



HAL
open science

Optimisation du codage HEVC par des moyens de pré-analyse et/ou de pré-codage du contenu

Nicolas Dhollande

► **To cite this version:**

Nicolas Dhollande. Optimisation du codage HEVC par des moyens de pré-analyse et/ou de pré-codage du contenu. Sciences de l'ingénieur [physics]. Ecole doctorale Matisse, 2016. Français. NNT : . tel-01758094

HAL Id: tel-01758094

<https://inria.hal.science/tel-01758094>

Submitted on 4 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

Ecole doctorale Matisse

présentée par

Nicolas Dhollande

Préparée à l'unité de recherche IRISA SIROCCO,
Thomson Video Networks et à la fondation b<>com

**Optimisation du
codage HEVC
par des moyens de
pré-analyse et/ou de
pré-codage du
contenu**

Thèse soutenue à Rennes
le 21 avril 2016

devant le jury composé de :

Daniel Ménard

Prof. INSA Rennes/*Examineur*

Anissa Mokraoui

Prof. Univ. Paris XIII/*Rapporteur*

Michel Kieffer

Prof. Supélec Paris Sud/*Rapporteur*

Mohamed-Chaker Larabi

MdC. Univ. Poitiers/*Examineur*

Christine Guillemot

DR. INRIA Rennes/*Directrice de thèse*

Olivier Le Meur

MdC. Univ. Rennes 1/*Co-directeur de thèse*

Xavier Ducloux

Responsable de programmes de recherche,
Thomson Video Networks/*Examineur*

Remerciements

Je souhaiterais tout d'abord remercier mes directeurs de thèse, Christine Guillemot et Olivier Le Meur, pour leur encadrement durant cette thèse, leurs remarques pertinentes, mais aussi leurs re-lectures, qui m'ont permis d'améliorer ce rapport et les publications.

Je tiens à exprimer ma gratitude à Xavier Ducloux, qui m'a accompagné depuis mon stage de fin d'étude, puis pendant cette thèse au sein de Thomson Video Networks et de b<>com. Il m'a apporté son expérience dans le domaine de la compression vidéo, et des conseils toujours avisés.

Je remercie également l'équipe Technologie de Thomson Video Networks pour leur accueil depuis près de 4 ans : Claude, Patrick, Eric, Raoul et Chantal.

Je remercie tous mes compagnons de thèse : Franck, Thibaud, Antonin, Cambodge, Andrei, mais aussi Fabrice, Victorien, David, Adrien, Pierre et Ulrich avec qui on a passé d'excellents moments, pendant et en dehors du travail. Merci à toute l'équipe MediaImmersif et à b<>com, pour nous avoir assuré d'excellentes conditions pour réaliser cette thèse.

Enfin, je tiens à remercier mes amis, ma famille, mes parents, pour leurs soutiens et encouragements sans failles.

Table des matières

Introduction générale	1
I Introduction à la norme HEVC (<i>High Efficiency Video Coding</i>)	5
1 Description du schéma de compression vidéo HEVC	6
1.1 Développement de la norme HEVC en standardisation	6
1.2 Représentation du signal vidéo	8
1.2.1 Structure	8
1.2.2 Transmission	8
1.2.3 Conversion couleur et sous-échantillonnage	10
1.3 Le schéma de codage hybride de HEVC	11
1.3.1 Partitionnement initial du signal vidéo	11
1.3.2 Structure de codage hiérarchique HEVC	13
1.3.3 Généralités sur la prédiction	14
1.3.4 Prédiction Intra	15
1.3.5 Prédiction Inter et codage du mouvement	16
1.3.6 Transformation et codage du résiduel	19
1.3.7 Autres outils HEVC	20
1.4 Contrôle du codeur et optimisation.	22
1.4.1 Optimisation au sens débit-distortion	22
1.4.2 Estimation des modes de codage	23
1.4.3 Estimation des paramètres de mouvement	23
1.4.4 Estimation des modes de prédiction Intra	24
1.5 Conclusion	25
2 Performance et complexité du codage HEVC	26
2.1 Méthodologie utilisée	26
2.1.1 Configurations de codage	26
2.1.2 Séquences utilisées	27
2.1.3 Réduction de complexité	28
2.1.4 Qualité de codage	28
2.2 Performance de la structure de codage HEVC	30
2.3 Construction de la structure de codage CU dans le modèle de référence	31
2.4 Réduction de complexité envisageable	32
2.4.1 Réduction de complexité maximum	32
2.4.2 Réduction de complexité effective en évaluant les modes optimaux uniquement	32
2.5 Conclusion	33

