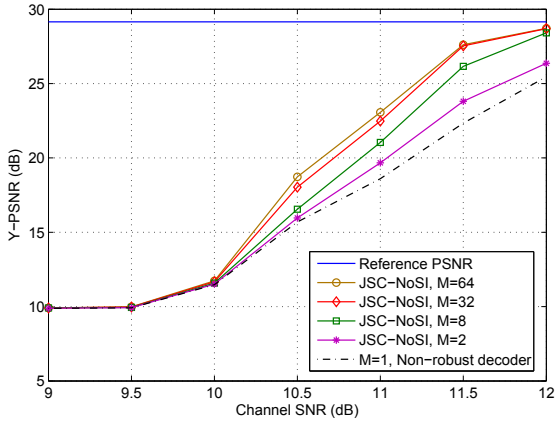
Figure 3.6 shows the average peak signal to noise ratio (PSNR) on the luminance



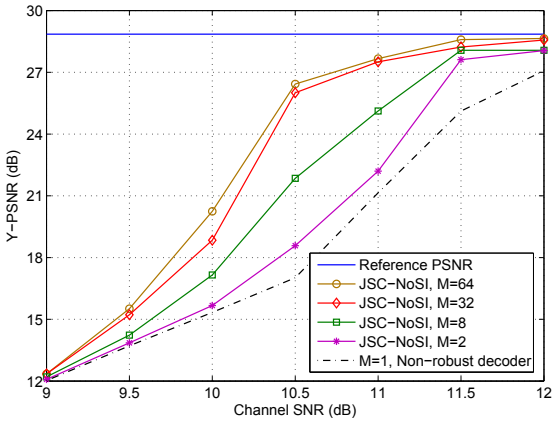
(a) `foreman.qcif`, block size 22×18



(a) `foreman.qcif`, block size 11×9



(c) `mobile.cif`, block size 22×18



(d) `mobile.cif`, block size 11×9

Figure 3.6.: Y-PSNR (dB) as function of the channel SNR (dB) for the non-robust decoder and the JSC-NoSI scheme, for different block sizes.

component Y (Y-PSNR) for the reconstructed `foreman.qcif` and `mobile.cif` sequences using a standard decoder and the JSC-NoSI scheme, for blocks sizes of 22×18 and 11×9 pixels. Different values of the parameter M of the $M-$algorithm have been considered. As expected, for both block sizes, increasing M increases the JSCD performance in terms of PSNR, since more candidates are examined by the $M-$algorithm. For both sequences, the reconstruction PSNR of the standard decoder is lower for blocks of size 22×18, since the initial PEB is more important. For a given value $M > 1$, gains in terms of reconstruction PSNR and in terms of channel SNR are more important for a smaller block size, *e.g.*, for $M = 32$ and for a channel SNR of 11.5 dB, gains in PSNR reach up to 5 dB and up to 2.5 dB for blocks of size 11×9 and 22×18 respectively, for the `foreman.qcif` sequence. Moreover, for $M = 64$ and for a channel SNR of 10.5 dB, gains in PSNR reach up

to 9 dB and up to 4 dB for blocks of size 11×9 and 22×18 respectively, for the `mobile.cif` sequence.

This difference in performance is explained by the fact that the PEB that was corrected is higher for blocks of size 11×9, as deduced from Tables 3.2 and 3.3. Note that the performance is more important for `mobile.cif`, which contains a richer texture; a single erroneous block which is corrected may significantly increase the reconstruction PSNR.

3.3.3.2. Influence of the Additional Side Information on JSCD Performance

In this section, the improvement brought by the use of the additional side information is evidenced. Tables 3.4 and 3.5 compare the PEB when using the JSC-NoSI and the JSC-SI schemes for the `foreman.qcif` and `mobile.cif` sequences respectively, and for blocks of 11×9 pixels. As expected, the use of the side information reduces the PEB present in the bit stream.

	JSC-NoSI				JSC-SI			
Channel SNR (dB)	9	10	11	12	9	10	11	12
Initial PEB	42.19	21.87	7.24	1.98	42.19	21.87	7.24	1.98
PEB, $M = 2$	32.93	13.61	3.36	0.86	32.32	13.01	2.74	0.73
PEB, $M = 8$	24.26	7.42	2.03	0.77	22.86	6.38	1.81	0.69
PEB, $M = 32$	20.09	5.21	1.95	0.71	17.34	3.66	1.75	0.62

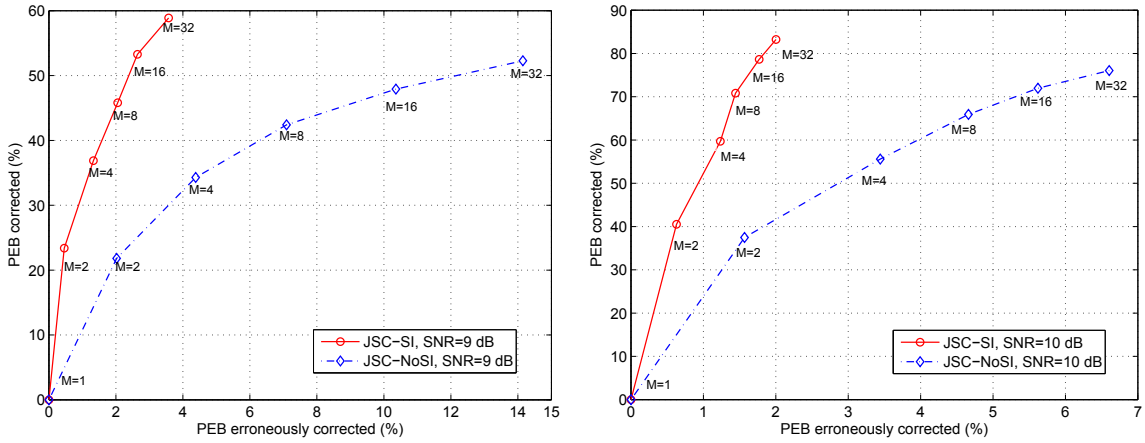
Table 3.4.: Percentage of erroneous blocks (PEB) as function of M and the channel SNR for the JSC-NoSI and the JSC-SI schemes, `foreman.qcif` sequence.

	JSC-NoSI				JSC-SI			
Channel SNR (dB)	9	10	11	12	9	10	11	12
Initial PEB	55.74	28.84	9.83	1.86	55.74	28.84	9.83	1.86
PEB, $M = 2$	45.48	17.90	4.67	1.00	45.00	17.44	3.40	0.72
PEB, $M = 8$	33.78	8.92	2.63	0.69	31.79	7.66	1.34	0.62
PEB, $M = 32$	28.20	6.08	1.62	0.57	23.77	4.09	0.86	0.48
PEB, $M = 64$	26.41	5.43	1.41	0.52	20.88	3.17	0.82	0.41

Table 3.5.: Percentage of erroneous blocks (PEB) as function of M and the channel SNR for the JSC-NoSI and the JSC-SI schemes, `mobile.cif` sequence.

The performance of the JSCD schemes based on the syntax-compliance tests introduced in Section 3.3.1 depends on their ability to detect and correct the erroneous 3D blocks. For a given channel SNR and a given value of M, some of the corrupted blocks may be deemed as corrected by the $M-$algorithm, while they still contain some errors. As mentioned in Section 3.3.1, this comes from the fact that some of

the errors do not lead to a desynchronization between the arithmetic encoder and the arithmetic decoder.



(a) Channel SNR= 9 dB (b) Channel SNR= 10 dB

Figure 3.7.: PEB corrected (%) as function of the PEB erroneously corrected (%) for the JSC-SI and the JSC-NoSI schemes, `foreman.qcif` sequence, block size 11×9.

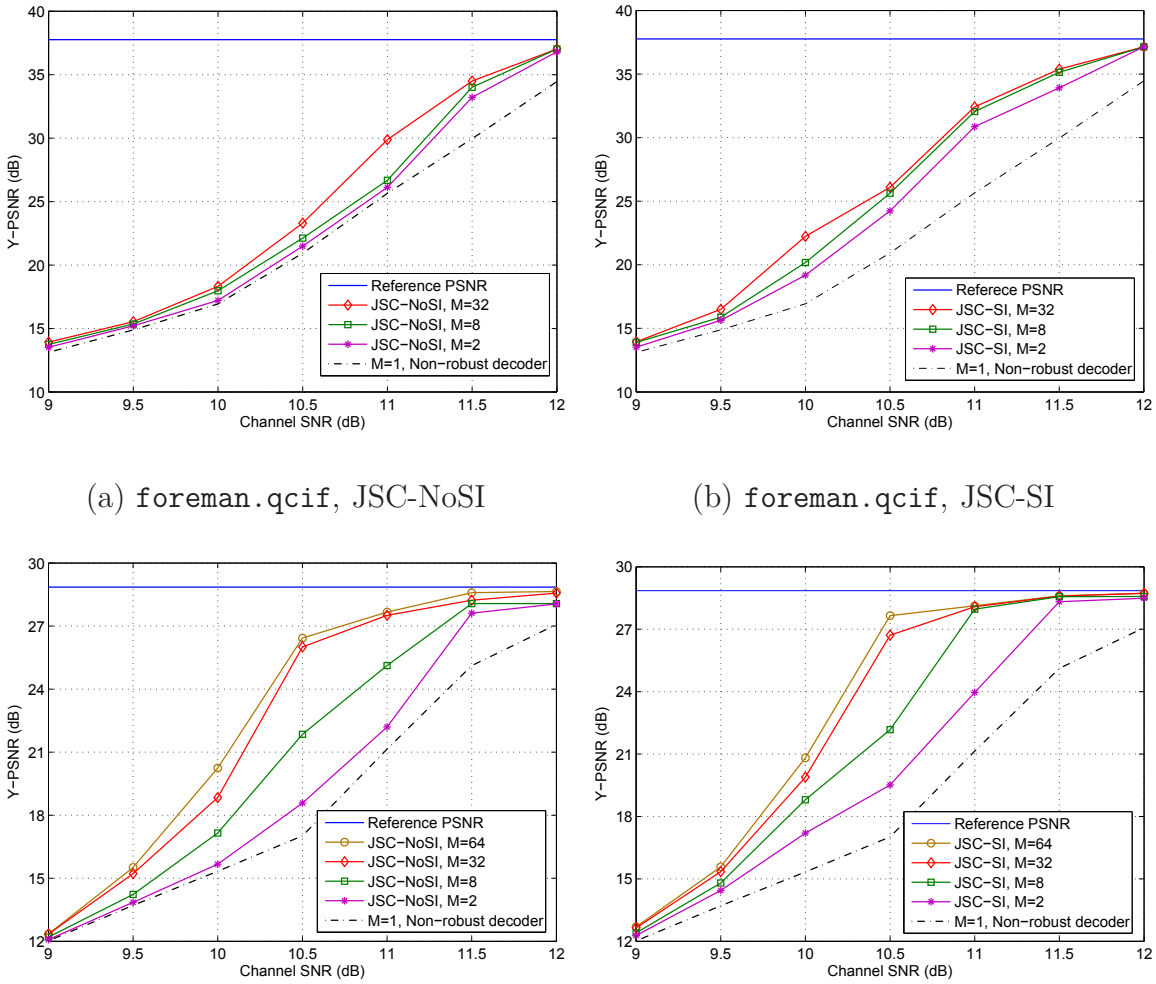
Figure 3.7 illustrates the PEB that are corrected as function of the PEB deemed as corrected, while they still contain some errors, for channel SNRs of 9 dB and 10 dB, and for the `foreman.qcif` sequence. Each curve is parametrized by $M = 2^q$, $q = 0, \ldots, 5$. For $M = 32$ and a channel SNR equal to 10 dB only 2 % of the corrupted blocks are erroneously corrected using the additional side information, while they are about 6.5 % without the use of this information. Moreover, almost 83 % of the corrupted blocks are well-corrected when using JSC-SI, in comparison with the 75 % when using JSC-NoSI.

The gains brought by the use of the side information reach up to about 2 dB in PSNR as illustrated in Figure 3.8 for both `foreman.qcif` and `mobile.cif` sequences.

Figure 3.9 illustrates the qualitative improvement brought by the JSCD schemes (JSC-SI and JSC-NoSI) when compared to the standard decoder for the `mobile.cif` sequence.

3.3.3.3. Decoding Complexity of the JSCD Schemes

In this part, an upper bound of the decoding complexity for the JSCD schemes is provided. Let $n_{2^q}^{\mathrm{d}}$ be the number of 3D blocks that are detected as erroneous in the bit stream for a given parameter $M = 2^q$, $q \in \mathbb{N}$ of the $M-$algorithm. The $N_{\mathrm{b}} - n_1^{\mathrm{d}}$ 3D blocks, initially deemed as correct, are processed only once by the arithmetic decoder. The number of 3D blocks, deemed corrected when increasing q by one, is



(a) `foreman.qcif`, JSC-NoSI (b) `foreman.qcif`, JSC-SI

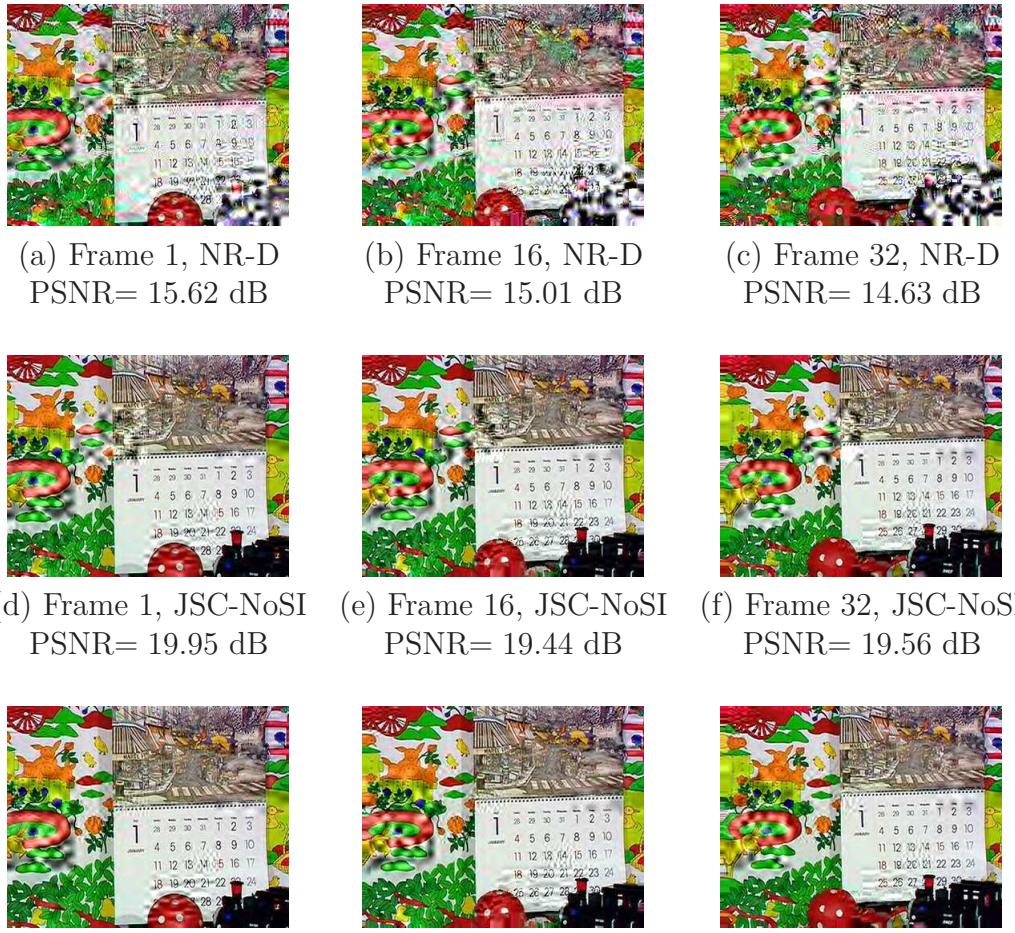
(c) `mobile.cif`, JSC-NoSI (d) `mobile.cif`, JSC-SI

Figure 3.8.: Y-PSNR (dB) as function of the channel SNR (dB) for the non-robust decoder, the JSC-NoSI scheme and the JSC-SI scheme, block size 11×9.

$n^{\mathrm{d}}_{2^q} - n^{\mathrm{d}}_{2^{q+1}}$. Each 3D block, detected as erroneous when $M = 2^q$ and deemed correct when $M = 2^{q+1}$, is processed by the arithmetic decoder at most 2^{q+1} times. Finally, the $n^{\mathrm{d}}_{2^{q_{\max}}}$ remaining 3D blocks, detected as erroneous after using $M_{\max} = 2^{q_{\max}}$, $q_{\max} \in \mathbb{N}^*$ are processed $2^{q_{\max}}$ times by the arithmetic decoder. An upper bound $D(q_{\max})$ of the number of 3D block decoding trials is then

$$D(q_{\max}) = (N_{\mathrm{b}} - n^{\mathrm{d}}_1) + \sum_{q=0}^{q_{\max}-1} \left(n^{\mathrm{d}}_{2^q} - n^{\mathrm{d}}_{2^{q+1}} \right) 2^{q+1} + n^{\mathrm{d}}_{2^{q_{\max}}} 2^{q_{\max}}. \tag{3.43}$$

With the standard decoder, the N_{b} 3D blocks are only processed once. The relative decoding complexity of JSC decoding compared to the standard non-robust decoder



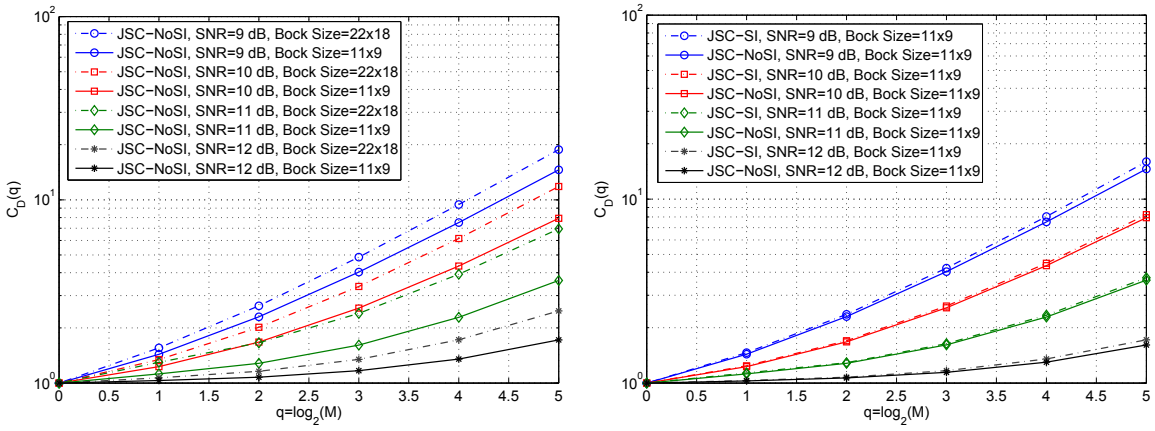
(a) Frame 1, NR-D PSNR= 15.62 dB
(b) Frame 16, NR-D PSNR= 15.01 dB
(c) Frame 32, NR-D PSNR= 14.63 dB
(d) Frame 1, JSC-NoSI PSNR= 19.95 dB
(e) Frame 16, JSC-NoSI PSNR= 19.44 dB
(f) Frame 32, JSC-NoSI PSNR= 19.56 dB
(g) Frame 1, JSC-SI PSNR= 21.19 dB
(h) Frame 16, JSC-SI PSNR= 20.66 dB
(i) Frame 32, JSC-SI PSNR= 21.44 dB

Figure 3.9.: Reconstructed frames of `mobile.cif`, sent over an AWGN channel with SNR= 10 dB and decoded using the non-robust decoder (NR-D), the JSC-NoSI and JSC-SI schemes with $M = 64$.

is then

$$C_{\mathrm{D}}\left(q_{\max}\right) = \frac{D\left(q_{\max}\right)}{N_{\mathrm{b}}}. \tag{3.44}$$

The relative decoding complexity $C_{\mathrm{D}}\left(q_{\max}\right)$, $q_{\max} = \log_2 M$ of the JSC-NoSI scheme is evaluated for the `foreman.qcif` sequence for different block sizes and channel SNRs. The results are reported in Figure 3.10-(a). The complexity increases when the channel SNR decreases, as the number of erroneous 3D blocks to process increases. Moreover, for a given channel SNR, the decoding complexity is higher for large blocks since the PEB is larger for these blocks. For example, for a channel SNR of 11 dB and $M = 16$, the JSC-NoSI decoding complexity is equal to four times the decoding complexity of a standard decoder, when blocks of size 22×18



(a) (b)

Figure 3.10.: Relative decoding complexity $C_{\mathrm{D}}(q)$ of the JSC decoding schemes, as function of $q = \log_2(M)$ for different block sizes and channel SNRs, `foreman.qcif` sequence.

are considered. With blocks of size 11×9, this complexity is reduced to less than three times the complexity of the standard decoder.

Figure 3.10-(b) compares the relative decoding complexity $C_{\mathrm{D}}(q_{\max})$ of the JSC-NoSI and the JSC-SI schemes, for blocks of size 11×9 and for various SNR levels. The complexity slightly increases with the use of the side information, as the proportion of erroneous blocks that are detected is increased. With a channel SNR equal to 11 dB, the decoding complexity of both schemes is less than four times the decoding complexity of a standard decoder. With an SNR of 12 dB, it is less than twice.

3.4. JSCD of Multiple Layers Bit Streams

In this section, the JSC decoding presented in Section 3.3 is applied to multiple layers bit streams.

3.4.1. Layered Bit Stream Generation

As mentioned in Section 3.2, the Vidwav encoder produces a fully embedded and scalable output by generating a multiple-layer bit stream formed by N_{L} layers. To each layer L_m, $m = 1 \ldots, N_{\mathrm{L}}$ is assigned a target bit rate and a spatial and temporal resolution.

The layered bit stream generation is illustrated in Figure 3.11. To build the bit stream of layer L_m at some target bit rate, it is necessary to determine, for each

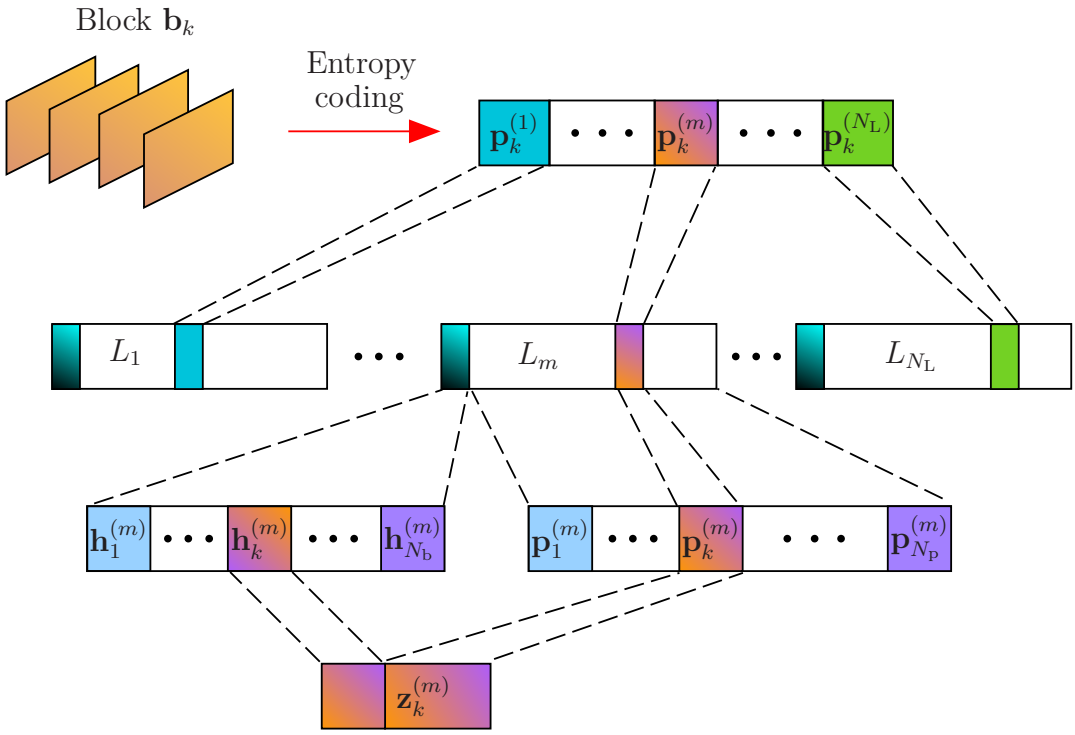


Figure 3.11.: Illustration of the layered bit stream generation.

coding block $\mathbf{b}_k$, how many bytes of its embedded payload $\mathbf{p}_k$ have to be included into this layer. The R-D slope information, obtained at the end of each coding pass during the entropy coding of $\mathbf{b}_k$, is used to perform an R-D optimization which determines the number of coding passes $n_k^{(m)}$ of $\mathbf{p}_k$ that have to be considered in L_m.

Let $\mathbf{p}_k^{(m)}$ be the substream of the payload $\mathbf{p}_k$ that is included into layer L_m. The header $\mathbf{h}_k^{(m)}$ associated to $\mathbf{p}_k^{(m)}$ contains information such as its size $\ell_k^{(m)}$ in bytes, the number $n_k^{(m)}$ of coding passes involved in $\mathbf{p}_k^{(m)}$, as well as the R-D slope information associated to these coding passes. One has $\ell_k = \sum_{m=1}^{N_\mathrm{L}} \ell_k^{(m)}$ and $n_k = \sum_{m=1}^{N_\mathrm{L}} n_k^{(m)}$.

The header $\mathbf{h}_k^{(1)}$ contains also other information of $\mathbf{h}_k$ that is independent of the way the payload $\mathbf{p}_k$ is partitioned between the different layers, such as the spatio-temporal subband index of the 3D block $\mathbf{b}_k$ and the associated motion vectors. Thus, one may write

$$\mathbf{p}_k = \left(\left(\mathbf{p}_k^{(1)}\right)^T, \ldots, \left(\mathbf{p}_k^{(N_\mathrm{L})}\right)^T \right)^T \tag{3.45}$$

and

$$\mathbf{h}_k = \left(\left(\mathbf{h}_k^{(1)}\right)^T, \ldots, \left(\mathbf{h}_k^{(N_\mathrm{L})}\right)^T \right)^T. \tag{3.46}$$

The layers L_m, $m = 1, \ldots, N_{\mathrm{L}}$ are then formed as follows

$$L_m = \left(\left(\mathbf{h}_1^{(m)}\right)^T, \ldots, \left(\mathbf{h}_{N_{\mathrm{b}}}^{(m)}\right)^T, \left(\mathbf{p}_1^{(m)}\right)^T, \ldots, \left(\mathbf{p}_{N_{\mathrm{b}}}^{(m)}\right)^T \right)^T. \tag{3.47}$$

Finally, the vector $\mathbf{z}$ to be transmitted is reorganized in layers

$$\mathbf{z} = \left((L_1)^T, \ldots, (L_{N_{\mathrm{L}}})^T \right)^T. \tag{3.48}$$

3.4.2. JSCD Scheme

The JSCD scheme presented in this section relies on the same compliance tests introduced in Section 3.3.1, which are applied here to each substream $\mathbf{p}_k^{(m)}$, $k = 1, \ldots, N_{\mathrm{b}}$ of each layer L_m, $m = 1, \ldots, N_{\mathrm{L}}$.

As defined in Section 3.3.1, let $\Lambda\left(\mathbf{p}_k^{(m)}\right)$ be the number of bits processed when decoding $n_k^{(m)}$ passes of a given payload $\mathbf{p}_k^{(m)}$. The number of bits $\Lambda\left(\mathbf{h}_k^{(m)}\right)$ that should be processed in $\mathbf{p}_k^{(m)}$ is deduced from the header $\mathbf{h}_k^{(m)}$. As the header $\mathbf{h}_k$ defined in (3.46) is assumed to be reliably transmitted, then $\Lambda\left(\mathbf{h}_k^{(m)}\right) = \ell_k^{(m)}$.

For a given layer $m = 1, \ldots, N_{\mathrm{L}}$, if for some estimate

$$\widetilde{\mathbf{p}}_k^{(1:m)} = \left(\left(\widetilde{\mathbf{p}}_k^{(1)}\right)^T, \ldots, \left(\widetilde{\mathbf{p}}_k^{(m)}\right)^T \right)^T,$$

the decoding of $\sum_{i=1}^{m} n_k^{(i)}$ passes requires strictly more or strictly less than $\sum_{i=1}^{m} \Lambda\left(\mathbf{h}_k^{(i)}\right)$ bits, then $\widetilde{\mathbf{p}}_k^{(1:m)} \neq \mathbf{p}_k^{(1:m)}$.

As mentioned in Section 3.3.1, the lengths $\Lambda\left(\mathbf{h}_k^{(m)}\right)$ are known in bytes at the decoder side. The syntax compliance test t^{B} defined in (3.37) may then be used without introducing any additional information. In this case, error detection is possible when the desynchronization between the arithmetic encoder and decoder is sufficient, *i.e.*, when more or less bytes than expected are consumed. The most reliable compliance test t^{b} defined in (3.33) may be considered, provided that each header $\mathbf{h}_k^{(m)}$ is supplemented with three bits indicating the number of bits used to pad the last byte of $\mathbf{p}_k^{(m)}$. Erros leading to a single bit desynchronization may then be detected using t^{b} for each substream $\mathbf{p}_k^{(m)}$.

The tests t^{b} and t^{B} may be performed to estimate each substream $\mathbf{p}_k^{(m)}$ of each layer L_m, $m = 1, \ldots, N_{\mathrm{L}}$ as follows:

1. for $m = 1$, $\mathbf{p}_k^{(1)}$ is estimated as in Section 3.3 and $\widehat{\mathbf{p}}_k^{(1)}$ is obtained by (3.40) or by (3.41).
2. for $m = 2, \ldots, N_{\mathrm{L}}$, assume that estimates $\widehat{\mathbf{p}}_k^{(i)}$, $i = 1, \ldots, m-1$ of $\mathbf{p}_k^{(i)}$ have been obtained:

a) as in Section 3.3.2, the $M-$algorithm is used to get the M most likely candidates $\widetilde{\mathbf{p}}(1), \ldots, \widetilde{\mathbf{p}}(M)$ to estimate $\mathbf{p}_k^{(m)}$

b) $k = 1$

c) do

i. if

$$\Lambda\left(\widehat{\mathbf{p}}_k^{(1:m-1)}, \widetilde{\mathbf{p}}(k)\right) = \sum_{i=1}^{m} \Lambda\left(\mathbf{h}_k^{(i)}\right), \tag{3.49}$$

return $\widehat{\mathbf{p}}_k^{(m)} = \widetilde{\mathbf{p}}(k)$

ii. else $k = k + 1$

d) while $k \leqslant M$.

If no vector $\widetilde{\mathbf{p}}(k)$, $k = 1, \ldots, M$ satisfying (3.49) is found, the vectors $\mathbf{p}_k^{(m)}, \ldots, \mathbf{p}_k^{(N_\mathrm{L})}$ are dropped and the recovery of the content of the 3D block $\mathbf{b}_k$ is obtained by entropy decoding the sequence $\widehat{\mathbf{p}}_k^{(1:m-1)}$.

3.4.2.1. Efficiency of the Compliance Tests in the Multi-Layer Case

As discussed in the single-layer case, the compliance tests t^b and t^B allow only to prove that some payload $\mathbf{p}$ is not syntax-compliant.

Generalizing to the multi-layer case, the bits contained in some corrupted sequence

$$\left(\left(\widetilde{\mathbf{p}}_k^{(1)}\right)^T, \ldots, \left(\widetilde{\mathbf{p}}_k^{(m)}\right)^T\right)^T, \ m = 1, \ldots, N_\mathrm{L}$$

may lead to the processing of exactly $\sum_{i=1}^{m} \ell_k^{(i)}$ bits. Therefore, for $m = 2, \ldots, N_\mathrm{L}$ the fact that

$$\Lambda\left(\widetilde{\mathbf{p}}_k^{(1:m)}\right) = \sum_{i=1}^{m} \Lambda\left(\mathbf{h}_k^{(i)}\right)$$

does not imply that $\widetilde{\mathbf{p}}_k^{(m)} = \mathbf{p}_k^{(m)}$, for each $m = 2, \ldots, N_\mathrm{L}$.

As a consequence, for $m = 2, \ldots, N_\mathrm{L}$ and for a given sequence of estimates $\widehat{\mathbf{p}}_k^{(1)}, \ldots, \widehat{\mathbf{p}}_k^{(m-1)}$ verifying

$$\Lambda\left(\widehat{\mathbf{p}}_k^{(1:m')}\right) = \sum_{i=1}^{m'} \Lambda\left(\mathbf{h}_k^{(i)}\right), \ m' = 1, \ldots, m-1,$$

the fact that

$$\Lambda\left(\widehat{\mathbf{p}}_k^{(1:m-1)}, \widetilde{\mathbf{p}}_k^{(m)}\right) \neq \sum_{i=1}^{m} \Lambda\left(\mathbf{h}_k^{(i)}\right)$$

does not necessarily imply that $\widetilde{\mathbf{p}}_k^{(m)} \neq \mathbf{p}_k^{(m)}$. Indeed, the arithmetic decoder may not be desynchronized while decoding a corrupted sequence

$$\left(\left(\widetilde{\mathbf{p}}_k^{(1)}\right)^T, \ldots, \left(\widetilde{\mathbf{p}}_k^{(m-1)}\right)^T\right)^T,$$

but a desynchronization may happen while decoding the sequence

$$\left(\left(\widetilde{\mathbf{p}}_k^{(1)}\right)^T, \ldots, \left(\widetilde{\mathbf{p}}_k^{(m-1)}\right)^T, \left(\mathbf{p}_k^{(m)}\right)^T\right)^T.$$

Therefore, for $m = 2, \ldots, N_{\mathrm{L}}$ the syntax-compliance tests t^{b} and t^{B} are less efficient than for $m = 1$, since they are based on the estimations $\widehat{\mathbf{p}}_k^{(1)}, \ldots, \widehat{\mathbf{p}}_k^{(m-1)}$, obtained at previous layers and which may be corrupted.

3.4.3. Simulation Results

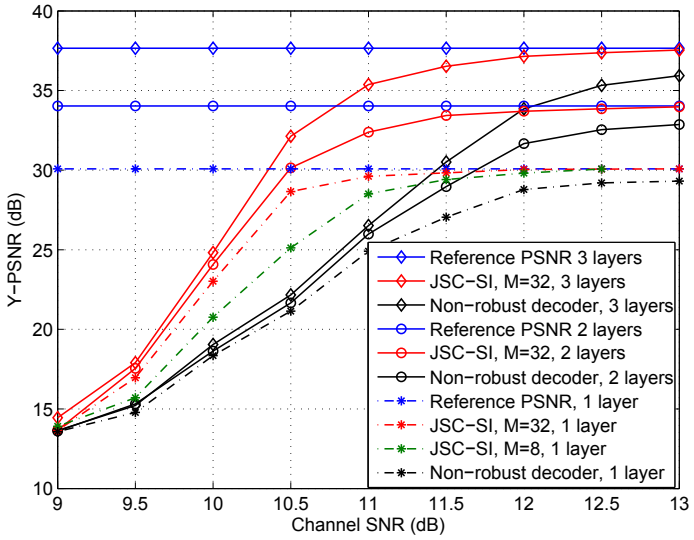


Figure 3.12.: Y-PSNR (dB) comparison of the standard non-robust decoder and JSC-SI, as a function the channel SNR (dB) when one, two and three layers are decoded, `foreman.qcif`, block size 11×9.

In this part, the simulation results obtained for the multiple-layer case are presented. The `foreman.qcif` sequence is encoded using three layers, corresponding to cumulative bit rates 32 kbps (1 layer), 64 kbps (2 layers) and 128 kbps (3 layers). Side information is generated for each layer and the JSC-SI decoding is performed using the syntax-compliance test t^{b}.

Figure 3.12 compares the average PSNR obtained at the output of the JSC-SI decoder and at the output of the non-robust decoder. The decoding of one, two, and three layers is performed. Note that the improvement in terms of PSNR brought

(a) PSNR=21.01 dB (b) PSNR=29.40 dB

Figure 3.13.: Frame 6 of `foreman.qcif` transmitted over an AWGN channel with SNR = 11 dB. Layer L_1 is decoded using the non-robust decoder (a) and using the JSC-SI decoder with $M = 32$ (b).

(a) PSNR=32.35 dB (b) PSNR=32.63 dB

Figure 3.14.: Frame 6 of `foreman.qcif` transmitted over an AWGN channel with SNR = 11 dB. Layer L_1 is decoded using the JSC-SI decoder with $M = 32$. Layer L_2 is decoded using the non-robust decoder (a) and using the JSC-SI decoder with $M = 32$ (b).

by decoding the third layer becomes less important for low channel SNRs. This is mainly due to the fact that corrupted substreams which are not detected as erroneous at the two first layers do not allow the corrupted substreams at the third layer to be corrected. Even substreams at third layer, which are error-free, may be deemed as erroneous as discussed in Section 3.4.2.1. The JSC-SI scheme is thus less efficient for higher (low-priority) layers. However, note that at an equivalent bit rate of 128 kbps, the JSC-SI scheme working on a three-layered bit stream provides more improvement in PSNRs than the JSC-SI scheme working on a single-layered bit stream, see Figure 3.8-(b). For example, for $M = 32$ and for a channel SNR= 10.5 dB, the gain in PSNR brought when considering three layers is about 10 dB when compared to the non-robust decoder, while it is about 5 dB in the single layer case.

Figures 3.13, 3.14 and 3.15 show the qualitative improvement brought by the proposed JSC-SI scheme on the 6−th frame of the decoded `foreman.qcif` sequence, when compared to the non-robust decoder. The layers are successively decoded and

(a) PSNR=35.75 dB (b) PSNR=36.30 dB

Figure 3.15.: Frame 6 of `foreman.qcif` transmitted over an AWGN channel with SNR = 11 dB. Layers L_1 and L_2 are decoded using the JSC-SI decoder with $M = 32$. Layer L_3 is decoded using the non-robust decoder (a) and using the JSC-SI decoder with $M = 32$ (b).

the visual improvement is illustrated for each decoded layer.

