

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique

Université Mentouri de Constantine  
Faculté des Sciences de l'Ingénieur  
Département d'Electronique

Mémoire  
Présenté en vue de l'obtention du diplôme de Magister en  
électronique

Option : Traitement du Signal

Présenté par :  
Mr Boudjada Ahmed

Thème

Codage Vidéo Distribué

Devant le jury :

Président :	Abdelhak Bennia	Prof Univ. Constantine
Rapporteurs:	Mohammed Khamadja	Prof Univ. Constantine
Examineurs:	Abdelfatah Charef	Prof Univ. Constantine
	Attaf Farrouki	M.C Univ. Constantine

Promotion 2008

# Résumé

Une propriété importante des codeurs vidéo de Wyner-Ziv est le décalage de l'estimation de mouvement au décodeur. Cette opération est effectuée sur les images clés qui sont codées en mode Intra pour interpoler les images Wyne-Ziv codées.

Dans ce mémoire, nous proposons une nouvelle architecture d'un codeur de Wyner-Ziv à base de bloc. Au lieu de coder l'image entière par un codage Intra ou Wyner-Ziv, on va combiner les deux modes pour coder tous les macros blocs de l'image, une petite partie de chaque macro bloc est d'abord codé en mode Intra avec une certaine qualité, ensuite les autres parties sont graduellement codées en mode Wyne-Ziv pour améliorer la qualité de l'image décodé. Cet arrangement la, nous permet d'augmenter l'efficacité de l'estimation partielle de mouvement progressivement, pour générer la meilleure information de bord.

En plus d'être plus pratiques, les simulations obtenus avec le codeur proposé ; montrent des résultats prometteurs qui présente une bonne opportunité pour diminuer le gap entre les performances des codeurs Wyner-Ziv et des codeurs Inter traditionnel, ainsi d'avoir un codeur Wyner-Ziv robuste aux erreurs de canal.

# Summary

An important property of Wyner-Ziv video coders is the shift of motion estimation to the decoder. This operation is performed on intra key frames and interpolated to Wyne-Ziv coded frames.

In this thesis, we propose a new block-based architecture of a Wyner-Ziv coder that does not use key frames and exploits partial matching motion estimation at the decoder. A small part of each macroblock is intra coded first with a given quality and the other parts are gradually coded. The decoder can thus try to estimate the real motion of the macroblock and gradually improved motion is used to generate better side information.

In addition to being more practical, the simulations with the proposed coder show promising results and good opportunities to bridge the efficiency gape between the Wyner-Ziv coders and conventional inter coding and also a practical robust Wyner-Ziv coder to channel losses.

## ملخص

تعود بداية البحث في مجال ضغط المعلومات بدون ضياع و التي تأتي من عدة منابع متشابهة إلى سنة 1973، حيث برهن J.K.Wolf و D.Slepian بأن حد عملية ضغط المعلومات بصورة موزعة و إعادة جمعها بصورة مشتركة يتمثل في الأنثروبي المشترك لهذه المنابع. الشيء المثير للاهتمام هو أن هذا الحد يساوي الحد الناتج عن عملية ضغط المعلومات معا.

إن التطبيق الميداني لهذه النظرية يمر عبر تشفير عشوائي، و الذي يستحيل تطبيقه في الواقع. تجدر الإشارة بأن التشفير عبر القناة قد يسمح بالوصول إلى هذا الحد النظري و خصوصا أن المعلومات متشابهة إذ أن كل واحدة تعتبر صورة مشوشة للأخرى.

برغم من أن بداية دراسة تشفير المعلومات بصورة موزعة يعود إلى سنة 1970، إلا أن الجهود لتطوير حلول عملية لضغط الفيديو حسب Wyner – Ziv قد بدأت مؤخرا ، و يعود ذلك إلى ظهور تطبيقات جديدة نذكر منها على سبيل المثال : أنظمة الفيديو الهوائية ، أنظمة المراقبة ، إضافة إلى ظهور الكاميرات المستخدمة في الهواتف المحمولة ...، و التي تتطلب احتياجات أخرى تختلف عن تلك التي تمنحها أنظمة تشفير الفيديو المعتمدة و نقصد بذلك MPEG-X و H.26x و خصوصا أن هذه التطبيقات تتميز بمصادر طاقة محدودة (بطاريات) على مستوى المشفر. إن عملية استغلال التشابه الموجود في الصور المتجاورة زمنيا في أنظمة تشفير الفيديو المعتمدة MPEG-X و H.26x يمر عبر عملية معقدة لتحديد الحركات عبر هذه الصور، و التي تكون سبب في استهلاك كمية معتبرة من الطاقة في المشفر. يجدر التنويه بأن عملية استغلال التشابه في نظام تشفير الفيديو حسب Wyner – Ziv يتم فقط عبر جهاز فك الشفرات. إذا إضافة إلى تبسيط عمل المشفر ، يهدف نظام التشفير الجديد إلى إظهار مناعة أكبر نحو الأخطاء التي يمكن أن تقع أثناء عملية إرسال المعلومات بحيث أن هذا النظام لا يعتمد على حلقة التنبؤ المعتادة في المشفر و التي ينجم عنها عدة أخطاء.

تتمثل الأهداف الرئيسية لهذه المذكرة في مراجعة الأعمال الحالية في مجال تشفير الفيديو بصورة موزعة إضافة إلى وضع و تطوير حل تطبيقي يسمح بتشفير الفيديو بهذه الصورة.

# Remerciements

Louange à dieu le miséricordieux qui nous permis de bien accomplir ce modeste travail.

Je tiens à remercier vivement mon directeur de mémoire, le professeur Mohammed KHAMADJA d'avoir dirigé ce travail, ainsi que pour toutes les facilités matérielles qu'il m'accordées pour l'accomplissement de ce travail.

Je remercie en particulier Mr Said BENIERBAH pour m'avoir co-dirigé avec compétence tout le long de ce travail, ainsi que pour toute l'aide qu'il m'a apporté.

Je remercie le professeur Abdelhak BENNIA d'avoir accepter de présider le jury, et également le professeur Abdelfatah CHAREF et le maître de conférences Attaf FARROUKI qui ont acceptés d'être membre du jury.

A l'ensemble des enseignants qui ont fait de leur mieux pour nous instruire.

A toute personne qui a contribué de prés ou de loin, directement ou indirectement à l'accomplissement de ce travail.

# Dédicaces

De tout mon cœur je dédie ce modeste travail

A ma chère mère, la lumière qui nous a guidés vers le chemin de savoir.

A mon cher père, pour leur sacrifice.

A mes chères sœurs.

A toute ma famille.

A mes chers amis chaque un et son nom.

A mes ceux que j'aime.

# Table des matières

## Chapitre 1 Etat de l'art du codage vidéo

1. Le signal vidéo .....	4
1.1 Définition d'une image et des types d'images.....	4
1.2 Définition de la vidéo numérique.....	5
2. Principe de la compression vidéo.....	6
2.1 Redondance spatiale .....	6
2.2 Redondance temporelle.....	7
2.3 Redondance psycho-visuelle .....	8
3. Mesures de performance.....	8
3.1 Taux de compression.....	8
3.2 Entropie.....	8
3.3 Mesures de distorsion.....	10
4. Codage vidéo traditionnel.....	11
4.1 Cadre théorique.....	11
4.2 L'approche générale de codage vidéo traditionnel.....	11
5. Codage vidéo distribué.....	15
5.1 Cadre théorique.....	15
5.1.1 Théorème de Slepian-Wolf.....	15
5.1.2 Théorème de Wyner-Ziv.....	17
5.2 L'approche générale du codage vidéo de Wyner-Ziv.....	19
5.2.1 Encodeur de Wyner Ziv .....	20
5.2.2 Décodeur de Wyner- Ziv .....	22
5.3 Les solutions du CVD les plus intéressantes .....	24
5.3.1 Approche suivie par l'équipe de Stanford .....	24
5.3.1.1 Codage.....	24
5.3.1.2 Décodage .....	25
5.3.1.3 Performances.....	26
5.3.2 Approche suivie par l'équipe de Berkeley.....	27
5.3.2.1 Codage.....	28
5.3.2.2 Décodage.....	31
5.3.2.3 Performances.....	32

## Chapitre 2 Codage vidéo de Wyner-Ziv avec turbo code

1. Implémentation d'un turbo code poinçonné.....	34
1.1 Turbo encodeur.....	35
1.1.1 Encodeurs RSC .....	35
1.1.2 Entrelacement.....	38
1.1.3 Poinçonnage de turbo code.....	39
1.2 Turbo décodeur.....	41
1.2.1 Algorithme de décodage SISO dans un canal AWGN.....	42
1.2.2 Opération de décision .....	47
2. Implémentation d'un CVD Intra/WZ.....	48
2.1 Encodeur Wyner-Ziv.....	51
2.1.1 Quantification.....	51
2.1.2 Turbo encodeur .....	51
2.2 Décodeur Wyner-Ziv.....	52
2.2.1 Estimation des images .....	52
2.2.2.1 Turbo décodeur .....	53
2.2.2.2 Algorithme de décodage SISO dans un canal Laplacienne.....	53
2.2.3 Reconstruction.....	57

## Chapitre 3 Résultats et discussion

1. Résultats de simulation du turbo code.....	60
1.1 L'évaluation du changement de nombre des itérations sur l'efficacité de décodage.....	61
1.2 L'évaluation du changement de nombre des permutations sur l'efficacité de décodage.....	62
1.3 L'évaluation du changement de la longueur L sur l'efficacité de décodage.....	63
2. Résultats de simulation du CVD.....	64
2.1 L'évaluation du changement de facteur de la quantification Intra sur l'efficacité du CVD.....	64
2.2 L'évaluation du changement de la longueur des séquences vidéo sur l'efficacité du codage.....	70
2.3 Comparaison des performances de CVD par apport à la norme Intra H.264.....	71



# Liste des Figures

Figure 1.1 : Schéma d'un lien de communication radio-mobile utilisant le principe de codage vidéo distribué.....	4
Figure 1.2 : Elément d'une image : le pixel.....	5
Figure 1.3 : Exemple d'exploitation de la redondance temporelle.....	7
Figure 1.4 : Paradigme de codage traditionnel.....	11
Figure 1.5 : Codage d'une image I.....	12
Figure 1.6 : Codage d'une image P ou B.....	12
Figure 1.7 : Exemple de configuration de codage d'une séquence vidéo.....	13
Figure 1.8 : Diagramme fonctionnel d'un codeur vidéo traditionnel.....	14
Figure 1.9 : Les normes de codage vidéo.....	14
Figure 1.10 : Compression distribué de deux séquences aléatoires discrètes i.i.d.....	15
Figure 1.11 : Région du taux achevé selon le théorème de Slepian-Wolf [1].....	16
Figure 1.12 : Rapport entre le codage canal et le codage de Slepian-Wolf.....	17
Figure 1.13 : Compression avec perte et avec information latéral au décodeur.....	18
Figure 1.14 : diagramme fonctionnel d'un codeur de Wyner-Ziv.....	19
Figure 1.15 : architecture du codeur vidéo Wyner-Ziv du groupe de Stanford [41]..	25
Figure 1.16 : Architecture d'un codeur PRISME [40].....	27
Figure 1.17 : Architecture d'un décodeur PRISME [40].....	27
Figure 1.18 : Sélection des coefficients DCT [40].....	29
Figure 1.19 : Contenu de bitstream [40].....	30
Figure 2.1 : Structure d'un Turbo codeur.....	35
Figure 2.2 : Encodeur RSC.....	36
Figure 2.3 : Diagramme de treillis d'un encodeur RSC.....	37
Figure 2.4 : Séquence de parité de 16 bits à la sortie d'un encodeur RSC.....	40
Figure 2.5 : Composition de premier bloc de 4 bits.....	40
Figure 2.6 : Implémentation d'un Turbo décodeur.....	41
Figure 2.7 : Architecture du codeur vidéo distribué.....	48
Figure 2.8 : L'ordre de codage des blocs WZ (blanc) et Intra (gris) pour un MB.....	49
Figure 2.9: l'information disponible au décodeur après chaque étape .....	50
Figure 2.10 : distribution de la résiduel $x$ de la séquence vidéo QCIF de Foreman ...	55
Figure 2.11 : Fonction de reconstruction d'un quantificateur scalaires uniforme.....	58

Figure 3.1 : Effets du nombre des itérations.....	61
Figure 3.2 : Effets du nombre des permutations.....	62
Figure 3.3 : Effets de la longueur $L$ .....	63
Figure 3.4 : Performance débit-distorsion total des 11 premières images de la séquence Hall QCIF, 15 Hz avec les facteurs de quantifications de codeur H.264 $q=20, q=24, q=28$ et $q=32$ respectivement.....	65
Figure 3.5 : Performance débit-distorsion total des 11 premières images de la séquence Foreman QCIF, 15 Hz avec les facteurs de quantifications de codeur H.264 $q=20, q=24, q=28$ et $q=32$ respectivement.....	65
Figure 3.6 : Performance débit-distorsion total des 11 premières images de la séquence Soccer QCIF, 15 Hz avec les facteurs de quantifications de codeur H.264 $q=20, q=24, q=28$ et $q=32$ respectivement.....	66
Figure 3.7 : Performance débit-distorsion WZ des 11 premières images des séquence Hall, Foreman et Soccer QCIF, 15 Hz avec un facteur de quantification de codeur H.264 $q=24$ .....	68
Figure 3.8 : Performance débit-distorsion total des 11 et 101 premières images de la séquence Hall QCIF, 15 Hz avec un facteur de quantification de codeur H.264 $q=20$ .....	70
Figure 3.9 : Performance débit-distorsion total des 11 et 101 premières images de la séquence Foreman QCIF, 15 Hz avec un facteur de quantification H.264 $q=28$ .....	70
Figure 3.10 : Performance débit-distorsion total des 11 et 101 premières images de la séquence Soccer QCIF, 15 Hz avec un facteur de quantification de codeur H.264 $q=28$ .....	71
Figure 3.11 : Performance débit-distorsion total des 11 et 101 premières images de la séquence Hall QCIF, 15 Hz avec CVD ( $q=20$ ) et avec codage Intra H.264.....	72
Figure 3.12 : Performance débit-distorsion total des 11 et 101 premières images de la séquence Foreman QCIF, 15 Hz avec CVD ( $q=28$ ) et avec codage Intra H.264.....	72
Figure 3.13 : Performance débit-distorsion total des 11 et 101 premières images de la séquence Soccer QCIF, 15 Hz avec CVD ( $q=28$ ) et avec codage Intra H.264.....	73

# Liste des Tableaux

Tableau : 2.1 Les états des transitions possibles de l'encodeur RSC.....	38
Tableau : 2.2 Entrelacement d'une séquence à 5 bits.....	39
Tableau : 3.1 : Les pertes en PSNR en (dB) de la séquence Hall.....	66
Tableau : 3.2 : Les pertes en PSNR en (dB) de la séquence Foreman.....	67
Tableau : 3.3 : Les pertes en PSNR en (dB) de la séquence Soccer.....	67
Tableau : 3.4 : Les débits Intra de la séquence Hall.....	69
Tableau : 3.5 : Les débits Intra de la séquence Foreman .....	69
Tableau : 3.6 : Les débits Intra de la séquence Soccer.....	69

# Liste des abréviations

APP	A Posteriori Probability
AWGN	additive white Gaussian noise
BCJR	Bahl, Cocke, Jelinek, et Raviv
CR	taux de compression
CVD	codage vidéo distribué
$d$	distorsion
dB	Décibel
DCT	Discrete Cosine Transform
EM	estimation du mouvement
I	codage Intra
LAPP	logarithme du rapport de probabilité posteriori
MAP	Maximum A Posteriori
MB	macro block
ML	Maximum Likelihood
MSE	erreur quadratique moyenne
PDWZ	pixel domaine Wyner-Ziv
PRISM	Power-efficient, Robust, hghcompression, Syndrome-based Multimedia Coding
PPSNR	rapport signal sur bruit crête à crête
Q	quantification
QCIF	Quarter Common Intermediate Format for images
RSC	récurifs systématiques convolutionnels
SDA	somme de la différence absolue
SISO	Soft-In Soft-Out
SNR	rapport signal sur bruit
SOVA	Soft-Output Viterbi Algorithm
SVH	système visuel humain
SW	Slepian-Wolf
TCD	Transformée en Cosinus Discrète
TCP	turbo code poinçonné
WZ	Wyner-Ziv

# Introduction générale

Depuis quelques années, le monde du multimédia a assisté à une véritable explosion de ses applications. Les nouvelles technologies qui paraissaient révolutionnaires à leur arrivée sur le marché, sont désormais entrées dans la vie de tous les jours. Il est maintenant impensable de ne pas pouvoir téléphoner, tout comme il serait impossible pour les entreprises et les particuliers de se passer de l'Internet. Si l'échange de la parole et du texte a longtemps paru suffisant, il s'agit maintenant de transmettre des documents vidéo et audio de natures diverses. Or, la vidéo numérique pose par sa taille importante, de nombreux problèmes, que ce soit pour la transmission (surtout pour les applications Temps Réel) ou pour le stockage. Dans ce contexte, il était devenu indispensable d'élaborer un standard international de compression vidéo.

Les deux organismes les plus actifs dans la normalisation des systèmes de compression vidéo sont l'UIT-T (Union International des Télécommunications) [1] et l'ISO/MPEG (Moving Picture Experts Group) [2]. Les codeurs issus des ces deux groupes de normalisation reposent sur des principes de base communs, mais différents de part leur configuration choisie en fonction de l'application.

Ces dernières années avec les applications naissantes telles que les réseaux de surveillance sans fil, les réseaux des sondes multimédia et les caméras PC sans fil, l'architecture du codage vidéo traditionnel (des deux organisations) est défiées. Ces applications ont des exigences très différentes de celles des systèmes vidéo traditionnels. Ces applications ont stimulé les efforts pour développer un nouveau paradigme de codage nommé par le codage vidéo distribué (CVD), qui est basé sur les deux théorèmes de Slepian-Wolf (1973) et celle de Wyner-Ziv (1976).

## Notre contribution

L'objectif principal de ce mémoire est de :

Examiner et analyser les solutions du codage vidéo distribuées les plus pertinentes disponibles dans la littérature.

Implémenter un turbo code poinçonné pour l'exploité dans un codeur vidéo distribué dans le domaine des pixels.

# Organisation du mémoire

Ce mémoire décrit en détail le développement d'une solution du codage de Wyner-Ziv dans le domaine des pixels (à base des pixels).

- Le principe de base de la compression vidéo traditionnel est rappelé dans le Chapitre 1. On va examiner aussi dans ce chapitre, les approches de codage vidéo distribuées les plus pertinentes, en particulier le travail des deux groupes de recherche qui sont responsables du développement des systèmes les plus appropriés du codage vidéo distribué de nos jours : Le groupe de Kannan Ramchandran à Berkeley et le groupe de Bernd Girod à Stanford [3] et [4].
- Dans le chapitre 2, notre codeur vidéo de Wyner-Ziv dans le domaine des pixels est présenté, en particulier on va décrire l'implémentation d'un turbo code poinçonné.
- Dans le chapitre 3, les résultats des performances de notre turbo code et de codeur vidéo distribué sont examinés.

# Chapitre 1

## Etat de l'art du codage vidéo

### Introduction

Afin de réduire le débit, les normes de compression vidéo actuelle comme H26x et MPEGx utilisent la corrélation temporelle en mettant en œuvre un codage prédictif compensé en mouvement. Ce choix se traduit par des encodeurs dont la complexité est en général grandement plus élevée (5 à 10 fois) que celle du décodeur. Cette différence est donc liée au fait que l'estimation et la compensation de mouvement sont calculées et effectuées dans l'encodeur, tandis que le décodeur va simplement utiliser les vecteurs de mouvement pour reconstruire l'image décodée. Ce schéma de conception asymétrique est tout à fait adapté au cas où le contenu vidéo est compressé une seule fois et décompressé plusieurs fois. C'est le cas, par exemple, de la diffusion TV où on utilise un seul encodeur complexe et coûteux avec plusieurs décodeurs simples. C'est aussi le cas des terminaux mobiles qui téléchargent ou reçoivent le flux des vidéos à partir de serveurs. Cependant, elles sont peu adaptées pour des applications où le codage doit être réalisé sur des terminaux à faible capacité, par exemple un mobile transmettant des flux vidéo vers une station de base Figure 1.1, pour des services de type MMS ou peer-to-peer, ou bien pour le flux des séquences d'images par une multitude de caméras.

Pour faciliter ces possibilités dans le cas de la vidéo il est plus judicieux de rechercher un schéma de codage dual du précédent avec un encodeur de complexité relativement limitée et un décodeur disposant d'une puissance de traitement nettement plus importante. Dans ce contexte le codage de sources distribuées CSD (DSC : Distributed Source Coding) peut être vu comme une façon différente d'exploiter cette corrélation temporelle en considérant, par exemple, que dans deux images successives, la deuxième constitue une version bruitée de la première et en exploitant cette corrélation au décodeur.

Le codage de sources distribuées est introduit par Slepian-Wolf [5] en 1973. Dans ce cas, il est possible de réaliser sans perte en efficacité de compression, un codage séparé et un décodage conjoint de deux sources corrélées  $X$  et  $Y$  à valeurs discrètes.

Plus tard ce principe a été étendu par Wyner et Ziv [6] au cas des sources Gaussiennes à valeurs continues (compression avec perte). De ces résultats théoriques, le défi du codage vidéo distribué s'annonce : Quel système de codage doit être développé afin d'approcher les limites du débit suggéré par la théorie de l'information ?

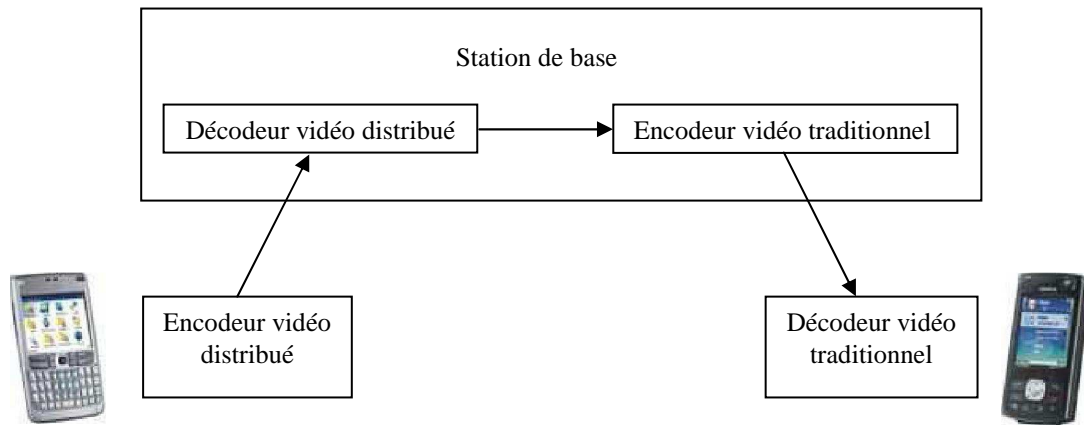


Figure 1.1 : Schéma d'un lien de communication radio-mobile utilisant le principe de codage vidéo distribué

La Figure 1.1 montre l'utilisation du codage de sources distribuées dans un schéma de communication radio-mobile. Les deux terminaux mobiles dans la Figure 1.1 présentent des complexités au codage et au décodage très faibles grâce à l'utilisation respective d'un encodeur vidéo distribué et d'un décodeur vidéo traditionnel (MPEG-X ou de H.26X). En outre, ce schéma de codage doit être robuste aux erreurs de canal de sorte que le décodeur puisse récupérer la scène avec une qualité de reconstruction élevée.

## 1. Le signal vidéo

### 1.1 Définition d'une image et des types d'images

Une image est stockée en mémoire sous forme de collection de points élémentaires appelés pixels. Nous pouvons considérer une image numérique comme une page de nombres organisés en tableau ou en matrice. Chaque nombre représente les caractéristiques du pixel. La position de chaque pixel peut être exprimée par deux coordonnées sur l'axe horizontal X et l'axe vertical Y comme le montre la Figure ci-dessous.



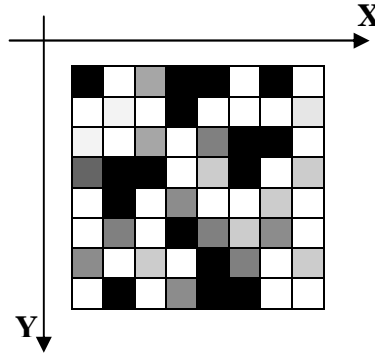


Figure 1.2 : Élément d'une image : le pixel

Le codage d'un pixel dépend du type d'image et nous en recensons trois types

- Les images à deux niveaux : une image en noir et blanc est l'exemple le plus courant. Toute image décrite avec deux valeurs correspond à ce type.
- Les images à plusieurs niveaux de gris : les images de nos téléviseurs en noir et blanc sont de ce type. Ce sont des images dont les pixels ont des niveaux de gris représentés avec des nombres allant de 0 à 255. Notons que par convention le niveau de gris 0 correspond à du noir, et le niveau 255 à du blanc.
- Les images couleurs : la couleur peut être codée, soit par superposition de trois couleurs primaires RVB, soit par composition d'informations de luminance et de chrominance  $YCrCb$ .

## 1.2 Définition de la vidéo numérique

La vidéo est une succession d'images animées. Le principe fondamental de la vidéo est que l'œil humain a la possibilité de retenir pendant un certain temps (de l'ordre du dixième de seconde) toute image imprimée sur la rétine. Il suffit donc de faire défiler un nombre suffisant d'images par seconde, pour que l'œil ne se rende pas compte qu'il s'agit d'images distinctes.

La vidéo numérique est tout simplement une suite d'images formées d'une matrice de pixels. Pour obtenir des images en couleur, il faut utiliser au moins 8 bits par pixel, ce qui correspond à 256 couleurs. En fait, avec 8 bits par pixel, on obtient de la vidéo numérique noir et blanc de haute qualité. Pour la vidéo numérique couleur, on utilise 8 bits pour chaque couleur RVB, soit donc 24 bits par pixel, ce qui correspond à environ 16,8 millions de couleurs.