II	Méthodes de réduction de complexité appliquées au codage des images Intra	35
3	Approches conventionnelles pour la réduction de complexité	36
3.1	Réduction de la complexité appliquée au partitionnement <i>quad-tree</i>	37
3.1.1	Approches basées sur les statistiques de codage	37
3.1.2	Approches basées sur la pré-analyse du contenu	37
3.1.3	Re-configuration de l'approche récursive du partitionnement	38
3.2	Réduction de la complexité appliquée au choix du meilleur mode de prédiction	38
3.2.1	Pré-sélection basée sur l'activité directionnelle de la texture	39
3.2.2	Autres méthodes	39
3.3	Réduction de complexité appliquée à la représentation des résidus	40
3.3.1	<i>Zero-blocks</i>	40
3.3.2	Re-configuration de l'approche récursive du partitionnement	40
3.4	Conclusion	41
4	Inférence de modes dans un codage basé pré-codage de sous-résolution	42
4.1	Présentation de l'architecture de test	43
4.1.1	Principe	43
4.1.2	Filtre de sous-échantillonnage	43
4.2	Inférence du partitionnement	45
4.2.1	Analyse des partitionnements HD et UHD	45
4.2.2	Stratégies d'inférence du partitionnement	46
4.2.3	Résultats expérimentaux	48
4.3	Inférence du mode de prédiction	49
4.3.1	Analyse des corrélations entre les modes HD et UHD	50
4.3.2	Stratégies de pré-sélection du mode UHD	51
4.3.3	Résultats expérimentaux	52
4.4	Conclusion	54
5	Codage accéléré par la pré-analyse locale du contenu	55
5.1	Motivations	55
5.2	Algorithme de partitionnement rapide à 2 classes	57
5.2.1	Apprentissage des seuils	57
5.2.2	Partitionnement rapide	58
5.2.3	Résultats expérimentaux	58
5.3	Algorithme de partitionnement rapide à 3 classes	60
5.3.1	Apprentissage des seuils	60
5.3.2	Partitionnement rapide	60
5.3.3	Résultats expérimentaux	61
5.4	Analyse du comportement de l'algorithme	62
5.5	Autres optimisations possibles	64
5.5.1	Simplification du choix des modes de prédiction et des TU	64
5.5.2	Combinaison avec l'algorithme de pré-codage	65
5.6	Apprentissage <i>online</i>	66
5.6.1	Principe	66
5.6.2	Résultats expérimentaux	67
5.7	Conclusion	68

III Extension à un contexte de diffusion vidéo (images inter-prédites)	71
6 Approches conventionnelles pour la réduction de complexité	72
6.1 Types de terminaisons rapides	72
6.2 Approches basées sur la corrélation du voisinage spatio-temporel	74
6.3 Approches basées sur les caractéristiques de codage du bloc courant	75
6.4 Conclusion	76
7 Codage rapide optimisé pour la diffusion au format UHD	78
7.1 Simplification des outils de codage HEVC	78
7.2 Limitation du nombre d'images de référence	80
7.3 Conclusion	81
8 Inférence de modes dans un codage basé pré-codage de sous-résolution	82
8.1 Analyse des partitionnements et des types de prédictions HD et UHD	82
8.2 Inférence du partitionnement	84
8.2.1 Stratégies d'inférence du partitionnement	84
8.2.2 Résultats expérimentaux	85
8.3 Inférence conditionnelle aux types de prédictions	85
8.3.1 Condition sur le mode SKIP	85
8.3.2 Stratégie d'inférence optimisée	86
8.3.3 Résultats expérimentaux	87
8.4 Inférence du partitionnement et des types de prédiction, combinée à une configuration de codage rapide.	88
8.4.1 Principe	88
8.4.2 Résultats expérimentaux	89
8.5 Conclusion	89
Conclusion générale	91
Bibliographie	94
Annexes	99
A Séquences utilisées	101
A.1 EBU-UHD	101
A.2 ClasseB-HD	102
B Performance HEVC	102
C Codage rapide des images I basé sur le pré-codage de sous-résolution	103
D Codage rapide des images I basé sur la pré-analyse	103
D.1 Opérateurs de pré-analyse	103
D.2 Seuils de décision	105
D.3 Résultats expérimentaux : algorithme à 3 classes	105
E Codage rapide des images B basé sur le pré-codage de sous-résolution	106
E.1 Résultats expérimentaux : inférence du partitionnement	106
E.2 Résultats expérimentaux : inférence conditionnelle aux types de pré-dictions	106
E.3 Résultats expérimentaux : configuration de codage optimisé	107

Publications

112

Introduction générale

Les travaux présentés dans cette thèse traitent la compression des vidéos et des solutions rapides de codage HEVC adaptées à la très haute définition –ou *Ultra High Definition*– (UHD).

Contexte

Dès la fin des années 80, et l'apparition de la vidéo numérique, la compression vidéo est devenue indispensable. Depuis, des standards de compression vidéo ont été développés avec un taux de compression toujours plus élevé. H.262/MPEG-2 (1994) et H.264/AVC (2003) sont les deux principaux standards ayant connu un succès industriel majeur : l'un porté par les premières chaînes de télévision numérique et le format physique DVD, l'autre plus récemment, par l'avènement des chaînes de télévision au format haute définition –ou *High Definition*– (HD), le *BluRay* et la vidéo sur Internet.