3.5. Conclusion

In this chapter, we have presented an efficient joint-source channel video decoding scheme which exploits the source residual redundancy. Simulations have shown that this scheme is able to correct transmission errors introduced by error-prone channels and improves the reconstruction of the corrupted video bit streams. Moreover, the additional redundancy, introduced in the headers at the encoder side, improves the decoding performance at the price of a slight modification of the syntax of the bit stream and of a little increase in the bit rate. The number of erroneous blocks is then significantly reduced.

In the next chapter, we propose to apply this robust decoding scheme in a multiple description context.

4. JSC Decoding of Multiple Description Encoded Video

In Chapter 3, we have proposed a joint-source channel decoding technique which combats errors occurring during a noisy transmission.

In this chapter, we consider a more general transmission scheme consisting of a mixed network, formed by a packet-loss wired part followed by a noisy wireless channel. The effect of packet losses occurring on the wired part is mitigated by multiple description coding, which avoids packet retransmission and reduces transmission delays. The aim of the joint source channel decoding approach described in Chapter 3, is to correct transmission errors introduced by the wireless part of the network and remaining after channel decoding. This allows to increase the number of error-free packets which are processed by the multiple description decoder. The redundancy introduced between the descriptions is then used to exploit the received and corrected packets and reconstruct the video sequence.

4.1. Problem Formulation

Assume that some video sequence has to be encoded and transmitted from a server to a client via a heterogeneous network. A typical application context, illustrated in Figure 4.1, is video streaming to a mobile client connected to some 3G/4G base station or WiFi access point. The network consists of a wired part introducing some losses *e.g.*, packet-switched network such as Internet, and a wireless part introducing some noise in the transmitted bit stream.

As in Chapter 3, the frames of the video sequence are assumed to be represented by a vector $\mathbf{x}$. In Chapter 3, we have assumed that the transmission were losses-free. In this chapter, the wired part of the network is not reliable. To mitigate packet losses occurring in this part *e.g.*, due to congestion or router failure, a multiple description (MD) encoder is employed to produce two correlated descriptions $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ of $\mathbf{x}$

$$\mathbf{x}^{(i)} = \mathbf{F}^{(i)}\left(\mathbf{x}\right), \; i = 1, 2, \tag{4.1}$$

where $\mathbf{F}^{(i)}\left(\cdot\right)$ is the coding function for the i-th description. Prior to transmission, the descriptions are quantized and entropy coded to get

$$\mathbf{z}^{(i)} = \mathbf{E}\left(\mathbf{x}^{(i)}\right), \tag{4.2}$$

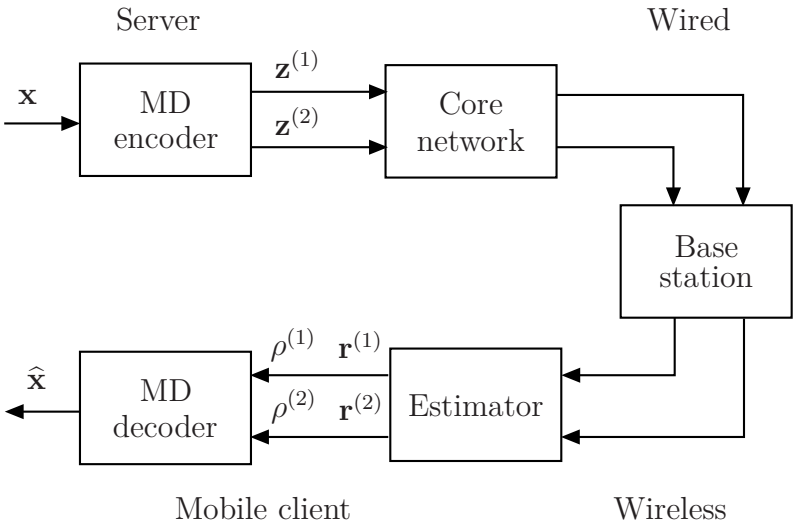


Figure 4.1.: Heterogeneous video transmission scheme of multiple description bit streams.

where as in Chapter 3, $\mathbf{E}(\cdot)$ denotes the entropy coding function assumed independent of the description[1] . The entropy-coded descriptions $\mathbf{z}^{(i)}$, $i = 1, 2$ are then packetized into $N_{\mathrm{p},i}$ packets $\mathbf{z}_k^{(i)}$ of more or less similar lengths to get

$$\mathbf{z}^{(i)} = \left(\left(\mathbf{z}_1^{(i)} \right)^T, \ldots, \left(\mathbf{z}_{N_{\mathrm{p},i}}^{(i)} \right)^T \right)^T .$$

Some packets $\mathbf{z}_k^{(i)}$ may be lost in the wired part of the network, some are received affected by noise introduced by the wireless part of the network. The channel model introduced in Chapter 3, is adopted here and similar notations are used for each description to model the channel state information

$$\boldsymbol{\rho}^{(i)} = \left(\rho_1^{(i)}, \ldots, \rho_{N_{\mathrm{p},i}}^{(i)} \right)^T$$

and the decoder soft information

$$\mathbf{r}^{(i)} = \left(\left(\mathbf{r}_1^{(i)} \right)^T, \ldots, \left(\mathbf{r}_{N_{\mathrm{p},i}}^{(i)} \right)^T \right)^T .$$

The content of $\mathbf{r}_k^{(i)}$, $k = 1, \ldots, N_{\mathrm{p},i}$ may only be exploited if $\rho_k^{(i)} \geqslant 1$, *i.e.*, if the packet is not lost.

[1] In some MD coding schemes, such as MD scalar quantization not considered here, $\mathbf{F}^{(i)}$ may incorporate the quantization step.

4.1.1. Optimal MAP Estimation

The aim here is to evaluate the MAP estimate $\widehat{\mathbf{x}}_{\mathrm{MAP}}$ of $\mathbf{x}$ knowing $\boldsymbol{\rho}^{(i)}$ and $\mathbf{r}^{(i)}$, $i = 1, 2$

$$\widehat{\mathbf{x}}_{\mathrm{MAP}} = \arg\max_{\mathbf{x}} p\left(\mathbf{x}|\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}\right). \tag{4.3}$$

Marginalizing (4.3) with respect to $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$, one gets

$$\widehat{\mathbf{x}}_{\mathrm{MAP}} = \arg\max_{\mathbf{x}} \sum_{\mathbf{z}^{(1)},\mathbf{z}^{(2)}} p\left(\mathbf{x}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}|\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}\right), \tag{4.4}$$

where

$$p\left(\mathbf{x}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}|\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}\right) = \frac{p\left(\mathbf{x}\right) p\left(\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}|\mathbf{x}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) p\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}|\mathbf{x}\right)}{p\left(\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}\right)}. \tag{4.5}$$

Since $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$ are deduced from $\mathbf{x}$, in (4.5) one has

$$p\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}|\mathbf{x}\right) = \begin{cases} 1 & \text{if } \mathbf{z}^{(i)}{=}\mathbf{E}\left(\mathbf{F}^{(i)}\left(\mathbf{x}\right)\right), \ i = 1, 2 \\ 0 & \text{otherwise.} \end{cases}$$

Using the fact that $\mathbf{x}$ does not provide more information than $\mathbf{E}\left(\mathbf{F}^{(i)}\left(\mathbf{x}\right)\right)$, $i = 1, 2$ on $\boldsymbol{\rho}^{(i)}$ and $\mathbf{r}^{(i)}$, from (4.5) one gets

$$\widehat{\mathbf{x}}_{\mathrm{MAP}} = \arg\max_{\mathbf{x}} p\left(\mathbf{x}\right) p\left(\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}|\mathbf{E}\left(\mathbf{F}^{(1)}\left(\mathbf{x}\right)\right), \mathbf{E}\left(\mathbf{F}^{(2)}\left(\mathbf{x}\right)\right)\right). \tag{4.6}$$

Again, the estimator in (4.6) is not easily obtained. This estimator has been employed, *e.g.*, in [1], where MD is performed by duplicating spatial subbands adequately quantized to minimize redundancy. However, (4.6) is unpractical for the MD scheme we have considered.

As in Section 3.1.2.2 of Chapter 3, one may proceed by estimating first $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$, accounting for the fact that these entropy-coded descriptions have been generated from the same video signal $\mathbf{x}$, before deducing an estimate for $\mathbf{x}$. The resulting *constrained* MAP estimator of $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$ knowing $\boldsymbol{\rho}^{(i)}$ and $\mathbf{r}^{(i)}$, $i = 1, 2$, is

$$\left(\widehat{\mathbf{z}}^{(1)}, \widehat{\mathbf{z}}^{(2)}\right) = \arg\max_{\left(\mathbf{z}^{(1)},\mathbf{z}^{(2)}\right)\in\mathcal{S}} p\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}|\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}\right), \tag{4.7}$$

where

$$\mathcal{S} = \left\{\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) \text{ st } \exists\mathbf{x} \mid \mathbf{z}^{(1)}{=}\mathbf{E}\left(\mathbf{F}^{(1)}\left(\mathbf{x}\right)\right) \text{ and } \mathbf{z}^{(2)} = \mathbf{E}\left(\mathbf{F}^{(2)}\left(\mathbf{x}\right)\right)\right\} \tag{4.8}$$

is the set of all pairs of entropy-coded descriptions that may be generated from all the possible $\mathbf{x}$.

Using Bayes' rule, (4.7) becomes

$$\left(\widehat{\mathbf{z}}^{(1)}, \widehat{\mathbf{z}}^{(2)}\right) = \arg \max_{\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) \in \mathcal{S}} p\left(\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)} | \mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) p\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right). \quad (4.9)$$

In (4.9), one has

$$p\left(\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)} | \mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) = p\left(\mathbf{r}^{(1)}, \mathbf{r}^{(2)} | \boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) p\left(\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)} | \mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right). \quad (4.10)$$

Assuming that the transmission of each description follows its own route, with independent transmission conditions, in the wired part of the network, one has

$$p\left(\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)} | \mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) = \prod_{i=1}^{2} p\left(\boldsymbol{\rho}^{(i)} | \mathbf{z}^{(i)}\right). \quad (4.11)$$

Furthermore, assuming that the realizations of the wireless part of the network are independent for each description, one gets

$$p\left(\mathbf{r}^{(1)}, \mathbf{r}^{(2)} | \boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) = \prod_{i=1}^{2} p\left(\mathbf{r}^{(i)} | \boldsymbol{\rho}^{(i)}, \mathbf{z}^{(i)}\right). \quad (4.12)$$

The assumptions made in Section 3.1.2.2 of Chapter 3 for a single description bit stream and which concern the independence of the congestion conditions and of the wireless channel realizations, for each transmitted packet, may be considered here for each description $\mathbf{z}^{(i)}$. One finally obtains

$$\left(\widehat{\mathbf{z}}^{(1)}, \widehat{\mathbf{z}}^{(2)}\right) = \arg \max_{\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) \in S} p\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) \prod_{i=1}^{2} \prod_{k=1}^{N_{\mathrm{p},i}} p\left(\mathbf{r}_k^{(i)} | \rho_k^{(i)}, \mathbf{z}_k^{(i)}\right) p\left(\rho_k^{(i)} | \mathbf{z}_k^{(i)}\right). \quad (4.13)$$

Note that in (4.13), when $\rho_k^{(i)} = 0$, the packet has been lost before reaching the wireless part of the network. In this case, $p\left(\mathbf{r}_k^{(i)} | \rho_k^{(i)}, \mathbf{z}_k^{(i)}\right)$ is independent of $\mathbf{z}_k^{(i)}$. When $\rho_k^{(i)} \geqslant 1$, the evaluation of $p\left(\mathbf{r}_k^{(i)} | \rho_k^{(i)}, \mathbf{z}_k^{(i)}\right)$ depends on the values of $\mathbf{z}_k^{(i)}$ and $\rho_k^{(i)}$ and on the channel model. Moreover, in (4.13), determining whether a pair $(\mathbf{z}^{(1)}, \mathbf{z}^{(2)})$ belongs to the set $\mathcal{S}$, depends on the MD coding scheme used to generate the two descriptions. This scheme is described in the next section.

4.2. MD Coding Scheme

As in Chapter 3, the number of frames considered in $\mathbf{x}$ is denoted by N_{f}, and $N = N_{\mathrm{f}} N_{\mathrm{r}} N_{\mathrm{c}}$ is the total number of entries of $\mathbf{x}$, where N_{r} and N_{c} denote the number of rows and columns in each frame respectively.

The considered MD coding scheme relies on a redundant motion-compensated wavelet decomposition with J levels in the temporal domain based on the Haar filter bank [148]. We have implemented the MD scheme of [148], on the Vidwav coder presented in Section 3.2, which is originally designed for single description coding. This MD scheme belongs to the family of frame expansion, presented in Section 2.4.2.3 of Chapter 2.

The redundancy between the descriptions is introduced by an oversampled decomposition at the last temporal level J, leading to a redundancy factor in terms of number of frames equal to $1 + \left(\frac{1}{2}\right)^J$. The J-th temporal decomposition is represented by an $N(1 + (\frac{1}{2})^J) \times N$ redundant transform matrix T_J^{r}. The redundant vector of temporal subbands $\mathbf{y}^{\mathrm{r}}$ of size $N(1 + (\frac{1}{2})^J)$, is obtained as follows

$$\mathbf{y}^{\mathrm{r}} = T_J^{\mathrm{r}} T_{J-1} \dots T_1 \mathbf{x} = \left((\mathbf{y}_1^{\mathrm{r}})^T, \dots, \left(\mathbf{y}_{N_{\mathrm{f}}^{\mathrm{r}}}^{\mathrm{r}}\right)^T \right)^T, \tag{4.14}$$

where $N_{\mathrm{f}}^{\mathrm{r}} = N_{\mathrm{f}}(1 + (\frac{1}{2})^J)$ and $\mathbf{y}_n^{\mathrm{r}}$, $n = 1, \dots, N_{\mathrm{f}}^{\mathrm{r}}$ is of size $N_{\mathrm{r}} N_{\mathrm{c}}$. The temporal subbands $\mathbf{y}^{\mathrm{r}}$ are spatially transformed to get

$$\mathbf{s}^{\mathrm{r}} = S^{\mathrm{r}} \mathbf{y}^{\mathrm{r}} = \left((\mathbf{s}_1^{\mathrm{r}})^T, \dots, \left(\mathbf{s}_{N_{\mathrm{f}}^{\mathrm{r}}}^{\mathrm{r}}\right)^T \right)^T, \tag{4.15}$$

where S^{r} is the $N^{\mathrm{r}} \times N^{\mathrm{r}}$ block diagonal matrix, with $N^{\mathrm{r}} = N_{\mathrm{f}}^{\mathrm{r}} N_{\mathrm{r}} N_{\mathrm{c}}$, of the same structure as the matrix S defined in (3.18).

The vector of redundant spatio-temporal coefficients $\mathbf{s}^{\mathrm{r}}$ is further partitioned into two descriptions. Let $\mathcal{N}^{(1)}$ and $\mathcal{N}^{(2)}$ be a partition of $\{1, \dots, N_{\mathrm{f}}^{\mathrm{r}}\}$. Then the description $\mathbf{x}^{(i)}$ contains all $\mathbf{s}_n^{\mathrm{r}}$ with $n \in \mathcal{N}^{(i)}$. This partitioning can be represented by $\mathbf{x}^{(i)} = P^{(i)} \mathbf{s}^{\mathrm{r}}$, $i = 1, 2$, where $P^{(i)}$ is a $\frac{N^{\mathrm{r}}}{2} \times N^{\mathrm{r}}$ binary matrix defining the partition of frames in the spatio-temporal subbands within each description. As in [148], a balanced partitioning scheme is considered, to allow an independent reconstruction of each description with similar quality. The MD encoding functions $\mathbf{F}^{(i)}$ introduced in (4.1) are then

$$\mathbf{F}^{(i)} = P^{(i)} S^{\mathrm{r}} T_J^{\mathrm{r}} T_{J-1} \dots T_1, \ i = 1, 2. \tag{4.16}$$

As in the case of single description coding, each $\mathbf{x}^{(i)}$ is organized into $N_{\mathrm{b},i}$ 3D blocks $\mathbf{b}_k$, $k = 1, \dots, N_{\mathrm{b},i}$. These blocks are further independently entropy-coded using the 3D-ESCOT algorithm to get $\mathbf{z}_k^{(i)}$, leading to the bit streams

$$\mathbf{z}^{(i)} = \left(\left(\mathbf{z}_1^{(i)}\right)^T, \dots, \left(\mathbf{z}_{N_{\mathrm{b},i}}^{(i)}\right)^T \right)^T = \mathbf{E}(\mathbf{x}^{(i)}), \ i = 1, 2.$$

In this chapter, one layered bit streams are considered for each description. Note however that multiple layer bit streams may be generated for each description, leading to a scalable MD coding scheme. Along this chapter, the notation $(\cdot)^{(i)}$, $i = 1, 2$ is used to refer to which description a given sequence belongs.

4.3. JSCD of the Descriptions

The aim of this section is to provide a practical implementation of (4.13), which is adapted to the MD coding scheme presented in Section 4.2.

As in Chapter 3, we assume for simplicity sake that each packet $\mathbf{z}_k^{(i)}$ introduced in Section 4.1 contains a single 3D block. In this case, $N_{\mathrm{p},i} = N_{\mathrm{b},i}$, $i = 1, 2$. Moreover, all the pairs $\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) \in \mathcal{S}$ are assumed to be equally probable *a priori*, so that (4.13) becomes

$$\left(\widehat{\mathbf{z}}^{(1)}, \widehat{\mathbf{z}}^{(2)}\right) = \arg \max_{\left(\mathbf{z}_{1:N_{\mathrm{b},1}}^{(1)}, \mathbf{z}_{1:N_{\mathrm{b},2}}^{(2)}\right) \in \mathcal{S}} \prod_{i=1}^{2} \prod_{j=1}^{N_i} p\left(\mathbf{r}_j^{(i)} | \rho_j^{(i)}, \mathbf{z}_j^{(i)}\right) p\left(\rho_j^{(i)} | \mathbf{z}_j^{(i)}\right), \quad (4.17)$$

where

$$\mathbf{z}_{1:N_{\mathrm{b},i}}^{(i)} = \left(\left(\mathbf{z}_1^{(i)}\right)^T, \ldots, \left(\mathbf{z}_{N_{\mathrm{b},i}}^{(i)}\right)^T\right)^T.$$

Now let us consider the sets

$$\mathcal{S}^{(i)} = \left\{\mathbf{z}^{(i)} \text{ st } \exists \mathbf{x} \mid \mathbf{z}^{(i)} = \mathbf{E}\left(\mathbf{F}^{(i)}(\mathbf{x})\right)\right\} \quad (4.18)$$

and

$$\mathcal{S}_k^{(i)} = \left\{\mathbf{z}_k^{(i)} \text{ st } \exists \mathbf{x} \mid \exists \left(\tilde{\mathbf{z}}_1^{(i)}, \ldots, \tilde{\mathbf{z}}_{k-1}^{(i)}, \tilde{\mathbf{z}}_{k+1}^{(i)}, \ldots, \tilde{\mathbf{z}}_{N_{\mathrm{b},i}}^{(i)}\right), \left(\left(\tilde{\mathbf{z}}_1^{(i)}\right)^T, \ldots, \left(\tilde{\mathbf{z}}_{k-1}^{(i)}\right)^T, \left(\mathbf{z}_k^{(i)}\right)^T, \left(\tilde{\mathbf{z}}_{k+1}^{(i)}\right)^T, \ldots, \left(\tilde{\mathbf{z}}_{N_{\mathrm{b},i}}^{(i)}\right)^T\right)^T = \mathbf{E}\left(\mathbf{F}^{(i)}(\mathbf{x})\right)\right\}. \quad (4.19)$$

The sets $\mathcal{S}^{(i)}$ and $\mathcal{S}_k^{(i)}$ contain respectively all the entropy-coded descriptions and all the entropy-coded 3D blocks $\mathbf{z}_k^{(i)}$, $k = 1, \ldots, N_{\mathrm{b},i}$ one may obtain considering all the possible values of $\mathbf{x}$.

Remark 2. One has

$$\mathcal{S} \subset \mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \subseteq \left(\bigotimes_{k=1}^{N_{\mathrm{b},1}} \mathcal{S}_k^{(1)}\right) \times \left(\bigotimes_{k=1}^{N_{\mathrm{b},2}} \mathcal{S}_k^{(2)}\right). \quad (4.20)$$

Consider $\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) \in \mathcal{S}$. From (4.8), there exists some sequence $\mathbf{x}$ such that $\mathbf{z}^{(i)} = \mathbf{E}(\mathbf{F}^{(i)}(\mathbf{x}))$, $i = 1, 2$, which implies $\mathbf{z}^{(i)} \in \mathcal{S}^{(i)}$ and $\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) \in \mathcal{S}^{(1)} \times \mathcal{S}^{(2)}$. Consider now $\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) \in \mathcal{S}^{(1)} \times \mathcal{S}^{(2)}$. For $i = 1, 2$

$$\exists \mathbf{x}_i \text{ st } \mathbf{z}^{(i)} = \left(\left(\mathbf{z}_1^{(i)}\right)^T, \ldots, \left(\mathbf{z}_{N_{\mathrm{b},i}}^{(i)}\right)^T\right)^T = \mathbf{E}(\mathbf{F}^{(i)}(\mathbf{x}_i)), \quad (4.21)$$

and thus $\mathbf{z}_k^{(i)} \in \mathcal{S}_k^{(i)}$, $k = 1, \ldots, N_{\mathrm{b},i}$, leading to $\mathbf{z}^{(i)} \in \left(\bigotimes_{k=1}^{N_{\mathrm{b},i}} \mathcal{S}_k^{(i)}\right)$ and $(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}) \in \left(\bigotimes_{k=1}^{N_{\mathrm{b},1}} \mathcal{S}_k^{(1)}\right) \times \left(\bigotimes_{k=1}^{N_{\mathrm{b},2}} \mathcal{S}_k^{(2)}\right)$.

In general, $\mathcal{S} \neq \mathcal{S}^{(1)} \times \mathcal{S}^{(2)}$. Consider $\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) \in \mathcal{S}^{(1)} \times \mathcal{S}^{(2)}$ and let $\mathcal{X}\left(\mathbf{z}^{(i)}\right) = \left\{\mathbf{x} \text{ st } \mathbf{z}^{(i)} = \mathbf{E}(\mathbf{F}^{(i)}(\mathbf{x}))\right\}$. If $\mathcal{X}\left(\mathbf{z}^{(1)}\right) \cap \mathcal{X}\left(\mathbf{z}^{(2)}\right) \neq \emptyset$ then $\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) \in \mathcal{S}$. Assume now that for all $\left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}\right) \in \mathcal{S}^{(1)} \times \mathcal{S}^{(2)}$, one has $\mathcal{X}\left(\mathbf{z}^{(1)}\right) \cap \mathcal{X}\left(\mathbf{z}^{(2)}\right) \neq \emptyset$. This means that $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$ are independent, which is not the case in a general MD coding scheme, as soon as it introduces some redundancy. Similarly, to show that $\mathcal{S}^{(1)} \times \mathcal{S}^{(2)} = \left(\bigotimes_{k=1}^{N_{\mathrm{b},1}} \mathcal{S}_k^{(1)}\right) \times \left(\bigotimes_{k=1}^{N_{\mathrm{b},2}} \mathcal{S}_k^{(2)}\right)$, it suffices to show that $\left(\bigotimes_{k=1}^{N_{\mathrm{b},i}} \mathcal{S}_k^{(i)}\right) \subset \mathcal{S}^{(i)}$, $i = 1, 2$. For that purpose, one has to show that $\bigcap_{k=1}^{N_{\mathrm{b},i}} \mathcal{X}\left(\mathbf{z}_k^{(i)}\right) \neq \emptyset$ for all $\mathbf{z}_{1:N_{\mathrm{b},i}}^{(i)} \in \left(\bigotimes_{k=1}^{N_{\mathrm{b},i}} \mathcal{S}_k^{(i)}\right)$. Necessary conditions for this is that the quantization and entropy-coding operations of each $\mathbf{z}_k^{(i)}$ are performed independently and that each individual $\mathbf{F}^{(i)}$ does not introduce any redundancy.

Similarly to the single description coding case, on may derive a suboptimal estimator for each component $\mathbf{z}_k^{(i)}$, $k = 1, \ldots, N_{\mathrm{b},i}$ of $\mathbf{z}^{(i)}$, $i = 1, 2$, using (4.17) and (4.20)

$$\widehat{\mathbf{z}}_k^{(i)} = \arg \max_{\mathbf{z}_k^{(i)} \in \mathcal{S}_k^{(i)}} p\left(\mathbf{r}_k^{(i)} | \rho_k^{(i)}, \mathbf{z}_k^{(i)}\right) p\left(\rho_k^{(i)} | \mathbf{z}_k^{(i)}\right). \tag{4.22}$$

This estimator may be used only when $\rho_j^{(i)} \geqslant 1$. The estimator $\left(\widehat{\mathbf{z}}_{1:N_{\mathrm{b},1}}^{(1)}, \widehat{\mathbf{z}}_{1:N_{\mathrm{b},2}}^{(2)}\right)$ is a suboptimal solution of (4.17) since $\left(\widehat{\mathbf{z}}_{1:N_{\mathrm{b},1}}^{(1)}, \widehat{\mathbf{z}}_{1:N_{\mathrm{b},2}}^{(2)}\right)$ is not necessarily in S, see (4.20). In what follows the estimation process of $\mathbf{z}_k^{(i)}$ is described when $\rho_k^{(i)} \geqslant 1$, then when $\rho_k^{(i)} = 0$.

4.3.1. When $\rho_k^{(i)} \geqslant 1$

As in Chapter 3, each transmitted packet $\mathbf{z}_k^{(i)}$, $k = 1, \ldots, N_{\mathrm{b},i}$ consists of a header $\mathbf{h}_k^{(i)}$ and a payload $\mathbf{p}_k^{(i)}$

$$\mathbf{z}_k^{(i)} = \left(\left(\mathbf{h}_k^{(i)}\right)^T, \left(\mathbf{p}_k^{(i)}\right)^T\right)^T.$$

The received vector

$$\mathbf{r}_k^{(i)} = \left(\left(\mathbf{r}_{k,\mathrm{h}}^{(i)}\right)^T, \left(\mathbf{r}_{k,\mathrm{p}}^{(i)}\right)^T\right)^T$$

contains soft information on $\mathbf{h}_k^{(i)}$ given by $\mathbf{r}_{k,\mathrm{h}}^{(i)}$ and on $\mathbf{p}_k^{(i)}$ given by $\mathbf{r}_{k,\mathrm{p}}^{(i)}$.

Moreover, the header $\mathbf{h}_k^{(i)}$ is assumed to be received without errors. As mentioned earlier, the term $p\left(\rho_k^{(i)} | \mathbf{z}_k^{(i)}\right)$ in (4.22) does not depend on the content of $\mathbf{z}_k^{(i)}$ but on

its length which is stored in $\mathbf{h}_k^{(i)}$. Thus, the estimation problem in (4.22) reduces to evaluate

$$\widehat{\mathbf{p}}_k^{(i)} = \arg \max_{\mathbf{p}_k^{(i)} \text{ st } \left(\mathbf{h}_k^{(i)}, \mathbf{p}_k^{(i)}\right) \in \mathcal{S}_k^{(i)}} p\left(\mathbf{r}_{k,\mathrm{p}}^{(i)} | \rho_k^{(i)}, \mathbf{p}_k^{(i)}\right). \tag{4.23}$$

The estimation in (4.23) is performed as in the case of single description bit streams, presented in Section 3.3 of Chapter 3 by using a JSC decoder which exploits the residual redundancy left in each description. The syntax compliance tests t^{b} and t^{B} introduced in Section 3.3.1 are considered for each $\mathbf{p}_k^{(i)}$ to eliminate the elements which are not syntax-compliant by using the $M-$algorithm as described in Section 3.3.2.

4.3.2. When $\rho_k^{(i)} = 0$

When $\rho_j^{(i)} = 0$, the estimator in (4.22) cannot be used since neither $\mathbf{r}_{k,\mathrm{h}}^{(i)}$ nor $\mathbf{r}_{k,\mathrm{p}}^{(i)}$ are available. All vectors $\widetilde{\mathbf{z}}_k^{(i)}$ are equally likely. The most reasonable estimate for $\mathbf{z}_k^{(i)}$ is to take the all zero vector and to replace the coefficients in the corresponding spatio-temporal subband by zeros, or to ignore these coefficients when performing reconstruction, as will be presented in the next section.

4.4. Reconstruction

In this section, we propose to reconstruct the video stream $\mathbf{x}$ from $\left(\widehat{\mathbf{z}}_{1:N_{\mathrm{b},1}}^{(1)}, \widehat{\mathbf{z}}_{1:N_{\mathrm{b},2}}^{(2)}\right)$ obtained as described in the previous section. Consider the vector

$$\widetilde{\mathbf{x}}^{(i)} = \left(\left(\widetilde{\mathbf{x}}_1^{(i)}\right)^T, \ldots, \left(\widetilde{\mathbf{x}}_{N_{\mathrm{b},i}}^{(i)}\right)^T\right)^T,$$

with $\widetilde{\mathbf{x}}_k^{(i)} = \mathbf{E}^{-1}(\widehat{\mathbf{z}}_k^{(i)})$ if $\rho_k^{(i)} \geqslant 1$ and $\widetilde{\mathbf{x}}_k^{(i)} = \mathbf{0}$ otherwise, where $\mathbf{E}^{-1}$ represents the entropy decoding function associated to $\mathbf{E}$. By combining $\widetilde{\mathbf{x}}^{(1)}$ and $\widetilde{\mathbf{x}}^{(2)}$, and assuming that all the corrupted blocks have been efficiently corrected, one gets the vector $\widetilde{\mathbf{s}}^{\mathrm{r}}$ which is a version of the transmitted vector of spatio-temporal subbands $\mathbf{s}^{\mathrm{r}}$, affected by quantization noise and block losses. The aim is to recover $\mathbf{x}$ from $\widetilde{\mathbf{s}}^{\mathrm{r}}$.

In a first time, we consider the case where there are no lost blocks, *i.e.*, $\rho_k^{(i)} \geqslant 1$, $k = 1, \ldots, N_{\mathrm{b},i}$, $i = 1, 2$. Then $\widetilde{\mathbf{s}}^{\mathrm{r}}$ is only affected by the quantization noise and the reconstruction of $\mathbf{x}$ corresponds to *central decoding*. After inverse spatial transform of $\widetilde{\mathbf{s}}^{\mathrm{r}}$, one obtains the vector of temporal subbbands $\mathbf{w} = (S^{\mathrm{r}})^{-1}\,\widetilde{\mathbf{s}}^{\mathrm{r}}$.

As mentioned in Section 4.2, the redundancy between the two descriptions is introduced at the last temporal level J by the means of oversampled motion-compensated temporal filtering. In [148], the central decoding is performed on $\mathbf{w}$ to recover the

frames of the temporal approximation subband at level $J-1$ (the detail frames of the temporal levels $1, .., J-1$ do not contain any redundancy). The reconstruction for a temporal approximation frame at level $J-1$ is performed by the central decoder either by taking into account only one reconstructed description, given by one of the side decoders, or by computing the mean of the reconstructions provided by the two side decoders, as proposed in [148]. We point out that this central decoding corresponds to multiplying $\mathbf{w}$ by the pseudo-inverse of the redundant temporal transform T^{r} when no motion estimation/compensation is considered.

Let us now consider the case where blocks have been lost. The spatio-temporal approximation subband contains the most important part of the bit stream. A block lost in this subband can seriously damage the whole reconstruction of the video sequence. To limit the damage propagation when losses occur within this subband, we identify the frames of the temporal approximation subband at level $J-1$ that would be affected by these losses. Once these frames are identified, they are reconstructed by taking into account only the description wherein these blocks have been correctly received; otherwise, if the same blocks have been corrupted in both descriptions, the central decoding is performed as in [148]. For the temporal detail subbands, the lost blocks are simply replaced with zero coefficients, losing thus the details of the inverse motion-compensation.

In this reconstruction approach, neither spatial nor temporal concealment has been considered. Only the temporal correlation between the descriptions has been exploited to reconstruct the temporal approximation frames at level $J-1$.

4.5. Simulation Results

For the experimental part, the `foreman.qcif` sequence has been encoded at a frame rate $R_{\mathrm{f}} = 15$ frames per second (fps) using the temporal MD coding scheme presented in Section 4.2, with $J = 3$ temporal decomposition levels and 3 spatial decomposition levels. The 3D blocks are of size 11×9, as in the Chapter 3, the proposed JSC decoding scheme has proven to be more efficient for this block size than for a block size of 22×18. The number of frames considered in $\mathbf{x}$ is $N_{\mathrm{f}} = 32$.

In the next section, we provide some statistics about the proportion of headers and data generated by the MD coder and the amount of redundancy introduced by the additional side information (SI) to allow the use of the syntax-compliance test t^{b}.

4.5.1. Bit Stream Organization

For both single description (SD) and MD coding schemes, the generated bit stream is put into packets of approximately the same size, with a length limit of $L_{\max} = 1000$ bytes. Each packet contains an integer number of encoded 3D blocks. The

headers of these 3D blocks are gathered in the beginning of each packet. These headers are then protected using a strong FEC, to ensure their good reception, even when the wireless part of the channel is noisy. Alternatively, one may use the FEC provided by the network, such as the check sum of UDP-Lite [90, 108], which may then be used as error-correcting code as detailed in [108] to improve the reliability of the header in presence of channel impairments. In what follows, we assume that thanks to the strong FEC or using the technique in [108], one gets error-free headers at decoder side.

Figure 4.2 presents the percentage of headers $\phi_{\mathrm{H}}^{\mathrm{SD}}(R)$ and $\phi_{\mathrm{H}}^{\mathrm{MD}}(R)$ in the generated SD and MD bit streams as function of the coding rate R. In both cases, as the rate increases, ϕ_{H} decreases. Moreover for the same coding rate R, one has $\phi_{\mathrm{H}}^{\mathrm{MD}}(R) > \phi_{\mathrm{H}}^{\mathrm{SD}}(R)$. There is some redundancy in the headers generated by the MD coder (number of frames N_{f}, number of spatial and temporal levels, number of descriptions, *etc.*). In what follows, we assume that using properly-designed header compression, one manages to get $\phi_{\mathrm{H}}^{\mathrm{MD}}(R) \simeq \phi_{\mathrm{H}}^{\mathrm{SD}}(R)$.

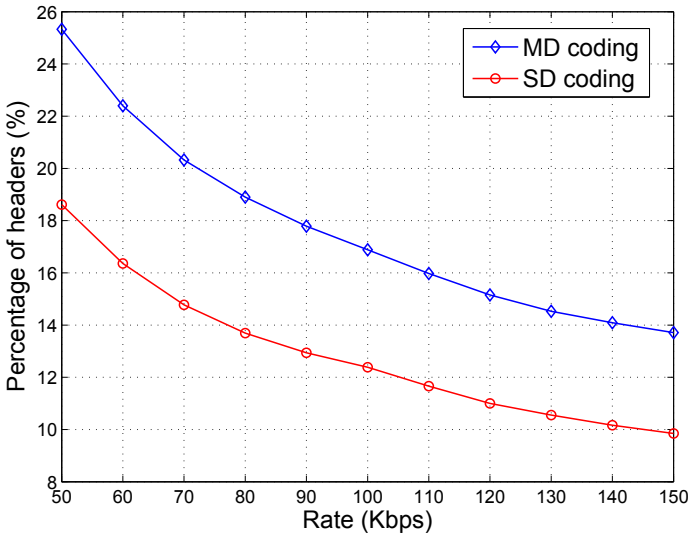


Figure 4.2.: Percentage of headers (%) in the total bit stream as function of the coding rate (kbps), for MD and SD coding.

As mentioned before, the length in bytes of each encoded 3D block is already present in the headers. Transmitting this size in bits requires three additional bits per 3D block. The corresponding amount of redundancy is

$$\phi_{\mathrm{SI}} = \frac{\sum_{i=1}^{2} 3N_{\mathrm{b},i}}{\sum_{i=1}^{2}\sum_{k=1}^{N_{\mathrm{b},i}} \ell(\mathbf{h}_k^{(i)}) + \ell(\mathbf{p}_k^{(i)})}, \tag{4.24}$$

where $\ell(\mathbf{h}_k^{(i)})$ and $\ell(\mathbf{p}_k^{(i)})$ denote the lengths of $\mathbf{h}_k^{(i)}$ and $\mathbf{p}_k^{(i)}$ respectively. This redundancy has been evaluated for several coding rates going from 50 kbps to 150 kbps and found to be almost constant and equal to 1 %. The 3 bits per 3D block may be

inserted into the headers to implement the syntax compliance test t^{b} instead of t^{B} as described in Section 3.3.1.

4.5.2. Performance of the JSCD Scheme

In this section, we present the performance of the robust JSC decoding scheme applied to both descriptions as presented in Sections 4.3 and 4.4. We assume in a first time, that no packet is lost in the wired part of the network and that the wireless link is described by a BPSK modulation and an additive white Gaussian noise (AWGN) channel with a known signal-to-noise ratio (SNR), *i.e*, $\rho_k^{(i)} = 1$, $k = 1, \ldots, N_{\mathrm{b},i}$, $i = 1, 2$. The first N_{f} frames of `foreman.qcif` are encoded at 75 kbps. The parameter M of the $M-$algorithm is increased to $M = 1024$.