## 2. Principe de la compression vidéo

La compression vidéo est nécessaire pour la transmission des données vidéo numériques dans les réseaux à bande passante limitée d'aujourd'hui ainsi que pour les applications où le stockage constitue une limite. Imaginons, pour transmettre des données vidéo numériques à 24 bits par pixel brut échantillonné selon une résolution spatiale de 720x480 et une résolution temporelle de 30 images par seconde, il faudrait un débit binaire de 248 Mb/s.

Le principe fondamental de la compression vidéo est de réduire autant que possible les redondances d'informations dans les données sans que cela soit perceptible par l'œil humain. Il y a donc un compromis entre le taux de compression et la qualité de l'image. En fait, celle-ci devient de plus en plus médiocre avec l'augmentation du taux de compression.

Deux grandes méthodes de compression existent : la compression sans perte et la compression avec perte [7] :

- Dans le cas de la compression sans perte, les données décodées à l'arrivée par le récepteur sont strictement identiques aux données codées au départ par l'émetteur. Ce type de compression, permettant au mieux un taux de compression de 2:1, est évidemment insuffisant pour la compression vidéo.
- Dans le cas de la compression avec perte, les données à la sortie du décodeur sont différentes par rapport à celles à l'entrée du l'encodeur. C'est ce type de compression qui est le plus utilisé en vidéo car on peut accepter des pertes d'informations qui ne sont pas toujours visibles à l'œil et qui se traduisent par de nets gains de compression. En compression avec pertes, on peut atteindre des taux de compression allant jusqu'à 300:1.

La compression des données vidéo numériques sans dégradation significative de la qualité est possible lorsque les séquences vidéo affichent un degré élevé [8] :

- de redondance spatiale : corrélation entre les pixels voisins.
- de redondance temporelle : corrélation entre les images vidéo.
- de redondance psycho-visuelle : propriétés de la vue humaine.

### 2.1 Redondance spatiale

C'est la corrélation dans chaque image prise indépendamment des autres qui présente des zones uniformes plus ou moins grandes dans lesquelles les pixels ont des valeurs

très voisines. On peut diminuer cette redondance en codant chaque image séparément en codage Intra.

## 2.2 Redondance temporelle

Deux images qui se suivent dans une séquence vidéo sont quasiment identiques. Ceci implique que la différence entre une image et la suivante soit relativement faible sauf lors d'un changement de plan. Autrement dit, la position d'un bloc de pixels varie généralement peu d'une image à l'autre. Le but est alors de ne coder (ou stocker) que ce qui est modifié lors du passage d'une image à une autre. Ce type de codage est appelé codage Inter. La redondance des données, est illustrée dans la Figure 1.3.



(a) Image 143



(b) Image 144



(c) Image de différence

Figure 1.3 : Exemple d'exploitation de la redondance temporelle

Dans ces deux images, on remarque qu'il n'y a pas une grande différence puisque la camera n'a pas changé ni de plan ni de position. Donc, la seule différence se trouve dans les parties qui ont subi un mouvement. Ceci peut être démontré par la Figure 1.3 (c) qui représente la différence entre les deux images successives (Figure 1.3 (a) et

Figure 1.3 (b)) dont on a ajouté 128 afin de mieux visualiser les redondances temporelles. La partie grise de cette image correspond à la similitude des deux images (a) et (b).

## 2.3 Redondance psycho-visuelle

En plus d'éliminer les redondances temporelles et spatiales, la redondance psycho-visuelle est généralement réduite elle aussi. La mesure la plus significative est une résolution réduite du détail des couleurs par rapport au détail de la luminance qui permet un meilleur rapprochement avec les caractéristiques de la perception humaine. Notant que des résultats expérimentaux ont prouvé que l'œil est beaucoup plus sensible aux variations fines d'intensité lumineuse (luminance) qu'à celles de la couleur (chrominance).

## 3. Mesures de performance

### 3.1 Taux de compression

En plus du débit  $R$  qui est le nombre de bits par pixels, une mesure courante pour déterminer le degré de compression obtenu est le taux de compression  $CR$  (compression ratio). Il est défini par:

$$CR = \frac{\text{nombre de bits de l'image originale}}{\text{nombre de bits de l'image comprimée}} \quad (1.1)$$

Pour une même méthode de compression et un même  $CR$  réalisés sur des images distinctes, la qualité obtenue peut être très variable d'une image à l'autre. Les propriétés statistiques des images originales jouent un rôle prépondérant dans le résultat obtenu. Par exemple avec une image sur-échantillonnée, donc très redondante, il est facile d'obtenir des taux élevés.

### 3.2 Entropie

En théorie de l'information l'entropie d'une variable aléatoire  $X$  est une mesure quantitative de l'incertitude associée aux valeurs prises par celle-ci. Elle est le nombre de bits moyens nécessaires pour décrire  $X$ , notée  $H(X)$ . Cette quantité, exprimée en

bits/symbole.

Soit une variable aléatoire discrète  $X$  qui prend des valeurs dans un ensemble de possibilité  $\Omega$ , l'entropie est définie par :

$$H(X) = - \sum_{x \in \Omega} p(x) \log_2(p(x)) \quad (1.2)$$

Sachant que l'entropie conjointe de deux variables aléatoires ( $X \in \Omega$  et  $Y \in \Psi$ ) avec une distribution (conjointe)  $Pr(X = x, Y = y)$  est définie par :

$$H(X, Y) = - \sum_{(x, y) \in \Omega \times \Psi} p(x, y) \log_2(p(x, y)) \quad (1.3)$$

L'entropie conjointe de deux événements est toujours supérieure ou égale à l'entropie de chaque événement ( $H(X, Y) \geq H(X)$  et  $H(X, Y) \geq H(Y)$ ). Cela est dû au fait que  $p(x, y) = p(y/x)p(x) \leq p(x)$ . L'entropie conjointe peut aussi s'écrire sous la forme suivante :

$$H(X, Y) = H(X) + H(Y/X) = H(Y) + H(X/Y) \quad (1.4)$$

Où  $H(Y/X)$  et  $H(X/Y)$  sont les entropies conditionnelles respectivement de  $Y$  sachant  $X$  et de  $X$  sachant  $Y$ , sont définies par :

$$H(X/Y) = - \sum_{(x, y) \in \Omega \times \Psi} p(x, y) \log_2(p(x/y)) \quad (1.5)$$

$$H(Y/X) = - \sum_{(x, y) \in \Omega \times \Psi} p(x, y) \log_2(p(y/x)) \quad (1.6)$$

Par analogie, l'entropie d'une image numérique est inversement liée à la probabilité d'apparition des niveaux de gris dans l'image. Plus une valeur de gris  $k$  est rare, plus sa probabilité d'apparition  $p(k)$  est faible, et cela contribue à une entropie globale plus grande.

### 3.3 Mesures de distorsion

La distorsion ( $d$ ) est l'erreur introduite par l'opération de compression, due au fait qu'éventuellement l'image reconstruite n'est pas exactement identique à l'image originale.

La mesure de distorsion utilisée généralement en compression d'image est l'erreur quadratique moyenne  $MSE$ . Cette grandeur est définie par la moyenne des écarts au carré entre le pixel  $(m,n)$  de l'image originale  $I(m,n)$ , et le pixel  $(m,n)$  de l'image reconstruite  $I'(m,n)$ .

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [I(m,n) - I'(m,n)]^2 \quad (1.7)$$

On référence cette erreur par rapport à la dynamique de l'image en décibels. On obtient un rapport signal sur bruit crête pour une image dont le maximum est  $2^R - 1$  dénoté  $PSNR$  (peak SNR). Si le minimum de l'image est nulle (image bien cadrée) on obtient le rapport signal sur bruit crête à crête noté  $PPSNR$  (peak to peak SNR). Lorsque la reconstruction est parfaite, la  $MSE$  est nulle et le  $PPSNR$  tend vers l'infini.

$$PSNR \text{ ou } PPSNR = 10 \cdot \log_{10} \frac{(2^R - 1)^2}{MSE} \text{ dB} \quad (1.8)$$

On peut dire que ces mesures de distorsion sont objectives et simples à calculer. Certaines méthodes de compression recherchent le meilleur compromis entre la performance et la distorsion, et optimisent des courbes taux-distorsion ou  $R(d)$  (rate distorsion).

Après qu'on a donné quelques généralités sur le signal et la compression vidéo, les sections suivantes servent à décrire le principe de base du codage vidéo. Pour les deux topologies du codage : codage vidéo traditionnel et codage vidéo distribué, le cadre théorique sur lequel les approches générales reposent ainsi que l'état de l'art seront ainsi décrit.

## 4. Codage vidéo traditionnel

### 4.1 Cadre théorique

Le codage vidéo traditionnel tenté de donner une réponse à des questions comme : Quelle est le débit minimum,  $R$ , nécessaire pour que deux séquences statistiquement dépendants  $X$  et  $Y$ , (par exemple deux images consécutives d'une séquence vidéo), puissent être parfaitement récupérés, c'est à dire sans erreurs, par un codage/décodage commun (Inter)? La Figure 1.4 décrit une telle question. Comme il est connu, la réponse à cette question est dérivée de la théorie de l'information, l'entropie commune  $R = R_x + R_y = H(X, Y)$ .

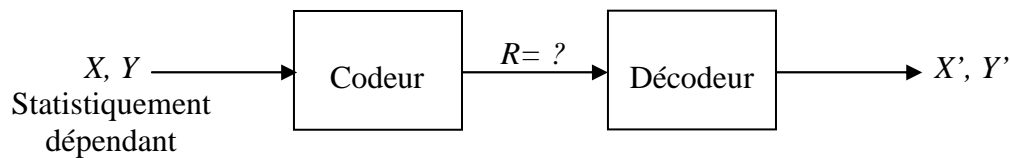


Figure 1.4 : Paradigme de codage traditionnel

Quand les séquences statistiquement dépendants,  $X$  et  $Y$ , sont indépendamment codés et décodés (Intra), les résultats pour la question correspondante sont également bien connus :  $R_x \geq H(X)$  et  $R_y \geq H(Y)$ . Dans cette situation, le nombre minimum de bits par source symbole nécessaire pour coder  $X$  et  $Y$  est donné par l'entropie de chaque source,  $H(X)$  et  $H(Y)$ , respectivement. Le débit total de transmission  $R$  associé est donné par  $R = R_x + R_y \geq H(X, Y)$ .

### 4.2 L'approche générale de codage vidéo traditionnel

Les codeurs normalisés exploitent les redondances présentées précédemment qui sont la redondance spatiale, temporelle et psycho-visuelle pour faire la compression. Dans un codeur vidéo hybride à compensation du mouvement et à transformée, la prédiction compensée du mouvement réduit en premier lieu les redondances temporelles. Un codage par transformée est ensuite appliqué à l'image de différence correspondante pour réduire les redondances spatiales [8]. Dans le cas des sources fortement corrélées, comme les images naturelles, la capacité de compression de la Transformée en Cosinus Discrète, ou TCD (Discrete Cosine Transform: DCT), est très proche de celle de la transformée optimale. La DCT est généralement suivie par

une quantification et un codage entropique permettant la compression des données ainsi obtenues.

Trois modes de codage sont possibles et conduisent à distinguer trois types d'image :

- Les images Intra, dites de type I : Les images Intra sont codées intégralement, sans aucune référence aux images voisines de la séquence vidéo. C'est la redondance spatiale qui est exploitée et éliminée à l'aide de la DCT. Chaque image est décomposée en blocs de 8x8 pixels et transformés en matrices de coefficients fréquentiels classés selon l'amplitude des signaux. Ces coefficients sont ensuite quantifiés et codés en utilisant un codage entropique comme montre la Figure 1.5. Les images I sont les plus coûteuses en débit mais servent de point de référence dans une séquence vidéo.
- Les images Inter prédites, dites de type P : Les images Prédicatives exploitent à la fois la redondance spatiale et la redondance temporelle des images d'une séquence vidéo. Elles sont codées à partir de l'image I ou P précédente à l'aide de vecteurs de mouvement obtenu par estimation de mouvement à l'encodeur. Les images sont découpées en blocs de 16 x 16 pixels. Les vecteurs de mouvement sont ensuite calculés en fonction du déplacement de chacun de ces blocs de pixels d'une image à la suivante puis codés avec un codage entropique.
- Les images Inter prédites bidirectionnelles, dites de type B : elles ne sont pas utilisées par tous les codeurs. Elles sont codées avec une estimation du mouvement par rapport à une image précédente et une image suivante. Les images de type I et P servent de référence aux images P et B. Le codage d'une image P et B est illustré par la Figure 1.6.

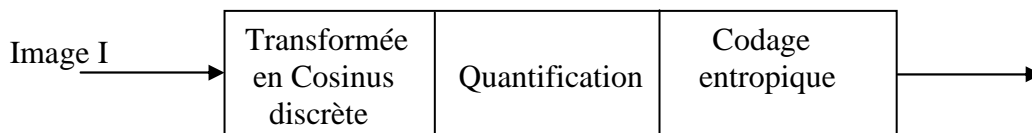


Figure 1.5 : Codage d'une image I

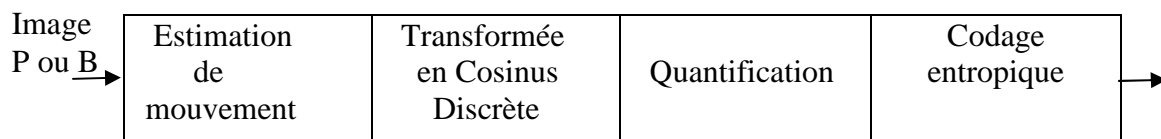


Figure 1.6 : Codage d'une image P ou B



Le codage d'une séquence vidéo consiste en une succession d'images Intra et Inter agencées de manière à satisfaire les contraintes de débit, de qualité et complexité Figure 1.7.

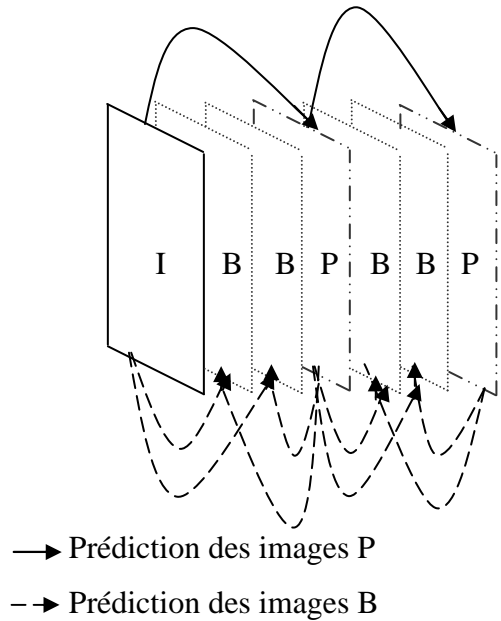
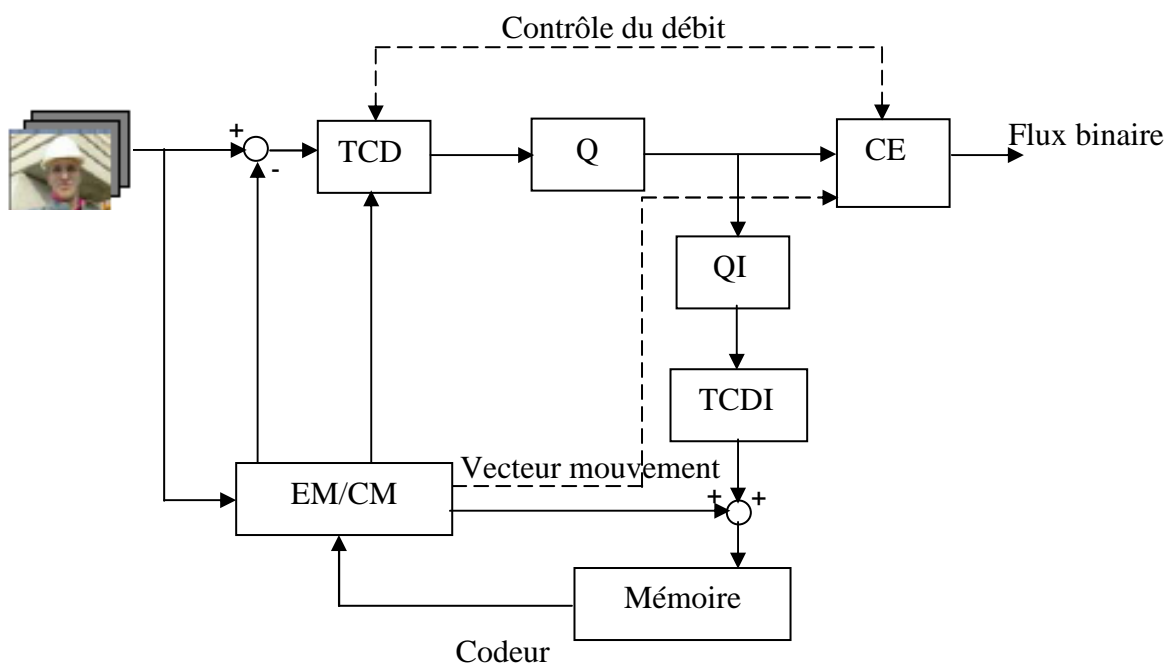


Figure 1.7 : Exemple de configuration de codage d'une séquence vidéo

Le diagramme fonctionnel d'un codeur/décodeur vidéo selon les principes exposés précédemment est présenté par la Figure 1.8.



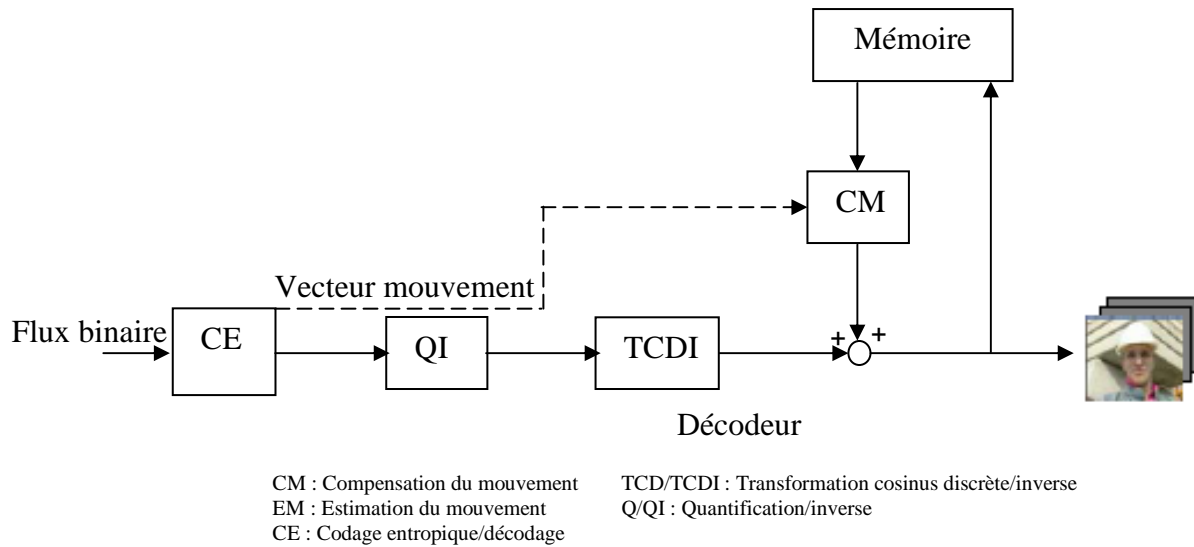


Figure 1.8 : Diagramme fonctionnel d'un codeur vidéo traditionnel

Au cours des dernières années, l'intérêt pour le multimédia a entraîné l'investissement d'une somme importante d'efforts de recherche sur le codage vidéo dans les universités et l'industrie, efforts qui se sont traduits par l'élaboration de plusieurs normes des deux organisations, UIT-T : H.261 [9], H.263 [10], H.264 [11] ou bien de l'ISO/CEI MPEG-1 [12], MPEG-2 [13], MPEG-4 [14]. Ces normes visent un vaste éventail d'applications aux exigences variées quant au débit binaire, à la qualité d'image et à la complexité ainsi qu'à l'amélioration des rapports de compression. La Figure 1.9 représente la classification des normes de codage vidéo.

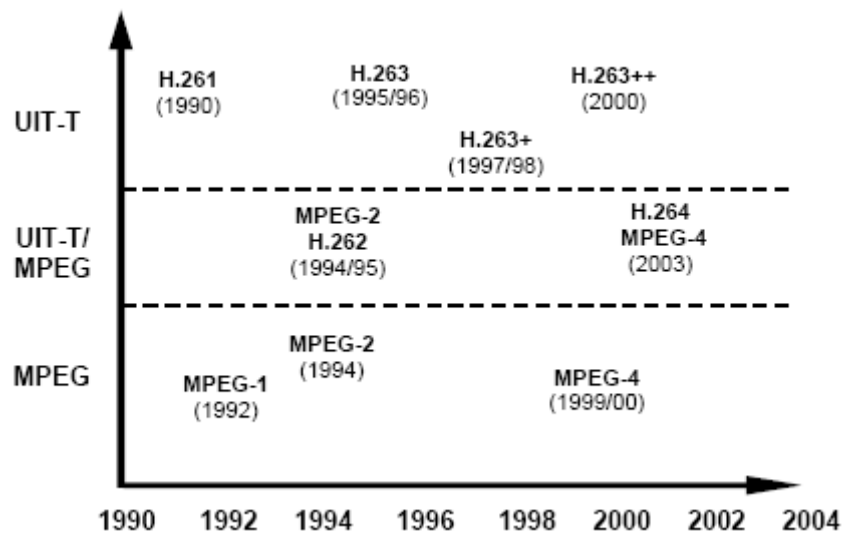


Figure 1.9 : Les normes de codage vidéo

## 5. Codage vidéo distribué

### 5.1 Cadre théorique

Le codage vidéo distribué (CVD) est un nouveau paradigme de codage vidéo basé sur deux théorèmes de la théorie de l'information : théorèmes de Slepian-wolf (1973) et de Wyner-Ziv (1976).

Bien que l'étude du codage distribué ait commencé aux années 1970, les efforts de développement des solutions pratiques du codage vidéo distribué sont plus récents. Le problème qu'émerge avec ce nouveau type de système de communication vidéo est le suivant: Comment obtenir une compression vidéo efficace et à basse complexité de l'encodeur, notamment quand le codage vidéo traditionnel ne fournit pas une solution acceptable ?

#### 5.1.1 Théorème de Slepian-wolf

Comme il a été vu dans la section 4.1, le codage et le décodage séparé de deux séquences,  $X$  et  $Y$  avec un débit  $R_x \geq H(X)$  et  $R_y \geq H(Y)$ , respectivement, permet une reconstruction sans erreurs au décodeur qui est référé au codage sans perte. Considérant maintenant la situation décrite sur la Figure 1.10 où  $X$  et  $Y$  proviennent de deux sources qui ne peuvent pas communiquer au niveau de l'encodeur, mais qui sont décodées conjointement (codage de source distribué).

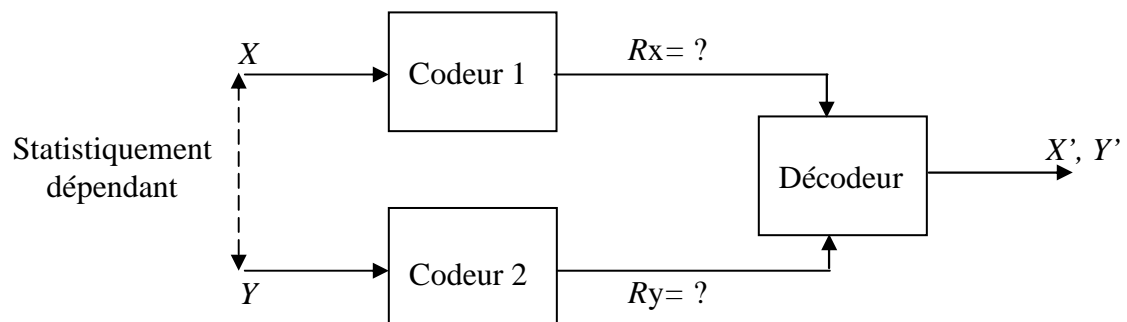


Figure 1.10 : Compression distribuée de deux séquences aléatoires discrètes statistiquement dépendantes

Le théorème de Slepian-Wolf [5] montre que le résultat de la compression est le même que lorsque la corrélation est exploitée par l'encodeur. Ce théorème a été étendu au cas des sources à valeurs continues Gaussiennes et codage avec pertes par

Wyner et Ziv [6].

Slepian-Wolf a montré aussi que :

Si les deux sources sont codées et décodées conjointement le débit minimal total qui peut être utilisé est  $R = H(X, Y)$ .

Si les deux sources sont codées et décodées indépendamment les débits seront  $R_x \geq H(X)$  et  $R_y \geq H(Y)$  et le débit total sera  $R = R_x + R_y \geq H(X, Y)$ .

Si maintenant les sources sont codées indépendamment mais décodées conjointement, les combinaisons possibles de  $R_x$  et  $R_y$  déterminés par Slepian et Wolf sont :

$$R_x \geq H(X/Y) \quad (1.9)$$

$$R_y \geq H(Y/X) \quad (1.10)$$

$$R_x + R_y \geq H(X, Y) \quad (1.11)$$

Où  $H(X/Y)$  est l'entropie conditionnelle de  $X$  si  $Y$  est connue et  $H(Y/X)$  est l'entropie conditionnelle de  $Y$  si  $X$  est connue.

Donc le décodage conjoint même avec un codage indépendant donne le même résultat (sans aucune perte d'efficacité) que le cas du codage et décodage conjoint des deux sources. La Figure 1.11 montre la région de débit pour laquelle la compression distribuée de deux sources statistiquement dépendantes  $X$  et  $Y$ , est recouverte avec une probabilité d'erreur arbitrairement petite selon le théorème de Slepian-Wolf. Sur cette Figure, les lignes verticales, horizontales et diagonales correspondent aux équations (1.9), (1.10) et (1.11), respectivement.