L'usage de la vidéo connaît une formidable croissance au travers du réseau Internet. Dans un trafic Internet mondial en constante augmentation, la part de la vidéo ne cesse d'augmenter, et on estime qu'elle pourrait représenter 80% du trafic Internet en 2019. Les services Netflix et YouTube représentent à eux seul la moitié du trafic IP dans le monde. Il y a plusieurs raisons à cette croissance. Le nombre d'internautes ne cesse d'augmenter, il devrait atteindre 3.9 milliards en 2019 et avec un débit de plus en plus élevé pour chacun. Dans le même temps, la population s'équipe de terminaux capables de se connecter à Internet et de lire la vidéo (smartphones, tablettes, TV connectées). La résolution HD est maintenant un standard, que ce soit sur un écran plat de grande diagonale ou sur un smart-phone utilisé en mobilité. Actuellement, les premiers écrans et terminaux au format UHD apparaissent sur le marché. Le format UHD va vite peser sur le trafic car il contient 8 fois plus d'information que la HD (4 fois en résolution spatiale, 2 fois en fréquence d'images). Cela ne se fera pas sans le nouveau standard *High Efficiency Video Coding* (HEVC), défini en 2013. En effet, HEVC promet un taux de compression au moins doublé par rapport à H.264/AVC. Le format UHD devrait par ailleurs continuer à s'enrichir sur le plan de la fréquence d'images (*High Frame Rate* (HFR)), la dynamique (*High Dynamic Range* (HDR)) et l'étendue de l'espace de couleurs (*Wide Color Gamut* (WCG)).

Un nouveau standard introduit de nouveaux outils de compression et inévitablement une complexité de calcul nettement supérieure pour augmenter le taux de compression. HEVC n'échappe pas à cette règle : il est entre 2 et 10 fois plus complexe côté codeur par rapport à H.264/AVC. L'optimisation des algorithmes de compression est donc un enjeu majeur pour l'adoption d'un nouveau format vidéo, notamment l'UHD comprimée avec HEVC. Pour être compétitif, un codeur temps réel de type industriel doit être économiquement réaliste, avoir une consommation électrique mesurée tout en offrant la meilleure qualité de codage possible.

Objectifs industriels

Les travaux de la thèse ont été menés dans un environnement de recherche industrielle. L'entreprise *Thomson Video Networks* cherche à perfectionner les techniques de codage sur ses solutions, que ce soit dans les produits émergents HEVC UHD, mais aussi dans les produits existants : codeurs SD, HD, au format MPEG-2 et H.264/AVC.

Les travaux ont consisté à développer de nouvelles architectures de codage, adaptées aux nouveaux formats UHD. Le but est d'**obtenir le meilleur compromis entre réduction de complexité et maintien de la performance de compression HEVC**.

Voici les critères que nous avons établi pour réaliser ces travaux :

- *Rester en conformité avec le standard HEVC* : les travaux ont pour but de réduire la complexité de codage, en respectant la syntaxe et les mécanismes imposés par le standard.
- *Travailler à partir d'un logiciel de référence* : le modèle de référence HEVC, aussi appelé *Test Model HM*, est implémenté en C/C++ et compare les modes de codage HEVC selon un critère objectif (le rapport signal sur bruit, PSNR) et sans *a priori* sur le contenu vidéo. Ce logiciel est utilisé par l'état de l'art scientifique de la réduction de complexité pour évaluer la performance des algorithmes. Une implémentation rapide cherchera davantage à simplifier la boucle de codage en apportant de l'*a priori* sur le contenu, en faisant des approximations de calcul et en tenant compte de la qualité subjective. Dans notre cas, une mesure objective suffit à comparer la qualité de codage de l'algorithme par rapport à la référence, et une comparaison des temps d'exécution suffit à déterminer la réduction de complexité.
- *Parallélisation et optimisations dédiées* : les résultats présentés dans cette thèse ont été réalisés sans optimisations liées à une architecture de calcul spécifique, le but étant de démontrer la performance obtenue sur la boucle de codage (choix des modes de prédiction optimaux). Toutes techniques d'*hyperthreading*, de parallélisation, ou d'optimisations bas niveau (vectorisation, SSE) sont proscrites.
- *Estimation de mouvement* : l'estimation de mouvement est un sujet à part entière. Son implémentation dans les codeurs industriels correspond souvent à des algorithmes propriétaires.

Contributions majeures

Les travaux réalisés au cours de cette thèse s'articulent autour de 2 méthodes pour réduire la complexité d'un codeur HEVC : l'**utilisation d'une passe de pré-codage de sous-résolution** et la **pré-analyse du contenu vidéo source**. Ces méthodes ont pour point commun de vouloir réduire le nombre de modes de prédiction évalués par la boucle de codage, notamment pour obtenir le partitionnement optimal de l'image.