Figure 4.3 shows the residual bit error rate (BER) for various channel SNRs as function of M when using the test t^{B} defined in (3.37) (JSC-NoSI) and when using the test t^{b} defined in (3.33), requiring the SI (JSC-SI). The results for both descriptions are similar and have been averaged in all the presented figures. The value $M = 1$ corresponds to a plain standard *central* decoder. As expected, increasing M decreases the BER in the compressed bit stream. Moreover, the use of the side information decreases the BER and is especially useful when M is large. It allows to gain more than 0.5 dB in terms of channel SNR at 9.5 dB or higher.

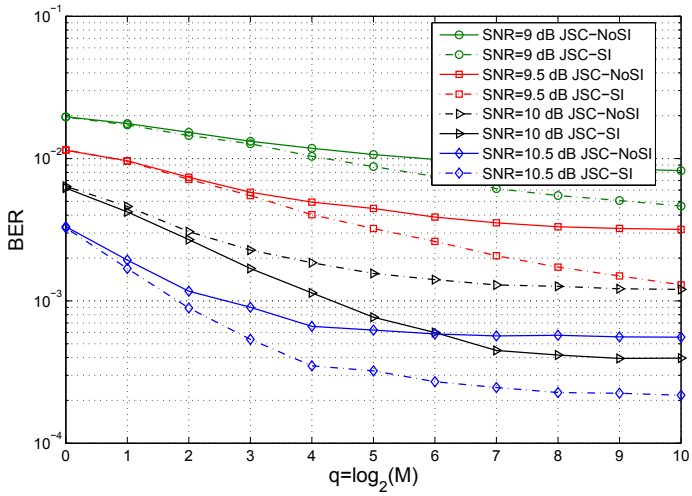


Figure 4.3.: BER as function of $q = \log_2(M)$, for various channel SNRs.

As discussed in Chapter 3, the performance of the JSC decoding schemes based on the syntax-compliance tests t^{b} and t^{B} depends on their ability to detect and correct erroneous 3D blocks. With a channel SNR of 9 dB, initially with JSC-NoSI, about 9 % of the erroneous 3D blocks were deemed as correct and thus missed. With JSC-SI, the percentage of erroneous blocks (PEB) missed is reduced to 3 %.

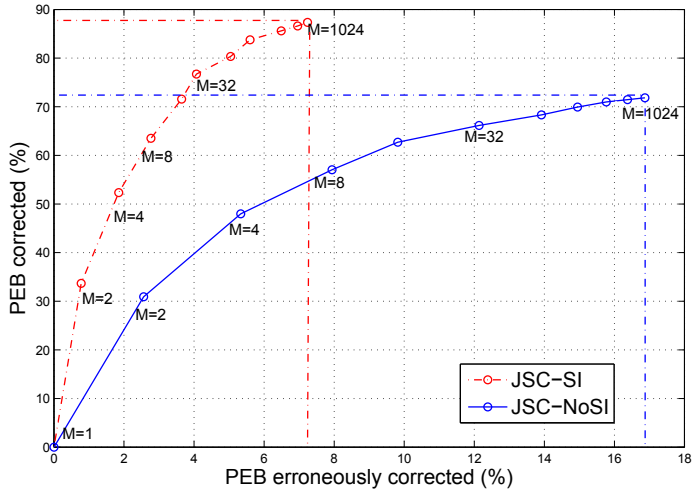


Figure 4.4.: PEB corrected (%) as function of the PEB erroneously corrected (%) for the JSC-SI and the JSC-NoSI schemes, channel SNR of 9 dB.

Figure 4.4 illustrates the PEB that are *well* corrected as function of the PEB that are *erroneously* corrected for the JSC-SI and JSC-NoSI schemes and for a channel SNR of 9 dB. For $M = 1024$, only 7.2 % of the corrupted blocks are erroneously corrected using the SI while they are 16.9 % without SI. Moreover 87.4 % of the corrupted blocks are well corrected when considering JSC-SI to be compared with 71.8 % when considering JSC-NoSI. As in SD coding, the additional side information reduces the amount of corrupted blocks which are erroneously corrected.

Figure 4.5 presents the performance of the JSC decoding schemes in terms of the reconstruction PSNR, for different values of the channel SNR and of the parameter M. The reference PSNR for the MD decoder without channel impairment is 34.14 dB. The gain obtained by the JSC-SI scheme reaches up to 6 dB in PSNR when compared to the JSC-NoSI scheme, for a channel SNR of 9 dB and for $M = 1024$. As previously mentioned, the curve for $M = 1$ corresponds to the standard non-robust MD decoder. Compared to this non-robust decoder, the gain in PSNR reaches up to 7 dB considering the JSC-NoSI scheme and up to 10 dB using the JSC-SI scheme, with $M = 1024$ and a channel SNR of 9.5 dB.

In terms of decoding complexity, the upper bound $D(q_{\max})$ defined in 3.43 is evaluated here as follows

$$D(q_{\max}) = (N_{\mathrm{b},1} + N_{\mathrm{b},2} - n_1^{\mathrm{d}}) + \sum_{q=0}^{q_{\max}-1} \left(n_{2^q}^{\mathrm{d}} - n_{2^{q+1}}^{\mathrm{d}}\right) 2^{q+1} + n_{2^{q_{\max}}}^{\mathrm{d}} 2^{q_{\max}}, \tag{4.25}$$

where $n_{2^q}^{\mathrm{d}}$ denotes the number of 3D blocks of both descriptions, deemed as corrupted for $M = 2^q$, $q \in \mathbb{N}$. The relative decoding complexity of the JSC decoding scheme

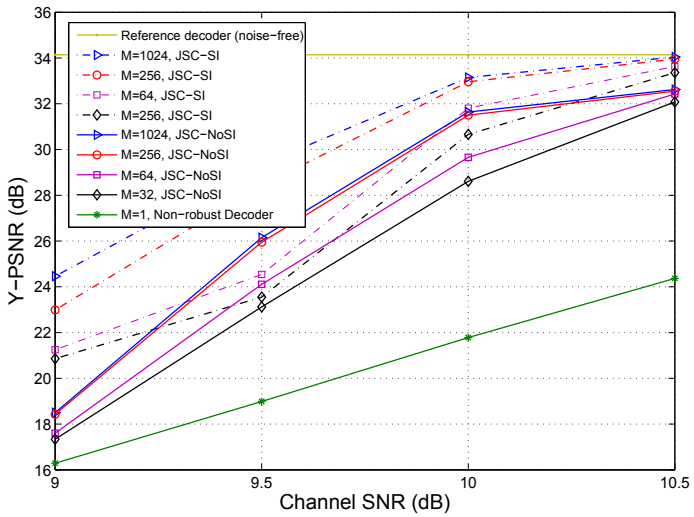


Figure 4.5.: Y-PSNR (dB) as function of the channel SNR (dB) for JSC-SI and JSC-NoSI for different values of M.

compared to a standard MD decoder is computed as

$$C_{\mathrm{D}}\left(q_{\max}\right) = \frac{D\left(q_{\max}\right)}{N_{\mathrm{b},1} + N_{\mathrm{b},2}}. \tag{4.26}$$

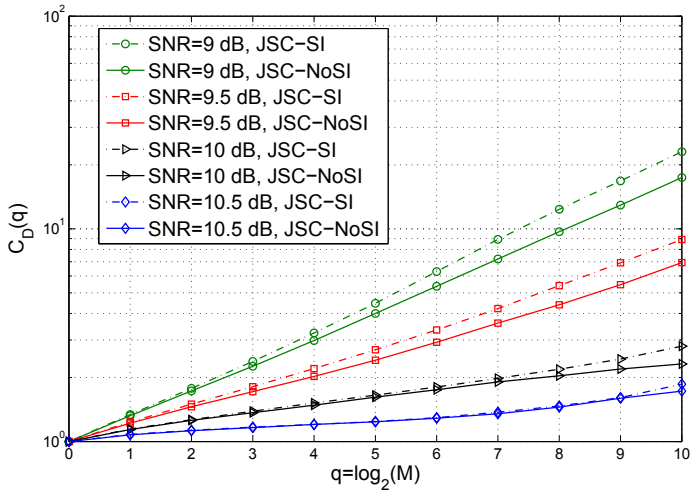


Figure 4.6.: Relative decoding complexity $C_{\mathrm{D}}\left(q\right)$ of the JSC-SI and the JSC-NoSI decoding schemes as function of $q = \log_2 M$, for different channel SNRs.

Figure 4.6 presents $C_{\mathrm{D}}\left(q_{\max}\right)$ with $q_{\max} = \log_2 M$ for the JSC-SI and JSC-NoSI schemes and for different values of the channel SNR. As in the case of SD coding, the complexity increases when the channel SNR decreases, since the number of

erroneous 3D blocks increases. Moreover, the complexity increases with the use of the SI since the detection rate of corrupted blocks is more important. With a channel SNR above 10 dB, the JSC decoding complexity is less than three times the decoding complexity of a standard decoder. With an SNR above 10.5 dB, it is less than twice.

The performance of JSCD obtained for the considered MD scheme is similar to the one obtained in the SD case. As discussed in the Chapter 2, the aim of using an MD scheme is to enhance the robustness of the transmitted data to transmission errors and/or to channel erasures. The price of such a robustness is an increase of the total bit rate due to redundancy introduction between the descriptions.

In the next section, the MD scheme proposed in Section 4.3 is compared to a classical SD coding scheme introducing the same amount of redundancy.

4.5.3. Comparison to an SD Scheme Combined with a FEC

In this section, the proposed JSC coding schemes are compared to an SD coding scheme introducing the same amount of redundanct. The PSNR as function of the coding rate for the SD scheme and MD schemes (without and with the use of the side information) is depicted in Figure 4.7 in absence of channel impairments (errors and losses).

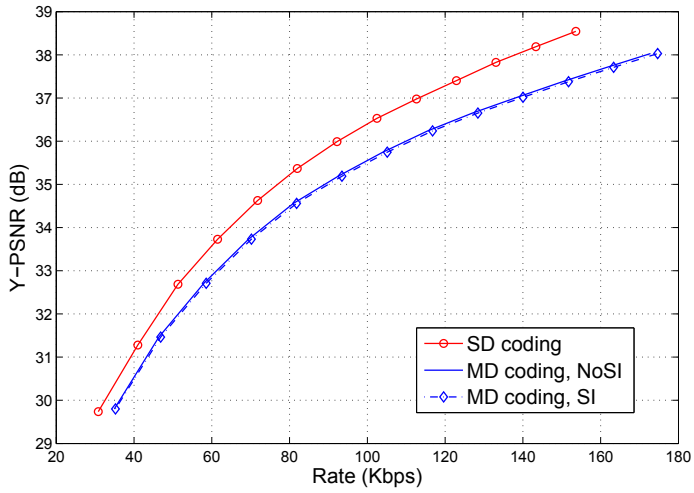


Figure 4.7.: Y-PSNR (dB) as function of the coding rate (kbps) for the SD coding scheme and the MD coding scheme with and without the use of the additional side information (SI).

Let $R^{\mathrm{d}}_{\mathrm{MD}}$ and $R^{\mathrm{d}}_{\mathrm{SD}}$ be the bit rates corresponding to the transmission of the texture (payloads of the entropy-coded blocks) of the MD and SD coding schemes. For an equivalent reconstruction quality, one has $R^{\mathrm{d}}_{\mathrm{MD}} \geqslant R^{\mathrm{d}}_{\mathrm{SD}}$. To get an equal bit rate, the

rate of the FEC to be used to protect the data generated by the SD coding scheme is equal to

$$\eta_{\text{NoSI}} = \frac{R^{\text{d}}_{\text{MD}}}{R^{\text{d}}_{\text{SD}}}, \tag{4.27}$$

without the use of the additional side information. When this information is considered, the rate of the FEC to be used is

$$\eta_{\text{SI}} = \frac{3\beta(N_1 + N_2)R_{\text{f}}/N_{\text{f}} + R^{\text{d}}_{\text{MD}}}{R^{\text{d}}_{\text{SD}}}, \tag{4.28}$$

where $3\beta(N_1 + N_2)R_{\text{f}}/N_{\text{f}}$ is the rate due to the three additional bits stored in the headers to use the test $t^{\text{b}}$and β accounts for the redundancy that may be introduced to protect the headers of the 3D block.

In what follows, $\beta = 1.2$. Packet-erasures FECs [134, 15, 142], bit-error correcting FECs or a combination of both FECs may be associated to the SD coding scheme. Here, we choose to focus on bit-error correcting FECs implemented using Reed Solomon codes. The JSC multiple description coding schemes are compared to an SD coding scheme followed by a Reed Solomon code and denoted by SD-FEC.

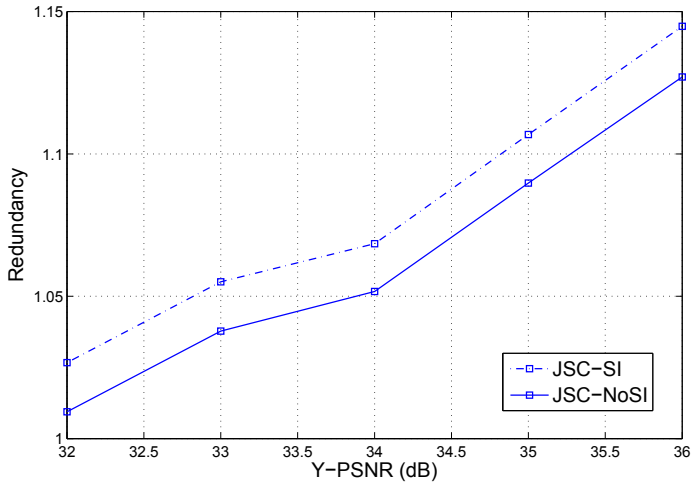


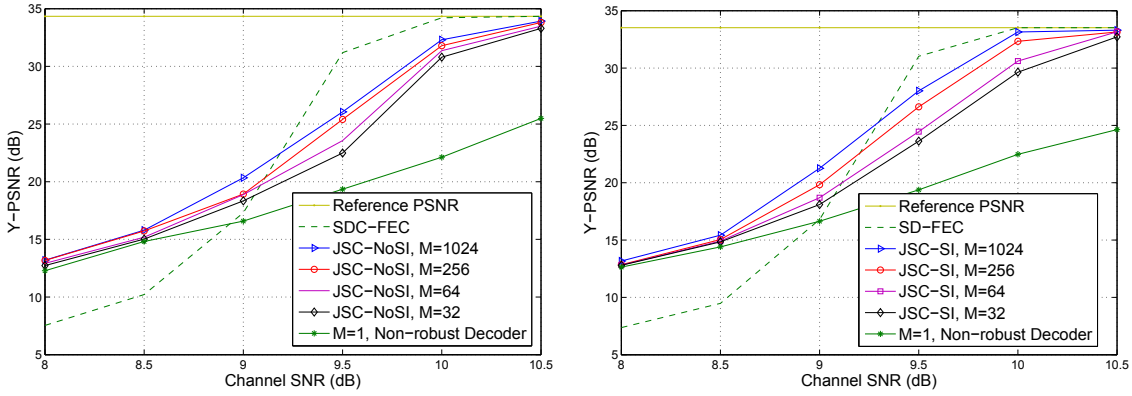
Figure 4.8.: Redundancy η_{NoSI} and η_{SI} as function of the Y-PSNR (dB).

The values of η_{NoSI} and η_{SI} are shown in Figure 4.8, as function of the reconstruction PSNR. As one can see, the redundancy increases with the PSNR. The performance comparison is done considering the same amount of redundancy for the same reconstruction PSNR between SD-FEC and JSC-SI, or between SD-FEC and JSC-NoSI. For JSC-NoSI, a redundancy $\eta_{\text{NoSI}} = 1.06$ is obtained at 78 kbps with a PSNR of 34.34 dB, whereas $\eta_{\text{SI}} = 1.06$ is obtained for JSC-SI at 67.5 kbps leading to a PSNR of 33.5 dB. Moreover for JSC-NoSI, a redundancy $\eta_{\text{NoSI}} = 1.04$ is obtained at 62 kbps with a PSNR of 33 dB, whereas $\eta_{\text{SI}} = 1.04$ is obtained for JSC-SI at 55.5 kbps leading to a PSNR of 32.4 dB.

In a first set of experiments, the wired network is assumed to be lossless, *i.e*, impairments are only introduced by the wireless channel. In a second set of experiments, packet losses as well as channel impairments are considered.

4.5.3.1. Lossless Wired Network

Here again, the wired part of the network is assumed lossless. For various channel SNRs, Figure 4.9 compares the PSNR obtained with SD-FEC and JSC-NoSI for $\eta_{\text{NoSI}} = 1.06$. It also compares SD-FEC and JSC-SI for $\eta_{\text{SI}} = 1.06$. The JSC-NoSI performs better than SD-FEC for a channel SNR less than 9.15 dB while JSC-SI is better than SD-FEC when the channel SNR is less than 9.3 dB. The degradation in PSNR is quite abrupt with the SD-FEC scheme, and smoother with both JSC schemes.



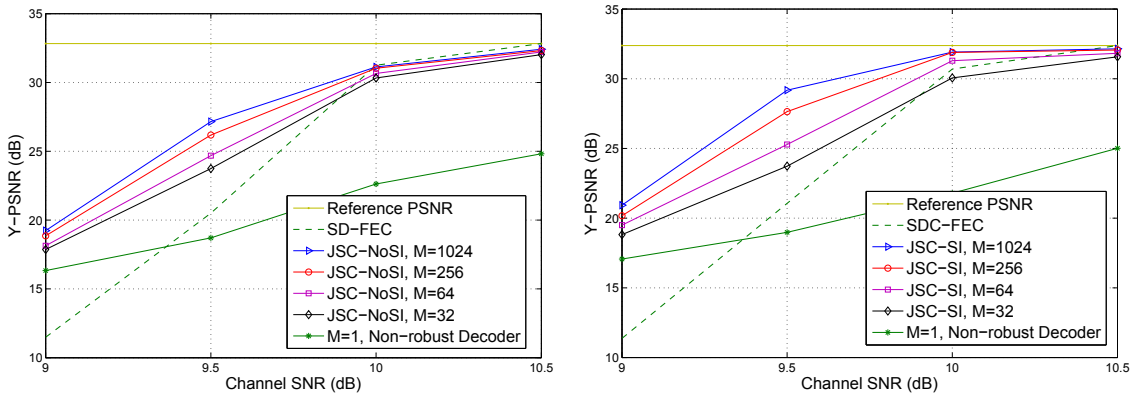
(a) JSC-NoSI (a) JSC-SI

Figure 4.9.: Y-PSNR (dB) as function of the channel SNR (dB) for SD-FEC, JSC-NoSI and JSC-SI with $\eta_{\text{NoSI}} = 1.06$.

Figure 4.10 compares the PSNR obtained with SD-FEC and the JSC schemes, as function of the channel SNR when $\eta_{\text{NoSI}} = \eta_{\text{SI}} = 1.04$. With this smaller redundancy, JSC-NoSI performs better than SD-FEC for a channel SNR less than 10 dB. The JSC-SI scheme performs better than the SD-FEC one for a channel SNR less than 10.4 dB. At a channel SNR of 9.5 dB, a gain in PSNR of about 7 dB is obtained with JSC-NoSI and about 8 dB with JSC-SI.

4.5.3.2. Lossy Wired Network

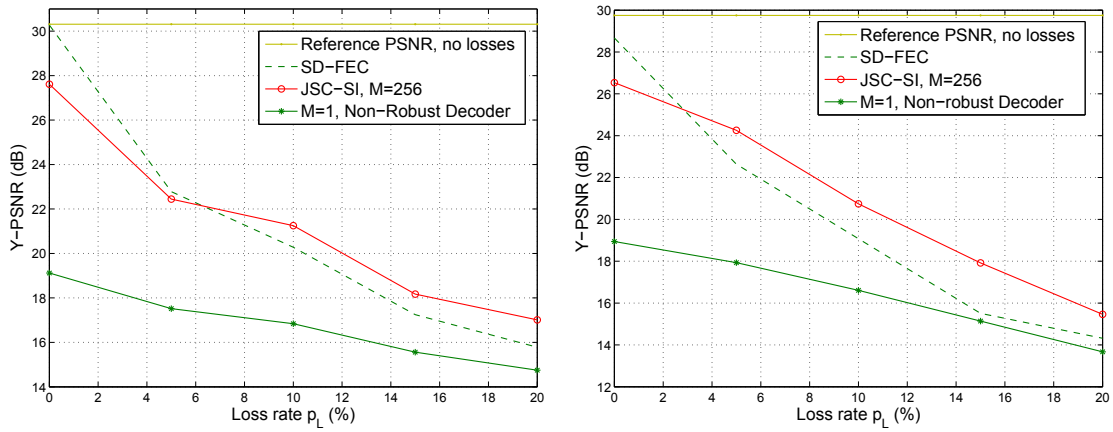
Now, the wired part of the network introduces packet losses. The first $N_{\text{f}} = 256$ frames of `foreman.qcif` are encoded at the same frame rate $R_{\text{f}} = 15$ fps. The wired part of the network is modeled by a memoryless packet erasure channel with loss probability p_{L}. The wireless part is still modeled as an AWGN channel. At the



(a) JSC-NoSI (b) JSC-SI

Figure 4.10.: Y-PSNR (dB) as function of the channel SNR (dB) for SD-FEC, JSC-NoSI and JSC-SI with $\eta_{\text{NoSI}} = \eta_{\text{SI}} = 1.04$.

decoder side, the approach described in Section 4.4 is adopted, in order to mitigate the effect of packet losses in the JSC case. For the SD case, lost data are replaced by zeros. Moreover, for the JSC decoding schemes, $M = 256$ is considered.



(a) $\eta_{\text{SI}} = 1.06$ (b) $\eta_{\text{SI}} = 1.04$

Figure 4.11.: Y-PSNR (dB) as function of p_{L} for SDC-FEC and JSC-SI for a channel SNR of 10 dB.

Figure 4.11 shows the performance of SD-FEC and JSC-SI in terms of reconstruction PSNR as function of p_{L} for $\eta_{\text{SI}} = 1.06$ and for $\eta_{\text{SI}} = 1.04$. Without losses, SD-FEC outperforms JSC-SI. However, with $p_{\text{L}} \geqslant 5$ % (resp $p_{\text{L}} \geqslant 3$ %), JSC-SI outperforms SD-FEC by up to 1 dB (resp. 2 dB) for $\eta_{\text{SI}} = 1.06$ (resp.$\eta_{\text{SI}} = 1.04$). The difference is mainly due to the compensation of the lost 3D blocks in one description by the other description, and by the use of the JSC decoding scheme in the MD schemes.

4.6. Conclusion

In this chapter, we have applied joint source channel decoding exploiting the source residual redundancy in a multiple description coding context. The aim was to present a typical application of such a scheme, *e.g.*, video broadcasting over mixed architectures, in which retransmission of the lost data is not allowed. The redundancy introduced between the descriptions is exploited to compensate packet losses, whereas the joint decoding allows to increase the number of error-free packets reaching the multiple description decoder, and thus to improve its performance.

In the considered multiple description scheme, the redundancy is introduced by using an overcomplete temporal decomposition which operates an expansion of the input video signal to generate the two descriptions. The linear relations between the entries of the descriptions may be exploited to detect and correct some of the transmission errors. However, this redundancy is only exploitable after entropy decoding and inverse spatial transform. This makes it difficult to use this redundancy directly from the channel soft outputs. Moreover due to motion estimation and compensation, a single entry in one of the descriptions may be involved in several linear relations making the error detection and correction particularly difficult.

In the next chapter, we study, in a more theoretical context, schemes operating such a signal expansion. The aim is to explore the way the introduced redundancy could be exploited to detect and correct transmission errors in a joint source channel coding framework.

5. Robust Estimation from Noisy Overcomplete Signal Representation

In Chapter 4, multiple description coding based on a redundant transform in the temporal domain has been combined with joint source-channel decoding techniques in order to enhance the robustness of transmitted video content to transmission errors and losses. Such signal expansion introduces some redundancy within the data to be transmitted, which may be efficiently exploited to detect and correct the transmission errors.

In this chapter, the general problem of estimating the input signal of some redundant linear transform is addressed and two efficient estimation approaches are proposed. These estimation schemes are then applied to oversampled filter banks used in image coding applications and a comparison of their performance is provided.

5.1. Introduction

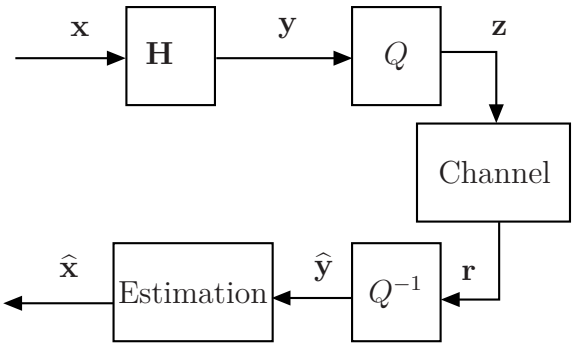


Figure 5.1.: Considered transmission scheme.

The considered transmission scheme is presented in Figure 5.1. The vector $\mathbf{x} \in \mathbb{R}^N$ to be transmitted is assumed to be the realization of some random (yet correlated in the case of video sequences) vector $\mathbf{X}$. This vector goes through an $M \times N$

transform matrix $\mathbf{H}$. The resulting vector $\mathbf{y} = \mathbf{H}\mathbf{x}$ is first quantized to get the vector of quantization indexes $\mathbf{z}$, then transmitted through some noisy channel.

The problem one is faced with is to estimate the input vector $\mathbf{x}$. Classical approaches handle this problem by using the hard decisions of the channel decoder, then by performing a reconstruction which minimizes a given metric, *e.g.*, the mean squared error. In the particular expansion case ($M > N$), these approaches are not efficient, since they do not fully exploit the redundancy introduced in the output vector $\mathbf{y}$, *i.e.*, the linear dependencies between the entries of $\mathbf{y}$ are not taken into account during the estimation of $\mathbf{x}$, in order to detect, *e.g.*, the transmission errors.

As discussed in the previous chapters, the redundancy present in $\mathbf{y}$ may efficiently be exploited at the receiver side to perform joint source-channel decoding (JSCD). To implement JSCD at the application layer, we assume as in Chapters 3 and 4 that permeable protocol layers [75, 42] and robust header recovery mechanisms [101] are implemented, so that the soft information coming from the channel decoders at physical layer reaches the application layer. The channel represents then all what is between the quantization output and the inverse quantization input, including the entropy coding, the network packetization and the physical channel, as well as the corresponding robust depacketization and entropy decoding modules [42]. At the output of this channel, the soft information $\mathbf{r}$ on the transmitted bits is obtained.

In this chapter, data redundancy is introduced by the mean of a redundant linear transform. A first direction to exploit such a *structured* redundancy between the samples of $\mathbf{y}$ is to perform a *consistent estimation* of the input $\mathbf{x}$, *i.e.*, an estimation in the subspace of the vectors that could have been obtained from the expansion matrix $\mathbf{H}$. A second proposed direction to estimate $\mathbf{x}$ considers the estimation problem illustrated in Figure 5.1 as a *linear mixing problem*, which may be handled using belief propagation [120] in order to evaluate the marginal posterior probability distribution of each entry of $\mathbf{x}$ knowing $\mathbf{r}$ [128, 129].

In the sequel, we present the linear mixing problem in Section 5.2. Then the optimal estimator is derived in Section 5.2.1. The proposed reconstruction approaches, based on consistent estimation and belief propagation are described in Sections 5.3 and 5.4. Finally, the application to oversampled filter banks is presented in Section 5.6.

5.2. Problem Formulation

Let $p_{\mathbf{X}}(\mathbf{x})$ be the probability density function (PDF) of the random vector $\mathbf{X} = (X_1, \ldots, X_M)^T$. The vector $\mathbf{y}$ obtained at the output of $\mathbf{H}$ is a realization of the random vector $\mathbf{Y} = (Y_1, \ldots, Y_M)^T$. Along this chapter, the study is limited to the case of an expansion ($M > N$), *i.e.*, the output vector

$$\mathbf{y} = \mathbf{H}\mathbf{x}, \tag{5.1}$$

contains more samples than the input vector $\mathbf{x}$ and the $M \times N$ matrix $\mathbf{H}$ is thus a *tall* matrix. Moreover, this matrix is assumed to be of full rank N. Each component y_m, $m = 1, \ldots, M$ of $\mathbf{y}$ is quantized with a scalar quantizer Q of rate ρ. The quantization intervals $[s(0)], \ldots, [s(2^\rho - 1)]$ form a partition of $\mathbb{R}$.

After quantization, one gets a vector $\mathbf{z}$ of quantization indexes, which is a realization of the random vector $\mathbf{Z} = (Z_1, \ldots, Z_M)^T$ with

$$Z_m = Q(Y_m) = z \Leftrightarrow Y_m \in [s(z)] = [\underline{s}(z), \overline{s}(z)], \; z \in \{0, \ldots, 2^\rho - 1\}, \quad (5.2)$$

where for a given quantization index $z \in \{0, \ldots, 2^\rho - 1\}$, $\underline{s}(z)$ and $\overline{s}(z)$ denote the lower and upper bounds of the interval $[s(z)]$ respectively.

The quantization indexes z_m are then transmitted through some noisy channel, as described in Section 5.1. This channel is assumed to be memoryless, property which may be ensured by employing proper interleavers prior to channel coding. The channel output related to z_m is assumed to be described by the vector $\mathbf{r}_m \in \mathbb{R}^\rho$ (or $\mathbb{C}^\rho$), $m = 1, \ldots, M$. The effect of the transmission channel is then described by the channel transition probability $p_{\mathbf{R}|Z}(\mathbf{r}|z)$.

The considered problem consists in the evaluation of the MAP estimate of $\mathbf{x}$

$$\widehat{\mathbf{x}}_{\text{MAP}} = \arg \max_{\mathbf{x} \in \mathbb{R}^N} p_{\mathbf{X}|\mathbf{R}}(\mathbf{x}|\mathbf{r}_{1:M}). \quad (5.3)$$

In the sequel, for a set of k vectors $\mathbf{a}_1, \ldots, \mathbf{a}_k$, the following notation is used $\mathbf{a}_{1:k} = \left(\mathbf{a}_1^T, \ldots, \mathbf{a}_k^T\right)^T$.

5.2.1. Optimal MAP Estimator

In this part, we propose to derive the optimal estimator for (5.3). Using Bayes' rule, one may rewrite (5.3) as

$$\widehat{\mathbf{x}}_{\text{MAP}} = \arg \max_{\mathbf{x} \in \mathbb{R}^N} p_{\mathbf{R}|\mathbf{X}}(\mathbf{r}_{1:M}|\mathbf{x}) \, p_{\mathbf{X}}(\mathbf{x}), \quad (5.4)$$

where

$$\begin{aligned} p_{\mathbf{R}|\mathbf{X}}(\mathbf{r}_{1:M}|\mathbf{x}) &= \sum_{\mathbf{z}_{1:M}} p_{\mathbf{R},\mathbf{Z}|\mathbf{X}}(\mathbf{r}_{1:M}, \mathbf{z}_{1:M}|\mathbf{x}) \\ &= \sum_{\mathbf{z}_{1:M}} p_{\mathbf{R}|\mathbf{Z},\mathbf{X}}(\mathbf{r}_{1:M}|\mathbf{z}_{1:M}, \mathbf{x}) \, p_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}_{1:M}|\mathbf{x}). \end{aligned} \quad (5.5)$$

In (5.5), once $z_1, \ldots, z_M$ are known, $\mathbf{x}$ does not bring additional knowledge on $\mathbf{r}_1, \ldots, \mathbf{r}_M$, and since the channel is memoryless, one gets

$$p_{\mathbf{R}|\mathbf{Z},\mathbf{X}}(\mathbf{r}_{1:M}|\mathbf{z}_{1:M}, \mathbf{x}) = \prod_{m=1}^{M} p_{\mathbf{R}|Z}(\mathbf{r}_m|z_m). \quad (5.6)$$

Moreover, since a scalar quantization is considered, one has

$$p_{\mathbf{Z}|\mathbf{X}}\left(\mathbf{z}_{1:M}|\mathbf{x}\right) = p_{\mathbf{Z}|\mathbf{Y}}\left(\mathbf{z}_{1:M}|\mathbf{H}\mathbf{x}\right) = \prod_{m=1}^{M} p_{Z|Y}\left(z_m|\mathbf{h}_m^T\mathbf{x}\right), \tag{5.7}$$

where $\mathbf{h}_m^T$ is the m-th row of $\mathbf{H}$. According to (5.2), one has

$$p_{Z|Y}\left(z_m|\mathbf{h}_m^T\mathbf{x}\right) = I_{[s(z_m)]}\left(\mathbf{h}_m^T\mathbf{x}\right), \tag{5.8}$$

where the indicator function is

$$I_{[s]}\left(x\right) = \begin{cases} 1 & \text{if } x \in [s], \\ 0 & \text{otherwise.} \end{cases} \tag{5.9}$$

Finally, combining (5.5)–(5.9) in (5.4), one obtains

$$\widehat{\mathbf{x}}_{\text{MAP}} = \arg\max_{\mathbf{x}\in\mathbb{R}^N} p_{\mathbf{X}}\left(\mathbf{x}\right) \sum_{\mathbf{z}_{1:M}} \prod_{m=1}^{M} p_{\mathbf{R}|Z}\left(\mathbf{r}_m|z_m\right) I_{[s(z_m)]}\left(\mathbf{h}_m^T\mathbf{x}\right). \tag{5.10}$$

For a given $\mathbf{x} \in \mathbb{R}^N$, by definition of the indicator function and due to the fact that the quantization intervals form a partition of $\mathbb{R}$, the sum over $\mathbf{z}_{1:M}$ consists of a single non-zero term when $z_m = Q\left(\mathbf{h}_m^T\mathbf{x}\right)$, $m = 1, \ldots, M$. Thus (5.10) becomes

$$\widehat{\mathbf{x}}_{\text{MAP}} = \arg\max_{\mathbf{x}\in\mathbb{R}^N} p_{\mathbf{X}}\left(\mathbf{x}\right) \prod_{m=1}^{M} p_{\mathbf{R}|Z}\left(\mathbf{r}_m|Q\left(\mathbf{h}_m^T\mathbf{x}\right)\right). \tag{5.11}$$

Obtaining $\widehat{\mathbf{x}}_{\text{MAP}}$ requires the maximization of the function

$$f\left(\mathbf{x}, \mathbf{r}_{1:M}\right) = p_{\mathbf{X}}\left(\mathbf{x}\right) \prod_{m=1}^{M} p_{\mathbf{R}|Z}\left(\mathbf{r}_m|Q\left(\mathbf{h}_m^T\mathbf{x}\right)\right) \tag{5.12}$$

over all possible $\mathbf{x} \in \mathbb{R}^N$. The evaluation of $f\left(\mathbf{x}, \mathbf{r}_{1:M}\right)$ is easy, however, due to the quantization, this function is only piecewise-continuous, which makes its maximization difficult, especially when N is large.

5.2.2. Running Example

In this example, $M = 3$ and $N = 2$. Consider a vector $\mathbf{x} \in \mathbb{R}^N$ such as $p_X(x_n) \sim \mathcal{N}(0, 1)$, $n = 1, \ldots, N$ and an $M \times N$ transform matrix $\mathbf{H}$ defined as follows

$$\mathbf{x} = \begin{pmatrix} -0.62 \\ 1.42 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix}.$$

The components of the vector $\mathbf{y} = \begin{pmatrix} -0.62, 1.54, -0.92 \end{pmatrix}^T$ obtained at the output of $\mathbf{H}$ are quantized using a scalar quantizer Q of step $\Delta = 1.03$ and rate $\rho = 2$ bits. The resulting vector $\mathbf{z} = (1, 3, 1)^T$ of quantization indexes is then modulated using the binary phase-shift keying (BPSK) modulation, before being transmitted through an AWGN channel with an SNR level of 2 dB. The observation at the channel output is

$$\mathbf{r}_{1:M} = \begin{pmatrix} -0.65 & -0.97 \\ -1.01 & -1.44 \\ 0.07 & -0.92 \end{pmatrix}. \tag{5.13}$$

The aim is to estimate $\mathbf{x}$ from $\mathbf{r}_{1:M}$. First, the estimator obtained from the channel's hard decisions is presented. Then the brute-force estimation scheme using the channel's soft information is described.

Estimation using the hard decisions of the channel

Using the hard decisions (HD) of the channel, the vector $\widehat{\mathbf{z}}_{\text{HD}} = (3, 3, 1)^T$ is obtained at the channel output. As one can see, the first component of $\mathbf{z}$ has been corrupted. After inverse quantization, the vector $\widehat{\mathbf{y}}_{\text{HD}} = Q^{-1}(\widehat{\mathbf{z}}_{\text{HD}}) = (1.54, 1.54, -0.51)^T$ is used to perform least squares estimation

$$\widehat{\mathbf{x}}_{\text{HD}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\widehat{\mathbf{y}}_{\text{HD}} = (0.68, 1.19)^T.$$

The mean squared error (MSE) obtained using this estimator is $e_{\text{HD}} = 0.8715$.

Brute-force estimation using the soft information of the channel

The function $f(\mathbf{x}, \mathbf{r}_{1:M})$ expressed in (5.12) is numerically evaluated using discretized PDFs, in order to obtain the values for $p_X(x_n)$, $n = 1, \ldots, N$ and for the product $\prod_{m=1}^{M} p_{\mathbf{R}|Z}(\mathbf{r}_m|Q(\mathbf{h}_m^T\mathbf{x}))$. The range considered for the random variables X_n is from -10 to 10, and the number of points on which these PDFs are evaluated is equal to 1024. One obtains then a 1024×1024 matrix representing $f(\mathbf{x}, \mathbf{r}_{1:M})$. A 2-D plot of the level curves of this matrix is presented in Figure 5.2. As one can see, the obtained $f(\mathbf{x}, \mathbf{r}_{1:M})$ is discontinuous.

The estimate $\widehat{\mathbf{x}}_{\text{MAP}}$ is obtained by choosing the vector corresponding to the maximal value in the matrix representing $f(\mathbf{x}, \mathbf{r}_{1:M})$. One obtains $\widehat{\mathbf{x}}_{\text{MAP}} = \begin{pmatrix} -1.00, 1.73 \end{pmatrix}^T$. The MSE obtained using this estimator is $e_{\text{MAP}} = 0.1203$.