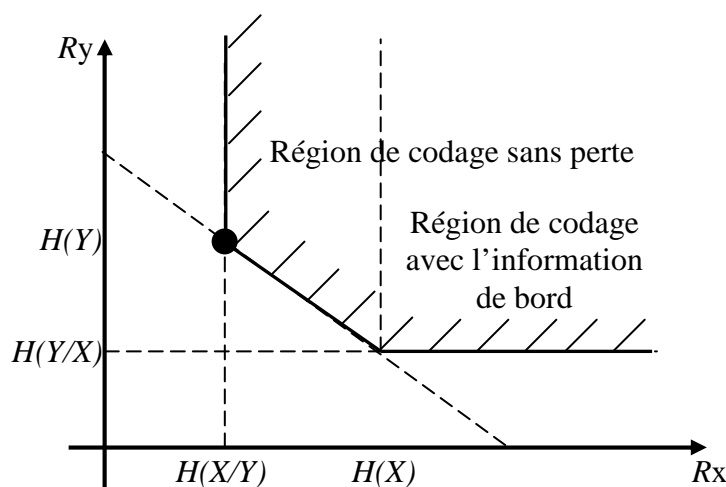


Figure 1.11 : Région du débit achevé selon le théorème de Slepian-Wolf [5]

Un dispositif intéressant de codage de Slepian-Wolf est le rapport qu'il a avec le codage canal. Ce rapport, déjà étudié en 1974 par Wyner [15].

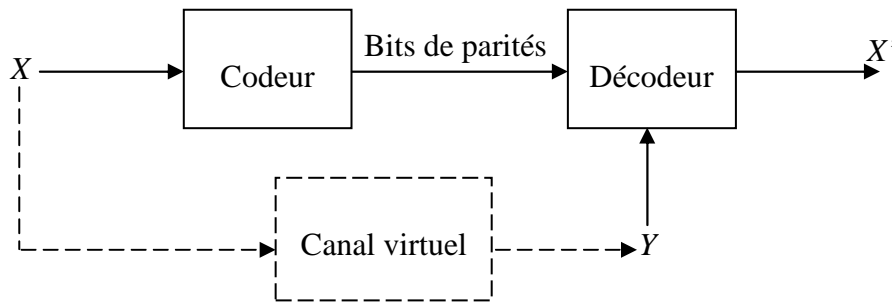


Figure 1.12 : Rapport entre le codage canal et le codage de Slepian-Wolf

Si nous considérons  $Y$  comme une version bruitée de  $X$ . Le problème peut ainsi être traité en utilisant des concepts et des outils de codage canal.

Les séquences  $X$  et  $Y$  sont donc codés indépendamment. Si  $Y$  est très corrélé à  $X$ , on peut considérer  $Y$  comme une version bruitée de  $X$  que l'on a transmis sur un canal virtuel, Ce canal virtuel n'existe pas pratiquement. Le concept de canal virtuel est utilisé pour modéliser la différence entre les deux séquences  $X$  et  $Y$  comme étant dues aux bruit de ce canal. Donc on peut coder les bits de parité de  $X$  et les utiliser pour corriger les erreurs qui se trouvent dans  $Y$ .

Donc, au lieu de transmettre  $X$  et les bits de parité (comme on le fait sur un canal réel), on transmet seulement les bits de parité et on utilise  $Y$  au niveau du décodeur pour trouver  $X$ .

### 5.1.2 Théorème de Wyner-Ziv

En 1976, A.Wyner et J. Ziv [6] ont étudié un cas particulier du théorème de Slepian-Wolf qui Correspond au point du débit  $(H(X/Y), H(Y))$  identifié sur la Figure 1.11 par le point noire. Ce cas étudie le codage d'une séquence  $X$  dont la séquence  $Y$  est connue au décodeur. La Figure 1.13 illustre un tel scénario ; dans la littérature, ce cas est connu par la compression avec perte et information de bord au décodeur.

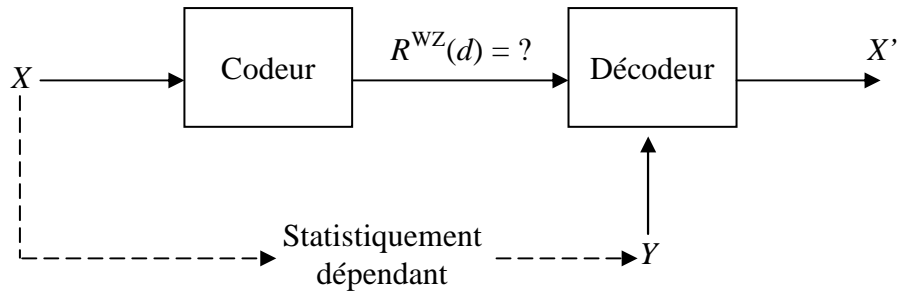


Figure 1.13 : Compression avec perte et avec information de bord au décodeur

Le concept de codage de Wyner-Ziv est également bien adapté au codage vidéo. Dans ce cas, certaines images de la séquence vidéo sont codées par un codage vidéo traditionnel (les normes MPEG-x ou H.26x), typiquement le mode du codage Intra. Les autres sont codées par la topologie de Wyner-Ziv (qui correspond à la séquence  $X$  dans La Figure 1.13). Le décodeur se sert des images codées par les normes traditionnelles pour générer une estimation de l'image courante (l'information de bord) par les techniques d'interpolation et d'extrapolation des images. Cette information de bord est ensuite employée dans le processus de décodage de l'image courante.

Si on considère la situation décrite sur la Figure 1.13, soit  $X$  et  $Y$  deux séquences aléatoires statistiquement dépendantes dont  $X$  est la séquence à coder, nommé par l'information de source, et  $Y$  l'information de bord qui est considérée disponible au décodeur. Le travail de Wyner et Ziv est concentré à établir le débit minimum  $R^{WZ}(d)$  nécessaire pour coder et reconstruire  $X$  avec une moyenne de distorsion au dessous de  $d$ . Les résultats obtenus par Wyner et Ziv ont indiqué que lorsque la dépendance statistique entre  $X$  et  $Y$  est exploitée seulement au décodeur, le débit de transmission augmente par rapport au cas où la corrélation est exploitée à l'encodeur et au décodeur, pour la même moyenne de distorsion  $d$  [6]. Mathématiquement, le théorème de Wyner et Ziv peut être décrit par :

$$R^{WZ}(d) \geq R_{X/Y}(d) \quad d \geq 0 \quad (1.12)$$

Où  $R^{WZ}(d)$  représente le débit du codage minimum de Wyner-Ziv (pour  $X$ ) et  $R_{X/Y}(d)$ , représente le débit nécessaire minimum pour coder  $X$  quand  $Y$  est simultanément disponible à l'encodeur et au décodeur, toujours pour la même moyenne de distorsion

*d.* Dans la littérature [16],  $R^{\text{WZ}}(d)$  et  $R_{X|Y}(d)$  s'appellent les fonctions du débit de distorsion. Notant que lorsque  $d = 0$ , c'est à dire quand il n'y a aucune distorsion, on trouve de nouveau le résultat de Slepian-Wolf, c'est à dire  $R^{\text{WZ}}(0) = R_{X|Y}(0)$ . Ceci signifie qu'il est possible (théoriquement) de reconstruire la séquence  $X$  avec une probabilité d'erreur arbitrairement petite même lorsque la corrélation entre  $X$  et l'information de bord  $Y$  est exploitée seulement au décodeur.

En conclusion, les théorèmes de Slepian-Wolf et de Wyner-Ziv bien connu dans la théorie de l'information, suggèrent qu'il est possible de compresser deux signaux statistiquement dépendants d'une manière distribuée (codage séparé, décodage conjoint) en utilise un débit semblable à celui employé dans un système où les signaux sont codés et décodés ensemble, c'est à dire comme dans le codage vidéo traditionnel.

Il faut bien noter que ces résultats sont uniquement théoriques et asymptotiques. Le travail théorique de Slepian-Wolf et de Wyner-Ziv (comme tous les résultats de la théorie de l'information) ne précise pas comment on peut pratiquement réaliser un tel système.

## 5.2 L'approche générale du codage vidéo de Wyner-Ziv

L'évolution de l'état de l'art des arrangements du CVD le long des ces dernières années a définie trois outils principaux pour approcher les limites suggérées par la théorie de l'information : Transformation, quantification et le codage canal.

Figure 1.14.

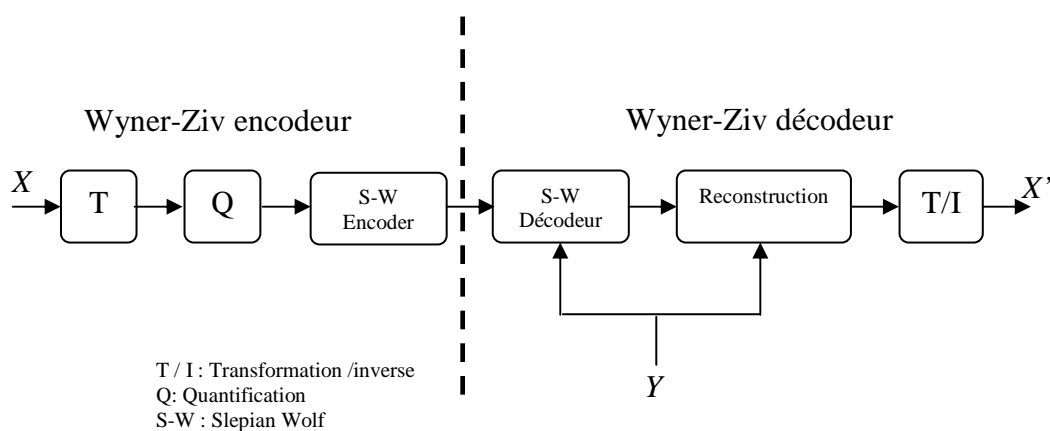


Figure 1.14 : Diagramme fonctionnel d'un codeur de Wyner-Ziv

Une transformation est appliquée d'abord aux échantillons de la séquence  $X$  pour permettre une représentation plus compacte des images. Les coefficients transformés sont ensuite quantifiés, afin d'exploiter les limitations perceptibles du système visuel humain (SVH) générant un flux de symbole quantifié. L'encodeur de Slepian-Wolf produit enfin l'information redondante (par exemple bits de parité) à partir de flux des symboles quantifiés.

Dans le décodeur de Wyner-Ziv, le flux des symboles quantifiés est décodé par un décodeur commun qui se sert de l'information de bord  $Y$ . le flux décodé et l'information de bord  $Y$  sont après utilisés ensemble dans un module de reconstruction pour estimer les coefficients transformés. En fin pour reconstruire la séquence  $X$  ;  $X'$ , une opération inverse de la transformation est effectuée.

Le principe de fonctionnement de chaque module de diagramme fonctionnel de la Figure 1.14, est présenté avec plus de détail dans la section suivante

### 5.2.1 Encodeur de Wyner Ziv :

- La première étape pour coder l'information de source dans l'architecture proposée est la transformation. Typiquement, la transformation est appliquée sur des blocs de  $n \times n$  échantillons afin de les décorréler. Le résultat de ce processus est la concentration de l'énergie des blocs dans quelques coefficients de grandes valeurs. Ces coefficients contiennent la majeure partie de l'information, ils s'appellent les coefficients de la basse fréquence. il est possible alors de réduire le débit binaire en transmettant seulement ces coefficients au décodeur.

Pradhan et Ramchandran [17] sont les premiers qui utilisent le codage à base d'ondelette dans un scénario du codage de Wyner-Ziv. Rebollo-Monedero et al ont noté dans [18] que la transformation cosinus discret (DCT) est un choix asymptotiquement optimal pour le module de la transformation, dans le terme des performances du taux de distorsion.

- La quantification est une technique de codage de la source, employée pour compresser une gamme des valeurs dans une simple valeur.

Flynn et Gray [19] ont considéré le problème de la quantification dans un système constitué par deux sondes séparées observant une cible commune. Dans ce contexte, différents algorithmes de la quantification ont été examinés afin



d'obtenir une bonne performance globale. Shamai, Verdú et Zamir ont présenté des quantificateurs conçus pour la compression avec perte et avec l'information de bord connu au décodeur [20]. Servetto [21] a contribué à la construction d'une famille des quantificateurs afin d'atteindre le débit ciblé  $H(X/Y)$ . Fleming et Effros [22] ont considéré la quantification des vecteurs dans l'étude de l'optimisation du taux de distorsion des vecteurs quantifiés pour le codage sur réseau avec un débit fixe. Muresan et Effros [23] ont démontré le rapport entre le scalaire de la quantification et l'histogramme de la segmentation afin de développer un algorithme pour la conception d'un quantificateur optimale. Rebollo-Monedero, Zhang et Girod [24] ont étudié la conception optimale des quantificateurs considérant le scénario du codage de Wyner-Ziv pour des sources gaussiennes et des intervalles disjoints de la quantification.

- L'encodeur de Slepian-Wolf est employé pour réduire le débit de transmission de l'information de la source. Le processus de l'encodeur de Slepian-Wolf est effectué par des techniques de codage canal.

Pradhan et Ramchandran [25] ont fait les premières étapes vers un arrangement pratique pour le codage de source distribué utilisant les syndromes. Leur solution est connue par DISCUS (DIstributed Source Coding Using Syndromes). Les résultats prouvent que les performances du codage distribués ont bénéficié de l'utilisation des techniques du codage canal. Plus tard, Wang et Orchard [26] ont atteint les meilleurs résultats en termes du rapport SNR en employant un code de treillis. Le turbo code a été après examinés par García-Frías et Zhao [27], [28], et Bajcsy et Mitran [29], [30] sur des séquences aléatoires binaires. Aaron et Girod [31] ont également proposé un système basé sur un turbo code et l'ont examiné sur des séquences aléatoires binaires et gaussiennes. Plus récemment, les nouveaux codeurs de Slepian-Wolf basés sur le turbo code ont été conçus par Liveris et al. [32] et Stankovic, Liveris, Xiong et Georgiades [33] aussi bien que des codeurs basés sur le code LDPC par Stankovic et al. [33], Schonberg, Pradhan et Ramchandran [34], et Coleman, Lee, Medard et Effros [35]. D'une façon générale, les résultats obtenus montrent que les performances sont plus près des limites du Slepian-Wolf et de Wyner-Ziv.

### 5.2.2 Décodeur de Wyner- Ziv :

Le décodeur de Wyner-Ziv effectue des opérations semblables à ceux réalisées à l'encodeur, bien que dans un ordre inverse.

- La première opération effectuée par le décodeur de Wyner-Ziv est le processus de décodage de Slepian-Wolf. Dans ce cas, le décodeur utilise le flux reçu et l'information de bord afin d'estimer les symboles quantifiés qui sortent du module de la quantification.
- Une fois que les symboles quantifiés sont décodés, la deuxième étape exécutée au décodeur de Wyner-Ziv est la reconstruction. Dans cette étape, une estimation des coefficients transformés de l'information de source est obtenue en utilisant les symboles quantifiés décodés ainsi que l'information de bord  $Y$ .

En 2002, Aaron, Zhang et Girod [36] ont considéré la fonction de reconstruction donnée par l'espérance conditionnelle donnée par (1.13) pour reconstruire les valeurs de l'information  $X$ .

$$E(X / q', Y) \tag{1.13}$$

Ils ont supposé une distribution Laplacian pour modéliser le résiduel entre les éléments correspondants de  $X$  et  $Y$ ; dans (1.13),  $q'$  représente le flux des symboles quantifiés décodés.

- En fin pour reconstruire l'information de la source, la transformation inverse opère sur les coefficients transformés estimés dans le module de reconstruction.

A titre d'exemple on va décrire brièvement quelques approches qui sont disponibles dans la littérature pour le codage vidéo distribué. Notant que ces approches comprennent presque tous les modules mentionnés précédemment (transformation, quantification et codage canal) selon le but de chaque approche.

Une première approche connue sous le nom de PRISM (Power-efficient, Robust, high-compression, Syndrome-based Multimedia Coding) a été proposée par Puri et Ram-chandran [37] pour la transmission de multimédia sur les réseaux sans fil en utilisant des syndromes. Le but principal de cette approche est de combiner la faible complexité et la robustesse du codage des images en mode Intra avec l'efficacité de compression des images en mode Inter.

En 2002, se servant des turbos codes, Aaron, Zhang et Girod [36] ont présenté les résultats d'un codeur vidéo utilisant un schéma de codage en mode Intra et de

décodage en mode Inter où différentes images sont indépendamment codées mais sont conjointement décodées. En 2003, Zhu, Aaron et Girod [38] ont proposé une approche du codage vidéo distribué à basse complexité sous le nom de "distributed compression for large cameras arrays". Dans cette solution, des vues multiples et corrélées d'une scène sont indépendamment codées avec un encodeur de Wyner-Ziv dans le domaine pixel mais sont conjointement décodées à un nœud central. Dans [38] les auteurs ont effectué une comparaison de performances entre le codeur de Wyner-Ziv dans le domaine pixel et un système de codage utilisant la norme JPEG-2000. Les résultats démontrent que pour des débits faibles, la solution présentée dans [38] atteint des valeurs de PSNR meilleures que JPEG-2000 avec une complexité de codage inférieure.

En 2004, Aaron, Rane, Setton et Girod [16] ont proposé une architecture semblable à celle de [36] en utilisant la transformée en cosinus discrète (DCT : Discrete Cosine Transform). Les résultats obtenus avec la nouvelle solution de codage sont meilleurs en terme d'efficacité de codage par rapport à ceux de [36] (au coût d'une complexité légèrement élevée au codage associé à la transformée DCT).

Une autre solution de codage vidéo distribué avec une faible complexité à l'encodeur a été proposée par Aaron, Rane et Girod [39]. Dans cette solution, mis à part le flux résultant du processus de codage des images, l'encodeur transmet également des informations supplémentaires sur l'image courante pour aider le décodeur lors de la phase d'estimation de mouvement.

En 2004, Rane, Aaron et Girod [40] ont présenté une autre approche visant la robustesse de la transmission des images vidéo. Spécifiquement, le but de cette approche est de rendre le flux obtenu après le codage séparé plus résistant aux erreurs de transmission d'un canal bruité. Dans ce cas, les mêmes images sont codées à la fois avec un codeur vidéo distribué et un codeur traditionnel (codeur MPEG dans [40]). Les bits générés avec les deux encodeurs passent dans un canal bruité. Le décodage s'effectue d'une manière conjointe pour reconstruire les images codées. Cette approche peut être vue comme une application de codage systématique source-canal.

En conclusion, les performances du taux de distorsion associé à l'architecture de Wyner-Ziv proposée sur la Figure 1.14 dépendent des techniques employées pour mettre en application chaque module de l'architecture. En pratique, il y a deux principaux secteurs d'intérêt au codage vidéo distribué, la faible complexité de l'encodeur (comme dans [16] et [39]), et la robustesse de la transmission (comme dans

[3], [25] et [40]). En plus, les codeurs vidéo distribués sont utilisés dans deux domaines différents : domaine pixel ([36], [41], [42] et [43]) et domaine transformé ([3], [16], [39], [40] et [44]).

Dans la littérature, il y a essentiellement deux groupes de recherche qui ont de nos jours responsables du principal développement des systèmes du codage vidéo distribué : le groupe de Bernd Girod (université de Stanford) et le groupe de Kannan Ramchandran à Berkeley (université de la Californie). Donc, la section suivante est conçue pour représenter l'approche suivie par chaque groupe.

## **5.3 Les solutions du CVD les plus intéressants**

### **5.3.1 Approche suivie par l'équipe de Stanford**

Dans le but de répondre aux exigences d'efficacité du codage et de la faible complexité, le groupe de Stanford a présenté plusieurs solutions du codage, par exemple [4], [16], [36] et [45], qui sont basées sur les théorèmes de Slepian-Wolf et de Wyner-Ziv :

#### **5.3.1.1 Codage**

Dans [36], une séquence vidéo est divisée en deux groupes d'images : les images de Wyner-Ziv (les images paires de la séquence vidéo) et les images clés (les images impaires de la séquence vidéo) ; chaque image de Wyner-Ziv est codé Pixel par Pixel d'une façon indépendante des images clés et des autres paires images. Pour décoder l'image de Wyner-Ziv, l'information de bord (une estimation de l'image de Wyner-Ziv) est produite par les techniques d'interpolation.

Une autre approche a été présentée dans [4] où le nombre des images de Wyner-Ziv par rapport aux images clés peut changer ; les images clés sont codées en mode Intra avec la norme H.263+ traditionnelle et les images de Wyner-Ziv sont codées comme dans [36]. En utilisant les images précédemment reconstruites (Wyner-Ziv et clés), les techniques d'interpolation ou d'extrapolation sont utilisées pour générer l'information de bord.

Dans [16], une architecture semblable à celle dans [4] est présentée. La différence principale est que dans [16] la transformation du codage est considérée dans le codage de Wyner-Ziv.

La Figure 1.15 représente l'architecture de codeur vidéo utilisée dans [4]

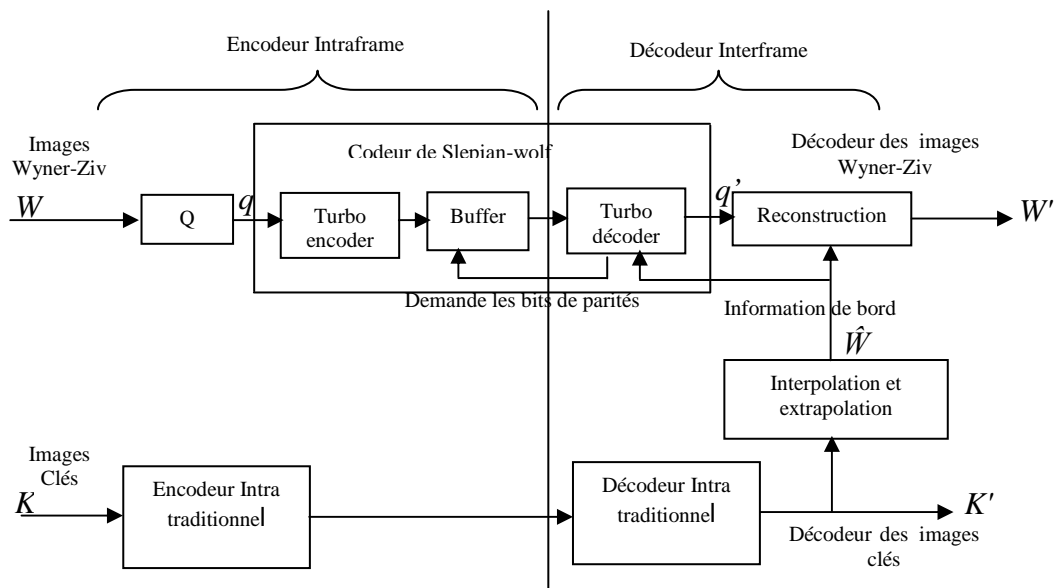


Figure 1.15 : Architecture du codeur vidéo Wyner-Ziv du groupe de Stanford [4]

Un premier sous-ensemble d'images impaires, régulièrement espacé dans la séquence, va être codé en mode Intra et constitué des images clés (Key Frames) désignées par  $K$ . Ces images sont codées et décodées en utilisant un codeur en mode Intra traditionnel. Le deuxième sous-ensemble d'images (l'ensemble des images paires qui se trouvent entre les Key Frames) est codé en utilisant le codeur de Wyner-Ziv. Ces images, désignées par images  $W$  (images Wyner-Ziv) sont codées en mode Intra et décodées en mode Inter.

Les pixels de chaque image Wyner-Ziv,  $W$ , sont quantifiés avec un quantificateur scalaire uniforme à  $2^M$  niveaux. Les symboles quantifiés, sont codés avec un turbo codeur poinçonné. Seuls les bits de parité sont stockés dans une mémoire tampon. Le nombre de bits de parité transmis au décodeur dépend de la qualité de l'information de bord  $\hat{W}$ .

### 5.3.1.2 Décodage

Les images  $K$  sont décodées avec un décodeur en mode Intra traditionnel. Pour chaque image Wyner-Ziv  $W$  codée, le décodeur produit une information de bord, image  $\hat{W}$ , par interpolation moyennée à partir des images clés adjacentes décodées [4].

Dans [36], la première technique d'interpolation utilisée était une simple moyenne des pixels des images paires  $K$  se trouvant dans la même position que le pixel de l'image  $W$ . Pour utiliser l'information de bord  $\hat{W}$  au turbo décodeur, le modèle de corrélation considéré entre les pixels de l'image  $W$  et leurs correspondants des images  $\hat{W}$  (l'information de bord) est une distribution Laplacienne. L'estimation des paramètres de la distribution Laplacienne est déterminée en observant les statistiques des images précédemment décodées.

Le turbo décodeur utilise l'information de bord  $\hat{W}$  et les bits de parité reçus pour récupérer les symboles quantifiés. Si le décodeur n'arrive pas à décoder les symboles originaux, alors une demande de bits de parité additionnels de la mémoire tampon du codeur de Wyner-Ziv sera effectuée via le canal de retour (feed-back). Ainsi, le canal de retour est nécessaire pour adapter le débit de transmission aux statistiques de dépendance entre l'information de bord et l'image à coder. Les processus de décodage et de demande de bits de parité sont répétés jusqu'à ce qu'une probabilité d'erreur symbole acceptable soit atteinte.

Le débit binaire d'une image Wyner-Ziv est déterminé en fonction de la corrélation entre celle-ci et l'information de bord  $\hat{W}$  correspondante. Puisque l'information de bord est disponible seulement au décodage, la décision sur le nombre des bits de parité à transmettre ne peut être prise que par le turbo décodeur et non par l'encodeur. Ainsi, le mécanisme de contrôle de débit est réalisé au niveau du décodeur. L'avantage du contrôle de débit au décodeur est de réduire au minimum le fardeau de cette tâche à l'encodeur. Après le décodage turbo (les pixels quantifiés sont estimés), chaque pixel de l'image  $W$  est reconstruit selon les valeurs du pixel de l'information de bord  $\hat{W}$  et le symbole décodé (l'équation (1,13)).