Réduction de complexité du codage des images Intra :

Dans un premier temps, nous souhaitons traiter les images Intra, qui sont très importantes dans les applications de diffusion (*broadcast*). Même si peu d'images sont codées en Intra (uniquement les images clés), leur codage doit être réalisé avec la plus grande attention. Une image de référence mal traitée peut propager des informations de codage non optimales dans la suite du *bitstream*. Le codage Intra est aussi utilisé au sein des images inter-prédites (type

P/B), notamment pour les images éloignées temporellement de leur référence. La prédiction Intra HEVC est en effet très compétitive avec la prédiction Inter en terme d'efficacité de codage. Enfin, certaines applications peuvent nécessiter de travailler avec des flux codés entièrement en Intra. Lorsqu'une chaîne de télévision doit rapatrier ses contenus vidéos depuis le lieu de production, des liens de contribution vidéo à haut débit sont mis en place pour transporter le signal avec la plus grande fidélité (codage sans perte ou visuellement sans perte) et une très faible latence, pour faciliter la manipulation du signal. De la même manière, les applications liées à l'imagerie médicale exigent une qualité d'image parfaite ainsi qu'une manipulation image par image.

Trois contributions concernent une méthode de pré-codage appliquée aux images Intra :

1. L'intégration d'un codage de sous-résolution destiné à réduire le nombre de modes de codage à évaluer. Ce pré-codage est réalisé à partir d'une sous-résolution de la vidéo originale.
2. Une méthode d'inférence du partitionnement quad-tree obtenu dans le codage de sous-résolution.
3. Une méthode de pré-sélection des modes directionnels Intra en tenant compte du mode utilisé dans le codage de sous-résolution.

Caractériser le contenu vidéo peut apporter de nombreuses informations sur le codage HEVC optimal. Ainsi, nous présentons notre méthode de pré-analyse du contenu source, appliquée aux images Intra, à travers deux contributions :

1. Un codeur a tendance à sélectionner de grands blocs pour les zones homogènes de l'image alors qu'il préfère subdiviser les textures fines et complexes en petits blocs. Nous montrons que la variance d'un bloc de luminance permet de prédire efficacement si coder cette région avec un seul bloc est optimal ou si il faut la coder avec des plus petits blocs.
2. Une méthode de seuillage permettant d'éliminer les tailles de blocs les moins probables de la procédure de partitionnement, réduisant ainsi la complexité de codage.

Réduction de complexité du codage des images Inter :

Dans un deuxième temps, nous proposons des solutions de réduction de complexité pour les images Inter (images inter-prédites de type B). La majorité des applications vidéos utilise conjointement les images Intra et Inter, pour une efficacité de compression maximale. Ces images sont plus complexes à coder, car les outils de codage sont plus nombreux et coûteux. Par ailleurs, les images inter-prédites peuvent être composées de blocs Intra. La combinatoire d'outils est considérable, néanmoins, cela offre des marges de simplifications supplémentaires pour un codeur rapide.

Nous avons choisi de travailler exclusivement sur la structure de codage hiérarchique (appelée *Random access*), qui est utilisée dans le domaine de la diffusion (*broadcast*). La méthode de pré-analyse du contenu, utilisée pour le codage Intra, n'est pas adaptée aux prédictions temporelles. Par contre, nous avons développé une approche basée sur le pré-codage de sous-résolution.

Nous exposons deux contributions majeures dans cette partie, dédiées aux images Inter :

1. Une méthode efficace d'inférence des modes de prédiction s'appuyant sur une première passe de codage de sous-résolution.

2. Une restriction de modes HEVC appliquée au modèle de référence, cohérente avec les implémentations optimisées pour la diffusion. Nous obtenons un codage UHD fortement accéléré avec une perte de qualité limitée par rapport à la référence. Nous appliquons également ces restrictions à notre méthode d'inférence (contribution 1.) afin d'évaluer le potentiel de nos travaux dans une implémentation déjà optimisée.

L'ensemble de ces méthodes a été intégré au sein du codeur de référence HEVC. La réduction de complexité et la perte de compression sont systématiquement évaluées. Nous vérifions également la qualité subjective sur écran.

Organisation du document

Première Partie - Description de la norme HEVC :

- Le **chapitre 1** décrit le paradigme général d'un codeur vidéo, les concepts de prédiction, de transformation et codage du résiduel et d'optimisation de la boucle de codage. Une description des outils du standard HEVC est réalisée. Le but est d'apporter les éléments de vocabulaire pour comprendre la suite des développements.
- Le **chapitre 2** décrit premièrement le protocole de mesure de la performance utilisé dans ce manuscrit. Ensuite, une analyse est menée sur les aspects qualité et complexité de HEVC. Elle permet d'appréhender la nouvelle structure de codage introduite par HEVC, son bénéfice sur le taux de compression et sa complexité. Ce chapitre permet de pointer les aspects de la boucle de codage ayant un potentiel de réduction de complexité important.

