

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



Université des Frères Mentouri Constantine

Faculté des Sciences de la technologie

Département d'électronique

Laboratoire de traitement de signal

N° d'Ordre :

Série :



Mémoire

Présenté en vue de l'obtention du diplôme de

Magister en Electronique

Option : Traitement du Signal

Développement d'un Système de Codage Distribué pour la Surveillance

Présenter par

Boudjefdjouf Nadir

Devant la commission d'Examen:

Président : Toufik Laroussi

Prof Univ. F.M. Constantine.

Rapporteur: Said Benierbah

MCA Univ. F. M. Constantine.

Examineur: Zohir Hamoudi

Prof Univ. F. M. Constantine.

Examineur: Mohamed Chemachema

MCA Univ. F. M. Constantine

Promotion 2015\2016

Résumé :

Les nouvelles applications de la vidéo, telles que la vidéoconférence sur des réseaux sans fil, la vidéosurveillance, les caméras PC et les téléphones mobiles, souffrent de la limitation des ressources de calcul et des ressources énergétiques. Dans ce cadre, le codeur classique par prédiction n'est pas efficace en raison de sa complexité de codage. Ainsi, il est plus intéressant de faire appel à des codeurs vidéo de faible complexité. Afin de répondre à ces exigences, un nouveau modèle de codage vidéo a été développé sous le nom codage vidéo distribué (CVD). Ce dernier se distingue par une faible complexité au niveau du codeur et les opérations complexes sont transférées au décodeur.

Notre contribution consiste à implémenter un système de surveillance pratique utilisant le codage vidéo distribué et fonctionnant sur le réseau WIFI. Nous avons implémenté tout le système de codage et de décodage, ainsi que le protocole de communication entre le codeur et le décodeur, pour gérer l'opération d'échange d'informations. Nous proposons aussi des techniques pour accélérer le fonctionnement du système, notamment pour la création de l'information adjacente et la gestion du débit.

Abstract:

The new video applications, such as video conference over wireless networks, video surveillance, PC cameras and mobile phones suffer from computation and energy resources limitation. In this context, the conventional predictive coding is not effective because its encoder is very complex. For this, it is more interesting to use low-complexity video encoders. To meet these requirements, the distributed video coding was developed having a low complexity at the encoder.

Our contribution is the implementation of practical surveillance system, using distributed video coding and running on the wireless (WIFI) network. We have implemented all the operations of encoding, decoding and communication protocol between the encoder and decoder to manage the information's exchange. We have also proposed some techniques to speed up the system, particularly for the creation of the side information and rate management.

ملخص

التطبيقات الجديدة للفيديو، مثل فيديو المؤتمرات عبر الشبكات اللاسلكية، المراقبة بالفيديو، كاميرات الكمبيوتر و الهواتف النقالة، تعاني من محدودية موارد الحساب و الطاقة. في هذا السياق، الترميز التقليدي غير فعال لان عملية الترميز معقدة، لذا من الضروري استعمال نظام يكون ترميزه سهلا. لكي نصل إلى هذا المبتغى، هناك نظام جديد اسمه الترميز الموزع للفيديو. هذا الأخير مخصص لأنظمة الفيديو التي يكون نظام ترميزها سهلا. مساهمتنا تتمثل في تحقيق نظام فيديو للمراقبة عملي اعتمادا على نظام ترميز موزع للفيديو. يشتغل عبر شبكة اتصال لاسلكية. قمنا بإنجاز نظام يسمح بترميز و تفكيك الفيديو، و بروتوكول للاتصال بين المرسل و المستقبل لنتمكن من التحكم في عملية تبادل المعلومات بينهما، كما اقترحنا تقنيات تسمح لنا بتسريع اشتغال هذا النظام، هاته التقنيات تتمثل في الحصول على المعلومات الجانبية بطريقة سريعة و التحكم في كمية المعلومات المتبادلة بين المرسل و المستقبل.

Remerciement

Je tiens à remercier vivement mon Directeur de Mémoire, Mr Saïd BENIERBAH, Maitre de Conférences A, d'avoir dirigé avec compétence ce travail, ainsi que pour toutes les facilités matérielles qu'il m'a accordées pour l'accomplissement de ce travail.

Je remercie le Professeur Toufik Laroussi d'avoir accepté de présider le Jury, et également le Professeur Zohir Hamoudi et le Dr Mohamed Chemachema (MCA) qui ont accepté d'être membres.

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

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

Liste des acronymes :

CVD	Codage Vidéo Distribué
LDPC	Low Density Parity Check
CNM	Correlation Noise Model
DCT	Discrete cosine Transform
SAD	Sum Absolute Difference
FPGA	Field Programmable Gate Array
VLSI	Very Large Scale Integration
BP	Belief Propagation
GOP	Group Of Picture
SI	Side Information
CIF	Common Intermediate Format
MSB	Most Significant Bit
LSB	Low Significant Bit
MAD	Mean Absolute Difference
SPMD	Simple Program Multiple Data

Table des matières

Introduction générale	1
Chapitre 1 : Etat de l'art du codage vidéo distribué	3
1 Introduction	3
2 Théorie du codage vidéo distribué	3
2.1 Théorie de l'information	3
2.2 Théorie de codage source distribué	4
2.2.1 Théorème de Slepian-Wolf (codage des sources corrélés sans perte)	4
2.2.1.1 Codage de source sans perte avec information adjacente	5
2.2.2 Théorème de Wyner et Ziv (Codage de sources corrélées avec pertes)	6
2.3 Codage vidéo distribué	7
2.4 Les architectures de codage vidéo distribué	7
2.4.1 Architecture de codage vidéo distribué Stanford	7
2.4.2 Architecture de codage vidéo distribué de Berkeley	9
2.5 Applications des vidéos pour le codage vidéo distribué	10
2.5.1 Système de surveillance sans fil	10
2.5.2 Réseau des capteurs vidéo multi-vu	10
2.5.3 Vidéo mobile sans fil	11
2.5.4 Distribution du flux de vidéo	12
2.6 Evolution du codage vidéo distribué	12
2.7 Implémentation réelle du codage vidéo distribué	15
3 Conclusion	18

Chapitre 2 : Implémentation d'un CVD pour Vidéo surveillance	20
1 Introduction	20
2 Les défis à l'implémentation du CVD fonctionnant en temps réel	20
2.1 Accélération l'opération d'estimation de mouvement	20
2.2 La complexité du décodeur LDPC	21
2.3 Retour de canal à partir du décodeur vers le codeur	21
2.4 Le nombre de GOP optimal pour la génération de l'information adjacente	21
3 Système proposé	21
3-1 Opérations au niveau du codeur	22
3.1.1 Acquisition de l'image	22
3.1.2 Conversion de l'image en plans de bits	23
3.1.3 Codage de paquets par le codeur LDPC	23
4 Opérations au niveau du décodeur	24
4.1 Estimation de mouvement et génération de l'information adjacente	25
4.1.1 Estimation de mouvement vers l'avant	25
4.1.2 Estimation de mouvement vers l'arrière	25
4.1.3 Compensation de mouvement bidirectionnelle	26
4.2 Modélisation du canal de transmission (virtuel)	26
4.3 Estimation du débit minimal initial pour décoder les paquets de l'image WZ	26
4.3.1 Méthode de la valeur médiane pour calculer le débit minimal	27
4.3.2 Estimation débit minimal basé sur la corrélation temporelle	27
4.3.3 Estimation de débit minimal basé sur la corrélation de plan de bit	27

4.3.4 Réduire le nombre de retour de canal	28
4.4 Décodage de l'image WZ par propagation de croyances LDPC	29
4.5 Protocole de communication entre le décodeur et codeur WZ	29
4.6 Reconstruction de la séquence de vidéo	31
5 Conclusion	31
Chapitre 3 : Résultats et discussion	
1 Introduction	32
2 Présentation des séquences de test	32
3 Evaluation et résultats	34
3.1 Comparer la différence entre le débit initial et final pour décoder la trame WZ ...	34
3.2 Comparaison entre le nombre de retour de canal et le nombre de bits utilisés pour décoder l'image WZ	38
3.3 Les résultats de la performance Débit-PSNR	40
3.4 Réduire le nombre de retours de canal	42
4 Les Résultat pour notre système de surveillance en temps réel	43
5 Conclusion	46
Conclusion générale	47
Suggestion de travaux futurs	48

Liste des Figures :

Figure 1.1 Codage distribué de deux source X et Y statistiquement corrélées	5
Figure 1.2 Région réalisable de débit pour le codage de Slepian-Wolf	6
Figure 1.3 Codage de source WZ avec information adjacente	6
Figure 1.4 Architecture de codage vidéo WZ Stanford (avec DCT)	8
Figure 1.5 Architecture codeur vidéo WZ Berkeley :(a) codeur et (b) décodeur	9
Figure 1.6 Capteurs de vidéo par multi caméras	11
Figure 1.7 Architecture de transcodage pour la vidéo sans fil	12
Figure 1.8 Décodage LDPC par algorithme propagation de croyance	17
Figure 1.9 Architecture Parallèle de décodage CVD GOP	18
Figure 2.1 Notre système de surveillance basé sur le CVD	22
Figure 2.2 Conversion de l'image en plans de bits	23
Figure 2.3 Les nouvelles positions des syndromes envoyés à partir de codeur vers le décodeur	24
Figure 2.4 Sélection des positions des bits de syndrome du codeur selon la valeur de R_{\min}	24
Figure 2.5 Algorithme pour réduire le nombre de retour de canal	28
Figure 2.6 Réorganisation des bits de syndrome selon l'ordre de codeur	29
Figure 2.7 Protocole de communication entre codeur et décodeur	30
Figure 3.1 Séquence de surveillance dans une rue (Street)	33
Figure 3.2 Séquence de surveillance dans une station de métro (Subway)	33

Figure 3.3 Séquence de surveillance dans un hall(Hall Monitor)	33
Figure 3.4 Les débits finaux et initiaux estimé par corrélation temporelle. (a) pour la séquence Street, (b) pour la séquence Subway et (c) pour la séquence Hall Monitor	34
Figure 3.5 Les débits finaux et initiaux estimés par corrélation entre les plans de bit. (a) pour la séquence Street, (b) pour la séquence Subway et (c) pour la séquence Hall Monitor	36
Figure 3.6 Les débits finaux et initiaux estimé par valeur médiane. (a) pour la séquence Street, (b) pour la séquence Subway et (c) pour la séquence Hall Monitor	37
Figure 3.7 Nombre le retour de canal pour décoder l'image WZ. (a) pour Street , (b) pour subway et (c) pour Hall Monitor	39
Figure 3.8 Nombre de bit utilisé pour décoder l'image WZ. (a) pour Street, (b) pour Subway et (c) pour Hall Monito	40
Figure 3.9 PSNR de l'image en fonction de débit. (a) pour Street, (b) pour Subway et (c) pour Hall monitor	41
Figure 3.10 Nombre de retour de canal.(a) pour Street, (b) pour Subway et (c) pour Hall monitor	42
Figure 3.11 Etablissement de connexion entre le codeur et décodeur	44
Figure 3.12 Opération codage et décodage de l'image intra entre codeur (a) et décodeur (b)	44
Figure 3.13 Résultat d'une opération de codage réelle. (a) Image intra au niveau de codeur et (b) Image intra au niveau de décodeur	44
Figure 3.14 Protocole de communication entre le codeur (a) et décodeur (b)	45
Figure 3.15 Résultat d'une opération de codage réelle. Image WZ au niveau de codeur (a) et décodeur (b)	45
Figure 3.16 Influence de bruit sur la reconstruction de l'image au niveau de décodeur	46

Introduction générale

Avec l'émergence des nouvelles applications de communications visuelles, tels que la vidéo sans fil, les caméras de surveillance, les réseaux de capteurs de faible puissance, les appareils de communication mobile et beaucoup d'autres, le problème de réaliser des opérations de communication simple est devenu très important. En fait, ces derniers systèmes possèdent une puissance de calcul limitée et des sources d'énergie très faibles, donc il est nécessaire de trouver un modèle de compression, qui utilise un codeur simple et de faible complexité. Ce modèle est connu sous le nom de codage vidéo distribué (CVD).

Le principe de base du CVD, est de basculer la complexité de compression du codeur vers le décodeur. Ce modèle est basé sur le théorème de Slepian & Wolf (SW) [1] d'une part, et Wyner & Ziv (WZ) [2] d'autre part. Ces deux théorèmes ont prouvé que sous certaines conditions l'encodage de deux sources corrélées peut se faire conjointement ou indépendamment sans qu'il n'y ait de perte d'efficacité de transmission à partir du moment où le décodage lui est effectué conjointement.

Le grand défi pour implémenter ces nouvelles applications de vidéo, basé sur le CVD, en temps réel, est de réduire la complexité ou d'accélérer les opérations complexes de compression au niveau du décodeur. Alors dans ce mémoire, nous allons proposer des solutions qui vont permettre d'accélérer les opérations complexes au niveau du décodeur comme l'accélération de la création de l'information adjacente, réduction du nombre de retour canal et de gérer les opérations d'échange des données entre le codeur et le décodeur.

Notre contribution

L'objectif principal de ce mémoire est de :

- Proposer une opération de création d'information adjacente rapide au niveau du décodeur CVD.
- Accélérer le décodage par l'amélioration du contrôle du débit.
- Implémenter un système CVD pour la surveillance.

Organisation du mémoire

Notre mémoire décrit en détail la solution pour implémenter un système de surveillance basé sur le CVD.

Dans le chapitre 1, nous avons abordé les notions de base qui permettent de comprendre le CVD, comme la théorie de l'information, les architectures CVD existantes. Ensuite l'évolution et les applications réelles du CVD.

Dans le chapitre 2, nous proposons une application de surveillance basée sur le CVD dans le domaine pixel. Cette application est composée de deux parties. Une partie codeur, qui a pour rôle d'acquérir la vidéo et la coder par le codeur LDPC (Low Density Parity Check), ensuite de l'envoyer par réseau WIFI vers le décodeur. Une partie décodeur, dont le rôle consiste à recevoir les paquets de la vidéo, les décoder avec un décodeur LDPC à l'aide de l'information adjacente et la modélisation du canal de bruit, et enfin stocker et afficher la vidéo décodée.

Dans le chapitre 3, nous présentons les résultats et les performances de système de codage vidéo distribué pour la surveillance, ainsi réalisé.

Chapitre 1

Etat de l'art du codage vidéo distribué

1 Introduction :

Plusieurs techniques de codage vidéo ont été proposées pour exploiter la corrélation qui existe dans une séquence vidéo. Leur but est de coder la séquence vidéo avec le minimum de complexité et un débit réduit. Le codeur utilisant la prédiction par compensation du mouvement (inter) est le plus utilisable dans le codage vidéo. Ce type de codeur est utilisé pour les applications vidéo qui nécessitent d'être codées une seule fois et être décodées plusieurs fois, parce que leur codage est complexe et leur décodage simple.

Cependant, dans des applications telles que la surveillance sans fil (qui utilise des dispositifs de faible puissance), les caméras de téléphone et PC sans fils, le codeur inter n'est pas convenable parce que ces applications nécessitent un codeur simple (caméra de téléphone). Pour satisfaire aux besoins de ces nouvelles applications, il faut avoir un codeur simple même si le décodeur est complexe.

Différents résultats de la théorie de l'information, particulièrement le codage source distribué, proposent de résoudre ce problème par exploitation des statistiques de source au niveau du décodeur. Ces résultats sont exploités pour proposer une nouvelle architecture pour le codage vidéo. Cette architecture c'est le codage vidéo distribué.

2 Théorie du codage vidéo distribué

2.1 Théorie de l'information

La théorie de l'information décrit les aspects les plus fondamentaux des systèmes de communication. Cette théorie s'intéresse à la construction et à l'étude de modèles mathématiques à l'aide essentiellement de la théorie des probabilités. Elle permet d'établir des lois, qui assurent l'envoi des données d'une façon efficace, fiable et avec le moindre coût possible.

Ainsi la quantité moyenne d'information d'une source X est mesurée par l'entropie $H(X)$ qui est le nombre moyen minimal de symboles binaires par lettre nécessaires pour représenter la source. Soit X une variable aléatoire qui prend des valeurs dans une source définie par l'ensemble $\{x_1, x_2, \dots, x_n\}$ avec les probabilités correspondantes $\{p_1, p_2, \dots, p_n\}$ ou $P(X = x_i) = p_i$. L'entropie de X est définie par:

$$H(X) = \sum_{i=1}^n -P(x_i) \log_2 P(x_i) \quad (1.1)$$

L'entropie conjointe $H(X, Y)$ mesure la quantité d'informations contenue dans un système de deux variables aléatoires $X(x_1, x_2, \dots, x_n)$ et $Y(y_1, y_2, \dots, y_n)$. Elle est définie par:

$$H(X, Y) = - \sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log_2 p(x_i, y_j) \quad (1.2)$$

D'une manière analogue, on peut définir l'entropie conditionnelle $H(X/Y)$ comme l'incertitude qui reste sur X lorsque Y est connue :

$$H(X/Y) = - \sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log_2 p(x_i/y_j) \quad (1.3)$$

La relation entre l'entropie conjointe et conditionnelle est donnée par l'équation suivante :

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

2.2 Théorie de codage source distribué

Le codage de source distribué consiste à coder indépendamment deux ou plusieurs sources de séquences aléatoires dépendantes entre elles. Les codeurs envoient des trains de bits séparés à un décodeur unique qui peut fonctionner conjointement sur tous les trains de bits entrants et ainsi exploiter les dépendances entre les sources.

2.2.1 Théorème de Slepian-Wolf (Codage de sources corrélées sans pertes)

Comme le montre la Figure-1.1-, X et Y sont deux sources corrélées, le codage de ces sources est effectué d'une manière indépendante au niveau du codeur, mais le décodage se fait de façon conjointe. Slepian et Wolf [1,2], ont démontré qu'en exploitant la corrélation entre les deux sources X et Y au niveau du décodeur, on peut atteindre un résultat similaire à celui que l'on trouve lors de leur traitement conjoint au niveau de codeur. Avec la seule connaissance de la corrélation entre les sources à l'émission, on peut réaliser un taux de compression optimal sans avoir à transmettre deux fois l'information redondante se trouvant dans la première source X comme dans la deuxième Y .

Les conditions nécessaires pour que le décodeur réussisse à décoder l'information reçue avec une probabilité d'erreur proche de zéro sont :

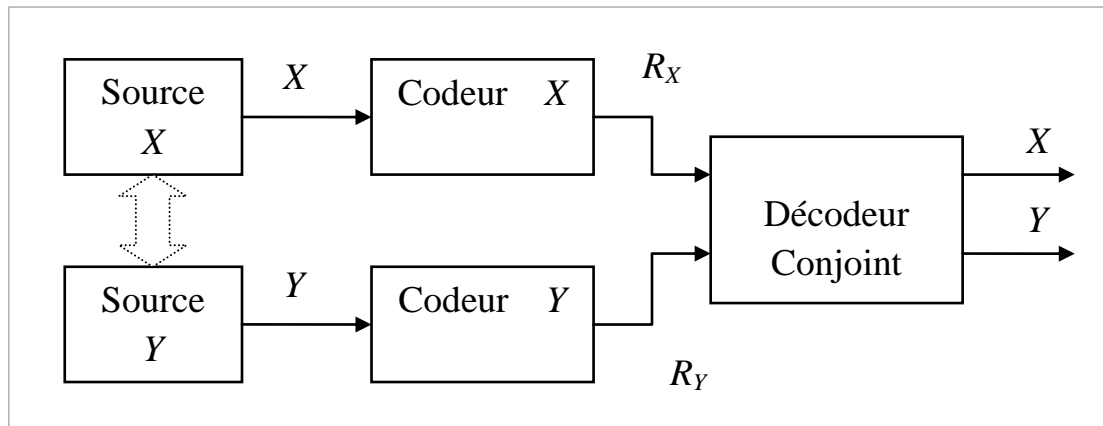


Figure 1.1 Codage distribué de deux source X et Y statistiquement corrélées

$$R_X \geq H(X/Y) \quad (1.5)$$

$$R_Y \geq H(Y/X) \quad (1.6)$$

$$R_X + R_Y \geq H(X, Y) \quad (1.7)$$

Ou R_X et R_Y , représentent respectivement le débit nécessaire pour assurer la reconstruction de l'information X et Y au niveau du décodeur.