Implementing this brute-force MAP estimation is very complex for high dimensional input vectors $\mathbf{x}$ and for dense matrices $\mathbf{H}$.

In the two next sections, two different suboptimal estimation approaches are presented. The first one uses linear programming and tools from interval analysis [73] to perform a consistent estimation of $\mathbf{x}$, whereas the second one uses belief propagation [120] to handle the MAP estimation problem in (5.3).

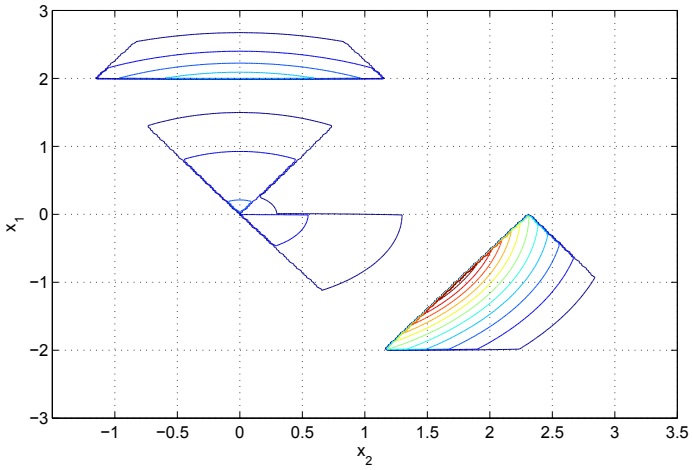


Figure 5.2.: Contour plot of $f(\mathbf{x}, \mathbf{r}_{1:3})$ for $\mathbf{r}_{1:3}$ defined in (5.13) .

5.3. Consistent Estimation of $\mathbf{x}$

In this section, the redundancy introduced by $\mathbf{H}$ is exploited and the *bounded* quantization noise is explicitly taken into account. A suboptimal implementation of the MAP estimator presented in Section 5.2.1 is derived by performing the estimation in the subspace of all the *consistent* quantization indexes, *i.e*, indexes that can result from the quantization of a vector $\mathbf{y}$ which may be generated at the output of the expansion matrix $\mathbf{H}$.

First, in Section 5.3.1, we propose to perform the estimation of $\mathbf{x}$ in a particular case, where the noise introduced by the channel is negligible. The estimation approach, in the general case, is described in Section 5.3.2.

5.3.1. Negligible Channel Noise

Assume that the noise introduced by the channel is very small. It is likely that $p_{\mathbf{R}|Z}(\mathbf{r}_m|z_m)$ will be vanishing for all values of z_m, except for some $z_m^* \in \{0, \ldots, 2^\rho - 1\}$, which is very likely to correspond to the actual transmitted quantization index $Q(y_m)$. In this case, when considering in (5.10) the sum over $\mathbf{z}_{1:M}$, only the term corresponding to $\mathbf{z}_{1:M}^*$ may be kept and one gets

$$\widehat{\mathbf{x}}_{\text{MAP}} = \arg\max_{\mathbf{x}\in\mathbb{R}^N} \; p_{\mathbf{X}}(\mathbf{x}) \prod_{m=1}^{M} p_{\mathbf{R}|Z}(\mathbf{r}_m|z_m^*)\, I_{[s(z_m^*)]}\left(\mathbf{h}_m^T\mathbf{x}\right). \tag{5.14}$$

Now, the product $\prod_{m=1}^{M} p_{\mathbf{R}|Z}(\mathbf{r}_m|z_m^*)$ does not depend any more on $\mathbf{x}$ and (5.14) becomes the following constrained optimization problem

$$\begin{aligned} \widehat{\mathbf{x}}_{\text{MAP}} &= \arg\max_{\mathbf{x}\in\mathbb{R}^N} p_{\mathbf{X}}(\mathbf{x}), \\ \text{st } \mathbf{h}_m^T\mathbf{x} &\in [s(z_m^*)], \; m = 1, \ldots, M \end{aligned} \tag{5.15}$$

which may be written with linear constraints as follows

$$\begin{aligned} \widehat{\mathbf{x}}_{\mathrm{MAP}} &= \arg \max_{\mathbf{x} \in \mathbb{R}^N} p_{\mathbf{X}}(\mathbf{x}), \\ \text{st } \mathbf{H}\mathbf{x} &\leqslant \overline{\mathbf{s}}(\mathbf{z}^*_{1:M}) \\ -\mathbf{H}\mathbf{x} &\leqslant -\underline{\mathbf{s}}(\mathbf{z}^*_{1:M}) \end{aligned} \tag{5.16}$$

where, for a given $\mathbf{z}_{1:M} \in \{0, \ldots, 2^\rho - 1\}^M$, $\overline{\mathbf{s}}(\mathbf{z}_{1:M}) = (\overline{s}(z_1), \ldots, \overline{s}(z_M))^T$ and $\underline{\mathbf{s}}(\mathbf{z}_{1:M}) = (\underline{s}(z_1), \ldots, \underline{s}(z_M))^T$.

5.3.1.1. Running Example

Let us consider the example presented in Section 5.2.2, where $N = 2$, $M = 3$ and

$$\mathbf{x} = \begin{pmatrix} -0.62 \\ 1.42 \end{pmatrix}, \ \mathbf{H} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix}.$$

The vector of quantization indexes $\mathbf{z} = (1, 3, 1)^T$ is BPSK modulated, then transmitted through an AWGN channel introducing a negligible noise, *e.g.*, with and SNR equal to 12 dB. The observation at the channel output is

$$\mathbf{r}_{1:3} = \begin{pmatrix} -0.97 & -1.20 \\ -0.92 & -1.33 \\ 1.17 & -0.59 \end{pmatrix}.$$

Table 5.1 shows the values of $p_{\mathbf{R}|Z}(\mathbf{r}_m|z_m = z)$ for $m = 1, 2, 3$ and $z = \{0, \ldots, 3\}$. The vector of quantization indexes $\mathbf{z}^* = (1, 3, 1)^T$, maximizing $\prod_{m=1}^3 p_{\mathbf{R}|Z}(\mathbf{r}_m|z_m)$, is equal to the transmitted vector $\mathbf{z}$.

$p(\mathbf{r}_m\|z_m)$	$z_m = 0$	$z_m = 1$	$z_m = 2$	$z_m = 3$
$m = 1$	0.00	1.00	0.00	0.00
$m = 2$	0.00	0.00	0.00	1.000
$m = 3$	0.00	1.00	0.00	0.00

Table 5.1.: Obtained values for $p(\mathbf{r}_m|z_m)$.

Finally, one selects $\mathbf{z}^* = (1, 3, 1)^T$ to solve (5.16), where instead of maximizing $p_{\mathbf{X}}(\mathbf{x})$ one minimizes $\mathbf{x}^T\mathbf{x}$, since $\mathbf{X} \sim \mathcal{N}(0, \mathbf{I}_2)$. One gets a simple quadratic problem with linear constraints. The obtained solution is $\widehat{\mathbf{x}}_{\mathrm{MAP}} = (-0.51, 0.89)^T$, which is reasonably close to $\mathbf{x} = (-0.62, 1.42)^T$.

5.3.2. General Case

Let us now assume that the noise introduced by the channel is no longer negligible. The vector $\mathbf{z}^*$ maximizing $p_{\mathbf{R}|\mathbf{Z}}(\mathbf{r}_{1:M}|\mathbf{z}_{1:M})$ may not correspond to the vector of transmitted indexes $Q(\mathbf{y}_{1:M})$. It may even not correspond to a vector of quantization indexes that may be obtained from a given $\mathbf{Hx}$, due to the relations between the entries of $\mathbf{Hx}$.

The proposed suboptimal implementation of (5.3) is to keep in (5.10), in the sum over $\mathbf{z}_{1:M}$, only the vector $\mathbf{z}^{\mathrm{f}}_{1:M}$ maximizing $p_{\mathbf{R}|\mathbf{Z}}(\mathbf{r}_{1:M}|\mathbf{z}_{1:M})$ and which is *consistent* (or feasible), *i.e.*, for which there exists some $\mathbf{x} \in \mathbb{R}^N$ satisfying $Q(\mathbf{Hx}) = \mathbf{z}^{\mathrm{f}}_{1:M}$. Then (5.16) is solved with $\mathbf{z}^{\mathrm{f}}_{1:M}$.

Let

$$\mathcal{Z} = \left\{ \mathbf{z} \in \{0, \ldots, 2^\rho - 1\}^M \mid \exists \mathbf{x} \in \mathbb{R}^N,\ \mathbf{z} = Q(\mathbf{Hx}) \right\} \tag{5.17}$$

be the set of consistent vectors of quantization indexes. The algorithm performing the consistent MAP estimation is described in the sequel.

5.3.2.1. Consistent MAP Estimation Algorithm

The proposed consistent MAP estimation algorithm has two steps:

1. Evaluate

$$\mathbf{z}^{\mathrm{f}}_{1:M} = \arg\max_{\mathbf{z} \in \mathcal{Z}} \prod_{m=1}^{M} p_{\mathbf{R}|Z}(\mathbf{r}_m|z_m). \tag{5.18}$$

2. Solve

$$\begin{aligned} \widehat{\mathbf{x}}_{\mathrm{MAP}} = \arg\max_{\mathbf{x}}\ & p_{\mathbf{X}}(\mathbf{x}). \\ \mathrm{st}\ & \mathbf{Hx} \leqslant \overline{\mathbf{s}}\left(\mathbf{z}^{\mathrm{f}}_{1:M}\right) \\ & -\mathbf{Hx} \leqslant -\underline{\mathbf{s}}\left(\mathbf{z}^{\mathrm{f}}_{1:M}\right) \end{aligned} \tag{5.19}$$

The first step of this algorithm may be performed as follows

1. Sort the values of $\mathbf{z}_{1:M} \in \{0, \ldots, 2^\rho - 1\}^M$, in decreasing order of $p_{\mathbf{R}|\mathbf{Z}}(\mathbf{r}_{1:M}|\mathbf{z}_{1:M})$, in a list $\mathcal{L}_{\mathbf{z}} = \left\{\mathbf{z}^{(1)}_{1:M}, \ldots, \mathbf{z}^{(L)}_{1:M}\right\}$, with $L = 2^{\rho M}$. Thus

$$p_{\mathbf{R}|\mathbf{Z}}\left(\mathbf{r}_{1:M}|\mathbf{z}^{(\ell_1)}_{1:M}\right) \geqslant p_{\mathbf{R}|\mathbf{Z}}\left(\mathbf{r}_{1:M}|\mathbf{z}^{(\ell_2)}_{1:M}\right), \tag{5.20}$$

   if $\ell_1 < \ell_2$, with $\ell_1, \ell_2 = 1, \ldots, L$.
2. $\ell = 1$
3. Do

a) If $\mathbf{z}^{(\ell)} \in \mathcal{Z}$

b) $\mathbf{z}^{\mathrm{f}} = \mathbf{z}^{(\ell)}$; End;

c) Else $\ell = \ell + 1$;

4. While $\ell \leqslant L$.

Our aim now is to describe how Step 3 (a) may be performed. Consider the set

$$\mathcal{X}(\mathbf{z}) = \left\{ \mathbf{x} \in \mathbb{R}^N \mid \mathbf{z} = Q(\mathbf{H}\mathbf{x}) \right\} \tag{5.21}$$

containing all the vectors $\mathbf{x}$ from which the quantization vector $\mathbf{z}$ may be obtained. Alternatively, this set may be defined as follows

$$\mathcal{X}(\mathbf{z}) = \bigcap_{m=1}^{M} \mathcal{X}_m(z_m), \tag{5.22}$$

where, for a given $z \in \{0, \ldots, 2^\rho - 1\}$, $\mathcal{X}_m(z) = \left\{ \mathbf{x} \in \mathbb{R}^N \mid z = Q\left(\mathbf{h}_m^T \mathbf{x}\right) \right\}$.

By definitions (5.17) and (5.22), one has

$$\mathcal{X}(\mathbf{z}) \neq \emptyset \Longleftrightarrow \mathbf{z} \in \mathcal{Z}. \tag{5.23}$$

For a given $\ell = 1, \ldots, L$, in order to determine whether $\mathbf{z}^{(\ell)} \in \mathcal{Z}$, or equivalently, whether $\mathcal{X}\left(\mathbf{z}^{(\ell)}\right) \neq \emptyset$, one may use, *e.g.*, linear programming [8] or interval analysis tools [73].

5.3.2.2. Consistency Test using Linear Programming

For a given $\ell = 1, \ldots, L$, consider the maximization problem

$$\max_{\mathbf{x} \in \mathcal{X}\left(\mathbf{z}_{1:M}^{(\ell)}\right)} \mathbf{a}^T \mathbf{x},$$

$\mathbf{a}$ being any non-zero vector of size N. This is equivalent to the following linear problem

$$\begin{aligned} \max_{\mathbf{x} \in \mathbb{R}^N} \ & \mathbf{a}^T \mathbf{x}. \\ \text{st } & \mathbf{H}\mathbf{x} \leqslant \overline{\mathbf{s}}\left(\mathbf{z}_{1:M}^{(\ell)}\right) \\ & -\mathbf{H}\mathbf{x} \leqslant -\underline{\mathbf{s}}\left(\mathbf{z}_{1:M}^{(\ell)}\right) \end{aligned} \tag{5.24}$$

If a solution $\mathbf{x}$ is found for (5.24), then $\mathcal{X}\left(\mathbf{z}^{(\ell)}\right) \neq \emptyset$ and $\mathbf{z}^{(\ell)}$ is consistent.

Note that one could try to directly solve (5.16) with $\mathbf{z}_{1:M}^{(\ell)}$, starting from $\ell = 1$ and increasing ℓ until a solution exists. Nevertheless, solving first (5.24) for various $\mathbf{z}_{1:M}^{(\ell)}$ with increasing ℓ is much less complex than solving (5.16), especially if $p_{\mathbf{X}}(\mathbf{x})$ is a general PDF.

5.3.2.3. Consistency Test using Interval Analysis

Another approach to determine whether a quantization vector $\mathbf{z}$ belongs to $\mathcal{Z}$ involves the *parity-check* matrix $\mathbf{P}$ associated to $\mathbf{H}$. Since $\mathbf{H}$ is of rank N, there exists an $(M-N)\times M$ matrix $\mathbf{P}$ of rank $(M-N)$ such that

$$\mathbf{PHx}=\mathbf{0},\ \forall \mathbf{x}\in\mathbb{R}^N. \tag{5.25}$$

One has

$$\mathbf{Py}=\mathbf{0}\Longleftrightarrow \exists \mathbf{x}\in\mathbb{R}^N \text{ st } \mathbf{y}=\mathbf{Hx}. \tag{5.26}$$

From (5.26), one deduces

$$\exists \mathbf{y} \text{ st } \mathbf{Py}=\mathbf{0} \text{ and } \mathbf{z}=Q(\mathbf{y})\Longleftrightarrow \exists \mathbf{x} \text{ st } \mathbf{z}=Q\left(\mathbf{Hx}\right). \tag{5.27}$$

Using the definition of $\mathcal{Z}$ in (5.17) and (5.27), one obtains

$$\mathbf{z}\in\mathcal{Z}\Longleftrightarrow \exists \mathbf{y}\in\mathbb{R}^M,\ \mathbf{z}=Q(\mathbf{y}) \text{ and } \mathbf{Py}=\mathbf{0}. \tag{5.28}$$

Now let us apply the equivalence (5.2) to all the components z_m, $m=1,\ldots,M$ of a given vector $\mathbf{z}$

$$\mathbf{z}=Q(\mathbf{y})\Longleftrightarrow \mathbf{y}\in[\mathbf{s}(\mathbf{z})], \tag{5.29}$$

where $[\mathbf{s}\left(\mathbf{z}_{1:M}\right)]=\left(\left[s\left(z_1^{(\ell)}\right)\right],\ldots,\left[s\left(z_M^{(\ell)}\right)\right]\right)^T$ is a *box*, *i.e.*, a vector of intervals. The box $\mathbf{P}\left[\mathbf{s}\left(\mathbf{z}_{1:M}\right)\right]$, which can be evaluated using basic interval additions and multiplications [73], is such that

$$\{\mathbf{Py} \text{ with } \mathbf{y}\in\left[\mathbf{s}\left(\mathbf{z}_{1:M}\right)\right]\}\subset\mathbf{P}\left[\mathbf{s}\left(\mathbf{z}_{1:M}\right)\right]. \tag{5.30}$$

Using (5.29) and (5.30)

$$\mathbf{z}=Q(\mathbf{y}) \text{ and } \mathbf{Py}=\mathbf{0}\Rightarrow\mathbf{0}\in\mathbf{P}\left[\mathbf{s}\left(\mathbf{z}\right)\right]. \tag{5.31}$$

Finally, from (5.28) and (5.31) one gets

$$\mathbf{z}\in\mathcal{Z}\Rightarrow\mathbf{0}\in\mathbf{P}\left[\mathbf{s}\left(\mathbf{z}\right)\right]. \tag{5.32}$$

One may rewrite (5.32) as

$$\mathbf{0}\notin\mathbf{P}\left[\mathbf{s}\left(\mathbf{z}\right)\right]\Rightarrow\mathbf{z}\notin\mathcal{Z}. \tag{5.33}$$

The test (5.33) allows to prove that a vector of quantization indexes $\mathbf{z}$ is not consistent, but it is *unable* to prove that some $\mathbf{z}$ is consistent, as the inclusion (5.30) is strict. This test allows to eliminate some of the inconsistent vectors $\mathbf{z}^{(\ell)}$, $\ell=1,\ldots,L$ and has a complexity between $O\left(M\right)$ and $O\left(M\left(M-N\right)\right)$, which is in general lower than solving directly (5.24).

5.3.2.4. Running Example

In this example, we consider the same transmission scheme in Sections 5.2.2 and 5.3.1.1, where the same transform matrix $\mathbf{H}$ is used. A parity-check matrix associated to $\mathbf{H}$ is $\mathbf{P} = (1, 1, 1)$. Recall that the input signal is $\mathbf{x} = (-0.62, 1.42)^T$ and the output signal is $\mathbf{y} = (-0.62, 1.54, -0.92)^T$.

A 2-bit quantizer Q is considered with

$$\begin{aligned}[s(0)] &=] -\infty, -1.03] \\ [s(1)] &= [-1.03, 0.00] \\ [s(2)] &= [0.00, 1.03] \\ [s(3)] &= [1.03, +\infty[.\end{aligned}$$

The obtained vector of quantization indexes is thus $\mathbf{z} = (1, 3, 1)^T$. This vector is BPSK modulated, then transmitted over an AWGN channel with a channel SNR of 2 dB. The observation

$$\mathbf{r} = \begin{pmatrix} -0.65 & -0.97 \\ -1.01 & -1.44 \\ 0.07 & -0.92 \end{pmatrix}$$

is obtained. One may sort the various combinations of $\mathbf{z}_{1:3}$ by decreasing value of $p(\mathbf{r}_{1:3}|\mathbf{z}_{1:3})$ to obtain $\mathbf{z}_{1:3}^{(\ell)}$, $\ell = 1, \ldots, L$, with $L = 64$. In what follows, we will only consider the four first vectors $\mathbf{z}_{1:3}^{(\ell)}$, $\ell = 1, \ldots, 4$. Table 5.2 shows the obtained values for $p_{\mathbf{R}|\mathbf{Z}}\left(\mathbf{r}_{1:3}|\mathbf{z}_{1:3}^{(\ell)}\right)$, $\ell = 1, \ldots, 4$.

ℓ	1	2	3	4
$\mathbf{z}_{1:3}^{(\ell)}$	$(3, 3, 1)^T$	$(3, 3, 3)^T$	$(1, 3, 1)^T$	$(1, 3, 3)^T$
$p_{\mathbf{R}|\mathbf{Z}}\left(\mathbf{r}_{1:3}|\mathbf{z}_{1:3}^{(\ell)}\right)$	0.42	0.33	0.05	0.04

Table 5.2.: Obtained values for $p_{\mathbf{R}|\mathbf{Z}}\left(\mathbf{r}_{1:3}|\mathbf{z}_{1:3}^{(\ell)}\right)$, for $\ell = 1, \ldots, 4$.

We first consider the consistency test based on linear programming described in Section 5.3.2.2. No solution is found for the linear problem in (5.24), when $\ell \in \{1, 2, 4\}$. For $\ell = 3$, a solution is found for (5.24).

Let us illustrate this result by visualizing the sets $\mathcal{X}(\mathbf{z}^{(\ell)})$, $\ell = 1, \ldots, 4$ introduced in (5.21). Figure 5.3 illustrates $\mathcal{X}_3(3)$ and Figure 5.4 illustrates $\mathcal{X}\left(\mathbf{z}_{1:3}^{(\ell)}\right) = \bigcap_{m=1}^{3} \mathcal{X}_m(z_m)$, $\ell = 1, \ldots, 4$. As one can see, $\mathcal{X}\left(\mathbf{z}_{1:3}^{(\ell)}\right) = \emptyset$, $\ell \in \{1, 2, 4\}$ and $\mathcal{X}\left(\mathbf{z}_{1:3}^{(3)}\right) \neq \emptyset$.

Consider the consistency test (5.33) based on the parity-check matrix $\mathbf{P}$, which is performed by using interval analysis, as described in Section 5.3.2.3. The results of

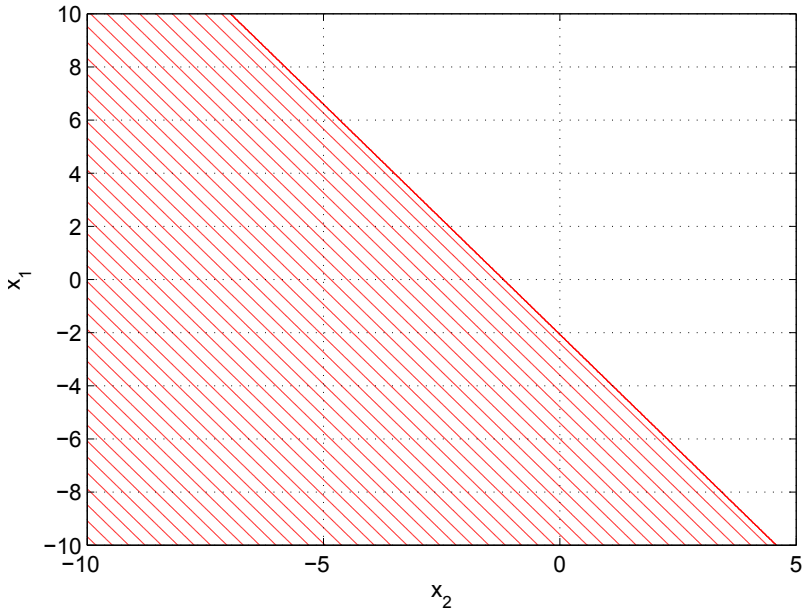


Figure 5.3.: Visualization of $\mathcal{X}_3(3)$.

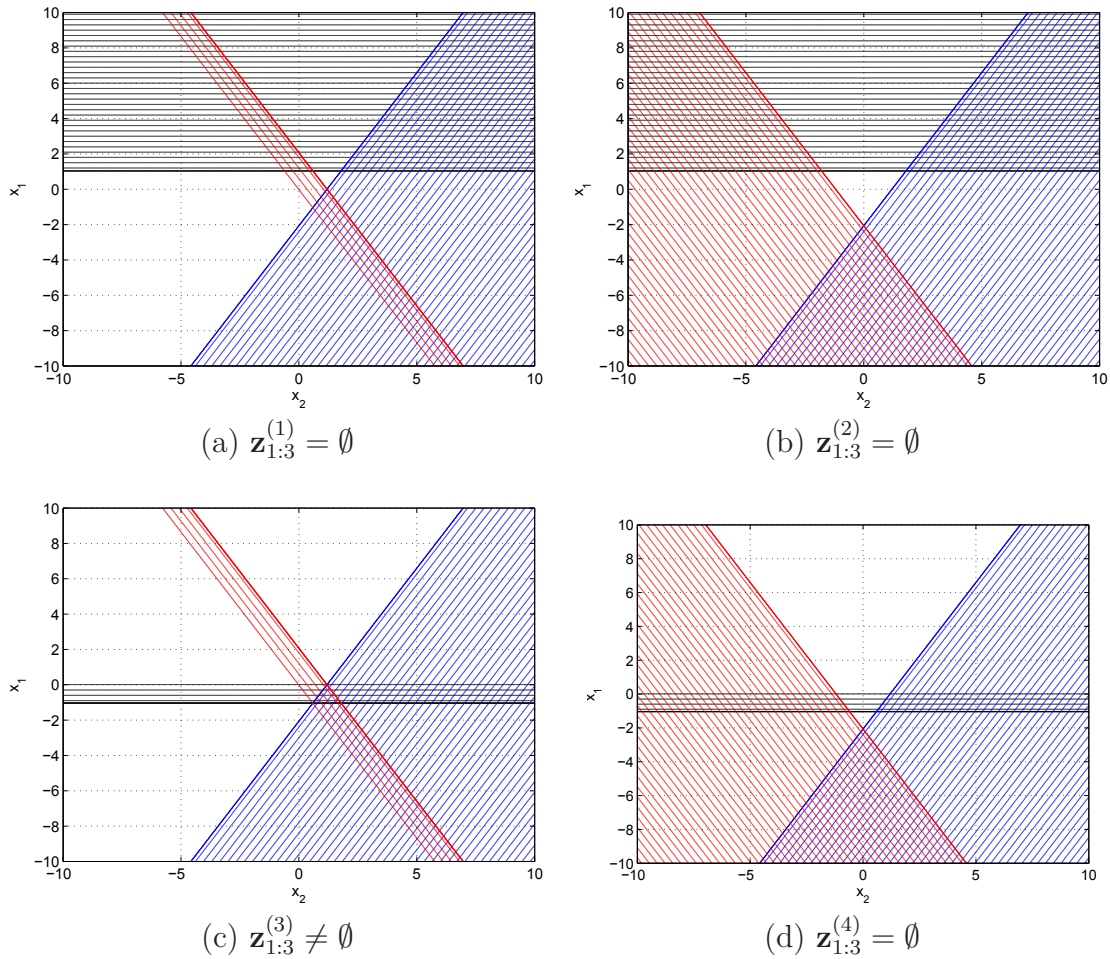
$\mathbf{z}^{(\ell)}$	PCT
$\mathbf{z}^{(1)} = (3,3,1)^T$	$\mathbf{0} \notin \mathbf{P}[\mathbf{s}(\mathbf{z}^{(1)})]$
$\mathbf{z}^{(2)} = (3,3,3)^T$	$\mathbf{0} \notin \mathbf{P}[\mathbf{s}(\mathbf{z}^{(2)})]$
$\mathbf{z}^{(3)} = (1,3,1)^T$	$\mathbf{0} \in \mathbf{P}[\mathbf{s}(\mathbf{z}^{(3)})]$
$\mathbf{z}^{(4)} = (1,3,3)^T$	$\mathbf{0} \notin \mathbf{P}[\mathbf{s}(\mathbf{z}^{(4)})]$

Table 5.3.: The 4 most likely $\mathbf{z}^{(\ell)}$ and the corresponding parity-check test (PCT).

this test are reported in Table 5.3. Again, only $\mathbf{z}^{(3)}$ is deemed consistent with the parity-check test (PCT).

Table 5.4 evaluates the execution times t_{LP} and t_{QP} required to solve the linear program (5.24) and the quadratic program (5.16) respectively with $\mathbf{z}_{1:3}^{(\ell)}$, $\ell = 1, \ldots, 4$, as well as the execution time t_{PCT} required to perform the consistency test (5.33), based on the parity-check matrix $\mathbf{P}$. As expected, the parity-check test takes less time than solving the linear program (5.24). For example, finding out that $\mathbf{z}^{(1)}$ is inconsistent takes 5 times less time when considering the parity-check test (5.33) than when (5.24) is solved.

Selecting $\mathbf{z}^{\mathrm{f}} = \mathbf{z}^{(3)} = (1,3,1)^T$ to solve (5.16), the MAP estimate $\widehat{\mathbf{x}}_{\mathrm{MAP}} = (-0.51, 0.89)^T$ is obtained.



(a) $\mathbf{z}_{1:3}^{(1)} = \emptyset$ (b) $\mathbf{z}_{1:3}^{(2)} = \emptyset$

(c) $\mathbf{z}_{1:3}^{(3)} \neq \emptyset$ (d) $\mathbf{z}_{1:3}^{(4)} = \emptyset$

Figure 5.4.: Visualization of $\mathcal{X}\left(\mathbf{z}_{1:3}^{(\ell)}\right)$, $\ell = 1, \ldots, 4$.

5.3.3. Practical Implementation

In order to implement the consistent MAP estimation algorithm, one has to sort the possible values of $\mathbf{z}_{1:M}$ in decreasing order of $p_{\mathbf{R}|\mathbf{Z}}(\mathbf{r}_{1:M}|\mathbf{z}_{1:M})$. The number of possible combinations of $\mathbf{z}_{1:M}$ is $L = 2^{\rho M}$, which may be intractable even for moderate values of M.

For $m = 1, \ldots, M$, one may sort the possible values of z_m in decreasing order of $p_{\mathbf{R}|Z}(\mathbf{r}_m|z_m)$. One then keeps only the N_m values of z_m associated to the largest $p_{\mathbf{R}|Z}(\mathbf{r}_m|z_m)$. This leads to $\prod_{m=1}^{M} N_m$ possible combinations of values for $\mathbf{z}_{1:M}$, which may in turn be sorted in decreasing order of $p_{\mathbf{R}|Z}(\mathbf{r}_{1:M}|\mathbf{z}_{1:M})$. Again, only the best N' vectors $\mathbf{z}_{1:M}^{(1)}, \ldots, \mathbf{z}_{1:M}^{(N)}$ may be kept. The consistent MAP estimation may then be applied to these N' vectors. The choice of N_m and N' depends on the level of the channel noise, on the relative values of $p_{\mathbf{R}|Z}(\mathbf{r}_m|z_m)$, and on the targeted complexity-efficiency trade-off.

ℓ	1	2	3	4
$\mathbf{z}_{1:3}^{(\ell)}$	$(3,3,1)^T$	$(3,3,3)^T$	$(1,3,1)^T$	$(1,3,3)^T$
t_{QP}(seconds)	0.0058	0.0066	0.0072	0.0164
t_{LP}(seconds)	0.0024	0.0030	0.0014	0.0022
t_{PCT}(seconds)	0.00048	0.00048	0.00047	0.00047

Table 5.4.: Execution time of (5.24), (5.16) and (5.33) for $\mathbf{z}^{(\ell)}$, $\ell = 1, \ldots, 4$.

On the other hand, solving (5.16) may be very complex when $p_{\mathbf{X}}(\mathbf{x})$ is a general PDF and when N is large. One could alternatively consider the least square estimator $\widehat{\mathbf{x}}_{\mathrm{LS}}$ of $\mathbf{x}$ defined as

$$\widehat{\mathbf{x}}_{\mathrm{LS}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T Q^{-1}\left(\mathbf{z}^{\mathrm{f}}\right), \tag{5.34}$$

where Q^{-1} denotes the inverse quantization operation associated to Q.

For the example in Section 5.3.2.4, one obtains $\widehat{\mathbf{x}}_{\mathrm{LS}} = (-0.68, 1.19)^T$, which is closer to $\mathbf{x} = (-0.62, 1.42)^T$ than $\widehat{\mathbf{x}}_{\mathrm{MAP}} = (-0.51, 0.89)^T$, in the MSE sense.

5.4. Estimation by Belief Propagation

In this section, a different approach to estimate $\widehat{\mathbf{x}}_{\mathrm{MAP}}$ defined in (5.11) is presented. This approach applies the belief propagation (BP) algorithm to compute the posterior probability distributions $p_{X|\mathbf{R}}(x_n|\mathbf{r}_{1:M})$, $n = 1, \ldots, N$ of each entry x_n of the input vector $\mathbf{x} = (x_1, \ldots, x_n)^T$, knowing the observation $\mathbf{r}_{1:M}$.

5.4.1. BP algorithm

BP is a general estimation algorithm where the dependencies between the variables are represented by a factor graph [95]. Estimates (*beliefs*) of the marginal distributions of the variables are iteratively updated via a *message passing* procedure along the edges of this graph [80, 95, 38]. This algorithm has proven to be efficient in numerous applications, including iterative decoding of low-density parity-check (LDPC) codes and turbo codes, which typically work over finite fields [105, 97]. It has also been applied in real field problems, such as random linear mixing and compressed sensing problems [38, 32, 128].

The problem considered here is to estimate the input vector $\mathbf{x}$ from the noisy observations $\mathbf{r}$ of linear measurement $\mathbf{y} = \mathbf{H}\mathbf{x}$. This estimation problem has been adressed with BP by Rangan *et al.* in [128, 129, 77], where the linear relations between the variables are represented by a factor graph $\mathcal{G}_{\mathbf{H}}$, associated to the transform matrix $\mathbf{H}$, see Figure 5.5. This graph is formed by two types of nodes: the variable

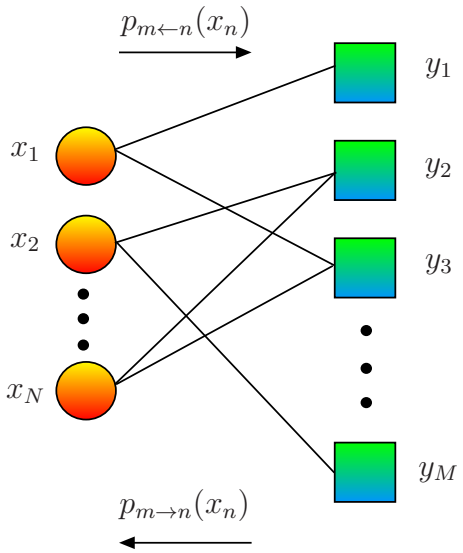


Figure 5.5.: Factor graph for the linear mixing estimation problem.

nodes $n = 1 \ldots, N$, describing the variables x_n and the factor nodes $m = 0, \ldots, M$, describing the variables y_m. Let

$$\mathcal{E}_{\mathbf{H}} \{(n, m) \in \{1, \ldots, N\} \times \{1, \ldots, M\} \mid \mathbf{H}(m, n) \neq 0\}$$

be the set of edges connecting these nodes: an edge between node n and node m signifies that the variables x_n and y_m are linearly dependent. The set of variable nodes connected to a factor node m is denoted by $\mathcal{N}(m)$. Similarly, the set of factor nodes connected to some variable node n is denoted by $\mathcal{N}(n)$. These different nodes *talk* to each others by exchanging messages (*beliefs*) along the edges of $\mathcal{G}_{\mathbf{H}}$.

An implementation of BP, which is adapted to the MAP estimation considered in (5.11), is presented in what follows. The exchanged messages are beliefs on the posterior probability distributions of each entry x_n, which are iteratively updated by passing them along $\mathcal{E}_{\mathbf{H}}$. The BP algorithm presented here is inspired by the one presented by Rangan in [128] and has the following steps:

1. *Initialization:*

   a) set the current iteration $i = 1$

   b) $\forall (n, m) \in \mathcal{E}_{\mathbf{H}}$ set the messages sent by n to m to the initial distribution of the random variable X_n

   $$p^x_{m\leftarrow n}(i, x_n) = p^x_n(i, x_n) = p_X(x_n). \tag{5.35}$$

2. *Linear mixing:*

   a) assume that the random variables X_n are independent and that $X_n \sim p^x_{m\leftarrow n}(i, x_n)$

b) $\forall(n,m) \in \mathcal{E}_{\mathbf{H}}$ compute the distribution $p^{y}_{m\to n}(i, y_{m\to n})$ of the random variables

$$Y_{m\to n} = \sum_{\substack{n'\in\mathcal{N}(m)\\ n'\neq n}} \mathbf{H}(m,n')X_{n'}. \tag{5.36}$$

3. *Output Update:*

a) $\forall(n,m) \in \mathcal{E}_{\mathbf{H}}$ compute the likelihood function

$$p^{u}_{m\to n}(i,u_m) = \int_{y_{m\to n}} p_{\mathbf{R}|Y}(\mathbf{r}_m|u_m + y_{m\to n}) p^{y}_{m\to n}(i, y_{m\to n}) dy_{m\to n}. \tag{5.37}$$

4. *Input update:*

a) $\forall(n,m) \in \mathcal{E}_{\mathbf{H}}$ update the message sent by n to m

$$p^{x}_{m\leftarrow n}(i+1,x_n) = \alpha p_X(x_n) \prod_{\substack{m'\in\mathcal{N}(n)\\ m'\neq m}} p^{u}_{m'\to n}(i, \mathbf{H}(m',n)x_n), \tag{5.38}$$

where α is a normalization constant obtained by imposing that $p^{x}_{m\leftarrow n}(i+1,x_n)$ sum up to 1.

b) $\forall n = 1,\ldots,.N$ update the distribution

$$p^{x}_{n}(i+1,x_n) = \beta p_X(x_n) \prod_{m\in N(n)} p^{u}_{m\to n}(i, \mathbf{H}(m,n)x_n), \tag{5.39}$$

where β is a normalization constant obtained by imposing that $p^{x}_{n}(i+1,x_n)$ sum up to 1.

5. *Incrementation:*

a) $i = i+1$

b) return to Step. 2 until a sufficient number of iterations is performed.

The message $p^{x}_{m\leftarrow n}(i,x_n)$, sent by the variable node n to the factor node m at iteration i, expresses the beliefs of the variable node n about the states in which X_n could be. In our case, it expresses the posterior probability distribution of x_n as *believed* by the neighboring factor nodes $y_{m'} \in \mathcal{N}(n)$, $m' \neq m$ at iteration i. The message $p^{y}_{m\to n}(i, y_{m\to n})$, sent by the factor node m to the variable node n at iteration i, allows to evaluate the likelihood function $p^{u}_{m\to n}(i,u_m)$, which measures how likely the observation $\mathbf{r}_m$ is obtained when $X_n = x_n$. When $\mathcal{G}_{\mathbf{H}}$ has no cycles, this message passing procedure is likely to converge to a consensus that determines the true marginals $p_{X|\mathbf{R}}(x_n|\mathbf{r}_{1:M})$.

