

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

UNIVERSITE MENTOURI CONSTANTINE
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE

M E M O I R E

Présentée Pour obtenir

LE DIPLOME DE MAGISTRE EN ELECTRONIQUE

Option : TRAITEMENT DU SIGNAL

Par

Mr : MEKIDECHE Mohammed

Codage des Images Vidéo Couleur
par Compensation Temporelle et Spectrale

Devant le jury :

Président : BENNIA Abdelhak
Rapporteur : KHAMADJA Mohammed
Examineurs : CHAREF Abdelfetah
FARROUKI Attef.

Professeur UMC.
Professeur UMC.
Professeur UMC.
M. de conférence UMC.

Remerciements

Louange à dieu le misericordieux, qui m'a permis d'accomplir ce travail.

Je tiens à remercier vivement mon directeur de mémoire :

Le professeur M. Chammed KHAM ADJA

D'avoir dirigé ce travail, ainsi pour toutes les facilités matérielles qu'il m'a accordé pour

L'accomplissement de ce travail.

Comme je tiens à remercier en particulier le docteur : Said BENYERBAH, pour m'avoir co-dirigé avec compétence tout le long de ce travail, ainsi pour tous les efforts fournis à mon égard.

Je remercie le professeur : Abdelhak BENNIA d'avoir accepté de présider le jury, et également

le professeur : Abdelfetah CHAREF, et le professeur : Attef FARROUKI qui ont accepté

d'être membres du jury

Je remercie toute ma famille, ainsi que tous les amis et collègues, pour leur soutien

et encouragement au cours de mes études

Résumé

En terme du codage vidéo, les normes MPEGx et H.26x utilisent le modèle YUV des images couleur décorréélées spectralement, à la suite d'une transformation linéaire des images RGB. Pour la décorrélation spatiotemporelle, ces normes adoptent une méthode dite Hybride. Cette méthode est basée sur la prédiction par compensation temporelle de l'image courante à partir de l'image précédente en exploitant le vecteur mouvement (MV), et le codage par transformation de l'image résiduelle. Dans le cas des images multispectrales fixes, il existe plusieurs méthodes développées pour la décorrélation spatiale des pixels voisins dans la même bande, et la décorrélation spectrale des pixels dans la même position dans les différentes bandes. Il existe des algorithmes de compression qui utilisent la quantification vectorielle. D'autres algorithmes se basent sur la combinaison des transformations orthogonales telles que : (DCT, DWT, KLT). Et d'autres, combinent les méthodes prédictives intra et inter bandes avec le codage par transformation. En fait, les images RGB sont des images multispectrales à trois bandes, auxquelles les algorithmes de codage précités peuvent être appliqués. Néanmoins, ces algorithmes considèrent des images multispectrales fixes. Dans notre travail, nous avons développé un codeur vidéo des images couleurs modèle RGB, basé sur la méthode de compensation spectrale. Pour la comparaison, nous avons intégré à notre codeur un programme pour le codage vidéo des images couleurs modèle YUV, basé sur la méthode de compensation temporelle. Alors, nous avons obtenu un codeur à plusieurs modes de fonctionnement. Le premier mode MC (Motion Compensated) qui code l'image YUV par compensation temporelle en utilisant le vecteur mouvement, donne les meilleurs résultats, ce qui prouve l'efficacité de la décorrélation temporelle. Cet algorithme est utilisé comme référence pour le développement de nos algorithmes. En fait, le deuxième mode SC (Shift compensated) qui code l'image RGB par décorrélation spectrale en utilisant les vecteurs de décalage (entre les valeurs des pixels dans la même position dans les bandes G et R d'une part, et dans les bandes B et R d'autre part) a donné des résultats acceptables. Nous avons ensuite amélioré le mode SC duquel, nous avons dérivé le nouveau mode noté SC* (Où, les vecteurs de décalage sont ignorés en phase de codage, car, on peut les déduire au niveau du décodeur à partir du vecteur mouvement et la trame de référence). Alors, les performances du mode SC* se trouvent très rapprochées de celles du mode MC.

Abstract

In term of video coding, the standards MPEGx and H.26x use the YUV model of colour images spectrally decorrelated, after a linear transformation of RGB images. For spatial-temporal decorrelation, these standards adopt the Hybride method. This method is based on temporal compensated prediction of current image from preceding image using the motion vector (MV), and transform coding of residual image. In the case of still multispectral images, there are several methods developped for spatial decorrelation of the neighbouring pixels in the same band, and spectral decorrelation of the pixels in the same position into different bands. There is some algorithms of compression use vector quantization. Other algorithms based on combining orthogonal transforms such as (DCT, DWT, KLT) . And others combine predictive methods intra or inter bands with transform coding. In fact, RGB images are three bands multispectral images, to which the preceding algorithms can be applied. But, these algorithms consider only multispectral images. In our work, we have developed a video coder for RGB model of colour images, based on spectral method. For comparison, we had integrated to our coder a program for video coding YUV model of colour images, based on temporal compensation method. Then we obtain a coder operating with several modes. This first one MC (Motion-Compensated) mode which is coding YUV image with temporal compensation shows the best results that proofs the temporal compensated efficiency. This algorithm is used as a reference to compare our algorithms with The second mode SC (Shift-compensated) which Codes RGB images using shift vectors (between pixels value in the same position in G and R bands on the one hand, and in B and R bands on the other hand), yields good results. Then, we improved the SC from which we derived the new mode denoted by SC* (where the shift vectors are ignored in coding phase because, they can be deducted at decoder from motion vector and reference frame). Then, the performances of SC* mode are more close to those for MC mode.

Introduction générale

L'objectif du développement des techniques de compression vidéo est d'économiser les ressources de stockage et l'optimisation des canaux de communication. En fait, les recherches actuelles évoluent dans le sens du développement de nouvelles alternatives, en combinant différentes méthodes acquises, par la réalisation des algorithmes efficaces de compression. Il est clair que, le seul facteur commun entre les différents algorithmes est la réduction au maximum de l'information redondante.

- **Contexte de l'étude**

Dans les applications vidéos, une image en couleur est décrite par un contenu spectral de trois bandes de couleurs fondamentales RGB (Red, Green, Blue), obtenues par filtrage optique, à l'intérieur de la camera, lors de la prise de la scène. Comme les informations portées par les trois bandes sont d'égale importance, alors, on est obligé de conserver la même résolution pour chaque bande, ce qui nécessite un espace de stockage plus grand. En fait, l'espace trichromatique RGB est considéré comme primaire et la plupart des normes de compression développent leurs algorithmes suite à un changement de l'espace de représentation primaire RGB à l'espace secondaire YUV ou encore, YCrCb (luminance, différence de chrominance rouge, différence de chrominance bleue), par application d'une transformation linéaire réversible.

Comme, le choix de cet espace permet une décorrélation spectrale, par la séparation du signal de luminance de ceux de la chrominance, alors on peut soumettre les signaux de chrominance auxquels l'œil est moins sensible à un sous échantillonnage. Par contre on conserve la résolution du signal de luminance qui porte les détails les plus significatifs pour l'œil humain. Le sous échantillonnage des signaux U, V améliore les performances de la compression en profitant des méthodes psychovisuelles. Néanmoins, l'inconvénient de la transformation de l'espace des couleurs est la perte des données dû au calcul et aux sous échantillonnage, et du temps investi pour effectuer la transformation. En outre, les exigences actuelles de l'application de l'imagerie dans les domaines scientifiques et militaires tels que, reconnaissance, prospection et télédétection, nécessitent des images à plusieurs bandes (multispectrales).

Alors, ce grand nombre de bandes spectrales, complique l'utilisation d'une quelconque transformation de l'espace des couleurs. En plus les bandes étant d'égale importance, alors, on est obligé de conserver la même résolution pour les bandes constitutives de l'image, ce qui limite la compression.

- **Approche suivie**

Dans notre travail on s'intéresse à développer un codeur vidéo polyvalent qui envisage la compression des images couleurs, modélisées dans les deux espaces YUV et RGB. Pour le codage des images YUV, on adopte la méthode hybride basée sur la prédiction de l'image courante par compensation temporelle, et le codage spatial par DCT de l'image résiduelle. Pour le codage des images RGB, nous développons une méthode de compression basée sur la compensation spectrale.

On signale que cette dernière méthode est basée sur l'estimation des vecteurs de décalage SVs (Shift Vectors), entre les pixels qui se trouvent dans la même position de la bande G et la bande R et entre les pixels qui se trouvent dans la même position de la bande B et la bande R. Ensuite, on utilise ces deux vecteurs pour la prédiction par compensation du décalage SCP (Shift Compensated Prediction). Pour mettre en évidence l'efficacité de notre codeur, nous allons comparer les résultats obtenus à l'issue du codage d'une séquences formées d'images YUV par compensation temporelle et la même séquence d'images RGB codée par compensation spectrale.

- **Objectifs visés**

On vise par le développement de ce codeur, une application en codage vidéo des images modélisées RGB par compensation spectrale, ce codeur peut être étendu au codage des images multispectrales (nombre de bandes supérieures à trois). Le travail avec le modèle du sous échantillonnage 4:4:4 (même résolution pour les différentes bandes) trouve des applications dans les productions d'images de hautes qualités. En plus, par la combinaison des modes de codage par la méthode de compensation temporelle et spectrale, on vise également la correction des erreurs de transmission et l'optimisation des débits.

- **Organisation de la mémoire**

Notre manuscrit est scindé en trois chapitres :

-Dans le premier chapitre, nous présentons un état de l'art en décrivant en premier lieu le but de la compression, les différents types de redondance et les méthodes de son exploitation pour compresser les images. En deuxième lieu, nous présentons le principe du codage vidéo des images couleurs par la compensation temporelle et spectrale. Pour la compensation temporelle nous expliquons en l'estimation du mouvement et la prédiction temporelle. Pour la compensation spectrale nous évoquons quelques méthodes intéressantes concernant le codage des images multispectrales fixes et les séquences vidéo. Nous terminons le chapitre par la description des critères de qualité.

-Dans le deuxième chapitre, nous présentons le codeur développé pour le codage vidéo des images couleurs modélisées dans les deux espaces RGB et YUV, en donnant les deux versions du codeur et du décodeur. Dans ce chapitre également nous expliquons en détail notre méthode de codage des images vidéo multispectrales appliquée aux images RGB.

-Dans le troisième chapitre nous procédons à l'étude et à la comparaison des résultats de simulation obtenus, après le codage d'une séquences formées d'images YUV par compensation temporelle, et le codage de la même séquence mais formée d'images RGB par compensation spectrale.

Table des matières

Pages

Remerciements	I
Résumé	II
Abstract	III
Introduction générale	IV

Chapitre1

1.1 Introduction	1
1.2 Espaces de représentation des images	2
1.3 Modèles RGB et YUV des images couleurs	2
1.4 Sous échantillonnage des signaux de chrominance U et V	3
1.5 Formats usuels des images	4
1.6 Nécessité de la compression	5
1.7 Notion de redondance	6
1.7.1 La redondance spatiale	7
1.7.2 La redondance temporelle	7
1.7.3 La redondance spectrale	7
1.7.4 La redondance psychovisuelle	8
1.7.5 La redondance statistique	8
1.8 Codage vidéo par compensation temporelle	9
1.8.1 Estimation du mouvement	10
1.8.2 Prédiction par compensation de mouvement	13
1.8.3 Codage par transformation	16
1.9 Codage des images multispectrales	21
1.9.1 Méthodes de prédiction	22
1.9.2 Méthodes de transformations	23
1.9.3 Méthodes de la quantification vectorielle	24
1.10 Codage vidéo des images multispectrales	25
1.11 Critères de performance	28
1.12 Conclusion	30

Chapitre 2

2.1 introduction	32
2.2 Approche au codage vidéo des images YUV et RGB	32
2.3 Présentation du codeur version1	35
3.3.1 Fonctionnement du codeur version 1 en mode Intra	36
3.3.2 Fonctionnement du codeur version 1 en mode MC	37
2.3.3 Fonctionnement du codeur version 1 en mode SC	38
2.4 Principe du décodage	45
2.4.1 Décodeur version1	45
2.4.2 Fonctionnement du décodeur version 1 en mode Intra	46
2.4.3 Fonctionnement du décodeur version 1 en mode MC	46
2.4.4 Fonctionnement du décodeur version 1 en mode SC	47
2.5 Présentation du codeur version2	48
2.5.1 fonctionnement du codeur version 2 en mode SC*	49
2.6 Décodeur version2	50
2.6.1 Fonctionnement du décodeur version2 en mode SC*	52
2.7 Conclusion	53

Chapitre 3

3.1 Introduction	55
3.2 considérations préliminaires	55
3.3 Codage de la séquence 'CarPhone' YUV	56
3.3.1 Mode de codage YUV(MC)	57
3.4 Codage de la séquence 'CarPhone' RGB	62
3.4.1 Mode de codage RGB(SC)	62
3.4.2 Mode de codage RGB(SC)*	67
3.5 Comparaison entre les modes de codage	69
3.6 Conclusion	71
3.7 Conclusion générale	72

CHAPITRE 1

ETAT DE L'ART

1.1 Introduction

La compression ou codage de l'image, est un processus qui consiste à éliminer certaines informations dites redondantes de l'image, et de simplifier d'autres, en faisant en sorte que les modifications apportées échappent le plus possible à la perception visuelle humaine. Ce processus de codage devient d'autant plus délicat, que débit recherché est réduit. Dans le cas des images fixes, les différentes techniques de compression développées s'appuient sur une analyse du contenu de l'image, en tirant profit de sa structure interne afin de supprimer la redondance qui peut être déduite des informations restantes. Par exemple, une image comporte des plages uniformes comportant un nombre de pixels identiques, dans ce cas, il suffit de retenir uniquement la valeur du niveaux de gris des pixels et leurs nombre, ce qui permet de coder les plages uniformes d'une manière compacte. Tandis que, dans le cas d'une séquence vidéo des images YUV, étant donné que la vitesse du mouvement est souvent plus petite que la fréquence de répétition des trames, alors, la compression peut exploiter le fait qu'il existe une similitude entre deux images successives. Donc, on peut considérer que deux images consécutives peuvent être identiques, à la différence d'un mouvement près facilement estimable. En fait, la décorrélation temporelle par compensation du mouvement entre la trame courante et la trame précédente dans le cas des images YUV [1], a permis d'obtenir des taux de compression considérables. En plus, comme le pouvoir perceptuel humain est plus sensible aux détails portés par le signal de luminance Y qu'aux signaux de chrominance U et V, alors, ces derniers signaux peuvent être sous échantillonnés, ce qui apporte un gain en espace et augmente la compression. Dans le cas des séquences vidéo des images RGB, en plus de la redondance temporelle entre les trames, il est nécessaire de tenir compte de la redondance spectrale entre les différentes bandes constituant les trames. Alors, il faut prévoir des méthodes de codage qui mettent en œuvre la décorrélation spectrale. Dans le cas des images multispectrales fixes, la décorrélation spectrale est effectuée par plusieurs méthodes proposées dans [2], [3], [4]. Mais pour la compression des séquences vidéo des images multispectrales, les travaux sont encore récents et les publications sont encore rares. En fait, nous avons reçu à obtenir la publication [5], qui cite une méthode statistique basée sur la matrice de covariance pour la décorrélation temporelle, et la méthode JPEG 2000, pour la décorrélation spatiale. Nous introduisons une méthode de codage basée sur la compensation spectrale inter bandes. Dans cette méthode, on emploie la redondance spectrale qui se trouve dans un décalage quasi uniforme, entre les pixels qui se trouvent dans la même position à travers la dimension spectrale.

1.2 Espaces de représentation des images

Dans l'espace 2D, l'image est décrite par une fonction $I(x, y)$ qui indique la valeur du niveau du gris du pixel, qui se trouve dans la position x, y du repère $(0, \overset{\pm}{x}, \overset{\pm}{y})$. Dans le cas de l'espace 3D, on associe une troisième dimension qui peut être spatiale z , temporelle t ou spectrale. Alors une image 3D volumique sera décrite par la fonction $I(x, y, z)$, une séquence vidéo sera décrite par la fonction $I(x, y, t)$ et une image multi spectrale sera décrite par la fonction $I(x, y, \lambda)$. La figure 1.1 montre une illustration des différents espaces de représentation de l'image.

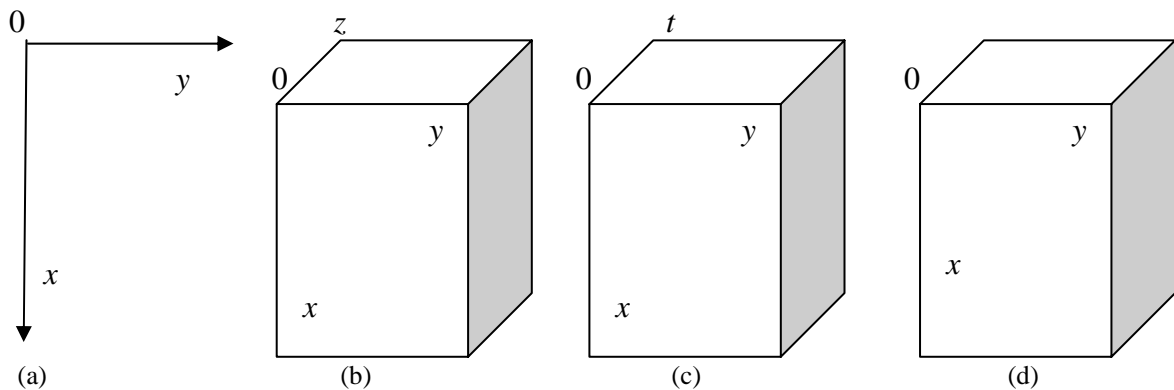


Figure 1.1 Espaces de représentation des images

(a) espace 2D, (b) Image 3D volumique (c) séquence vidéo, (d) image multi spectrale

1.3 Modèles RGB et YUV des images couleurs

Une image en couleur est obtenue après la capture d'une scène en mesurant la lumière réfléchie avec des modalités différentes. Ces modalités consistent à prendre pour chaque bande de fréquence la mesure de l'intensité lumineuse réfléchie ayant traversé un filtre correspondant. Le modèle le plus familier de représentation d'une image par trois bandes spectrale est l'espace des couleurs primaires RGB (Red, Green, Blue). La compression par décorrélation temporelle des images modélisées RGB est insuffisante, à cause de l'existence d'une forte corrélation spectrale entre les bandes. En plus, les trois bandes RGB portent des détails significatifs pour l'œil humain, ce qui interdit le sous échantillonnage. Alors, on a réussi à résoudre le problème par l'adoption d'un autre modèle de représentation défini par l'espace YUV. La représentation de l'image dans cet espace, permet la séparation du signal de luminance Y des signaux de chrominance U et V. L'avantage de cette séparation des signaux,

est la décorrélation spectrale d'une part, et la possibilité du sous échantillonnage des signaux de chrominance U et V d'autre part. La conversion typique de l'espace RGB à l'espace YUV adoptée par la CCIR (International Radio Consultative Committee) Recommandation 601-1, utilisée en JPEG, est donnée par l'expression mathématique suivante [6] :

$$\begin{array}{rcccc}
 Y & 0.299000 & 0.587000 & 0.114000 & R \\
 U & = -0.168736 & -0.331264 & 0.500002 & G \\
 V & 0.500000 & -0.418688 & -0.081312 & B
 \end{array} \quad (1.1)$$

En fait, la transformation précédente contient des valeurs négatives pour les composantes chroma U et V, tandis que la luminance Y est toujours positive. Dans le but de représenter les signaux chroma U et V dans l'ensemble des nombres entiers non signés à 8bits, on effectue une opération de décalage par 128 en prenant les valeurs arrondies des niveaux du gris des signaux U et V. L'expression mathématique utilisée est la suivante :

$$\begin{array}{rcccc}
 Y & 0.29900 & 0.58700 & 0.11400 & R & 0 \\
 U & = -0.16874 & -0.33126 & 0.50000 & G & + 128 \\
 V & 0.50000 & -0.41869 & -0.08131 & B & 128
 \end{array} \quad (1.2)$$

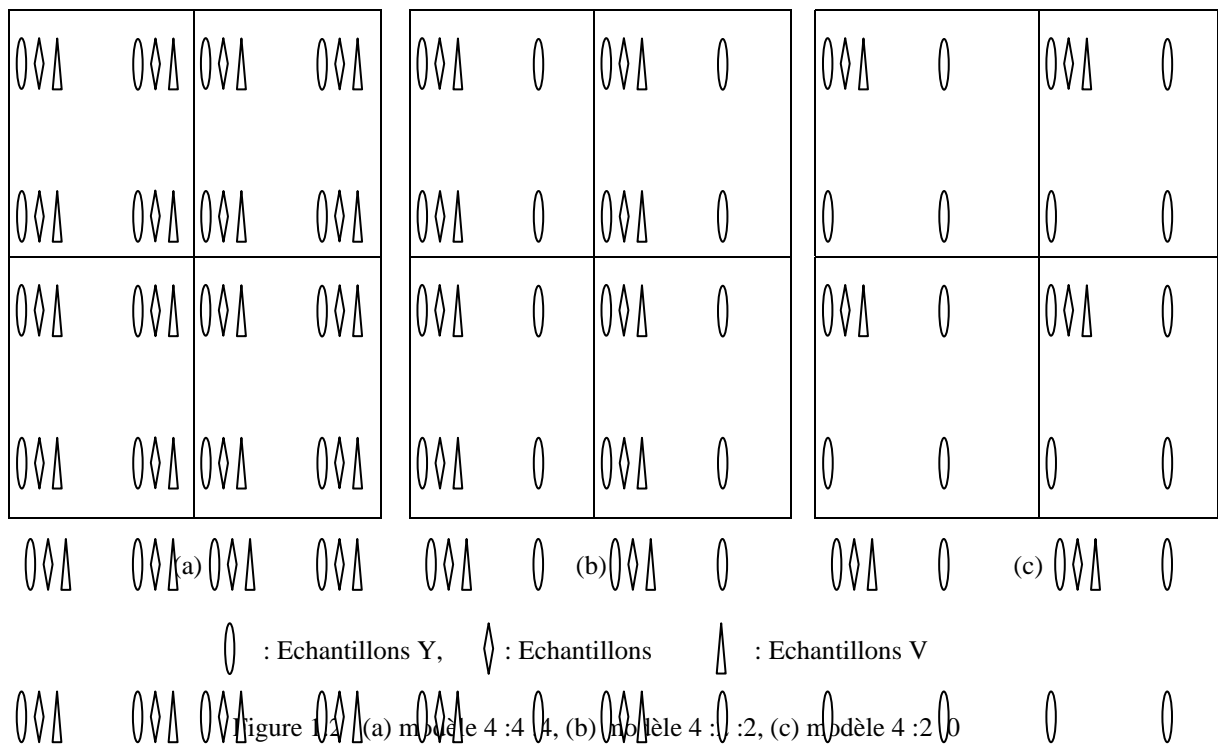
La transformation inverse qui produit la représentation des composantes R, G et B dans l'ensemble des nombres non signés à 8 bits, est donnée par l'expression suivante :

$$\begin{array}{rcccc}
 R & 1.00000 & 0.00000 & 1.40210 & Y & 0 \\
 G & = 1.00000 & -0.34414 & -0.71414 & U & - 128 \\
 B & 1.00000 & 1.77180 & 0.00000 & V & 128
 \end{array} \quad (1.3)$$

1.4 Sous échantillonnage des signaux de chrominance U et V

Il est connu que les détails les plus significatifs, pour l'œil humain, d'une image modélisée dans l'espace YUV sont contenus dans le signal de luminance Y, tandis que les détails les moins significatifs sont contenus dans les signaux de chrominance U et V. Alors, dans le but d'augmenter la compression par une diminution du nombre d'échantillons on soumet les signaux de chrominance U et V, auxquels l'œil est moins sensible à un sous échantillonnage symbolisé par le modèle Y : U : V. On signale que ce modèle est choisi en fonction de l'application visée. Par exemple, on utilise dans la HDTV (High Definition TV)

le modèle 4:4:4 qui signifie : pour 4 pixels Y il y'a 4 pixels U et 4 pixels V. Dans la production en couleur de haute qualité on utilise le modèle 4:2:2 qui signifie : pour 4 pixels Y il y'a 2 pixels U et 2 pixels V. Le modèle qui permet une compression très poussée le plus utilisé actuellement dans la télévision digitale, et le stockage DVD (Digital Versatile Disk) est le 4:2:0 qui signifie : pour 4 pixels Y il y'a 1 pixel U et 1 pixel V. La figure 1.2 suivante montre une illustration de trois modes d'échantillonnage des signaux de chrominance U et V.



1.5 Formats usuels des images

Actuellement, il existe une variété de formats adoptés en compatibilité de l'application visée. A titre d'exemple, les trois types de formats usuels, utilisant le sous échantillonnage 4:2:0 sont :

- Le format CIF (Common Intermediate Format) de 30 trames par seconde, utilisé en vidéoconférences.
- Le format QCIF (Quarter CIF) sa résolution est la moitié de celle du format CIF, elle est de 15 trames par seconde et utilisée en vidéo téléphonie.
- le format du groupe ITU-R 601 de 30 trames par seconde, utilisée pour les émissions numériques TV de grand public. Le tableau 1.1 représente le tableau des formats usuels cités et leur utilisation.