2.2.1.1 Codage de source sans pertes avec information adjacente

Reconsidérons le schéma de compression de SW qui est indiqué par la Figure -1.2-. L'information Y est envoyée en premier et reconstruite au niveau du décodeur. Étant donné que les informations X et Y sont corrélées, la reconstruction de Y fournit une certaine information à propos de X. Dans ce scénario, Y peut-être dénotée comme information adjacente (SI : Side Information), cette information est disponible au niveau du décodeur pour la reconstruction de X.

Pour assurer une reconstitution de X avec une probabilité d'erreur proche de zéro, il faut assurer un débit de l'information X sachant qu' Y est au niveau de décodeur :

$$H(X/Y) \leq R_X \leq H(X) \quad (1.8)$$

$$H(Y/X) \leq R_Y \leq H(Y) \quad (1.9)$$

Cette équation (1.8) est vérifiée si le décodeur dispose de l'information adjacente Y. Sans cette information adjacente le débit devait être $R_X > H(X)$. Ce scénario est connu sous le nom de « compression sans perte » de X avec une information adjacente Y disponible uniquement au niveau du décodeur.

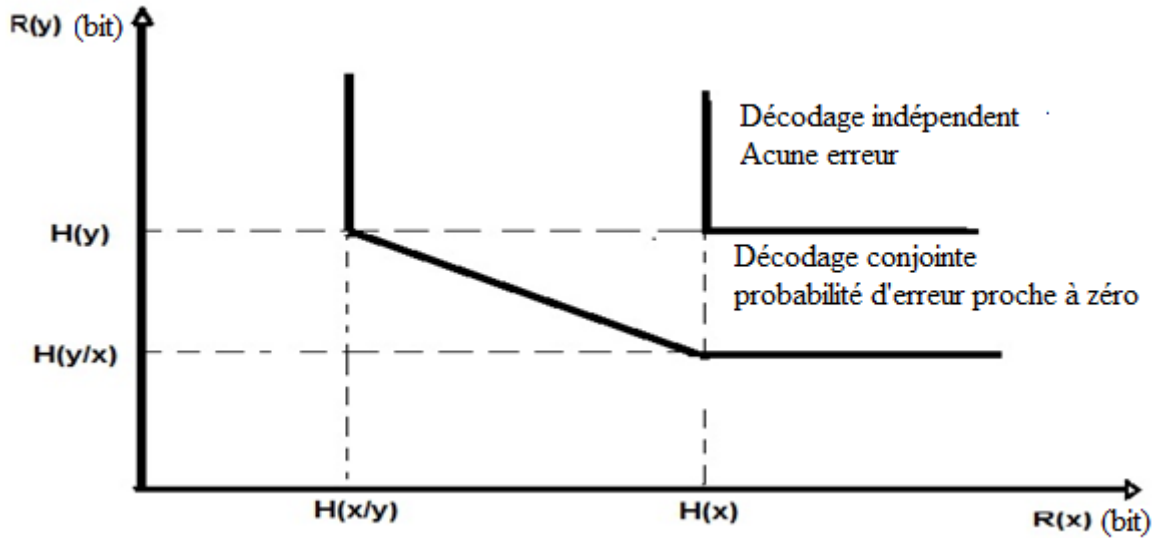


Figure 1.2 Réglion réalisable de débit pour le codage de Slepian-Wolf

2.2.2 Théorème de Wyner et Ziv (Codage de sources corrélées avec pertes)

La plupart des systèmes de compression tolèrent un certain niveau de distorsion entre l'information originale et l'information décodée. Selon la Figure -1.3-, Wyner et Ziv proposent un système de compression avec perte [3] [4], en définissant la fonction débit-distorsion ($R_{X|Y}(D)$) qui consiste à donner le débit minimum $R_{X|Y}$ requis pour assurer une reconstruction ne dépassant pas une certaine moyenne de distorsion D .

Wyner et Ziv ont démontré que le débit de l'information nécessaire pour obtenir X en présence de l'information adjacente Y au niveau du décodeur est :

$$R_{X|Y}^{WZ}(D) \geq R_{X|Y}(D) \quad (1.9)$$

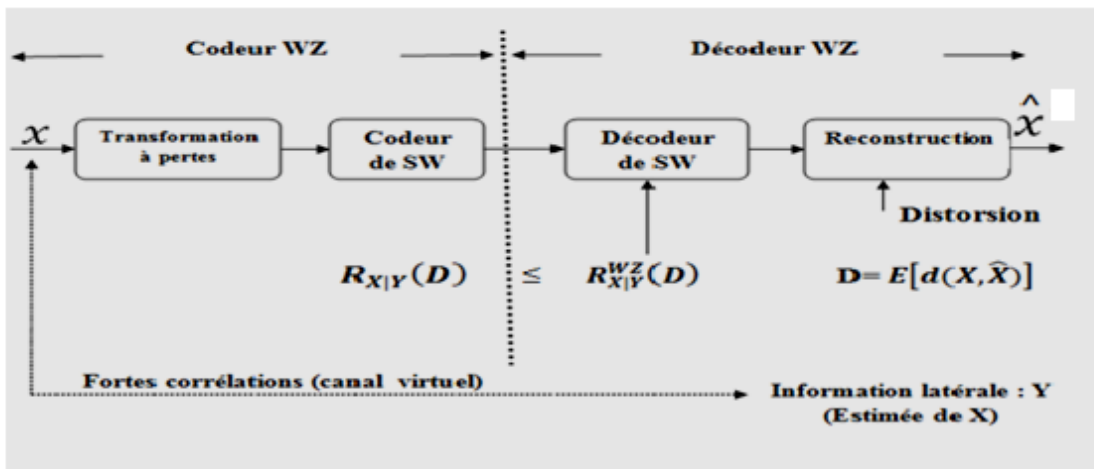


Figure 1.3 Codage de source WZ avec information adjacente

2.3 Codage vidéo distribué

La compression vidéo repose sur deux principes de base, le premier exploite les redondances spatiales qui existent dans chaque image, le second repose sur l'idée qu'une image peut-être très similaire aux images voisines (redondance temporelle). Le codage vidéo distribué est basé sur ce dernier principe qui est une opération coûteuse en termes de calcul car elle utilise des techniques très complexes comme l'estimation du mouvement.

Le schéma de base d'un CVD consiste en un codeur simple et un décodeur qui peut être complexe. Donc l'estimation du mouvement est transférée au décodeur.

2.4 Les architectures de codage vidéo distribué

2.4.1 Architecture de codage vidéo distribué de Stanford

Cette architecture de codage vidéo a été proposée en 2002 [5] pour le domaine des pixels et plus tard étendu au domaine de transformation (en ajoutant la transformation DCT). Ce codeur WZ est représenté par la Figure 1-4 et fonctionne comme suit :

La séquence vidéo est divisée en images WZ et des images clés, les images paires sont les images clés et les images impaires sont les images WZ. Les images clés sont codées en mode intra par le codeur H.264, les images WZ sont codées par le codeur WZ. Pour les images WZ, les pixels seront quantifiés et converti en plans de bits. Les plans de bits seront ensuite codés par un turbo-codeur pour obtenir les syndromes ; ces syndromes sont ensuite envoyés sur la demande du décodeur. Dans le cas de l'utilisation de la DCT, une transformation DCT est appliquée pour obtenir les symboles, ensuite ces symboles sont regroupés pour former les bandes de coefficients DCT avant leur conversion en plans de bits.

Le décodeur crée les informations adjacentes pour chaque trame codée WZ par l'opération d'interpolation ou d'extrapolation. Une fois que les informations adjacentes et les statistiques sur les résidus pour chaque pixel ou chaque bande de coefficients DCT sont connues, le turbo décodeur reçoit un bloc de bits de parité à partir du codeur et lance l'opération du décodage.

Pour décider si plus de bits sont nécessaires pour que le décodage réussisse, le décodeur utilise un critère de demande d'arrêt. Après avoir réussi à décoder le plan de bits MSB des pixels ou d'une bande DCT, le turbo-décodeur procède de façon similaire avec les plans de bits restants associés à la même bande.

Une fois que tous les plans de bits des pixels ou des bandes DCT (bande par bande) sont décodés avec succès, les plans de bits sont regroupés pour former les pixels ou les coefficients décodés quantifiés pour reconstruire les pixels ou les coefficients DCT.

Pour le cas de la DCT, toutes les bandes ne sont pas transmises et les coefficients des bandes DCT pour lesquelles aucun bit n'est transmis, sont remplacés par les coefficients des bandes DCT correspondant de l'information adjacente. Après que toutes les bandes de DCT sont reconstitués, la transformation DCT inverse (IDCT) est appliquée pour obtenir l'image WZ décodée.

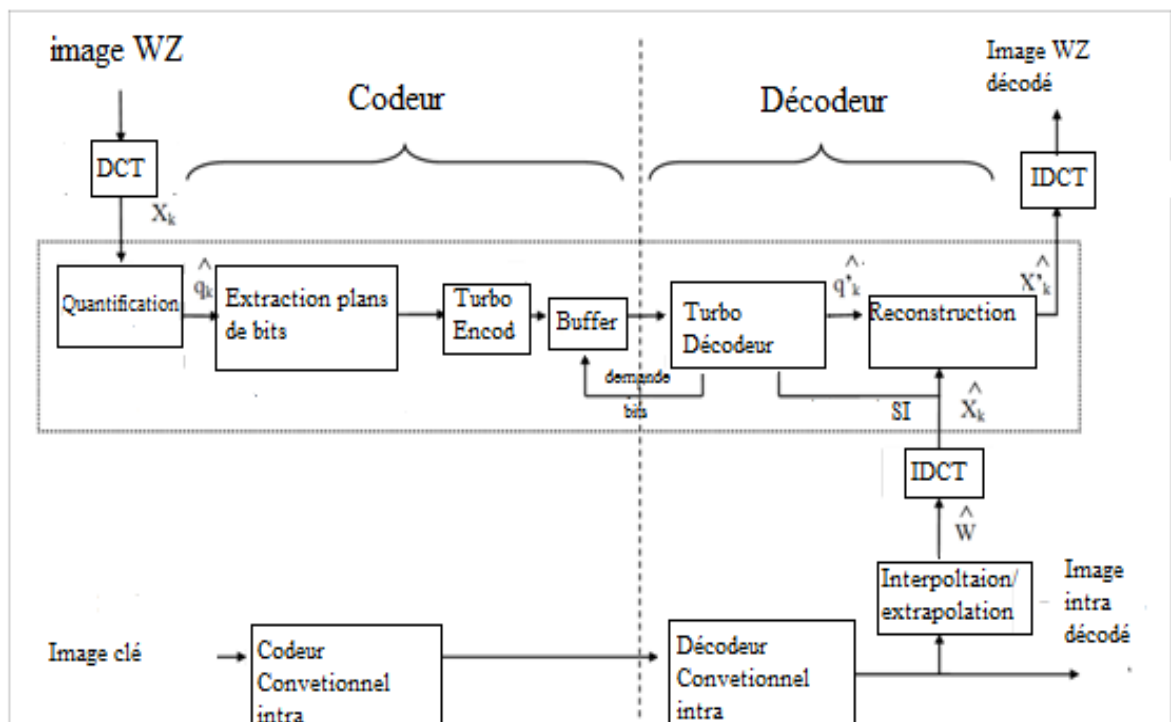


Figure 1.4 Architecture de codage vidéo WZ Stanford (avec DCT)

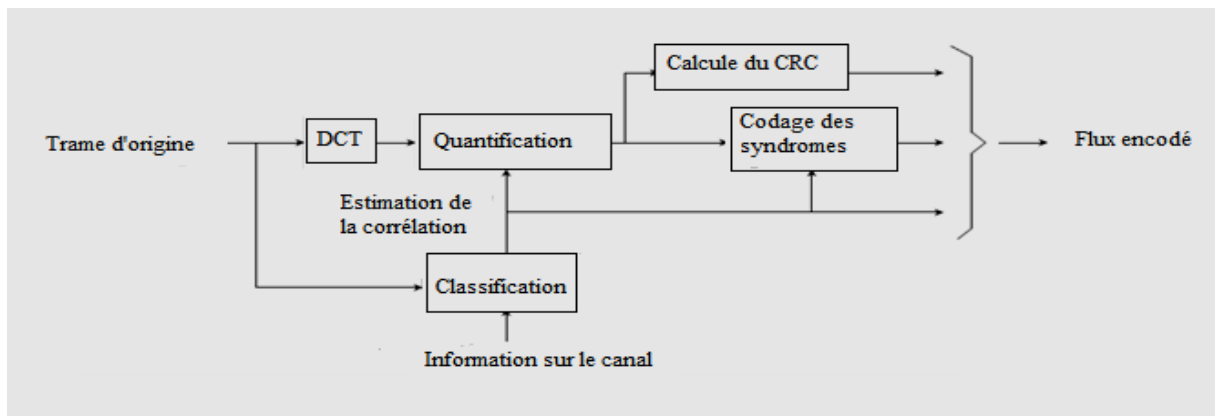
Il existe un autre système de codage vidéo de Wyner-Ziv populaire dans la littérature, qui a été développé dans le projet de DISCOVER [6]. Il est basé sur l'architecture de Stanford dans le domaine transformé avec les petits changements suivants :

- 1/ la transformation DCT est effectuée sur des blocs de (4 x 4).
- 2/ La taille de GOP (Groupe of Pictures) est flexible (fixe ou variable).
- 3/ Le codage des coefficients DCT est effectuée par le codeur LDPC ou turbo code.

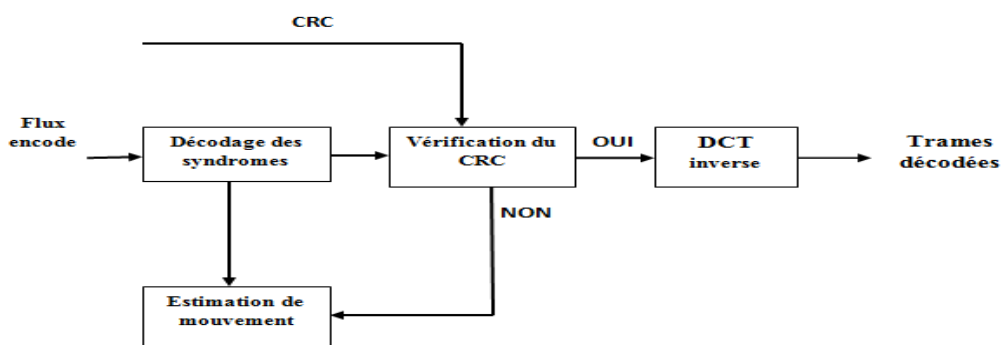
2.4.2 Architecture de codage vidéo distribué de Berkeley

Une autre approche du codage vidéo WZ a été proposée par l'université de Berkeley [7] [8], connue dans la littérature comme PRISM (Power-efficient, Robust, high compression Syndrome Based Multimedia Coding) est illustrée dans la Figure 1.5. Cette approche fonctionne comme suit :

Chaque image de la séquence vidéo est divisée en blocs de 8×8 , ensuite une transformation DCT est appliquée sur chaque bloc, enfin une quantification scalaire est appliquée aux coefficients DCT. Avant que l'opération de codage ne commence, chaque bloc est classifié dans une des catégories prédéfinies en fonction de la corrélation entre le bloc courant et le bloc de la trame précédente. Dans l'étape de classification, on décide du mode de codage pour chaque bloc de l'image courante : soit aucun codage (skip), soit codage intra ou codage WZ. Pour les blocs qui appartiennent à la classe de décodage WZ, seuls les bits les moins significatifs sont codés, car on suppose que les bits les plus significatifs peuvent être déduits à partir de l'information adjacente.



(a)



(b)

Figure 1.5 Architecture de codeur vidéo WZ Berkeley : (a) codeur et (b) décodeur

Le décodeur génère les blocs candidats qui jouent le rôle de l'information adjacente, chaque bloc candidat conduit à un bloc décodé, à partir duquel une signature de CRC est générée. Afin de détecter si le décodage a réussi, cette dernière est comparée à la valeur de CRC reçue en provenance du codeur, si les deux CRC sont égaux, le bloc candidat est considéré décodé, sinon le décodeur revient à l'étape de génération de SI et un nouveau candidat de bloc est utilisé. Une fois que tous les blocs quantifiés sont décodés, une transformation DCT inverse est appliquée sur les coefficients quantifiés pour reconstruire l'image WZ.

2.5 Applications des vidéos pour le codage vidéo distribué

Ce qui a donné de l'importance au codage vidéo distribué est le fait qu'il est convenable pour les applications vidéo, ayant une faible puissance de calcul et une énergie limitée, qui ont besoin de faire appel aux codeurs simples. C'est le cas des nouvelles applications vidéo telles que les caméras de surveillance et la téléphonie mobile. Dans ce qui suit, on va voir des exemples et des scénarios d'application [9].

2.5.1 Système de surveillance sans fil

L'exploitation d'un réseau de caméras sans fil à faible puissance se fait principalement pour la surveillance. Le réseau de surveillance peut comprendre de nombreuses caméras qui détectent la même scène avec des positions de visualisation différentes et communiquent l'information à un serveur de surveillance central. Les caméras voisines peuvent généralement capter partiellement des zones interférées et par conséquent leurs séquences vidéo associées sont corrélées. Chaque caméra code sa propre vidéo et l'envoie vers le serveur central où le décodage sera effectué. Le nombre des caméras (codeur) est généralement beaucoup plus élevé que le nombre de décodeurs (typiquement un), donc il est possible de réduire le coût du système en utilisant des codeurs de faible complexité et seulement un décodeur de complexité plus élevée.

2.5.2 Réseau de capteurs vidéo multi-vue

Les progrès dans la conception des capteurs et réseaux de communications permettent un large éventail d'applications comme le suivi d'objet, le suivi de l'activité et des événements. Une scène ou un objet est acquis par de nombreuses caméras situées à des positions spatiales fixes, par exemple pour les effets spéciaux de cinéma et acquisition multi-vue (Figure -1.6-), Il y aura des encodeurs associés à chaque caméra et le décodage sera effectué à la station de base. Il est hautement souhaitable d'utiliser un codeur à faible

complexité, faible coût et faible consommation d'énergie, et un décodeur plus complexe. Dans ce type de réseaux, le décodeur est une unité centrale de traitement avec des capacités de calculs élevés chargée de recueillir et de traiter toutes les informations reçues.



Figure 1.6 capteurs de vidéo par multi caméras

2.5.3 Vidéo mobile sans fil

La vidéo mobile sans fil est une communication qui se fait à travers des appareils photos de téléphones (vidéoconférence), les participants à la conférence utilisent des dispositifs mobiles pour coder et décoder la vidéo. L'exigence principale dans cette application est d'avoir une faible complexité au niveau du codeur et du décodeur dans chaque dispositif mobile, puisque la consommation d'énergie et l'autonomie de la batterie sont étroitement liées à la complexité de l'encodeur / décodeur. Afin de contourner ces contraintes, le décodeur classique (H.264) et le codeur CVD sont nécessaires dans chaque dispositif de mobile, parce que le décodage du codeur classique est simple et son encodage est complexe, alors que pour le CVD c'est le cas contraire.

Le flux des données doit être transformé d'un codage CVD en un codage classique (H.264 / AVC) à l'intérieur des serveurs du réseau, ce qui permet un codage /décodage de faible complexité au niveau des mobiles. Les opérations de codage et de décodage de vidéo entre deux mobiles sont faites comme suit :

Mobile 1 : codage vidéo avec le codeur CVD et transmission à la station de base. La station de base décode la vidéo par un CVD ensuite l'encode par un codeur (H.264) et l'envoie vers Mobile 2 qui décode la vidéo reçue par un décodeur H.264.