After N_{i} iterations of the BP algorithm, the MAP estimate of $\mathbf{x}$ defined in (5.3) is obtained as follows

$$\widehat{\mathbf{x}}_{\mathrm{MAP}} = \left(\widehat{x}_{1,\mathrm{MAP}}, \ldots, \widehat{x}_{N,\mathrm{MAP}}\right)^T, \tag{5.40}$$

where, for $n = 1, \ldots, N$

$$\widehat{x}_{n,\mathrm{MAP}} = \arg\max_{x_n \in \mathbb{R}} p_n^x(N_{\mathrm{i}}, x_n).$$

The steps of the BP presented here are derived from the sum-product algorithm, which is a generic representation of the message-passing algorithms operating in a factor graph [81, 95]. A presentation of the sum-product algorithm is provided in Appendix C.1, where it is shown how the BP algorithm presented in this section can be viewed as a particular instance of this generic algorithm.

5.4.2. Running Example

In this section the example of Section 5.3.2.4 is considered. The input signal is $\mathbf{x} = (-0.62, 1.42)^T$ and the output signal is $\mathbf{y} = (-0.62, 1.54, -0.92)^T$. Moreover, the same 2-bit quantizer Q is used.

The MAP estimation based on the BP algorithm described in Section 5.4.1 is implemented by considering probability mass functions approximating the continuous distributions. The range considered for the input variables x_n, $n = 1, 2$ is from -10 to 10 and the number of points on which the probability distribution functions are evaluated is set to 1024. The considered resolution is then of 20/1024. The total number of iterations of the BP algorithm is equal to 10. At each iteration, the messages $p^x_{m \leftarrow n}(i, x_n)$ and $p^u_{m \rightarrow n}(i, u_m)$ are vectors of 1024 entries, where the probability distribution is evaluated. The MAP estimate described in (5.40) is then obtained: $\widehat{\mathbf{x}}_{\mathrm{MAP}} = (-1.33, 2.20)^T$.

5.5. Comparison between the two Estimation Schemes

In this section, the two estimation schemes presented in Sections 5.3 and 5.4 are compared. The same running example introduced previously is considered: $N = 2$, $M = 3$, $X \sim \mathcal{N}(0, \mathbf{I}_2)$ and

$$\mathbf{H} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix}.$$

A 4−bit quantizer Q is used here for each entry of $\mathbf{y} = \mathbf{H}\mathbf{x}$. The components of $\mathbf{z} = Q(\mathbf{y})$ are BPSK modulated, then transmitted over an AWGN channel, with an SNR varying between 0 dB and 14 dB. The number of noise realizations is set to 3000 for each value of the channel SNR.

In the sequel, we describe the different reconstruction approaches and compare them.

Consistent estimation scheme

The consistent MAP estimation scheme presented in Section 5.3 is performed, and the most likely feasible vector of quantization indexes $\mathbf{z}^{\mathrm{f}}$ defined in (5.18) is obtained.

Two MAP estimates of the input $\mathbf{x}$ are considered. The first one is obtained by solving the quadratic program (QP) in (5.19) with $\mathbf{z}^{\mathrm{f}}$. This estimate is reffered to as $\widehat{\mathbf{x}}_{\mathrm{QP}}$.

The second estimate we consider is the one obtained using least squares (LS) reconstruction, as follows

$$\widehat{\mathbf{x}}_{\mathrm{LS}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T Q^{-1}\left(\mathbf{z}^{\mathrm{f}}\right).$$

Estimation using BP

The MAP estimation based on the BP algorithm and presented in Section 5.4.1 is performed as described in the example of Section 5.4.2. The MAP estimate defined in (5.40) is then obtained. In this section, this estimate is reffered to as $\widehat{\mathbf{x}}_{\mathrm{BP}}$.

Estimation scheme using the hard decisions of the channel

This estimation scheme uses the hard decisions taken by a classical non-robust decoder on the received vector $\mathbf{r}_{1:M}$. As described in Section 5.2.2, after demodulation and inverse quantization, the received vector $\widehat{\mathbf{y}}_{\mathrm{HD}}$ is obtained, and the input vector may be reconstructed using LS estimation as:

$$\widehat{\mathbf{x}}_{\mathrm{HD}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\widehat{\mathbf{y}}_{\mathrm{HD}}.$$

Reference reconstruction scheme

The scheme serving as a reference considers the noise-free signal $\widehat{\mathbf{x}}_{\mathrm{REF}}$, reconstructed after inverse quantization of $\mathbf{z}$. Let $\widehat{\mathbf{y}}_{\mathrm{REF}} = Q^{-1}(\mathbf{z})$. Then

$$\widehat{\mathbf{x}}_{\mathrm{REF}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\widehat{\mathbf{y}}_{\mathrm{REF}}.$$

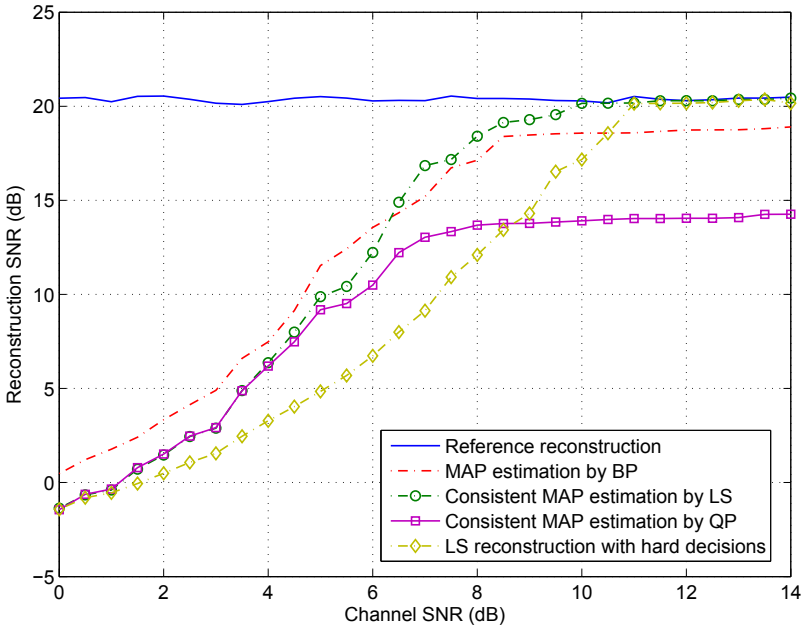


Figure 5.6.: SNR (dB) of the reconstructed signals as function of the channel SNR (dB).

Figure 5.6 shows the average SNR of the signal reconstructed using the estimation schemes previously described, as function of the channel SNR.

For a channel SNR less than 8.5 dB, all the proposed estimation schemes outperform the classical reconstruction approach based on the hard decisions of the channel. For example, for a channel SNR of 6 dB, the gain observed in terms of reconstruction SNR is of 4 dB, and of 5.5 dB, when considering $\widehat{\mathbf{x}}_{\mathrm{QP}}$, and $\widehat{\mathbf{x}}_{\mathrm{LS}}$, respectively. For the same channel SNR level, the gain in reconstruction SNR is more important when considering estimation by BP, and is almost equal to 7 dB.

For a channel SNR less than 4 dB, the consistent estimates $\widehat{\mathbf{x}}_{\mathrm{LS}}$ and $\widehat{\mathbf{x}}_{\mathrm{QP}}$ lead to a similar reconstruction SNR performance. For a higher channel SNR level, $\widehat{\mathbf{x}}_{\mathrm{LS}}$ performs better than $\widehat{\mathbf{x}}_{\mathrm{QP}}$ and the gap between the two estimates becomes more important with the channel SNR. This is explained by the fact that for a negligible channel noise, the most likely vector of quantization indexes $\mathbf{z}^*$ is usually the one which has been transmitted as presented in Section 5.3.1. The least squares estimate $\widehat{\mathbf{x}}_{\mathrm{LS}}$ in this case minimizes the quantization error, since there are no transmission errors, and is then more efficient than maximizing the source's *a priori* $p_{\mathbf{X}}(\mathbf{x})$ over the set $\mathcal{X}(\mathbf{z}^*)$.

The estimation scheme using the BP algorithm is more efficient than the consistent estimation schemes at low channel SNRs. For a channel SNR higher than 6.5 dB, $\widehat{\mathbf{x}}_{\mathrm{LS}}$ becomes more efficient than $\widehat{\mathbf{x}}_{\mathrm{BP}}$ and for a channel SNR higher than 10.5 dB, $\widehat{\mathbf{x}}_{\mathrm{HD}}$ outperforms $\widehat{\mathbf{x}}_{\mathrm{BP}}$. This may be explained by the fact that the evaluation of $\widehat{\mathbf{x}}_{\mathrm{BP}}$ involves the computation of the marginal probability distributions instead of

the joint distribution of $\mathbf{x}$. Moreover, the BP algorithm may not converge to the exact marginals due to the presence of cycles in the factor graph. Both effects do not appear when the impact of the channel noise is significant.

5.6. Application to Oversampled Filter Banks

In this section, oversampled filter banks [151] are considered as a practical example of schemes operating a signal expansion. The estimation approaches presented in Sections 5.3 and 5.4 are described and applied to theses filter banks.

In the next section, a brief presentation of oversampled filter banks is provided. A more detailed presentation is provided in Appendix B.

5.6.1. Brief Presentation of OFBs

An oversampled filter bank (OFB) is a filter bank whose number of output subbands is larger than the downsampling factor. These subbands form then a redundant representation of the input signal.

A typical $M-$band OFB with a downsampling factor of $N < M$ is presented in Figure 5.7. This OFB is formed by M analysis filters h_m, $m = 1, \ldots, M$ assumed of finite impulse responses (FIR). The polyphase representation of this OFB is the $M \times N$ matrix

$$\mathbf{H}(z) = \sum_{k=0}^{K-1} \mathbf{H}_k z^{-k} \tag{5.41}$$

where $K \in \mathbb{N}^*$ and $\mathbf{H}_k$, $k = 0, \ldots, K-1$ is a sequence of $M \times N$ matrices which can be constructed from the filters h_m, $m = 1, \ldots, M$, see Appendix B.

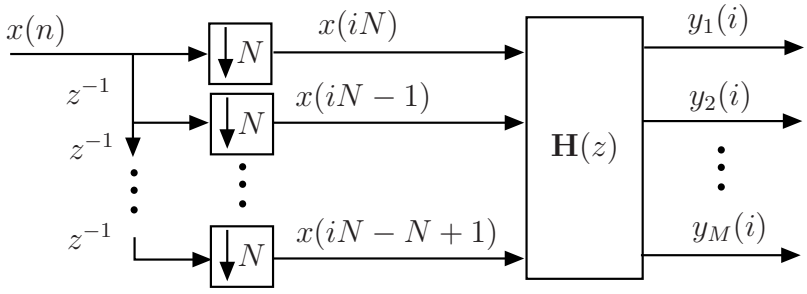


Figure 5.7.: Analysis stage of an $M-$band oversampled filter bank.

Assuming further that $\mathbf{H}(z)$ has full column rank, there exists an FIR filter bank, having the polyphase representation given by the an $(M-N) \times M$ matrix $\mathbf{P}(z)$

which verifies

$$\mathbf{P}(z)\mathbf{H}(z) = \mathbf{0},\ \forall z \in \mathbb{C} \tag{5.42}$$

and

$$\mathbf{P}(z) = \sum_{k=0}^{K'-1} \mathbf{P}_k z^{-k},$$

where $K' \in \mathbb{N}^*$ and $\mathbf{P}_k$, $k = 0, \ldots, K'-1$ is a sequence of $(M-N) \times M$ matrices, see Appendix B. The matrix $\mathbf{P}(z)$ is the parity-check matrix associated to $\mathbf{H}(z)$.

5.6.2. Signal Expansion using OFBs

Let $\mathbf{x} \in \mathbb{R}^{VN}$ and $\mathbf{y} \in \mathbb{R}^{VM}$ be the input and output vectors of an OFB, where V is some positive integer . One may write

$$\mathbf{x} = \left(\left(\mathbf{x}^1\right)^T, \ldots, \left(\mathbf{x}^V\right)^T \right)^T$$

and

$$\mathbf{y} = \left(\left(\mathbf{y}^1\right)^T, \ldots, \left(\mathbf{y}^V\right)^T \right)^T,$$

where

$$\mathbf{x}^i = \left(x\left(iN - N + 1\right), \ldots, x\left(iN\right)\right)^T$$

and

$$\mathbf{y}^i = \left(y\left(iM - M + 1\right), \ldots, y\left(iM\right)\right)^T$$

are the input and the output of the OFB at time instant $i = 1, \ldots, V$.

It is shown in Appendix B that

$$\mathbf{y}^i = \sum_{k=0}^{K-1} \mathbf{H}_k \mathbf{x}^{i-k}, \tag{5.43}$$

with $\mathbf{x}^i = \mathbf{0}$, if $i \leqslant 0$.

Let $\mathbf{H}_{K-1:0} = (\mathbf{H}_{K-1}, \ldots, \mathbf{H}_0)$ be the $M \times KN$ matrix constructed from the matrices $\mathbf{H}_k$ and let

$$\mathbf{x}^{i-K+1:i} = \left(\left(\mathbf{x}^{i-K+1}\right)^T, \ldots, \left(\mathbf{x}^i\right)^T \right)^T$$

be the vector containing all the input samples affecting the OFB output at time instant i. Then (5.43) may be written as

$$\mathbf{y}^i = \mathbf{H}_{K-1:0}\mathbf{x}^{i-K+1:i}. \tag{5.44}$$

Using (5.44), one can write the whole OFB operations as the linear expansion presented in (5.1), where

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \mathbf{0} \\ \mathbf{H}_1 & \mathbf{H}_0 & \mathbf{0} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{H}_{K-1} & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & \ddots & \mathbf{0} & \vdots \\ \vdots & \ddots & \mathbf{H}_{K-1} & \ddots & \mathbf{H}_1 & \mathbf{H}_0 & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{H}_{K-1} & \cdots & \mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix} \tag{5.45}$$

is an $VM \times VN$ block-diagonal matrix.

Now let us write (5.42) in the temporal domain for each output $\mathbf{y}^i$, $i = 1, \ldots, V$ of the OFB

$$\sum_{k=0}^{K'-1} \mathbf{P}_k \mathbf{y}^{i-k} = \mathbf{0}, \tag{5.46}$$

where $\mathbf{y}^i = \mathbf{0}$, if $i \leqslant 0$. At each time instant i, the relation (5.46) defines a set of $M - N$ equations, which may be used to check whether a given subband vector

$$\left(\left(\mathbf{y}^{i-K'+1}\right)^T, \ldots, \left(\mathbf{y}^i\right)^T \right)^T$$

could have been generated by the OFB. These equations are referred to as the parity-check equations and may be rewritten as (5.25), where

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_0 & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \mathbf{0} \\ \mathbf{P}_1 & \mathbf{P}_0 & \mathbf{0} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{P}_{K'-1} & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & \ddots & \mathbf{0} & \vdots \\ \vdots & \ddots & \mathbf{P}_{K'-1} & \ddots & \mathbf{P}_1 & \mathbf{P}_0 & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{P}_{K'-1} & \cdots & \mathbf{P}_1 & \mathbf{P}_0 \end{pmatrix} \tag{5.47}$$

is the parity-check matrix associated to $\mathbf{H}$ defined in (5.45).

5.6.2.1. Example and Simulation Results

In this example, an OFB based on the Haar filter bank is considered with $N = 4$ and $M = 6$. Let $\mathbf{x} \in \mathbb{R}^{12}$ be the input vector. Then $V = 3$. The components of $\mathbf{x}$ are i.i.d., have a normal distribution, and are of variance $\sigma_x^2 = 1$. The transform matrix $\mathbf{H}$ that has been considered is of size 18×12 and is defined as follows

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_1 & \mathbf{H}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix}, \tag{5.48}$$

where

$$\mathbf{H}_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \text{ and } \mathbf{H}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 \end{pmatrix}. \tag{5.49}$$

The associated parity-check matrix is

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{P}_1 & \mathbf{P}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_1 & \mathbf{P}_0 \end{pmatrix},$$

where

$$\mathbf{P}_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & -0.5 & 0.5 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \text{ and } \mathbf{P}_1 = \begin{pmatrix} 0 & -1 & 1 & 1 & 0.5 & 0.5 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{5.50}$$

Each component of the vector $\mathbf{y} \in \mathbb{R}^{18}$, obtained at the output of $\mathbf{H}$, is quantized using the scalar quantization function Q over $\rho = 4$ bits, then BPSK modulated and transmitted over an AWGN channel with an SNR level between 0 dB and 14 dB.

The estimation schemes described in Section 5.5 are compared and the results are reported in Figure 5.8. The same observations as the running example of Section 5.4.2, may be made here. At low channel SNRs, the BP algorithm leads to a better reconstruction SNR when compared to the consistent estimation. Moreover, the consistent estimation using the least squares reconstruction outperforms the consistent estimation using quadratic programming, for moderate to large values of the channel SNR.

5.6.3. Iterative Implementation of the Consistent MAP Estimator

The aim of this section is to provide an iterative implementation of the consistent MAP estimation presented in Section 5.3, when the expansion matrix $\mathbf{H}$ has the

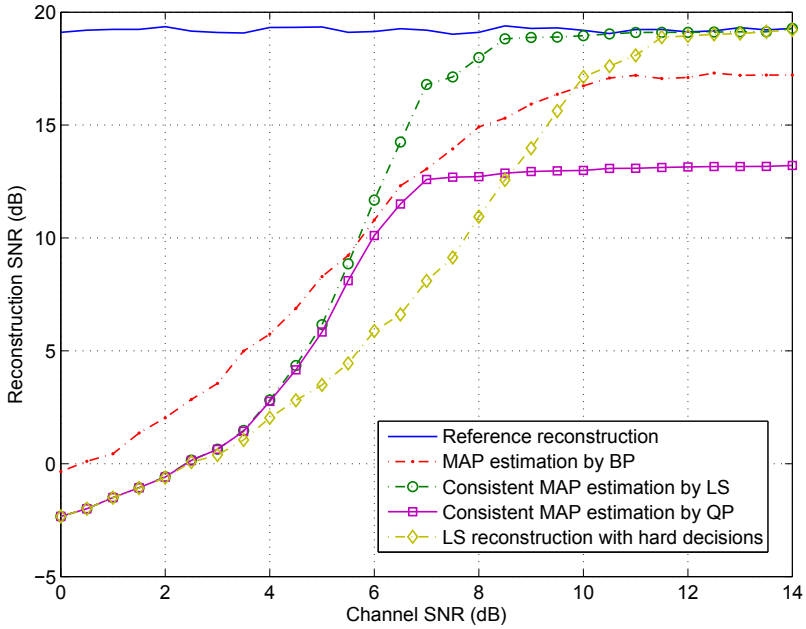


Figure 5.8.: SNR (dB) of the reconstructed signals as function of the channel SNR (dB), for the OFB based on the Haar filters.

structure described by (5.45), as in the case of OFBs. This iterative implementation is particularly useful when considering large-scale input and output signals.

As in Section 5.3, the aim is to estimate the source outcome $\mathbf{x} \in \mathbb{R}^{VN}$ from the reception of some channel output vector

$$\mathbf{r} = \left(\left(\mathbf{r}^1\right)^T, \ldots, \left(\mathbf{r}^V\right)^T \right)^T .$$

Assume, at time instant i, that estimates $\widehat{\mathbf{x}}^j$, $j = i-K+1, \ldots, i-1$ have been obtained for the components of $\mathbf{x}$ at the $K-1$ previous time instants.

The MAP estimate of $\mathbf{x}^i$ using the knowledge of $\mathbf{r}_{1:M}^i$ and $\widehat{\mathbf{x}}^j$ is then

$$\widehat{\mathbf{x}}^i = \arg \max_{\mathbf{x}^i \in \mathbb{R}^N} p_{\mathbf{X}|\mathbf{R}} \left(\mathbf{x}^i | \mathbf{r}_{1:M}^i, \widehat{\mathbf{x}}^1, \ldots, \widehat{\mathbf{x}}^{i-1} \right). \tag{5.51}$$

Using Bayes' rule, one gets

$$\widehat{\mathbf{x}}^i = \arg \max_{\mathbf{x}^i \in \mathbb{R}^N} p_{\mathbf{R}|\mathbf{X}} \left(\mathbf{r}_{1:M}^i | \mathbf{x}^i, \widehat{\mathbf{x}}^1, \ldots, \widehat{\mathbf{x}}^{i-1} \right) p_{\mathbf{X}} \left(\mathbf{x}^i | \widehat{\mathbf{x}}^1, \ldots, \widehat{\mathbf{x}}^{i-1} \right), \tag{5.52}$$

where

$$\begin{aligned} p_{\mathbf{R}|\mathbf{X}} \left(\mathbf{r}_{1:M}^i | \mathbf{x}^i, \widehat{\mathbf{x}}^1, \ldots, \widehat{\mathbf{x}}^{i-1} \right) &= \sum_{\mathbf{z}_{1:M}} p_{\mathbf{R},\mathbf{Z}|\mathbf{X}} \left(\mathbf{r}_{1:M}^i, \mathbf{z}_{1:M} | \mathbf{x}^i, \widehat{\mathbf{x}}^1, \ldots, \widehat{\mathbf{x}}^{i-1} \right) \\ &= \sum_{\mathbf{z}_{1:M}} p_{\mathbf{R}|\mathbf{Z}} \left(\mathbf{r}_{1:M}^i | \mathbf{z}_{1:M} \right) p_{\mathbf{Z}|\mathbf{X}} \left(\mathbf{z}_{1:M} | \mathbf{x}^i, \widehat{\mathbf{x}}^1, \ldots, \widehat{\mathbf{x}}^{i-1} \right). \end{aligned} \tag{5.53}$$

The second term of the sum (5.53) may be written as

$$p_{\mathbf{Z}|\mathbf{X}}\left(\mathbf{z}_{1:M}|\mathbf{x}^i,\widehat{\mathbf{x}}^1,\ldots,\widehat{\mathbf{x}}^{i-1}\right)=p_{\mathbf{Z}|\mathbf{Y}}\left(\mathbf{z}_{1:M}|\mathbf{H}_0\mathbf{x}^i+\mathbf{H}_{K-1:1}\widehat{\mathbf{x}}^{i-K+1:i-1}\right). \quad (5.54)$$

Now using (5.2), one obtains

$$\begin{aligned} p_{\mathbf{Z}|\mathbf{X}}\left(\mathbf{z}_{1:M}|\mathbf{H}_0\mathbf{x}^i+\sum_{u=1}^{K-1}\mathbf{H}_u\widehat{\mathbf{x}}^{i-u}\right) &= I_{[\mathbf{s}(\mathbf{z}_{1:M})]}\left(\mathbf{H}_0\mathbf{x}^i+\mathbf{H}_{K-1:1}\widehat{\mathbf{x}}^{i-K+1:i-1}\right), \\ &= I_{[\mathbf{s}(\mathbf{z}_{1:M})]-\mathbf{H}_{K-1:1}\widehat{\mathbf{x}}^{i-K+1:i-1}}\left(\mathbf{H}_0\mathbf{x}^i\right). \quad (5.55) \end{aligned}$$

Finally, one gets

$$\widehat{\mathbf{x}}^i=\arg\max_{\mathbf{x}^i\in\mathbb{R}^N} p_{\mathbf{X}}\left(\mathbf{x}^i|\widehat{\mathbf{x}}^1,\ldots,\widehat{\mathbf{x}}^{i-1}\right)\sum_{\mathbf{z}_{1:M}} p_{R|Z}\left(\mathbf{r}^i_{1:M}|\mathbf{z}_{1:M}\right) I_{[\mathbf{s}(\mathbf{z}_{1:M})]-\mathbf{H}_{K-1:1}\widehat{\mathbf{x}}^{i-K+1:i-1}}\left(\mathbf{H}_0\mathbf{x}^i\right). \quad (5.56)$$

Consider the set

$$\mathcal{Z}^i=\left\{\mathbf{z}_{1:M}\in\{0,\ldots,2^\rho-1\}^M \mid \exists\mathbf{x}^i\in\mathbb{R}^N \text{ st } \mathbf{H}_0\mathbf{x}^i\in[\mathbf{z}_{1:M}]-\mathbf{H}_{K-1:1}\widehat{\mathbf{x}}^{i-K+1:i-1}\right\} \quad (5.57)$$

of feasible quantization indexes, defined at each time instant i.

As in Section 5.3.2, in the sum (5.56) over all the vectors $\mathbf{z}_{1:M}$, one keeps only

$$\mathbf{z}^{\mathrm{f}}=\arg\max_{\mathbf{z}_{1:M}\in\mathcal{Z}^i} p_{\mathbf{R}|\mathbf{Z}}\left(\mathbf{r}^i_{1:M}|\mathbf{z}_{1:M}\right). \quad (5.58)$$

Finally, the following linearly-constrained optimization problem derived from (5.56) is obtained

$$\begin{aligned} \widehat{\mathbf{x}}^i=\arg\max_{\mathbf{x}^i\in\mathbb{R}^N} & p_{\mathbf{X}}\left(\mathbf{x}^i|\widehat{\mathbf{x}}^1,\ldots,\widehat{\mathbf{x}}^{i-1}\right). \quad (5.59)\\ \text{st } & \mathbf{H}_0\mathbf{x}^i\leqslant\overline{\mathbf{s}}\left(\mathbf{z}^{\mathrm{f}}\right)-\mathbf{H}_{K-1:1}\widehat{\mathbf{x}}^{i-K+1-i-1}\\ & -\mathbf{H}_0\mathbf{x}^i\leqslant-\underline{\mathbf{s}}\left(\mathbf{z}^{\mathrm{f}}\right)+\mathbf{H}_{K-1:1}\widehat{\mathbf{x}}^{i-K+1-i-1} \end{aligned}$$

In what follows, we will present a practical implementation of this estimator, using interval analysis.

5.6.3.1. Practical Implementation

Assume that the estimation process at the previous instants $j=i-K+1,\ldots,i-1$ has been able to provide boxes $[\widehat{\mathbf{x}}^j]$, such that $\mathbf{x}^j\in[\widehat{\mathbf{x}}^j]$. The set of feasible quantization indexes at time instant i, defined in (5.57), becomes

$$\begin{aligned} \mathcal{Z}^i=\Big\{\mathbf{z}_{1:M}\in\{0,\ldots,2^\rho-1\}^M \mid & \exists\mathbf{x}^i\in\mathbb{R}^N \text{ and } \exists\widetilde{\mathbf{x}}^{i-K+1:i-1}\in\left[\widehat{\mathbf{x}}^{i-K+1:i-1}\right]\\ & \text{st } \mathbf{H}_0\mathbf{x}^i\in[\mathbf{z}_{1:M}]-\mathbf{H}_{K-1:1}\widetilde{\mathbf{x}}^{i-K+1:i-1}\Big\}. \quad (5.60) \end{aligned}$$

Assume further that the estimation process in the previous instants $j' = i - K' + 1, \ldots, i-1$ has been able to provide boxes $\left[\widehat{\mathbf{y}}^{j'}\right]$, such that $\mathbf{y}^{j'} \in \left[\widehat{\mathbf{y}}^{j'}\right]$.

Using (5.27)–(5.30), (5.46) and the definition in (5.60)

$$\mathbf{0} \notin \mathbf{P}_0 \left[s\left(\mathbf{z}\right)\right] + \mathbf{P}_{i-K'+1:i-1} \left[\widehat{\mathbf{y}}^{i-K'+1:i-1}\right] \Rightarrow \mathbf{z} \notin \mathcal{Z}^i.$$

Now consider the set

$$\mathcal{X}^{(i)}\left(\mathbf{z}\right) = \left\{\mathbf{x}^i | \exists \widetilde{\mathbf{x}}^{i-K+1:i-1} \in \left[\widehat{\mathbf{x}}^{i-K+1:i-1}\right] \text{ st } H_0 \mathbf{x}^i \in \left[s\left(\mathbf{z}\right)\right] - H_{K-1:1} \widetilde{\mathbf{x}}^{i-K+1:i-1}\right\}, \tag{5.61}$$

defined for a given $\mathbf{z} \in \{0, , 2^\rho - 1\}^M$ at time instant i. Therefore

$$\mathcal{X}^{(i)}\left(\mathbf{z}\right) \neq \emptyset \Leftrightarrow \mathbf{z} \in \mathcal{Z}^i.$$

The set $\mathcal{X}^{(i)}\left(\mathbf{z}\right)$ is called a polytope [159]. An outer approximation[1] $\left[\mathcal{X}^{(i)}\left(\mathbf{z}\right)\right]$ of this polytope may be obtained using basic operations on the intervals $\left[\widehat{\mathbf{x}}^j\right]$, $j = i - K + 1, \ldots, i-1$ and $\left[s\left(\mathbf{z}\right)\right]$ [73]. Since

$$\mathcal{X}^{(i)}\left(\mathbf{z}\right) \subset \left[\mathcal{X}^{(i)}\left(\mathbf{z}\right)\right], \tag{5.62}$$

one has

$$\left[\mathcal{X}^{(i)}\left(\mathbf{z}\right)\right] = \emptyset \Rightarrow \mathcal{X}^{(i)}\left(\mathbf{z}\right) = \emptyset. \tag{5.63}$$

Therefore, for a given vector of quantization indexes $\mathbf{z} \in \{0, \ldots, 2^\rho - 1\}^M$, if an empty box $\left[\mathcal{X}^{(i)}\left(\mathbf{z}\right)\right]$ is obtained as an outer approximation of $\mathcal{X}^{(i)}\left(\mathbf{z}\right)$, then $\mathbf{z} \notin \mathcal{Z}^i$.

Estimation at $i = 1$, input: $\mathbf{r}^{(1)}$, output $[\widehat{\mathbf{x}}^1]$, $[\widehat{\mathbf{y}}^1]$, $\widehat{\mathbf{y}}^1$

The estimation of $\mathbf{z}^{\mathrm{f}}$ at initialization is performed as described in Section 5.3.2.1, using only $\mathbf{r}^1$, with $\mathbf{H} = \mathbf{H}_0$ and $\mathbf{P} = \mathbf{P}_0$

1. Evaluate $\mathbf{z}^{\mathrm{f}}$;
2. Evaluate $\left[\mathcal{X}^{(1)}\left(\mathbf{z}^{\mathrm{f}}\right)\right]$;
3. Return $[\widehat{\mathbf{x}}^1] = \left[\mathcal{X}^{(1)}\left(\mathbf{z}^{\mathrm{f}}\right)\right]$, $\widehat{\mathbf{y}}^1 = \left[s\left(\mathbf{z}^{\mathrm{f}}\right)\right]$ and $\widehat{\mathbf{y}}^1 = Q^{-1}\left(\mathbf{z}^{\mathrm{f}}\right)$.

[1] An outer approximation of $\mathcal{X}^{(i)}\left(\mathbf{z}\right)$ is a box enclosing $\mathcal{X}^{(i)}\left(\mathbf{z}\right)$.

Estimation at $i \geqslant 2$, input: $\mathbf{r}^i$, $\left[\widehat{\mathbf{x}}^{i-K+1:i-1}\right]$, $\left[\widehat{\mathbf{y}}^{i-K'+1:i-1}\right]$, output: $[\widehat{\mathbf{x}}^i]$, $[\widehat{\mathbf{y}}^i]$, $\widehat{\mathbf{y}}^i$

1. Sort the vectors of quantization indexes $\mathbf{z} \in \{0, \ldots, 2^\rho - 1\}^M$ in decreasing order of $p_{\mathbf{R}|\mathbf{Z}}(\mathbf{r}^i|\mathbf{z})$ in a list $\mathcal{L}_i = \left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(L)}\right\}$, with $L = 2^{\rho M}$.
2. Do
   a) If $\mathbf{0} \in \mathbf{P}_0 \left[s\left(\mathbf{z}^{(\ell)}\right)\right] + \mathbf{P}_{i-K'+1:i-1} \left[\widehat{\mathbf{y}}^{i-K'+1:i-1}\right]$
      i. If $\left[\mathcal{X}^{(i)}\left(\mathbf{z}^{(\ell)}\right)\right] \neq \emptyset$
         A. Return $[\widehat{\mathbf{x}}^i] = \left[\mathcal{X}^{(i)}\left(\mathbf{z}^{(\ell)}\right)\right]$, $[\widehat{\mathbf{y}}^i] = \left[s\left(\mathbf{z}^{(\ell)}\right)\right]$ and $\widehat{\mathbf{y}}^i = Q^{-1}\left(\mathbf{z}^{(\ell)}\right)$; End;
      ii. Else
         A. $\ell = \ell + 1$;
   b) Else
      i. $\ell = \ell + 1$;
3. While $\ell \leqslant L$.

5.6.3.2. Simulation Results

In this section, the OFB described in the example of Section 5.6.2.1 is considered. Recall that the polyphase representations of this OFB and the associated parity-check filter are

$$\mathbf{H}(z) = \sum_{k=0}^{1} \mathbf{H}_k z^{-k}$$

and

$$\mathbf{P}(z) = \sum_{k=0}^{1} \mathbf{P}_k z^{-k},$$

where the matrices $\mathbf{H}_k$ and $\mathbf{P}_k$, $k = 0, 1$ are defined in (5.49) and (5.50) respectively.

Two types of signals are considered: a discrete-valued signal formed by 4 lines of the image `Lena.pgm` and a continuous-valued signal corresponding to the realizations of a zero-mean unit variance correlated Gaussian signal, with a correlation coefficient $c = 0.9$. The first $NV = 2000$ samples are kept for each signal, with $V = 500$. The output vector $\mathbf{y}$ is then of length $MV = 3000$. The components of the vector $\mathbf{y}^i$, $i = 1, \ldots, V$ are quantized with a rate $\rho = 4$. A BPSK modulation of the resulting quantized indexes is then performed before their transmission over an AWGN channel with an SNR level between 6 dB and 11 dB.

The iterative algorithm described in Section 5.6.3.1 is used. The obtained results are averaged over 250 noise realizations for both signals.

Figures 5.9 and 5.10 show the average SNR of the reconstructed signals as function of the channel SNR. The noiseless signal reconstructed after inverse quantization (in blue) serves as reference. The signal estimated using the algorithm presented in Section 5.6.3.1 (in green) has a higher SNR than the one estimated using the hard decisions of the channel (in yellow). The gains reach up to 10 dB in terms of reconstruction SNR and up to 3 dB in terms of channel SNR.

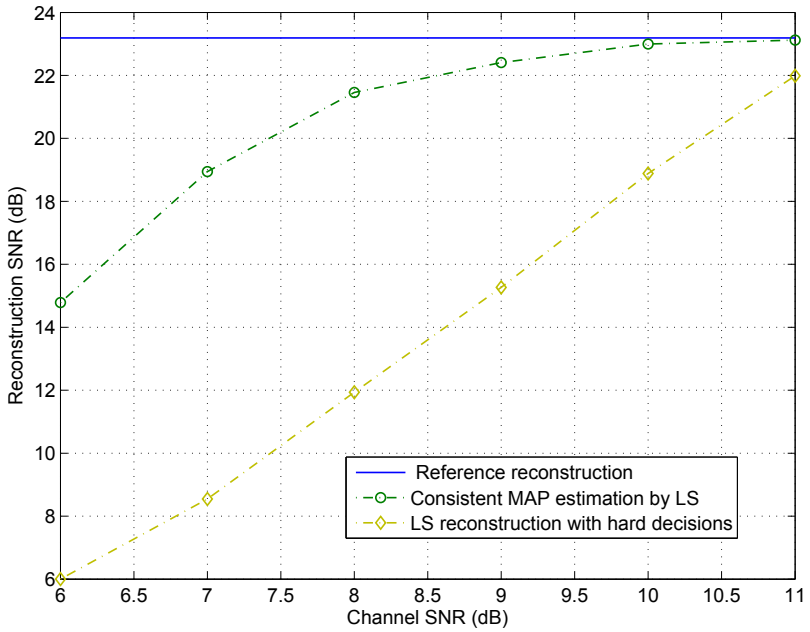


Figure 5.9.: SNR of the reconstructed signals as function of the channel SNR, for the 4 lines of `Lena.pgm`.

5.7. Conclusion

In this chapter, we have considered the problem of estimating a vector $\mathbf{x}$ from its quantized and noisy linear measurements. As the complexity of the optimal MAP estimator is intractable in general, two different suboptimal solutions have been proposed. The first approach exploits the linear dependencies between the entries of the output vector, as well as the fact that the quantization noise is bounded, to perform a consistent estimation of the input vector $\mathbf{x}$. Leveraging on techniques from interval analysis, it is possible to quickly eliminate solutions which are not consistent. The second technique uses the belief propagation, implemented by a message passing algorithm on a factor graph, to evaluate the a posteriori probability distributions of each entry of $\mathbf{x}$.

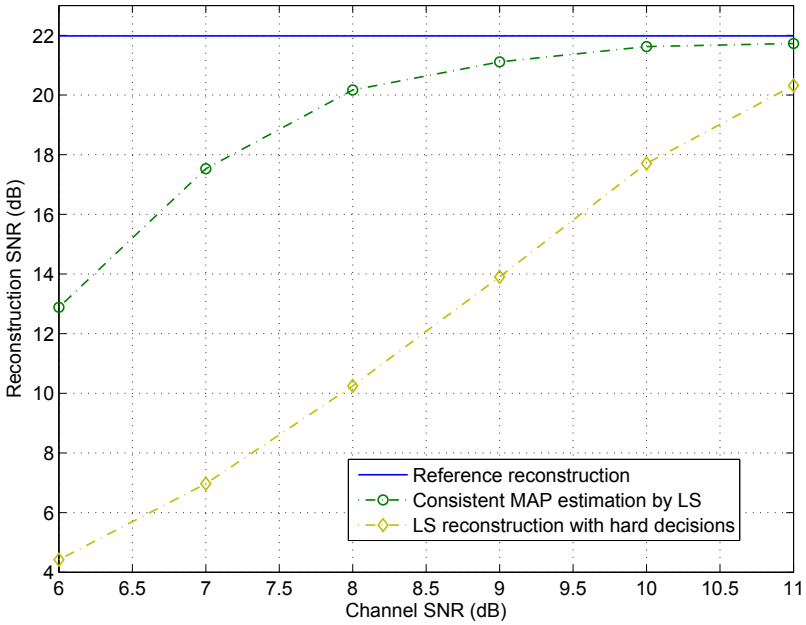


Figure 5.10.: SNR (dB) of the reconstructed signal as function of the channel SNR (dB) for the continuous-valued one-dimensional signal.