### 5.3.1.3 Performances

Les performances débit-distorsion du codeur vidéo de Wyner-Ziv dans le domaine pixel de la solution de Stanford se situent entre celles des codeurs vidéo traditionnels en modes Inter et Intra. Dans [45], au même débit et pour les séquences Hall Monitor et Salesman, la solution proposée présente un gain de performances en terme de PSNR de 2 à 5 dB par rapport au codeur vidéo traditionnel en mode Intra. Cependant, les performances du codeur H.263+ en mode Inter restent meilleures que celles de la solution proposée.

### 5.3.2 Approche suivie par l'équipe de Berkeley

Cette section présentera une architecture du codage vidéo dont le but principal est d'augmenter la robustesse du codage Interframe à la propagation des erreurs de transmission. Cette approche a été présentée par le groupe du Berkeley [3] sous le nom de Power-efficient, Robust, High-compression, Syndrome-based Multimedia coding ou PRISM, comme elle est habituellement référencée dans la littérature.

Cependant, l'approche de PRISM utilise le concept d'information de bord d'une façon différente de la description donnée à la section 5.1.2. L'innovation de la solution PRISM est d'avoir plusieurs candidats d'information de bord de telle façon que lors de la procédure de décodage, le meilleur élément est sélectionné pour la reconstruction de l'image courante. Ainsi, l'approche présentée par le groupe de Berkeley ne demande pas l'utilisation de la voie de retour pour améliorer la reconstruction des images codées.

Nous allons brièvement présenter l'architecture de l'encodeur et du décodeur de la solution PRISM pour le codage vidéo distribué. Plus de détails se trouvent dans [3].

La Figure 1.16 et la Figure 1.17 illustrent les architectures d'un encodeur et d'un décodeur PRISM, respectivement.

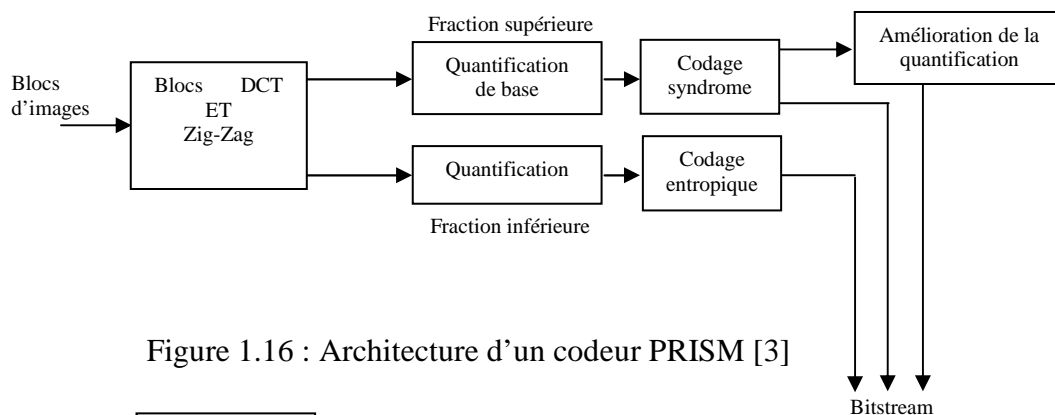


Figure 1.16 : Architecture d'un codeur PRISM [3]

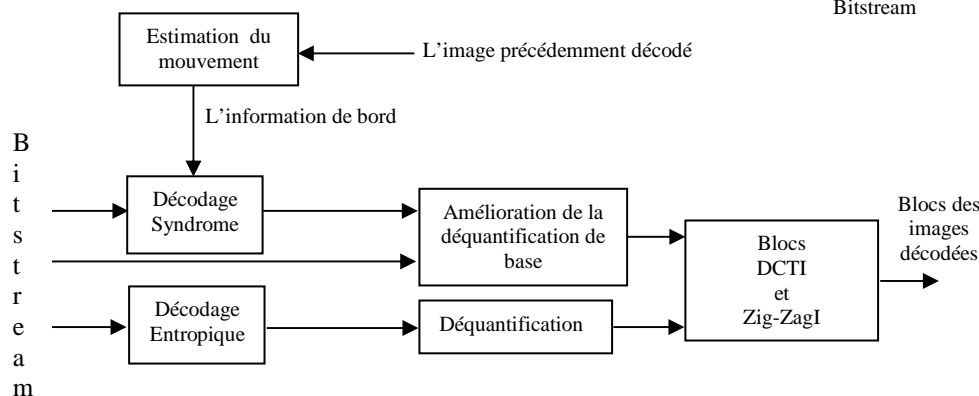


Figure 1.17 : Architecture d'un décodeur PRISM [3]

### 5.3.2.1 Codage

Considérons l'architecture de codage du système PRISM illustrée sur la Figure 1.16. Chaque image de la séquence vidéo est d'abord divisée en blocs de pixels. La taille des blocs est  $8 \times 8$  ou  $16 \times 16$ . Avant de présenter les différentes opérations effectuées par l'encodeur de PRISM pour coder une image, une étape appelée classification est réalisée.

#### Classification

Dans l'étape de classification, chaque bloc  $8 \times 8$  ou  $16 \times 16$  est classifié dans une des classes prédéfinies. Cette classification est réalisée selon la corrélation (dépendance statistique ou bien bruit de corrélation) entre chaque bloc d'image courante et son correspondant (même position) dans l'image précédente. Dans [3], la valeur de la variance du bruit de corrélation est déterminée par l'erreur quadratique entre chaque bloc d'image courante et son correspondant dans l'image précédente. L'étape de classification permet de décider quel codage est adapté pour chaque bloc de l'image courante : aucun codage (classe aucun codage), codage source traditionnel (classe de codage en mode Intra) ou codage par syndrome (classe de codage de Wyner-Ziv).

Ainsi, les blocs d'images classifiés dans la classe aucun codage (corrélacion temporelle grande) ne sont ni codés ni transmis. Les blocs classifiés dans la classe de codage en mode Intra (corrélacion très faible) sont traditionnellement codés. Par contre, les blocs classifiés dans la classe de codage par syndrome sont codés avec un codeur de Wyner-Ziv basé sur la technique DISCUS. Une attention particulière sera donnée aux procédures du codage/décodage de ces blocs. Les classes de mode de codage choisies pour les blocs d'une image courante sont alors transmises au décodeur comme information d'en-tête.

#### Dé-corrélacion spatiale par transformée DCT

Une transformée DCT est appliquée à chaque bloc de l'image courante. Les coefficients de la transformée de chaque bloc sont par la suite scrutés en zig-zag.

Pour les blocs classifiés dans la classe de codage par syndrome, la solution PRISM, comme l'indique la Figure 1.16, code les coefficients de basse fréquence en utilisant le codeur de Wyner-Ziv, tandis que ceux de haute fréquence sont codés classiquement (c'est-à-dire quantifiés puis codés avec un codeur entropique). Typiquement, plusieurs



coefficients de haute fréquence ont des valeurs proches de zéro et donc le codage entropique va utiliser peu de bits pour les compresser. Par contre, les coefficients de basse fréquence ont des valeurs élevées et le codage par syndrome permettra de réduire le débit binaire requis pour les transmettre (seul le syndrome est transmis au décodeur comme dans DISCUS).

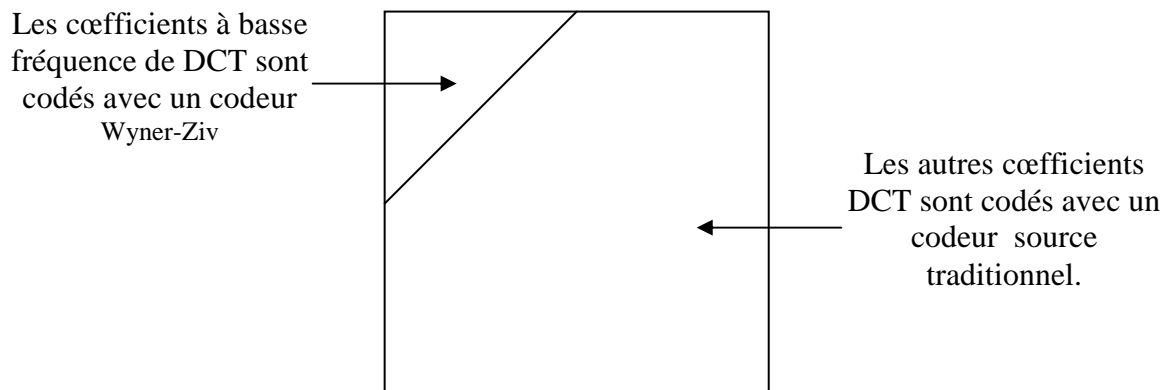


Figure 1.18 : Sélection des coefficients DCT [3]

### **Quantification**

Les coefficients de la transformée, qui sont scrutés en zig-zag, sont quantifiés. Le coefficient DC (le coefficient de la transformée avec la plus basse fréquence) et un nombre restreint de coefficients AC (présentant la haute fréquence) sont quantifiés en utilisant le quantificateur scalaire adapté. Le choix des pas de quantification de ces coefficients est dépendant du niveau de corrélation de chaque bloc 8 x 8 ou 16 x 16 de l'image courante déterminé dans l'étape de classification.

### **Codage par syndrome**

La compression des coefficients de basse fréquence (environ 20 % du nombre total des coefficients) quantifiés avec un quantificateur adapté est réalisée grâce au codage par syndrome. L'approche PRISM utilise le principe du codeur DISCUS. Un treillis à 128 états d'un codeur traditionnel de taux 1/2 est utilisé et seulement le syndrome obtenu à la fin du codage est transmis au décodeur. Les bits de syndrome sont incorporés dans le flux binaire comme l'indique la Figure 1.19.

Bits de syndrome	Bits de CRC	Bits d'amélioration	Bits du codeur de source
------------------	-------------	---------------------	--------------------------

Figure 1.19 : Contenu de flux [3]

### **Amélioration de la quantification**

La compression d'un signal avec différents pas de quantification permet d'atteindre une meilleure qualité de reconstruction (distorsion désirée) au décodeur. Comme il a été mentionné ci-dessus, pour les coefficients de basse fréquence qui sont codés par syndrome, le choix du pas de quantification est limité par la corrélation déterminée à l'étape de classification afin de réduire au minimum la probabilité d'erreur de décodage. Pour atteindre une qualité souhaitable de reconstruction, le pas de quantification du quantificateur adapté est raffiné (amélioration de la quantification). Le processus d'amélioration de quantification est juste une subdivision progressive de l'intervalle de base du quantificateur, en des intervalles de taille égale correspondants au pas de quantification cible. L'indice associé à l'intervalle amélioré (ou raffiné), faisant partie de l'intervalle de base, est transmis au décodeur. Les bits d'amélioration (indice associé aux intervalles raffinés) forment une autre composante du flux de la Figure 1.19.

### **Codage entropique**

Les coefficients de haute fréquence quantifiés avec un quantificateur scalaire sont traditionnellement codés en utilisant un codeur à longueur variable (CLV) : codeur de Huffman. Les bits générés, appelés bits du codeur de source, sont incorporés dans le flux comme l'indique la Figure 1.19.

### **Code CRC**

Ici et contrairement au codage vidéo traditionnel, c'est le décodeur qui effectue l'estimation de mouvement. Cependant, pour chaque bloc de l'image courante codé, le décodeur ne connaît pas la meilleure information de bord générée par estimation de mouvement de l'image précédente. Pour remédier à ce problème et améliorer la tâche de l'estimation de mouvement, l'encodeur transmet non seulement le syndrome mais également un contrôle par redondance cyclique (CRC de force suffisante) de la séquence des mots de code obtenus après quantification. Ce contrôle sert comme une

signature de la séquence des mots de code. Les bits de contrôle de CRC sont introduits dans le flux binaire (Figure 1.19) sous le nom de bits de CRC.

Pour résumer, la complexité du codeur dépend essentiellement des opérations de la transformée et de codage entropique [3]. Par conséquent, cette complexité est semblable à celle d'une solution traditionnelle de codage en mode Intra.

### **5.3.2.2 Décodage**

L'architecture du décodeur de PRISM est présentée à la Figure 1.17. Pour la classe aucun codage, les blocs reconstruits sont égaux à ceux ayant la même position dans l'image précédente. Pour les blocs classifiés dans la classe de codage en mode Intra, le décodage traditionnel à savoir décodage entropique et déquantification (opérations inverses à celles effectuées à l'encodeur) est appliqué. Quant aux blocs classifiés de la classe de codage de syndrome, le procédé de décodage est décrit dans les cinq étapes suivantes :

#### **Estimation de mouvement**

La tâche la plus importante exécutée au décodeur de PRISM est l'estimation de mouvement qui fournit les informations de bord nécessaires pour décoder efficacement le mot de code associé au syndrome reçu. Les informations de bord (le meilleur bloc parmi tous ses voisins) sont obtenues à partir de l'estimation de mouvement à  $\frac{1}{2}$  pixels de l'image reconstruite précédemment.

#### **Décodage par syndrome**

Chaque information de bord générée par l'opération d'estimation de mouvement est utilisée en association avec les bits de syndrome reçus pour décoder les mots de code correspondant à la version quantifiée des coefficients à basse fréquence. Comme dans DISCUS, vu qu'un codeur traditionnel (avec un treillis à 128 états) est utilisé lors du codage, le décodage peut être réalisé en utilisant l'algorithme de Viterbi [46], [47]. Cet algorithme est alors employé pour identifier la séquence des mots de code la plus proche de l'information de bord utilisée. Un contrôle de détection d'erreur est effectué sur la séquence décodée en utilisant le code CRC. Si le CRC est valide, alors le décodage est déclaré bien réussi. Autrement, une autre information de bord est utilisée et le processus entier de décodage par syndrome est répété. Le CRC est employé

comme une signature fiable et unique pour chaque bloc et peut identifier facilement la meilleure information de bord. Dans PRISM, le décodeur n'a pas besoin d'une voie de retour pour améliorer les performances de décodage. Cependant, un décodage multi-hypothèse qui entraîne l'utilisation de décodage canal multiple est utilisé.

### **Amélioration de la déquantification**

Après l'estimation de la séquence des mots de code, la reconstruction des coefficients de basse fréquence est effectuée en utilisant un déquantificateur. Afin d'avoir une meilleure qualité de reconstruction (faible distorsion), la déquantification utilise la séquence des mots de code estimée et les bits d'amélioration transmis par l'encodeur. À cette étape, un algorithme d'estimation linéaire est exécuté pour retrouver les coefficients de Wyner-Ziv estimés.

### **Décodage entropique et déquantification associée**

Les bits reçus du codeur de source (correspondants au codage en mode Intra des versions quantifiées des coefficients à haute fréquence) sont décodés en utilisant des opérations inverses à celles effectuées à l'encodeur à savoir un décodage entropique et une reconstruction utilisant un déquantificateur.

### **Transformée DCT inverse**

Après le balayage en zig-zag inverse des coefficients estimés, la transformée DCT inverse est alors appliquée pour retrouver les pixels des blocs reconstruits. Ainsi, le processus de décodage de PRISM est terminé.

## **5.3.2.3 Performances**

Afin d'évaluer les performances du système PRISM, les auteurs dans [3] ont codé les 15 premières images des séquences Mother and Daughter (352 x 288), Carphone (176 x 144) et football (352 x 240). La première image de chaque séquence vidéo est codée en mode Intra. Les performances de PRISM ont été comparées à celles d'un système de codage vidéo H.263+ (codage en mode Intra et Inter). Pour les trois séquences, les performances de PRISM sont situées entre celles du codeur de H.263+ avec un codage en modes Inter et Intra. Pour un même débit, le gain en performances de PRISM par rapport au codeur de H.263+ avec un codage en mode Intra est de 1.7, 2.5

et plus de 8 dB, respectivement pour les séquences football, Carphone et Mother and Daughter.

Les auteurs dans [3] ont testé la robustesse de la solution PRISM proposée. Ils ont montré l'impact de la perte d'une image de la séquence football lors du codage avec les systèmes PRISM et H.263+. La robustesse de PRISM est beaucoup plus élevée que celle de H.263+. En effet, étant donné qu'il n'y a pas de propagation d'erreur entre les images (due à l'absence de la prédiction à l'encodeur), la perte d'une image en utilisant PRISM a un effet négligeable sur la qualité de la séquence vidéo décodée.

## Conclusion

Les applications naissantes qui nécessitent des conditions du codage tout à fait différentes à ceux visées par les standards MPEG-x et H.26x, telles que la consommation d'une faible puissance à l'encodeur, ont stimulé le développement d'un nouveau paradigme du codage capable de satisfaire ces dernières exigences. Ce nouveau paradigme du codage est établi sur un codage de source distribué qui est théoriquement étudié dans les années 1970 par Slepian, Wolf, Wyner et Ziv.

Une grande partie du travail qui a été effectué dans le codage vidéo distribué se rapporte au codage de Wyner-Ziv qui est un cas particulier du codage vidéo distribué. Jusqu'ici, les résultats obtenus prouvent que le codage vidéo de Wyner-Ziv peut fournir des solutions du codage intéressantes pour quelques applications.

L'état de l'art du codage vidéo présenté dans ce chapitre sert comme point de départ dans le chapitre 2 où il décrit l'implémentation d'un codeur pratique du pixel domaines.

# Chapitre 2

## Codage vidéo de Wyner-Ziv avec turbo code

### Introduction

Dans le chapitre précédent, différents schémas de mise en œuvre du codage vidéo distribué permettant d'approcher les limites théoriques ont été décrites. Dans ce chapitre, nous allons présenter notre contribution au codage vidéo distribuée, qui consiste d'implémenter un turbo code poinçonner pour l'exploiter dans un codeur vidéo distribué. En premier lieu, pour tester l'efficacité du turbo code, on va considérer dans l'implémentation du turbo décodeur que le canal de transmission est gaussien, alors que, pour adapter le canal de transmission dans le schéma du codage vidéo distribué, le canal de transmission est considéré Laplacienne.

### 1. Implémentation d'un turbo code poinçonné

Les codes correcteurs d'erreurs (codage de canal) sont des solutions permettant d'améliorer la qualité des communications numériques. Le principe du codage canal est d'introduire de la redondance dans la séquence d'information binaire dans le but de corriger les erreurs de transmissions durant la réception de l'information. Les turbos codes sont une famille des codes correcteurs d'erreurs construits par concaténation de codes élémentaires [48]. Les turbos codes sont le résultat de deux innovations majeures : la concaténation de deux codes pour le codage et le décodage itératif. Actuellement, ces codes sont considérés comme étant les plus efficaces pour le codage avec des gains de 2 à 4 dB par rapport aux codes correcteurs d'erreurs classiques. Le turbo code découvert par Berrou et al. [49] est constitué de deux encodeurs récursifs systématiques convolutionnels (RSC) concaténés en parallèle et séparés par un entrelaceur. Le turbo décodeur est composé de deux décodeurs SISO (Soft-In Soft-Out) concaténés en série via un entrelaceur/ déentrelaceur. Le décodage est basé sur l'échange d'information extrinsèque d'un décodeur à l'autre jusqu'à

l'obtention des performances désirées. Chaque décodeur calcule les probabilités à posteriori (A Posteriori Probability: APP) pour estimer le bit d'information. L'algorithme de décodage utilisé dans chaque SISO pour calculer les APP est celui de Bahl et al. [50] appelé algorithme BCJR (Bahl, Cocke, Jelinek, et Raviv) ou bien MAP (Maximum A Posteriori). Par la suite en va décrire en détails les différents modules de l'encodeur/décodeur turbo qu'on a implémenté.

## 1.1 Turbo encodeur

La Figure 2.1 illustre une structure d'un turbo encodeur utilisant une concaténation parallèle de deux encodeurs RSC identiques d'un taux de  $\frac{1}{2}$ . Entre les deux encodeurs RSCs, un entrelaceur aléatoire de  $L$  bits est employé pour décorréliser les  $L$  bits de la séquence d'entrée ( $X$ ). Donc, la séquence d'entrée est codée dans son ordre naturel et dans son ordre permuté, par les deux encodeurs RSC<sub>*i*</sub> ( $i=1,2$ ). Les symboles  $S_i$  et  $P_i$  représentent les séquences produites par l'encodeur RSC<sub>*i*</sub>, qui correspondent à la séquence des bits systématiques et de parités respectivement. Pour obtenir des rendements plus élevés, un poinçonnage des séquences de redondance  $P_i$  est effectué, le processus de poinçonnage consiste à diviser les séquences des bits de parités  $P_i$  à des petits blocs, ensuite les envoyer au turbo décodeur bloc par bloc. La transmission des blocs s'arrête lorsque certains critères de la convergence soient satisfaits.

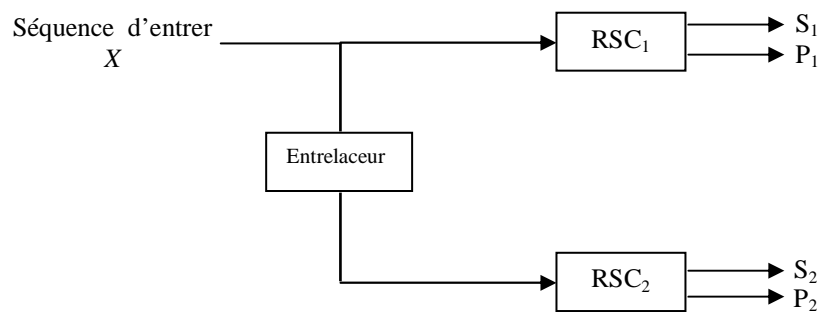


Figure 2.1 : Structure d'un turbo encodeur

### 1.1.1 Encodeurs RSC

Nous avons utilisé des encodeurs convolutifs récurrents systématique. Explicitons ces différents termes :

- Convolutif : la décision est prise en fonction des probabilités de chacun des bits.
- Récurrents : on exécute plusieurs fois le même algorithme jusqu'à ce que l'on

obtienne des probabilités satisfaisantes.

- Systématique : le mot d'information initial est récupéré directement dans le mot de code reçu (Si).

Un encodeur RSC est typiquement caractérisé par une matrice génératrice  $G$  qui permet d'obtenir la sortie de l'encodeur RSC pour une entrée donnée RSCin.

Généralement, la sortie de l'encodeur RSC, RSCout est donné par (2.1).

$$RSCout = RSCin.G \quad (2.1)$$

Dans notre implémentation (Figure 2.1), deux encodeurs RSC d'un taux de  $\frac{1}{2}$  ont été employés. Avec une matrice génératrice polynomial donné par :

$$\begin{bmatrix} 1 & g2(D) \\ & g1(D) \end{bmatrix} \quad (2.2)$$

Où  $g1(D)$  et  $g2(D)$  sont deux polynômes dont  $D$  dénote le retard,  $g1(D)$  et  $g2(D)$  sont exprimés par l'équation (2.3), pour  $i=1$  et  $i=2$ , respectivement.

$$g_i(D) = g_{i0} + g_{i1} \times D + g_{i2} \times D^2 + \dots + g_{im} \times D^m \quad i = 1, 2 \quad (2.3)$$

Le degré  $m$  de polynômes, indique la mémoire de l'encodeur RSC, c'est à dire le nombre des éléments  $D$  de registres à décalage utilisés dans l'implémentation de l'encodeur RSC. La Figure 2.2 illustre un encodeur (RSC) d'un taux de  $\frac{1}{2}$  avec une mémoire  $m = 2$  (4 états) et avec une matrice génératrice donnée par l'équation (2.4).

$$\begin{bmatrix} 1 & \frac{1+D+D^2}{1+D^2} \end{bmatrix} \quad (2.4)$$

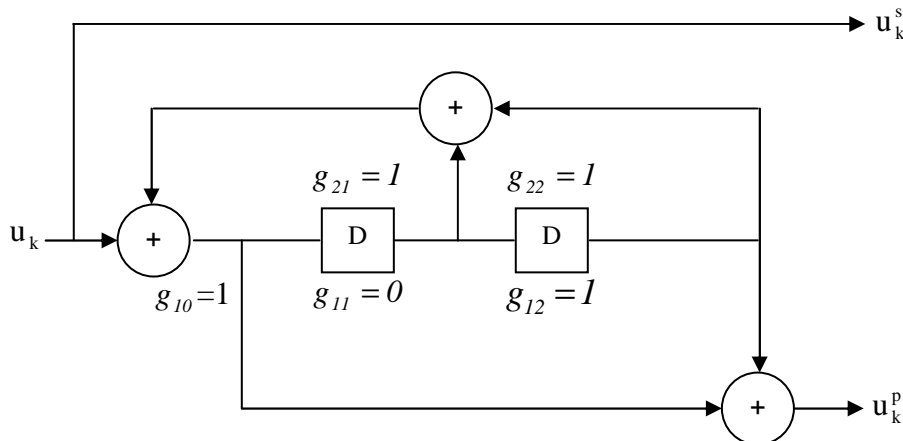


Figure 2.2 : Encodeur RSC



Le symbole  $u_k$  représente le  $k^{\text{ème}}$  bit (ou le bit au temps  $k$ ) d'une séquence de  $L$  bits à l'entrée de l'encodeur RSC. Les symboles  $u_k^s$  et  $u_k^p$  (le bit systématique et le bit de parité, respectivement) représentent les deux sorties de l'encodeur RSC au temps  $k$ . Le bit de parité au temps  $k$ ,  $u_k^p$  est généré par :

$$u_k^p = g_{10} + (s_1^{k-1} \cdot g_{11} + s_2^{k-1} \cdot g_{12}) \tag{2.5}$$

Avec  $g_{10} = u_k + (s_1^{k-1} \cdot g_{21} + s_2^{k-1} \cdot g_{22})$ ; l'opérateur  $(.)$  est équivalent à l'opérateur « ou » exclusif. Les coefficients  $s_i^{k-1}$  ( $i = 1, 2$ ) correspondent sur la Figure 2.2, aux valeurs de registre à décalage (0 ou 1) de gauche à droite avant d'introduire le  $k^{\text{ème}}$  bit d'entrée dans l'encodeur RSC, c'est à dire les valeurs de registre à décalage au temps  $k-1$ . L'ensemble des valeurs  $(s_1 s_2)^{k-1}$  représente l'état  $S$  de l'encodeur RSC au temps  $(k-1)$ . Après l'introduction du  $k^{\text{ème}}$  bit d'entrée à l'encodeur, les valeurs du registre à décalage changent (selon la valeur du bit d'entrée) résultant la transition de l'état de l'encodeur RSC  $S_{k-1}$  à  $S_k$ . Pour chaque état  $S_{k-1}$  de l'encodeur RSC, lorsque le bit  $u_k$  change il y a deux transitions d'état possibles dans l'encodeur RSC ; chacune des deux transitions est associée à une valeur du bit  $u_k$  (0 ou 1). Sans compter le bit d'entrée  $u_k$ , les deux transitions d'état sont aussi caractérisées par le bit de parité  $u_k^p$ .