Mobile 2 : codage vidéo avec le codeur CVD et transmission à la station de base. La station de base décode la vidéo par un CVD ensuite l'encode par un codeur (H.264) et l'envoie vers Mobile 1 qui décode la vidéo reçue par un décodeur H.264.

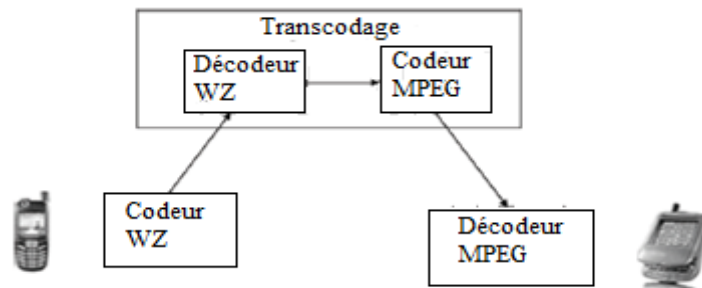


Figure 1.7 Architecture de transcoding pour la vidéo sans fil.

2.5.4 Distribution du flux de vidéo

La croissance énorme de l'internet nous a donné la possibilité de réaliser des systèmes de flux vidéo qui permettent à un utilisateur de recevoir et visualiser simultanément une séquence vidéo à partir d'un serveur distant. Il est possible d'effectuer un « flux distribué (Distributed streaming) » afin de donner au récepteur le débit maximum possible de données. Le flux vidéo est transmis au récepteur par différents émetteurs dans un mode distribué, afin de réduire le débit binaire du côté de l'émetteur et d'assurer le débit nécessaire au niveau du récepteur. Dans ce contexte, il est possible de considérer CVD comme une nouvelle technologie qui peut être utilisée pour effectuer un système de streaming vidéo plus flexible et fiable.

2.6 Evolution du codage vidéo distribué

2.6.1 Information adjacente

L'efficacité du codage vidéo distribué dépend principalement de l'amélioration de la qualité de l'information adjacente. Pour cette raison des efforts ont été faits pour améliorer la génération de cette dernière.

En 2002, l'équipe de Stanford propose d'utiliser la moyenne des trames clés pour générer la SI [10]. Cette méthode est simple mais n'exploite pas bien la corrélation entre les trames de référence. Pour mieux exploiter cette corrélation, en 2005, Ascenso et al. [11] ont proposé d'utiliser une interpolation par compensation du mouvement entre les images intra .

Ainsi, pour générer la SI d'une image WZ, une estimation de mouvement bidirectionnelle est réalisée entre les deux trames de référence intra précédente et suivante. Une fois que le vecteur de mouvement bidirectionnel est obtenu, une compensation de mouvement bidirectionnelle est effectuée pour générer la trame interpolée. Dans la même année, X. Artigas et L. Torres [12] ont proposé un algorithme pour améliorer la qualité de la trame décodée progressivement. Chaque fois qu'un plan de bits est décodé, avec succès, il sera utilisé pour améliorer la SI, et cette dernière va nous aider à décoder les plans de bits suivants. De la même façon, Ascenso et. al. [13] ont proposé de générer les SI en tenant compte des bits de l'image partiellement décodée, ce qui améliore la qualité de SI.

Pour les applications CVD en temps réel, il est plus approprié de générer SI par extrapolation, car l'interpolation de l'image va interrompre l'ordre original des trames et nécessite que certaines trames futures soit disponibles. Pour cela, Natario et. al. [14] proposent une estimation de mouvement par extrapolation : le vecteur de mouvement est estimé en utilisant deux trames de référence précédemment décodées.

[15] et [16] proposent d'évaluer des SI différentes à chaque itération du processus de propagation de croyance (Belief Propagation) du décodage LDPC et de choisir la meilleure de ces différentes SI ; celle qui arrête le décodage itératif avec succès.

Dans le cas où il ya beaucoup de variation dans la scène de la vidéo, la variation du vecteur de mouvement entre les blocs d'image n'est pas toujours linéaire. Ce qui a amené Akinola et. al. [17] à estimer un mouvement non-linéaire entre les images intra pour améliorer la SI, afin de permettre la reconstruction parfaite de l'image WZ. Pour réduire la complexité, [18], [19] et [20] propose de limiter la recherche de mouvement au niveau des blocs par estimation de la probabilité de déplacement de chaque bloc. Les résultats de probabilités pour chacun des vecteurs de mouvement contribuent à l'amélioration de la SI. A chaque fois qu'un groupe de blocs a été décodé avec succès, les vecteurs de mouvement devront être recalculés en mettant à jour le modèle de bruit de corrélation et la SI.

Laraba.M [21] propose une nouvelle méthode pour générer l'information adjacente pour les séquences de vidéo périodiques comme le battement de cœur, mouvement d'une caméra de surveillance, et rotation de satellite autour de la terre. L'idée consiste à diviser la séquence vidéo en plusieurs périodes, ensuite une seule période d'images est codée et envoyée vers le décodeur, cette période peut être utilisée comme une information adjacente pour décoder la séquence vidéo prochaine.

Benierbah. S et Khamadja. M [22] ont proposé une approche hybride entre les blocs intra et les blocks WZ dans l'image pour créer l'information adjacente. Chaque image est

divisée en des blocs 16×16 , ces blocs sont divisés en des blocs de 4×4 , ensuite les premiers blocs de 4×4 de tous les blocs de 16×16 seront codés par le codeur intra pour générer l'information adjacente, qui va contribuer à décoder le reste des blocs de l'image (les trois derniers blocs de 4×4 pour chaque bloc de 16×16) par le décodeur WZ, au début, le décodeur génère l'information adjacente initiale à partir des blocs intra déjà décodés pour estimer les blocs WZ, ensuite le décodeur utilise les blocs WZ et les blocs intra pour améliorer progressivement l'information adjacente, cette dernière est utilisée avec les bits de syndromes qui sont envoyés par le codeur pour obtenir l'image final décodée.

Haj taib [23] propose une architecture progressive de la SI avec deux groupes de blocs. Au niveau du codeur, chaque trame WZ codée est divisée en deux ensembles : le premier ensemble des blocs est envoyé vers le décodeur, qui génère SI en utilisant la méthode d'interpolation [24]. Ce premier ensemble décodé est utilisé pour améliorer la qualité de SI pour le deuxième ensemble.

2.6.2 Modélisation de bruit du canal

L'un des aspects les plus importants qui influe sur le rendement du codeur CVD est le modèle de bruit du canal qui est utilisé pour estimer la distribution de bruit entre les SIs et la trame WZ.

C. Brites et. al. [25] ont proposé un algorithme qui permet d'estimer en ligne le bruit au niveau de la trame WZ pour le domaine pixel du codeur CVD. Il utilise l'erreur entre les trames de référence intra précédente et suivante pour approximer la variance entre SI et trame WZ. La distribution du bruit est supposée être la distribution Laplacienne. Le paramètre de la fonction densité de probabilité est calculé à partir de la variance estimée. Pour améliorer le paramètre de la distribution Laplacienne, Luong et. Al [26] ont proposé un algorithme pour estimer de manière adaptative ce paramètre en utilisant la méthode de regroupement, pour exploiter la corrélation de toutes les bandes de fréquence. Il a également été proposé de combiner leurs algorithmes avec le modèle de bruit dans [27] pour optimiser de manière adaptative les informations adjacentes pour le décodage LDPC.

Dans la même année, Song et. al. [28] proposent une méthode de raffinement progressif pour modéliser le bruit de corrélation (CNM) qui utilise les plans de bits précédemment décodés et les erreurs de quantification pour affiner le bruit de corrélation estimé.

En 2012, Luong et.al.[29] ont proposé d'affiner la trame résiduelle en exploitant la corrélation des coefficients voisins. Les résidus de trame qui sont déjà décodés sont utilisés pour influencer la distribution du bruit de la trame courante et donc exploiter la corrélation

temporelle. L'optimisation adaptative des paramètres de bruit est obtenue par des essais multiples de convergence dans le processus de décodage LDPC.

2.6.3 Contrôle de débit

Durant l'opération de décodage des images WZ, le décodeur demande plusieurs syndromes à partir du codeur d'une manière itérative, cette demande influe largement sur le temps du décodage. Pour réduire le nombre d'itérations, plusieurs méthodes ont été proposées pour estimer le débit de syndrome initial, qui permet de décoder les images WZ rapidement avec un nombre de retour de canal minimal.

Brites et. al. [30] proposent une méthode pour estimer le débit initial pour décoder le plan de bit de l'image WZ. L'idée consiste à utiliser le débit moyen qui permet de décoder les plans de bit des images WZ précédentes pour décoder le plan de bit de l'image WZ courante.

Fu et. al. [31] ont proposé d'estimer le débit initial pour décoder le plan de bit de l'image WZ au niveau du codeur. Au début, il commence par estimer la SI par la méthode de l'interpolation bilinéaire ou par l'estimation de mouvement bidirectionnelle, ensuite il calcule la probabilité d'erreur (différence entre la SI et l'image originale WZ) en utilisant la SI et la modélisation de canal de bruit pour un nombre de plan de bit déterminé qu'on veut envoyer vers le décodeur, enfin il calcule le débit initial à l'aide du théorème de Slepian-Wolf [1,2].

2.6.4 Codage de canal

Les codes correcteurs d'erreurs de l'information pendant la transmission jouent un rôle très important dans le CVD. Les deux codes les plus utilisés, dans CVD, sont le turbo code et le LDPC qui ont un débit proche de la limite de Shannon (capacité). Après avoir utilisé le turbo-code pendant une longue période, de nos jours on utilise le codeur LDPC qui est mieux que le turbo-code pour les mots de code de longueur considérable, grâce à la variation du nombre d'itération de décodage, la possibilité de l'implémentation en parallèle, ainsi que la minimisation du temps de décodage et de l'erreur de décodage de la trame.

2.7 Implémentations réelles du codage vidéo distribué

L'implémentation en temps réel de CVD peut être faite par deux méthodes : software et hardware.

2.7.1 Aspect software

L'aspect software est basé sur l'amélioration des algorithmes de calcul dans les blocs de CVD (décodeur LDPC, estimation de mouvement, et DCT) de manière efficace.

2.7.1.1 Réduire la complexité du calcul de la DCT

L'implémentation réelle de la DCT demande des opérations de calcul très intensives. Par conséquent, beaucoup d'efforts ont été développés pour concevoir des architectures rentables capables de diminuer cette complexité de calcul. Par exemple, on peut citer le travail de Lee[32], qui est basé sur la décomposition de la matrice DCT en coefficients pairs et impairs et celui de Liu et. al. [33] qui proposent un algorithme qui calcule la DCT par récursivité.

2.7.1.2 Décodage LDPC parallèle

Dans le décodeur LDPC simple, la corrélation entre les plans de bit n'est pas entièrement exploitée lors du décodage. Pour cela, Pan et al [34], proposent un codeur Wyner-Ziv avec de multiples décodeurs LDPC, pour affiner de manière itérative le décodage de plan de bits au cours du processus de décodage et prendre l'inter-corrélation des plans de bits en compte. Le codeur Wyner-Ziv avec plusieurs décodeurs LDPC fonctionnent sur tous les plans de bits disponibles à la fois. On exploite la corrélation entre eux en échangeant les informations et en mettant à jour la distribution du modèle de bruit pendant le décodage. Si un plan de bit est décodé avec succès, le décodeur LDPC correspondant ne demande plus les bits de syndrome à partir du codeur, le reste des décodeurs LDPC qui n'ont pas encore décodé leurs plans de bits concernés sont réinitialisés à l'aide des nouvelles SI, qui ont été actualisées par les plans de bits décodés avec succès.

2.7.2 Aspect hardware

L'aspect hardware consiste à implémenter une solution hardware qui permet d'effectuer les mêmes opérations de l'aspect software, mais d'une façon efficace et rapide.

2.7.2.1 Implémentation du SAD (Sum Absolute Difference) avec FPGA

L'algorithme SAD est utilisé dans l'estimation de mouvement, pour calculer le vecteur de mouvement entre les blocs des images. Le coût de calcul de cet algorithme peut être un problème important dans les applications de codage vidéo distribué en temps réel. Afin de réduire la complexité des calculs, différentes architectures VLSI sont proposées [35] parmi lesquelles, on trouve les circuits logiques programmables FPGA [36] qui ont été choisis pour accélérer le calcul de la SAD.

2.7.2.2 Implémentation du décodeur LDPC avec VLSI (Very large scale integration)

Le défi principal dans le décodeur LDPC est la mise en œuvre d'une solution capable de gérer efficacement l'échange des messages lors du décodage itératif de propagation de

croissance(BP). Pour cela, Zhong et. al. [37] proposent des implémentations de décodeur LDPC complètement parallèle et partielle. Dans un décodeur LDPC entièrement parallèle, les nœuds de variable et de contrôle sont reliés entre eux par un réseau d'interconnexion, représentant la connectivité du graphe de Tanner. Il est clair que ces décodeurs peuvent atteindre de très haut débit de décodage. L'inconvénient de cette méthode est qu'elle souffre d'une grande complexité pour sa mise en œuvre, à cause de la complexité du schéma de routage. Par conséquent, le décodeur entièrement parallèle est efficace seulement pour les codes LDPC avec une longueur de code courte.

Pour le décodeur LDPC partiel, le but est d'établir un compromis entre la complexité du matériel et la vitesse de décodage, en utilisant un multiplexage temporel dans lequel un certain nombre de nœuds de variable ou nœuds de contrôle est relié à une unité de décodage unique.

2.7.2.3 Implémentation de décodeur LDPC utilisant la propagation de croyance parallèle par multi-CPU

Le but de l'utilisation de l'architecture parallèle multi-CPU [38] est d'accélérer l'exécution de l'algorithme de propagation de croyances du décodeur LDPC comme le montre par la Figure 1.8.

Au niveau des cœurs du processeur, l'approche « un seul program plusieurs données » (SPMD) est utilisée. Cette approche est utile dans les scénarios où le même bloc d'instructions est exécuté en plusieurs itérations sur des données différentes.

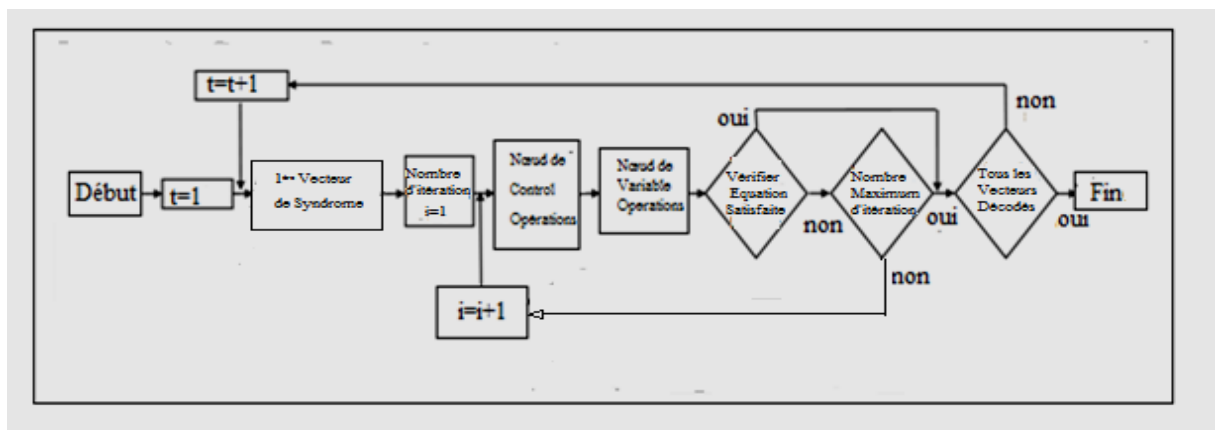


Figure 1.8 Décodage LDPC par algorithme propagation de croyance

Dans l'algorithme de décodage de propagation de croyance (BP), comme représenté sur la Figure-1.8, il existe des boucles à trois niveaux. La boucle de plus haut niveau est celle du niveau de blocs, où l'algorithme de décodage est répété pour chaque bloc de syndromes reçu.

Cette boucle peut être exécutée par l'approche SPMD en parallèle par l'attribution de blocs de syndrome aux différents cœurs, car il n'y a pas de dépendances entre les données des blocs.

La boucle de niveau intermédiaire est celle du niveau d'itération, où à chaque itération, les messages sont échangés entre les nœuds de variables et les nœuds de contrôle. Cette boucle ne peut être parallélisée par l'approche SPMD.

Le niveau plus interne consiste en deux boucles pour calculer les messages des nœuds de variables et les messages des nœuds de contrôle. Dans ces boucles, les mêmes calculs sont effectués pour chaque nœud de variable ou nœud de contrôle. Ces boucles peuvent être exécutées en parallèle en utilisant l'approche SPMD.

2.7.2.4) Décodage parallèle de vidéo distribué GOP utilisant des processeurs multi-cœur

La complexité du décodeur augmente lorsque plusieurs plans binaires sont décodés. A ce point, Garcia et. Al. [39] proposent une architecture de décodage de WZ qui distribue la complexité de décodage à travers plusieurs unités de traitement (Figure -1.9-). Cette architecture permet le décodage parallèle de la structure GOP. Chaque procédure de décodage GOP est réalisée indépendamment par un cœur (processeur) différent. En outre, pour chaque trame WZ dans un GOP, l'information adjacente est calculée et ensuite divisée en plusieurs parties. Ensuite, chaque partie de la trame est affectée à n'importe quel cœur qui exécute la procédure de décodage itératif pour décoder la partie correspondante de la trame WZ.

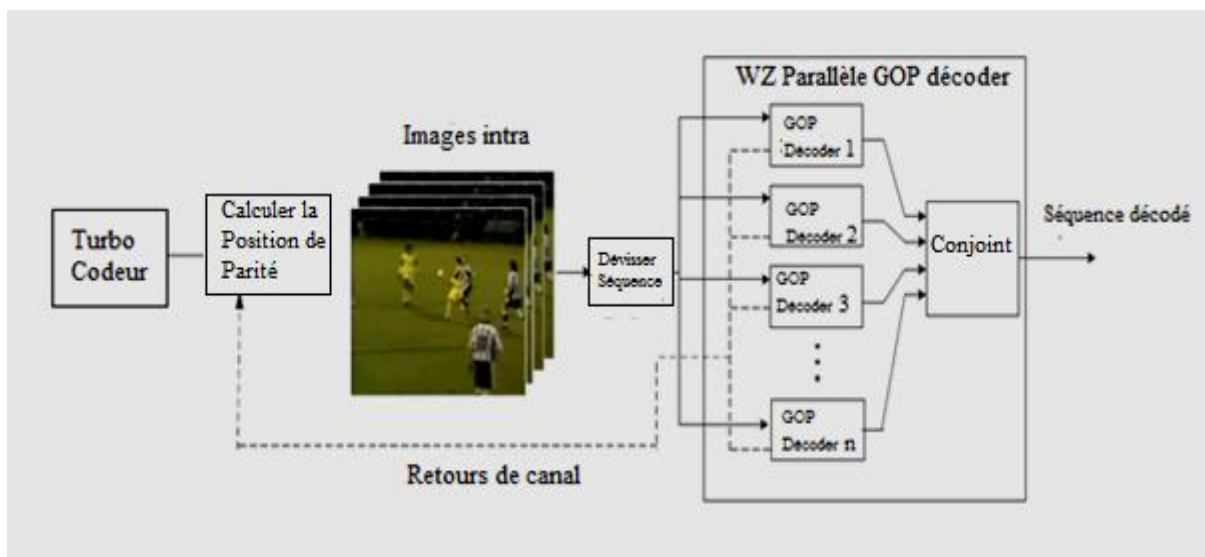


Figure 1.9 Architecture Parallèle de décodage CVD GOP

3) Conclusion

Avec l'émergence des nouvelles applications de vidéo, qui nécessitent un codeur simple dû à la consommation d'énergie, le codage classique comme MPEG.X et H.26X sont

inappropriés, car ces deux derniers utilisent des codeurs très complexes. Donc, il est nécessaire de trouver un nouveau modèle de codage de vidéo qui a un codeur simple, ce modèle est le codage vidéo distribué, ce dernier permet de basculer les opérations complexes du codeur vers le décodeur.

