UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH
FACULTÉ DES SCIENCES ET TECHNIQUES FÈS
DÉPARTEMENT D'INFORMATIQUE

PROJET DE FIN D'ETUDES

MASTER SCIENCES ET TECHNIQUES
SYSTÈMES INTELLIGENTS & RÉSEAUX

AMÉLIORATION DES ALGORITHMES DE CODAGES DES VIDÉOS 3D

LIEU DE STAGE : ECOLE NATIONALE DES SCIENCES APPLIQUÉES AGADIR (LABSIV)

RÉALISÉ PAR : HAMZA HAMOUT

SOUTENU LE : 14/06/2016

ENCADRÉ PAR :

MR. ARSALANE ZARGHILI
MR. ABDERRAHMANE ELYOUSFI

DEVANT LE JURY COMPOSÉ DE :

PR. ARSALANE ZARGHILI
PR. AHLAME BEGDOURI
PR. ABDELALI BOUSHABA
PR. ILHAM CHAKER

ANNÉE UNIVERSITAIRE 2015-2016

Remerciements

Au terme de ce stage de recherche, je voudrais remercier tous ceux qui, d'une manière ou d'une autre, ont contribué à la réalisation de mon mémoire.

Je tiens à remercier Mr. Arsalane ZAGHRILI d'avoir accepté et évalué mon sujet de recherche, je tiens aussi à le remercier pour la qualité de son encadrement exceptionnel, sa patience et sa compréhension durant la préparation de ce projet. Ceci m'a permis de réaliser mes recherches dans de bonnes conditions.

Je souhaite exprimer toute ma reconnaissance à Mr. Abderrahmane ELYOUSFI, un professeur habilité à ENSA d'Agadir, je le remercie pour son encadrement et de ses nombreuses heures consacrées au partage de son savoir et son expérience que ce soit au niveau professionnel ou personnel.

Mes remerciements s'adressent également à l'ensemble des professeurs du département informatique au sein de la Faculté des Sciences et Technique de Fès, pour nous fournir une formation à haute qualité.

Je tiens à remercier sincèrement, ma famille et mes fidèles amies, Rahma AZAMZ, Ichrak BENAMRI, Wiam AMRAOUI, de m'avoir accompagné et encouragé d'aller jusqu'au bout.

Table des matières

Introduction générale

Aujourd'hui, les applications multimédias professionnelles ou grand public mettent de plus en plus en scène des contenus 3D dans des contextes industriels divers. La conception Assisté par Ordinateur (CAO) par l'industrie automobile ou aéronautique, les nouveaux services de télémédecines, les industries du jeu vidéo, des films d'animation 3D, la télévision tridimensionnelle (3DTV) sont quelques exemples représentatifs de domaines où la vidéo 3D joue un rôle incontournable.

Ainsi, les représentations telles que les vidéos multi-vues (MultiView Video en anglais MVV) permettent la création de vidéos 3D. Il s'agit de plusieurs séquences vidéo conventionnelles prises avec plusieurs caméras synchronisées et à des positions différentes dans la scène. Lorsque l'on associe ces vidéo à des vidéos dites des profondeurs on parle de données MultiView Video-plus Depth, MVD, le format MVD reste jusqu'à aujourd'hui le format le plus utilisé.

Par conséquent la migration vers ces technologies, et les applications multimédias, nécessite la création de nouveaux standards de compression vidéo tels que le H.264/AVC (Advanced Video Coding) et le HEVC (High Efficency Video COding). Ces standards sont caractérisés par des hautes performances de codage en termes de taux de compression et qualité vidéo par rapport aux normes précédentes. Cependant, ces performances entrainent de grandes complexités de calcul ce qui rend la tâche de codage difficile, ainsi que la représentation MDV qui est en revanche liée aux coûts importants de stockage et de transmission. D'où la nécessité d'élaborer et de mettre en œuvre des outils de compression efficaces dédiés et optimisés pour ce type de contenu.

Le rapport est organisé comme suit, le premier chapitre représente un état d'art sur la norme H.264. Dans le deuxième et troisième chapitre, une description du nome HEVC et 3D-HEVC respectivement. Dans le chapitre quatre on propose deux implémentation pour l'amélioration du codeur 3D-HEVC, et on retrouvera finalement la conclusion.

Chapitre 1 : Codage des vidéos 2D (H.264)

Chapitre 1 : Codage des vidéos 2D (H.264)

1.1 Les formats des vidéos

Un flux vidéo est composé d'une succession d'images qui défilent à un rythme fixe pour donner l'illusion du mouvement, par exemple 25 images ou bien 30 images par seconde. On trouve généralement un flux audio et d'autres informations comme des sous-titres, des menus, des chapitrages, des interactions et des métadonnées, des informations de propriétés sur la vidéo comme la date de création, le nom de la vidéo, son auteur, etc.

Il existe une multitude de formats vidéo, AVI (Audio Video Interleave) de Microsoft, MPEG (Moving Picure Expert Group) de l'organisme du même nom, FLV (Flash Video) d'Adobe. Ce qu'il faut savoir, c'est qu'un format résulte de deux concepts techniques distincts : les conteneurs et les CoDecs.

1.1.1 Le concept du conteneur

Le conteneur décrit la structure du fichier. Il est utilisé pour stocker la vidéo, son flux d'images, flux audio et métadonnées, selon un schéma bien défini. Il précise notamment quel codec vidéo et potentiellement audio sont utilisés. Il peut également intégrer des sous-titres ou des chapitrages**.** Le principal objectif du conteneur est donc d'organiser la coexistence entre l'image, le son, éventuellement du texte et d'autres données liées. Dans le langage du monde vidéo, on parle de **multiplexage**.

On peut imaginer un conteneur comme une boite avec plusieurs objets à l'intérieur (figure 1.1) mais rangés d'une façon bien définie. Le conteneur utilisé est généralement identifié grâce à l'extension du fichier de la vidéo. C'est pour cette raison que cette notion est souvent confondue avec la notion de format, car il arrive que le nom du format soit identique au nom de son conteneur (tableau 1.1).

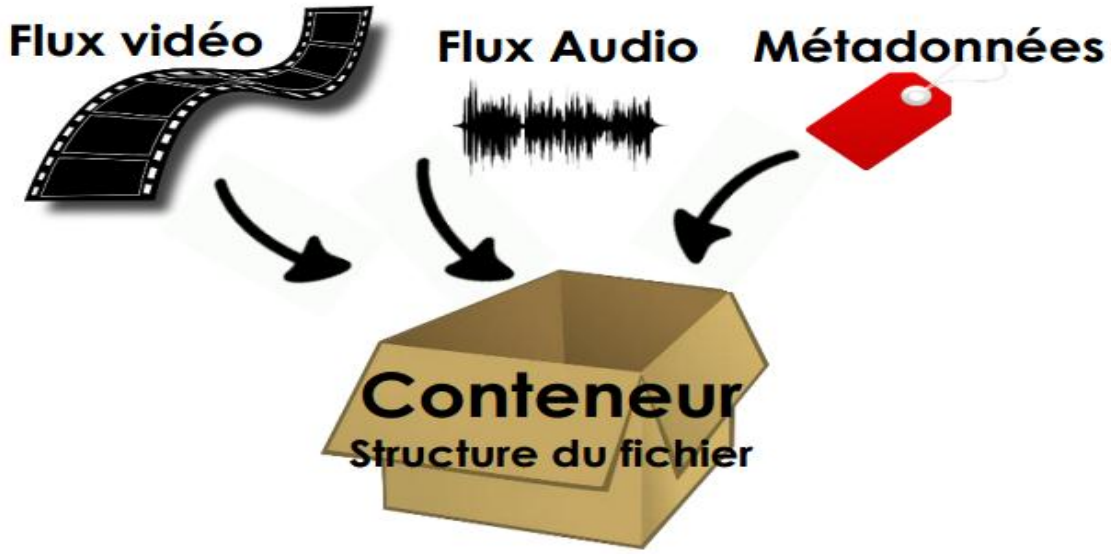


FIG. 1.1-Le conteneur qui structure du fichier.

Format	Conteneur
AVI	AVI
WMV	ASF
MP4	MP4
MPEG	PS

TAB. 1.1-Tableau des formats et ses conteneurs

1.1.2 Le concept du CoDec

On retrouve donc dans chaque conteneur les données audio et vidéo. Mais en amont, ces données doivent être encodées pour correspondre au format attendu en optimisant la compression avec perte de qualité minimum. C'est le rôle de CoDec, abréviation de « **Co**deur/**Dé**codeur ». Il propose une méthode pour encoder les signaux vidéo et audio selon un format attendu par le conteneur (figure 1.2). S'il l'on reprend l'image de notre conteneur en tant qu'une boite, le codec décrit la méthode pour ranger ou déballer correctement les différents objets composants la vidéo. L'efficacité d'un codec se mesure d'une part dans ses capacités de compression, mais aussi de décompression, c'est-à-dire à rétablir la vidéo lors de sa diffusion au plus près de sa qualité d'origine et dans un débit performant.

Ce qu'il faut comprendre c'est que les conteneurs les plus utilisés supportent différents CoDecs, et un CoDec ne peut pas être utilisé avec n'importe quel conteneur. Il y a une question de compatibilité (tableau 1.2).

FIG. 1.2-La méthode pour encoder/décoder des signaux vidéos.

Conteneur	CoDec
MP4	MPEG-4, XVID, H264.
AVI	MJPEG, DVIX, MPEG-4, XVID.

TAB. 1.2-La compatibilité des conteneurs avec les CoDec.

1.2 Les types des images et fonctionnement

1.2.1 Les images I

Les images de type I, sont des images codées en mode Intra-Frame. Ce sont des images de référence appelées également images clés (KeyFrame en anglais). Ce sont des images autonomes, c'est-à-dire qui peuvent être décodées sans références à d'autres images (Fig 1.3) [1]. Elles permettent d'assurer la cohésion de la séquence vidéo, comme elles permettent aussi de garantir la qualité résultante de la compression. La première image de la séquence vidéo est de type I, en général il y en a une ou deux par seconde [2].

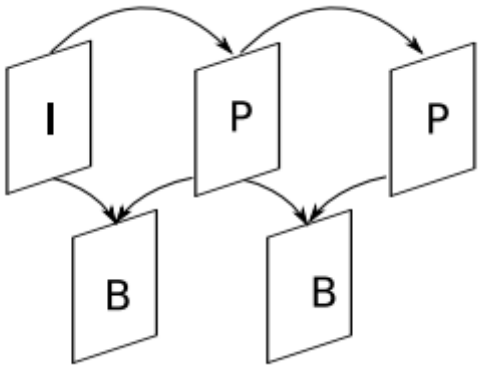


FIG. 1.3-Images de référence, prédites, et bi-prédites.

1.2.2 Les images P

Les images de type P, sont des images codées en mode Inter-Frame. Ce sont des images Prédictives qui fait référence aux parties (macroblocs[1]) des images I et/ou P antérieures pour le codage de l'image (figure 1.3). Une image P nécessite généralement moins de bits qu'une image I, mais elle peut être sensible aux erreurs en raison de la dépendance complexe vis-à-vis des images P et/ou I antérieures [2].

1.2.3 Les images B

Les images de type B, sont des images aussi codées en mode Inter-Frame. Ce sont des images Bi-prédictives ou images prédites bi-directionnellement, appelées aussi images prédites en arrière (backwards-predicted frames ne anglais) [1]. Les images B sont assez similaires aux images P, à la différence qu'elles peuvent être prédites à partir de deux images de référence, une antérieure et l'autre postérieure à l'image courante (figure 1.3), ce qui donne une meilleure compression, mais induit un retard au niveau de décodeur, puisque il doit décoder la prochaine image I ou P, afin d'être utilisées comme référence future par l'image B. le codage/décodage des images B est donc plus complexe et requiert des mémoires de grandes tailles [2].

1.3 La structure et la taille de GOP

1.3.1 La structure de GOP

Un GOP[2] débute toujours par une image I. Ensuite, plusieurs images P suivent à des intervalles réguliers. Dans les espaces entre deux images P ou entre une image P et une image I, une ou plusieurs images B soit intercalées. Certains codeurs vidéo permettent d'utiliser des GOP contenant plus d'une image I.

Plus le flux généré par un codeur contient des images codées en mode intra (I), plus il est éditable. Cependant, la taille des images codées en intra (en termes de bits) est plus importante que celle des images P ou B. Augmenter le nombre d'images I au sein du GOP aura donc pour conséquence l'augmentation de la taille de la vidéo encodée.

[1]macroblocs : ce sont des blocs de taille 16x16 pixels, c'est l'unité de base pour la prédiction en mode Intra-Frame.

[2]GOP : Group Of Picture.

Afin de limiter la bonde passante ou l'espace de stockage nécessaire, les vidéos pour la diffusion sur internet n'ont généralement qu'une seule image I par GOP.

1.3.2 La taille de GOP

La distance entre deux image I successives est appelée la taille du GOP. Les standards de codages utilisent généralement des GOP de taille entre 15 et 18, ce qui signifie qu'il y a une image I toutes les 14 ou 17 images (combinaison d'images P et B) (figure 1.4).

La structure de GOP est souvent indiquée par deux nombres, par exemple M=3 et N=12. Le premier indique la distance entre deux images d'ancrage références (I ou P), le second indique la distance entre deux images codées en intra (I), c'est la longueur du GOP. La structure du GOP de l'exemple où M=3 et N=12 est alors **IBBPBBPBBPBBP**. [2][3]

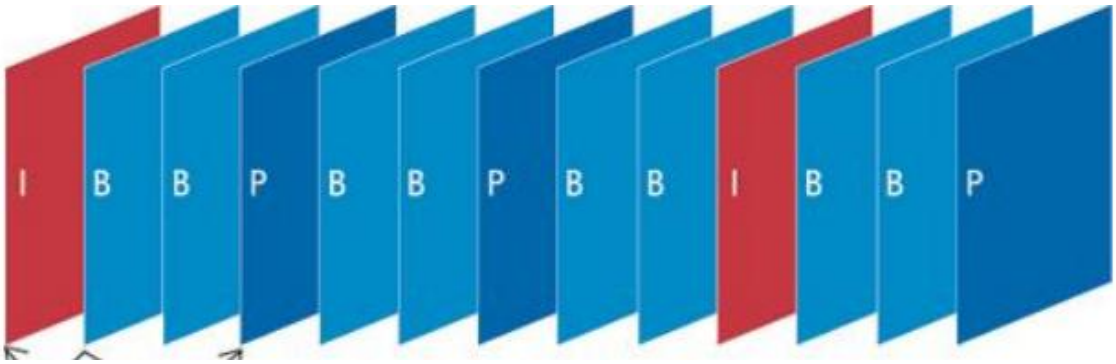


FIG. 1.4-Exemple de GOP où M=3 et N=12.

1.4 Espace de représentation des données images

1.4.1 Espaces couleurs

Pour travailler sur des images numériques, on discrétise ces images. On représente donc une image par une fonction à deux variables, les coordonnées, qui renvoient la couleur au point demandé de l'image. La représentation classique d'une image numérique utilise un espace de couleur dit RGB (figure 1.5). Cela correspond à une représentation discrète de l'image où l'on quantifie la couleur en chaque point par trois valeurs, Rouge, Vert, Bleu (RVB ou RBG en anglais). Cette représentation a cependant plusieurs défauts, elle utilise une grande quantité d'information et ne tient pas compte le fait que l'œil humain ne perçoit pas les couleurs en RVB mais au travers deux types de cellules, les une perçoivent la luminosité (noir et blanc), et les autres la coloration. [2]

Les CoDecs ne compressent donc pas les images RVB, mais utilisent un espace de couleurs plus approprié, de façon à mieux profiter du format de la vision humaine. Le format adopté est un format dit YCrCb (ou bien YUV) (figure 1.6). Y représente la luminosité (ou luma), U représente la première valeur de chrominance (ou chroma), V représente la deuxième valeur de chrominance (ou chroma). Il faut donc 3 composantes indépendantes pour pourvoir avoir un espace de couleur complet [2].

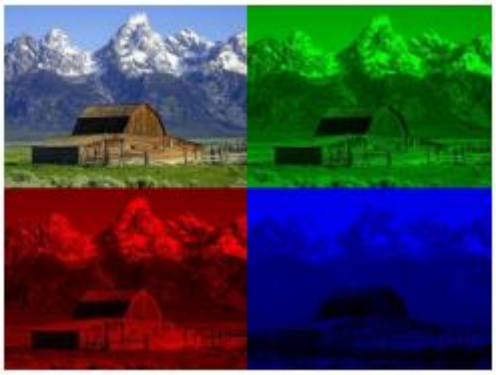


FIG. 1.5-Esapce couleur RGB.

FIG. 1.6-Espace couleur YCbCr.

Les formules permettant de passer d'un espace RVB a YUV (formule 1.1), et inversement (formule 1.2) [2] sont assez simples, et correspondent à des produits matriciels, en effet ce sont des changements de bases, puisque ce sont des espace vectoriels de domension3.

$$\begin{cases} Y = (0.257 * R) + (0.504 * G) + (0.098 * B) + 16 \\ Cr = (0.439 * R) - (0.368 * G) - (0.071 * B) + 128 \\ Cb = -(0.148 * R) - (0.291 * G) + (0.439 * B) + 128 \end{cases} \quad (1.1)$$

$$\begin{cases} B = 1.164 * (Y - 16) + 2.018(U - 128) \\ G = 1.164(Y - 16) - 0.813(V - 128) - 0.391(U - 128) \\ R = 1.164(Y - 16) + 1.596(V - 128) \end{cases} \quad (1.2)$$

1.4.2 Echantillonnage d'espace couleur YCrCb

L'utilisation du format YCrCb permet de profiter d'une caractéristique de l'œil humain, sa capacité de distinction des couleurs est plus faible que celle de distinction de la luminosité, puisque la composante Y porte plus d'information que les composantes Cr et Cb.

Cela permet de réduire la quantité d'information des espaces de chrominance par rapport à celle de luminosité, on fait donc ce qu'on appelle de sous-échantillonnage sur les composantes U et V selon un schéma présenté comme un rapport entre trois termes J[3] :a[4] :b[5]. Cela correspond en général à les réduire de moitié de résolution, et donc à diviser par 4 leurs tailles en mémoire, pour une perte de qualité assez faible.

On distingue plusieurs espaces YCrCb aux caractéristiques différentes, d'où la définition des nouveaux formats (figure 1.7):

- ✓ Le format 4 :2 :0, où les chrominances sont sous-échantillonnées par un facteur de deux horizontalement et verticalement.
- ✓ Le format 4 :1 :1, où les chrominances sont sous-échantillonnées d'un facteur deux verticalement et horizontalement.
- ✓ Le format 4 :2 :2 où le sous-échantillonnage d'un facteur deux se fait uniquement sur les colonnes de chaque chrominance.
- ✓ Le format 4 :4 :4 où les chrominances restent inchangées.

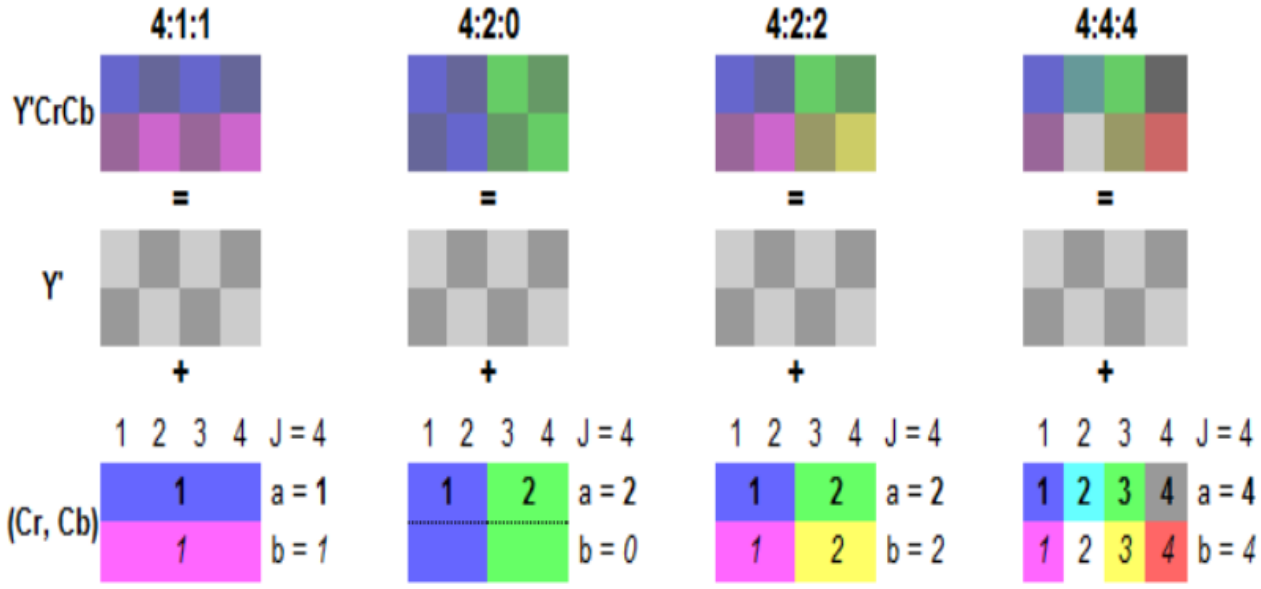


FIG. 1.7-Les nouveaux formats d'espace YCrCb.

[3]J : La taille horizontale du bloc d'échantillonnage de référence normalement 4.

[4]a : Le nombre d'échantillons de chrominance dans la première ligne de J pixels.

[5]b : Le nombre d'échantillons de chrominance additionnels dans la deuxième ligne de J pixels.

1.5 La prédiction en mode Intra-Frame

La prédiction intra n'utilise pas d'image de référence. Elle exploite la redondance spatiale d'une image (image I), c'est-à-dire, le fait que les pixels dans une image sont spatialement corrélés. Des codeurs comme H.264 à la particularité dans le domaine spatial, se référant aux échantillons voisins de blocs déjà codés et non dans le domaine transformé (voir section 1.7). La prédiction dépend bien sûr de la taille de la partition considérée. Il existe donc plusieurs méthodes en fonction du block :

- ✓ Prédiction intra de blocs de luminance de taille 16x16.
- ✓ Prédiction intra de blocs de luminance de taille 4x4.
- ✓ Prédiction intra de blocs de chrominance de taille 8x8.

1.5.1 Prédiction intra de blocs 16x16 de luminance

Elle utilise des larges blocs de taille 16x16, ce qui donne une faible précision de prédiction, mais moins de bits seront nécessaires pour coder la prédiction. La prédiction est créée directement à partir des échantillons de la gauche ou au-dessous ou bien une combinaison de ceux-ci (figure 1.8) [3]. Quatre modes sont alors possibles, comme le montre la figure 1.9. Le tableau 1.3 précise les calculs réalisés.

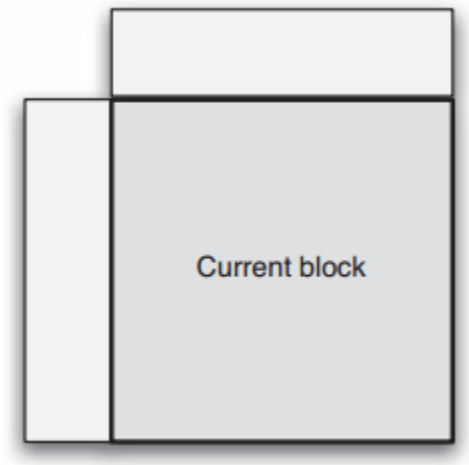


FIG. 1.8-La source des échantillons pour intra-prédiction des blocs de 16x16.

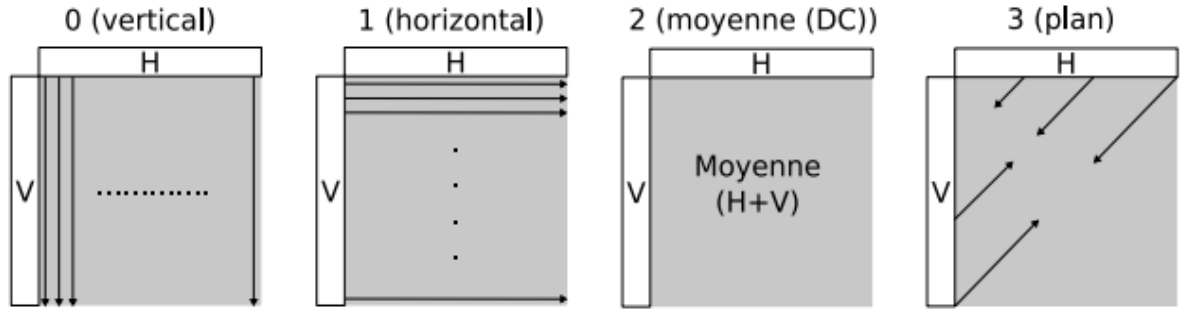


FIG. 1.9-Les quatre modes d'intra-prédiction des blocs de 16x16.