La Figure 2.3 illustre le diagramme de treillis d'un encodeur RSC avec une matrice génératrice donnée par l'équation (2.2), on suppose que les encodeurs RSC commencent à l'état 0,  $S_0 = 00$  ; l'état final  $S_L$  est considéré inconnu.

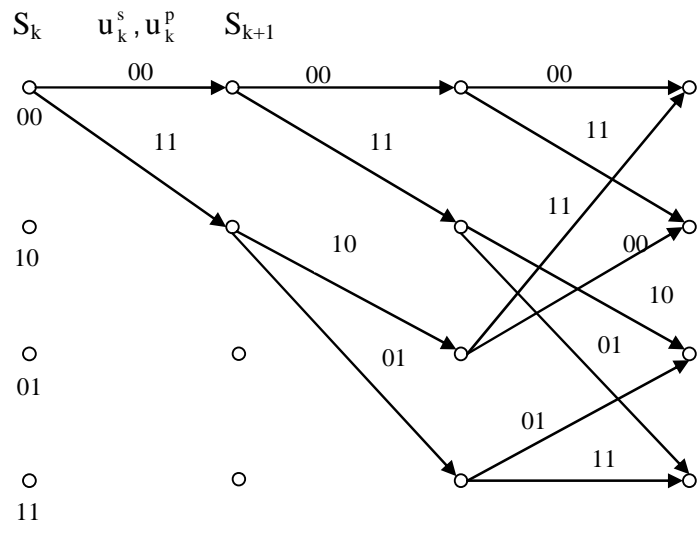


Figure 2.3 : Diagramme de treillis d'un encodeur RSC

Si on décale le premier bit de la séquence d'entrée  $u_1$  dans l'encodeur RSC, deux transitions d'état sont possibles : la transition d'état associée à  $u_1=0$  et celle associée à  $u_1=1$  ; sur la Figure 2.3, chaque état est représenté par un cercle et chaque transition d'état est représentée par une flèche reliant deux cercles. À chaque valeur de bit  $u_1$  (0,1) il lui correspond également une séquence de sortie, formé par les deux bits  $u_k^s$  et  $u_k^p$ . Décalant le bit  $u_2$  dans l'encodeur RSC, deux transitions d'état sont possibles à chaque état résultant du décalage  $u_1$  dans l'encodeur RSC, et ainsi de suite. Le Tableau 2.1 contient, pour chaque état de l'encodeur RSC, les bits  $u_k$  introduits dans l'encodeur RSC et les bits de sortie  $u_k^s$  et  $u_k^p$  correspondant.

Tableau : 2.1 Les états des transitions possibles de l'encodeur RSC

$u_k^s$	$S_k$	$S_{k+1}$	$u_k^p$
0	00	00	0
1		10	1
0	01	10	0
1		00	1
0	10	11	1
1		01	0
0	11	01	1
1		11	0

On note que le diagramme de treillis de l'encodeur RSC est utilisé dans la procédure de turbo décodage, comme il sera décrit dans la section 1.2.1.

### 1.1.2 Entrelacement

La technique d'entrelacement ou de permutation qui consiste à disperser des données dans le temps, a toujours rendu de grands services en communications numériques. On l'utilise pour réduire les effets des atténuations plus ou moins longues dans les transmissions affectées d'évanescences, et plus généralement dans des situations où des perturbations peuvent altérer des symboles consécutifs. Dans le cas de turbo code aussi, la permutation permet de lutter efficacement contre l'apparition de paquets d'erreurs, sur l'une au moins des dimensions du code composite.

Notant qu'un déentrelaceur permet de restaurer l'ordre de la séquence d'entrée de l'entrelaceur.

Le Tableau : 2.2 illustre un entrelaceur avec une séquence d'entrée de longueur égale à 5 bits (pour la simplicité).

Tableau : 2.2 Entrelacement d'une séquence à 5 bits

Séquence d'entrer	Bits d'entrer	Structure de l'entrelaceur	Bits de sortie
$u_1$	0	2	1
$u_2$	1	5	0
$u_3$	1	4	1
$u_4$	1	1	0
$u_5$	0	3	1

L'entrelaceur attribue à chaque index de la séquence d'entrée, un autre index de la séquence de sortie avec une correspondance univoque. Selon la structure de l'entrelaceur représentée sur le Tableau 2.2, le bit de la séquence d'entrée situé dans la 1<sup>ère</sup> position ( $u_1$ ) sera inséré dans la 2<sup>ème</sup> position de la séquence de sortie ; le 2<sup>ème</sup> bit d'entrée ( $u_2$ ) sera inséré dans la 5<sup>ème</sup> position de la séquence de sortie et ainsi de suite. La structure de déentrelaceur est obtenue à partir de la structure de l'entrelaceur :

$$\text{déentrelaceur}[\text{entrelaceur}[j]] = j \quad (2.6)$$

Il y a plusieurs types d'entrelaceurs désignés pour les turbos codes, tels que les entrelaceurs aléatoires, entrelaceurs par bloc et les entrelaceurs convolutionnel [51] ; Notant que les différents types d'entrelaceurs peuvent conduire à des différentes performances [51]. Dans l'encodeur de la Figure 2.1, un entrelaceur pseudo-aléatoire a été adopté. Notant que les  $L$  bits de la séquence d'entrée ne peuvent pas être entrelacés par un mode complètement aléatoire afin de rendre le turbo décodage possible. C'est-à-dire, que le turbo décodeur doit connaître le modèle employé par l'entrelaceur du turbo encodeur pour pouvoir accomplir la tâche de décodage.

### 1.1.3 Poinçonnage de turbo code

Afin de transmettre la quantité minimum des bits requis pour le décodage, le turbo encodeur est basé sur une structure d'un turbo code poinçonné (TCP) [52].

Notant que seuls les bits de parité sont poinçonnés. En effet, un poinçonnage des bits systématiques entraîne une dégradation rapide du seuil de convergence de décodage.

Généralement dans une structure d'un TCP, après codage de la séquence d'entrée de  $L$  bits, la séquence de parité produit par chaque encodeur RSC est divisé en  $P$  blocs de  $(L/P)$  bits chacun, où  $L$  est la longueur de l'entrelaceur et  $P$  s'appelle souvent la période de poinçonnage.

La Figure 2.4 et la Figure 2.5 illustrent un processus de division d'une séquence de parité de 16 bits ( $L=16$ ), si on considère une période de poinçonnage = 4 ; pour la simplicité, les nombres sur la Figure 2.4 et 2.5 indiquent juste la position de chaque bit de parité.



Figure 2.4 : Séquence de parité de 16 bits à la sortie d'un encodeur RSC



Figure 2.5 : Composition de premier bloc de 4 bits

Donc le premier bloc de  $(16/4)$  bits est constitué par les bits de parité situés aux positions 1, 5, 9 et 13 (ou en général aux positions 1,  $P+1$ ,  $2P+1$ , etc..) ; les bits aux positions 2, 6, 10 et 14 (ou 2,  $P+2$ ,  $2P+2$ , etc..) constituent le deuxième bloc de  $(16/4)$  bits et ainsi de suite. Quand le décodeur demande plus de bits de parité par l'intermédiaire du canal de retour, l'encodeur transmet un bloc de  $(16/4)$  bits, donc une fraction du nombre total des bits de parité  $L$ ; évidemment, aucun bloc n'est envoyé deux fois.

Dans notre implémentation les blocs des deux séquences  $P1$  et  $P2$  sont transmis alternativement au décodeur. L'encodeur RSC1 commence par l'envoi d'un bloc de  $(L/P)$  bits de la séquence de parité  $P1$ . Si le décodeur demande plus de bits de parité par l'intermédiaire du canal de retour, l'encodeur RSC2 transmet un bloc de  $(L/P)$  bits de  $P2$ . Si plus de bits de parité sont demandé, l'encodeur RSC1 envoie un autre bloc de  $(L/P)$  bits de  $P1$ , et ainsi de suite jusqu'à ce que il n'y a plus de bits de parité demandé, ou tous les bits de parité  $P1$  et  $P2$  sont transmis.

Notant que l'opération de dépointonnage correspond à l'insertion des bits neutres égale à 0 à l'entrée du décodeur pour l'ensemble des bits non transmis [53].

## 1.2 Turbo décodeur

Le décodage d'un turbo code binaire s'appuie sur le schéma de principe de la Figure 2.6. Le turbo décodeur est constitué de deux décodeurs identiques SISO (Soft-Input, Soft-Output), un à chaque encodeur RSC utilisé dans l'implémentation de turbo code. La boucle permet à chaque décodeur de tirer profit de l'ensemble des informations disponibles. Puisque les deux décodeurs SISO échangent l'information entre eux, le turbo décodeur s'appelle un turbo décodeur itératif. En fait, c'est l'arrangement itératif de décodage qui met la référence du terme turbo.

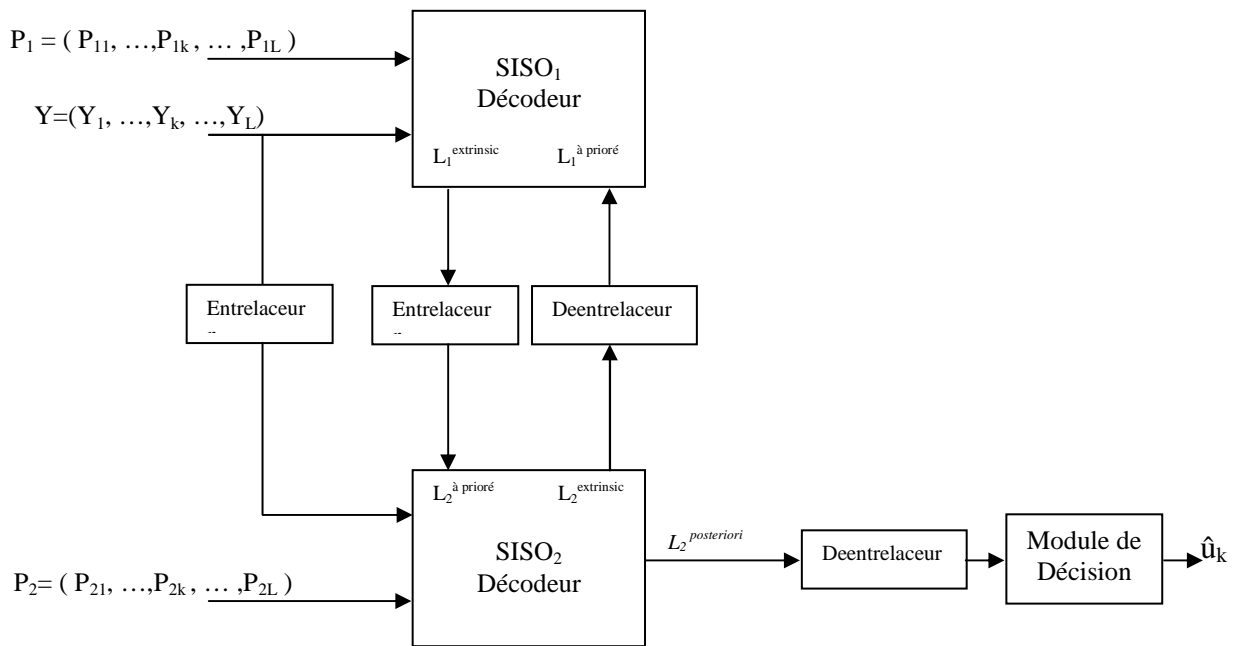


Figure 2.6 : Implémentation d'un Turbo décodeur

Les entrées du décodeur SISO<sub>i</sub> (soft inputs) sont représentées par les symboles  $Y$ ,  $P_i$  et  $L_i^{\text{à priori}}$  ( $i = 1, 2$ ) ; dont  $Y$  est l'information systématique,  $P_i$  l'information de parité et  $L_i^{\text{à priori}}$  est l'information à priori. L'information de canal correspond à l'information systématique  $Y$  et l'information de parité produites par les encodeurs RSC1 et RSC2.

La sortie des décodeurs SISO<sub>i</sub> (soft output) est appelée l'information extrinsèque ou l'information de fiabilité. L'information extrinsèque est représentée par le symbole  $L_i^{\text{extrinsic}}$  ( $i = 1, 2$ ). Généralement, l'amplitude de l'information extrinsèque pour chaque bit  $u_k$  indique la fiabilité de réception (confiance) du bit, le signe de l'information extrinsèque correspond à la valeur de bit  $u_k$  ( $u_k = 1$  si le signe est positif,  $u_k = 0$  si le signe est négatif). Rappelant que le symbole  $u_k$  représente le  $k^{\text{ème}}$  bit de la séquence du

$L$  bits à l'entrée de turbo encodeur.

Le décodage SISO a pour rôle d'augmenter le rapport signal sur bruit, c'est-à-dire de fournir à la sortie une information extrinsèque plus fiable qu'à l'entrée, grâce à la redondance locale ( $Y$ ). Donc, l'information à priori d'un décodeur SISO  $L^{\text{à priori}}$  correspond à l'information extrinsèque calculée par l'autre décodeur SISO.

Puisque la séquence de  $L$  bits d'entrée du RSC2 est entrelacée, il est logique que l'entrelaceur et le déentrelaceur du  $L$  bits entre les deux décodeurs SISO arrangent l'information extrinsèque dans l'ordre approprié à chaque décodeur SISO. Pour la même raison, il est nécessaire d'entrelacer l'information systématique  $Y$ . Notant que la même structure du l'entrelaceur est préétablie pour le turbo encodeur / décodeur.

La procédure itérative de décodage s'arrête quand certains critères de la convergence donnés sont satisfaits [51]. La convergence signifie que les probabilités à posteriori ne changent pas d'une itération à l'autre c'est à dire que les deux SISO ont trouvé les mêmes résultats. Après que les critères de la convergence soient accomplis, une estimation de la séquence à l'entrée de turbo encodeur est obtenue, en se basant sur la décision de la dernière itération de l'information à posteriori (représentée par  $L2$  *posteriori* sur la Figure 2.6).

Après avoir discuté le schéma de principe de turbo décodeur, la section suivante décrit l'algorithme de décodage SISO ainsi que l'opération de décision.

### 1.2.1 Algorithme de décodage SISO dans un canal AWGN

La procédure de décodage d'un turbo code est effectué par les algorithmes du maximum de la vraisemblance (Maximum Likelihood ML), comme le Soft-Output Viterbi Algorithm (SOVA), ou l'algorithme de maximum à posteriori (MAP) [51]. Typiquement, les algorithmes de ML sont employés pour estimer la séquence d'information la plus probable qui a été transmis, alors que l'algorithme de MAP est employé quand les bits d'information les plus probables déjà transmis ont besoin d'être estimé.

Puisque dans le codeur de domaine pixel de WZ (qu'on va décrire dans les sections suivantes) le turbo encodeur est effectué au niveau de bit  $u_k$  (chaque bit  $u_k$  est indépendamment turbo codé voire la section 1.1), l'algorithme de décodage utilisé dans l'implémentation de décodeur SISO est l'algorithme de décodage MAP proposé par Bahl, Cocke, Jelinek et Raviv en 1974 [50].

Bahl et al, en montré dans [50] que l'algorithme MAP, réduit au minimum le taux des bits erronés, une fois qu'il est employé dans un processus de décodage des codes convolutionnel. En 1993, Berrou, Glavieux et Thitimajshima [49] ont utilisé l'algorithme de MAP au décodage itératif du turbo codes. Cependant, Berrou et al. Ont modifié l'algorithme de décodage de MAP dans [50].

Dans la présentation suivante de l'algorithme de décodage SISO la valeur de bit logique 0 est représentée par la valeur  $-1$  et la valeur de bit logique 1 est représentée par la valeur  $+1$ . La procédure itérative de décodage SISO, dans notre implémentation est faite à l'aide d'un algorithme de maximum à posteriori MAP, qui opère sur les logarithmes de vraisemblances associés aux bits systématiques et de parité. L'estimation de chaque bit  $u_k$  dénotée par  $\hat{u}_k$ , est obtenue à partir du logarithme du rapport de probabilité à posteriori (LAPP) défini par (2.7) [53]

$$L(\hat{u}_k) = \ln \left( \frac{p(u_k = +1/r)}{p(u_k = -1/r)} \right) \quad (2.7)$$

Dont  $P(u_k = \pm 1 | r)$  est la probabilité à posteriori, le symbole  $r = (r_1, r_2, \dots, r_L)$  avec  $r_k = (r_k^s, r_k^p) = (Y_k, P_{ik})$  représente l'information du canal pour chaque bit  $u_k$ , notant que l'opérateur  $\ln(\cdot)$  est le logarithme naturel.

Appliquant le théorème de Bayes à (2.7),  $L(\hat{u}_k)$  peut être exprimé comme une somme de deux termes, où le deuxième terme représente l'information a priori du décodeur SISO pour tous les bits  $u_k$ .

$$\begin{aligned} L(\hat{u}_k) &= \ln \left( \frac{p(r/u_k = +1) \times p(u_k = +1)}{p(r/u_k = -1) \times p(u_k = -1)} \right) \\ &= \ln \left( \frac{p(r/u_k = +1)}{p(r/u_k = -1)} \right) + \ln \left( \frac{p(u_k = +1)}{p(u_k = -1)} \right) \end{aligned} \quad (2.8)$$

Dans (2.8),  $p(\cdot)$  dénote la fonction de la densité de probabilité conditionnelle (pdf) de  $r$  sachant que le symbole  $u_k = \pm 1$  a été transmis.

Comme il est mentionné dans la section 1.1.1, chaque encodeur RSC est caractérisé par un diagramme de treillis semblable à celui représenté sur la Figure 2.3. On se basant sur la connaissance du diagramme, la probabilité à posteriori  $P(u_k = +1 | r)$  est

écrite [53]

$$p(u_k = +1/r) = \sum_{S^+} p(S_{k-1}, S_k, r) / p(r) \quad (2.9)$$

Où  $S^+$  représente toutes les transitions possibles  $S_{k-1} \rightarrow S_k$  considérant l'entrée  $u_k = +1$ .

Par analogie

$$p(u_k = -1/r) = \sum_{S^-} p(S_{k-1}, S_k, r) / p(r) \quad (2.10)$$

Où  $S^-$  représente toutes les transitions possibles  $S_{k-1} \rightarrow S_k$  considérant l'entrée  $u_k = -1$ .

Le diagramme de treillis doit donc être connu par le turbo décodeur afin d'effectuer la procédure de décodage [51].

En substitution (2.9) et (2.10) dans (2.7) on trouve

$$L(\hat{u}_k) = \ln \left( \frac{\sum_{S^+} p(S_{k-1}, S_k, r) / p(r)}{\sum_{S^-} p(S_{k-1}, S_k, r) / p(r)} \right) \quad (2.11)$$

La probabilité d'une transition à partir d'un état particulier, avec la séquence observé associé,  $P(S_{k-1}, S_k, r)$  est exprimée par un produit de trois probabilités [53]

$$p(S_{k-1}, S_k, r) = \alpha_{k-1}(S_{k-1}) \times \gamma_k(S_{k-1}, S_k) \times \beta_k(S_k) \quad (2.12)$$

La probabilité conjoint d'arrivé à une branche et à un état particulier avec la séquence observé associé qui mène à cette état  $(r_1, r_2, \dots, r_k)$ , est calculé à partir de (2.13) [53].

$$\alpha_k(S_k) = p(S_k, r_{1,k}) = \frac{\sum_{S_{k-1}} \alpha_{k-1}(S_{k-1}) \times \gamma_k(S_{k-1}, S_k)}{\sum_{S_k} \sum_{S_{k-1}} \alpha_{k-1}(S_{k-1}) \times \gamma_k(S_{k-1}, S_k)} \quad (2.13)$$



Si en considère que l'état initial de l'encodeur RSC,  $S_0$ , est connu, l'initialisation de  $\alpha_k(S_k)$  est donné par [53]

$$\alpha_0^i(S_k) = \begin{cases} 1, & S_k = S_0 \\ 0, & S_k \neq S_0 \end{cases}, i = 1, 2 \quad (2.14)$$

Dont  $i$  représente l'index de l'encodeur RSC (1 pour RSC1 et 2 pour RSC2).

La probabilité de sortir d'un état particulier  $S$ , à travers une branche avec la séquence observée ( $r_k, r_{k+1}, \dots, r_L$ ) qu'elle finit le treillis est donné par (2.15) [53].

$$\beta_{k-1}(S_{k-1}) = p(r_{k,L} / S_{k-1}) = \frac{\sum_{S_k} \beta_k(S_k) \times \gamma_k(S_{k-1}, S_k)}{\sum_{S_k} \sum_{S_{k-1}} \alpha_{k-1}(S_{k-1}) \times \gamma_k(S_{k-1}, S_k)} \quad (2.15)$$

L'équation (2.16) exprime les conditions limités de  $\beta_k^1(S_k)$  (c'est-à-dire. pour  $k=L$ ) en supposent que le dernier état de l'encodeur RSC1  $S_L$  est connu [53].

$$\beta_L^1(S_k) = \begin{cases} 1, & S_k = S_L \\ 0, & S_k \neq S_L \end{cases} \quad (2.16)$$

Si l'état final de l'encodeur RSC1  $S_L$  est inconnu, la condition de la limite est donnée par [51]

$$\beta_L^1(S_k) = \frac{1}{2^m}, \quad S_k = 1, 2, \dots, L \quad (2.17)$$

Où  $m$  est la mémoire de l'encodeur RSC. La probabilité de l'encodeur RSC2  $\beta_L^2(S_k)$  est initialisée avec les valeurs de la probabilité  $\alpha_L^2(S_k)$  pour tous les états  $S_k$  [53].

A partir de [53], la probabilité  $\gamma_k(S_{k-1}, S_k)$  est écrite

$$\gamma_k(S_{k-1}, S_k) = p(u_k) p(r_k / u_k) \quad (2.18)$$

Et la probabilité à priori  $P(u_k)$  [53] est exprimé par

$$p(u_k = \pm 1) = \frac{e^{-\frac{1}{2}L^{\text{apriori}}(u_k)}}{1 + e^{-L^{\text{apriori}}(u_k)}} \times e^{-\frac{u_k}{2}L^{\text{apriori}}(u_k)} = A_k \times e^{-\frac{u_k}{2}L^{\text{apriori}}(u_k)} \quad (2.19)$$

Dont  $L^{\text{à priori}}(u_k)$  est définie par l'équation (2.20)

$$L^{\text{apriori}}(u_k) = \ln\left(\frac{p(u_k = +1)}{p(u_k = -1)}\right) \quad (2.20)$$

Dans l'équation (2.18), la probabilité  $p(r_k / u_k)$  est la pdf conditionnel de  $r_k$  sachant que  $u_k$  a été transmis. Le model du canal AWGN est un simple canal qui est commun dans les systèmes de communication. Pour tester notre turbo code (dans le chapitre suivant), la transmission des symboles  $\pm 1$  est modelé par un bruit gaussien blanc.

La probabilité  $p(r_k / u_k)$  est donc exprimée par une distribution gaussienne

$$\begin{aligned} p(r_k / u_k) &= p(r_k^s / u_k) p(r_k^p / u_k) \\ &= B_k e^{-\left(\frac{r_k^s u_k + r_k^p u_k^p}{\sigma^2}\right)} \end{aligned} \quad (2.21)$$

En peut exprimer la probabilité  $\gamma_k(S_{k-1}, S_k)$

$$\gamma_k(S_{k-1}, S_k) = A_k B_k e^{\left(\frac{1}{2}(u_k L^{\text{apriori}}(u_k) + u_k L_c r_k^s + u_k^p L_c r_k^p)\right)}, \quad L_c = \frac{2}{\sigma^2} \quad (2.22)$$

Puisque le terme  $\gamma_k(S_{k-1}, S_k)$  s'apparaît au numérateur et au dénominateur dans (2.11); le produit  $A_k \times B_k$  est indépendant de  $u_k$  et par conséquent il peut être annulé.

L'équation (2.11) est réécrite utilisant (2.12) et (2.22)

$$\begin{aligned}
L(\hat{u}_k) &= \ln \left[ \frac{\sum_{S^+} \alpha_{k-1}(S_{k-1}) e^{\frac{1}{2} [L^{\text{a priori}}(u_k) + L_c r_k^s + u_k^p L_c r_k^p]} \beta_k(S_k)}{\sum_{S^-} \alpha_{k-1}(S_{k-1}) e^{\frac{1}{2} [-L^{\text{a priori}}(u_k) - L_c r_k^s + u_k^p L_c r_k^p]} \beta_k(S_k)} \right] \\
&= L^{\text{a priori}}(u_k) + L_c r_k^s + \ln \left[ \frac{\sum_{S^+} \alpha_{k-1}(S_{k-1}) e^{\left(\frac{u_k^p L_c r_k^p}{2}\right)} \beta_k(S_k)}{\sum_{S^-} \alpha_{k-1}(S_{k-1}) e^{\left(\frac{u_k^p L_c r_k^p}{2}\right)} \beta_k(S_k)} \right] \quad (2.23) \\
&= L^{\text{a priori}}(u_k) + L_c r_k^s + L^{\text{extrinsèque}}(u_k)
\end{aligned}$$

Dont le premier élément  $L^{\text{a priori}}(u_k)$  est l'information à priori,  $L_c r_k^s$  est l'information qui provient de bruit et le dernier terme est l'information extrinsèque.

### 1.2.2 Opération de décision

Après que les itérations de décodage soient finies, L'estimation  $\hat{u}_k$  de chaque bit d'entrer  $u_k$  du turbo codeur, résulte du seuillage de la dernière information à posteriori de SISO2. La valeur de seuil est mise à zéro (valeur centrale entre +1 et -1).

