Ministère de l'enseignement Supérieur et de la recherche Scientifique

وزارة التعليم العالي والبحث العلمي

Badji Mokhtar Annaba University
Université Badji Mokhtar – Annaba
Faculté de Technologie

Département Electronique

جامعة باجي مختار– عنابـــة

كـليـة التكنـولـوجيـا

قســم الإلكترونيك

Thèse

Présentée pour obtenir le diplôme de

Doctorat En-Sciences

Spécialité : Electronique

Par :

Brai Radhia

Thème :

Optimisation du HEVC basée sur la réduction de la complexité calculatoire. Application à l'IPTV pour le contrôle de la congestion et l'amélioration du QoS

Thèse soutenue le mardi 19 novembre 2024

Devant le jury composé de :

Président:	Pr.	BENOUARET	Mohamed	UBMA Annaba
Directeur de thèse :	Pr	DOGHMANE	Noureddine	UBMA Annaba
Co-directeur :	pr	BEKHOUCH	Amara	Université de Souk- Ahras
Examinateur :	Dr	BOUKAACHE	Abdenour	Université Guelma
Examinateur :	Pr	BOUDEN	Toufik	Université de Jijel
Invité :	Dr	KOUADRIA	Nasreddine	UBMA Annaba

Remerciements

Je suis reconnaissante à mon Dieu "الله " pour m'avoir donné la santé, la patience,
La connaissance et toutes les ressources nécessaires pour terminer cette thèse.

Au terme de ce travail, je suis bienheureuse de pouvoir exprimer toute ma gratitude envers les personnes qui m'ont aidé en contribuant à l'aboutissement de cette thèse.

Je tiens à exprimer ma gratitude particulière à mon directeur de thèse,
le professeur « DOGHMANE Noureddine », qui m'a proposé le sujet guidée pendant ces années dans mes activités de recherche.

Je vous remercie d'avoir écouté et guidé sans oublier vos qualités humaines.

Je tiens à remercier, Monsieur « BEKHOUCH Amara », M.C.A à l'Université de Souk-Ahras, pour son soutien et ses remarques constructives tout au long de ce travail.

Enfin, un grand merci à notre source d'amour ma mère AKILLA.

Et ma famille.

``تحسين HEVC بناءً على تقليل التعقيد الحسابي: التطبيق على IPTV للتحكم في الازدحام وتحسين جودة الخدمة ''

الملخص:

يوفر HEVC ما بين 25٪ إلى 50٪ من معدل نقل البيانات، مقارنةً بسابقه، وهو Advanced Video Coding H.264، وبالتالي يدعم دقة تصل إلى UHD. 8 K ومع ذلك، فإن خفض معدلات التدفق التي يوفرها ترميز HEVC يؤدي إلى زيادة التكلفة الحسابية لعملية التشفير .يمكن أن يصبح هذا التعقيد عقبة حقيقية خاصة بالنسبة لتدفق الفيديو في الوقت الحقيقي وأيضًا لتطبيقات VANET (شبكة المركبات المخصصة) مثل السلامة المرورية والمراقبة بالفيديو .يعود التحسن في معدلات التدفق وكذلك الزيادة في تكلفة الحساب إلى استخدام قدرات التشفير والتنبؤ والتحويل الكبيرة والمتعددة الأحجام .في الواقع، يعتمد المبرمج H264 على كتل كبيرة هيكلية بأحجام 4x4 8x8 و 16x16، بينما يعتمد H.265على وحدات شجرة الترميز(CTUs) ، تختار وحدات التحكم في العمليات الأحجام 8×8، 16x16 و 32x32 و 64x64 .يقترح هذا العمل طريقة قرار سريعة بحجم CU (وحدة التشفير) لتقليل تكلفة حساب HEVC على أساس التجانس المكاني .مقارنة بنموذج الاختبار المعياريHM16.13 ، يتم تقليل متوسط وقت التشفير بحوالي 40٪ لتسلسلات الفيديوQCIF / CIF ، و35٪ إلى 43٪ لتسلسلات اختبار الفئة A و Bو C. يتم الحصول على هذه التخفيضات المهمة في وقت التشفير مع فقدان ضئيل للجودة ومتوسط زيادة في معدلات التدفق لا تتجاوز 0.89٪ لأنماط التكوين الثلاثة (الكل داخل، والوصول العشوائي، والتأخير المنخفض)

كلمات مفتاحية:اختيار حجم CU ، تقليل التعقيد، التجانس المكاني ,HEVC/H265

‘’ Optimisation du HEVC basée sur la réduction de la complexité calculatoire. Application à l'IPTV pour le contrôle de la congestion et l'amélioration du QoS’’

Résumé :

HEVC offre des économies de débit de 25 à 50 % par rapport à son prédécesseur, l'Advanced Video Coding H.264, et prend en charge des résolutions allant jusqu'à 8K UHD. Cependant, cette réduction du débit dans HEVC entraîne une augmentation du coût de calcul lors de l'opération d'encodage. Cette complexité peut devenir un véritable obstacle, en particulier pour le streaming vidéo en temps réel et les applications VANET (Vehicular Ad-Hoc Network) telles que la sécurité routière et la vidéosurveillance. L'amélioration du débit et l'augmentation du coût de calcul sont attribuables à l'utilisation de blocs de codage, de prédiction et de transformation de différentes tailles. En effet, le codeur H.264 est basé sur des macroblocs de structure de tailles 4x4, 8x8 et 16x16, tandis que le H.265 dépend des unités d'encodage en arbre (Coding Tree Units - CTU) qui sélectionnent les tailles de blocs 8x8, 16x16, 32x32 et 64x64. Dans ce travail, une méthode de décision rapide de la taille de l'unité de codage (Coding Unit - CU) est proposée pour réduire le coût de calcul du HEVC en se basant sur l'homogénéité spatiale. Comparé au modèle de test de référence HM16.13, le temps de codage moyen est réduit d'environ 40 % pour les séquences vidéo CIF / QCIF et de 35 % à 43 % pour les séquences de test des classes A, B et C. Ces réductions significatives du temps de codage sont obtenues avec une perte de qualité négligeable et une augmentation moyenne du débit qui ne dépasse pas 0,89 % pour les trois modes de configuration (tout intra, accès aléatoire et faible délai).

Mots clés : Choix de la taille de CU, réduction de complexité, homogénéité spatiale. HEVC/H265.

‘Optimization of HEVC Based on Computational Complexity Reduction. Application to IPTV for Congestion Control and QoS Improvement’’

Abstract:

HEVC offers 25 to 50% bitrate savings compared to its predecessor, Advanced Video Coding H.264, and supports resolutions up to 8K UHD. However, the reduction in bitrate provided by HEVC leads to an increase in the computational cost of the encoding operation. This complexity can become a real hindrance, especially for real-time video streaming and VANET (Vehicular Ad-Hoc Network) applications such as road safety and video surveillance. The improvement in bitrate and the increase in computational cost are due to the use of larger and multi-sized coding blocks, prediction, and transformation. In fact, the H.264 encoder is based on macroblocks with structure sizes of 4x4, 8x8, and 16x16, while H.265 depends on Coding Tree Units (CTUs), where CTUs select block sizes of 8x8, 16x16, 32x32, and 64x64. This work proposes a fast Coding Unit (CU) size decision method to reduce the computational cost of HEVC based on spatial homogeneity. Compared to the reference test model HM16.13, the average encoding time is reduced by approximately 40% for CIF/QCIF video sequences and from 35% to 43% for Class A, B, and C test sequences. These significant reductions in encoding time are achieved with negligible loss of quality and an average increase in bitrate that does not exceed 0.89% for the three configuration modes (All Intra, Random Access, and Low Delay).

Key words: Choice of CU size, complexity reduction, spatial homogeneity, HEVC/H265

LISTE DES ACRONYMES

AMVP	Advanced Motion Vector Prediction
BO	Band Offsets
CABAC	Context-adaptive binary arithmetic coding
CB	Coding Blocks
CIF/QCIF	Common Intermediate Format/Quarter Common Intermediate Format
CRT	Cathode Ray Tube
CTB	Coding Tree Block
CTU	Coding TREE Unit
CU	Coding Unit
DCT/ DST	Discrete Cosine Transform/ Discrete Sine Transform
EBSP	Encapsulate Byte Sequence Payload
EO	Edge Offset
GOP	groupe of pictures
HD	High Definition
HEVC	High Efficiency Video Coding
HM	HEVC Test Model
HVS	the human visual system
IP	Internet Protocol
MDC	Multiple Description Coding
MPEG	Moving Picture Experts Group
PSNR	Peak Signal to Noise Ratio
PU	Prediction unit
QOE	Quality of Experience
Qp	Quantization Parameter
RBSP	raw byte sequence payload
RPS	reference picture set
SAO	Sample adaptive offsets
SD	Standard Definition
SEI	Supplemental Enhancement Information
SMCV	Simplified Mass Center Vector
SODB	String of Data Bits
SSIM	structural similarity index measure
TI/SI	Information temporelle/Information spatiale
TU	Transformation unit
UHD	Ultra High Definition
VPS	*Video Parameter Set*
VR	Virtual reality
VUI	video usability information
WPP	*Wavefront parallel processing*

LISTE DES FIGURES

Figure 1. 1 : La représentation des données vidéo 8

Figure 1. 2 : Exemples d'applications vidéo. 9

Figure1 .3 : Espaces colorimétriques RGB et YUV 9

Figure1.4 : La notation de sous-échantillonnage de chrominance. 11

Figure 1.5 : Quelque Représentation de sous-échantillonnage de chrominance 11

Figure 1.6 : Séquence vidéo 12

Figure 1.7 : (a) affichages progressifs, (b) Affichages entrelacés 13

Figure 1.8 : Représentation d'un test de résolution..... 14

Figure 1.9 : Redondance spatiale et temporelle..... 16

Figure 1.10: Schéma bloc d'un processus de codage et de décodage vidéo de base 17

objets de cette manière, les différences entre les images peuvent être encore réduites[8]. 17

Figure 1.11 : Les normes de compression vidéo. 20

Figure 1.12 : H.261 format de trame CIF, macro bloc, GOB. 22

Figure 1.13 : Structure du GOP MPEG-2..... 24

Figure 1.14 : Format d'image H.263 (QCIF). 25

Figure 1.15: Exemple de partitions CTU dans VVC. 30

Figure 2.1 : Représentation NAL units..... 35

Figure 2.2 : Structure de l'en-tête d'unité NAL à 2 octets 35

Figure. 2.3 : Schéma fonctionnel typique d'un encodeur HEVC. 38

Figure 2.4 : Partition d'une image..... 39

Figure 2. 5 : (a) Le partitionnement d'un CTB ,(b) CTU et CTB, (c) CTB divisé en CB 41

Figure 2.6: Illustration des types de fractionnement du PU dans HEVC. 42

Figure 2.7 : Les modes Intra prédiction de HEVC 43

Figure 2.8 : Estimation du mouvement. 45

Figure 2.9: Processus de décision pour le mode PU de HM16.9. 45

Figure 2.10 : la matrice de transformation directe 32x32.....47
Figure 2.11: (a) transformation directe et quantification, (b) transformation inverse et quantification 48
Figure 2.12 : Types et catégories de bords pour la compensation de décalage de bord.....51
Figure. 2.13 : Les Trois opérations clés dans CABAC 53
Figure.4. 1 : Valeur SMCV en fonction du nombre d'CU.....77
Figure 4.2 : Décision du mode CU basée sur l'amplitude du centre de masse.. 78
Figure 4.3 : Structure des données dans HEVC. 79
Figure 4.4 : Création d'un index.....80
Figure 4.5 : Décision du mode CU basée sur l'amplitude du centre de masse. 80
Figure 4.6 : Représentation spatio-temporelle de la séquence vidéo. 84
Figure 4.7 : Diagramme spatio-temporel pour JCT-VC. Séquences des classes A-B-C.....88
Figure 4.8 : SI et TI des vidéos de classe C décodées. 93

LISTE DES TABLEAUX

Tableau 1.1 : Principalement des outils de codage récemment développés utilisés dans H.266/VVC. 28

Tableau 2.1 Types d'unités NAL, significations et classes 36

Tableau 4.1: Le fichier de configuration d'HM 16.13 83

Tableau 4.2 : Les séquences vidéo originales incluent la résolution et les images. 84

Tableau 4.3 : Efficacité de codage HM16.13 sous les configurations RA, LD et Intra 86

Table 4.4 : Les séquences de vidéos de Class-A-B-C-E 87

Tableau 4.5 : Évaluation des performances de la méthode recommandée par rapport aux séquences utilisées 89

Tableau 4.6. Comparaison des performances. 89

Tableau 4. 7 : Valeurs TI et SI des vidéos de classe C décodées. 92

Table des matières

Remerciements i

Liste des acronymes viii

Liste des figures ix

Liste des tableaux xi

Table des matières xii

Introduction générale : 1

CHAPITRE 1 : LES NORMES DE COMPRESSION VIDEO

1.1 Introduction 7

1.2 Représentations de données vidéo 7

1.2.1 Définitions de vidéo 8

1.2.2 Un espace colorimétrique 9

1.2.3 Sous-échantillonnage de la chrominance 10

1.2.4 Chronométrage Vidéo 12

1.2.5 RÉSOLUTION VIDÉO 13

1.3 Compression 15

1.3.1 Les paramètres des optimisations de codage vidéo 18

1.4 Les normes de codage vidéo 19

1.4.1 Le processus d'adoption de la normalisation 20

1.4.2 H.120 21

1.4.3 H.261 21

1.4.4 MPEG-2/DVB 23

1.4.5 H.263 24

1.4.6 MPEG-4 25

1.4.7 H.264/AVC 26

1.4.8 H.265/HEVC 27

1.4.9 H.266/VVC 27

1.5 Conclusion 30

CHAPITRE 2: LE CODAGE VIDEO A HAUTE EFFICACITE (HEVC)

2.1 Introduction 32

2.2 Caractéristiques principales 32

2.3 Abstraction réseau et syntaxe de haut niveau 34

2.3.1 Format de flux d'octets 34
2.4 La couche de codage vidéo (VCL) 37
2.4.1 Images et partitions 38
2.4.2 Transformation et quantification HEVC 46
2.4.3 Filtres « In-Loop » 50
2.4.4 Codage entropique 52
2.5 Conclusion 55

CHAPITRE 3 : LES TRAVAUX RECENTS POUR REDUCTION DE LA COMPLEXITE DU HEVC

3.1 Introduction 57
3.2 Un codage spatio-temporel pour Intercodage HEVC 57
3.3 Algorithme de recherche d'arborescence inter-CU 59
3.4 Technique de détection précoce des sauts 60
3.5 Algorithme de partitionnement CU rapide 61
3.6 Algorithme de décision rapide de la taille de l'intra CU 62
3.7 Détermination précoce de la taille de la CU 62
3.8 Arbres de décision pour la réduction de la complexité 63
3.9 Algorithme basé sur la classification par blocs arborescents 63
3.10 Prédiction efficace de la profondeur CU et du mode PU 64
3.11 Architecture VLSI d'optimisation de la distorsion 64
3.12 Une approche multi-résolution' de Jha et Chaudhary 65
3.13 Algorithme de codage intertrame dans HEVC 65
3.14 Décision de profondeur d'unité d'arbre de codage rapide 66
3.15 Détection d'homogénéité spatiale et temporelle 66
3.16 Décision rapide de taille CU basée homogénéité temporelle 67
3.17 Informations de bord et codage spatio-temporel 67
3.18 Corrélation statistique et spatio-temporelle pour HEVC 67
3.19 Décision bayésienne et l'apprentissage en ligne 68
3.20 Apprentissage automatique pour le codage intra HEVC' 68
3.21 Un algorithme de partition CU rapide 69

3.22 Conclusion.....70

CHAPITRE 4 : REDUCTION DE LA COMPLEXITE DU CODEUR HEVC PAR HOMOGENEITE SPATIAL

4.1 Introduction.....73

4.2 L'algorithme de La décision rapide proposée.....73

4.2.1 Détection de l'homogénéité spatiale d'une CU.....74

4.3 Détection améliorée de l'homogénéité spatiale.....78

4.4 Résultats et discussion.....81

4.4.1 Les Mesures de test.....81

4.4.2 Les Conditions de test pour le codage HEVC.....82

4.4.3 Les Séquences video de test CIF et QCIF.....83

4.4.4 Teste des Séquences des classes A, B, C et E.....86

4.4.5 Évaluation des distorsions spatio-temporelles (classe C).....90

4.5 Conclusion.....93

CONCLUSION GENERALE.....95

REFERENCES.....98

Introduction Générale

Introduction générale :

Aujourd'hui, les contenus médiatiques sont consommés sur une multitude d'appareils et de plateformes différents, allant des téléviseurs intelligents et des smartphones aux ordinateurs et aux tablettes. Cette évolution est motivée par l'émergence, au cours des dernières années, de technologies de compression numérique à haute efficacité qui ont révolutionné le concept de diffusion, de stockage et d'échange de contenus. Parallèlement, l'utilisation de services vidéo tels que la vidéo à la demande (VoD), le streaming vidéo et les communications vidéo personnelles devient courante parmi les utilisateurs, répondant ainsi aux besoins du monde en ligne connecté d'aujourd'hui.

Selon diverses études, la diffusion vidéo représente désormais la majorité du trafic réseau à l'échelle mondiale, notamment sur les réseaux sans fil et cellulaires. Ce volume ne cesse de croître à un rythme très rapide, porté à la fois par le nombre croissant de services et d'utilisateurs, mais également par l'augmentation de la résolution spatiale et temporelle des formats vidéo, et par la demande d'une expérience de meilleure qualité.

Le phénomène de l'augmentation de la résolution a connu une croissance exponentielle au cours de la dernière décennie, grâce aux nouvelles capacités des terminaux d'acquisition, principalement les caméras vidéo, les caméras de Smartphones et même les caméras sportives, qui prennent en charge des résolutions allant du format Définition Standard (SD) au format Haute Définition (HD) jusqu'au format Ultra Haute Définition (UHD) actuel avec plus de 8 méga pixels. Le stockage et la transmission de ces expériences vidéo de haute qualité engendrent des quantités de données considérables. Parallèlement, les appareils capables d'enregistrer, d'afficher et de distribuer des résolutions ultra-élevées sont devenus abordables tant pour les consommateurs domestiques que mobiles. Pour favoriser l'adoption généralisée de ces technologies vidéo immersives, des avancées en matière de compression vidéo sont essentielles.

Au cours des 30 dernières années, le codage vidéo a été un moteur de la recherche académique et industrielle, conduisant au développement de produits et services captivants. Les principes fondamentaux du codage vidéo n'ont pas changé au fil des ans, les codecs et normes de codage vidéo les plus récents reposant toujours sur les principes

de codage hybride basé sur des blocs établis dans la Recommandation H.261 en 1988. Cependant, les avancées dans les outils de codage vidéo ont permis d'obtenir des réductions remarquables du débit binaire tout en maintenant des objectifs de qualité vidéo similaires.

Pour répondre aux exigences croissantes en matière de stockage et de transmission vidéo, le critère de référence pour le développement de nouvelles normes a été fixé à une réduction de 50% du débit binaire. H.264/Advanced Video Coding (AVC) a atteint une réduction de 50% par rapport à Mpeg-2/H.262, et le High Efficiency Video Coding (HEVC) a atteint une réduction de 50% par rapport à H.264/AVC. À l'heure actuelle, la norme HEVC, introduite au début de 2013, représente la pointe de la technologie en matière de codage vidéo.

HEVC introduit de nouveaux outils de codage et améliore les algorithmes existants pour offrir une meilleure efficacité de compression. Parmi les innovations notables, on retrouve la partition en blocs quadtree, une grande variété de tailles de blocs, une densité élevée de modes angulaires, davantage de tailles de transformation et de prédiction, un filtre supplémentaire en boucle pour améliorer la qualité perceptuelle, ainsi qu'un vecteur de mouvement avancé pour l'inter-codage. Cependant, ces outils, à eux seuls, ne garantissent pas une efficacité optimale. Il est essentiel de prendre des décisions optimales à chaque étape de l'encodage, notamment pour la subdivision quadtree, la taille de transformation, le mode angulaire et le vecteur de mouvement, afin d'atteindre les meilleures performances. L'implémentation d'une prise de décision optimale constitue ainsi l'élément clé de l'architecture de l'encodeur.

Traditionnellement, cette tâche a été réalisée en évaluant exhaustivement toutes les combinaisons de paramètres disponibles, et en utilisant une fonction de coût, la combinaison offrant les meilleures performances, c'est-à-dire la meilleure qualité avec la plus faible distorsion, est sélectionnée comme décision optimale. Cependant, le grand nombre de combinaisons disponibles dans la nouvelle norme rend cette prise de décision optimale basée sur la force brute une solution peu pratique pour la plupart des applications de ce domaine, en raison de la complexité informatique extrêmement élevée requise. Au cours des dernières années, la communauté de recherche a déployé des efforts considérables pour résoudre ce problème, et plusieurs nouvelles approches et algorithmes

ont été proposés pour accélérer les différents outils de HEVC. Cette thèse s'inscrit dans ce domaine de recherche, en mettant principalement l'accent sur la conception d'algorithmes à faible complexité pour la réduction du temps de codage HEVC.

Motivation

La complexité de calcul accrue pose un défi significatif pour l'implémentation en temps réel des nouvelles normes de codage vidéo sur les plates-formes embarquées avec des ressources de calcul, de mémoire et d'énergie limitées. L'introduction de nouveaux contenus vidéo en 4K UHD et à 360 degrés avec des résolutions spatiales élevées (8K, 16K) et temporelles (120 images par seconde) complique davantage l'obtention de performances de codage et de décodage en temps réel.

Malgré les avancées dans le domaine des microélectroniques pour les plates-formes embarquées multi-cœurs, l'augmentation exponentielle de la complexité de calcul lors du processus de codage ne peut pas être entièrement compensée. Par conséquent, il est crucial de se concentrer sur les aspects algorithmiques pour réduire la complexité de l'encodeur HEVC tout en maintenant le débit binaire et la qualité offerts par les nouveaux outils de codage.

En raison de la densité d'énergie limitée des batteries des systèmes embarqués modernes, les concepteurs doivent explorer des méthodes pour réduire encore la complexité de calcul et proposer des approches pour adapter et contrôler cette complexité, minimisant ainsi la consommation d'énergie lors du processus de codage HEVC. Les travaux de recherche de cette thèse se focalisent principalement sur la propose d'une méthode de prise de décision rapide de la taille de l'unité de codage (CU) afin de réduire le coût de calcul de HEVC en se basant sur l'homogénéité spatiale.

Contributions de la thèse

L'objectif principal de cette thèse est de résoudre les problèmes liés à la réduction de la complexité de calcul de la norme de compression vidéo HEVC, tout en maintenant un équilibre satisfaisant entre le débit binaire et la qualité vidéo. Plus précisément, les contributions majeures de cette recherche se concentrent sur les aspects suivants :

- Nouvelle méthode adaptative de décision rapide de la taille des CU : Cette thèse propose une nouvelle méthode permettant de prendre rapidement des décisions adaptatives sur la taille de l'unité de codage (CU) dans HEVC. La méthode utilise les caractéristiques vidéo lors du processus de codage et intègre l'homogénéité spatiale pour réduire le coût de calcul lié à la partition en quadtree.

- Étude statistique des séquences vidéo : L'approche proposée commence par une étude statistique utilisant différentes séquences vidéo. Cette étude couvre les quatre angles du plan d'information temporelle/spatiale (TI/SI) et explore plusieurs valeurs de paramètres de qualité (QP). L'objectif de l'étude est d'analyser la relation entre l'amplitude du vecteur de centre de masse simplifié (SMCV) de la CU et les meilleures tailles intra-blocs pour ces différentes séquences.

- Détermination des seuils : Basée sur l'étude statistique, quatre seuils sont déterminés. Ces seuils correspondent aux valeurs de SMCV couvrant au moins 90 % des CUs codées avec des tailles de 64 ou 32, 32 ou 16, 16 ou 8, et enfin 8 ou 4 respectivement. Ces seuils jouent un rôle crucial dans le processus de prise de décision concernant la taille des CUs.

- Prise de décision rapide pour différentes séquences vidéo : Une fois l'étude statistique terminée, la méthode proposée permet une prise de décision rapide pour la taille des CUs de n'importe quelle autre séquence vidéo. Cette décision est prise en fonction des différentes complexités de la texture de la séquence. Ainsi, les CUs peuvent être classés dans plusieurs classes de complexité de texture correspondant à des CUs homogènes de tailles (64×64), (32×32), (16×16), (8×8) ou (4×4) respectivement.

Ces contributions visent à améliorer l'efficacité de la codification HEVC en réduisant les coûts de calcul et en optimisant le processus de décision de la taille des CUs en se basant sur les caractéristiques vidéo et l'homogénéité spatiale.

Organisation de la thèse

En plus de cette introduction, qui sert à la fois de motivation et de présentation générale du problème, le manuscrit est structuré en quatre chapitres, qui sont organisés de la manière suivante :

Dans le premier chapitre, nous aborderons la représentation des données vidéo, en mettant l'accent sur les concepts fondamentaux utilisés dans leur encodage. Nous commencerons par introduire quelques généralités, puis nous examinerons de manière concise les aspects techniques et les paramètres des optimisations de codage vidéo. Ensuite, nous retracerons brièvement l'historique de l'évolution des normes de compression vidéo, en mettant en évidence les points clés de chaque norme, depuis le standard H.261 jusqu'au standard VVC.

Dans le deuxième chapitre, nous aborderons les concepts fondamentaux de la compression à haute efficacité (HEVC). Nous présenterons en détail les caractéristiques principales et les syntaxes de haut niveau de cette norme. De plus, nous examinerons de près les notions clés liées à la couche de codage vidéo (VCL) dans le contexte de la norme HEVC.

Le troisième chapitre offre un état de l'art des différentes techniques de codage de la norme HEVC, axées sur la réduction de la complexité calculatoire. Nous nous penchons particulièrement sur l'étude des méthodes exploitant les caractéristiques vidéo, telles que l'homogénéité spatiale et temporelle. De plus, nous analysons les méthodes qui se concentrent sur l'optimisation des décisions relatives aux unités de codage (CU) et nous les examinons en détail au sein de ce chapitre.

Dans le quatrième chapitre, L'approche que nous avons proposée visant à améliorer le temps d'encodage tout en garantissant un compromis acceptable entre la qualité vidéo et le débit binaire. Nous commencerons par une description détaillée de l'algorithme de décision rapide pour la taille des unités de codage (CU). Ensuite, nous présenterons une méthode améliorée de détection de l'homogénéité spatiale. Pour compléter notre étude, nous exposerons les mesures et les conditions de test utilisées pour le codage HEVC, ainsi que les données employées pour évaluer notre approche. Enfin, nous examinerons les résultats obtenus en utilisant ces données de test.

1 .

CHAPITRE 1 : LES NORMES DE COMPRESSION VIDEO

1.1 Introduction

L'image animée, sous toutes ses formes (cinéma, télévision, vidéo, etc.), est l'une des inventions les plus répandues et les plus avancées de notre époque. Au fil du temps, le développement d'algorithmes de compression performants et de technologies de traitement vidéo a permis une transition réussie du domaine analogique vers le domaine numérique. Aujourd'hui, cette transition est largement intégrée dans de nombreux processus de production et de distribution vidéo, révolutionnant ainsi l'industrie.

L'objectif principal dans la conception des nouveaux systèmes vidéo numériques a été de réduire les exigences en termes de stockage et de bande passante, tout en préservant une qualité supérieure à celle de la vidéo analogique. Cette priorité a permis de relever de nombreux défis, notamment en termes de gestion des ressources et de transmission des contenus.

La compression vidéo est devenue un élément clé de cette évolution, offrant des moyens efficaces de réduire la taille des fichiers vidéo sans compromettre la qualité visuelle. Les méthodes et normes de compression actuelles sont le fruit de recherches approfondies et d'avancées technologiques constantes.

Ce chapitre entame son exploration en fournissant un aperçu des éléments essentiels de la vidéo. Nous examinerons les différents aspects techniques et artistiques qui composent l'image animée, tels que la résolution, le format d'affichage, la fréquence d'images, les espaces colorimétriques, etc.

La deuxième section de ce chapitre se concentre sur les méthodes de compression vidéo utilisées aujourd'hui, ainsi que les normes qui régissent cette discipline.

1.2 Représentations de données vidéo[1]

Les représentations vidéos sont prolongées avec les cinq tendances suivantes, comme le montre la (Fig. 1.1) résolution spatiale (x, y) : la résolution spatiale de la vidéo (x, y) augmente continuellement pour améliorer la clarté de la vidéo on est passe des format initiaux tel que le format(CIF) (320×240) à la définition standard (SD) (720p),puis a la haute définition HD (1080p) au maintenant 4K (3840×2160) et 8K (7680×4320) , ces

résolutions peuvent encore être étendues à des milliards de pixels dépassant ainsi la fidélité de la vision humaine.

2) Angle de vue et profondeur (V_x, V_y, V_z) : les formats vidéos sont développés de la 2D a la stéréo (2-vues), puis aux multivue, vidéo a point de vue libre, 360°VR.

3) Spectre (λ) indiquant la fidélité des couleurs et la résolution d'amplitude : les vidéo ont évolué du noir/blanc, aux video couleur avec RVB.

4) Échantillonnage temporel (t) : Avec le développement des technologies de capture et de photographie computationnelle, la fréquence d'images vidéo est passée de 25/30 images par seconde (fps) pour la vidéo SD à 60 fps pour la vidéo HD, et il est probable qu`elle atteindra bientôt 120fps ou plus.

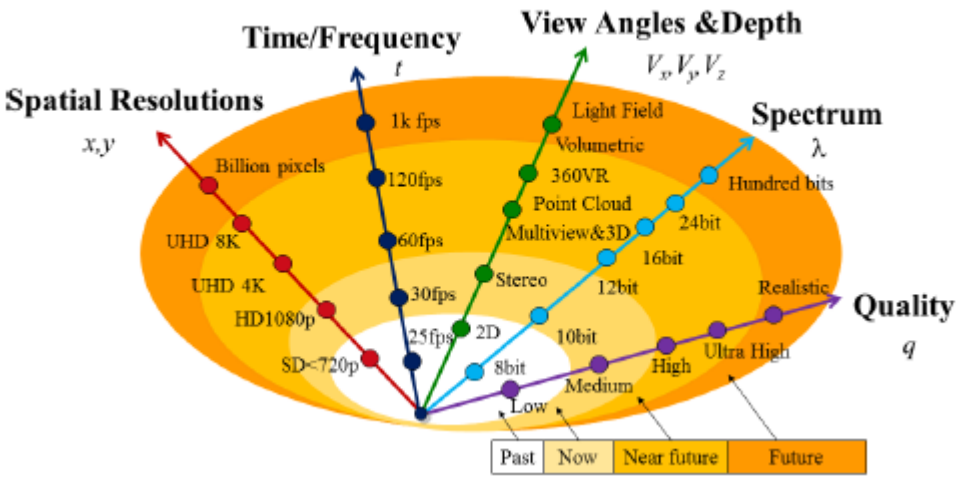


Figure 1. 1 : *La représentation des données vidéo [1]*

1.2.1 Définitions de vidéo

Avec l'essor de la communication et de l'affichage multimédia, de nombreuses applications vidéo ont émergé, telles que la télédiffusion, le cinéma, la vidéo à la demande, la visioconférence, la vidéo mobile, la vidéosurveillance, le contrôle à distance, la robotique, la vidéo 3D et la réalité virtuelle (VR), qui permet des expériences immersives et une perception visuelle réaliste, comme l'illustre la Fig. 1.2. Ces applications vidéo sont désormais largement utilisées dans divers domaines de la vie quotidienne, notamment dans la fabrication, la communication, la sécurité nationale, l'armée, l'éducation, la santé et le divertissement. Aujourd'hui, les données vidéo

représentent une part majoritaire du trafic Internet, et leur volume continue de croître de manière exponentielle chaque année [1, 2].

t 

Figure 1. 2 : *Exemples d'applications vidéo.*

1.2.2 Un espace colorimétrique

C'est une représentation mathématique de la couleur. Initialement, la vidéo ne contenait que des informations in noir et blanc ou en niveaux de gris. Lorsque la diffusion couleur a été développée, des tentatives ont été faites pour transmettre une vidéo couleur en utilisant des données d'espace colorimétrique RVB (rouge, vert, bleu), mais cette technologie occupait beaucoup de bande passante, d'autres espaces colorimétriques alternatifs ont donc été développés.