Chapitre 2

**Implémentation de notre système de surveillance basé sur
le codage vidéo distribué**

1 Introduction :

Notre travail est basé sur le développement d'un système de codage vidéo distribué, appliqué dans le domaine de la surveillance. Notre système est composé de deux parties dont les fonctions principales sont respectivement le codage et le décodage des données vidéo. Au niveau du codeur, les principales fonctions réalisées sont : la capture de la vidéo, l'encodage des données, et leur transmission vers le décodeur. Au niveau du décodeur, les fonctions exécutées sont : la réception des données, le décodage, l'affichage et le stockage vidéo.

Dans ce chapitre, nous allons développer, en détails, les différentes parties qui constituent le système que nous avons proposé. En particulier nous allons détailler le protocole de communication pour gérer l'opération d'échange des données entre le codeur et le décodeur, l'accélération de la génération de l'information adjacente et la réduction du nombre de retours de canal pour décoder les paquets de l'image WZ.

2 Les défis à l'implémentation du CVD fonctionnant en temps réel

Dans cette section, nous allons détailler les points essentiels pour améliorer l'efficacité du codage vidéo distribué en terme de temps de traitement. Pour pouvoir transmettre un nombre d'images par seconde suffisant, il faut que le temps d'exécution des différentes opérations effectuées soit court. En fait, ce sont essentiellement les opérations de décodage qui sont très complexes et leur fonctionnement affecte directement le nombre d'images à transmettre. Donc, c'est ces opérations qu'il faut accélérer pour un fonctionnement en temps réel du système.

2.1 Accélération de l'opération d'estimation de mouvement

Les techniques utilisées pour générer l'information adjacente au niveau du décodeur influent sur la performance du taux de distorsion dans le CVD. Plus la corrélation est forte entre l'information adjacente et l'image originale, moins de bits de parité seront envoyés vers le décodeur afin de décoder l'image WZ.

Dans les séquences vidéo contenant un mouvement faible, comme la surveillance, il est préférable de réduire le nombre de comparaison entre les blocs d'images afin de déterminer rapidement le vecteur de mouvement. Pour cela, on a défini un seuil max de différence entre les blocs d'images, pour activer l'estimation du mouvement. Pendant le calcul du vecteur de mouvement, on commence par calculer la différence entre les deux blocs des images clés de référence (ce qui est équivalent au vecteur de mouvement (0,0)). Si cette différence ne dépasse pas le seuil max, on ignore le vecteur de mouvement pour ce bloc et on passe au bloc suivant.

2.2 La complexité du décodeur LDPC

Le codage LDPC impose une certaine complexité au niveau du décodeur. Cette complexité est due à l'algorithme de décodage, qui est basé sur la propagation de croyance. Cette méthode utilise un certain nombre d'itérations pour échanger les données entre les nœuds de contrôle et les nœuds variables, pour obtenir la trame décodée. Ce nombre d'itérations prend un temps considérable, qui influe sur l'implémentation du CVD en temps réel. Pour résoudre ce problème, la solution consiste à implémenter le décodeur LDPC en parallèle, afin d'accélérer le temps d'exécution de l'algorithme de propagation de croyance.

2.3 retour de canal à partir du décodeur vers le codeur

Durant le décodage des plans de bits, le décodeur demande itérativement les bits de parité à partir du codeur par l'intermédiaire d'un canal de retour jusqu'à ce que le décodage réussisse. Par conséquent, le décodeur doit effectuer un contrôle de débit pour garantir un nombre minimum de retour de canal et un débit minimum de bits de parité, qui seront envoyés par la suite vers le codeur. Dans certains cas, le retard de décodage sera trop élevé, car plusieurs opérations de requêtes de bits additionnels peuvent être nécessaires pour décoder les données. La complexité de calcul au niveau du décodeur sera aussi importante, car après chaque nouvelle réception de bits, le décodage LDPC est effectué. En plus, même si le retard de décodage est acceptable, il pose un problème de stockage de flux des données codées jusqu'à ce que le décodage soit réussi. Pour surmonter ce problème, le décodeur doit recevoir le nombre requis de bits de parité ou syndromes en une seule itération. Ceci n'est possible que si le décodeur peut estimer le débit des données, qui permet de décoder la trame WZ, ou si l'estimation de la SI au niveau du décodeur est proche de l'image WZ originale.

2.4 Le nombre de GOP optimal pour la génération de l'information adjacente

Dans le cas où la variation de la scène à surveiller est faible, il est préférable d'utiliser un nombre élevé d'images par GOP afin de générer la SI correctement et réduire par conséquent le débit de transmission de trames, vers le décodeur. Mais le nombre élevé de trames par GOP demande une zone de mémoire considérable et un temps de retard important pour coder les images, ce qui s'oppose au principe de la simplicité du codeur. Pour notre système CVD pratique à faibles ressources, nous avons fixé la valeur maximale du GOP à 2.

3 Système proposé

Note travail consiste à implémenter un système de surveillance, dont le principe de fonctionnement est le suivant : on commence par l'acquisition des images d'une scène vidéo à partir d'une webcam, située au niveau du codeur (PC), ensuite le codeur code les images et les

envoi vers le décodeur (un autre PC) à travers un réseau sans fils (WIFI), pour les décoder. Le langage de programmation utilisé pour l'implémentation de notre système est le C++.

Le système que nous avons développé est illustré par la figure 2-1. Il se compose, comme toutes les architectures de codage vidéo distribué de deux parties : une partie qui concerne le codage de la vidéo et une partie qui concerne son décodage. Dans la suite de cette section, nous allons détailler les différentes parties de ce système, en insistant sur les parties qui sont particulières à notre contribution.

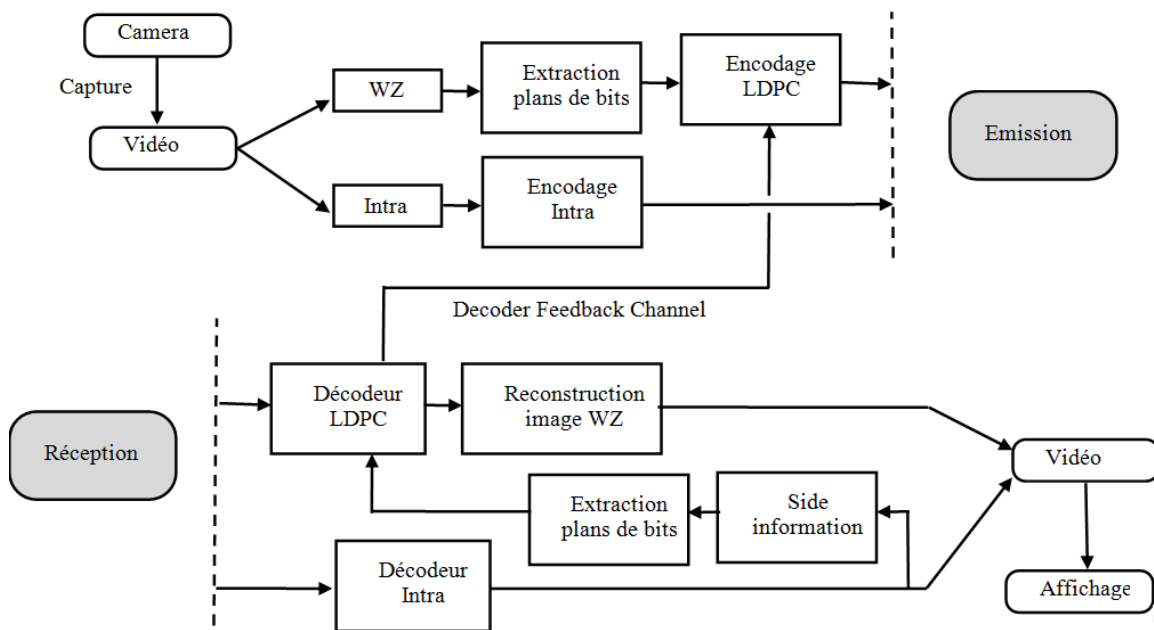


Figure-2.1- Notre système de surveillance basé sur le CVD

3-1 Opérations au niveau du codeur

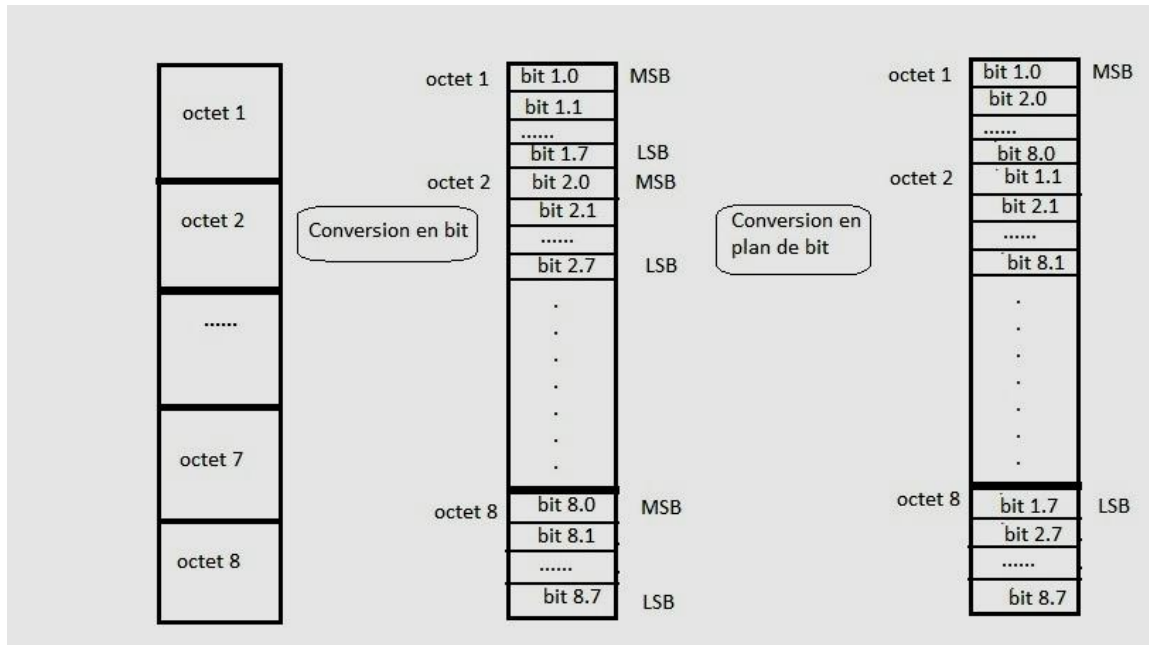
3.1.1 Acquisition de l'image

Le but de cette opération est de capturer les images CIF (Common Intermediate Format) de longueur 288 pixels et largeur 352 pixels, à partir d'une caméra ou d'une Webcam. Les images sont ensuite divisées en images intra et images WZ. Les images intra sont codées par le codeur intra H.264 et directement envoyées vers le décodeur. Par contre, le codage et décodage des images WZ se font dans le domaine pixels. Pour coder l'image WZ on passe par plusieurs étapes :

- Conversion de l'image en plans de bits.
- Codage de l'image par un codeur LDPC.
- Envoi des données résultantes vers le décodeur.

3.1.2 Conversion de l'image en plans de bits

La conversion de l'image en plans de bits est une opération nécessaire pour le codeur LDPC binaire. Les valeurs des pixels de l'image sont converties en bits, qui seront réorganisés en plans de bits (Figures -2.2) à partir des bits de poids le plus fort (MSB) suivis par les bits de poids inférieur jusqu'aux bits de poids le plus faible (LSB).



Figures 2.2 Conversion de l'image en plans de bits

3.1.3 Codage de paquets par le codeur LDPC

Les plans de bits seront divisés en plusieurs paquets de taille 6336 bits chacun. Ces paquets sont ainsi prêts à être codés par le codeur LDPC. Chaque paquet X du plan de bits est ensuite multiplié par la matrice de parité H , de taille 6336 pour obtenir les syndromes S .

$$S = H * X \quad (2.1)$$

Les syndromes LDPC sont ensuite accumulés [39]. Les bits de source (X_1, X_2, \dots, X_N) sont additionnés (modulo 2) en accord avec la structure du graphe de Tanner du code LDPC, afin de générer le syndrome $S = [S_1, S_2, \dots, S_M]^T$. Les bits de celui-ci sont ensuite accumulés (modulo 2) afin de produire le syndrome accumulé $a = [a_1, a_2, \dots, a_m]^T$.

Avant d'envoyer le syndrome accumulé au décodeur, nous sélectionnons le positionnement des bits de syndrome à envoyer, selon le débit imposé par le décodeur. Le positionnement des prochains bits demandés par le décodeur est déterminé par un tableau (Figure 2.3). Le but est de ne pas envoyer les bits syndromes qui ont été émis précédemment.

32	65	48	15	56	23	31
----	----	----	----	----	----	-------	----

Figure 2.3 Les nouvelles positions des syndromes envoyées à partir de codeur vers le décodeur

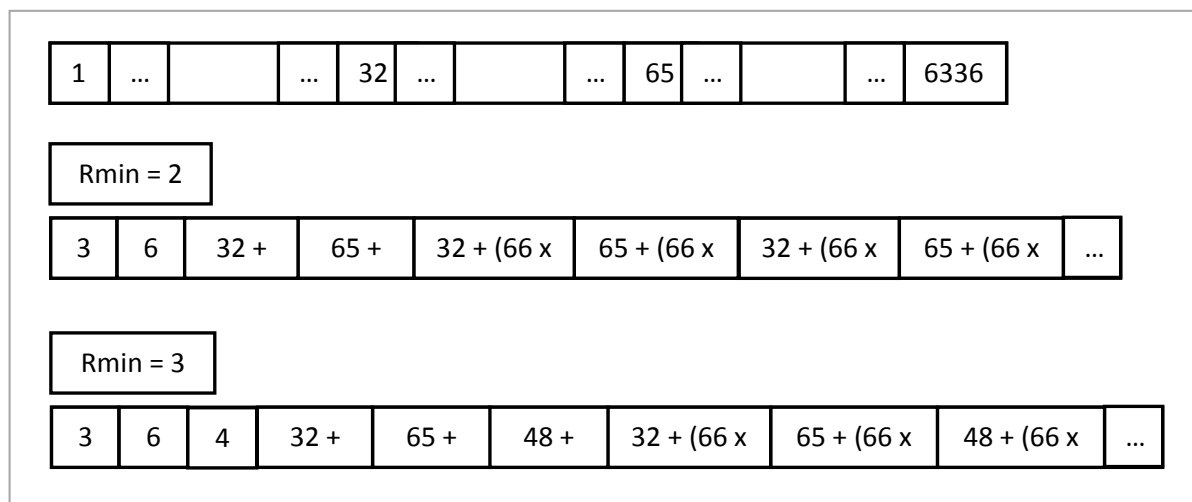


Figure 2.4 Sélection des positions des bits de syndrome du codeur selon la valeur de R_{min} .

Les bits de syndrome accumulés sont envoyés progressivement, suivant le débit minimum imposé par le décodeur (R_{min}), en sélectionnant les positions des bits à envoyer. Ces positions sont déterminées à partir d'un tableau (Figure 2.3) contenant les nouvelles positions des bits de syndrome à envoyer exigées par le décodeur, au cas où il ne parvient pas à accomplir le décodage et passe à une valeur de débit (R_{min}) plus élevée, et en relevant les valeurs de bits de syndrome à une période de 66 emplacements. L'intérêt est d'éviter d'envoyer les bits qui étaient déjà envoyés précédemment. Les bits de syndrome sont ensuite envoyés vers le décodeur.

4 Opérations au niveau du décodeur

Au niveau du décodeur, la première image intra reçue est décodée et affichée. Cette image et l'image intra suivante sont utilisées pour décoder les images WZ. Ensuite pour les images WZ, une estimation de mouvement est faite entre les deux images intra précédente ($i-1$) et suivante ($i+1$), pour générer l'information adjacente de l'image WZ courante (i). Après, l'information adjacente obtenue est transformée en plans de bits qui aident à décoder l'image WZ. Ensuite, une modélisation du canal de transmission par le modèle Laplacien est effectuée. Cette dernière permet de donner une information initiale sur les bits de l'image WZ

décodée. D'autre part, l'opération de décodage par le décodeur LDPC se fait par un protocole de communication entre le codeur et le décodeur. Après le décodage des trames WZ, les images sont affichées dans l'ordre chronologique et d'une façon alternée : une image WZ est affichée suivi par une image intra.

Le décodage de l'image WZ est détaillé par les étapes suivantes :

4.1 Estimation de mouvement et génération de l'information adjacente

Pour notre système (vidéo surveillance), on propose d'utiliser l'estimation de mouvement bidirectionnelle entre l'image intra précédente et suivante pour déterminer le vecteur de mouvement entre les blocs des deux images intra (vecteur mouvement avant et arrière) et prendre la moyenne des deux vecteurs. Ensuite, on utilise la compensation de mouvement pour estimer la SI de l'image originale WZ.

Dans les séquences vidéo de surveillance, le changement de mouvement entre les images vidéo est lent, à cause de ça on a fixé un seuil MAD_{min} , pendant le calcul de la différence moyenne absolue (MAD) entre les blocs de l'image pour déterminer le vecteur de mouvement. Si la MAD est inférieure à MAD_{min} , on considère que le vecteur de mouvement est nul et on passe à un autre bloc pour déterminer son vecteur de mouvement.

En plus, pour réduire la complexité de la recherche, on a limité la zone de recherche à ± 3 pixels verticalement et horizontalement.

4.1.1 Estimation de mouvement vers l'avant

Dans cette étape, nous allons déterminer les vecteurs de mouvement vers l'avant entre l'image intra précédente ($i-1$) et suivante ($i+1$). On découpe l'image intra $i+1$ sous forme de blocs de 16×16 . Puis dans chaque bloc on recherche le bloc de l'image $i-1$ qui lui ressemble le plus dans une zone de recherche de ± 3 pixels verticalement et horizontalement avec un pas de recherche de 1 pixel. La ressemblance (déterminer le vecteur de mouvement vers l'avant (d_{xf}, d_{yf})) des blocs est évaluée à l'aide la valeur minimale de MAD.

4.1.2 Estimation de mouvement vers l'arrière

Dans cette étape, nous allons déterminer les vecteurs de mouvement vers l'arrière entre l'image intra $i+1$ et $i-1$. Pour cela on découpe l'image intra $i-1$ sous forme de blocs de 16×16 , pour chaque bloc de cette dernière, on recherche le bloc dans l'image $i+1$ qui lui ressemble le plus avec une zone de recherche de ± 3 pixels verticalement et horizontalement et un pas de recherche de 1 pixel. La ressemblance (déterminer le vecteur de mouvement vers l'arrière (d_{xb}, d_{yb})) des blocs est évaluée à l'aide la valeur minimale de MAD.

4.1.3 Compensation de mouvement bidirectionnelle

L'image estimée SI est obtenue à partir de la moyenne des deux images intra avec leur vecteur de mouvement.

$$SI(x, y) = (intra_{i-1}(x + dx_b, y + dy_b) + intra_{i+1}(x + dx_f, y + dy_f))/2 \quad (2.2)$$

L'information adjacente obtenue est transformée en plans de bits qui seront utilisés par le décodeur LDPC pendant le décodage de l'image WZ.