Mode	Description
Mode 0 (vertical)	Extrapolation à partir des échantillons supérieurs (H).
Mode 1 (horizontal)	Extrapolation à partir des échantillons gauches (V).
Mode 2 (DC)	Moyenne de H et V.
Mode 3 (plan)	Une fonction linéaire plane est ajustée à H et V. cela fonction bien sur les zones de luminance variant doucement.

TAB. 1.3-La description des quatre modes d'intra-prédiction des blocs de 16x16.

1.5.2 Prédiction intra de blocs 4x4 de luminance

La prédiction de blocs 4x4 de luminance est une spécificité de H.264 par rapport aux autres standards qui utilisent des blocs 8x8. Cela apporte une meilleure précision dans la prédiction mais génère et nécessite plus de calculs. Neuf modes de prédiction sont possibles pour les blocs 4x4 selon la direction de prédiction. La figure 1.10 présente la source des échantillons pour intra prédiction. Sur la figure 1.11 on a les neuf modes, les flèches indiquent la direction de prédiction. Pour les modes 3 à 8, les pixels prédits sont formés par une moyenne pondérée des pixels [A-M] (figure 1.12). Le tableau 1.4 précise la description de chaque mode [3].

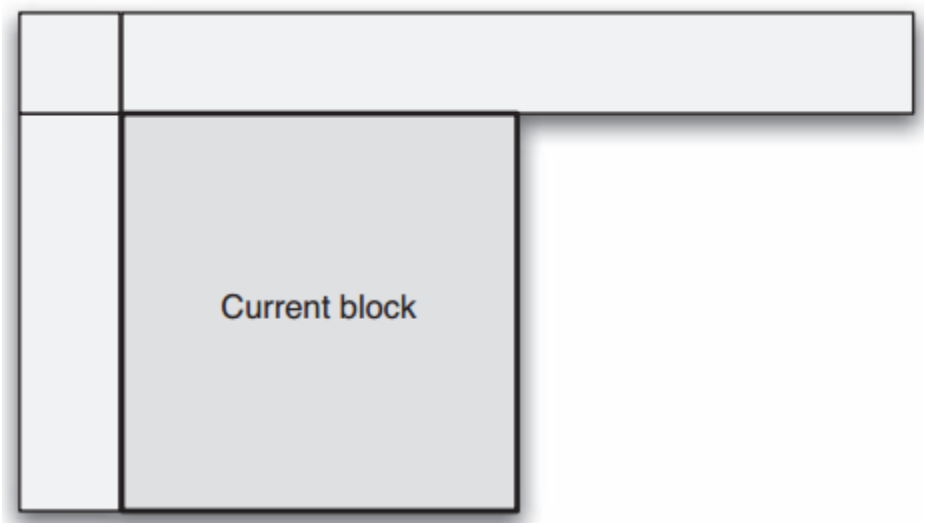


FIG. 1.10- La source des échantillons pour intra-prédiction des blocs de 4x4.

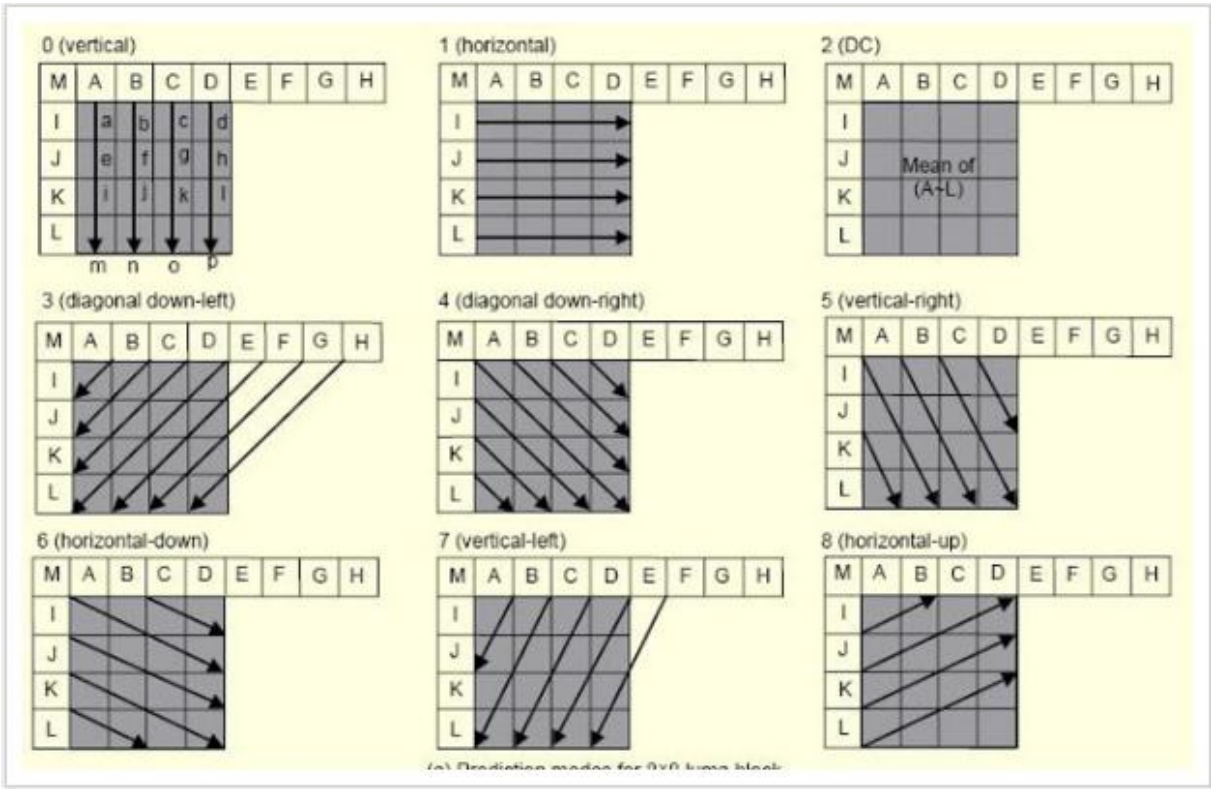


FIG. 1.11-Les neuf modes d'intra-prédictions des blocs de 4x4.

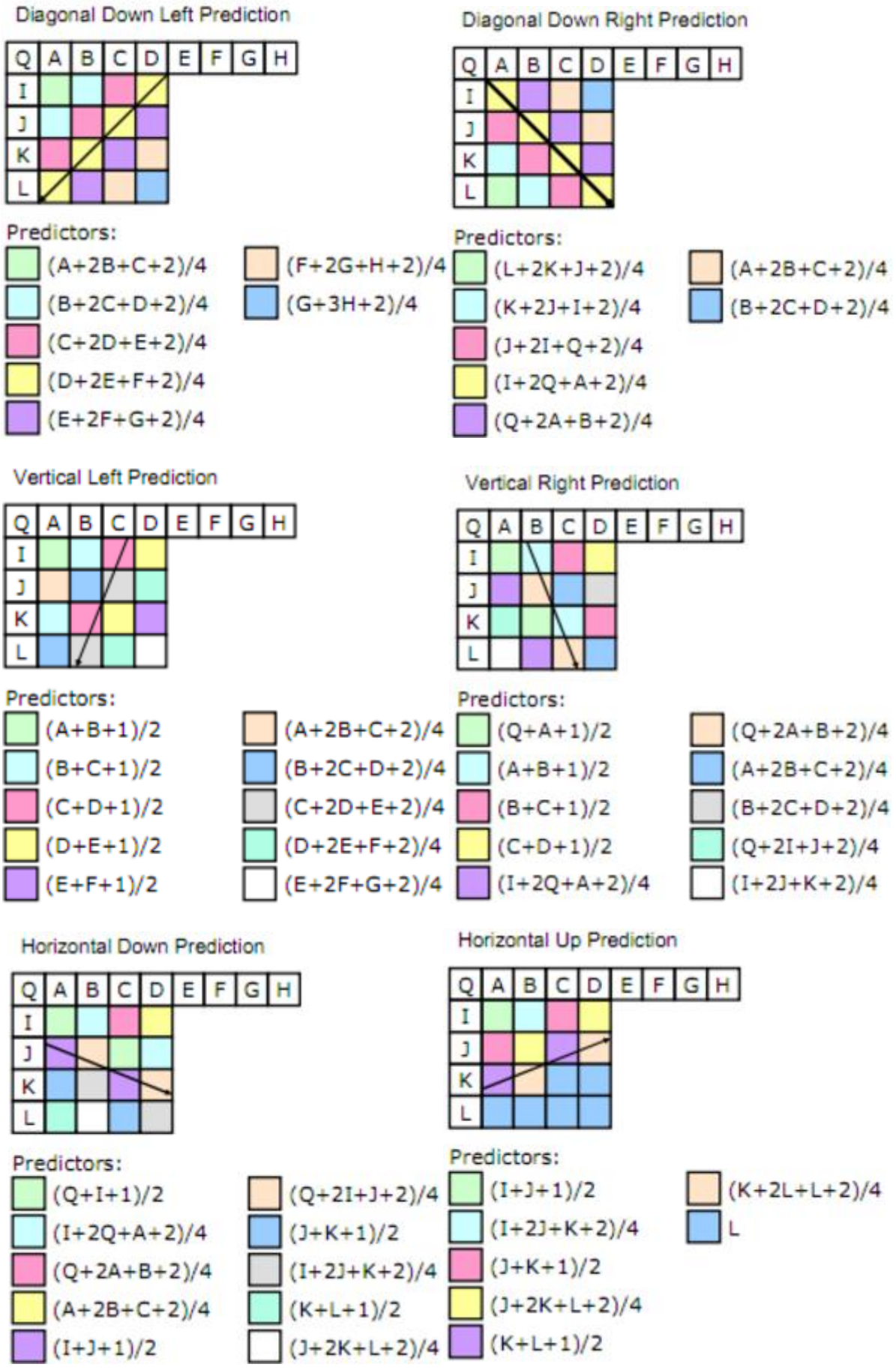


FIG. 1.12-Le calcule de la moyenne pondérée des pixels [A-M] pour les modes de 3 à 8.

Mode	Description
Mode 0 (vertical)	Les échantillons supérieurs A, B, C et D sont extrapolés verticalement.
Mode 1 (horizontal)	Les échantillons gauches I, J, K et L sont extrapolés horizontalement.
Mode 2 (DC)	Tous les échantillons sont prédits par la moyenne de [A-D] et [I-L].
Mode 3 (diagonale bas-gauche)	Les échantillons sont interpolés avec angle de 45entre le bas-gauche et le haut-droit.
Mode 4 (diagonale bas-droite)	Les échantillons sont interpolés avec un angle de 45° vers le bas et vers la droite.
Mode 5 (vertical-droit)	Extrapolation avec un angle de 26,6° à gauche de la verticale (*largeur*/*hauteur*=1/2)
Mode 6 (horizontal-bas)	Extrapolation avec un angle de 26,6° en-dessous de l'horizontale.
Mode 7 (vertical-gauche)	Extrapolation (ou interpolation) avec un angle de 26,6° à droite de la verticale.
Mode 8 (horizontal-haut)	Interpolation avec un angle de 26,6° au-dessus de l'horizontale.

TAB. 1.4- La description des neufs modes d'intra-prédiction des blocs de 4x4.

1.5.3 Prédiction intra de blocs de 8x8 de chrominance

Chaque bloc de 8x8 de chrominance est prédit à partir des échantillons de chrominance déjà codés au-dessus et à gauche. Les deux composantes Cb et Cr utilisent toujours la même prédiction. Quatre modes de prédiction existent, similaires aux modes 16x16 de luminance (voir la figure 1.9). Le tableau 1.5 [3] présente ces différents modes.

Mode	Description
Mode 0 (DC)	Moyenne de H et V.
Mode 1 (horizontal)	Extrapolation à partir des échantillons gauches (V).
Mode 2 (vertical)	Extrapolation à partir des échantillons supérieurs (H).
Mode 3 (plan)	Une fonction linéaire plane est ajustée à H et V.

TAB. 1.5-La description des quatre modes d'intra-prédiction des blocs de 4x4 de chrominance.

En pratique, on ne code pas le bloc prédit du bloc courant, mais plutôt son résiduel, qui est le résultat de la soustraction de bloc courant et le bloc prédit en intra (figures 1.13, 1.14, 1.15). Donc lors de codage en mode intra on suit les étapes suivantes :

- ✓ Calculer le bloc prédit du bloc courant à partir des blocs déjà codés pour tous les modes.
- ✓ Choisir le mode le plus optimal, celui qui minimise le SAE[6].
- ✓ Calculer le bloc résiduel, en effectuant une soustraction entre le bloc courant et le bloc prédit.
- ✓ Le bloc résiduel qui sera codé, va être envoyé sous forme d'un flux binaire.

Pour le décodage, on reconstruit le bloc courant suivant les étapes suivantes :

- ✓ Décodant le flux binaire, on trouve le bloc résiduel.
- ✓ On calcule le bloc prédit à partir des blocs déjà décodés, en utilisant le même mode utilisé dans le codage.
- ✓ On reconstruit le bloc courant, en calculant la somme entre le bloc résiduel et le bloc prédit.

FIG. 1.13-image courante.

FIG. 1.14-image prédite en mode intra

FIG. 1.15-image résiduel.

[6]SAE: the Sum of Absolute Errors, ou SAD: the sum of absolute Difference.

1.6 La prédiction en mode Inter-Frame

Inter-prédiction est le processus de prédiction des blocs d'échantillons de luminances et chrominances à partir des images de références qui ont été préalablement codées et transmises. Cela implique la recherche et la sélection dans une région de prédiction (voir la section 1.6.1), la génération d'un bloc de prédiction et le soustrayant du bloc origine, afin de former le bloc résiduel, qui sera ensuite codé et transmit [3].

L'image de référence est choisie parmi une liste d'images de références déjà codée et stockée dans un DPB[7] qui contient des images antérieures, postérieures de l'image courante. Les images dans DPB sont indexées, à savoir qu'elles sont listées dans un ordre bien particulier comme il est montré dans le tableau 1.6. Donc la prédiction peut être générer à partir d'une région de prédiction dans une unique image de référence, pour macrobloc P ou B, ou bien à partir de deux régions de prédiction dans des images de références, pour un macrobloc B. Le décalage entre la position de partition courante et la région de prédiction dans une image de référence, est un vecteur de mouvement (voir la section 1.6.2).

La liste	Description
Liste 0 (P)	Une liste unique de toutes les images de référence. Par défaut, la première image dans la liste est l'image la plus récemment décodée.
Liste 0 (B)	Une liste de toutes les images de référence. Par défaut, la première image dans la liste est l'image actuelle avant l'image actuelle dans l'ordre d'affichage.
Liste 1 (B)	Une liste de toutes les images de référence. Par défaut, la première image dans la liste est l'image actuelle dans l'ordre d'affichage.

TAB. 1.6-les différentes listes d'images de références.

1.6.1 Estimation du mouvement

Estimation du mouvement d'un macrobloc, consiste à trouver une région de 16x16 dans une image de référence (figure 1.16)[3], qui correspond étroitement au macrobloc courant. Le macrobloc k6de référence est déjà codé, qui peuk6t être situé soit dans une image antérieure ou bien postérieure. Le macrobloc de référence qui sera sélectionné, c'est celui qui minimise le critère de correspondance (par exemple SAD).

[7] DPB : Decoded Picture Buffer.

Soit I_t l'image à l'instant **t** et I'_{t-1} l'image référence. Il s'agit, pour tout bloc B_t de taille HxL de I_t , de rechercher dans I'_{t-1} le bloc B'_{t-1} le plus proche. Cette proximité est exprimée via le calcul d'une distance, la plus classique étant la SAD (Sum of Absolute Difference) (formule 1.3) [3]:

$$SAD_{(B_t(x,y),B'_{t-1}(x,y))}(u,v) = \sum_{j=0}^{H-1}\sum_{i=0}^{L-1} |B_t(x+i,y+j) - B'_{t-1}(x+u+i,y+v+j)| \quad (1.3)$$

On cherche donc le déplacement au pixel prés minimisant la distance (formule 1.4) :

$$(u',v') = min_{(u,v)\varepsilon F} SAD(B_t(x,y), B'_{t-1}(x+u,y+v)) \quad (1.4)$$

Il existe plusieurs méthodes de recherche du bloc référence, qui correspond au bloc courant :

- ✓ Estimation par la recherche brute.
- ✓ Estimation par la recherche en diamant.
- ✓ Estimation par la recherche en hexagonal.
- ✓ Estimation par la recherche hiérarchique.

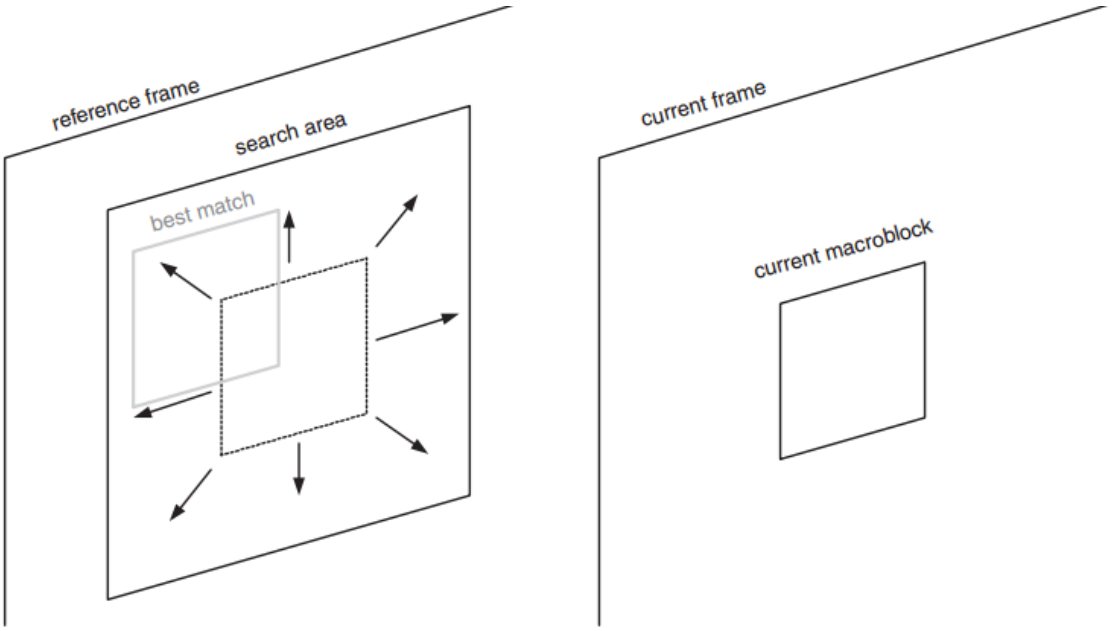


FIG. 1.16-Estimation du mouvement.

1.6.2 Compensation du mouvement

Les échantillons de luminances et chrominances qui correspondent à la région sélectionnée dans l'image de référence, sont soustrait du macrobloc courant pour produire un macrobloc résiduel, qui sera codé et transmis avec un vecteur de mouvement (figure 1.17) [3], qui décrit la position de la meilleure région de prédiction par rapport à la position du macrobloc courant(figure 1.18) [3].

Le bloc $\mathbf{B'_{t-1}}$ choisi devient le prédicteur du bloc courant de taille **L**x**H**. On opère alors une soustraction entre ces deux blocs pour obtenir un bloc résiduel (formule 1.5) (figure 1.18).

$$\hat{B}(x,y) = B_t(x,y) - B'_{t-1}(x+u', y+v') \tag{1.5}$$

Le bloc résiduel est ensuite codé et transmis, ainsi que le vecteur de mouvement correspondant (u', v'). Dans le coté de décodeur, il recrée le bloc prédicteur d'après le vecteur de mouvement (u', v') qu'il a reçu, en suite il décode le bloc résiduel afin de l'ajouter au prédicteur et reconstruit une version du bloc original (formule 1.6):

$$B'(x,y) = B'_{t-1}(x+u', y+v') + \hat{B}_t(x,y) \tag{1.6}$$

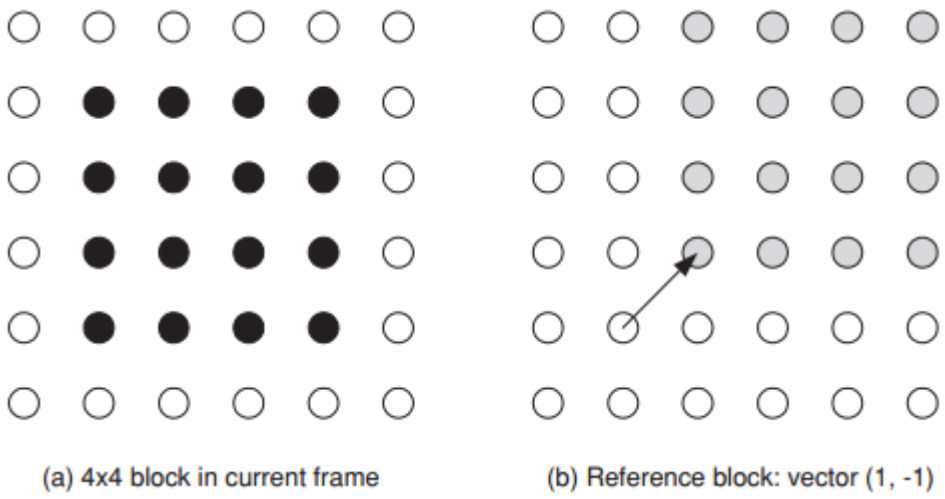
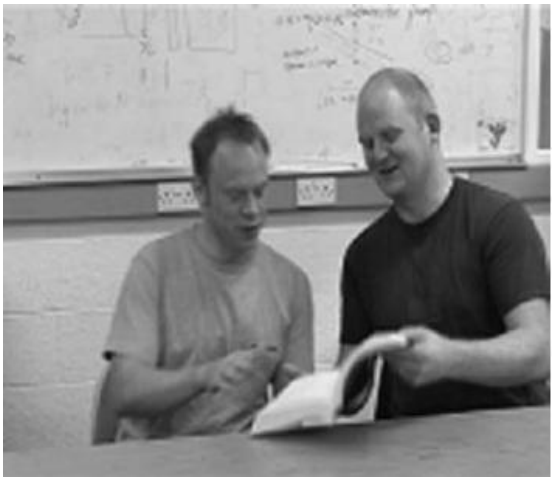


FIG. 1.17-Exemple de vecteur du mouvement.



(a)Image de référence(t-1).

(b) image courante(t).

(c)résiduel sans compensation.

(d) résiduel avec compensation.

FIG. 1.18-Estimation et compensation du mouvement.

1.7 Transformation, Quantification, Codage Entropique

Les images naturelles sont souvent difficiles à compresser dans leur état original en raison de la forte corrélation entre les échantillons voisins. L'estimation et la compensation de mouvement permettent de réduire la corrélation temporelle locale dans l'image résiduelle et ainsi de faciliter la compression de l'image originale. L'objectif du bloc de traitement suivant, est d'exploiter d'avantage la corrélation résiduelle spatiale d'une image ou des données résiduelles pour obtenir une forme efficacement compressible par le codeur entropique. Classiquement ce bloc est constitué d'une transformation qui sera suivie d'une quantification, ensuite du codage entropique (figure 1.19).