.

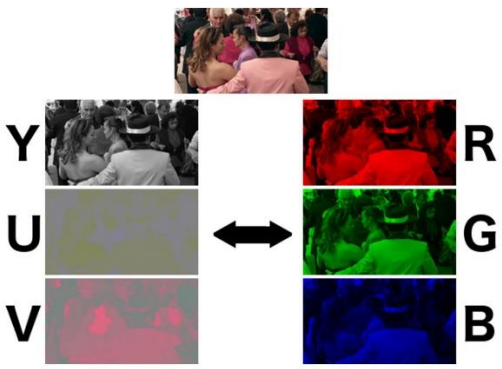


Figure1 .3 : *Espaces colorimétriques RGB et YUV*

La vidéo couleur peut être représentée de diverses manières. (R G B, YIQ, YC_bC_r, YUV et d'autres…), elles sont toujours toutes mathématiquement liées au RVB.

La conversion réelle entre le format RVB standard et le format YUV et vice-versa est représenter sur équation 1.1. [3] Voir figure 1.3

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1{,}402 \\ 1 & -0{,}34414 & -0{,}71414 \\ 1 & 1{,}772 & 0 \end{bmatrix} . \begin{bmatrix} y \\ U-128 \\ V-128 \end{bmatrix} \qquad 1.1$$

L'espace colorimétrique RVB à de grandes lacunes, dont la principale est qu'il n'est pas intuitif, il nous est donc difficile de connaître les attributs cognitifs de la couleur exprimés par une valeur à partir de sa valeur RVB. Ensuite, l'espace colorimétrique RVB est l'un des espaces colorimétriques les plus inégaux, car la différence visuelle entre deux couleurs ne peut pas être exprimée comme la distance entre deux points de couleur. De plus, la corrélation entre RVB est très élevée et l'espace RVB est sensible au bruit dans les zones de faible intensité.

Il existe de nombreux espaces colorimétriques, où les composants de luminance et de chrominance sont séparés, tel qu' YCbCr, HSV et Lab, la formule de conversion entre YC_bC_r et RVB sont les suivants 1.2

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.299 & 0{,}587 & 0{,}114 \\ -0{,}169 & -0{,}331 & 0{,}5 \\ 0{,}5 & -0{,}419 & -0{,}081 \end{bmatrix} . \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad 1.2$$

1.2.3 Sous-échantillonnage de la chrominance [4, 5]

Une méthode pour réduire le volume d'une image numérique est le sous-échantillonnage de chrominance. Il implique de réduire le nombre d'échantillons à traiter. (Figure 1.4)

Notation de sous-échantillonnage de chrominance :

Au début de la vidéo numérique, la notation du sous-échantillonnage était logique ; malheureusement, la technologie a dépassé cette notation. La Figure 1.4 montre la notation du sous-échantillonnage chromatique : le premier chiffre indique la référence d'échantillonnage horizontal de la luminance. Le deuxième chiffre spécifie le sous-

échantillonnage horizontal de CB et CR par rapport à la luminance. Le troisième chiffre spécifiait à l'origine le sous-échantillonnage horizontal de CR. La notation s'est développée sans anticiper le sous-échantillonnage vertical ; un troisième chiffre égal à zéro désigne désormais un sous-échantillonnage vertical 2:1 de CB et CR. [4, 5]

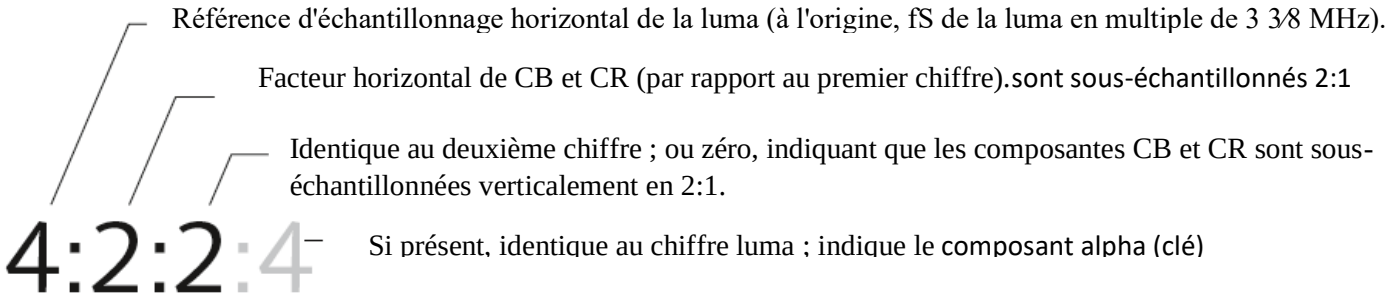


Figure1.4 : *La notation de sous-échantillonnage de chrominance.*

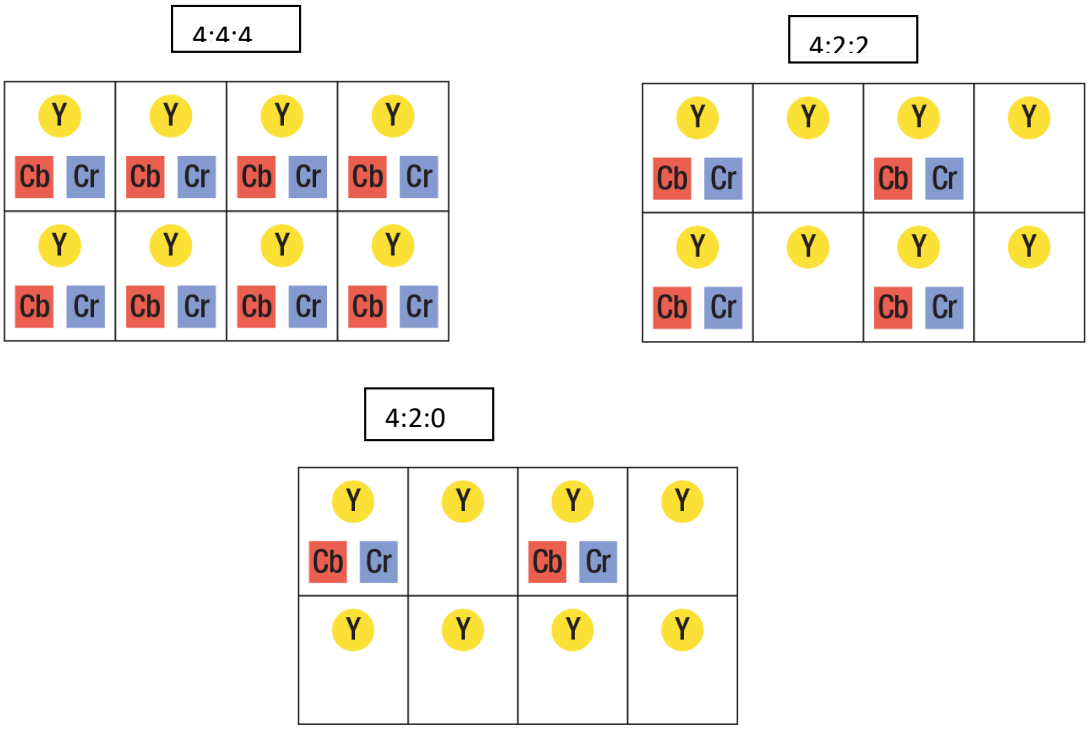


Figure 1.5 : *Quelque Représentation de sous-échantillonnage de chrominance*

- La structure 4:4:4 : aucun sous-échantillonnage.
- La structure 4:2:2 : Les composants des chrominances CB et CR sont chacun sous-échantillonné par un facteur de 2 horizontalement.

- La structure 4:2:0 : est utilisée dans JPEG/JFIF, MPEG-2 et H.264, H265, les chrominances CB et CR sont chacun sous-échantillonnés par un facteur de 2 horizontalement et un facteur de 2 verticalement.
- La structure 4:1:1 : indique un sous-échantillonnage de la chrominance par un facteur de 4 horizontalement

1.2.4 Chronométrage Vidéo

Bien qu'il semble que la vidéo soit un mouvement continu, il s'agit en réalité d'une collection d'images fixes, qui changent suffisamment rapidement pour donner l'illusion d'un mouvement continu, comme le montre la figure 1.6. Cela se produit généralement 50 ou 60 fois par seconde pour la vidéo grand public et 70 à 90 fois par seconde pour les écrans d'ordinateur. Des informations de synchronisation spéciales, appelées synchronisation verticale, sont utilisées pour indiquer le début d'une nouvelle image.

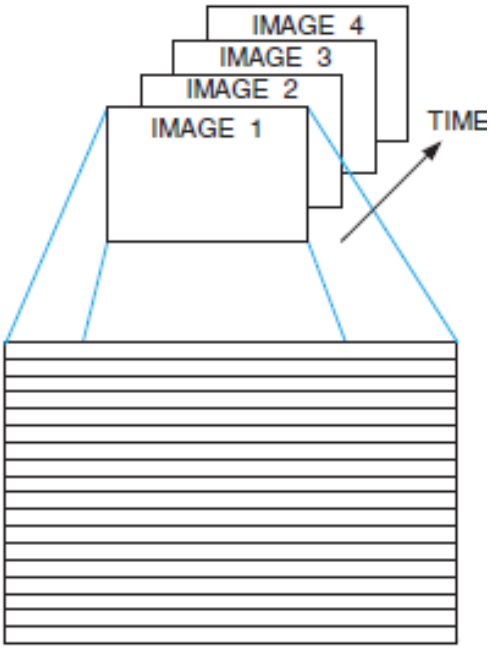


Figure 1.6 : *Séquence vidéo*

Chaque image fixe est également composée de lignes de balayage, c'est-à-dire des lignes de données qui apparaissent séquentiellement les unes après les autres sur l'écran, comme le montre la figure 1.6. Des informations de synchronisation supplémentaires, appelées synchronisation horizontale, sont utilisées pour indiquer quand une nouvelle ligne de balayage commence.

Le signal de synchronisation composite est une combinaison de synchronisation verticale et horizontale. Il y a deux techniques d'affichage vidéo : les affichages progressifs et non entrelacés. Pour les écrans progressifs qui « peignent » une image sur l'écran, comme un CRT (tube à rayons cathodiques), chaque image est affichée en commençant par le coin supérieur gauche de l'écran, en se déplaçant vers le bord droit de l'écran. Le balayage descend ensuite d'une ligne et répète le balayage de gauche à droite. Ce processus est répété jusqu'à ce que tout l'écran soit actualisé, comme illustré à la figure 1.7(a). Au début de la télévision, une technique appelée entrelacement a été créé pour diminuer la quantité d'informations envoyées pour chaque image. En transférant les lignes impaires, suivies des lignes paires (comme le montre la figure 1.7(b), la quantité d'informations envoyées pour chaque image a été réduite de moitié. Aujourd'hui, la plupart des émissions (y compris la TVHD) sont encore transmises en mode entrelacé.

La plupart des écrans CRT sont encore entrelacés, tandis que les écrans LCD, plasma et les écrans d'ordinateur sont progressifs.

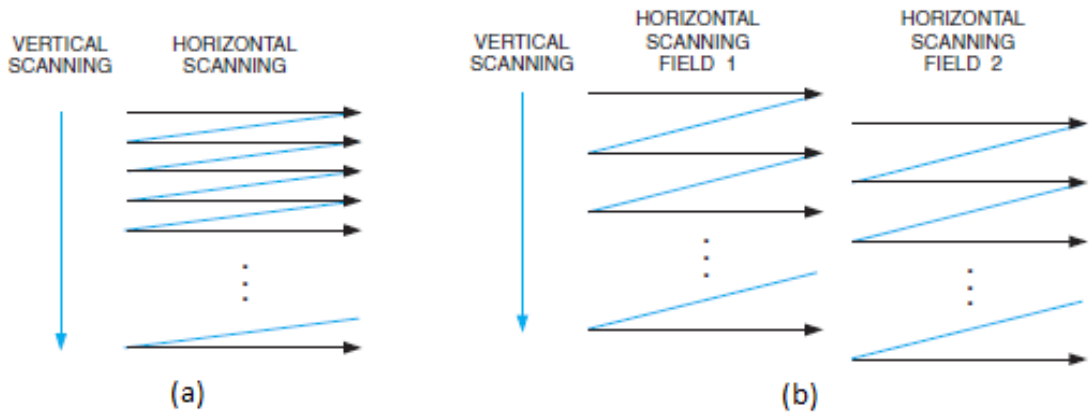


Figure 1.7 : *(a) affichages progressifs, (b) Affichages entrelacés*

1.4.1 Résolution vidéo :

En informatique, le nombre de pixels détermine la résolution. (Il s'agit de la plus petite unité d'un graphique ou d'une image) disposée en largeur et en hauteur sur un écran ou une image. En termes simples, la résolution mesure le nombre de pixels d'une image (Résolution = nombre de pixels en largeur × nombre de pixels en hauteur). SD (environ 720 × 480), HD (1280 × 720) et Full HD (1920 × 1080) ont tous un nombre de pixels différent. Par exemple, une "résolution" de 1152 × 864 signifie que le système possède

un total d'environ un million de pixels (soit un mégapixel, ou 1 Mpx). Cependant, à mesure que la fréquence des détails de l'image augmente, le traitement du signal et les effets optiques entraînent une diminution de la réponse, même dans les limites imposées par l'échantillonnage [4, 6].

En vidéo, la résolution fait référence au nombre maximum de paires de lignes (ou cycles) pouvant être résolues sur l'écran d'affichage. La résolution désigne donc le nombre de paires de lignes (ou cycles) résolues sur la surface de l'écran, exprimée en cycles par hauteur d'image (C/PH) ou en cycles par largeur d'image (C/PW). Un cycle équivaut à une paire de lignes de film. Dans un système numérique, il faut au moins deux échantillons — pixels, lignes de balayage ou lignes TV — pour représenter une paire de lignes. Cependant, la résolution effective peut être sensiblement inférieure au nombre de paires de pixels en raison des effets de filtrage optique, électro-optique et électrique. La résolution limite est définie comme la fréquence à laquelle les détails sont enregistrés avec seulement 10 % de la réponse en basse fréquence du système.

La figure 1.8 illustre un graphique de test de résolution. Dans l'orientation présentée, il balaie les fréquences horizontales et peut être utilisé pour estimer la résolution horizontale. Lorsqu'il est tourné à 90°, il permet de balayer les fréquences verticales, et ainsi d'estimer la résolution verticale.

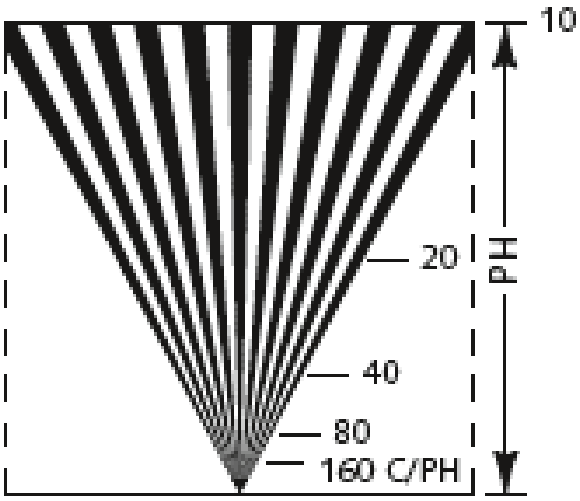


Figure 1.8 : *Représentation d'un test de résolution*

1.3 Compression

La vidéo génère de grandes quantités de bande passante et nécessite beaucoup d'espace de stockage. Par exemple, les vidéos TV ont des débits de données en Mb/s (pour la TV HD) ou en Gb/s. Les méthodes de compression efficaces sont essentielles pour faciliter la gestion de tels débits de données.

La compression consiste à réduire la redondance des données en convertissant ces dernières de manière réversible (par codage), afin de les représenter avec moins de bits. Cela permet un stockage et une transmission plus efficaces des données. Le processus inverse est appelé décompression (ou décodage). Les logiciels et le matériel capables d'encoder et de décoder sont appelés décodeurs. L'ensemble des deux opérations (encodage et décodage) forme un codec. Des algorithmes génériques de compression sans perte, qui assurent la reconstruction parfaite des données initiales, peuvent être utilisés pour les images et la vidéo. Cependant, ces algorithmes n'atteignent qu'une réduction de données d'environ 2:1 en moyenne, ce qui n'est pas suffisant pour une compression vidéo efficace. Lors de la compression vidéo, deux types spécifiques de redondance peuvent être exploités :

- **Redondance spatio-temporelle** :
  Généralement, les valeurs de pixels sont corrélées avec leurs voisins, à la fois dans la même image et entre les images successives (voir figure 1.9).
- **Redondance psycho-visuelle** :
  Le système visuel humain n'est pas également sensible à tous les schémas. Par conséquent, l'algorithme de compression peut rejeter des informations qui ne sont pas perçues par l'observateur. C'est ce qu'on appelle la compression avec perte.



Figure 1.9 : *Redondance spatiale et temporelle*

En vidéo analogique, ces deux types de redondances sont exploités grâce à des techniques de codage couleur et d'entrelacement basées sur la vision. La vidéo numérique offre des méthodes de compression supplémentaires, qui sont décrites ci-après.

a vidéo numérique bénéficie de méthodes de compression spéciales. Elle peut être grossièrement classée en deux catégories : les méthodes basées sur des modèles, telles que la compression fractale, et les méthodes basées sur la forme d'onde, telles que la compression DCT ou par ondelettes. Les codecs et standards vidéo actuels appartiennent à cette dernière catégorie et comprennent les étapes suivantes [7] (voir figure 1.10) :

- **Transformation** : Pour faciliter l'exploitation des redondances psychovisuelles, les images sont transformées dans un domaine où différentes gammes de fréquences, avec des sensibilités variables du système visuel humain, peuvent être séparées. Cela peut être réalisé à l'aide de la transformée en cosinus discrète (DCT), par exemple. Cette étape est réversible, c'est-à-dire qu'aucune information n'est perdue.
- **Quantification** : Après la transformation, la précision numérique des coefficients de transformée est réduite afin de diminuer la quantité de bits présents dans le flux. Le degré de quantification appliqué à chaque coefficient est généralement déterminé par la visibilité de la distorsion résultante pour un observateur

humain ; les coefficients haute fréquence peuvent être quantifiés plus grossièrement que les coefficients basse fréquence, par exemple.

- **Codage** : Une fois que les données ont été quantifiées en un ensemble fini de valeurs, elles peuvent être codées sans perte en exploitant la redondance entre les coefficients quantifiés dans le flux binaire. Le codage entropique, qui repose sur le fait que certains symboles apparaissent beaucoup plus fréquemment que d'autres, est souvent utilisé pour ce processus. Deux des schémas de codage entropique les plus populaires sont le codage de Huffman et le codage arithmétique[7, 8].

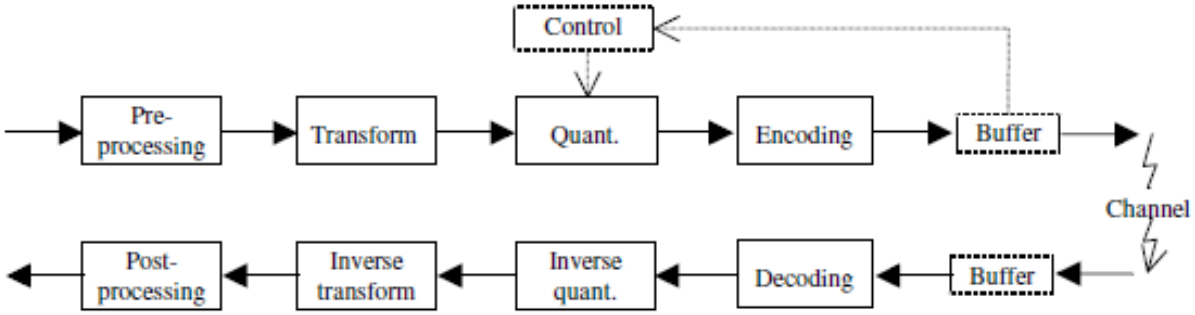


Figure 1.10: Schéma bloc d'un processus de codage et de décodage vidéo de base

Une composante importante de la compression vidéo numérique consiste à exploiter la similarité entre les images successives d'une séquence au lieu de coder chaque image séparément. Alors que cette redondance temporelle pourrait être prise en charge par une transformation spatio-temporelle, Pour les raisons d'efficacité de mise en œuvre, une approche hybride de domaine spatial et de domaine de transformation est souvent utilisée à la place. Une méthode simple de compression temporelle est la différenciation de trame, où seules les différences pixel par pixel entre les trames successives sont codées. Une compression plus élevée peut être obtenue en utilisant l'estimation de mouvement, une technique pour décrire une image basée sur le contenu des images proches à l'aide de vecteurs de mouvement. En compensant les mouvements des objets de cette manière, les différences entre les images peuvent être encore réduites[8].

1.3.1 Les paramètres des optimisations de codage vidéo

Il y a trois exigences concernant le codage vidéo : un haut niveau de compression, une faible complexité et une haute qualité visuelle..

1.3.1.1 Taux de compression

La compression en tant que rapport de la taille (généralement en bits ou octets) de l'image ou de la vidéo d'origine (d'entrée) B_i à celle de l'image ou de la vidéo (de sortie) utilisée pour le stockage ou la transmission B_o :

$$CR = \frac{B_i}{B_O} \tag{1.3}$$

L'efficacité de la compression est l'exigence la plus importante d'une norme de codage. L'objectif de chaque norme de codage vidéo est de doubler le taux de compression par rapport à la norme précédente. Comme des outils de codage de plus en plus avancés ont été développés dans les nouvelles normes, le taux de compression approche à sa limite et presque saturé. Explorer les redondances vidéo et améliorer encore le taux de compression devient vital et extrêmement difficile. [1, 6]

1.3.1.2 Complexité de codage :

Les complexités de codage des algorithmes de codage et de décodage sont liées au coût du matériel, à l'accès à la mémoire, à la capacité de calcul, etc., qui sont directement proportionnels au coût de production, d'utilisation et de maintenance. Comme des algorithmes de codage plus avancés et plus compliqués ont été adoptés dans les normes en cours pour améliorer l'efficacité du codage, cela entraîne des dizaines ou des centaines de fois la charge de complexité par rapport à son prédécesseur. De plus, à mesure que les résolutions vue-spatiale-temporelle, la profondeur de bits et le nombre de vues augmentent, le coût de calcul du codec vidéo est augmenté de manière multiplicative, probablement des millions de fois. Pour permettre une diffusion en temps réel de contenus tels que l'UHD, la 3D et la réalité virtuelle (VR), il est crucial de réduire les coûts de calcul à l'aide de techniques d'optimisation, afin d'assurer une expérience fluide et de haute qualité.

- **Qualité visuelle :**

La qualité visuelle ou QoE puisque les vidéos doivent être compressées autant que possible tout en conservant la qualité visuelle. Actuellement, la qualité de l'image/vidéo est principalement mesurée par le rapport crête-signal sur bruit (PSNR) ou l'erreur quadratique moyenne (MSE) en raison de sa simplicité. Cependant, le PSNR et le MSE ne peuvent pas vraiment refléter la qualité perçue du HVS, qui est un système non linéaire compliqué. Bien que de nombreux facteurs perceptuels importants aient été révélés dans des perspectives psychologiques et physiologiques, les connaissances sur le HVS sont encore très limitées. Il est difficile de développer une métrique de qualité efficace qui puisse être largement applicable au codage vidéo pour explorer les redondances perceptuelles.

1.4 Les normes de codage vidéo

Dans les années 1990, le développement des normes de codage vidéo a été porté par deux espaces d'application parallèles : la communication vidéo en temps réel et la distribution ou diffusion de contenu vidéo. Les spécifications correspondantes ont été publiées par deux principaux organismes de normalisation, l'Union internationale des télécommunications (UIT) et l'Organisation internationale de normalisation/Commission électrotechnique internationale (ISO/CEI). La Figure 1.11 synthétise l'évolution des différentes recommandations ou normes de compression vidéo. Les cinq normes de pointe (H.261, MPEG-4, H.264/AVC, HEVC et VVC dans des rectangles rouges) ,Depuis trois décennies, quatre générations ont été publiés. [1].

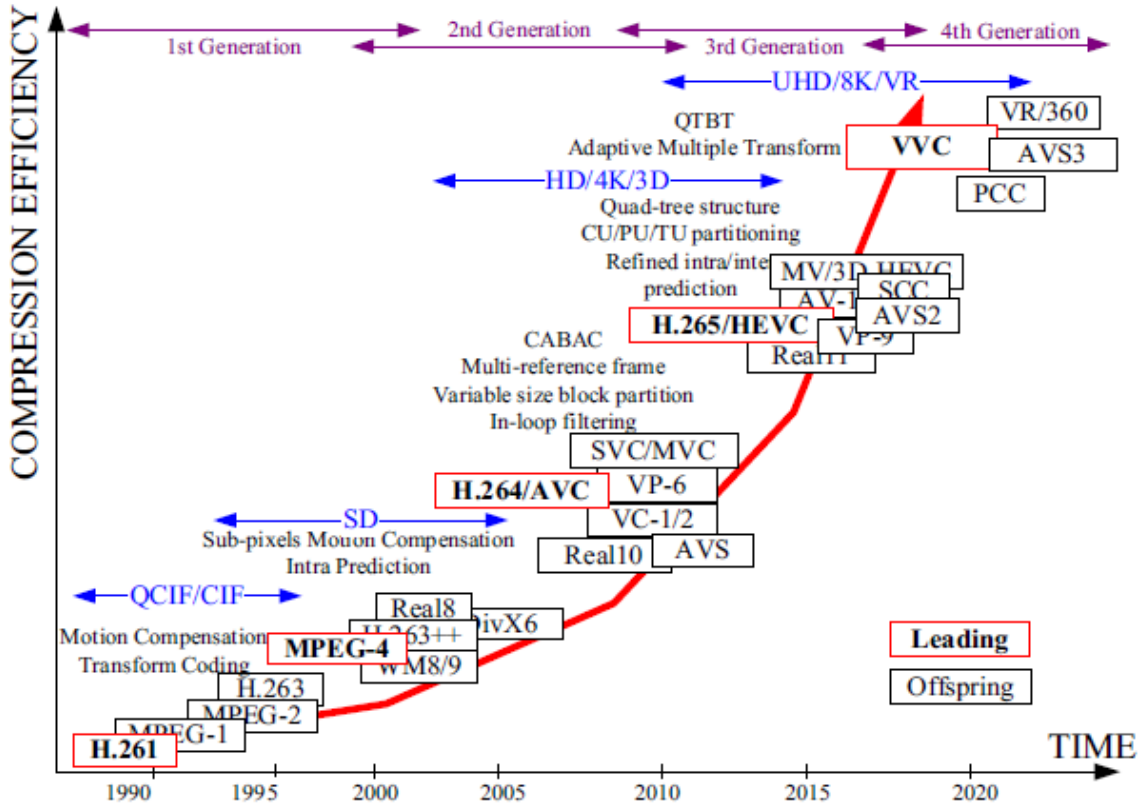


Figure 1.11 : *Les normes de compression vidéo.*

Chaque génération de norme de codage vidéo, tous les 7 à 10 ans, depuis l'introduction de H.120 en 1984 jusqu'aux codecs H.265/HEVC et H.266/VVC les plus récents, a constamment divisé par deux le débit binaire requis pour une qualité vidéo équivalente. Cette partie présente les fonctionnalités qui ont permis ces progrès.

1.4.1 Le processus d'adoption de la normalisation

La normalisation suit un cheminement bien défini, avec 4 étapes suivantes :

- **Définition des exigences** : Ici, le champ d'application de la norme est finalisé et les exigences et les objectifs du processus de normalisation sont définis.
- **Divergence** : un appel à propositions formel est alors normalement lancé, permettant aux experts de l'industrie et du milieu universitaire de présenter et de comparer leurs méthodes et leurs résultats, généralement par rapport à une norme existante en utilisant un ensemble prédéfini de données et de conditions de test.
- **Convergence** : Les meilleurs algorithmes apportés lors de la phase de divergence sont sélectionnés. Ceci est normalement réalisé grâce à l'utilisation d'un modèle de test évolutif tel que le modèle de test HEVC (HM) ou le modèle de test VVC

(VTM). En utilisant la version actuelle, ce modèle évolue à travers plusieurs itérations comme référence pour des propositions ultérieures dans différents domaines techniques clés (KTA, pour HM) ou pour diverses expériences de base (CE, pour VTM). Ceux-ci, en cas de succès, sont incorporés dans une nouvelle version améliorée du modèle. Ce processus se poursuit jusqu'à ce que les spécifications de performances requises soient satisfaites.

- **Vérification** : lors de la vérification, la norme résultante et son flux binaire sont validés pour leur conformité. Des processus de test de conformité sont ensuite définis pour les tests de conformité des produits.

1.4.2 H.120

Le groupe d'étude SG.XV du CCITT a commencé ses travaux sur H.120 en 1980 et a produit la première norme internationale de codage vidéo numérique en 1984, suivie d'une deuxième version en 1988[9]. H.120 s'adressait aux applications de visioconférence à 2,048 Mb/s et 1,544 Mb/s pour les systèmes TV 625/50 et 525/60, respectivement. Cette norme n'a jamais été un succès commercial, en partie parce qu'elle était basée sur des stratégies de codage différentes pour différentes régions internationales, mais principalement parce que sa qualité d'image (en particulier la qualité temporelle) était insuffisante. H.120 a mis en œuvre une stratégie de réapprovisionnement conditionnel, dans laquelle chaque trame était divisée en régions modifiées et inchangées. Les régions modifiées ont été codées avec une modulation par impulsions codées différentielles intrachamp (DPCM) dans les parties 1 et 2 de la norme, bien que la partie 3 (pour une utilisation aux États-Unis) a utilisé la prédiction de fond et la prédiction inter champ compensée en mouvement.

