

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Mentouri Constantine 1  
Faculté des Sciences de la technologie  
Département d'Electronique

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

**Option:** Traitement de Signal

**Présenté par :**

Laraba Mossaab

**Thème:**

Codage vidéo distribué des séquences vidéo  
périodiques

**Devant la commission de jury :**

Encadreur : Mohammed Khamadja

Prof Univ. Constantine.

Président : Abdelfatah Charef

Prof Univ. Constantine.

Examineur: Toufik Laroussi

Prof Univ. Constantine.

Examineur: Faouzi Soutani

Prof Univ. Constantine.

Membre invité : Said Benierbah

MCA Univ. Constantine.

Promotion 2013

## **Remerciements :**

*Louange à Allah, Seigneur de l'univers, Le Tout Miséricordieux, Le Très Miséricordieux, qui nous a permis d'accomplir ce modeste travail.*

*Tout d'abord je remercie vivement mon directeur de mémoire, Le professeur Mohammed KHAMADJA d'avoir dirigé ce travail, et pour ses précieux et judicieux conseils.*

*Ensuite, Le Docteur Said BENI ERBAH pour m'avoir codirigé. Je tiens à lui exprimer ma gratitude et ma reconnaissance pour l'aide précieuse qu'il m'a apporté, tout le long de ce travail.*

*J'adresse aussi mes remerciements aux trois professeurs, Abdel fatah CHAREF et Toufik Laroussi et Faouzi Soul tani pour avoir accepté d'être membres de mon jury.*

*A tous les enseignants qui ont participé à ma formation.*

# *\*\*\*\*Dédicace\*\*\*\**

*Je dédie ce Travail à :*

*Mes chers parents pour leur sacrifice et  
encouragement.*

*Mes frères, Housseem, Mohamed, Mostafa, et ma  
petite sœur Amira.*

*Toute ma famille.*

*Tous mes amis.*

*Mossaab.*

## Résumé :

Le codage vidéo distribué (CVD) est devenu un domaine de recherche très intéressant, à cause de sa simplicité et ses résultats prometteurs, mais il y a encore de nombreux problèmes qui attendent d'être résolus. Dans ce mémoire nous allons concentrer notre travail sur l'application du CVD sur des séquences vidéo périodiques, où nous allons proposer une nouvelle technique de génération de l'information adjacente adaptée spécifiquement pour ce type de séquences vidéo. Cette technique est très simple contrairement aux méthodes de génération de l'information adjacente utilisées dans l'état de l'art tels que : l'interpolation et l'extrapolation, qui sont très complexes puisque elles sont basées sur l'estimation de mouvement. Les résultats obtenus par notre codeur CVD sont très prometteurs, et montrent que malgré la simplicité de cette méthode, elle reste très efficace.

**Mots clés :** CVD, séquences vidéo périodiques, information adjacente.

## ملخص :

عملية ضغط الفيديو بطريقة موزعة أصبحت مجال بحث مثير للاهتمام، نظرا لبساطتها و نتائجها الواعدة، لكن لا يزال هناك العديد من المشاكل في انتظار حلها. في هذه المذكرة سوف نتطرق إلى تطبيق عملية ضغط الفيديو بطريقة موزعة على مقاطع فيديو دورية، أين سنقترح تقنية جديدة لتوليد المعلومة الجانبية. أعدت خصيصا لتطبيقها على هذا النوع من مقاطع الفيديو. هذه التقنية غير معقدة و سهلة التطبيق مقارنة مع التقنيات الموجودة في المراجع مثل: الاستيفاء و الاستقراء، التي تعتبر معقدة جدا باعتبار أنها تعتمد بصفة كبيرة على تقدير الحركة. النتائج المتحصل عليها باستعمال هذه التقنية هي نتائج واعدة جدا وتبرز انه رغم بساطتها إلا أنها تبقى فعالة جدا.

## Summary :

Distributed Video Coding (DVC) has become a very interesting field of research, because of its simplicity and promising results, but there are still many problems waiting to be solved. In this thesis, we focus our study on the Application of DVC on periodic video sequences. We will introduce a new way of side information creation, suited for this type of video sequences. This solution is very simple unlike other techniques used in state of the art for side information creation such as interpolation and extrapolation, which are very complex because they are based on motion estimation. The obtained results show that although this technique is very simple, but it remains very effective.

**Key words:** DVC, periodic video sequences, side information.

## Table des matières

<b>Introduction générale</b> .....	1
<b>Chapitre 1 : Etat de l'art du codage vidéo distribué</b> .....	<b>3</b>
<b>Introduction</b> .....	4
<b>1. Notion de base de la théorie de l'information</b> .....	4
1.1. Quantité d'information .....	4
1.2. L'entropie .....	5
1.3. Entropie conjointe et conditionnelle .....	5
1.4. Codage entropique (codage statistique à longueur variable).....	6
<b>2. Les Codes LDPC</b> .....	7
2.1. Représentations d'un code LDPC .....	7
2.1.1. Représentation matricielle .....	7
2.1.2. Représentation graphique .....	7
2.2. Procédure de codage et décodage d'un code LDPC .....	8
2.2.1. Opération de codage .....	8
2.2.2. Opération de décodage .....	8
<b>3. Codage de source distribué (CSD)</b> .....	9
3.1. Théorème de Slepian-Wolf .....	11
3.2. Théorème de Wyner-Ziv .....	13
3.2.1. Principe du codage Wyner – Ziv .....	14
<b>4. Le codage vidéo</b> .....	15
4.1. Eléments de base .....	15
4.1.1. La vidéo numérique .....	15
4.1.2 La compression vidéo .....	16
4.1.3. Codage vidéo traditionnel (La norme H.264/ AVC) .....	17
4.2. Codage vidéo distribué .....	19
4.2.1. Présentation des Systèmes CVD .....	19
4.2.2. Les solutions du CVD les plus intéressantes .....	22

1. L'information adjacente (Side information).....	23
1.1. Définition .....	23
1.2. Différentes techniques de Génération de l'information adjacente .....	23
1.2.1. Estimation de l'information adjacente par l'interpolation .....	23
1.2.1.1 Estimation de mouvement bidirectionnelle avec lissage spatial.....	24
1.2.1.2. Estimation de mouvement bidirectionnelle non linéaire avec plusieurs Images de référence.....	26
1.2.2. Estimation de l'information adjacente par l'extrapolation .....	27
1.2.2.1. Estimation du mouvement avec Lissage spatial du mouvement .....	27
2. Les modèles proposés dans la littérature .....	29
1. Le modèle CVD proposé par l'équipe de Stanford .....	29
1.1 La procédure de codage .....	29
1.2. La procédure de décodage .....	30
2. Le modèle CVD proposé par l'équipe de Berkeley (PRISM) .....	32
2.1. La procédure de codage .....	33
2.2. La procédure de décodage .....	34
3. DISCOVER Wyner-Ziv video codec .....	36
3.1. La procédure de codage .....	36
3.2. La procédure de décodage .....	37
<b>5. Conclusion .....</b>	<b>38</b>

## **Chapitre 2 :**

<b>Implémentation d'un CVD pour séquences vidéo périodiques.....</b>	<b>39</b>
<b>Introduction .....</b>	<b>40</b>
<b>1. Le mouvement périodique .....</b>	<b>40</b>
1.1. Définition .....	40
1.2. Exemples des phénomènes périodiques .....	40
1.2.1. Le Battement du cœur (rythme cardiaque) .....	40

1.2.2. Camera de surveillance .....	41
1.2.3. Oscillation d'un pendule .....	41
1.2.4. Satellite artificiel .....	42
1.2.5. D'autres exemples du phénomène périodique .....	42
<b>2. Présentation d'un CVD pour des séquences périodique .....</b>	<b>43</b>
<b>3. Idée de base de notre travaille .....</b>	<b>44</b>
<b>4. La procédure de codage .....</b>	<b>46</b>
4.1. Transformation DCT .....	47
4.2. Quantification .....	47
4.3 Conversion en plan de bits .....	49
4.4. Codage de canal (Codeur LDPC et Buffer) .....	50
<b>5. La procédure de décodage .....</b>	<b>50</b>
5.1 Génération de l'information adjacente .....	50
5.2 Modélisation du modèle de bruit .....	50
5.2.1. Estimation du paramètre $\alpha$ .....	51
5.3. Décodage LDPC (correction) .....	51
5.4 Quantificateur inverse et reconstruction .....	52
5.5. DCT Inverse .....	53
<b>6. Conclusion .....</b>	<b>54</b>

## **Chapitre 3 :**

<b>Résultats et discussion.....</b>	<b>55</b>
-------------------------------------	-----------

<b>Introduction .....</b>	<b>56</b>
---------------------------	-----------

<b>1. L'évaluation de la séquence de test du train .....</b>	<b>57</b>
--	-----------

1.1. La séquence de test du train .....	58
---	----

1.2. Evaluation des résultats .....	58
-------------------------------------	----

<b>2. Evaluation de la séquence de la grande muraille de chine .....</b>	<b>59</b>
--	-----------

2.1. Caractéristique de la séquence la grande muraille .....	59
--	----

2.2. Evaluation des résultats .....	60
<b>3. Evaluation de la séquence de l'Electrocardiogramme (ECG) .....</b>	<b>61</b>
3.1. Caractéristique de la séquence de test de l'ECG .....	61
3.2. Evaluation des résultats .....	62
<b>4. Conclusion .....</b>	<b>63</b>
<b>Conclusion générale .....</b>	<b>64</b>



## Liste des Figures :

1.1. Graphe de Tanner pour l'exemple de matrice de contrôle de parité (1.7).....	8
1.2. Codage de source distribué avec plusieurs séquences vidéo corrélés. ....	10
1.3 Codage distribué de deux sources statistiquement corrélées et i.i.d, X et Y.....	11
1.4 - Illustration du théorème de Slepian-Wolf pour deux sources.....	12
1.5. Relation entre le codage de canal et le codage de Slepian-Wolf.....	13
1.6. Le codage WZ de X en utilisant l'information adjacente Y disponible au niveau du décodeur seulement. ....	13
1.7 - Structure de codage de base du H.264/AVC .....	19
1.8 - Codeurs vidéo avec complexité limitée. téléphone portable, caméra sans fil, capsule endoscopique, endoscope sans fil. ....	20
1.9. Prédiction de l'image K en utilisant la technique de l'interpolation.....	24
1.10. L'architecture d'estimation par interpolation .....	25
1.11. Estimation du mouvement bidirectionnel .....	25
1.12. Illustration de la méthode (HOMI).....	26
1.13. Prédiction de l'image K en utilisant la technique de l'Extrapolation.....	27
1.14 architecture d'extrapolation proposée par Natário et al. ....	28
1.15. Projection du mouvement .....	28
1.16. Architecture du codeur CVD de l'équipe Stanford .....	29
1.17. Le zig-zag scan des coefficients DCT .....	30
1.18. Compensation de mouvement par interpolation.....	31
1.19. Architecture du codeur PRISM .....	32
1.20. Codage par syndrome et codage entropique dans un seul bloc de $8 \times 8$ .....	34
1.21. Architecture du décodeur PRISM.....	35
1.22. Le CODEC DISCOVER .....	36
2.1. Rythme cardiaque d'un Homme en bonne santé.....	40
2.2. Camera de surveillance qui filme des séquences vidéo périodique.....	41
2.3. Le mouvement périodique réalisé par l'oscillation du pendule.....	41
2.4. Rotation d'un satellite autour de la terre.....	42
2.5. Exemples des phénomènes périodiques.....	43

2.6. Exploitation de l'information adjacente pour le codage de la séquence vidéo périodique. ....	45
2.7. L'architecture de codage et décodage CVD proposé par notre approche.....	46
2.8. Quantificateur à zone morte (dead-zone quantizer).....	48
2.9 Huit matrices de quantifications associées à des différentes performances de distorsion de débit .....	49
3.1. Illustration du mouvement périodique réalisé par le train.....	57
3.2. Performance débit-distorsion de la séquence du train .....	59
3.3. Illustration de La séquence de la grande muraille (seulement la première période). ....	59
3.4. Performance débit-distorsion de la séquence de la grande muraille. ....	60
3.5. Illustration de La séquence de L'ECG. ....	61
3.6. Performance débit-distorsion de la séquence de L'ECG.....	62

## Liste des Tableaux :

Tableau 1.1. Exemple d'un code VLC pour $M = 4$ .....	6
Tableau 3.1 les caractéristiques de la séquence du train. ....	58
Tableau 3.2 les caractéristiques de la séquence de la grande muraille.....	60
Tableau 3.2 les caractéristiques de la séquence de L'ECG.. ....	62

## **Introduction générale :**

Depuis les dix dernières années nous assistons à une métamorphose dans la façon dont nous communiquons, les médias numériques sont devenus une partie intégrante de notre mode de vie. Les progrès spectaculaires dans le domaine informatique et les technologies de communication ont conduit à une prolifération des médias numériques et de leur intégration dans les dispositifs et les activités quotidiennes.

Aujourd'hui, nous assistons à une évolution vers la production et le partage de vidéos pour les applications en temps réel, tels que la vidéoconférence sur des réseaux sans fil / cellulaires, la vidéosurveillance, caméras PC sans fil, caméras des téléphones mobiles et beaucoup d'autres. Ces applications disposent de ressources limitées, de faible puissance, contrairement au serveur central, qui est généralement très puissant. Dans ce cas, un codeur vidéo de faible complexité est nécessaire (même au détriment d'un décodeur complexe).

Afin de répondre à ces exigences, un nouveau modèle de codage vidéo a été développé dans les dix dernières années, sous le nom de codage vidéo distribué (CVD). Contrairement au codage traditionnel, ce modèle bascule la complexité du codage vers le décodeur, en utilisant les principes de codage de source avec perte contrôlée, en exploitant l'information adjacente (side information) au niveau du décodeur. Il est également connu sous le nom de codage Wyner-Ziv ((WZ) coding ). Ce modèle de codage est basé sur deux théorèmes principaux, Slepian and Wolf [1] pour le cas sans perte et WZ [2] pour le cas avec perte.

La grande problématique dans le cadre CVD, c'est comment générer l'information adjacente au niveau du décodeur car c'est elle qui contrôle la qualité totale du codage. Alors dans ce mémoire, nous allons proposer une nouvelle technique de génération de l'information adjacente adaptée spécifiquement pour des séquences vidéo périodiques.

Différemment de ce qui est présenté dans la littérature, où l'information adjacente est créée après le codage, dans notre cas l'information adjacente est créée avant la procédure de codage et de décodage. Ceci constitue l'originalité et l'apport de notre travail au codage vidéo distribué.

## **Organisation du mémoire :**

Ce mémoire décrit en détail une nouvelle approche CVD basée sur une nouvelle technique de génération de l'information adjacente. Ce modèle est développé précisément pour l'appliquer sur des séquences vidéo périodiques.

- Pour commencer, dans le premier chapitre nous allons présenter l'état de l'art du CVD qui permettra, mieux par la suite, la compréhension de notre contribution.

- Le deuxième chapitre décrit en détail l'implémentation de notre propre modèle CVD, conçu spécifiquement pour les séquences vidéo périodiques.

- Dans le dernier chapitre, les performances de notre modèle seront discutées.

# **Chapitre 1**

## **Etat de l'art du codage vidéo distribué**

## **Introduction :**

Dans ce chapitre, nous allons présenter l'état de l'art du CVD. Nous commencerons par les notions de base de la théorie de l'information et les théorèmes de codage de source distribué (CSD), qui ont contribué à la naissance du CVD. Ensuite nous présenterons les techniques de génération de l'information adjacente et les approches CVD les plus connues dans la littérature avec des présentations détaillées de leurs procédures de codage et décodage.

## **1. Notion de base de la théorie de l'information :**

La théorie de l'information représente le fondement du CVD. Bien que les premières tentatives de mesure de l'information datent des années vingt, ce n'est qu'à partir de 1948, grâce aux travaux effectués par Claude Shannon, que la théorie de l'information a pris sa forme actuelle. C'est une discipline fondamentale du domaine des communications. Elle consiste à déterminer les limites (imposées par les lois physiques) de la compression et la transmission de l'information à partir d'une source. Elle est par ailleurs en constante évolution au regard des exigences contemporaines où l'information à stocker et à transmettre devient de plus en plus importante.

### **1.1. Quantité d'information:**

Nous allons introduire le concept de mesure d'information, en supposant une source d'information discrète et d'alphabet fini. Le point de vue de la théorie de l'information conduit à nous intéresser à des sources et des canaux aléatoires. De ce fait, le destinataire, qui ne dispose que du message reçu, peut seulement faire des hypothèses au sujet du message émis et choisit le plus vraisemblable. Il traite donc le message délivré par la source comme un processus aléatoire.

Puisque son caractère imprévisible apparaît comme un attribut essentiel de l'information, on est conduit à identifier la mesure quantitative de l'information à la mesure de l'imprévisibilité. On mesurera ainsi la quantité d'information  $I(x)$  apportée par la réalisation d'un événement  $x$  de probabilité  $p(x)$  par une certaine fonction croissante de son improbabilité soit :

$$I(x) = f \left[ \frac{1}{P(x)} \right] \quad (1.1)$$

Où  $f$  est une fonction croissante et continue. Dans la théorie de l'information,  $f$  est une fonction logarithmique. Alors la quantité d'information  $I(x)$  deviendra :

$$I(x) = \log_b \frac{1}{p(x)} = -\log_b p(x) \quad (1.2)$$

## 1.2. L'entropie :

L'entropie représente la moyenne pondérée de la quantité d'information de chaque symbole délivré par la source. En d'autres termes, c'est l'espérance mathématique de l'information élémentaire donnée par chaque réalisation possible  $X = x_i$  de la source, l'entropie  $H(x)$  est donnée par :

$$H(x) = E[I_x(x_i)] = -\sum_{i=0}^N p_x(x_i) \log_P p_x(x_i) \quad (1.3)$$

## 1.3. Entropie conjointe et conditionnelle:

Considérons maintenant deux variables aléatoires  $X$  et  $Y$  à valeurs tirées de deux espaces d'épreuves  $\mathcal{X} = (x_0 \ x_1 \ \dots \ x_{N-1})$  et  $\mathcal{Y} = (y_0 \ y_1 \ \dots \ y_{N-1})$ . La définition précédente appliquée aux éléments  $(x_i, y_j)$  du produit cartésien de ces espaces, conduit à l'entropie conjointe donnée par :

$$H(x, y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{x,y}(x, y) \log_b p_{x,y}(x, y) \quad (1.4)$$

$$= -\sum_{i=0}^N \sum_{j=0}^M p_{x,y}(x_i, y_j) \log_b p_{x,y}(x_i, y_j) \quad (1.5)$$

Où  $\mathcal{X}$  et  $\mathcal{Y}$  sont respectivement les espaces d'épreuves de  $X$  et  $Y$ . D'une manière analogue, on peut définir l'entropie conditionnelle  $H(X/Y)$  comme l'incertitude qui reste sur  $X$  lorsque  $Y$  est connue, l'entropie conditionnelle  $H(X/Y)$  est donnée par :



$$H(X/Y) = - \sum_{i=0}^N \sum_{j=0}^M p_{x,y}(x_i, y_j) \log_b p_{x,y}(x_i, y_j) \quad (1.6)$$

#### 1.4. Codage entropique (codage statistique à longueur variable) :

Le codage entropique (ou codage statistique à longueur variable) est une méthode de codage de source sans pertes, dont le but est de transformer la représentation d'une source de données pour sa compression ou sa transmission sur un canal de communication. Les principaux types de codage entropique sont le codage de Huffman et le codage arithmétique.

On se donne une source discrète (données, fichier, . . .) dont chaque symbole  $x$  prend une parmi  $M$  valeurs possibles  $\{x_1, x_2, \dots, x_M\}$ . On appelle cela une source  $M$ -aire  $X$ .

La statistique de cette source est caractérisée par une distribution de probabilité  $p(x)$ , qu'on suppose connue (ou estimée) sous la forme  $\{p_1, p_2, \dots, p_M\}$ . Où  $p_i$  est la probabilité d'occurrence du symbole  $x_i$ .

Le codeur code chaque symbole de source  $x_i$  par un mot de code  $c_i$ . Un code à longueur variable (VLC : Variable-Length Code) est tel que les différents mots de code n'ont pas nécessairement la même longueur en bits. On note  $l_i$  la longueur en bits du mot de code  $c_i$  et

Voici un exemple de code VLC pour  $M = 4$  :

$x_i$	$p_i$	$l_i$	$c_i$
0	1/2	1	0
1	1/4	2	10
2	1/8	3	110
3	1/8	3	111

**Tableau 1.1. Exemple d'un code VLC pour  $M = 4$ .**

## 2. Les Codes LDPC :

Les codes LDPC (low-density parity-check) sont une classe des codes correcteurs d'erreur en blocs linéaires, découverts en 1963 par Gallager [3]. Ce nom leur vient des caractéristiques de leur matrice de contrôle de parité, qui comporte plus de 0 que de 1.