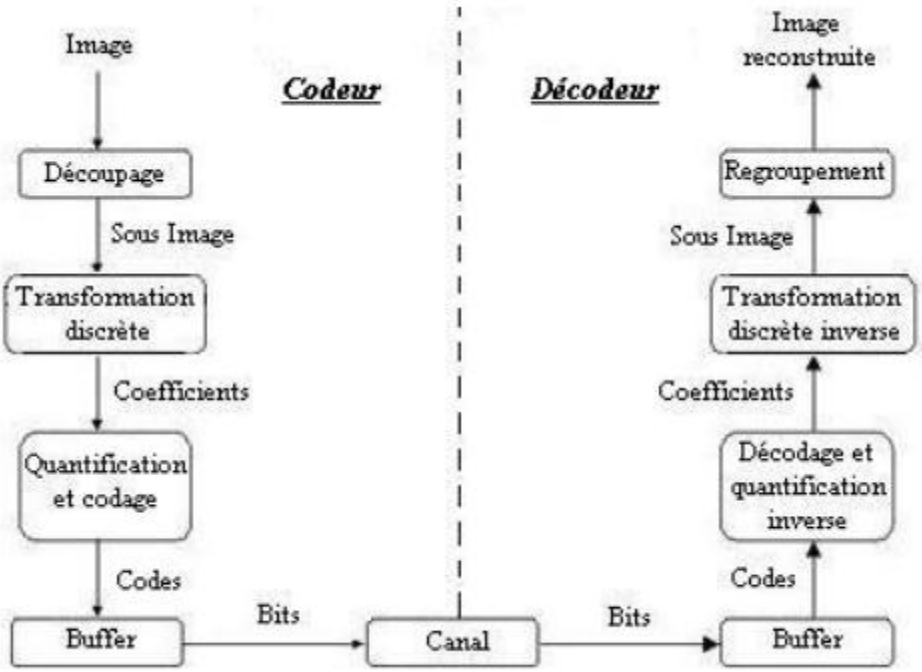


FIG. 1.19-Le processus de transformation, quantification, codage.

1.7.1 Transformation

Dans la transformation, l'image de dimension NxN est subdivisée en sous images ou blocs de taille réduite, car la quantité de calcul demandée pour effectuer la transformation sur l'image entière, est très élevée. Chaque bloc subit une transformation mathématique orthogonale inversible linéaire du domaine spatial vers le domaine fréquentiel, indépendamment des autres blocs (transformée en un ensemble de coefficients plus ou moins indépendants) [1] [3]. Parmi les transformations linéaires existantes :

- ✓ Transformation de Karhunen-Loeve (KLT).
- ✓ Transformation de Fourrier discrète (DFT).
- ✓ Transformation de Hadamard (HT).
- ✓ Transformation en ondelettes (WT).
- ✓ Transformation en cosinus discrète (DCT).

Par la suite on va s'intéresser à la transformation en cosinus discrète (DCT).

La DCT, est une transformation mathématique qui transforme un ensemble de données d'un domaine spatial en un spectre de fréquence et inversement (IDCF). C'est la plus utilisée parmi les transformations citées [1].

Elle permet schématiquement de changer l'échelle de mesure, en passant d'une échelle définissant un pixel en fonction de sa position en x et en y à une échelle définissant la fréquence d'apparition de ce pixel dans un bloc de pixels, en effet, il est possible de supprimer des informations sans altérer le résultat final, contrairement à un bloc de pixel où la disparition brute de plusieurs éléments est immédiatement visible.

La DCT est effectuée sur une matrice carrée NxN de valeurs de pixels et donne une matrice carrée NxN de coefficients de fréquence. Le temps de calcul requis pour chaque élément dans la DCT dépend de la taille de la matrice.

Vu la difficulté d'appliquer la DCT sur la matrice entière, celle-ci est décomposée en blocs de taille 8x8 pixels.

A la sortie de la matrice de la DCT, la valeur de la position (0,0) est appelée le coefficient continu (DC[8]), cette valeur représente une moyenne de la grandeur d'ensemble de la matrice d'entrée, ce coefficient est plus grand d'un ordre de grandeur à toute valeur dans la matrice de la DCT, par convention, les 64 valeurs transformées (de chaque bloc) sont positionnées d'une certaine manière, ainsi la valeur moyenne de tous ces coefficients est placée en haut à gauche de ce bloc. Plus on s'éloigne des coefficients continus, plus leurs grandeurs diminuent. Ce qui signifie que la DCT concentre la représentation de l'image en haut à gauche de la matrice de sortie. Les coefficients en bas et à droite de cette matrice contiennent moins d'information utile (figure 1.20) [1].

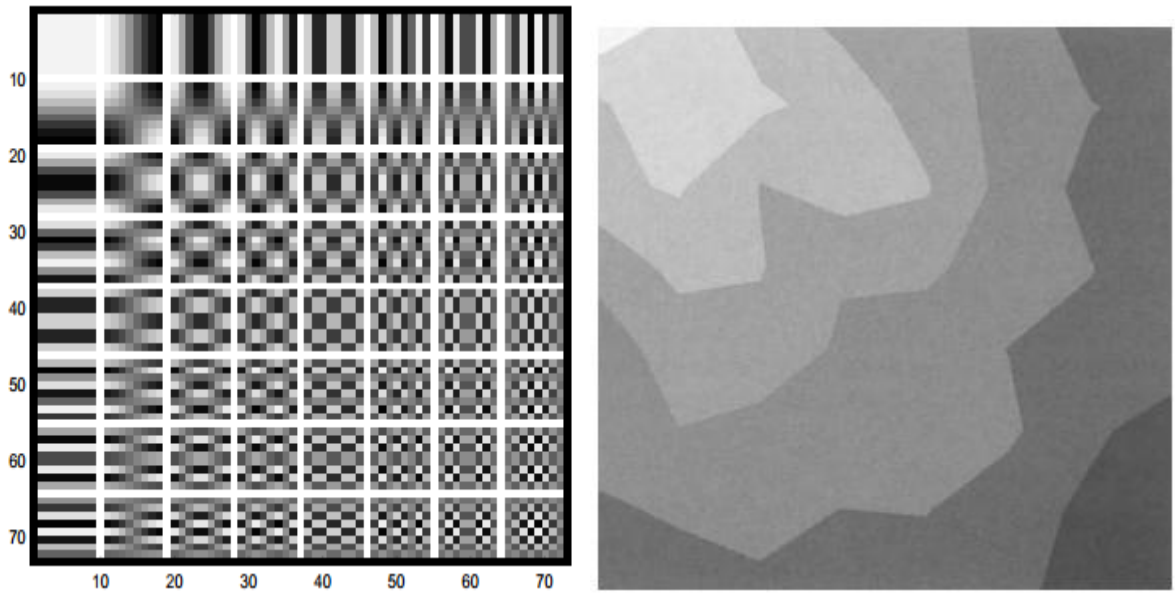


FIG. 1.20-La transformation en DCT des blocs de taille 8x8.

[8]DC c'est le pixel d'indice (0,0), et le reste on les appelés AC coefficient.

Transformer DCT direct : pour u,v=0,1 ,..,7.

$$S(u,v) = \frac{C(u)}{2}.\frac{C(v)}{2}\sum_{y=0}^{7}\sum_{x=0}^{7} s(x,y)cos\frac{(2x+1)u\pi}{16}cos\frac{(2y+1)v\pi}{16} \quad (1.7)$$

Transformation DCT inverse (IDCT) : pour x,y=0,1,..,7.

$$s(x,y) = \sum_{v=0}^{7}\frac{C(v))}{2}\sum_{u=0}^{7}\frac{C(u)}{2}S(u,v)cos\frac{(2x+1)u\pi}{16}cos\frac{(2y+1)v\pi}{16} \quad (1.8)$$

Ou s(x,y) : les échantillons d'entrées et S(u,v) : les coefficients de la DCT obtenu.

$$C(u) = C(v) = \begin{cases} 1 & \text{si } v,u > 0, \\ \frac{1}{\sqrt{2}} & \text{si } u = v = 0 \end{cases} \quad (1.9)$$

1.7.2 Quantification

La quantification est utilisée pour réduire la précision des données d'une image après l'étape de transformation en supprimant les valeurs insignifiant (exemple : les coefficients de la DCT proches de zéro). Au sein d'un codeur vidéo ou d'images fixes, la quantification a donc pour but de mettre à zéro les coefficients insignifiants tout en conservant un nombre réduit de coefficients significatifs non nuls. La sortie d'un quantificateur est typiquement un tableau clairsemé de coefficients quantifiés, contenant principalement des zéros.

Formellement, un quantificateur transforme chaque échantillons d'un signal Y en un échantillon discret quantifié Z ayant une gamme plus réduite de valeurs. Il est ainsi possible d'encoder le signal quantifié avec moins de bits. En compression vidéo, les coefficients issus de la transformation sont quantifiés. Un quantificateur fait correspondre à chaque échantillon du signal d'entrée une valeur quantifiée, et un quantificateur vectoriel fait correspondre directement à un groupe d'échantillons (un vecteur) un groupe de valeurs quantifiées. On obtient bien la compression mais cela implique dans les deux cas une dégradation irréversible du signal. On parle ici de distorsions du signal (ou d'erreur de quantification).

1.7.2.1 Quantification scalaire

Un exemple simple de quantification scalaire d'arrondi d'un nombre décimal à l'entier le plus proche, la fonction se fait du domaine $\mathbb{R}$ vers le domaine $\mathbb{Z}$. c'est une méthode de compression avec pertes (non réversible) puisqu'il n'est plus possible de déterminer la valeur exacte du nombre réel original à partir de l'entier arrondi [1].

Un exemple plus général d'une quantification uniforme est donné par :

$$Z = arrondi(\frac{F}{QP}) \qquad F' = Z.QP \tag{1 .10}$$

Où QP est le pas de quantification, F le nombre à quantifié, Z est la valeur quantifiée et F' la valeur reconstruite. Les intervalles des valeurs quantifiées en sortie sont espacés de façon uniforme.

1.7.2.2 Quantification vectorielle

La quantification vectorielle, par opposition, consiste en une représentation d'un groupe d'échantillons par un vecteur plutôt que par un scalaire. L'espace des échantillons est tout d'abord partitionné en cellules. Pour chaque cellule obtenue, un vecteur de représentation est choisi et indexé dans un dictionnaire. Ainsi c'est l'indice du vecteur de représentation qui est codé et non le vecteur lui-même. La quantification vectorielle est efficace pour le codage audio, elle est cependant moins adaptée à la compression de séquence vidéo. En effet, la diversité importante des séquences vidéo naturelles La mise en œuvre de ce dictionnaire ne permet pas d'obtenir un dictionnaire de vecteurs optimal pour la compression vidéo, et la mise en œuvre de ce dictionnaire n'étant pas forcément évidente et aussi la nécessité de sa transmission au décodeur, alors la quantification scalaire qui a été retenue dans les principaux standards de compression malgré une plus faible distorsion à niveau de représentation égaux [1].

1.7.3 Codage Entropique

Le codeur entropique convertit une série de symboles représentant les éléments de la séquence vidéo en un flux binaire compressé approprié pour la transmission ou le stockage. Les symboles présents en entrée du codeur entropique peuvent représenter les coefficients transformés, quantifiés et réordonnées, les vecteurs de mouvements (un vecteur de déplacement pour chaque bloc compensé en mouvement), etc. Donc le codage se fait en parcourant les éléments dans l'ordre imposé par une séquence appelée Séquence Zigzag (figure 1.21). Les éléments sont parcourus en commençant par les basses fréquences puis ensuite en traitant les fréquences de plus en plus élevées. Étant donné qu'il y a beaucoup de composantes de hautes fréquences qui sont nulles, la séquence zigzag engendre de longues suites de 0 consécutifs. D'une part, les suites de valeurs nulles sont simplement codées en donnant le nombre de 0 successifs. D'autre part, les valeurs non nulles seront codées. Il existe pour cela des méthodes de codages :

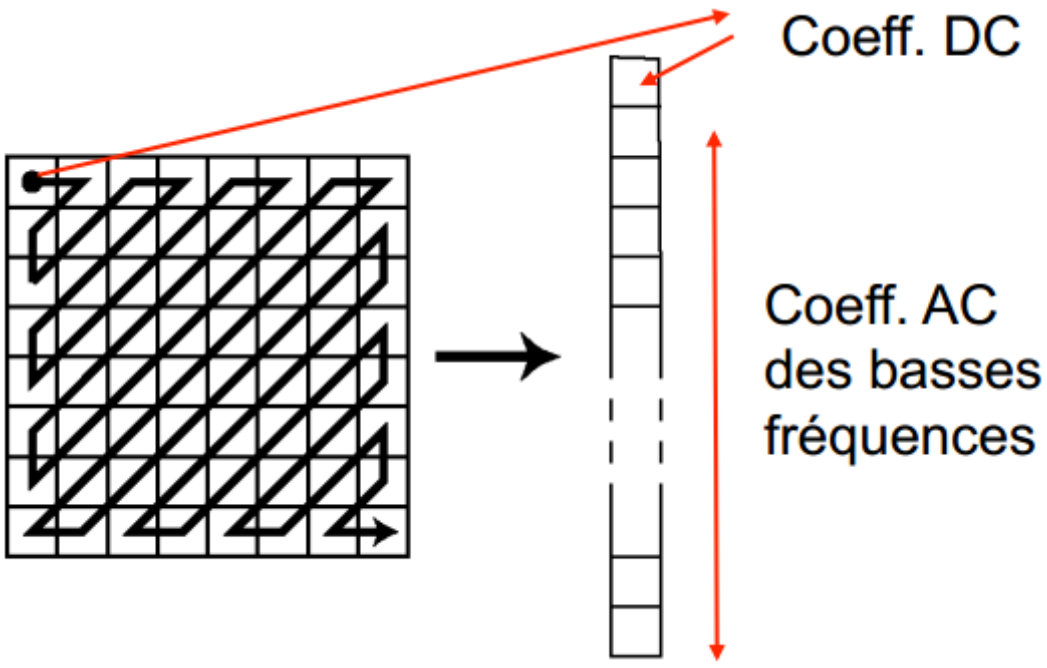


FIG. 1.20-La lecture des coefficients de la matrice DCT en Zigzag.

1.7.3.1 Run length Encoding (RLE)

C'est un codage par plage, qui est une méthode simple et efficace pour compresser des sources comprenant des répétitions successives importantes de symboles identiques. En effet, au lieu de coder N fois de façon indépendante un symbole S, seul le couple (N, S) est codé. Par exemple la chaine symboles suivante : **AAAAAABBBBCCDEDCCCBBA**, sera codée comme suit : **6A4B2CAD1E1D3D2B1A**. Ce procédé est très utilisé en compression d'images, notamment sur les coefficients DCT nuls en sortie de quantification et dans les formats Bitmap ou encore PCX.

1.7.3.2 Ziv, Lempel et Welsh (LZW)

C'est une technique de codage de symboles par blocs distincts. Ces blocs sont représentés par un ensemble de mots construits dynamiquement pendant l'encodage et stockés au fur et à mesure dans un dictionnaire. Ce dictionnaire est également reconstruit dynamiquement au décodeur, ce qui présente l'avantage de ne pas avoir à le transmettre. Le codage d'un symbole se déroule en deux phases : le codage à proprement parler et la mise à jour du dictionnaire (figure 1.21).

- ✓ Pour le codage, on recherche dans le dictionnaire le mot le plus long permettant de représenter les prochains symboles de la source. L'index de ce mot est alors transmis.
- ✓ Pour la mise à jour du dictionnaire, on forme un nouveau mot avec le symbole venant d'être codé auquel on adjoint le prochain symbole. Ce nouveau mot est alors stocké dans le dictionnaire.

Suite de symboles à coder : aaabdbccbbcaabcc

Symbole en entrée	Dictionnaire		Indice à transmettre
	index	code associé	
	0	a	
	1	b	
	2	c	
	3	d	
a	4	aa	0
a a	5	aab	4
b	6	bd	1
d	7	db	3
b	8	bc	1
c	9	cc	2
c	10	cb	2
b	11	bc	1
b c	12	bca	11
a a b	13	aabc	5
c c	12		9

FIG. 1.21-Exemple de codage de LZW.

1.7.3.3 Codage à longueur variable

Un codeur à longueur variable transforme les symboles d'entrée en une série de mots de code (VLC en anglais), ceux-ci sont donc de longueur variable mais contiennent un nombre entier de bits. Les symboles ayant une fréquence d'occurrence importante sont représentés avec des mots de code de faible taille, tandis que les symboles moins fréquents sont représentés avec des mots de code de taille plus importante. Pour un nombre suffisamment grand de symbole, cela conduit à la compression moyenne des données.

1.7.3.3.1 Codage de Huffman

Le codage de type Huffman assigne un code de longueur variable à chaque symbole. D'après la méthode originale proposée par Huffman en 1952 [1], alors l'algorithme suit trois étapes :

- ✓ Les symboles à coder, sont classés par fréquence décroissante.
- ✓ Ensuite, tant qu'il reste plus d'un élément à coder, les deux éléments de plus faible fréquence sont regroupés pour former un nouveau symbole dont la fréquence est la somme de leurs fréquences respectives. Un bit est assigné à chaque symbole fusionné, « 1 » pour celui qui a la fréquence la plus élevée, « 0 » pour l'autre. Ce nouveau symbole est alors inséré dans la liste des éléments à coder en fonction de sa fréquence et le processus est réitéré.
- ✓ Lorsque tous les symboles ont été traités, on parcourt de la droite vers la gauche les bits assignés aux différents éléments pour obtenir leurs représentants (figure 1.22).

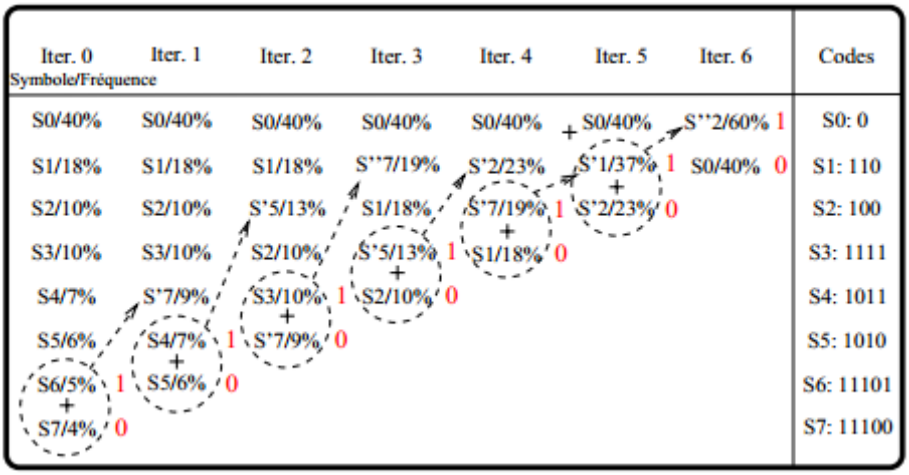


FIG. 1.22-exemple de codage de Huffman.

1.7.3.3.2 Codage Arithmétique

Le codage arithmétique amène une alternative pratique au codage de type Huffman et permet de se rapprocher des taux théoriques maximum de compression. Un codeur arithmétique convertit une séquence de symboles de données en un simple nombre décimal et s'approche du nombre décimal optimal de bits requis pour représenter chaque symbole [1]. Le codeur procède de la façon suivante :

- ✓ Tout d'abord, le processus est initialisé en assignant à chaque symbole, un intervalle compris entre 0 et 1 de longueur égale à la probabilité $\mathbf{P_i}$. l'intervalle correspondant au premier symbole présent est sélectionné comme intervalle courant.
- ✓ Ensuite, cet intervalle est divisé en **L** sous-intervalles de longueurs $\mathbf{L_i * P_i}$ **où** $\mathbf{L_i}$ correspondant à la longueur de l'intervalle courant.
- ✓ Ce processus est réitéré pour chaque symbole à coder jusqu'au dernier (figure 1.23). Au final, on obtient un intervalle par symbole dont la borne inférieure, est exprimée en un rationnel binaire, qui sert de représentation au symbole en question. Evidemment, la longueur des intervalles étant proportionnelle aux probabilités des symboles, des problèmes de précision se posent sur leurs représentations.

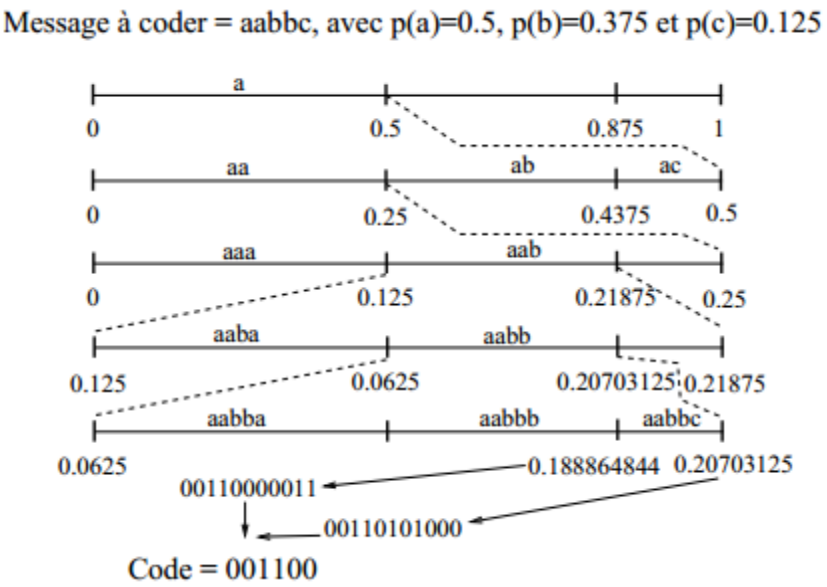


FIG. 1.23-Principe de codage arithmétique.

1.7.3.4 Context-based Adaptive Variable Length Coding (CAVLC)

Le codage entropique CAVLC (Context-based Adaptive Variable Length Coding) est utilisé pour le codage des coefficients de transformée quantifiés [1] [3]. Ce codage entropique parcourt les blocs 4x4 en zigzag et tire avantage des blocs 4x4 quantifiés :

- ✓ Après prédiction, transformation et quantification, les blocs contiennent principalement des éléments nuls, CAVLC les représente de maniéré compacte.
- ✓ Après le parcours en zigzag, les coefficients non nuls et les coefficients nuls, sont souvent des séquences de 1 et -1. CAVLC signale le nombre de ces coefficients de manière compacte.
- ✓ Le nombre des coefficients non nuls dans les blocs voisins est corrélé. Le nombre de coefficients est codé avec une table de correspondance.
- ✓ L'amplitude des coefficients non nuls tend à être plus grande au début du réarrangé (après des coefficients de la composante continue). CAVLC en tire avantage en adaptant le choix des tables de correspondances (LUT[9]) en fonction des amplitudes déjà codées.

1.7.3.5 Context-Adaptive Binary Arithmetic Coding (CABAC)

La méthode CABAC (context-Adaptive Binary Arithmetic Codign) disponible dans le profil principal, améliore encore le codage entropique [1] [3]. Le procédé de codage applique les étapes suivantes :

1. Binarisation : CABAC utilise un codage arithmétique binaire, c'est-à-dire, ne codant que les éléments binaires (0 ou 1). Les symboles non binaires sont convertis en code binaire avant le codage arithmétique. Ce procédé est semblable au procédé de transformation d'un symbole en un mot de code de longueur variable, mais le code binaire est ensuite codé par le codeur arithmétique avant d'être transmis.

Les étapes 2,3 et 4 sont répétées pour chaque bit ou bin du symbole converti en binaire.

2. Sélection du modèle de contexte : c'est un modèle de probabilité pour un ou plusieurs bins du symbole convertis en binaires. Ce modèle est choisi parmi une sélection de modèles disponibles en fonction des statistiques de symboles récemment codés. Il enregistre la probabilité pour chaque bin du symbole converti en binaire.
3. Codage arithmétique : il code chaque bin en fonction du modèle de probabilité sélectionné.

[9]LUT : Look Up Table.

4. Mise à jour de la probabilité : le modèle de contexte sélectionné est mis à jour par rapport à la valeur courante codée (si la valeur du bin étant un 1 alors le compteur de 1 est incrémenté).

Par rapport au CAVLC, CABAC garantit en général une réduction supplémentaire du débit binaire de 10 à 15% lors du codage de signaux télévisuels pour une même qualité.

Le codage entropique, dernier élément du codeur avant la génération du flux binaire est un élément important dans le codeur. La compression peut être efficace et permet une dernière optimisation du codage des données répétitives.