1.4.3 H.261

H.261[10] a suivi H.120 en 1989 et a été le premier codec vidéo à avoir été largement adopté. Il était basé sur un modèle p×64 kbps (p= 1, . . ., 30), destiné aux applications de conférence ISDN. H.261 a été le premier algorithme de compression hybride basé sur des blocs à utiliser une combinaison de transformation (la transformée en cosinus discrète DCT), de DPCM temporel et de compensation de mouvement. Cette

architecture a résisté à l'épreuve du temps car toutes les principales normes de codage vidéo reposent depuis sur elle.

- **Types d'images et caractéristiques principales Macrobloc, GOB et format de trame**

H. 261 a introduit une hiérarchie de base dans le processus de codage d'image qui, avec quelques modifications, est toujours utilisée aujourd'hui. Les images H.261, au format CIF ou QCIF, sont représentées comme une séquence de groupes de blocs (GOB), comprenant chacun un certain nombre de macroblocs (MB). Un macrobloc, , comprend quatre blocs de luminance spatiaux transformés en DCT 8x8 et deux blocs de chrominance transformés en DCT 16x16 spatiaux (sous-échantillonnés). Un diagramme montrant la hiérarchie des images est donné à la Fig. 1.12 pour le cas d'une trame CIF. Comme on peut le voir, il y a 33 Mo dans un GOB, ce qui donne une dimension GOB de 176 × 48 pixels, et il y a 12 GOB dans une trame CIF, ce qui donne une dimension de trame de 352 × 288 pixels. H.261 ne prend en charge que les trames I et P – les trames B ne sont pas prises en charge. L'estimation et la compensation de mouvement font partie intégrante du processus de codage avec une taille de bloc fixe de 16×16 et une fenêtre de recherche de (−15 · · · +15) pixels.

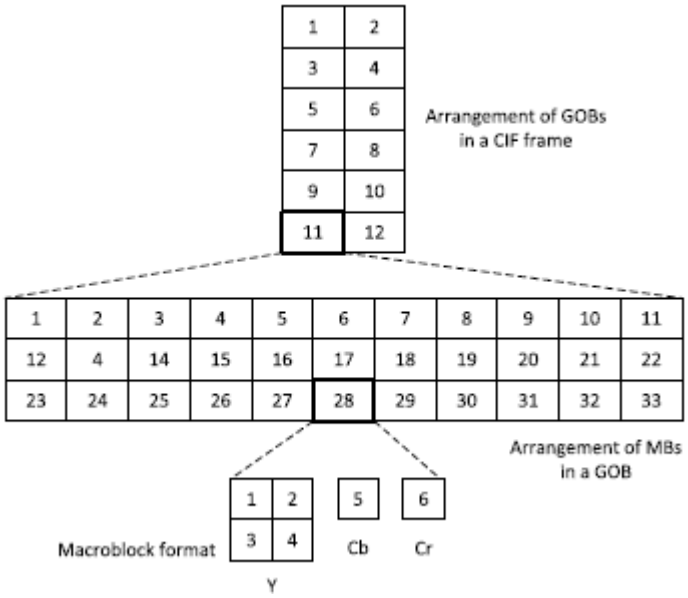


Figure 1.12 : *H.261 format de trame CIF, macro bloc, GOB.*

1.4.4 MPEG-2/DVB

En 1988, le Moving Picture Experts Group (MPEG) a été fondé, délivrant un algorithme de codage vidéo destiné aux supports de stockage numériques à 1,5 Mbps en 1993. Il a été suivi en 1995 par MPEG-2[11], spécifiquement destiné à la diffusion vidéo numérique émergente. MPEG-2 a joué un rôle déterminant, grâce à son inclusion dans tous les décodeurs pendant plus d'une décennie, en soutenant véritablement la révolution de la diffusion numérique.

- **Types d'images et caractéristiques principales**

MPEG-2 (ISO/IEC-13818:2000), également connu sous le nom de H.262, est une norme de codage audiovisuel générique prenant en charge une gamme d'applications à des débits binaires d'environ 2 à 30 Mbps. Il utilise une architecture DCT hybride basée sur des blocs à compensation de mouvement (similaire à H.261), utilisant des blocs 8 × 8 et des macroblocs 16 × 16 avec une estimation de mouvement basée sur des blocs de translation avec une précision d'un demi-pixel. En raison de sa concentration sur la diffusion de télévision numérique, MPEG-2 prend nécessairement en charge les formats d'image progressifs et entrelacés. La norme de codage a été définie principalement pour les environnements sans erreur et le codage de canal a été ajouté en fonction du scénario d'application (par exemple, diffusion terrestre, par câble ou par satellite). MPEG-2 a été défini pour prendre en charge une large gamme d'applications allant de la capture et de l'édition du traitement en studio à la distribution et à la diffusion. Il est intimement lié aux normes de diffusion de contenu de diffusion DVB-T, DVB-S et DVB-C. MPEG-2 prend en charge trois types d'image : intra (I), codé sans référence à d'autres trames (moins efficace) ; prédit (P), codé sur la base de la prédiction d'une trame I ou P précédente ; et prédit de manière bidirectionnelle (B), prédit à partir d'images P et/ou I mais non utilisé comme base pour d'autres prédictions (le plus efficace). Il utilise une structure de groupe d'images (GOP) comme illustré à la Fig. 1.12 et a été le premier standard à introduire des trames B prédites de manière bidirectionnelle.

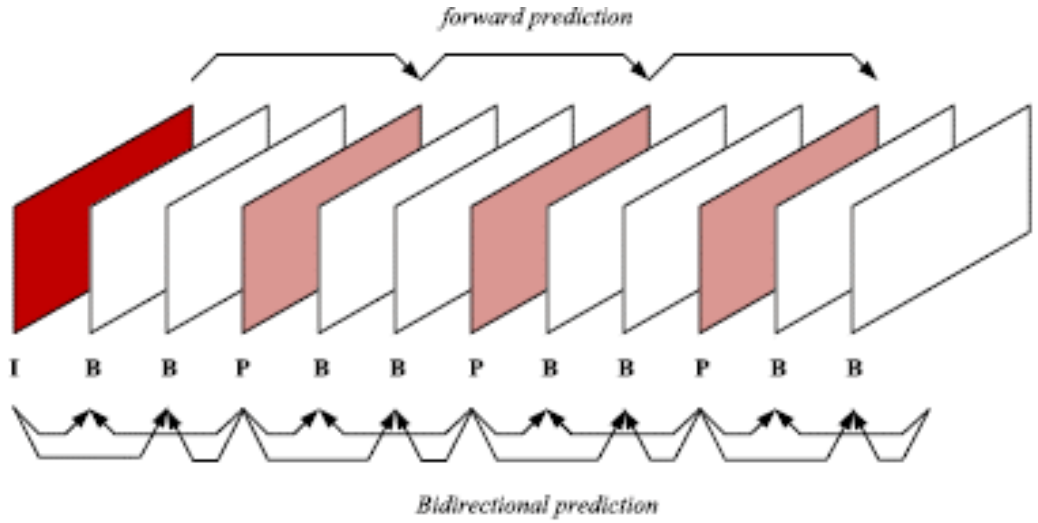


Figure 1.13 : *Structure du GOP MPEG-2.*

1.4.5 H.263

H.263 [11] a été défini par ITU-T SG15, à partir de 1993 avec l'objectif de coder à des débits binaires inférieurs à 64 kbps. L'application initiale était axée sur le RTPC et les premières applications radio mobiles à des débits binaires compris entre 10 et 24 kb/s. Bien que la vidéo mobile soit plus lente à décoller que prévu, H.263 a néanmoins eu un impact significatif sur les applications de conférence et de surveillance, ainsi que sur les premiers flux Internet. En particulier, H.263 a été utilisé pour encoder le contenu Flash Video pour des sites tels que YouTube et Myspace. Real Vidéo était initialement basé sur celui-ci, et il a été spécifié dans plusieurs services vidéo ETSI 3GPP. Le codec a d'abord été conçu pour être intégré dans le cadre H.324 pour les applications à commutation de circuits, mais a été largement utilisé dans H.323 (visioconférence basée sur RTP) et d'autres wrappers de streaming IP.

- **Types d'images et caractéristiques principales** :

Macrobloc, GOB et format de trame H.263 prend en charge les formats d'image suivants : sub-QCIF (88 × 72), QCIF (176 × 144), CIF (352 × 288), 4CIF (704 × 576) et 16CIF (1408 × 1152), le tout au format YCbCr 4:2:0. Comme avec H.261, chaque image est divisée en un certain nombre de groupes de blocs (GOB) définis comme un nombre entier, k, de rangées de macroblocs (MB). Pour SQCIF, QCIF et CIF, k=1 ; pour

4CIF, k=2 ; et pour 16CIF, k=4. La structure d'une image QCIF H.263 est illustrée à la Figure. 1.13.

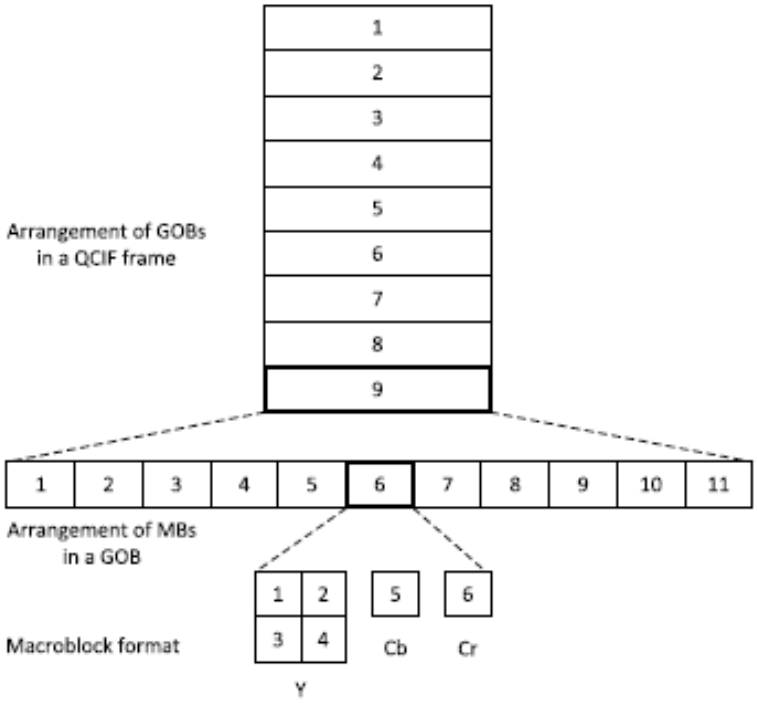


Figure 1.14 : *Format d'image H.263 (QCIF).*

1.4.6 MPEG-4

MPEG-4 était un projet très ambitieux qui cherchait à introduire de nouvelles approches utilisant des méthodes basées sur les objets ainsi que sur les formes d'onde. La norme MPEG-4 a été ratifiée en 2000, sous le nom d'ISO/IEC 14496, offrant potentiellement des améliorations d'encodage significatives au-delà de MPEG-2. Il s'est toutefois avéré, en général, trop complexe. Il est intéressant de comparer les documents de normes pour H.261 et MPEG-4. , et cela a gagné du terrain en formant la base de la technologie émergente des appareils photo numériques de l'époque. Le premier

MPEG-4 partie 2 offrait peu d'avantages pratiques sur MPEG-2 ou H.263 et a été rapidement dépassé par MPEG-4 part 10/AVC, maintenant connu sous le nom de H.264/AVC. [12]

1.4.7 H.264/AVC

ITU-T/SG16/Q6 (Video Coding Experts Group) a commencé à travailler en 1998 sur un projet appelé H.26L. L'équipe vidéo conjointe (JVT) entre VCEG et MPEG a été formée en 2001 pour établir un projet de norme commun connu sous le nom de H.264/MPEG-4 AVC. Le projet final pour les formalités l'approbation a été produite en mars 2003[13] , avec l'extension de codage vidéo évolutif (SVC) finalisée en 2007. MPEG-4 (partie 10), ou H.264/AVC, est de loin la norme de codage vidéo la plus répandue à ce jour. De la même manière que MPEG-2 a soutenu la révolution de la diffusion numérique, La vidéo sur Internet a été grandement facilitée par H.264/AVC, des services mobiles, des services OTT, de l'IPTV et de la TVHD. H.264/AVC est un format obligatoire pour les lecteurs Blu-ray et est utilisé par la plupart des sites de streaming Internet, y compris Vimeo, YouTube et iTunes. Il est utilisé dans Adobe Flash Player et Microsoft Silverlight et a également été adopté pour la diffusion HDTV par câble, satellite et terrestre. L'UIT-T, en partenariat avec l'ISO/CEI, a livré le H.264/AVC en 2004.

L'objectif du projet H.264/AVC était de fournir :

- Efficacité de codage améliorée : une réduction ciblée du débit binaire moyen de 50 % pour une fidélité fixe, par rapport à toute autre norme vidéo.
- Robustesse aux erreurs : outils pour gérer la perte de paquets et la gestion de la congestion.
- Convivialité réseau : les principaux objectifs étaient les réseaux mobiles et Internet – séparation des couches de codage par l'abstraction du réseau.
- Adaptation aux contraintes de délai : Pour fournir des modes à faible délai.
- Spécification de syntaxe simple : Éviter une quantité excessive de fonctionnalités optionnelles ou de configurations de profils.

H.264/AVC prend en charge le traitement des formats progressifs et entrelacés de manière cohérente comme une seule unité. Le fait que la trame soit progressive ou entrelacée est indiqué dans le PPS et un codage trame-trame adaptatif peut être utilisé (soit pour coder une trame comme une seule unité soit comme deux trames) mais cela n'a aucun impact sur la façon dont l'image est décodée via la mémoire tampon du

décodeur. H.264/AVC prend en charge les trames I, les trames P et les trames B. Les trames B sont appelées trames B généralisées dans H.264, ce qui signifie qu'ils peuvent être prédits à partir d'une ou deux trames de référence. H.264/AVC prend en charge une très large gamme de formats d'image de QCIF à 4K UHDTV. Le macrobloc dans H.264 est basé sur un agencement d'échantillons 16 × 16, comprenant des données résiduelles de luminance et de chrominance. La différence avec H.264/AVC est que cette structure peut être subdivisée de différentes manières selon les modes de prédiction disponibles.

1.4.8 H.265/HEVC

HEVC, officiellement connu sous le nom de ISO/IEC MPEG-H Part 2 et ITU-T H.265, a été développé en collaboration entre ISO/IEC MPEG et ITU-T VCEG. La norme a été approuvée en avril 2013[14] et offre la possibilité d'améliorer jusqu'à 50 % l'efficacité de la compression.

Sur AVC et cible les solutions pour les services de vidéo mobile, de diffusion et de streaming avec des exigences de qualité accrues et des expériences plus immersives. La réduction du débit binaire pour des résolutions vidéo plus grandes, la prise en charge du traitement parallèle et la facilité d'intégration avec les mécanismes de transport appropriés sont les principaux avantages d'HEVC. (Ces points sont détaillés dans le chapitre 2.)

1.4.9 H.266/VVC

Offrir une meilleure prise en charge pour les formats vidéo immersifs et d'améliorer les capacités de compression au-delà de celles offertes par les normes actuelles, une nouvelle norme de codage vidéo, H.266/Versatile Video Coding (VVC), a été développée par une équipe conjointe de l'UIT-T et des experts ISO/CEI (l'équipe conjointe d'experts vidéo JVET) et a finalement été approuvée en juillet 2020. L'appel formel à propositions VVC a été lancé en 2017 dans le but d'améliorer considérablement l'efficacité du codage par rapport au HEVC. Vingt et un groupe de promoteurs de 32 organisations industrielles et universitaires ont soumis 46 solutions : 22 pour le contenu à plage dynamique standard (SDR) et 12 pour le contenu à plage dynamique élevée (HDR) et le contenu vidéo à 360 degrés. Environ 40 nouvelles fonctionnalités ont été

adoptées. La version la plus récente du modèle de test VVC (VTM) peut atteindre un gain de codage supérieur à 30 % par rapport à l'encodeur de référence HEVC (HM) sur le contenu SDR, avec des gains de performances encore plus importants pour le contenu HDR et à 360 degrés. [15, 16]

L'un des principaux objectifs de VVC était d'améliorer l'efficacité de la compression pour le contenu UHD, HDR et Wide Color Gamut (WCG) et d'encoder des formats vidéo immersifs tels que 360 degrés. Bien que VVC utilise toujours une architecture de codage hybride traditionnelle basée sur des blocs, il inclut de nombreuses nouvelles fonctionnalités liées au partitionnement d'image, à la prédiction intra et inter, à la transformation, à la quantification, au codage entropique et au filtrage en boucle. Le tableau 1.1 résume les principaux nouveaux outils de codage adoptés dans VVC.

Tableau 1.1 : *Principalement des outils de codage récemment développés utilisés dans H.266/VVC.*

Fonction codec	Approche
Cloisons d'images	Une structure quadtree avec des partitions multitypes imbriquées utilisant la segmentation binaire et ternaire La taille maximale de la CTU pour les blocs luma est de 128 × 128. La taille de transformation maximale pour les blocs luma est de 64 × 64.
Intra-prédiction	65 intra-modes directionnels, plus ajustement de surface et prédiction DC Intra-lissage dépendant du mode Intra-prédiction multiligne de référence
Interprédiction	Interprédiction basée sur un modèle de mouvement affine Précision du vecteur de mouvement de1/ 16 pixels Intra- et inter-prédictions combinées Résolution de vecteur de mouvement adaptative
Transformer et quantifier	Plusieurs sélections de transformations primaires Quantification dépendante avec QP max augmenté à 63
Codage entropique	CABAC avec un modèle adaptatif de mise à jour à double probabilité
Filtre en boucle	Filtre de boucle adaptatif Cartographie Luma avec mise à l'échelle de la chrominance
Codage vidéo à 360 degrés	Compensation de mouvement enveloppant horizontal

a. Syntaxe de haut niveau :

- **Unités NAL :** Par rapport à HEVC, VVC réduit le nombre de types NALU (NUT) de 63 à 32, dont 13 NUT pour le VLC, 7 pour les jeux de paramètres et les en-têtes d'image, et 12 pour les autres NUT non-VLC.
  Jeux de paramètres Trois nouveaux ensembles de paramètres ont été utilisés en plus de ceux existants dans HEVC, y compris les informations de capacité de décodage (DCI), un ensemble de paramètres d'adaptation (APS) et un en-tête d'image (PH). DCI comprend des informations de profil, de niveau et de sous-profil, tandis qu'APS contient des paramètres adaptatifs tels que les coefficients de filtre de boucle adaptative (ALF). PH contient des PPS et des SPS qui s'appliquent à toutes les tranches d'une image.

b. Partitionnement d'image :

Afin de mieux gérer les mouvements complexes et les limites d'objets, VVC prend en charge une flexibilité accrue dans la manière dont les images peuvent être partitionnées. Il utilise le même concept CTU que dans HEVC, mais avec sa taille de bloc luma maximale étendue à 128 × 128. Chaque CTU peut être divisée en un arbre multitype imbriqué via une segmentation binaire et ternaire. VVC a également simplifié les types d'unités de partition multiples (CU, PU et TU) définis dans HEVC et ne les active que lorsqu'une CU est supérieur à la taille de transformation maximale. En plus de la segmentation quadtree dans HEVC avec des formes CU carrées, VVC prend également en charge diverses formes CU rectangulaires via le fractionnement binaire vertical (SPLIT_BT_VER), le fractionnement binaire horizontal (SPLIT_BT_HOR), le fractionnement ternaire vertical (SPLIT_TT_VER) et le fractionnement ternaire horizontal (SPLIT_TT_HOR). Un exemple de partitions CTU est illustré à la Figure 1.14. La taille de transformation maximale prise en charge dans VVC est passée à 64 × 64 pour les composants luma et 32 × 32 pour les canaux chroma.

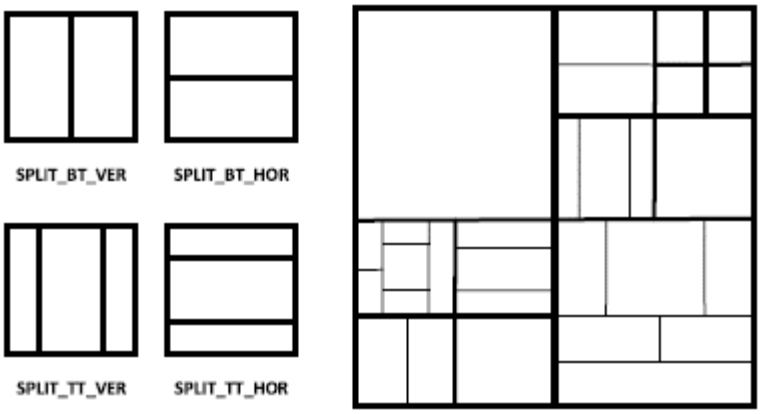


Figure 1.15: *Exemple de partitions CTU dans VVC.*

1.5 Conclusion

Ce chapitre aborde les représentations vidéo et les différentes étapes de compression, ainsi que les caractéristiques clés de toutes les normes majeures de compression vidéo, de H.120 à H.261, H.263 et MPEG-2 H.264/AVC, jusqu'à H.265/HEVC et le nouveau codec H.266/VVC. Bien que ces normes reposent largement sur une architecture de transformation hybride avec compensation de mouvement, les innovations continues de ces 30 dernières années ont permis de diviser en moyenne par deux le débit binaire tous les 7 à 10 ans, tout en maintenant une qualité vidéo équivalente. Ces progrès sont impressionnants, cependant, la demande en contenu vidéo de haute qualité à grande échelle ne cesse d'augmenter, et de nouvelles approches seront nécessaires pour répondre aux exigences futures.

2 .

CHAPITRE 2: LE CODAGE VIDEO A HAUTE EFFICACITE (HEVC)

2.1 Introduction

Le codage vidéo HD (HEVC) est en cours de préparation en tant que norme de codage vidéo la plus récente par le groupe d'experts en codage vidéo de l'UIT-T et le groupe d'experts en images animées ISO/IEC. L'objectif principal de l'effort de normalisation HEVC est d'offrir des performances de compression considérablement améliorées par rapport aux normes existantes, avec une réduction du débit binaire d'environ 50 % pour une qualité vidéo perceptuellement équivalente. Ce chapitre présente l'analyse du HEVC ainsi que des évaluations de ses performances. Nous examinerons de plus près le codage vidéo et la structure syntaxique du HEVC. Le lecteur verra qu'une grande partie de la technologie HEVC a été extraite de H.264/AVC, avec certaines fonctionnalités de H.264 omises et d'autres fonctionnalités nouvelles ou modifiées ajoutées, le cas échéant, pour améliorer les performances.

2.2 Caractéristiques principales

HEVC réduit les exigences de débit binaire environ de moitié par rapport à son prédécesseur pour la même qualité d'image. Pour ce faire, il utilise l'approche hybride éprouvée qui nous a bien servi depuis H.261, mais avec l'ajout de plusieurs fonctionnalités nouvelles et importantes.

Ajoutent à la complexité au codec, beaucoup offrent en fait une simplicité par rapport au H.264/AVC et facilitent le traitement parallèle, compatible avec les développements matériels récents. HEVC, comme les autres normes , peut faire des compromis entre la complexité de calcul, le taux de compression et la robustesse aux erreurs, et le temps de retard de traitement, selon les exigences de l'application, et cette plage de points de fonctionnement est définie par son profil et sa structure de niveau. L'un des principaux objectifs de HEVC est la prochaine génération d'écrans et de systèmes d'acquisition HDTV qui présentent des images à balayage progressif, des fréquences d'images plus élevées et des résolutions allant jusqu'à UHDTV. Une grande partie de la technologie HEVC a été extraite de H.264/AVC, avec certaines fonctionnalités H.264 omises et d'autres fonctionnalités nouvelles et modifiées ajoutées le cas échéant pour améliorer les performances. Les principales nouvelles fonctionnalités de HEVC concernent sa

structure de blocs, où des super macroblocs, appelés unités d'arbre de codage (CTU), sont utilisés, où la taille d'une CTU peut aller jusqu'à 64 × 64 échantillons ; la manière dont ces CTU sont partitionnées en blocs de codage, de transformation et de prédiction (PB) de taille variable à l'aide d'une structure quadtree ; intra-prédiction considérablement améliorée, prenant en charge 35 modes ; et l'utilisation du décalage adaptatif à l'échantillon (SAO) en conjonction avec le DBF dans la boucle d'inter-prédiction pour réduire les artefacts de bande et de sonnerie. Dans le modèle de test HEVC (HM)[17], trois modes de codage par défaut ont été définis pour différents scénarios d'évaluation: Tous les accès intérieurs, à faible délai et aléatoires [18].

a. **Tout mode intra (All intra mode)**: Toutes les trames sont codées sous forme d'images intra (I). Cette configuration reflétée des applications telles que l'édition où l'accès par image à la séquence doit être actif
b. **Mode à faible retard (Low delay mode**) : toutes les trames sont codées sous forme d'images I ou de trames interprédites. L'ordre des trames de codage est identique à l'ordre temporel. Cette configuration est évaluée à l'aide d'images P avec uni-prédiction uniquement ou d'images B avec uni- et bi-prédiction
c. **Mode d'accès aléatoire (Random access mode) :** Une structure hiérarchique est utilisée avec 15 images B consécutives dans un GOP (cela peut être configuré). De grands décalages QP sont appliqués à différents niveaux de couche. Les applications de streaming et de disséminations sont présentes dans cette configuration. Ces trois paramètres de configuration sont évalués dans une configuration correspondant au profil principal de HEVC et dans une configuration pour une efficacité de codage maximale avec une plus grande profondeur de traitement des bits internes. Pour l'évaluation des tests, des séquences vidéo d'une durée de 10 s sont couramment utilisées. Cette durée est considérée comme suffisante pour une évaluation qualifiée de la qualité visuelle de la séquence reconstruite, tout en maintenant à un niveau acceptable l'effort de production du train binaire codé pour l'évaluation.

2.3 Abstraction réseau et syntaxe de haut niveau

La syntaxe de haut niveau fournit l'encapsulation des données vidéo codées pour un traitement ultérieur. Il structure l'information pour le transport et la rend accessible et également consultable. Il fournit en outre des moyens de coder tous les paramètres de haut niveau spécifiés et des informations secondaires supplémentaires. Afin de faire la distinction entre les informations encodées et la forme sous laquelle ces informations sont représentées, une couche de codage vidéo (VCL) et une couche d'abstraction de réseau (NAL) sont définies. Toutes les données VCL et non-VCL sont encapsulées dans des unités NAL pour la transmission. Le schéma de syntaxe de haut niveau est conceptuellement hérité de H.264 | AVC avec quelques révisions et extensions pour correspondre aux exigences modifiées et permettre une fonctionnalité améliorée.

2.3.1 Format de flux d'octets

Comme H.264 | AVC, la spécification HEVC fournit un format de flux d'octets autonome pour la représentation du flux d'unités NAL ce format peut être utilisé par ex. pour diffuser les données vidéo codées vers ou depuis un fichier, ou sur un réseau. L'utilisation du format de flux d'octets n'est pas nécessaire si la couche de transport fournit d'autres moyens pour encapsuler les données brutes de l'unité NAL[17].

2.3.1.1 Couche d'abstraction réseau (NAL units)

La syntaxe de haut niveau de HEVC conserve un certain nombre d'éléments utilisés dans H.264/AVC pour fournir une abstraction du réseau et la base de la résilience aux erreurs. Les NALU sont toujours utilisées comme interface de base avec le mécanisme de transport et HEVC prend en charge 63 types NALU (VCL et non-VCL), la couche d'abstraction du réseau (NAL) définit l'encapsulation des données vidéo codées pour le transport et le stockage. Les informations codées sont organisées en unités NAL, qui contiennent soit des données de la couche de codage vidéo (VCL), soit des données non VCL. La couche de codage vidéo comprend toutes les données de tranche et de CTU. Les données non-VCL incluent les ensembles de paramètres et d'autres informations comme détaillé ci-dessous. L'ensemble ordonné de toutes les unités NAL VCL et non

VCL qui représentent la séquence vidéo codée est appelé flux d'unités NAL. Selon le contexte, le flux unitaire NAL ou le flux d'octets selon le format de flux d'octets sont également appelés flux binaires[17, 19].

La charge utile d'une unité NAL (NALU) est une séquence de NU octets. Les deux premiers octets contiennent l'en-tête d'unité NAL, les octets restants sont appelés charge utile de séquence d'octets bruts (RBSP). Le dernier octet du RBSP contenant des données codées comprend le bit d'arrêt RBSP ('1') suivi de bits '0' de sorte que l'alignement des octets est réalisé. La concaténation des bits des octets RBSP avant le bit d'arrêt RBSP constitue la chaîne de bits de données (SODB). La SODB contient toutes les données codées nécessaires au processus de décodage. Dans la SODB, les bits des octets sont concaténés du MSB au LSB. Une illustration des octets d'unité NAL comprenant RBSP et SODB, voire à la Figure. 2.1.

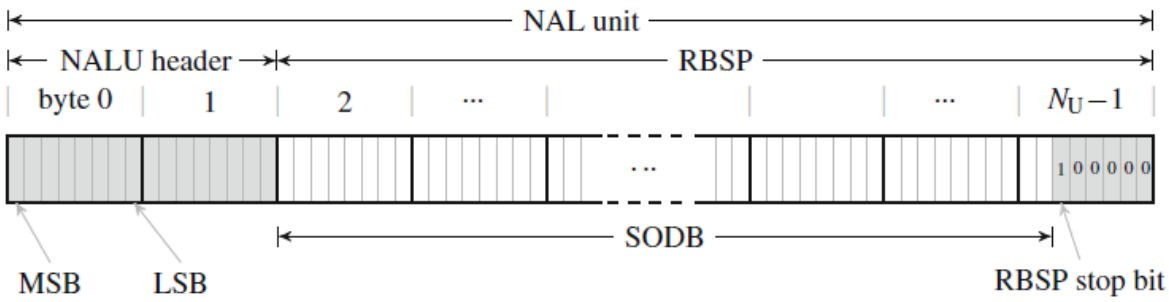


Figure 2.1 : *Représentation NAL units*

L'en-tête d'unité NAL identifie le type d'unité NAL et sa relation avec d'autres unités NAL dans le flux d'unités NAL. La structure de l'en-tête d'unité NAL est représentée dans la figure. 2.2.

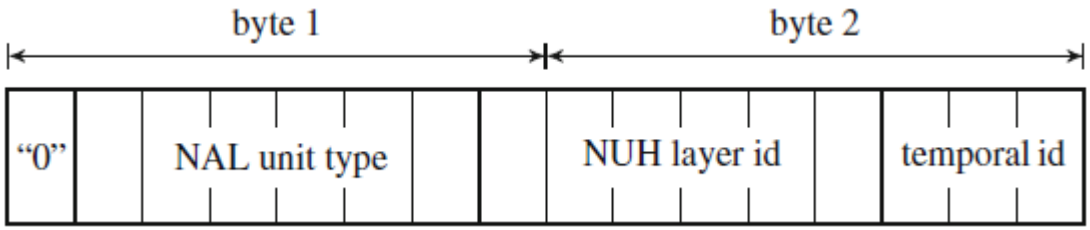


Figure 2.2 : *Structure de l'en-tête d'unité NAL à 2 octets*

Les unités NAL sont classées en unités NAL VCL et non-VCL selon qu'elles contiennent respectivement des images codées ou d'autres données associées. Dans la norme HEVC, plusieurs types d'unités VCL NAL identifiant des catégories d'images

pour l'initialisation du décodeur et à des fins d'accès aléatoire sont inclus. Le tableau 2.1 répertorie les types d'unités NAL et leurs significations et classes de types associées dans la norme HEVC[19,20].