Deuxième Partie - Réduction de complexité du codage des images Intra :

- Le **chapitre 3** décrit les approches conventionnelles utilisées pour réduire la complexité des codeurs Intra, en HEVC mais aussi dans le précédent standard H.264/AVC.
- Le **chapitre 4** décrit notre méthode de pré-codage de sous-résolution. Après une comparaison entre plusieurs stratégies d'inférence du partitionnement, une méthode de pré-sélection rapide des modes Intra est présentée.
- Le **chapitre 5** décrit notre méthode de pré-analyse du contenu utilisant la variance pour accélérer la procédure de partitionnement.

Troisième Partie - Réduction de complexité du codage des images Inter :

- Le **chapitre 6** décrit les approches conventionnelles appliquées aux images inter-prédites.
- Le **chapitre 7** propose des restrictions de modes de codage HEVC permettant de simplifier significativement la combinatoire de la boucle de codage, contre une perte d'efficacité de compression très limitée.
- Le **chapitre 8** applique les techniques de pré-codage de sous-résolution aux outils de codage des images Inter.

Enfin, la thèse est conclue par un résumé de nos contributions et des perspectives d'études complémentaires. En annexe, les séquences utilisées ainsi que des résultats expérimentaux par séquence sont détaillés. Enfin, les documents produits durant la thèse sont listés en fin de manuscrit.

Première partie

Introduction à la norme HEVC (*High Efficiency Video Coding*)

Chapitre 1

Description du schéma de compression vidéo HEVC

Tous les travaux présentés dans cette thèse sont réalisés dans le cadre normatif d'HEVC. Par conséquent, nous débuterons par une présentation du standard HEVC et de ses outils. Comme tous les codeurs modernes tels que H.262/MPEG-2 et H.264/AVC, HEVC utilise un paradigme de codage hybride, cherchant à exploiter à la fois les redondances spatiales et les redondances temporelles des vidéos. En proposant plus d'outils de codage, HEVC promet une réduction du débit de 50% par rapport à H.264/AVC, à qualité subjective équivalente. Pour atteindre cette performance, un codeur doit réaliser un processus complexe de compétition entre tous ces modes de codage, afin de déterminer la meilleure solution pour coder le signal. Cette notion, essentielle pour la compréhension de ce manuscrit, est développée dans ce chapitre.

1.1 Développement de la norme HEVC en standardisation

A l'heure actuelle, le standard de compression le plus diffusé est H.264/AVC (ou MPEG-4/AVC) [1]. Il a été normalisé en 2003 grâce à la collaboration des groupes *Video Coding Expert Group* (VCEG) de l'ITU-T et MPEG (Moving Picture Expert Group) de l'ISO/IEC au sein du groupe *Joint Video Team* (JVT). Il est largement utilisé dans les supports physiques *BluRay*, par les chaînes de télévisions françaises au format HD de la Télévision Numérique Terrestre (TNT) ainsi que pour l'IP-TV, le satellite, mais aussi la vidéo à la demande (VOD et SVOD) et la vidéo en streaming sur internet.

Face à la diffusion massive des contenus vidéo et des écrans au format HD, au déploiement des services vidéos dans un contexte de mobilité, et à l'émergence des services vidéo au format UHD, la question d'un nouveau standard plus performant s'est posée dès 2009.

Dans la continuité des travaux de standardisation de H.264/AVC, le groupe VCEG a développé un modèle de codage basé sur le modèle H.264/AVC : le JM-KTA (pour *Key Technical Area*) afin de préparer la future génération de *codec*. De nouvelles techniques ont été évaluées, telles que l'estimation de mouvement sous pixel-ique 1/8ème, le filtrage d'interpolation adaptatif (AIF), ou encore la prédiction de mouvement compétitive (MV-Comp) sur laquelle se base l'actuel codage du mouvement AMVP (*Advanced Motion Vector Prediction*) de HEVC (voir section 1.3.5). Les gains mesurés avoisinaient déjà les 20% par rapport à H.264/AVC.

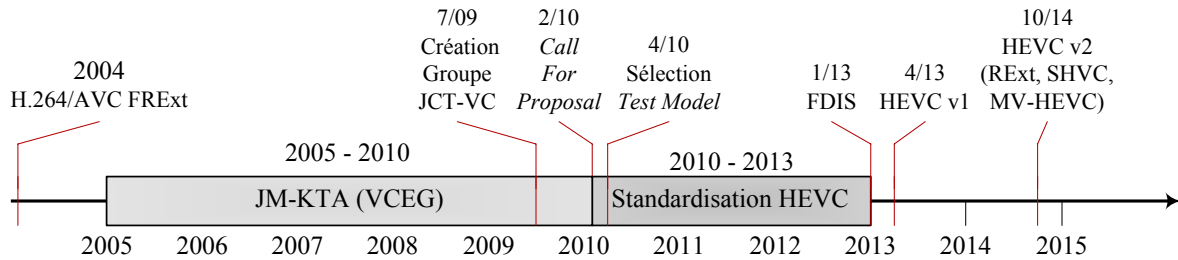


FIGURE 1.1 – Déroulement de la standardisation HEVC.