1.8 Rate-distortion Optimization Cost

Le schéma de RD-Cost est illustré dans la figure 1.24. Au début le bloc de prédiction est soustrait de bloc courant original CU_ORG afin d'avoir le bloc résiduel de bloc courant CU_RES. Le CU_RES va subir par la suite une transformation (T), et une quantification(Q). Après la quantification, on utilise CABAC pour coder le CU_RES après la quantification et calculer le nombre de bits R [4].

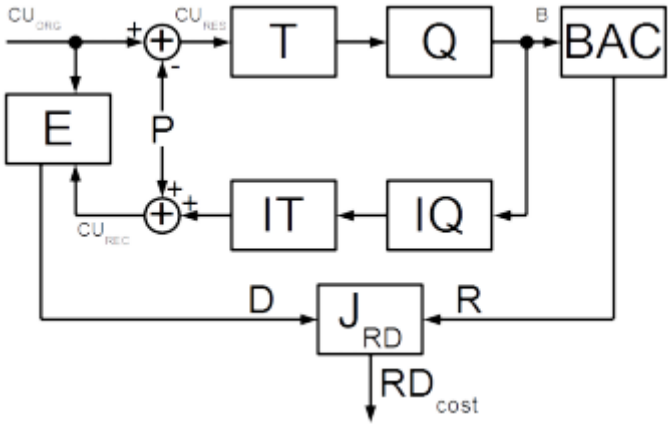


FIG. 1.24- Rate-distortion optimization (RDO) blocks estimation.

CU_RES quantifié va subir le traitement inverse, c'est-à-dire, une quantification inverse (QI), puis une transformation inverse (IT), afin de trouver le bloc reconstruit CU_REC. on calcule la distorsion D par une soustraction entre le CU_ORG et le CU_REC. Finalement on applique la formule (1.10) [4] pour calculer le RDO-Cost. RDO-Cost permet la sélection du mode le plus optimal pour la taille, c'est celui qui minimise le RDO-Cost.

$$J_{RD} = D + \lambda R, \ \lambda = 0.85 * 2^{(QP-12)/3}. \quad (1.10)$$

Chapitre 2: High Efficiency Video Coding (HEVC)

Chapitre 2: High Efficiency Video Coding (HEVC)

HEVC est la nouvelle norme de compression vidéo, elle est désigné d'être le successeur de H.264/AVC, le format de codage vidéo H.265 a été lancé dans des produits dès 2013. En Janvier 2013, de nombreux représentants des télécoms, de l'informatique, de l'industrie, de l'électronique et du monde de la télévision ont approuvé la publication d'un brouillon pour la norme High Efficiency Video Coding (HEVC). C'est le format de codage vidéo H.265 qui doit prendre la suite de la norme actuelle H.264/AVC. Si H.264 est utilisé pour l'encodage et le décodage vidéo HD et Full HD, H.265 a été conçu pour accompagner le contenu 4K (4 096 x 2 160 pixels), d'améliorer les performances du H.264 et de doubler le niveau de compression sans pour autant compromettre la qualité de l'image. On dit que le H.265 peut faire de 35 à 67% de mieux que le H.264 en compression [5].

H.265 est mis au point dans le cadre d'une collaboration entre le groupe Moving Picture Experts Group (**MPEG**) et International Télécommunication Union (**ITU-T**). La norme HEVC a été finalisée au début de l'année 2013 et a adapté un domaine de la moitié. Pour les services de télévision, cela devrait prendre plus de temps.

La norme HEVC a adopté la même architecture basic de codage vidéo de la précédente norme H.264/AVC, cette architecture est basée sur :

- ✓ Block hybrid coding scheme: Advanced intra and inter coding modes.
- ✓ Motion compensated prediction.
- ✓ Transform coding with high efficiency entropy coding: Context Adaptive Binary arithmetic Coding (**CABAC**).
- ✓ Flexible quad-tree partitioning structure with a large block size 64 x 64.
- ✓ New partitions: Coding units, Prediction units and Transform units.

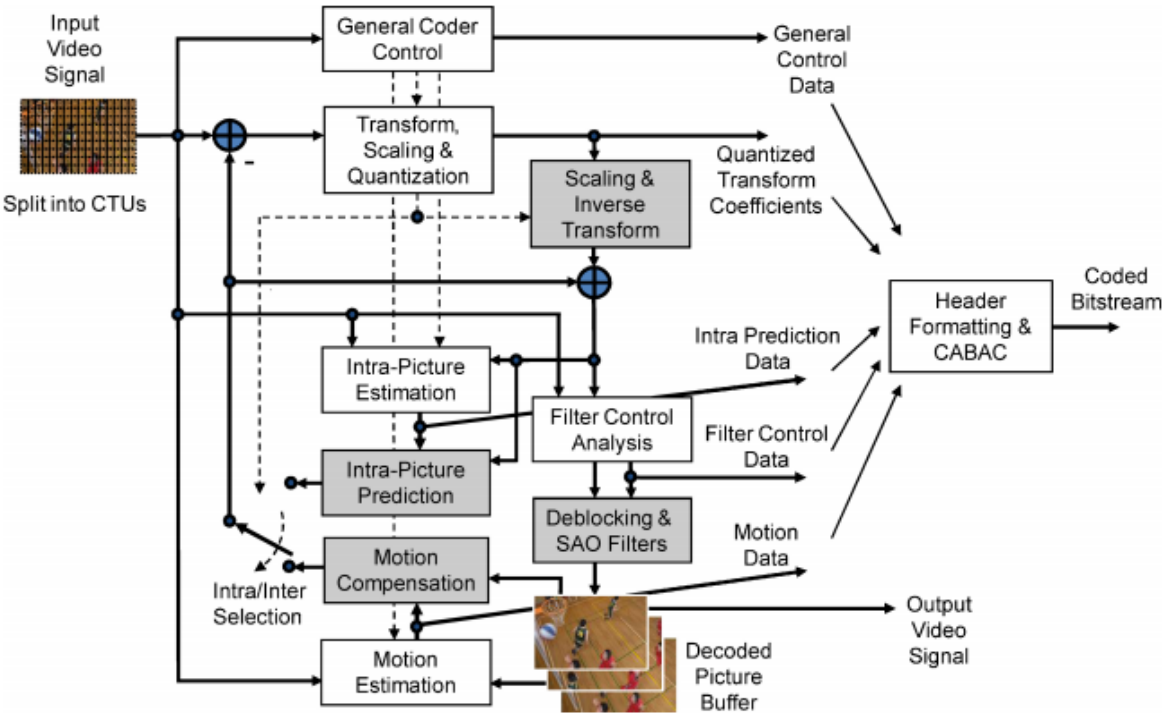


FIG. 2.1-Model HEVC d'encodeur vidéo.

2.1 Echantillonnage et la représentation de l'image

Pour la représentation de couleur de signal vidéo, HEVC utilise l'espace couleur YCrCb avec le format 4 :2 :0. Cette séparation de représentation couleur en trois composantes sont appelés Y, Cr et Cb. La composante Y est nommée Luma, les deux autres composantes Cr, Cb sont appelés Chroma [6].

Le système visuel humain est très sensible au Luma que chroma, donc le format 4 :2 :0 est utilisé, où chaque composante de chroma est sous-échantillonnée par un facteur de deux horizontalement et verticalement. Chaque pixel de chaque composante est représenté par 8 ou 10-bit pour la précision, et 8-bit est le cas le plus typique.

2.2 La structure de codage

Les méthodes de partitionnement des images d'une séquence vidéo sont présentées, et la terminologie pour le traitement des données spécifiques d'une image est introduite.

Pour le codage, une image est devisée en arbre de blocs de codages (CTB : Coding Tree Block) en forme des carrées. Un ensemble consécutif d'arbre de blocs est regroupé en tranches. Les CTBs ont une taille configurable pour chaque séquence vidéo codée, et remplace le « MacroBlock », qui a été utilisé dans la précédente norme H.264 [5] [6].

Un CTB est la racine d'un arbre de blocs de codages, qui sont à leur tour partitionnés pour la prédiction, et pour la transformation afin de coder les blocs de résiduels.

2.2.1 Blocs et Unités

L'entité de base de codage est l'arbre de blocs de codages et son correspondant arbre d'unité de codage. Les CTU (Coding Tree Unit) contient les CTBs des composantes d'espace couleurs YUV (figure 2.2). Un CTB est la racine d'un arbre quaternaire (Quadtree en anglais) partitionné en blocs de codages (CB : Coding BLock). Un bloc de codage est devisé en une ou plusieurs blocs de prédictions (PB : Prediction Block), qui constitue la racine d'un arbre de partitionnement quaternaire de blocs de transformation (TB : Transform Block). En conséquence, une unité de codage (CU : Coding Unit), contient l'unité de prédiction (PU : Prediction Unit), et l'ensemble des unités de transformations (TU : Transform Unit). Alors un PU contient les informations communes pour toutes les composantes d'espace couleur, un TU contient la structure de syntaxe de codage résiduel séparé pour chaque composante d'espace couleur. L'emplacement et la taille de CBs, PBs, et TB de la composante Luma sont appliqués au correspondant unité de codage (CU), unité de prédiction (PU), et unité de transformation (TU). En conséquence, l'emplacement et la taille pour les blocs de chrominances sont dérivés de son correspond bloc en Luma [5] [7].

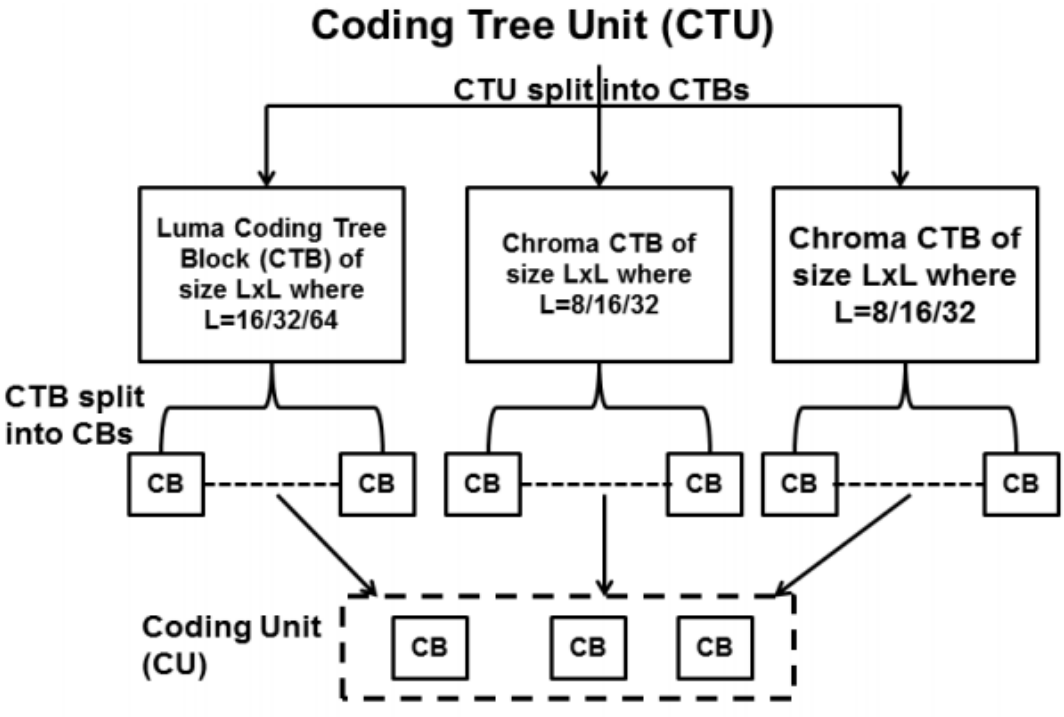


FIG. 2.2-la structure de subdivision des CTU en CB.

2.2.2 Coding Tree Block & Coding Bloc

L'image est partitionnée en arbre blocs de codages, en forme de carré de taille 2^n x 2^n, ou n ∈ {4, 5, 6}, la plus petite taille d'un CTB est de 16x16 qui correspond à la taille de MacroBloc tel qu'elle est utilisés dans la précédente norme H.264. En modifiant la taille de CTB jusqu'à 64 x 64, la structure de codage peut être adaptée à l'image de grande résolution (1920 x 1088).

Un CTB est la racine d'une structure d'arbre quaternaire de blocs de codages, d'une taille carrée. Le bloc de codage de son tour est la racine de la structure de bloc de prédiction et l'arbre quaternaire résiduel. Un CTB peut être représenté par un seul CB, comme il peut être devisé en quatre partitions de demi-taille verticalement et horizontalement. Chacun de ces partitions peut être lui-même un CB, ou peut-être encore une fois partitionné, et ainsi de suite. En HEVC, un CB peut avoir une taille minimale de 8 x 8. En résultant, l'arbre de partitionnement quaternaire est numérisé à l'aide d'un Z-scan afin de former la séquence de codage pour un CB. La figure 2.3 est un exemple d'un CTB partitionné en CB, et montre aussi à Z-scan ordre [5] [7].

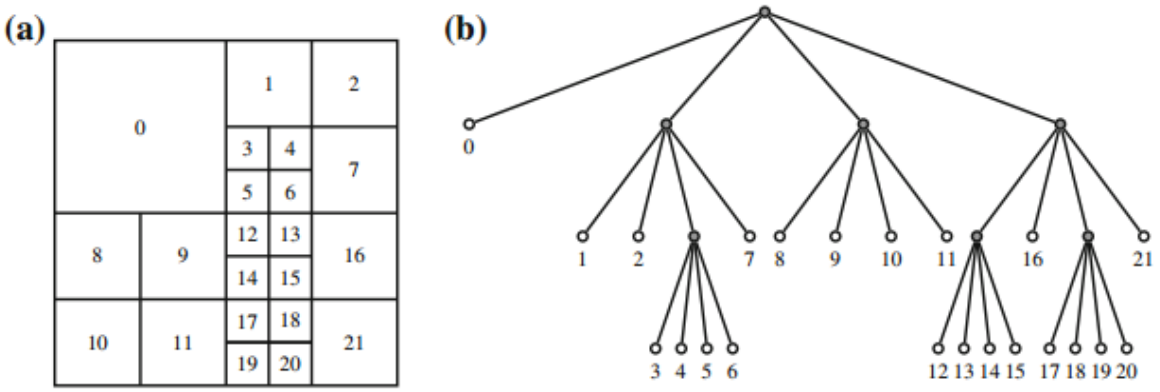


FIG. 2.3-Un CTB partitionné en CB. **a** Partitionnement spatial. **b** l'Arbre quaternaire correspondant.

2.2.3 Bloc de Prédiction

Pour la compensation du mouvement inter prédiction et intra prédiction, un CB est partitionné en bloc de prédiction (PB). Pour intra prédiction, la taille PB est en général identique à celle de CB, sauf la taille minimale d'un CB qui est de 8 x 8 et celle d'un PB est 4 x 4. Pour Inter prédiction d'un CB, huit différentes partions pour un PB sont valables. Le partitionnement d'un PB pour intra et inter sont représentés dans la figure 2.4 [5] [7].

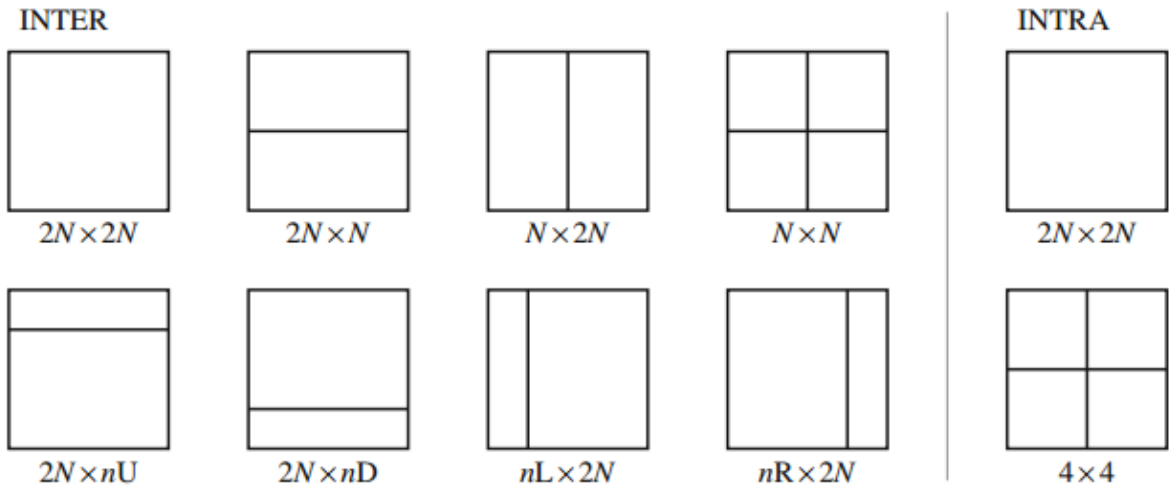


FIG. 2.4-Partitionnement un CB de taille 2N x 2N en PB, avec n=N/2. Pour les partitionnements asymétriques de CB, les indices U, D, L, et R désignes respectivement Up, Down, Left, et Right. Pour le codage intra, seul les CBs de taille 8 x 8 sont autorisés d'être partitionnés en quatre blocs de taille 4 x 4.

2.2.4 Transform Tree and Transform Block

Un CB est partitionné en PB, chaque PB est une racine d'un autre arbre quaternaire appelé arbre de transformation (Transform tree). Les feuilles de cet arbre sont des TB (Transform Block). La taille de TB définit la taille de l'objet de la transformation résiduelle. La taille minimale d'un TB est de 4 x 4, la taille maximale est de 64 x 64. Un TB de taille 64 x 64 subit implicitement une subdivision en quatre TB de taille 32 x 32 comme une taille maximale valable pour un TB. Un exemple de TB partitionnement est représenté dans la figure 2.5 [5] [7] [8].

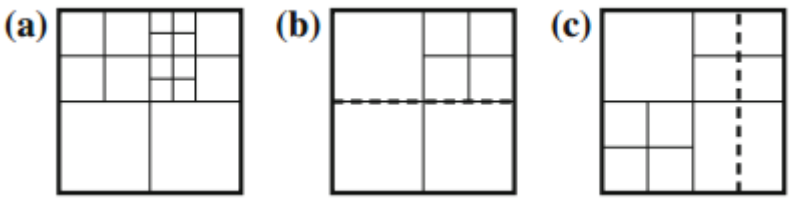


FIG. 2.5-Exemple d'une CB partitionné en PB et TB.

2.3 Intra Prédiction

La décision pour appliquer intra ou inter prédiction est indiquée par un flag dans chaque CU. Pour une image de type I, seulement le mode intra sera appliqué, par contre dans une image de type P ou B chaque CU peut être codé soit en intra ou bien en inter.

Les méthodes d'intra prédiction peuvent être classifiées en deux catégories, la première catégorie est la méthode de prédiction angulaire, la deuxième est la prédiction PLANAR et la prédiction DC. Le nombre total de la prédiction intra supporté par HEVC est de 35 modes, comme il est listé dans la table 2.1 [10], et représenté dans la figure 2.6 [5].

Numéro de mode d'Intra prédiction	**Le nom associé**
0	INTRA_PLANAR
1	INTRA_DC
2 . . . 34	INTRA_ANGULAR[i], i=2 . . . 34

TAB. 2.1-La relation entre le numéro de mode intra prédiction et son nom associé.

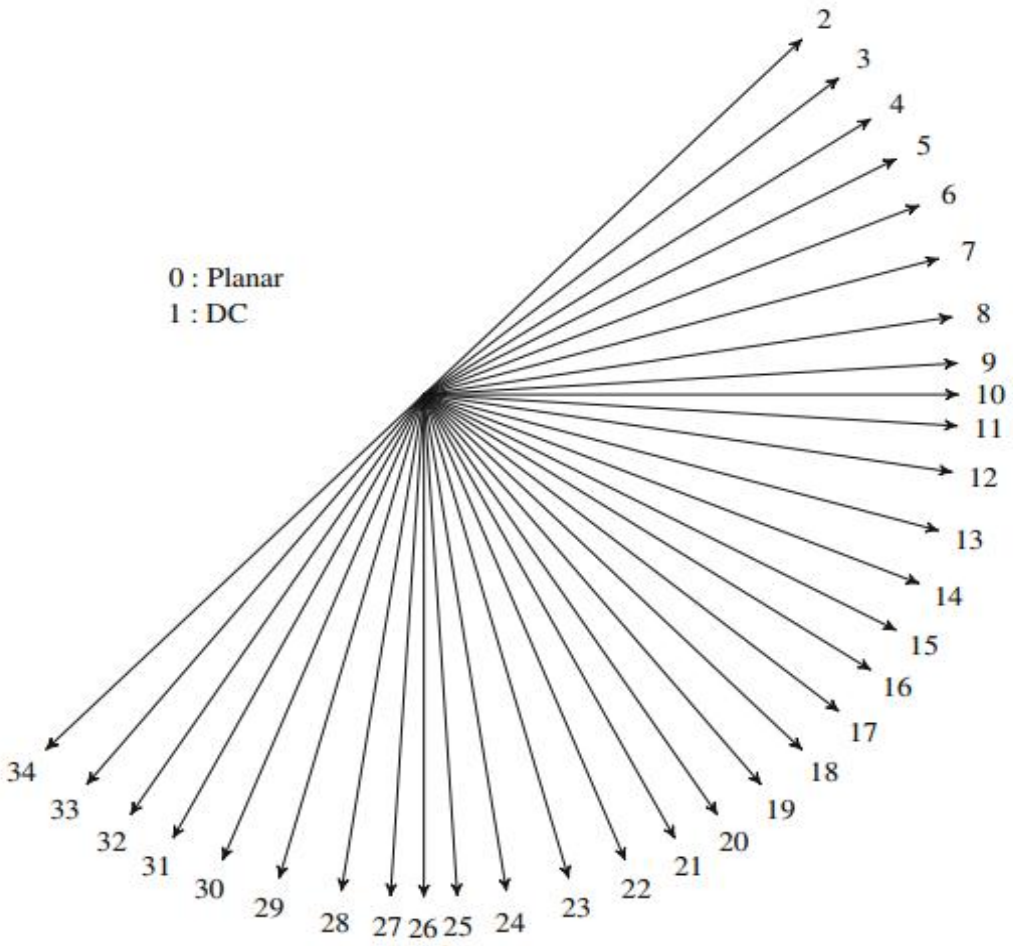


FIG. 2.6-Mode Inra prédiction, et la prédiction angulaire.

Toutes les modes intra prédiction en HEVC utilisent des échantillons de blocs de références reconstruites, comme il est illustré dans la figure 2.7 [10].

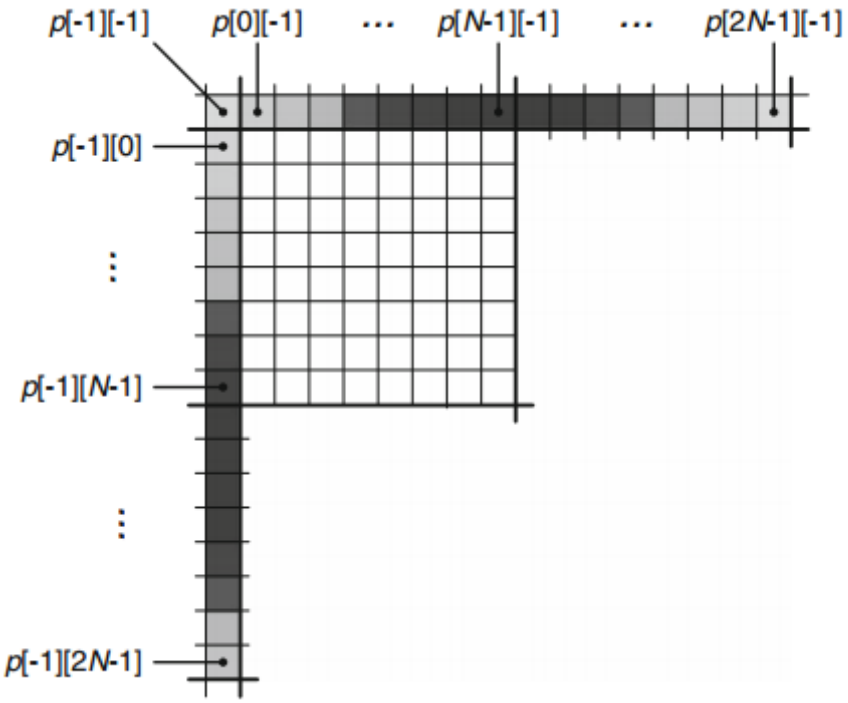


FIG. 2.7-Exemple des échantillons de références utilisés pour les modes de prédictions.