The proposed techniques are generic and may find application in the schemes performing signal expansion. Such schemes are, for example, oversampled filter banks, which provide an overcomplete representation of the signal placed at its input. We have shown that, for these applications, the linear mixing matrix has a particular structure, which leads to practical implementations of the consistent MAP estimator.

6. Conclusions and Perspectives

The extensive usage of heterogeneous best-effort networks puts the transmission reliability and the processing complexity at the core of any communication system. During the few past years, increasing efforts have been dedicated to the development of practical joint source-channel coding and decoding schemes responding to these issues.

In this thesis, we investigated some of these schemes and mainly focused on proposing techniques to increase the robustness of multimedia contents transmitted over unreliable networks, with limited transmission delay and complexity.

To conclude this dissertation, we provide in this chapter a synthesis of our contributions, before opening up some perspectives to the pursuit of this work.

6.1. Synthesis of the Contributions

Joint Source-Channel Decoding

We have proposed an efficient joint source-channel decoder for an error-resilient video transmission. The proposed joint decoder employs the residual redundancy left in the bit stream by the video coder, combined with bit reliability information provided by the channel, in order to detect and correct transmission errors. We have shown that this technique leads to a manageable complexity, while significantly improving the decoding performance.

Application to Multiple Description Coding

We applied the joint source-channel decoder to multiple description video streams transmitted over a mixed architecture, consisting of a wired lossy network and a wireless noisy channel. The joint decoding increases the number of error-free packets received by the multiple description decoder. As a consequence, the lost packets are better compensated and the decoder performance is improved.

Robust Estimation Techniques

Also, we focused on studying joint source-channel coding schemes relying on a redundant transform.

We proposed a consistent estimation technique exploiting the structured redundancy introduced by such schemes and accounting for the bounded nature of the quantization noise. More precisely, we have shown how the estimation problem can be formulated as a constrained optimization problem, to which suboptimal solutions may be delivered by solving several linear programs or by using tools from interval analysis. In the presence of transmission errors, we illustrated how this estimation scheme improves the decoding performance when compared to a classical approach based on the hard decisions of the channel.

Further, we applied this consistent estimation to recover the input of an oversampled filter bank from its noisy subbands. We proposed an iterative implementation of this estimation scheme, particularly adapted to oversampled filter banks.

Alternatively, we proposed to apply the belief propagation algorithm, widely used in inference problems, to estimate the signals placed at the input of an oversampled filter bank. More precisely, we used belief propagation to compute the probability distribution of each entry of the input vector, knowing the noisy subbands. We have shown that this approach is particularly efficient when the noise introduced by the channel is significant.

6.2. Perspectives

At the end of this thesis, several directions may be investigated to pursue this work and open up some new perspectives for future development.

First of all, the joint source-channel decoding scheme may be improved by exploiting other sources of redundancy. One may even consider to add some supplementary redundancy on the transmitted packets in the corresponding headers, such as the number of non-zero coefficients within a spatio-temporal subband, the check-sum of some video frame or a part of it, etc. The amount of this information may be chosen according to the importance of the transmitted packets. Moreover, an optimization of the decoding complexity as a function of the packets sensitivity may also be considered. More precisely, the number of candidates examined by the joint decoder may be adapted to the current layer and to the spatio-temporal subband of the processed block.

The application of the proposed joint source-channel decoder to multiple description coding has proven to be efficient for small redundancy levels. Indeed, combining both error correcting and packet erasure codes should increase the performance of classical single description schemes, at higher redundancy levels. The multiple description

scheme could further benefit from this redundancy increase, by duplicating, for example, some spatio-temporal subbands in both descriptions, to better compensate the effect of packet losses.

To connect our work on redundant joint source channel coding schemes and on joint source-channel decoding, a direction to further investigate is how to use the proposed consistent estimation technique and the belief propagation algorithm to improve the multiple description decoding. Indeed, the two descriptions are generated by an oversampled filter bank in the temporal domain. It is thus possible to exploit the structured redundancy introduced by this filter bank jointly with the residual redundancy, in order to improve the joint decoding of the descriptions.

More generally, a great deal of effort still needs to be made to optimize the proposed estimation algorithms in the case of oversampled filter banks with a dense analysis matrix and a reduced oversampling ratio. In particular, the estimation algorithm based on belief propagation may be improved by including the parity-check constraints in the factor graph. Finally, in order to apply these estimation approaches to image and video coding applications, it seems necessary to study its performance when oversampled filter banks with high coding gains are considered.

A. Sum-Product Algorithm

Most of practical algorithms have to deal with multivariables *global* functions which are very complex to handle. Usually such *global* functions admit a factorization into a product of *local* functions, each one depending on a subset of variables, which are easier to handle. Such a factorization can be represented by a bipartite graph, called factor graph [81, 95].

It is shown in [81, 95] how a single generic algorithm, namely the sum-product algorithm (SPA), can encompass a large variety of practical algorithms dealing with complicated global functions. The SPA operates by *message-passing* in the factor graph associated with the global function to compute its different marginal functions.

The aim of this appendix is to briefly present the SPA and to show how the BP algorithm presented in Section 5.4.1 of Chapter 5 can be regarded as a particular instance of the SPA.

A.1. Problem Formulation

Consider a global function $g(x_1, \ldots, x_N)$ of N variables $x_n \in A_n$ where A_n is some *domain* or alphabet. Assuming that the summation operation is well defined for g, one may introduce the marginal functions g_n, $n = 1, \ldots, N$ associated with g as follows

$$\begin{aligned} g_n(x_n) &= \sum_{x_1 \in A_1} \cdots \sum_{x_{n-1} \in A_{n-1}} \sum_{x_{n+1} \in A_{n+1}} \cdots \sum_{x_1 \in A_N} g(x_1, \ldots, x_N) \\ &= \sum_{\sim\{x_n\}} g(x_1, \ldots, x_N), \end{aligned} \tag{A.1}$$

where the notation $\sum_{\sim\{x_n\}}$ has been introduced in [81] to indicate that the summation is over all the variables except x_n.

Assume now that g factors into a product of local functions f_j as follows

$$g(x_1, \ldots, x_N) = \prod_{j \in J} f_j(X_j), \tag{A.2}$$

where J is a discrete set of indexes and $X_j \subset \{x_1, \ldots, x_N\}$. The factorization (A.2) may be visualized by a bipartite factor graph $\mathcal{G}$ which is formed by two types of

nodes, the *variable* nodes (VN) representing x_n, $n = 1, \ldots, N$ and the *factor* nodes (FN) representing f_j, $j \in J$. An edge between the VN x and the factor node f signifies that the variable x is an argument of the local function f.

A.1.1. Simple Example

Consider for example a global function $g(x_1, x_2, x_3, x_4, x_5)$ of 5 variables which admits the following factorization

$$g(x_1, x_2, x_3, x_4, x_5) = f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4) f_E(x_3, x_5).$$

Therefore, the index set $J = \{A, B, C, D, E\}$ and the local functions f_A, f_B, f_C, f_D, f_E of g take their values respectively in $X_A = \{x_1\}$, $X_B = \{x_2\}$, $X_C = \{x_1, x_2, x_3\}$, $X_D = \{x_3, x_4\}$, $X_E = \{x_3, x_5\}$. The factor graph $\mathcal{G}$ corresponding to g is represented in Figure A.1.

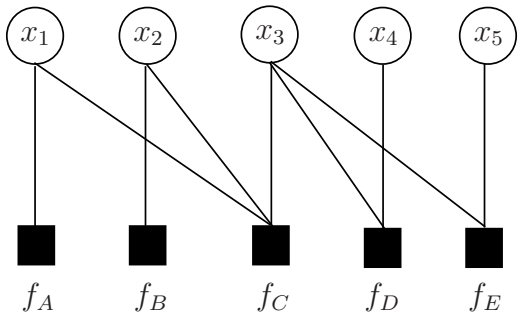


Figure A.1.: Factor graph associated to the global function g.

A.2. Computation of the Marginals

The aim of the SPA is to compute the marginals $g_n(x_n)$ defined in (A.1) by exchanging a set of *messages* between the variable nodes and the factor nodes of $\mathcal{G}$. By *message*, one means a given representation or description of the marginal functions, *e.g*, the a posteriori probability distribution of x_n, the mean and variances of x_n, etc. The message sent by a variable node x to a factor node f is denoted by $\mu_{f \leftarrow x}(x)$ and the message sent by the factor node f to the variable node x is denoted by $\mu_{f \rightarrow x}(x)$.

For cycle-free graphs, authors in [81] showed how each single marginal function $g_n(x_n)$ may be computed separately by using the graph representation in a tree of root x_n. The message passing procedure starts at the leaves nodes. Each leaf VN sends the trivial unit function to its parent, and each leaf FN f sends a description of f to its parent. Each non-leaf node waits for the messages of all its children before

computing the message to be sent to its parent. The message computation follows a single rule which will be introduced further in this section. The computation terminates at the root x_n where the marginal $g_n(x_n)$ is obtained as the *product* of all the messages received at x_n. This algorithm is referred to as the single$-n$ sum-product algorithm.

The marginals $g_n(x_n)$ may also be computed simultaneously, which is more efficient than computing each marginal separately. This algorithm is refereed to as the sum-product algorithm. All the possible instances of the single$-n$ sum-product algorithm are *overlayed* on the graph $\mathcal{G}$ and the message passing procedure is initialized at the leaves nodes as in the single$-n$ sum-product algorithm. Each non-leaf node u remains idle until it has received messages coming from all but one of its neighboring nodes v. The node u is then able to compute the message to be sent to node v which is temporarily considered as its parent. Once this message is sent, node u returns to the idle state until a *return message* arrives from v. Once this message arrives, node u sends it to each of its neighbors other than v, each being regarded in turn as a parent of u. The algorithm terminates when two messages have been exchanged along each edge of $\mathcal{G}$, one in each direction. The marginals $g_n(x_n)$ are then computed as in the case of the single$-n$ sum-product algorithm.

The computation of the messages passed along the edges of $\mathcal{G}$ is performed according to the following rule [81]:

The Sum-Product update rule: *The message sent from a node v on an edge e is the product of the local function at v (or the unit function if v is a variable node) with all messages received at v on edges* other *than e, summarized for the variable associated with e.*

According to the SPA update rule, the message $\mu_{f\leftarrow x}(x)$ is computed as follows

$$\mu_{f\leftarrow x}(x) = \prod_{h\in\mathcal{N}(x)\setminus\{f\}} \mu_{h\to x}(x), \tag{A.3}$$

where $\mathcal{N}(x)$ is the set of neighbors of x and the message $\mu_{f\to x}(x)$ is obtained as follows

$$\mu_{f\to x}(x) = \sum_{\sim\{x\}} \left(f(X) \prod_{y\in\mathcal{N}(f)\setminus\{x\}} \mu_{f\leftarrow y}(y) \right), \tag{A.4}$$

where $X = \mathcal{N}(f)$ is the set of neighbors of the FN f, *i.e*, its set of arguments.

Here, the product or summary of the messages involved in (A.3) and (A.4), shall be understood as appropriate descriptions of the pointwise product or summary. For example, if the messages are parametrization of the functions, then the product of theses messages is the parametrization of the resulting product function and not necessarily the product of the messages themselves.

Figures A.2 and A.3 illustrate the computation of $\mu_{f\leftarrow x}(x)$ and $\mu_{f\to x}(x)$ respectively.

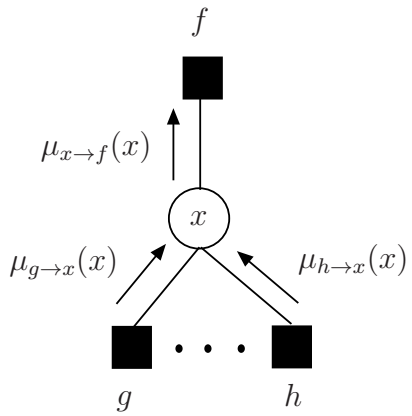


Figure A.2.: Message computation at the variable node x.

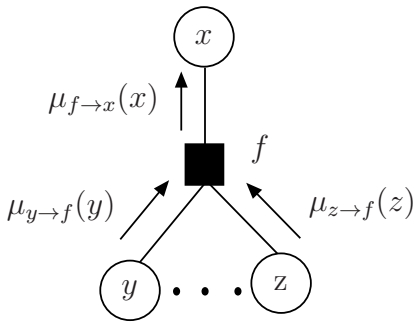


Figure A.3.: Message computation at the factor node f.

The update rule at the VN x takes the simple form in (A.3) because there is no local function to consider [1] and the summary $\sum_{\sim\{x\}}$ over a product of functions of x is equal to that product. The computation of $\mu_{f\to x}(x)$ in (A.4) is less trivial, since it involves function multiplications, followed by the summary operator $\sum_{\sim\{x\}}$.

Finally the marginal functions are exactly obtained as follows

$$g_n(x_n) = \prod_{f\in\mathcal{N}(x_n)} \mu_{f\to x_n}(x_n). \tag{A.5}$$

For a more detailed analysis of the SPA update rule and the computation formula (A.5) which gives the exact value of g_n in the cycle-free case, one may refer to [81] (Appendix-A).

A.2.1. Detailed Example

The computation of the marginals associated with the function g presented in the example of A.1.1 is illustrated in Figure A.4, where all the steps of SPA are detailed.

[1] the local function taken here is the unit function

A.3. Sum-Product Algorithm for Factor Graphs with Cycles

The SPA presented so far considered cycle-free factor graphs. In the case of factor graphs with cycles, the computations introduced in the cycle-free case may be applied by following the same update SPA rule, since all the updates are local. However due to the presence of cycles, the SPA has no natural termination and the messages may be passed infinitely along the same edge. The exact summary in (A.4) usually includes an infinite number of terms.

The number of iterations of the SPA is usually decided according to whether the algorithm has converged or not. The convergence occurs when identical messages are obtained between two successive iterations. Nonetheless, the major problem of graphs with cycles is that the convergence is usually not perfect, messages may oscillate between a set of values or states, or the convergence may be obtained for a wrong fixed point. Therefore, unlike the cycle-free case, the results obtained when operating the SPA on graphs with cycles are not considered as exact.

In some applications, *e.g*, Markov chains, the associated factor graphs are naturally cycle-free. In other applications, *e.g*, decoding of LDPC and Turbo codes, the underlying graphs contain cycles but the algorithm performances can still achieve very good performance as shown in [14, 97].

It is shown in [81] how equivalent cycle-free representations may be obtained from a factor graph with cycles, *e.g* for Bayesian networks. Such representations are usually achieved by operating a number of graph transformations. Note however that the cycles elimination comes at the expense of an increase of the complexity. In [167], two modified versions of belief-propagation decoding, referred to as normalized belief-propagation and offset belief-propagation are proposed to improve the performance of BP in graphs with cycles.

A.4. Belief Propagation

The SPA described in Section A.1 can encompass a wide number of practical algorithms such as the forward/backward algorithm, the Viterbi algorithm, the belief propagation (BP) algorithm, *etc* [81].

In this section its is shown how BP presented in Section 5.4.1 of Chapter 5, may be derived as a particular instance of the SPA to compute the a posteriori probability distributions (APPDs) $p(x_n|\mathbf{r}_{1:M})$.

The function to maximize in the MAP estimation problem of Chapter 5, is recalled

$$\begin{aligned} g(x_1,\ldots,x_N) &= p(x_1,\ldots,x_N|\mathbf{r}_{1:M}) \\ &= p(x_1,\ldots,x_N)p(\mathbf{r}_{1:M}|x_1,\ldots,x_N) \end{aligned}$$

Assuming that the variables x_n are independent with prior $p_{X_n}(x_n)$, the joint APPD g admits the following factorization

$$g(x_1, \ldots, x_N) = \prod_{n=1}^{N} p_{X_n}(x_n) \prod_{m=1}^{M} p\left(\mathbf{r}_m | \sum_{n=1}^{N} \mathbf{H}(m,n) x_n\right). \tag{A.6}$$

The graph $\mathcal{G}_H$ associated to the factorization (A.2) has N variables nodes x_n, M factor nodes f_m, each of which represents the likelihood function $p\left(\mathbf{r}_m | \sum_{n=1}^{N} \mathbf{H}(m,n) x_n\right)$ and N (leaf) factor nodes h_n which represent the *a priori* distribution of the variables x_n. Note, that these N factor nodes have not been mentioned in Section 5.4.1 of Chapter 5, for simplicity sake. However they are implicitly taken into account in the initialization step of the presented BP algorithm.

In deed, the SPA starts at these FN since they are the (only) leaves of $\mathcal{G}_H$. Each node h_n sends the message $p_{X_n}(x_n)$ to x_n. At this level, there is no computation to be made, these messages are simply transferred from each VN x_n to its neighboring FN f_m as described in (5.35) .

From now on, the nodes x_n and f_m are refereed to as n and m respectively. Moreover, the messages $\mu_{f_m \leftarrow x_n}(x_n)$ and $\mu_{f_m \rightarrow x_n}(x_n)$ exchanged at iteration i of the BP algorithm are denoted by $p^x_{m \leftarrow n}(i, x_n)$ and $p^u_{m \rightarrow n}(i, u_m = u_m)$ respectively (with $u_m = H(m,n)x_n$), as introduced in Section 5.4.1, to refer to the fact that these messages represent PDFs.

At the level of each FN m, the messages product $\prod_{n \in \mathcal{N}(m)} p^x_{m \leftarrow n}(i, x_n)$ is defined here as the distribution $p_{m \rightarrow n}(i, y_{m \rightarrow n})$ of the random variable $y_{m \rightarrow n}$ defined in (5.36).

The function $f_m(X)$, $X = \mathcal{N}(f_m)$ defined in (A.4) is the likelihood $p\left(\mathbf{r}_m | \sum_{n=1}^{N} \mathbf{H}(m,n) x_n\right)$ which takes the elements of $\mathcal{N}(m)$ as argument, $\mathbf{r}_m$ being a fixed parameter of this function.

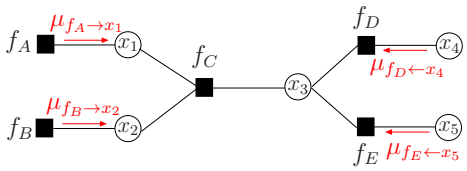
The message computation in (A.4) is performed as described by (5.38), where the summary operator $\sum_{\sim\{x_n\}}$ is defined as the integration over the (continuous) random variable $y_{m \rightarrow n}$.

At the level of each VN node n, the message computation in (A.3) is performed as described by (5.38), where the actual product of the messages is performed[2]. The same observation is made for the computation (A.5) of the marginals $g_n(x_n) = p(x_n | \mathbf{r}_{1:M})$ which performed in (5.39).

[2]It is the product of likelihoods sent by the FN nodes $m' \in \mathcal{N}(n) \backslash \{m\}$

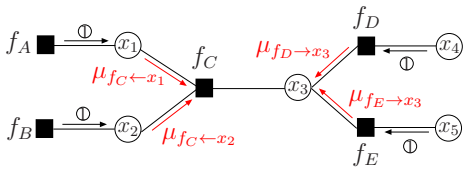
Step 1 (initialization)

$$\mu_{f_A \to x_1}(x_1) = \sum_{\sim\{x_1\}} f_A(x_1) = f_A(x_1)$$
$$\mu_{f_B \to x_2}(x_2) = \sum_{\sim\{x_2\}} f_B(x_2) = f_B(x_2)$$
$$\mu_{f_D \leftarrow x_4}(x_4) = 1$$
$$\mu_{f_E \leftarrow x_5}(x_5) = 1$$



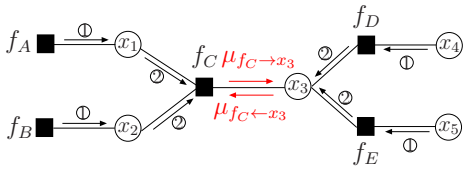
Step 2

$$\mu_{f_C \leftarrow x_1}(x_1) = \mu_{f_A \to x_1}(x_1)$$
$$\mu_{f_C \leftarrow x_2}(x_2) = \mu_{f_B \to x_2}(x_2)$$
$$\mu_{f_D \to x_3}(x_3) = \sum_{\sim\{x_3\}} f_D(x_3, x_4)\mu_{f_D \leftarrow x_4}(x_4)$$
$$\mu_{f_E \to x_3}(x_3) = \sum_{\sim\{x_3\}} f_E(x_3, x_5)\mu_{f_E \leftarrow x_5}(x_5)$$



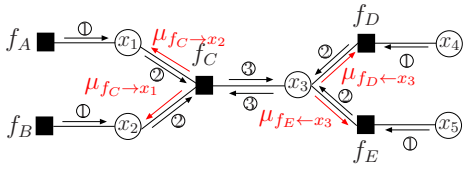
Step 3

$$\mu_{f_C \to x_3}(x_3) = \sum_{\sim\{x_3\}} f_C(x_1, x_2, x_3)\mu_{f_C \leftarrow x_1}(x_1)\mu_{f_C \leftarrow x_2}(x_2)$$
$$\mu_{f_C \leftarrow x_3}(x_3) = \mu_{f_D \to x_3}(x_3)\mu_{f_E \to x_3}(x_3)$$



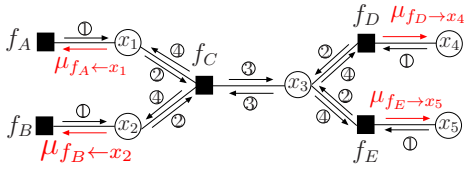
Step 4

$$\mu_{f_C \to x_1}(x_1) = \sum_{\sim\{x_1\}} f_C(x_1, x_2, x_3)\mu_{f_C \leftarrow x_2}(x_2)\mu_{f_C \leftarrow x_3}(x_3)$$
$$\mu_{f_C \to x_2}(x_2) = \sum_{\sim\{x_2\}} f_C(x_1, x_2, x_3)\mu_{f_C \leftarrow x_1}(x_1)\mu_{f_C \leftarrow x_3}(x_3)$$
$$\mu_{f_D \leftarrow x_3}(x_3) = \mu_{f_C \to x_3}(x_3)\mu_{f_E \to x_3}(x_3)$$
$$\mu_{f_E \leftarrow x_3}(x_3) = \mu_{f_C \to x_3}(x_3)\mu_{f_D \to x_3}(x_3)$$



Step 5

$$\mu_{f_A \leftarrow x_1}(x_1) = \mu_{f_C \to x_1}(x_1)$$
$$\mu_{f_B \leftarrow x_2}(x_2) = \mu_{f_C \to x_2}(x_2)$$
$$\mu_{f_D \to x_4}(x_4) = \sum_{\sim\{x_4\}} f_D(x_3, x_4)\mu_{f_D \leftarrow x_3}(x_3)$$
$$\mu_{f_E \to x_5}(x_5) = \sum_{\sim\{x_5\}} f_E(x_3, x_5)\mu_{f_E \leftarrow x_3}(x_3)$$



Step 6 (End)

$$g_1(x_1) = \mu_{f_A \to x_1}(x_1)\mu_{f_C \to x_1}(x_1)$$
$$g_2(x_2) = \mu_{f_B \to x_2}(x_2)\mu_{f_C \to x_2}(x_2)$$
$$g_3(x_3) = \mu_{f_C \to x_3}(x_3)\mu_{f_D \to x_3}(x_3)\mu_{f_E \to x_3}(x_3)$$
$$g_4(x_4) = \mu_{f_D \to x_4}(x_4)$$
$$g_5(x_5) = \mu_{f_E \to x_5}(x_5)$$

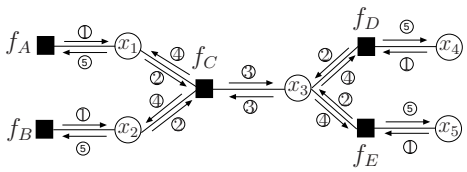


Figure A.4.: Example of the computation steps of the SPA.

B. Oversampled Filter Banks

Oversampled filter banks [151] are filter banks whose number of output subbands is larger than the decimation factor. These filter banks provide several advantages over the critically sampled filter banks, such as increased design freedom [16], noise shaping and reduction [16, 17], as well as robustness to erasures and transmission errors [56, 131, 54, 79, 86, 130, 2].

This appendix briefly introduces the oversampled filter banks and some of their properties. The aim is to show that the estimation schemes presented in Chapter 5 may be applied to these filter banks by exploiting the overcomplete representation they provide.

In Section B.1 a description of the oversampled filter banks in both time and polyphase domains is provided. In Section B.2, the parallelism between these filter banks and error-correcting codes is evidenced.

B.1. Introduction to Oversampled Filter Banks

B.1.1. Time Domain Analysis

Consider an $M-$band filter bank (FB), wherein each band is subsampled by an integer factor N. The analysis and synthesis filters have the impulse responses $\{h_m(n)\}_{n\in\mathbb{Z}}$ and $\{g_m(n)\}_{n\in\mathbb{Z}}$ respectively, where $m = 1, \ldots, M$, see Figure B.1.

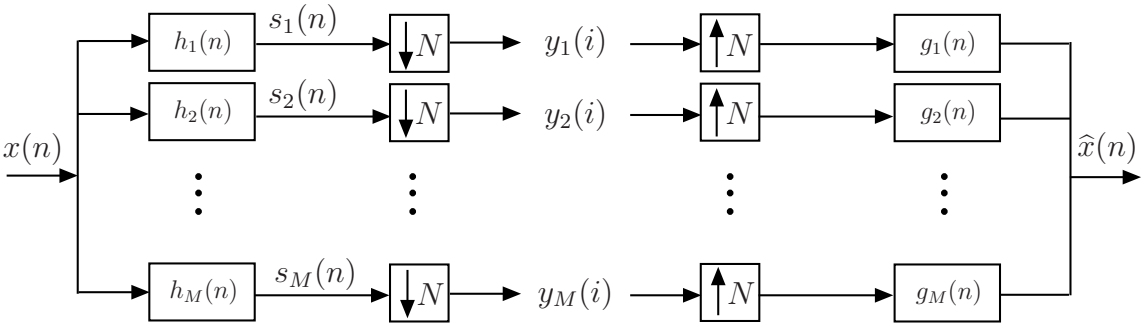


Figure B.1.: Block diagram of an $M-$band filter bank.

The input signal $x(n)$, $n \in \mathbb{N}$, goes through the M analysis filters and M output signals $s_m(n)$, $m = 1, \ldots, M$ are obtained. The $n-$th sample of the $m-$th analysis filter output, $s_m(n)$ is expressed as

$$s_m(n) = \sum_{\ell=-\infty}^{+\infty} h_m(\ell) x(n-\ell),\ n \in \mathbb{N}. \tag{B.1}$$

After downsampling by N, downsampled subband signals are obtained

$$y_m(i) = \sum_{\ell=-\infty}^{+\infty} h_m(\ell) x(iN-\ell),\ i \in \mathbb{N}. \tag{B.2}$$

The reconstructed signal $\widehat{x}(n)$, at the synthesis stage, is obtained as follows

$$\widehat{x}(n) = \sum_{m=0}^{M-1} \sum_{i=-\infty}^{+\infty} y_m(i) g_m(n-iN),\ n \in \mathbb{N}. \tag{B.3}$$

In critically sampled FBs, one has $N = M$ and the subband signals $y_m(i)$, $i \in \mathbb{N}$, $m = 1, \ldots, M$ contain exactly the same number of samples, per unit of time, as the input signal $x(n)$, $n \in \mathbb{N}$.

In the case of oversampled filter banks (OFBs), $M > N$ and the output subbands form a redundant representation of the input signal.

When the analysis filters h_m, $m = 1, \ldots, M$ are of finite impulse responses (FIR) and of maximal length L, (B.2) becomes

$$y_m(i) = \sum_{\ell=0}^{L-1} h_m(\ell) x(iN-\ell),\ i \in \mathbb{N}. \tag{B.4}$$

Without loss of generality, one can take L as the smallest multiple of N that is larger than the largest length of the filters impulse responses (zeros may be added to impulse responses if necessary). Then $L = N \times K$, where K is some positive integer. Consider the $M \times N$ matrices

$$\mathbf{H}_k = \begin{pmatrix} h_1(kN+N-1) & \cdots & h_1(kN) \\ \vdots & \ddots & \vdots \\ h_M(kN+N-1) & \cdots & h_M(kN) \end{pmatrix},\ k = 0, \ldots, K-1. \tag{B.5}$$

The temporal relation between the input and the output of the OFB, at time instant $i \in \mathbb{N}$, can be written as

$$\mathbf{y}^i = \sum_{k=0}^{K-1} \mathbf{H}_k \mathbf{x}^{i-k}, \tag{B.6}$$

where

$$\mathbf{x}^i = (x(iN-N+1), \ldots, x(iN))^T$$

and

$$\mathbf{y}^i = (y(iM - M + 1), \ldots, y(iM))^T,$$

with $y(iM - m) = y_{M-m}(i)$, $m = 0, \ldots, M - 1$.

The vectors $\mathbf{x}^i$ and $\mathbf{y}^i$ contain the polyphase components at time instant i of the input signal $\mathbf{x}$ and the output subband signal $\mathbf{y}$, respectively. The relation in (B.6) may be rewritten as follows

$$\mathbf{y}^i = \mathbf{H}_{K-1:0}\mathbf{x}^{i-K+1:i}, \tag{B.7}$$

where $\mathbf{H}_{K-1:0} = (\mathbf{H}_{K-1}, \ldots, \mathbf{H}_0)$ is the $M \times KN$ matrix formed by the matrices $\mathbf{H}_k$, $k = 0, \ldots, K - 1$, defined in (B.5) and the vector

$$\mathbf{x}^{i-K+1:i} = \left(\left(\mathbf{x}^{i-K+1}\right)^T, \ldots, \left(\mathbf{x}^i\right)^T\right)^T \tag{B.8}$$

contains all the input samples affecting the OFB output at time instant i.

In (B.7), the oversampling character of the filtering operation is not obvious due to the length L of the analysis filters, which may exceed the length M of the output vector $\mathbf{y}^i$. To illustrate the redundancy introduction, consider $J > K$ successive output vectors $\mathbf{y}^{i-J+1}, \ldots, \mathbf{y}^i$ gathered in the vector $\mathbf{y}^{i-J+1:i}$ of size JM, which is defined similarly to (B.8), as follows

$$\mathbf{y}^{i-J+1:i} = \left(\left(\mathbf{y}^{i-J+1}\right)^T, \ldots, \left(\mathbf{y}^i\right)^T\right)^T. \tag{B.9}$$

Using (B.7)–(B.9), one gets

$$\mathbf{y}^{i-J+1:i} = \mathcal{H}\mathbf{x}^{i-J+1-K+1:i}, \tag{B.10}$$

where the matrix

$$\mathcal{H} = \begin{pmatrix} \mathbf{H}_{K-1} & \cdots & \mathbf{H}_1 & \mathbf{H}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{K-1} & \cdots & \mathbf{H}_1 & \mathbf{H}_0 & \mathbf{0} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{H}_{K-1} & \cdots & \mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix} \tag{B.11}$$

is of size $JM \times (J + K - 1)N$. When enough output vectors are gathered in (B.9), *i.e.*, $J > (K - 1)N/(M - N)$, the matrix $\mathcal{H}$ becomes *tall*, *i.e*, with more rows than columns and the redundancy introduction performed by OFBs becomes more evident.

The time-domain analysis of OFBs presented in this section aims at providing the basic filtering operations carried out on the input signal in the time domain. In the next section a similar, but more general, analysis of OFBs is carried out in the $z-$domain, by using the polyphase representation of OFBs.

B.1.2. Polyphase Domain Analysis

Polyphase representation of FBs is a classical description of the input-output relationships of FBs [151]. This representation simplifies computations and leads to efficient implementations. Figures B.2 and B.3 illustrate the polyphase representations $\mathbf{H}(z)$ and $\mathbf{G}(z)$ of given analysis and synthesis FBs, respectively. In these representations, the downsampling and upsampling operations are performed before analysis and synthesis filtering, which simplifies the computations, *e.g.*, as it can be seen in (B.2), where the computation of $s_m(n)$ is not necessary when n is not a multiple of N.

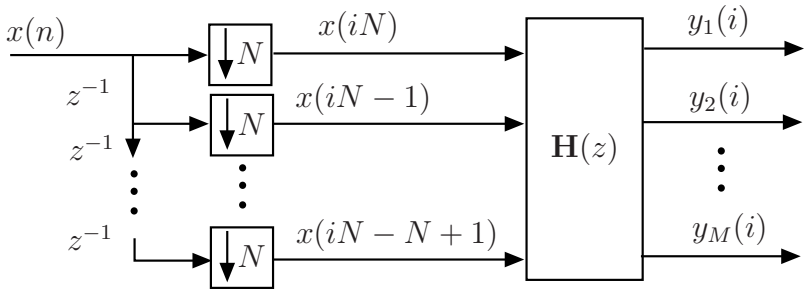


Figure B.2.: Polyphase representation of an OFB at the analysis stage.

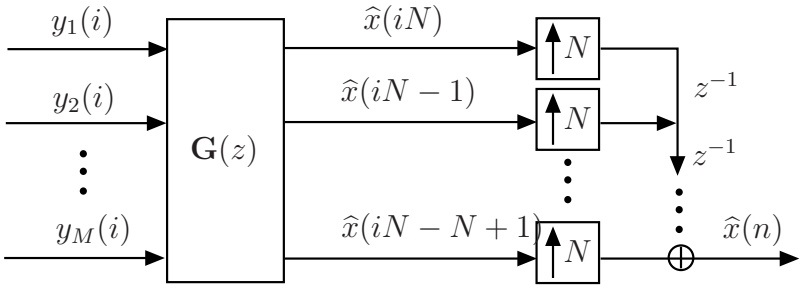


Figure B.3.: Polyphase representation of an OFB at the synthesis stage.

The $M \times N$ polyphase analysis matrix $\mathbf{H}(z)$ is expressed as

$$\mathbf{H}(z) = \sum_{k=-\infty}^{+\infty} \mathbf{H}_k z^{-k}, \tag{B.12}$$

and the relation between the input $\mathbf{X}(z)$ and the output $\mathbf{Y}(z)$ of the OFB in the polyphase domain is given by

$$\mathbf{Y}(z) = \mathbf{H}(z)\mathbf{X}(z). \tag{B.13}$$

The relation (B.13) is more general than (B.10), as it is not limited to the FIR case.

In a similar way, the $N \times M$ polyphase matrix $\mathbf{G}(z)$ at the synthesis stage is expressed as

$$\mathbf{G}(z) = \sum_{k=-\infty}^{+\infty} \mathbf{G}_k z^{-k}, \tag{B.14}$$

where

$$\mathbf{G}_k = \begin{pmatrix} g_1(kN) & \cdots & g_M(kN) \\ \vdots & \ddots & \vdots \\ g_1(kN+N-1) & \cdots & g_M(kN+N-1) \end{pmatrix}, \tag{B.15}$$

and the reconstructed signal is obtained as follows

$$\widehat{\mathbf{X}}(z) = \mathbf{G}(z)\mathbf{Y}(z). \tag{B.16}$$

B.1.3. Perfect Reconstruction

A filter bank satisfies the perfect reconstruction property $\widehat{\mathbf{X}}(z) = \mathbf{X}(z)$[1] if

$$\mathbf{G}(z)\mathbf{H}(z) = \mathbf{I}_N, \tag{B.17}$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix.

In the case of critically sampled FBs, the matrices $\mathbf{H}(z)$ and $\mathbf{G}(z)$ are $N \times N$ square matrices of full rank N and (B.17) admits a unique solution $\mathbf{G}(z) = \mathbf{H}^{-1}(z)$ [151, 158, 31].

For OFBs, the matrix $\mathbf{H}(z)$ is tall and perfect reconstruction is satisfied if $\mathbf{H}(z)$ has full normal column rank N [31]. In this case, the solution of (B.17) is not uniquely determined and any inverse of $\mathbf{H}(z)$ is a valid solution. It has been shown in [10, 92, 18], that any left inverse $\mathbf{G}(z)$ of $\mathbf{H}(z)$ can be written as follows

$$\mathbf{G}(z) = \widehat{\mathbf{G}}(z) + \mathbf{R}(z)\left(\mathbf{I}_M - \mathbf{H}(z)\widehat{\mathbf{G}}(z)\right), \tag{B.18}$$

where $\widehat{\mathbf{G}}(z)$ is any particular solution of (B.17) and $\mathbf{R}(z)$ is any $N \times M$ matrix, whose elements $(\mathbf{R}(z))_{n,m}$ satisfy $\left|(\mathbf{R}(z))_{n,m}\right| < \infty$. Using the fact that $\widehat{\mathbf{G}}(z)\mathbf{H}(z) = \mathbf{I}_N$, it is easy to see that $\mathbf{G}(z)$ defined in (B.18) is a solution of (B.17).

In case of OFBs, the non-uniqueness of the synthesis FB for a given analysis FB $\mathbf{H}(z)$ offers freedom in the design of $\mathbf{G}(z)$ through the elements of $\mathbf{R}(z)$ [18]. This allows to construct synthesis FBs with desired properties, which is particularly useful in the case of erasure channels where the content of one or more subbands may be erased [79, 130, 2].