### 2.1. Représentations d'un code LDPC :

Un code LDPC est représenté par deux façons : représentation matricielle et graphique.

#### 2.1.1. Représentation matricielle :

Un code LDPC  $(n, k)$  est représenté par sa matrice de contrôle de parité  $H$  de taille  $(n \times m)$ , où  $m = n - k$ . On peut définir deux paramètres décrivant ces matrices,  $W_c$  et  $W_r$ . Où  $W_r$  est le nombre de 1 dans une ligne et  $W_c$  celui d'une colonne. si on veut construire une matrice à faible densité, on doit vérifier que  $W_c \ll n$  et  $W_r \ll m$ . L'équation (1.7) représente une matrice de contrôle de parité  $H$  pour un code LDPC (10,5).

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad (1.7)$$

#### 2.1.2. Représentation graphique :

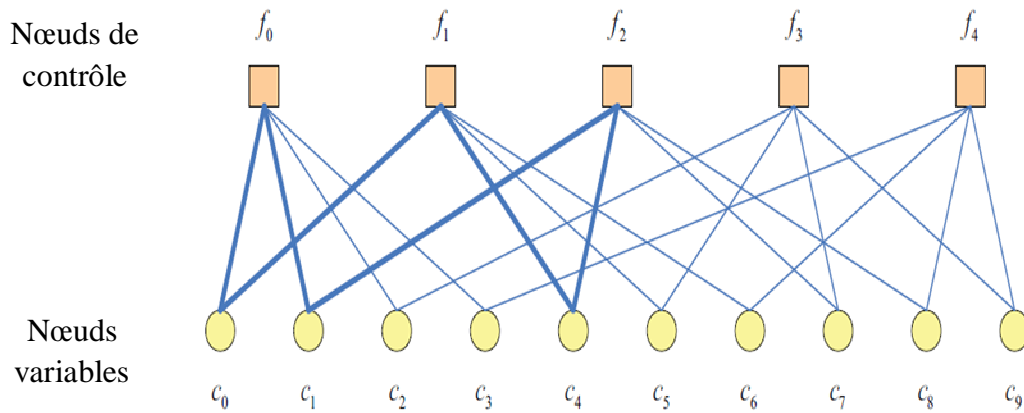
Le graphe de Tanner (1981) est un graphe biparti utilisé pour représenter les codes LDPC d'une façon plus efficace. Non seulement il fournit une représentation complète des codes LDPC, mais il facilite également la compréhension des algorithmes de décodage. Le graphe de Tanner comprend deux types de nœuds : les  $v$ -nœuds (nœuds variables) et les  $c$ -nœuds (nœuds de contrôle, ou syndromes). Le graphe de Tanner d'un code est dessiné selon les règles suivantes:

- Un code LDPC  $(n, k)$  est composé de  $m = (n - k)$   $c$ -nœuds (le nombre de bits de parité) et  $(n)$   $v$ -nœuds (le nombre de bits d'un mot à coder).

- Si l'élément  $h_{ij}$  de la matrice  $H$  vaut 1, alors le  $c$ -nœud  $f_i$  est connecté au  $v$ -nœud  $C_j$ .

Le graphe de Tanner pour le code LDPC généré par la matrice  $H$  (1.7) est représenté dans la figure 1.1. On remarque que les deux  $c$ -nœuds  $f_0$  et  $f_1$  sont reliés à  $C_0$  par le fait que, dans la

colonne de rang zéro en  $H$ ,  $h_{00} = h_{10} = 1$  tandis que  $h_{20} = h_{30} = h_{40} = 0$ . De cette façon on fait pour tous les c-nœuds et v-nœuds pour avoir le graphe de Tanner.



**Figure 1.1. Graphe de Tanner pour l'exemple de matrice de contrôle de parité (1.7).**

## 2.2. Procédure de codage et décodage d'un code LDPC :

### 2.2.1. Opération de codage :

Le codage d'un mot  $C$  (v- nœud) en un syndrome  $S$  (c-nœud) s'effectue par l'équation (1.8).

$$\mathbf{S} = \mathbf{H} \mathbf{C} \quad (1.8)$$

### 2.2.2. Opération de décodage :

Toutes les stratégies effectives de décodage des codes LDPC peuvent être classées dans une classe nommée par les algorithmes de passage du message (Message Passing Algorithms (MPA)), on les nomme ainsi car à chaque itération des algorithmes les messages sont transmis à partir des c-nœuds vers les v-nœuds et à partir des v-nœuds vers les c-nœuds. Bien que parfois elles viennent sous différents noms tels que : les algorithmes somme-produit (sum-product algorithm (SPA)) et l'algorithme de propagation de croyance (belief propagation algorithm (BPA)).

Le Décodeur LDPC corrige le mot de code reçu erroné en utilisant le BPA, si le mot de code corrigé vérifie l'équation de parité de contrôle  $\mathbf{C} \mathbf{H}^T = \mathbf{0}$ , alors l'opération de se termine et

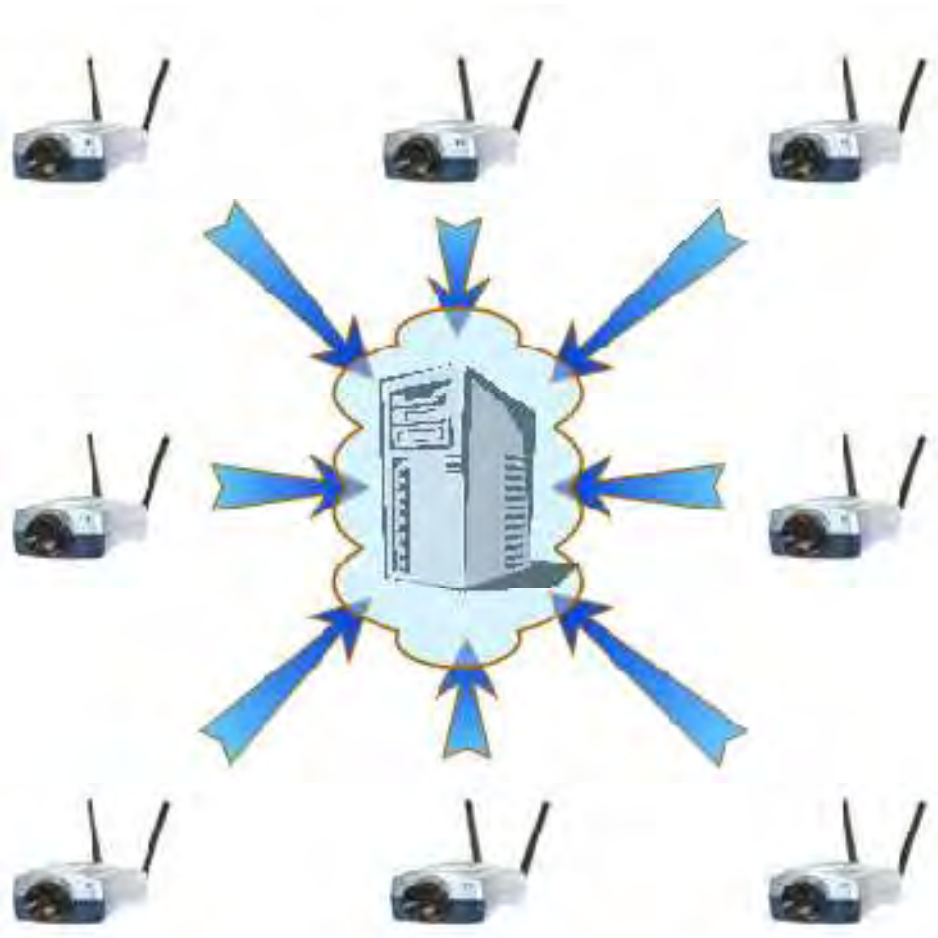
le décodage est déclaré réussi, sinon le décodeur demande plus de bit de parité du codeur jusqu'à ce que le mot de code corrigé vérifie l'équation de parité de contrôle.

### **3. Codage de source distribué(CSD) :**

Le CSD est une technique de compression qui repose sur le codage de deux sources (ou plus) d'informations aléatoires corrélées sans aucune communication les unes avec les autres. En modélisant la corrélation entre plusieurs sources au niveau du décodeur, le CSD est capable de basculer la complexité de calcul du codeur vers le décodeur. Un train de bits indépendants est envoyé par chaque codeur vers un seul décodeur qui effectue un décodage conjoint de tous les trains de bits reçus, en exploitant la corrélation statistiques entre eux.

La Figure 1.2 illustre une séquence vidéo acquise par une caméra de surveillance sans fil, où plusieurs caméras captent la même scène à partir de plusieurs positions différentes. Tandis que les caméras ne partagent pas l'information les unes avec les autres, leurs séquences vidéo associées sont généralement corrélés puisque les caméras voisines captent des zones partiellement recouvertes. La caméra envoie un flux de bits indépendants vers un décodeur centralisé, qui réalise un décodage conjoint de tous les flux de bits reçus en exploitant la corrélation entre eux.

Donc, il est possible de réduire la complexité du processus de codage en exploitant la corrélation entre plusieurs séquences codées juste au niveau du décodeur. Puisque la consommation d'énergie est un problème majeur dans un réseau de captage sans fil, comme le scénario de surveillance sans fil, les systèmes de codage vidéo classiques ne sont pas bien adaptés pour une telle application, car la tâche d'estimation de mouvement effectuée au niveau du décodeur consomme beaucoup d'énergie avec un taux de calcul colossale.



**Figure 1.2 - Codage de source distribué avec plusieurs séquences vidéo corrélées.**

Alors, on peut dire que l'exploitation de la corrélation entre les sources peut conduire à un taux de compression plus faible. Ainsi, il est intéressant de se demander quel est le taux minimum nécessaire pour la transmission de  $X$  et  $Y$ , lorsque  $Y$  n'est pas connue au niveau du codeur, mais elle est connue au niveau du décodeur. C'est logique que le taux de compression nécessaire doit être moins significatif que la somme des entropies, du fait de l'exploitation de la corrélation entre  $X$  et  $Y$ .

En 1973, D. Slepian et J. K. Wolf [1] ont proposé la limite théorique qui assure une compression sans perte dans le cas d'une compression distribuée de deux sources statistiquement corrélées, indépendantes et identiquement distribuées (i.i.d),  $X$  et  $Y$ . Cette limite a été étendue à une étude plus générale où il existe plus que deux sources par T. M. Cover in 1975 [4]. Selon ces limites le taux nécessaire de compression est moins significatif

que la somme des entropies. Plus tard, des nouvelles limites théoriques, dans le cas de compression avec perte, ont été présentées par A. D. Wyner et J. Ziv in 1976 [2].

Dans les sections qui suivent les deux théorèmes de Slepian-Wolf et WZ sont présentés.

### 3.1. Théorème de Slepian-Wolf :

Le théorème de Slepian-Wolf est adressé au codage indépendant de deux séquences statistiquement corrélées, independently and identically distributed ( i.i.d), comme illustré par la figure 1.3.

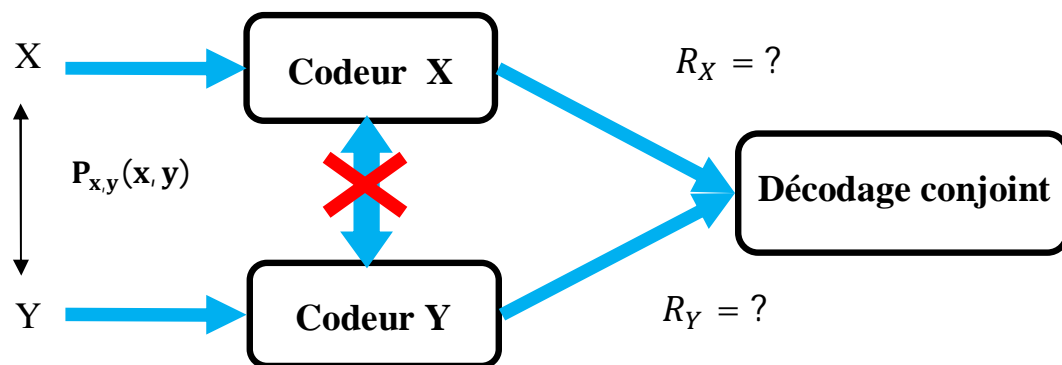


Figure 1.3 Codage distribué de deux sources statistiquement corrélées et i.i.d, X et Y.

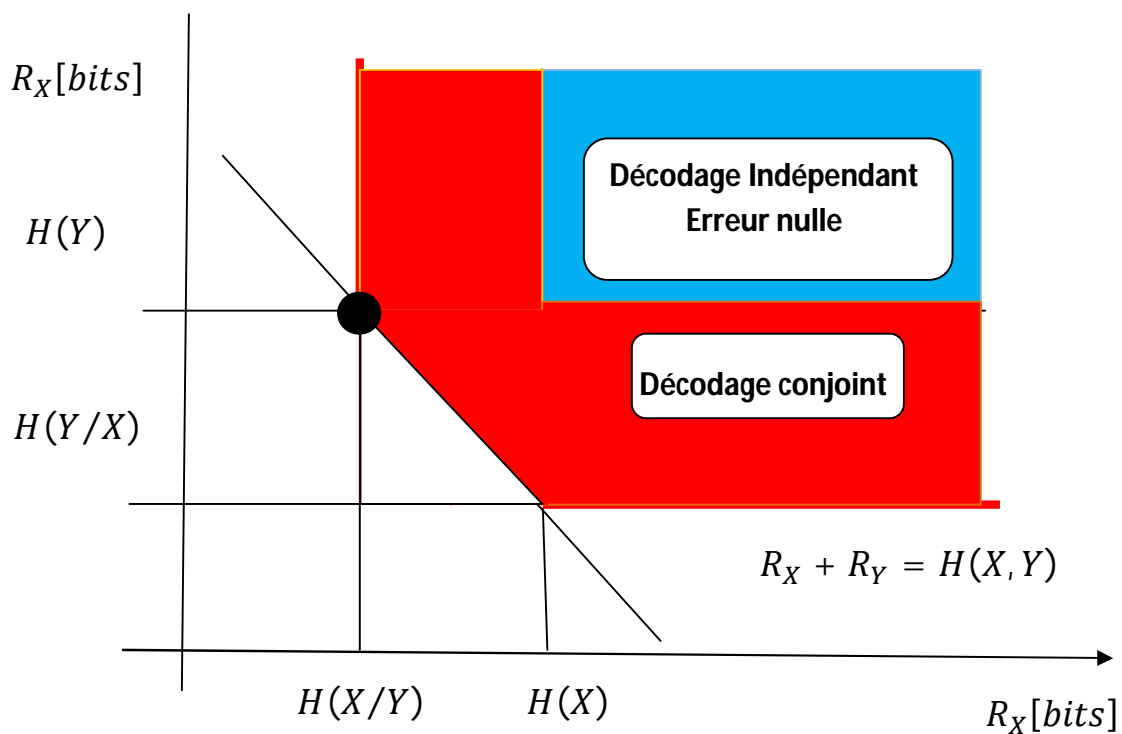
Le théorème de Slepian-Wolf stipule que le débit optimal, obtenu lors d'un codage et décodage conjoint de plusieurs sources corrélées, peut théoriquement être atteint par un codage indépendant et un décodage conjoint de ces sources en satisfaisant la transmission à des débits limités par l'équation (1.8). Ce théorème peut être considéré comme un aspect *distribué* du codage entropique [5].

$$R_x \geq H(x | y)$$

$$R_y \geq H(y | x) \tag{1.9}$$

$$R_x + R_y \geq H(x, y)$$

Ces limites de débit sont présentées dans la figure 1.4.



**Figure 1.4 - Illustration du théorème de Slepian-Wolf pour deux sources.**

La figure 1.4 illustre la région du débit obtenu par la compression distribuée de deux sources corrélées et (i.i.d.) X et Y, qui permet une reconstruction avec une erreur de probabilité très petite.

Comme la reconstruction des deux séquences, X et Y, est parfaite (en négligeant la faible probabilité d'erreur de décodage), le codage de Slepian-Wolf est généralement considéré comme un codage de source distribuée sans perte [6].

Une caractéristique très intéressante du codage de Slepian-Wolf est la relation qu'il a avec le codage de canal. La relation entre X et Y, déjà étudiée dans les années 1970 par Wyner [7], sera explicitée à partir de la figure 1.5.

Soient X et Y deux séquences statistiquement corrélées, on peut considérer qu'il existe entre X et Y un canal de corrélation virtuel, soit X l'entrée du canal virtuel et Y la sortie de ce canal. Les bits d'information envoyés sur le canal à un débit  $R_x \geq H(X|Y)$  sont considérés comme des bits de parité des codes correcteurs d'erreurs.

Logiquement, plus X et Y sont corrélées, moins les erreurs devront être corrigées, et plus de compression peut être atteint.

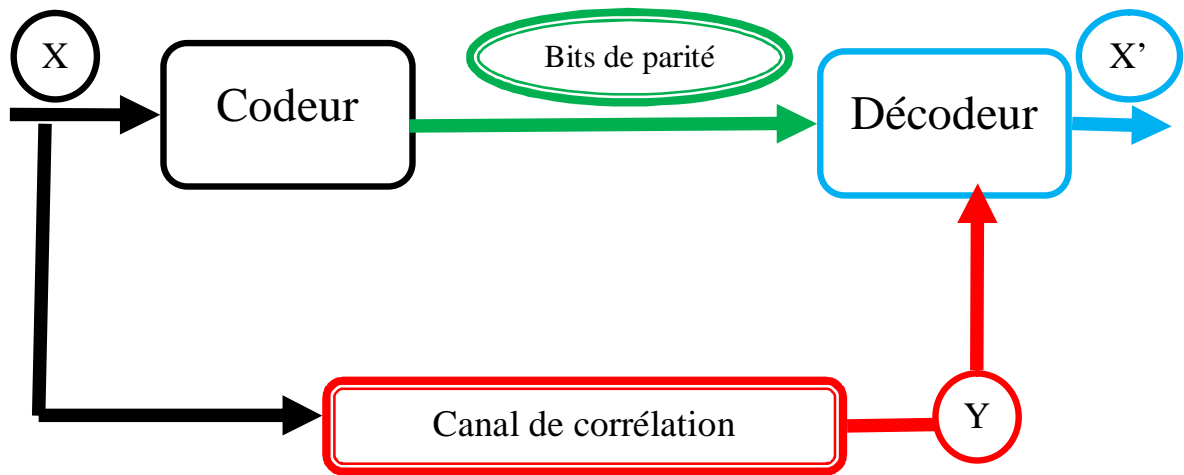


Figure 1.5. Relation entre le codage de canal et le codage de Slepian-Wolf

### 3.2. Théorème de Wyner-Ziv (WZ) :

Le travail de Slepian et Wolf a été développé en 1976 pour le cas avec perte par Wyner et Ziv [2]. Le Théorème de WZ effectue le traitement d'une source X dans le cas d'une compression avec perte en exploitant l'information adjacente Y qui se trouve au niveau du décodeur seulement, comme le montre la figure 1.6, Ce cas correspond au point  $(H(X|Y), H(Y))$  présenté dans la figure 1.4 comme un point noir (compression en exploitant l'information adjacente Y).

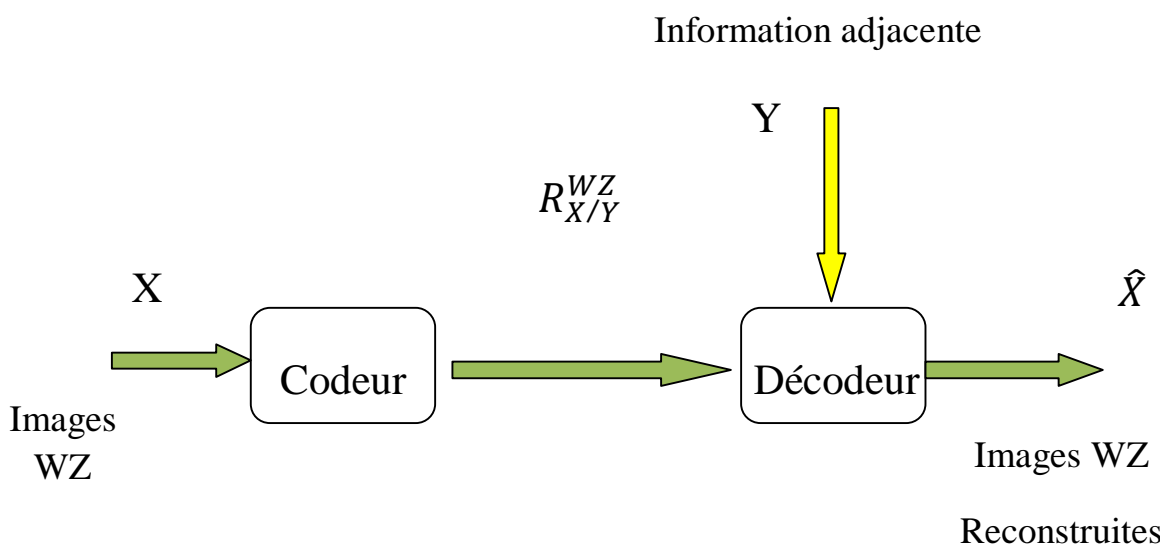


Figure 1.6. Le codage WZ de X en utilisant l'information adjacente Y disponible au niveau du décodeur seulement.