2.3.1 Intra Prédiction Angulaire

La prédiction angulaire en HEVC est conçue pour modéliser efficacement les différentes structures directionnelles. L'ensemble de directions des prédictions disponibles a été sélectionné pour fournir un bon compromis entre la complexité et l'efficacité de codage. L'échantillon procédé pour la prédiction est conçu pour avoir de faible exigence en matière de calcul et d'être cohérent entre les différentes tailles de bloc et les directions de prédictions. Cela a été trouvé particulièrement important que le nombre et la taille des blocs et les directions de prédictions prises en charge par le codage intra en HEVC, qui dépassent de loin ceux de la précédente norme H.264. En HEVC il y a quatre blocs d'intra prédiction d'une taille allant de 4 x 4 à 32 x 32, dont chacun prend en charge 33 directions distinctes de prédictions. Dans la figure 2.8, un exemple de la prédiction suivant 33 directions [5] [7] [8].

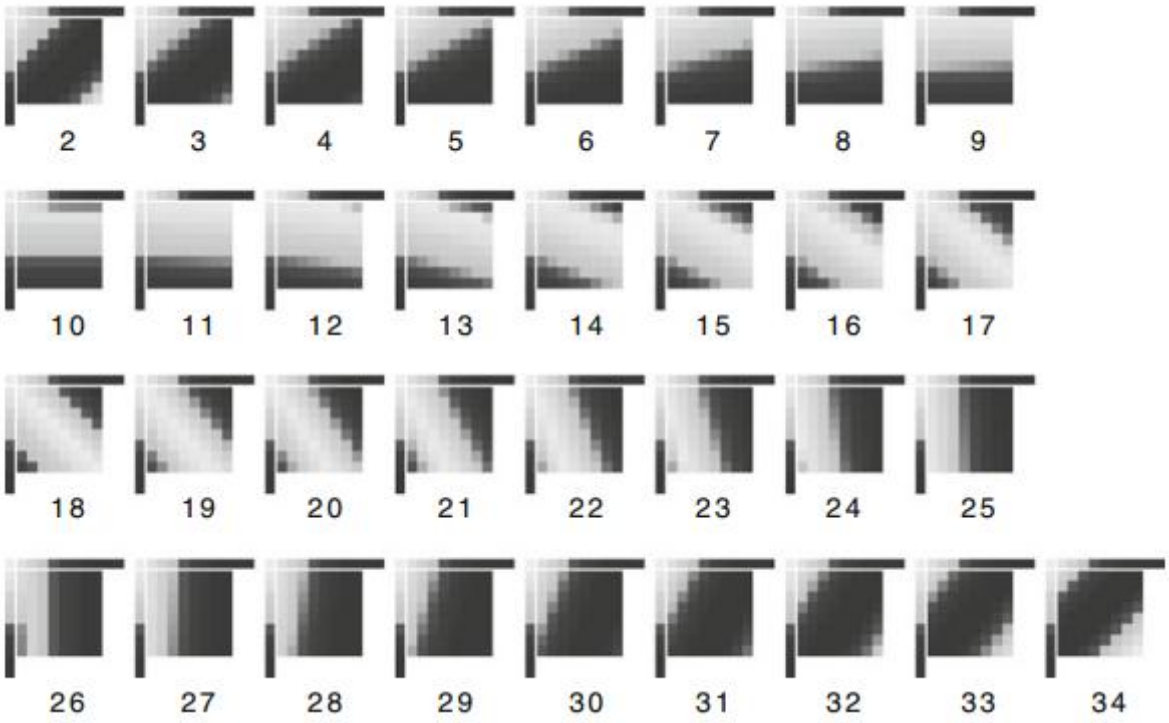


FIG. 2.8-Un exemple de prédiction pour un bloc de taille 8 x 8 suivant 33 directions.

2.3.2 Intra Prédiction Planar

Dans le mode de prédiction planar, les structures de gradients dans un bloc peuvent être estimées. La prédiction pour un bloc est générée par une moyenne pondérée de quatre des échantillons de références en fonction de l'emplacement de l'échantillon comme il est montré dans la figure 2.9 [8]. Les valeurs de prédictions sont dérivées par la formule 2.1 [8].

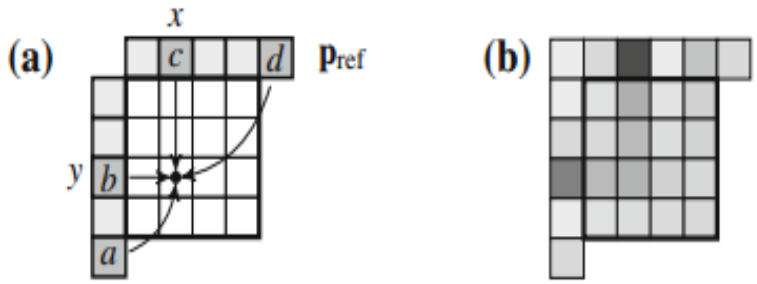


FIG. 2.9-Planar Intra prédiction.

$$\mathbf{B}_c(x,y) = \frac{y+1}{2N_c} p_{ref}(-1, N_c) + \frac{N_c - 1 - x}{2N_c} p_{ref}(-1, y) + \frac{N_c - 1 - y}{2N_c} p_{ref}(x, -1) + \frac{x+1}{2N_c} p_{ref}(N_c, -1). \quad (2.1)$$

2.3.3 DC Intra Prédiction

Pour prédiction DC, la moyenne des valeurs des échantillons adjacents horizontalement et verticalement, est calculée par la formule suivante 2.2 [8].

$$v_{DC} = \frac{1}{2N_c}\left(\sum_{x=0}^{N_c-1} p_{ref}(x,-1) + \sum_{y=0}^{N_c-1} p_{ref}(-1,y)\right). \tag{2.2}$$

Les échantillons dans le bloc de prédiction, sont affectés à la moyenne DC. Pour luma, les échantillons (0, 0), (X, 0), et (0, Y), à savoir les échantillons à gauche et en haut qui limitent le bloc, sont traités séparément. A ces endroits, les valeurs de prédictions est une moyenne de pondération de VDC et les valeurs de références voisines.

$$\mathbf{B}_c(0,0) = \frac{1}{4}\left(p_{ref}(-1,0) + p_{ref}(0,-1) + 2v_{DC}\right), \tag{2.3}$$

$$\mathbf{B}_c(x,0) = \frac{1}{4}\left(p_{ref}(x,-1) + 3v_{DC}\right), \tag{2.4}$$

$$\mathbf{B}_c(0,y) = \frac{1}{4}\left(p_{ref}(-1,y) + 3v_{DC}\right). \tag{2.5}$$

2.4 Inter Prédiction

L'inter prédiction en HEVC peut être considéré comme une amélioration constante, et une généralisation de toutes les parties communes entre les précédentes normes de codages vidéo, par exemple H.264/AVC. La prédiction de vecteur de mouvement a été renforcée avec Advanced Motion Vector Prediction (AMVP), basée sur la compétition de vecteurs de mouvements. Une nouvelle technique d'inter prédiction est nommée Mode Merge. Cette technique de fusion permet de simplifier de façon significative la signalisation des données de mouvements par bloc, en déduisant tous les données de mouvements à partir de blocs déjà codés.

2.4.1 Advanced Motion Vector Prediction

Comme dans la précédente norme de codage de vidéo, les vecteurs de mouvements en HEVC sont codés en termes de composante horizontale (X), et verticale (Y), comme une différence que l'on appelle Motion Vector Prediction (MVP). Le calcul de la différence de deux composantes de vecteurs de mouvement, est précisé par les deux équations suivantes :

$$\mathrm{MVD}_x = \Delta x - \mathrm{MVP}_x \tag{2.6}$$

$$\mathrm{MVD}_y = \Delta y - \mathrm{MVP}_y \tag{2.7}$$

Le vecteur de mouvement du bloc courant est généralement corrélé avec les vecteurs de mouvements des blocs voisins dans l'image courante, ou dans l'image précédemment codée, car les blocs voisins sont susceptibles de correspondre au même objet mobile avec un mouvement semblable, et les objets ne sont pas susceptibles de changer brusquement de mouvement. Par conséquent, le fait d'utiliser les vecteurs de mouvements des blocs voisins en tant que prédicteurs, réduit la taille de la différence de vecteurs signalés. Les MVPs sont généralement dérivés de vecteurs de mouvement déjà décodés à partir des blocs voisins spatiaux, ou à partir de blocs temporellement voisins dans l'image de Co-localisées.

En HEVC, l'approche implicitement dérivée de MPV a été remplacée par une technique connue sous le nom de la compétition de vecteurs de mouvement, qui signale explicitement qu'un MVP sélectionné à partir d'une liste de MVPs, qui sera utilisée comme vecteur de mouvement.

Le concept final de la liste des candidats en AMVP, contient les MVPs suivants :

- ✓ Deux candidats spatiaux seront dérivés de cinq blocs voisins spatialement.
- ✓ Un seul candidat temporel sera dérivé à partir de deux blocs temporellement voisins.
- ✓ Un vecteur de mouvement nul lorsque les MVPs spatiaux, temporels, ou les deux ne sont pas valables.

2.4.1.1 les candidats spatiaux

Deux candidats spatiaux A et B sont dérivés à partir de cinq spatiaux blocs voisins, comme il est illustré dans la figure 2.9(**b**). Pour le candidat A, les données de mouvements de deux blocs A0 e A1 aux cornes bac à gauche, sont vérifiées si l'un des blocs candidats contient un indice de référence qui est égal à l'indice de référence de bloc courant. Le premier vecteur de mouvement trouvé sera considéré comme candidat A. lorsque tous les indices de références de A0 et A1 pointent à une image de référence différente de celle de l'indice de bloc courant, le vecteur de mouvement associé ne peut être utilisé.

Pour le candidat B, les candidats B0, B1, et B2 sont vérifiés séquentiellement de la même façon que les candidats A0 et A1 ont été vérifiés [7].

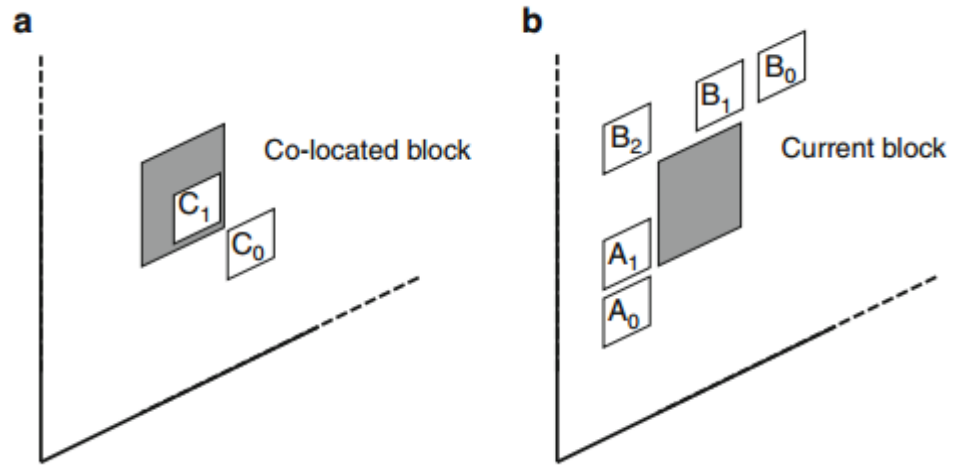


FIG. 2.9-Motion Vector Prediction & Merge Mode candidats. (**a**) Temporel. (**b**) Spatial.

2.4.1.2 les candidats temporels

L'image Co-localisée, est une image de référence qui est déjà décodée, il est possible d'envisager également des données de mouvements à partir du bloc de la même position, à partir du bloc à droite du bloc Co-localisé, ou du bloc ci-dessous. En HEVC le bloc vers la droite en bas et au centre du bloc actuel, est été déterminé à être le plus approprié pour fournir un bon vecteur de mouvement temporel (TMVP : Temporal Motion Vector Prediction). Ces candidats sont illustrés dans la figure 2.9(**a**), où C0 représente le voisin en bas à droite, et C1 représente le bloc central. Là encore, les données de mouvements de C0 sont considérées d'abord comme premier candidat, si elles ne sont pas valables, les données de mouvements de C1 seront utilisées pour dériver TMVP [7].

2.4.2 ModeMerge

Il y a une seule différence principale entre AMVP et Merge Mode. La liste des AMVP ne contient que des vecteurs de mouvements pour une liste de référence, alors que le Merge Mode, contient toutes les données de mouvements, y compris si l'information d'image comprend une ou deux listes de références utilisées, ainsi que l'indice de référence est un vecteur de mouvement pour chaque liste. Cela réduit considérablement la signalisation des données de mouvements. Dans l'ensemble, la liste des candidats de Merge Mode est la suivante:

- ✓ Quatre candidats sont dérivés à partir de cinq candidats spatiaux.
- ✓ Un seul candidat temporel est dérivé à partir de deux candidats temporels.
- ✓ Fusion des candidats supplémentaires, y compris les candidats bi-prédiction combinés, et vecteur de mouvement nul.

2.4.2.1 les candidats spatiaux

Les premiers candidats de la liste Mode Merge, sont les candidats spatiaux. Ici les même blocs utilisés dans AMVP, seront utilisés en Mode Merge, voir la figure 2.9(**b**). Afin de tirer une liste de prédictions de vecteurs mouvements, un MVP est dérivé de A0 et A1, et un autre de B0, B1, et B2 respectivement dans cet ordre. Pour Mode Merge, on peut insérer jusqu'à quatre candidats dans la liste de Mode Merge après une vérification séquentielle de A1, B1, B0, A0, et B2. Au lieu de vérifier si un bloc voisin est valable et contient l'information de mouvement, certains contrôles de redondances supplémentaire sont effectués avant des prendre tous les données de mouvements du bloc voisins en tant que Mode Merge candidats [7]. Ces contrôles de redondance peuvent être divisés en deux catégories :

- ✓ Evité d'avoir des candidats avec des données de mouvements redondantes dans la liste.
- ✓ Empêcher la fusion de deux partitions qui pouvaient être exprimées autrement.

Dans la conception finale, deux comparaison sont effectuées par candidat, résultant en cinq comparaisons globale, considérant l'ordre des candidats suivant {A1, B1, B0, A0, B2}, B0 vérifie uniquement B1, A0 que A1, B2 que B1 et A1.

2.4.2.2 Les candidats temporels

La dérivation de vecteur de mouvement pour Mode Merge temporel candidat, est la même pour TMVP décrit dans la section 2.4.1.2. Mode Merge candidat comprend tous les données de mouvements, et la TMVP n'a qu'un seul vecteur de mouvement, la dérivation de la totalité des données de mouvements dépend uniquement du type de slice. Pour les slices de type B, un TMVP est dérivé pour chaque liste d'images de références. En fonction de la validité de TMVP pour chacune des listes de références, le type de prédiction est défini comme étant une bi-prédiction, ou définie sur la liste où le TMVP est valable. Tous les indices d'images de références associés sont mis à zéro. En conséquence, pour des slices de type P, un seul TMVP est dérivé de la liste 0 avec un indice d'image de référence égal à zéro.

Lorsque, au moins un TMVP est valable, et il est ajouté à la liste de Mode Merge candidats, aucun contrôle de redondance n'est effectué. Cela rend la construction de la liste de Mode Merge indépendante de l'image Co-localisé. Ce qui améliore la résistance aux erreurs.

2.5 Transformation, Quantification, Codage

HEVC utilise une transformation de codage de l'erreur de prédiction, de la même manière que la précédente norme. Le bloc résiduel est partitionné en plusieurs carrées TBs, comme il est décrit dans la section 2.2.4. Les tailles de blocs supportées par la transformation sont, des blocs de tailles 4 x 4, 8 x 8, 16 x 16, 32 x 32 [5][7][9].

2.5.1 La transformation

Deux dimensions de transformation sont calculées en appliquant 1-D transformation dans les directions horizontal et vertical. Les éléments de noyau sont des matrices de transformations dérivées en rapprochant des fonctions DCT des bases, en prenant en considération la limitation de la gamme dynamique nécessaire pour le calcul de la transformation, et la maximisation de la précision. Pour la simplicité, une seule matrice des nombres entiers, et de taille 32 x 32 est utilisée, et des versions sous-échantillonnées sont utilisées pour les autres tailles, à titre exemple, la matrice de transformation de taille 16 x 16 représentée ci-dessous. Les matrices de tailles 8 x 8, et 4 x 4, peuvent être dérivées par l'utilisation des premières huit lignes 0, 2, 4, …, et les premières quatre lignes 0, 4, 8, …., respectivement.

Bien que la norme HEVC spécifie la transformation simplement en termes de valeurs d'une matrice, les valeurs entières de la matrice ont été sélectionnées pour avoir des propriétés de la symétrie, ce qui permet une implémentation partielle est rapide des facteurs avec moins d'opérations mathématiques, et les grandes transformations peuvent être construites en utilisant la plus petite transformation, comme bloc de construction.

$$H = \begin{bmatrix}
64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\
90 & 87 & 80 & 70 & 57 & 43 & 25 & 9 & -9 & -25 & -43 & -57 & -70 & -80 & -87 & 90 \\
89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 & -89 & -75 & -50 & -18 & 18 & 50 & 75 & 89 \\
87 & 57 & 9 & -43 & -80 & -90 & -70 & -25 & 25 & 70 & 90 & 80 & 43 & -9 & -57 & -87 \\
83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 & 83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 \\
80 & 9 & -70 & -87 & -25 & 57 & 90 & 43 & -43 & -90 & -57 & 25 & 87 & 70 & -9 & -80 \\
75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 & -75 & 18 & 89 & 50 & -50 & -89 & -18 & 75 \\
70 & -43 & -87 & 9 & 90 & 25 & -80 & -57 & 57 & 80 & -25 & -90 & -9 & 87 & 43 & -70 \\
64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\
57 & -80 & -25 & 90 & -9 & -87 & 43 & 70 & -70 & -43 & 87 & 9 & -90 & 25 & 80 & -57 \\
50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 & -50 & 89 & -18 & -75 & 75 & 18 & -89 & 50 \\
43 & -90 & 57 & 25 & -87 & 70 & 9 & -80 & 80 & -9 & -70 & 87 & -25 & -57 & 90 & -43 \\
36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 & 36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 \\
25 & -70 & 90 & -80 & 43 & 9 & -57 & 87 & -87 & 57 & -9 & -43 & 80 & -90 & 70 & -25 \\
18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 & -18 & 50 & -75 & 89 & -89 & 75 & -50 & 18 \\
9 & -25 & 43 & -57 & 70 & -80 & 87 & -90 & 90 & -87 & 80 & -70 & 57 & -43 & 25 & -9
\end{bmatrix}.$$

Pour le bloc de transformation de taille 4 x 4, un nombre des entiers de transformation est dérivée de la fonction DST (*discrete sine transform),* ce dernier est appliqué sur des blocs résiduels de luma, dans une image intra, en utilisant la matrice ci-dessous. L'utilisation de la DST est limitée seulement au bloc luma de taille 4 x 4.

$$H = \begin{bmatrix}
29 & 55 & 74 & 84 \\
74 & 74 & 0 & -74 \\
84 & -29 & -74 & 55 \\
55 & -84 & 74 & -29
\end{bmatrix}.$$

2.5.2 Quantification

Etant donné que les lignes de la matrice de transformation, sont des approximations de valeurs proches de façon uniforme des fonctions de bases mises à l'échelle de la DCT orthonormé, l'opération de pré-échelonnage, qui est incorporée dans la dé-quantification de H.264/AVC**,** n'est pas nécessaire en HEVC. Cet évitement de fréquence spécifique à la fonction de mises à l'échelle de base, est utile pour réduire la taille de la mémoire intermédiaire, en particulier lorsque l'on considère que la taille de la transformation peut être aussi grande que 32 x 32 [5].

Pour la quantification, HEVC utilise essentiellement la même URQ (Uniform Reconstruction Quantization), contrôlée par un paramètre de quantification (QP) comme dans H.264/AVC. La gamme des valeurs QP est définie entre 0-51, et une augmentation de 6 fois de la taille de pas de quantification, telle que la cartographie des valeurs QP à l'étape d'approximativement logarithmique de la taille. En HEVC les matrices de quantifications sont aussi supportées, HEVC utilise 20 matrices de quantifications qui dépondent de la taille et le type de bloc de transformation (voir la figure 2.10) [8] :

- ✓ Luma : Intra 4 x 4, Inter 4 x 4, Intra 8 x 8, Inter 8 x 8, Intra 16 x 16, Inter 16 x 16, Intra 32 x 32, Inter 32 x 32.
- ✓ Cb : Intra 4 x 4, Inter 4 x 4, Intra 8 x 8, Inter 8 x 8, Intra 16 x 16, Inter 16 x 16.
- ✓ Cr : Intra 4 x 4, Inter 4 x 4, Intra 8 x 8, Inter 8 x 8, Intra 16 x 16, Inter 16 x 16.

Default 4x4 for IntraLuma, IntraCb, IntraCr, InterLuma, InterCb, InterCr

16	16	16	16
16	16	16	16
16	16	16	16
16	16	16	16

(Flat matrix)

Default 8x8 for IntraLuma, IntraCb, IntraCr

16	16	16	16	17	18	21	24
16	16	16	16	17	19	22	25
16	16	17	18	20	22	25	29
16	16	18	21	24	27	31	36
17	17	20	24	30	35	41	47
18	19	22	27	35	44	54	65
21	22	25	31	41	54	70	88
24	25	29	36	47	65	88	115

Default 8x8 for InterLuma, InterCb, InterCr

16	16	16	16	17	18	20	24
16	16	16	17	18	20	24	25
16	16	17	18	20	24	25	28
16	17	18	20	24	25	28	33
17	18	20	24	25	28	33	41
18	20	24	25	28	33	41	54
20	24	25	28	33	41	54	71
24	25	28	33	41	54	71	91

FIG. 2.10-Matrice de quantification de TB de taille 4 x 4 et 8 x 8.

Pour réduire la mémoire nécessaire pour stocker les fréquences spécifiques, seules les matrices de quantification de taille 4 x 4 et 8 x 8 sont utilisées. Pour des TB de taille 16 x 16 et 32 x 32, une matrice de taille 8 x 8 est utilisée pour la construction de matrices de quantification de taille plus grande (voir la figure 2.11).

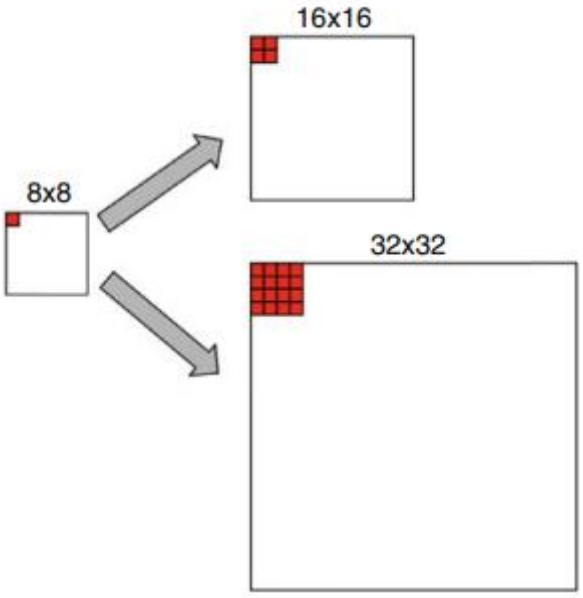


FIG. 2.11-Construction des matrices de quantifications de tailles 16 x 16, 32 x 32, à partir d'une matrice de quantification de taille 8 x 8.

2.5.3 Codage entropique

HEVC utilise une seule méthode de codage entropique CABAC, au lieu de deux (CAVLC, CABAC) dans la norme H.264. L'algorithme de base de CABAC reste inchangé (voir la section 1.7.3.5).

2.6 D'autre outils

En addition de la structure de codage et de ces outils concernant la prédiction, un effort particulier a été mené sur les filtres et le parallélisme du standard.