[1] It is assumed here that no delay is introduced, but the theory presented in this section can easily be extended to perfect reconstruction with non-zero delay.

Consider the matrix $\widetilde{\mathbf{H}}(z) = (\mathbf{H}^*(z^{-1}))^T$, where $(\cdot)^*$ stands for the complex conjugation operation. A particular synthesis FB, that can be used in (B.18), is

$$\widehat{\mathbf{G}}(z) = \left(\widetilde{\mathbf{H}}(z)\mathbf{H}(z)\right)^{-1}\widetilde{\mathbf{H}}(z), \tag{B.19}$$

which is called the para-pseudo-inverse of $\mathbf{H}(z)$.

The next section shows how an OFB with perfect reconstruction property can be assimilated to a channel code.

B.2. OFBs as Error Correcting Codes

A parallelism between OFBs and channel codes has been evidenced in [85, 86] by introducing the notions of generator matrix, code subspace and parity-check matrix, as well as the associated operators.

These operators are presented in the time domain in Section B.2.1, then they are described using the corresponding polyphase matrices in Section B.2.2.

B.2.1. Time Domain

In this section, the study is presented for real-valued signals and filters, but it can be extended to the general complex case.

Consider an integer $J > (K-1)N/(M-N)$. As presented in Section B.1.1, at time instant i, one has

$$\mathbf{y}^{i-J+1:i} = \mathcal{H}\mathbf{x}^{i-J+1-K+1:i},$$

where $\mathcal{H}$ is defined in (B.11). The input vector $\mathbf{x}^{i-J+1-K+1:i}$ span the whole real space $\mathbb{R}^{(J+K-1)N}$, whereas the output vector $\mathbf{y}^{i-J+1:i}$ belongs to the subspace $\mathcal{C}$ of $\mathbb{R}^{JM}$, generated by the columns of $\mathcal{H}$.

The matrix $\mathcal{H}$ may then be seen as the generator matrix of an error-correcting code in the real field. The subspace $\mathcal{C}$ is considered as the *code subspace* associated to the analysis filter bank for a given integer J, in the sense that any subband signal obtained from the generator matrix $\mathcal{H}$ should belong to $\mathcal{C}$

$$\mathbf{y} \in \mathcal{C} \iff \exists \mathbf{x} \in \mathbb{R}^{(J+K-1)N} \text{ st } \mathbf{y} = \mathcal{H}\mathbf{x}. \tag{B.20}$$

When the subband signal is corrupted by noise, a possible decoding approach consists in projecting the corrupted vector $\tilde{\mathbf{y}}^{i-J+1:i}$ onto $\mathcal{C}$ using the orthogonal projection operator $\Pi_{\mathcal{C}}$, defined in [86]

$$\Pi_{\mathcal{C}} = \mathcal{H}\left(\mathcal{H}^T\mathcal{H}\right)^{-1}\mathcal{H}^T \tag{B.21}$$

and verifying

$$\Pi_{\mathcal{C}}\mathbf{y} = \mathbf{y}, \forall \mathbf{y} \in \mathcal{C}. \tag{B.22}$$

The projection onto $\mathcal{C}$ is illustrated in Figure B.4, where $\mathcal{C}$ is a two-dimensional subspace of a three-dimensional space.

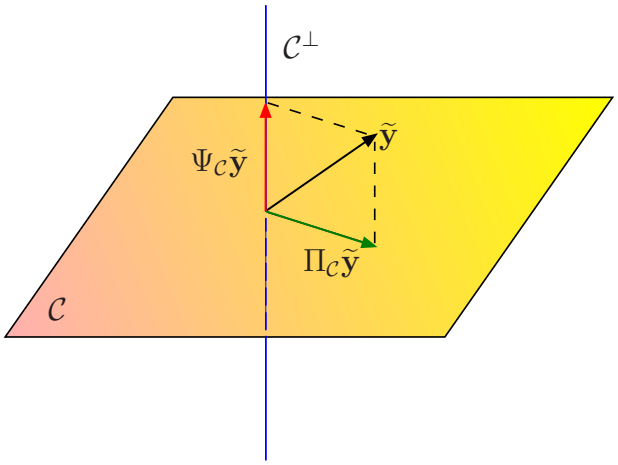


Figure B.4.: Illustration of the projection onto the subspace code $\mathcal{C}$ [87].

Similarly to error-correcting codes, the concept of parity-check operator $\Psi_{\mathcal{C}} = \Pi_{\mathcal{C}^{\perp}}$ is also defined in [86]

$$\Psi_{\mathcal{C}} = I_{JM} - \Pi_{C}, \tag{B.23}$$

where I_{JM} is the identity operator defined in the $JM-$dimensional space $\mathbb{R}^{JM}$. Using (B.22)–(B.23), one has

$$\Psi_{\mathcal{C}}\mathbf{y} = \mathbf{0}, \ \forall \mathbf{y} \in \mathcal{C}. \tag{B.24}$$

The syndromes $\mathbf{s} \in \mathbb{R}^{JM-(J+K-1)N}$ are then computed by applying $\Psi_{\mathcal{C}}$ on any received vector $\mathbf{y} \in \mathbb{R}^{JM}$

$$\mathbf{s} = \Psi_{\mathcal{C}}\mathbf{y}. \tag{B.25}$$

Similarly to channel coding

$$\Psi_{\mathcal{C}}\mathbf{y} \neq \mathbf{0} \Longrightarrow \mathbf{y} \notin \mathcal{C}. \tag{B.26}$$

Using (B.26), one may detect the occurence of transmission errors in the received subband vector $\mathbf{y}$.

B.2.2. Polyphase Domain

In Section B.2.1, the subspace interpretation of OFBs as error-correcting codes is illustrated by stacking up J successive subband vectors, which is somewhat restrictive.

In this section, the parallelism between OFBs and error-correction codes is drawn in the more general polyphase domain framework.

B.2.2.1. Smith-McMillan Decomposition

As mentioned in Section B.1.3, in the case of OFBs, the perfect reconstruction property is satisfied if $\mathbf{H}(z)$ has full column rank N. In this case, $\mathbf{H}(z)$ admits a Smith-McMillan decomposition [151] of the form

$$\mathbf{H}(z) = \mathbf{U}(z) \begin{pmatrix} \mathbf{\Lambda}(z) \\ \mathbf{0} \end{pmatrix} \mathbf{W}(z), \tag{B.27}$$

where $\mathbf{U}(z)$ and $\mathbf{W}(z)$ are unimodular[2] matrices of sizes $M \times M$ and $N \times N$ respectively, and $\mathbf{\Lambda}(z)$ is an $N \times N$ diagonal matrix, see Figure B.5.

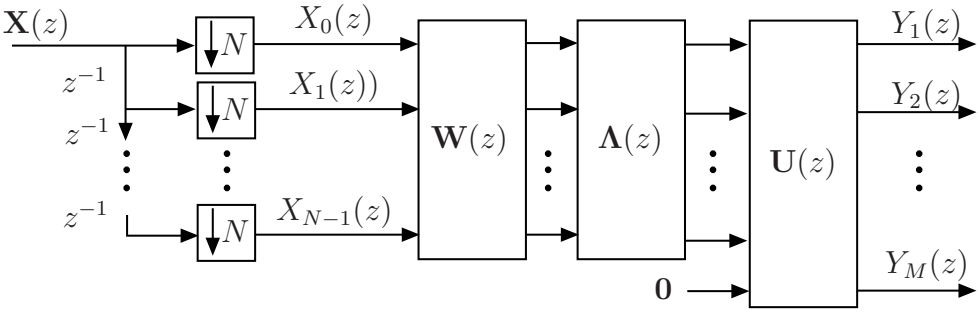


Figure B.5.: Illustration of the Smith-McMillan decomposition [85].

The decomposition (B.27) allows a more clear understanding of the different operations performed at the analysis stage of an OFB. First an $N \times N$ multiple input multiple output (MIMO) filter $\mathbf{\Lambda}(z)\mathbf{W}(z)$ transforms an N-input sequence $\mathbf{X}(z) = (X_1(z), \ldots, X_N(z))^T$ into a sequence of N subbands. These subbands are padded with $M - N$ null subbands. The M resulting subbands enter then an $M \times M$ MIMO filter $\mathbf{U}(z)$. At the output of $\mathbf{U}(z)$, an M-subband vector $\mathbf{Y}(z) = (Y_1(z), \ldots, Y_M(z))^T$ is obtained.

A mapping as the one illustrated by Figure B.5 cannot span the entire output space. The output subband vector $\mathbf{Y}(z)$ only spans a subspace $\mathcal{C}$ of $\mathbb{R}^M$ of dimension N.

[2]Unimodular matrices are polynomial matrices with polynomial inverse [151].

This code space $\mathcal{C}$ is spanned by the columns of $\mathbf{H}$, or equivalently, by the first N columns of $\mathbf{U}(z)$ [87, 85, 86].

The analogy between $\mathbf{H}$ and the generator matrix of an error-correcting channel code becomes more clear through the Smith-McMillan form, and OFBs may then be considered as *real-valued error-correcting* codes [104, 25]. The orthogonal projection onto the code subspace $\Pi_{\mathcal{C}}$ defined in (B.21) is implemented by using the para-pseudo-inverse $\widehat{\mathbf{G}}(z)$ defined in (B.19), at the synthesis stage [17]. It is shown in [85, 87] how the parity-check matrix and the syndrome associated to $\mathbf{H}(z)$ may be computed using (B.27). This is the topic of next part.

B.2.2.2. Parity-Check Matrix and Syndrome Computation

Since the matrix $\mathbf{U}(z)$ is unimodular, it is invertible and its inverse may be written as follows

$$\mathbf{U}^{-1}(z) = \begin{pmatrix} \mathbf{V}(z) \\ \mathbf{P}(z) \end{pmatrix}, \tag{B.28}$$

where $\mathbf{V}(z)$ and $\mathbf{P}(z)$ are matrices of sizes $N \times M$ and $(M-N) \times N$, respectively. From (B.27)–(B.28), one gets

$$\begin{pmatrix} \mathbf{V}(z) \\ \mathbf{P}(z) \end{pmatrix} \mathbf{H}(z) = \begin{pmatrix} \mathbf{\Lambda}(z)\mathbf{W}(z) \\ \mathbf{0} \end{pmatrix}, \tag{B.29}$$

and thus

$$\mathbf{P}(z)\mathbf{H}(z) = \mathbf{0}. \tag{B.30}$$

Consider a subband signal $\mathbf{Y}(z)$ obtained at the output of the OFB from some input signal $\mathbf{X}(z)$. Then, using (B.30), one obtains

$$\begin{aligned} \mathbf{P}(z)\mathbf{Y}(z) &= \mathbf{P}(z)\mathbf{H}(z)\mathbf{X}(z) \\ &= \mathbf{0}. \end{aligned} \tag{B.31}$$

Now assume that $\mathbf{Y}(z)$ is corrupted by some error signal of polyphase representation $\mathbf{E}(z)$. The corrupted subband signal $\widetilde{\mathbf{Y}}(z) = \mathbf{Y}(z) + \mathbf{E}(z)$ verifies

$$\mathbf{P}(z)\widetilde{\mathbf{Y}}(z) = \mathbf{P}(z)\mathbf{E}(z). \tag{B.32}$$

Therefore $\mathbf{P}(z)$ is considered as the parity-check matrix associated to $\mathbf{H}(z)$ and

$$\mathbf{S}(z) = \mathbf{P}(z)\widetilde{\mathbf{Y}}(z)$$

is the syndrome associated to $\widetilde{\mathbf{Y}}(z)$, which has to be equal to zero in the error-free case, and may be different from zero if some transmission errors are encountered[3].

[3] As in error-correcting codes, some transmission errors lead to a null syndrome and are thus undetectable.

The parity-check matrix $\mathbf{P}(z)$ may be written as follows

$$\mathbf{P}(z) = \sum_{k=-\infty}^{+\infty} \mathbf{P}_k z^{-k}, \tag{B.33}$$

where $\mathbf{P}_k$, $k \in \mathbb{Z}$ are $(M-N) \times M$ matrices.

In the case of FIR analysis filters, the parity-check filter $\mathbf{P}(z)$ is also FIR [87] and (B.33) becomes

$$\mathbf{P}(z) = \sum_{k=0}^{K'-1} \mathbf{P}_k z^{-k}, \tag{B.34}$$

where $K' \in \mathbb{N}^*$.

C. Image Denoising by Adaptive Lifting Schemes

This appendix addresses the problem of image denoising by using adaptive lifting schemes. Such methods can adapt themselves to the analyzed signal, allowing to keep useful information for denoising applications. However, these schemes result in non-isometric transforms, impacting thus the denoising approaches performing the noise energy estimation in the subband domain. In [118, 119], the authors proposed a model to evaluate the subband energies of an uncorrelated signal when using such adaptive lifting schemes. Based on this work, an estimation of the noise energies in the wavelet transform domain is proposed in the sequel for image denoising.

C.1. Introduction

During its acquisition or transmission, an image is often corrupted by noise. The aim of denoising techniques is to remove this noise while keeping as much as possible the important features of the image. A particular interest has been dedicated to denoising by wavelet thresholding [35, 39, 36]. The principal motivation relies on the energy compaction property inherent to wavelet transforms: the small detail coefficients are more likely due to noise whereas the large ones are due to important signal features [72]. These small coefficients can be then thresholded without affecting the significant features of the image.

The lifting scheme (LS), introduced by Sweldenes [145] is a practical way for constructing wavelets, which leads to the so called second generation wavelet transforms. It is popular because it has the capability of adjusting the wavelet transform to complex geometries and offers a simple, yet efficient, implementation of the first generation wavelet transforms. However, an important limitation of the LS is that it does not cope well with the sudden changes in the input signal, that hide important information in many applications, such as denoising. It becomes desirable to have a lifting scheme that is able to adapt itself to the data. The adaptive lifting schemes (ALS) have been designed for this particular purpose [124, 29, 68, 125, 106].

The idea behind the ALS application in the particular case of denoising via thresholding, is that these schemes allow to perfectly preserve the original characteristics

of the input signal, offering a sparse representation, which make the thresholding rules more effective than in the case of the traditional non-adaptive LS. The most known thresholding methods are VisuShrink [35] and SureShrink [36].

In this work, we have considered a soft thresholding method. The threshold selection algorithm proposed by Donoho in [35] for VisuShrink is used to derive a specific threshold for each subband when using an ALS. This approach relies on the estimation of the noise energy in each subband when an ALS is used, which is not a trivial task. Usevitch [150] has shown, for generic linear wavelet filter banks, that for an uncorrelated signal, the energy in the spatial domain is the weighted sum of subband energies. This allows, for example, to estimate the distortion introduced by the quantization noise, but also it can be used to analyze other kinds of noise. This result has been generalized to non-linear ALS in [118, 119, 23] and the obtained subband weights have been used to perform optimal resource allocation.

The contribution of the work presented in this annex is to use these weights in order to derive estimates of the noise energy in each subband, prior to the application of the soft thresholding procedure.

C.2. Adaptive Lifting Schemes

C.2.1. Classical Lifting Schemes

A typical lifting stage is composed of three steps : Split, Predict and Update as shown in Figure C.1.

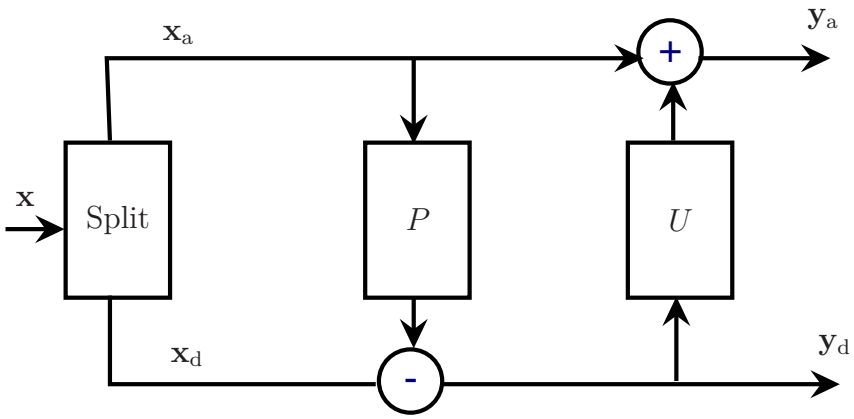


Figure C.1.: Analysis stage of a classical lifting scheme.

The input signal $\mathbf{x}$ is first split into its even and odd polyphase components, called the approximation signal $\mathbf{x}_a$ and respectively, the detail signal $\mathbf{x}_d$. The odd samples of $\mathbf{x}$ are then predicted from the neighboring even ones. The predictor operator P is a linear combination of these samples and generally, it is chosen such that it gives

a good estimate of $\mathbf{x}_\mathrm{d}$. The obtained signal $\mathbf{y}_\mathrm{d} = \mathbf{x}_\mathrm{d} - P(\mathbf{x}_\mathrm{a})$ has smaller energy than $\mathbf{x}_\mathrm{d}$. Finally, the even samples of $\mathbf{x}$ are transformed into a low-pass filtered and subsampled version $\mathbf{y}_\mathrm{a}$ of the original signal $\mathbf{x}$. This is performed by using an updating operator U which is a linear combination of the elements of $\mathbf{y}_\mathrm{d}$. The approximation signal $\mathbf{y}_\mathrm{a} = \mathbf{x}_\mathrm{a} + U(\mathbf{y}_\mathrm{d})$ is then obtained.

The principal disadvantage of the LS described above is that the linear filtering structure is fixed and therefore it can not match well the sharp transitions in the signal. The lifting schemes with adaptive prediction (APLS) [50, 29, 106] or adaptive update (AULS) [124, 68, 125] have been designed to overcome this limitation by using a filter that is able to adapt itself to the input signal. In what follows, the APLS are briefly described.

C.3. Adaptive Prediction Lifting Scheme

Let $\mathbf{x}$ be the input signal of length N and $\mathbf{y}_{i,j}$ the wavelet subband obtained from $\mathbf{x}$, where $i \in \mathcal{I}$ identifies the decomposition level starting from 0, and $j \in \mathcal{J}$ identifies the channel. Usually $\mathcal{J} = \{0, 1\}$, with 0 used for the low-pass and 1 for the high-pass channel, but more channels can be used, as in the case of multi-dimensional transforms. For example, the subbands produced by one decomposition level are called $\mathbf{y}_{0,0}$ and $\mathbf{y}_{0,1}$.

In the APLS, the adaptivity is built into the prediction step of the lifting scheme as shown in Figure C.2.

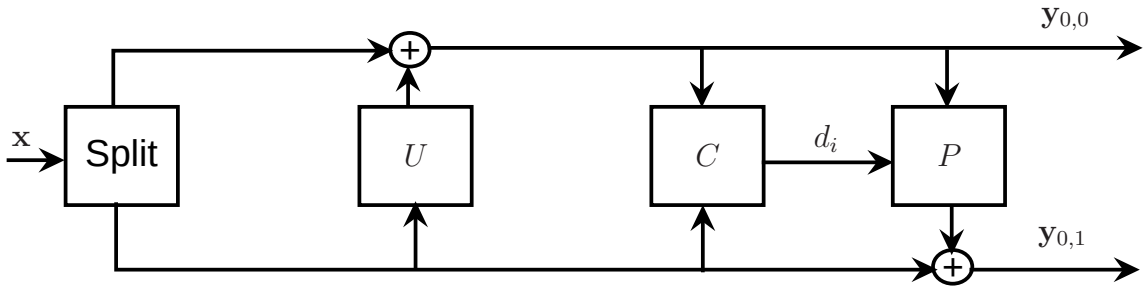


Figure C.2.: Adaptive prediction lifting scheme.

Such schemes are based on the design of a data-dependent prediction filter in order to minimize the predicted detail signal. In this work, the considered APLS is the one proposed by Claypoole *et al.* in [29], which lowers the order of the prediction filter near jumps to avoid prediction across discontinuities, and uses higher order predictors where the signal is locally smooth.

The choice of the prediction operator to be used is made at each decomposition level i. A decision map $d_i(\cdot)$ is computed at each decomposition level i. The decision map allows to discriminate the smooth parts of the signal to be filtered from its

textured parts. Usually, the decision map takes its value in a discrete index set $\{1, \ldots, N_{\mathrm{f}}\}$, each value h of this set being associated to a given prediction filter $\{\gamma_h(n),\, n \in \mathbb{Z}\}$. The position k, *e.g.* $d_i(k) = h$, $h = 1, \ldots, N_{\mathrm{f}}$, indicates that the $h-$th prediction filter γ_h has to be used. Once the decision maps are calculated, the following equations are used for the analysis stage at level i

$$\begin{aligned}
\mathbf{y}_{i,0}(k) &= \mathbf{y}_{i-1,0}(2k) + \sum_{n\in\mathbb{Z}} \beta(n)\mathbf{y}_{i-1,0}(2k+1-2n) \\
\mathbf{y}_{i,1}(k) &= \mathbf{y}_{i-1,0}(2k+1) - \sum_{n\in\mathbb{Z}} \gamma_{d_i(k)}(n)\mathbf{y}_{i,0}(k)(k-n),
\end{aligned}$$

where $\mathbf{y}_{i-1,0} = \mathbf{x}$, if $i = 0$, and $\{\beta(n),\, n \in \mathbb{Z}\}$ represents the non-adaptive update filter. The synthesis stage at level i is described by

$$\begin{aligned}
\widetilde{\mathbf{y}}_{i-1,0}(2k+1) = &\qquad \widetilde{\mathbf{y}}_{i,1}(k) + \sum_{n\in\mathbb{Z}} \gamma_{d(k)}(n)\widetilde{\mathbf{y}}_{i,0}(k-n) \\
\widetilde{\mathbf{y}}_{i-1,0}(2k) = &\qquad \widetilde{\mathbf{y}}_{i,0}(k) - \sum_{n\in\mathbb{Z}} \beta(n)\widetilde{\mathbf{y}}_{i-1,0}(2k+1-2n)
\end{aligned}$$

where $\widetilde{\mathbf{y}}_{i-1,0}$, if $i = 0$, is the reconstructed input signal $\widetilde{\mathbf{x}}$.

As one can notice, this system is non-linear since the prediction operator depends on the decision map which, at its turn, depends on the input signal.

C.4. Distortion Estimation in the Transform Domain

For generic linear wavelet filter banks, Usevitch showed [150] that the energy σ^2 (in the spatial domain) of an uncorrelated one-dimensional signal is related to the energies $\sigma^2_{i,j}$ of the wavelet subbands $\mathbf{y}_{i,j}$ by the following linear relation

$$\sigma^2 = \sum_{i,j} \frac{1}{2^{i+1}} w_{i,j}\sigma^2_{i,j}. \tag{C.1}$$

The weights $w_{i,j}$ are computed as the norm of the synthesis polyphase matrix $\mathbf{G}_{i,j}$ used to reconstruct the subband $\mathbf{y}_{i,j}$ of size $N_{i,j}$ [150]

$$w_{i,j} = \frac{2}{N} \sum_{n,m} \mathbf{G}_{i,j}(m,n)^2. \tag{C.2}$$

This approach has been extended in [118, 119] to the case of the inherently non-linear ALS, for which no polyphase representation exists. The basic idea was to look at the overall ALS as a linear time-varying system, which is possible once the decision maps $d_i(\cdot)$ are given. In fact, the authors have shown that the non-linearity of the system depends only on the decision maps and not on the whole input signal. Therefore, the weights depend only on the values of $d_i(\cdot)$, and more precisely on the choice of the prediction filters.

Consider the polyphase synthesis matrix $\mathbf{G}_{i,j}^{(h)}$ corresponding to the value h of the decision map $d_i(\cdot)$. This matrix can be considered as the polyphase synthesis matrix used in the non-adaptive lifting case, where the $h-$th prediction filter γ_h is always used. Let $N_{i,j}^{(h)}$ be the number of times the $h-$th filter is used in the subband $\mathbf{y}_{i,j}$. In [118, 119] is shown that

$$w_{i,j} = \sum_{h=1}^{N_\mathrm{f}} \frac{2N_{i,j}^{(h)}}{N} w_{i,j}^{(h)} = \sum_{h=1}^{N_\mathrm{f}} \rho^{(h)} w_{i,j}^{(h)}, \tag{C.3}$$

where $w_{i,j}^{(h)}$ is obtained from (C.2) and $\rho^{(h)} = \frac{2N_{i,j}^{(h)}}{N}$ measures the relative frequency of using filter h for the current subband $y_{i,j}$.

C.5. Application to Image Denoising

In the APLS approach, the prediction operator adapts itself to the input signal so that the characteristics of the original signal are very well preserved. This property has been successfully exploited to perform optimal resources allocation [118, 119], by estimating the distortion, introduced by quantization, in the transform domain.

In this work, the APLS properties are exploited for the purpose of image denoising. Let the image input signal be $\mathbf{x}(k,\ell)$, k, $\ell = 1,\ldots,N$. Assume that this signal is corrupted by an additive noise ϵ of energy σ^2 assumed independent of $\mathbf{x}$. The corrupted image $\mathbf{z}$ is then obtained as

$$\mathbf{z}(k,\ell) = \mathbf{x}(k,\ell) + \epsilon(k,\ell). \tag{C.4}$$

The goal is to denoise $\mathbf{z}(k,\ell)$ and to obtain an estimate $\widehat{\mathbf{x}}(k,\ell)$ of $\mathbf{x}(k,\ell)$.

Let I denotes the coarsest scale of the APLS decomposition. As introduced in Section C.3, the same notation $\mathbf{y}_{i,j}$ is kept for the subbands obtained from $\mathbf{x}$, where $i \in \mathcal{I}$, $j \in \mathcal{J}$, $\mathcal{I} = \{0,\ldots,I-1\}$ and $\mathcal{J} = \{0,\ldots,3\}$, the only difference being that the two-dimensional case is considered here. The noisy subbands obtained from $\mathbf{z}$ are denoted by $\widetilde{\mathbf{y}}_{i,j}$. In the case of an orthogonal wavelet transform, the coefficients in each noisy subband $\widetilde{\mathbf{y}}_{i,j}$ are i.i.d, as $\mathcal{N}(0,\sigma^2)$. In the case of an APLS, this result does not hold anymore since the considered transform is neither isometric nor linear. The standard deviation $\sigma_{i,j}$ of the noise within the subband $\widetilde{\mathbf{y}}_{i,j}$ is not equal to the standard deviation σ of the noise introduced the spatial domain.

In what follows, wavelet thresholding is used for removing the noise. It consists in thresholding only the wavelet coefficients of the detail subbands (*i.e.*, $\widetilde{\mathbf{y}}_{i,1}$, $\widetilde{\mathbf{y}}_{i,2}$ and $\widetilde{\mathbf{y}}_{i,3}$), while keeping the low resolution coefficients unchanged. The soft thresholding method in [37] is considered here as it gives the best performances when coupled with an undecimated discrete wavelet transform [30]. For an one-dimensional signal of length M, Donoho and Johnstone [35] proposed the universal threshold, $T_\mathrm{U} = \sigma\sqrt{2\log(M)}$, which results in an asymptotic optimal estimate in the minimax sense.

In the case of APLS, the soft thresholding is applied to each noisy detail subband $\widetilde{\mathbf{y}}_{i,j}$, $i \in \mathcal{I}$, $j = 1, 2, 3$, with a specific threshold $T_{i,j} = \sigma_{i,j}\sqrt{2\log(N_i^2)}$, where $N_i^2 = \frac{N^2}{4^{i+1}}$ is the number of coefficients in the subband $\widetilde{\mathbf{y}}_{i,j}$ and $\sigma_{i,j}$ is the noise standard deviation of the corrupted detail subband $\widetilde{\mathbf{y}}_{i,j}$. The problem is thus to obtain a good estimation $\widehat{\sigma}_{i,j}$ of $\sigma_{i,j}$ within each subband. The relation in C.1 is only valid when $\sigma_{i,j}$ represents the energy of an uncorrelated signal, which might not be the case when using APLS on a noisy signal.

Consider the equation C.1 for the two-dimensional APLS, wherein the energy of the noise introduced in the spatial domain is $\sigma^2 = \sum_{i,j} \frac{1}{4^{i+1}} w_{i,j}\sigma_{i,j}^2$. As explained in [150], the use of a non-orthogonal transform results in weighting the energy in each subband. The weights can be seen as a measure of the closeness of the biorthogonal filters to the class of orthogonal filters. The introduction of these weights allows thus to approach the behavior of the orthogonal transform in the sense that the equality between the energies in the subbands, which is verified by an orthogonal transform, is changed into an equality between the weighted energies when a non-orthogonal transform is used. This relation may be written as

$$w_{i,j}\sigma_{i,j}^2 \approx w_{i',j'}\sigma_{i',j'}^2, \tag{C.5}$$

where $i \neq i'$ and $j \neq j'$.

At the first decomposition level, the noise energy of the diagonal details subband $y_{0,3}$ may be estimated [35, 39] as: $\widehat{\sigma}_{0,3}^2 = \left(\frac{m}{0.6745}\right)^2$, where m is the median absolute deviation of the wavelet diagonal details at the finest decomposition level. From (C.5), an estimation of $\sigma_{i,j}$ can be derived

$$\widehat{\sigma}_{i,j} = \sqrt{\left(\frac{w_{0,3}}{w_{i,j}}\right)}\widehat{\sigma}_{0,3}. \tag{C.6}$$

One should point out that the equation (C.5), from which (C.6) is derived, relies on the assumption of equality between the weighted subband energies. In the next section, the energies expressions in (C.6) are used to achieve soft thresholding.

C.6. Simulation Results

C.6.1. Noise Standard Deviation Estimation

In this subsection, the aim is to evaluate the correctness of the noise standard deviation estimation provided by equation (C.6). A white Gaussian noise with a standard deviation σ is introduced in the original image, which is further transformed using an APLS with five decomposition levels, as described in Section C.3. The equation (C.6) is then used to calculate the estimations $\widehat{\sigma}_{i,j}$ of the noise standard deviation

	Barbara			Lena		
σ	30	50	80	30	50	80
Subband $\mathbf{y}_{0,3}$	0.86	0.51	0.12	0.51	0.20	0.03
Subband $\mathbf{y}_{0,2}$	0.71	0.48	0.30	0.54	0.22	0.16
Subband $\mathbf{y}_{0,1}$	0.98	0.55	0.32	0.58	0.32	0.17
Subband $\mathbf{y}_{1,3}$	8.16	8.71	8.77	8.84	9.02	9.17
Subband $\mathbf{y}_{1,2}$	7.25	7.57	7.52	7.61	7.82	7.77
Subband $\mathbf{y}_{1,1}$	5.09	5.56	5.71	5.64	5.68	5.84

Table C.1.: Relative error of the standard deviation estimation in some of the detail subbands for the images `Barbara.pgm` and `Lena.pgm`, corrupted with a spatial noise of standard deviation $\sigma \in \{30, 50, 80\}$.

within each subband $\tilde{\mathbf{y}}_{i,j}$. The percentage of relative errors of this estimation are reported in Table C.1 for the six first subbands.

For the first decomposition level the estimation is accurate, but the estimation errors become more important in the second level of decomposition and so on. The reason is that the signal is not entirely decorrelated as supposed in [150] and imposed in [119]. However, these estimations are good enough in the particular context of denoising by soft thresholding as will be shown in the next subsection.

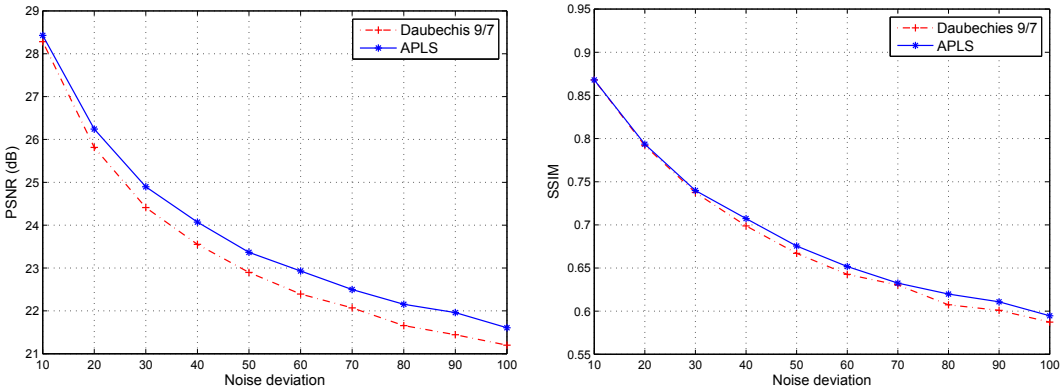
C.6.2. Denoising by Soft Thresholding

In this subsection, a soft thresholding method is used on the detail subbands of the noisy image, obtained with the considered APLS. A specific threshold is used for each detail subband as described in Section C.5. This denoising approach is compared to the conventional soft thresholding approach performed on the same noisy image, but this time transformed with the classical, non-adaptive 9/7 wavelet transform. Three different images have been considered: `Lena.pgm`, `House.pgm` and `Barbara.pgm`. PSNR and SSIM curves as function of the introduced noise are reported in Figures C.3, C.4 and C.5.

There is an improvement in PSNR and SSIM with the proposed approach when compared to the 9/7 filter. When considering the image `House.pgm`, the obtained gain is up to 1 dB in PSNR and 0.03 in SSIM.

Figure C.6 compares the visual reconstruction quality for the image `Lena.pgm` for both denoising methods, *e.g.* when the image is transformed with the non-adaptive 9/7 filter and further denoised using the soft thresholding method, and when the APLS scheme is used and the proposed denoising approach, described in Section C.5, is adopted.

The noise introduced in the image has a standard variation of 30. One can see that the APLS scheme allows to better preserve the edges. For example, the nose, mouth



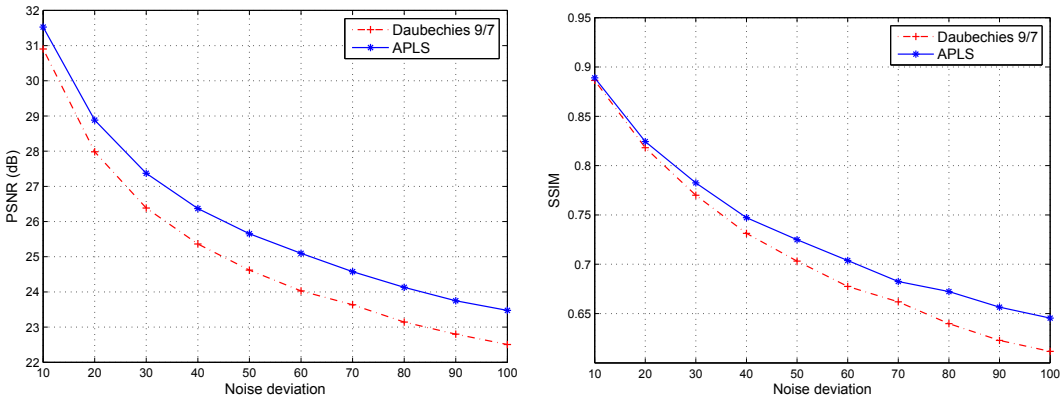
(a) PSNR performance (b) SSIM performance

Figure C.3.: PSNR and SSIM performance as function of the spatial noise introduced in the image `Lena.pgm`.

and eyes areas in Figure C.6 are recovered with more details. The non-adaptive 9/7 wavelet transform tends in general to oversmooth the image whereas the APLS scheme catches better the details. Note however that there are some artifacts in the image denoised using the proposed scheme. The reason is that the soft threshloding method introduces a bias in the estimation of the wavelet coefficients of important amplitude. Other techniques that lead to an unbiased estimation, as the SURE method, may be considered and should give a higher visual quality. Also, the energy estimation within each subband is performed using (C.6), based on the assumption in (C.5).

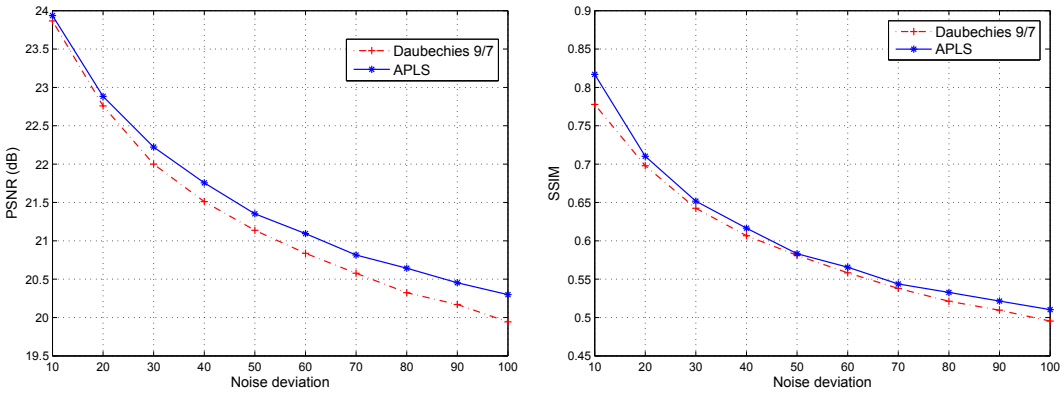
C.7. Conclusion

In this appendix, we proposed an approach for image denoising via soft thresholding, by using an adaptive lifting scheme. This approach, based on the estimation of the energy in the transform subbands, gives better performance than the classical non-adaptive 9/7 wavelet transform. Other thresholding methods, for example the SURE method, may be considered to improve the visual quality of the reconstructed images. Moreover, the SureShrink technique, which is an hybrid of the universal thresholding method, and the SURE threshold may be investigated as well, the choice of the threshold in this case depends on the energy of a particular subband. In addition, more elaborated expressions than (C.6) may be developed for estimating the subbands energies in presence of noise.



(a) PSNR performance (b) SSIM performance

Figure C.4.: PSNR and SSIM performance as function of the spatial noise introduced in the image `House.pgm`.



(a) PSNR performance (b) SSIM performance

Figure C.5.: PSNR and SSIM performance as function of the spatial noise introduced in the image `Barbara.pgm`.