JCT-VC

Grâce à l'expérience du groupe JVT dans les précédents standards, les deux organismes ITU-T et ISO/IEC ont formé un nouveau groupe d'experts : *Joint Collaborative Team on Video Coding (JCT-VC)* [2]. Un appel à propositions (CfP, pour *Call for Proposals*) est émis par le groupe en 2009 (voir frise en figure 1.1) pour préparer le futur standard [3]. Pour favoriser une adoption rapide du standard, l'objectif était de réduire le débit de 50% par rapport à H.264/AVC à qualité subjective équivalente, contre une augmentation de complexité raisonnable sur le codeur et le décodeur. 27 propositions techniques provenant d'entreprises (BBC, Samsung, HHI...), de regroupements d'entreprises (l'exemple en [4] est une proposition conjointe de France Telecom, NTT DoCoMo, Panasonic et Technicolor) et de laboratoires académiques sont soumises en 2010. 5 propositions se détachent avec 30-35% de gains objectifs. Les meilleures idées et outils sont regroupés au sein d'un modèle logiciel appelé *TMuC (Test Model under Consideration)* qui évoluera par la suite en *Test Model HM*. La période de standardisation aboutit à la soumission du *Final Draft International Standard (FDIS)*, en Janvier 2013.

Publication du standard

En Avril 2013, le standard HEVC version 1 est publié par l'ITU-T, sous le nom *ITU-T Recommendation H.265 : High efficiency video coding* [5], puis l'ISO/IEC le publie sous le nom *ISO/IEC 23008-2 : High efficiency video coding* ou *MPEG-H Part. 2*. Seuls trois profils de codage sont définis dans cette version 1 :

- le *Main Profile* pour les applications les plus courantes, utilisant une profondeur de couleurs 8 bits,
- le *Main Profile 10* utilisant les mêmes outils mais avec 10 bits de profondeur,
- le *Still Picture Profile*, définissant le codage des images fixes (1 seule image Intra).

Tan et al. [6] montrent que l'objectif fixé au début de la standardisation est largement atteint avec le *Main Profile* HEVC. On atteint un gain en compression de 62% par rapport à H.264/AVC à qualité subjective égale sur des contenus HD et de 64% sur des contenus UHD.

La version 2 du standard, publiée en octobre 2014 [7], intègre 19 nouveaux profils correspondants à des *range extensions (RExt)*. Ils adressent certaines applications professionnelles, du profil *Monochrome* au profil *High Throughput 4 :4 :4 16 Intra profile*. Les profils de codage scalable (*HEVC Scalability Extension (SHVC)*) et de codage multi-vues (*Multiview Main*) sont également ajoutés.

La version 3 ([8]) est en cours de développement et intégrera un profil pour le 3D-HEVC et un profil *Screen Content Coding*, dédié au codage de contenus synthétiques. Les performances de compression sont très prometteuses dans ce contexte.

Nos travaux sont appliqués uniquement dans le cadre du codage *Main Profile* mais nos contributions sont compatibles avec les autres profils.

1.2 Représentation du signal vidéo

1.2.1 Structure

Comme l'illustre la figure 1.2, une séquence vidéo est une série d'images capturées à un certain intervalle de temps T_V entre chaque image. Chaque image successive est acquise pendant un intervalle de temps t_A plus court. Typiquement, une fréquence vidéo F_V de 24 Hertz (Hz) est utilisée au cinéma, soit 24 images par seconde nécessaire à donner l'impression du mouvement.

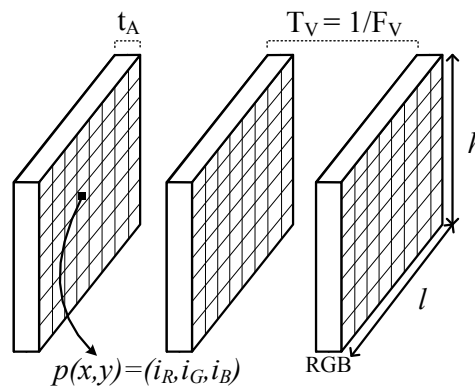


FIGURE 1.2 – Structure d'une vidéo.