Concernant les filtres intégrés au codage, un filtre anti-bloc est toujours utilisé et permet de réduire les artéfacts liés au traitement par bloc bien que les tailles plus grandes utilisées dans HEVC réduisent partiellement ce défaut. Deux filtres additionnels sont aussi définis, Sample-Adaptive Offset Filter et Adaptive Loop Filter. Le premier est appliqué après le filtre anti-bloc et consiste à classifier les pixels de la LCU en fonction de leurs intensités puis à appliquer un offset pour chaque catégorie qui est dérivée par rapport à la LCU originale et transmis au décodeur. Le second filtre est quant à lui appliqué avant la copie de l'image dans le buffer et permet d'améliorer la qualité à la fois objective et subjective de l'image. L'image est partitionnée en zones de différents niveaux d'activité auxquelles est appliqué un filtre selon deux formes sélectionnées en fonction de leur efficacité [9].

Par rapport au parallélisme, deux outils sont notamment définis pour permettre un codage efficace pour des applications exploitant le parallélisme : le codage en Wavefront et les Tiles. Le premier outil se base sur une technique bien connue des concepteurs d'encodeurs consistant à traiter les CU en parallèle ligne par ligne avec un décalage du nombre de CU nécessaires pour conserver les dépendances. En particulier, cette méthode permet d'avoir du mouvement. Sur le même principe, l'outil ajoute en plus une gestion astucieuse des probabilités nécessaires au codeur entropique CABAC. L'outil des Tiles voit quant à lui le problème différemment en divisant les images en plusieurs rectangles à encode en parallèle. L'intégration de tels outils dans la norme et donc leur exploitation lors du décodage est un réel avantage pour le développement de futurs encodeurs compétitifs.

Chapitre 3: Extension 3D de High Efficiency Video Coding (3D-HEVC)

Chapitre 3: Extension 3D de High Efficiency Video Coding (3D-HEVC)

La vidéo 3D connait un développement rapide ces dernières années, des sorties de plus en plus fréquentes de films 3D à l'émergence de nouveaux services 3D comme la télévision 3D (3DTV) et la Free Viewpoint TV (FTV). Bien que la vidéo 3D n'a pas encore atteint le succès attendu (du principalement aux inconforts visuels et nécessite de porter des lunettes 3D tout au long de la séance), elle est attendue à conquérir rapidement le marché avec le visionnage auto-stéréoscopique, qui requiert un nombre important de vues à multiplexer simultanément au récepteur [11].

Le format stéréo classique (mettant en jeu deux vues uniquement) étant du coup insuffisant, le format multi-vue (MVV pour Multi-View Video), composé de plusieurs vues représentant la même scène mais légèrement décalées les unes par rapport aux autres, permet de subvenir aux exigences de la 3DTV et la FTV. Or le cout de transmission des différentes vues dans le format MVV peut toutefois rester exorbitant, même si les corrélations inter-vues sont exploitées avec des codeurs tels que Multi-view Vidéo Coding (MVC). Le format Multi-view plus Depth (MVD voir la figure 3.1) est une alternative intéressante à MVV car il introduit des vidéos de profondeur qui sont moins couteuses à coder que les vidéos de texture, et qui peuvent être utilisées pour synthétiser autant des vues nécessaires au récepteur avec la technique de Depth Image Based Rendering (DIBR) voir la figure 3.2 [12].

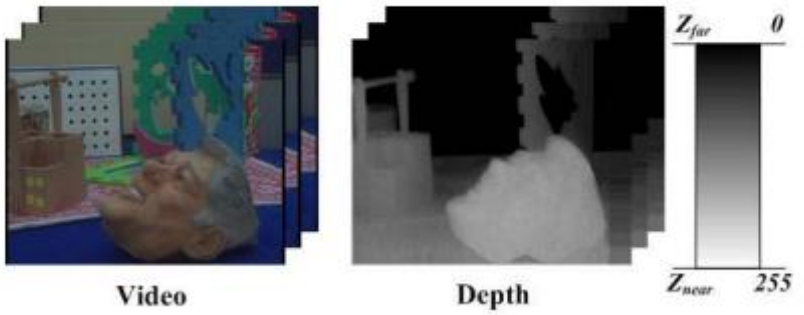


FIG. 3.1-Vidéo plus depth.

Or il n'y a actuellement aucun standard conçu pour coder efficacement des vidéos 3D dans un format MVD. Le groupe MPEG s'est penché sur le sujet en 2011, et l'extension 3D du standard HEVC (3D-HEVC) permettrait un codage efficace des données MVD. C'est dans ce contexte où se situent justement les travaux de ce mémoire. L'objectif est de développer des outils permettant d'augmenter l'efficacité de codage des vues dépendantes, des vidéos de profondeur, et des vidéos synthétisées dans 3D-HEVC. Le fil conducteur des approches proposées est l'amélioration de la prédiction des informations de codage transmises dans un flux binaire 3D-HEVC [13].

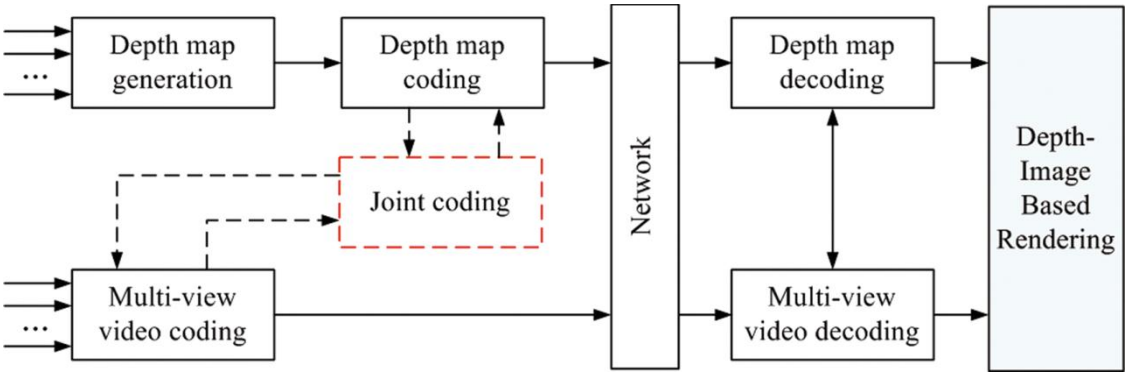


FIG. 3.2-Depth-Image Based rendering.

3.1 3D Video Coding

Certains systèmes d'affichage de 3DTV peuvent utilisés un groupe de séquences vidéo capturées par des caméras multiplexe simultanément de la même scène pour montrer l'effet naturel et vivant. Habituellement, le système choisira 8 ou 16 vues pour représenter la scène 3D, alors que la vidéo stéréo classique adopte seulement deux caméras.

3.1.1 Stereo Video Coding

Conventionnellement, le codage vidéo stéréo est le plus simple, et représente le traditionnel de la vidéo 3D. Correspondant à la distance des yeux humains, deux caméras simultanément pendant la capture de la même scène, pour acquérir la vidéo stéréo à partir de légères différentes point de vue. Les deux images bénéficient simultanément, de la compression en prévoyant une image à partir de l'autre. C'est-à-dire, après avoir encodée une séquence indépendante, le pair apparié peut être prédit en référence à la précédente vidéo en exploitant la corrélation entre les caméras adjacentes [11].

Multi-vue profil (MVP) est spécifiée dans l'UIT-T Rec. H.262/ISO/IEC 12818-2 il y a environ 15 ans. Le schéma de codage de la vidéo MVP à deux couches (couche de base, couche d'amélioration), comme illustré en figure 3.3 [14]. La couche de base de la vidéo est codée en MPEG-2 en flux de bits. Avec le but d'une meilleure efficacité, la couche d'amélioration, est codée en exploitant à la fois la corrélation temporelle et spatiale.

Dans les applications de vidéo stéréoscopique, la couche de base est assignée à la vue de l'œil gauche et la couche d'amélioration à la vue de l'œil droit. Tandis que la couche de base est assignée au profil principal. Le flux de bits peut être également affiché en deux dimensions.

Puisque la méthode garantit la compatibilité ascendante, la vidéo stéréo est plus facile d'être sauvagement déployée en peu de temps. Toutefois, l'amélioration de l'efficacité de compression est relativement limitée, puisque la prédiction temporelle est déjà bien développée, en particulier lors de la capture de plusieurs vues dans des distances proches.

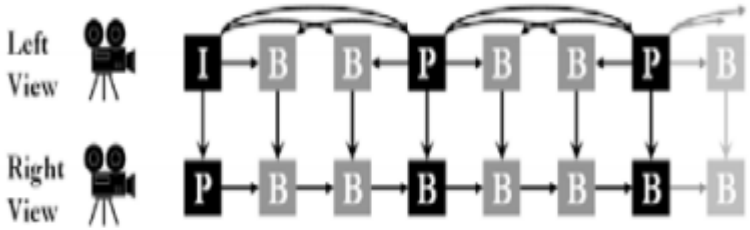


FIG. 3.3-La prédiction pour la conventionnel Video Coding.

3.1.2 Multi-view Video Coding

Multi-vues vidéo est acquise en prenant simultanément la capture de la même scène de différente angles. La première étape de Frée-Viewpoint de vidéo, l'utilisateur peut profiter de la scène d'une manière interactive à partir de plusieurs orientations.

Indépendamment de codage des vidéos séparées, appelés simulcast, la méthode la plus simple en utilisant un codec H.264/AVC. Toutefois interviews peuvent être statiquement supprimés, pour diminuer la quantité de données. Ces redondances peuvent être classées en deux type, inter-vue similitude entre la caméra adjacente, et la similitude de temporelle entre les images successive de chaque vidéo. La technique de compensation de mouvement, qui est bien développées pour une seule vue, peut être utilisée pour la prédiction temporelle. Egalement la technique de compensation des disparités, peut être utilisée pour réduire la redondance inter-vue. Comme il est montré dans la figure 3.4 [14] [15] [16], les algorithmes qui sont basés sur la hiérarchique image B, qui est supporté par la norme H.264/AVC et HEVC, approuvent une meilleur performance, et ont été sélectionnés par JVT comme multi-vue modèle commun.

MVC est spécifié par ISO/IEC 14496-10 | ITU-T Rec. H.264 support le codage direct de la synchronisation des informations à partir de plusieurs points de vues, en utilisant un seul flux, et exploite la redondance inter-caméra pour réduire le débit binaire.

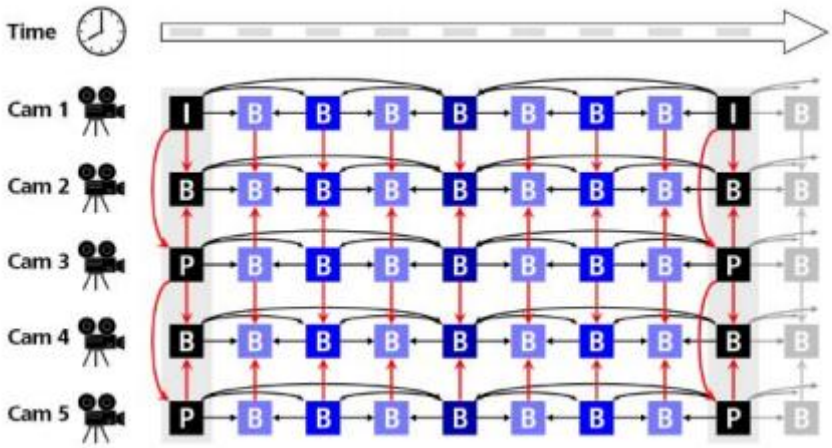


FIG. 3.4-le schéma de la structure Multi-view Video Coding.

En raison de la corrélation temporelle, des expériences menées dans le cadre de la normalisation MPEG, ont montré que MVC dépasse la diffusion simultanée d'une manière significative avec le gain PSNR (Peak Signal-to-Noise Ratio), de sorte que les demandes d'industrie, ISO/MPEG et ITU/VCEG ont décidé d'élaborer une norme MVC spécifique que l'extension de H.264/AVC, et ont terminé la tâche en juillet 2008.

3.1.3 Multi-view plus Depth Video Coding

Multi-view plus depth (MVD) est l'extension des vidéos plus depth, où les spectateurs ne seront plus limités dans la zone de visualisation fixe, et angle de vision. MVD est composé de plusieurs vidéos en couleur avec plusieurs associés des données de profondeurs comme il est montré dans la figure 3.5 [14]. Comparé à MVC, MVD n'a pas besoin de transmettre autant de séquence de même scène, pour des vues intermédiaire, elles peuvent être générer par la technique DIBR du côté du récepteur. A savoir, moins de vue et son information de profondeur associée, sont nécessaires pour être transmises simultanés. Par conséquent, les taux de bits vont être efficacement réduits.

FIG. 3.5-Multi-view plus depth.

Etant donné que les données de profondeurs sont lisses et monochromes, il peut être facilement comprimé lors de l'utilisation de codage vidéo standard, et exploitant la redondance inter-vue. Une efficace solution optimale est proposée par une estimation conjointe, et le codage de la carte de profondeur en utilisant la programmation dynamique le long de l'arbre des coefficients d'ondelette. En outre, coder conjointement les données de profondeurs et multi-vues est une autre option, à l'aide des informations de mouvement de la vidéo de texture correspondante, où la prédiction de la synthèse de profondeur. La performance de compression de la video/depth, peut être améliorée en effectuant l'allocation de débit pour video/depth.

Bien que le multi-view video peut prendre en charge une variété des applications d'affichages 3D, la grande quantité de données et un véritable problème, surtout lorsque le nombre des caméras augmentes.

3.2 La structure de codage Multi-view plus Depth

L'algorithme de codage basé sur le format MVD, dans lequel chaque image vidéo est associée à une carte de profondeur (Depth Map). L'algorithme de codage peut être également utilisé pour un format multi-view sans la carte de profondeur. Les images vidéo, lorsqu'elles sont présentes, les cartes de profondeurs sont codées en unité d'accès par unité d'accès, comme il est illustré dans la figure 3.6. Une unité d'accès comprend toutes les images vidéo et les cartes de profondeurs associées, qui correspondent au même instant dans le temps. Unité VCL, et Non-NAL contenant des paramètres de la caméra, peuvent être en outre associées à une unité d'accès. Il devait être noté que l'ordre de codage des unités d'accès ne doit pas être identique à la capture ou l'affichage. D'une manière générale, les données reconstruisent des unités d'accès déjà codées, et peuvent être utilisées pour un codage efficace de l'unité d'accès actuel. L'accès aléatoire est activé par ce qu'on appelle des unités d'accès aléatoires, dans lequel, les images vidéo et les cartes de profondeurs associées, sont codées sans référence à des unités d'accès précédemment codées. En outre, une unité d'accès ne référencie à aucune unité d'accès qui précède l'unité d'accès aléatoire précédent pour le codage [17].

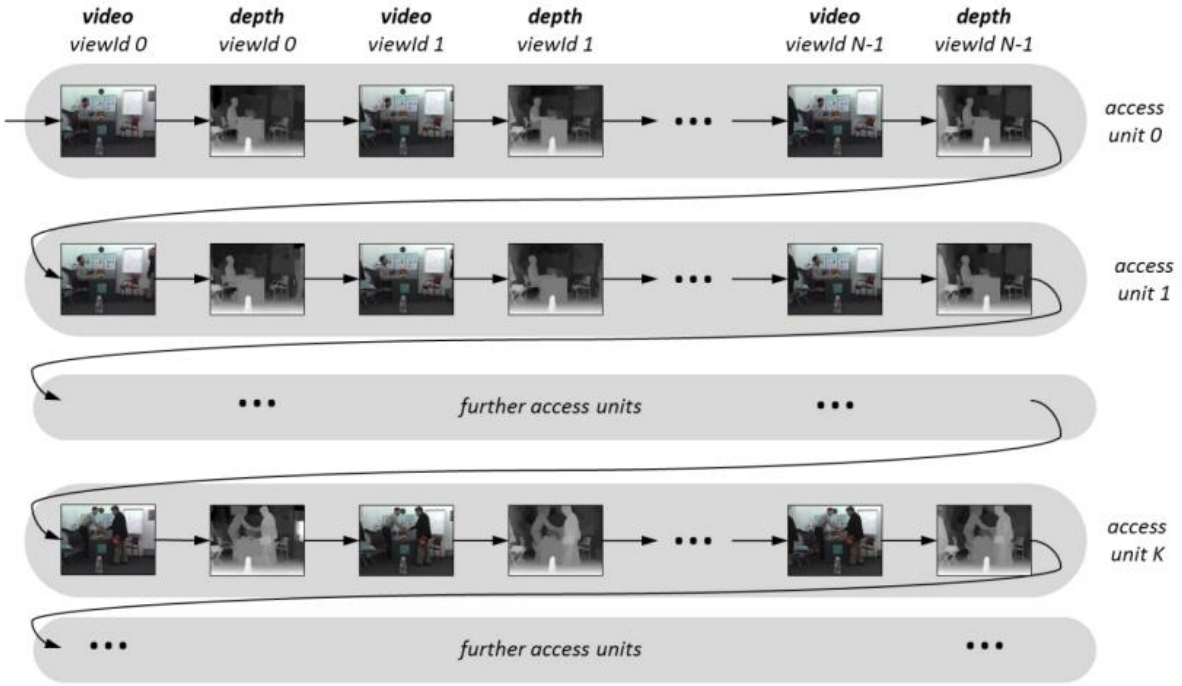


FIG. 3.6-La structure des unités d'accès et l'ordre de codages des vues.

Les images vidéo et les cartes de profondeurs correspondant à une position d'une caméra particulier, sont indiquées par un identifiant de vue (ViewId). Toutes les images vidéo et des cartes de profondeurs qui appartiennent à la même caméra, sont associées à la même valeur de ViewId. Les identificateurs d'affichage sont utilisés pour spécifier l'ordre de codage à l'intérieur d'une unité d'accès, et la détection des vues manquantes dans l'erreur d'environnement.

A l'intérieur d'une unité d'accès, l'image vidéo, lorsqu'elle est présente, sa carte de profondeur associée avec ViewId vaut 0, et elles sont codées en premier, suivies par l'image vidéo et sa carte de profondeur avec ViewId égal à 1, etc. L'image vidéo et sa carte de profondeur avec une valeur particulière de ViewId, sont transmises après tous les images vidéo et ses cartes de profondeurs, avec des valeurs plus petites de ViewId. Pour des vues indépendantes, l'image vidéo est toujours codée avant la carte de profondeur associée. Pour des vues dépendantes, les images vidéo peuvent être codées avant ou après sa carte de profondeur (à savoir, la carte de profondeur avec la même valeur ViewId). Il faut noter que la valeur de ViewId ne représente pas nécessairement la disposition des appareils dans le réseau des caméras. Pour l'ordre de la reconstruction des images vidéo et sa carte de profondeur après le décodage, chaque valeur ViewId est associée à un autre identifiant appelé View ordre index (VOI). Le VOI, est un entier signé, qui spécifie l'ordre des vues codées de gauche à droite. Si une vue A à une valeur de VOI plus petite qu'une vue B, l'appareil photo de la vue 1 se trouve à gauche de la caméra de vue B [17].

La structure de base de codage des vidéos 3D est représentée sur le schéma dans la figure 3.7. Chaque composante de signal est codée en utilisant un codeur basé sur HEVC. Le flux binaires résultant, ou plus précisément, la résultante de Network Abstraction Layer (NAL), les unités sont multiplexés pour former le flux binaire de la vidéo 3D. La vue de base, ou la vue indépendante, est codée en utilisant un codeur HEVC non modifié. Tenu compte de flux binaire de la vidéo 3D, les unités NAL contenant les données de la couche de base, peuvent être identifiées par l'analyse de jeu de paramètre et d'unité NAL-entête des tranches codées de l'unité NAL. Basé sur ces données, le sous flux de bits de la vue de base, peut être extrait et directement codé en utilisant un décodeur classique HEVC.

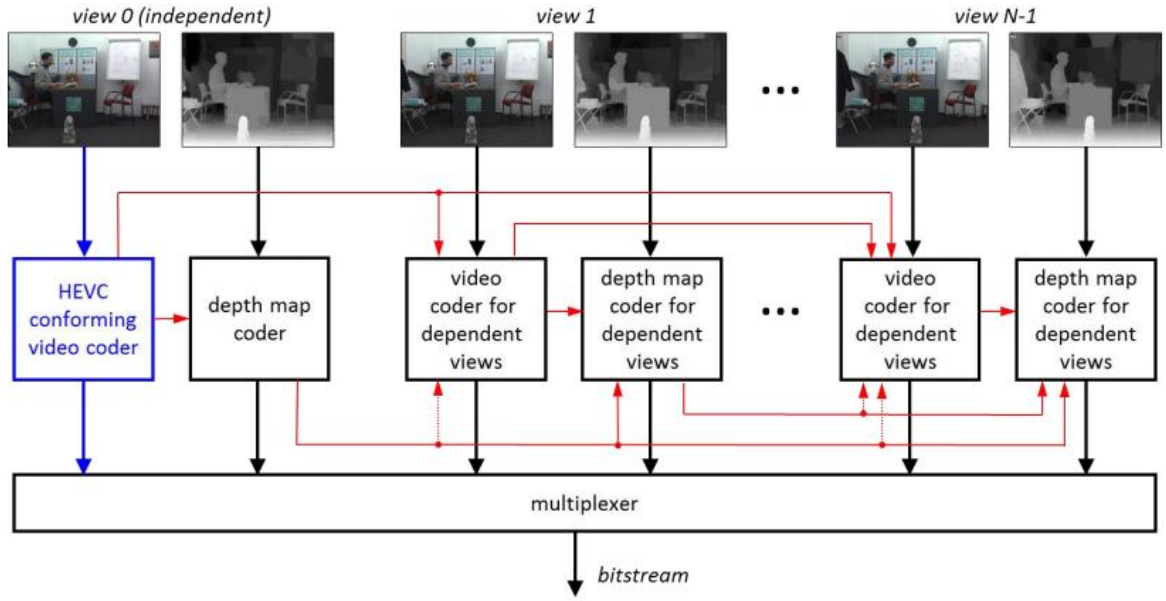


FIG. 3.7-La structure de base de codeur avec la prédiction Inter-Composante (les lignes rouges).

Pour coder des vues dépendantes et les données de profondeurs, les codeurs HEVC modifiés sont utilisés, qui sont étendus en incluant des outils de codages supplémentaires et des techniques de prédictions inter-composantes, qui utilisent des données déjà codées à l'intérieur de la même unité d'accès, comme il est déjà montré dans la figure 3.7 [17]. Pour permettre à un rejet facultatif des données de profondeurs à partir du flux de bits, par exemple, pour supporter le décodage d'une vidéo stéréo adapté aux écrans stéréo classique, la prédiction inter-composante peut être configurée de façon à ce que les images vidéo soient décodées indépendamment des données de profondeurs.

3.3 Technique de Codage 3D-HEVC

L'extension 3D de HEVC de codage vidéo (3D-HEVC), a été développé pour le format vidéo 3D, qui utilise des données de profondeurs, allant de la vidéo stéréo classique à multi-vue vidéo, ainsi que les données de profondeurs (MVD), avec deux ou plusieurs vues et leurs associés : les cartes de profondeurs. Le format évolutif est obtenu par le codage de chaque vue vidéo et son associé carte de profondeurs, en utilisant une structure de codage vidéo 2D qui est basé sur la technologie HEVC. La structure de base 3D-HEVC est déjà montrée dans la figure 3.7.

Afin d'assurer la compatibilité descendante avec la vidéo 2D, une vue de base ou une vue indépendante est codé, en utilisant un codec HEVC entièrement compatible. Cela comprend la prédiction spatiale dans une image de prédiction spatiale, et la prédiction temporelle entre les images à différentes instances de temps, transformation, le codage de résidu de prédiction, et le codage entropique.

Pour le codage des dépendants vues, les données de profondeurs, des codeurs HEVC modifiés, sont utilisés, qui sont tendu en incluant des outils de codage supplémentaires, et des techniques de prédictions inter-composantes, qui utilisent des données de composantes déjà codés aux même instants du temps, comme il est indiqué par les flèches rouges dans la figures 3.7.

3.3.1 Advanced Texture Coding in 3D-HEVC

Pour atteindre une plus grande efficacité de codage, des outils avancés exploitent de la redondance d'inter-vue, le codage a été étendu et évalué.