4.2 Modélisation du canal de transmission virtuel

Pour améliorer l'efficacité du codage, au niveau du décodeur WZ, ce dernier a besoin d'informations fiables pour modéliser la distribution de bruit résiduel qui représente la différence entre l'image originale WZ au niveau du codeur et l'image WZ estimée au niveau du décodeur. La distribution de Laplace est habituellement utilisée pour modéliser ce bruit. La fonction de distribution Laplacienne pour le codage WZ est donnée par l'équation suivante:

$$f(C_x^{b_k}(u, v) - C_y^{b_k}(u, v)) = \frac{\alpha}{2} e^{-\alpha |C_x^{b_k}(u, v) - C_y^{b_k}(u, v)|} = \frac{\alpha}{2} e^{-\alpha |C_R^{b_k}(u, v)|} \quad (2.3)$$

Où f est la fonction de densité de probabilité (fdp). $C_x^{b_k}(u, v)$ Représente les pixels de l'image WZ estimée dans le plan de bit b_k à la position (u, v) .

$C_y^{b_k}(u, v)$ Représente les pixels de l'information adjacente y dans le plan de bit b_k à la position (u, v) .

Le paramètre α de la fdp Laplacienne est estimé pour chaque paquet par l'équation suivante :

$$\alpha^2 = \frac{2}{\sigma^2} \quad (2.4)$$

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N (X_{i+1}(n) - X_{i-1}(n))^2 \quad (2.5)$$

N représente le nombre d'octets du paquet à décoder, X_{i-1} et X_{i+1} représente respectivement la trame intra précédente et suivante.

4.3 Estimation du débit minimal initial pour décoder les paquets de l'image WZ

Afin de réduire le nombre de requêtes de bits de syndromes effectuées par le décodeur pour décoder l'image de WZ, on estime le nombre minimum de bits de syndromes ou le débit minimal (R_{min}) qui doit être envoyé par le codeur. Pour cela, on a exploité les trois algorithmes qui sont proposé par [40].

4.3.1 Méthode de la valeur médiane pour calculer le débit minimal

Pour réduire la complexité du calcul et estimer le débit minimal, [40] propose d'estimer initialement le débit minimal pour un paquet i ($i=0\dots15$) pour un plan de bit j ($j=0\dots7$) de l'image courante. Cette estimation est basée sur le débit final des bits syndrome des dernières trois images WZ décodées précédemment. On prend la valeur médiane des trois valeurs finales des trois images WZ et on les multiplie par le facteur $(1-k)$, sachant que $k=0.1$ pour les 5 premiers paquets et 0.05 pour les autres paquets.

$$INC(i, j) = (1 - k) * \text{median}(FNC_{-1}(i, j), FNC_{-2}(i, j), FNC_{-3}(i, j)) \quad (2.6)$$

INC est le débit initial des bits de syndrome utilisé pour décoder le paquet de l'image WZ.

FNC est le débit final des bits de syndrome qui permet décodé le paquet de l'image WZ.

Dans notre cas, dans le domaine pixels, on prend la valeur médiane des trois valeurs finales de débit du paquet du même plan de bits.

4.3.2 Estimation débit minimal basée sur la corrélation temporelle

Cet algorithme exploite la stationnarité temporelle du débit final qui permet de décoder les bits de syndrome de l'image courante.

L'opération consiste à estimer le débit minimal initial des syndromes pour décoder le paquet d'ordre j de plan de bit i de l'image courante. Cette opération est effectuée en se basant sur le débit final de décodage des trois paquets des images précédentes de plan de bit i et de paquet j , en plus du débit final du paquet j de plan de bit $i-1$ de l'image courante.

$$INC_{t=i+3}(bp = 1, paquet = 1) = aFNC_{t=i}(bp = 1, paquet = 1) + a^2FNC_{t=i+1}(bp = 1, paquet = 1) + a^3FNC_{t=i+2}(bp = 1, paquet = 1) + INC_{t=i+3}(bp = 0, paquet = 1) \quad (2.7)$$

$a=0.54$ si il y a une sous-estimation de débit minimal pour la trame syndrome ($i+2$) et $a=0.47$ si il y'a une surestimation.

4.3.3 Estimation du débit minimal basée sur la corrélation de plans de bits

Dans cette méthode, le calcul du débit minimal est basé sur la corrélation entre les plans de bit de l'image courante. Pour déterminer le débit initial pour un paquet d'ordre j qui est situé dans un plan de bit i pour l'image courante, on utilise le débit final qui a permis de décoder le paquet d'ordre j de plan de bit $i-1$ de l'image courante et le paquet d'ordre j de plan de bit i de l'image précédente.

$$INC_{image2}(bp = 1, paquet = 1) = FNC_{image2}(bp = 0, paquet = 1) + a [FNC_{image1}(bp = 1, paquet = 1) - FNC_{t=imag1}bp = 0, paquet = 1)] \quad (2.8)$$

$$\left\{ \begin{array}{l} a=0.5 \text{ Si } FNC_{image1}(bp=1,paquet=1)= INC_{image1}(bp=1,paquet=1) \text{ ou } \\ FNC_{image2}(bp=0,paquet=1)= INC_{image2}(bp=0,paquet=1) \\ a=1 \text{ Si non} \end{array} \right.$$

4.3.4 réduire le nombre de retour de canal

Notre idée consiste à envoyer un débit initial et un débit complémentaire en même temps (Figure-2.5-) pour décoder les plans de bits suivants et courants, respectivement. Le but est de réduire le nombre de retour canal entre le codeur et le décodeur.

A chaque début de décodage d'un paquet, on envoie le débit initial. Si le débit initial n'est pas suffisant pour décoder ce paquet, alors on envoie en même temps un débit complémentaire pour décoder le paquet courant et un débit initial pour les autres paquets qui sont situés dans le même plan de bit. Cette opération est répétée jusqu'à ce qu'on atteigne le nombre total des paquets dans un plan de bit.

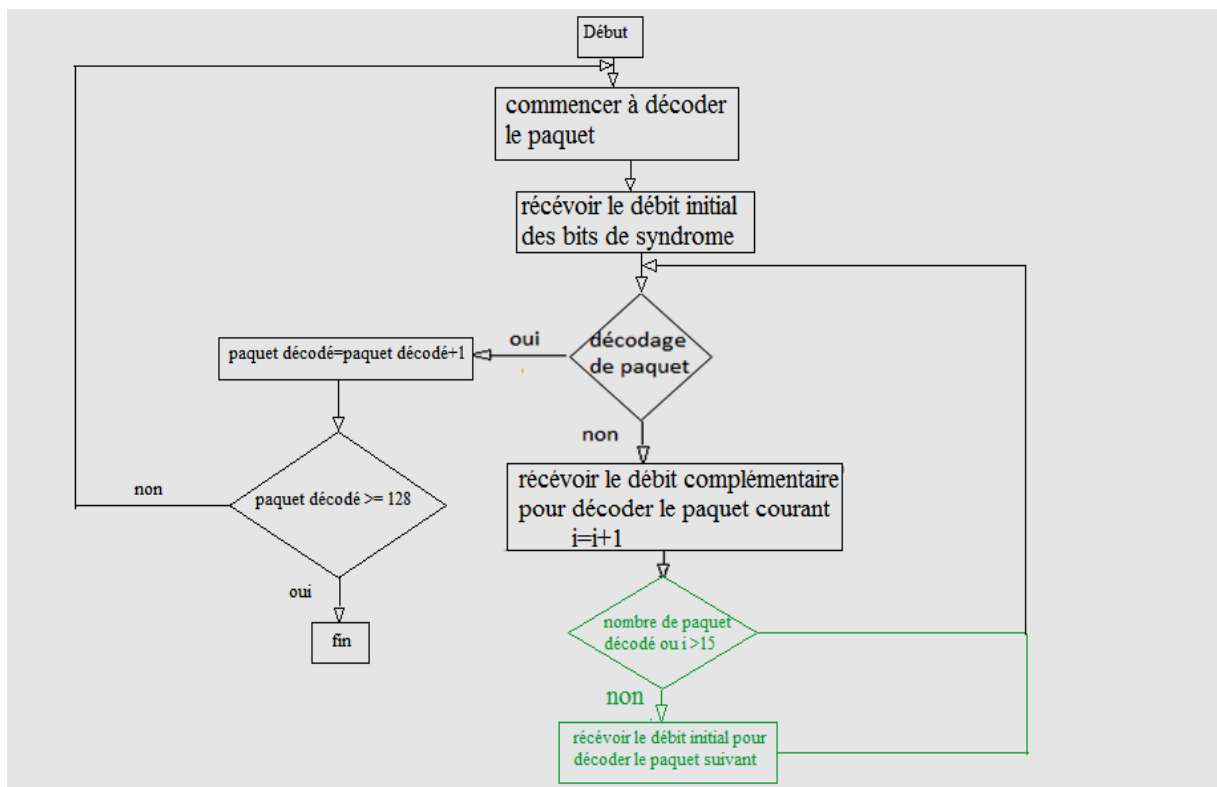


Figure 2.5 Algorithme pour réduire le nombre de retour de canal

4.4 décodage de l'image WZ par propagation de croyances LDPC

La modélisation Laplacienne du bruit entre la SI et la trame WZ originale permet d'estimer les probabilités de bit de l'image WZ, qui sont calculés en tenant compte des plans de bits déjà décodés. Cette probabilité aide le décodeur LDPC pendant l'opération de décodage. Les syndromes qui ont été reçus par le décodeur, sont réorganisés selon l'ordre dans lequel ils ont été organisés dans le codeur. Ces bits de syndrome sont ensuite utilisés par le décodeur LDPC. L'algorithme BP est utilisé pour récupérer les bits de la trame WZ codée. L'algorithme BP est une approche de décodage soft, qui utilise le rapport de vraisemblance logarithmique(LLR) entre les nœuds de source et les nœuds du syndrome.

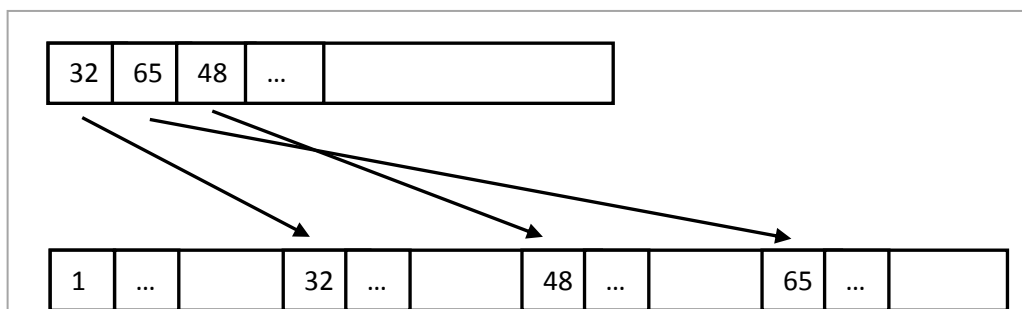


Figure 2.6 Réorganisation des bits de syndrome selon l'ordre de codeur

Dans la pratique, la propagation de croyance est exécutée pour un nombre maximum d'itérations de 100 ou jusqu'à ce que le paquet de syndrome est décodé.

4.5 Protocole de communication entre le décodeur et codeur WZ

L'algorithme de codage LDPC travaille sur un paquet de taille 6336 bits. Alors chaque image WZ est divisée en 128 paquets de taille 6336 bits chacun. Nous avons développé un protocole de communication entre le codeur et décodeur CVD. Le protocole de communication comme illustré par la Figure-2.6- entre le codeur et le décodeur consiste à échanger les informations et les données. Les informations sont envoyées par le décodeur vers le codeur, et elles représentent l'état d'avancement et le succès du décodage dans un paquet de l'image actuelle. Pour chaque paquet, de chacune des images de la séquence vidéo à coder, les données (syndromes) sont envoyées par le codeur vers le décodeur progressivement jusqu'à ce que le paquet soit décodé avec succès.

Au début du décodage de chaque paquet, le décodeur estime le débit minimum (R_{MIN}) de données nécessaires pour décodé le paquet WZ. Cette valeur de R_{MIN} est envoyée au codeur, le codeur envoie alors les bits de syndrome codés de taille ($R_{MIN} \times 96$) au décodeur.

Le décodeur qui reçoit ces données lance l'opération de décodage en exploitant les bits reçus, l'information adjacente et le modèle de bruit de corrélation. Si le décodage du paquet en cours ne réussit pas, le décodeur envoie une information ($R_D = 1$) au codeur pour l'informer de la nécessité d'envoyer un autre bloc de bits de syndrome. Ensuite le codeur envoie un bloc de bits de syndrome de taille ($R_D \times 96$), au décodeur. Une fois ce bloc de données reçu, le décodeur relance l'opération de décodage à nouveau. Cette opération de communication des données se répète itérativement jusqu'à ce que le décodage soit effectué avec succès ou bien jusqu'à l'envoi du paquet entier de syndromes.

Si après l'envoi d'un bloc de bits de syndrome, le décodage du paquet est effectué avec succès, le décodeur réinitialise la valeur de R_D à 0. Il transmet le paquet décodé dans une zone mémoire séparée pour être réassemblé, la valeur de R_D est envoyée au codeur pour l'informer que le décodage est effectué avec succès. Le codeur passe alors au paquet suivant et le transmet au buffer d'émission.

Après le décodage de tous les paquets de l'image, on passe à l'image WZ suivante pour la décoder. Les opérations du protocole de communication sont répétées pour chaque paquet d'image WZ.

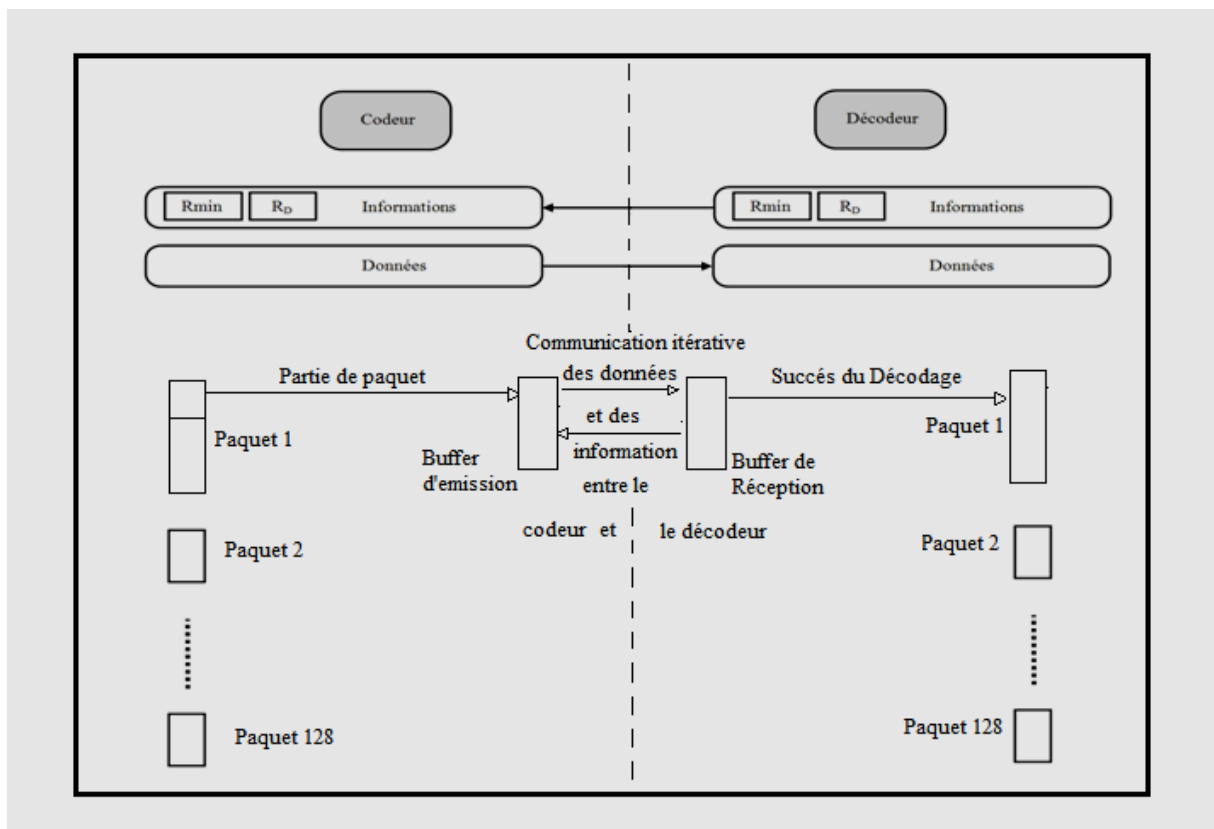


Figure 2.7 Protocole de communication entre codeur et décodeur

Dans le cas où le syndrome complet d'un paquet est envoyé au décodeur, et que celui-ci ne puisse pas décoder, on multiplie le syndrome complet du paquet par l'inverse de la matrice de parité H pour obtenir le paquet décodé.

4.6 Reconstruction de la séquence de vidéo

Pour notre système on a décodé seulement les quatre plans de bit les plus significatifs, donc il reste les quatre plans de bit les moins significatifs, ces derniers seront reconstruit comme suit :

- On remplit les quatre plans de bit les moins significatifs par zéro.
- Pour chaque valeur décodée des quatre plans de bit les plus significatifs, on peut déterminer la valeur min et la valeur max de l'intervalle auquel peut appartenir la valeur décodée finale, des 8 plans de bit décodé.
- Si la valeur du SI appartient à l'intervalle déterminé précédemment, on remplace les quatre plans de bit les moins significatifs par les quatre derniers plans de bit de l'information adjacente (la valeur décodée est égale à SI).
- Si la valeur du SI est supérieure à la valeur max de l'intervalle, on remplace la valeur décodée par cette dernière. Si la valeur de SI est inférieure à valeur min de l'intervalle, on remplace la valeur décodée par cette dernière.

Après que tous les paquets de l'image WZ sont décodés, on utilise un algorithme inverse de l'opération de codage du plan de bit. Le résultat est mis dans une zone mémoire. Si le nombre de paquets décodés atteint le nombre des paquets par image, on combinera, de manière appropriée, les images intra et WZ pour former une séquence vidéo complète. Ensuite on passe à l'affichage et stockage de la vidéo.

5 Conclusion :

Dans ce chapitre, nous avons présenté notre contribution qui consiste à implémenter un système de surveillance basé sur le CVD. Cette contribution est destinée à améliorer les performances des opérations du décodeur telles que : l'accélération de la génération de l'information adjacente, en réduisant le nombre de comparaisons entre les blocs des images. On a aussi réduit le nombre de retour canal entre le codeur et le décodeur par l'envoi du débit complémentaire et initial en même temps pour décoder le paquet courant et suivant respectivement. Et enfin, on a proposé et réalisé un protocole de communication qui permet de gérer l'opération d'échanges des données entre le codeur et le décodeur.

Chapitre 3

Résultats et discussion

1 Introduction :

Dans ce chapitre, nous allons analyser les performances de notre système de codage vidéo distribué appliqué sur des séquences de vidéos de surveillance. Ensuite, on va comparer ces résultats avec les systèmes de codage classiques intra H.264. Nos tests sont basés sur trois critères :

- 1- calculer le PSNR de l'image décodée en fonction du débit de données. On va essentiellement comparer les résultats de notre système avec le codage intra de H.264.
- 2- la différence entre le débit initial et le débit final utilisé pour le décodage des paquets des images WZ. Dans notre travail, nous avons utilisé trois algorithmes de [40] pour estimer le débit minimal initial nécessaire pour décoder les paquets des images WZ. On a étudié l'influence de chaque algorithme sur le débit de décodage et le nombre de retours canal.
- 3- le nombre de retours canal entre le codeur et le décodeur pour décoder l'image WZ. Pour chaque méthode proposée par [40] pour estimer le débit initial, on a calculé le nombre de retours canal qui permet de décoder l'image WZ, ensuite, nous allons observer l'impact de notre idée sur le nombre de retours canal.

Pour tout test effectué sur ces séquences vidéo, on s'intéresse seulement à la luminance Y de l'image, on décode chaque image avec seulement 4 plans de bit maximum.