Une image numérique est constituée d'un ou plusieurs canaux de couleurs contenant chacun un tableau d'échantillons d'intensités i_i . Le vecteur d'échantillons de chaque canal aux coordonnées (x, y) est appelé *pixel* : $\vec{p}(x, y) = (i_r, i_g, i_b)$. Pour la représentation des couleurs, on utilise les canaux de couleurs rouge, vert et bleu (RGB), soit 3 canaux. Si chaque échantillon est décrit sur 256 niveaux d'intensité (8 bits de profondeur), il est possible de décrire environ 1.67 millions de couleurs différentes (256^3) au travers d'un pixel RGB.

Chaque image est capturée et représentée par un certain nombre de lignes et de colonnes. Le ratio entre largeur l et hauteur h est typiquement de 16/9 pour les formats vidéos récents. Le format *Full HD* contient 1920 colonnes et 1080 lignes (1920x1080), alors que le format *UHD* contient quatre fois plus de pixels (3840x2160). A ne pas confondre, le format 4K, destiné au cinéma dispose de 4096x2048 pixels pour un ratio d'aspect 2 : 1. Historiquement, pour répondre à des problématiques liées aux anciens écrans cathodiques, les vidéos étaient transmises de manière entrelacée. Les lignes paires et impaires de l'image sont séparées en trames, appelées *fields* en anglais. La *bottom field* et la *top field* sont transmises successivement, à des instants temporels différents, ce qui simule une cadence d'affichage double. Actuellement, les formats entrelacés, tels que le 1080i50 (soit des *fields* de 1920x540 pixels transmises à 50Hz), restent les plus utilisés pour la diffusion traditionnelle de la télévision (TNT ou satellite). L'UHD est un format dit progressif : les pixels des images sont tous capturés au même instant. Nos travaux étant basés sur l'UHD ne considèrent donc que les formats progressifs.

1.2.2 Transmission

Un schéma classique de diffusion de la vidéo numérique est décrit dans la figure 1.3.

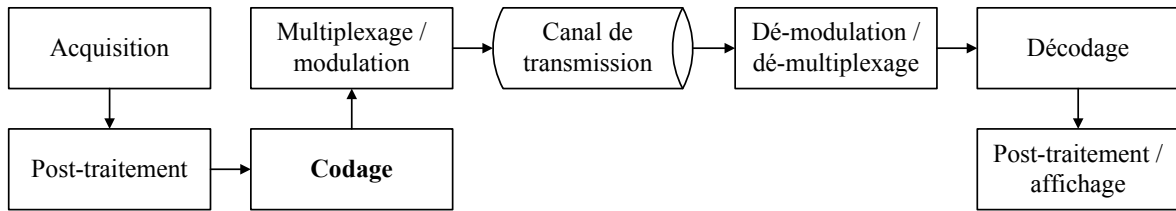


FIGURE 1.3 – Schéma général d'un système de transmission vidéo.

Acquisition : La source vidéo est acquise dans un format numérique brut, non comprimé, ou comprimé visuellement sans dégradations.

Post-traitement : Plusieurs opérations peuvent être réalisées sur le signal : montage, conversion de formats, correction des couleurs et dé-bruitage.

Codage : Le codage consiste à transformer le signal vidéo traité en un flux binaire comprimé pour répondre aux contraintes de débit du canal de transmission, aux contraintes liées à l'affichage en bout de chaîne (formats supportés, capacité de traitement du décodeur, résolution, cadence). Le flux binaire est une séquence de bits appelé *bitstream*. Pour le transport, que ce soit pour HEVC ou les autres standards, le *bitstream* est encapsulé dans des *NAL units*, chacune étant précédée par un *start code*. Les *NALU* peuvent encapsuler un segment de données image (VCL), ou bien des informations de contrôle liées à plusieurs images codées. Le flux de *NALU* produit, par exemple codé en HEVC, doit être conforme au standard HEVC, respecter sa syntaxe, afin d'être compréhensible par n'importe quel décodeur HEVC. Cette étape fait l'objet de nos travaux : nous cherchons à réduire sa complexité dans le cas HEVC.

Multiplexage, modulation : Le multiplexage permet de combiner le flux vidéo avec des données audio, méta-données, et toutes informations nécessaires à la bonne synchronisation des données au moment de la restitution. Le protocole *MPEG-TS (Transport Stream)* est utilisé pour transporter les données audio-vidéo pour la TNT, le câble ou satellite (standards DVB et ATSC). Le signal est aussi transformé au niveau de l'amplitude et/ou de la phase (modulation) afin de transmettre le maximum de données binaires dans un intervalle de fréquence disponible, tout en anticipant les erreurs possibles de transmission.

Canal de transmission : Dans le contexte de la diffusion (*broadcast*), la transmission est uni-directionnelle. Aucun retour de l'utilisateur n'est nécessaire, contrairement aux applications conversationnelles (visio-conférence). Une latence dans la transmission est tout à fait tolérable, ce qui autorise des mécanismes complexes de traitement des pertes de paquets.