### 3.2.1. Principe du codage WZ:

Soit  $X$  et  $Y$  deux séquences aléatoires statistiquement corrélées et i.i.d, où  $X$  est la séquence à coder (information principale), et  $Y$  l'information adjacente disponible au niveau du décodeur. Les travaux de WZ ont permis d'établir les limites théoriques de débit-distorsion en d'autres termes ils ont permis de déterminer le débit minimum  $R^{wz}(\mathbf{d})$  nécessaire pour coder  $X$  en garantissant sa reconstruction avec une distorsion inférieure à la distorsion moyenne  $\mathbf{d}$ .

Lorsque la corrélation statistique entre  $X$  et  $Y$  est exploitée seulement au niveau du décodeur, le débit de transmission augmente par rapport au cas où la corrélation est exploitée au codeur et au décodeur, pour la même distorsion moyenne  $\mathbf{d}$ . Mathématiquement, le théorème de WZ peut être décrit par :

$$R^{wz}(\mathbf{d}) \geq R_{X/Y}(\mathbf{d}), \quad \mathbf{d} \geq 0 \quad (1.10)$$

Où  $R^{wz}(\mathbf{d})$  représente le débit du codage minimum de WZ (pour  $X$ ) et  $R_{X/Y}(\mathbf{d})$  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  $\mathbf{d}$ . Dans la littérature  $R^{wz}(\mathbf{d})$  et  $R_{X/Y}(\mathbf{d})$  s'appellent les fonctions du débit de distorsion. Notant que lorsque  $\mathbf{d} = 0$ , c'est à dire quand il n'y a aucune distorsion, on trouve de nouveau le résultat de Slepian-Wolf, c'est-à-dire  $R^{wz}(\mathbf{0}) = R_{X/Y}(\mathbf{0})$ .

Le concept de codage de WZ est bien adapté au codage vidéo. Dans ce cas, certaines images de la séquence vidéo sont codées en utilisant des techniques de codage vidéo classiques (les normes MPEG-X ou H.26X), les autres sont codées en WZ (qui correspond à la séquence  $Y$  dans La Figure 1.6). Le décodeur se sert des images codées par les standards classiques pour générer une estimation de l'image courante (l'information adjacente) par les techniques d'interpolation et d'extrapolation des images. Cette information adjacente est ensuite employée dans le processus de décodage de l'image courante.

En conclusion, les théorèmes de Slepian-Wolf et de WZ bien connus dans la théorie de l'information, suggèrent qu'il est possible de compresser deux signaux statistiquement corrélés

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 ensembles, c'est à dire comme dans le codage vidéo classique.

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

## **4. Le codage vidéo :**

### **4.1. Eléments de base :**

#### **4.1.1. La vidéo numérique :**

En principe, la vidéo numérique est une succession d'images, appelées plus communément trames, affichées selon un certain nombre d'images par seconde (*frame rate* exprimé en fps) pour créer l'illusion d'une animation [8]. Chaque pixel d'une image est préalablement codé suivant un modèle de représentation des couleurs. Le plus connu est l'espace colorimétrique RGB (*Red-Green-Blue*) correspondant aux trois couleurs primaires caractéristiques de la perception humaine [9].

Cet espace est basé sur une synthèse additive des couleurs, c'est-à-dire que le mélange des trois composantes dans des proportions diverses permet de reproduire à l'écran une part importante du spectre visible sans avoir à spécifier une multitude de fréquences lumineuses. Pour les applications les plus courantes les composantes sont codées selon une échelle de 256 valeurs (0 à 255 soit 8 bits) correspondant à leur niveau d'intensité. Ceci détermine aussi la profondeur du pixel qui représente le nombre total de bits sur lequel il est codé soit 24 bits pour ce cas ci. Le nombre de couleurs pouvant être ainsi représenté est de  $256 \times 256 \times 256$  possibilités, soit près de 16 millions de couleurs.

Bien que le système RGB soit très utilisé dans l'acquisition et l'affichage de l'image, pour ce qui est du traitement des couleurs et de la compression en revanche, le système de luminance-chrominance est plus efficace et donc plus largement utilisé. Ceci est étroitement lié à la perception des couleurs du système visuel humain. En effet, il est connu que ce dernier est plus sensible au vert qu'au rouge et moins sensible au bleu.

Ainsi, une égale représentation du rouge, vert et bleu mène à une représentation inefficace des données. Les normes internationales de compression vidéo utilisent préférentiellement la représentation colorimétrique YCrCb où Y correspond à la luminance et Cr-Cb aux

chrominances rouge et bleu respectivement [9]. La relation reliant les deux espaces RGB et YCrCb est simplement linéaire:

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.369 & -0.071 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix}$$

Des expériences psycho visuelles ont montré que le système de vision humaine est plus sensible aux hautes fréquences spatiales de la luminance Y qu'à celles des chrominances Cr et Cb. Ceci a permis par la suite le sous-échantillonnage spatial de ces dernières sans apporter de détérioration notable au contenu visuel.

### 4.1.2 La compression vidéo :

La compression vidéo repose sur deux principes de base [8]. Le premier exploite les redondances spatiales qui existent dans chaque image. Le second repose sur la caractéristique qu'une image peut-être très similaire aux images voisines. C'est ce que l'on appelle la redondance temporelle. Une technique typique de compression vidéo devrait donc débiter par encoder la première image par une méthode classique de compression d'image, puis encoder chacune des images successives en identifiant les différences avec les images antérieures, et enfin d'encoder ces différences.

Si l'image est très différente de ses voisines, elle devra être encodée indépendamment. Dans la littérature, cette image sera dite *intra* alors qu'une image codée par l'intermédiaire de ses prédécesseurs sera identifiée comme une image *inter*. Nous verrons plus tard que ce qui distingue cette compression traditionnelle de la compression vidéo dite *distribuée* est que pour cette dernière les images subissent un encodage indépendant les unes des autres. Le processus de décodage tiendra cependant compte de la corrélation entre les images.

La compression vidéo se fait généralement avec perte. Encoder une image  $T_i$  à l'aide d'une image antérieure  $T_{i-1}$  introduit de la distorsion. Il est aussi montré qu'encoder l'image  $T_{i+1}$  à l'aide de l'image  $T_i$  augmente cette distorsion. Pour éviter qu'une distorsion trop importante n'apparaisse et ne dégrade le rendu vidéo, les images intra sont régulièrement utilisées dans la séquence pour maintenir une certaine cohérence dans le rendu vidéo.

En définitive, trois types d'images sont nécessaires pour effectuer le codage différentiel et le codage bidirectionnel procurant un minimum d'erreurs de propagation:

- **Image I ou K** (*Intra*) : intra-codée et ne nécessite pas d'information additionnelle pour être décodée.
- **Image P** (*Prédictive*) : déduite d'une image antérieure qui peut être de type I ou P.
- **Image B** (*Bidirectionnel*) : se calcule de manière bidirectionnelle depuis les images antérieures ou postérieures de type I ou P.

Usuellement, une séquence vidéo est segmentée selon des groupes d'images consécutifs (GOP : *Group of Pictures*) débutant par une image I et suivis d'images de type P puis B dans des positions bien spécifiées (exemple d'ordre d'affichage: ( I P P I P .....)). Les images constituant le GOP peuvent avoir des ordonnancements différents suivant l'encodage ou l'affichage.

En effet, les images **B** sont encodées après le calcul des images **P** bien qu'elles apparaissent antérieurement à ces dernières lors de l'affichage. D'après l'exemple précédent la séquence d'encodage serai [10] : ( IP P P IP ..... )

En règle générale, la rapidité d'un encodeur importe peu dès l'instant que le décodage (et donc l'affichage) soit rapide [8]. Pour la compression vidéo distribuée en revanche, l'asymétrie de complexité se trouve inversée : l'encodeur est simple mais le décodeur complexe.

#### **4.1.3. Codage vidéo traditionnel (La norme H.264/ AVC) :**

H.264, ou MPEG-4 AVC, est la norme de codage traditionnel la plus récente qui se trouve sur le marché. C'est une norme de codage vidéo issue du travail commun de normalisation de UIT-T *Video Coding Experts Group* (VCEG) ainsi que l'ISO *ICEI Moving Picture Experts Group* (MPEG) [11]. Elle est de plus le produit d'un effort de partenariat connu sous le nom *Joint Video Team* (JVT). Actuellement la norme est devenue partie intégrante de la nouvelle génération de supports vidéo haute définition (HD-DVD, Blu-Ray) ainsi que d'autres applications comme la télévision sur téléphone portable. La norme (cf. [12]) , publiée en 2003 définit cinq profils différents et globalement 15 niveaux répartis selon ces profils [10]. Chaque profil correspond à un sous-ensemble de syntaxe utilisé pour la représentation des données codées. Ceci permet de faire varier la complexité du codec selon les besoins d'une application particulière.

La norme de codage permet d'atteindre une bien meilleure compression vidéo que toutes les normes précédentes avec un débit deux ou trois fois moindre pour la même qualité au

décodage [10]. Cependant cela se traduit par une complexité accrue à 1 encodage. Par rapport aux normes antérieures, elle présente quelques innovations majeures [10] :

- La compensation de mouvement peut être effectuée avec plusieurs images de référence déjà codées. Il est ainsi possible d'avoir jusqu'à 32 images utilisées comme référence et jusqu'à 4 références différentes pour un même *macrobloc*. Cette fonctionnalité permet une réduction significative du débit pour des vidéos comportant des scènes très rapides.
- La norme H.264 reconnaît qu'un *macrobloc* peut-être similaire à un autre dans la même image. De ce fait, l'estimation et la compensation de mouvement sont étendues à l'intra-codage (intra prédiction) , recherchant ainsi les similitudes internes à l'image. Bien entendu, cette recherche est limitée à des portions de l'image restant disponibles avant le décodage des macroblocs.
- Les macroblocs peuvent être de forme rectangulaire (les formes permises sont ainsi 16 x 8, 8 x 16, 8 x 8). Dans le profil 1, les macroblocs sont découpés selon des blocs 4 x 4 pour la luminance et 2 x 2 pour les chrominances; de plus larges blocs 8 x 8 sont permis dans les profils supérieurs. Ce degré de liberté supplémentaire permet une meilleure estimation de mouvement, spécialement sur des régions hautement détaillées où la correspondance avec des macroblocs fonctionne mal.
- La H.264 inclut une version modifiée de la DCT avec des valeurs entières dans la matrice de transformation H . En fait, les valeurs de H sont légèrement différentes si la transformation est appliquée sur des intra-blocs, des blocs d'erreur résiduelle ou bien des blocs de luminance et de chrominance. Les valeurs de la matrice H sont choisies de telle sorte que les transformations directe et inverse s'opèrent seulement par des décalages de bits ou des additions permettant ainsi de réduire la complexité calculatoire.
- À la place d'un codage à longueur variable, tels que le codage de Huffman ou le codage arithmétique conventionnel, la norme H.264 introduit deux autres techniques pour encoder l'erreur résiduelle transformée, les vecteurs de mouvement, etc. , soient:
  1. le codage arithmétique contextuel (*CABAC : Context-adaptive binary arithmetic coding*), une technique sophistiquée pour compresser sans perte des éléments syntaxiques dans le flux vidéo.
  2. le codage Golomb exponentiel est utilisé afin d'encoder des valeurs de paramètres uniques tandis qu'un codage adaptatif de type Huffman (*CAVLC : Contextadaptive Huffman variable-length coding*) est utilisé comme amélioration du codage à longueur variable traditionnel.

Le schéma de principe de l'encodeur de la norme H.264 est représenté par la figure 1.7.

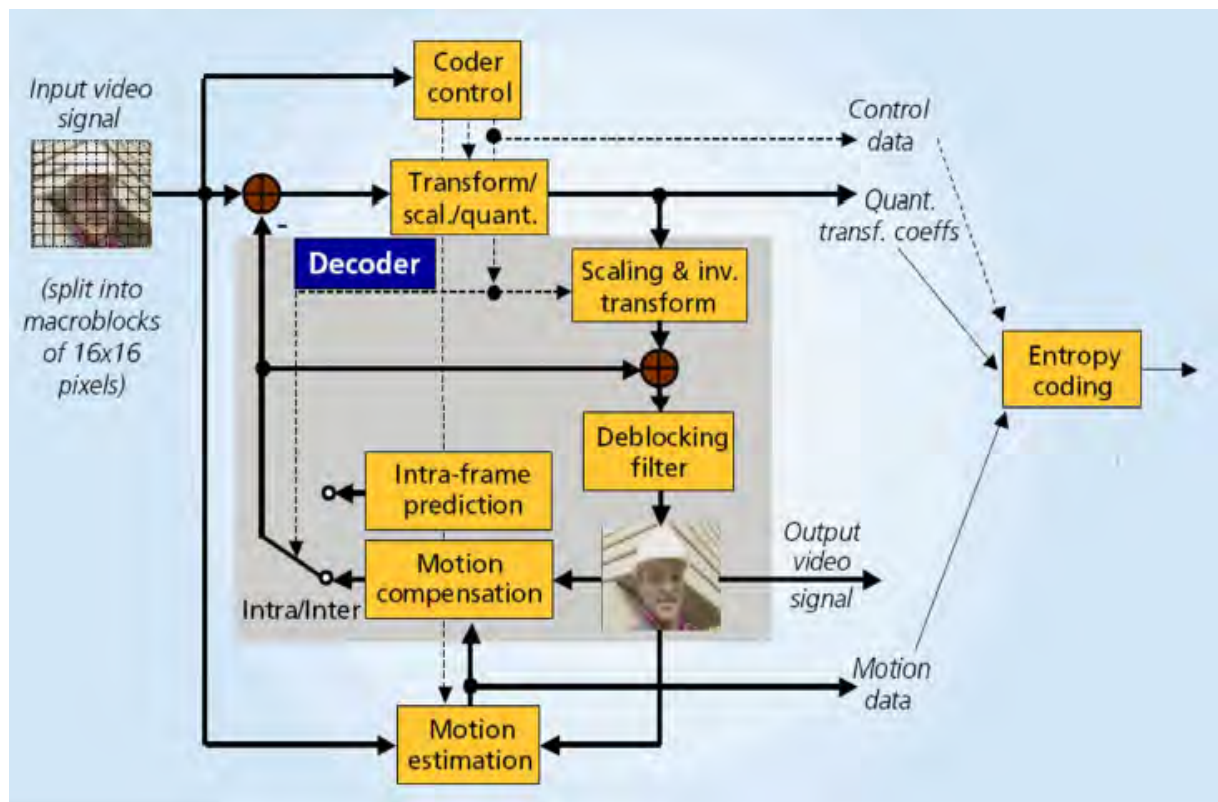


Figure 1.7 - Structure de codage de base du H.264/AVC[11]

## 4.2. Codage vidéo distribué :

### 4.2.1. Présentation des Systèmes CVD :

Les codeurs vidéo traditionnels, tel que H.264/AVC, peuvent être considérés comme un système de codage de source avec la disponibilité de l'information adjacente à la fois dans le codeur et le décodeur. Pour être plus précis, il existe une image prédite  $Y$  de chaque image originale  $X$ , qui est créée au niveau du codeur et elle est connue à la fois au codeur et au décodeur. Cette image prédite, ou en d'autre terme l'information adjacente, est créée en utilisant la prédiction temporelle (estimation de mouvement) et spatiale.

Les encodeurs vidéo qui exploitent les informations temporelles et spatiales sont généralement appelés **les codeurs vidéo hybride**. Dans ces codeurs, le résiduel entre l'image source à coder  $X$  et sa prédite  $Y$ , ainsi que les vecteurs de mouvement (VMs), sont envoyés au décodeur. Ils sont utilisés là-bas avec les images précédemment décodées afin d'obtenir une reconstruction  $\hat{X}$  de  $X$ . Ce processus est bien illustré par le diagramme du CODEC de H.264/AVC dans la Figure 1.7.

Comme on peut le voir sur cette figure, le côté codeur du H.264/AVC est très complexe. Plus précisément, il comprend les deux modules de prédiction d'images : l'inter (prédiction temporelle) et intra (prédiction spatiale), qui effectuent des calculs très lourds. De l'autre côté, le décodeur n'a pas besoin de faire beaucoup de calculs, juste des opérations simples, telle que l'addition des blocs d'images décodées déjà existantes dans l'image résiduelle en utilisant les VMs, ensuite la dé-quantification et les opérations de la DCT inverse. Il s'agit d'une configuration maître-esclave classique, où le codeur est le maître et le décodeur est l'esclave.

Mais, avec l'explosion des technologies précisément les applications sans fils comme les caméras sans fils, téléphones portables et d'autre, les codeurs traditionnel comme le H.264/AVC ne sont pas très adaptés à ces applications car le cote codeur est complexe et donc consomme beaucoup d'énergie.

Ensemble, les théorèmes Slepian- Wolf et le WZ suggèrent qu'il est possible de compresser deux signaux statistiquement corrélés d'une manière distribuée (codage séparé, décodage conjoint), en approchant l'efficacité du codage des systèmes de codage prédictif conventionnels avec moins de complexité et donc consommer beaucoup moins d'énergie. Cela a donné naissance à une nouvelle approche de codage, dans la quelle le codeur peut être très simple, tandis que le décodeur, qui est la partie qui exploite la corrélation statistique entre les sources, sera la partie la plus complexe du système de codage vidéo.

Dans ces conditions, un nouveau paradigme de codage vidéo, connu sous le nom CVD qui est basée sur les théorèmes ci-dessus, a vu le jour dans les dix dernières années. Le CVD peut être considéré comme une application particulière des résultats théoriques du CSD.



**Figure 1.8 - Codeurs vidéo avec complexité limitée. téléphone portable, caméra sans fil, capsule endoscopique, endoscope sans fil.**

Selon le paradigme CVD, l'information adjacente est créée au niveau du décodeur. Notamment, l'opération complexe de prédiction temporelle (estimation de mouvement(EM)) dans le codeur est évitée et donc la complexité du codeur va tombée de manière significative. En général, les systèmes CVD visent à combiner les caractéristiques du codage intra frame qui fournissent un codage à faible complexité et une robustesse aux erreurs de transmission, avec l'efficacité de la compression du codage inter frame. La Figure 1.8 présente un exemple d'applications dans lequel les codeurs à faible complexité sont nécessaires.

Un diagramme de haut niveau d'un système CVD typique est présenté dans la figure 1.6. Comme on peut le voir dans cette figure l'information adjacente est créée au niveau du décodeur et donc il n'y a aucune exploitation de la corrélation temporelle au niveau du codeur entre les images voisines. Au lieu de cela, cette corrélation est exploitée au décodeur en utilisant les informations temporelles qui existent dans les images déjà décodées.

L'objectif principal du CVD est de fournir des performances de compression comparables à celle des systèmes de codage vidéo hybride, comme MPEG-2 et H.264. Selon les théorèmes Slepian-Wolf et WZ, cet objectif est théoriquement réalisable, dans certains cas, comme dans le cas d'une gaussienne. Cependant, il ya certaines hypothèses de ces théorèmes qui ne sont pas nécessairement satisfaites sous le cadre du CVD.

Par exemple, la distribution conjointe de la source et l'information adjacente ne sont pas connues dans la pratique, et elles varient spatialement et temporellement tout le long de la séquence vidéo, à moins que la vidéo soit très statique. En particulier, le cas de la distribution gaussienne conjointe qui, d'après WZ, ne subit aucune perte de débit par rapport au codage classique, n'est pas nécessairement satisfait, ainsi une certaine perte de débit est inévitable. Donc, on s'attend à rencontrer une certaine dégradation des performances par rapport aux codeurs vidéo hybrides. Un système de CVD a deux avantages majeurs sur les codeurs vidéo classiques tels que le MPEG-2 et H.264.



## **1. complexité :**

Le CVD permet de concevoir des décodeurs de différentes complexités. Par exemple, la création de l'information adjacente peut être effectuée en utilisant des algorithmes simples (comme simplement copier les images précédemment décodées ou d'autres un peu plus complexes, tels que les algorithmes basés sur l'interpolation ou l'extrapolation du mouvement. De plus, dans un environnement qui permet au codeur d'être complexe à un certain degré, le codeur peut être adapté en conséquence, par exemple en effectuant une estimation de mouvement limitée.

## **2. Robustesse aux erreurs de dérive :**

Dans les systèmes de codages vidéo classiques, une erreur de transmission ou une perte de paquets peut conduire à une prédiction erronée de l'image courante au niveau du décodeur. Dans ce cas. Cette différence peut entraîner une reconstruction erronée au niveau du décodeur, et cette erreur peut même affecter aussi la reconstruction d'autres images, Ce n'est pas le cas dans le CVD, puisque dans un codeur de base de CVD l'image est souvent codée indépendamment de toute autre image. Ainsi, même dans le cas d'une erreur de transmission, une seule image peut être affectée.