2 Présentation des séquences de test :

Dans nos tests, nous avons utilisé trois séquences de surveillance de différents scènes. Pour chaque séquence de surveillance, l'évaluation du test est faite pour les 101 images de format CIF avec une fréquence de 30 images/sec :

Street (Figure 3.1): Cette séquence est capturée par une caméra de surveillance installée dans une rue [41]. Dans cette séquence, le changement de mouvement entre les images de vidéo est très lent.

Subway (Figure 3.2): Cette séquence de vidéo est obtenue à partir d'une caméra installée dans une station de métro [42]. Dans cette séquence, le mouvement entre les images de la vidéo est plus rapide.

Hall Monitor (Figure 3.3): Cette séquence est capturée par une caméra de surveillance installée dans un hall [43]. Dans cette séquence, le changement de mouvement entre les images de vidéo est moyen.



Figure 3.1 Séquence de surveillance dans une rue (Street)



Figure 3.2 Séquence de surveillance dans une station de métro (Subway)



Figure 3.3 Séquence de surveillance dans un hall (Hall Monitor)

3 Evaluation et résultats :

3.1 Comparaison de la différence entre le débit initial et final pour décoder une trame WZ

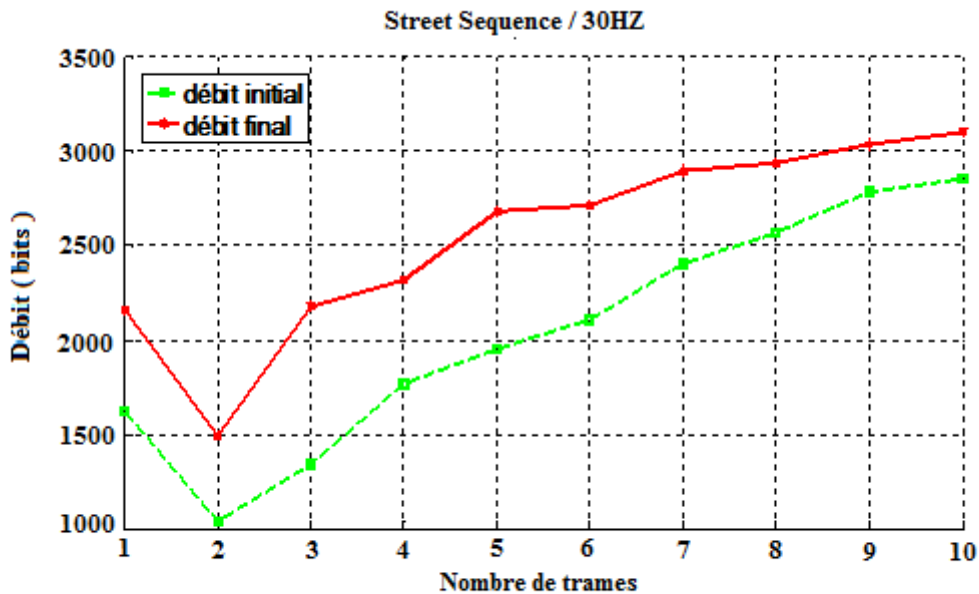
Dans ce test, on prend 10 images WZ et chaque image WZ contient 16 paquets. Pour chaque paquet WZ, on estime le débit minimum initial et on détermine le débit final qui permet de décoder ce dernier. Ensuite on calcule le débit moyen initial et final en bits pour les 16 paquets de l'image WZ. Dans ce test, on va utiliser les trois algorithmes proposés par [40].

$$\text{Débit initial moyen} = \frac{1}{16} \sum_{i=1}^{16} \text{Débit initial de paquet}_i \quad (3.1)$$

$$\text{Débit final moyen} = \frac{1}{16} \sum_{i=1}^{16} \text{Débit final de paquet}_i \quad (3.2)$$

Les Figures 3.4 (a), (b) et (c), représentent la différence moyenne entre le débit initial et final et cela en utilisant l'algorithme basé sur la corrélation temporelle (TC), on remarque que la différence moyenne entre le débit initial et final est 637 bits pour la Figure 3.4 (a) et (c) et 938 bits pour la Figure 3.4 (b).

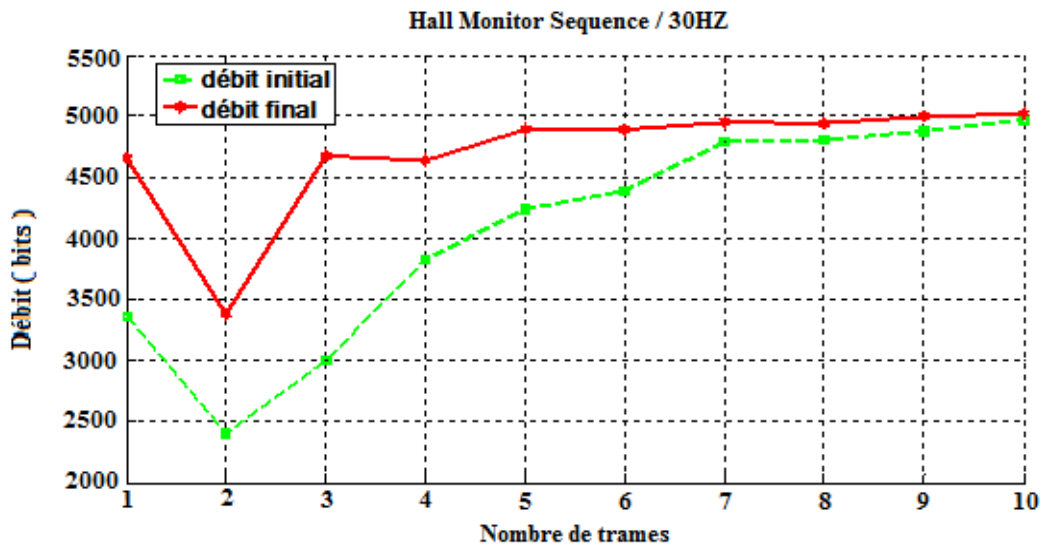
Puisque cette différence est importante, on considère donc que cette technique n'est pas efficace pour estimer le débit initial utilisé pour le décodage des paquets de l'image WZ. On a aussi remarqué que pour beaucoup d'images, le débit initial a dépassé le débit final.



(a)



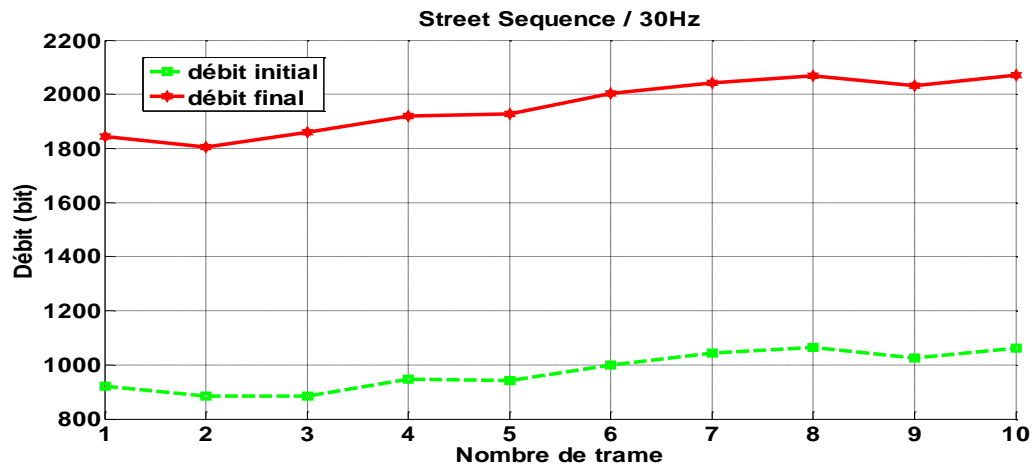
(b)



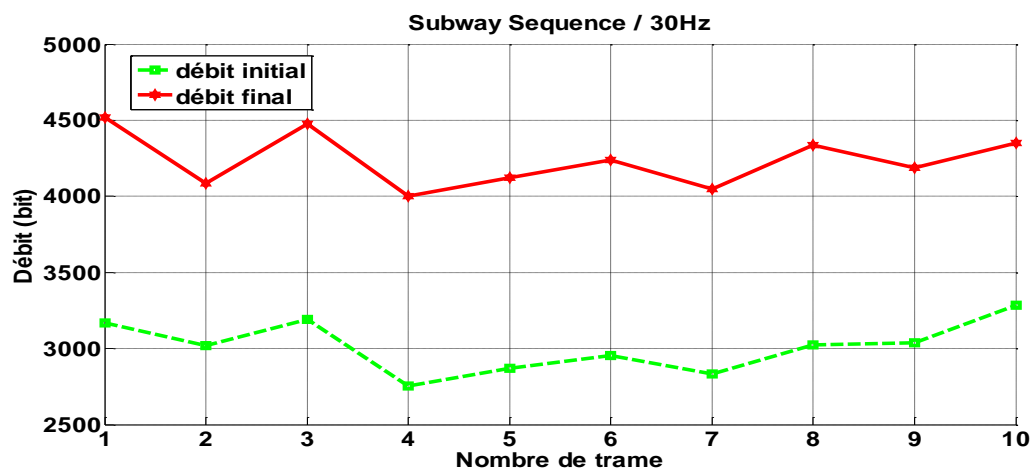
(c)

Figure 3.4 les débits finaux et initiaux estimés par corrélation temporelle. (a) pour la séquence Street, (b) pour la séquence Subway et (c) pour la séquence Hall Monitor.

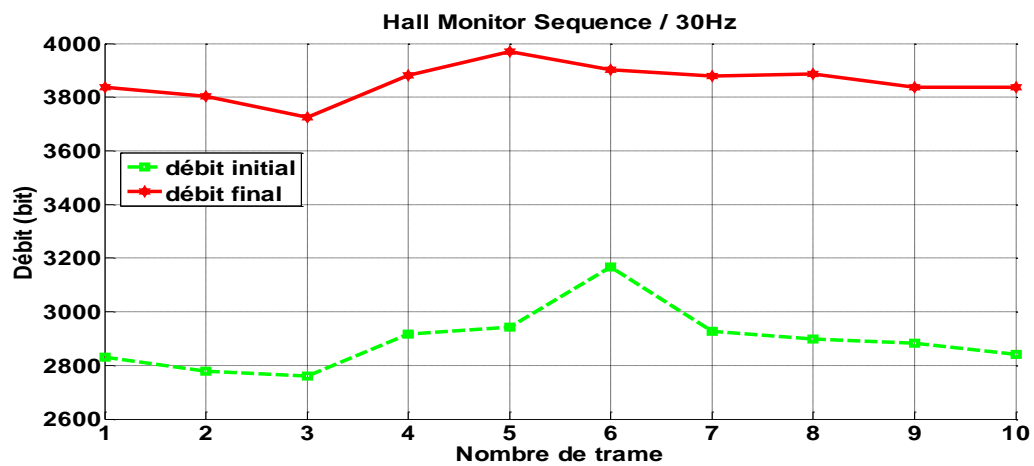
La différence moyenne pour le décodage des images WZ est encore plus élevée dans le cas de l'utilisation de l'algorithme de corrélation entre les plans de bits. Les figures 3.5 (a), (b) et (c) montrent ce résultat (748 bits pour (a), 1222 bits pour (b) et 963 bits pour (c)). L'autre inconvénient dans cette méthode est le temps de décodage qui est très important.



(a)



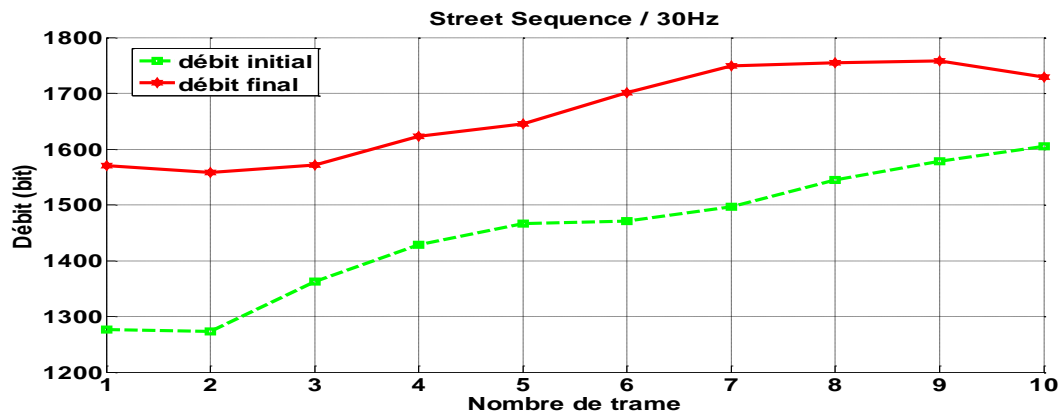
(b)



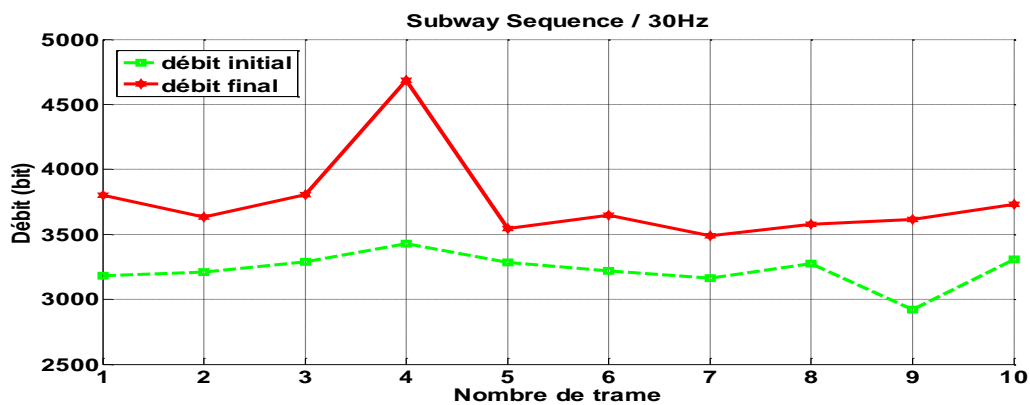
(c)

Figure 3.5 les débits finaux et initiaux estimés par corrélation entre les plans de bit. (a) pour la séquence Street, (b) pour la séquence Subway et (c) pour la séquence Hall Monitor

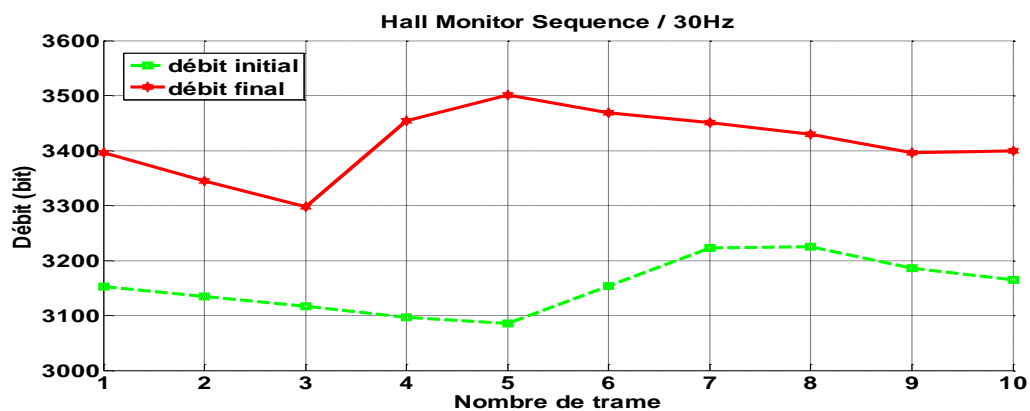
Pour l'algorithme basé sur la valeur médiane (Figure-3.6), on remarque que la différence moyenne entre les débits initial et final pour décoder les paquets des images WZ est la plus petite (182 bit pour (a), 526 pour (b) et 262 pour (c), alors le débit minimal initial et final sont proches. Donc, l'avantage c'est d'utiliser moins de retours canal pour le décodage des paquets de l'image WZ.



(a)



(b)



(c)

Figure 3.6 les débits finaux et initiaux estimés par valeur médiane. (a) pour la séquence Street, (b) pour la séquence Subway et (c) pour la séquence Hall Monitor.

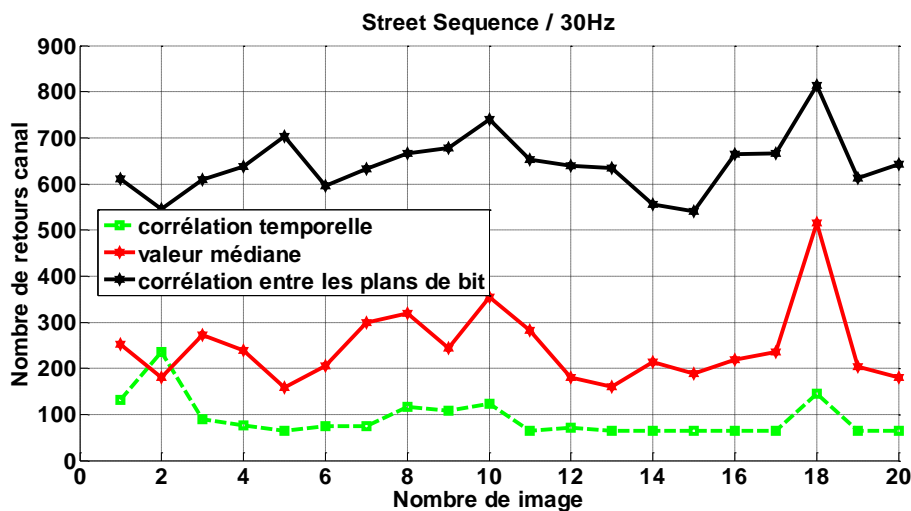
3.2 Comparaison entre le nombre de retours canal et le nombre de bits utilisés pour décoder l'image WZ

Dans cette comparaison, on prend 20 images WZ pour chaque image décodée et on calcule le nombre de retours canal (Figure 3.7) et le nombre de bits qui permet de décoder chaque image (Figure -3.8). Ensuite on compare les trois algorithmes par rapport à ces deux critères.

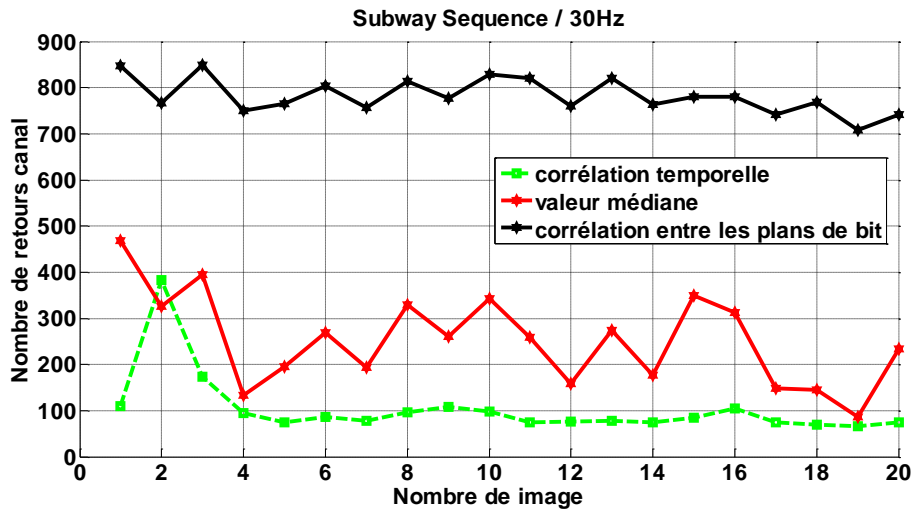
Dans la première méthode (TC), on remarque que le nombre de retours canal est très petit (entre 100 et 70 pour (a), entre 100 et 80 pour (b) et entre 150 et 250 pour (c)), donc le temps de décodage est plus court. L'inconvénient c'est que le débit initial peut dépasser le débit final de décodage des paquets de l'image WZ.