Format	Luminance resolution	Chrominance resolution	Frames per second	Bits per second (uncompressed)
IUT-R601	858x525	429x525	30	216 Mbps
CIF	352x288	176x144	30	36.5 Mbps
QCIF	176x144	88x72	15	4.6 Mbps

Tableau 1.1 Quelques formats usuels et leurs caractéristiques

1.6 Nécessité de la compression

On signale que la numérisation d'une image 2-D, produit une quantité énorme de données après les opérations d'échantillonnage et de quantification des niveaux de gris. En tenant compte des contraintes de limitation de la capacité de stockage et des canaux de communication disponibles, il est impératif de ramener le débit nécessaire à une valeur compatible avec ces limitations. A titre d'exemple, une image monochrome de qualité standard, à 256 niveaux de gris, de résolution 512x512, après avoir été numérisée et codée sur 8 bits, requiert une mémoire de $(512 \times 512 \times 8 = 2097152 \text{ bits})$, si la cadence d'émission est de 25 images par seconde alors sa transmission requiert un débit de $(512 \times 512 \times 8 \times 25 = 52,428 \text{ Mbps})$ et nécessite un canal de transmission de largeur $(52,428 / 2 = 26 \text{ Mhz})$. Dans les missions spatiales lointaines, la faible puissance des émetteurs embarqués limite généralement le débit d'information à 10 bits/s. Dans ces conditions, la transmission d'une image durerait $(512 \times 512 \times 8 / 10 \times 3600 = 55,55 \text{ heures})$ [7].

Dans le cas de l'enregistrement magnétique sur magnétoscope numérique, le débit maximal est limité à 20 Mbps. Dans le cas d'une image multispectrale captée par un camera à six bandes de résolution 1920x1080, lorsqu'elle est codée sur 16 bits, requiert un espace mémoire de $1920 \times 1080 \times 6 \times 16 = 23.7 \text{ mega bytes}$. En plus, si la camera utilisée possède une cadence de capture de 30 trames par seconde alors, la taille nécessaire pour une seconde sera d'environ 712 mega bytes [5]. Alors, toutes ces considérations imposent la nécessité de la compression des images pour optimiser l'exploitation des supports de stockage et de transmission

1.7 Notion de redondance

On peut définir la redondance comme l'ensemble des données qui se répètent dans l'image et que l'on peut soustraire sans porter atteinte à sa qualité. Les données résiduelles sont pertinentes et garantissent une qualité acceptable de l'image lors de sa reconstruction. La redondance des données est un élément essentiel dans le domaine de la compression de l'image. En plus, c'est une entité qui est mathématiquement quantifiable. Si on note par n_1, n_2 , les nombres d'informations transportées dans deux ensembles de données [7], [8], on définit la donnée redondante relative R_D et le taux de compression par les relation suivantes :

$$R_D = 1 - 1/C_R \quad (1.4)$$

$$C_R = n_1/n_2 \quad (1.5)$$

-Si $n_1 = n_2$, on aura $C_R = 1$ et $R_D = 0$, cela signifie absence de redondance.

-Si $n_1 \gg n_2$, on aura $C_R \rightarrow \infty$ et $R_D \rightarrow 1$, cela signifie que le premier ensemble (n_1) contient plus de données redondantes que le second (n_2).

-Si $n_1 \ll n_2$, on aura $C_R \rightarrow 0$ et $R_D \rightarrow -$, cela signifie que le premier ensemble (n_1) contient moins de données redondantes que le second (n_2).

Par exemple, on dit que le taux de compression est de (10 :1), cela veut dire que le premier ensemble (n_1) transporte 10 unités informationnelles avant compression, réduite à 1 unité informationnelle transportée par le deuxième ensemble (n_2) après compression. Dans le cas d'une image fixe, on peut supprimer jusqu'à 70% des informations sans toucher à sa qualité. Alors, on dit que le taux de compression est de (3 :1), si on tolère quelques pertes non perceptibles, on peut atteindre (10 :1). Dans le cas des séquences vidéo on peut atteindre des rapports de compression nettement supérieurs de (15 :1) à (40 :1).

1.7.1 La redondance spatiale

Dans une image il existe généralement des zones homogènes, c'est-à-dire, présence d'un nombre de pixels ayant la même valeur de niveaux de gris. On constate qu'il est inutile de coder chaque pixel séparément, car, il suffit de coder uniquement le nombre et la valeur. Comme on peut aussi effectuer des opérations de prédiction et d'interpolation des uns à partir des autres. La redondance spatiale à l'intérieur de l'image peut être mise en évidence par une technique de transformations dans le domaine des fréquences, tel que : DCT (Discrete Cosine Transform).

1.7.2 La redondance temporelle

En vidéo il existe une forte corrélation entre deux images voisines à cause du mouvement. On peut considérer que deux images voisines sont identiques à la différence d'un mouvement près. On note que la technique mise en jeu pour coder une image vidéo, est la compensation temporelle entre deux images voisines après l'estimation du mouvement. Dans ce cas il suffit de transmettre uniquement le vecteur mouvement et l'erreur de prédiction. On note que la décorrélation temporelle conduit à des taux de compression élevés.

1.7.3 La redondance spectrale

La redondance spectrale est due à la corrélation entre les pixels qui se trouvent dans la même position dans les différentes bandes constituant l'image. Cette redondance spectrale peut être supprimé soit par exploitation du décalage entre les valeurs des pixels dans la même position dans les bandes selon la dimension spectrale, soit par transformation de l'image de l'espace primaire RGB à l'espace secondaire YUV.

La figure 1.3 montre les trois bandes RGB d'une image de la séquence 'foramen'. On voit que les trois bandes sont d'égale importance car, elles contiennent des détails significatifs pour l'œil humain, ce qui prouve une grande corrélation spectrale entre elles. La figure 1.4 montre les trois composantes YUV de l'image 'foreman'. On voit que la composante de luminance Y contient tous les détails significatifs alors que, les composantes de chrominance U et V portent des détails moins significatifs. Ceci montre que la transformation de l'espace RGB à l'espace YUV provoque une décorrélation spectrale.

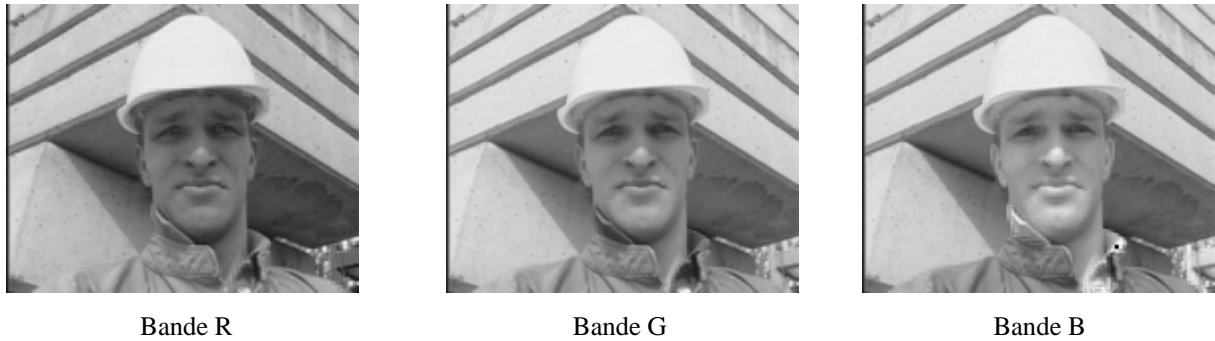


Figure 1.3 Les trois bandes RGB d'une image de la séquence 'Foreman'.

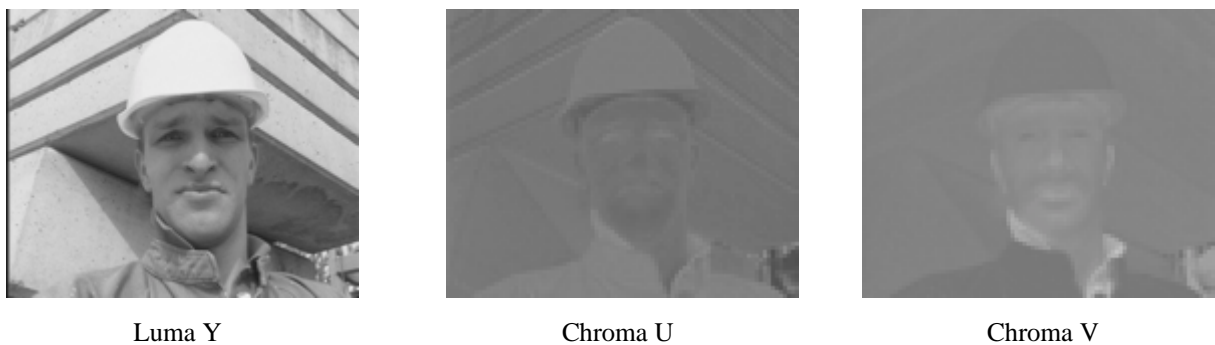


Figure 1.4 Les trois composantes YUV d'une image de la séquence 'Foreman'.

1.7.4 La redondance psychovisuelle

Appelée aussi redondance subjective, elle fait appel au concept de codage perceptuel en tirant profit de la faiblesse du système visuel humain. En fait, elle consiste à coder les détails jugés pertinents et significatifs pour notre système visuel avec un nombre de bits limité, tout en rejetant les détails non perceptibles pour un observateur non averti. Dans ce cas on procède à une pondération psychovisuelle qui consiste en une opération de quantification non linéaire, qui introduit des pertes de résolution irréversibles. L'élimination de la redondance psychovisuelle s'accomplit avec la condition de maintenir un compromis entre le désir de préserver une qualité convenable et la nécessité de réduire suffisamment le débit au regard de la capacité du support de l'information.

1.7.5 La redondance statistique

Cette redondance est définie par le caractère statistique d'émission des symboles par une source. Alors, dans un alphabet il existe des symboles dont la fréquence d'apparition est

grande, il convient de leur attribuer des mots codes plus courts. Tandis que, les symboles qui se répètent rarement on leur associe des mots codes plus longs. Cette opération de codage à caractère statistique est aussi appelée codage entropique

1.8 Codage vidéo par compensation temporelle

Le processus de la compression vidéo consiste à représenter d'une façon plus compacte une séquence d'images par un nombre réduit de bits, après élimination des différents types de redondances. Il est clair que, les différentes méthodes utilisées pour la suppression de la redondance déterminent les différents types de codage.

En fait, Le codage par compensation temporelle consiste à effectuer une compensation de mouvement MC (Motion compensated) entre deux images consécutives. Ce mode de codage est très efficace à cause de la décorrélation dans la dimension temporelle. Ce type de codage est dit aussi Hybride car, il combine la prédiction par compensation temporelle de l'image courante en utilisant le vecteur calculé par estimation du mouvement, et le codage spatial par DCT de l'erreur de prédiction.

Pour la compression et la décompression des images vidéo on fait appel à une paire complémentaire dite CODEC (Coder / Decoder). La tâche du codeur est la compression des données vidéo, tandis que le décodeur décompresse les données reçues pour reproduire les données vidéo d'origine.

Algorithme appliqué

L'algorithme appliqué dans le codage par compensation temporelle est le suivant :

- 1- Estimation de mouvement par calcul du MSE (Mean Square Error) entre le bloc courant et l'ensemble des blocs de la région de recherche dans l'image de référence.
- 2- Sélection du bloc qui donne la plus petite erreur.
- 3- Soustraction du bloc sélectionné du bloc courant pour avoir le bloc différence de déplacement, DFD (Displaced Frame Difference).
- 4- Codage et transmission du DFD.
- 5- Codage et transmission du vecteur mouvement MV.

1.8.1 Estimation du mouvement

L'estimation du mouvement ME (Motion estimation) consiste à déterminer le vecteur mouvement MV (Motion Vector), en se basant sur le principe de la correspondance des blocs BMA (Block Matching). On signale que les blocs à correspondre appartiennent à deux trames successives de la séquence vidéo. Le principe de l'estimation du déplacement par la méthode de correspondance de blocs repose sur les hypothèses suivantes [9] :

- Les objets sont indéformables et se translatent perpendiculairement à l'axe optique de la caméra.
- Les objets dans l'image courante $I(t)$, sont généralement des rectangles appelés blocs et ne correspondent pas à des objets réels.
- Le centre d'un bloc dans l'image courante $I(t)$, est noté $P_{i,j}$.

La méthode consiste donc à rechercher dans l'image précédente (de référence) $I(t-1)$, le bloc correspondant en explorant une zone de recherche dans toutes les directions. Par exemple, La position pour laquelle l'erreur quadratique moyenne MSE est minimale détermine le vecteur mouvement. La figure 1.5 montre les positions d'un bloc avant et après le déplacement.

Si (i_1, j_1) et (i_2, j_2) désignent respectivement les coordonnées du centre du macrobloc dans l'image de référence $I(t-1)$ et l'image courante $I(t)$, alors on définit les composantes du vecteur mouvement ($\vec{D} = \vec{dx} + \vec{dy}$) du point P_{i_1, j_1} dans l'image $I(t-1)$ au point P_{i_2, j_2} dans l'image $I(t)$ par :

$$dx = i_2 - i_1 \quad (1.7)$$

$$dy = j_2 - j_1$$

Soient : $Xm = |dx|_{\max} = |dy|_{\max}$ et M, N les dimensions du macro bloc, alors les dimensions de la zone de recherche sont :

$$M+2 Xm \text{ et } N+2 Xm . \quad (1.8)$$

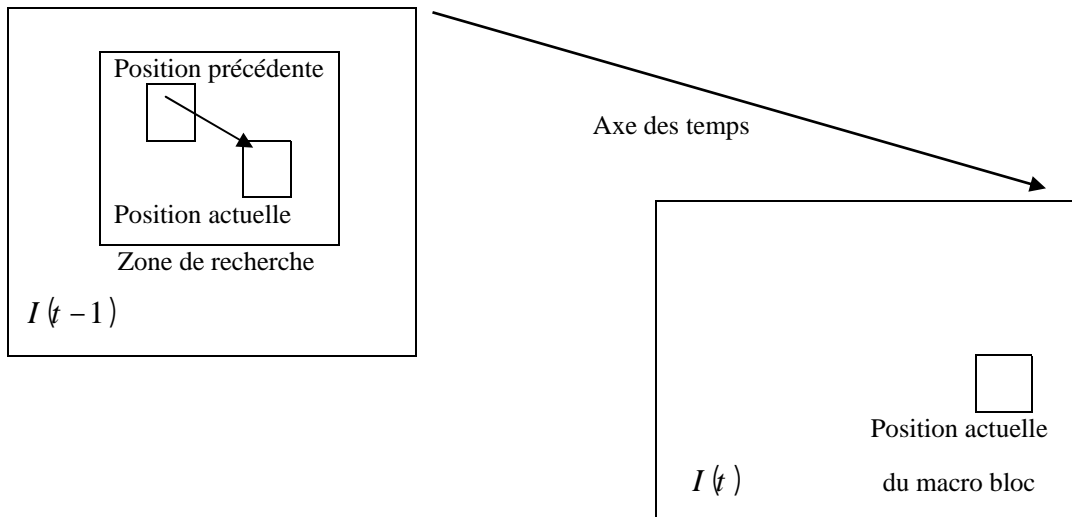


Figure 1.5 Estimation de mouvement par le principe de la correspondance des blocs (BM).

Exemple

L'estimation du mouvement est un processus qui consiste à déterminer le vecteur de déplacement du centre du bloc courant, se déplaçant dans une zone de recherche. On signale que le bloc courant se trouve dans l'image courante F_n , tandis que, la zone de recherche de dimension plus grande se trouve dans l'image de référence F_{n-1} . La figure 1.6 (a) montre le bloc courant de 3x3 pixels à correspondre. Dans la figure 1.6 (b) se trouvent représentés : La zone de recherche qui est un tableau de 5x5 pixels, le bloc courant en pointillés gras occupant la position centrale (0,0), le bloc courant en trait gras en position finale (1,1), la flèche en trait gras indique le vecteur mouvement (MV) du bloc courant qui s'est déplacé de la position (0,0) en position (1,1) dans la zone de recherche, et enfin la flèche en pointillés fins indique la hiérarchie du balayage en spirale. Dans la figure 1.6 (c) se trouvent les coordonnées des différentes positions du centre du bloc courant à correspondre dans la zone de recherche.

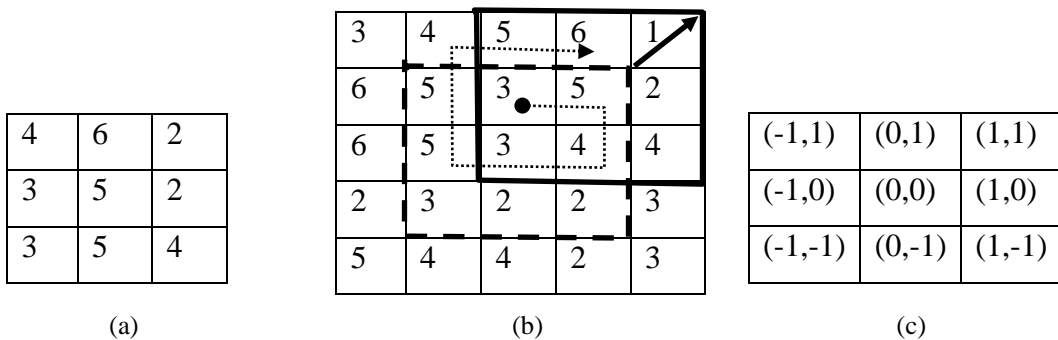


Figure 1.6 Illustration du principe du BMA

Le calcul de l'erreur quadratique moyenne (MSE) est effectué pour chaque position du bloc pendant le balayage en spirale ne dépassant pas ± 1 pixel. Les des valeurs du MSE calculées pour chaque position dans la zone de recherche sont données dans le tableau 1.2. D'après les calculs du MSE on trouve la valeur minimale 0.33 pour la position (1,1) dans l'image de référence, c'est-à-dire que le bloc courant qui était en position (0,0) dans l'image courante s'est déplacé de $dx = 1$ et $dy = 1$ pour se retrouver en position (1,1) dans l'image de référence. Donc on parvient à l'estimation de mouvement dont le vecteur de déplacement est : $D = [1,1]$. Comme le décodeur à besoin de ce vecteur pour reconstruire l'image courante, donc il faut prévoir son codage et sa transmission.

Positions (x, y)	(0,0)	(1,0)	(1,-1)	(0,-1)	(-1,-1)	(-1,0)	(-1,1)	(0,1)	(1,1)
MSE	4.88	2	3.44	3.22	1.77	2.55	3.66	4.66	0.33

Tableau 1.2 les valeurs du MSE selon les différentes positions

Codage VLC (Variable Length Coding) du vecteur mouvement

Pour le codage du vecteur mouvement on utilise le codage VLC dit (Exponential Golomb Code) qui est un codage entropique [1], où chaque mot code est structuré comme suit :

[M zeros][1][INFO]

Avec :

$M = \log_2(\text{code_num} + 1)$ désigne le nombre des zéros qui précèdent l'information, et où l'index code_num est donné dans les tables de consultation.

[1] Niveau logique 1 qui suit les M zéros.

$\text{INFO} = \text{code_num} + 1 - 2^M$ désigne M-bits véhiculant l'information.

Le mot est décodé comme suit :

1-Lecture des M premiers 0 suivis par 1,

2-Lire le champ M-bits INFO.

3- $\text{code_num} = 2^M + \text{INFO} - 1$

Exemple

Soit à coder le vecteur mouvement $D = [1,1]^T$.

D'après le tableau 1.3 (b) des deux composantes du vecteur D ont pour valeur (value) égale à 1, le *code_num* correspondant est aussi égal à 1, alors, on aura M=1 bit de valeur logique 0 suivi par la valeur logique 1, et INFO=1bit de valeur logique 0. Le tableaux 1.3 (a) donne le mot code (codeword) qui correspond au *code_num*=1, et enfin D (1.1) aura pour mot code (010,010).

<i>code_num</i>	codeword
0	1
1	010
2	011
3	00100
4	00101
...	...

(a)

Value	<i>code_num</i>
0	0
1	1
-1	2
2	3
-2	4
...	...

(b)

Tableaux 1.3 de consultation (Exponential Golomb Code)

1.8.2 Prédiction par compensation de mouvement

La notion de prédiction par compensation du mouvement MCP (Motion-Compensated prediction) constitue la base de la prédiction temporelle, qui consiste à prédire le bloc courant à partir du bloc correspondant dans l'image précédente en se servant du vecteur mouvement. C'est-à-dire, le bloc dans l'image courante à coder est équivalent au translaté par \vec{D} du bloc correspondant dans l'image précédente de référence. Si on désigne par :

$(x, y, t - 1)$, le bloc de référence dans l'image de référence $I(t - 1)$

(x, y, t) , le bloc courant dans l'image courante $I(t)$.

$_p(x, y, t)$, le bloc courant prédit, alors :

-Dans le cas des images non stationnaires (en mouvement), la prédiction temporelle est décrite par la relation suivante :

$$(x, y, t) = (x - dx, y - dy, t - 1) \quad (1.9)$$

Et l'erreur de prédiction $e(x, y, t)$, appelée DFD (Displaced Frame Difference) sera :

$$e(x, y, t) = (x, y, t) - {}_p(x, y, t) = (x, y, t) - (x - dx, y - dy, t - 1) \quad (1.10)$$

-Dans le cas des images stationnaires (fixes), on a $d_x = d_y = 0$, il en résulte :

$${}_p(x, y, t) = (x, y, t - 1) \quad (1.11)$$

De même :

$$e(x, y, t) = (x, y, t) - {}_p(x, y, t) = (x, y, t) - (x, y, t - 1) \quad (1.12)$$

Une fois l'erreur de prédiction DFD (Displaced Frame Difference) obtenue, on lui applique le codage par transformée. En fait, il existe trois types d'images en rapport avec le mode et la direction de la prédiction temporelle. Ces images forment un ensemble appelé GOP (Group Of Pictures), figure 1.7. Un GOP commence toujours par une image I et se termine à la dernière image précédant la prochaine image I .

Images de type I

Ces images sont codées indépendamment des autres par prédiction intra, chaque image est décrite par elle-même, elle contient tous les éléments nécessaires à sa reconstruction. Elles sont utilisées dans les algorithmes de codage JPEG. Les images de type I sont caractérisées par un taux de compression faibles et sont assez volumineuses.

Images de type P

Ces images sont obtenues par une prédiction avant, à partir des images passées de type I ou P en appliquant la technique d'estimation et compensation du mouvement MCP. Elles sont utilisées dans les algorithmes de codage MPEG. Ces images sont typiquement moins

volumineuses qu'une image de type I , mais peuvent transmettre des erreurs, car elles servent également de référence.

Images de type B

Ces images sont obtenues par prédiction bidirectionnelle avant ou arrière, à partir des images passées ou futures de type I ou P , toujours par compensation de mouvement. Elles offrent le taux de compression le plus élevé, et elles ne propagent pas d'erreurs car, elles ne sont pas utilisées en tant que référence. Leur taille est six fois moins volumineuse qu'une image type I

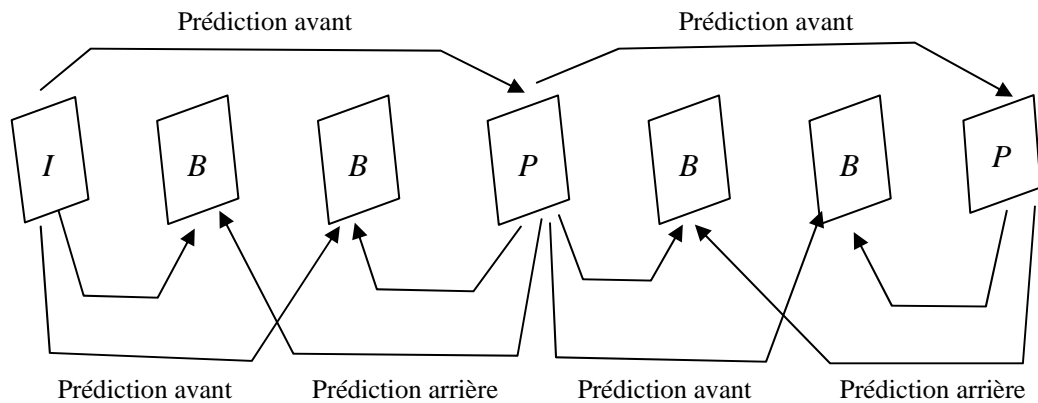


Figure 1.7 Exemple d'un GOP contenant les trois types d'images I , P et B

Exemple

Par application de l'équation (1.10), on calcule à partir du bloc courant (x, y, t) donné en figure 1.8 (a) et du bloc courant prédit (x, y, t) donnée en figure 1.8 (b), l'erreur de prédiction $e(x, y, t)$ donnée en figure 1.8 (c). On signale que dans notre codeur, la prédiction est faite à partir des images précédentes type I ou P , et que la meilleure ressemblance pour la position (1,1) donne la meilleure prédiction pour laquelle la MSE est minimale. Le codeur vidéo effectue le même processus pour tous les blocs de l'image courante. Notons que la technique de compensation de mouvement montre que l'erreur de prédiction est très faiblement corrélée, ce qui présente alors une amélioration très poussée de ce codage par rapport au codage spatial classique dans le cas de l'image fixe. Cette erreur de prédiction

maintenant passe à l'étape de transformation en *DCT* , puis, à l'étape de quantification Q suivant des tables normalisées en prenant les valeurs arrondies.