### **4.2.2. Les solutions du CVD les plus intéressantes :**

Les premiers systèmes WZ pratiques ont été développés autour des années 2002, à l'Université de Stanford et l'Université de Berkeley. L'architecture Stanford se caractérise par un codage à base d'images (domaine de pixels), en utilisant les turbo codes comme codes correcteurs d'erreurs et un canal de retour, tandis que la solution de l'université de Berkeley (nommé PRISM) se caractérise par un codage à base de blocs, en utilisant une transformée DCT comme méthode de codage, sans canal de retour.

Plus tard, d'autres architectures ont été proposées. Elles sont principalement basées sur les modèles précédemment mentionnés. Les systèmes les plus récents sont DISCOVER [13] et VISNET II [14], où DISCOVER a gagné plus de popularité grâce à sa simplicité et sa bonne performance. DISCOVER est basé sur l'architecture de Stanford, avec deux changements majeurs, étant un modèle de bruit amélioré avec l'utilisation des codes LDPC au lieu des turbos codes, et un code CRC. VISNET II est similaire à DISCOVER, avec l'utilisation de l'estimation de mouvement pour créer l'information adjacente et un filtre de déblocage [15] qui est appliqué sur les images décodées.

Dans les sections qui suivent nous allons présenter un peu en détail ces systèmes de codage CVD, mais comme l'essentiel de notre travail repose sur l'utilisation de l'information adjacente au préalable on va expliciter ce qu'est l'information adjacente et les manières de sa création dans la littérature.

## **1. L'information adjacente (Side information):**

### **1.1. Définition :**

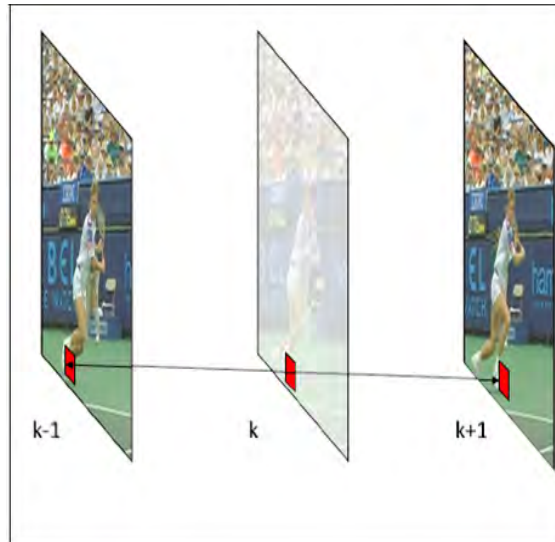
L'information adjacente est une version bruitée (erronée) de l'image originale qui va être codée en mode WZ, elle est générée toujours au niveau du décodeur par des différentes techniques, pour être ensuite exploitée à la reconstruction des images originales. Ceci est fait par l'utilisation des codes correcteurs d'erreur comme le LDPC ou les Turbo codes. Plus la qualité de l'information adjacente est bonne (la corrélation entre les images WZ est élevée) moins les bits de parités seront envoyés du codeur vers le décodeur, ce qui permet d'avoir un codec CVD avec des performances de haut niveau. Dans ce qui suit nous allons présenter les techniques les plus utilisées pour la création de l'information adjacente dans la littérature.

### **1.2. Différentes techniques de Génération de l'information adjacente :**

La génération de l'information adjacente est l'étape la plus importante dans tout système CVD. Elle joue un rôle critique dans la détermination de la qualité de ce codec,

#### **1.2.1. Estimation de l'information adjacente par l'interpolation :**

L'estimation par l'interpolation est l'une des techniques les plus utilisées pour générer l'information adjacente dans les codecs CVD. En général l'interpolation estime l'information adjacente d'une image WZ, qui se trouve dans une position temporelle  $k$ , en se basant sur deux images de référence déjà décodées, la précédente ( $k-1$ ) et la suivante ( $k+1$ ) comme l'illustre la figure 1.9.



**Figure 1.9. Prédiction de l'image K en utilisant la technique de l'interpolation.**

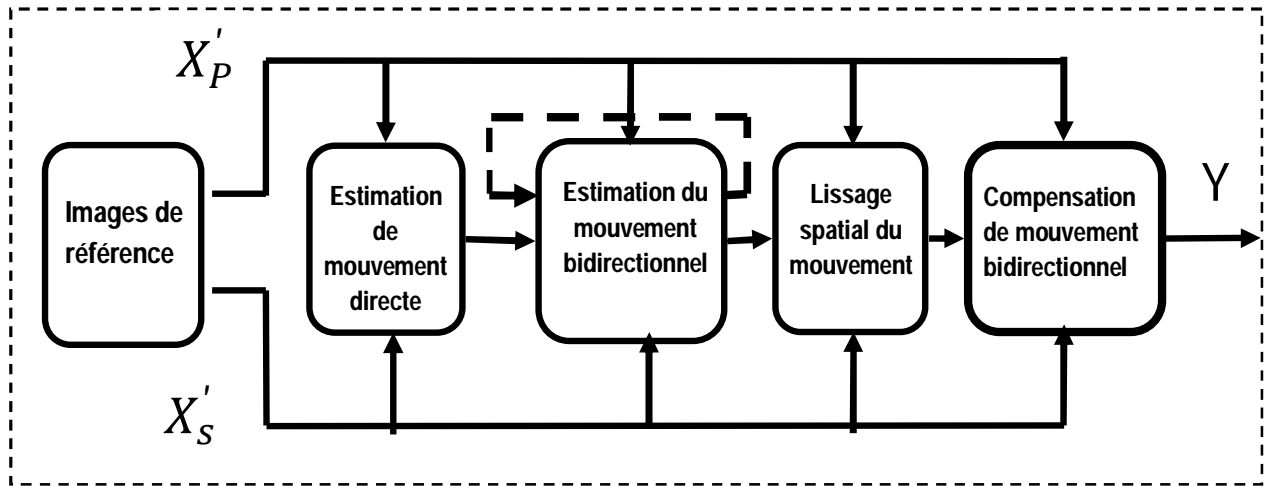
Les premières solutions de création de l'information adjacente dans la littérature ont été proposées par Aaron et al en 2002, qui ont proposé les deux premières architectures d'estimation de l'information adjacente par interpolation : la première consiste à prendre la moyenne des deux images-clés les plus proches, alors que la deuxième se fait en utilisant la compensation de mouvement par interpolation basant sur des vecteurs de mouvement symétriques. Ces premières méthodes sont très simples, mais plutôt inefficaces, et donc de nombreuses améliorations ont été proposées depuis. Elles ont permis d'améliorer les performances des codeurs CVD. Ces techniques améliorées seront présentées dans ce qui suit.

#### **1.2.1.1 Estimation de mouvement bidirectionnelle avec lissage de mouvement spatial :**

En 2005, Ascenso et al. ont développé une nouvelle technique de création de l'information adjacente, appelée *motion compensated frame interpolation (MCFI)* [16], qui a été largement utilisée dans la littérature. En 2006, les mêmes auteurs ont amélioré cette technique en donnant une nouvelle architecture [17]. Cette architecture est illustrée dans la figure 1.10.

#### **Procédure de génération de l'information adjacente :**

La **MCFI** dans [17], crée l'information adjacente pour une image WZ qui se trouve à la position temporelle  $k$ , en se basant sur deux images clés déjà décodées, la précédente ( $X_b'$ ) qui se trouve à la position temporelle  $(k-1)$  et la suivante ( $X_f'$ ) qui se trouve à la position temporelle  $(k+1)$  pour un GOP de taille 2.



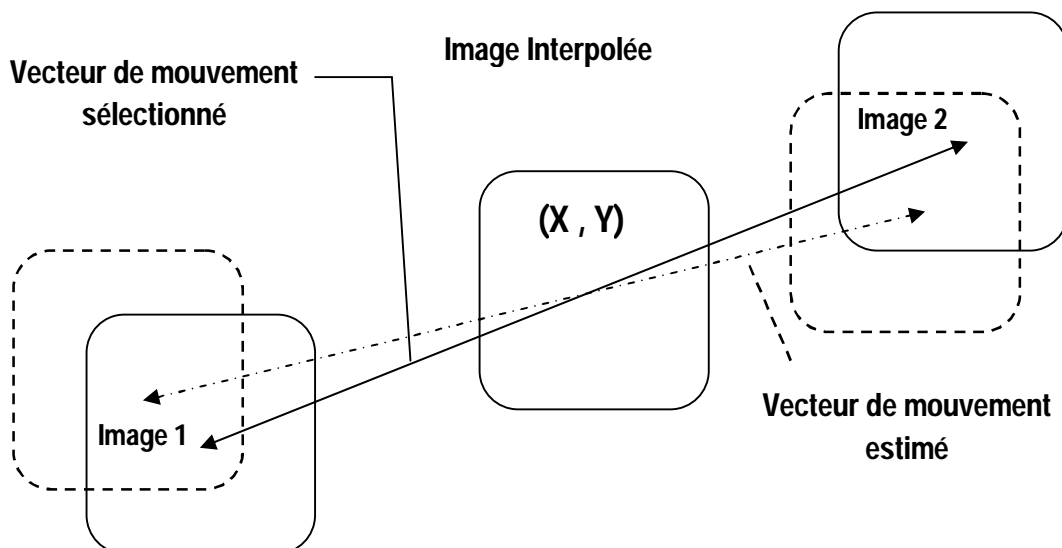
**Figure 1.10. L'architecture d'estimation par interpolation**

**- Estimation de mouvement directe:**

L'algorithme d'estimation du mouvement par recherche globale des blocs similaire, est appliqué pour estimer le mouvement entre les images précédemment décodées ( $X_f'$ ) et ( $X_b'$ ).

**- Estimation du mouvement bidirectionnel:**

L'estimation du mouvement bidirectionnel affine les vecteurs de mouvement obtenus par l'étape précédente. Cette technique sélectionne la trajectoire linéaire entre l'image clé suivante et précédente comme passant au centre des blocs de l'image interpolée comme illustrée dans la figure 1.11.



**Figure 1.11. Estimation du mouvement bidirectionnel**

### - Lissage spatial du mouvement:

L'algorithme de Lissage spatial du mouvement [18], est utilisé pour lisser le champ de vecteur de mouvement final.

### - compensation de mouvement bidirectionnel:

Après avoir obtenu le champ de vecteur de mouvement final, l'image interpolée peut être remplie en utilisant simplement la compensation de mouvement bidirectionnelle, comme défini dans les schémas de codage vidéo traditionnels [19].

### 1.2.1.2. Estimation de mouvement bidirectionnelle non linéaire avec plusieurs images de référence:

En 2010, Petrazzuoli et al. Ont proposés une méthode appelée, high order motion interpolation (HOMI) [20] où l'information adjacente est générée en utilisant quatre images de référence, deux images précédentes ( $k-3$  et  $k-1$ ) et deux images suivantes ( $k+1$  et  $k+3$ ), pour une image WZ qui se trouve dans la position temporelle  $k$  comme illustré dans la figure 1.12.

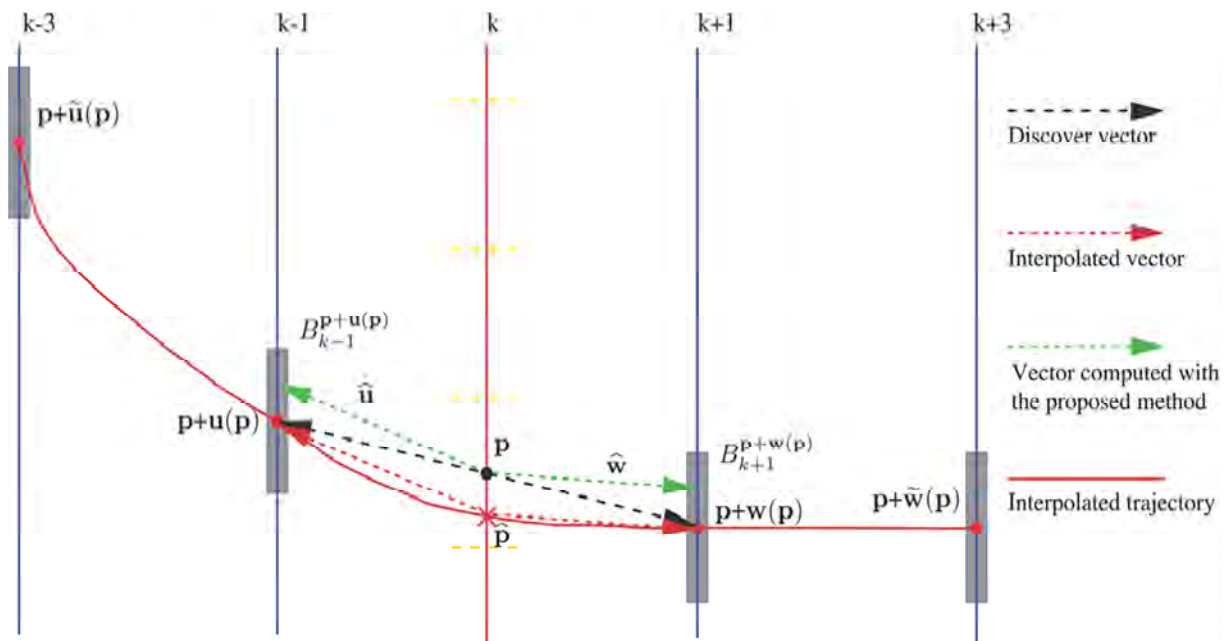


Figure 1.12. Illustration de la méthode (HOMI)

Afin de créer l'information adjacente qui se trouve dans la position temporelle  $k$ , la technique de MCFI est d'abord utilisé pour estimer le mouvement entre les images  $k-1$  et  $k+1$  pour avoir les vecteurs de mouvement  $u(p)$  et  $w(p)$  connus comme ('Discover vector'), ensuite une mise en correspondance de bloc est appliqué pour trouver dans  $k-3$  et  $k+3$  le bloc qui semble le

plus au blocs  $B_{k-1}^{p+u(p)}$  et  $B_{k+1}^{p+w(p)}$  respectivement pour obtenir les vecteur de mouvement  $\tilde{u}(p)$  et  $\tilde{w}(p)$  respectivement, la valeur interpolée, qui se trouve dans la position  $\tilde{p}$ , est obtenue en appliquant l'interpolation entre les quatre images de référence. Finalement après avoir obtenu les vecteurs de mouvement finaux  $\tilde{u}$  et  $\tilde{w}$ , le bloc de l'information adjacente estimée est ensuite rempli en utilisant une compensation de mouvement bidirectionnelle.

### 1.2.2. Estimation de l'information adjacente par l'extrapolation :

L'extrapolation estime l'information adjacente en se basant seulement sur les images clés précédemment décodées comme l'illustre la figure 1.13. Cette technique est moins efficace que l'interpolation. L'avantage majeur de l'extrapolation réside dans le fait qu'elle coûte moins en envoyant moins d'images clés codées intra. Dans ce qui suit nous allons présenter les techniques d'estimation de l'information adjacente par l'extrapolation les plus utilisées dans la littérature.

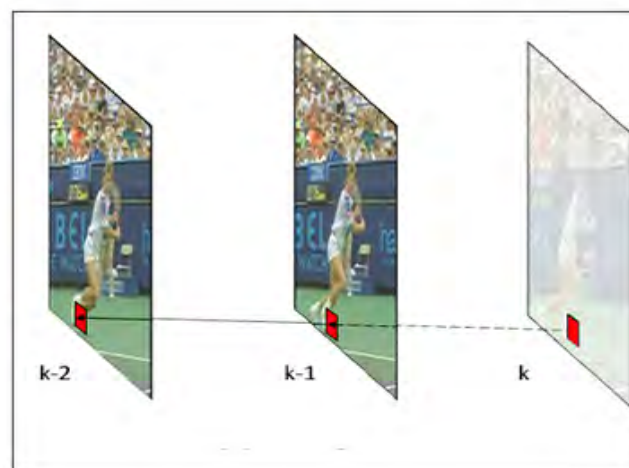


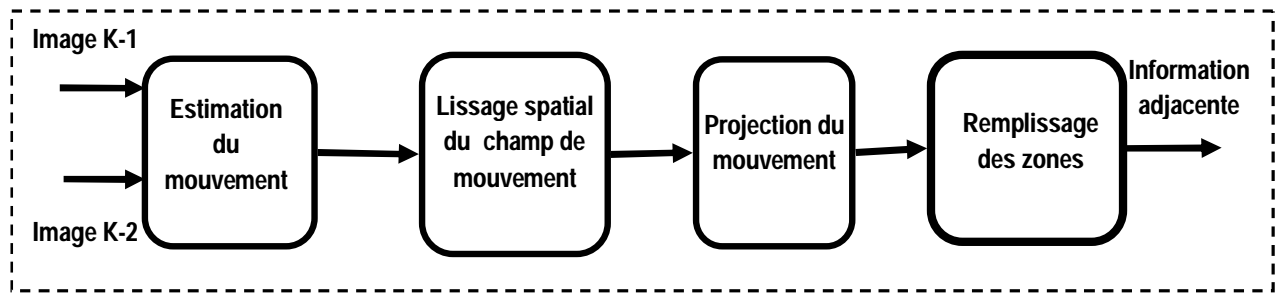
Figure 1.13. Prédiction de l'image K en utilisant la technique de l'Extrapolation.

#### 1.2.2.1. Estimation du mouvement avec Lissage spatial du mouvement:

En 2005, Natário et al [21]. a proposé une méthode de génération de l'information adjacente en utilisant une architecture d'extrapolation comme illustré dans la figure 1.14.

#### Procédure de génération de l'information adjacente :

Cette méthode génère l'information adjacente en exploitant les deux images clés précédemment décodés k-1 et k-2.

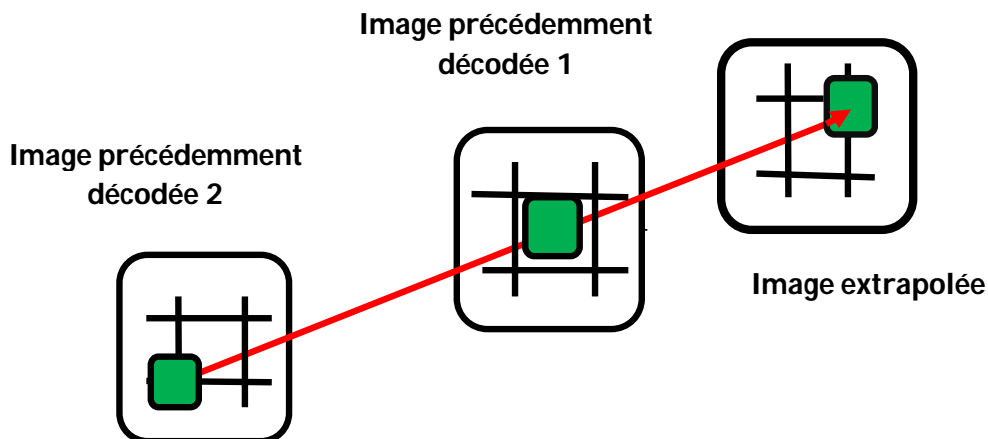


**Figure 1.14** architecture d'extrapolation proposée par Natário et al.

1- Les vecteurs de mouvement sont estimés pour chaque block de  $8 \times 8$  en utilisant une estimation du mouvement.

2- Pour chaque block, un nouveau vecteur de mouvement est calculé, en prenant la moyenne de tous les vecteurs de mouvement voisins. Ce qui permet de capturer le vrai mouvement et alors améliorer la qualité de l'information adjacente.

3- Les valeurs des pixels seront transmises de l'image  $k-2$  vers l'image  $k-1$  et de l'image  $k-1$  vers l'image estimée  $k$  en utilisant le champ de mouvement obtenu précédemment comme illustre par la figure 1.15.



**Figure 1.15. Projection du mouvement**

4- Chaque fois qu'un pixel est estimé par plus d'un pixel dans les images précédentes, cette position va prendre la valeur moyenne de ces pixels, Lorsque'aucun pixel des images précédentes n'est attribué à l'image estimée, cette valeur est estimée par une interpolation spatiale locale.

## 2. Les modèles proposés dans la littérature :

### 1. Le modèle CVD proposé par l'équipe de Stanford :

Le système CVD de Stanford a été proposé par un groupe de chercheurs de l'université de Stanford. Cette architecture de codage vidéo a d'abord été proposée pour le domaine de pixel [22] et a ensuite été étendue au domaine de transformation [23]. Le codec de Stanford dans le domaine de transformation est présenté dans la figure 1.16 et fonctionne comme suit :

La séquence vidéo est divisée en deux types d'images, les images clés (les images impaires de la séquence vidéo) et les images WZ (les images paires de la séquence vidéo). Les images clés sont codées en mode intra, en utilisant le codeur intra H.263+, ou le plus développé (plus complexe) H.264/AVC intra mode, et les images restantes sont codées par les techniques du CVD.