Ensuite le taux des bits erronés  $P_e$  est calculé, dont il provient de la comparaison entre chaque bit estimé  $\hat{u}_k$  et bit d'information  $u_k$ , ce processus s'appelle la détection idéale des erreurs. Si  $P_e$  est plus grand qu'un certain seuil donné, le décodeur demande plus de bits de parité de l'encodeur par l'intermédiaire du canal de retour. Autrement, la procédure de turbo décodage est terminée et la sortie du turbo décodeur est la séquence  $\hat{u}_k$ . Dans notre implémentation le seuillage de  $P_e$  est fixé à  $1 \times 10^{-3}$ .

Dans la section suivante, en va décrire en détail les différents modules constituant notre CVD, notant qu'il y a quelque changement dans l'algorithme de décodage SISO exposé précédemment, dont le canal est modelé par un bruit Laplacian.

## 2. Implémentation d'un CVD Intra/WZ

Dans les propositions de l'état de l'art du CVD, la séquence vidéo est divisée en deux groupes : les images clés et les images Wyner-Ziv [36]. Pour décoder une image WZ, une estimation de celle-ci est produite par les techniques d'interpolation appliquées sur les images clés Intra codées/décodées. Dans d'autres propositions, le codage Intra à base de blocs est utilisé pour coder certains blocs des images WZ qui sont difficiles à prédire [54]. De même, notre proposition adopte une structure à base de blocs. Dans une seule macro-bloc MB, une partie est codée Intra et l'autre partie est codée WZ. Cet arrangement-là, nous permet d'augmenter l'efficacité de l'estimation de mouvement, dont l'information Intra est employée pour estimer partiellement le mouvement et prédire le reste du MB.

La Figure 2.7 illustre l'architecture du codeur qu'on a implémenté. L'architecture générale est semblable à celle proposée par Aaron et al dans [36] : tous les deux se servent d'un quantificateur, d'un codeur de Slepian-Wolf à base d'un turbo code, d'un module d'interpolation/estimation et d'un module de reconstruction. Mais il y a cependant quelques différences capitales.

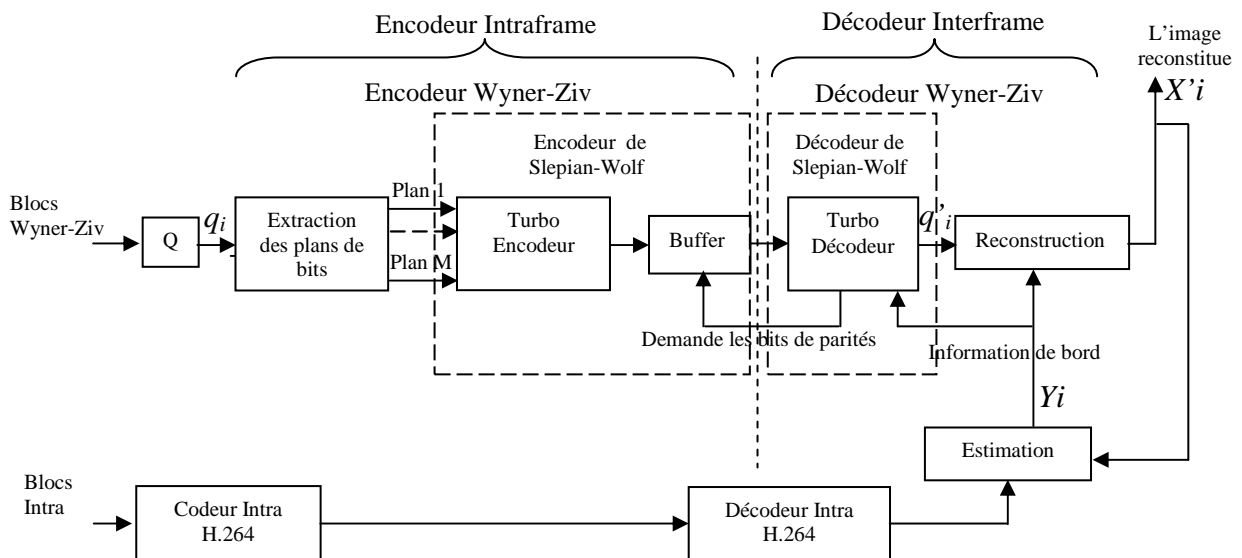


Figure 2.7 : Architecture du codeur vidéo distribué

Au lieu de coder l'image entière par un codage Intra ou WZ, on va combiner les deux modes pour coder tous les MBs de l'image, c'est à dire un codage mixte Intra/WZ. Notre approche propose une structure du codage où chaque image vidéo est divisée en

MBs de 16x16 (pixels) qui sont aussi divisé en blocs de 4x4.

1	3	1	3
4	2	4	2
1	3	1	3
4	2	4	2

3	1	3	1
2	4	2	4
3	1	3	1
2	4	2	4

4	2	4	2
1	3	1	3
4	2	4	2
1	3	1	3

2	4	2	4
3	1	3	1
2	4	2	4
3	1	3	1

Figure 2.8 : L'ordre de codage des blocs WZ (blanc) et Intra (gris) pour un MB

Pour chaque image un ordre de codage est imposé selon la Figure 2.8, les blocs gris sont codés Intra, cependant les blocs restant (blanc) sont codés WZ. Le résultat est une opération de codage à 4 étapes.

Pour une image donnée, un seul ordre est imposé pour coder toute les MBs de l'image entière Figure 2.8 (ordre 1, 2, 3 ou 4). Les premiers blocs codés sont les blocs Intra. Dans le module d'estimation, une estimation partielle de mouvement utilisant les quatre blocs Intra décodés de chaque MB de l'image actuelle et l'image précédemment décodée  $Y_i$  considéré comme une information de bord est effectué. Le vecteur de mouvement obtenu est employé pour prédire et compensé (MCP) les blocs restants de chaque MB (la compensation est faite a partir de l'image précédemment décodé). La deuxième étape consiste de coder les blocs WZ numéroté par 2, Figure 2.8, et ainsi de suite pour la troisième et la quatrième étape, qui consiste à coder les blocs WZ numérotés par 3 et 4 respectivement. Dans la première étape seul les blocs Intra sont employés pour estimer le mouvement, cependant les autres étapes exploitent en même temps les blocs Intra et WZ (décodés). A chaque étape on a une image complète au décodeur, dont une partie de celle-ci est obtenue à partir de l'image courante  $X_i$  (blocs décodés), et l'autre partie résulte du processus de la compensation, le flux binaire de WZ améliore la qualité de l'image compensée à chaque étape. En conclusion, la qualité de l'image finale dépend de paramètre de la quantification utilisée dans le codage des blocs Intra, et de l'efficacité de MCP.

On note que la corrélation partielle entre l'information de bord  $Y_i$  et les MBs de l'image actuel  $X_i$ , est examinée à chaque étape, afin de bien modéliser le canal virtuel. Pour permettre une qualité constante à travers toute la séquence vidéo, c'est à dire éviter l'accumulation des erreurs dans certaines régions, l'ordre du codage change périodiquement dans la séquence vidéo (Figure 2.8). La première image d'une

séquence vidéo (N° 0) est codée entièrement Intra, la deuxième image (N° 1) est codée avec l'ordre du codage montré sur la Figure 2.8 (1). De même pour la troisième, la quatrième et la cinquième image (N° 2, N° 3 et N° 4) qui adapte l'ordre du codage montré sur la Figure 2.8 (2), Figure 2.8 (3) et Figure 2.8 (4) respectivement. Cette opération continue avec les images restantes, et après une période de 4 images on revient toujours au mode du codage de la Figure 2.8 (1).

Sur la Figure 2.9 on peut voir le résultat de codage (à 4 étapes) considérant une seule image. En notant que 4 sur 16 blocs de chaque MB sont codés Intra, c'est à dire que 25% de la séquence vidéo est codée Intra. Si on compare notre structure avec celle qui utilise le codage WZ avec les images clés, notre structure est équivalente à l'insertion de 3 images WZ entre 2 images clés.

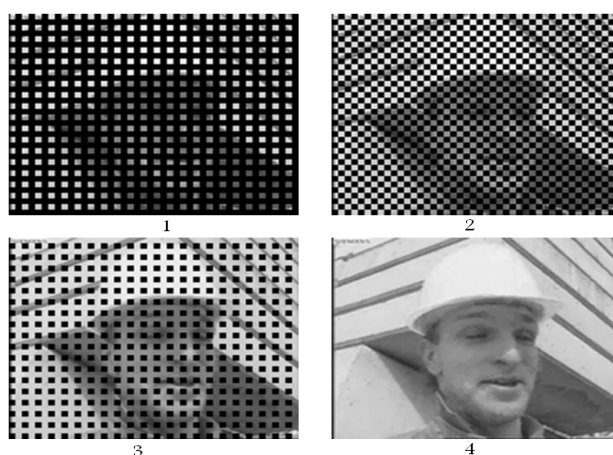


Figure 2.9: L'information disponible au décodeur après chaque étape

Dans notre approche, les blocs Intra des 4 images successives sont groupés ensemble pour construire une image complète, l'image résultante est ensuite codée par le codeur Intra H.264/AVC 4x4 standard avec transformation DCT. Pour le codage de WZ, les blocs WZ de l'image entière à une étape du codage donnée, sont groupés et codés ensemble par un turbo code.

Puisque le codage standard Intra H264/AVC est une technique de codage connue, le processus de codage ne sera pas détaillé ici, donc une attention particulière sera donnée aux procédures du codage/décodage de Wyner-Ziv.

Dans les sections suivantes, l'architecture et l'implémentation de chaque module du codeur de domaine pixel Wyner-Ziv (PDWZ) représenté sur la Figure 2.7 est décrit en détail.

## 2.1 Encodeur Wyner-Ziv

Considérons l'architecture de codage illustrée sur la Figure 2.7. Chaque image de la séquence vidéo est d'abord divisée en MB (16x16 qui sont aussi divisés en blocs de 4x4). La procédure du codage distribué de Wyner-Ziv à une étape donnée est décrite comme suit :

Les blocs Wyner-Ziv d'une image  $X_i$  (grouper à une étape donnée) sont d'abord échantillonnés Pixel par Pixel, puis quantifiés utilisant  $2^M$  niveau uniforme, ensuite une extraction des plans de bit est effectuée sur le flux des symboles quantifiés  $q_i$ , chaque plan de bit résultant est indépendamment introduit dans le turbo code. L'information redondante (de parité) produite par le turbo encodeur, pour chaque plan de bit est stockée dans un buffer, et transmise par des petites quantités selon la demande de décodeur, par l'intermédiaire du canal de retour.

### 2.1.1 Quantification

Les Pixel de chaque blocs Wyner-Ziv d'une image  $X_i$ , sont quantifiés utilisant un quantificateur scalaire uniforme avec  $2^M$  niveaux; le paramètre  $M$  correspond au nombre de bits requis pour qu'un pixel soit dans l'un des niveaux de la quantification  $2^M$ . Si on variant la valeur de  $M$ , différentes performances de taux de distorsion peuvent être atteintes, puisque chaque valeur de  $M$  à un taux de distorsion associé. Quatre points du taux de distorsion ont été considérés dans l'évaluation des performances de notre codeur qui correspondent aux valeurs de  $M = 1, 2, 3$  et  $4$ .

Après la quantification des blocs WZ, les symboles quantifiés sont convertis en flux binaire. Ensuite les bits des symboles quantifiés de la même importance (par exemple les bits les plus significatifs) sont groupés ensemble, formant un vecteur d'un plan de bit donné.

### 2.1.2 Turbo encodeur

Après la formation des  $M$  vecteurs des plans de bit associés aux symboles quantifiés, chaque vecteur (plan de bit) est introduit dans l'encodeur de Slepian-Wolf, qui comporte typiquement un turbo encodeur et un buffer. Commencé par le vecteur du plan de bit le plus significatif, le turbo encodeur implémenté dans la section 1.1 génère les séquences de parité P1 et P2. Notant que les séquences systématiques S1 et S2 montrées sur la Figure 2.1 sont retirées. Cependant, ces séquences sont remplacées par

l'information reconstruit  $X'i$  (utilisant l'information de bord  $Yi$  et les blocs décodés Intra ou WZ). Les bits de parité produits par le turbo encodeur sont puis stockés dans un buffer, ensuite poinçonné et transmis selon la demande du décodeur par l'intermédiaire du canal de retour.

## 2.2 Décodeur Wyner-Ziv

Au décodeur, le module d'estimation est exploité pour générer une estimation de l'image  $Xi, Yi$ . Cette estimation est ensuite utilisé pour obtenir le flux des symboles quantifiés décodés  $q'i$ . L'image  $Yi$ , connue sous le nom d'information de bord, ainsi que le flux  $q'i$ , sont également employée dans le module de reconstruction, pour accomplir la tâche de la reconstruction de  $X'i$ .

### 2.2.1 Estimation des images

La plus part des technique existent dans la littérature suppose que les images impaires de la séquence vidéo  $X_{2i-1}$  et  $X_{2i+1}$  (appelé les images clés) sont disponibles au décodeur, une interpolation bilinéaire (moyenne) est applique entre les images clés  $X_{2i-1}$  et  $X_{2i+1}$  pour générer l'information de bord. Subjectivement, cette technique née pas une solution pratique, dont il considère que l'image  $X_{2i+1}$  qui né pas encore codée est disponible au décodeur (sans aucune compression).

Le codage vidéo avec prédiction et compensation du mouvement à base de bloc est devenu un outil très populaire dans ces dernières années. C'est cet arrangement du codage qui est le plus souvent employé dans les normes du codage traditionnel, ainsi que la grande partie déployé dans les nouvelles applications telles que le codage vidéo sur Internet et sur les canaux sans fil. Dans notre approche en se serve de cet arrangement de codage avec l'assortiment partiel pour accélérer et augmenter l'efficacité d'EM [55], au lieu d'utiliser une extrapolation temporelle nous employons une estimation spatiale pour identifier les vecteurs de mouvement. L'assortiment partiel est basé sur la supposition que le mouvement est spatialement doux. Donc, en se basant sur l'EM de quelques pixels (du MB) pour décoder le MB entière, d'autre part, cette estimation est employée pour examiner la corrélation entre l'information de bord et les blocs non transmis encore au décodeur (utilisant le MB décodé).



Notant que la somme de la différence absolu (SDA) est employé pour spécifier l'assortiment partielle entre les blocs décodés de l'image  $X_i$  et celle possible rechercher dans  $Y_i$ .

### 2.2.2.1 Turbo décodeur

La structure du turbo code poinçonné employé dans notre codeur PDWZ, permet d'adopter le taux de Wyner-Ziv, selon la similitude entre les blocs WZ de l'image  $X_i$  (a codé) et celle de l'information de bord correspondante  $Y_i$ . C'est à dire, plus que la similitude entre les blocs décodés et son estimation faite au décodeur augmente, plus que les bits de parité (bit de Wyner-Ziv) nécessaire d'être envoyé au décodeur diminue.

Afin d'obtenir une estimation des symboles quantifiés  $q'_i$ , le turbo décodeur exploite les séquences de parité poinçonné, P1 et P2, envoyé par le turbo encodeur pour corriger les erreurs dans l'information de bord résultant du module d'estimation.

Dans l'architecture du codeur PDWZ représentée sur la Figure 2.7, le turbo décodeur décrit dans la section 1.2 est exploité avec une autre modélisation de canal virtuel, adéquate au codage vidéo distribué.

### 2.2.2.2 Algorithme de décodage SISO dans un canal Laplacienne

Dans notre turbo codeur, un autre model Laplacienne est implémenté afin de bien modélisé le canal virtuel, entre l'information de la source  $X_i$  (l'image a codé) et l'information de bord  $Y_i$  qui est une version bruité de l'image  $X_i$  généré au décodeur (voir Figure 2.7).

Reprenant l'équation (2.11) et (2.12)

$$L(\hat{u}_k) = \ln \left( \frac{\sum_{S^+} p(S_{k-1}, S_k, r) / p(r)}{\sum_{S^-} p(S_{k-1}, S_k, r) / p(r)} \right) \quad (2.11)$$

$$p(S_{k-1}, S_k, r) = \alpha_{k-1}(S_{k-1}) \times \gamma_k(S_{k-1}, S_k) \times \beta_k(S_k) \quad (2.12)$$

Le calcul des probabilités  $\alpha_k(S_k), \beta_{k-1}(S_{k-1})$  reste le même à celui développé dans la section 1.2.1, cependant le calcul de  $\gamma_k(S_{k-1}, S_k)$  sera changé.

Si on réécrit l'équation (2.18)

$$\gamma_k(S_{k-1}, S_k) = p(u_k) p(r_k / u_k) \quad (2.18)$$

Dont  $P(u_k)$  est donné par (2.19).

Rappelant que la probabilité  $P(r_k / u_k)$  est la pdf conditionnel de  $r_k$  sachant que  $u_k$  a été transmis. Ou,  $r_k = (r_k^s, r_k^p) = (Y_k, P_{ik})$  avec  $P_{ik}$  est le  $k^{\text{ème}}$  bit de parité reçu au décodeur ( $i$  représente l'index de l'encodeur RSC) et  $Y_k$  est le  $k^{\text{ème}}$  bit systématique, pour chaque  $u_k$ .

### **Modeler le résiduel de l'information de parité**

Pour calculer la probabilité  $\gamma_k(S_{k-1}, S_k)$ , il est nécessaire de savoir comment modéliser la pdf conditionnel  $P(r_k / u_k)$ . Dans notre codeur PDWZ, on a supposé qu'aucune erreur n'est introduite dans la transmission des bits de parité (c'est à dire, dans le flux de Wyner-Ziv). Alors, les bits de parité transmis sont les mêmes bits reçus au décodeur.

Dans ce cas la probabilité de recevoir le bit de parité  $P_{ik} = \pm 1$  sachant que  $u_k^p = \pm 1$  a été transmis égale à 1, et la probabilité de recevoir  $P_{ik} = \mp 1$  sachant que  $u_k^p = \pm 1$  a été transmis égale à zéro. Donc, la pdf conditionnel  $p(r_k^p / u_k) = p(p_{ik} / u_k)$  peut être décrit par la fonction delta de Dirac  $\delta(t)$  typiquement défini par (2.24) et (2.25) [56].

$$\int_{-\xi}^{\xi} \delta(t) dt = 1 \quad (2.24)$$

$$\delta(t) = 0, \quad t \neq 0 \quad (2.25)$$

Dont  $\varepsilon$  représente une valeur arbitrairement petite.

Donc, pour pouvoir calculer  $p(p_{ik}/u_k)$ , il est nécessaire de rapprocher la fonction delta  $\delta(t)$ . La distribution gaussienne avec une moyenne zéro et une variance arbitrairement petite  $\sigma^2$  peut bien rapprocher la fonction delta de Dirac.

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

$$\approx c \times e^{-\frac{x^2}{2\sigma^2}} \quad (2.26)$$

### Modeler le résiduel de l'information systématique

Comme il est déjà mentionné, l'information de bord  $Y_i$  généré au module d'estimation est considérée comme une version bruitée de l'image a codée  $X_i$ .

La Figure 2.10 montre la distribution du résiduel  $x$ , c'est à dire la différence de la luminance entre les pixels de  $X_i$  et  $Y_i$  correspondants, pour la séquence vidéo QCIF de Foreman. Une distribution Laplacian donnée par (2.27) est également tracée sur la Figure 2.12, avec le paramètre alpha égal à 0.58.

$$f(x) = \frac{\alpha}{2} e^{-\alpha|x|} \quad (2.27)$$

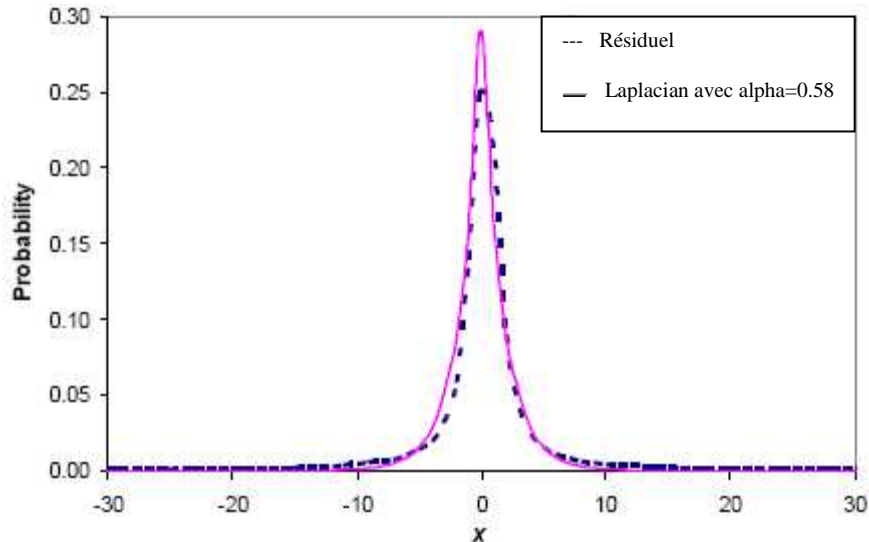


Figure 2.10 : Distribution de la résiduel  $x$  de la séquence vidéo QCIF de Foreman [57]

En peut remarquer de la Figure 2.10 que la distribution Laplacian se rapproche bien de la distribution résiduelle. Donc, le résiduel entre  $X_i$  et  $Y_i$  est modelé par une distribution Laplacian pour modéliser le bruit dans le canal virtuelle de transmission de la Figure 2.7. Rappelant que La corrélation partielle entre l'information de bord  $Y_i$  et les MBs de l'image actuel  $X_i$  décodé compensé est examiné à chaque étape afin de bien modélisé le canal virtuelle, ou le paramètre alpha est calculé pour chaque MB

$$\alpha^2 = \frac{2}{\sigma_{Y,X}^2} \quad (2.28)$$

### Modeler la pdf conditionnelle

De [58] la probabilité  $P(r_k | u_k)$  est écrite

$$\begin{aligned} p(r_k / u_k) &= p(r_k^s / u_k) p(r_k^p / u_k) \\ &\approx e^{-\alpha |r_k^s - u_k|} \times e^{-\frac{(r_k^p - u_k^p)^2}{2\sigma^2}} \\ &= e^{-\alpha |r_k^s - u_k|} \times e^{-\frac{(r_k^p)^2 + (u_k^p)^2 - 2(r_k^p \times u_k^p)}{2\sigma^2}} \\ &= B_k \times e^{-\alpha |r_k^s - u_k|} \times e^{-\frac{(r_k^p \times u_k^p)}{\sigma^2}} \end{aligned} \quad (2.29)$$

En Substituent (2.19) et (2.29) dans (2.18) en trouve

$$\gamma_k(S_{k-1}, S_k) \approx A_k \times e^{\frac{u_k}{2} L_{\text{apriori}}(u_k)} \times B_k \times e^{-\frac{r_k^p \times u_k^p}{\sigma^2}} \times e^{-\alpha |r_k^s - u_k|} \quad (2.30)$$

Alors l'équation (2.30) peut être rapprochée par [53]

$$\gamma_k(S_{k-1}, S_k) \sim e^{\frac{u_k}{2} L_{\text{apriori}}(u_k)} \times e^{-\frac{r_k^p \times u_k^p}{\sigma^2}} \times e^{-\alpha |r_k^s - u_k|} \quad (2.31)$$

Puisque le terme  $\gamma_k(S_{k-1}, S_k)$  s'apparaît au numérateur et au dénominateur dans (2.11); le produit  $A_k \times B_k$  est indépendant de  $u_k$  et par conséquent il peut être annulé.

Donc, l'équation (2.11) peut être réécrit utilisant (2.12) et (2.31)

$$\begin{aligned}
 L(\hat{u}_k) &= \ln \left[ \frac{\sum_{S^+} \alpha_{k-1}(S_{k-1}) \times e^{\frac{u_k}{2} L^{\text{apriori}}(u_k)} \times e^{\frac{r_k^p \times u_k^p}{\sigma^2}} \times e^{-\alpha |r_k^s - u_k|} \times \beta_k(S_k)}{\sum_{S^-} \alpha_{k-1}(S_{k-1}) \times e^{\frac{u_k}{2} L^{\text{apriori}}(u_k)} \times e^{\frac{r_k^p \times u_k^p}{\sigma^2}} \times e^{-\alpha |r_k^s - u_k|} \times \beta_k(S_k)} \right] \\
 &= L^{\text{apriori}}(u_k) + \ln \left[ \frac{\sum_{S^+} \alpha_{k-1}(S_{k-1}) \times \beta_k(S_k) \times e^{\frac{r_k^p \times u_k^p}{\sigma^2}} \times e^{-\alpha |r_k^s - u_k|}}{\sum_{S^-} \alpha_{k-1}(S_{k-1}) \times \beta_k(S_k) \times e^{\frac{r_k^p \times u_k^p}{\sigma^2}} \times e^{-\alpha |r_k^s - u_k|}} \right] \quad (2.32)
 \end{aligned}$$

Le premier élément  $L^{\text{apriori}}(u_k)$  est l'information à priori, et le deuxième terme indique l'information extrinsèque qui est échangée entre les deux modules SISO.

Notant, qu'on va exploiter la même opération de décision décrite à la section 1.2.2 dans le turbo décodeur de PDWZ.

### 2.2.3 Reconstruction

La dernière étape pour décoder les blocs WZ est le processus de reconstruction.

Les  $M$  plans de bit décodés sont groupés ensemble, pour former le stream  $q'i$  qui correspond aux valeurs des Pixels quantifié des blocs WZ de l'image  $X_i$ .