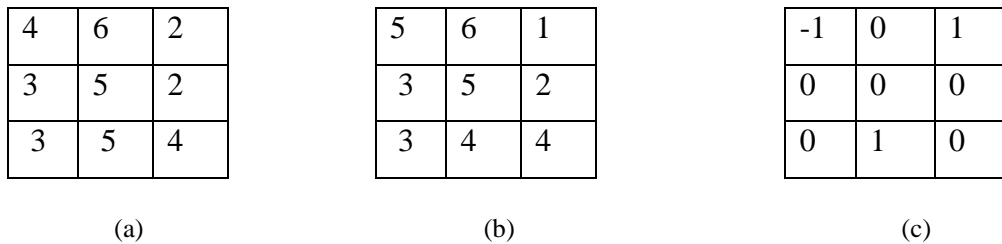


Figure 1.8 (a) Bloc courant, (b) Bloc courant prédit, (c) Erreur de prédiction

1.8.3 Codage par transformation

Le codage par transformation consiste à transformer le plan image par une opération linéaire dans un plan fréquentiel où, la séparation entre l'information pertinente et redondante est plus aisée. Enfin, on obtient des coefficients mieux décorrélés. Pour cela, il existe un nombre de transformations orthogonales tels que : *DCT* (Discrete cosine Transform) et *DWT* (Discrete Wavelet Transform). Une image contient beaucoup de plages uniformes. Le spectre de la transformation fait apparaître une sorte de concentration de l'énergie autour des fréquences basses. Comme la plus grande partie de l'énergie est concentrée dans les coefficients de basses fréquences, alors on attribue peu de bits pour coder les coefficients se trouvant dans la partie des hautes fréquences. On note que l'opération de transformation est par sa nature réversible. Mais, suivie par une opération non conservative de quantification, fait que ce type de codage rentre dans la catégorie de codage avec pertes, c'est-à-dire, irréversible. En fait ce codage est à la base de plusieurs normes reconnues de compression des images, comme *JPEG* (Joint Photographic Expert Group) pour les images fixes, et *MPEGx* (Motion Picture Expert Group) pour les images en mouvement développées par l'ISO (International Standard Organisation) , et les normes H.26*

Quantification et seuillage des coefficients

Le but de la quantification est d'améliorer davantage la compression, en représentant les coefficients *DCT* avec une précision suffisante, pour restituer la qualité recherchée de l'image. Autrement dit, le but de cette étape revient à rejeter toute information qui paraît visuellement non significative. La loi de quantification est adaptée en fonction de la courbe de la sensibilité du système visuel humain, ce qui rend l'opération non linéaire. Rappelons que la

sensibilité du système visuel humain est maximale aux basses fréquences, et décroît au fur et à mesure que la fréquence augmente. Alors, il est nécessaire d'effectuer une pondération fréquentielle, à l'aide d'une table normalisée. Cette table est établie conformément à la courbe de sensibilité visuelle dans le but de répartir les erreurs de codage sur les coefficients à hautes fréquences. Une autre approche de seuillage consiste à éliminer même les coefficients inférieurs à un seuil tolérable.

La quantification est aussi adaptable au contexte local du signal. En effet, en général une image comporte différentes zones d'activité, il est nécessaire d'utiliser des quantificateurs spécifiques pour chaque zone différente. On note, qu'il existe des tables normalisées de quantification de luminance et de chrominance. A titre d'exemple, le standard H264 (AVC) utilise une quantification scalaire dont l'implémentation est compliqué à cause des critères requis sur l'arithmétique ainsi que sur les opérations du seuillage (post-scaling et pre-scaling). L'opération directe de quantification est régie par la relation générale suivante [1] :

$$Z_{ij} = Round \frac{Y_{ij}}{Q_{step}} \quad (1.13)$$

Où Y_{ij} c'est le coefficient transformé, Z_{ij} c'est la valeur quantifiée et Q_{step} c'est le pas de quantification. Après le seuillage du coefficient Y_{ij} à l'aide d'un facteur multiplicatif PF on obtient :

$$Z_{ij} = Round W_{ij} \frac{PF}{Q_{step}} \quad (1.14)$$

Généralement, le pas de quantification est faible pour les basses fréquences ce qui permet de conserver les détails de chrominance qui sont généralement portés par les basses fréquences. Par contre, le pas devient considérable pour se débarrasser des détails inutiles de hautes fréquences. La figure 1.9 illustre les opérations de seuillage et de quantification pour un pas ($Q_{step} = 2$), en (a) les échantillons de départ, en (b) les coefficients DCT, en (c) le bloc résultant après les opérations de seuillage et quantification.

5	11	8	10
9	8	4	12
1	10	11	4
19	6	15	7

(a)

140	-1	-6	7
-19	-39	7	-92
22	17	8	31
-27	-32	-59	-21

(b)

17	0	-1	0
-1	-2	0	-5
3	1	1	2
-2	-1	-5	-1

(c)

Figure 1.9 Exemple de transformation et quantification des échantillons

Balayage des coefficients

Après l'opération de quantification, on s'aperçoit que les coefficients significatifs sur la matrice transformée sont concentrés dans la partie des basses fréquences (partie haute-gauche). Tandis que les coefficients situés dans la partie des hautes fréquences (partie basse-droite) sont généralement non significatifs et nuls. De plus on remarque une répétition des coefficients, dont la plupart sont des zéros. Alors il s'avère plus convenable d'effectuer un balayage qui permet un réarrangement des coefficients selon un ordre répondant aux exigences du codage par plage. Par exemple, dans le standard H264 AVC on adopte le balayage en zigzag montré dans la figure 1.10. Le réarrangement des coefficients après le balayage en zigzag, engendre la suite 0, 3, 0, 1, -1, -1, 0, 1, 0...

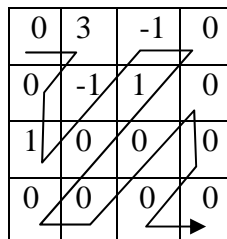


Figure 1.10 Balayage en zigzag des coefficients quantifiés

Codage CAVLC des coefficients

Le codage CAVLC (Context-based adaptive variable length coding) des coefficients, après réarrangement par le balayage en Zigzag, est basé sur les constatations suivantes [1] :

- 1) La plupart des coefficients obtenus après quantification sont nuls, donc il est convenable de compacter la représentation en exploitant le codage par plage.
- 2) On remarque que les coefficients correspondants aux hautes fréquences sont une traînée de +/-1, on les note par ('Trailing 1s' ou 'T1s').
- 3) Les coefficients non nuls des blocs adjacents ne sont pas corrélés, et par conséquent il est possible de construire des tables de consultation (Look-up table) pour les coder.

4) La valeur des coefficients (level) est élevée au début du balayage, proche du DC, et tend à diminuer vers la fin. Il est avantageux de choisir d'une manière adaptative les tables de consultation, selon le tableau 1.4.

N	Tables pour (ceff_token)
0,1	Num-VLCO
2,3	Num-VLC1
4,5,6,7	Num-VLC2
8 ou plus	FLC (Fixed Length Code)

Tableau 1.4 Sélection des tables de codage VLC

Le procédé du codage CAVLC d'un bloc de coefficients se résume comme suit :

1) Codage du nombre des coefficients non nuls et le nombre des +/-1 successifs, désigné par (ceff_token). Ceci se fait par consultation d'une table à choisir selon un paramètre N, qui dépend du nombre des coefficients non nuls dans les deux blocs qui se trouvent en haut et à gauche du bloc courant à coder.

2) Codage du signe des coefficients +/-1 consécutifs (T1). Pour chaque T1 signalé par coeff_token on utilise un bit (0 = + et 1 = -). On note que ces T1 sont codés dans l'ordre inverse en commençant par les hautes fréquences.

3) Codage des autres coefficients non nuls (Level).

4) Le codage se fait toujours dans l'ordre inverse, en commençant par la haute fréquence vers le coefficient DC. Pour le codage des coefficients (Level), avec leurs signes inclus, on fait un choix adaptatif entre 7 tables VLC de consultation. Ce choix dépend de la magnitude de chaque succession des coefficients (Level) codés.

5) le codage du nombre de zéros avant le dernier coefficient non nul (TotalZeros). On utilise aussi le codage VLC. Les zéros au départ au début du balayage seront ignorés.

6) Codage du nombre des zéros avant chaque coefficients non nul (run_before). Là aussi, on utilise le codage VLC. Le codage se fait toujours dans l'ordre inverse en commençant par les hautes fréquences. On ne code le (run_before) dans le cas (a) il n'y a pas de zéros à coder à gauche et (b) pour le dernier coefficient non nul.

Exemple

Soit à coder le bloc donné en figure 1.15, dont le balayage en zigzag engendre la suite 0,3,0,1,-1,-1,0,1,0...,on aura: TotalCoeffs=5(indexés de la haute fréquence (7) à la basse fréquence(0)), TotalZeros=3, T1s=3 (En fait, il y'a 4 (un) successives dont seulement 3 peuvent être codés comme un cas spécial).

Le codage complet est donné au tableau 1.5.

Le flot binaire (bitstream) transmis pour ce bloc est : 000010001110010111101101.

Element	Value	Code
Coeff_token	TotalCoeffs=5, T1s=3	0000100
T1 sign (4)	-	0
T1 sign (3)	-	1
T1 sign (2)	-	1
Level (1)	+1 (use Level_VLC0)	1
Level (0)	+3 (use Level_VLC1)	0010
TotalZeros	3	111
Run_before (4)	Zerosleft=3, run_before=1	10
Run_before (3)	Zerosleft=2, run_before=0	1
Run_before (2)	Zerosleft=2, run_before=0	1
Run_before (1)	Zerosleft=2, run_before=1	01
Run_before (0)	Zerosleft=1, run_before=1	No code required, last coefficient.

Tableau 1.5 Codage CAVLC d'un bloc de 4x4

Exemple du décodage

A la réception le décodeur doit effectuer les opérations inverses du codage CAVLC décrit précédemment pour reconstruire les données d'origine. Alors, le décodage du flot binaire (bitstream) 000010001110010111101101 transmis est donné au tableau 1.6.

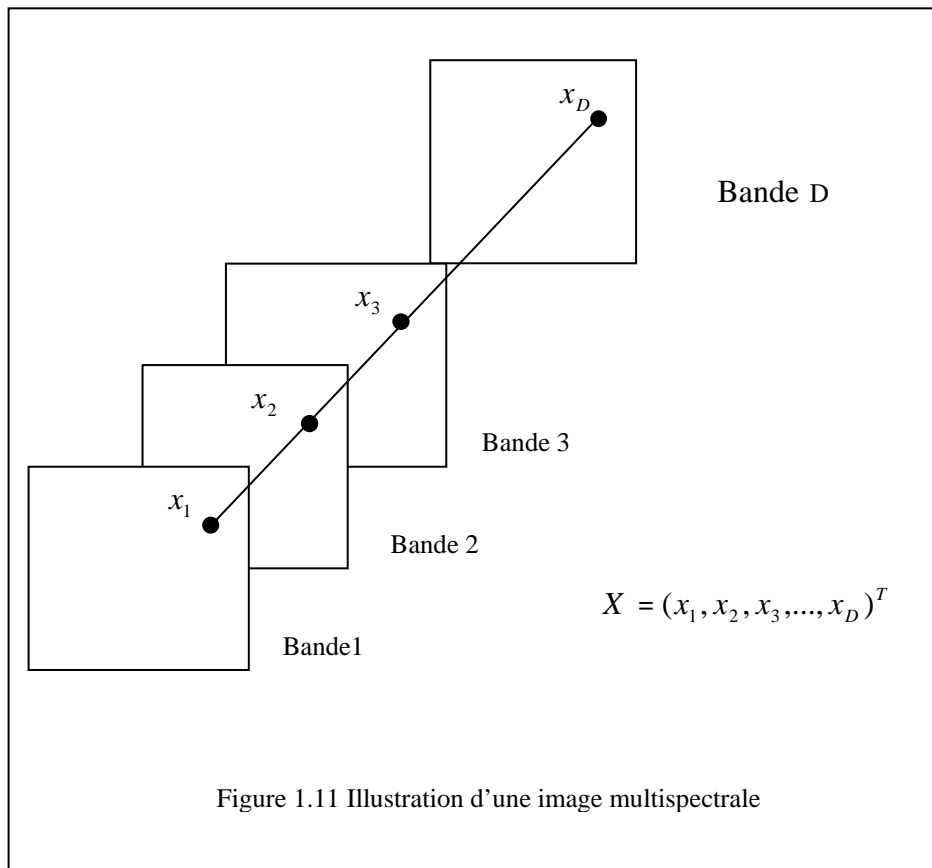
Code	Element	Value	Output
0000100	coeff_token	TotalCoeffs=5,T1s=3	Empty
0	T1 sign	+	<u>1</u>
1	T1 sign	-	<u>-1</u> , 1
1	T1 sign	-	<u>-1</u> , -1, 1
1	Level	+1	<u>1</u> -1, -1, 1
0010	Level	+3	<u>3</u> , 1, -1, -1, 1
111	TotalZeros	3	3, 1, -1, -1, 1
10	run_before	1	3, 1, -1, -1, <u>0</u> , 1
1	run_before	0	3, 1, -1, -1, 0, 1
1	run_before	0	3, 1, -1, -1, 0, 1
01	run_before	1	3, <u>0</u> , 1, -1, -1, 0, 1

Tableau 1.6 Exemple de décodage des coefficients

1.9 Codage des images multispectrales

Une image multispectrale est une image qui regroupe plusieurs plans d'images appelés bandes spectrales. Les algorithmes de compression des images multispectrales sont classés selon l'exploitation de la redondance dans les deux dimensions spatiale et spectrale. La méthode la plus simple consiste à décomposer une image multispectrale en un ensemble d'images monochromes, ensuite, on applique les méthodes de compression usuelles

Une image multispectrale ayant un nombre de D bandes est illustrée en figure 1.11. Les composantes du vecteur X sont les valeurs du niveau de gris des pixels qui se trouvent dans la même position dans les différentes bandes. Le vecteur X est de dimension D, et le nombre de ces vecteurs est égal à la résolution de la bande. Par exemple, une image modélisée dans l'espace RGB est considérée comme une image multispectrale à trois bandes R, G et B.



1.9.1 Méthodes de prédiction

Il existe plusieurs algorithmes de prédiction qui exploitent la redondance spatiale et spectrale pour la compression des images multispectrales. Ces algorithmes effectuent la décorrélation spatiale des pixels voisins dans la même bande, et la décorrélation spectrale entre les pixels de la même position dans les bandes voisines. Pour la décorrélation spatiale des pixels voisins, dans [10] on exploite les modes de prédiction intra de la norme JPEG des images fixes, dans [11] on utilise une méthode différentielle. Pour la décorrélation spectrale, on sait que chaque bande correspond à une même scène, donc, il existe un certain niveau de corrélation spectrale entre les bandes. Comme, les bandes correspondent à différentes longueurs d'ondes, alors, il existe une gamme dynamique différente entre elles, ce qui limite l'application de la méthode différentielle sur ce type d'images. Cependant, il a été remarqué qu'il existe une relation linéaire entre les pixels dans la même position dans les différentes

bandes, donc une prédiction linéaire peut être mise en œuvre pour la suppression de la redondance spectrale entre les bandes.

On considère m bandes B_i ($0 \leq i < m$), on définit une bande unité B_m qui ne peut pas être prédite à partir des autres bandes. Chaque bande B_i peut être obtenu par prédiction linéaire à partir d'une autre bande B_j ($0 \leq j < m$) avec la bande B_m incluse d'après [4].

$$P_{ij} = a_{ij}B_j + c_{ij}B_m \quad (1.15)$$

$$D_{ij} = B_i - P_{ij} \quad (1.16)$$

$$B_i = P_{ij} + D_{ij} \quad (1.17)$$

Où P_{ij} est la bande courante prédite. D_{ij} est la bande résiduelle qui va être codée et transmise. a_{ij} Et c_{ij} sont des coefficients de prédiction déterminés par minimisation de l'erreur quadratique moyenne de prédiction $PMSE = \|D_{ij}\|^2 / n$, n est le nombre de pixels par composante.

1.9.2 Méthode des transformations

Il existe aussi, des algorithmes qui combinent un ensemble de transformation. Par exemple, pour la décorrélation spatiale on exploite la DCT ou DWT. Pour la décorrélation spectrale on utilise la KLT. Les méthodes basées sur la transformation se répartissent en trois classes qui sont selon [3].

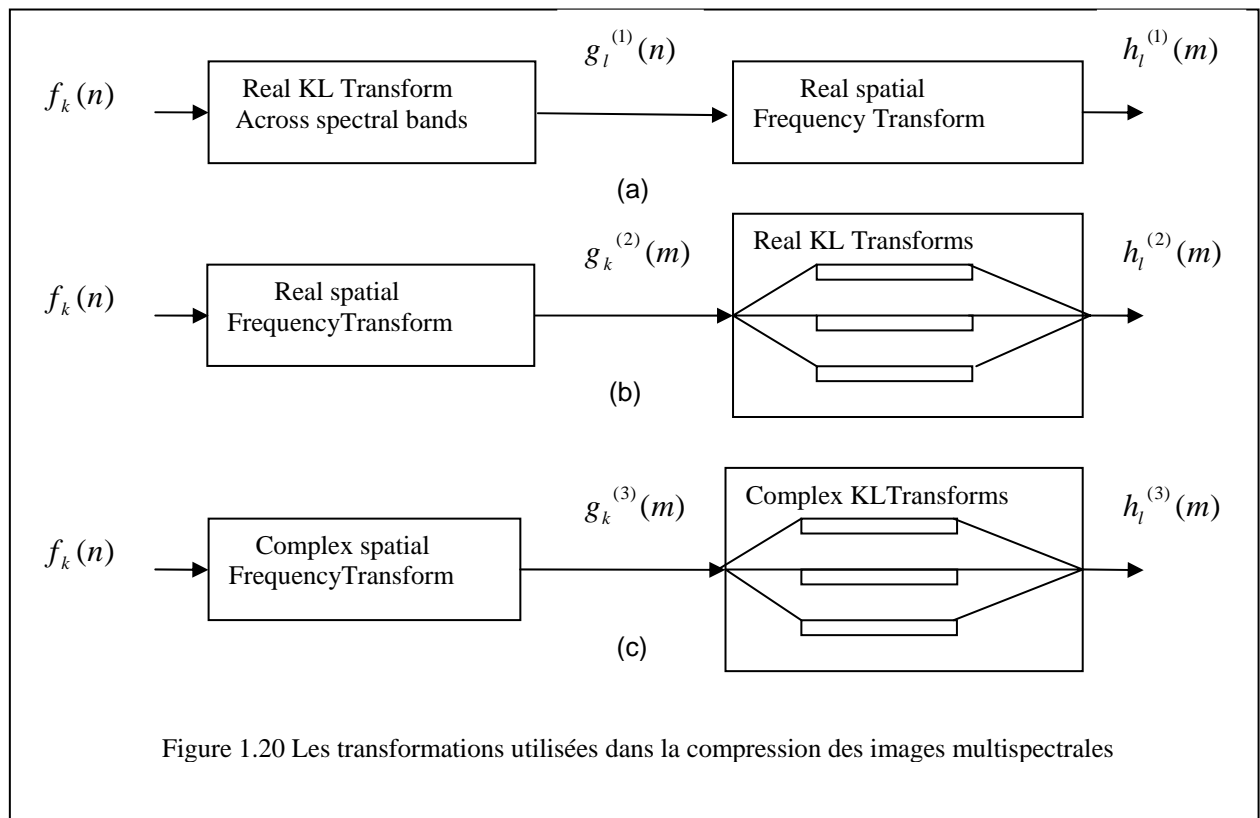
1-Méthode de transformation spectrale-spatiale : En premier lieu, pour la décorrélation spectrale on applique la transformation KLT à travers les composantes spectrales, ensuite on applique la DCT pour la compression de chaque composante. Dans cette transformation l'enregistrement des bandes est nécessaire, et doivent avoir la même résolution.

2- Méthode de transformation spatiale-spectrale :

En premier lieu, on doit appliquer une transformation spatiale DCT ou DWT sur chaque bande, puis, on applique en deuxième lieu la transformation KLT pour chaque coefficient de la DCT, ou pour chaque bande de la DWT.

Dans cette transformation l'enregistrement de l'ensemble des bandes est nécessaire, mais, l'égalité de la résolution n'est pas exigée.

3- Méthode de transformation spatiale-spectrale complexe : Si les composantes spectrales ne sont pas enregistrées alors, la transformation doit être complexe pour conserver l'information de phase entre celles-ci. Car le décalage spatial d'une composante par rapport à l'autre (mauvais enregistrement) correspond à un décalage dans le domaine fréquentiel. Dans ce type de transformation la condition sur l'enregistrement et la résolution des bandes n'est pas exigée. Après chaque transformation mise en œuvre, les coefficients seront soumis aux opérations de quantification et du codage VLC. Les trois méthodes sont illustrées par le schéma de la figure 1.12.



1.9.3 Méthode de la quantification vectorielle

La quantification vectorielle (VQ) est une alternative de codage utilisée dans le domaine la compression des images. L'approche de la (VQ) est justifiée par la considération de plusieurs échantillons à quantifier en même temps, contrairement à la quantification scalaire qui ne considère qu'un seul échantillon à la fois. En plus, la représentation vectorielle

convient mieux pour les images multispectrales car, chaque composante x_i du vecteur X représente un pixel dans la même position dans une bande B_i . Dans le cas des images multispectrales ayant un nombre égal à D bandes, la quantification vectorielle est limitée à cause de la grande dimension D des vecteurs d'entrée. Alors, on est obligée de surmonter ce problème par l'introduction d'autres méthodes.

Par exemple dans [13] on introduit la méthode du quantificateur vectoriel partitionné PVQ (Partitioned Vector Quantizer). Dans cette méthode on partitionne le vecteur d'entrée, de dimension D , en un ensemble de N segments que l'on quantifie par un VQ à recherche exhaustive à L -niveaux. En fait, le partitionnement peut être adaptatif par minimisation de l'erreur quadratique moyenne, et on peut avoir donc des sous vecteurs de dimensions différentes. Par conséquent, on aura pour chaque segment un codebook et un QV approprié.

Un quantificateur vectoriel partitionné (PVQ) est composé de N (VQ) à recherche exhaustive $Q_i = (A_i, F_i, P_i)$, à L -niveaux et de dimension d_i , tel que :

$$d_i = D \quad 1 \leq i \leq N$$

Avec :

$A_i = \{c_1^i, c_2^i, \dots, c_L^i\}$ est un sous ensemble indexé de R^{d_i} appelé codebook

$P_i = \{S_1^i, S_2^i, \dots, S_L^i\}$ est une partition de R^{d_i} et ses classes d'équivalence S_j^i satisfont

$$\bigcup_{j=1}^L S_j^i = R^{d_i} \quad \text{et} \quad S_h^i \cap S_k^i = \emptyset \quad \text{pour} \quad h \neq k.$$

La fonction $F_i : R^{d_i} \rightarrow A_i$ définit la relation entre le codebook et la partition tel que $F(X) = c_j^i$ si et seulement si $X \in S_j^i$, l'indexe j du centroïde c_j^i est le résultat de la quantification du sous vecteur de dimension d_i . Cet index sera transmis au décodeur.

1.10 Codage vidéo des images multispectrales

On rappelle qu'une séquence vidéo d'images multispectrales est composée de trames. Chaque trame est formée par un nombre de bandes spectrales captées par une camera à plusieurs canaux. Ces séquences sont stockées dans des bases de données. Dans [5] l'auteur considère une image multispectrale formée de six bandes, et pour sa compression il utilise deux méthodes qui sont :

1-La méthode PCA (Principal Component Analysis) pour la décorrélation temporelle.

Cette méthode est basée sur l'estimation de la matrice de corrélation C tel que :

$$C = AA^T \quad (1.18)$$

Où A est une matrice produite par une simple trame et sa dimension est égale au produit (longueur du spectre) x (hauteur de l'image) x (largeur de l'image). A partir de la covariance on calcule les valeurs propres qu'on ordonne selon un ordre décroissant comme suit :

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_L \geq 0.$$

Où L est le nombre de canaux dans une trame. Les vecteurs propres correspondant u_1, u_2, \dots, u_L forment un ensemble de vecteurs orthogonaux car la matrice de covariance est symétrique. En fonction de l'application, un nombre de vecteurs propres peut être sélectionné pour créer un espace optimal. La matrice A est multipliée par l'ensemble des vecteurs propres U_i (Les vecteurs propres dans les colonnes).

$$S = U_i^T A, \quad 1 \leq i \leq L \quad (1.19)$$

Dans beaucoup de cas, seulement un petit nombre de vecteurs propres (deux, trois ou quatre) peuvent représenter assez bien le spectre original des couleurs. La première bande interne de S contient la plus grande énergie spectrale de l'image. Il est possible d'utiliser la première bande interne comme échelle du gris de l'image, par exemple, pour les méthodes traditionnelles de l'estimation du mouvement. Il est possible de reconstruire l'image spectrale \hat{A} par la sélection des vecteurs propres et les images des bandes internes comme suivent :

$$\hat{A} = U_i^T S, \quad 1 \leq i \leq L \quad (1.20)$$

Les auteurs de cette publication utilisent les mêmes valeurs propres pour cinq différentes trames dans le temps ou on a besoin de calculer six valeurs propres pour une seconde de vidéo.

2-La méthode de prédiction du standard JPEG 2000 pour la décorrélation spatiale.

Pour l'élimination de la redondance spatiale, il existe plusieurs méthodes disponibles, par exemple, l'auteur adopte celle du standard le plus connu JPEG2000, qui utilise la transformation en ondelettes pour les bandes internes. Après la méthode d'analyse des composants PCA, les bandes internes de l'image sont quantifiées et codés sur 8 bits.

La figure 1.13 illustre le principe de codage vidéo des images multispectrales. La figure 1.14 montre un codeur typique utilisant la méthode de compression statistique PCA et la méthode JPEG2000.