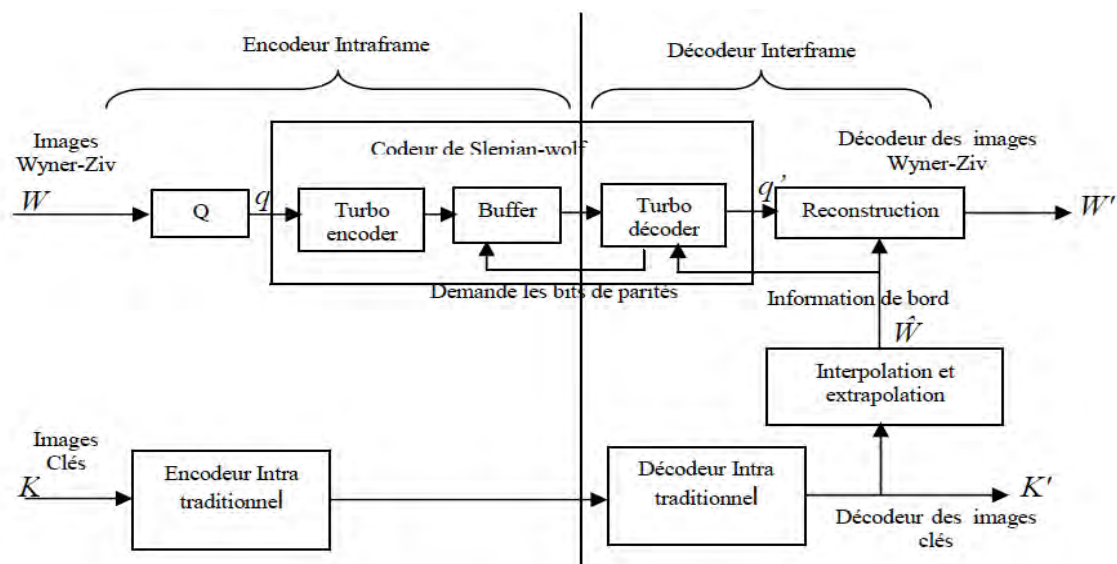


Figure 1.16 Architecture du codeur CVD de l'équipe Stanford

### 1.1 La procédure de codage :

#### Au niveau du codeur :

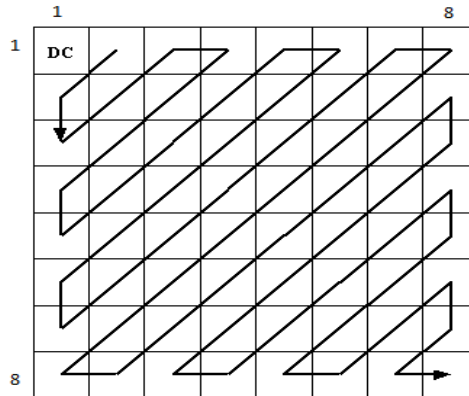
##### - Classification :

La séquence vidéo est divisée en des images clés et des images WZ, formant tous les deux ce qu'on appelle un GOP (group of pictures) ou un groupe d'images. Typiquement, un GOP de longueur deux est choisie, c'est-à-dire que la première image dans chaque GOP est une image clé et la deuxième est une image WZ, qui est codée en utilisant les principes de CSD comme décrit ci-dessous.



### - Transformation :

Une transformation par bloc, généralement DCT de 4×4 ou 8×8 est appliquée à chaque image de type WZ. Ensuite les coefficients résultant de la transformation de l'image WZ entière, sont scannés et regroupés en zig-zag comme illustré sur la figure 1.17 (pour un bloc de 8×8), formant ainsi les bandes de coefficient DCT.



**Figure 1.17 Le zig-zag scan des coefficients DCT**

### - Quantification :

Chaque bande DCT est uniformément quantifiée, en utilisant un quantificateur scalaire uniforme à  $2^M$  niveaux. Ensuite, les coefficients quantifiés sont transformés en bits. Les bits des coefficients quantifiés sont ensuite regroupés pour former des plans de bit.

### - Codage Turbo :

Chaque bande DCT est turbo-codé, en commençant par le bit de plan le plus significatif (MSB). Les bits de parité sont générés pour chaque plan de bits, stockés dans un buffer et ensuite envoyés au décodeur en morceaux à la demande du décodeur, par l'intermédiaire d'un canal de retour. Ces bits sont utilisés pour la correction de l'information adjacente (estimation des images WZ), qui est créé au niveau du décodeur.

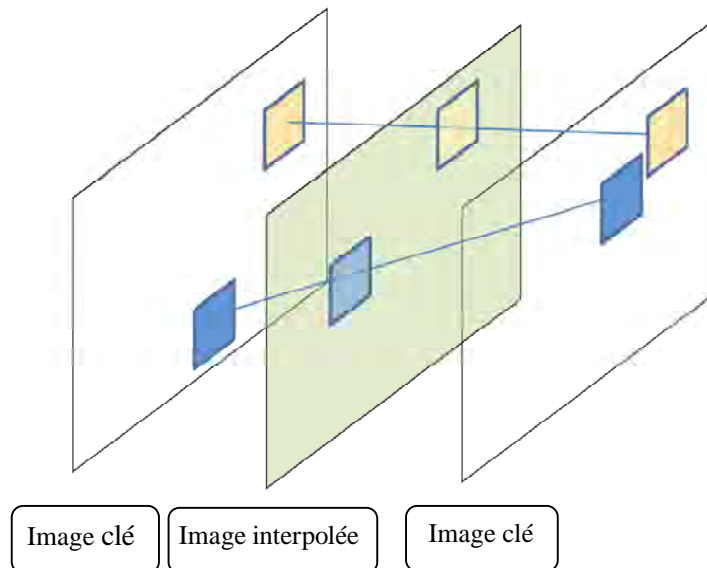
## 1.2. La procédure de décodage :

### Au niveau du décodeur :

#### - Création de l'information adjacente :

Le décodeur crée l'information adjacente qui sert comme une version estimée (ou "bruitée") des images WZ. Ceci est fait par une compensation de mouvement par l'interpolation, en

utilisant les images décodées qui sont les plus proches de l'image courante. Dans cette méthode, des vecteurs de mouvement sont créés au niveau du décodeur en utilisant une estimation de mouvement entre les deux images clés voisines (déjà décodés) où l'une est l'image clé précédente et l'autre est l'image clé suivante. Ces variétés améliorées sont ensuite utilisés pour la prédiction de l'image WZ. Ce processus est illustré à la figure 1.18.



**Figure 1.18. Compensation de mouvement par interpolation.**

**- Modélisation du bruit de corrélation :**

La différence statistique entre l'image source et l'information adjacente est modélisée par une distribution de Laplace. Le paramètre  $\alpha$  de la distribution laplacienne est estimé en utilisant une phase d'entraînement hors-ligne.

**- Turbo décodage:**

Chaque plan de bits (à partir du bit de plan le plus significatif) est Turbo décodé en utilisant l'information adjacente et le modèle de distribution de bruit. Le turbo-décodeur reçoit du codeur des morceaux successifs de bits de parité, 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 WZ sera effectuée via le canal de retour. La demande de bits de parité est répétée jusqu'à une probabilité d'erreur acceptable.

### - Reconstruction :

Après l'opération du turbo décodage, les plans de bits associés à leurs bandes de DCT sont regroupés, afin de former les symboles quantifiés décodés.

### - Transformée inverse :

Après que toutes les bandes DCT soient connues, le décodeur effectue une dé-quantification et une DCT inverse afin d'obtenir les images WZ décodées.

### - Réorganisation :

Après avoir décodé toute les images WZ, Les images clés sont décodées avec un décodeur en mode Intra traditionnel. Enfin, pour obtenir la séquence vidéo décodée, les images WZ et Les images clés décodées sont placées dans le bon ordre.

## 2. Le modèle CVD proposé par l'équipe de Berkeley (PRISM) :

L'approche proposée par l'équipe Berkeley est connu dans la littérature sous le nom PRISM (Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding) [24,25], fonctionne dans le domaine de transformation. Contrairement au codec Stanford, qui est dans le domaine pixel, l'approche Berkeley est basée sur des blocs. Chaque bloc dans une image peut être codé avec l'un des nombreux modes disponibles, en fonction de sa corrélation avec le bloc de prédiction. Un tel bloc peut être soit un bloc Co-localisé ou un bloc de compensation de mouvement [25]. Le concept du GOP n'est pas employé dans PRISM. Nous allons brièvement présenter l'architecture de l'encodeur et du décodeur de la solution PRISM pour le codage vidéo distribué.

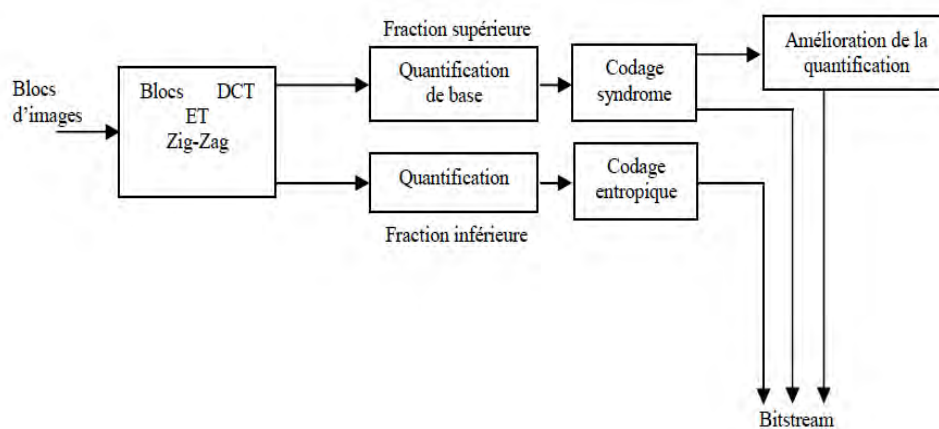


Figure 1.19 - Architecture du codeur PRISM [24].

La Figure 1.19 illustre l'architecture du codeur PRISM.

## 2.1. La procédure de codage :

### - Classification :

Chaque bloc de  $8 \times 8$  ou  $16 \times 16$  est classé dans l'une de trois classes prédéfinies en fonction de la corrélation statique entre le bloc de l'image courante et son correspondant dans l'image précédente. L'étape de la classification permet de décider quel type de codage est bien adapté pour chaque bloc de l'image courante: aucun codage (classe de saut), codage classique (classe de codage en mode Intra) et classe du codage par syndrome. Les classes de mode de codage choisies pour les blocs de l'image courante sont ensuite transmises au décodeur comme information d'en-tête.

### - Transformation DCT :

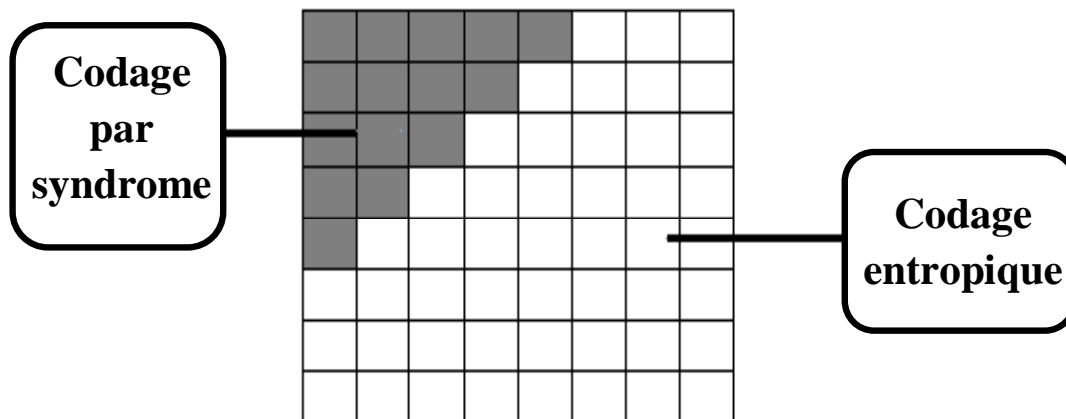
Une transformée DCT  $8 \times 8$  est appliquée à chaque bloc de l'image courante. Les coefficients de la transformée de chaque bloc sont par la suite scannés et regroupés en zig-zag.

### - Quantification :

Un Quantificateur scalaire est appliqué sur les coefficients DCT en fonction de la qualité de reconstruction souhaitée. Les coefficients DC (les basses fréquences) et un petit groupe des coefficients AC (qui sont proches des DC dans le zig-zag scanne) sont quantifiés avec un petit pas de quantification, puisque la plupart de l'énergie est concentrée dans cette zone et donc donne plus d'informations sur l'image que les autres coefficients, qui sont quantifiés avec un pas de quantification plus grand.

### - Codage par syndrome :

Dans ce mode, les coefficients de basse fréquence (généralement les 20% les plus bas, qui sont appelés les coefficients WZ) sont codées en utilisant le concept de syndrome. Comme l'indique la figure 1.20 Puisque le nombre de bits nécessaires pour coder un syndrome est inférieur au nombre de bits nécessaires pour coder un mot de code. Alors en utilisant le codage par syndrome (code BCH [26]), la transmission des mots de code associés au coefficient quantifiés est remplacée par la transmission des syndromes et donc la compression est réalisée.



**Figure 1.20 - Codage par syndrome et codage entropique dans un seul bloc de 8×8.**

**- codage entropique :**

Les coefficients de haute fréquence quantifiés, qui n'ont pas été codés par syndrome, sont alors codés par un codeur entropique traditionnel en utilisant par exemple un codeur à longueur variable (CLV) comme le codeur de Huffman.

**- Code CRC (Cyclic Redundancy Check):**

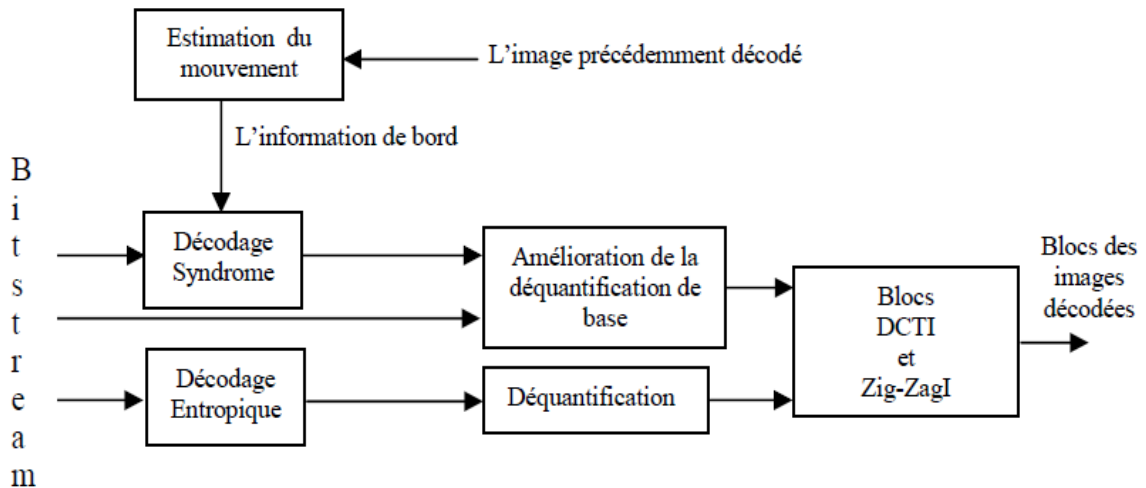
Un code CRC [27] est calculé pour tous les coefficients WZ dans chaque bloc. La longueur de ce code est habituellement considérée comme 16 bits. Ce code CRC est utilisé dans le processus de décodage pour sélectionner le meilleur candidat (l'information adjacente) dans le processus de décodage par syndrome.

**2.2. La procédure de décodage :**

La Figure 1.21 illustre l'architecture du décodeur PRISM.

**- Création de l'information adjacente :**

Chaque bloc constitué des coefficients WZ a plusieurs candidats (plusieurs informations adjacentes) qui sont obtenus à partir du bloc décodé précédemment en utilisant un algorithme d'EM à ½ pixel de précision, Autrement dit, tous les blocs voisins dans l'image précédente (dans une plage de recherche) peuvent toutes servir comme information adjacente.



**Figure 1.21 - Architecture du décodeur PRISM [24].**

**- Modélisation du bruit de corrélation :**

La différence statistique entre l'image source et l'information adjacente est modélisée par une distribution de Laplace. Le paramètre de distribution  $\alpha$  est estimé au cours d'une phase d'apprentissage hors ligne.

**- Décodage par syndrome :**

Pour chaque bloc de  $8 \times 8$  ou  $16 \times 16$ , les bits de syndrome reçus avec l'un des candidats sont utilisés pour décoder la séquence de mots de code quantifiée, à partir des bits de syndromes reçus, il est possible d'obtenir plusieurs séquences de mots de code quantifiés. L'algorithme de Viterbi [28] est utilisé pour trouver, dans cet ensemble des séquences de mots de code quantifiés, la séquence la plus proche de l'information adjacente.

Si la séquence la plus proche identifiée ne correspond pas au CRC reçu, le processus de décodage par syndrome est effectué à nouveau en utilisant une autre information adjacente générée par une recherche de mouvement. Le processus de décodage par syndrome s'arrête lorsque la séquence la plus proche identifiée correspond au CRC reçu. Le CRC est utilisé comme une signature fiable et unique pour chaque bloc et permet d'identifier la meilleure information adjacente.

**- Décodage entropique:**

Une fois que les coefficients WZ quantifiés sont reconstruits, les coefficients codés en mode entropique (hautes fréquences) sont intra décodés.

### - Transformée inverse :

La Dé-quantification et la DCT inverse sont appliquées sur les coefficients WZ quantifiés décodés.

## 3. Le modèle CVD proposé par l'équipe de DISCOVER:

Le codec DISCOVER [29] est probablement l'un des codecs CVD les plus efficaces en distorsion de débit qui sont actuellement disponibles. Le codec DISCOVER (DIStributed COding for Video sERvices) est basé sur l'architecture CVD de Stanford, avec une meilleure génération de l'information adjacente, l'utilisation d'un codeur LDPC et un CRC de 8 bits pour la vérification. La figure 1.22 illustre l'architecture de codage proposé par l'équipe de DISCOVER. Une description de l'opération du codage et du décodage du codec DISCOVER est donnée ci-dessous.

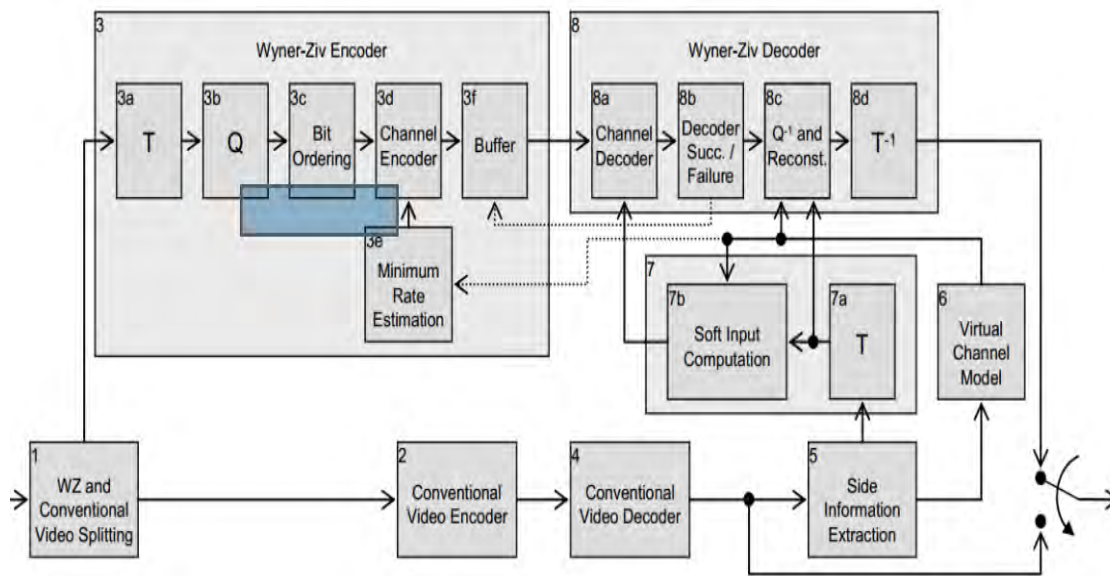


Figure 1.22 : Le CODEC DISCOVER [29]

### 3.1. La procédure de codage :

#### - Classification :

Un algorithme adaptatif de sélection de la taille du GOP [30] contrôle l'insertion des images clés entre les images WZ. Cet algorithme construit des GOPs, formé chacun par les images les plus corrélés. La première image dans chaque GOP est une image clé.