Tableau 2.1 Types d'unités NAL, significations et classes

Type	Signification	Classe
0, 1	Slice segment of ordinary trailing picture	VCL
2, 3	Slice segment of TSA picture	VCL
4, 5	Slice segment of STSA picture	VCL
6, 7	Slice segment of RADL picture	VCL
8, 9	Slice segment of RASL picture	VCL
10–15	Reserved for future use	VCL
16–18	Slice segment of BLA picture	VCL
19, 20	Slice segment of IDR picture	VCL
21	Slice segment of CRA picture	VCL
22–31	Reserved for future use	VCL
32	Video parameter set (VPS)	non-VCL
33	Sequence parameter set (SPS)	non-VCL
34	Picture parameter set (PPS)	non-VCL
35	Access unit delimiter	non-VCL
36	End of sequence	non-VCL
37	End of bitstream	non-VCL
38	Filler data	non-VCL
39, 40	SEI messages	non-VCL
41–47	Reserved for future use	non-VCL
48–63	Unspecified (available for system use)	non-VCL

2.3.1.2 Jeux de paramètres [19]

Les mêmes jeux de paramètres que dans H.264/AVC sont utilisés, complétés par un nouveau jeu de paramètres vidéo (VPS). VPS peut inclure des métadonnées pour décrire les caractéristiques générales des couches codées, comme cela pourrait être utile dans les futures extensions évolutives. Les informations contenues dans l'en-tête NALU prennent également en charge l'identification des couches temporelles.

2.3.1.3 Jeux d'images de référence et listes d'images de référence

Il est important de gérer le contenu du tampon d'image décodée (DPB) pour assurer la synchronisation entre l'encodeur et le décodeur. L'ensemble d'images conservé est appelé ensemble d'images de référence (RPS). HEVC prend en charge deux listes d'images de référence, RPL0 et RPL1. Dans le cas d'une prédiction unidirectionnelle, une image de référence peut être sélectionnée dans l'une ou l'autre des listes et dans le

cas d'une prédiction bidirectionnelle, une image doit provenir de chaque liste. Certaines améliorations ont été apportées à HEVC pour rendre l'identification du RPS plus robuste que dans H.264/AVC [21].

2.3.1.4 Métadonnées d'informations SEI et VUI

La syntaxe inclut prendre en charge de différents type de métadonnées appelées SEI et VUI. De telles données fournissent des informations sur la synchronisation des images vidéo, l'interprétation correcte de l'espace colorimétrique utilisé dans le signal vidéo, des informations d'emballage de trame stéréoscopique 3D, d'autres informations d'indication d'affichage, etc.

2.3.1.5 WPP traitement parallèle du front d'onde

Lorsque WPP est activé, une tranche est divisée en rangées de CTU. La première ligne est traitée de manière ordinaire, la deuxième ligne peut commencer à être traitée après que seulement deux CTU ont été traitées dans la première ligne, la troisième ligne peut commencer à être traitée après que seulement deux CTU ont été traitées dans la deuxième ligne, et ainsi de suite. Les modèles de contexte du codeur entropique de chaque ligne sont déduits de ceux de la ligne précédente avec un retard de traitement de deux CTU. WPP fournit une forme de parallélisme de traitement à un niveau de granularité assez fin, c'est-à-dire au sein d'une tranche. WPP peut souvent fournir de meilleures performances de compression que les mosaïques (et éviter certains artefacts visuels pouvant être induits par l'utilisation de tiles).

2.4 La couche de codage vidéo (VCL)

Semblable à H.264, HEVC utilise une structure de codage hybride combinant une compensation de mouvement avec une transformation de bloc et un codage entropique. Plusieurs innovations ont été incorporées dans HEVC qui offrent des gains de performances significatifs .la figure 2.3 représente le schéma fonctionnel d'un encodeur vidéo hybride, qui pourrait créer un flux binaire conforme à la norme HEVC

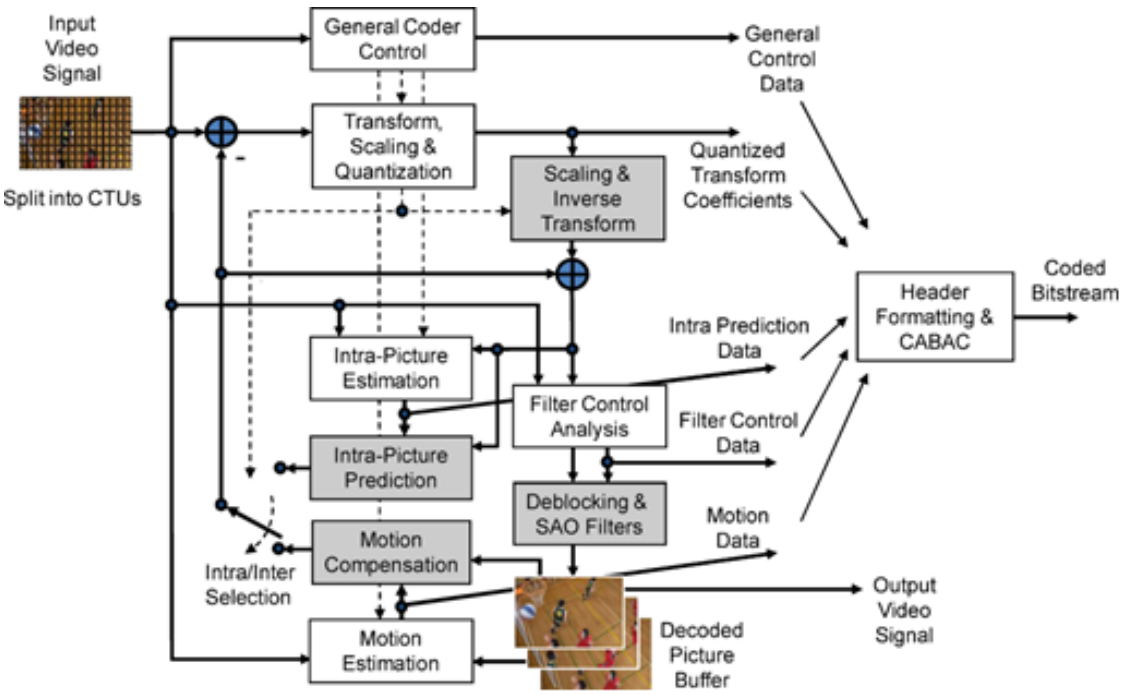


Figure. 2.3 : *Schéma fonctionnel typique d'un encodeur HEVC.*

2.4.1 Images et partitions

HEVC prend en charge un vaste éventail de tailles et de fréquences d'images. Les types d'images de base pris en charge dans HEVC sont identiques à ceux de H.264/AVC, à la petite exception près d'une version modifiée de la trame IDR, maintenant connue sous le nom d'image à accès aléatoire propre (CRA).

Une unité de codage HEVC (CU) peut contenir jusqu'à trois blocs (tableaux d'échantillons de luminance et de chrominance) et toute information requise pour décoder ces blocs. Chaque CU est subdivisée en unités de prédiction (PU) et celles-ci sont codés à l'aide de prédictions intra ou inter. Les paramètres de prédiction appropriés (inter- ou intra-) sont signalés au sein de chaque PU. Un arbre quaternaire résiduel (RQT) est ensuite utilisé pour diviser la CU en unités de transformation (TU) dans lesquelles les informations résiduelles sont codées. Ces unités contiennent toutes des blocs de codage (CB), des PB et des blocs de transformation (TB) pour les composants luma et chroma constitutifs, comme décrit ci-dessus. Cette structure est décrite plus en détail dans [22]. La décomposition de l'image HEVC est décrite plus en détail ci-dessous.

Le traitement d'image parallèle est réalisé à l'aide de :

- Slice/Segments de Slice
- Tiles
- Traitement parallèle du front d'onde (WPP)

2.4.1.1 Slice et Tiles

Les slices sont des zones d'image qui peuvent être décodées de manière indépendante. Dans une trame, chaque slice peut être décodée sans nécessiter de données provenant d'une autre slice de la même image. Elles introduisent toutefois une surcharge en raison de la présence d'en-têtes de slice et du fait que chaque frontière de slice interrompt la prédiction entre les slices.

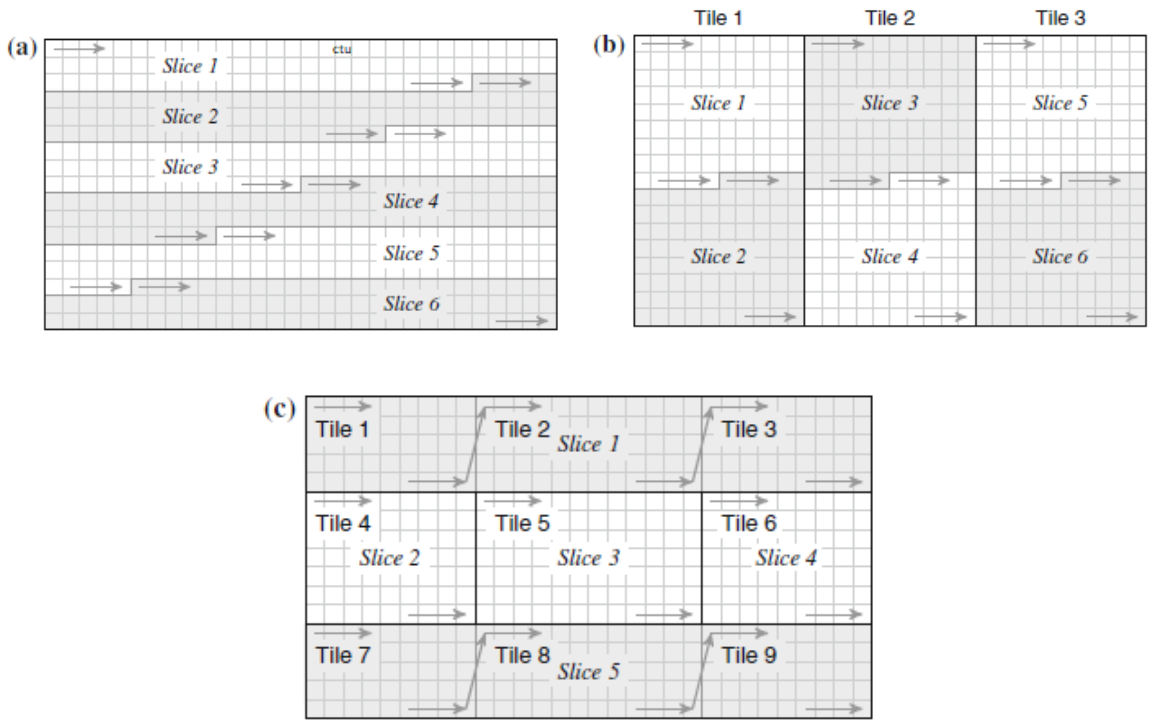


Figure 2.4 : *Partition d'une image de 30 × 17 CTU de codage en tranches et tiles. (a)slice seulement. (b) slice en tiles. (c) Tiles en slice*

L'option permettant de partitionner une image en régions rectangulaires appelées tiles a été spécifiée. L'objectif principal des tiles est d'augmenter la capacité de traitement parallèle plutôt que de fournir une résistance aux erreurs. Les tiles sont des régions indépendamment décodables d'une image qui sont codées avec certaines informations

d'en-tête partagées. Les tiles peuvent en outre être utilisées à des fins d'accès spatial aléatoire à des régions locales d'images vidéo. Une configuration de tiles typique d'une image consiste à segmenter l'image en régions rectangulaires avec des nombres approximativement égaux de CTU dans chaque tiles. Les tiles fournissent un parallélisme à un niveau de granularité plus grossier (image/sous-image), et aucune synchronisation sophistiquée des threads n'est nécessaire pour leur utilisation. Voir figure 2.4.

2.4.1.2 CTB (Coding Tree Block) et Coding Blocks (CB)

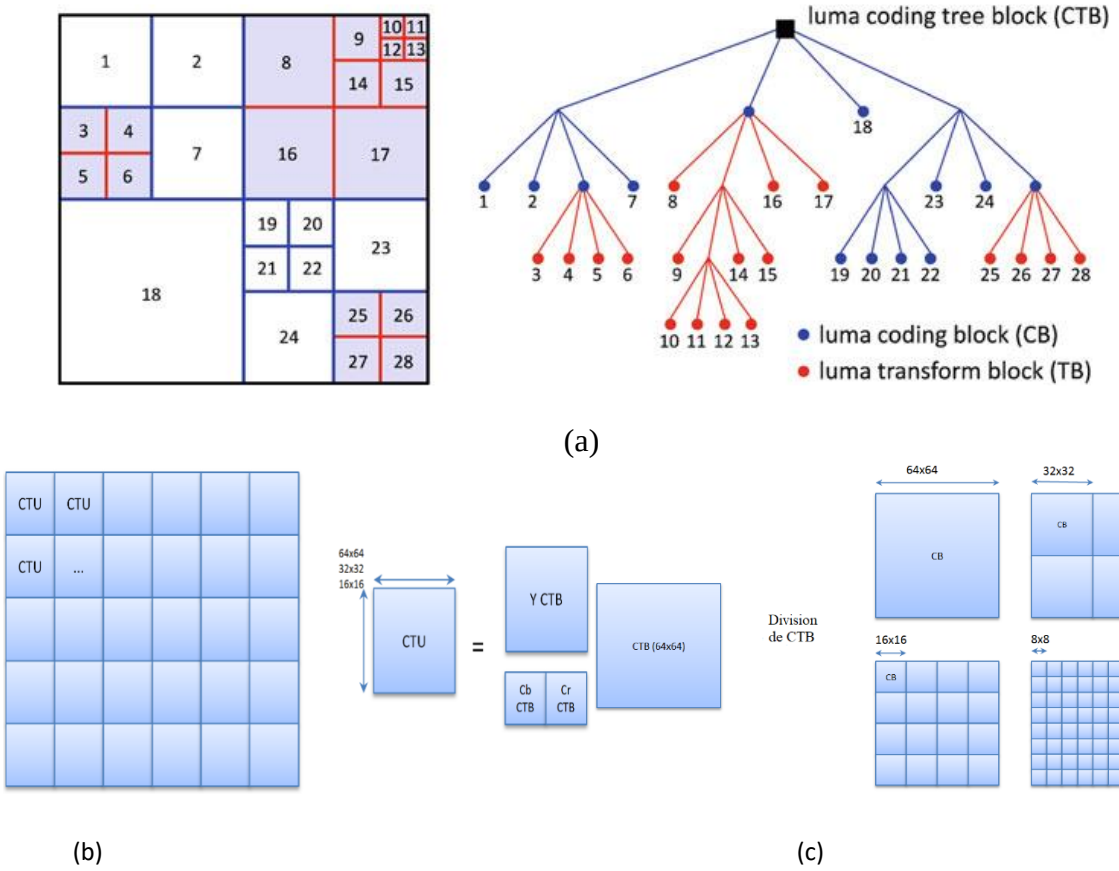
L'image est partitionnée en (CTB) de $2^n \times 2^n$ échantillons, où $n \in \{4, 5, 6\}$ [19]sont spécifié dans le profil principal. La plus petite taille de CTB de 16 x 16 correspond à la taille de macrobloc telle qu'utilisée dans les normes de codage vidéo précédentes. En modifiant la taille du CTB jusqu'à 64 x 64, la structure de codage peut être adaptée à des résolutions d'image plus importantes. Le CTB est la racine d'une structure d'arbre quaternaire de blocs de codage (CB) de taille carrée. Le bloc de codage est à son tour la racine de la structure du bloc de prédiction et l'arbre quaternaire résiduel. Un CTB peut être représenté par un seul CB, ou il peut être divisé en quatre cloisons de taille moitié horizontale et moitié verticale. Chacune de ces partitions peut être un CB lui-même ou peut être en outre divisée en une taille moitié horizontale et moitié verticale, et ainsi de suite.

2.4.1.3 CTU (Coding tree units)et CTB (coding tree block)

HEVC remplace le macrobloc des normes antérieures par un CTU (L × L) où L = 16, 32, 64. Cela permet un codage plus efficace pour les formats à plus haute résolution. Le CTU est composé de blocs d'arbre de codage luma et chroma (CTB). Chaque image est segmentée en séquences de CTU dans l'ordre de balayage raster, et chacune de ces séquences de CTU est appelée slice. Chaque slice a un en-tête qui lui permet d'être décodée indépendamment de toutes les autres tranches de l'image. Les CTB peuvent être partitionnés en CU comprenant des CB luma et chroma avec des tailles de bloc plus petites à l'aide d'une décomposition quadtree. La taille de la plus grande CU (LCU) est

de 64 × 64. La manière dont la CTU est décomposée n'est pas spécifiée dans la norme (elle est non normative) mais constitue néanmoins une partie importante du traitement RDO du codeur. Un exemple de partitionnement CTU est donné à la Figure 2.5(a). Chaque CU est subdivisée en TU et en PU. Cette approche offre à l'encodeur beaucoup plus de flexibilité pour optimiser les partitions de blocs en fonction du contenu vidéo[23].

Les CTB de chaque CTU sont partitionnés en blocs de codage (CB), comme indiqué par une structure quadtree (Figure 2.5 (b), (c).). Lorsqu'un luma CTB est divisé par l'arbre quaternaire, les CB luma et chroma sont divisés ensemble, et un CB luma peut être aussi petit que 8x8 (accompagné de deux CB 4x4 chroma) ; seul l'échantillonnage de couleur 4:2:0 est pris en charge, de sorte que les structures de chrominance correspondantes ont toujours la moitié de la taille du luma à la fois horizontalement et verticalement. Un CB luma avec les deux CB chroma correspondants et les éléments de syntaxe associés est appelé une unité de codage (CU).



(a)

(b) (c)

Figure 2. 5 : *(a) Le partitionnement d'un CTB luma (noir) en blocs de codage (bleu) et blocs de transformation (rouge), (b) CTU et CTB, (c) CTB* divisé *en CB*

2.4.1.4 Unités de prédiction (PU)

Le choix du genre de prédiction (intra, inter) est pris pour chaque CU. Les unités de codage peuvent encore être trop grandes, par exemple pour de très petits objets en mouvement. Pour cette raison, les CB peuvent être divisés en blocs de prédiction,

auxquels sont attribués, par exemple, un vecteur de mouvement. Une CU peut être découpé en une, deux ou quatre PU selon le type de découpage des PU. HEVC définit deux formes de séparation pour la CU codée intra et huit formes de séparation pour la CU codée inter. Contrairement à la CU, la PU ne peut être scindée qu'une seule fois [22].

- **Type de fractionnement PU**

chaque CU dans HEVC peut être classée en trois catégories : skipped CU, inter coded CU, et inra coded CU . Une CU codée en inter utilise un schéma de compensation de mouvement pour la prédiction du bloc actuel, tandis qu'une CU codée en intra utilise des échantillons reconstruits voisins pour la prédiction. Une skipped CU est une forme spéciale de CU intercodée où la différence de vecteur de mouvement et l'énergie résiduelle sont égales à zéro. Le type de fractionnement PU est spécifié différemment Pour chaque catégorie, voir la Figure 2.6

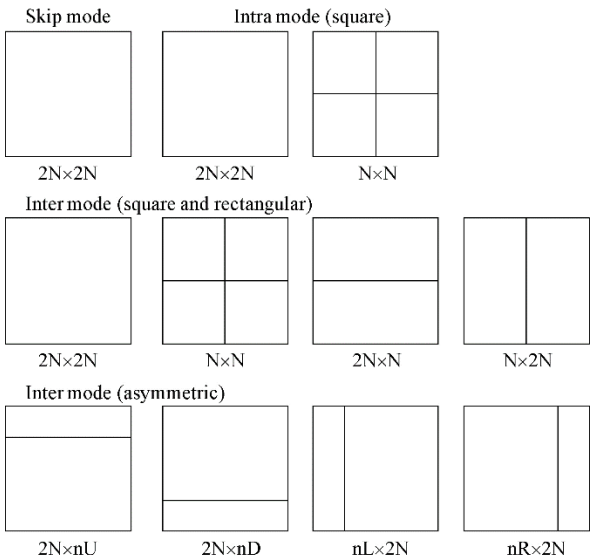


Figure 2.6: *Illustration des types de fractionnement du PU dans HEVC.*

- **Codage intra prédiction**

Le nombre de directions de prédiction intra dépend de la taille du CU, comme le montre la Figure 2.7 : 33 modes angulaires pour 8x8. **[24]**

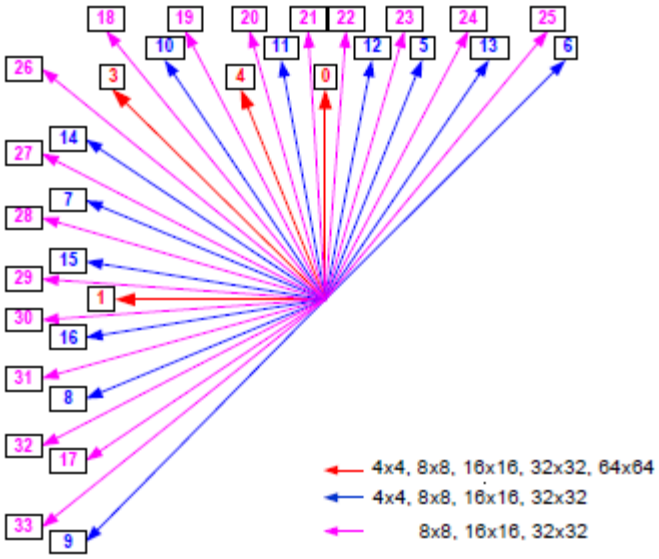


Figure 2.7 : *Les modes Intra prédiction de HEVC*

Comme pour H.264/AVC, le mode DC correspond à la valeur moyenne des échantillons de la colonne gauche et de la ligne supérieure, et I_PCM permet de coder directement les valeurs des échantillons. De plus, deux nouveaux modes intra ont été introduits : le mode "planar" et le mode "LM Chroma", où les échantillons de chrominance sont prédits à partir des échantillons de luminance reconstruits. Les bords supérieur et gauche des prédictions Intra DC et angulaires sont filtrés. La direction intra et la taille du CU déterminent le nombre de prises de filtre (Mode Dependent Intra Smoothing). Le décodage du mode Intra se fait en deux étapes. Premièrement, les trois modes les plus probables (MPM) sont dérivés des CU codés voisins. Ensuite, un arbre de mode de décodage permet de dériver le mode intra à partir d'un indicateur binaire et de un ou deux éléments de syntaxes supplémentaires.

- **Codage Inter Prédiction :** La prédiction à compensation de mouvement est généralement l'outil clé pour une représentation efficace du contenu vidéo. Les changements entre les images successives

dans la séquence vidéo sont approchés par un déplacement par bloc entre les zones de l'image actuelle et les images précédemment codées qui sont disponibles dans la mémoire tampon d'image décodée à la fois du côté du codeur et du côté du décodeur. Étant donné que la prédiction à compensation de mouvement est effectuée entre les images, ce mode de prédiction est appelé mode Inter Prédiction. La précision spécifiée des informations de mouvement pour la compensation de mouvement est d'un quart d'échantillon pour la composante luma et d'un huitième échantillon pour les composantes chroma. Les méthodes avancées de dérivation de vecteur de mouvement et de prédiction de vecteur de mouvement mises en œuvre jouent un rôle clé dans l'efficacité de compression de HEVC. Les filtres d'interpolation de sous-échantillon fournissent un signal de prédiction amélioré aux emplacements de sous-échantillon par rapport aux filtres d'interpolation des spécifications de codage vidéo précédentes. Pour prédire le vecteur de mouvement, HEVC introduit AMVP (Advanced Motion Vector Prediction) en utilisant la corrélation spatiale et temporelle du PU actuel

De plus, il adopte un mode de fusion dans lequel les PU forment une zone fusionnée qui partage toutes les données de mouvement et les informations de mouvement ne sont signalées qu'une seule fois [25].

HEVC adopte un processus de décision de mode pour trouver le mode de prédiction optimal, qui traverse tous les modes PU possibles pour trouver celui avec le moindre coût RD en utilisant le multiplicateur de Lagrange. L'estimation de mouvement est le processus de recherche du meilleur PU correspondant du PU actuel dans une fenêtre de recherche d'image de référence et le coût RD peut être calculé par la formule suivante :

$$J_{min}\ J_{mode} = D_{mode} + \lambda_{mode} \cdot R_{mode} \qquad (2.1)$$

où D_{mode} donne le résidu de prédiction du compensation de mouvement,

λ_{Mode} représente le multiplicateur lagrangien et R_{mode} spécifie la consommation de bits pour la décision de mode (MV, index d'images de référence, etc.)Figure 2.7.

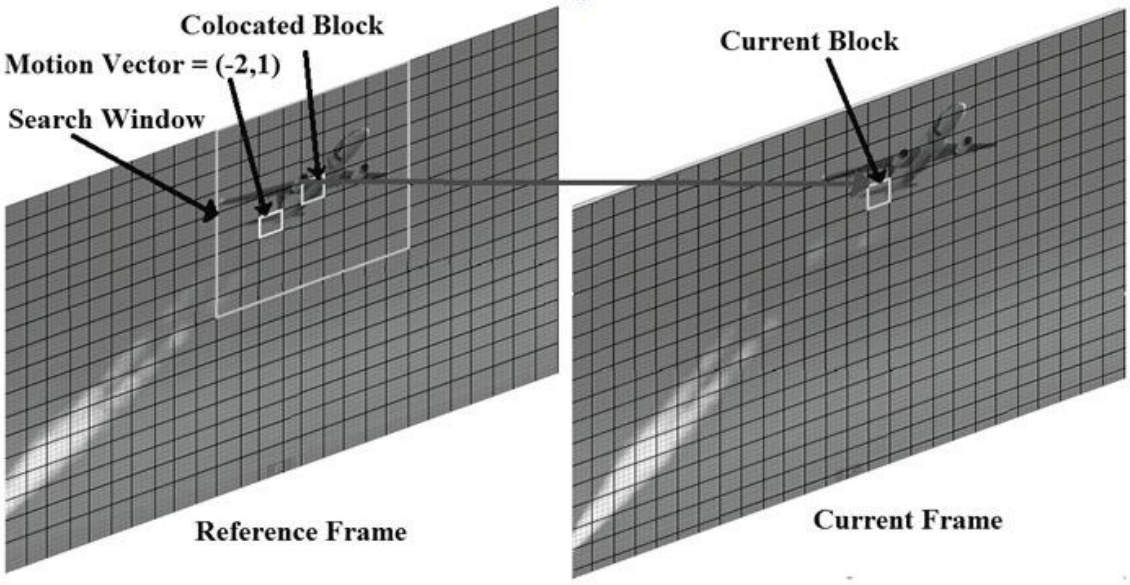


Figure 2.8 : *Estimation du mouvement.*

Dans chaque mode PU illustrer à la Figure 2.8. , l'algorithme ME recherche dans une fenêtre de recherche des cadres de référence et renvoie celui avec le moins de coût RD (figure 2.9).

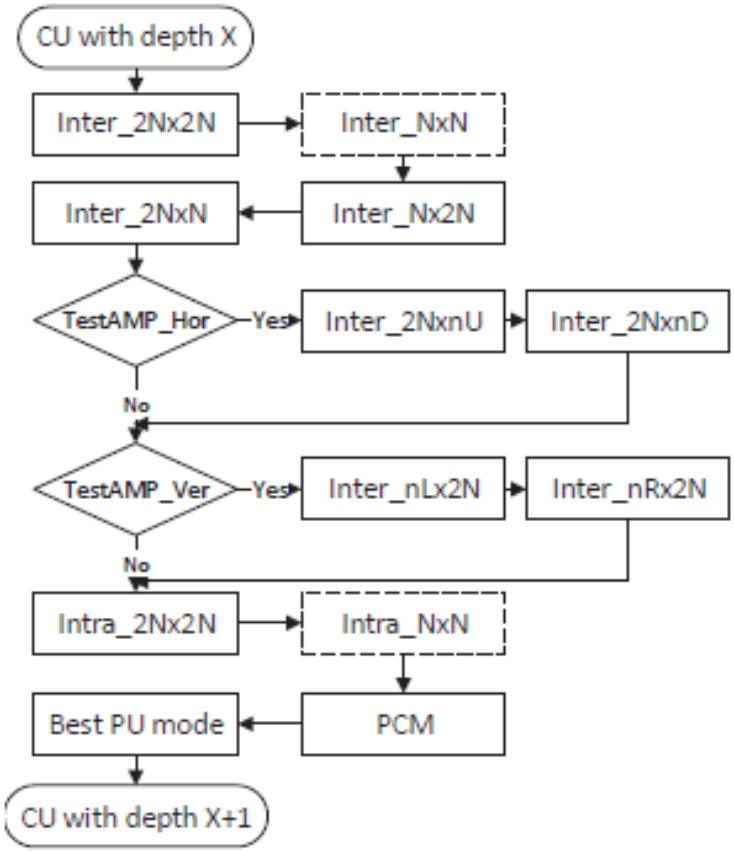


Figure 2.9: *Processus de décision pour le mode PU de HM16.9.*

2.4.2 Transformation et quantification HEVC

TU spécifie l'unité de base pour la transformation et la quantification. Lorsque la taille d'une CU est déterminée, la CU est partitionnée en PU pour la prédiction et chaque PU est ensuite subdivisée en TU afin de trouver le schéma de partitionnement avec le moindre coût de distorsion de débit (RD). La transformation cosinus discrète (DCT) avec une autre transformation sinusoïdale discrète (DST) est utilisée par HEVC avant la quantification pour obtenir un codage sans perte des blocs de transformation. Ces deux transformations sont implémentées dans l'encodeur HEVC à l'aide du processeur.

2.4.2.1 Estimation DCT et DST[23]

HEVC utilise une approximation de précision finie de la transformée en cosinus discrète bidimensionnelle de tailles 8x8 à 32x32 et une transformée en sinus discrète alternative de taille 4x4 qui est utilisée pour les blocs résiduels de prédiction luma intra. La conception a pris en compte un certain nombre de facteurs, par exemple la préservation des propriétés de symétrie et la profondeur de bits limitée pour une mise en œuvre plus facile Blocs de prédiction intra luma 4x4 qui est une transformation entière basée sur le DST. La norme spécifie des matrices pour la transformation de base de chaque taille de bloc (4x4,8x8,16x16,32x32) ainsi qu'une matrice pour la transformation alternative des blocs de prédiction intra luma 4x4 qui est une transformation entière basée sur le DST. Chaque coefficient de matrice est représenté par 8 bits (y compris le bit de signe) et mis à l'échelle selon un facteur de $2^{6+m/2}$ par rapport à la DCT orthonormée. Les matrices sont conçues pour préserver les propriétés d' (anti-)symétrie, ce qui nous permet d'obtenir la moitié droite de la matrice à partir de la moitié gauche. De plus, les coefficients de matrices plus petites (4x4, 8x8, 16x16) peuvent être dérivés de la matrice 32x32, la matrice de transformation inverse est également définie comme des matrices de transformation directe transposées.