La procédure de reconstruction de chaque pixel est décrit sur la Figure 2.11, dont  $X_i$  et  $Y_i$  sont des entiers et  $Q = [ab]$  est un intervalle de quantification, l'image  $X'i$  et reconstruit selon la relation suivante :

$$X_i' = \begin{cases} a & Y_i < a \\ Y_i & a \leq Y_i \leq b \\ b & Y_i > b \end{cases} \quad (2.33)$$

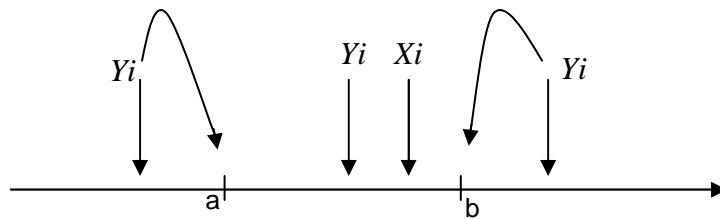


Figure 2.11 : Fonction de reconstruction d'un quantificateur scalaire uniforme

- Si la valeur du pixel de l'information de bord appartient à l'intervalle de symbole quantifié décodé  $q'_i$ , la valeur de pixel reconstruite est rendue égale à la valeur de Pixel de l'information de bord.
- Si la valeur de Pixel de l'information de bord est inférieure à la petite borne de l'intervalle de symbole quantifié décodé  $q'_i$ , la valeur de pixel reconstruite prend la valeur de cette borne.
- Si la valeur de pixel de l'information de bord est supérieure à la borne supérieure de l'intervalle de symbole quantifié décodé  $q'_i$ , la valeur de pixel reconstruite prend la valeur de cette borne.

L'erreur de reconstruction est alors garanti d'être inférieur au pas de la quantification  $|X_i - X'_i| \leq |a - b|$  alors plus que  $M$  est élevé plus que l'erreur de reconstruction diminuée.

Enfin les blocs résultant du décodage WZ à une étape donné, sont remplacés à leur position correspondant dans l'image décodée  $X'_i$ .

## Conclusion

Dans ce chapitre nous avons discuté une nouvelle architecture plus pratique du codage vidéo distribué. Dans la plupart des codeurs de l'état de l'art du CVD, la séquence vidéo est divisé en deux groupe, les images clés et les images Wyner-Ziv. Certain architecture suppose que les images clé sont disponibles au décodeur sans aucune compression (images originale). Notre codeur combine les deux modes de codage Intra et Wyner-Ziv pour coder chaque MB d'une image donnée. Pour coder les blocs WZ on a implémenté un turbo code poinçonné pour réduire la redondance de flux de Wyner-Ziv. On s'est servi du codage vidéo prédictif à base de bloc dans le module d'estimation, comme on a employé l'assortiment partiel pour accélérer et augmenter l'efficacité d'EM.

Dans le chapitre suivant l'évaluation des performances de turbo code poinçonné et de CVD de PDWZ est montrée.

# Chapitre 3

## Résultats et discussion

### Introduction

Dans ce chapitre, les résultats expérimentaux de turbo code poinçonné et du codeur vidéo distribué sont présentés. Leurs performances sont mesurées à l'aide du taux de compression et de la relation débit-distorsion respectivement. Les résultats du CVD sont comparés à la fin aux performances de la relation débit-distorsion de la norme H.264 Intra.

Pour l'évaluation des performances de la relation débit-distorsion de CVD, on a sélectionné trois séquences vidéo caractérisées par des quantités de mouvement différentes. La séquence de Hall filme différents objets avec plusieurs détails et des quantités de mouvements relativement petites, la séquence est enregistrée à l'aide d'une caméra fixe. La séquence de Foreman est un exemple d'une conférence vidéo, dont elle montre un mouvement irrégulier, elle est enregistrée par une caméra portable. En fin la séquence de Soccer inclut un mouvement rapide, elle filme un match de football, la séquence est enregistrée par une caméra fixe.

### 1. Résultats de simulation du turbo code

Cette section présente les résultats des performances du turbo code poinçonné, ces résultats sont mesurés à l'aide de taux de compression CR, dans un canal AWGN. Les résultats obtenus servent comme une référence dans l'implémentation du turbo code poinçonné dans un canal Laplacien pour le CVD. Les simulations emploient le turbo code implémenté dans la section 1 du chapitre précédent, avec une longueur de la séquence d'entrée  $X$  (pour les deux premiers tests),  $L = 1000$  [53], pour fixer la période de poinçonnage  $P$  on a effectué plusieurs tests, la valeur de  $P = 21$  donne les meilleurs résultats (en terme de taux de compression), d'autre part, le seuillage du taux des bits erronés  $P_e$  doit être choisi assez petit pour diminuer le nombre des erreurs des bits décodés, pour cela on va choisir  $P_e = 1 \times 10^{-3}$ , on note que cette valeur est la plus souvent employée dans la littérature [4], [16] et [31].



## 1.1 L'évaluation du changement du nombre d'itérations sur l'efficacité de décodage

A chaque itération il y a un échange des informations extrinsèques entre les deux décodeurs SISO<sub>i</sub> ( $i = 1,2$ ), donc l'augmentation du nombre des itérations va augmenter le temps de calcul, ce qui est inacceptable pour des applications en temps réel. Pour limiter le nombre des itérations nécessaires pour satisfaire la convergence du décodeur, on va chercher à déterminer l'inverse du taux de compression  $1/CR$ , en fonction de la variance du bruit, ou plus précisément en fonction du rapport signal sur bruit SNR, ainsi que le nombre des itérations considérées dans le processus de décodage. La séquence d'entrée générée aléatoirement dans l'encodeur est transmise dans le canal AWGN. Dans l'évaluation des performances du codeur vidéo distribué (sections suivantes), 4 points du taux de distorsion sont considérés ( $M = 1, 2,3,4$ ). Ces points correspondent respectivement au codage/décodage du premier plan de bit le plus significatif, des deux plans de bit les plus significatifs, des trois plans de bit et des quatre plans de bit les plus significatifs de chaque image. Donc le canal de transmission de chaque taux de distorsion est modélisé par des bruits différents qui constituent l'estimation des plans de bits les moins significatifs qui désignent les détails dans une image pour lesquels le module d'estimation (de la Figure 2.7), est moins efficace. Donc trois SNR différents sont considérés dans cette simulation : 1 dB, 1,5 dB et 2 dB respectivement. Figure 3.1

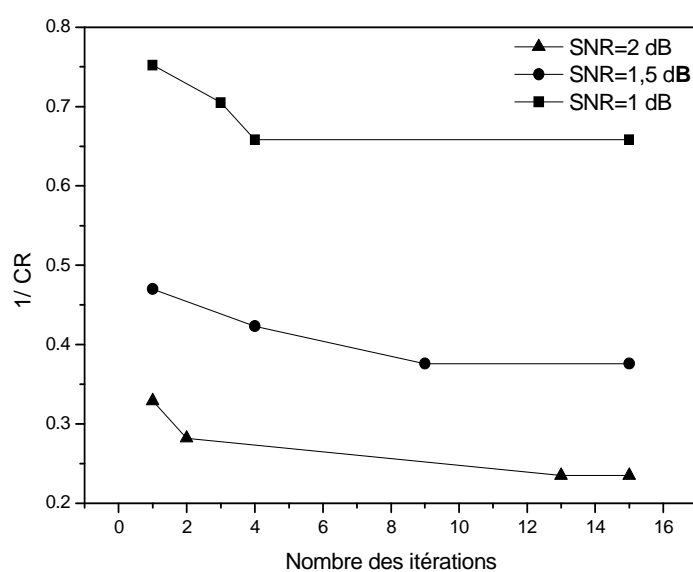


Figure : 3.1 Effets du nombre des itérations

L'augmentation du nombre des itérations à moins d'effet pour les faibles valeurs du SNR, par rapport aux valeurs élevées. Pour un SNR = 1 dB, lorsque le nombre des itérations augmente de 1 à 4, les performances du turbo décodeur s'améliorent considérablement (c'est à dire que le taux de la compression augmente), parce que les décodeurs 1 et 2 échange l'information entre eux pour prendre des décisions plus précises. Plus le nombre des itérations augmentent, plus les performances du turbo décodeur s'améliorent. Cependant, après que le nombre des itérations atteint une certaine valeur, l'amélioration sera moins considérable. Cela peut être expliqué par l'échange de la même information entre les décodeurs 1 et 2, l'augmentation du nombre des itérations n'apporte pas plus d'information. On peut conclure que 4 itérations sont suffisantes pour avoir des résultats raisonnables pour les valeurs faibles de SNR, et 13 itérations pour les valeurs élevées du SNR.

## 1.2 L'évaluation du changement du nombre des permutations sur l'efficacité du décodage

Trois permutations sont considérées dans la simulation représentée sur la Figure 3.2 : 500, 600 et 1000 permutations. Pour avoir plusieurs points, le canal AWGN est modélisé par les SNRs 1 dB, 1,5 dB, 2 dB, 2,5 dB, 3 dB et 3,5 dB respectivement, la séquence d'entrée  $X$  est de longueur  $L = 1000$ , le nombre des itérations du processus du décodage est fixé à  $N = 13$  (itérations suffisantes pour avoir des résultats raisonnables pour les valeurs faibles ou élevées de SNR, Figure 3.1).

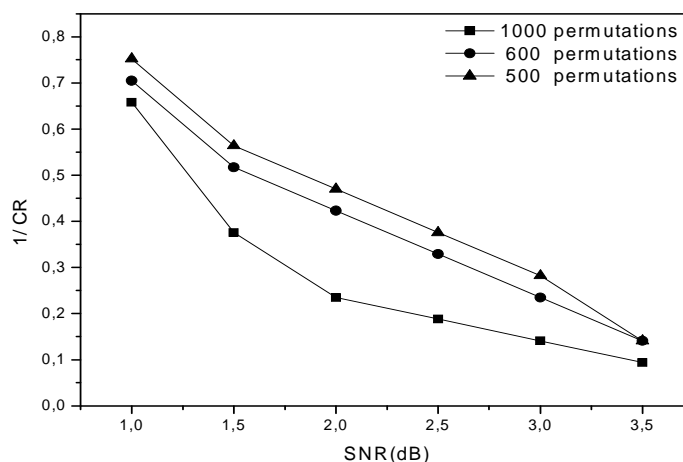


Figure 3.2 : Effets du nombre des permutations

Le nombre de permutations dans la séquence d'entrée du module d'entrelacement affecte également les performances du turbo code. Si on augmente le nombre de permutations, les performances de décodage augmentent. Comme Shannon la montré [59], les codes aléatoires avec des blocs à grandes longueurs peuvent donner un débit de transmission pratiquement égal à la capacité du canal. Par conséquent le nombre de permutations doit être choisi assez grand, par exemple 1000, puisque les petites valeurs peuvent impliquer un manque de l'aspect aléatoire requis pour atteindre une bonne performance pour les turbos codes.

### 1.3 L'évaluation du changement de la longueur $L$ sur l'efficacité de décodage

La Figure 3.3 montre les performances du turbo code poinçonné dans un canal AWGN, avec les SNRs 1 dB, 1,5 dB, 2 dB, 2.5 dB, 3 dB et 3.5 dB respectivement, le nombre des itérations est fixé à  $N=13$ , trois longueurs de la séquence d'entrée sont considérées dans la simulation :  $L = 1000$ ,  $L = 2000$  et  $L = 6336$  (la longueur de la séquence  $X$  dans le codeur vidéo distribué est  $L = 6336$ ) respectivement.

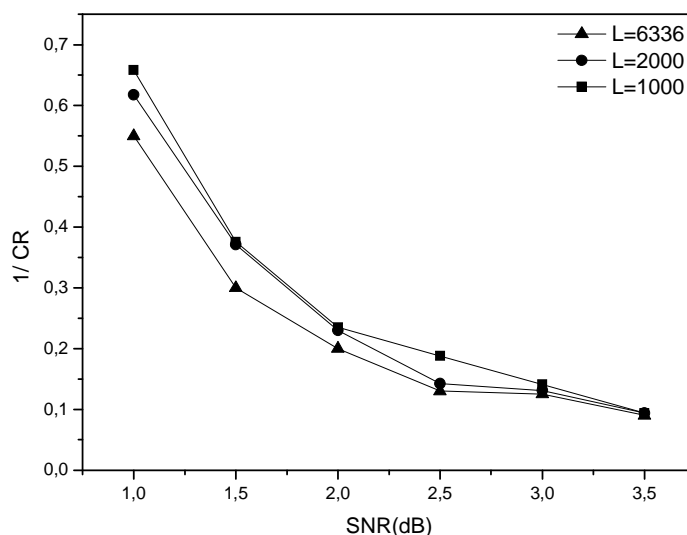


Figure 3.3 : Effets de la longueur  $L$

Si on augmente la longueur  $L$  de la séquence d'entrée, la corrélation entre les bits adjacents entrelacés sera diminuée, donc les performances de décodage augmentent.

Les résultats de la simulation vérifient cette conclusion. Cependant, puisque le turbo code est un bloc code, alors l'augmentation de la longueur des séquences va augmenter le temps de calcul pour accomplir le processus de décodage.

## **2. Résultats de simulation du CVD**

Dans l'évaluation des performances de la relation débit-distorsion du CVD, les séquences de format QCIF de Hall, Foreman et Soccer avec une fréquence de 15 HZ ont été employées. Dans le schéma de codage vidéo distribué, les 4x4 blocs WZ de l'image entière, à une étape donnée, sont groupés et codés ensemble avec une période de poinçonnage  $P = 21$ , d'autre part le nombre des itérations du processus de décodage  $N$  est fixé à 13 itérations. Pour une image QCIF avec 99 MB, à chaque étape on code 6336 pixels WZ. Pour le codage Intra, les blocs (Intra) des quatre images successives sont groupés ensemble, l'image résultante (complète) est ensuite introduite dans le codeur Intra H.264/AVC. Notant que dans l'évaluation des performances de la relation débit-distorsion, on n'a considéré que le débit-distorsion de la luminance, avec 4 points du taux de distorsion qui correspondent à  $M = 1, 2, 3$  et  $4$  respectivement. Les paramètres d'évaluation décrite dans cette section sont communs pour tous les tests d'évaluations du CVD.

### **2.1 L'évaluation du changement de facteur de la quantification Intra sur l'efficacité du CVD**

Les performances de la relation débit-distorsion du codeur vidéo distribué avec différents facteurs de quantification du codeur Intra H.264 sont examinées. Pour cela, Les 11 premières images de la séquence de Hall, Foreman et Soccer sont codées avec les facteurs de quantification :  $q=20$ ,  $q=24$ ,  $q=28$  et  $q=32$  respectivement [11]. Notant que le débit total est donné par la somme des débits WZ et Intra.

Pour mieux visualiser l'effet du changement de facteur de la quantification de codeur Intra H.264, les performances de la relation débit-distorsion totales des 11 premières images obtenues par la séquence de Hall, Foreman et Soccer sont superposées sur les trois figures suivantes.

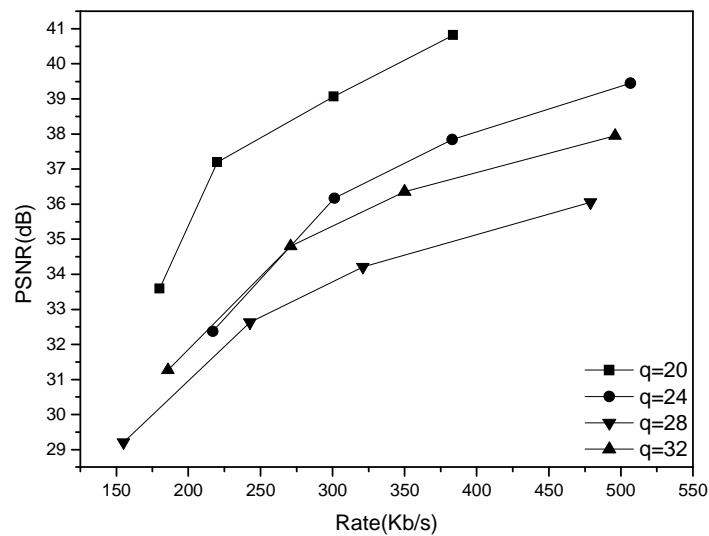


Figure 3.4 : Performance débit-distorsion total des 11 premières images de la séquence Hall QCIF, 15 Hz avec les facteurs de quantifications de codeur H.264  $q=20$ ,  $q=24$ ,  $q=28$  et  $q=32$  respectivement

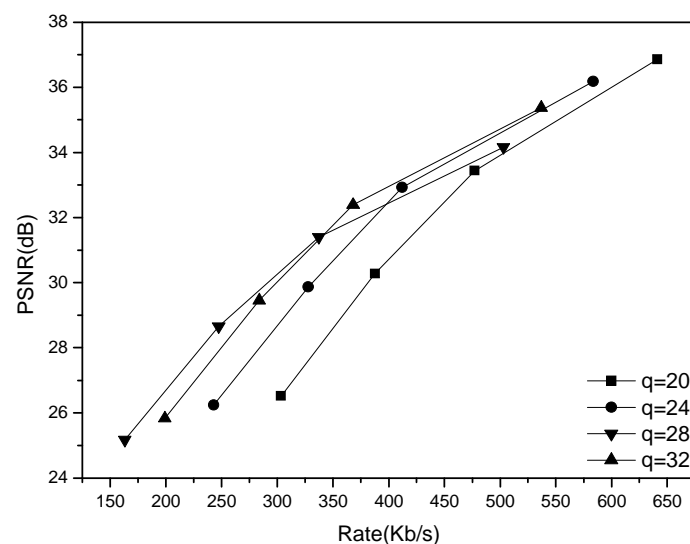


Figure 3.5 : Performance débit-distorsion total des 11 premières images de la séquence Foreman QCIF, 15 Hz avec les facteurs de quantifications de codeur H.264  $q=20$ ,  $q=24$ ,  $q=28$  et  $q=32$  respectivement

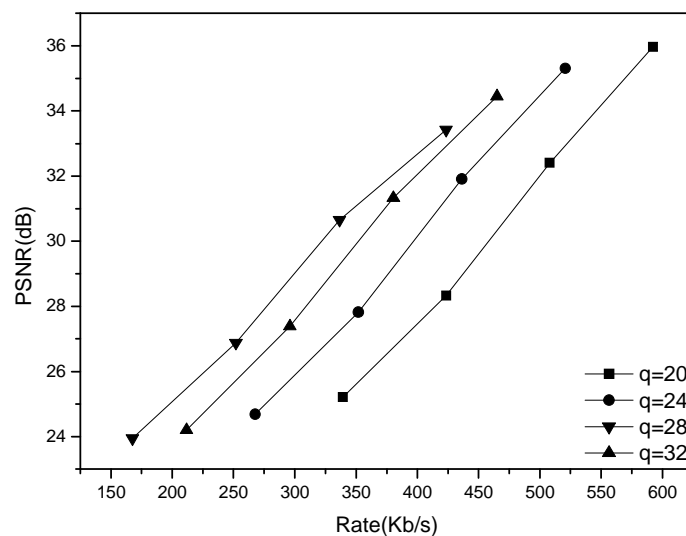


Figure 3.6 : Performance débit-distorsion total des 11 premières images de la séquence Soccer QCIF, 15 Hz avec les facteurs de quantifications de codeur H.264  $q=20$ ,  $q=24$ ,  $q=28$  et  $q=32$  respectivement

A partir des Figures 3.4, 3.5 et 3.6 on peut constater que plus le facteur de la quantification  $q$  augmente, plus les performances en termes de PSNR chutent, car il y a des erreurs de reconstruction (pertes d'information) qui sont dues à l'irréversibilité de l'opération de la quantification (voir la section 2.2.3).

Les pertes en PSNR des trois séquences Hall, Foreman et Soccer avec  $M=4, 3, 2$  et  $1$  pour  $q=24$ ,  $q=28$  et  $q=32$  par rapport à celui où  $q=20$ , sont regroupées dans les trois tableaux suivants.

Tableau 3.1 : Les pertes en PSNR en (dB) de la séquence Hall

Q=20	Q=24	Q=28	Q=32
M = 4	1,37	2,87	4,76
M = 3	1,23	2,71	4,86
M = 2	1,03	2,4	4,57
M = 1	1	2,33	4,39

Tableau 3.2 : Les pertes en PSNR en (dB) de la séquence Foreman

Q=20	Q=24	Q=28	Q=32
M = 4	0,68	1,49	2,7
M = 3	0,51	1,05	2,04
M = 2	0,41	0,82	1,62
M = 1	0,28	0,69	1,35

Tableau 3.3 : Les pertes en PSNR en (dB) de la séquence Soccer

Q=20	Q=24	Q=28	Q=32
M = 4	0,66	1,52	2,55
M = 3	0,53	1,08	1,76
M = 2	0,51	0,94	1,45
M = 1	0,50	0,81	1,27

On peut voir que les performances du PSNR de la séquence de Hall sont plus sensibles que celles de Foreman et Soccer à l'augmentation de facteur de la quantification. Par ce que la séquence de Hall contient beaucoup de textures et de détails par rapport aux séquences Foreman et Soccer qui englobent plusieurs régions continues. Donc, pour coder les séquences Foreman et Soccer on peut choisir par exemple un facteur de quantification  $q=24$  ou  $q=28$ , pour la séquence de Hall le facteur de la quantification doit être petit  $q=20$ .

D'autre part, l'augmentation du facteur de la quantification (pour les trois séquences) a moins d'effet (en PSNR) sur les débits faibles ( $M=1$ ) (Tableaux 3.1, 3.2 et 3.3), alors que la dégradation du PSNR pour les débits élevés est plus importante, parce que les plans de bits les moins significatifs (détails) qui contribuent le plus à la dégradation des performances.

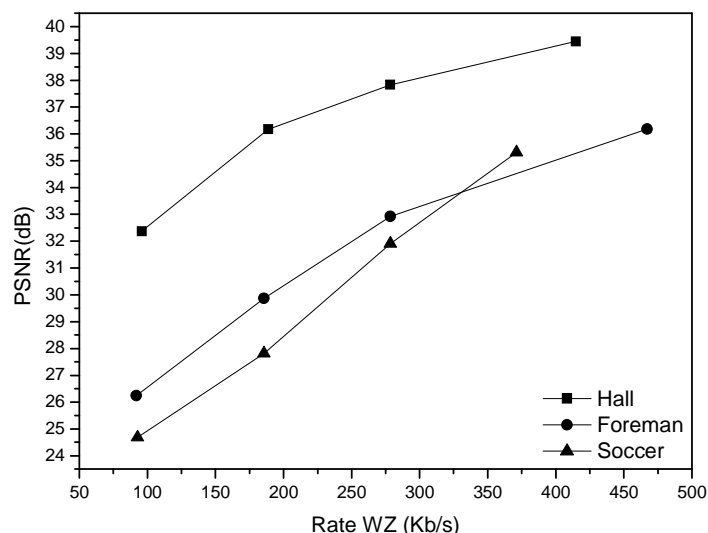


Figure 3.7 : Performance débit-distorsion WZ des 11 premières images des séquences Hall, Foreman et Soccer QCIF, 15 Hz avec un facteur de quantification de codeur H.264  $q=24$

Les performances de la relation débit-distorsion de WZ, des trois séquences sont également montrées Figure 3.7. Le débit WZ de Soccer et Foreman (pour un PSNR donné) est toujours supérieur au débit de la séquence de Hall Figure 3.7 (quelle que soit la valeur de  $q$ ); c'est à dire que les nombres de bits de parité envoyé au décodeur, pour corriger les erreurs dans les séquences de Soccer et Foreman, est plus grand que celui nécessaire pour décoder la séquence de Hall. Cela est dû au mouvement rapide et irrégulier qui caractérise les séquences de Soccer et Foreman, alors que la scène des 11 premières images de la séquence de Hall est presque fixe. Donc, les images de la séquence de Hall sont très corrélées, ce qui permet une bonne estimation de l'information de bord au décodeur, l'estimation et la compensation de mouvement dans les séquences de Soccer et Foreman est par contre moins efficace.

Pour évaluer les performances du codage Intra, les débits Intra ( $R_I$ ) ajoutés au débit WZ pour les trois séquences Hall, Foreman et Soccer avec les facteurs de la quantification  $q=20$ ,  $q=24$ ,  $q=28$  et  $q=32$  sont montrés sur les trois tableaux suivant.



Tableau 3.4 : Les débits Intra de la séquence Hall

$R_I$	q=20	q=24	q=28	q=32
M = 4	135,01	92,18	57	27,4
M = 3	144,86	104,55	72	43
M = 2	154,13	112,71	80	51
M = 1	162,29	121,14	87,01	59,1

Tableau 3.5 : Les débits Intra de la séquence Foreman

$R_I$	q=20	q=24	q=28	q=32
M = 4	176,85	116,39	72,79	36
M = 3	192,39	133,55	89,67	52,83
M = 2	201,38	141,98	98,1	61,82
M = 1	210,6	151	106,5	70,25

Tableau 3.6 : Les débits Intra de la séquence Soccer

$R_I$	q=20	q=24	q=28	q=32
M = 4	221,03	149,53	93,66	49,29
M = 3	229,47	157,97	102,1	58,01
M = 2	237,90	166,38	110,54	66,45
M = 1	246,34	174,84	118,98	74,88

On peut constater que le coût du codage Intra des séquences de Soccer et Foreman est plus grand que le coût de Hall. On note que la configuration de codeur H.264 implémentée dans notre CVD (les blocs Intra des 4 images successives sont groupés ensemble), ne fournit pas une bonne exploitation de la corrélation spatiale par rapport à une image normale. Le module de la TCD du codeur H.264, exploite mieux la redondance spatiale élevée (par rapport aux séquences Soccer et Foreman) dans la séquence de Hall pour diminuer le coût du codage Intra.

## 2.2 L'évaluation du changement de la longueur des séquences vidéo sur l'efficacité du codage

Dans notre deuxième test nous évaluons l'effet de l'augmentation de la longueur des séquences de Hall, Foreman et Soccer sur les performances de la relation débit-distorsion. Les 11 et les 101 premières images des trois séquences sont codées avec les facteurs de quantification  $q=20$ ,  $q=28$  et  $q=28$  respectivement. Les résultats obtenus sont montrés sur les figures suivantes.