#### **- Transformation DCT :**

Chaque image WZ est d'abord transformée en utilisant une DCT  $4 \times 4$ , comme celle utilisée dans le H.264 [11]. Cette transformation est plus efficace que la DCT ordinaire et réduit les erreurs de décodage. Les coefficients DCT obtenus sont ensuite organisés en 16 bandes.

#### **- Quantification :**

Chaque bande DCT  $B_k$  ( $k = 1, 2, \dots, 16$ ) est uniformément quantifiée, en utilisant un quantificateur scalaire uniforme avec  $2^{M_k}$  niveaux. Le nombre de niveaux de quantification varie généralement d'une bande à l'autre. Après que les coefficients quantifiés seront transformés, ils seront ensuite regroupés pour former les plans de bits.

#### **- Codage LDPCA :**

Pour chaque plan de bits, les bits de syndrome sont créés en utilisant un codeur LDPC Accumulateur, Le codeur WZ stocke ces syndromes accumulés dans une mémoire tampon (buffer) et le transmet initialement que quelques-uns d'entre eux en paquets. Ces paquets initiaux correspondent au débit théorique minimal  $R_{min}$ . Si le décodeur WZ échoue, alors plusieurs bits de syndrome seront demandés par le buffer du codeur en utilisant un canal de retour. Pour faciliter la détection des erreurs résiduelles au décodeur un CRC de 8 bits est également transmis.

### **3.2. La procédure de décodage :**

#### **- Création de l'information adjacente :**

Le décodeur construit l'information adjacente pour chaque image WZ codée en utilisant l'estimation de mouvement et la compensation de mouvement par interpolation entre les images clés précédentes et les suivantes. Les vecteurs de mouvement obtenus sont lissés à l'aide d'un filtre médian pour améliorer la cohérence.

#### **- Modélisation du bruit de corrélation :**

Le codec DISCOVER utilise une distribution de Laplace pour modéliser le bruit de corrélation (distribution d'erreurs entre les bandes DCT de l'information adjacente et les images WZ). Le paramètre de distribution laplacienne  $\alpha$  est estimé en ligne au niveau du décodeur et prend en compte la variabilité spatiale et temporelle des statistiques de bruit de corrélation.



### **Décodage LDPC :**

Le modèle de distribution laplacienne est ensuite utilisé pour convertir les coefficients DCT de l'information adjacente en entrées souples (soft-input) dans le décodeur LDPC. Ensuite le décodeur LDPC corrige les erreurs de bits de l'information adjacente en utilisant la procédure de propagation de croyance (belief propagation).

Pour savoir si le décodage est réussi ou non, la distance de Hamming est calculée entre le syndrome reçu et celui généré à l'aide du plan de bits décodé suivie d'un CRC. Si la distance de Hamming est nulle et la somme de contrôle calculée par le CRC sur le plan de bits décodé est égale à la valeur reçue par le codeur, alors le décodage est déclaré réussi. Sinon le décodeur LDPCA demande plus de syndromes par l'intermédiaire du canal de retour jusqu'à ce que le décodage réussisse.

### **- Reconstruction :**

Après la réussite du décodage les valeurs quantifiées originales sont reconstruites, ensuite la dé-quantification et la transformation DCT inverse sont appliquées pour avoir les valeurs réelles de l'image WZ.

## **5. Conclusion :**

Les techniques de codage traditionnelles telles que MPEG.X et H.26X ne sont pas bien adaptées aux nouveaux besoins technologiques à cause de la complexité de calcul au niveau du codeur. Comme nous avons vu dans ce chapitre le CVD permet de basculer la complexité vers le décodeur pour avoir un codeur le plus simple possible. Les modèles les plus intéressants qui se trouvent sur le marché, sont : Stanford , PRISM et DISCOVER. Ce dernier est celui qui a gagné le plus de popularité, parce qu'il est très efficace et fournit une meilleure génération de l'information adjacente. Dans le chapitre qui suit nous allons présenter notre approche pratique du CVD et son application sur les séquences périodiques.

## **Chapitre 2**

### **Implémentation d'un CVD pour des séquences vidéo périodiques.**

## Introduction :

Dans ce chapitre nous allons nous intéresser beaucoup plus au côté pratique c'est à dire l'implémentation du CVD, mais avec notre propre approche adaptée pour les séquences vidéo périodiques. Cette approche consiste en une nouvelle technique de génération de l'information adjacente. Dans la première section on justifiera l'utilisation des mouvements périodiques, ensuite nous présenterons notre approche CVD, avec une nouvelle technique de génération de l'information adjacente adaptée pour ces séquences vidéo périodiques.

## 1. Le mouvement périodique :

### 1.1. Définition :

Un mouvement est dit périodique lorsqu'il se répète identique à lui-même au bout d'un intervalle de temps réguliers, souvent noté  $T$ , appelé période. On dit souvent qu'au cours d'une période le système effectue un cycle. Ce phénomène est très répandu dans la nature et dans notre vie quotidienne, tels que la rotation de la terre sur elle-même et sur le soleil, oscillation d'un pendule, mouvement des aiguilles d'une montre, battement du cœur, mouvement d'une caméra de surveillance ou même un satellite qui gravite autour d'une planète ..... etc.

### 1.2. Exemples des phénomènes périodiques :

#### 1.2.1. Les Battements du cœur (rythme cardiaque) :

Le rythme cardiaque est le nombre de battements cardiaques (pulsations) par unité de temps (généralement la minute). Comme illustré sur la figure 2.1.



Figure 2.1. Rythme cardiaque d'un Homme en bonne santé.

Selon les espèces animales, la fréquence cardiaque est très inégale, chez la baleine, le plus gros mammifère actuel, la fréquence cardiaque est inférieure à 20 battements par minute, chez l'Homme elle est d'environ 80 battements par minute. Ainsi, le rythme cardiaque est considéré comme un phénomène périodique. Partons de l'idée, que tous les humains en bonne santé ont un rythme cardiaque 80 battements par minute, on peut utiliser cette caractéristique comme information adjacente, et ensuite l'exploiter dans un système CVD.

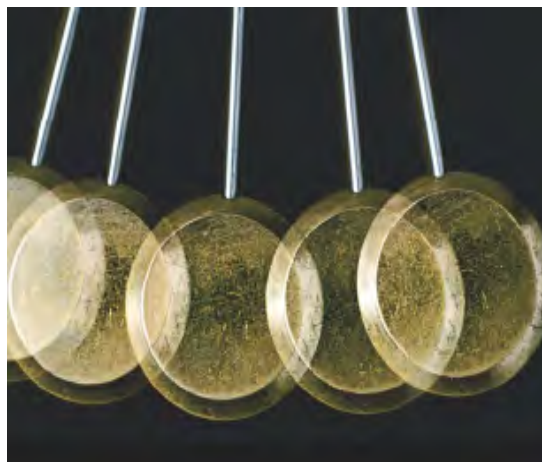
### **1.2.2. Camera de surveillance :**

Quand la camera de surveillance filme une séquence, elle réalise un mouvement périodique de droite → gauche et de gauche → droite, ainsi nous pourrions exploiter les séquences vidéo périodiques acquises dans notre modèle CVD.



**Figure 2.2 Camera de surveillance qui filme des séquences vidéo périodique.**

### **1.2.3. Oscillation d'un pendule :**

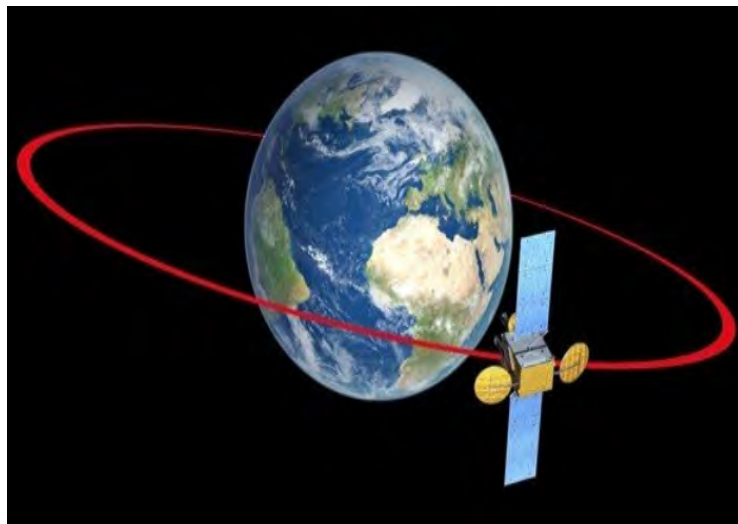


**Figure 2.3. Le mouvement périodique réalisé par l'oscillation du pendule.**

Le pendule est l'un des phénomènes périodiques les plus connus en physique, composé d'un solide de petite dimension fixée à l'extrémité d'un fil et oscillant sous l'effet de la pesanteur. Comme illustré sur la figure 2.3.

#### **1.2.4. Satellite artificiel :**

Un satellite artificiel est un objet fabriqué par l'être humain, envoyé dans l'espace à l'aide d'un lanceur et gravitant autour de la terre ou d'une autre planète. La vitesse imprimée par la fusée au satellite lui permet de se maintenir pratiquement indéfiniment dans l'espace en décrivant une orbite autour du corps céleste. Celle-ci, définie en fonction de la mission du satellite, peut prendre différentes formes : géostationnaire, elliptique et circulaire. Ce qui rend le satellite l'un des phénomènes périodiques les plus intéressants à étudier.



**Figure 2.4. Rotation d'un satellite autour de la terre.**

Dans notre cas, nous sommes intéressés par les satellites qui ont pour mission de prise d'images de la Terre ou d'autres planètes. De très nombreux usages (scientifiques, civils et militaires) sont faits de ces images.

#### **1.2.5. D'autres exemples du phénomène périodique :**

La figure 2.5 illustre d'autres exemples des phénomènes périodiques, qu'on trouve dans notre vie quotidienne, tels que la machine à coudre, les aiguilles de la montre, les moulins à vent ....etc.



**2.5. Exemples des phénomènes périodiques.**

## **2. Présentation d'un CVD pour des séquences périodique :**

Dans cette section nous allons présenter une nouvelle approche de codage vidéo distribué. On a vu dans le chapitre précédent les différentes techniques utilisées dans la littérature pour la génération de l'information adjacente comme l'interpolation et l'extrapolation. Cependant, le grand inconvénient de ces techniques c'est la complexité de l'implémentation avec leur coût de calcul énorme ce qui donne un système CVD complexe. Dans notre modèle CVD une nouvelle technique de génération de l'information adjacente va être présentée, Une technique simple mais qui reste très efficace.

Ce qui distingue notre modèle CVD de tous les modèles qu'on trouve dans la littérature c'est l'absence de classification des images. Dans toutes les approches présentées dans le chapitre 1, tel que le CVD proposé par l'équipe de stanford ou l'équipe de Berkeley ou même dans l'architecture récente proposée par l'équipe de Discover, la séquence vidéo est divisé en deux types d'images, le premier type ce sont les images clés qui vont être codées en mode intra par un codeur classique généralement le H.26X ou le MPEG, et le deuxième type d'images ce sont dites images WZ qui vont être codées en mode codage de source. Dans notre proposition les images clés n'existent pas. Il y'a juste les images WZ, ce qui rend le codage beaucoup plus simple. Mais cette approche est applicable qu'aux séquences vidéo

périodiques, c'est à dire des séquences vidéo qui réalisent un mouvement périodique tel que les caméras vidéo de surveillance, les battements du cœur, le pendule, un satellite qui tourne de façon circulaire et régulière sur la terre ....etc. notre technique utilisée pour générer l'information adjacente va être détaillée dans la section qui suit.

### **3. Idée de base de notre travail :**

Différemment de ce qui est présente dans la littérature, où l'information adjacente est créée après le codage, dans notre cas l'information adjacente est créée avant la procédure de codage et de décodage. Ceci constitue l'originalité et l'apport de notre travail au codage vidéo distribué.

Nous savons bien qu'il existe beaucoup de modèles de caméra de surveillance, mais le modèle qui nous intéresse c'est la camera de surveillance dynamique qui bouge de gauche → droite et de droite → gauche en réalisant un mouvement périodique.

Quand la caméra tourne, elle filme une séquence vidéo périodique par exemple de durée de 2 minutes (120 sec) , et elle réalise par exemple des périodes de 15 secondes chacune, ce qui veut dire qu'on a une séquence vidéo composée de 8 périodes (  $120 \text{ sec} \div 15 \text{ sec} = 8 \text{ per}$  ) , en d'autre terme la camera va revenir sur le même endroit chaque 15 sec. Dans notre cas, on commence par filmer la scène par un balayage d'une période. Ces images (période) vont servir d'information adjacente. La figure 2.6 va nous permettre d'expliquer comment la séquence utilisée comme information adjacente va être exploitée pour reconstruire la séquence vidéo.

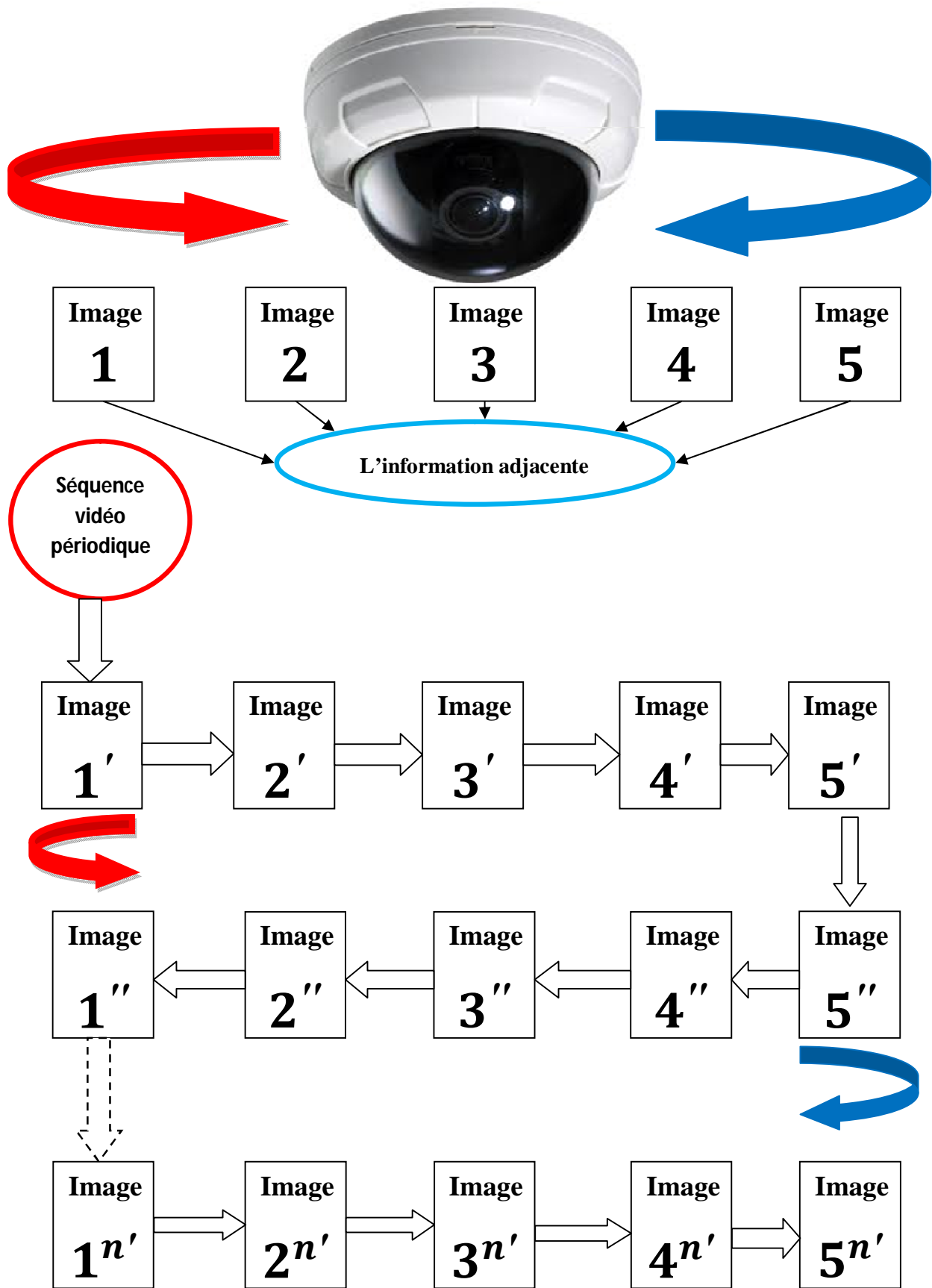


Figure 2.6. Exploitation de l'information adjacente pour le codage de la séquence vidéo périodique.



La caméra réalise un mouvement périodique de droite → gauche (de l'image 1' vers l'image 5') et de gauche → droite (de l'image 5' vers l'image 1'), dans cet exemple nous avons imposé que la séquence vidéo soit composée de 5 images par période et que la camera filme une seule image par seconde juste pour simplifier la présentation.

L'information adjacente c'est la période (image 1 image 5) prise par la caméra vidéo quand la scène est vide (absence de mouvement extérieur). Ensuite cette période est codée en mode intra par un codage classique (H264), et elle est stockée dans le décodeur comme information adjacente. Chaque image WZ est alors décodée par rapport à sa correspondante dans l'information adjacente. Par exemple l'image WZ numéro 1' est reconstruite en utilisant l'image numéro 1 de l'information adjacente et l'image WZ numéro  $4^n$ ' est reconstruite en utilisant l'image numéro 4 de l'information adjacente est ainsi de suite. Dans ce qui suit nous allons détailler la procédure de codage et décodage.

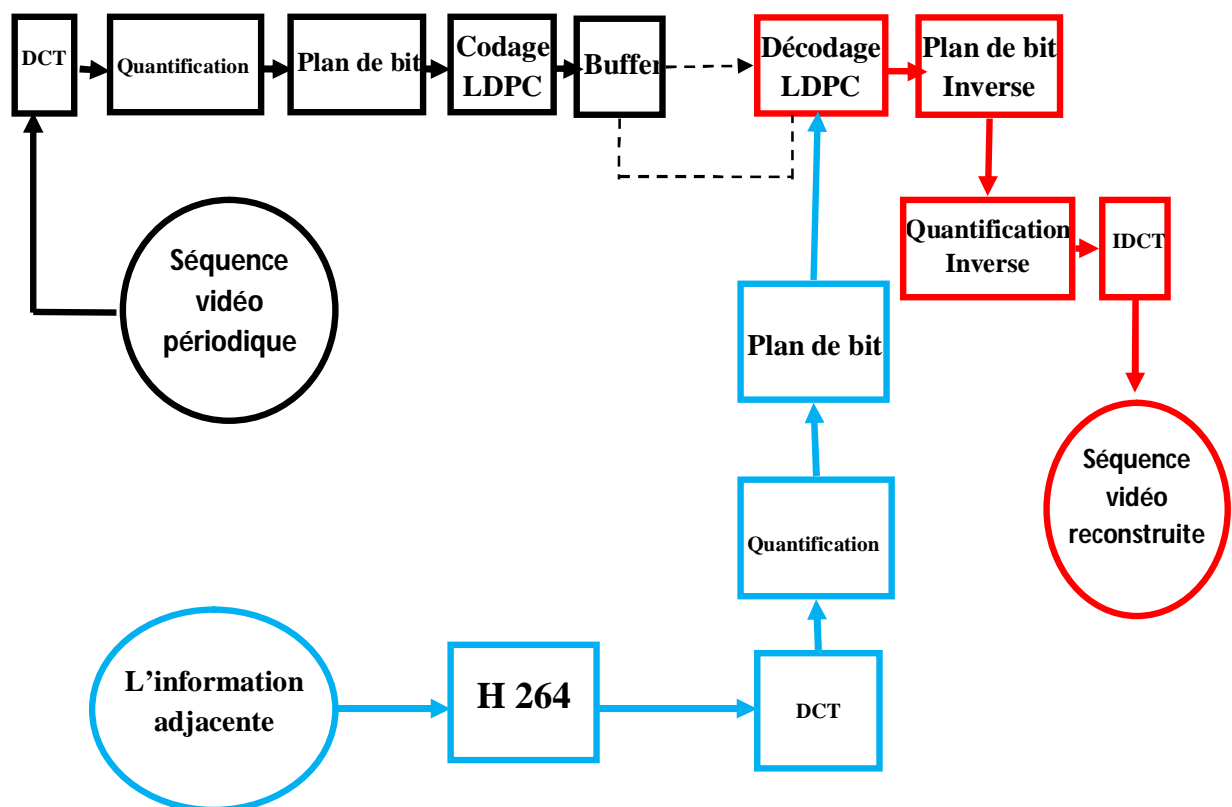


Figure 2.7. L'architecture de codage et décodage CVD proposée par notre approche.

#### 4. La procédure de codage :

On rappelle que dans notre cas il n'y a pas de partition des images. Nous travaillerons qu'avec des images WZ, c'est-à-dire qui seront codées WZ.

Considérons l'architecture de codage illustrée par la Figure 2.7. La procédure de codage passe par les étapes suivantes.