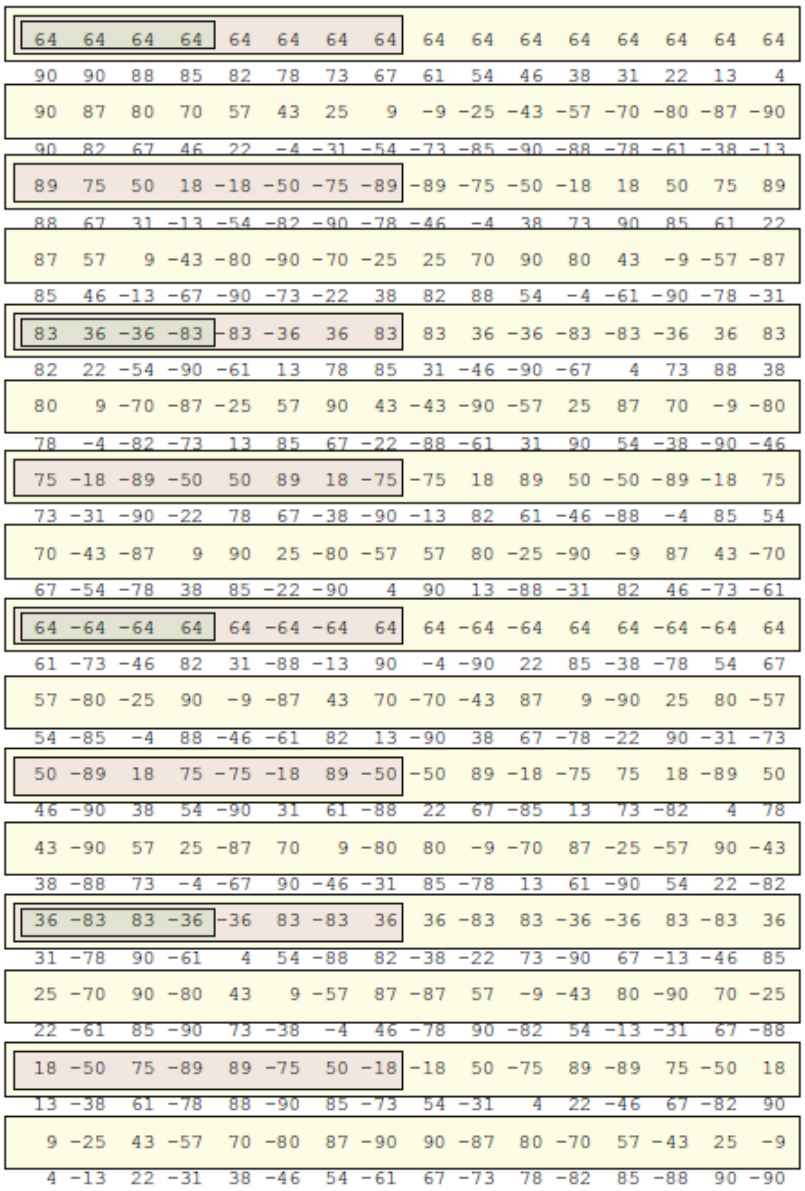


Figure 2.10 : *Moitié gauche de la matrice de transformation directe 32x32. les zones en surbrillance montrent des matrices 16x16, 8x8 et 4x4 qui peuvent être dérivées de la matrice 32x32[23].*

Les coefficients de transformation sont ensuite obtenus en appliquant une transformation 1D à deux étages pour les colonnes et les lignes respectivement, ce qui implique une multiplication matricielle des coefficients prédits résultants. Avec leurs matrices respectives. Enfin, les coefficients de transformation passent par la quantification à l'aide d'une échelle Q-step. L'inverse de cette procédure est la déquantification suivie d'une autre transformation en deux étapes qui est la multiplication de la transposition des matrices mentionnées. Ces étapes sont décrites dans la figure 2.10 : C est la matrice DCT orthonormée. D est l'approximation mise à l'échelle de la matrice DCT

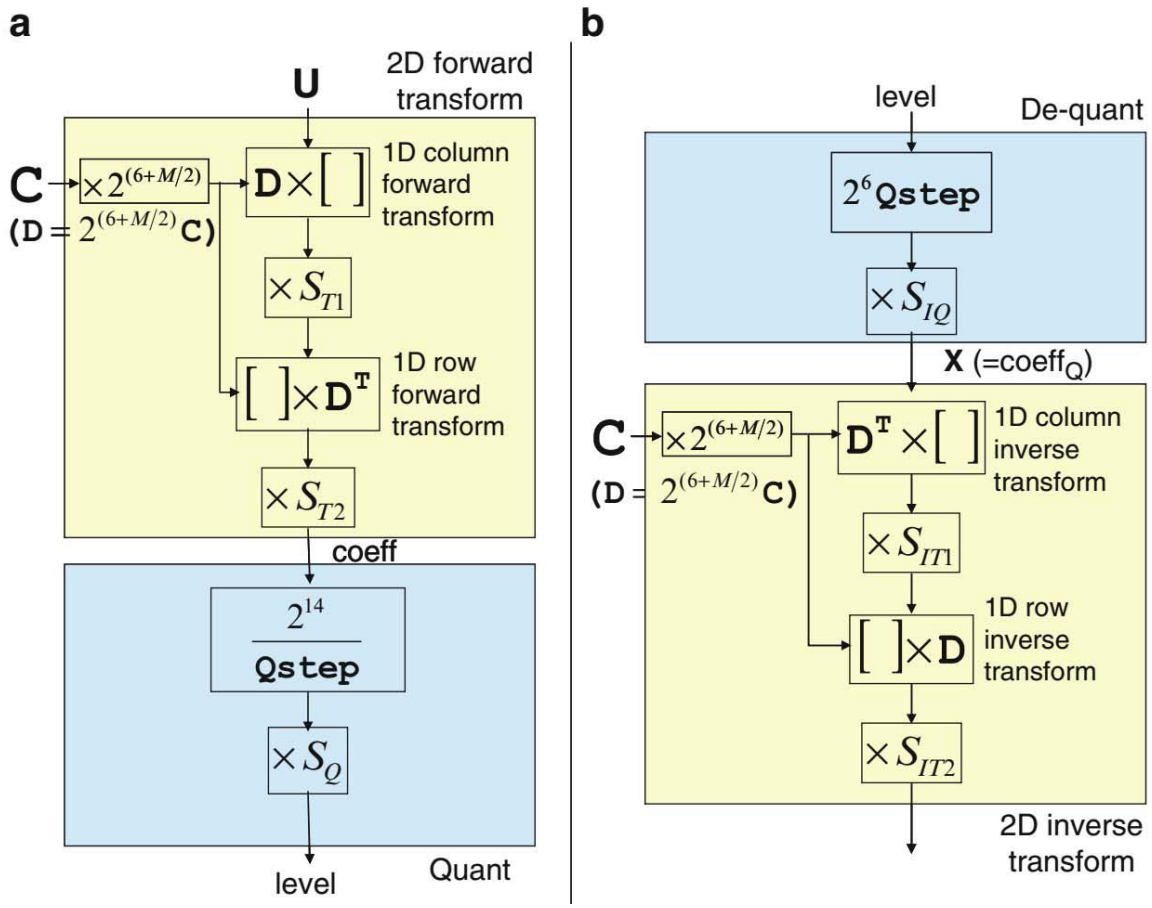


Figure 2.11: *(a) transformation directe et quantification, (b) transformation inverse et quantification [23]*

Comme mentionné précédemment, les matrices sont mises à l'échelle par $2^{6+m/2}$, donc afin de préserver la norme du bloc résiduel, un développement supplémentaire de la forme de S_{T1}, S_{T2}, S_{IT1}, S_{IT2} est nécessaire. Ces facteurs sont décrits dans la norme et sont conçus de manière à être applicables à l'aide d'opérations de décalage logique. Soit B la profondeur de bits tandis que $M=\log_2 N$ où N est la taille de la transformation, alors les facteurs d'échelle sont comme indiqué ci-dessous :

$$ST1=2^{-(B+M-9)},\ S_{T2}=2^{-(M+6)},\ _{SIT1}=2^{-7},\ S_{IT2}=2^{-(20-B)} \qquad (2.2)$$

Une procédure similaire est appliquée pour DST alternatif qui fournit une réduction de débit binaire de 1% sur les images intra. La matrice DST est illustrée ci-dessous :

$$A_4 = \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix} \qquad (2.3)$$

La quantification consiste en une division par une taille de pas de quantification (Qstep) et un arrondi ultérieur, et la quantification inverse est la multiplication par nombre de pas de quantification. Un paramètre de quantification (QP) est utilisé pour déterminer la taille du pas de quantification dans HEVC. QP peut prendre 52 valeurs de 0 à 51 pour des séquences vidéo 8 bits. Une augmentation de 1 dans QP signifie une augmentation de la taille du pas de quantification d'environ 12 %. Une augmentation de 6 augmente la taille du pas de quantification d'un facteur 2. Il est nécessaire de définir la taille de pas absolue qui est liée à la gamme des valeurs QP en addition de déterminer la différence relative entre les tailles de pas de deux valeurs QP consécutives. Cela a été accompli en choisissant Qstep=1 pour QP=4.

La relation entre QP et la taille de pas de quantification équivalente pour une transformée orthonormée est maintenant donnée par :

$$QSTEP = (2^{1/6})^{QP-4} \tag{2.4}$$

Où:

$$G=[G_0,G_1,G_2,G_3,G_4,G_5]^T=[2^{-4/6},2^{-3/6},2^{-2/6},2^{-1/6},2^0,2^{1/6}]^T \tag{2.5}$$

L'approximation en point fixe dans HEVC est donnée par :

$$g=[g_0,g_1,g_2,g_3,g_4,g_5]^T=[40,45,51.57,64,72]^T \tag{2.6}$$

Dans HEVC, l'encodeur peut signaler s'il faut ou non utiliser des matrices de quantification permettant une mise à l'échelle dépendante de la fréquence. La mise à l'échelle dépendante de la fréquence est utile pour effectuer une quantification basée sur le système visuel humain (HVS) où les coefficients de basse fréquence sont quantifiés avec une taille de pas de quantification plus fine par rapport aux coefficients de haute fréquence dans le bloc de transformation [26]. Quantification basée sur HVS peut fournir une meilleure qualité visuelle que la quantification indépendante de la fréquence sur certaines séquences vidéo. HEVC utilise les 20 matrices de quantification suivantes qui dépendent de la taille et du type du bloc de transformation :

- Luma : Intra4X4, Inter 4X 4, Intra 8X 8, Inter 8 X8, Intra 16X 16, Inter 16 X16, Intra 32X 32, Inter 32 X32 .
- Cb : Intra4X 4, Inter 4 X4, Intra 8X 8, Inter 8X 8, Intra 16 X16, Inter 16X 16 .

- Cr : Intra 4X 4, Inter 4 X4, Intra 8X 8, Inter 8X 8, Intra 16X 16, Inter 16X 16

2.4.3 Filtres « In-Loop »

HEVC comprend deux étapes de filtrage « In-Loop » avant de mettre à jour sa mémoire tampon d'image décodée : un processus DBF (Deblocking filter**)** et un processus SAO (Sample Adaptive Offset). Le filtre de déblocage est identique au filtre H.264/AVC, qui vise à réduire les artefacts de blocage dus au codage par blocs. SAO est nouvellement introduit dans HEVC et est appliqué sur tous les échantillons en ajoutant une simple valeur de décalage à chaque pixel en fonction de la catégorie.

2.4.3.1 Le filtre de déblocage (De-blocking filter)

Le filtre de déblocage HEVC est identique à celui de H.264/AVC mais simplifié et amélioré pour un meilleur support (en particulier pour la structure de traitement parallèle). Il est utilisé dans la boucle de prédiction inter-images. Cependant, la conception est simplifiée en ce qui concerne ses processus de prise de décision et de filtrage, et est rendue plus conviviale pour le traitement parallèle.

2.4.3.2 SAO (Sample adaptive offsets)

Une cartographie d'amplitude non linéaire est introduite dans la boucle de prédiction inter-images après le filtre de déblocage. Son objectif est de mieux reconstruire les amplitudes du signal d'origine a l'aide d'une table de correspondance qui décrit quelques paramètres supplémentaires qui peuvent être déterminés par analyse d'histogramme côté codeur.

SAO représente une procédure qui corrige les exemples décodés en ajoutant des décalages conditionnellement à chaque exemple. Cela fait suite à l'application des filtres de déblocage. Dans les premières étapes, SAO subdivise les images en parties conçues par LCU pour produire les données statistiques locales. Chaque pièce peut être développée par EO (Edge Offset) et BO (Band Offsets).

La compensation de décalage de bord (edge-offset) corrige les distorsions entre les pixels d'origine et les pixels filtrés par dégroupage en ajoutant la valeur de décalage aux valeurs de pixels filtrés par dégroupage pour chaque catégorie de bord. Dans le décalage de bande (Band Offsets), la valeur de décalage est sélectionnée en fonction de la valeur

d'échantillon. Toute la plage d'amplitude de l'échantillon est divisée en 32 bandes égales (chaque bande a son propre décalage) qui sont ensuite divisées en deux groupes, une bande centrale composée de 16 bandes et un autre groupe contenant les bandes restantes. Un encodeur sélectionne une bande de groupe pour appliquer SAO[27].

La décision de sélectionner la compensation de décalage de bord ou la compensation de décalage de bande pour le filtrage SAO est prise dans un sens RDO durant le processus de codage HEVC[28].

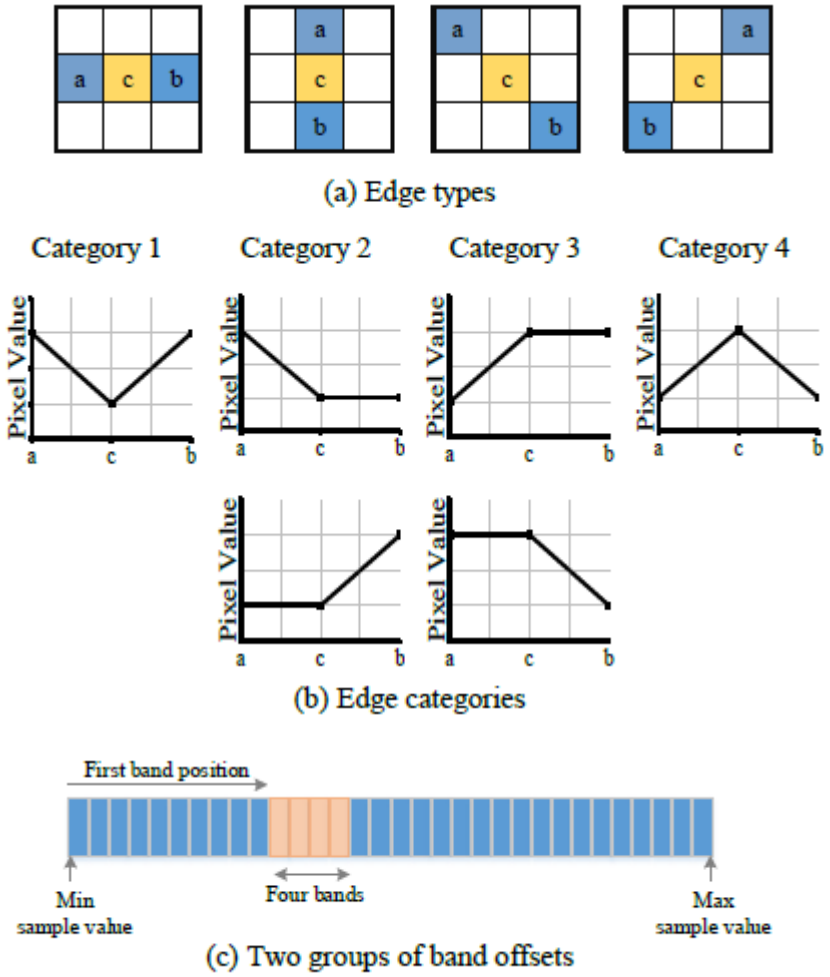


Figure 2.12 : *Types et catégories de bords pour la compensation de décalage de bord*

La figure 2.12 décrit les quatre types de bords et les quatre catégories de filtrage HEVC SAO pour la compensation de décalage de bord, et illustre les bandes d'histogramme de valeurs de pixel filtrées par déblocage pour la compensation de décalage de bande. Comme le montre la figure 2.12-(a), il existe quatre types de bords définis avec les quatre modèles directionnels 1-D tels que les directions horizontale, verticale, diagonale 135° et directions diagonales 45^{0}. En utilisant un type de bord, le

pixel actuel filtré par déblocage à l'emplacement c est classé dans l'une des quatre catégories de bord en le comparant aux deux autres valeurs de pixel situées en a et b. C'est-à-dire que la valeur de pixel actuelle en c est inférieure (ou supérieure) à ses deux valeurs de pixel voisines en a et b, elle est classée dans la catégorie 1 (ou 4). Si la valeur de pixel actuelle est la même que l'une des deux et est inférieure (ou supérieure) à l'autre, elle est classée dans la catégorie 2 (ou 3). Dans un bloc CTU, chaque pixel filtré par déblocage en c est classé dans l'une des quatre catégories de bord à l'aide d'un modèle de type de bord donné. Ensuite, tous les pixels du bloc CTU sont regroupés en fonction de leurs catégories de bord et la valeur de décalage moyenne est calculée pour chaque catégorie de bord. Ceci est fait pour les quatre types de bords et le meilleur type de bord est sélectionné dans un sens RDO pour le taux total et la distorsion due aux quatre valeurs de décalage. Par conséquent, le type de bord final et les valeurs de décalage résultantes pour les quatre catégories de bords sont écrits dans des flux binaires

La figure 2.12-(c) illustre la compensation de décalage de bande pour le filtrage SAO[28] . Tous les pixels du bloc CTU sont regroupés en 32 groupes en fonction de leurs valeurs pour la compensation de décalage de bande. Les valeurs de décalage moyennes pour seulement quatre groupes consécutifs sont transmises avec les premières informations de position de bande au côté décodeur pour corriger la distorsion entre les pixels d'origine et reconstruits.

2.4.4 Codage entropique

Le codage arithmétique binaire adaptatif (CABAC) est utilisé pour le codage entropique. Ceci est similaire au schéma CABAC dans H.264 / MPEG-4 AVC, mais a été amélioré par plusieurs améliorations pour augmenter sa vitesse de débit et meilleures performances de compression et moins de besoins de mémoire de contexte.

HEVC utilise Context Adaptive Binary Arithmetic Coder (CABAC) comme principal moteur de codage entropique et se compose de trois étapes : la binarisation des éléments de syntaxe, la modélisation du contexte ou le mode de contournement et le codage arithmétique binaire[29].

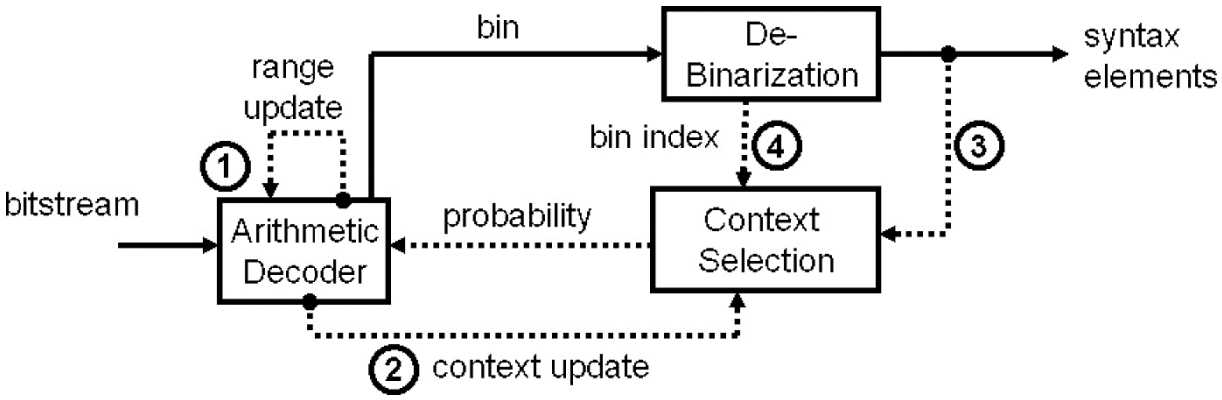


Figure. 2.13 : *Les Trois opérations clés dans CABAC*

2.4.4.1 Binarisation

Plusieurs processus de binarisation différents sont utilisés dans HEVC, notamment unaire (U), unaire tronqué (TU), Exp-Golomb d'ordre k (EGk) et longueur fixe (FL). Ces formes de binarisation ont également été utilisées dans H.264/AVC. Ces diverses méthodes de binarisation peuvent être expliquées en termes de la manière dont elles signaleraient une valeur non signée N.

1) Le codage unaire implique la signalisation d'une chaîne bin de longueur N+1, où les N premiers bins sont 1 et le dernier bin est 0. Le décodeur recherche un 0 pour déterminer quand l'élément de syntaxe est complet.

2) Le codage unaire tronqué a une case de moins que le codage unaire en fixant un maximum sur la plus grande valeur possible de l'élément syntaxique (cMax). Lorsque N+1<cMax, la signalisation est la même qu'en codage unaire. Cependant, lorsque N+1=cMax, tous les bins sont 1. Le décodeur recherche les bins de 0 à cMax pour déterminer quand l'élément de syntaxe est complet.

3) $k^{ième}$ ordre Exp-Golomb est un type de code universel. La distribution peut être modifiée en fonction du paramètre k.

4) La longueur fixe utilise un nombre fixe de cases, ceil(log2(cMax+1)), avec les bits les plus significatifs signalés avant les bits les moins significatifs.

2.4.4.2 Modélisation du contexte

La modélisation de contexte fournit une estimation de probabilité précise requise pour atteindre une efficacité de codage élevée. En conséquence, il est hautement adaptatif et différents modèles de contexte peuvent être utilisés pour différents casiers et La probabilité de ce modèle de contexte est mise à jour en fonction des valeurs des casiers codés auparavant. Les casiers avec des distributions similaires partagent souvent le même modèle de contexte. Le modèle de contexte pour chaque bin peut être sélectionné en fonction du type d'élément de syntaxe, de la position du bin dans l'élément de syntaxe (binIdx), de la luma/chroma, des informations voisines, etc. Un changement de contexte peut se produire après chaque bin. Les modèles de probabilité sont stockés sous forme d'entrées 7-b [6-b pour l'état de probabilité et 1-b pour le symbole le plus probable (MPS)] dans une mémoire de contexte et adressés à l'aide de l'indice de contexte calculé par la logique de sélection de contexte. HEVC utilise la même méthode de mise à jour de probabilité que H.264/AVC ; cependant, la logique de sélection de contexte a été modifiée pour améliorer le débit.

2.4.4.3 Codage arithmétique[29]

Le codage arithmétique est basé sur la division d'intervalle récursive. Une plage, avec une valeur initiale de 0 à 1, est divisée en deux sous-intervalles en fonction de la probabilité de la case. Les bits codés fournissent un décalage qui, lorsqu'il est converti en une fraction binaire, sélectionne l'un des deux sous-intervalles, qui indique la valeur de la case décodée. Après chaque casier décodé, la plage est mise à jour pour être égale au sous-intervalle sélectionné, et le processus de division d'intervalle se répète. La plage et le décalage ont une précision de bit limitée, de sorte qu'une renormalisation est nécessaire chaque fois que la plage tombe en dessous d'une certaine valeur pour éviter le sous-dépassement. La renormalisation peut se produire après le décodage de chaque classe. Le codage arithmétique peut être effectué en utilisant une probabilité estimée (codée par le contexte) ou en supposant une probabilité égale de 0,5 (codée par contournement). Pour les casiers codés de contournement, la division de la plage en sous-intervalles peut être effectuée par un décalage, alors qu'une table de consultation

est requise pour les casiers codés contextuels. HEVC utilise le même codage arithmétique que H.264/AVC.

2.5 Conclusion

La norme HEVC émergente a été développée et normalisée en collaboration par les organisations ITU-T VCEG et ISO/IEC MPEG. HEVC représente un certain nombre d'avancées dans la technologie de codage vidéo. Sa conception de couche de codage vidéo est basée sur des concepts de codage vidéo combinés de mouvement basés sur des blocs conventionnels, mais avec quelques différences importantes par rapport aux normes antérieures.

3 .

CHAPITRE 3 : LES TRAVAUX RECENTS POUR REDUCTION DE LA COMPLEXITE DU HEVC

3.1 Introduction

Récemment, l'optimisation du partitionnement CU pour les encodeurs HEVC a attiré beaucoup d'attention des chercheurs. Un élément majeur de la norme HEVC est la structure de partitionnement utilisée qui a un impact direct sur l'efficacité de la compression. Les différentes tailles de CU, PU et TU adoptées par la norme HEVC ont assuré un compromis entre débit et qualité. Cependant, le nombre élevé de formes possibles de CU a considérablement augmenté le temps de codage. Ainsi, la plupart des chercheurs se concentrent sur cette étape pour l'optimisation du codeur HEVC.

L'objectif de ce chapitre est d'étudier la plupart des méthodes de réduction du taux de calcul pour HEVC.

3.2 Un codage spatio-temporel pour Intercodage HEVC [30]

Proposé un modèle de codage CU rapide pour les codeurs HEVC basé sur des paramètres de codage spatio-temporel. Ils ont utilisé les paramètres de décalage d'échantillon adaptatifs (SAO) pour le bloc CU actuel sous codage pour estimer la complexité de la texture qui affecte la partition CU. Les auteurs ont utilisé les paramètres de codage spatial et temporel de manière optimale sans calculs supplémentaires. Cependant, l'idée de base de la solution proposée dans [30] est l'utilisation du filtre Sample-Adaptive-Offset (SAO), qui peut être considéré comme l'un des processus les plus gourmands en temps de calcul dans HEVC. La compensation de décalage de bande (band offset) n'est pas utilisée dans l' analyse de complexité de texture car elle ne transmet pas les informations de texture .Ainsi, il convient de noter dans le schéma de codage CU rapide proposé que les informations de catégorie de bord obtenues pendant le processus SAO basé sur RDO sont stockées dans la trame suivante pour le schéma de codage CU rapide. La méthode est basée premièrement sur la mesure de la complexité du bloc CTU sur la base d'une métrique objective de la complexité de la texture appelée la densité des bords, définie comme

$$\tau = (1/M^2) \sum_y^M \sum_x^M bord(x, y) \qquad (4.1)$$

Où M indique la taille de la CTU et bord(x, y) peut prendre une valeur de front (0 ou 1) à la position (x, y) dans la CTU.

Il à trouver, Les catégories 1 et 4 ont la première place dans la CTU de complexité (τ= 0,32), tandis que les catégories 2 et 3 ont la première place dans la CTU homogène (τ= 0,06).Cela a motivé notre décision rapide de fractionnement de CU basée sur les paramètres SAO. Partant de ce constat, nous pouvons établir deux hypothèses : Hypothèse 1 - si la Catégorie 1 et la Catégorie 4 sont dominantes dans un bloc CU, le bloc CU est fortement texturé et a tendance à être scindé en bloc CU de plus petites tailles ; et Hypothèse 2 - lorsque la catégorie 1 et la catégorie 4 ne sont pas dominantes, le bloc CU est supposé être une région de texture homogène et a tendance à être codé dans une taille CU plus grande sans autre division. Il convient de noter que la plus grande taille de CU ne signifie pas une taille de CU physique, mais elle indique la taille de CU actuelle.

De plus, Le schéma de codage CU rapide se compose de deux parties : la méthode de détection du mode CU SKIP précoce et la méthode de décision fractionnée Fast CU :

- **Amélioration de la détection précoce du mode SKIP de CU basée sur Complexité du texte des blocs CU :** Pour les blocs CU à texture complexe, ils ont des valeurs de coût RD relativement plus élevées que les blocs CU à texture simple. Par conséquent, il est souhaitable d'effectuer la détection précoce du mode SKIP en tenant compte de la complexité de la texture du bloc CU actuel. C'est-à-dire que les informations de catégorie de bord des paramètres SAO peuvent être utilisées de manière utile dans la détection précoce du mode SKIP. Il convient de noter que, puisque les valeurs des paramètres SAO pour le bloc CU actuel ne sont pas disponibles, nous utilisons les valeurs des paramètres SAO du bloc colocalisé dans la trame codée la plus récente pour le bloc CU actuel.
- **Proposition d'une décision rapide de fractionnement de CU basée sur des paramètres de codage spatio-temporel** : Effectue un élagage précoce du partitionnement CU structuré en quadri-arbre. Il se compose de trois étapes opérationnelles :

(1) acquisition de données de paramètres de codage spatio-temporel : les données de catégorie de bord SAO dans une trame sont stockées en mémoire uniquement pour la décision de division CU rapide dans le codage de trame suivant.

(2) classification CU pour la complexité de la texture : Le but est d'utiliser la tendance à la division CU en fonction de la complexité de la texture des blocs CU. Cette étape se compose de deux sous-étapes : la classification grossière de CU et les sous-étapes d'affinement de la classification. La classification CU approximative consiste à décider rapidement des classes de complexité de texture pour le bloc CU actuel uniquement en utilisant les informations de catégorie de bord SAO. La sous-étape de raffinement de la classification corrige la décision imprécise pour la complexité de la texture.

(3) Décision rapide de fractionnement de CU.

Cette proposition d'encodage CU rapide global basé sur la détection combinée améliorée du mode CU SKIP précoce et la décision de division rapide de l'UC a entraîné des gains de temps d'encodage moyens de 49,6 % et 42,7 % avec des pertes de débit BD moyennes de 1,4 % et 1,0 % pour un accès aléatoire et un faible délai.

3.3 Algorithme de recherche d'arborescence inter-CU [31]

L'objectif est de modifier la décision du mode HM en fonction des caractéristiques statistiques liées aux taux de sélection des types carrés et non carrés, afin de déterminer le meilleur mode CU en comparant les coûts RD uniquement pour les types carrés, dans le cadre d'une recherche descendante. Cependant, l'utilisation de ces caractéristiques statistiques peut limiter l'applicabilité de cette approche dans des structures de codage plus complexes. L'algorithme propose une décision "square-type-first inter-CU" (SFM).

Pour les modes carrés, les unités de prédiction (PU) disponibles incluent le mode skip, la prédiction par inter, et la prédiction intra. Les modes PU non carrés, quant à eux, sont uniquement utilisés pour l'interprédiction. Les modes inter 2N x 2N et intra 2N x 2N sont considérés comme des modes PU de type carré, tandis que d'autres

modes PU, tels que inter N x N et intra N x N, sont également classés comme types carrés, mais sous certaines conditions.