Dans la deuxième méthode (corrélation entre les plans de bits), le nombre de retours canal est très important (entre 500 et 600 pour (a), entre 700 et 850 pour (b) et entre 1100 et 1300 pour (c)), alors le temps de décodage est très grand, mais le nombre de bits qui est utilisé pour décoder la trame WZ est le plus petit.

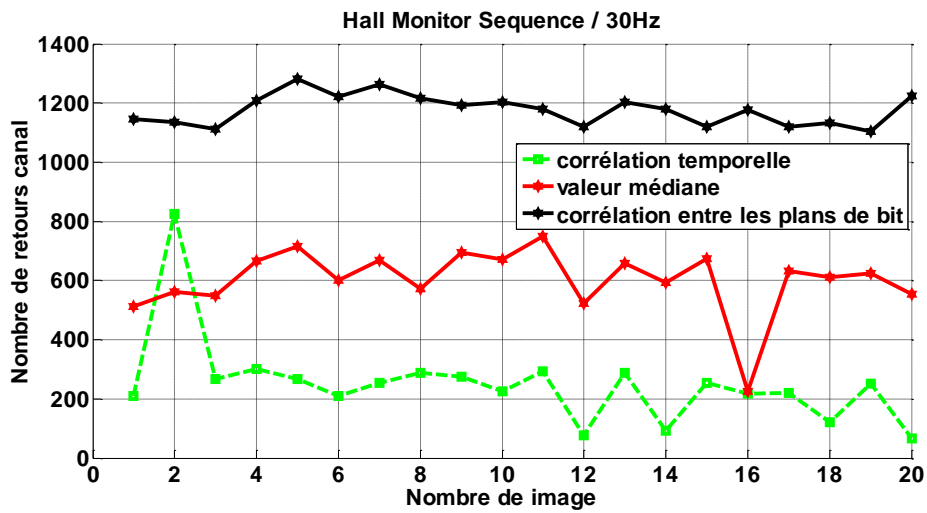
L'algorithme de la valeur médiane est la solution intermédiaire parce qu'il présente un compromis entre le nombre de retours canal et le nombre de bits utilisés pour le décodage de l'image WZ.



(a)

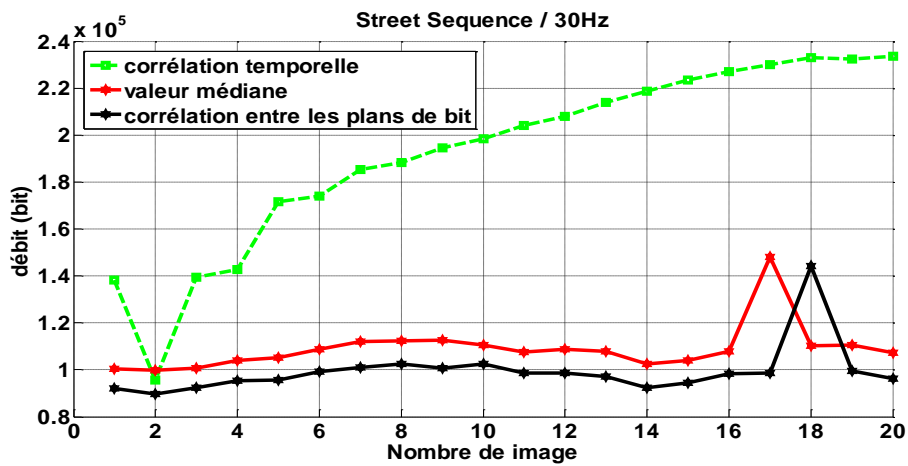


(b)

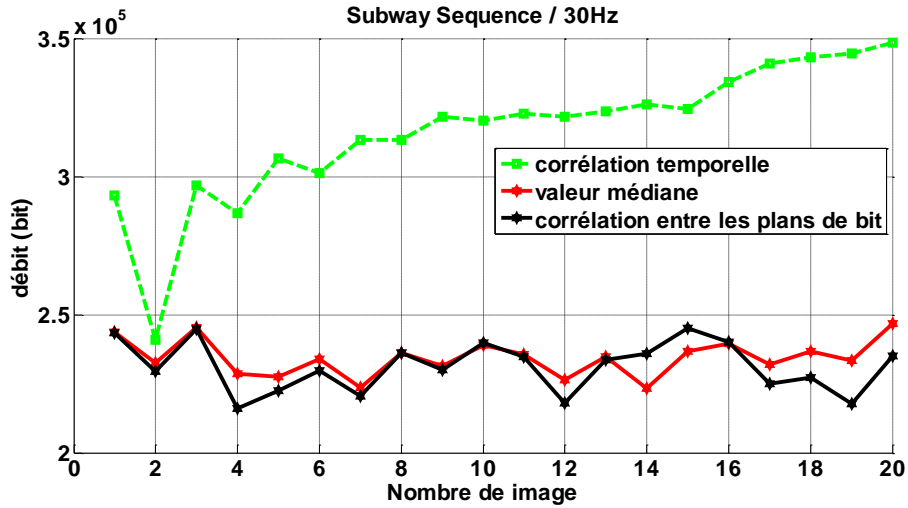


(c)

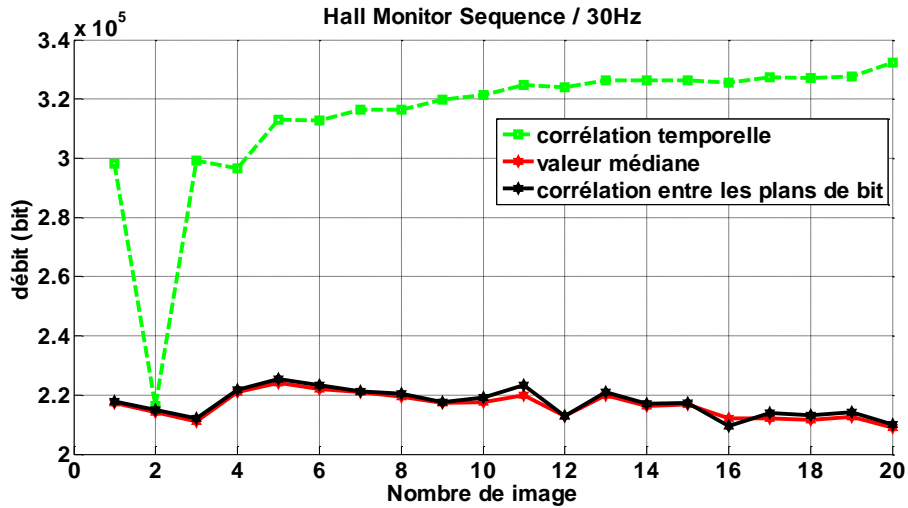
Figure 3.7 nombre le retour de canal pour décoder l'image WZ. (a) pour Street, (b) pour subway et (c) pour Hall Monitor.



(a)



(b)



(c)

Figure 3.8 nombre de bits utilisés pour décoder l'image WZ. (a) pour Street, (b) pour Subway et (c) pour Hall Monitor.

3.3 Les résultats de la performance Débit-PSNR

Dans ce test on prend 50 images. Ces images sont codées par le codeur intra et WZ. Pour chaque codeur, on a calculé le PSNR en fonction du débit de données, utilisé pour décoder les images.

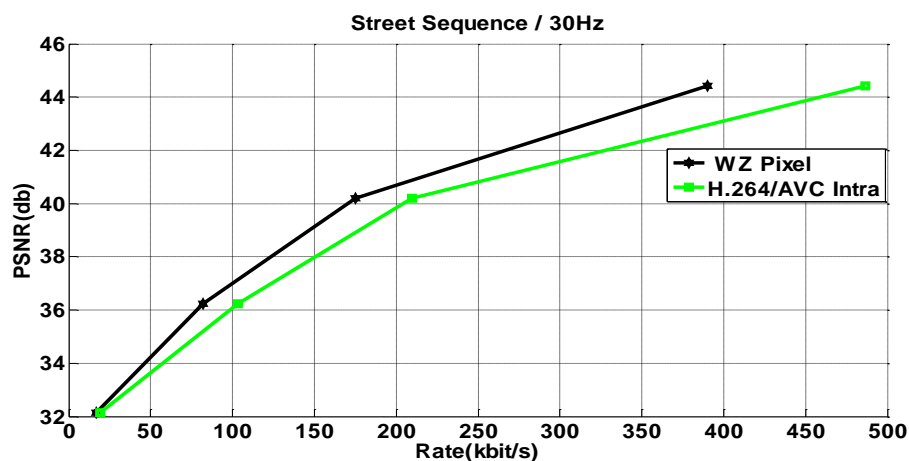
Les Figures 3.9 (a), (b) et (c) représentent le résultat de débit-distorsion pour les trois séquences vidéo surveillance. Le PSNR de l'image décodée varie selon la qualité du SI et le débit de chaque plan de bit envoyé par le codeur. On remarque que plus le débit est grand, plus la qualité de l'image décodé augmente ; ce qui est normal. La variation du PSNR est lente par rapport à l'augmentation du débit, parce que plus on augmente le niveau du plan de

bit, plus les erreurs apparaissent entre le SI et l'image originale, donc c'est pour ça que le taux de croissance du PSNR est plus lent par rapport à l'augmentation du débit.

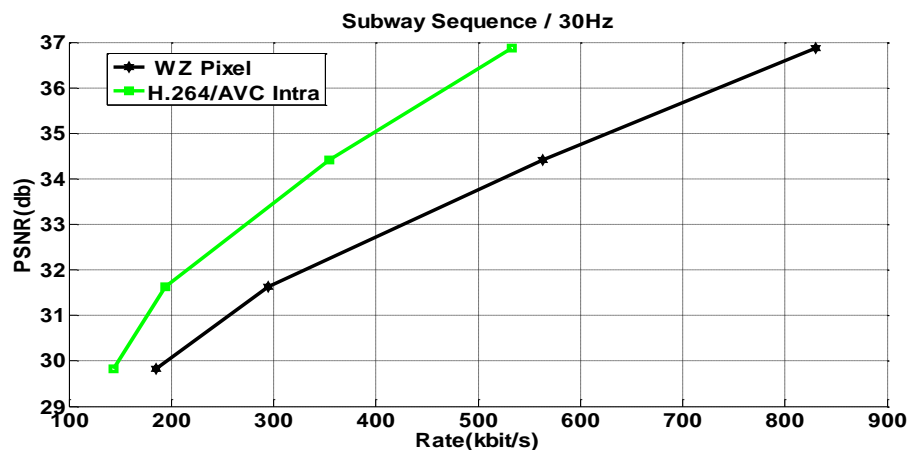
Dans certains cas, le PSNR de l'image décodée est grand malgré que le débit utilisé pour décoder l'image WZ est petit (Figure 3.9 (a)), ça veut dire que la qualité de la SI est bonne. Dans d'autres cas, la qualité de l'image décodée est mauvaise malgré que le débit utilisé pour décoder l'image WZ est grand (Figure 3.10 (b)), ceci veut dire que la qualité de la SI est mauvaise.

D'après la Figure 3.9 (a) et (c), on remarque que notre système WZ surpasse les résultats obtenus par le codeur classique intra H.264, mais pour la Figure 3.9 (b), les résultats du codeur intra H.264 sont meilleurs que notre système WZ.

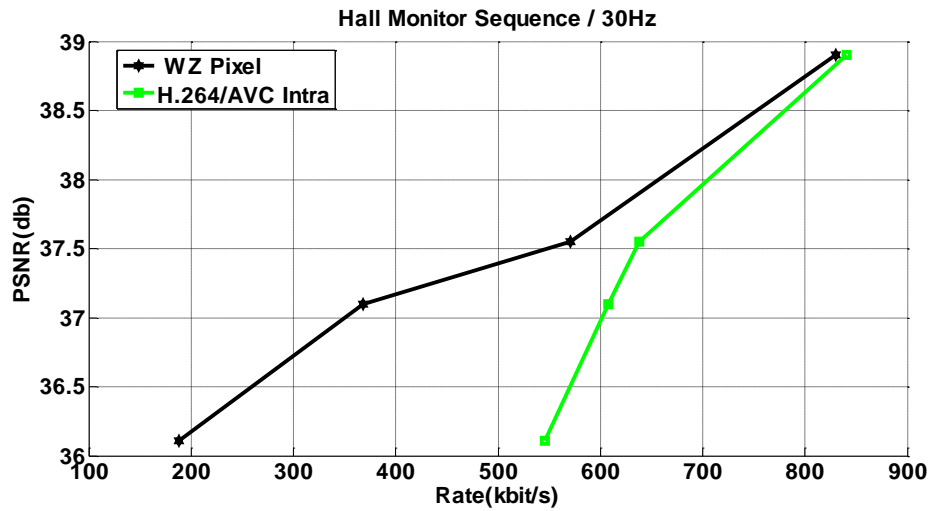
D'après les résultats que nous avons obtenus (Figure 3.9 (a), (b) et (c)), on a constaté que pour les séquences vidéo qui contiennent un mouvement moyen ou faible, il est préférable d'utiliser le codage WZ. Par contre, pour les séquences vidéo qui contiennent un mouvement fort, il est préférable d'utiliser le codeur intra H.264.



(a)



(b)

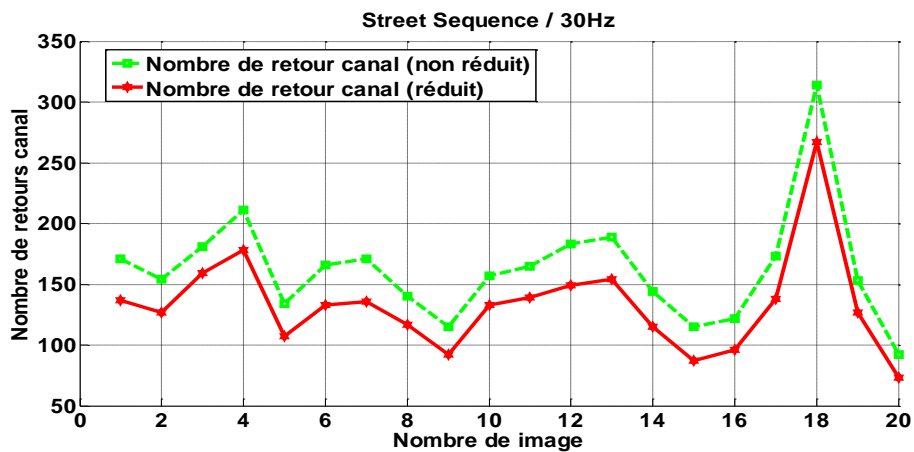


(c)

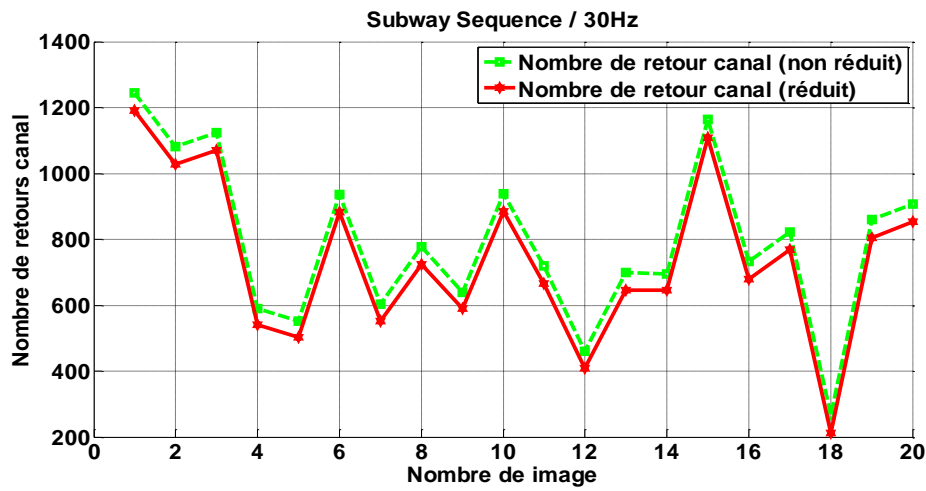
Figure 3.9 PSNR de l'image en fonction de débit. (a) pour Street, (b) pour Subway et (c) pour Hall monitor.

3.4 Réduire le nombre de retours canal

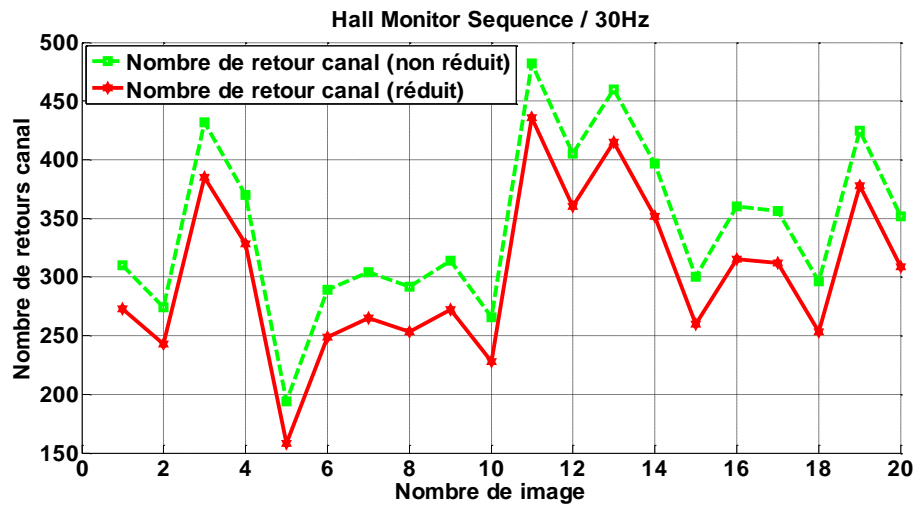
Dans ce test, on utilise notre proposition pour réduire le nombre de retours canal entre le codeur et le décodeur. Selon la Figure 3.10 (a), (b) et (c), le nombre de retours canal a diminué d'une façon considérable. Donc, cette technique permet de réduire le temps de décodage de l'image WZ.



(a)



(b)



(c)

Figure 3.10 nombre de retour de canal.(a) pour Street, (b) pour Subway et (c) pour Hall monitor.

4 Les Résultat pour notre système de surveillance en temps réel.

Dans cette partie, nous allons présenter les résultats obtenus par notre système de surveillance pour un codage en temps réel, cette présentation est illustrée par notre propre séquence vidéo.

Nous avons installé une webcam dans le hall du laboratoire. Le codeur et le décodeur se trouvent dans des endroits différents. La webcam est située au niveau du codeur pour l'acquisition et codage de la vidéo, ensuite les images codées sont envoyées vers le décodeur à travers un réseau wifi.

Pour gérer l'opération du codage et décodage, nous avons implémenté le protocole de communication qui permet de suivre l'état d'avancement du décodage au niveau du décodeur.

Au début, le décodeur établit la connexion avec le codeur, ensuite le codeur lance l'opération de capture et de codage des images vidéo, qui sont envoyées vers le décodeur pour le décodage, l'affichage et le stockage.