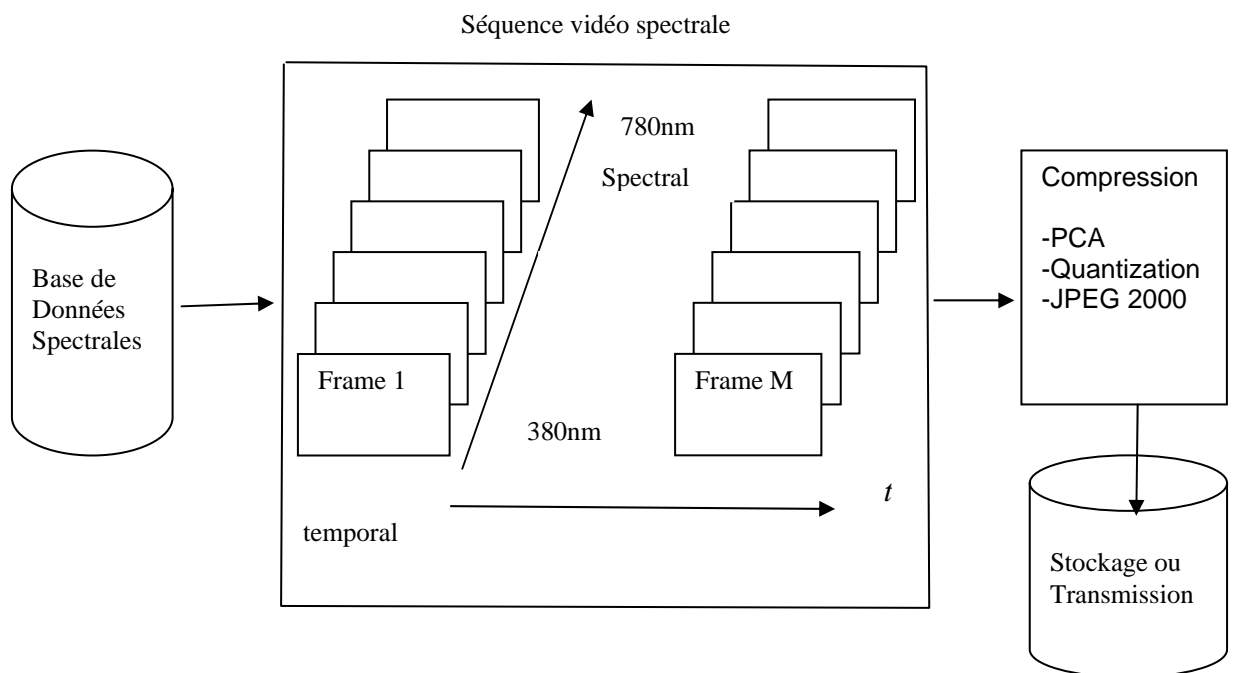


Figure 1.13 Schéma du codage vidéo multi spectral

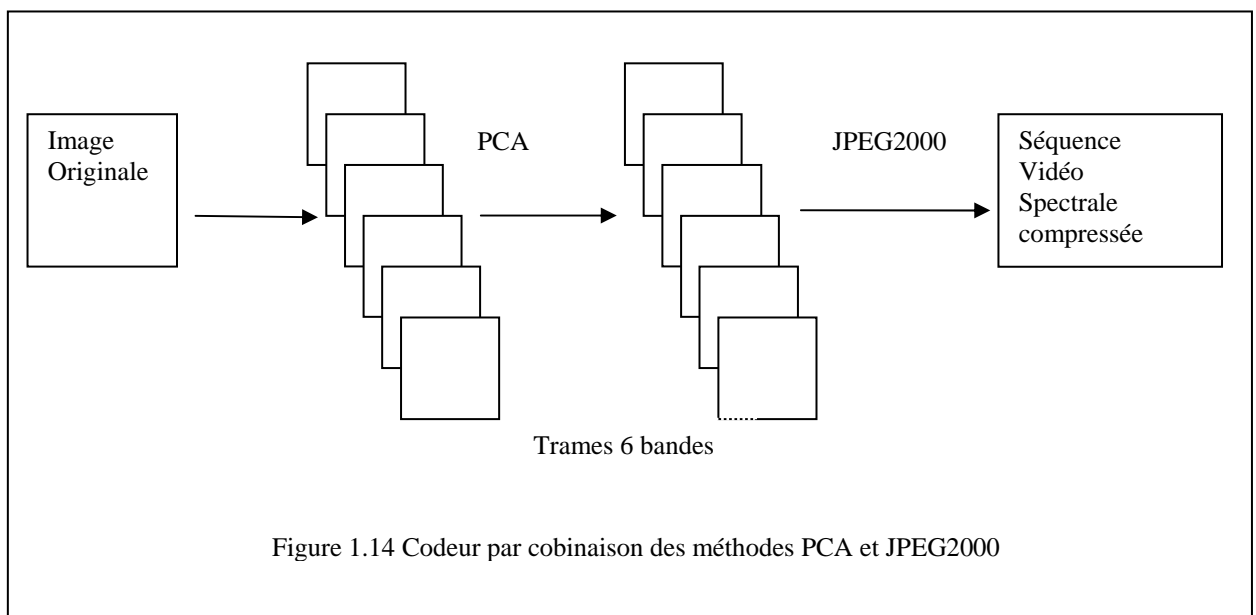


Figure 1.14 Codeur par combinaison des méthodes PCA et JPEG2000

Dans le cas d'une autre méthode, publiée dans [12], utilise une séquence d'images multispectrales à 6 bandes (48 bit/pel) comme données de test avec les nouveaux standards comme MPEG-4 studio Profile et H.164/AVC. L'auteur désigne par xyzX les formats séparés visible/invisible, où X désigne le format des six bandes invisibles. Pour la comparaison avec le standard H.264/AVC qui adopte le format du sous échantillonnage 4:2:0, X est remplacé par le format 420420, et pour la comparaison avec le standard MPEG-4, X est remplacé par le format 444444 entier, sans effectuer le sous échantillonnage des signaux de chrominance.

1.11 Les Critères de performances

La conception du type de codage nécessite la prise en compte des propriétés statistiques de la source, cas du codage entropique sans perte, et de tromper l'organe de la vision le mieux possible dans le cas du codage psycho visuelle avec perte. L'efficacité des différentes méthodes de codage est évaluée sur la base des critères suivants :

Le nombre d'éléments binaires par échantillon

Le taux de compression dépend des caractéristiques de l'image traitée, de la qualité d'une image, ainsi que de la vitesse de compression et de décompression désirée. A titre d'exemple, en télévision dans le cas d'une image de 576 lignes utiles, balayage entrelacé de 25 image par seconde, le débit binaire D_b , et le nombre de bit par échantillon N_b , sont liés par la relation générale suivante :

$$D_b = (f_y + f_u + f_v) \times D_a \times 576 \times 25 \times N_b \quad (1.21)$$

Où, f_y, f_u, f_v sont les fréquences d'échantillonnage respectivement, luminance, différence de couleur rouge, différence de couleur bleue, et D_a la durée d'une ligne active.

Qualité de l'image restituée

Parmi les critères objectifs de mesure de performance et de comparaison très souvent utilisés, on peut citer : L'erreur quadratique moyenne (MSE , Mean Square Error), l'erreur

moyenne absolue (*MAE*, Mean Absolute Error), Le rapport crête du signal sur le bruit (*PSNR*, Peak Signal to Noise Ratio).

Soit une image de résolution $M \times N$ codée sur n bits, et soient C_{ij} et R_{ij} les échantillons de l'image d'origine, et de l'image reconstruite respectivement, On a :

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \quad (1.22)$$

$$MAE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \quad (1.23)$$

$$PSNR_{dB} = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (1.24)$$

Pour la mesure des critères de qualité subjectives de l'image reconstruite. L'imagerie professionnelle utilise une échelle de qualité quantifiée en cinq (5) niveaux de visibilité de dégradations donnés dans le tableau 1.7.

niveau	Visibilité de dégradations
5	Imperceptible
4	Perceptible non gênant
3	Légèrement gênant
2	Gênant
1	Très gênant

Tableau 1.7 Niveaux de visibilité de dégradations

Robustesse aux erreurs de transmissions

Il faut signaler que, contrairement au codage de source, qui consiste à réduire le plus possible le nombre d'éléments binaires par échantillon, le codage de canal consiste à rajouter des bits redondants, mais très utiles pour la détection et la correction des erreurs de transmission qui affectent le canal. Ces codes de gestion des erreurs sont généralement des

bits de parité. Malgré que, cette opération augmente le flux binaire représentant l'information, il faut prendre en compte cette robustesse. Les systèmes DPCM sont les plus sensibles aux erreurs de transmission qui ont tendance à se propager en chaîne sur les lignes et les colonnes et dégrade de façon importante l'image reconstruite.

Complexité de la réalisation

Enfin, il existe des critères objectifs pour le choix d'un algorithme. Il faut tenir compte du temps d'exécution en rapport direct avec la puissance de calcul matérialisée par le nombre des opérations d'addition et de multiplication, en outre la précision exprimé par le nombre de bits nécessaires pour la représentation des données et enfin la facilité d'adaptation de l'algorithme à toutes les spécifications.

1.12 Conclusion

La redondance est un facteur essentiel et utile pour la représentation en grande partie de l'information, mais en même temps un facteur pesant sur le processus de codage de celle-ci. Donc, le but de l'élimination de la redondance, n'est pas son inutilité à contribuer à la représentation de l'information non codée, mais son inutilité à contribuer à la représentation de l'information pertinente codée. Comme il existe plusieurs types de redondances connues en traitement de l'image, alors différentes méthodes et algorithmes ont été développés pour son élimination au cours du codage. Ces méthodes de codage sont caractérisées par les différents critères de performance qui justifient l'intérêt d'une méthode par rapport à une autre.

CHAPITRE 2

DECODEUR DEVELOPPE

2.1 Introduction

Le processus de compression vidéo des images en couleur s'appuie sur des algorithmes développés autour de plusieurs fonctionnalités exigées, en vue de répondre aux différents critères de qualité, aussi bien subjectifs qu'objectifs. Le codage par compensation temporelle constitue un facteur commun entre plusieurs normes réputées telles que la famille des normes MPEGx et H.26x. Un CODEC (Coder / Decoder) est structuré en blocs qui contribuent en complémentarité, à assurer la fonction du codage et du décodage. Dans ce chapitre, nous allons présenter et analyser le schéma de notre codeur vidéo développé en deux versions. Notons que, pour le mode de fonctionnement par compensation temporelle noté MC (Motion-Compensated), les deux versions sont équivalentes.

Mais, pour le mode de fonctionnement par compensation spectrale noté SC (Shift-Compensated), les deux versions diffèrent. En fait, dans la version 1, les vecteurs de décalage SVs (Shift Vectors) entre la bande G et R d'une part, et entre B et R d'autre part, sont codés et transmis. Alors que, dans le cas de la version 2 le mode de fonctionnement par compensation spectrale, noté SC*, les vecteurs SVs ne sont ni codés ni transmis, car on peut les déduire au décodage en utilisant le vecteur mouvement MV (Motion Vector). On signale que le MV est toujours codé et transmis, quelque soit la version et/ou le mode de codage. En terme d'utilisation, le mode MC est utilisé pour coder les séquences vidéo des images modélisées YUV, et les modes SC et SC* sont utilisés pour coder les séquences vidéo des images modélisées RGB. Le mode Intra, qui est en réalité un sous mode inclus dans chacun des modes (MC, SC, SC*), sert uniquement au codage spatial de la première trame de la séquence des images modélisées YUV ou de la première trame de la séquence des images modélisées RGB.

2.2 Approche au codage vidéo des images YUV et RGB

La méthode DPCM (Difference Pulse Code Modulation) était la première approche envisagée, en raison de la grande corrélation entre les pixels voisins. Cette technique est caractérisée par la simplicité du calcul et d'implémentation. Néanmoins, les résultats obtenus n'ont pas permis une avancée suffisante, car même après la compression les fichiers restent volumineux. La technique de compensation temporelle MC (Motion-Compensated) n'est pas une extension du codage spatial, car elle est seulement utilisée dans les séquences vidéo. Elle est donc, un moyen pour supprimer la redondance temporelle causée par le mouvement.

Le système de codage Hybride, est une combinaison du codage par transformation avec le codage par compensation de mouvement. Cette méthode est généralement appliquée aux images YUV, dont la redondance spectrale a été éliminée à l'issue de la transformation de l'espace des couleurs primaire RGB. L'introduction de la technique de compensation de mouvement dans les séquences vidéo, marque une nouvelle époque dans le domaine de codage. Elle a permis de relancer la compression rendant de ce fait plusieurs applications possibles. Le codage par compensation spectrale noté SC (Shift-Compensated), est une méthode développée parallèlement à la méthode Hybride, qui s'applique directement sur les images RGB. Cette méthode est basée sur l'élimination de la redondance spectrale supportée par les pixels dans la même position dans les bandes spectrales. La décorrélation spectrale est mise en œuvre grâce à l'estimation du décalage entre les bandes G et R, et entre les bandes B et R. Il résulte de cette estimation deux vecteurs de décalage, l'un entre la bande G et R, et l'autre entre la bande B et R. Les bandes G et B sont les bandes courantes à coder par rapport à la bande de référence R.

Par analogie avec la compensation temporelle (MC) et la prédiction par compensation du mouvement (MCP) dans la méthode de décorrélation temporelle, on a la compensation spectrale (SC) et la prédiction par compensation du décalage (SCP) dans la méthode de décorrélation spectrale. Ces paramètres marquent la différence entre les deux méthodes de codage, et le facteur commun entre les deux reste toujours le codage par transformation de l'erreur de prédiction DFD (Displaced Frame Difference) dans le cas de la compensation temporelle, ou SFD (Shift Frame Difference) dans le cas de la compensation spectrale. Les paramètres à transmettre sont : le vecteur mouvement (MV), les vecteurs de décalage (SVs), l'erreur de prédiction temporelles (DFD), et l'erreur de prédiction spectrale (SFD). On va voir au cours du chapitre dans la version 2 qu'il n'est pas nécessaire de coder et transmettre les (SVs), car, on peut les déduire au niveau du décodeur en utilisant le (MV).

Une séquence vidéo des images couleurs YUV ou RGB, est constituée d'une suite de trames. Chaque trame est composée de trois images monochromes Y, U et V ou R, G et B. Le processus de traitement de chaque trame, consiste à subdiviser chaque image monochrome de la trame en un ensemble de macro blocs MB ensuite, les macro blocs sont traités l'un après l'autre. Dans notre codeur, nous adoptons le partitionnement du standard H.264/AVC. Pour les opérations d'estimation du mouvement et des décalages, ainsi que pour les opérations de prédiction temporelle et spectrale, on adopte les macro blocs de 16x16 pixels, Tandis que, pour les opérations de transformation DCT et DCT^{-1} , de quantification Q et Q^{-1} , et du codage CAVLC on adopte des blocs de 4x4 pixels. Ces derniers blocs sont obtenus par

subdivision des macro blocs de 16x16 pixels en 4 blocs de 4x4 pixels). La figure 2.1 illustre le processus de traitement d'une séquence vidéo des images couleurs au cours du codage.

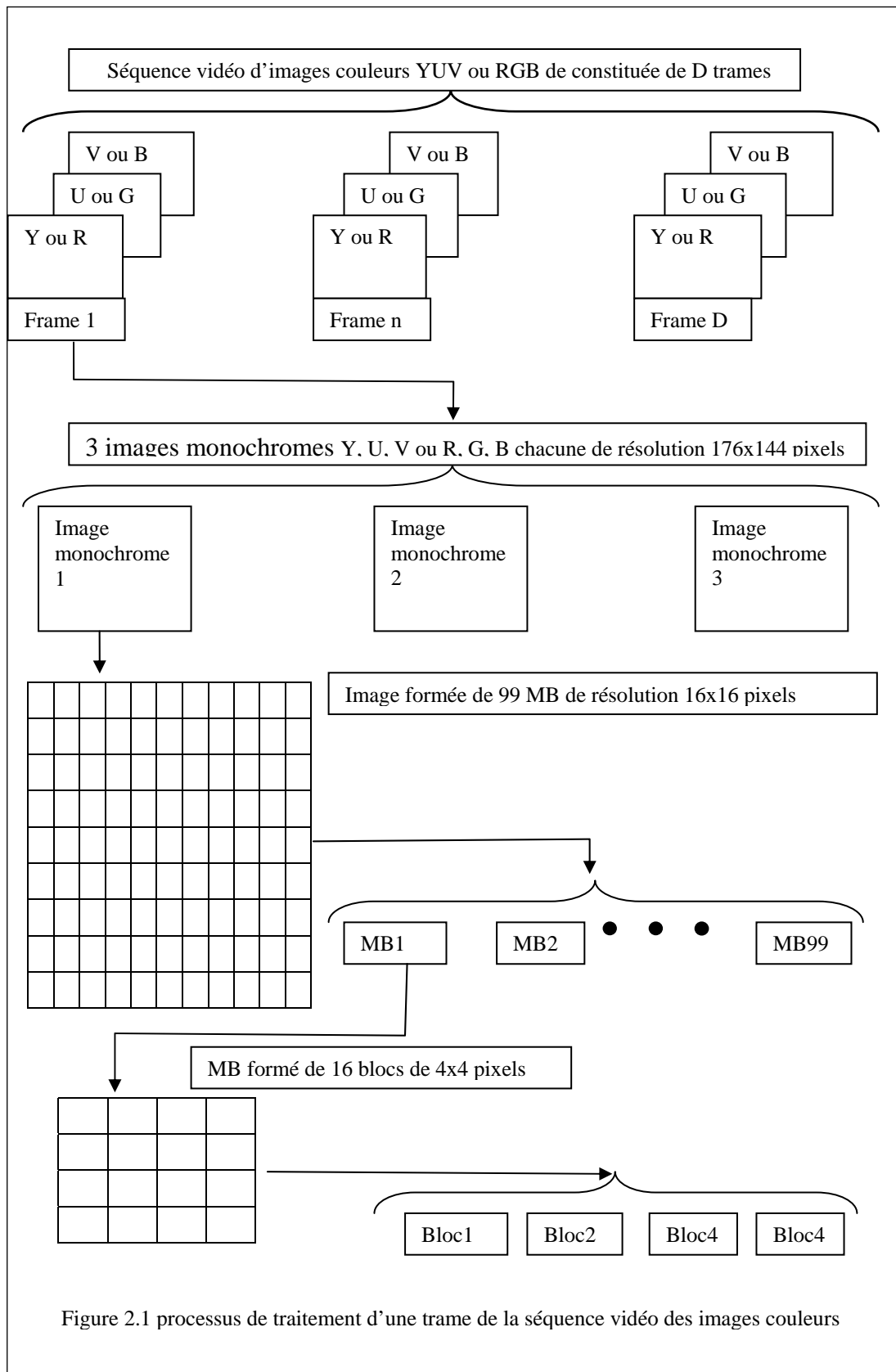


Figure 2.1 processus de traitement d'une trame de la séquence vidéo des images couleurs

2.3 Présentation du codeur version 1

Le schéma du codeur développé version 1 est donné en figure 2.2. Les inverseurs I et S permettent la commutation de trois modes de fonctionnement selon le tableau 2.1. Ces trois modes de fonctionnement sont : Le mode de codage spatial Intra, le mode de codage par compensation temporelle MC (motion-Compensated), et enfin le mode de codage par compensation spectrale SC (Shift-Compensated). On note que dans cette version, les paramètres qui seront codés et transmis sont : l'erreur de prédiction (DFD), le vecteur mouvement (MV), et les vecteurs de décalage (SVs).

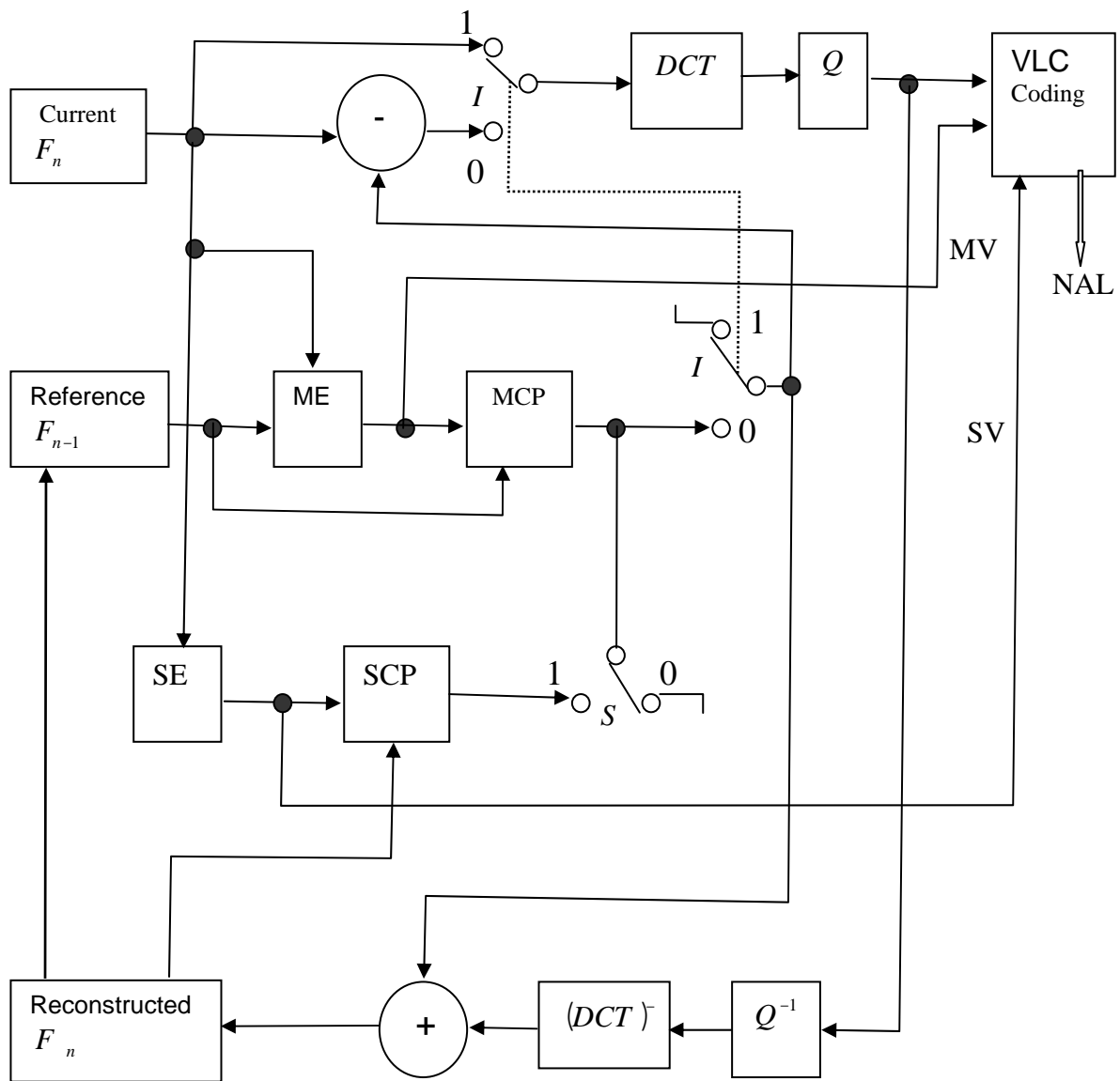


Figure 2.2 schéma bloc du codeur version 1.

I	S	Mode
1	X	Intra
0	0	MC
0	1	SC

Tableau 2.1 commutation des modes de fonctionnement du codeur version 1 et 2

2.3.1 Fonctionnement du codeur version 1 en mode Intra

Le mode Intra est sélectionné par l'inverseur I mis en position 1 quelque soit l'état de l'inverseur S. Il correspond au codage direct des données par transformation détaillé dans la section 1.8.3 de l'état de l'art. On doit signaler que, ce mode est appliqué uniquement à la première trame de la séquence vidéo des images couleurs YUV ou RGB. En fait, Dans le cas d'une trame YUV le codage mode Intra concerne les trois composantes Y, U, V, et dans le cas d'une trame RGB, le codage mode Intra concerne les bandes R, G et B. Le schéma équivalent figure 2.3 comprend deux parties qui sont : la chaîne directe de codage et la chaîne de reconstruction. La chaîne directe commence par l'étage d'entrée qui contient le macro bloc F_n de résolution 16x16 pixels de la trame courante d'ordre n de la séquence vidéo des images couleurs YUV ou RGB. Dans le cas d'une séquence d'images YUV, F_n appartient à la composante Y ou U ou V. Dans le cas d'une séquence d'images RGB, F_n appartient à la bande R ou G ou B. Maintenant, F_n passe à l'étage DCT où il subit un partitionnement en bloc de 4x4 pixels avant la phase de transformation par DCT. Après, les blocs de 4x4 des coefficients transformés obtenus passent à la quantification et le seuillage dans l'étage Q . A la fin de la quantification nous obtenons un bloc dont les coefficients non nuls sont concentrés en haut et à gauche, et les autres sont nuls. Après le balayage en zigzag des coefficients quantifiés, on obtient une suite de coefficients qui passent à la dernière phase de codage VLC pour produire des unités NAL (network Abstraction Layer) pour la transmission ou le stockage. Dans la chaîne de retour, pour la reconstruction du macro bloc courant F_n , les coefficients à la sortie de l'étage de quantification sont soumis à la quantification inverse dans l'étage Q^{-1} , puis, à la transformation en cosinus discrète inverse dans l'étage DCT^{-1} . Enfin, on obtient les échantillons du macro bloc d'origine. Maintenant, le macro bloc reconstruit F_n va être mémorisé et devient un macro bloc de référence F_{n-1} de la trame précédente

d'ordre $n - 1$, dite de référence. Le schéma équivalent du mode de codage Intra est donné dans la figure 2.4.