Concernant les modes PU non carrés, HEVC peut utiliser des modes inter N x 2N, inter 2N x N, ainsi que quatre modes inter-AMP. La méthode proposée détermine le mode PU en deux étapes distinctes pour les types carrés et non carrés. Dans le cas des modes carrés, l'algorithme sélectionne le meilleur mode CU en comparant les coûts RD à l'aide d'une recherche descendante (top-down RD cost) dans une exploration de type profondeur d'abord (DFS).

Si une UC (Unité de Codage) n'est pas une SUC (Sous-Unité de Codage), elle est divisée en quatre sous-UC pour identifier les meilleurs modes pour ces sous-unité. Les calculs de coût RD effectués pour le CU de niveau supérieur sont également réalisés pour les sub-UC, mais uniquement pour les modes carrés. Concernant les modes non carrés, les calculs RD sont conditionnels, et se basent sur la comparaison des coûts RD du meilleur mode CU avec ceux des sub-UC correspondantes. Les calculs RD pour les modes non carrés sont ensuite ignorés. Le coût pour les modes non carrés et la somme des coûts RD des sub-UC sont ensuite comparés, et le meilleur mode est sélectionné.

L'algorithme proposé permet d'éliminer efficacement les modes PU non carrés. De plus, cet algorithme peut être accéléré grâce à des conditions de terminaison anticipée (Early Termination Conditions - ETC), permettant ainsi d'améliorer la vitesse des encodeurs HEVC utilisant la méthode de décision SFM.

Les résultats expérimentaux montrent que la méthode suggérée pour HM 13.0 offre un gain de performance (ATS) de l'ordre de 65,72 à 67,64 % par rapport à l'encodeur HM 13.0, et un gain de 33,29 à 61,46 % par rapport aux algorithmes CFM, ECU et ESD utilisés dans HM. De plus, cette méthode n'entraîne qu'une réduction de l'efficacité de la compression de 2 % ou moins en termes de BD-BR.

3.4 Technique de détection précoce des sauts [32]

La technique est Appelée détection précoce des sauts (ESD) ; est basée sur l'identification de régions immobiles et homogènes dans une séquence vidéo, en plus, un nouveau calcul basé sur la différence d'entropie qui peut prédire plus précisément le

saut de blocs de code dans une séquence vidéo naturelle. Dans cette approche, le gain de temps est associé d'une perte considérable de qualité de codage.

L'idée repose sur l'utilisation des images de référence de l'encodeur HEVC. Deux groupes d'images de référence sont sélectionnés (liste0 et liste1), puis deux vecteurs de mouvement sont calculés. Si la différence entre ces deux vecteurs est égale à 0, cela indique qu'une région est immobile, et cette région est alors prise en compte pour la détection du mode "skip".

La deuxième étape consiste à comparer les distances entre les modes "skip" et "non-skip" pour identifier les régions homogènes, en se basant sur la variance de l'unité de codage (CU) actuelle et les valeurs moyennes estimées. Si une CU satisfait à ces deux conditions, elle est considérée comme stationnaire.

La détection du mode "skip" dans les CU non stationnaires repose sur le calcul des distances en utilisant la moyenne et les valeurs actuelles de la différence absolue et de l'entropie entre les modes "skip" et "non-skip". Le mode "skip" est ensuite déterminé en fonction de la valeur de cette distance.

3.5 Algorithme de partitionnement CU rapide [33] :

Les grands blocs s'adaptent aux zones lisses, tandis que les petits blocs sont beaucoup plus utilisés avec les zones texturées. On décrit un algorithme de division CU pour la rapidité de configuration intra. Le partitionnement CU est basé sur l'analyse de la texture des régions ; le partitionnement est donc arrêté prématurément pour les régions homogènes. Cela permet de déterminer la meilleure taille de CU sans tester toutes les profondeurs. Pour Cette méthode l'appel récursif n'est maintenu que s'il y a un besoin de partitionnement. Pour une CU donnée, on commence par calculer le maximum des différences absolues (notées MAD) entre chaque pixel Pi,j (i et j sont les coordonnées du pixel P) et ses huit voisins . Ensuite, nous calculons la somme (S) de tous les MAD pour prendre une décision sur l'homogénéité de la CU à la profondeur actuelle. Si la somme S est supérieure à un seuil Thres[uiDepth], alors le bloc CU correspondant est annoté comme texturé et ainsi, le partitionnement peut se poursuivre jusqu'à la profondeur suivante. Au contraire, si la somme S est inférieure au seuil, le

partitionnement est terminé. Les CTU incomplètes (qui ne correspondent pas à une LCU complète) situées aux limites de l'image sont exclues de l'analyse mentionnée ci-dessus car elles ne suivent pas le même schéma de partitionnement récursif Une étude préliminaire pour la sélection des valeurs de seuil pour chaque profondeur a été réalisée afin d'assurer une meilleure qualité/compression compromis d'efficacité.

Dans cet algorithme, la valeur du seuil dépend de la taille du bloc considéré (Depth). Un ensemble d'expériences a été mené afin de choisir une valeur seuil qui résout le compromis entre l'efficacité de la compression et le temps d'encodage. Cet algorithme a la capacité de réduire le temps d'encodage jusqu'à 41 % en moyenne pour différentes

3.6 Algorithme de décision rapide de la taille de l'intra CU [34]

Ils ont proposé un algorithme de décision rapide du GVR (**(large gray value)** à grande valeur de gris utilisant la vérification de la plage de valeurs de gris. notre algorithme proposé peut réduire le temps de codage efficacement sans perte évidente de taux de distorsion . La complexité de calcul de 35,8% ainsi que faire 0,10 dB diminution du PSNR et augmentation de 2,33% du débit.

La première étape CU sera divisée si elle a une grande plage de valeurs de gris (significant luma differences). Puis calcule le seuil GVR qui signifie que CU ne sera pas divisé si son GVR est inférieur au seuil.

3.7 Détermination précoce de la taille de la CU [35]

Ils ont proposé une nouvelle approche de réduction de la complexité des processus de codage intra-mode. Cette méthode a défini un nouveau terme appelé le coût moyen des pixels (APC) pour une CU. L'APC pour chaque CU est défini comme le coût RD complet de la CU divisé par le nombre de pixels dans cette CU. Cette valeur est ensuite comparée à une valeur seuilδ, Si le coût total calculé de CU est inférieur à δ, on considère que le fractionnement n'apportera pas nécessairement d'amélioration du coût. Par conséquent, le processus de division est arrêté.

Cela se traduit par un gain de temps considérable le codeur avec une détérioration négligeable de la qualité et du débit. Cette méthode, a produit une réduction moyenne

de 31,3% en temps de compression en HM 10,0 avec une augmentation de 0,7% du débit et une diminution négligeable de PSNR (-0,03 dB) en utilisant un seuil APC de 20.

3.8 Arbres de décision pour la réduction de la complexité [36]

Ils ont exploré l'utilisation d'un algorithme basé sur un arbre de décision pour la partition CTU. L'essence même de cette méthode est d'utiliser un classificateur d'arbre de décision pour chaque profondeur de la partition CU. La nouvelle partie de la méthode proposée consiste à utiliser des arbres de décision «élagués manuellement» au lieu de SVM pour les critères de décision. Celles-ci sont extrêmement rapides et simples à mettre en œuvre car elles se contentent de filtrer les unités de codage en fonction de quelques valeurs de caractéristiques. Même si l'arbre de décision est l'un des algorithmes de 'machine learning' les plus simples, importer tous les critères d'un modèle DT (**Decision Trees)** formé sur plusieurs milliers d'échantillons peut devenir complexe. Par conséquent, pour plus de simplicité, l'outil a été écrit de telle sorte que deux valeurs de seuil peuvent être définies par l'utilisateur. Ces seuils supprimeraient les parties de l'arbre qui ne répondaient pas aux critères spécifiés et conserveraient les nœuds optimaux, ce qui ne laisserait que le modèle le plus simple et le plus efficace. Un seuil indique le pourcentage de précision des critères de décision, et l'autre indique le pourcentage d'échantillons classés par ce critère. Dans un scénario idéal, une solution pourrait être trouvée qui couvre un nombre élevé d'échantillons d'entrée et qui est également très précise (peu d'erreurs de classification).

Une réduction prometteuse du temps d'encodage de 42,1% en moyenne. Cependant, une sélection manuelle des seuils requis doit être appliquée par cet algorithme. Sur la base d'une analyse statistique.

3.9 Algorithme basé sur la classification par blocs arborescents [37]

Ils Ont présenté un algorithme de décision de profondeur CU rapide pour le codeur HEVC. Tout d'abord, la décision actuelle de CU est prédite à l'aide d'un ensemble de poids optimisés de décisions de CU environnantes. Ensuite, un tableau correspondant doit être trouvé à partir de l'analyse des véritables décisions actuelles possibles de CU.

Ce tableau est utilisé pour atteindre la précision de la prédiction cible. Troisièmement, l'arrêt précoce des processus de codage est obtenu à l'aide de 3 méthodes d'évaluation : vérification du coût RD, vérification de l'homogénéité du mouvement et vérification du mode SKIP. L'algorithme peut réduire le temps de codage de 57% avec le profil Low Delay et 49% coder l'heure avec le profil Random Access, tout en conservant les tailles PSNR et bitrate. Et comparé à deux algorithmes de pointe, notre algorithme améliore le gain de temps de 23 et 13% avec le profil Low Delay, et 7 et 3% avec le profil Random Access.

3.10 Prédiction efficace de la profondeur CU et du mode PU [38]

Ils ont exploré l'utilisation de trois classes de CTU comme référence pour calculer les niveaux de profondeur de CU prévus. Ces trois classes de CTU sont obtenues à partir des treize CTU voisines en utilisant la méthode des k-moyennes. Ensuite, afin de simplifier l'algorithme de sélection de PU, les modes PU rarement utilisés sont ignorés pour chaque profondeur de CU. Cependant, cette méthode nécessite d'effectuer un grand nombre de pré-calculs avant d'éliminer les PU peu utilisées, ce qui peut être un réel inconvénient pour cet algorithme, notamment dans les applications temps-réel. les résultats peut atteindre 56,71% du temps enregistrement avec profil de configuration à faible retard et 59,76% avec profil de configuration à accès aléatoire, tandis que les valeurs de BDBR (Bjontegaard Delta Bit Rate) ne sont respectivement que de 1,0517% et de 0,9918%.

3.11 Architecture VLSI d'optimisation de la distorsion [39]

Un cadre sans quantification est proposé pour réduire la complexité de l'optimisation du débit et de la distorsion. Cette approche est basée sur un modèle de taux résiduel utilisant la position et l'amplitude de coefficients de transformée non nuls. Cependant, la grande dépendance d'un modèle de débit résiduel à la résolution vidéo et aux paramètres de codage limite et réduit son applicabilité.

Dans cette référence, un algorithme rapide et une architecture VLSI de RDO dans HEVC sont présentés. La proposition de flux RDO. La quantification, la

déquantification, la transformation inverse et le CABAC sont remplacés par des estimations de débit et de détorsion pour restituer les complexités et les délais. chaque (syntax elements(SE)) est d'abord binarisé en bacs. Les bits finaux sont ensuite générés à partir des bacs bases sur les modèles de probabilité. classons les (SE) en deux groupes, respectivement pour les résidus et les en-têtes. Le débit binaire global est ainsi la somme des deux parties correspondantes. Alors que le RDO proposé vise à être sans quantification, nous dérivons une méthode qui estime directement la distorsion à partir des coefficients transformés, supprimant l'exigence des étapes Q, Q^{-1} et T^{-1}.

Pour la partie distorsion, les paramètres de mise à l'échelle sont tronqués pour réduire le coût de multiplication. En conséquence, la complexité du temps de codage d'environ 62 % peut être réduite avec une perte d'efficacité de codage de 1,85 % pour le "tout-intra". Pour l'architecture VLSI, ils ont conçu une architecture reconfigurable qui s'applique à toutes les unités de transformation.

3.12 Une approche multi-résolution' de Jha et Chaudhary [40]

Ont exploré une méthode compatible avec les standards H.264 et HEVC. En changeant les trois techniques traditionnelles de compensation de mouvement ; par une méthode basée sur le filtrage, un modèle d'ordre linéaire et un cadre multi-résolution qui permet l'estimation des paramètres.

3.13 Algorithme de codage intertrame dans HEVC [41]

Il détermine de manière adaptative la plage de profondeur de CU en utilisant une propriété de bloc d'arbre dans laquelle de petits niveaux de profondeur sont toujours sélectionnés dans des blocs dans des régions homogènes et de grands niveaux de profondeur sont sélectionnés sur des blocs d'arbre avec mouvement actif ou texture riche.

Ils ont proposé un algorithme de décision rapide de la taille de la CU pour déterminer la plage de profondeur de la CU (y compris le niveau de profondeur minimum et maximum) et ignorer certains niveaux de profondeur spécifiques rarement utilisés dans le cadre précédent et les CU voisines. De plus, l'algorithme proposé introduit également

des méthodes de terminaison précoce, basées sur la vérification de l'homogénéité du mouvement ‘Method 1: motion homogeneity checking based ET), la vérification du coût RD (Method 2: RD cost checking based ET) et la vérification du mode SKIP pour passer ME sur des tailles de CU inutiles (Method 3: SKIP mode checking based ET). Les résultats expérimentaux démontrent que l'algorithme proposé peut réduire considérablement la complexité de calcul tout en maintenant les mêmes performances de RD que le codeur HEVC original.

3.14 Décision de profondeur d'unité d'arbre de codage rapide [42]

Ils ont proposé l’algorithme de décision de profondeur quadtree CTU pour réduire la complexité de calcul de HEVC. Dans cet article, l’algorithme de résolution en profondeur quadtree CTU est rapidement réduit à la complexité de calcul de HEVC. Il étudie la corrélation entre les CTU spatiales et temporelles voisines de la profondeur. Ensuite, cet algorithme saute l’arbre quaternaire de profondeur 3 en utilisant la corrélation entre le mode de prédiction et la CTU. Les résultats expérimentaux montrent que le temps de codage moyen de 40%, tout en maintenant une performance de distorsion de fréquence comparable.

3.15 Détection d'homogénéité spatiale et temporelle [43]

Ils ont proposé un algorithme de décision de taille CU optimisé pour réduire le cout de calcul du partitionnement quad-tree au moyen d’une analyse et d’une classification d’homogénéité spatiale et temporelle, qui est directement appliquée à l’image d’entrée. De plus, cette préanalyse d’image est effectuée à l’aide d’unités logiques et de matériel embarqué sur un GPU. Évitant ainsi des états d'attente inutiles, donc le coût de calcul associé à ce processus est nul pour le processeur en charge du processus de codage. En comparaison avec le modèle de test HM16.2 de référence, le temps de codage est réduit jusqu'à 32,69%, avec une perte de qualité négligeable et une augmentation maximale du débit BD de 1,2% pour la configuration P à faible retard.

3.16 Décision rapide de taille CU basée homogénéité temporelle [44]

Un algorithme de décision de taille de CU optimisé est proposé pour réduire le cout de calcul HEVC en utilisant la classification d'homogénéité temporelle qui est directement appliquée à l'image d'entrée à l'aide d'un GPU.

3.17 Informations de bord et codage spatio-temporel [45]

Ils ont proposé un schéma de codage. inter-image rapide utilisant des informations provenant de paramètres de codage spatiotemporel, pour réduire la complexité du codeur HEVC. Il consiste en une méthode de décision rapide de tous les modes

2 N × 2 N : une méthode de décision rapide en profondeur CU et une méthode de décision rapide en mode PU. Ce schéma utilise l'information de bord pour exprimer la complexité de la structure et utilise la différence d'information de bord entre CU actuelle et ses CU spatiotemporelles pour exprimer la similarité de bord (ES) dans une trame et le mouvement de bord (EM) entre deux trames adjacentes. Et puis, utilise ES et EM comme paramètres d'assistant coopèrent avec les niveaux de profondeur CU et les coûts RD du mode PU des CU spatiotemporels pour accomplir la fin précoce de la division CU et la sélection du mode PU. Les résultats expérimentaux montrent que la méthode de codage inter-trame rapide proposée peut réduire considérablement les coûts de calcul avec une perte RD négligeable. Il y a en moyenne 53,7 et 54,9% de gain en temps de codage, mais uniquement avec des pertes de débit binaire différentiel (BDBR) de 1,5 et 1,8% en moyenne pour différentes séquences d'essai en accès aléatoire et en faible retard.

3.18 Corrélation statistique et spatio-temporelle pour HEVC [46]

Il a décrit un algorithme de faible complexité pour HEVC avec deux approches, Détermination précoce du mode SKIP (EDSM(Early Determination of SKIP Mode)) et Finalisation anticipée du processus de sélection du cadre de référence (ETRFS(Early Termination of Reference Frame Selection)) pour réduire la complexité de codage de HEVC. Ceci, en exploitant la corrélation spatiotemporelle et la corrélation statistique. Néanmoins, l'efficacité de cette approche dépend directement de la précision de la

prédiction et de la sélection des caractéristiques. Une faible précision entraine une perte significative des performances RD basé sur l'apprentissage automatique, l'algorithme proposé peut réduire le temps de codage de 45,01% en moyenne dans un contexte RA et de 46,16% en moyenne dans un contexte LD, tout en conservant une perte négligeable d'efficacité de codage.

3.19 Décision bayésienne et l'apprentissage en ligne [47]

A a développé un algorithme CU de terminaison précoce basé sur une règle de décision bayésienne. Dans ce cadre, cette règle est utilisée pour estimer une fonction de vraisemblance et déterminer la probabilité d'une nouvelle scène. Afin de prédire la taille de la CU, le modèle doit être mis à jour pour chaque nouvelle image. L'algorithme proposé permet une réduction notable de 36 % du temps de codage. Toutefois, sa précision varie en fonction de la scène traitée, ce qui impacte la performance globale. Les résultats montrent qu'en configuration All Intra, l'algorithme offre une réduction de 36,69 % du temps d'encodage, tout en augmentant de 1,08 % le débit delta Bjøntegaard binaire (BDBR). Lorsqu'il est intégré à l'approche SCC rapide existante, l'algorithme réduit le temps d'encodage de 48,83 % et améliore le BDBR de 1,78 %. Les algorithmes CFM, ECU et ESD utilisés dans HM révèlent que la méthode proposée ne réduit l'efficacité de compression que de 2 % ou moins en termes de BD-BR. Néanmoins, pour intégrer cet algorithme dans un codeur vidéo en temps réel, des travaux supplémentaires sont nécessaires.

3.20 Apprentissage automatique pour le codage intra HEVC' [48]

Un algorithme rapide a été proposé pour optimiser le processus de division des unités de codage (CU) dans le codage intra-HEVC. Selon les auteurs, l'utilisation de l'analyse discriminante linéaire de Fisher (FLDA) et du classificateur des k plus proches voisins (k-NN) permet de déterminer rapidement la taille des CU. Cet algorithme réduit la complexité de calcul du codeur HEVC en s'appuyant sur une analyse statistique de la complexité de l'image et des informations de codage. La mesure de complexité, fondée sur la magnitude des bords (EM) calculée à l'aide de l'opérateur de Sobel, est utilisée

pour effectuer une séparation anticipée et une décision AOLP. Dans cette étude, FLDA et le classificateur k-NN sont combinés.

L'EM, basé sur l'opérateur Sobel, ainsi que les coûts RD (SATD J et full J) obtenus lors du codage HEVC, servent de caractéristiques pour une classification binaire. FLDA transforme une fonction 2D en une fonction 1D plus facilement séparable en utilisant la direction de projection optimale (w).

L'algorithme proposé calcule la probabilité directement à l'aide d'une méthode non paramétrique, le k-NN, et détermine la classe appropriée de la CU actuelle en utilisant les voisins transformés par FLDA. Les résultats expérimentaux montrent que cet algorithme permet de réduire le temps de codage de 54,0 % avec une dégradation négligeable de l'efficacité de codage (0,68 %) par rapport à la version HM 16.15 dans la configuration AI.

3.21 Un algorithme de partition CU rapide [49]

Ils proposent un algorithme de codage intrarapide composé de deux parties : une prédiction de la plage de profondeur CTU et deux règles de décision bayésiennes sont appliquées pour accélérer le processus de partition CU.

La complexité du codage intra HEVC est assez élevée, car l'encodeur doit parcourir les 35 modes de prédiction et une vérification itérative est appliquée pour tous les niveaux de profondeur CU possibles. Par conséquent, l'accélération du processus de codage intra devrait se concentrer sur la réduction de l'ensemble de candidats de mode et sur le saut de certaines vérifications de profondeur de CU redondantes. Compte tenu de cette motivation, un algorithme de codage intra rapide est proposé dans cet article. Le but de l'algorithme proposé est de réduire la complexité de calcul dans la décision de taille de CU et la décision de mode intra.

Dans l'algorithme proposé, la plage de profondeur peut être divisée en trois catégories : $R1 = [0, 1]$, $R2 = [0, 2]$, $R3 = [1, 3]$, qui correspondent à une région très homogène, homogène et inhomogène , respectivement. Et puisque les blocs voisins contiennent généralement des textures similaires dans les images naturelles, la plage de profondeur de la CTU actuelle peut être prédite par les CTU gauche et supérieure.

En outre, deux règles de décision bayésiennes à taux d'erreur minimum sont appliquées pour ignorer davantage certaines vérifications de profondeur de CU redondantes, afin d'optimiser le processus de partition CTU. Pour la partie de décision intra-mode rapide, un ensemble de modes de prédiction adaptative basé sur les informations de profondeur est généré pour simplifier le processus RMD, moins de modes sont évalués au lieu d'une traversée pour tous les modes de prédiction intra. Et pendant ce temps, le processus MPM est ajusté pour ignorer davantage certaines vérifications de mode inutiles. Aussi, un rapport directionnel est calculé pour estimer l'orientation approximative de chaque CU, ainsi les modes angulaires non pertinents peuvent être davantage éliminés. Les résultats de l'expérience montrent que l'algorithme proposé peut réaliser un gain de temps de 52,9 % en moyenne tout en ne causant qu'un taux de BD de 1,28 %. Augmenter par rapport aux séquences de test standard HEVC. Aussi, une simulation étendue est réalisée sur des séquences en ultra haute définition et rapporte 51,88% gain de temps avec une perte de qualité de codage négligeable. Pendant ce temps, par rapport à d'autres œuvres relatives,

3.22 Conclusion

Malgré les résultats acceptables obtenus dans ce travail connexe, la majorité de ces techniques souffrent toujours soit d'une complexité de calcul, soit d'une augmentation des débits binaires résultants, soit d'une réduction de la qualité des vidéos reconstruites en termes de PSNR et SSIM. La raison principale de la complexité de calcul dans la plupart des cas est due au grand nombre de pré-calculs effectués avant le codage réel. Pour cette raison, cette étude de recherche propose une méthode adaptative de décision de taille Fast CU, le coût de calcul du partitionnement quad-tree basé sur l'analyse de l'homogénéité spatiale. La spécificité de la méthode proposée est la classification des UC en plusieurs classes selon leurs complexités de texture. Cela permet une décision de taille de CU rapide et donc une réduction du coût de calcul HEVC.

4 .

CHAPITRE 4 : REDUCTION DE LA COMPLEXITE DU CODEUR HEVC PAR HOMOGENEITE SPATIAL

4.1 Introduction

La réduction de la complexité du codeur HEVC/H265 vise à améliorer les performances de compression en intégrant une forme d'intelligence au sein du codeur.

Cette approche introduit une nouvelle méthode de décision rapide pour la taille des CU (unités de codage) dans HEVC. Elle repose sur des mécanismes de décision rapide tout au long du processus d'encodage des CU. L'algorithme analyse les caractéristiques pertinentes et l'homogénéité spatiale, permettant ainsi de réduire le coût de calcul associé au partitionnement en arbre quadratique (quad-tree).

L'organisation de ce chapitre est la suivante : dans la première partie, intitulée L'algorithme de décision rapide pour la taille de CU, sont présentées les notions d'homogénéité spatiale et de détection améliorée de l'homogénéité spatiale. La deuxième partie présente des résultats expérimentaux visant à évaluer les performances de la stratégie proposée, en utilisant des séquences vidéo CIF/QCIF ainsi que des séquences des classes A, B, C et E dans le cadre des tests JCT-VC. Enfin, le chapitre se termine par une conclusion.

4.2 L'algorithme de La décision rapide proposée

Les recherches actuelles sur l'optimisation de la taille des unités de codage (CU) se concentrent sur le développement d'un algorithme visant à détecter des régions spatiales homogènes afin d'accélérer le processus de codage HEVC. Le nouvel algorithme proposé peut être résumé de la manière suivante :

1. La détection de l'homogénéité spatiale repose sur le SMCV (Simplified Mass Center Vector) pour le codage HEVC des trames d'entrée. Il est important de préciser qu'en premier lieu, nous menons une étude statistique, qui avait été initialement conçue pour le codec H.264. Cette étude, réalisée à partir de diverses séquences vidéo ayant des caractéristiques temporelles et spatiales distinctes, et pour différentes valeurs de QP, devrait permettre de définir quatre seuils.

2. Amélioration de la détection de l'homogénéité spatiale basée sur la structure des données dans le modèle HM, en utilisant les trames d'entrée et la création d'index pour une CU homogène. Cela se fait en comparant le SMCV de chaque CU des images d'entrée avec les différents seuils calculés précédemment.
3. Prise de décisions rapides sur la taille intra-CU pour les régions spatialement homogènes et l'index créé.

Tout d'abord, la direction vecteur du centre de masse du bloc est utilisée pour déterminer la direction de corrélation du bloc dans HEVC. Ensuite, la création d'index pour des CU homogènes est présentée, suivie de l'optimisation du processus de décision de mode. Nous avons appliqué et testé cette nouvelle approche sur le modèle de test HEVC HM16.13 [50]. Ce travail vise principalement à réduire le coût de calcul associé à l'encodeur HEVC en s'appuyant sur l'homogénéité spatiale des images et la forte corrélation entre les images voisines.

4.2.1 Détection de l'homogénéité spatiale d'une CU

L'homogénéité est une information locale et correspond au caractère uniforme d'une région. En effet, une région qui a des textures avec une propriété spatiale similaire a tendance à être homogène. Pour déterminer des régions homogènes dans une image, de nombreuses techniques ont été proposées. Dans [51] et[52] la direction du vecteur de centre de gravité de bloc est utilisée pour déterminer la direction de corrélation de bloc dans H.264/AVC. Ceci est plus adapté à une mise en œuvre dans un algorithme de décision en mode rapide pour l'intra-prédiction dans ce codage vidéo. Dans le codage Intra de H.264/AVC, les propriétés locales des trames sont estimées en utilisant des tailles de bloc de 4 × 4, 8 × 8 et 16 × 16. Les auteurs dans [52] proposent de déterminer la complexité de la texture à partir simplement de l'amplitude du vecteur central de masse simplifié (SMCV) de chaque bloc MB de 16 × 16. Tandis que dans [53] et [54] les auteurs proposent de sélectionner l'amplitude SMCV ainsi que la variance comme caractéristiques extraites de chaque CU 64×64, mais cette fois pour l'intra-codage 3D-HEVC. Contrairement au H264 basé sur des macroblocs de structure de tailles 4×4, 8×8 et 16×16, le HEVC utilise des UC de tailles 4×4, 8×8, 16×16, 32×32 et 64×64 blocs.

Par conséquent, dans notre travail, nous adaptons l'algorithme décrit dans [51], pour estimer la complexité de la texture de chaque bloc d'entrée 64 × 64 et ainsi sélectionner la taille de CU la plus appropriée. Comme dans [52-54] , nous commençons par une étude statistique sur de nombreuses séquences vidéo différentes. Cette étude statistique concerne principalement la relation entre l'Amplitude SMCV (AMP_{SMCVCU}) du CU et les meilleures tailles intra bloc pour ces différentes séquences. Le SMCV est ensuite calculé pour chaque bloc CU N×N (où N est 64, 32, 16, 8 ou 4) de coordonnées (i,j) en utilisant (4.1). Soit I(i,j) la valeur de luminance du pixel en (i,j) dans N×N CU ;

l'AMP_{SMCVCU}, défini par l'Eq. (4.4), est basé sur deux coordonnées dx et dy, représentant respectivement le degré de différence dans les directions verticale et horizontale, comme indiqué dans les équations (2) et (3).

AMP_{SMCVCU} est utilisé pour déterminer la complexité de texture de CU. En effet, lorsque cette amplitude est plus élevée, le CU est moins homogène. Bien évidemment, il est possible de proposer plusieurs niveaux de complexité de l'CU étudiée en fonction de la valeur de $AM_{PSMCVCU}$:

$$SMCV\ \vec{V} = \{dx_{i,j}, dy_{i,j}\} \tag{4.1}$$

Où $d_{xi,j}$ et $d_{yi,j}$ représentent les degrés de différence dans les directions verticale et horizontale, donnés respectivement par les équations (4.3) et (4.4).

$$dx_{i,j} = \sum_{y=0}^{\frac{N}{2}-1} \sum_{x=0}^{N-1} \left[\left(\frac{N}{2} - y \right) . \left| I_{(x+x_0, y+y_0)} - I_{(x+x_0, N+y_0-y-1)} \right| \right] \tag{4.2}$$

(x_0, y_0) Est les coordonnées de pixel à gauche d'une CU et $I_{x,\,y}$ représente l'intensité des pixels de la CU.

$$dy_{i,j} = \sum_{x=0}^{\frac{N}{2}-1} \sum_{y=0}^{N-1} \left[\left(\frac{N}{2} - x \right) . \left| I_{(x+x_0, y+y_0)} - I_{(N+x_0-x-1, y+y_0)} \right| \right] \tag{4.3}$$

Ceci conduit finalement à l'expression de $AMP_{SMCVCU}(\vec{V})$:

$$AMP_{SMCVCU}(\vec{V}_{i,j}) = (dx_{i,j} + dy_{j,i}) \tag{4.4}$$

L'étude statistique réalisée sur différentes séquences vidéo, couvrant les quatre angles du plan TI/SI et pour plusieurs valeurs de QP, permet de déterminer quatre seuils Th_1, Th_2, Th_3 et Th_4. L'étude statistique de l'estimation de la complexité de la texture des CU

proposées est similaire à un système de reconnaissance à partir d'un ensemble étiqueté de données d'apprentissage. Cette étude statistique réalisée fonctionne comme suit :

- Pour toutes les CU des différentes séquences choisies, nous obtenons le SMCV valeur et le mode de taille intra codé, avec (64, 32, 16, 8 ou 4).
- On calcule le nombre d'CU codées avec 64, 32, 16, 8 et aussi pour 4 selon les valeurs SMCV
- Les seuils Th_1, Th_2, Th_3 et Th_4 sont des valeurs SMCV correspondant à au moins 90% du plus grand pourcentage de chaque CU des CU codées de tailles 4×4, 8×8, 16×16, 32×32 et 64× 64, respectivement.