- 1- Transformation DCT.
- 2- Quantification.
- 3- Conversion en plan de bits.
- 4- Codage LDPC.

Chaque étape va être brièvement expliquée.

## **4.1 Transformation DCT:**

La transformée DCT est largement utilisée par les normes de codage vidéo traditionnels qu'on trouve dans l'état de l'art. Dans notre architecture, on applique la DCT utilisé par le standard de codage vidéo traditionnel H.264 AVC [31], sur des blocs  $4 \times 4$ .

## **4.2 Quantification :**

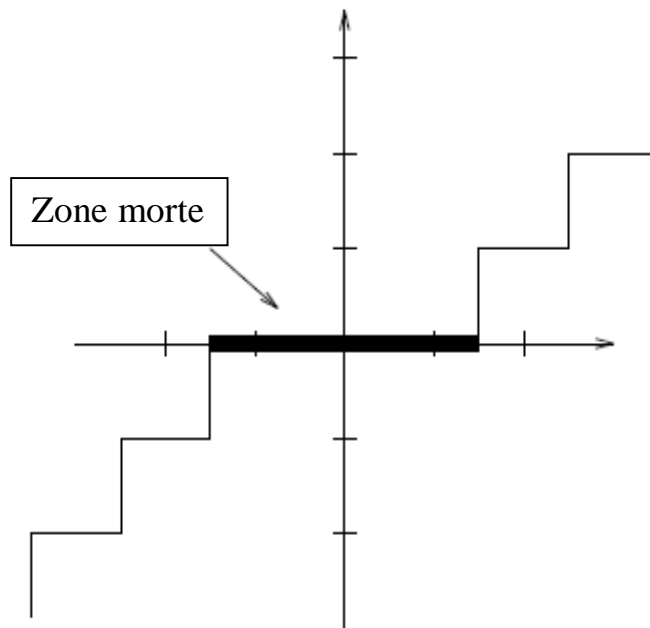
Après l'application de la transformée DCT au niveau du codeur, chaque bande DCT est quantifié en utilisant séparément avec un nombre prédéfini de niveaux (pas de quantification), en fonction de la qualité voulue des images WZ.

- Les coefficients DCT représentant les fréquences spatiales les plus faibles sont quantifiés en utilisant un quantificateur scalaire uniforme avec des petits pas de quantification, tandis que les hautes fréquences seront quantifiées avec des pas de quantifications plus grandes, sans dégradation significative de la qualité visuelle de l'image décodée.
- Le nombre de niveaux de quantification par bande de coefficients DCT est supposé être connu par le codeur et le décodeur.
- Un quantificateur scalaire uniforme est utilisé pour quantifier la bande de coefficients DC, qui sont des valeurs positives et représentent les valeurs moyennes dans chaque block de  $4 \times 4$ , alors ils sont quantifiés avec un intervalle dynamique entre  $[0 \text{ et } 2^{11}(2048)]$ .
- les bandes restantes, appelées bandes de coefficients AC, contiennent des valeurs positives et d'autre négatives.
  
- Les bandes AC sont quantifiées en utilisant un quantificateur à zone morte (dead-zone quantizer), pour avoir la possibilité de bloquer les coefficients les plus faibles (avoir beaucoup de zéros), ce qui permet d'améliorer la performance du codage WZ. La figure 2.8 illustre un

quantificateur à zone morte. L'intervalle dynamique  $[-\text{MaxVal}_b ; \text{MaxVal}_b]$  est calculé séparément pour chaque bande AC et transmis au décodeur, Le pas de quantification est donné par :

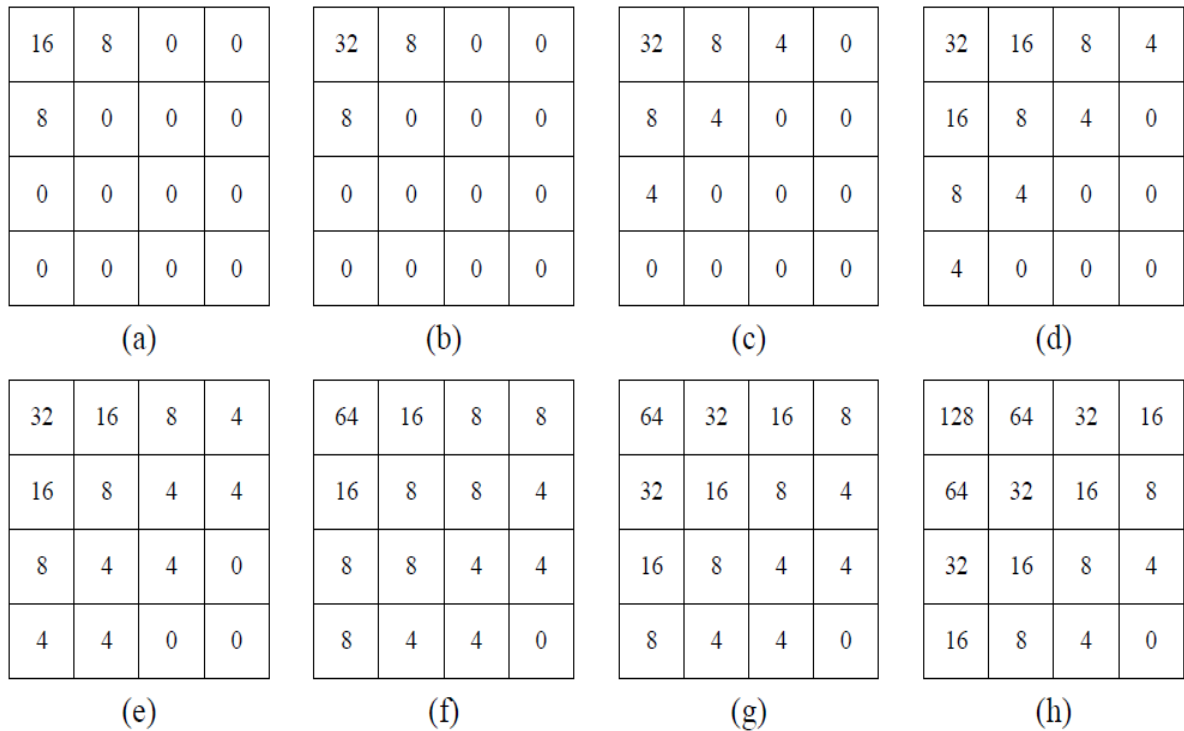
$$W_b = \left\lceil \frac{2 \text{MaxVal}_b}{2^{M_b}} \right\rceil \quad (2.1)$$

Où  $2^{M_b}$  représente le niveau de quantification de chaque bande AC.



**Figure 2.8 quantificateur à zone morte (dead-zone quantizer)**

En fonction de la qualité de codage voulue, 8 points débit-distorsion ont été considérés, où chaque point correspond à une qualité de codage. Chaque matrice de  $4 \times 4$  illustrée dans la figure 2.9, correspond à un point de débit-distorsion, c'est à dire que chaque matrice correspond à une qualité de codage.



**Figure 2.9 Huit matrices de quantifications associées à des différentes performances de distorsion de débit.**

- La matrice (a) représente le débit binaire le plus bas (la qualité de codage la plus faible) et la plus haute en terme débit-distorsion, tandis que la matrice (h) représente le débit binaire le plus haut (la meilleur qualité de codage) et la plus basse en terme débit-distorsion.
- Dans chaque matrice de quantification de 4×4 la valeur à chaque position dans la figure 2.9, indique le nombre de niveaux de quantification associé à la bande de coefficients DCT correspondante
- Dans la figure 2.9, la valeur 0 signifie qu'aucun bit WZ n'est transmit au décodeur pour la bande correspondante, le décodeur va remplacer cette valeur par la valeur de l'information adjacente disponible au niveau du décodeur.

### 4.3. Conversion en plan de bits :

Après la quantification de toutes bandes des coefficients DCT, les symboles quantifiés (représentants des valeurs entières) sont convertis en flux binaire, Les bits de symboles quantifiés sont regroupés ensemble en fonction de leur position binaire en commençant par les bits les plus significatifs, qui seront ensuite codés indépendamment par un codeur LDPC.

#### 4.4. Codage de canal (Codeur LDPC et Buffer) :

Pour le codage de canal, on utilise les codes LDPCA [32], car ils sont plus efficace et plus performant que les turbo code pour le codage à faible densité.

Un codeur LDPCA se compose d'un codeur LDPC qui est un créateur de syndrome et un accumulateur. Pour chaque plan de bits, les syndromes sont créés en utilisant le code LDPC ensuite ils seront accumulé modulo 2 pour produire les syndromes accumulés. Après le codeur WZ stocke ces syndromes accumulés dans un buffer (mémoire tampon), et ne transmet au début que quelques-uns d'entre eux en paquets. Ces paquets initiaux correspondent au débit théorique minimal  $R_{min}$ , si le décodeur n'arrive pas à accomplir l'opération de décodage avec succès, alors il va demander plus de syndromes accumulés à partir de la mémoire tampon (buffer) en utilisant le canal de retour.

#### 5. La procédure de décodage :

la procédure de décodage passe par les étapes suivantes .

- 1- Modélisation du bruit.
- 2- Décodage LDPC.
- 3- Dé-quantification et reconstruction.
- 4- Transformation DCT inverse.

Chaque étape va être brièvement expliquée.

##### 5.1 Modélisation du bruit:

Dans un codeur CVD, l'efficacité du décodage des images WZ dépend de façon critique de la capacité de modéliser la corrélation statistique entre l'information WZ au codeur et l'information adjacente disponible au niveau du décodeur. Il s'agit d'une tâche très complexe, puisque l'information originale n'est pas disponible au niveau du décodeur, et que la qualité de l'information adjacente (la version bruitée de l'information originale) varie tout au long de la séquence vidéo c-à-dire que la distribution de l'erreur n'est pas constante dans le temps.

Dans notre application nous utilisons une distribution Laplacienne pour modéliser le bruit de corrélation (la distribution de l'erreur entre les bandes DCT de l'information adjacente et les bandes DCT des images WZ). La distribution de Laplace est donné par :

$$f(x) = \frac{\alpha}{2} e^{-\alpha|x|} \quad (2.2)$$

### 5.1.1. Estimation du paramètre $\alpha$ :

Le paramètre de la distribution laplacienne  $\alpha$  est estimé online au niveau du décodeur et prend en considération les changements temporels et spatiaux des statistiques du bruit de corrélation, cette technique est plus efficace que la technique qui estime le paramètre  $\alpha$  hors-ligne c'est-à-dire qui a été estimé au préalable sur un grand nombre d'images. Dans notre application le paramètre  $\alpha$  est estimé comme suit :

Pour le décodage de l'image WZ (la séquence vidéo périodique) numéro 1 on exploite la différence entre l'image numéro 2 et l'image numéro 1 de l'information adjacente, puisque au début nous n'avons aucune information sur la séquence originale. Après le décodage de la première image, on peut accéder à cette information. Pour l'estimation du paramètre  $\alpha$  de la deuxième image, on exploite la différence entre l'image numéro 2 de l'information adjacente et l'image 1 récemment décodée et ainsi de suite pour toutes les images. Ex, pour l'estimation du paramètre  $\alpha$  de l'image numéro 20 on exploite la différence entre l'image numéro 20 de l'information adjacente et l'image 19 déjà décodée.

Ensuite le modèle de distribution laplacienne est utilisé pour convertir les coefficients DCT de l'information adjacente en informations d'entrées souples (soft-input) pour le décodeur LDPC.

### 5.2. Décodage LDPC (correction):

En utilisant les entrées souples (soft-input) calculées par le modèle de canal virtuel, le décodeur de canal (le décodeur LDPC), connu aussi sous le nom décodeur WZ, corrige les bits erronés de l'information adjacente en utilisant la procédure de propagation de croyance **belief propagation** sur les premiers paquets de syndromes accumulés envoyés par la mémoire tampon (buffer) du codeur.

Pour déterminer la réussite de l'opération de décodage une vérification d'erreur de syndrome (syndrome check error) est calculée, .i.e. la distance de Hamming entre le syndrome reçu et celui généré par le plan de bits décodé,

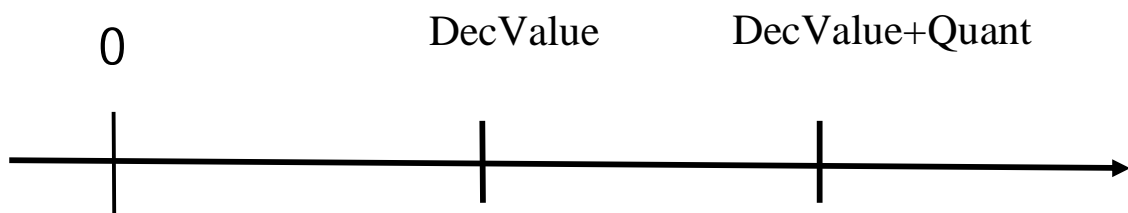
Si la distance de Hamming est différente de zéro, alors le décodeur passe à l'itération suivante. Après un certain nombre d'itérations, si la distance de Hamming reste différente de zéro, alors le plan de bits décodé est supposé erroné et le décodeur LDPCA demande plus de syndromes

accumulés par l'intermédiaire du canal de retour. Si la distance de Hamming est égale à zéro, et la valeur de plan de bit décodée égale à celle reçue par le décodeur, alors le décodage est déclaré réussi. Sinon le décodeur demande plus de syndromes accumulés jusqu'à la réussite du décodage.

### 5.3. Dé-quantification et reconstruction :

Après la réussite du décodage de tous les plans de bit de chaque bande, une conversion inverse de plan de bits avec une dé-quantification sont appliquées et les symboles quantifiés sont reconstruits, mais ces valeurs sont juste une approximation des vraies valeurs de la séquence originale. Alors pour avoir les valeurs originales le décodeur va procéder comme suit :

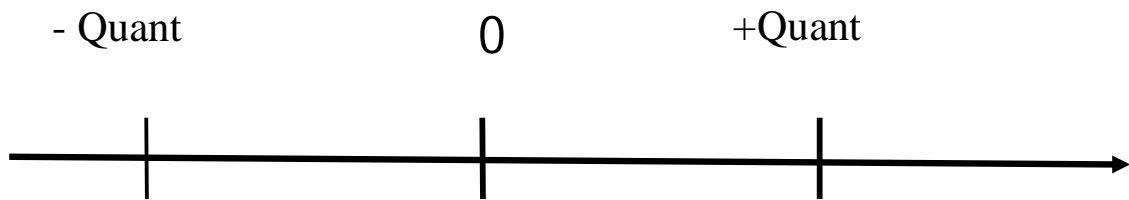
#### 1. Cas1 (DecValue > 0) :



Où DecValue est la valeur décodée et **Quant** est le pas de quantification, avec Y la valeur de l'information adjacente.

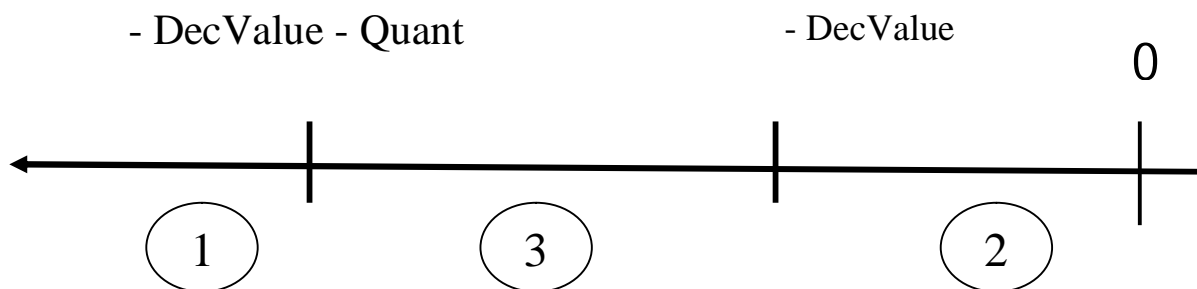
$$\text{DecValue} = \begin{cases} Y & \text{si } \text{DecValue} \leq Y < \text{DecValue} + \text{Quant} \\ \text{DecValue} & \text{si } Y < \text{DecValue} \\ \text{DecValue} + \text{Quant} & \text{si } Y \geq \text{DecValue} + \text{Quant} \end{cases}$$

**2. Cas2 (DecValue = 0):**



$$\text{DecValue} = \begin{cases} + \text{Quant}, & \text{si } Y \geq \text{Quant} \\ -\text{Quant}, & \text{si } Y \leq -\text{Quant} \\ Y & \text{si } -\text{Quant} < Y < +\text{Quant} \end{cases}$$

**3. Cas3 (DecValue < 0):**



$$\text{DecValue} = \begin{cases} Y & \text{si } -\text{DecValue} - \text{Quant} < Y \leq -\text{DecValue} \\ \text{DecValue} & \text{si } Y \geq -\text{DecValue} \\ \text{DecValue} + \text{Quant} & \text{si } Y < -\text{DecValue} - \text{Quant} \end{cases}$$

**5.4. DCT Inverse :**

Afin de reconstituer l'image WZ une DCT inverse de 4x4 est appliquée sur toutes les bandes afin d'obtenir les images WZ décodées.



## **6. Conclusion :**

Dans ce chapitre nous avons présenté une nouvelle technique de génération de l'information adjacente, une technique très simple si on la compare avec les techniques trouvées dans la littérature comme l'interpolation et l'extrapolation, qui sont plus complexes puisque elles font appel à une opération d'estimation de mouvement, qui est très complexe en terme de calcul.

# **Chapitre 3**

## **Résultats et discussion**

## **Introduction :**

Dans ce chapitre nous allons discuter les résultats expérimentaux de notre modèle CVD, sachant que le codeur et le décodeur sont tous les deux implémentés avec le langage de programmation C++, Afin d'évaluer les performances de débit-distorsion du codec proposé trois séquences d'images ont été proposés. Puisque notre proposition repose sur l'utilisation des séquences périodiques nous avons conçus ces trois séquences nous même, car dans la littérature on ne trouve pas des séquences de test périodiques. Dans les sections qui suivent nous allons donnés une présentation de ces séquencés de test.

Certaines conditions de test sont communes pour tous les processus d'évaluation des performances, ainsi ils seront indiqués une seule fois afin d'éviter les répétitions chaque fois qu'une séquence de test est considérée. Les conditions de test les plus communes sont listées ci- dessous :

- Seules les données de luminance Y sont considérées pour l'évaluation des performances de notre codec, parce que les données de luminance nous donnent suffisamment d'information sur la séquence vidéo.
- Les 8 matrices de quantification illustrées dans la figure 3.14 sont utilisées pour obtenir huit points de débit-distorsion différents.
- pour la génération de l'information adjacente on utilise la technique montrée dans le chapitre précédent, on prend une seule période de chaque séquence, on la code avec le codeur traditionnel H.264, ensuite on la stocke dans le décodeur.
- Pour chaque séquence de test, les performances débit-distorsion de notre codec sont comparées avec le mode intra du codeur à H.264.

Dans les sections qui suivent nous allons discutés et analysés les résultats obtenus pour les trois séquences.

## 1. L'évaluation de la séquence de test du train :

Avant de passer aux discussions et analyses des résultats, nous allons donner une description de cette séquence de test.

### 1.1. La séquence de test du train :

La séquence du train est une séquence de test périodique QCIF, elle est acquise par une caméra fixée en haut qui suit un train qui tourne sur sa piste en formant un mouvement périodique, comme illustré dans la figure 3.1.



(a)



(b)




(c)



(d)

**Figure 3.1 illustration du mouvement périodique réalisé par le train.**

Comme on peut voir dans la figure 3.1, la caméra est fixée en haut et filme un train qui réalise un mouvement périodique en tournant, partant du point de départ (a) passant par (b), (c) et (d) et ensuite il va retourner en (a) et ainsi de suite pour toute la séquence vidéo. Le tableau 3.1 présente les caractéristiques principales de cette séquence.