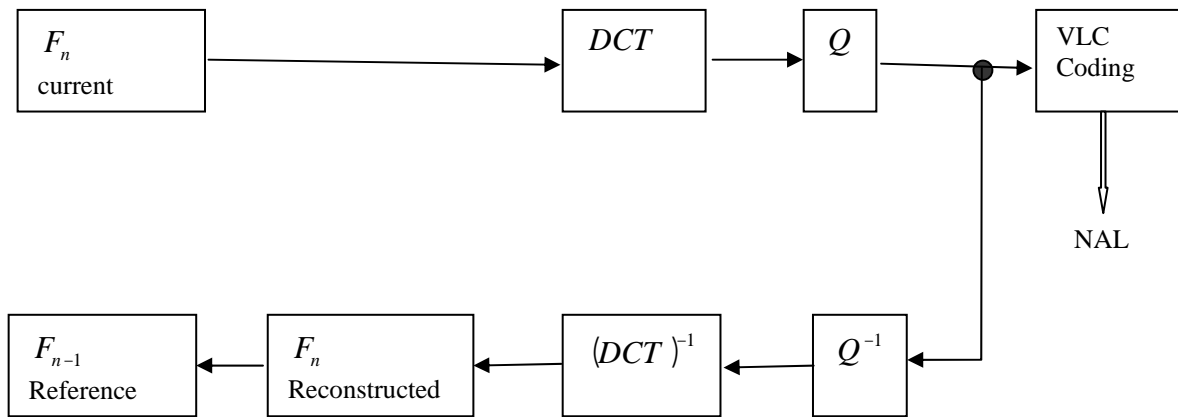


Figure 2.3 Schéma équivalent du codeur version 1 en mode Intra

2.3.2 Fonctionnement du codeur version 1 en mode MC

Ce mode est sélectionné en positionnant les inverseurs I et S sur 0. En fait, il correspond exactement au codage par compensation temporelle (MC) des séquences des images couleurs YUV. Le schéma équivalent est donné dans la figure 2.4. On constate que, en plus des étages du mode de codage Intra, l'apparition de deux nouveaux blocs qui sont : L'étage ME (Motion Estimation) où se fait l'estimation du vecteur mouvement MV, et l'étage MCP (Motion-Compensated prédiction) où se fait la prédiction temporelle. Pour l'estimation de mouvement dans l'étage ME, on applique le principe de correspondance de bloc BMA par minimisation de l'erreur entre le macro bloc F_n de l'image courante et le macro bloc F_{n-1} qui se trouve dans une zone de recherche de l'image de référence.

Ensuite, le vecteur de mouvement résultant MV est acheminé vers l'étage VLC pour être codé et transmis. Ce vecteur mouvement est aussi acheminé à l'étage MCP, qui servira avec le macro bloc de référence F_{n-1} pour la prédiction temporelle d'un modèle \hat{F}_n . En effet, pour avoir l'erreur de prédiction DFD le modèle \hat{F}_n est retranché du macro bloc F_n de l'image courante à coder. Maintenant, cette erreur DFD passe au codage par transformée comme on fait pour le mode Intra. Tandis que, pour la reconstruction du macro bloc F_n le modèle prédit \hat{F}_n est ajouté à l'erreur DFD obtenue par les opérations inverses Q^{-1} et $(DCT)^{-1}$.

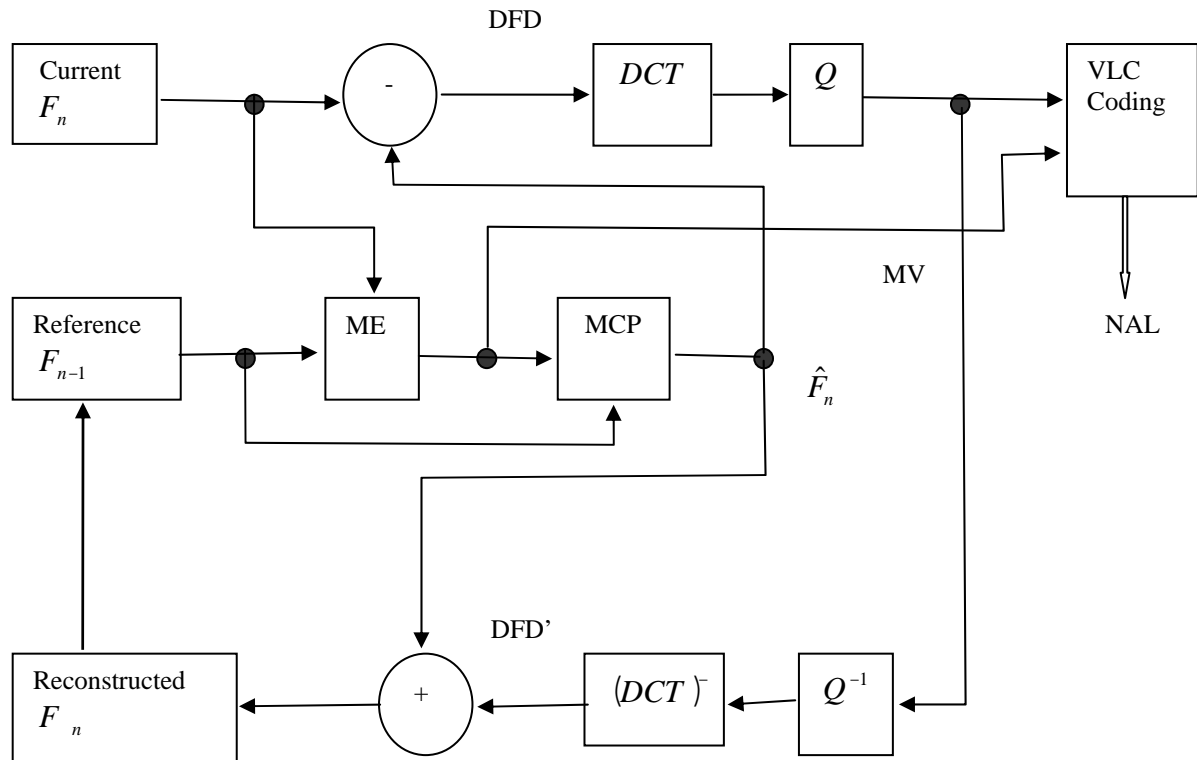


Figure 2.4 Schéma équivalent du codeur version 1 en mode MC

2.3.3 Fonctionnement du codeur version 1 en mode SC

Ce mode est obtenu en maintenant l'inverseur I en position 0 et l'inverseur S en position 1. On signale que ce mode est utilisé pour le codage par compensation spectrale des séquences d'images couleurs RGB. Dans ce cas, on note que la bande spectrale R de référence est toujours codée par compensation de mouvement MC à l'aide des deux étages ME et MCP. Tandis que les autres bandes G et B sont codées par compensation spectrale SC à l'aide des étages SE et SCP. Le schéma équivalent de ce mode de codage est donné dans la figure 2.5. Sur le schéma équivalent on constate encore l'apparition de deux autres étages qui sont : L'étage SE (Shift Estimation) où se fait l'estimation des vecteurs de décalage SVs (Shift Vectors), et l'étage SCP où se fait la prédiction spectrale. L'étage SE reçoit les macro blocs F_n des bandes G, B et R, et après partitionnement du macro bloc de la bande R, on estime les vecteurs de décalages SV_G et SV_B . En effet, SV_G représente le vecteur de décalage du macro bloc G par rapport au macro bloc R, et SV_B représente le décalage du macro bloc de la bande B par rapport au macro bloc de la bande R. Maintenant ces vecteurs de décalage vont être véhiculés vers l'étage final de codage VLC pour être codé et transmis.

Pour la prédiction spectrale l'étage SCP reçoit les vecteurs de décalage et les macro blocs F_n des bandes G et B reconstruites pour produire le modèle prédit \hat{F}_n . Maintenant, pour obtenir l'erreur de prédiction SFD (Shift Frame Difference), on retranche le modèle \hat{F}_n des macro blocs F_n de G et B. Tandis que, pour reconstruire le bloc courant F_n , on ajoute le modèle \hat{F}_n à l'erreur SFD' obtenue par les opérations de Q^{-1} et $(DCT)^{-1}$ sur l'erreur de prédiction SFD. Les particularités du mode de codage SC en ce qui concerne le partitionnement de la bande de référence R, l'estimation SE, et la prédiction sont détaillés dans l'algorithme que nous introduisons ci-après.

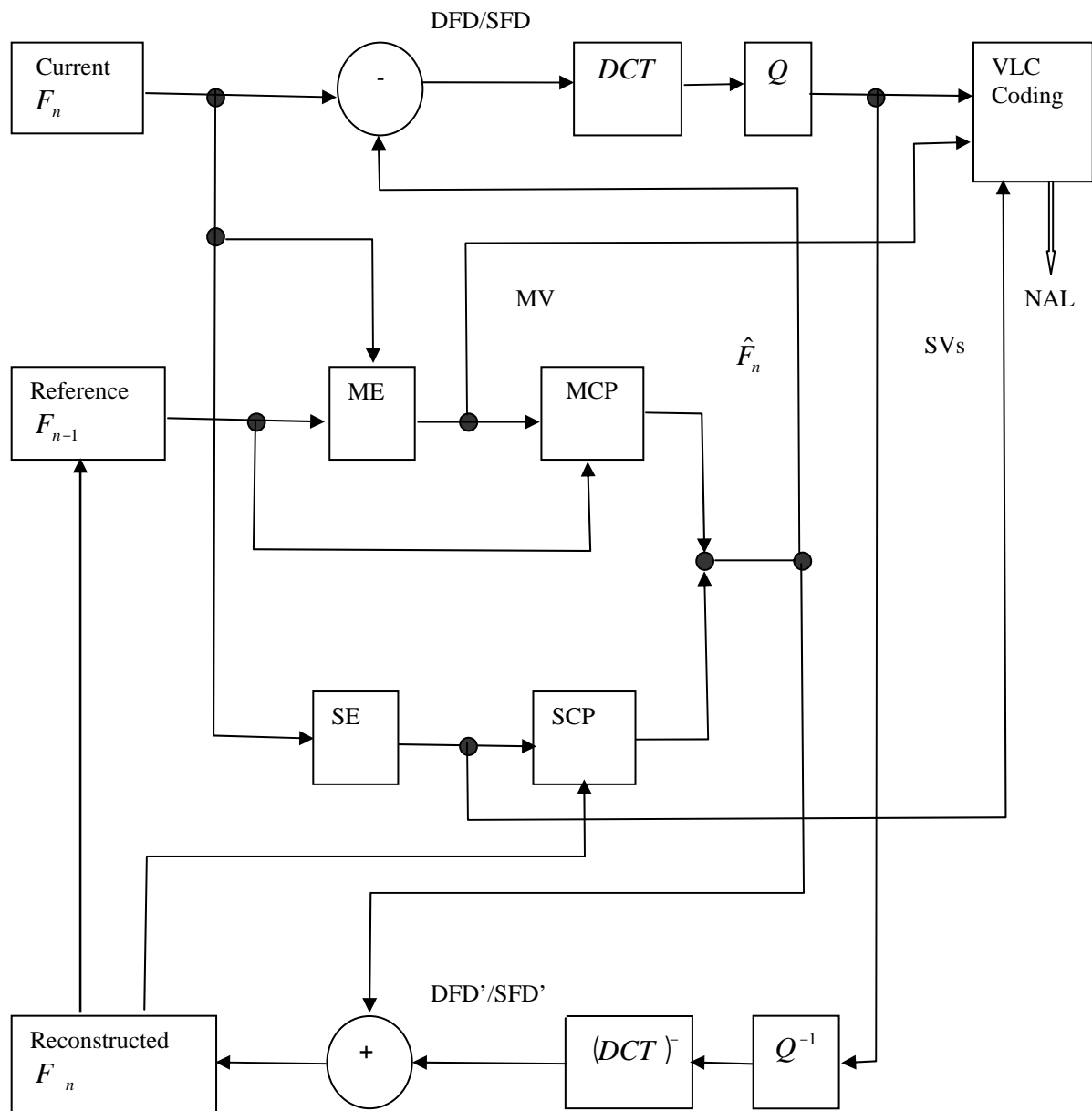


Figure 2.5 Schéma équivalent du codeur version 1 en mode SC

Algorithme du codage mode SC

1- Partitionnement du bloc de référence bande R en utilisant la condition

$$k \text{ PS} \quad R(x, y) < (k + 1) \text{PS}$$

Avec, $k = 0, 1, \dots, 15$ et $\text{PS} = 16$ désigne la taille du bloc utilisée

2- Estimation du décalage entre les pixels des partitions de la bande de référence R et les pixels occupant les mêmes positions dans les bandes courantes G et B à coder.

3- Prédiction par compensation des décalages estimés des pixels du bloc de référence pour produire un bloc très proche du bloc courant.

4- Calcul de l'erreur SFD (Shift Frame Difference) entre le bloc courant et le bloc prédit.

Exemple

Soient à coder les bloc G et B par rapport au bloc R donné dans la figure 2.6.

134	202	12	12
134	202	75	75
206	135	14	14
206	135	76	76

Bande R

51	238	190	190
51	238	47	47
240	50	192	192
240	50	46	46

Bande G

250	198	60	60
250	198	55	55
200	252	61	61
200	252	58	58

Bande B

Figure 2.6 Bloc de référence à partitionner

1) Partitionnement de la bande R

On commence par une phase de partitionnement du bloc de référence R, pour déterminer les différentes zones spectralement homogènes. Si on choisit la taille des partitions égale à 16, alors, avec 256 niveaux du gris on aura $256/16=16$ partitions. Pour déterminer à quelle partition appartient un pixel donné de la partition, on utilise le niveau du gris des pixels dans la bande R tel que :

$$P_p = \text{round}(R(x, y) / \text{PS}) \quad (2.1)$$

Où PS est la taille de la partition, et P_p désigne la partition d'ordre p . Les valeurs sont regroupées en utilisant la condition de partitionnement suivante :

$$\text{PS}.k \leq P_p[i] < \text{PS}.(k + 1) \quad (2.2)$$

Avec $PS = 16$ on aura :

$$16k \quad P_p[i] < 16(k+1), \quad k = 1, 2, \dots, 15. \quad (2.3)$$

Chaque partition non vide P_p , contient des éléments $P_p[i]$. La table de partitionnement est donnée dans le tableau 2.2, on voit quatre partitions seulement non vides, chacune contient quatre valeurs qui sont :

$$P_0[i] = \{12, 12, 14, 14\}.$$

$$P_4[i] = \{75, 75, 76, 76\}.$$

$$P_8[i] = \{134, 134, 135, 135\}.$$

$$P_{12}[i] = \{202, 202, 206, 206\}.$$

Les autres partitions sont vides.

Partitions	Éléments inclus dans la partition			
0 $P_0[i] < 16$	12	12	14	14
16 $P_1[i] < 32$				
32 $P_2[i] < 48$				
48 $P_3[i] < 64$				
64 $P_4[i] < 80$	75	75	76	76
80 $P_5[i] < 96$				
96 $P_6[i] < 112$				
112 $P_7[i] < 128$				
128 $P_8[i] < 144$	134	134	135	135
144 $P_9[i] < 160$				
160 $P_{10}[i] < 176$				
176 $P_{11}[i] < 192$				
192 $P_{12}[i] < 208$	202	202	206	206
208 $P_{13}[i] < 224$				
224 $P_{14}[i] < 240$				
240 $P_{15}[i] < 256$				

Tableau 2.2 Les partitions de R

2) Estimation du décalage (SE)

Une image RGB, présente une similarité entre les bandes, puisqu'elle représente la même structure physique prise avec des modalités différentes. Les niveaux des pixels d'une bande sont différents car, la lumière mesurée au cours de la prise de vue dépend de la nature de la scène, donc de la variation spatiale, et de la réflectivité. En outre, la différence entre les valeurs des pixels, dans la même position d'une bande à l'autre, dépend de la réponse en sélection spectrale des différents filtres utilisés. Dans ce dernier cas, on peut concevoir un modèle linéaire en considérant que l'effet du filtre est un décalage des valeurs des pixels dans la même position entre les différentes bandes, Et par conséquent, on peut dresser une relation mathématique entre les niveaux des pixels de deux bandes adjacentes.

Cette hypothèse convient mieux dans les régions homogènes. Après avoir achevé le partitionnement, on calcule la différence entre les valeurs des pixels se trouvant dans les mêmes positions dans le bloc courant à coder, par exemple, le bloc de la bande G et le bloc de la bande B et les valeurs de partitionnement de la bande R. La valeur du décalage est la valeur moyenne arrondie pour chaque ensemble de différence qui correspond à une partition. Donc pour chaque partition non vide, il existe une valeur du décalage non vide qu'on note S_p où, p désignent la bande et la partition correspondante respectivement. On calcule les composantes des vecteurs de décalage, entre les blocs courants des bandes G et B à coder par rapport à R, en utilisant la formule générale suivante :

$$S_p = E \left\{ G_p(x, y) - R_p(x, y) \right\} \quad (2.4)$$

Alors, les composantes du vecteur de décalage de la bande G par rapport à R seront :

$$S_p^G = E \left\{ G_p(x, y) - R_p(x, y) \right\} \quad (2.5)$$

$$S_0^G = \text{round} \left\{ (190 - 12) + (190 - 12) + (192 - 14) + (192 - 14) \right\} / 4 = 178$$

$$S_4^G = \text{round} \left\{ (47 - 75) + (47 - 75) + (46 - 76) + (46 - 76) \right\} / 4 = -29$$

$$S_8^G = \text{round} \left\{ (51 - 134) + (51 - 134) + (50 - 135) + (50 - 135) \right\} / 4 = -84$$

$$S_{12}^G = \text{round} \left\{ (238 - 202) + (238 - 202) + (240 - 206) + (240 - 206) \right\} / 4 = 35$$

De la même manière, les composantes du vecteur de décalage de la bande B par rapport à R seront comme suit :

$$S_p^B = E \left\{ B_p(x, y) - R_p(x, y) \right\} \quad (2.6)$$

$$S_0^B = \text{round} \{ (60 - 12) + (60 - 12) + (61 - 14) + (61 - 14) \} / 4 = 47$$

$$S_4^B = \text{round} \{ (55 - 75) + (55 - 75) + (58 - 76) + (58 - 76) \} / 4 = -19$$

$$S_8^B = \text{round} \{ (250 - 134) + (250 - 134) + (252 - 135) + (252 - 135) \} / 4 = 116$$

$$S_{12}^B = \text{round} \{ (198 - 202) + (198 - 202) + (200 - 206) + (200 - 206) \} / 4 = -5$$

On obtient les deux vecteurs de décalage de la bande G par rapport à R suivant :

$$SV_G = (S_0^G, 0, 0, 0, S_4^G, 0, 0, 0, S_8^G, 0, 0, 0, S_{12}^G, 0, 0, 0)^T$$

$$SV_G = (178, 0, 0, 0, -29, 0, 0, 0, -84, 0, 0, 0, 35, 0, 0, 0)^T$$

Et le vecteur de décalage de la bande B par rapport à R suivant :

$$SV_B = (S_0^B, 0, 0, 0, S_4^B, 0, 0, 0, S_8^B, 0, 0, 0, S_{12}^B, 0, 0, 0)^T$$

$$SV_B = (47, 0, 0, 0, -19, 0, 0, 0, 116, 0, 0, 0, -5, 0, 0, 0)^T$$

Maintenant on forme les matrices de décalage S_G et S_B données dans la figure 2.7 tel que :

$$S_G[x][y] = SV_G[\text{round}(R[x][y] / PS)] \text{ et } S_B[x][y] = SV_B[\text{round}(R[x][y] / PS)]$$

Où $x = 0, 1, \dots, 15$, $y = 0, 1, \dots, 15$ et $PS = 16$

-84	35	178	178
-84	35	-29	-29
35	-84	178	178
35	-84	-29	-29

S_G

116	-5	47	47
116	-5	-19	-19
-5	116	47	47
-5	116	-19	-19

S_B

Figure 2.7 Matrices de décalage

3) Prédiction par compensation des décalages (SCP)

Pour trouver la valeur prédite des blocs courants à coder de la bande G et la bande B, on ajoute les matrices S_G et S_B calculées auparavant, au bloc de référence de la bande R, car on code les blocs G et B par rapport au bloc R. On aura :

$$\hat{G}(x, y) = R(x, y) + S_G \quad (2.7)$$

$$\hat{B}(x, y) = R(x, y) + S_B \quad (2.8)$$

On déduit les blocs prédits donnés dans la figure 2.8.

50	237	190	190
50	237	46	46
241	51	192	192
241	51	47	47

$\hat{G}(x, y)$

250	197	59	59
250	197	56	56
201	251	61	61
201	251	57	57

$\hat{B}(x, y)$

Figure 2.8 Bloc prédits bandes G et B

4) Calcul de l'erreur de prédiction (SFD)

L'erreur de prédiction qui est la différence entre la valeur d'origine et la valeur prédite est donnée dans la figure 2.9.

$$e_G(x, y) = G(x, y) - \hat{G}(x, y) \quad (2.9)$$

$$e_B(x, y) = B(x, y) - \hat{B}(x, y) \quad (2.10)$$

1	1	0	0
1	1	1	1
-1	-1	0	0
-1	-1	-1	-1

$e_G(x, y)$

0	-1	-1	-1
0	-1	-1	-1
-1	-1	0	0
-1	-1	1	1

$e_B(x, y)$

Figure 2.9 Erreurs de prédiction SFD

Codage des coefficients et des vecteurs de décalage

Après les étapes de transformation en *DCT* et de quantification Q , la suite des coefficients des erreurs résiduelles $e_G(x, y)$ et $e_B(x, y)$ obtenues après le balayage en zigzag, va subir le codage CAVLC (Context-based adaptive variable length coding). Ce codage est détaillé dans la section 1.8.3 au chapitre 1. Dans cette version, il est nécessaire de coder et transmettre les vecteurs SVs pour chacune des bandes G et B car, le décodeur a besoin de ces

vecteurs de décalage SV_G et SV_B , pour reconstruire les blocs courant des bandes G et B. On note que les vecteurs SVs sont codés exactement comme le vecteur MV par le codage VLC dit (Exp-Golomb code) introduit dans la section 1.8.1 du chapitre 1.

2.4 Principe du décodage

En effet, le codage de l'information n'a de sens que si le décodage de celle-ci est réalisable. Car, le codeur lui-même intègre un décodeur pour reconstruire l'image courante qui va être mémorisée pour servir de référence pour la prédiction des images courantes. Donc le décodeur reconstruit le bloc en cours de la même manière que le fait le codeur. Il est clair que le codeur doit intégrer les fonctions qui se trouvent chez le décodeur, telles que la transformation inverse en cosinus discrète, la quantification inverse, la prédiction par compensation temporelle (MCP) et la prédiction par compensation spectrale (SCP).

2.4.1 Décodeur version 1

Cette version du décodeur est compatible avec le codeur version 1, où les deux paramètres MV et SVs sont codés et transmis. On note aussi, la présence des inverseurs I et S qui assurent la commutation des modes de décodage conformément aux modes de codage. Le schéma équivalent est donné dans la figure 2.10.

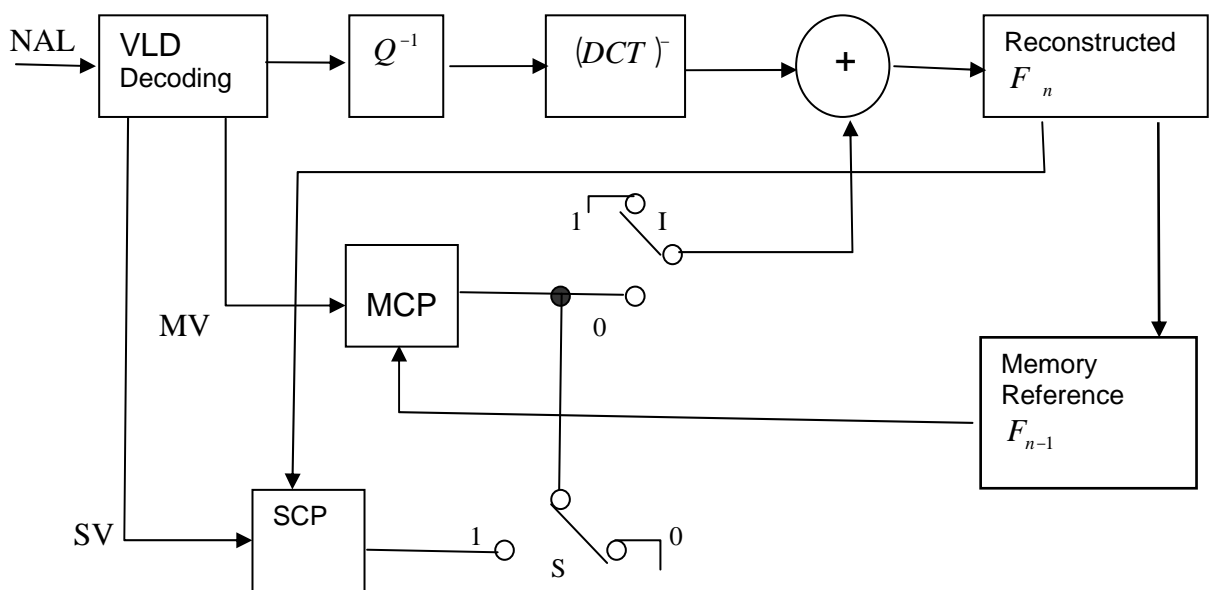


Figure 2.10 Schéma bloc du décodeur version 1

2.4.2 Fonctionnement du décodeur version 1 en mode Intra

Le fonctionnement de ce mode se fait en positionnant l'inverseur I en 1 quelque soit la position de l'inverseur S , on obtient le schéma équivalent donné dans la figure 2.11. On rappelle que ce mode de décodage concerne toujours la première trame de chaque séquence vidéo. C'est-à-dire, dans le cas d'une séquence d'images YUV le décodage mode Intra concerne les trois composantes Y, U et V de la première trame, et dans le cas d'une séquence d'images RGB le décodage mode Intra concerne les bandes R, G et B de la première trame. Le flux binaire reçu à travers le NAL (Network Abstract Layer) est décodé en se basant sur les tables de consultation du codage CAVLC. Nous avons adopté la méthode de décodage du standard H264/AVC dont nous avons donné un exemple à la fin de la section 1.8.3 du chapitre 1 de l'état de l'art. Ainsi, le décodage VLD (Variable Length Decoding), nous permet d'obtenir le bloc des coefficients quantifiés. Ensuite, après réarrangement par balayage en zigzag sous forme de tableaux, on applique les opérations inverses Q^{-1} et $(DCT)^{-1}$ pour obtenir le bloc reconstruit F_n qui sera mémorisé pour servir de référence pour les utilisations ultérieures. En effet, ce mode de décodage est le plus simple et constitue l'opération inverse du codage spatiale réalisé par transformation bidirectionnelle en fréquence.