Ainsi, nous avons observé que les meilleures performances sont atteintes lorsque les valeurs de ces seuils sont respectivement égales à 650, 7000, 32000 et 54000. L'étude statistique étant terminée, nous pouvons maintenant procéder à une décision rapide quant à la taille de l'CU à prendre en compte, pour toute séquence vidéo, selon la complexité de sa texture. Ainsi, nous pouvons classer ces CU en cinq classes de complexité de texture comme le montre la Figure 4.1 :

- ✓ Complexité de texture très faible lorsque $AMP_{SMCVCU} \leq Th_1$ correspondant à une CU (64×64) homogène,
- ✓ Complexité de texture faible lorsque $Th1 < AMP_{SMCVCU} \leq Th_2$ correspondant à une CU (32×32) homogène,
- ✓ Complexité de texture faible/élevée lorsque $Th_2 < AMP_{SMCVCU} \leq Th_3$ correspondant à une CU (16×16) homogène,
- ✓ Complexité de texture élevée lorsque $Th_3 < AMP_{SMCVCU} \leq Th_4$ correspondant à un CU (8×8) homogène,
- ✓ Complexité de texture très élevée lorsque $AMP_{SMCVCU} > Th4$ correspondant à un CU (4×4) homogène.

Dans la Figure 4.1, que l'on peut considérer comme une sorte d'histogrammes pour les différentes tailles d'CU (4×4, 8×8, 16×16, 32×32 et 64×64), l'abscisse représente le nombre de CU et l'ordonnée représente l'amplitude du SMCV. Comme nous avons

différentes tailles d'CU, nous les avons représentées par des couleurs différentes. On remarque clairement que les valeurs d'amplitude SMCV (AMP_{SMCVCU}) du $CU_{4\times4}$ et un peu moins ceux de $CU_{16\times16}$ sont les plus élevés.

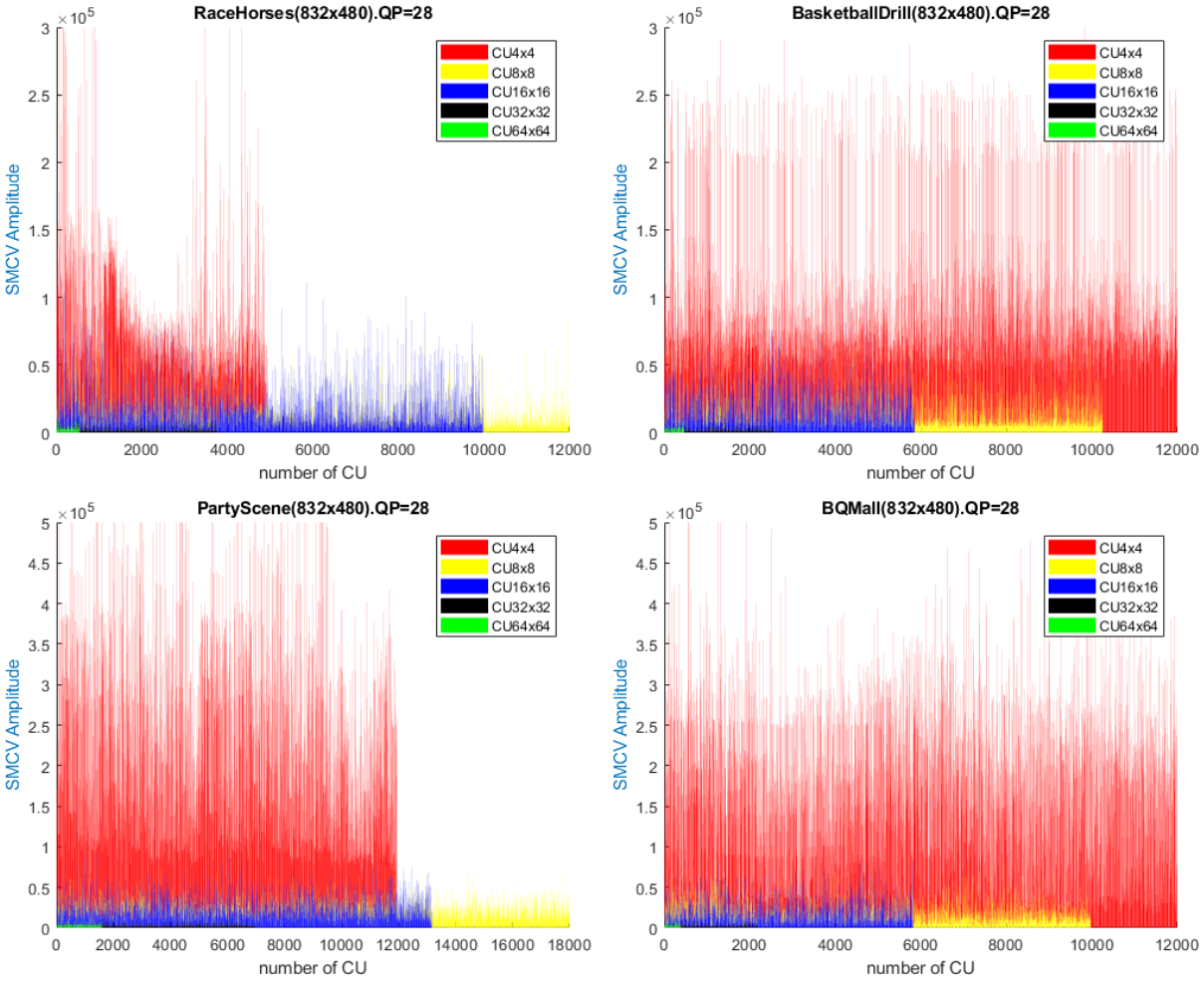


Figure.4. 1 : *Valeur SMCV en fonction du nombre d'CU par taille dans les séquences vidéo de classe C pour la valeur QP 28*

L'algorithme de décision proposé pour la taille de CU rapide est illustré à la figure4.5.premierement 0n calcule L'amplitude vectorielle SMCV (AMP_{SMCVCU}) de l'entrée CU. Si CU a une complexité de texture très faible ($AMP_{SMCVCU} \leq$ Th1), alors CU (64×64) homogène est sélectionnée ($CU_{64\times64}$ homogène représentée en blanc à gauche de la figure 4.2) ; Sinon, si une CU a une complexité faible/élevée ($Th_2 <$ AMPSMCVCU $\leq Th_3$), CU (16 × 16) homogène est sélectionné ($CU_{16\times16}$ homogène

représenté en blanc à gauche du CU $_{32 \times 32}$ de la Figure 4. 2) ; sinon, si une CU a une complexité élevée ($Th_3 < AMP_{SMCVCU} \leq Th_4$), CU (8 × 8) homogène est sélectionné

(CU $_{8 \times 8}$ homogène représenté en blanc juste en dessous de CU 16 × 16 de la Figure 4.2) ; sinon, un CU a une texture très élevée complexité ($AMP_{SMCVCU} > Th4$), CU (4 × 4) homogène est sélectionné

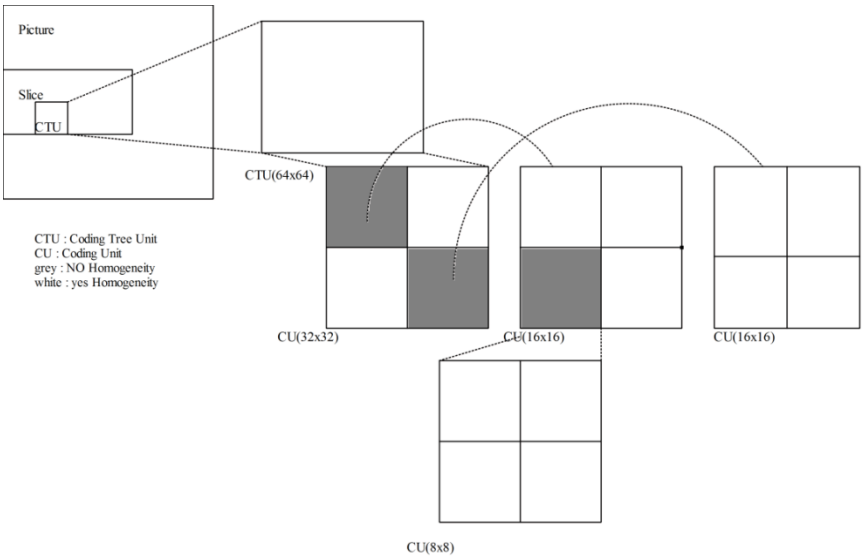


Figure 4.2 : *Décision du mode CU basée sur l'amplitude du centre de masse..*

4.3 Détection améliorée de l'homogénéité spatiale

Pour réduire davantage la complexité de l'encodeur HEVC, on tient compte de l'homogénéité spatiale pour chaque trame courante tout en tenant compte de la trame précédente. Pour cela, nous adoptons la structure des données et la méthode d'accès utilisées dans le logiciel de référence HEVC. En effet, dans le logiciel de référence HEVC, toutes les informations en CU sont accessibles par index (voir figure 4.3). De manière simple, chaque index est défini comme étant le numéro d'unité de stockage au sein de la CU. Pour pouvoir bénéficier de cette structure dans notre méthode alors un indice sera attribué à l'homogénéité spatiale estimée, à partir du SVMC.

- LCU: largest CU

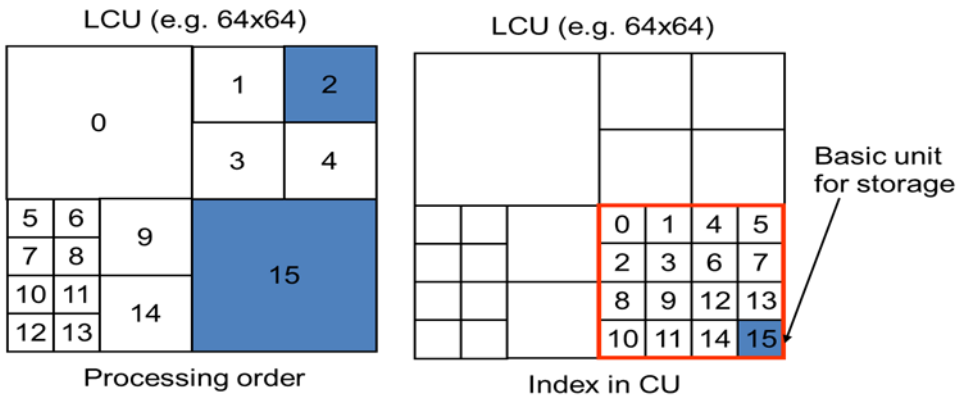


Figure 4.3 : *Structure des données dans HEVC.*

La décision de créer un index CU homogène est basée sur l'observation dans les parties temporelles et spatiales. Les caractéristiques de mouvement et de texture de ces images sont similaires. Par conséquent, nous pouvons prédire la profondeur de la CU potentielle en vérifiant la taille de ses préréglages de CU. La Figure 4.4 montre comment créer un index. Donc un CU de la trame précédente qui a été détectée homogène, enregistré sur un index, conduira à une décision rapide sur l'homogénéité de CU qui lui correspond, dans la trame courante. Sinon, il faudra appliquer le même principe décrit ci-dessus sur le CU de la trame courante. Cependant, compte tenu du grand nombre de CU de grandes tailles présentes dans les différentes trames des séquences vidéo, selon l'étude statistique de la figure.4.1, ainsi que la forte corrélation entre les trames voisines, cela doit conduire à une réduction significative de la complexité de codage.

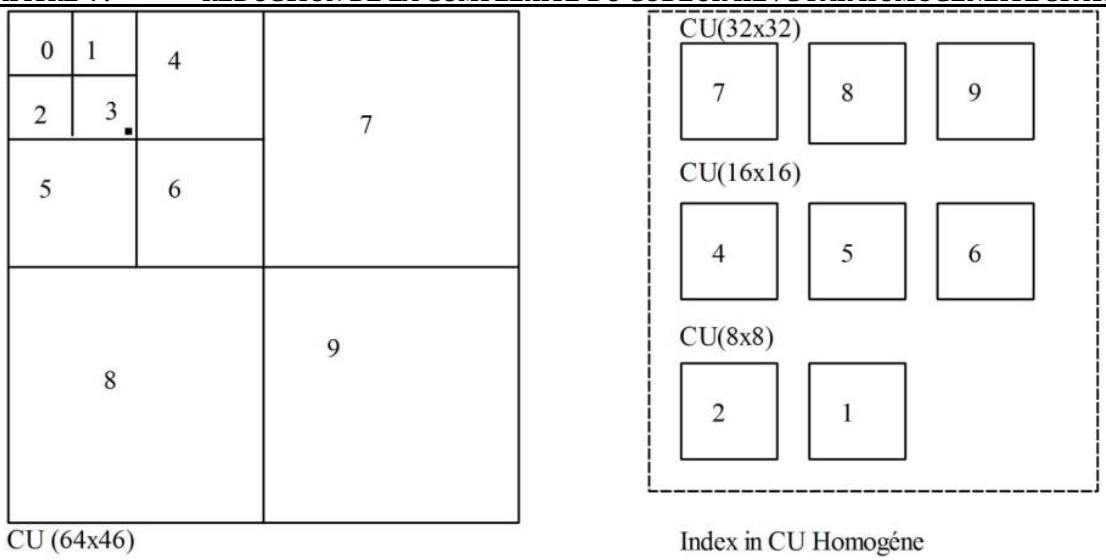


Figure 4.4 : *Création d'un index.*

Dans cette étude, la profondeur du candidat CU peut être prédite en vérifiant les préréglages CU. La figure 4.5 montre la relation entre les préréglages de CU dans le préréglage de trame et la CU codée actuelle dans notre trame. . Les préréglages CU signifient que la trame CU précédente a la même position que la CU actuelle encodée. L'algorithme proposé s'exécute de manière récursive pour des CU égales à 64 × 64, 32×32, 16 × 16, 8 × 8 ou 4 × 4.

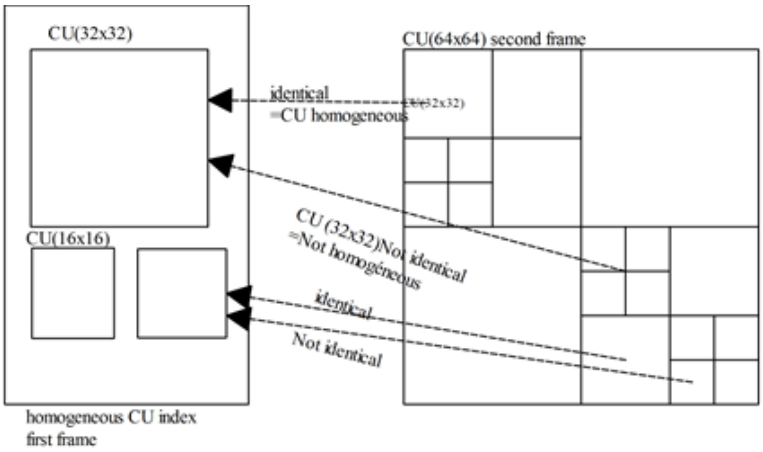


Figure 4.5 : *Décision du mode CU basée sur l'amplitude du centre de masse*.

L'algorithme de décision complet de la taille de CU rapide est présenté ci-dessous

Algorithm: Décision rapide de la taille CU

1: Procedure fast CU size (video)
2: For all frames ∈video sequence do
3: if First frame then
4: Initialize CU data
5: Calculation of the degrees of difference in vertical directions
6: Calculation of the degrees of difference in horizontal directions
7: SMCV vector amplitude (*AMPsmcvcu)* estimation.
8: $AMPsmcvcu \leftarrow AMP_{smvc}$;
9: if $AMPsmcvcu \leq TH_H_1$ then
10: Classify CU (64 × 64) homogeneous in an index
11: // Structural partitioning decision
12: else if ($Th_H_1 \leq AMP_{smcvcu}$) and ($AMP_{smcvcu} \leq Th_H_2$) then
13: Classify CU (32×32) homogeneous in an index
14: else if ($Th_H_2 \leq AMP_{smcvcu}$) and ($AMP_{smcvcu} \leq Th_H_3$) then
15: Classify the CU (16 × 16) homogeneous in an index
16: else if($Th_H_3 \leq AMP_{smcvcu}$) and ($AMPsmcvcu \leq Th_H_4$) then
17: Classify the CU (8 × 8) homogeneous in an index
18: else
19: Classify the CU (4 × 4) homogeneous in an index
20: end if
21: else other frame
22: If CU of subsequent frames == CU homogeneous index then
23: Classify the CU homogeneous in an index
24:else
25: back the *AMPsmvc* amplitude calculation step
26: end if
27: end if
28: end for
29: end procedure

4.4 Résultats et discussion

4.4.1 Les Mesures de test

Dans cette section, nous évaluons les performances de l'algorithme en termes d'augmentation du débit binaire, de rapport signal sur bruit de crête (PSNR) et de réduction de la complexité. Tous les algorithmes utilisés sont implémentés dans le modèle de test HM16.13 [50] et réalisée à l'aide d'un processeur Intel (R) Core (TM) 2 Duo 3,4 GHz avec 4,00 Go de RAM et Microsoft Windows 7 comme système d'exploitation .

Trois mesures sont définies par rapport au codeur H.265 du logiciel de référence HM16.13 :

Le taux de gain de temps d'encodage [55] : ΔTS (%) défini par :

$$\Delta Time(\%) = \frac{ProposalTime - ReferenceTime}{ReferenceTime} \times 100 \quad (4.5)$$

Soit la variation de débit [55] ΔBR (%) donnée :

$$\Delta BR(\%) = \frac{Proposal(bytes) - Reference(bytes)}{Reference(bytes)} \times 100 \quad (4.6)$$

Le ΔPSNR [55] la variation de la qualité vidéo est définie par :

$$\Delta PSNR(dB) = PSNRproposed - PSNR_{\text{Ref}} \quad (4.7)$$

Le moyen PSNR[23] pour une séquence de trames de couleur est donné par :

$$PSNR = \frac{6.PSNR_Y + PSNR_U + PSNR_V}{8} \quad (4.8)$$

Où Reference Time et Reference (bytes) sont des valeurs de la séquence décodée par le logiciel HM16.13.

4.4.2 Les Conditions de test pour le codage HEVC

Modes de configuration de l'encodeur : L'encodeur logiciel de référence HM16.13 [50] a trois configurations principales. Ces configurations [56], [57]incluent : le mode tout intra (AI), le mode d'accès aléatoire (RA) et un mode à faible retard (LDB).

Chaque image des séquences vidéo de test dans cette expérience est codée dans les trois configurations. Tout intra (AI), Accès aléatoire (RA), Faible retard LD **(** Low delay**)** .Le tableau 4.1 montre les paramètres des : encoder_randomaccess_main.cfg, encoder_intra_main.cfg et encoder_lowdelay_main.cfg.

Tableau 4.1: *Le fichier de configuration d'HM 16.13*

	Encoder RandomAccess_main	Encoder Intra_main	Encoder Lowdelay_main
Item	**Value**	**Value**	**Value**
MaxCUWidth	64	64	64
MaxCUHeight	64	64	64
MaxPartitionDepth	4	4	4
QuadtreeTULog2MaxSize	5	5	5
QuadtreeTULog2MinSize	2	2	2
QuadtreeTUMaxDepthInter	3	3	3
QuadtreeTUMaxDepthIntra	3	3	3
IntraPeriodDecodingRefres	32	1	-1
hType	1	0	0
GOPSize	16	1	4
IntraQPOffset	-3		-1
RDOQ	1	1	1
LoopFilterDisable	0	0	0
InternalBitDepth	8	8	8
SAO	1	1	1
QP	17, 22, 27, 32, 37,42	17, 22, 27, 32, 37,42	17, 22, 27, 32, 37,42

4.4.3 Les Séquences video de test CIF et QCIF

Pour les tests d'évaluation subjective HM16.13, nous avons travaillé sur douze séquences vidéo CIF et QCIF [58] et avec YUV, sous-échantillonnage de chrominance 4:2:0, où le tableau 2 nous donne les séquences vidéo utilisées. En fonction de l'information spatiale (SI) et de l'information temporelle (TI) [59] ces séquences ont été choisies comme illustré sur la Figure 4.6.

Tableau 4.2 : *Les séquences vidéo originales incluent la résolution et les images.*

	Video sequence	**Resolution (Pixel)**	**Frames**
CIF	Akiyo	352×288	300
	Forman	352×288	300
	Basketframe	416×240	200
	Mobile	352×288	300
	Stefan	352×288	90
	Waterfull	352×288	260
Q-CIF	News	176x144	300
	bridge-close	176×144	300
	carphone	176×144	382
	grandma	176×144	870
	miss-america	176×144	150
	Hall	176×144	200

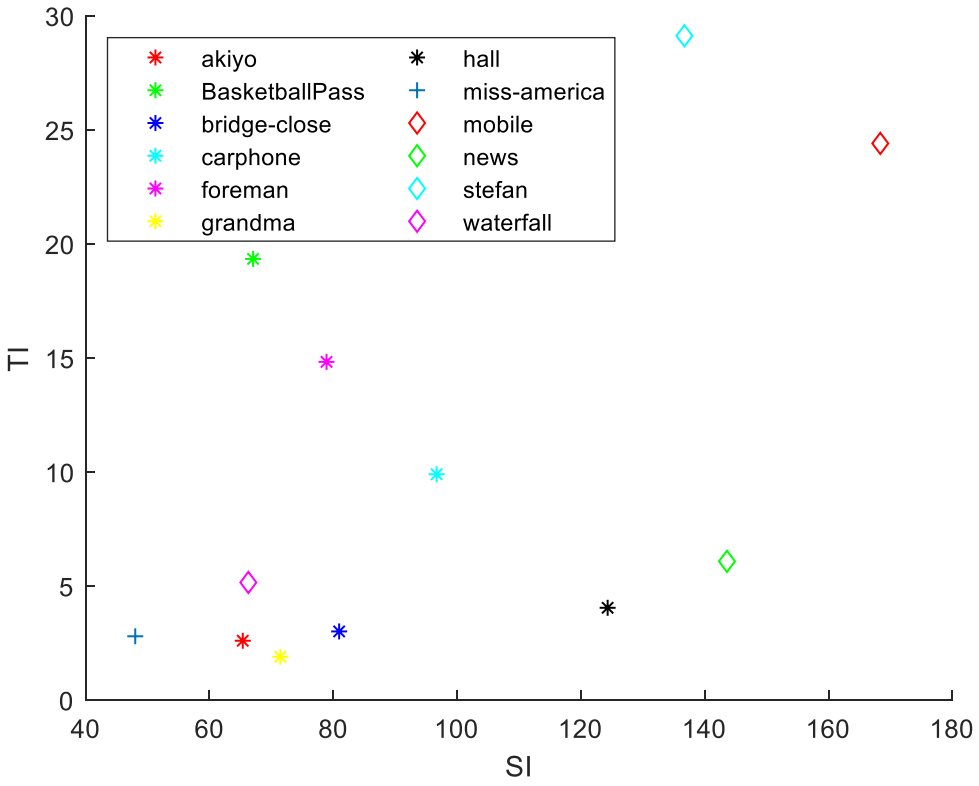


Figure 4.6 : *Représentation spatio-temporelle de la séquence vidéo.*

Pour vérifier cette méthode [60], les résultats obtenus à l'aide du HM modifié sont comparés aux résultats donnés par le HM d'origine, pour différentes séquences vidéo QCIF et CIF [36], les paramètres QP variaient de 17 à 42 avec un pas de 5 (valeurs QP : 17, 22, 27, 32, 37, 42). De ces comparaisons, on peut déduire que la méthode proposée conduit, pour toutes les séquences vidéo utilisées, à une très légère réduction moyenne du PSNR (ΔPSNR) entre -0,115 à -0,081 dB et une petite augmentation des débits de seulement 0,296% et 0,160%. En revanche, un gain significatif en temps de codage est obtenu. Par exemple, pour QP=28, les résultats de mesure globaux sont résumés dans le tableau 3, montrant ΔBR (débit binaire Δ) (%) pour tout le mode intra (AI), le mode d'accès aléatoire (RA) et le mode à faible retard (LD).

BD-BR (Bjontegaard Delta Bitrate), proposé en 2001, reste l'un des outils les plus largement utilisés pour calculer et comparer l'efficacité de compression de deux ou plusieurs codecs vidéo. La métrique Bjontegaard permet la comparaison des courbes RD en termes d'amélioration PSNR moyenne ou d'économies moyennes de débit binaire en pourcentage. A cet effet, les calculs de valeurs de taux Bjontegaard Delta (BD) que nous avons effectués, sont indiqués dans la valeur BDBR, y compris à la fois BD-PSNR et BD-Bit-rate (voir les tableaux 4.3 et4. 4).

Dans le tableau 4.3, les valeurs négatives et positives présentent respectivement les données décroissantes et croissantes. Tous les résultats expérimentaux sont obtenus pour les HEVC modifiés par rapport à ceux des HEVC d'origine. Il est évident de ce tableau, que le HEVC proposé atteint un taux moyen de gain de temps d'encodage pour toutes les séquences de test utilisées, de 41,78 %, 40,60 % et 39,82 % pour le mode d'accès aléatoire, tout mode intra et mode faible retard respectivement. Il y a une augmentation du BDBR de 1,53 % pour RA-Main, de 1,64 % pour tout le mode intra et de 1,73 % pour le mode à faible retard.

Tableau 4.3 : *Efficacité de codage HM16.13 sous les configurations RA, LD et Intra*

configuration	**Random Access (RA)**				**All intra (AI) modes**				**low delay (LD)**			
Séquences	**ΔTime**	**ΔBits**	**ΔPSNR**	**BDBR (%)**	**ΔTime**	**ΔBits**	**ΔPSNR**	**BDBR (%)**	**ΔTime**	**ΔBits**	**ΔPSNR**	**BDBR (%)**
akiyo	-40,01	0,009	-0,088	0,6	-39,1	0,1	-0,195	0,4	-41,32	0,008	-0,095	1,3
Forman	-42,2	-0,3	-0,175	0,5	-40,3	-0,2	-0,23	0,8	-42,1	-0,32	-0,19	2
Basketframe	-30,6	0,36	-0,085	1,3	-33,05	0,5	-0,29	0,3	-36,3	0,38	-0,12	0,8
mobile	-39,2	0,80	-0,101	1,7	-39,5	0,92	-0,2	2,5	-35	0,89	-0,23	2,1
stefan	-23,53	0,56	-0,09	0,9	-22,36	0,6	-0,14	0,7	-20,66	0,59	-0,2	1,8
waterfull	-43,12	-0,66	-0,1	2,1	-42,91	-0,56	-0,3	1,8	-41,95	-0,59	-0,14	0,1
foreman	-55,98	0,32	-0,004	1,5	-54,1	0,59	-0,09	1,9	-50	0,33	-0,021	3,1
bridge-close	-60,69	0,75	-0,36	2,2	-59,62	0,79	-0,46	3,4	-58	0,69	-0,39	1,2
carphone	-43,33	-0,09	-0,15	0,7	-39,25	-0,06	-0,2	1,3	-42,1	-0,09	-0,3	2
grandma	-45,02	-0,14	-0,33	0,1	-43	-0,05	-0,34	1,1	-41,01	-0,23	-0,63	0,5
carphone	-38,32	0,81	-0,9	3,2	-37	0,91	-0,87	2,8	-37,36	0,29	-0,69	1?4
hall	-39,33	-0,63	-0,15	3,6	-36,99	-0,42	-0,19	2,7	-32,09	-0.65	-0.14	4,1
Average	-41,78	0,15	-0,21	1,53	-40,60	0,26	-0,29	1,64	-39,82	0,18	-0,27	1,73

4.4.4 Teste des Séquences des classes A, B, C et E

L'équipe conjointe de collaboration sur le codage vidéo (JCT-VC) a classé les séquences vidéo de test en six classes de A à F. Chacune de ces différentes classes a sa propre résolution et est utilisée pour des applications cibles typiques[20] . La classe C en particulier est destinée à plusieurs applications comme les applications mobiles et VANET comme la sécurité routière et la vidéosurveillance. Souvent ce type d'application nécessite un délai de bout en bout strictement faible. En effet, le délai entre la capture et l'affichage de la vidéo ne doit pas dépasser quelques dixièmes de seconde pour la sécurité routière et quelques secondes pour la vidéosurveillance de classe A, B, C et E. A cet effet, les résolutions des séquences vidéo requises dans de telles applications sont relativement faibles. De plus, le temps d'encodage et de décodage doit être minimal. Dans cette section, nous comparons certaines des méthodes les plus avancées avec celle proposée. Nous utilisons la variation du débit binaire ΔBR [55] et

du taux de gain de temps d'encodage[55] ΔTS appliqués aux séquences vidéo de classe C dans les conditions de test JCT-VC [18] (voir tableau 4.4). Les informations spatiales (SI) et les informations temporelles (TI)[59] pour les séquences JCT-VC Classe A, B, C et E sont illustrées sur la Figure4.7. Comme nous pouvons le constater, les séquences de classe C sur lesquelles nous avons testé notre approche couvrent relativement le plan TI/SI ce qui permet une meilleure validation des résultats obtenus.

Table 4.4 : *Les séquences de vidéos de **C**lass-A-B-C-E*

Class	Sequence Name	Resolution	Frame	Typical Applications
A	Traffic	2560 × 1600	30	To evaluate 4K/8K video encoding performance or in works that aims to reduce the computing time.
	PeopleOnStreet	2560 × 1600	30	
B	Kimono	1920 × 1080	24	Evaluation of 1080p HDTV.
	ParkScene	1920 × 1080	24	
	Cactus	1920 × 1080	50	
C	RaceHorses	832 × 480	30	Mobile video applications.
	PartyScene	832 × 480	50	
	BasketballDrill	832 × 480	50	
	BQMall	832 × 480	60	
E	FourPeople	1280 × 720	60	Visual interactive communications.
	Johnny	1280 × 720	60	

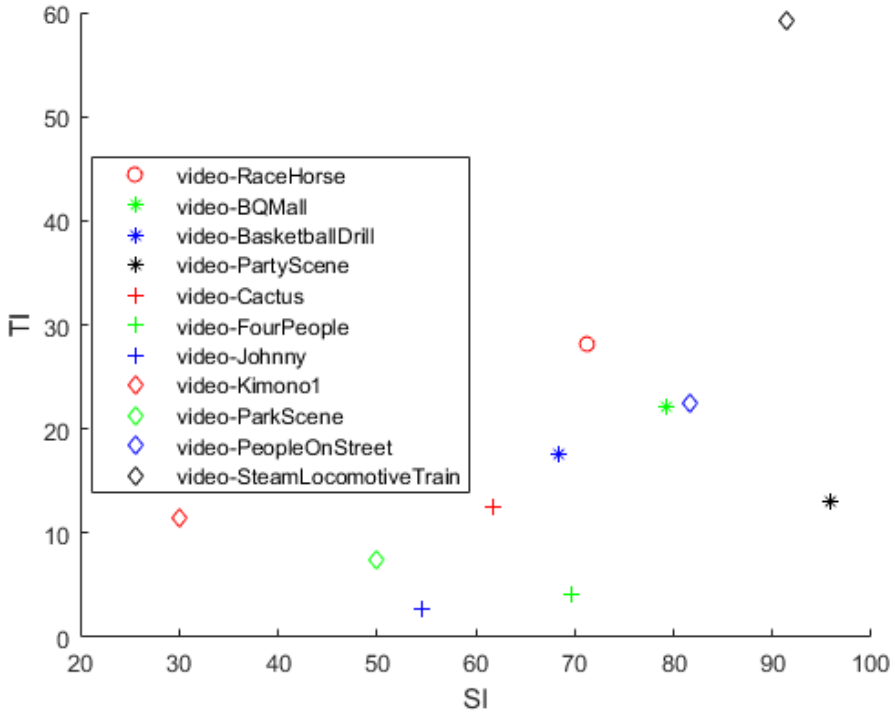


Figure 4.7 : *Diagramme spatio-temporel pour JCT-VC. Séquences des classes A-B-C.*

Comme précédemment, les résultats obtenus par notre approche, en termes de taux de gain de temps de codage ΔTS (%) et de variation de débit ΔBR (%) [55], pour les séquences vidéo de la classe C sont présentées dans le tableau 5 et pour les trois modes : AI, RA et LD. Ces tests ont été réalisés avec une configuration avec QP = 28. Ce fort gain de temps d'encodage s'accompagne de seulement 0,89%, 0,82% et 0,73% de débits en plus pour une même qualité en termes de PSNR, respectivement pour les modes AI, RA et LD. Il y a une augmentation du BDBR de 1,33 % pour RA-Main, 1,64 % tout en mode intra et 2,81 % en mode faible retard.