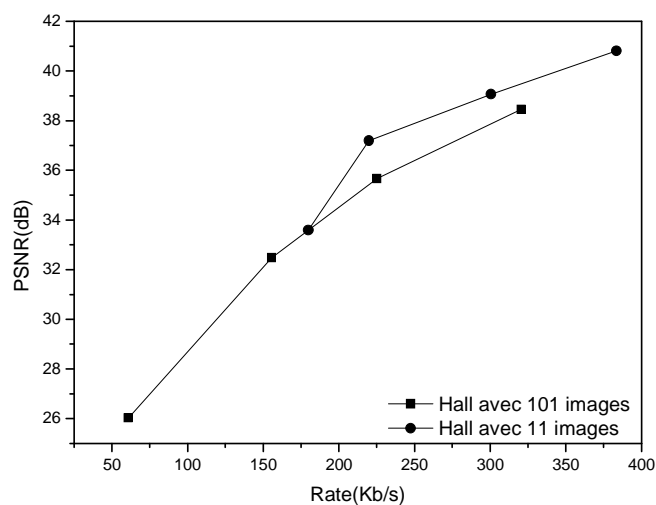


Figure 3.8 : Performance débit-distorsion total des 11 et 101 premières images de la séquence Hall QCIF, 15 Hz avec un facteur de quantification de codeur H.264  $q=20$

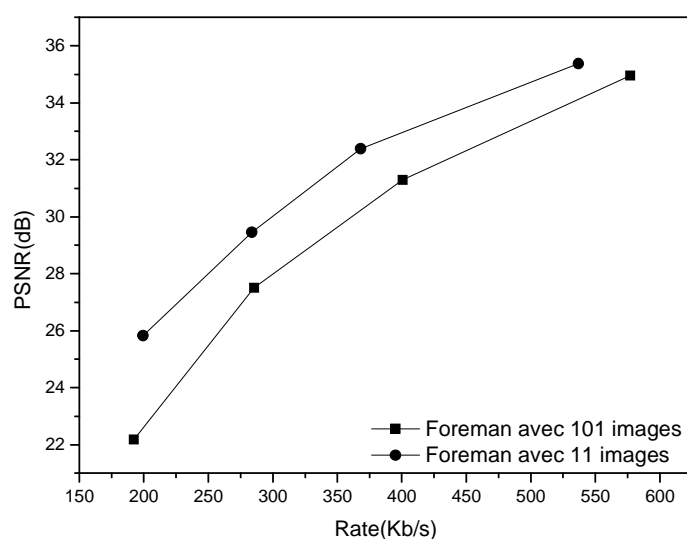


Figure 3.9 : Performance débit-distorsion total des 11 et 101 premières images de la séquence Foreman QCIF, 15 Hz avec un facteur de quantification H.264  $q=28$

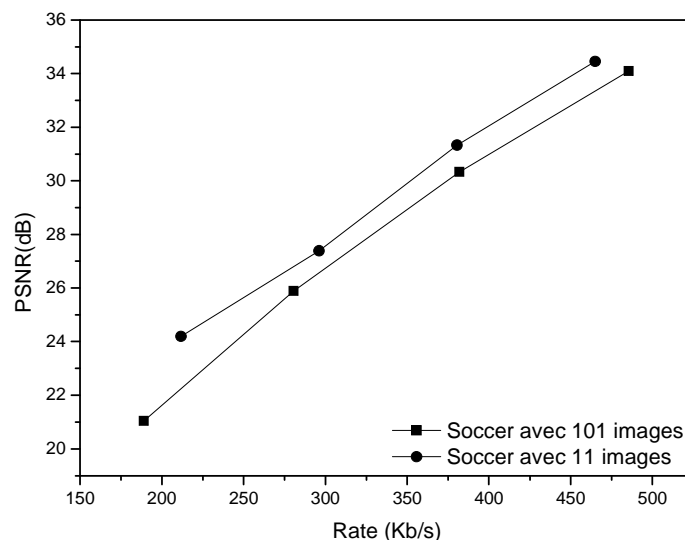


Figure 3.10 : Performance débit-distorsion total des 11 et 101 premières images de la séquence Soccer QCIF, 15 Hz avec un facteur de quantification de codeur H.264  $q=28$

Pour les trois séquences les performances de la relation débit-distorsion des 11 premières images sont les meilleures. Donc l'estimation du mouvement se dégrade quand la longueur des séquences augmente. Cela est dû d'une part à l'augmentation de l'activité et la longueur du mouvement au delà de l'image 10, 80 et 53 des séquences Hall, Foreman et Soccer respectivement, d'autre part il y a une sorte d'accumulation des erreurs, donc la dégradation de la qualité d'une image (PSNR) influe sur le reste des images.

### 2.3 Comparaison des performances de notre CVD par apport à la norme Intra H.264

Dans le troisième test nous comparons les performances de la relation débit-distorsion obtenus avec notre codeur vidéo distribué, et celle obtenue avec le codeur Intra H.264 standard, pour les 11 et les 101 premiers images de chaque séquence.

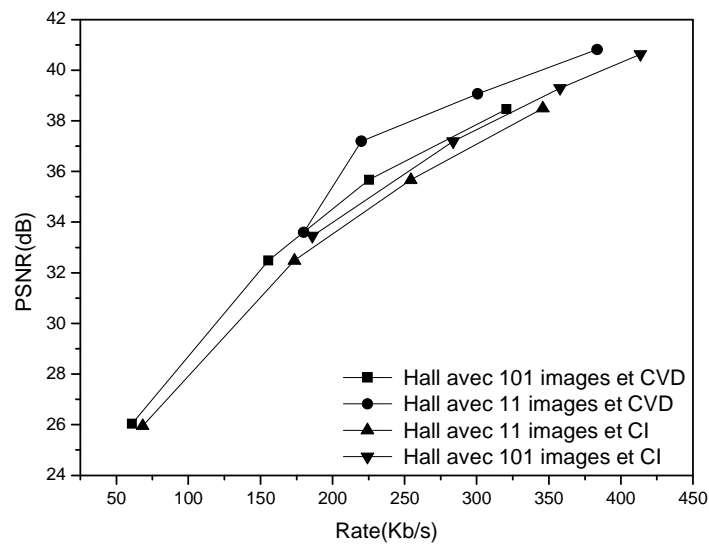


Figure 3.11 : Performance débit-distorsion total des 11 et 101 premières images de la séquence Hall QCIF, 15 Hz avec CVD ( $q=20$ ) et avec codage Intra H.264

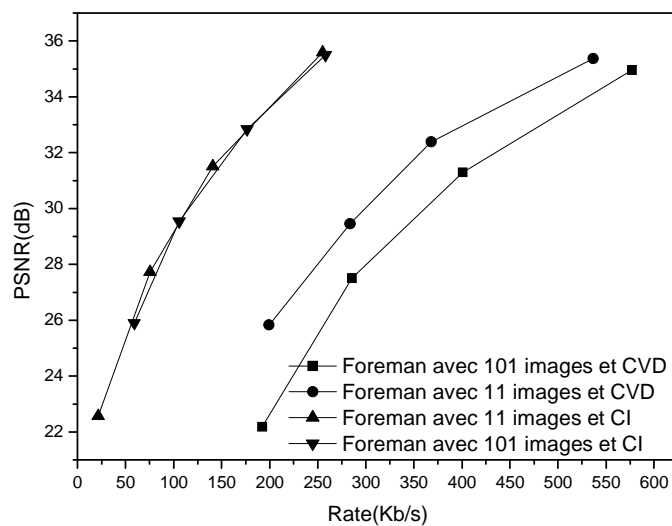


Figure 3.12 : Performance débit-distorsion total des 11 et 101 premières images de la séquence Foreman QCIF, 15 Hz avec CVD ( $q=28$ ) et avec codage Intra H.264

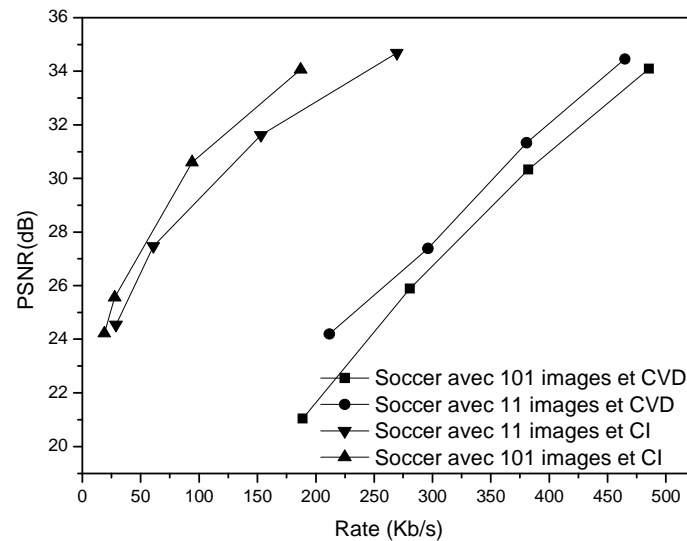


Figure 3.13 : Performance débit-distorsion total des 11 et 101 premières images de la séquence Soccer QCIF, 15 Hz avec CVD ( $q=28$ ) et avec codage Intra H.264

On peut voir que les performances de la relation débit-distorsion du CVD pour la séquence de Hall sont plus élevées que les performances du codeur H.264 Intra dont l'amélioration est d'environ 0.7 dB. Cependant, les performances de la relation débit-distorsion pour les séquences Soccer et Foreman du codeur H.264 Intra sont meilleures que celles obtenues par notre CVD.

Le gap entre les performances de notre CVD et le codeur Intra H.264 avec les séquences Soccer et Foreman est dû principalement à la faible efficacité de l'estimation du mouvement au décodeur, car la plupart des images des séquences Soccer et Foreman sont caractérisées par un mouvement rapide, long et complexe.

On peut conclure que lorsque l'efficacité de la compensation de mouvement augmente au décodeur, la qualité de reconstruction en termes de PSNR augmente et par conséquent le débit sera diminué.

## Conclusion

Dans ce chapitre, on a examiné les performances de turbo code et du codeur vidéo distribué implémentés dans le chapitre précédent.

Les résultats de simulation montre que le turbo code est un code correcteur d'erreur puissant dans un environnement bruité. Cependant, il y a plusieurs facteurs à considérer dans l'implémentation. On peut conclure que l'augmentation du nombre des itérations, ainsi que le nombre des permutations et la longueur de la séquence d'entrée  $X$  va améliorer les performances de turbo code, toutefois le temps de calcul augmente.

D'autre part l'évaluation du changement de facteur de la quantification Intra, et de la longueur des séquences vidéo sur l'efficacité du CVD, montre que plus le facteur de la quantification augmente, plus les performances en termes de PSNR chutent. De même que l'estimation du mouvement se dégrade quand la longueur des séquences vidéo augmente.

Les résultats obtenus dans la section 2.3 montrent que les performances de la relation débit-distorsion de notre CVD avec la séquence de Hall sont plus élevées que les performances du codeur H.264 Intra (0.7 dB). Alors que, lorsque l'efficacité de l'estimation de mouvement se dégrade au niveau de décodeur avec des mouvements rapides, longs ou complexes, les performances de la relation débit-distorsion du codeur H.264 Intra dépasse les performances de notre codeur pour la séquence Soccer et Foreman.

# Conclusion générale

Dans ce travail, nous nous sommes intéressés à l'étude du codage vidéo distribué. Ce type de codage concerne le cas des signaux fortement corrélés que l'on code séparément et décode conjointement. Le codage de sources distribuées a été considéré comme solution potentielle pour la compression de l'information dans des applications exigeant des encodeurs avec une faible complexité. En particulier, nous avons présenté l'intérêt de ce type de codage dans le domaine de la compression vidéo distribuée.

Dans un premier temps, on a commencé par donner des généralités sur les images et le signal vidéo. Ensuite, on a présenté une étude générale sur le principe de la compression vidéo. De plus, on a donné un aperçu sur les mesures des performances en particulier le taux de compression, l'entropie et la distorsion. Ensuite, on a présenté le principe fondamental du codage vidéo traditionnel ainsi que les efforts des deux organisations UIT-T et l'ISO/CEI.

Nous avons ensuite présenté les deux travaux de Dr D. Slepian et Dr J. K. Wolf qui démontrent que la limite d'une compression disjointe est la même que lors d'une compression conjointe. Pour approcher les limites théoriques suggérées par le théorème de Wyner-Ziv, l'évolution des arrangements du CVD le long des ces dernières années est présenté. Ensuite, on a étudié les architectures des deux groupes de recherche, qui sont de nos jours responsables des principaux développements des systèmes du codage vidéo distribué: le groupe de Bernd Girod (université de Stanford) et le groupe de Kannan Ramchandran à Berkeley (université de la Californie). Les résultats obtenus prouvent que le codage vidéo de Wyner-Ziv peut fournir des solutions du codage intéressantes pour quelques applications.

Dans le chapitre 2 de ce mémoire, nous avons implémenté un turbo code poinçonné pour l'exploiter dans notre codeur distribué, dont l'architecture générale est semblable à celle proposée par Aaron et al dans [36], mais il y a cependant quelques différences essentielles dont a parlé dans le chapitre 2.

Dans le chapitre 3, les performances du turbo code et du codeur vidéo distribué sont évaluées par simulation. Les résultats de simulation montrent que le turbo code est un code correcteur d'erreur puissant dans un environnement bruité. L'augmentation du facteur de la quantification Intra, et de la longueur des séquences vidéo dans le CVD dégrade les performances en PSNR, ainsi que l'estimation du mouvement. D'autre part, les performances de la relation débit-distorsion de notre CVD pour la séquence de Hall sont plus élevées que les performances du codeur H.264 Intra dont l'amélioration est de 0.7 dB. Alors que, lorsque l'efficacité de l'estimation de mouvement se dégrade au niveau de décodeur avec des mouvements rapides, longs ou complexes, les performances de la relation débit-distorsion pour les séquences Soccer et Foreman du codeur H.264 Intra sont meilleures à celle obtenue par le CVD.

En perspectives de ce travail, il serait intéressant de chercher des quantificateurs plus adaptés au contexte du codage vidéo distribué. Il serait bon également de :

- Tenter d'améliorer l'information de bord : ce qui veut dire l'amélioration de l'estimation du mouvement et de la prédiction au niveau du décodeur. Si une bonne prédiction est possible le nombre de bits de parité à transmettre peut être réduit.
- Amélioration du codage canal : adapter la demande de bits de parités au décodeur selon les variances du bruit de corrélation entre les blocks décodés et l'information de bord. Il faut aussi un effort pour la modélisation du canal virtuel ; c'est ce modèle qui détermine les probabilités utilisées par le décodeur du code canal. Donc, son influence sur le résultat est importante puisque un bon modèle peut aider à corriger les erreurs rapidement ce qui réduit le débit et améliore la qualité.
- Essayer d'exploiter d'autres informations au niveau du codeur pour aider le décodeur dans sa tâche.



---

# Bibliographies

- [1] UIT-T : Union Internationale des Télécommunication <http://www.itu.ch>.
- [2] MPEG: Motion Picture Expert Group, <http://www.mpeg.org>.
- [3] R. Puri and K. Ramchandran, «PRISM: A Video Coding Architecture Based on Distributed Compression Principles», ERL Technical Report, University of California, Berkeley, USA, March 2003.
- [4] A. Aaron, E. Setton and B. Girod, «Towards Practical Wyner-Ziv Coding of Video», Proceedings of IEEE International Conference on Image Processing, Vol. 3, pp. 869-872, Barcelona, Spain, September 2003.
- [5] J. Slepian and J. Wolf, «Noiseless Coding of Correlated Information Sources», IEEE Transactions on Information Theory, Vol. 19, No. 4, pp. 471-480, July 1973.
- [6] A. Wyner and J. Ziv, «The Rate-Distortion Function for Source Coding with Side Information at the Decoder», IEEE Transactions on Information Theory, Vol. 22, No. 1, pp. 1-10, January 1976.
- [7] E. Incerti, «Compression d'images : Algorithmes et Standards», Vuibert, Paris, 2003.
- [8] G. Côté and L. Winger, «Progrès Récents dans le Domaine de la Compression Vidéo», IEEE Canadian Review Printemps 2002.
- [9] H.261: Video Codec for Audiovisual Services at p x 64 kb/s. Recommendation H.261 à l'UIT-T. Mars 1993.
- [10] H.263: Video Coding for Low Bit rate Communication. Mars 1996.
- [11] F.Loras, «H.264/MPEG-4 AVC, un Nouveau Standard de Compression Vidéo», CORES A'03, Lyon, France, 16-17 Janvier 2003.
- [12] Standard MPEG-1 : ISO/IEC 11172-2, Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1,5 Mb/s.

- 
- [13] Standard MPEG-2: ISO/IEC 13818-2, Information Technology - Generic Coding of Moving Pictures and Associated Audio Information.
- [14] Standard MPEG-4: ISO/IEC 14496-2, Information Technology - Coding of Audio-Visual Objects.
- [15] A. Wyner, «Recent Results in the Shannon Theory», IEEE Transactions on Information Theory, Vol.20, No. 1, pp2-10, January 1974.
- [16] A. Aaron, S. Rane, E. Setton and B. Girod, «Transform-Domain Wyner-Ziv Codec for Video», Proceedings of SPIE Visual Communications and Image Processing Conference, San Jose, California, USA, January 2004.
- [17] S. Pradhan and K. Ramchandran, «Enhancing Analog Image Transmission Systems Using Digital Side Information: A New Wavelet-Based Image Coding Paradigm», Proceedings of IEEE Data Compression Conference, pp.63-72, Snowbird, Utah, USA, March 2001.
- [18] D. Rebollo-Monedero, A. Aaron and B. Girod, «Transforms for High-Rate Distributed Source Coding», Proceedings of Asilomar Conference on Signals, Systems and Computers, Pacific Grove, California, USA, November 2003.
- [19] T. Flynn and R. Gray, «Encoding of Correlated Observations», IEEE Transactions on Information Theory, Vol. 33, No. 6, pp. 773-787, November 1987.
- [20] S. Shamai, S. Verdu and R. Zamir, «Systematic Lossy Source/Channel Coding», IEEE Transactions on Information Theory, Vol. 44, No. 2, pp. 564-579, March 1998.
- [21] S. Servetto, «Lattice Quantization with Side Information», Proceedings of IEEE Data Compression Conference, pp. 510-519, Snowbird, Utah, USA, March 2000.
- [22] M. Fleming and M. Effros, «Network Vector Quantization», Proceedings of IEEE Data Compression Conference, pp. 13-22, Snowbird, Utah, USA, March 2001.
- [23] D. Muresan and M. Effros, «Quantization as Histogram Segmentation: Globally Optimal Scalar Quantizer Design in Network Systems», Proceedings of IEEE

- Data Compression Conference, pp. 302-311, Snowbird, Utah, USA, April 2002.
- [24] D. Rebollo-Monedero, R. Zhang and B. Girod, «Design of Optimal Quantizers for Distributed Source Coding», Proceedings of IEEE Data Compression Conference, pp. 13-22, Snowbird, Utah, USA, March 2003.
- [25] S. Pradhan and K. Ramchandran, «Distributed Source Coding Using Syndromes «DISCUS»: Design and Construction», Proceedings of IEEE Data Compression Conference, pp. 158-167, Snowbird, Utah, USA, March 1999.
- [26] X. Wang and M. Orchard, «Design of Trellis Codes for Source Coding with Side Information at the Decoder», Proceedings of IEEE Data Compression Conference, pp. 361-370, Snowbird, Utah, USA, March 2001.
- [27] J.Garcia-Frias, «Compression of Correlated Binary Sources Using Turbo Codes», IEEE Communications Letters, Vol. 5, No. 10, pp. 417-419, October 2001.
- [28] J.Garcia-Frias and Y. Zhao, «Data Compression of Unknown Single and Correlated Binary Sources Using Punctured Turbo Codes», Proceedings of 39th Allerton Conference on Communication, Control and Computing, Monticello, Illinois, USA, October 2001.
- [29] J. Bajcsy and P. Mitran, «Coding for the Slepian-Wolf Problem with Turbo Codes», Proceedings of IEEE Global Telecommunications Conference, Vol. 2, pp. 1400-1404, San Antonio, Texas, USA, November 2001.
- [30] P.Mitran and J. Bajcsy, «Near Shannon-Limit Coding for the Slepian-Wolf Problem», Proceedings of 21st Biennial Symposium on Communications, Kingston, Ontario, Canada, June 2002.
- [31] A. Aaron and B. Girod, «Compression with Side Information Using Turbo Codes», Proceedings of IEEE Data Compression Conference, pp. 252-261, Snowbird, Utah, USA, April 2002.
- [32] A. Livens, Z. Xiong and C. Georghiades, «Distributed Compression of Binary Sources Using Conventional Parallel and Serial Concatenated Convolutional Codes», Proceedings of IEEE Data Compression Conference, pp. 193-202, Snowbird, Utah, USA, March 2003.

- 
- [33] V. Stankovic, A. Liveris, Z. Xiong and C. Georghiades, «Design of Slepian-Wolf Codes by Channel Code Partitioning», Proceedings of IEEE Data Compression Conference, pp. 302-311, Snowbird, Utah, USA, March 2004.
- [34] D. Schonberg, S. Pradhan and K. Ramchandran, «Distributed Code Constructions for the Entire Slepian-Wolf Rate Region for Arbitrarily Correlated Sources», Proceedings of IEEE Data Compression Conference, pp. 292-301, Snowbird, Utah, USA, March 2004.
- [35] T. Coleman, A. Lee, M. Medard and M. Effros, «On Some New Approaches to Practical Slepian-Wolf Compression Inspired by Channel Coding», Proceedings of IEEE Data Compression Conference, pp. 282-291, Snowbird, Utah, USA, March 2004.
- [36] A. Aaron, R. Zhang and B. Girod, «Wyner-Ziv Coding for Motion Video», Proceedings of Asilomar Conference on Signals, Systems and Computers, Pacific Grove, California, USA, November 2002.
- [37] R. Puri and K. Ramchandran, «PRISM: A New Robust Video Coding Architecture Based on Distributed Compression Principles», Proceedings of 40<sup>th</sup> Allerton Conference on Communication, Control and Computing, Allerton, Illinois, USA, October 2002.
- [38] X. Zhu, A. Aaron and B. Girod, «Distributed Compression for Large Camera Arrays», IEEE Workshop on Statistical Signal Processing, pp. 30-33, St. Louis, Missouri, USA, September 2003.
- [39] A. Aaron, S. Rane and B. Girod, «Wyner-Ziv Video Coding with Hash-Based Motion Compensation at the Receiver», Proceedings of IEEE International Conference on Image Processing, Vol. 5, pp. 3097-3100, Singapore, October 2004.
- [40] S. Rane, A. Aaron and B. Girod, «Systematic Lossy Forward Error Protection for Error-Resilient Digital Video Broadcasting», Proceedings of SPIE Visual Communications and Image Processing Conference, San Jose, California, USA, January 2004.
- [41] A. Aaron, S. Rane, R. Zhang, and B. Girod, «Wyner-Ziv coding for video: Applications to compression and error resilience», Proceedings IEEE Data

- Compression Conference «DCC», pp. 93-102, Snowbird, USA, March 2003.
- [42] A. Aaron, E. Setton, and B. Girod, « Towards practical Wyner-Ziv coding of video», Proc. IEEE International Conference on Image Processing, Vol.3 , pp. 869-872, Barcelona, Spain, September 2003.
- [43] A. Aaron, S. Rane, D. ReboUo-Monedero, and B. Girod, « Systematic lossy forward error protection for video waveforms», Proc. IEEE International Conference on Image Processing, Vol. 1, pp. 609-612, Barcelona, Spain, September 2003.
- [44] A. Sehgal and N. Ahuja,«Robust predictive coding and the Wyner-Ziv problem», Proc. IEEE Data Compression Conference «DCC», pp. 103-112, March 2003.
- [45] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, « Distributed video coding» Proc. IEEE, Special issue on advances in video coding and delivery, vol. 93, No. 1, pp. 71-83, Invited paper, January. 2005.
- [46] A. J. Viterbi, «Error bounds for convolutional codes and an asymptotically optimum decoding algorithm», IEEE Trans. Inform. Theory, Vol. IT-13, pp. 260-269, April 1967.
- [47] G. D. Forney, «The Viterbi algorithm IEEE Proceedings», Vol. 61, pp.268-278, March 1973.
- [48] C.Berrou, A. Glavieux and P. Thitimajshima, «Near Shannon limit error-correcting coding and decoding», IEEE Int. International Conference on Communications. ICC'93, vol 2/3. May 1993, pp. 1067-1071.
- [49] C. Berrou, A. Glavieux and P. Thitimajshima, «Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes (1) », IEEE International Conference on Communications, Vol. 2, pp. 1064-1070, Geneva, Switzerland, June 1993.
- [50] L. R. Bahl, J. Cocke, F. Jeinek and J. Raviv, «Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate», IEEE Transactions on Information Theory, Vol. 20, No. 2, pp. 248-287, March 1974.
- [51] B. Vucetic and J. Yuan, «Turbo Codes Principles and Applications», Kluwer Academic Publishers, USA, 2000.

- 
- [52] D. Rowitch and L. Milstein, «On the Performance of Hybrid FEC/ARQ Systems using Rate Compatible Punctured Turbo Codes», IEEE Transactions on Communications, Vol. 48, No. 6, pp. 948-959, June 2000.
- [53] W. E. Ryan, «A Turbo Code Tutorial», New Mexico State University.
- [54] A. Trapanese, M. Tagliasacchi, S. Tubaro, J. Ascenso, C. Brites, and F. Pereira, «Embedding a block-based intra mode in frame-based pixel domain wyner-ziv video coding», in International Workshop on Very Low Bitrate Video Coding, Costa del Rei, Sardinia, Italy, September 2005.
- [55] Chun-Ho Cheung and Lai-Man, «Generalized Partial Distortion Search Algorithm for Fast Block Motion Estimation» journal or conference 2005.
- [56] A. B. Carlson, «Communication Systems», McGraw-Hill, 2000.
- [57] Ascenso, J., Brites, C, Pereira, F, «Motion Compensated Refinement for Low Complexity Pixel Based Distributed Video Coding», IEEE International Conference on Advanced Video and Signal Based Surveillance, Como, Italy, September 2005.
- [58] P.A.Regalia, «Iterative Decoding of Concatenated Codes: A Tutorial», EURASIP Journal on Applied Signal Processing, Vol. 2005, No. 6, pp. 762-774, June 2005.
- [59] C. E. Shannon, «A Mathematical Theory of Communication», Bell System Technical Journal, Vol. 27, pp. 379-423, 623-656, July, October, 1948.