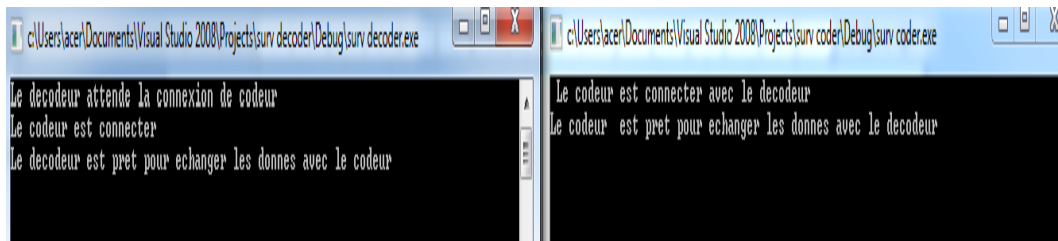
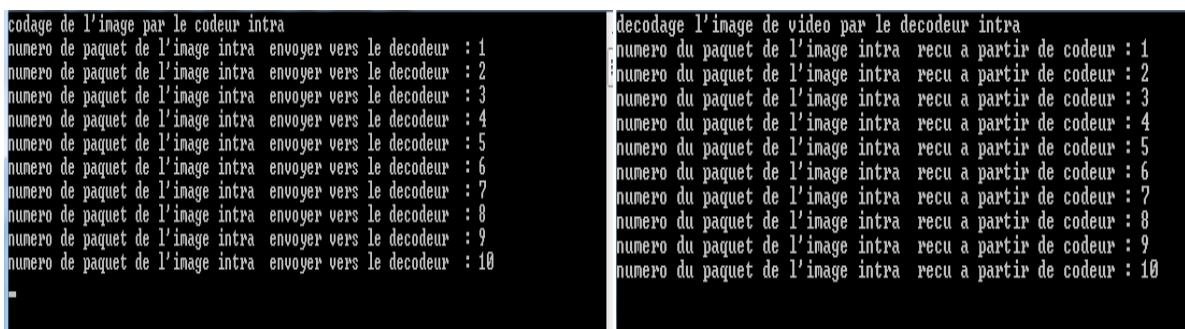


Figure 3.11 Etablissement de connexion entre le codeur et décodeur

Le codeur commence par coder l'image en mode intra et envoie l'information vers le décodeur. Le décodeur reçoit et décode cette image intra. Après, le codeur code l'image suivante en mode WZ et la transmet vers le décodeur pour la décoder. Cette opération est répétée cycliquement pour toutes les images de la vidéo.



(a)

(b)

Figure 3.12 Opération codage et décodage de l'image intra entre codeur (a) et décodeur (b)



(a)

(b)

Figure 3.13 Résultat d'une opération de codage réelle. (a) Image intra au niveau de codeur et (b) Image intra au niveau de décodeur.

Pour les images WZ de la séquence vidéo, au niveau du codeur, chaque image est divisée en 128 paquets. Le décodeur estime le débit initial nécessaire pour décoder chaque paquet de l'image WZ et l'envoi vers le codeur. Le codeur reçoit cette demande de débit et envoie les données demandées vers le décodeur. Si ce débit est suffisant pour décoder ce paquet, le décodeur passe au paquet suivant. Sinon, le décodeur demande au codeur plus de données pour le décodage.

```

c:\Users\acer\Documents\Visual Studio 2008\Projects\surv coder\Debug\surv coder.exe
envoyer le debit complementaire requirant par le decodeur :96 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
le paquet est decoder
numero de paquet envoye est :8
envoyer le debit initiale requirant par le decodeur :960 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
le paquet est decoder
numero de paquet envoye est :9
envoyer le debit initiale requirant par le decodeur :960 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
envoyer le debit complementaire requirant par le decodeur :96 bit
le paquet est decoder
numero de paquet envoye est :10
envoyer le debit initiale requirant par le decodeur :960 bit
envoyer le debit complementaire requirant par le decodeur :96 bit

c:\Users\acer\Documents\Visual Studio 2008\Projects\surv decoder\Debug\surv decoder.exe
Donner moi le debit initiale pour decoder le paquet :960bit
Debit complementaire requirant par le decodeur est :96 bit
Debit complementaire requirant par le decodeur est :96 bit
Debit complementaire requirant par le decodeur est :96 bit
Debit complementaire requirant par le decodeur est :96 bit
Debit complementaire requirant par le decodeur est :96 bit
Debit complementaire requirant par le decodeur est :96 bit
Debit complementaire requirant par le decodeur est :96 bit
le paquet est decoder
nombre de bit utiliser pour decoder le paquet est : 1536
numero paquet recu est :9
Donner moi le debit initiale pour decoder le paquet :960bit
Debit complementaire requirant par le decodeur est :96 bit
Debit complementaire requirant par le decodeur est :96 bit
Debit complementaire requirant par le decodeur est :96 bit
Debit complementaire requirant par le decodeur est :96 bit
Debit complementaire requirant par le decodeur est :96 bit
Debit complementaire requirant par le decodeur est :96 bit
Debit complementaire requirant par le decodeur est :96 bit
le paquet est decoder
nombre de bit utiliser pour decoder le paquet est : 1536
numero paquet recu est :10
Donner moi le debit initiale pour decoder le paquet :960bit
Debit complementaire requirant par le decodeur est :96 bit
  
```

(a) (b)

Figure 3.14 Protocole de communication entre le codeur (a) et décodeur (b)

Si tous les paquets des plans de bits sont décodés, on passe à l’affichage et stockage des images de la vidéo.



(a) (b)

Figure 3.15 Résultat d’une opération réelle de codage. Image WZ au niveau de codeur (a) et décodeur (b)

Durant l'émission des images vidéo à partir du codeur vers le décodeur, il peut y avoir un bruit qui va créer des erreurs. Ceci nécessite d'envoyer plus de débit pour les corriger ou les diminuer. Ce bruit peut aussi influencer l'opération de reconstruction de l'image vidéo.

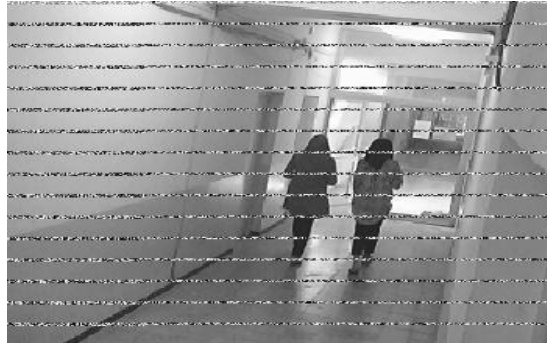


Figure 3.16 Influence de bruit sur la reconstruction de l'image au niveau de décodeur

Pour notre contribution à la réduction du nombre de retours canal entre le codeur et décodeur. On a remarqué que le temps de décodage de l'image WZ a diminué avec un taux de 12%.

Enfin, on remarque que notre application de surveillance n'est pas encore une application en temps réel, car le temps de décodage de l'image est important et cela est dû à la complexité du décodage LDPC et l'estimation du débit initial, qui n'est pas suffisamment proche du débit final pour permettre de décoder rapidement les paquets de l'image.

5 Conclusion

Dans ce chapitre, nous avons discuté les résultats obtenus par notre système de surveillance basé sur le CVD. Pour l'estimation du débit initial des données utilisé pour décoder les paquets de l'image, nous avons trouvé que la méthode basée sur la valeur médiane est la plus convenable. Pour notre application, cette technique présente un compromis acceptable entre le débit des données et le nombre de retours canal, entre le codeur et décodeur.

Pour réduire le nombre de retours canal, on a remarqué que notre proposition permet de réduire le nombre de retours canal d'une façon considérable et par conséquent elle permet de réduire le temps de décodage de l'image.

Conclusion générale

Le codage vidéo distribué est un nouveau modèle de codage vidéo. Son principe repose sur les travaux de Slepian-Wolf et de Wyner-Ziv. Contrairement au codage classique, la complexité du codage dans le CVD est transférée du codeur vers le décodeur. Ce transfert de complexité est très demandé par les nouvelles applications de vidéo telles que les caméras de téléphonie mobile et de vidéo sans fil qui ont une puissance de calcul limitée et une autonomie d'énergie faible.

Dans ce travail, nous avons implémenté un système de surveillance basé sur le CVD. Notre contribution consiste à implémenter un protocole de communication entre le codeur et le décodeur pour gérer l'opération du codage et décodage. Ensuite, nous avons exploité trois algorithmes proposés par [40] pour estimer le débit initial, pour décoder les images WZ. Après un test, on a choisi le meilleur qui est le plus convenable pour notre système de surveillance. Ensuite, on a proposé une méthode qui permet de réduire le nombre de retours canal entre le codeur et le décodeur et par conséquent réduire le temps de décodage.

Nous avons aussi proposé et réalisé un protocole de communication entre le codeur et le décodeur. Il consiste à échanger les informations de gestion de communication et des données d'image. Les informations sont envoyées par le décodeur au codeur, et elles représentent l'état d'avancement du décodage pour un paquet donné de l'image en cours. Les données sont des parties de l'image envoyées par le codeur au décodeur progressivement jusqu'à ce que l'image soit décodée avec succès. Au début du décodage de chaque paquet, le décodeur estime le débit minimum de données nécessaires pour décoder le paquet WZ et l'envoi vers le codeur, le codeur envoie alors les bits de syndrome codés au décodeur. Le décodeur reçoit ces données et lance l'opération de décodage. Si le décodage du paquet en cours ne réussit pas, le décodeur envoie une information au codeur pour l'informer de la nécessité d'envoyer un autre bloc de bits de syndromes. Ensuite le codeur envoie un bloc de bits de syndrome, le décodeur reçoit ce bloc de données et relance l'opération de décodage à nouveau. Cette opération de communication des données se répète itérativement jusqu'à ce que le décodage soit effectué avec succès ou bien jusqu'à l'envoi du paquet complet de syndromes.

Pour assurer un débit de données initial suffisant pour décoder l'image WZ, et diminuer le nombre d'itérations de retours canal entre le codeur et le décodeur, on a exploité les trois algorithmes qui sont proposés par [40]. On a appliqué ces trois algorithmes dans le cas

de notre système. On a trouvé que l'algorithme basé sur la valeur médiane est le plus convenable pour notre application.

Nous avons aussi introduit dans notre implémentation une technique pour réduire les délais dus à l'utilisation du canal de retour. On a proposé d'envoyer les données pour décoder le paquet de données WZ courant et les autres paquets suivant en même temps. Dans cette opération, on commence par décoder le paquet de données courant, si le débit initial n'est pas suffisant pour le décoder, on envoie en même temps des données nécessaires pour compléter son décodage et un débit initial de données pour décoder les paquets suivants. Ceci a permis de réduire le nombre de retours canal entre le codeur et le décodeur.

Suggestion de travaux futurs

Pour que le CVD soit compétitif et performant vis-à-vis des standards vidéo conventionnels, plusieurs étapes importantes sont à réaliser, avant de voir un dispositif vidéo opérationnel se basant sur ce nouveau paradigme.

Dans la plupart des architectures de codage vidéo distribué, la génération de l'information adjacente est basée beaucoup plus sur les images intra par interpolation ou extrapolation. Dans un décodeur WZ domaine pixel, le décodage de l'image se fait paquet par paquet, donc on peut exploiter chaque nouvelle information sur l'image WZ décodée pour améliorer la qualité de la SI, pour les paquets restants. L'estimation de mouvement peut être réalisée entre le paquet décodé et les autres paquets qui se trouvent dans la même position et dans d'autres plans de bits.

Pour réduire le nombre d'utilisation du canal, il faut trouver des algorithmes qui permettent d'estimer un débit minimal initial qui soit très proche du débit final pour décoder la trame WZ. Ceci permet d'éviter la fragmentation des données et de réduire le temps de décodage de la trame WZ.

Dans certaines applications, où le canal de retour est inexistant ou lorsque le délai de décodage ne peut être toléré, le contrôle du débit doit se faire au niveau du codeur. Il est donc nécessaire de trouver des méthodes efficaces pour générer la SI au niveau du codeur, sans utiliser beaucoup de ressources.

Finalement, l'algorithme de la propagation de croyance pour le décodage LDPC s'appuie sur le fait que chaque code produit peut être décodé indépendamment parce que les mots de code sont codés indépendamment. Donc, le parallélisme de la procédure du décodage permet de réduire le délai de décodage. Ceci permettra d'assurer le décodage de la vidéo en temps réel.

Bibliographie

- [1] B. Girod, A. Aaron, S. Rane and D. Rebollo-Monedero, "Distributed video coding," in Proc IEEE, Special Issue on Video Coding and Delivery, vol. 93, no. 1, pp. 71-83, Jan. 2005.
- [2] J. Slepian and J. Wolf, "Noiseless Coding of Correlated Information Sources," IEEE transactions on Information Theory, Vol. 19, No. 4, pp. 471-480, July 1973.
- [3] A. Wyner and J. Ziv, "the rate-distortion function for source coding with side information at the decoder" , IEEE transactions on information theory, vol. 22, no. 1, pp. 1-10, january 1976.
- [4] A. Boudjada, "Codage Vidéo Distribué", thèse magister 2008, www.umc.edu.dz/buc/theses/electronique/BOU5256.pdf 2008.
- [5] A. Aaron, B. Girod, "Compression with side information using turbo codes" IEEE Data Compression Conference, 2002.
- [6] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, M. Ouaret "the discover codec: architecture, techniques and evaluation", Picture Coding Symposium, Vol. 17, No. 9, pp. 1103-1120, November 2007.
- [7] R. Puri, A. Majumdar, K. Ramchandran , "PRISM: A video coding paradigm with motion estimation at the decoder" , IEEE Allerton Conference on Communication, Control, and Computing, Vol. 16, No. 10, pp. 2436-2448, 2007.
- [8] R. Puri, K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles", EECS Department, University of California, Berkeley, 2002.
- [9] F. Pereira, L. Torres, C. Guillemot, T. Ebrahimi, R. Leonardi, S. Klomp, "Distributed Video Coding: Selecting the most promising application scenarios," *Signal Processing: Image Communication*, vol. 23 no. 5, pp. 339-352, Jun. 2008.
- [10] A. Aaron, R. U. I. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in Proc of the 36th Asilomar Conference on Signals Systems and Computers, pp. 240-244, Pacific Grove, Calif, USA, Nov. 2002.
- [11] J. Ascenso, C. Brites, and F. Pereira, "Motion compensated refinement for low complexity pixel based distributed video coding", IEEE Conference Advanced Video and Signal Based Surveillance, 2005.
- [12] X. Artigas, L. Torres, "iterative generation of motion-compensated side information for distributed video coding", IEEE International Conference on Image Processing(ICIP), 2005, Vol 1, pp. 240-244, Sept. 2005.

- [13] J. Ascenso, C. Brites and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding" 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, July.2007.
- [14] L. Natario, C. Brites, J. Ascenso and F. Pereira, "Extrapolating Side Information for Low-Delay Pixel Domain Distributed Video Coding", Lectures Notes in Computer Science, Vol. 3893, pp. 16 - 21, 2006.
- [15] H. Luong, L. Raket, X. Huang, S. Forchhammer, "Side Information and Noise Learning for Distributed Video Coding using Optical Flow and Clustering,"IEEE transactions on, Image Processing, Vol. 21, pp. 4782 - 4796, Dec. 2012.
- [16] A. B. B. Adikari, W. A. C. Fernando, W. A. R. J. Weerakkody, "Multiple Side Information Streams for Distributed Video Coding," IET Digital Library , Vol. 25, pp. 1447 - 1449, December 2006.
- [17] Akinola, Mobolaji, Dooley, Laurence and Wong, Patrick "Wyner-Ziv side information generation using a higher order piecewise trajectory temporal interpolation algorithm", International Conference on Graphic and Image Processing (ICGIP), Manila, Philippines, Dec. 2010.
- [18] D. Varodayan, D. Chen, M. Flierl and B. Girod, "Wyner-Ziv coding of video with unsupervised motion vector learning", Journal Signal Processing: Image Communication, Vol. 23, pp. 369 - 378, June 2008.
- [19] M. Zheng, F. H. Ali, "Exploration and Exploitation of Reference Frames in Distributed Video Coding" ,IEEE Signal Processing Letters , Vol. 19, no. 7, pp. 411-414, July 2012.
- [20] R. Martins, C. Brites, J. Ascenso and F. Pereira, "Statistical motion learning for improved transform domain Wyner-Ziv video coding", Journal Image Processing, IET, Vol 4, no. 1, pp. 28-41, Feb. 2010.
- [21] Laraba.M, «Codage vidéo distribué des séquences vidéo périodiques», www.umc.edu.dz/buc/theses/electronique/LAR6470.pdf 2013.
- [22] Benierbeh. S, Khamadja. M, "Hybrid Wyner-Ziv and intra video coding with partial matching motion estimation at the decoder," Image Processing (ICIP), 16th IEEE International Conference on pp.2925 - 2928, 2009.
- [23] Mohamed HAJ TAIEB "Codage Vidéo Distribué utilisant les Turbo Codes", www.Theses.ulaval.ca/2013/30170/30170.pdf 2013.
- [24] J. Ascenso, C. Brites and, F. Pereira, "Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding", EURASIP, Slovak republic, July, 2005.

- [25] C. Brites, J. Ascenso, F. Pereira, “Modeling Correlation Noise Statistics at Decoder for Pixel Based Wyner-Ziv Video Coding”, Picture Coding Symposium (*PCS*), Beijing, China, April 2006.
- [26] H. V. Luong , X. Huang and S. Forchhammer, “Adaptive noise model for transform domain Wyner-Ziv video using clustering of DCT blocks”, Conference Multimedia Signal Processing (MMSP), IEEE 13th International Workshop on, October 2011.
- [27] X. Huang, S. Forchhammer, “Improved virtual channel noise model for transform domain Wyner–Ziv video coding”, IEEE International Conference Acoustics, Speech and Signal Processing, ICASSP , pp.921-924, April 2009.
- [28] J. Song, K. Wang, H .Liu, Y. Li, C. Wu, “Progressive correlation noise refinement for transform domain Wyner-Ziv Video Coding”, in Proc. ICIP, pp.2625-2628, 2011.
- [29] H. V. Luong, S. Forchhammer, “Noise residual learning for noise modeling in distributed video coding”, IEEE, Conference Picture Coding Symposium, pp.157-160, 2012.
- [30] C. Brites, J. Ascenso, and F. Pereira , “Feedback channel in pixel domain Wyner-Ziv video coding: myths and realities” ,14th European Signal Processing Conference (EUSIPCO), Florence, Italy, September 4-8, 2006.
- [31] C. Fu ,J. Kim, “Encoder rate control for block-based distributed video coding” , Multimedia Signal Processing (MMSP) ,2010 IEEE International Workshop on, USA ,pp. 333-338, Oct. 2010.
- [32] Y.P Lee et. al, “ A cost effective architecture for 8x8 two-dimensional DCT/IDCT using direct method”, Circuits and Systems for Video Technology, IEEE transactions on V 7 , Issue 3 , pp. 459-476, 2002.
- [33] K. Ray-Liu and C. T. Chiu, "Unified parallel lattice structures for time-recursive discrete cosine/sine/Hartley transforms", IEEE Transactions on ,Signal Processing, V 41 , N° 3 , pp. 1357-1377, 2002.
- [34] S. b. Pan, S.S. Chae and R.H. Park, “VLSI Architecture for block matching algorithms using systolic arrays" Journal of Signal Processing Systems, Vol.68, N°. 1, pp 127-138, July 2012.
- [35] J. Olivares, J. Hormigo, J. Villalba, and I. Benavides “Minimum Sum of Absolute Differences implementation in a single FPGA device" Dept. of Computer Architecture, University of Malaga, Spain.
- [36] H.V. Luong, X. Huang and S. Forchhammer, “multiple LDPC decoding using bit plane correlation for transform domain wyner-ziv video coding”.IEEE International Conference on: Acoustics, Speech and Signal Processing (ICASSP), 2011.

- [37] H. Zhong and T. Zhang “design of VLSI implementation-oriented LDPC codes”, Electrical, Computer and Systems Engineering Department Rensselaer Polytechnic Institute, Troy, NY, USA.
- [38] S. Momcilovic, Y. Wang, S Rane “Toward Real time Side Information Decoding on Multi-core Processors”, IEEE International Workshop on, (MMSP), pp. 321-326, 2010.
- [39] D. Varodayan, A. Aaron and B. Girod, “Rate-adaptive codes for distributed source, coding,” *EURASIP Signal Processing Journal, Special on Distributed Source Coding*, vol. 86, no 11, pp. 3123-3130, Nov. 2006.
- [40] M. Haj Taieb, J. Y. Chouinard and D. Wang, “Low complexity hybrid rate control schemes for distributed video coding”, ICSP1E , San Francisco, USA, 24-26 October2012.
- [41] http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html
- [42] <http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/>
- [43] <http://trace.eas.asu.edu/yuv/>