Tableau 4.5 : *Évaluation des performances de la méthode recommandée par rapport aux séquences utilisées*

		Random Access (RA)				**All intra (AI)**				**low delay (LD)**			
Class	**Video sequence**	**ΔTime**	**ΔBits**	**ΔPSNR**	**BDBR (%)**	**ΔTime**	**ΔBits**	**ΔPSNR**	**BDBR (%)**	**ΔTime**	**ΔBits**	**ΔPSNR**	**BDBR (%)**
class A	Traffic	-43,15	0,59	-0.04	0,6	-42,81	0,68	-0,03	0,4				
	PeopleOnStreet	-42,73	0,66	-0,07	0,92	-43,11	0,75	-0,07	0,5				
class B	Kimono	-45,53	0,95	-0,04	1,18	-44,9	0,83	-0,04	0,99	-45,33	1,06	-0.03	2,6
	Cactus	-46,85	1,21	-0,05	0,76	-46,06	1,16	-0.04	1,45	-45,25	1,15	-0,05	4,1
	ParkScene	-41,53	0,86	-0,03	0,93	-42,91	1,06	-0,03	2,55	-43,19	0,92	-0.02	2,18
class C	BasketBallDrill	-42,65	0,68	-0.04	0,81	-41,71	0,58	-0.05	0,67	-41,3	0,4	-0.03	3,16
	PartyScene	-40,93	0,96	-0.04	0,54	-41,91	0,96	-0,03	0,84	-40,19	0,89	-0,04	1,94
	RaceHorses	-41,53	1,13	-0.03	2,45	-40,59	0,97	-0.03	1,05	-39,33	0,96	-0.03	2,51
	BQMall	-43,85	0,81	-0.04	3,8	-41,96	0,76	-0.04	1,67	-43,25	0,65	-0.03	4,54
class E	FourPeople					-36,97	1,17	-0,05	1,89	-40,33	0,76	-0,04	2,45
	Johnny					-34,45	0,86	-0,03	0,49	-41,15	0,55	-0.03	1,8
	Average	**-43,19**	**0,87**	**-0,05**	**1,33**	**-35,15**	**0,89**	**-0,04**	**1,14**	**-42,15**	**0,82**	**-0,04**	**2,81**

Le tableau 4.6 présente les comparaisons de performances de la méthode proposée, toujours en termes de taux de gain de temps d'encodage et de variation des débits, par rapport à plusieurs autres techniques récentes, utilisant les quatre séquences vidéo de classe A, B, C et E pour les trois modes AI, RA et LD avec QP = 28.

Tableau 4.6. *Comparaison des performances.*

	Random Access (RA)				**All intra (AI) modes**				**low delay (LD)**			
Methods	**ΔTime**	**ΔBits**	**ΔPSNR**	**BDBR (%)**	**ΔTime**	**ΔBits**	**ΔPSNR**	**BDBR (%)**	**ΔTime**	**ΔBits**	**ΔPSNR**	**BDBR (%)**
[44]	-	-	-	-	-24,9	-	-	0,56	-39,27	-	-	1,24
[33]	-	-	-	-	-41,59	0,81	-0.05	1,73	-	-	-	-
[32]	-35,13	0,26	-0.06	0,46	-	-	-	-	-35,22	0.16	-0.03	0,59
[39]	-10,41	-	-	2,6	-	-	-	-	-9,72	-	-	3,31
[46]	-41,8	0.24	-0.02	0,42	-	-	-	-	-43,48	0.05	-0.02	0,51
Proposed	-43,19	0,87	-0,05	1,33	-35,15	0,89	-0,04	1,44	-42,15	0,82	-0,04	2,81

L'approche proposée donne une moyenne de taux de gain de temps d'encodage pour les quatre séquences vidéo de classe A, B, C et E et pour les trois modes parmi les plus

intéressants. En effet, la méthode proposée surpasse, en termes de gain de temps de codage, toutes les méthodes utilisées dans la comparaison, pour les trois modes AI, RA et LD. Cependant, cette nouvelle approche conduit à de très légères dégradations de la qualité (petites diminutions du PSNR) et également à une petite augmentation des débits. Ces limitations, en termes de diminution du PSNR et d'augmentation des débits binaires, peuvent être un peu plus importantes mais acceptables, par rapport aux méthodes visant à rendre le codec plus efficace. Notez que la méthode proposée conduit à la même réduction du temps de calcul à la fois en mode RA et en mode LD que dans [46] mais avec une certaine augmentation du débit binaire, ce qui est l'une de ses limitations. Par contre, la méthode proposée est applicable pour les trois modes RA, AI et LD alors que [46] n'a pas été proposé pour le mode AI. La référence [33] présente une réduction du temps de calcul plus importante que la méthode proposée mais uniquement pour le mode AI. Notez également que l'approche proposée vise à réduire la complexité grâce à des décisions rapides sur l'homogénéité spatiale des blocs CU en fonction de leurs tailles. Cette approche aura certainement un impact sur la dispersion des informations spatiales voire temporelles des séquences vidéo.

4.4.5 Évaluation des distorsions spatio-temporelles (classe C)

L'évaluation des performances des algorithmes de compression vidéo s'appuie fortement sur des métriques objectives, tels que MSE, PSNR et/ou SSIM. Le plus souvent, ces métriques, comme le MSE et le PSNR, sont basés sur la différence d'intensité entre les pixels de l'image originale et ceux de l'image reconstruite. Pour une séquence vidéo, la qualité de chaque image est évaluée indépendamment des autres images.

La qualité globale de l'ensemble de la séquence vidéo est alors obtenue sous la forme d'une moyenne temporelle. En revanche, les distorsions dues à la compression vidéo, notamment à bas débit, sont lisibles à l'œil humain et entraînent généralement des modifications de la structure spatio-temporelle des séquences vidéo reconstruites.

Nous proposons ici d'évaluer l'impact de ces méthodes de réduction des coûts de calcul HEVC, basées sur l'homogénéité spatiale, sur la distorsion des vidéos reconstruites. Ceci, en analysant les informations perceptives spatiales (SI) et les

informations perceptives temporelles (TI), diverses séquences d'images de test, ainsi que les originaux et ceux qui ont été encodés/décodés. Rappelons que TI et SI mesurent respectivement la quantité de variations de temps dans une séquence vidéo et la quantité de détails spatiaux de l'image[48]. À cet effet, le TI et le SI sont plus élevés respectivement pour les séquences à mouvement élevé et les scènes spatialement complexes. Puisqu'il s'agit de processus de codage avec perte, ils peuvent être représentés comme ajoutant des distorsions à chaque CU. Ces distorsions vont donc se propager spatialement aux CU voisines du fait entre autres de l'intra-prédiction et temporellement du fait de l'inter-prédiction. L'importance de ces distorsions peut être appréciée en comparant les plans TI/SI des séquences vidéo originales avec ceux reconstruits après un encodage HEVC original ou HEVC modifié selon l'approche proposée. Nous avons donc procédé au calcul du TI et du SI de chaque séquence vidéo individuellement pour le HEVC original et modifié en utilisant les trois modes AI, RA et LD. Ainsi, chaque plan TI/SI représente une seule vidéo pour les six cas dans ce cas HM-AI original, HM-RA original, HM-LD original, HM-AI modifié, HM-RA modifié et HM-LD modifié (voir Figure.4. 8). Comme on peut le voir à travers ces différents plans TI/SI une certaine dispersion des positions pour les différentes séquences par rapport à la séquence originale existe et elle est relativement importante. Cependant, cette dispersion des positions dans le plan TI/SI des séquences reconstruites à partir du HM modifié est plus importante pour le mode RA et moins importante pour le mode AI, quelle que soit la vidéo utilisée.

Comme on peut le voir à travers ces différents plans TI/SI une certaine dispersion des positions des différentes séquences par rapport à la séquence d'origine existe et elle est relativement importante. Cependant, cette dispersion des positions dans le plan TI/SI des séquences reconstruites à partir du HM modifié par rapport à la vidéo originale est plus importante pour le mode LD et moins importante pour le mode AI, quelle que soit la vidéo utilisée. Pour mieux apprécier ces différences, nous avons calculé les écarts minimum, maximum et standard des TI/SI pour chaque mode (voir tableau 4.7). Cela nous a permis d'une part de confirmer notre première appréciation à savoir que le mode LD conduit à une plus grande dispersion du TI/SI aussi bien pour le HM d'origine que modifié. Mais ce tableau montre également que le SI subit des changements plus

importants que le TI. Nous pouvons expliquer cela par l'impact plus important de l'homogénéité spatiale sur la décision relative à la taille de l'CU. Les changements temporels des distorsions spatiales peuvent modifier significativement leurs perceptions.

Tableau 4. 7 : *Valeurs TI et SI des vidéos de classe C décodées.*

Séquence vidéo		**MAX value**		**MIN value**		**the standard déviation**	
		SI	**TI**	**SI**	**TI**	**SI**	**TI**
BasketBallDrill	Originalvideo	76.86	20.81	62.22	0	2.57	1.76
	HM reconstructed-RA	71.12	21.18	50.62	7.67	3.60	1.48
	modified HM recontructed-RA	71.00	21.11	51.00	5.10	3.46	1.47
	HM reconstructed-AI	76.16	20.73	61.41	0	2.61	1.75
	modified HM recontructed-AI	76.10	20.66	61.23	0	2.59	1.72
	HM reconstructed-LD	63.40	20.64	53.41	5.84	1.97	1.89
	modified HM recontructed-LD	63.35	20.60	53.41	4.19	1.86	1.84
PartyScene	originalvideo	106.57	17.95	95.94	0	3.02	1.82
	HM reconstructed-RA	102.64	19.01	80.83	10.26	4.10	1.06
	modified HM recontructed-RA	102.59	19.00	80.79	9.99	4.00	1.00
	HM reconstructed-AI	106.31	17.91	95.92	0	2.99	1.79
	modified HM reconstructed-AI	106.40	17.89	95.90	0	2.94	1.78
	HM reconstructed-LD	97.58	16.03	89.91	8.33	2.43	1.19
	modified HM reconstructed-LD	97.53	16.00	89.89	8.20	2.40	1.20
RaceHorses	originalvideo	85.85	36.98	52.38	16.11	7.78	4.56
	HM reconstructed-RA	82.52	35.78	48.86	14.27	6.62	4.84
	modified HM reconstructed-RA	82.60	35.80	48.90	14.30	6.65	4.83
	HM reconstructed-AI	95.20	20.78	63.96	12.67	7.42	2.12
	modified HM recontructed-AI	95.25	20.79	63.94	12.64	7.30	2.10
	HM reconstructed-LD	76.94	35.89	56.16	26.24	5.40	2.96
	modified HM reconstructed-LD	76.90	35.80	56.20	26.30	5.42	2.95
BQMall	originalvideo	95.47	21.14	64.18	12.78	7.42	2.21
	HM reconstructed-RA	92.24	21.18	57.61	14.03	5.64	1.53
	modified HM reconstructed-RA	92.25	21.12	57.60	14.00	5.69	1.54
	HM reconstructed-AI	95.20	20.78	63.96	12.67	7.42	2.12
	modified HM recontructed-AI	95.21	20.74	63.96	12.65	7.40	2.13
	HM reconstructed-LD	85.80	20.74	71.07	13.51	5.11	2.32
	modified HM recontructed-LD	85.81	20.75	71.02	13.52	5.12	2.35

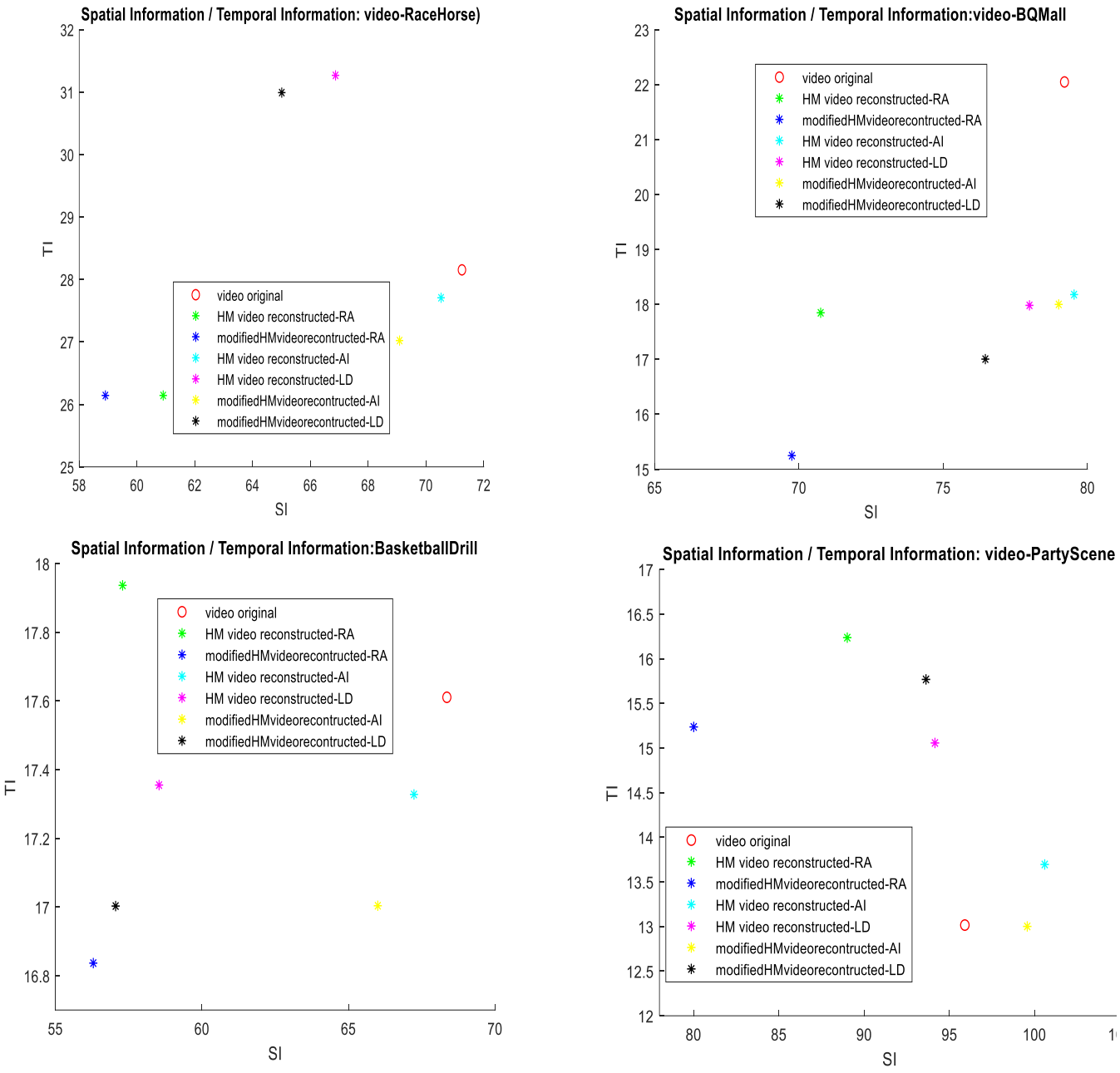


Figure 4.8 : *SI et TI des vidéos de classe C décodées.*

4.5 Conclusion

Ce chapitre présente un nouveau mécanisme de prise de décision rapide de la taille de CU afin de réduire le coût du calcul HEVC en fonction de son homogénéité spatiale, Cette méthode fonctionne en trois étapes : (1) La direction de corrélation de bloc dans HEVC est déterminée à l'aide de la direction du vecteur central de masse de bloc. . (2) Création d'index pour CU homogène. (3) A modeling of the decision process. Le logiciel de référence HEVC 16.13 est appliqué pour évaluer l'algorithme proposé. Les résultats

expérimentaux comparatifs ont démontré que l'algorithme proposé peut réduire considérablement la complexité de calcul de l'encodeur H.265. . En effet, le temps de codage est réduit à 40% pour les séquences vidéo CIF/QCIF et de 35% à 43% pour les séquences de test de classe A, B et C tout en conservant quasiment les mêmes performances R&D que l'encodeur d'origine pour les trois modes de configuration (Tout intra, Random Access et Low Delay). D'autre part, des comparaisons ont également été faites sur la façon dont ces approches impactent les informations perceptives spatiales (SI) et les informations perceptives temporelles (TI) des séquences utilisées.

CONCLUSION GENERALE

CONCLUSION GENERALE

Cette thèse présente un nouveau mécanisme de prise de décision rapide pour déterminer la taille des Coding Units (CU) dans le cadre de la compression HEVC, permettant ainsi de réduire le coût de calcul. Ce mécanisme repose sur l'homogénéité spatiale des CU et se décompose en trois étapes :

- Détermination de la corrélation des blocs dans HEVC en utilisant la direction vecteur du centre de masse des blocs.
- Création d'un index pour les CU homogènes.
- Modélisation du processus de prise de décision.

L'algorithme proposé est évalué à l'aide du logiciel de référence HEVC 16.13. Les résultats expérimentaux comparatifs démontrent que cet algorithme permet de réduire considérablement la complexité de calcul de l'encodeur H.265. En effet, le temps de codage est réduit de 40% pour les séquences vidéo CIF/QCIF et de 35% à 43% pour les séquences de test de classe A, B et C, tout en maintenant des performances R&D presque identiques à celles de l'encodeur d'origine pour les trois modes de configuration (Tout intra, Random Access et Low Delay).

Par ailleurs, des comparaisons ont également été réalisées pour évaluer l'impact de ces approches sur les informations perceptives spatiales (SI) et temporelles (TI) des séquences utilisées. L'efficacité de cet algorithme repose sur la détermination de l'homogénéité spatiale des CU à partir de l'amplitude de vecteur du centre de masse simplifié (SMCV) et de l'indice d'homogénéité temporelle.

Dans nos travaux futurs, nous prévoyons d'étendre cette approche à la norme VVC, qui adopte un schéma de codage vidéo hybride similaire à HEVC mais avec des Coding Tree Units (CTU) de taille maximale augmentée à 128 × 128 pixels. Ainsi, l'approche proposée dans cette thèse pour HEVC peut être appliquée au nouveau standard VVC.

En ce qui concerne le codage vidéo multi-vue (MVC), utilisé pour enregistrer une même scène sous différents angles, nous envisageons d'étendre l'idée de l'index contenant les données d'homogénéité spatiale à d'autres vues afin de réduire davantage

le temps de décodage de toutes les vues. Cette extension est rendue possible par la forte corrélation existante entre les points de vue d'une même séquence.

Ainsi, cette thèse propose une méthode novatrice de prise de décision rapide pour la taille des CU dans le cadre de la compression HEVC, avec des applications potentielles dans le nouveau standard VVC et le codage vidéo multi-vue (MVC).

REFERENCES

[1] Y. Zhang, S. Kwong, and S. Wang, "Machine learning based video coding optimizations: A survey," *Information Sciences,* vol. 506, pp. 395-423, 2020.

[2] K. Jack, *Digital video and DSP: Instant access*. Newnes, 2008.

[3] B. Gordon, N. Chaddha, and T.-Y. Meng, "A low-power multiplierless YUV to RGB converter based on human vision perception," in *Proceedings of 1994 IEEE workshop on VLSI signal processing*, 1994: IEEE, pp. 408-417.

[4] C. A. Poynton, *A technical introduction to digital video*. John Wiley & Sons, Inc., 1996.

[5] C. Poynton, *Digital video and HD: Algorithms and Interfaces*. Elsevier, 2012.

[6] F. Zhang and D. R. Bull, *Intelligent image and video compression: communicating pictures*. Academic Press, 2021.

[7] P. Tudor, "MPEG-2 video compression," *Electronics & communication engineering journal,* vol. 7, no. 6, pp. 257-264, 1995.

[8] S. Winkler, *Digital video quality: vision models and metrics*. John Wiley & Sons, 2005.

[9] I. T. Union-Telecommun, "Generic coding of moving pictures and associated audio information-Part 2: Video," *Int. Standards Org./Int. Electrotech. Comm.(ISO/IEC) JTC 1, Rec. H. 262 and ISO/IEC 13 818-2 (MPEG-2 Video),* 1994.

[10] S. CCITT and H. Recommendation, "261–Video Codec for Audiovisual Services at px64 kbit/s," *The International Telegraph and Telephone Consultative Committee,* vol. 115, 1990.

[11] I.-T. SG15, "Video coding for low bitrate communication," *Draft ITU-T Rec. H. 263,* 1996.

[12] I. E. Richardson, *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.

[13] I. Rec, "H. 264: Advanced video coding for generic audiovisual services," *ITU-T Rec. H. 264-ISO/IEC 14496-10 AVC,* 2005.

[14] J. Samelak, J. Stankowski, and M. Domański, "Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 26th Meeting: Geneva, CH, 12–20 January 2017."

[15] J. Chen, Y. Ye, and S. Kim, "Algorithm description for Versatile Video Coding and Test Model 10 (VTM 10), document JVET-S2001 of Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11," in *Proceedings of the 19th JVET Meeting, Geneva, Switzerland*, 2020, vol. 22.

[16] J. Chen, Y. Ye, and S. Kim, "JVET-Q2002-v3: Algorithm description for Versatile Video Coding and Test Model 8 (VTM 8)," in *Proceedings of the 17th JVET Meeting, Brussels, Belgium*, 2020, pp. 7-17.

[17] C. Rosewarne, K. Sharman, R. Sjöberg, and G. Sullivan, "High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Encoder Description Update 13| MPEG," in *38th JCT-VC Meeting, Brussels*, 2020.

[18] F. Bossen, "Common test conditions and software reference configurations. doc. jctvc-k1100," in *11th Meeting: Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG*, 2013, vol. 16.

[19] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on circuits and systems for video technology,* vol. 22, no. 12, pp. 1649-1668, 2012.

[20] M. Wien, "High efficiency video coding," *Coding Tools and specification,* vol. 24, 2015.

[21] R. Sjoberg *et al.*, "Overview of HEVC high-level syntax and reference picture management," *IEEE transactions on Circuits and Systems for Video Technology,* vol. 22, no. 12, pp. 1858-1870, 2012.

[22] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block partitioning structure in the HEVC standard," *IEEE transactions on circuits and systems for video technology,* vol. 22, no. 12, pp. 1697-1706, 2012.

[23] V. Sze, M. Budagavi, and G. J. Sullivan, "High efficiency video coding (HEVC)," in *Integrated circuit and systems, algorithms and architectures*, vol. 39: Springer, 2014, p. 40.

[24] P. Bordes, G. Clare, F. Henry, M. Raulet, and J. Viéron, "An overview of the emerging HEVC standard," in *International Symposium on Signal, Image, Video and Communications, ISIVC],(July 2012)*, 2012.

[25] B. Bross, "High efficiency video coding (HEVC) text specification draft 9 (SoDIS)," in *11th JCT-VC meeting, Oct. 2012*, 2012.

[26] M. Haque, A. Tabatabai, and Y. Morigami, "HVS model based default quantization matrices," *Document JCTVC-G880,* pp. 1-13, 2011.

[27] N. Archana, G. Keerthivasan, and B. Janakiraman, "SAO filter in HEVC: A survey," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020: IEEE, pp. 974-979.

[28] C.-M. Fu *et al.*, "Sample adaptive offset in the HEVC standard," *IEEE Transactions on Circuits and Systems for Video technology,* vol. 22, no. 12, pp. 1755-1764, 2012.

[29] A. Moffat, R. M. Neal, and I. H. Witten, "Arithmetic coding revisited," *ACM Transactions on Information Systems (TOIS),* vol. 16, no. 3, pp. 256-294, 1998.

[30] S. Ahn, B. Lee, M. J. I. T. o. C. Kim, and S. f. V. Technology, "A novel fast CU encoding scheme based on spatiotemporal encoding parameters for HEVC inter coding," vol. 25, no. 3, pp. 422-435, 2014.

[31] Y.-J. Ahn and D. J. J. o. R.-T. I. P. Sim, "Square-type-first inter-CU tree search algorithm for acceleration of HEVC encoder," vol. 12, pp. 419-432, 2016.

[32] K. Goswami, J.-H. Lee, K.-S. Jang, B.-G. Kim, and K.-K. J. J. o. R.-T. I. P. Kwon, "Entropy difference-based early skip detection technique for high-efficiency video coding," vol. 12, pp. 237-245, 2016.

[33] M. Maazouz, N. Batel, N. Bahri, N. J. E. S. Masmoudi, and a. I. J. Technology, "Homogeneity-based fast CU partitioning algorithm for HEVC intra coding," vol. 22, no. 3, pp. 706-714, 2019.

[34] H. Ding, X. Huang, and Q. J. O. Zhang, "The fast intra CU size decision algorithm using gray value range in HEVC," vol. 127, no. 18, pp. 7155-7161, 2016.

[35] K. Saurty, P. C. Catherine, and K. Soyjaudah, "Early CU size determination in HEVC intra prediction using average pixel cost," in *2014 Fourth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, 2014: IEEE, pp. 247-252.

[36] N. Westland, A. S. Dias, and M. Mrak, "Decision trees for complexity reduction in video compression," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019: IEEE, pp. 2666-2670.

[37] Z. Liu, T.-L. Lin, C.-C. J. M. T. Chou, and Applications, "HEVC coding-unit decision algorithm using tree-block classification and statistical data analysis," vol. 76, pp. 9051-9072, 2017.

[38] Z. Liu, T.-L. Lin, C.-C. J. J. o. V. C. Chou, and I. Representation, "Efficient prediction of CU depth and PU mode for fast HEVC encoding using statistical analysis," vol. 38, pp. 474-486, 2016.

[39] H. Sun, D. Zhou, L. Hu, S. Kimura, and S. J. I. T. o. M. Goto, "Fast algorithm and VLSI architecture of rate distortion optimization in H. 265/HEVC," vol. 19, no. 11, pp. 2375-2390, 2017.

[40] M. K. Jha, R. Chaudhary, S. D. Roy, M. Mathur, and B. J. M. S. Lall, "Restricted affine motion compensation and estimation in video coding with particle filtering and importance sampling: a multi-resolution approach," vol. 24, pp. 271-284, 2018.

[41] Y. Zhang, H. Wang, and Z. Li, "Fast coding unit depth decision algorithm for interframe coding in HEVC," in *2013 Data Compression Conference*, 2013: IEEE, pp. 53-62.

[42] Z. Pan, S. Kwong, Y. Zhang, J. Lei, and H. Yuan, "Fast coding tree unit depth decision for high efficiency video coding," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014: IEEE, pp. 3214-3218.

[43] D. G. Fernández, A. A. Del Barrio, G. Botella, C. J. M. T. García, and Applications, "Fast and effective CU size decision based on spatial and temporal homogeneity detection," vol. 77, pp. 5907-5927, 2018.

[44] D. Fernández, A. Del Barrio, G. Botella, and C. García, "Fast CU size decision based on temporal homogeneity detection," in *2016 Conference on Design of Circuits and Integrated Systems (DCIS)*, 2016: IEEE, pp. 1-6.

[45] Z. Yang, S. Guo, Q. J. M. T. Shao, and Applications, "A fast inter-frame encoding scheme using the edge information and the spatiotemporal encoding parameters for HEVC," vol. 76, pp. 24125-24142, 2017.

[46] X. Shang, G. Wang, T. Fan, and Y. J. J. o. E. I. Li, "Statistical and spatiotemporal correlation based low-complexity video coding for high-efficiency video coding," vol. 24, no. 2, pp. 023006-023006, 2015.

[47] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. J. I. T. o. I. P. Siu, "Online-learning-based Bayesian decision rule for fast intra mode and CU partitioning algorithm in HEVC screen content coding," vol. 29, pp. 170-185, 2019.

[48] D. Lee and J. Jeong, "Fast CU size decision algorithm using machine learning for HEVC intra coding," *Signal Processing: Image Communication,* vol. 62, pp. 33-41, 2018.

[49] W. Liao and Z. Chen, "A fast CU partition and mode decision algorithm for HEVC intra coding," *Signal Processing: Image Communication,* vol. 67, pp. 140-148, 2018.

[50] "HM (HEVC Test Model), ." https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.13/. (accessed.

[51] A. Elyousfi, "Gravity direction-based ultra-fast intraprediction algorithm for H. 264/AVC video coding," *Signal, Image and Video Processing,* vol. 7, no. 1, pp. 53-65, 2013.

[52] A. Elyousfi, A. El Hachimi, and H. Hamout, "Texture complexity based fast and efficient intra block mode decision algorithm for H. 264/AVC," in *2017 Intelligent Systems and Computer Vision (ISCV)*, 2017: IEEE, pp. 1-4.

[53] H. Hamout and A. Elyousfi, "Fast depth map intra-mode selection for 3D-HEVC intra-coding," *Signal, Image and Video Processing,* vol. 14, no. 7, pp. 1301-1308, 2020.

[54] H. Hamout and A. Elyousfi, "Fast depth map intra coding for 3D video compression-based tensor feature extraction and data analysis," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 30, no. 7, pp. 1933-1945, 2019.

[55] S. Abdellah, N. Mokhtar, and S. Amina, "Complexity reduction in the H. 264/AVC using highly adaptive fast mode decision based on macroblock motion activity," *Journal of Electronic Imaging,* vol. 24, no. 6, pp. 063015-063015, 2015.

[56] I.-K. Kim, K. McCann, K. Sugimoto, B. Bross, W.-J. Han, and G. Sullivan, "High efficiency video coding (HEVC) test model 10 (HM10) encoder description," in *Document: JCTVC-O1002, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 15th Meeting: Geneva*, 2013.

[57] F. Bossen, "Common test conditions and software reference configurations," *JCTVC-L1100,* vol. 12, no. 7, 2013.

[58] "xiph.Org , Derf's test media collection." https://media.xiph.org/video/derf,

. (accessed.

[59] P. ITU-T RECOMMENDATION, "Subjective video quality assessment methods for multimedia applications," 1999.

[60] R.Brai, A. Bekhouch, N. Doghmane, S.Harize, N. Kouadria ,`` Low calculation cost of HEVC coding unit size based on spatial homogeneity detection``, Journal of Visual Communication and Image Representation,elsevier 2023.