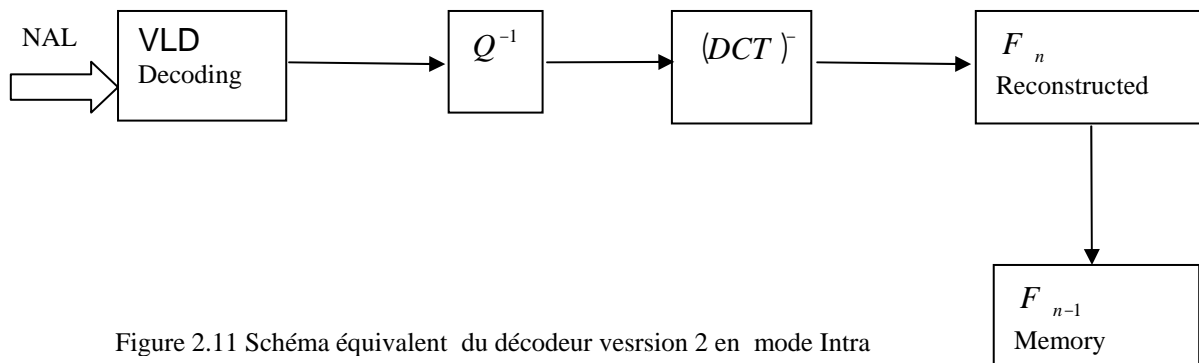


Figure 2.11 Schéma équivalent du décodeur version 2 en mode Intra

2.4.3 Fonctionnement du décodeur version 1 en mode MC

Ce mode est sélectionné par les inverseurs I et S en position 0, alors, on obtient le schéma équivalent donné en figure 2.12. Par ce mode, on décode les informations contenues dans le NAL qui sont, le code du vecteur MV et le code de l'erreur de prédiction DFD. Après décodage dans l'étage VLD, le MV est dirigé vers l'étage MCP où se fait la prédiction MCP, il en résulte le macro bloc prédit \hat{F}_n . Alors que, l'erreur subit les opérations inverses Q^{-1}

et $(DCT)^{-1}$, ce qui donne le bloc erreur de prédiction DFD d'origine. Enfin, on ajoute le bloc erreur DFD au macro bloc prédit \hat{F}_n pour obtenir le macro bloc courant reconstruit F_n qui sera mémorisé pour servir de référence F_{n-1} pour les prédictions ultérieures.

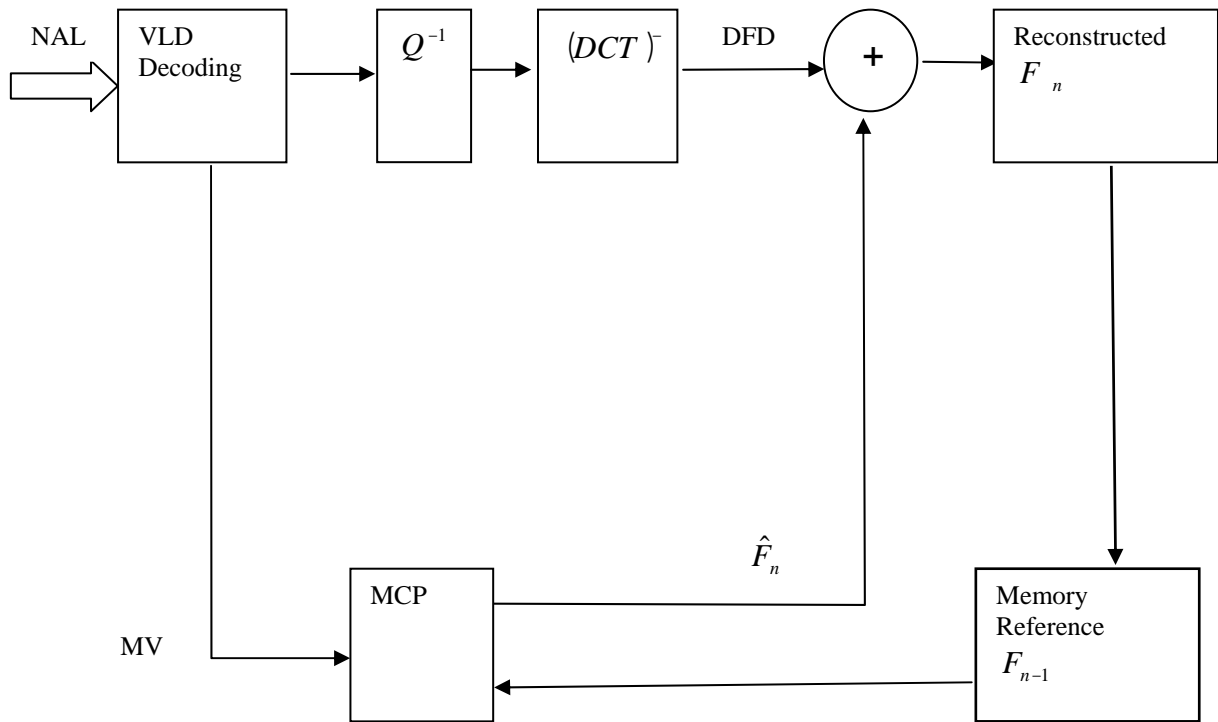


Figure 2.12 Schéma équivalent du décodeur version 1 en mode MC

2.4.4 Fonctionnement du décodeur version 1 en mode SC

Ce mode est sélectionné par l'inverseur I en position 0 et S en position 1, on obtient le schéma équivalent donné dans la figure 2.13. Ce décodage concerne une séquence d'images RGB dont la bande R a été codée par compensation temporelle, et les bandes G, B ont été codées par compensation spectrale. Donc, le NAL doit contenir les codes des paramètres suivants :

L'erreur de prédiction $e_R(x, y)$ contenue dans la DFD, le vecteur mouvement (MV) pour le décodage du bloc R qui va servir comme référence, et enfin les vecteurs SV_G et SV_B pour la reconstruction des blocs G et B en utilisant la bande de référence R. En fait, après le décodage VLD de tous ces paramètres on obtient : Le MV qui se dirige vers le bloc de prédiction MCP, tandis que, les blocs erreur des bandes G et B,

résultants après quantification et transformation inverse, sont ajoutés aux blocs des bandes G et B prédits par SCP pour donner les blocs courants reconstruits des bandes G et B.

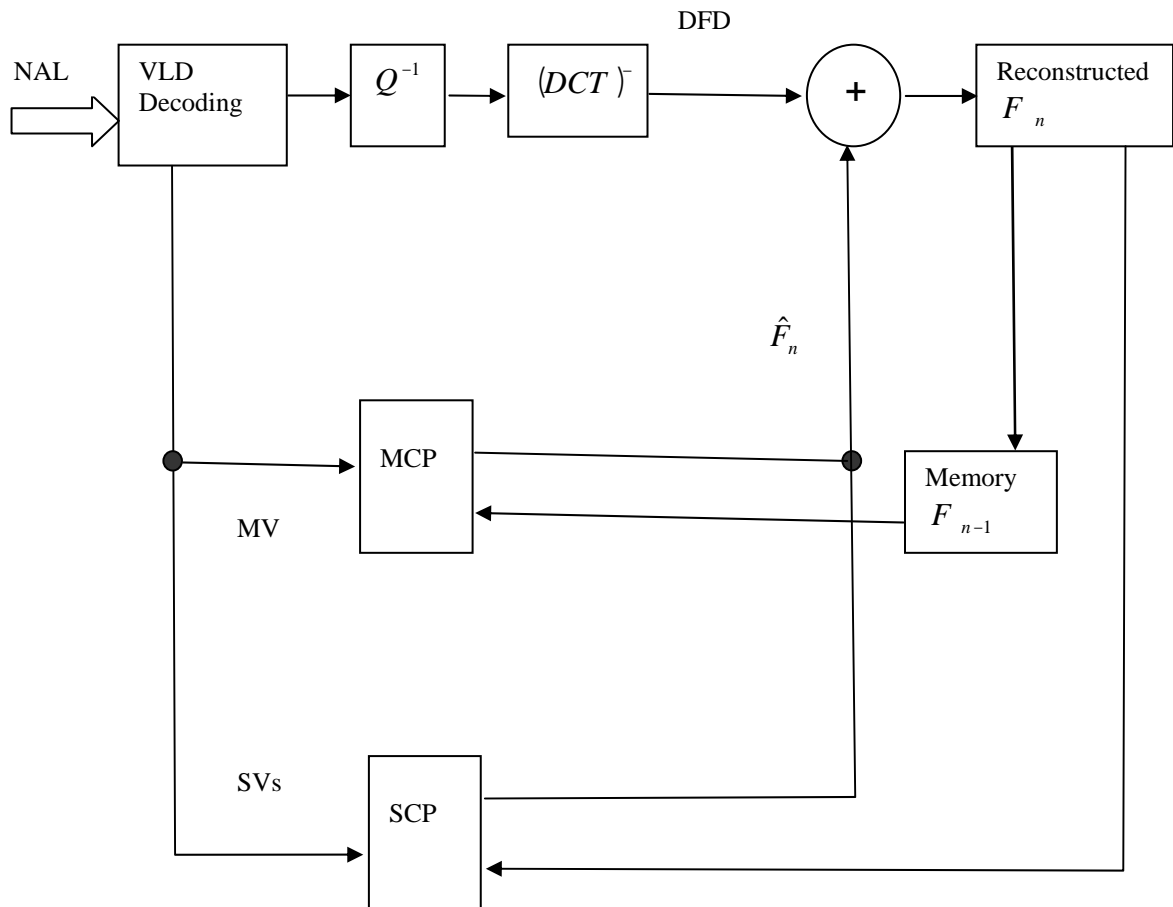


Figure 2.13 Schéma équivalent du décodeur version 1 en mode SC

2.5 Présentation du codeur version 2

L'introduction de cette version vise toujours l'objectif de la diminution du coût de codage. L'idée de base repose sur l'estimation des vecteurs de décalage SVs, en utilisant le vecteur mouvement MV au niveau du décodeur. Dans ce cas, seul le vecteur mouvement est codé et transmis, ce qui permet de réduire la longueur du flot binaire transmis. Le schéma bloc est donné dans la figure 2.14. Dans cette version, on trouve les mêmes étages que dans la première version. Les schémas équivalents des modes de codage Intra pour la première trame, et le mode MC pour le codage de la bande R restent inchangés. La seule différence se trouve dans l'étage SE qui reçoit les macro blocs F_n de la bande courante R de la trame courante, les macro blocs F_{n-1} des bandes G et B de la trame précédente, et le vecteur mouvement MV

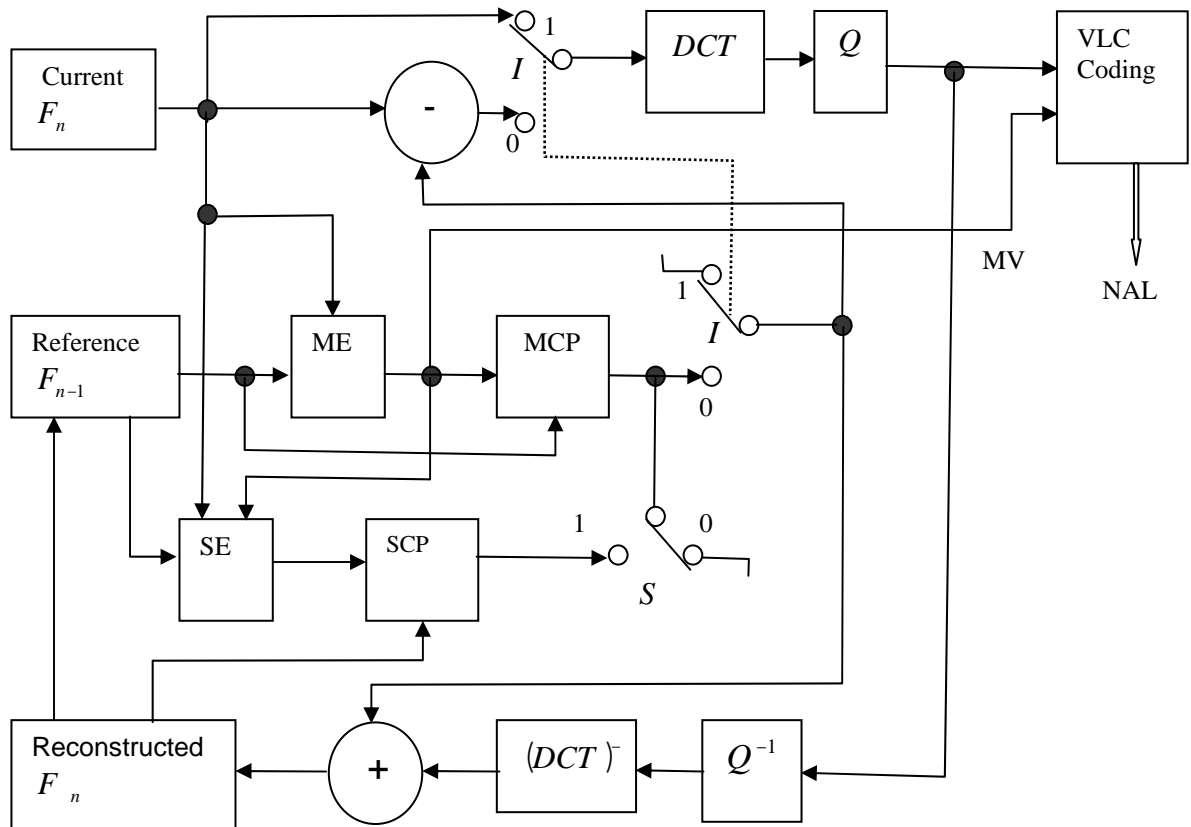


Figure 2.14 Schéma bloc du codeur version 2.

2.5.1 Fonctionnement du codeur version 2 en mode SC*

Ce mode est équivalent au mode SC seulement la notation par l'astérisque signifie l'absence des vecteurs de décalage SV_G et SV_B à la transmission. Le schéma équivalent est donné dans la figure 2.15. Pour l'estimation des SVs, l'étage SE reçoit les macro blocs F_n de la bande R de la trame courante, les macro blocs F_{n-1} des bandes G et B de la trame précédente, et le vecteur mouvement MV de l'étage ME. Comme l'estimation des vecteurs de décalage doit s'effectuer à partir des bandes R, G, et B de la même trame courante, donc, il faut prévoir l'estimation des bandes G et B en exploitant le vecteur mouvement MV. Dans ce cas, l'estimation des vecteurs de décalage SVs, se fait après la prédiction des macro blocs de la bande $G(x, y, t)$ et la bande $B(x, y, t)$ à partir du vecteur mouvement MV et les macro blocs des bandes de références reconstruites $G(x, y, t-1)$ et $B(x, y, t-1)$. On maintient toujours le partitionnement du bloc de la bande courante $R(x, y, t)$. On a les valeurs de prédiction temporelle qui sont calculées par les formules suivantes :

Pour le macro bloc courant de la bande G on a :

$$\hat{G}(x, y, t) = G(x - dx, y - dy, t - 1) \quad (2.11)$$

Pour le macro bloc courant de la bande B on a :

$$\hat{B}(x, y, t) = B(x - dx, y - dy, t - 1) \quad (2.12)$$

Ces valeurs de prédiction temporelle sont à leur tour utilisées pour calculer les valeurs de décalages tels que :

$$S_G = \hat{G}(x, y, t) - R(x, y) = G(x - dx, y - dy, t - 1) \quad (2.13)$$

$$S_B = \hat{B}(x, y, t) - R(x, y) = B(x - dx, y - dy, t - 1) \quad (2.14)$$

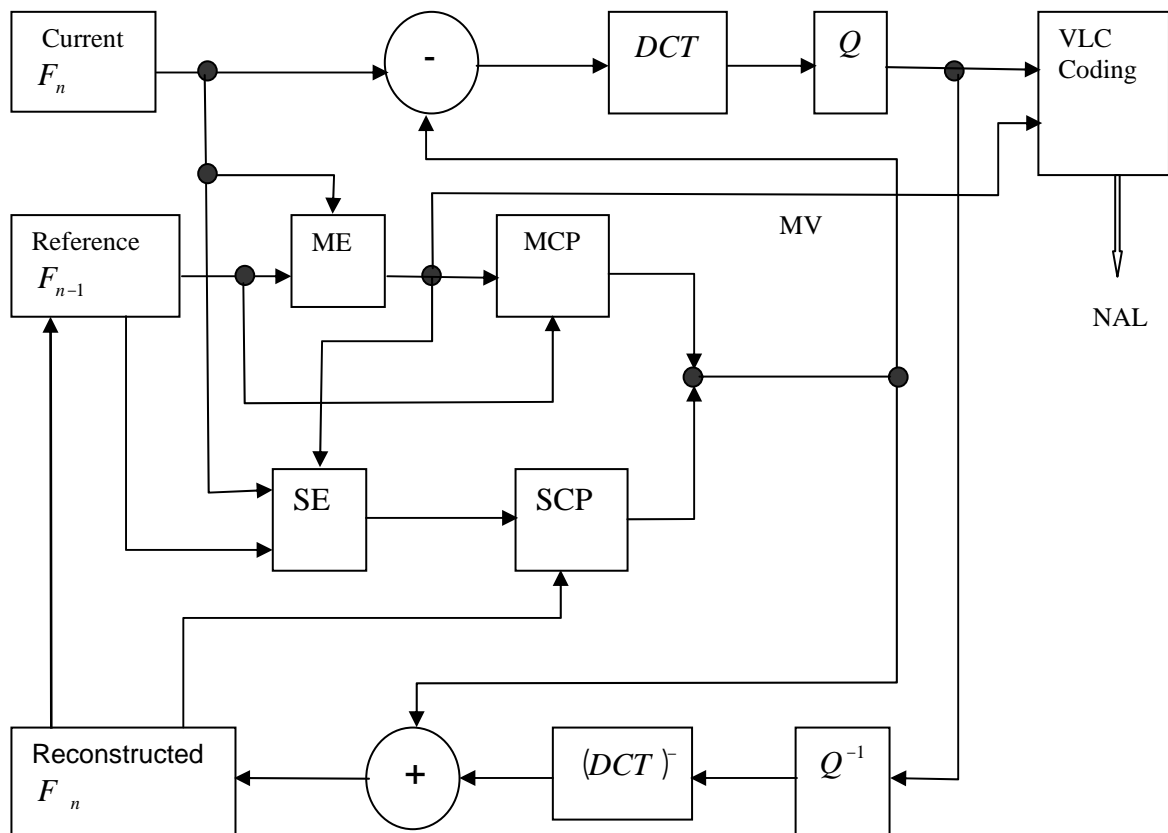


Figure 2.15 Schéma équivalent du codeur version 2 en mode SC*

2.6 Décodeur version 2

On signale que cette version est compatible avec la version 2 du codeur et que la différence entre cette version 2 du décodeur et la version précédente réside dans la méthode de prédiction SCP. Tandis que, les autres parties restent inchangées. Donc, on se limite uniquement à présenter le schéma du mode de décodage SC sujet à cette modification. Pour effectuer l'estimation du SV, en premier lieu le bloc SE a besoin des blocs des bandes références précédentes $G(x, y, t-1)$ et $B(x, y, t-1)$ de l'image de référence précédemment décodée et du vecteur mouvement pour l'estimation temporaire des bandes des blocs courants des bandes $G(x, y, t)$ et $B(x, y, t)$.

En deuxième lieu ces deux dernières bandes sont combinées avec le bloc courant de la bande $R(x, y, t)$ pour donner enfin le vecteur SV. Cette valeur va être à son tour utilisée pour l'estimation SCP de la même manière qu'elle a été expliquée en codage. Donc on voit bien que les vecteurs SVs n'ont pas été codés et transmis, mais ils sont déduits au niveau du décodeur, en bénéficiant de la méthode d'estimation et de prédiction temporelle. Le schéma du décodeur version 2, ainsi que celui du mode SCP sont donnés dans la figure 2.16.

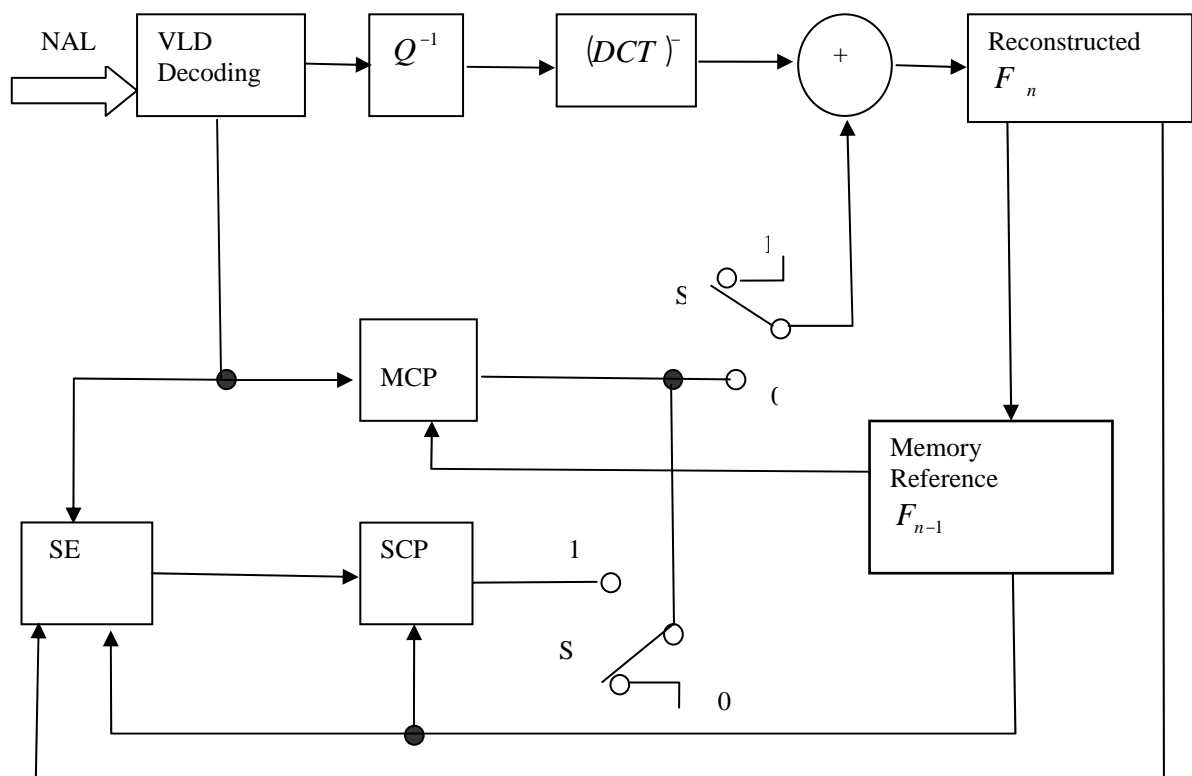


Figure 2.16 Schéma bloc du décodeur version 2

2.7 Conclusion

Le codage par compensation temporelle MC, constitue actuellement un outil très efficace dans le domaine du codage des séquences vidéos des images YUV. Son intérêt réside dans l'exploitation de la méthode d'estimation et compensation du mouvement, pour éliminer la redondance temporelle présente dans les images voisines. Ce type de codage considère la séquence vidéo, comme une image dont la troisième dimension est le temps t . Ceci permet l'utilisation d'un codage basé sur la prédiction temporelle MCP. Dans le codage par compensation spectrale SC appliquée en vidéo aux images RGB, on considère une trame constituée par un ensemble de trois bandes spectrales dont la troisième dimension est la longueur d'onde λ . Alors, dans ce cas pour supprimer la redondance spectrale, on peut appliquer la méthode de prédiction SCP, basée sur une autre notion d'estimation et compensation du décalage entre les bandes G et B par rapport à la bande R. Le codeur développé est en réalité un codeur bivalent, qui intègre les deux méthodes de compensation temporelle et spectrale sur les deux modèles d'image YUV et RGB. Dans le codeur de la première version, Le codage et la transmission des vecteurs SV_G et SV_B en parallèle au vecteur mouvement est nécessaire. Dans la deuxième version, seul le vecteur mouvement est codé et transmis, car les vecteurs SV_G et SV_B sont reconstruit au niveaux du décodeur en exploitant le vecteur mouvement. Alors, on a constaté que ce fait réduit le débit binaire résultant du codage. Le mode de la deuxième version noté SC* constitue une amélioration du codage par compensation spectrale.

CHAPITRE 3

RESULTATS DE SIMULATION

3.1 Introduction

L'efficacité d'un codeur, quelque soit son domaine d'application, ne peut être jugé qu'après une étude statistique des résultats de codage obtenus pour des séquences de test. Pour la comparaison des résultats, ces séquences doivent être choisies parmi les séquences de test standard reconnues en traitement de l'image. L'étude et la comparaison des résultats obtenus se basent sur les mesures objectives de qualité. Alors, ces mesures sont seules responsables de la détermination des performances de la méthode de codage adoptée. En effet, pour l'implémentation d'une méthode de codage, plusieurs formats reconnus sont utilisés tels que CIF, QCIF, ... Cependant, le choix d'un format semble moins important, quand il s'agit du choix du modèle du sous échantillonnage des composantes chrominances Y:U:V tels que, les modèles 4:4:4, 4:2:2, 4:2:0, ..., et du mode de représentation spatiale de l'image par exemple, une image modélisée YUV ou RGB.

Dans notre étude, nous avons choisis arbitrairement le petit format QCIF (Quarter Intermediate Format) (176x144) car, notre objectif est la comparaison entre deux modes de compensation est indépendant du choix de format. Mais, le modèle 4:4:4 a été choisi parce que nous utilisons les bandes RGB qui sont d'égale importance alors, aucune d'elles ne peut être échantillonnée à part. En plus, pour des raisons de compatibilité, on doit maintenir le même modèle du sous échantillonnage 4:4:4 pour les composantes Y, U et V, puisque on code les deux modèles YUV et RGB de la même image

3.2 Considérations préliminaires

1) Pour la simulation, nous avons utilisé la séquence de test standard 'CarPhone'. On procède au codage d'une séquence vidéo des images modélisées dans l'espace YUV par la méthode de compensation temporelle, et le codage de la même séquence des images modélisées RGB par la méthode de compensation spectrale. Il en résulte les trois modes suivants :

- Le mode noté YUV (MC) qui correspond au codage de la séquence vidéo des images YUV par compensation temporelle (MC). On rappelle que ce mode existe est utilisé par les standards mais, dans notre étude nous avons élaboré le programme car, on a besoin de son adaptation pour le codage de la bande R lors de la compensation spectrale.
- Le mode de codage noté RGB (SC) qui correspond au codage de la séquence vidéo des images RGB par compensation spectrale (SC) avec codage et transmission des vecteurs de décalage SVs.