Nom de la séquence vidéo	Train
	
Nombre total d'images	728
Nombre total des périodes	4
Nombre images évalués	364
Nombre des périodes évalués	2
Résolution spatiale	QCIF
Résolution temporelle (Image/Seconde)	15

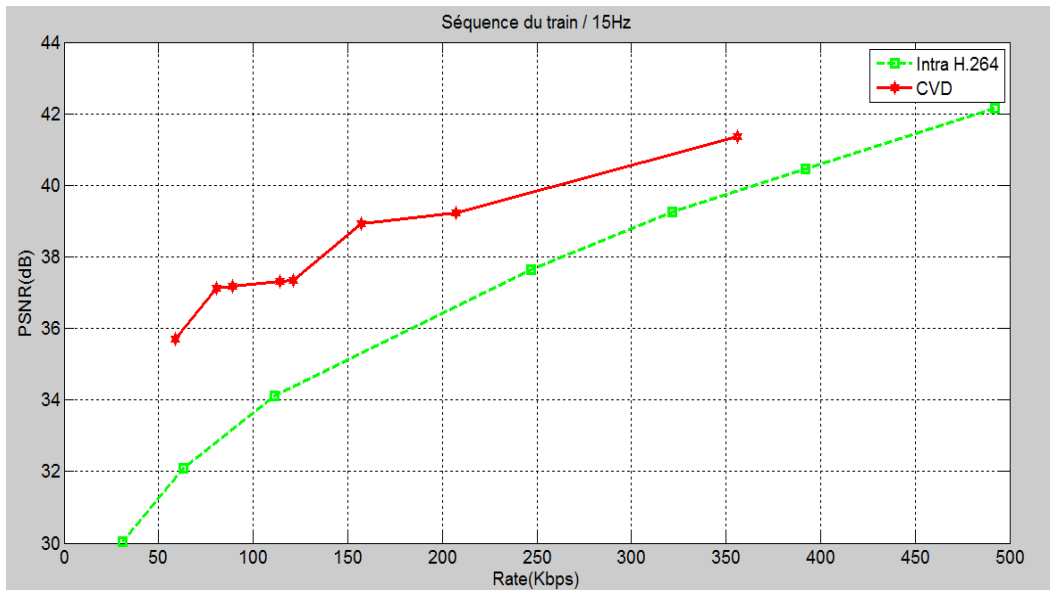
**Tableau 3.1 les caractéristiques de la séquence du train**

## 1.2. Evaluation des résultats:

La figure 3.2 illustre les résultats du débit-distorsion de la séquence du train.

Comme on peut voir dans la figure 3.2, les résultats obtenus par notre modèle CVD surpassent ceux du H.264 pour les 8 points de débit-distorsion, c'est-à-dire pour les basses, moyennes et hautes fréquences.

Pour les basses et moyennes fréquences, le PSNR obtenu par le CVD est meilleur d'environ 3 jusqu'à 4 décibels (en moyenne) que celui obtenu par le H.264, tandis que pour les hautes fréquences on remarque une petite dégradation du PSNR, qui est meilleur d'environ 1.8 décibels que celui obtenu par le H.264, à cause de la rapidité de mouvement du train.



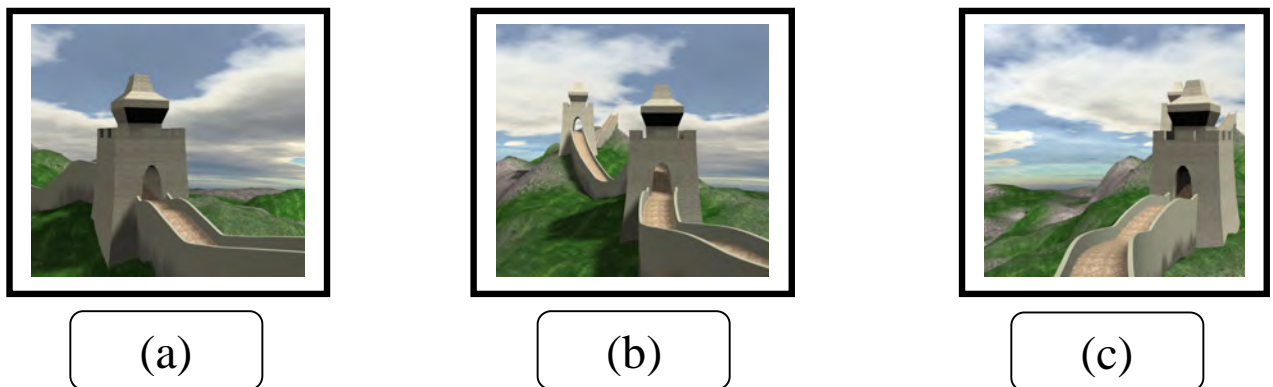
**Figure 3.2. Performance débit-distorsion de la séquence du train.**

## 2. Evaluation de la séquence de la grande muraille de chine:

### 2.1. Caractéristique de la séquence de la grande muraille:

Cette séquence est une séquence 3D est conçue par l'Autodesk 3ds Max, qui est un logiciel de modélisation et d'animation 3D. Ici la camera filme par balayage de gauche → droite et de droite → gauche, réalisant ainsi un mouvement périodique, comme illustré par la figure 3.3.

Pour cette séquence de test nous avons construit deux séquences périodiques, une originale et l'autre modifiée (des nouveaux objets entrent dans la scène + bruit), l'original va servir comme information adjacente et la deuxième va être codée par notre codeur WZ.



**Figure 3.3. Illustration de La séquence de la grande muraille (seulement la première période).**

Le tableau 3.2 présente les caractéristiques principales de cette séquence.

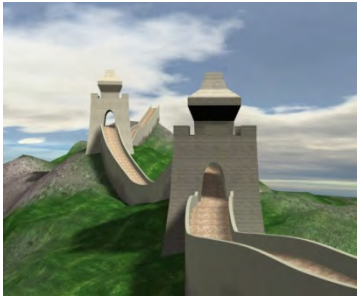
Nom de la séquence vidéo	La grande muraille
	
Nombre total d'images	480
Nombre total des périodes	3
Nombre images évalués	320
Nombre des périodes évalués	2
Résolution spatiale	QCIF
Résolution temporelle (Image/Seconde)	30

Tableau 3.2 les caractéristiques de la de la grande muraille.

## 2.2. Evaluation des résultats:

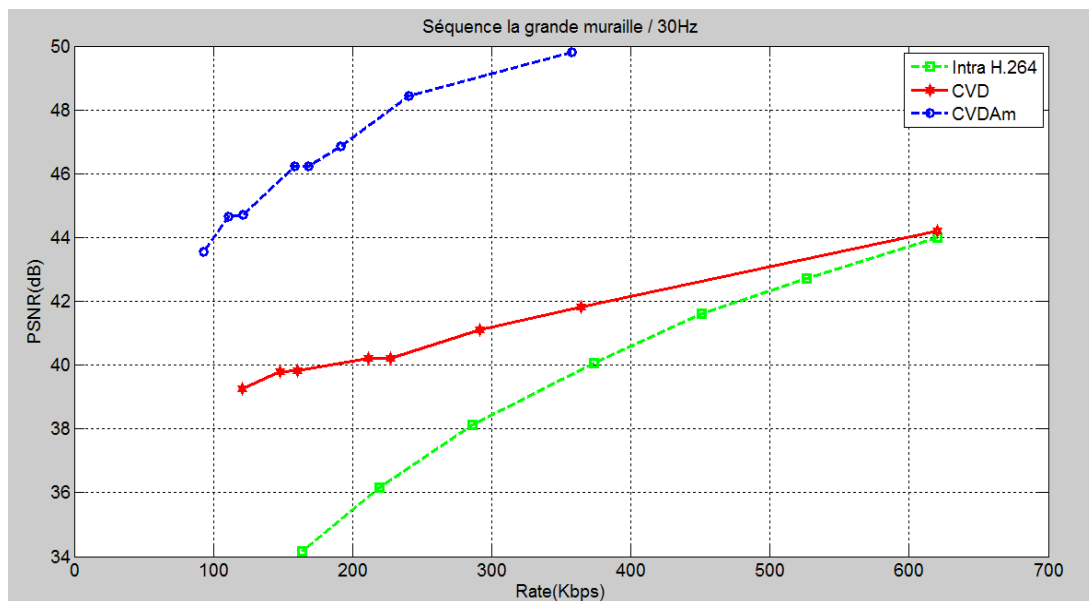


Figure 3.4. Performance débit-distorsion de la séquence de la grande muraille.

Pour cette séquence de test, on compare deux model CVD avec le codec traditionnel H.264. Pour le premier CVD (CVD), on utilise comme information adjacente la séquence modifiée (objets + bruit), et pour le deuxième CVDAm (CVD Amélioré) on enlève le bruit pour améliorer la qualité de l'information adjacente et ainsi améliorer les performances de notre codeur.

Comme on peut dans la figure 3.4, les résultats obtenus par le CVD surpassent ceux obtenus par le H.264 pour les 8 point débit-distorsion (faibles, moyennes et hautes fréquences).

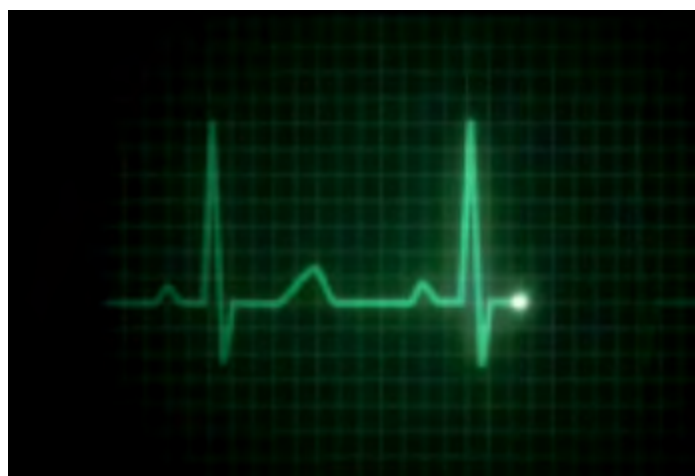
Dans les basses fréquences le débit obtenu par le CVD est meilleur d'environ 4 décibels que celui obtenu par le H.264, mais cet avantage dégrade de façon continue pour les moyennes et hautes fréquences à cause du bruit ajouté.

Tandis, que le CVSAm surpasse le CVD et le H.264 pour les 8 point du débit-distorsion. On voie très bien dans la figure 3.4 que le PSNR obtenu par CVDAm et meilleur d'environ 10 décibels pour les basses, moyennes et hautes fréquences à cause de la bonne qualité de l'information adjacente (manque du bruit).

### **3. Evaluation de la séquence de l'Electrocardiogramme (ECG) :**

#### **3.1. Caractéristique de la séquence de test de l'ECG :**

L'ECG est une représentation graphique de l'activité électrique du cœur, et comme pour un homme en bonne santé, le cœur effectue environ 80 battements par minute (en général), alors on peut considérer L'ECG comme une séquence périodique comme illustré par la figure 3.5.



**Figure 3.5. Illustration de La séquence de L'ECG.**



Le tableau 3.3 présente les caractéristiques principales de cette séquence.


Nom de la séquence vidéo	La grande muraille
	
Nombre total d'images	426
Nombre total des périodes	3
Nombre images évalués	284
Nombre des périodes évalués	2
Résolution spatiale	QCIF
Résolution temporelle (Image/Seconde)	30

Tableau 3.2 les caractéristiques de la séquence de L'ECG.

### 3.2. Evaluation des résultats:

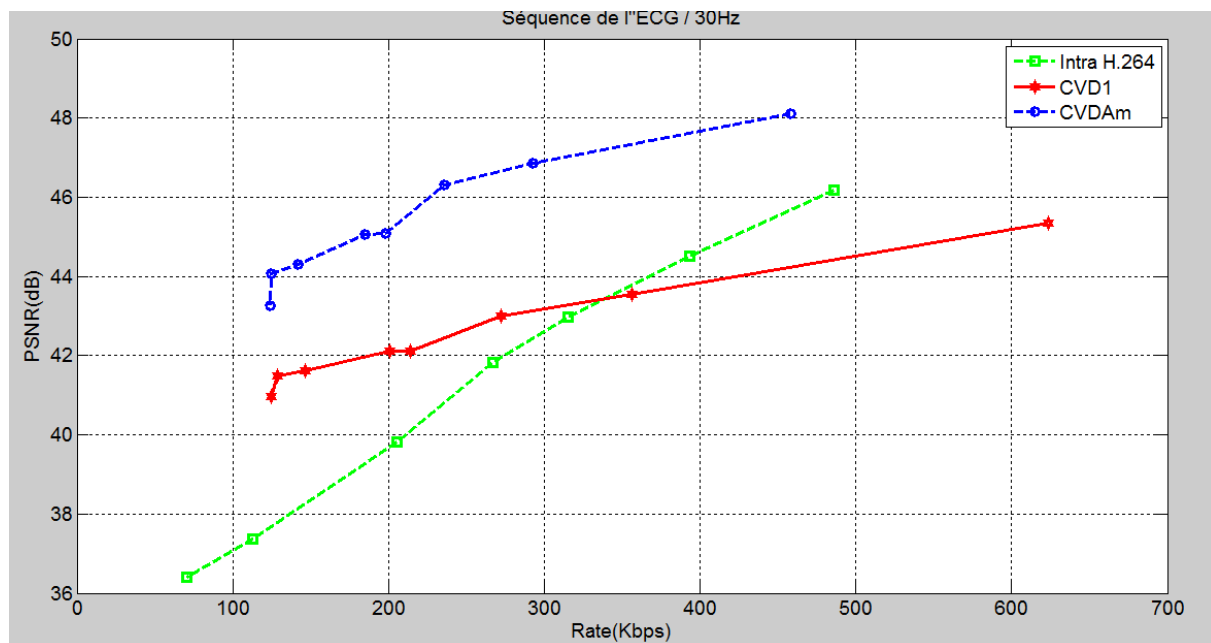


Figure 3.6. Performance débit-distorsion de la séquence de L'ECG.

Pour cette séquence de test on compare deux modèles CVD, le premier CVD1 où la période utilisée comme information adjacente est codé en mode intra H.264 et ensuite stockée dans le décodeur, et le deuxième CVDAm où n'est pas codée en mode intra (originale).

Le CVD1 surpasse le H.264 pour les basses et les moyennes fréquences, le PSNR obtenu par le CVD1 est meilleur d'environ 3 décibels que celui obtenu par le H.264 dans le cas des basses fréquences, et meilleur d'environ 1 jusqu'à 2 décibels dans le cas des moyennes fréquences. Tandis que dans le cas des hautes fréquences le H.264 surpasse le CVD1, le PSNR obtenu par le H.264 est meilleur d'environ 1.8 décibels que celui obtenu par le CVD1, cette dégradation de PSNR dans les hautes fréquences est due à cause de la rapidité et l'imprévisibilité du changement de l'activité électrique du cœur.

Pour le CVDAm (CVD Amélioré) qui est caractérisé par une meilleure qualité de l'information adjacente (une période originale) c'est-à-dire que la séquence WZ et l'information adjacente sont très corrélés. On voit que le CVDAm surpasse le CVD1 et le H.264 pour les basses, moyenne et hautes fréquences, avec un PSNR meilleur d'environ 2 jusqu'à 4 décibels de celui obtenu par le CVD1.

#### **4. Conclusion :**

Dans ce chapitre nous avons discuté les résultats obtenus par notre nouvelle approche CVD, qui repose sur une nouvelle technique de génération de l'information adjacente, adaptée spécifiquement pour des séquences vidéo périodique. Comme on a vu cette technique est très simple mais reste très efficace et donne de très bons résultats.

## **Conclusion générale :**

Le CVD et un nouveau modèle de codage vidéo à faible complexité, basée principalement sur les théorèmes de Slepian-Wolf et Wyner-Ziv, où la complexité de calcul est basculé ver le décodeur. La compression est achevé en exploitant la corrélation entre les images WZ et l'information adjacente.

Dans ce mémoire nous avons présente une nouvelle approche CVD basée sur une nouvelle technique de génération de l'information adjacente. Cette approche a été développée précisément pour l'appliquer sur des séquences vidéo périodiques.

Dans un premier temps, on a commencé par donner des généralités sur les éléments de bases de la théorie de l'information et le codage de source distribué. Ensuite nous avons présentés les techniques de génération de l'information adjacente utilisées dans l'état de l'art avec les architectures CVD les plus connus.

Le deuxième chapitre décrit en détail notre technique proposée pour créer l'information adjacente, avec l'implémentation de notre model CVD.

Dans le chapitre 3, les performances de notre codeur CVD sont évaluées, en les comparants avec le codeur traditionnel H.264. Cette approche donne de très bon résultats pour tous les séquences de test utilisés.

## Bibliographie:

- [1] D. SLEPIAN et J. WOLF, « Noiseless Coding of Correlated Information Sources » *IEEE Transactions on Information Theory* vol. 19, juillet 1973.
- [2] A. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, Jan 1976.
- [3] R. Gallager. Low density parity check codes. MA : MIT Press, 0(0), 1963. Cambridge.
- [4] T. Cover. A proof of the data compression theorem of Slepian and Wolf for ergodic sources (Corresp.). *IEEE Transactions on Information Theory*, 21(2):226 – 228, Mar 1975.
- [5] B. GIROD, A. M. AARON, S. RANE et D. REBOLLO-MONEDERO, « Distributed Video Coding », *Proceedings of the IEEE*, vol. 93, janvier 2005.
- [6] Fernando Pereira, Catarina Brites, and João Ascenso. Distributed video coding: Basics, codecs, and performance. In *Distributed Source Coding*, pages 189 – 245 Academic Press, Boston, 2009.
- [7] A. Wyner, “Recent Results in the Shannon Theory”, *IEEE Transactions on Information Theory*, Vol. 20, No. 1, pp. 2-10, January 1974.
- [8] D. SALOMON, *Data Compression. The Complete Reference*. Springer, 2004.
- [9] Y. Q. SHI et H. SUN, *Image and Video Compression for Multimedia Engineering. Fundamentals, Algorithms, and Standards*. CRC Press, 2000.
- [10] A. K. SALKINTZIS et N. PASSAS *Emerging Wireless Multimedia. Services and Technologies*. A Wiley-Interscience publication, 2006.
- [11] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560 – 576, July 2003.
- [12] ISO/IEC 14496-10 and ITU-T Rec. H.264, Advanced Video Coding 2003.
- [13] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret. The DISCOVER codec: Architecture, Techniques and Evaluation. In *Picture Coding Symposium (PCS'07)*, 2007.
- [14] J. Ascenso et al. The VISNET II DVC CODEC: Architecture, tools and performance. In *Proc. of the 18th European Signal Processing Conference (EUSIPCO)*, 2010.
- [15] R. Martins, C. Brites, J. Ascenso, and F. Pereira. Adaptive deblocking filter for transform.
- [16] J. Ascenso, C. Brites, F. Pereira, Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding, in: *Proceedings of EURASIP*

Conference on Speech and Image Processing, Multimedia Communications and Services (EC- SIPMCS), Smolenice, Slovak Republic, June 2005.

[17] J. Ascenso, C. Brites, F. Pereira, Content adaptive Wyner–Ziv video coding driven by motion activity, in Proceedings of IEEE International Conference on Image Processing (ICIP), Atlanta, GA, USA, October 2006.

[18] L. Alparone, M. Barni, F. Bartolini, V. Cappellini (1996) “Adaptively Weighted Vector-Median Filters for Motion Fields Smoothing”, *IEEE ICASSP*, Georgia, USA, May 1996.

[19] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra (2003) “Overview of the H.264/AVC Video Coding Standard”, *IEEE transactions on circuits and systems for video technology*, vol. 13, no. 7, July 2003.

[20] G. Petrazzuoli, M. Cagnazzo, B. Pesquet-Popescu, High order motion interpolation for side information improvement in DVC, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Dallas, TX, USA, March 2010.

[21] L. Natário, C. Brites, J. Ascenso, F. Pereira, Extrapolating side information for low-delay pixel-domain distributed video coding, in: Proceedings of International Workshop on Very Low Bitrate Video Coding (VLBV), Sardinia, Italy, September 2005.

[22] A. Aaron, R. Zhang, and B. Girod. Wyner-ziv coding of motion video. In Proc. Asilomar Conference on Signals and Systems, pages 240–244, 2002.

[23] A. Aaron, S. D. Rane, E. Setton, and B. Girod. Transform-domain Wyner-Ziv codec for video. In S. Panchanathan & B. Vasudev, editor, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, volume 5308 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, pages 520–528, January 2004.

[24] R. Puri and K. Ramchandran. Prism: an uplink-friendly multimedia coding paradigm. In IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings (ICASSP '03), volume 4, pages 856–9, April 2003.

[25] R. Puri, A. Majumdar, and K. Ramchandran. Prism: A video coding paradigm with motion estimation at the decoder. *IEEE Transactions on Image Processing*, 16(10):2436–2448, October 2007.

[26] S. Lin and D. J. Costello. Error Control Coding. Prentice Hall, 2004.

[27] G. Castagnoli, J. Ganz, and P. Graber. Optimum cycle redundancy-check codes with 16-bit redundancy. *Communications, IEEE Transactions on*, 38(1):111–114, Jan 1990.

- [28] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, April 1967.
- [29] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret. The DISCOVER codec: Architecture, Techniques and Evaluation. In *Picture Coding Symposium (PCS'07)*, 2007.
- [30] J. Ascenso, C. Brites, and F. Pereira. Content adaptive wyner-ziv video coding driven by motion activity. In *IEEE International Conference on Image Processing*, pages 605–608, Oct. 2006.
- [31] ISO/IEC International Standard 14496-10:2003, “Information Technology – Coding of Audio-visual Objects – Part 10: Advanced Video Coding”.
- [32] D. Varodayan, A. Aaron and B. Girod (2006) “Rate-Adaptive Codes for Distributed Source Coding”, *EURASIP Signal Processing Journal*, Special Section on Distributed Source Coding, vol. 86, no. 11, November 2006.