3.3.1.1 La prédiction de disparité compensée

Comme premier outil de codage des dépendantes vues, le concept de prédiction de disparité compensée (DCP) a été ajoutée comme une alternative à la prédiction de compensation de mouvement (MCP). MCP se réfère à la prédiction inter-image qui utilise des images déjà codées de la même vue en différents temps, tandis que DCP se réfère à la prédiction inter-image qui utilise aussi des images déjà codées de la différentes vues en même temps. DCP est également utilisé dans le prolongement du MVC H.264/AVC, et de même, la syntaxe de bloc d'arbre de codage, et de processus de décodage de HEVC, reste inchangé lors de l'ajout de DCP à notre codeur. Seule la syntaxe de haut niveau a été modifiée de telle sorte que l'image vidéo déjà codée dans la même unité d'accès, peut être insérée dans la liste d'images de références [12] [18].

3.3.1.2 Neighboring Block-Based Disparity Vector Derivation (NBDV)

En raison de la prédiction de compensation de disparité, des vecteurs de mouvements de disparités sont remplis dans un champ de mouvement avec des vecteurs de mouvements normaux. Comme son nom l'indique, le concept de NBDV dérive un vecteur de disparité pour un bloc actuel en utilisant un vecteur de mouvement de disparité disponible à partir des blocs voisins spatialement et temporellement. Les voisins spatiaux sont les même que celle utilisée dans HEVC pour la prédiction de mouvement tandis que les voisins temporelles, comprennent ceux qui suivent le pixel central du bloc Co-localisé dans deux liste de références [12] [17] [18].

Les blocs voisins, sont vérifiés dans un arbre prédéfinie, et une fois le vecteur de mouvement de disparité dans un bloc voisin est identifié à partir du champ de mouvement, le procédé NBDV se termine, et que le vecteur de disparité dérivée est égale à l'identifient de vecteur de mouvement de disparité. Le principal avantage de cette technique, c'est que le vecteur de disparité utilisé pour la prédiction inter-vue, peut être directement dérivée sans bits supplémentaires, et indépendantes de l'image de profondeur associée. Presque tous les outils de codage, peuvent être construits de manière efficace sur la technique NBDV sans accédés aux d'informations de profondeurs.

Divers simplificateurs ont été introduits pour NBDV. Les outils 3D-HEVC sont appliqués du niveau de l'unité de codage (CU), et un seul vecteur de disparité dérivé est utilisé pour chaque unité de prédiction (PU) de la CU. En outre, comme illustré plus en détails dans la figure 3.8, les blocs temporellement voisins (appelé ''Ct '') dans deux images de référence, sont contrôlés par la gauche et au-dessus des blocs voisin de CU courant. Les blocs temporels sont identifiés dans deux images. La première image utilise le vecteur de prédiction du mouvement temporelle en HEVC. La seconde est l'image d'accès aléatoire, au cas d'indisponibilité de l'image avec le niveau temporel plus bas, dans la liste d'image de référence. Il est a noté que la seconde image est choisis de telle sorte, qu'il soit plus probable de trouver des vecteurs de mouvement de disparité dans le bloc voisin dans le temps.

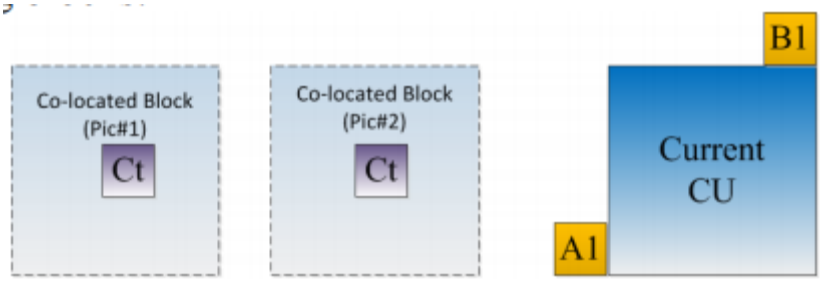


FIG. 3.8-Neighboring blocks in NBDV.

3.3.1.3 Inter-view motion prédiction

Les informations de mouvements entre les vues présentent un haut degré de corrélation, et infère d'une vue à une autre vue, conduit à des gains notables dans l'efficacité du codage, puisque la bonne prédiction réduisent généralement le débit binaire requis pour envoyer ces renseignement. Pour atteindre cela, la disparité, telle que celle obtenu par le procédé NBDV, est utilisé pour établis une correspondance entre les blocs dans chaque vue. Une illustration de prédiction de mouvement entre vue est représentée sur la figure 3.9, où le vecteur de mouvement de vue 1 est déduit de vecteur de mouvement de point de vue 0 de bloc correspondant au 1 en fonction de la disparité entre ces blocs comme dérivé par le procédé NBDV [12] [17] [18].

En 3D-HEVC, la prédiction de mouvement inter-vue est réalisée d'une manière qui est compatible avec le schéma de prédiction de mouvement dans HEVC, et se prolonge bien au-delà. En HEVC, une liste Marge candidats peut être formée, où chaque candidat pourrait contenir à la fois l'indice de référence et vecteur de mouvement de deux directions de prédictions correspondant à la liste d'image de référence, la liste 0 et la liste 1. Au niveau du décodeur, un indice est utilisé pour spécifier le candidat qui est utilisé de telle sorte que les informations de mouvement de PU courant peuvent être dérivées sans autre information transmise. L'idée principale de la prédiction inter-vue de mouvement dans 3D-HEVC est d'obtenir des candidats supplémentaires et les mettre dans la liste des candidats de Mode Merge, ainsi que le Mode Merge existant tel qu'il est résultant en HEVC. Basé sur le vecteur de disparité (DV), les trois candidats suivants peuvent être générés et insérés dans la liste de Merge candidats, qui peut contenir jusqu'à six entrées en 3D-HEVC.

- **Candidat de mouvement inter-vu**: le concept est représenté sur la figure 3.9. étant donnée un vecteur de disparité, un bloc correspondant au bloc PU courant dans une vue de référence de la même unité d'accès est identifié. Si le bloc en inter-vue et son image de référence a une valeur POC (Picture Ordre Count) égale à celle d'une entrée dans la liste de PU courant, son information de mouvement est tirée, pour un candidat de mouvement inter-vue.
- **candidat de mouvement de disparité** : le vecteur de mouvement pour ce candidat est généré par la conversion d'un vecteur disparité d'entrée en un vecteur de mouvement de disparité, et l'indice de référence est fixé à l'indice de référence de l'image de référence inter-vue associée avec le vecteur de disparité.

- **Candidat décalé** : ce candidat est généré avec le même procédé que celui défini ci-dessus avec un vecteur de disparité légèrement décalé, pour compenser dans le cas où le vecteur n'est pas assez précis. Tout d'abord, un candidat supplémentaire inter-vue est généré par un vecteur de disparité d'entrée égale à la DV avec un vecteur de décalage ayant une composante horizontale et une composante verticale égale à la demi largeur et demi hauteur de la PU courant, respectivement. Si le candidat supplémentaire de l'inter-vue est disponible, le candidat décalé est dérivé, sinon, le candidat décalé est fixé à un vecteur de mouvement disparité avec le vecteur de disparité égale à DV avec la composante horizontale décalée de 4.

Il est noté que le premier et le deuxième candidat supplémentaire sont les candidats de mouvement inter-vue, et le mouvement de disparité candidat dérivé avec le vecteur de disparité d'entrée DV.

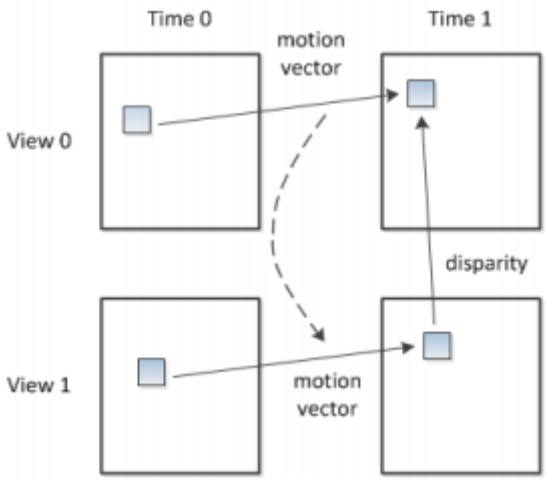


FIG. 3.9-Illustration de la prédiction de mouvement entre vues.

En 3D-HEVC, l'efficacité du codage du premier candidat supplémentaire a été améliorée par la prédiction de mouvement inter-vue sur la base Sub-PU, dans lequel des blocs plus petits (par exemple 8 x 8) de chaque PU peuvent tirer son propre information de mouvement. Chaque Sub-PU a son information de mouvement individuel provenant de la même manière que le candidat de mouvement inter-vue décrit ci-dessus, et la compensation de mouvement peut être effectué séparément pour elle.

3.3.1.4 Prédiction de Résiduel inter-vue

Aussi connu par Advanced Residual Prediction (ARP) est prise en charge uniquement en 3D-HEVC, ce mode de prédiction augmente la précision de la prédiction résiduel en calculant le bloc résiduel à la volée. Dans ARP, le vecteur de mouvement est aligné pour le bloc courant, et le bloc de référence, donc la similitude entre la prédiction résiduelle et le résidu signal le bloc actuel est beaucoup plus élevé et l'énergie restant apes ARP est réduit considérablement [12] [17] [18]. .

Comme illustrer dans la figure 3.10, il existe deux types de ARP design, ARP temporelle et inter-vue ARP. Dans ARP temporel, le facteur de prédiction résiduel est calculé comme une différence entre les deux blocs dans la vue de référence, dans lequel le premier est identifié uniquement par le DV, alors que ce dernier est identifié conjointement par le DV et le vecteur de mouvement en cours. L'information de mouvement utilisée pour obtenir le résidu à la vue de référence est alignée avec celle du bloc actuel.

En inter-vue ARP, un inter-vue résiduel est calculé comme la différence entre deux blocs de temps différents, le bloc de référence et la vue de référence (BaseRef). Dans le cas d'inter-vue ARP, le vecteur de mouvement de disparité (DMV) est signalé et utilisé pour identifier un bloc de référence d'inter-vue dans la vue de référence, dans lequel un vecteur de mouvement (mvLx) peut être dérivé et utilisé pour identifier d'avantage les deux blocs dans l'autre instant de temps (CurrRef et baseRef).

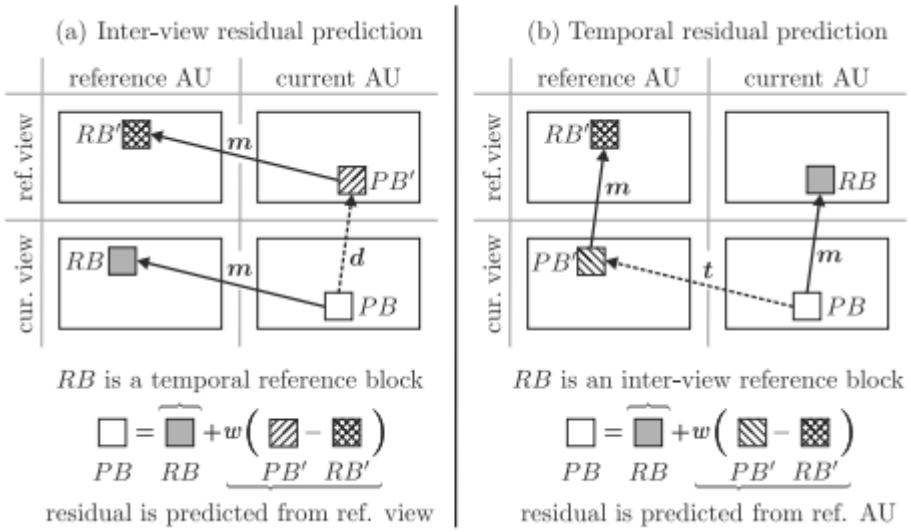


FIG. 3.10-Illustration d'ARP Temporelle et ARP Inter-vue.

En ARP, puisque le bloc résiduel doit être calculé à la volée. Effectuer une simple compensation de mouvement supplémentaire présente une grande complexité de calcul et plus d'accès à la mémoire. Pour résoudre ces problèmes, une conception d'une complexité assez fiable a été adoptée en 3D-HEVC avec des optimisations dans divers aspects : par exemple, l'interpolation bilinéaire est utilisée pour la compensation de mouvement du bloc courant, ainsi que, la génération du bloc résiduel.

3.3.2 Advanced Depth Coding in 3D-HEVC

Bien que la vidéo où l'information texture dans une vidéo Multi-View plus Depth (MVD) est directement affichée à l'utilisateur après le décodage, les cartes de profondeurs (Depth Map) sont utilisées pour générer des vues intermédiaires supplémentaires a de nouveau point de vue. Cette utilisation particulière de carte de profondeur, est combinée avec le fait que les cartes de profondeurs ont des caractéristiques qui sont différentes des données vidéo (par exemple, des grandes zones homogènes et des arrêts vivants), qui nécessitent des nouvelles méthodes de codages pour être pris en considération. Un état d'art de l'ensemble des outils de codage à base de profondeur, qui a été adopté dans 3D-HEVC est fournis dans cette section [12] [17] [18]. .

Il est à noter que la reconstruction des cartes de profondeurs n'est pas directement visualisée, mais plutôt utilisée pour synthétiser des vues textures supplémentaires, le taux de distorsion pour le codage de cartes de profondeurs est modifié à partir d'une formulation traditionnelle. Au lieu de l'optimiser directement selon la distorsion de la profondeur, une nouvelle optimisation de vue synthétisée (VSO : View Synthesis Optimization) est utilisée, et prend en considération la distorsion de vues vidéo intermédiaires. Les cartes de profondeurs sont encodées d'une manière à fournir une meilleure qualité pour des vues intermédiaires avec un taux minimal.

3.3.2.1 Carte de profondeur et la prédiction de disparité

Bien que les cartes de profondeurs présentent des caractéristiques différentes pour décrire la même scène, mais également ils partagent certaines caractéristiques communes, comme des vecteurs de mouvements similaires dans le temps de chaque point de vue, mais aussi des vecteurs de disparité similaires à travers des vues. En général, une vidéo et ses cartes de profondeurs associées, ont des vecteurs de mouvements de disparités identiques. Cependant, dans le codage vidéo, l'estimation de vecteur de mouvement et la disparité, qui minimise le taux global pour coder les données de mouvements/disparités ainsi que les informations résiduelles, sont recherchées. En conséquence, le vecteur de mouvement/disparité utilisé pour coder la vidéo et ses composantes de profondeurs, peut aussi dériver de vrais mouvements de la scène, ainsi que la différence en deux composantes.

Par conséquent, le concept général de la prédiction des composantes des mouvements de la disparité (MPC et DCP) est également appliqué pour les cartes de profondeurs, mais avec quelques changements pour des candidats spécifiques. Pour un mouvement similaire dans la vidéo, les paramètres de mouvement hérités de la carte de profondeur (MPI) sont utilisés, ce qui introduit un candidat de texture, en ce qui concerne le Mode Merge pour le codage de la carte de profondeur, ce qui permet la transmission des paramètres de mouvement à partir du signal de texture. Similaire à la prédiction de mouvement dans le codage de la texture, la prédiction de mouvement de la carte de profondeur est obtenue par l'ajout d'un nouveau candidat dans la liste des candidats de Mode Merge. Les candidats supplémentaires comprennent un candidat inter-vu de Mode Merge, et un Sub-PB paramètre de mouvement hérité (MPI).

Le candidat inter-vue de Mode Merge est similaire à celui obtenu pour le codage de la texture, avec un vecteur de disparité étant converti à partir d'une valeur de profondeur par défaut. Le candidat MPI, cependant utilise la dérivation des informations de mouvements de profondeurs des blocs Co-localisé de la texture déjà codée de la même vue. Ce processus de dérivation peut être également appliqué au niveau des Sub-PB, dans lequel Sub-PB identifie sa région co-localisé dans les images textures codées afin de saisir le vecteur de mouvement.

3.3.2.2 le codage intra des cartes de profondeurs

Pour mieux représenter la caractéristique particulière des cartes de profondeurs, chaque bloc de la carte peut être géométriquement partitionné et représenté plus efficacement en 3D-HEVC. Ces partitionnements non rectangulaires sont désignés collectivement comme des modes de modélisation de profondeurs (DMM : Depth Modeling Mode). Ici, deux méthodes pour séparer un bloc de la carte de profondeur en deux différentes partitions non rectangulaire sont appliquées. Chaque méthode a son propre model de partitionnement : d'abord, un « **Wedgelet** » à ligne droite, comme il est montré dans la figure 3.11 (**a**) utilise conjointement avec la signalisation « **Wedgelet** » direct. Ici le meilleur « **Wedgelet** » est demandé, ce qui donne une distorsion minimum entre le signal original et le rapprochement « **Wedgelet** ». Ces informations de partitionnement ne sont pas prédites à partir de blocs voisins, mais plutôt prises dans une liste de model possibles. D'autre part, un contour tel qu'il est représenté sur la figure 3.11 (**b**) en utilisant entant que modèle de partitionnement. Le contour est prédit à partir du bloc texture co-localisé et peut avoir n'importe quelle partitionnement [12] [18].

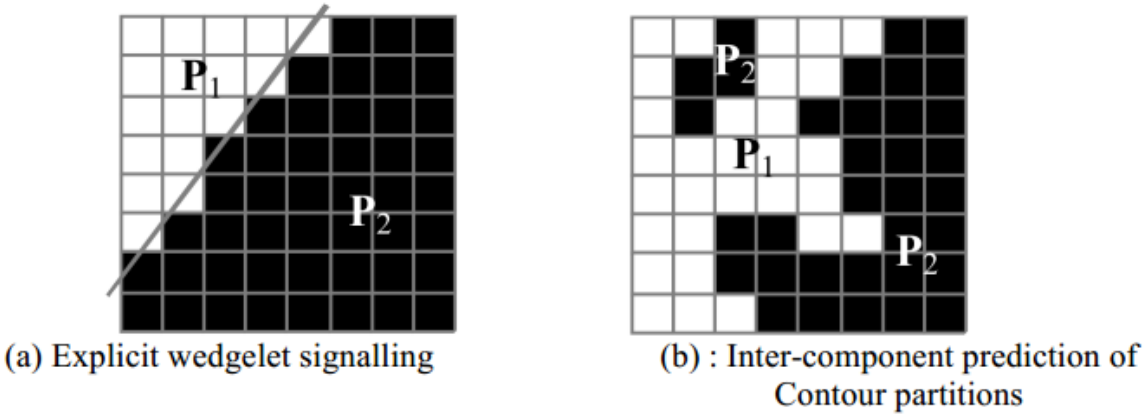


FIG. 3.11-Différents modèles de partitionnement du Bloc de carte de profondeur. (a) : Ligne droite. (b) : la dérivation d'un contour arbitraire du bloc de texture co-localisé.

Pour le partitionnement, tout d'abord, un seuil est calculé comme moyenne de l'échantillon de quatre coins du bloc (carte de profondeur pour **Intra_Wedge** et Texture pour **Inta_Contour**), puis un modèle binaire représente le SBP (Sub Bloc Partition) est obtenu en comparant les valeurs d'échantillon du bloc où la valeur de seuil a été prise. En raison de seuillage, la superficie d'un SBP peut également être disjointe. Apres la génération du modèle binaire qui définit les deux SBP, le même procédé de prédiction de l'échantillon est appliqué dans **Intra_Wedge** et **Intra_Contour**. Le processus est invoqué pour chacun de deux SBP et fournit une prédiction DC à partir d'un sous ensemble des échantillons décodés (voir la figure 3.12 (a)) des blocs adjacents de PB courant. Comment calcule-t-on la prédiction DC ? Elle dépend de lesquels des échantillons voisins **la**, **ta**, **lc**, et **tc** de PB courant sont également voisins échantillons du SBP courant comme représenté sur la figure 3.12 (**b**).

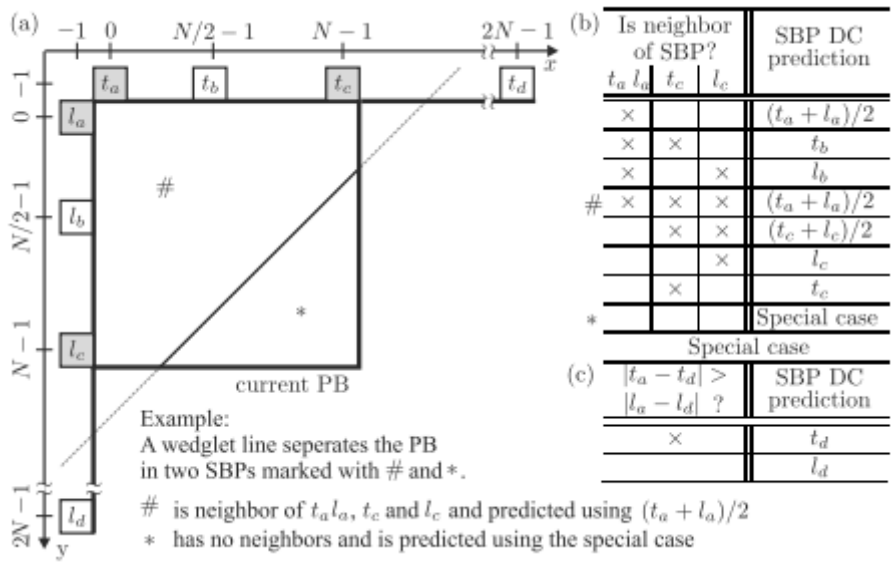


FIG. 3.12-Intra_Wedge et Intra_contour DC prédiction. (a) ta, **tb**, **tc**, **la**, **lb**, **lc** sont les échantillons voisins de courant PB. (b) prédiction dépend sur lesquels de **ta**, **la**, **tc**, et **lc** sont aussi des voisins de SBP courant. (c) les cas spéciaux.

Un cas particulier se produit lorsque aucun de quatre échantillons n'est pas un voisin de SBP courant (comme illustré dans la figure 3.12 par *). On suppose que les limites à l'intérieur de la SBP de PB est aligné avec un bord, qui coupe la frontière d'un bloc voisin, sont entre les positions de **lc** et **ld**, ou entre les positions de **tc** et **td**. Si la différence absolue de **ta** et **td** est supérieure à la différence absolue de **la** et **ld**, les valeurs d'échantillons de la SBP sont égales à td, comme il est indiqué sur la figure 3.12 (c). Sinon elles sont égales à **ld**.

Chapitre 4 : Implémentations & résultats

Chapitre 4 : Implémentations & résultats

Les algorithmes ont été mis en œuvre dans le codeur H.265 version HTM-14.1 [19], et testés en stricte conformité avec les conditions d'essais communs qui spécifient les conditions, les résultats à tirer, vers la fin de l'amélioration et les paramètres à respecter pour les tests [20]. Les séquences sur lesquelles les tests seront déroulés, sont listées dans le tableau 4.1. Les simulations sont effectuées sous un système de 64-bits, plate-forme Windows 7 professionnel avec un processeur Inter(R) Core™ i5 2.7Ghz.

Séquence	Résolution	Vue d'entrée	Nombre d'image
Balloons	1024 x 768	1-3-5	300
Kendo	1024 x 768	1-3-5	300
PoznanHall2	1920 x 1088	7-6-5	200
PoznanStreet	1920 x 1088	5-4-3	250
Shark	1920 x 1088	1-5-9	300

TAB. 4.1-les séquences de tests.

4-1 Fast encode decision for texture coding in 3D-HEVC

Dans cette partie nous améliorerons le codage des vues dépendantes en se basant sur les statistiques, comme exemple : l'utilisation du mode Merge et d'un niveau de subdivision pour les vues dépendantes, mais en se basant aussi sur des observations qui se résument sur la forte corrélation entre les vues dépendantes prise en même instant.

L'idée globalement est de minimiser le calcul du coût pour tous les modes ainsi que pour tous les niveaux de subdivisions (des blocs de taille 64, 32, 16, 8, 4) utilisés par le codeur 3D-HEVC.

4-1-1 Analyse statistique

La probabilité de la sélection de Mode Merge et le niveau de subdivision sont d'abord rapportés. La corrélation d'inter-vue dans le Mode Merge et la subdivision du bloc sont aussi analysées. Les séquences de tests sont listées dans le tableau 4.1. Toutes les statistiques sont effectuées sous les conditions de tests communes de 3D-HEVC [20].