- Le mode de codage noté RGB (SC)* qui correspond au codage de la séquence des images RGB en mode (SC). Donc, tout simplement c'est le mode précédent à la différence les vecteurs de décalage SVs ne sont pas codés et transmis.

2) Par respect à la norme H.264/AVC qui adopte la variation du paramètre de quantification (Q_p) de 1 à 51, alors les relevés des valeurs sont faits en variant le paramètre de quantification de 1 à 50 par incrément de cinq. Cette plage de variation est suffisante pour étudier l'effet de la quantification sur les images reconstruites.

3) Pour mieux juger les performances du codage nous prenons un nombre suffisant de trames de la séquence, par exemple une centaine ce qui permet d'avoir à coder des scènes différentes. Comme le débit est mesurable en kbps (kilobits per second), on préfère travailler avec la cadence une trame par seconde (1fps) pour limiter la longueur des valeurs relevées. Pour les autres cadences usuellement utilisées comme 10, 15, 25 et 30 trames par seconde, le débit peut être obtenu par une simple multiplication la cadence désirée par le débit relevé avec la cadence unité.

4) Pour l'étude des résultats on se base sur des courbes plus utilisées en traitement de l'image, qui décrivent les paramètres essentiels en compression tels que, le débit et la qualité.

- La courbe (Rate-Distorsion) qui traduit la dépendance entre le PSNR (Distorsion) et le débit (Rate). Ici, les relevés des valeurs sont faits pour Q_p variant de 1 à 50 par pas de 5.

- La courbe (Frame-Rate) qui traduit la dépendance entre le débit et le nombre des trames (Frame). Ici, les relevés des valeurs sont faits pour $Q_p = 25$.

- La courbe (Frame-Distorsion) qui traduit la dépendance entre la distorsion et le nombre des trames. Ici, les relevés des valeurs sont faits pour $Q_p = 25$.

Nous donnons également d'autres résultats sous formes de tableaux.

3.3 Codage de la séquence 'CarPhone' YUV

Cette séquence vidéo est composée de cent trames, chacune constitue une image modélisée dans l'espace secondaire des couleurs YUV. Le modèle du sous échantillonnage des composantes est 4 :4 :4 c'est-à-dire, la même résolution pour la composante de luminance Y que pour celles des chrominances U et V. Le format est QCIF (176x144), ce qui fait que la résolution de chaque trame est : 3x176x144 pixels. Nous adoptons un balayage progressif de

cadence une image par seconde (1fps). Pour le traçage des courbes on relève les valeurs concernant le débit (Rate), le nombre des trames (Frame) et la distorsion (PSNR). Les relevés des valeurs sont obtenus pour différentes valeurs du paramètre de quantification. La figure 3.1 montre les trois composantes Y, U et V de l'image 'CarPhone' obtenues après transformation de l'espace des couleurs primaire RGB.

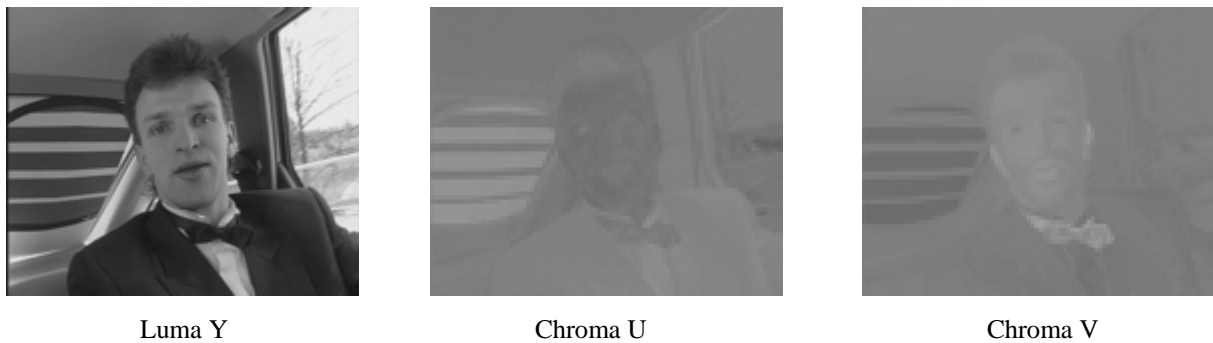


Figure 3.1 Image YUV 'CarPhone'

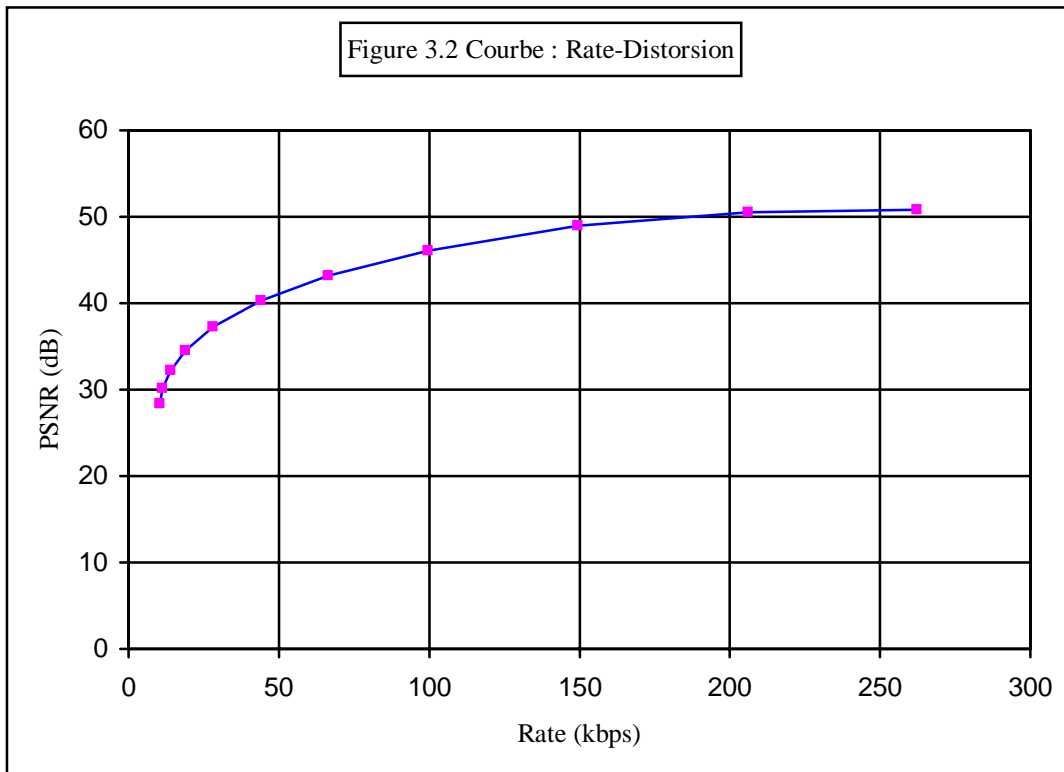
3.3.1 Mode de codage YUV (MC)

Dans ce mode de codage, les trois composantes Y, U et V de la trame sont codées par compensation temporelle. Chaque composante est codée en utilisant sa précédente et le vecteur mouvement. Donc, il est clair que le codeur agit par le même effet sur les trois composantes. On doit noter que, pour voir l'effet du paramètre de quantification sur la qualité de l'image reconstruite, les valeurs de la courbe (Rate-Distorsion) sont obtenues en variant Q_p de 1 à 50 par pas de 5. Alors que, pour les deux courbes (Frame-Rate) et (Frame-Distorsion) le paramètre de quantification Q_p est fixé à la valeur intermédiaire 25 pour avoir une image de qualité moyenne.

1) Résultats de variation de la distorsion en fonction du débit

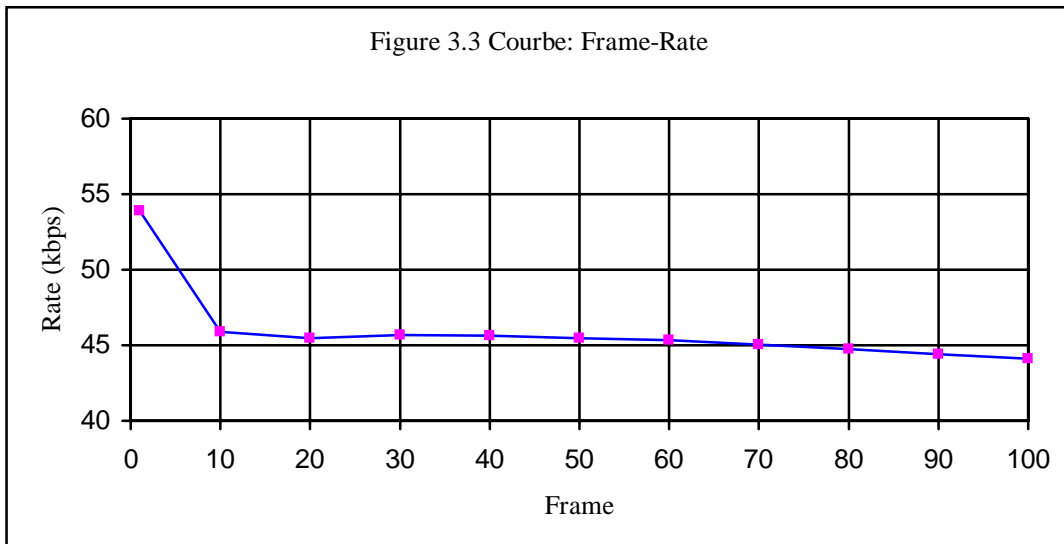
En effet, sachant que le but de la compression est de chercher toujours à diminuer le débit. Dans la figure 3.2, on constate que la diminution du débit provoquée par la réduction du nombre d'échantillons se traduit par la diminution du PSNR c'est-à-dire, la dégradation de la qualité de l'image reconstruite, car on a moins d'échantillons nécessaires pour la reconstruction. Au contraire, l'élévation du débit provoquée par une augmentation du nombre d'échantillons se traduit par une augmentation du PSNR c'est-à-dire, l'amélioration de l'image reconstruite, car on dispose de plus d'échantillons nécessaires à la reconstruction. Donc, on peut conclure que la distorsion est d'autant plus grande que le débit recherché est

plus petit, et vice versa. Alors, il faut garder un compromis entre le débit et la distorsion pour assurer au moins les qualités subjectives de l'image reconstruite après l'opération de compression. Il faut remarquer que, l'augmentation du paramètre de quantification provoque une diminution du débit, ce qui augmente la distorsion, et la diminution du paramètre de quantification provoque une augmentation du débit, ce qui diminue la distorsion. En fait, on doit noter l'importance du paramètre de quantification QP dans la régulation du débit.



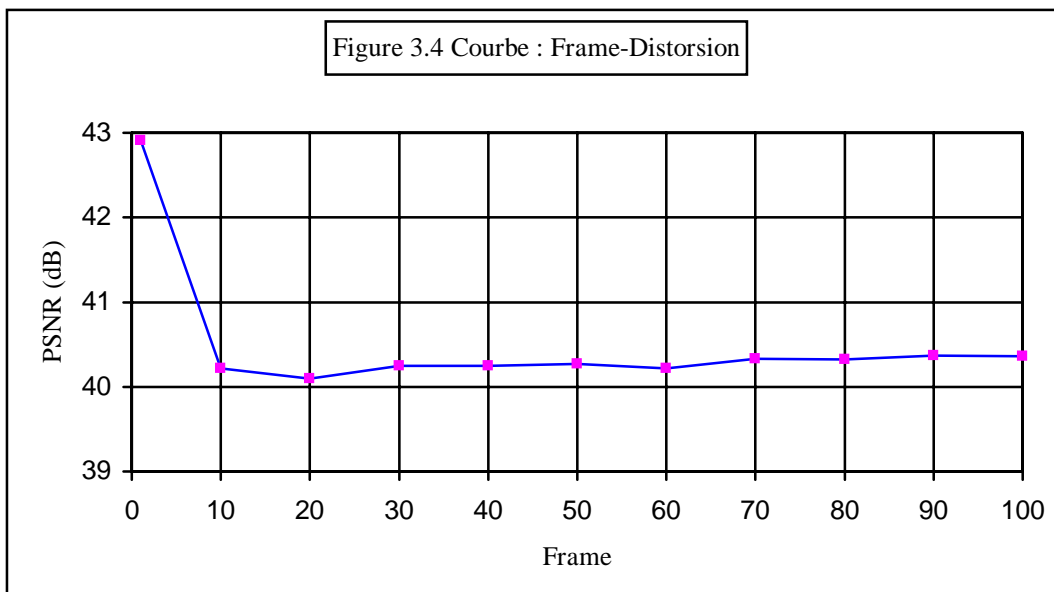
2) Résultats de la variation du débit en fonction des trames

La figure 3.3 montre la courbe (Frame-Rate). En fait, il est naturel que la première image de type *I* codée spatialement par DCT possède le plus grand débit 53.8 kbps, car aucune compensation n'est faite et l'opération du codage est effectuée directement sur les valeurs des échantillons quantifiés. Pour les autres images de la séquence de type *P* prédites temporellement, la valeur du débit chute et varie dans une plage très réduite comprise entre 44 et 46 kbps. La réduction du débit justifie l'efficacité de la compensation temporelle. Tandis que, la faible plage de variation est due à l'existence des zones homogènes dans les trames de la séquence, ainsi qu'à l'absence de rupture de scène et le faible mouvement.



3) Résultats de la variation de la distorsion en fonction des trames

La figure 3.4 montre la courbe Frame-Distorsion. Cette courbe présente aussi un PSNR élevé 42.91 dB pour la première image de type *I* puis, pour les autres images de type *P* le PSNR présente une décroissance rapide et varie dans une plage très réduite entre 40 et 40.5 dB. La relation de quasi proportionnalité entre le débit et le PSNR, fait que le comportement et l'interprétation de cette courbe soient analogues à celle de la courbe (Frame-Rate).



Comme nous voulons montrer l'effet du paramètre de quantification sur les images reconstruites, alors nous donnons le tableau 3.1 qui montre les mesures de qualité PSNR-Y, PSNR-U, et PSNR-V des trois composantes Y, U et V reconstruites. Par exemple, si nous prenons la trame numéro 10 de la séquence 'CarPhone'. Les valeurs du PSNR de ce tableau sont obtenues en choisissant les valeurs de Q_p qui provoquent une distorsion visible à l'œil sur l'image reconstruite.

Q_p	PSNR-Y (dB)	PSNR-U (dB)	PSNR-V (dB)	MeanPSNR (dB)
30	32.95	38.83	39.64	37.14
35	29.35	36.67	37.28	34.43
40	28.50	37.24	37.82	34.52
45	23.23	33.07	33.90	30.07
50	20.61	31.94	32.83	28.46

Tableau 3.1 Relevés des distorsions des composantes Y, U et V en fonction de Q_p

Pour concrétiser l'interprétation de la variation des PSNR données dans le tableau précédent, alors on donne dans la figure 3.5 de (a) à (e) les images reconstruites correspondantes aux trois PSNR. Nous constatons qu'à partir d'un $Q_p=30$, la distorsion commence à être visible sur les images des composantes, et au fur et à mesure que le paramètre de quantification Q_p augmente, les dégradations s'aggravent et les images commencent à perdre leurs détails. Cette dégradation est traduite par la décroissance des trois PSNR. Dans ce cas il faut remarquer que la perte de qualité affecte simultanément toutes les composantes Y, U et V codées par le mode de compensation temporelle YUV (MC).



Luma Y



Chroma U

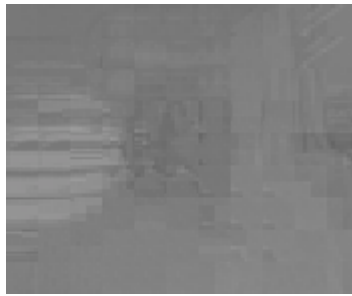


Chroma V

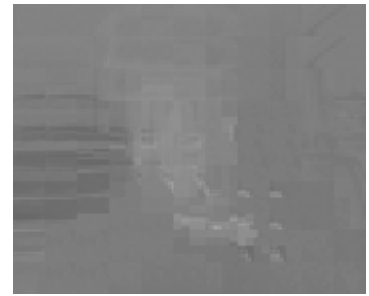
(a) $Q_p= 30$, MeanPSNR=37.14 dB



Luma Y

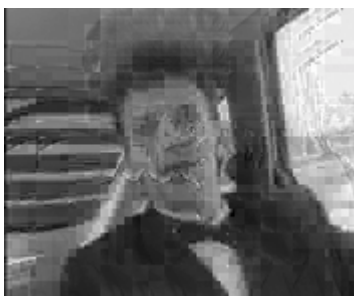


Chroma U



Chroma V

(b) Qp=35, MeanPSNR=34.43 dB



Luma Y



Chroma U



Chroma V

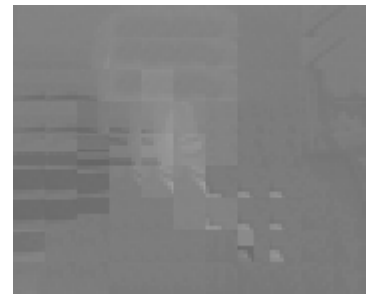
(c) Qp=40, MeanPSNR=34.52 dB



Luma Y



Chroma U



Chroma V

(d) Qp=45, MeanPSNR=30.07 dB



Luma Y



Chroma U



Chroma V

(e) Qp=50, MeanPSNR=28.46 dB

Figure 3.5 Les composantes Y, U et V de la trame numéro 10 reconstruites

3.4 Codage de la séquence ‘ CarPhone’ RGB

Il s’agit de la même séquence précédente de cent images prises dans l’espace de représentation des couleurs RGB. Donc, on maintient inchangé le modèle du sous échantillonnage (4 :4 :4), le format format QCIF (176x144) et la cadence (1fps). Les trois bandes R, G et B d’origine d’une trame de la séquence ‘CarPhone’ sont données dans la figure 3.6. On doit noter que les bandes RGB sont des primaires prises par la camera, alors le codage de l’image sous sa forme primaire RGB permet d’éviter la perte de temps mis pour la transformation de l’espace, et les pertes de données dus aux calculs en virgule flottante et au sous échantillonnage.



Figure 3.6 Les trois bandes R, G et B de L’image‘CarPhone’

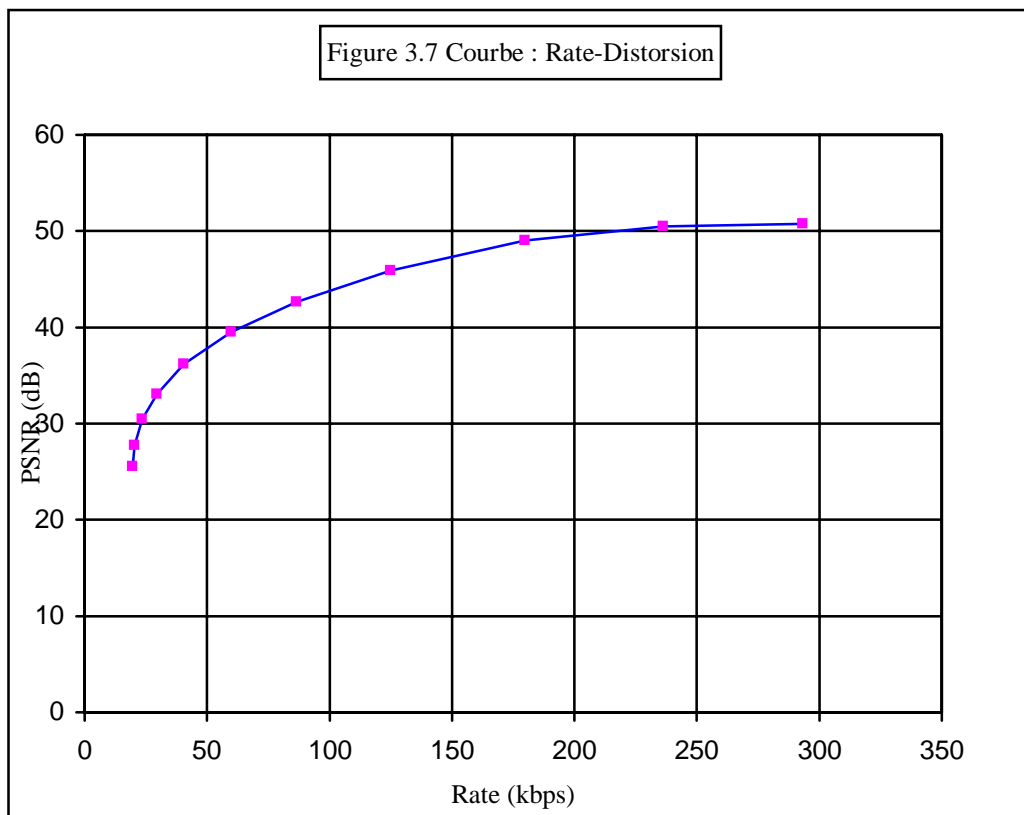
3.4.1 Mode de codage RGB (SC)

Dans ce mode de codage, la bande R de l’image est codée par compensation temporelle, tandis que, les bandes G et B sont codés par compensation spectrale. Alors, dans ce cas le codeur n’agit pas de la même manière sur les trois bandes. Par conséquent, la limite de distorsion de R ne sera pas la même que celles de G et B. En effet, dans ce mode on effectue le codage et la transmission des vecteurs SVs pour permettre au décodeur de reconstruire l’image reçue.

1) résultats de variation de la distorsion en fonction du débit

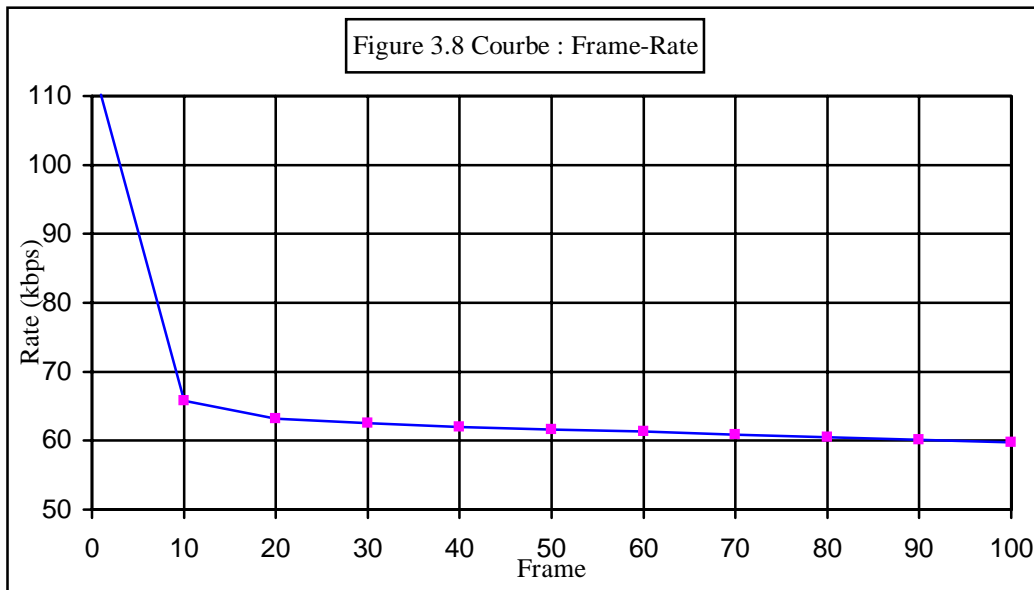
Dans la figure 3.7 on donne la courbe Rate-Distorsion en prenant le PSNR moyen des trois bandes. D’après les mesures de la distorsion, on constate que pour les faibles valeurs du

paramètre de quantification qui se trouvent au dessous de la limite de distorsion, le PSNR de la bande R est proche de la moyenne. Mais, pour les valeurs de Q_p qui se trouvent au dessus de la limite de distorsion, le PSNR de la bande R tend à s'éloigner de la moyenne, causant ainsi sa distorsion. Alors que, les bandes G et B résistent à la distorsion, car, leur PSNR restent proches de la moyenne. En quelque sorte les bandes G et B ne sont pas affectées par la distorsion au même instant que la bande R. Donc, on tire déjà la conclusion que les bandes G et B codées par compensation spectrale présente une certaine résistance à la distorsion.