(a) 9/7 wavelet filter,
PSNR= 24.4 dB, SSIM= 0.740

(b) APLS scheme,
PSNR= 24.9 dB, SSIM= 0.742

Figure C.6.: Visual reconstruction quality of the denoised image `Lena.pgm` when using the 9/7 wavelet filter and the APLS scheme, for $\sigma = 30$.

List of Acronyms

ARQ Automatic Repeat Request

AVC Advanced Video Coding

AWGN Additive White Gaussian Noise

BER Bit Error Rate

BP Belief Propagation

BPSK Binary Phase-Shift Keying

CRC Cyclic Redundancy Check

DCT Discrete Cosine Transform

FB Filter Bank

FEC Forward Error Correction

FIR Finite Impulse Response

HD Hard Decisions

JSC Joint Source-Channel

JSC-NoSI JSC with No Side Information

JSC-SI JSC with Side Information

JSCC Joint Source-Channel Coding

JSCD Joint Source-Channel Decoding

LDPC Low-Density Parity-Check

LS Least Squares

MAC Media Access Control

MAP Maximum a Posteriori

MCTF Motion-Compensated Temporal Filtering

MD Multiple Description

MDCT Multiple Descriptions with Correlating Transform

MDSQ Multiple Description Scalar Quantization

ML Maximum Likelihood

MSE Mean Squared Error

NALU Network Abstraction Layer Unit

NR-D Non Robust Decoder

OFB Oversampled Filter Banks

PCT Parity-Check Test

PDF Probability Density Function

PEB Percentage of Erroneous Blocks

PSNR Peak Signal to Noise Ratio

R-D Rate-Distorsion

RTP Real Time Protocol

SD Single Description

SNR Signal to Noise Ratio

UEP Unequal Error Protection

VLC Variable Length Code

Y-PSNR PSNR on the luminance component Y

List of Publications

1. M. Abid, M. Trocan, M. Kieffer and B. Pesquet-Popescu, "Joint source-channel decoding of multiple-description encoded video transmitted over heteregeneous networks", in *Signal Processing: Image Communication*, 2012, submitted.
2. M. Abid, M. Kieffer and B. Pesquet-Popescu, "Maximum a posteriori consistent estimation using interval analysis", in *the 16th IFAC Symposium on System Identification (SYSID)*, Belgium, July 11-13, 2012.
3. Q. Wang, M. Abid, M. Kieffer and B. Pesquet-Popescu, "MAP estimation of the input of an oversampled filter bank from noisy subbands by belief propagation", in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, March 25-30, 2012.
4. M. Abid, M. Kieffer and B. Pesquet-Popescu, "Reconstruction cohérente de l'entrée d'un banc de filtres suréchantillonnés à partir de sa sortie bruitée", in *Proceedings of GRETSI*, Bordeaux, September 5-8, 2011.
5. M. Abid, M. Kieffer and B. Pesquet-Popescu, "Consistent reconstruction of the input of an oversampled filter bank from noisy subbands", in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Barcelona, August 29 - September 2, 2011.
6. M. Abid, M. Kieffer and B. Pesquet-Popescu, "Joint source-channel coding/decoding of 3D-ESCOT bitstreams, in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Saint-Malo, October 4-6, 2010.
7. M. Abid, M. Kieffer, M. Cagnazzo and B. Pesquet-Popescu, "Robust decoding of a 3D-ESCOT bitstream transmitted over noisy channels", in Proceedings of the IEEE International Conference on Image Processing, (ICIP), Hong Kong, September 26-29, 2010.
8. M. Abid, M. Cagnazzo and B. Pesquet-Popescu, "Image denoising by adaptive lifting schemes", in *Proceedings of the European Workshop on Visual Information Processing (EUVIP)*", Paris, July 5-7, 2010.

Bibliography

[1] M. A. Agostini, M. Kieffer, and M. Antonini. MAP estimation of multiple description encoded video transmitted over noisy channels. In *Proceedings of IEEE International Conference on Image Processing, ICIP*, Cairo, Egypt, November 7-10 2009. 56, 91

[2] M. Akbari and F. Labeau. Recovering the output of an OFB in the case of instantaneous erasures in sub-band domain. In *Proc. MMSP*, St Malo, France, 2010, pp. 274-279. 6, 149, 153

[3] J. B. Anderson and S. Mohan. *Source and Channel Coding: An Algorithmic Approach*. Kluwer, 1991. 14, 58, 71, 73

[4] E. Baccaglini, T. Tillo, and G. Olmo. Image and video transmission: a comparison study of using unequal loss protection and multiple description coding. *Multimedia Tools and Applications*, 55(2):247–259, 2011. 19, 55

[5] S. Badri-Hoeher and P. Hoeher. Use of the list viterbi algorithm to compute the distance spectrum of trellis codes and isi channels. In *Information Theory, 2000. Proceedings. IEEE International Symposium on*, page 101, 25-30 June 2000. 57

[6] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, 20:284–287, 1974. 73

[7] R. Bauer and J. Hagenauer. On variable length codes for iterative source/channel decoding. In *Proc. IEEE Data Compression Conference*, pages 272–282, Snowbird, UT, 1998. 58

[8] Olivier Beaumont and Bernard Philippe. Linear interval tolerance problem and linear programming techniques. *Reliable Computing*, 7(6):433–447, 2001. 115

[9] N. El Beheiry, M. El Sharkawy, M. Lotfy, and S. Elnoubi. An adaptive fast and efficient spatial error concealment technique for block-based video coding systems. *IEEE International Midwest Symposium on Circuits and Systems*, pages pp. 663–668, August 2009. 59

[10] A. Ben-Israel and T. N. E. Greville. *Generalized Inverses: Theory and Applications*. Wiley-Interscience, 1974. 153

[11] C. Bergeron and C. Lamy-Bergot. Soft-input decoding of variable-length codes applied to the H.264 standard. In *Proc. IEEE Workshop on Multimedia Signal Processing*, pages 87–90, 29 Sept.-1 Oct. 2004. 6, 57

[12] C. Bergeron and C. Lamy-Bergot. Modelling h.264/avc sensitivity for error protection in wireless transmissions. In *Multimedia Signal Processing, 2006 IEEE 8th Workshop on*, pages 302–305, Oct. 2006. 6, 46

[13] R. Bernardini, M. Durigon, R. Rinaldo, and A. Vitali. Comparison between multiple description and single description video coding with forward error correction. In *Proceedings of IEEE International Workshop on Multimedia Signal Processing, MMSP*, Shangai, China, October 30-November 2 2005. 19, 55, 56

[14] C. Berrou, A. Glavieux, and P. Thitimajhima. Near shannon limit error correcting coding and decoding : Turbo-codes. *Proc. Of ICC, Geneva*, pages 1064–1070, 1993. 145

[15] R. E. Blahut. *Algebraic Codes for Data Transmission*. Cambridge University Press, February 2003. 103

[16] H. Bölcskei and F. Hlawatsch. Oversampled filterbanks: Optimal noise shaping, design freedom and noise analysis. In *Proc. IEEE Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 2453–2456, Munich, Germany, 1997. 149

[17] H. Bölcskei and F. Hlawatsch. Noise reduction in oversampled filter banks using predictive quantization. *IEEE Transactions on Information Theory*, 47(1):155–172, January 2001. 149, 157

[18] H. Bölcskei, F. Hlawatsch, and H. G. Feichtinger. Frame-theoretic analysis of oversampled filter banks. *IEEE Trans. Signal Processing*, 46(12):3256–3268, 1998. 153

[19] J.W. Byers, M. Luby, and M. Mitzenmacher. A digital fountain approach to asynchronous reliable multicast. *IEEE Journal on Selected Areas in Communications*, 20(8):1528–1540, 2002. 41

[20] M. Bystrom and J. W. Modestino. Combined source-channel coding schemes for video transmission over an additive white gaussian noise channel. *IEEE Journal on Selected Areas in Communications*, 18:880–890, 2000. 6, 42, 45, 47

[21] M. Bystrom and T. Stockhammer. Dependent source and channel rate allocation for video transmission. *IEEE Transactions on Wireless Communications*, 3(1):258– 268, 2004. 42, 45, 47

[22] T. André C. Lamy-Bergot, A. Mokraoui-Zergainoh and B. Pesquet-Popescu. Panorama des techniques de codage/décodage conjoint et techniques de diversité adaptées à la transmission de flux vidéo et html sur lien ip sans fil point/multipoint. *Revue Traitement du Signal*, 25, no.5 (in French), Octobre 2008. 6, 56, 57

[23] M. Cagnazzo and B. Pesquet-Popescu. Perceptual impact of transform coefficients quantization for adaptive lifting schemes. In *International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2010. 160

[24] G. Cheung and A. Zakhor. "bit allocation for joint source/channel coding of scalable video",. *IEEE Transactions on Image Processing*, 9:340–356, 2000. 6, 42, 45, 47

[25] J.C. Chiang, M. Kieffer, and P. Duhamel. Robust image transmission using oversampled filterbank. In *Proceedings of EUSIPCO*, pages 1325–1328, 2004. 157

[26] M. Chiani and M.G. Martini. Proportional unequal error protection for mpeg–4 video transmission. In *Proceedings (ICC 2001), Helsinki (Finland)*, 2001. 6, 46

[27] J. Chou and K. Ramchandran. Arithmetic coding-based continuous error detection for efficient arq-based image transmission. *Selected Areas in Communications, IEEE Journal on*, 18(6):861–867, June 2000. 58

[28] C. Christopoulos, A. Skodras, and T. Ebrahimi. The JPEG-2000 still image coding system: An overview. *IEEE Trans. Consumer Electron.*, 46:1103–1127, 2000. 45

[29] R. L. Claypoole, G. M. Davis, W. Sweldens, and R. G. Baraniuk. Nonlinear wavelet transforms for image coding via lifting. *IEEE Transactions on Image Processing,*, 12, no. 12(12):1449–1459, December 2003. 159, 161

[30] R. R. Coifman and D. L. Donoho. Translation invariant de-noising. In *Wavelets and Statistics, A. Antoniadis and G. Oppenheim, Eds. New York: Springer-Verlag,*, 1995, pp. 125–150. 163

[31] Z. Cvetković and M. Vetterli. Oversampled filter banks. *IEEE Transactions on Signal Processing*, 46(5):1245–1255, 1998. 153

[32] S. Sarvotham D. Baron and R. G. Baraniuk. Bayesian compressive sensing via belief propagation. *IEEE Transactions on Signal Processing*, 58(1):269–280, January 2010. 120

[33] L. B. Milstein D. N. Rowitch. Rate Compatible Punctured Turbo (RCPT) Codes in a hybrid FEC/ARQ system. *Proceedings of the IEEE Global Communication Conference*, 4:55–59, Phoenix, AZ, USA, 1997. 46

[34] B. Li W. Zhu Y. Q. Zhang D. Wu, Y. T. Hou and H. J. Chao. An end-to-end approach for optimal mode selection in internet video communication: theory and application. *IEEE Journal on Selected Areas in Communications*, 18, no. 6:977–995, 2000. 45

[35] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation via wavelet shrinkage. In *Biometrika*, 1994, vol. 81, pp. 425-455. 159, 160, 163, 164

[36] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. In *Journal of the American Statistical Association*, volume 90, pages 1200–1224, 1995. 159, 160

[37] D.L. Donoho. De-noising by soft-thresholding. In *IEEE Transactions on Information Theory*, volume 41, 1995. 163

[38] D.L. Donoho, A.Maleki, and A. Montanari. Message Passing Algorithm for Compressed Sensing. *Proceedings of the National Academy of Sciences*, 106 no. 45:18914–18919, 2009. 26, 120

[39] D.L. Donoho and I.M Johnstone. Threshold selection for wavelet shrinkage of noisy data. In *Proceedings of the IEEE Conference on Engineering in Medicine and Biology Society, Maryland, USA*, pages 24–25, 1994. 159, 164

[40] P. L. Dragotti, S. D. Servetto, and M. Vetterli. Optimal filter banks for multiple description coding: Analysis and synthesis. *IEEE Transactions on Information Theory*, 48(7):2036–2052, July 2002. 54

[41] R.J. Duffin and A. C. Schaeffer. A class of nonharmonic Fourier series. *Trans. Amer. Math. Soc.*, 72:341–366, 1952. 53

[42] P. Duhamel and M. Kieffer. *Joint source-channel decoding: A cross-layer perspective with applications in video broadcasting*. Academic Press, 2009. 6, 8, 20, 57, 63, 73, 108

[43] A. A. El Gamal and T. M. Cover. Achievable rate for multiple description. *IEEE Transactions on Information Theory*, 28(6):851–857, 1982. 6, 48

[44] A. K. Katsaggelos F. Zhai, Y. Eisenberg. *Handbook of Image and Video Processing*, chapter Joint source-channel coding for video communications. June 2005. 5, 43

[45] F. Farvardin. A study of vector quantization for noisy channels. *IEEE Transactions on Information Theory*, 36:799–809, 1990. 45

[46] N. Farvardin and V. Vaishampayan. Optimal quantizer design for noisy channels: An approach to combined source-channel coding. *IEEE Transactions on Information Theory*, 33(6):827–837, 1991. 45

[47] M. Fleming and M. Effros. Generalized multiple description vector quantization. In *Proceedings of the IEEE Data Compression Conference*, pages 3–12, 1999. 51

[48] J. E. Fowler and B. Pesquet-Popescu. An overview on wavelets in source coding, communications, and networks. *EURASIP Journal on Image and Video Processing*, pages 1–28, 2007. 55

[49] C. Zhang S. S. Gao G. F. Tu, J. J. Liu and S. D. Li. Studies and advances on joint source-channel encoding/decoding techniques in flow media communications. *Science China Information Sciences*, 53, no. 1:1–17, 2010. 5, 43

[50] Ö. N. Gerek and A. E. Çetin. Adaptive polyphase subband decomposition structures for image compression. *IEEE Transactions on Image Processing*, 9(10):1649–1659, October 2000. 161

[51] H. Gharavi and S. I. Gao. Spatial interpolation algorithm for error concealment april 2008, pp. 1153-1156. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 1153–1156, April 2008. 59

[52] V. K. Goyal. Multiple description coding: compression meets the network. *IEEE Signal Processing Magazine*, 18(5):74–93, September 2001. 6, 42, 48

[53] V. K. Goyal and J. Kovacević. Generalized multiple description coding with correlating transforms. *IEEE Trans. Information Theory*, 47(6):2199–2224, 2001. 49, 52

[54] V. K Goyal, J. Kovacevic, and J.A. Kelner. Quantized frame expansions with erasures. *Journal of Appl. and Comput. Harmonic Analysis*, 10(3):203–233, 2001. 6, 54, 149

[55] V. K. Goyal, J. Kovacević, and M. Vetterli. Multiple description transform coding : Robustness to erasures using tight frame expansion. In *Proc. IEEE Int. Symp. On Inform. Th.*, page 408, Boston, MA, August 1998. 53

[56] V. K. Goyal, Jelena Kovacevic, and Martin Vetterli. Quantized frame expansions as source-channel codes for erasure channels. In *Proceedings of Data Compression Conference*, pages 326–335, 1999. 6, 53, 54, 149

[57] V. K. Goyal, M. Vetterli, and N. T. Thao. Quantized overcomplete expansions in $\mathbb{R}^n$: Analysis, synthesis, and algorithms. *IEEE Transactions on Information Theory*, 44(1):16–31, 1998. 54

[58] M. Grangetto and P. Cosman. Map decoding of arithmetic codes with a forbidden symbol. *Proc. of ACIVS 2002, Ghent, Belgium*, 2002. 58

[59] M. Grangetto, P. Cosman, and G. Olmo. Joint source/channel coding and MAP decoding of arithmetic codes. *IEEE Trans. on Comm.*, 53(6):1007–1016, 2005. 58

[60] M. Grangetto, E. Magli, and G. Olmo. Robust video transmission over error-prone channels via error correcting arithmetic codes. *IEEE Comm. Letters*, pages 596–598, 2003. 58

[61] M. Grangetto, E. Magli, and G. Olmo. Joint source-channel iterative decoding of arithmetic codes. *Proc. of ICC, Paris*, pages 20–24, 2004. 58

[62] C. Guillemot and P. Siohan. Joint source-channel decoding with soft information: A survey. *Elsevier Journal on Applied Signal Processing, special issue on the turbo principle*, 6:906–927, 2005. 6, 56

[63] T. Guionnet and C. Guillemot. Soft decoding and synchronization of arithmetic codes: Application to image transmission over noisy channels. *IEEE Trans. on Image Processing*, 12(12):1599–1609, 2003. 58

[64] T. Guionnet, C. Guillemot, and E. Fabre. Soft decoding of multiple descriptions. In *Proceedings of IEEE International Conference on Multimedia and Expo, ICME*, volume 2, pages 601–604, August 26-29 2002. 56

[65] C. Yim H. Ha and Y. Y. Kim. Packet loss resilience using unequal forward error correction assignment for video transmission over communication networks. *Computer Communications*, 30, no. 12:3676–3689, 2007. 45

[66] O. Hadar, M. Huber, R. Huber, and S. Greenberg. New hybrid error concealment for digital compressed video. *EURASIP Journal on Advances in Signal Processing*, pages vol. 12, pp.1821–1833, July 2005. 59

[67] J. Hagenauer. Rate-compatible punctured convolutional codes (RCPC codes) and their applications. *IEEE Transactions on Communications*, 36(4):389–400, 1988. 41, 46

[68] Henk J. A. M. Heijmans, Béatrice Pesquet-Popescu, and Gemma Piella. Building nonredundant adaptive wavelets by update lifting. *Applied Computational Harmonic Analysis*, (18):252–281, May 2005, no. 18. 159, 161

[69] B. Hochwald and K. Zeger. Tradeoff between source and channel coding. *IEEE Transactions on Information Theory*, 43:1412–1424, 1997. 45

[70] ITU-T and ISO/IEC JTC 1. Advanced video coding for generic audiovisual services. Technical report, ITU-T Rec. H.264, and ISO/IEC 14496-10 AVC, nov. 2003. 39, 45

[71] S. Li J. Xu, Z.Xioong and Y. Zhang. *Three-dimensional embedded subband coding with optimal truncation (3D ESCOT)*. Applied and Computational Harmonic Analysis, May 2001, vol. 10, pp. 290-315. 10, 67

[72] Maarten Jansen. Noise reduction by wavelet thresholding. In *Springer Verlag, Lecture notes in Statistics, vol. 161,*, 2001. 159

[73] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag, London, 2001. 23, 111, 115, 116, 132

[74] H. Jenkac, G. Liebl, T. Stockhammer, and X. Wen. Flexible outer Reed-Solomon coding on RLC layer for MBMS over GERAN. In *IEEE Vehicular Technology Conference*, volume 5, pages 2777–2781, May 17-19 2004. 41

[75] H. Jenkac, T. Stockhammer, and W. Xu. Permeable-layer receiver for reliable multicast transmission in wireless systems. In *Proc. IEEE Wireless Communications and Networking Conference*, volume 3, pages 1805–1811, 13-17 March 2005. 63, 108

[76] S. Kaiser and M. Bystrom. Soft decoding of variable-length codes. In *Proc. IEEE ICC*, volume 3, pages 1203–1207, New Orleans, 2000. 6, 56

[77] U. Kamilov, V.K Goyal, and S. Rangan. Message-passing estimation from quantized samples. *arXiv:1105.6368v1 [cs.IT]*, 31 May 2011. 120

[78] M. Kim, H. Lee, and S. Sull. Spatial error concealment for H.264 using sequential directional interpolation. *IEEE Transactions on Consumer Electronics*, pages vol. 54, no. 4, pp. 1811–1818, November 2008. 59

[79] J. Kovacević, P. L. Dragotti, and V. K Goyal. Filter bank frame expansions with erasures. *IEEE Transactions on Information Theory*, 48(6):1439–1450, 2002. 6, 53, 54, 149, 153

[80] F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 47(2):pp. 498–519, February 2001. 26, 120

[81] F.R. Kschischang, B.J. Frey, and H.A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 2001. 123, 141, 142, 143, 144, 145

[82] W.-Y. KUNG, C.-S. KIM, and C.-C. J. KUO. Spatial and temporal error concealment techniques for video transmission over noisy channels. *IEEE transactions on circuits and systems for video technology*, 16:789–802, 2006. 59

[83] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, Boston, third edition, 2005. 62

[84] L. Meilhac S. Olivieri P. Verdi L. Camiciotti, C. Lamy. Joint Source-Channel Coding for 4G Multimedia Streaming. *2nd WWRF Meeting, WG3*, Helsinki, Finland, 2001. 42, 43

[85] F. Labeau. *Strategies for exploiting residual redundancy in joint source-channel coding*. PhD thesis, Université Catholique de Louvain, Louvain-la-Neuve, 2000. 29, 154, 156, 157

[86] F. Labeau, J.C. Chiang, M. Kieffer, P. Duhamel, L. Vandendorpe, and B. Mack. Oversampled filter banks as error correcting codes: theory and impulse correction. *IEEE Transactions on Signal Processing*, 53(12):4619 – 4630, 2005. 6, 29, 149, 154, 155, 157

[87] F. Labeau, L. Vandendorpe, and B. Macq. Oversampled filter banks as error correcting codes. In *International Symposium on Wireless Personal Multimedia Communications*, volume 3, pages 1265–1269, October 27-30 2002. 29, 155, 157, 158

[88] W. M. Lam, A. R. Reibman, and B. Liu. Recovery of lost or erroneously received motion vectors. In *Proceedings ICASSP*, volume 5, pages 417–420, 1993. 59

[89] M. H. Larsen, C. Weidmann, and M. Kieffer. Iterative decoding of entropy–constrained multiple description trellis–coded quantization. In *Proceedings of IEEE Global Telecommunications Conference, GLOBECOM*, San Francisco, United States, November 27-December 1 2006. 56

[90] L. A. Larzon, M. Degermark, L. E. Jonsson, and G. Fairhurst. The lightweight user datagram protocol (UDP-Lite). Technical Report RFC 3828, The Internet Society, 2004. 98

[91] C.M. Lee, M. Kieffer, and P. Duhamel. Soft decoding of VLC encoded data for robust transmission of packetized video. In *Proceedings of ICASSP*, pages 737–740, 2005. 6, 57

[92] S. Li and D. M. Healy, Jr. A parametric class of discrete Gabor expansions. *IEEE Transactions on Signal Processing*, 44(2):201–211, 1996. 153

[93] S. Lin and D. J. Costello. *Error Control Coding: Fundamentals and applications*. Prentice-Hall, Englewood Cliffs, 1983. 41

[94] S. Lin, D. J. Costello, Jr., and M. J. Miller. Automatic repeat request error-control schemes. *IEEE Communications Magazine*, 22:5–16, 1984. 41

[95] H.A. Loeliger. An Introduction to Factor Graph. *IEEE Signal Processing Magazine*, 04:28–41, 2004. 26, 120, 123, 141

[96] R. W. Heath Jr. M. F. Sabir and A. C. Bovik. Joint source-channel distortion modeling for mpeg-4 video. *IEEE Transactions on Image Processing*, 18, no 11:90–105, 2009. 42, 45

[97] D. J. C. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(3):399–431, 1999. 120, 145

[98] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic, Boston, MA, 1998. 53

[99] C. Marin, K. Bouchireb, M. Kieffer, and P. Duhamel. Joint exploitation of residual source information and mac layer crc redundancy for robust video decoding. *IEEE Transactions on Wireless Communications*, pages vol. 9, no. 7, pp. 2165–2175, July 2010. 58

[100] C. Marin, Y. Leprovost, M. Kieffer, and P. Duhamel. Robust header recovery based enhanced permeable protocol layer mechanism. In *Proceedings of IEEE Workshop in Signal Processing Advances in Wireless Communications, SPAWC*, pages 91–95, 2008. 8, 20, 56

[101] C. Marin, Y. Leprovost, M. Kieffer, and P. Duhamel. Robust MAC-Lite and soft header recovery for packetized multimedia transmission. *IEEE Transactions on Communications*, March 2010, vol. 58, no. 3, pp. 775-784. 8, 63, 108

[102] S. Marinkovic and C. Guillemot. Joint source-channel coding by means of oversampled filter banks codes. *Eurasip journal on applied signal processing*, (4):510–524, 2005. 54

[103] D. Marpe, H. Schwarz, G. Blattermann, and T Weigand. Final CABAC cleanup. JVT *of* ISO/IEC JTC1/SC29/WG11 & ITU-T SG16/Q.6 Doc. JVT-F039*, Awaji, Japan*, 2002. 39

[104] T. G. Marshall. Coding of real-number sequences for error correction: A digital signal processing problem. *IEEE Journal on Selected Areas in Communications*, 2(2):381–392, 1984. 54, 157

[105] R. J. McEliece, D. J. C. MacKay, and J. F. Cheng. Turbo decoding as an instance of Pearl' s 'Belief Propagation' Algorithm. *IEEE Journal on Selected Areas in Communications*, 16(2):140–152, February 1998. 120

[106] Nagita Mehrseresht and David Taubman. Spatially continuous orientation adaptive discrete packet wavelet decomposition for image compression. pages 1593–1596, Atlanta, GA (USA), October 2006. 159, 161

[107] T. Mladenov, S. Nooshabadi, and K. Kiseon. MBMS Raptor codes design trade-offs for IPTV. *IEEE Transactions on Consumer Electronics*, 56(3):1264–1269, August 2010. 41

[108] F. Mériaux and M. Kieffer. Robust IP and UDP-lite header recovery for packetized multimedia transmission. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, Dallas, United States, March 14-19 2010. 98

[109] H. Nguyen and P. Duhamel. Estimation of redundancy in compressed image and video data for joint source–channel decoding. In *Proc. GLOBECOM*, 2003. 57

[110] H. Nguyen and P. Duhamel. Iterative joint source–channel decoding of variable length encoded video sequences exploiting source semantics. In *Proceedings of ICIP*, 2004. 58

[111] H. Nguyen and P. Duhamel. Robust source decoding of variable-length encoded video data taking into accounts source constraints. *IEEE Transactions on Communications*, 53, no.7:pp. 1077–1084, July 2005. 57

[112] H. Nguyen and P. Duhamel. Iterative joint source-channel decoding of vlc exploiting source semantics over realistic radio-mobile channels. *IEEE Transactions on Communications*, 57, no. 6:1701–1711, June 2009. 57, 58

[113] H. Nguyen, P. Duhamel, J. Brouet, and D. Rouffet. Robust vlc sequence decoding exploiting additional video stream properties with reduced complexity. In *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, pages 375–378, June 2004. Taipei, Taiwan. 57

[114] J. K. Omura. On the Viterbi decoding algorithm. *IEEE trans. Information Theory*, 15(1):177–179, 1969. 7

[115] L. Ozarow. On a source-coding problem with two channels and three receivers. *Bell Syst. Techn. J.*, 59(10):1909–1921, 1980. 6, 48, 49

[116] J. Kovacevic P. L. Dragotti and V. Goyal. Quantized oversampled filter banks with erasures. In *Proceedings of the IEEE Data Compression Conference,*, pages 173–182, Snowbird, UT, March 2001. 6, 54

[117] G. Panza, E. Balatti, G. Vavassori, C. Lamy-Bergot, and F. Sidoti. Supporting network transparency in 4G networks. In *Proceedings of IST Mobile and Wireless Communication Summit*, June 19-23, 2005, Dresne, Germany. 8, 42, 43, 63

[118] S. Parrilli, M. Cagnazzo, and B. Pesquet-Popescu. Distortion evaluation in transform domain for adaptive lifting schemes. In *IEEE Workshop on Multimedia Signal Processing*, Cairns, Australia, 2008. 159, 160, 162, 163

[119] S. Parrilli, M. Cagnazzo, and B. Pesquet-Popescu. Estimation of quantization noise for adaptive-prediction lifting schemes. In *IEEE Workshop on Multimedia Signal Processing*, 2009. 159, 160, 162, 163, 165

[120] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988. 108, 111

[121] M. Pereira, M. Antonini, and M. Barlaud. Multiple description image and video coding for wireless channels. *EURASIP Special Issue on Recent Advances in Wireless Video*, 18(10):925–945, November 2003. 56

[122] L. Perros-Meilhac and C. Lamy. Huffman tree based metric derivation for a low-complexity soft VLC decoding. In *Proc. IEEE ICC*, volume 2, pages 783–787, 2002. 6, 56

[123] B. D. Pettijohn, W. Hoffman, and K Sayood. Joint source/channel coding using arithmetic codes. *IEEE Trans. on Comm.*, 49(5):826–836, 2001. 58

[124] G. Piella, B. Pesquet-Popescu, and H. J. A. M. Heijmans. Adaptive update lifting with a decision rule based on derivative filters. 9:329–332, October 2002. 159, 161

[125] Gemma Piella, Béatrice Pesquet-Popescu, and Henk J. A. M. Heijmans. Gradient-driven update lifting for adaptive wavelets. *Signal Processing: Image Communication*, 20(9-10):813–831, Oct.-Nov. 2005. 159, 161

[126] D. Zhang J. Xu G. Pau M. Trocan S. Brangoulo R. Xiong, X. Ji and V. Bottreau. Vidwav Wavelet Video Coding Specifications. Technical report, MPEG document, Poznan, July 2005. 10, 13, 39, 45, 46, 65, 72

[127] S. L. Regunathan R. Zhang and K.Roth. Video coding with optimal inter/intra-mode switching for packet loss resilience. *IEEE Journal on Selected Areas in Communications*, 18, no. 6:966–976, 2000. 45

[128] S. Rangan. Estimation with Random Linear Mixing, Belief Propagation and Compressed Sensing. *arXiv:1001.2228v2 [cs.IT]*, 18 May 2010. 27, 108, 120, 121

[129] S. Rangan. Generalized Approximate Message Passing for Estimation with random Linear Mixing. *arXiv:1010.5141v1 [cs.IT]*, 25 Oct 2010. 108, 120

[130] G. Rath and C. Guillemot. Frame-theoretic analysis of DFT codes with erasures. *IEEE Transactions on Signal Processing*, 52(2):447–460, 2004. 6, 54, 149, 153

[131] G. R. Redinbo. Decoding real block codes: Activity detection, wiener estimation. *IEEE Transactions on Information Theory*, 46(2):609–623, 2000. 6, 149

[132] I.S. Reed and G. Solomon. Polynomial codes over certain finite fields. *SIAM Journal of Applied Mathematics*, 8:300–304, 1960. 19, 41

[133] T. Richardson and U. Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008. 41

[134] L. Rizzo. Effective erasure codes for reliable computer communication protocols. *ACM Computer Communication Review*, 27(2):24–36, April 1997. 103

[135] Z. Rongfu. Content-adaptive spatial error concealment for video communication. *IEEE Transactions on Consumer Electronics*, pages vol. 50, no. 1, pp. 335–341, February 2004. 59

[136] G. Sabeva, S. Ben Jamaa, M. Kieffer, and P. Duhamel. Robust decoding of h.264 encoded video transmitted over wireless channels. In *Multimedia Signal Processing, 2006 IEEE 8th Workshop on*, October 2006. 6, 57

[137] K. Sayood and J. C. Borkenhagen. Use of residual redundancy in the design of joint source/channel coders. *IEEE trans. Communication*, 39(6):838–846, 1991. 6, 56

[138] K. Sayood, H. H. Otu, and N. Demir. Joint source/channel coding for variable length codes. *IEEE Trans. Commun.*, 48(5):787–794, may 2000. 6, 57

[139] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, 2007. 46

[140] S. D. Servetto, V. A. Vaishampayan, and N. J. A. Sloane. Multiple description lattice vector quantization. In *Proceedings of the IEEE Data Compression Conference*, pages 13–22, March 1999. 51

[141] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, October 1948. 5, 40, 55

[142] A. Shokrollahi. Raptor codes. *IEEE Transactions on Information Theory*, 52(6):2551–2567, June 2006. 41, 103

[143] C. Soldani, G. Leduc, F. Verdicchio, and A. Munteanu. Multiple description coding versus transport layer FEC for resilient video transmission. In *International Conference on Digital Telecommunications, ICDT*, page 20, August 29-31 2006. 19, 55, 56

[144] M. R. Stoufs, A. Munteanu, J. Barbarien, J. Cornelis, and P. Schelkens. Error protection of scalable soures: A comparative analysis of forward error correction and multiple description coding. In *Proceedings of IEEE International Conference on Digital Signal Processing*, pages 1–6, July 5-7 2009. 19, 55, 56

[145] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2):186–200, 1996. 159

[146] D. Taubman and A. Zakhor. Multirate 3-d subband coding of video. *IEEE Trans. Image Processing*, 3(5):572–588, 1994. 47

[147] R. Thobaben and J. Kliewer. On iterative source-channel decoding for variable-length encoded markov sources using a bit-level trellis. In *Proc. IV IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications (SPAWC'03)*, Rome, 2003. 58

[148] C. Tillier, T. Petrisor, and B. Pesquet-Popescu. A motion-compensated overcomplete temporal decomposition for multiple description scalable video coding. *EURASIP Journal on Image and Video Processing*, March 2007. 18, 55, 56, 93, 96, 97

[149] T. Tillo, M. Grangetto, and G. Olmo. A flexible error resilient scheme for jpeg 2000. In *Multimedia Signal Processing, 2004 IEEE 6th Workshop on*, pages 295–298, 29 Sept.-1 Oct. 2004. 57

[150] B. Usevitch. Optimal bit allocation for biorthogonal wavelet coding. pages 387–395, Snowbird, USA, March 1996. 160, 162, 164, 165

[151] P. P. Vaidyanathan. *Multirate Systems and Filterbanks*. Prentice-Hall, Englewood-Cliffs, NJ, 1993. 6, 20, 28, 126, 149, 152, 153, 156

[152] V. A. Vaishampayan. Design of multiple description scalar quantizers. *IEEE Transactions on Information Theory*, 39(3):821–834, May 1993. 50, 51

[153] V. A. Vaishampayan and J.-C. Batllo. Asymptotic analysis of multiple description quantizers. *IEEE Transactions on Information Theory*, 44(1):278–284, 1998. 51

[154] V. A. Vaishampayan and J. Domaszewicz. Design of entropy-constrained multiple-description scalar quantizers. *IEEE Transactions on Information Theory*, 40(1):245–250, January 1994. 51

[155] V. A. Vaishampayan, N. J. A. Sloane, and S. D. Servetto. Multiple description vector quantization with lattice codebooks: Design and analysis. *IEEE Transactions on Information Theory*, 47(5):1718–1734, July 2001. 51

[156] R. E. Van Dyck and D. J. Miller. Transport of wireless video using separate, concatenated and joint source-channel coding. *Proc. IEEE*, 87(10):1734–1750, 1999. 56

[157] R. Venkataramani, G. Kramer, and V. K. Goyal. Multiple description coding with many channels. *IEEE Transactions on Information Theory*, 49(9):2106–2114, September 2003. 6, 48

[158] M. Vetterli and J. Kovaˇcević. *Wavelets and Subband Coding*. Prentice Hall, Englewood Cliffs, 1995. 153

[159] E. Walter and L. Pronzato. *Identification of Parametric Models from Experimental Data*. Springer-Verlag, London, 1997. 132

[160] J. Wang, L. Wang, T. Ikenaga, and S. Goto. Standard deviation and intra prediction mode based adaptive spatial error concealment (SEC) in H.264/AVC. *EICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, pages vol. E.91–A, no. 10, pp. 2954–2962, October 2008. 59

[161] L. Wang, T. Ikenaga, and S. Goto. An adaptive spatial error concealment (SEC) with more accurate MB type decision in H.264/AVC. *5th International Conference on Visual Information Engineering*, pages pp. 759–764, 2008. 59

[162] T. Wang, M. T. Orchard, and A. R. Reibman. Multiple description image coding for noisy channels by pairing transform coefficients. In *Proceedings of IEEE International Workshop on Multimedia Signal Processing, MMSP*, pages 419–424, June 23-25 1997. 52

[163] Y. Wang, A. R. Reibman, and S.Lin. Multiple description coding for video delivery. In *Proceedings of the IEEE*, volume 93, pages 57–70, January 2005. 55

[164] Y. Wang and Q. Zhu. Error control and concealment for video communication: A review. *Proceedings of the IEEE*, 86:974–997, 1998. 52, 59

[165] J. K. Wolf, A. D. Wyner, and J. Ziv. Source coding for multiple description. *Bell System Technical Journal.*, 59(8):1417–1426, October 1980. 6, 48

[166] X. Yang and K. Ramchandran. Optimal multiple description subband coding. In *Proceedings of the IEEE International Conference on Image Processing, ICIP*, volume 1, pages 684–658, Chicago, October 1998. 54

[167] M. R. Yazdani, S. Hemati, and A. H. Banihashemi. Improving belief propagation on graphs with cycles. *IEEE Communications Letters*, 8(1):57–59, January 2004. 145

[168] R. W. Yeung. *A First Course in Information Theory*. Springer, New-York, 2002. 6, 57

[169] Y.Lu Q.Huang Y.Zhang, W.Gao and D.Zhao. Joint source-channel rate-distortion optimization for h.264 video coding over error-prone networks. *IEEE Transactions on Multimedia*, 9, no. 3:445–454, 2007. 42, 45

[170] J.Cai Z. He and C.W.Chen. Joint source-channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 12, no. 6:511–523, 2002. 45

[171] B. J. Kim Z. Xiong and W. A. Pearlman. "progressive video coding for noisy channels",. *IEEE International Conference on Image Processing*, 1:334–337, 1998. 46

[172] Y. Zhao and D. Tian. Spatial error concealment based on directional decision and intra prediction. *IEEE International Symposium on Circuits and Systems*, 3:2899–2902, May 2005. 59

[173] J. Zheng and L.-P. Chau. A motion vector recovery algorithm for digital video using lagrange interpolation. *IEEE Transactions on Broadcasting*, 49(4):383 – 389, 2003. 59

[174] Q.-F. Zhu and Y. Wang. *Error concealment in visual communications*. A. R. Reibman and M. T. Sun, eds. Compressed Video over Networks, pp. 217-250. Marcel Dekker, Inc., 2000. 59