4-1-1-1 Analyse le Mode Merge

Le tableau 4.2 montre la probabilité de la sélection du Mode Merge pour les vues dépendantes de textures de chaque séquence de test énumérée dans le tableau 4.1. À partir du tableau 4.2, on peut voir que la probabilité moyenne de sélection de Mode Merge pour les vues dépendantes est 97.1%. Bien que la probabilité de sélection de Mode Merge dans les vues dépendantes est élevée, mais le codeur de la 3D-HEVC doit encore examiner le RD-Cost de tous les modes inter et intra. Dans la plupart des cas, ces opérations pour le contrôle de modes inter et intra, peuvent être évitées sans sacrifier les performances de codage. Par conséquent, un **Fast Encode Decision** pour un CU est faisable, et il est efficace pour un gain de temps en codage [21].

Séquence	Merge mode(%)
Balloons	97.3
Kendo	96.5
PoznanHall2	97.5
PoznanStreet	97.5
Shark	97.6
Moyenne	97.08

TAB. 4.2-La probabilité de sélection Mode Merge pour les vues dépendantes.

4-1-1-2 le niveau de subdivision des blocs

Pour déterminer le meilleur mode de CTU, le HTM examine exhaustivement toutes les subdivisions possibles pour un CU allant de 0 à 3 (de la taille 64x64 jusqu'à 4x4). Le tableau 4.3 montre les probabilités de tous les quatre niveaux de vues dépendantes. Pour chaque Cu de chaque séquence.

A partir du tableau 4.3, on peut voir que les probabilités de subdivision suivant les niveaux 0, 1, 2, et 3 sont 77.08%, 17.28%, 4.56%, et 1.06%, respectivement. La probabilité de subdivision de niveau 0 et 1 est de 95.08%, et la probabilité de la subdivision pour le niveau 3 est seulement 1.06%. On peut en déduire que la vérification des performances de prédictions des derniers niveaux de subdivisions est généralement pas nécessaire, en particulier pour le dernier niveau de subdivision [21].

Séquence	Niveau 0 (%)	Niveau 1 (%)	Niveau 2 (%)	Niveau 3 (%)
Balloons	68.2	24.1	6.1	1.5
Kendo	73.9	20.5	4.6	1.0
PoznanHall2	84.2	12.7	2.8	0.2
PoznanStreet	83.8	11.2	4.0	1.0
Shark	75.3	17.9	5.3	1.6
Moyenne	77.08	17.28	4.56	1.06

TAB. 4.3-La probabilité de subdivision pour chaque niveau pour les vues dépendantes.

Afin d'analyser la corrélation entre le niveau de subdivision de CU courant et le niveau de subdivision de CU voisin dans l'inter-vue, le tableau 4.4 montre la probabilité des CU ayant un niveau de subdivision inferieur ou égale au maximal des niveaux de subdivision de cinq blocs voisins dans inter-vue, comme indiqué sur la figure 4.1 [21]. On peut voir qu'à partir de tableau 4.4, la précision moyenne est de 94.83%.

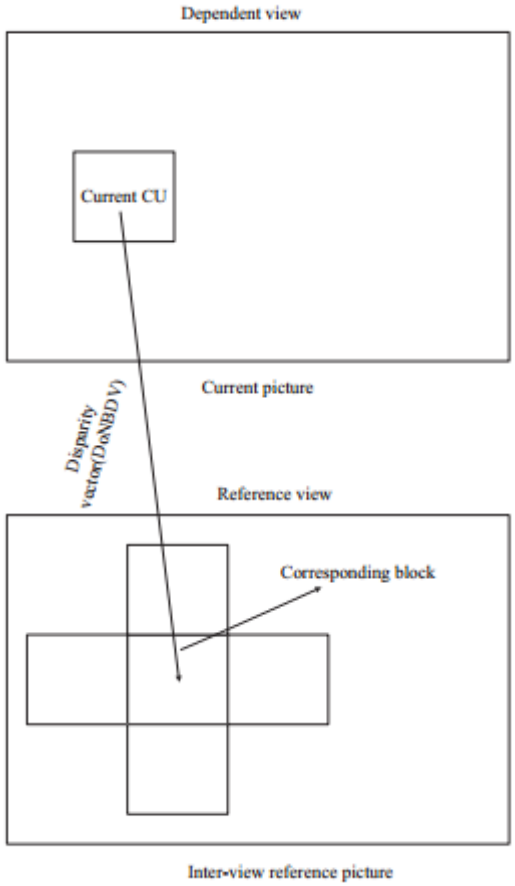


FIG. 4.1-Les cinq blocs voisins de bloc CU courant.

Séquences	Exactitude de la Prédiction (%)
Balloons	93.1
Kendo	95.4
PoznanHall2	96.7
PoznanStreet	89.8
Shark	96.9
Moyenne	94.38

TAB. 4.4- La précision de subdivision d'un CU en exploitant inter-vue.

4-1-2 Algorithme & résultats

Il y a une forte corrélation inter-vue entre les modes de codages et la structure de subdivision de différentes vues. Par conséquent, un algorithme de décision rapide du codage des vues dépendantes avec exploitation des corrélations inter-vue, est représenté dans la figure 4.2 [21]. Il utilise deux stratégies pour accélérer la décision de codeur H.265. La première consiste à prendre une pré-décision de Mode Merge, et l'autre consiste à déterminer le niveau de subdivision.

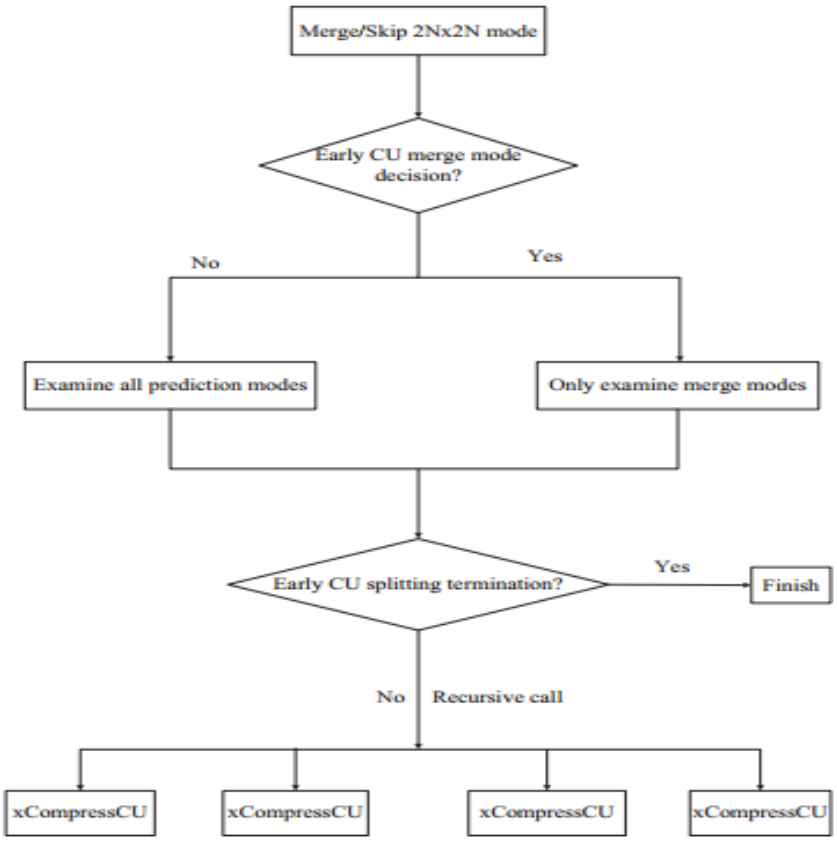


FIG. 4.2- l'algorithme de pré-décision de mode Merge et le niveau de subdivision.

Les deux stratégies sont basées sur deux conditions :

- ✓ Tous les cinq blocs inter-vus voisins de CU courant (comme le montre la figure 4.1) sont codés en Mode Merge, alors le CU courant va être codé en mode Merge.
- ✓ Le niveau de subdivision de CU courant est égal au max des niveaux de subdivision des cinq blocs inter-vue voisins de CU courant.

Le tableau 4.5 montre les résultats d'exploitation seulement pour la pré-décision du mode Merge, on remarque qu'on a pu avoir une réduction à 31,5 % au niveau du temps, avec une petite augmentation du taux de bit : 0.88%, et une dégradation au niveau de la qualité avec 0.35db.

Séquences	Réduction du temps(%)	Taux de bit(%)	Qualité (db)
Balloons	-33.5	0.4	-0.2
Kendo	-36.8	1.2	-0.3
PoznanHall2	-16.7	1.1	-1
PoznanStreet	-15.1	0.3	-0.2
Shark	-38.9	1.1	-0.2
Moyenne	-31.5	0.88	-0.35

TAB. 4.5- Les résultats de la pré-décision de mode Merge.

Le tableau 4.6 montre les résultats de la pré-décision de niveau de subdivision des CUs pour les vues dépendantes, on a bien pu gagner 38.7% au niveau du temps, avec une augmentation pas assez forte de 0.81% pour le taux de bit, avec une dégradation normale de la qualité : 0.4db.

Séquences	Réduction du temps(%)	Taux de bit(%)	Qualité (db)
Balloons	-40.2	1.4	-0.7
Kendo	-42.0	1.4	-0.8
PoznanHall2	-28.2	0.1	-0.4
PoznanStreet	-32.2	0.1	-0.2
Shark	-43.1	0.2	-0.2
Moyenne	-38.7	0.81	-0.4

TAB. 4.6- Les résultats de subdivision décision.

Le tableau 4.7, représente les résultats de combinaison des deux stratégies de pré-décisions, celle de mode Merge et celle de la subdivision de blocs, on a eu un niveau de réduction du temps de 21%, qui est assez satisfaisant, avec une augmentation du taux de bit : 2.7%, et une dégradation presque négligeable pour l'œil humain.

Séquences	Réduction du temps(%)	Taux de bit(%)	Qualité (db)
Balloons	-21.1	2.9	-0.8
Kendo	-24.4	3	-0.7
PoznanHall2	-10.0	1.89	-0.3
PoznanStreet	-14.9	2.4	-0.4
Shark	-25.7	3.3	-0.9
Moyenne	-20.9	2.70	-0.62

TAB. 4.7- Les résultats de la combinaison des deux améliorations.

On remarque très bien que le taux de bit qui est un facteur très important dans l'amélioration, est très bas au niveau la première stratégie pré-décision de mode Merge et la deuxième stratégie de pré-décision de niveau de subdivision, par contre lorsque on a combiné les deux stratégies, on a eu un taux de bit égal à 2.8% qui est un mauvais résultat par rapport à des autres améliorations qui ne dépassent pas 1% en taux de bits, mais cela ne nous empêche pas de dire que ces résultats en général sont assez satisfaisants.

4-2 Low complexity depth mode decision for 3D-HEVC

Le but de cette implémentation, est de faire une prédiction des modes de codages, inter et intra, en se basant sur le critère de la complexité des régions de carte de profondeurs. La formule (4.1) représente la complexité de CU courant, (x, y) sont les coordonnées de la position de CU courant, D_r (i, j) est la valeur de la carte de profondeur du CU courant relativement à la position (i, j), et D_a(x, y) est la moyenne des valeurs de CU courant. TDC permet de calculer la complexité des CU de taille 64 x 64 afin de décider, si la région est simple ou bien complexe. Dans la suite on va voir la relation entre la complexité d'une région et les modes de prédictions, en se basant sur des statistiques pour les deux modes codages inter et intra. Le seuil de la complexité est fixé à 40, si la DCT d'un CU est inférieure ou égale à 40, il est donc classé comme une région simple, sinon il est classé comme une région complexe, voir la formule (4.2) [22].

$$TDC(x,y) = \frac{1}{64 * 64} \sum_{i=1}^{64} \sum_{j=1}^{64} (D_r(i,j) - D_a(x,y))^2 \qquad (4.1)$$

$$\begin{cases} TDC <= T \;\; Treebloc \in simple\, deph\, region \\ TDC > T \;\; Treebloc \in complexe\, deph\, region \end{cases}, T = 40 \qquad (4.2)$$

4-2-1 Fast depth inter mode selection

Pour chaque CU de la carte de profondeur, le codeur 3D-HEVC va examiner le RDO-Cost de, Merge Mode, Skip Mode, Inter 2N x 2N, Intra 2N x 2N, Inter 2N x nU, Inter 2N x nD, Inter 2N x nR, Inter nL x 2N, DMM/RBC 2N x 2N, DMM/RBC N x N, afin de choisir le mode le plus performant pour coder un CU. Donc pour éviter les calculs de RDO-Cost inutiles, on va prédire le mode de codage, en se basant sur des statistiques de la relation entre le mode de codages et la complexité d'un CU.

Le tableau 4.6 montre pour des régions simple, le pourcentage des modes de codages sélectionnés comme meilleurs modes, il est clair que pour des régions simples les modes de codages les plus utilisés, sont Skip Mode et Mode Merge, par contre les autres modes de codages sont presque négligeables. De la même manière, dans le tableau 4.7, on aperçoit que les pourcentages des modes de codages sélectionnés comme meilleurs modes pour une région complexe sont importantes, ce qui signifie, qu'on ne peut pas les négligés. Par conséquent pour une région complexe tous les modes de codage sont pris en considération [22].

Séquences	Mode Skipe (%)	Mode Merge (%)	Inter 2Nx2N(%)	Inter 2NxN(%)	Inter Nx2N(%)	Modes inter asymetric(%)	Modes Intra (%)
Kendo	95.7	2.7	0.7	0.2	0.3	0.3	0.1
Balloons	96.5	2.0	0.5	0.2	0.3	0.4	0.1
Shark	89.3	4.8	1.7	0.8	0.9	0.7	1.7
PoznanStreet	96.6	1.9	0.5	0.1	0.1	0.6	0.2
PoznanHall2	99.0	0.6	0.1	0.1	0.1	0.1	0.0
Moyenne	94.7	2.7	0.8	0.4	0.4	0.4	0.6

TAB. 4.6- Les modes Inter prédictions pour des régions simples.

Séquences	Skipe Mode(%)	Merge Mode(%)	Inter 2Nx2N(%)	Inter 2NxN(%)	Inter Nx2N(%)	Small inter modes(%)	Intra modes(%)
Kendo	54.1	7.3	5.7	4.7	6.4	17.1	4.6
Balloons	56.2	6.9	5.1	4.2	6.1	18.3	3.2
Shark	29.3	10.6	6.5	7.3	7.5	28.4	10.4
PoznanStreet	51.9	7.6	4.1	3.1	6.2	21.6	5.5
PoznanHall2	63.4	7.1	2.6	1.8	3.1	16.2	5.8
Average	47.2	8.4	5.4	4.8	6.2	21.6	6.4

TAB. 4.7- Les modes Inter prédiction pour des régions complexes.

4-2-2 Adaptive Depth intra décision

Dans cette partie on va établir une liaison entre le mode de codage intra pour la carte de profondeur, à savoir intra-HEVC, DMM, et la complexité d'une région. Le tableau 4.8 montre le pourcentage des meilleurs modes de codage intra pour tous les simple CU, on remarque que pour une région simple le mode HEVC est sélectionné à 99.6% comme meilleur mode, par contre pour une région complexe tous les modes intra (HEVC, DMM) sont pris en considération [22].

Séquences	Région simple		Région complexe	
	HEVC (%)	DMM (%)	HEVC (%)	DMM (%)
Kendo	99.70	0.30	81.10	18.90
Balloons	99.60	0.40	77.80	22.20
Shark	99.80	0.20	80.90	19.10
PoznanStreet	99.50	0.50	75.70	24.50
PoznanHall2	99.70	0.30	80.20	19.80
Moyenne	99.60	0.40	78.60	21.40

TAB. 4.8-Les modes intra prédictions pour des simples régions et des complexes régions.

4-2-3 Algorithme et résultats

Selon l'analyse ci-dessus, y compris les stratégies basées sur une sélection rapide de mode intra et inter, l'algorithme de décision de mode de codage basé sur le critère de la complexité est comme suivant :

1. Pour chaque CU de taille 64 x 64 calculer la TDC par la formule (4.1).
2. Déterminer si la région est simple ou bien complexe par la formule (4.2).
3. Pour mode inter, si la région est simple, seulement Skip Mode et Mode Merge seront évalués, sinon tous les modes seront évalués.
4. Pour mode intra, si la région est simple, seulement le conventionnel intra HEVC va être utilisé, sinon tous les modes intra de 3D-HEVC (HEVC, DMM) seront examiner.

En ce qui concerne les résultats, les tests sont effectués sur les séquences montrées dans le tableau 4.1. Le tableau 4.9 montre les résultats de deux stratégies d'amélioration de codage inter et intra pour les cartes de profondeurs.

Concernant les résultats du tableau 4.9, on a pu gagner 24% du temps de codage, avec une augmentation faible au niveau du taux de bit :1.16%, et une dégradation vraiment négligeable au niveau de qualité : 0.06db qui est trop faible, ce qui nous a permis de conclure, que cette amélioration de codage intra et inter des blocs de la carte de profondeur est une amélioration assez forte, puisque on a pu gagner au niveau du temps avec des pertes négligeables au niveau de la qualité et du taux de bits.

Sequence	Taux de bit (%)	Qualité (db)	Réduction du temps(%)
Kendo	0.82	-0.05	-22.10
Balloons	1.07	-0.07	-23.50
Shark	1.21	-0.05	-18.15
PoznanStreet	1.15	-0.08	-24.54
PoznanHall2	1.52	-0.04	-30.90
Moyenne	1.16	-0.06	-23.84

TAB. 4.9- Les modes Intra prédiction pour des régions simples et pour des régions complexes.

Conclusion

Conclusion

L'optimisation des algorithmes de codages des vidéos 3D, spécialement, la représentation MVD (Multi-View Video plus Depth) qui est loin d'être résolu, principalement avec la nouvelle norme 3D-HEVC.

Au cours de mon stage de master au laboratoire **LabSiv**, j'ai commencé un état d'art sur la norme H.264 AVC pour le codage de vidéos 2D, afin d'avoir une base solide sur le codage vidéo. On a commencé a étudié la nouvelle norme qui a été finalisée en Janvier 2013, dont le but d'améliorer de plus le codage de vidéo 2D avec certaines conditions :
- Avoir 50% de réduction de taux de bits par rapport à la norme précédente H.264
- Garder la même qualité, avec une augmentation de la complexité jusqu'à quatre fois au maximum.

Par la suite, on a attaqué l'extension 3D de HEVC, qui a été finalisée en février 2015 après la finalisation de MV-HEVC en février 2014, la première extension Multi-vue de HEVC**.** 3D-HEVC est basé principalement sur HEVC et sur des nouvelles fonctionnalités qui permettent de traiter la corrélation inter-vue pour les vues dépendantes.

Comme la simulation, et implémentation des algorithmes d'amélioration de codage de vidéos 3D, on a étudié dans un premier temps le problème de codage inter pour les deux types images (Texture et carte de profondeur) sous forme d'analyse statistique afin de prouver l'origine de l'idée d'amélioration. Après l'implémentation des deux algorithmes, on a eu des résultats satisfaisants au niveau du temps avec une petite augmentation en taux de bit, et une diminution négligeable au niveau de la qualité surtout pour la deuxième amélioration de codage de la carte de profondeur, qu'on la considère comme meilleure par rapport à la première amélioration de codage des textures.

Comme perspective, l'amélioration de codage des vidéos 3D (3D-HEVC), ne se comporte pas seulement sur le codage inter et intra, mais aussi sur la décision de la taille de bloc de prédiction, au lieu de calculer le RDO-cost de tous les niveaux de subdivision (64, 32,16, 8, 4), soit pour le mode intra ou bien le mode inter, on cherche à réduire le calcul de RDO-cost, par une pré-décision de la taille des blocs. Comme on peut faire d'autres améliorations sur la prédiction angulaire, par une prédiction de la direction de la variation de chaque bloc en utilisant l'opérateur Sobel, au lieu de calculer RDO-cost de 33 directions, on aura besoin que de cinq directions, celle choisi par l'opérateur sobel et deux autre directions, à gauche et à droite de la direction choisi pour la précision de la prédiction angulaire.

Bibliographie

- [1] Olivier Brouard. ‘’Pre-analyse de la vidéo pour un codage adapte. Application au codage de la TVHD en flux H.264’’.
- [2] Mohammed BENABDELLAH. ‘’Outils de compression et de crypto-compression: Aplications aux images fixes et video ’’.
- [3] IAIN E. RICHARDSON. ‘’ The H.264 advanced video compression standard, second edition” WILEY.
- [4] Maxim P.Sharabayko, Oleg G.Ponomarev, "Fast Rate Estimation for RDO Mode Decision in HEVC". OPEN ACCESS entropy 19 December 2014.
- [5] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Trans. Circuits and Systems for Video Technology, 22(12), 1649-1668, (Dec. 2012).
- [6] K. R. Rao, Do Nyeon Kim, Jae Jeong Hwang. “Video Coding Standards, AVS China, H.264/MPEG-4 PART 10, HEVC, VP6, DIRAC and VC-1”. Springer, Signals and Communication Technology.
- [7] Vivienne Sze, Madhukar Budagavi, Gary J. Sullivan. “High Efficiency Video Coding (HEVC): Algorithms and Architectures”. Springer, Integrated Circuits and Systems.
- [8] Mathias Wien, “High Efficiency Video Coding, Coding Tools and Specification”. Springer, Signals and Communication Technology.
- [9] Jani Lainema, Frank Bossen, Woo-Jin Han, Junghye Min, and Kemal Ugur, “Intra Coding of the HEVC Standard”. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 22, NO. 12, DECEMBER 2012.
- [10] Sapna VASUDEVAN, “Implementation of fast residuel quadtree coding and fast intra prediction in High Efficiency Video Coding” December 2013.
- [11] Mark R. Pickering, “Stereoscopic and Multi-View Video Coding”. Academic Press Library in Signal Processing Volume 5, 2014, Pages 119-153.
- [12] Gerhard Tech, Ying Chen, Karsten Müller, Jens-Rainer Ohm, Anthony Vetro, Ye-Kui Wang, "Overview of the Multiview and 3D Extensions of High Efficiency Video Coding". IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 26, NO. 1, JANUARY 2016.
- [13] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, R. Tanger, “Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability”. Signal Processing: Image Communication22 (2007) 217–234.
- [14] Lianlian Jiang,Jiangqian He, Nan Zhang, Tiejun Huang, “An Overview of 3D Video Representation and Coding”. Springer, 3D.

- [15] Karsten Müller, Philipp Merkle, Gerhard Tech, Thomas Wiegand, "3D VIDEO FORMATS AND CODING METHODS". Proceedings of 2010 IEEE 17th International Conference on Image Processing September 26-29, 2010, Hong Kong.
- [16] Philipp Merkle, Aljoscha Smolic, Karsten Müller, Thomas Wiegand, “Multi-view video plus depth representation and coding”. 2007 IEEE International Conference on Image Processing, Sept. 16 2007-Oct. 19 2007.
- [17] Elie Gabriel Mora, “Multiview video plus depth coding for new multimedia”.Gerhard Tech, Krzysztof Wegner, Ying Chen, Sehoon Yea, “3D-HEVC Test Model 2”, JCT3VB1005_d0, 13–19 Oct. 2012.
- [18] Karsten Müller, Senior, Heike Schwarz, Detlev Marpe, Christian Bartnik,Sebastian Bosse, Heribert Brust, Tobias Hinz, Haricharan Lakshman, Philipp Merkle, Franz Hunn Rhee, Gerhard Tech, Martin Winken, and Thomas Wiegand, “3D High-Efficiency Video Coding for Multi-View Video and Depth Data”. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 22, NO. 9, SEPTEMBER 2013.
- [19] JCT-3V. (May 2015). MV- and 3D-HEVC Reference Software, HTM-14.1. https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-14.1/.
- [20] K. Mueller, A. Vetro, Common test conditions of 3DV core experiments, in:Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) document JCT3V-G1100, 7th Meeting, San José, US, January, 2014.
- [21] Na Zhang, Debin Zhao, Yi-Wen Chen, Jian-Liang Lin, Wen Gao. “Fast encoder decision for texture coding in 3D-HEVC”. Signal Processing: Image Communication Volume 29, Issue 9, October 2014, Pages 951–961.
- [22] Ming Chen, Yongshuang Yang, Qiuwen Zhang, Xiaoxin Zhao, Xinpeng Huangb, Yong Gan, ” Low complexity depth mode decision for HEVC-based 3D video coding”. Optik - International Journal for Light and Electron Optics, Volume 127, Issue 11, June 2016, Pages 4758–4767.