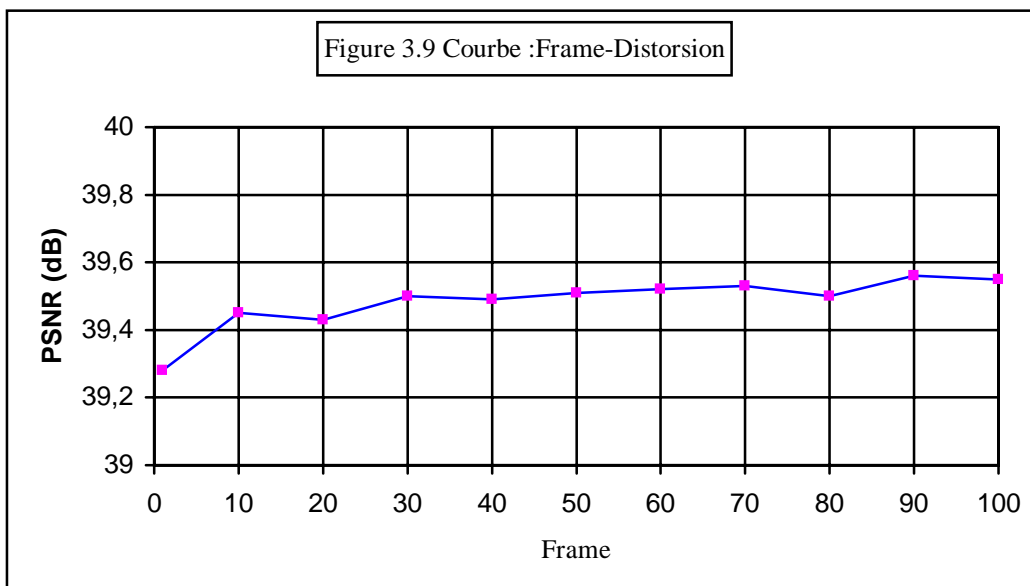
2) Résultats de la variation du débit en fonction des trames

La figure 3.8 montre la courbe Frame-Rate. On voit que le débit dans ce mode est aussi maximale 110.16 kbps pour la première image de type *I* codée directement par transformée, ensuite, pour les trames codées par compensation spectrale le débit chute et varie entre 59 et 66 Kbps. Donc, cette méthode n'est pas moins performante que la méthode de compensation temporelle. On note que dans la première image de type *I* les bandes R, G, et B sont codés spatialement et le débit peut être réduit si on exploite la redondance spectrale.



3) Résultats de la variation de la distorsion en fonction des trames.

La figure 3.9 montre la courbe Frame-Distorsion. Pour la première image le PSNR est minimale ce qui se traduit par la sensibilité à la distorsion des images type *I* codées spatialement. Pour les images de type *P* codées par compensation, on voit une faible augmentation du PSNR, mais la plage de variation est toujours réduite à cause des zones homogènes et le faible mouvement, et la continuité de la scène.



Le tableau 3.2 montre les mesures de qualité PSNR-R, PSNR-G, et PSNR-B des trois bandes R, G et B reconstruites de la trame numéro 10 de la séquence 'CarPhone'. Les trois PSNR sont donnés en fonction du paramètre de quantification Q_p variable de 30 à 50 par incrément de 5. On constate que les trois PSNR diminuent, au fur et à mesure que le paramètre de quantification Q_p augmente, mais, le PSNR-R est toujours plus petit que le PSNR-G et le PSNR-B quelque soit la valeur du Q_p . Pratiquement ceci veut dire que la bande R sera affectée par la distorsion avant l'affectation des autres bandes G et B. Sachant que, la bande R est codée par compensation temporelle et les bandes G et B sont codées par compensation spectrale. Alors, la résistance à la distorsion des bandes G et B est un atout pour le codage par compensation spectrale.

Q_p	PSNR-R (dB)	PSNR-G (dB)	PSNR-B (dB)	MeanPSNR (dB)
30	32.72	38.21	37.51	36.15
35	29.21	35.09	34.46	32.92
40	26.18	32.95	32.13	30.42
45	23.21	30.36	29.41	27.66
50	20.53	28.67	27.00	25.40

Tableau 3.2 Relevés des distorsions pour les bandes R, G et B en fonction de Q_p

L'interprétation pratique de la variation des PSNR des PSNR-R, PSNR-G, et PSNR-B en fonction du paramètre de quantification est matérialisé dans les images reconstruites des bandes R, G et B. Par exemple, si on prend la trame numéro 10 de la séquence 'CarPhone', on obtient les images de la figure 3.10 de (a) à (e). A partir d'un $Q_p=30$, on constate que la distorsion commence à être visible sur la bande R uniquement au fur et à mesure que le paramètre de quantification Q_p augmente, tandis que les bandes G et B semblent résister à la distorsion et gardent leurs détails. On peut interpréter ce phénomène par le fait que : d'une part, les coefficients transformés de l'erreur de prédiction dans le cas du codage par compensation spectrale possèdent des valeurs assez grandes que la quantification n'en peut annuler qu'une partie. D'autre part, l'absence de la propagation de l'erreur le long de la dimension spectrale limite la distorsion.

Sachant que, les bandes G et B sont codées par notre méthode de compensation spectrale, et que la bande R est codée par la méthode de compensation temporelle déjà utilisée dans les standards. Et comme, pour la même valeur du paramètre de quantification Q_p les images reconstruites des bandes G et B sont plus résistantes à la distorsion que les images de

la bande R. Alors, ceci constitue déjà un avantage de notre méthode de codage par compensation spectrale par rapport à la méthode de codage par compensation temporelle.



Bande R



Bande G



Bande B

(a) $Q_p=30$, MeanPSNR=36.15 dB



Bande R



Bande G



Bande B

(b) $Q_p=35$, MeanPSNR=32.92 dB



Bande R



Bande G



Bande B

(c) $Q_p=40$, MeanPSNR=30.42 dB



Bande R



Bande G



Bande B

(d) $Q_p=45$, MeanPSNR=27.66 dB



Figure 3.10 Les bandes R, G et B reconstruites de la trame numéro 10

3.4.2 Mode de codage RGB(SC)*

Ce mode de codage fonctionne de la même manière que le mode précédent à la différence, les vecteurs de décalage SV_G et SV_B ne seront pas codés et transmis dans ce mode. Ceci constitue un avantage car, l'absence des vecteurs en phase de codage nous permet de réduire le débit transmis. On rappelle que dans ce cas aussi la bande R de chaque trame est toujours codée par compensation temporelle, tandis que, les bandes G et B sont codés par compensation spectrale.

La courbe Rate-Distorsion donnée dans la figure 3.11 de ce mode RGB (SC)* possède la même allure que la courbe Rate-Distorsion du codage en mode RGB (SC), mais il existe entre les deux courbes un décalage (que nous allons montrer dans la section ultérieure réservée à la comparaison entre les modes de codage) dû à l'absence des vecteurs $SVs = \{SV_G, SV_B\}$. Par exemple pour un paramètre de quantification $Qp=25$, le débit en mode RGB (SC) est 59.76 kbps et en mode RGB (SC)* le débit n'est que 50.57 kbps, alors ceci constitue une amélioration de la méthode.

La figure 3.12 représente la courbe Frame-Rate, il est clair que l'absence des vecteurs SVs provoque une diminution du débit quelque soit la trame codée. La figure 3.13 montre la courbe Frame-Distorsion. elle est la même que celle du codage en mode RGB(SC) parce que la reconstruction dépend de la méthode de compensation et que les deux mode RGB(SC) et RGB(SC)* utilisent la même méthode de compensation spectrale.

Figure 3.11 Courbe : Rate-Distorsion

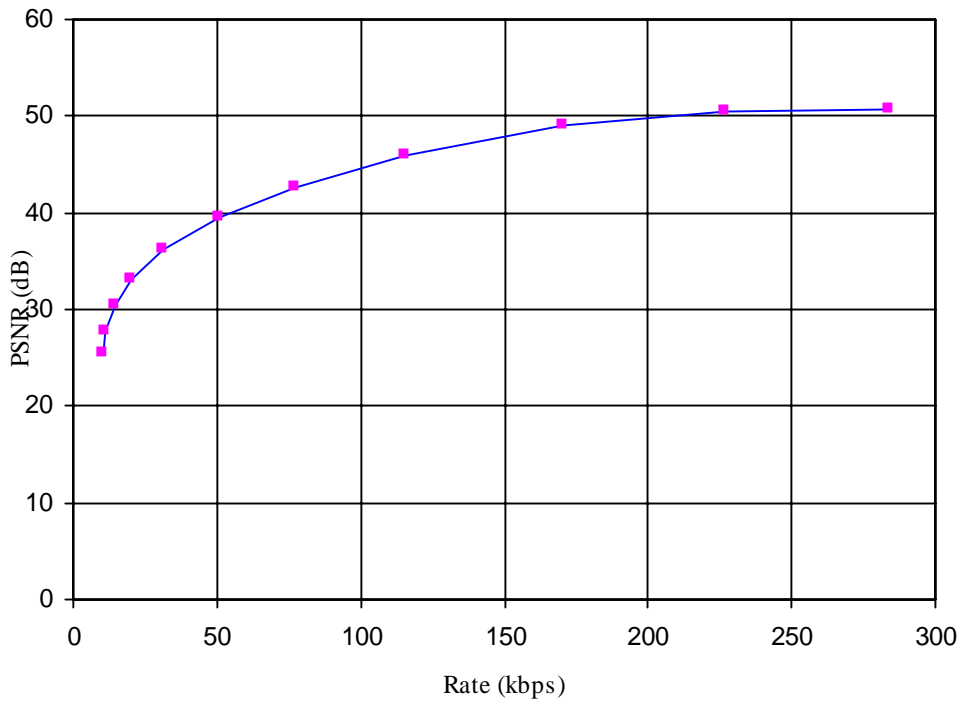
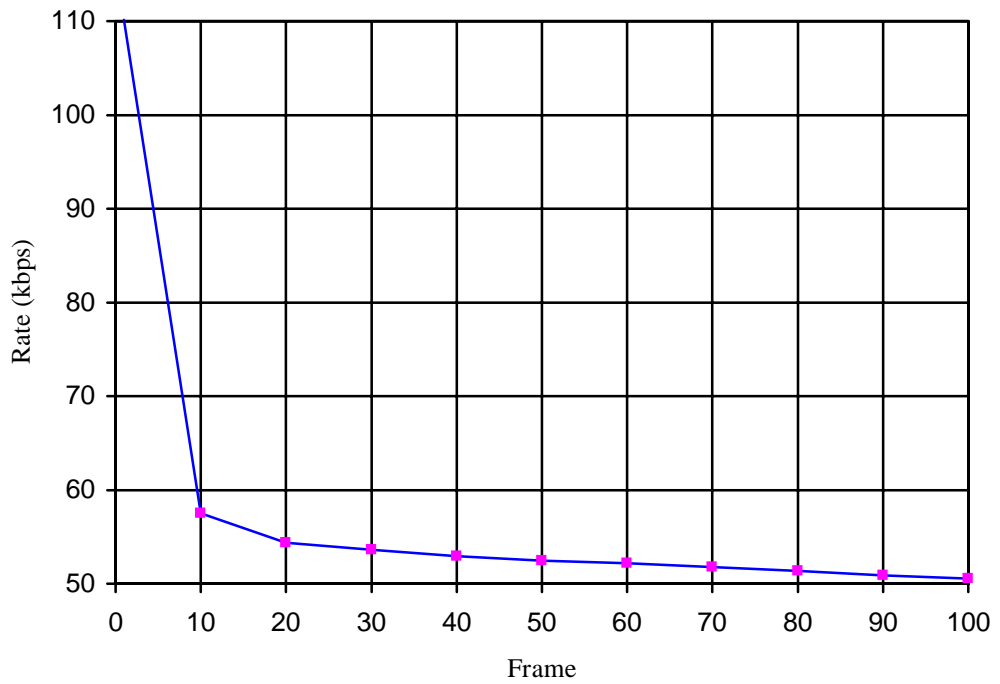
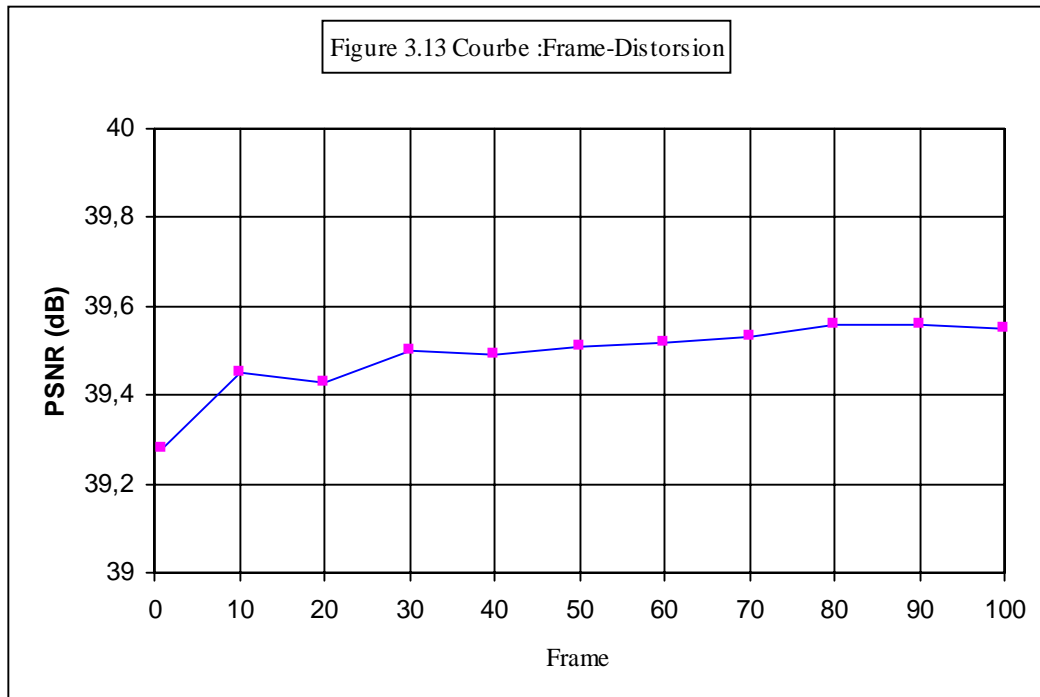


Figure 3.12 Courbe : Frame-Rate





3.5 Comparaison entre les modes de codage

La figure 3.14 montre les trois courbes correspondantes aux trois modes de codage étudiés. La courbe YUV (MC) en haut correspond au mode de codage par compensation temporelle qui donne les meilleurs résultats. Ce mode de codage est décrit par la méthode hybride utilisée par les normes MPEGX et H.264x et son efficacité revient à l'utilisation du principe de compensation temporelle entre deux images voisines. Alors, il permet d'obtenir des débits faibles, avec des mesures de qualités objectives admissibles, assurant une fidélité au cours de la construction des trames. La réduction du débit est dû à la faible erreur de prédiction DFD (Displaced Frame Difference) qui donne moins d'échantillons après les opérations de quantification et seuillage. Cette erreur résulte de la prédiction temporelle par estimation et compensation du mouvement. La courbe RGB(SCP) en bas, correspond au mode de codage par compensation temporelle de l'image bande R et le codage par compensation spectrale des images bandes G et B. C'est le mode qui donne le plus grand débit, à cause des valeurs élevées de l'erreur de prédiction SFD (Shift Frame Difference) par estimation et compensation des décalages, et les valeurs des vecteurs de décalage SV_G et SV_B codés et transmis. En fait, ce mode nous paraît indésirable à cause du grand débit qu'il introduit au cours du codage, mais, comme son avantage se trouve dans la résistance des

images reconstruite à la distorsion, on s'est intéressé à chercher une solution pour son amélioration. On a constaté que les bandes G (t) et B (t) de référence peuvent être déduites au niveau du décodeur en exploitant le vecteur mouvement MV, et les bandes de l'image précédente G (t-1) et B (t-1), donc il est inutile de transmettre les vecteurs SV_G et SV_B . Alors on a pensé à dériver le mode correspondant à la courbe RGB (SC)* qui ignore en phase du codage la transmission des vecteurs SV_G et SV_B . On a obtenu des résultats prometteurs, et on a réussi à rapprocher la courbe RGB (SC)* de la courbe YUV (MC). Pour le haut débit on voit que les trois courbes ont tendance à converger vers une limite supérieure. Cette convergence est dû au fait que, pour les grand débits, correspondants à des valeurs faibles du paramètre de quantification Q_p , l'image est reconstruite à partir d'un grand nombre d'échantillons, ce qui peut assurer la conservation de sa qualité.

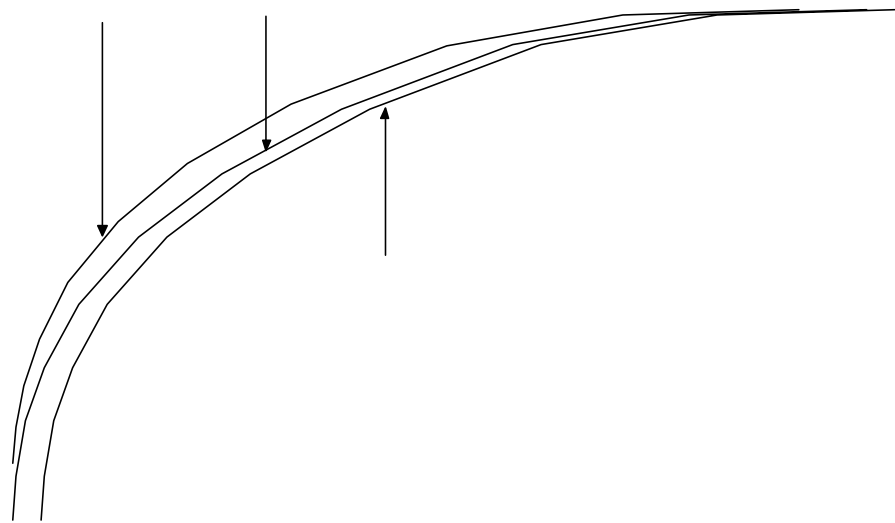


Figure 3.14 Courbes de comparaison entre les trois modes de codage

Pour mieux se focaliser sur la différence, nous donnons en figure 3.15, uniquement les deux courbes correspondantes aux deux meilleurs modes de codage rapprochés YUV (MC) et RGB (SC)*. A partir des relevées des valeurs on peut calculer la différence moyenne entre les distorsions $d_{moy} = 0,99$ dB, et la différence moyenne entre les débits $dx_{moy} = 10,27$ kbps.

Notons que cette différence est assez faible que le rapprochement des performances ne peut pas être exclue, et qu'elle dépend de la texture des images à coder.

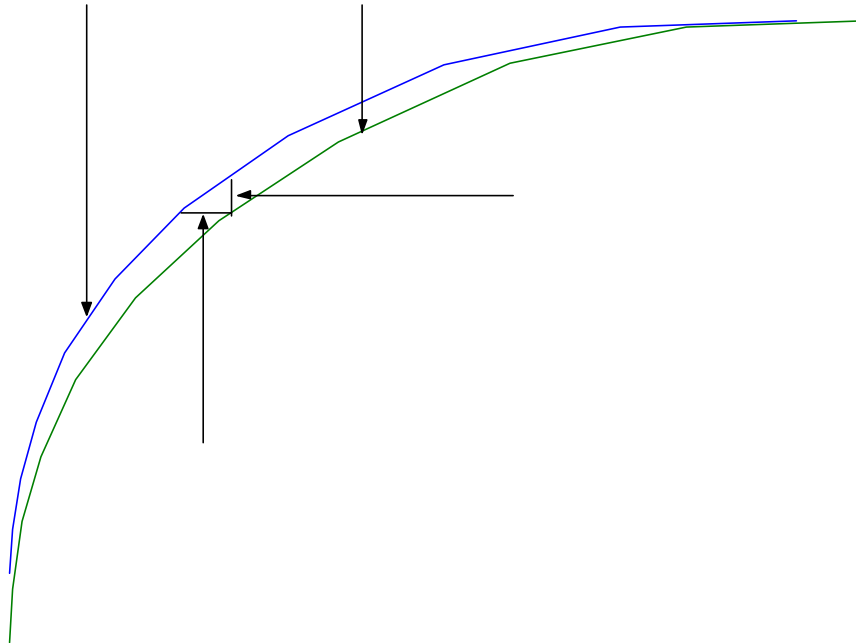


Figure 3.15 Courbes de comparaison entre les deux modes de codage

3.6 Conclusion

D'après l'étude des résultats de simulation on constate que le mode YUV(MC) qui code la séquence des images YUV par compensation temporelle est le plus performant et donne les meilleurs résultats. Le mode RGB (SC) qui code la séquence des images RGB par compensation spectrale entre les bandes en tenant compte des vecteurs de décalage SV_G et SV_B donne le plus grand débit. Malgré le débit élevé de ce mode, il parait intéressant à cause de la résistance à la distorsion des bandes G et B codées par décorrélation spectrale. Alors, ce mode a été amélioré par la dérivation du mode noté RGB(SC)*, plus économique grâce à l'absence des vecteurs SV_G et SV_B en phase de codage. Maintenant les performances des modes YUV (MC) et RGB (SC)* se trouvent rapprochées davantage, et que les images codées

par compensation spectrale sont plus résistantes à la distorsion lors de l'augmentation du paramètre de quantification.

Conclusion générale et perspectives.

Généralement, les normes internationales de codage vidéo utilisent les images modélisées YUV. Ces images sont obtenues par transformation linéaire de l'espace primaire RGB. Le but visé par cette transformation, est d'isoler le signal de luminance Y qui porte les détails les plus significatifs, des signaux de chrominance U et V. Il résulte de cette séparation une décorrélation spectrale, et la possibilité de soumettre les signaux U et V à un sous échantillonnage diminuant ainsi le nombre d'échantillons représentant l'image, ce qui améliore la compression. Pour la décorrélation spatiotemporelle, ces normes adoptent une méthode dite Hybride, basée sur la prédiction temporelle et le codage spatial.

En fait, nous constatons que la transformation de l'espace des couleurs RGB à l'espace des couleurs YUV, provoque différentes pertes de données dues aux erreurs de calcul et au sous échantillonnage des signaux chroma, en plus le temps mis pour effectuer la transformation. Alors, nous avons développé un codeur vidéo des images couleurs RGB, basé sur la compensation spectrale. A titre de comparaison, nous avons intégré à notre codeur un programme basé sur la compensation temporelle pour le codage vidéo des images YUV. Et par conséquent, notre codeur peut fonctionner en plusieurs modes commutables qui sont : Le mode YUV(MC), le mode RGB(SC) et le mode dérivé RGB(SC)*.

D'après les résultats de l'étude, il a été remarqué que le mode YUV(MC) qui code l'image YUV par compensation temporelle donne les meilleurs résultats. Le deuxième mode RGB(SC) qui code l'image RGB par décorrélation spectrale en tenant compte des vecteurs SVs en phase du codage donne des résultats acceptables. Nous avons ensuite, amélioré le mode RGB (SC) par l'introduction d'un mode dérivé noté RGB (SC*) dont les performances se trouvent très rapprochées du mode YUV (MC). Donc, le mode de la deuxième version noté RGB (SC)* constitue une amélioration du codage par compensation spectrale. On note que la différence entre ces deux derniers modes dépend de la texture des images. Et que les images codées par compensation spectrale sont plus résistantes à la distorsion lors de l'augmentation du paramètre de quantification

A titre de perspectives, on peut améliorer d'avantage le mode RGB (SC)* par, la réduction de la taille des partitions des blocs, et l'ordonnement des bandes pour réduire les décalages entre les bandes. Nous avons vue la possibilité de coder une séquence formée d'images modélisées YUV par compensation temporelle, et la même séquence mais formée d'images modélisées RGB par compensation spectrale. Donc, on peut mettre en œuvre une commutation automatique en phase du codage de telle sorte à sélectionner le mode qui donne le plus faible débit, alors ce qui permet d'optimiser le flot binaire à stocker ou à transmettre. En plus, la combinaison des deux modes peut être exploitée pour la correction des erreurs de transmission, ou en cas de perte des paquets de données en video streaming dans les réseaux Internet.

Références

- [1] Iain E.G. Richardson, H.264 and MPEG-4 Video Compression, Video Coding for Next-Generation Multimedia. 2003 John Wiley & Sons, Ltd. ISBN: 0-470-84837-5
- [2] G. Motta, F. Rizzo, and J.A. Storer, "Compression of hyperspectral imagery," Proceedings of Data Compression Conference DCC03, IEEE Computer Society Press, pp. 333-342, 2003.
- [3] D. Treutter, N. Memon, and C.A. Bouman, Handbook of Image and Video Processing, A. Bovik, Editor, Academic Press, 2000. Second Edition 2005.
- [4] R.E. Ladner, "Predictive Coding of Hyperspectral Images," Department of Computer Science and Engineering, Box 352350 University of Washington, Box 352500, Seattle, WA 98195-2500.
- [5] Purmonen, J., Hauta-Ksaru, M., Tuomela, J., Fukuda, H., Mitsui, M., and Yamaguchi, M., "Spectral video intraframe compression and database," Eighth International Symposium on Multispectral Color Science, Proc. of SPIE-IS&T, Electronic Imaging, SPIE vol. 6062, 60620N, © 2006.
- [6] T. Acharya, JPEG2000 Standard for image compression, Wiley Interscience, Edunburg Texas, 2004.
- [7] A. Boukhelif Techniques de compression d'images, office des publications universitaires, Edition : 2.06.4535, avril 2004.
- [8] R. C. Gonzalez, Digital Image Processing, Addison-Wesley Publishing Company, Inc, World Student Series, September 1993.
- [9] A. Martin et A. Pelat, Télévision numérique, Compression et transmission du signal, ellipses/édition marketing S.A. 1997.
- [10] Nacir D. Memon and K. Sayood, "Lossless compression of color images in the RGB domain," Computer Science and Mathematics, Arkansas State University, AR 72767-0070, Electrical Engineering, University of Nebraska Lincoln, NE 68588-0551.
- [11] N.D. Memon, K. Sayood and S. Magliveras, "New Techniques for Reversible Compression of Multispectral Data," in proc. the Computing in Aerospace 9 conference, pp. 148-156, San Diego, CA, March 1993.
- [12] S. Takamura and Y. Yashima, "Multiband Video Coding Using H.264/AVC, MPEG-4 studio Profile and JPEG2000", proceeding of the data Compression Conference (DCC'04) 1068-0.314/04\$20.00© 2004 IEEE.
- [13] S. Benyerbah, Application de l'estimation du mouvement à la compression des séquences d'images, thèse de doctorat en sciences, UMC 2005.
- [14] A. K. Jain, "Image data compression A review," Proc. IEEE, vol. 69, pp. 349-389, Mar. 1981.
- [15] B. Gorod, "Efficiency Analysis of Multihypothesis Motion Compensated Prediction for Video Coding," IEEE Transactions on Image Processing, vol. 9, No. 2, pp. 173-183, Feb 2000.