Dé-modulation, dé-multiplexage : Le processus inverse est réalisé pour extraire le flux comprimé.

Décodage : Le décodeur, décrit par le standard vidéo, est généralement intégré à l'appareil servant à l'affichage. Il reçoit et traite le *bitstream* afin de reconstruire la vidéo. Dans le cas d'un standard moderne comme HEVC, les images décodées sont chargées au fur et à mesure dans une mémoire tampon (*buffer*). Ces images servent de références au décodage des images suivantes. Ainsi, certains types de codages ne traitent pas les images dans le même ordre que celui de l'affichage (voir la section 1.3.1). La gestion des *buffers* est un aspect important du décodeur. Il doit pouvoir fournir les images décodées à l'affichage de manière à respecter la cadence d'image imposée. De plus, il doit être résilient à toute perte de paquets ou pic de complexité.

Post-traitement/affichage : Certains traitements complémentaires peuvent être réalisés avant l’affichage pour en améliorer le rendu (interpolation de mouvement ou d’échelle). L’affichage fait l’interface avec le système visuel humain (SVH ou *HVS* en anglais).

1.2.3 Conversion couleur et sous-échantillonnage

Le SVH étant plus sensible aux variations de luminosité que de couleur, les algorithmes de compression vidéo convertissent les images RGB dans l’espace couleur YUV. YUV comprend une composante de luminance Y (ou luma) et deux composantes de U et V (ou chroma). Les composantes U et V sont sous-échantillonnées par rapport à Y au moyen d’un rapport Y :U :V spécifique. Plusieurs échantillonnages YUV existent mais en diffusion, le format 4 :2 :0 est principalement utilisé : les composantes U et V sont 2 fois plus petites en largeur et hauteur que Y. Les positions des échantillons de chroma décrites par la figure 1.4 sont utilisées dans la spécification HEVC mais aussi dans les précédents standards.

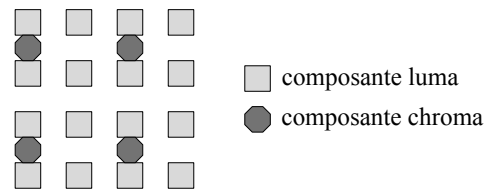


FIGURE 1.4 – Sous-échantillonnage YUV 4 :2 :0.

Selon la Recommandation ITU BT-709 [9], communément utilisée pour les applications HDTV, la conversion RGB vers YUV se fait de la manière suivante : soit E'_R , E'_G et E'_B , les signaux RGB basés sur les primaires couleurs après application d’une fonction de transfert opto-électronique. On définit la valeur de luma E'_Y comme une somme pondérée de E'_R , E'_G et E'_B et les signaux de chroma E'_U et E'_V par rapport à E'_Y , E'_B et E'_R :

$$\begin{aligned} E'_Y &= w_R \cdot E'_R + w_G \cdot E'_G + w_B \cdot E'_B \\ E'_U &= \frac{1}{2 \cdot (1 - w_B)} \cdot (E'_B - E'_Y) \\ E'_V &= \frac{1}{2 \cdot (1 - w_R)} \cdot (E'_R - E'_Y) \end{aligned} \quad (1.1)$$

sachant les constantes définies par BT-709 :

$$\begin{cases} w_R = 0.2126 \\ w_B = 0.0722 \\ w_G = 1 - w_R - w_B \end{cases}$$

$E'_Y = 0$ correspond à une valeur nominale de noir et $E'_Y = 1$, à une valeur nominale de blanc. Pour la compression, le signal doit être quantifié à une certaine précision entière, on parle de *bitdepth*. Typiquement, le *bitdepth* d’un signal de luminance est de 8 bits par pixels, ce qui correspond à 256 niveaux d’intensité par canal. Selon l’Annexe E de HEVC [5], la quantification du signal YUV en valeurs entières sur 8 bits est réalisée de la manière suivante :

$$\begin{aligned} Y &= \text{arrondi}(219 \cdot E'_Y + 16) \\ U &= \text{arrondi}(224 \cdot E'_U + 128) \\ V &= \text{arrondi}(224 \cdot E'_V + 128) \end{aligned} \quad (1.2)$$

1.3 Le schéma de codage hybride de HEVC

La structure de codage hybride, illustrée par la figure 1.5, fait partie intégrante des standards de compression modernes. Le codage hybride signifie que l'on combine deux méthodes dans le même système :

- Une estimation du mouvement est réalisée entre les images de la séquence vidéo. La prédiction des données compensées en mouvement est réalisée à partir des images précédemment codées (prédiction INTER). La prédiction INTRA utilise, quant à elle, les données du voisinage spatial causal pour la prédiction.
- L'erreur de prédiction obtenue est transformée dans le domaine fréquentiel 2D, pour ensuite réaliser une importante compression des données grâce à la quantification des coefficients transformés puis au codage entropique.

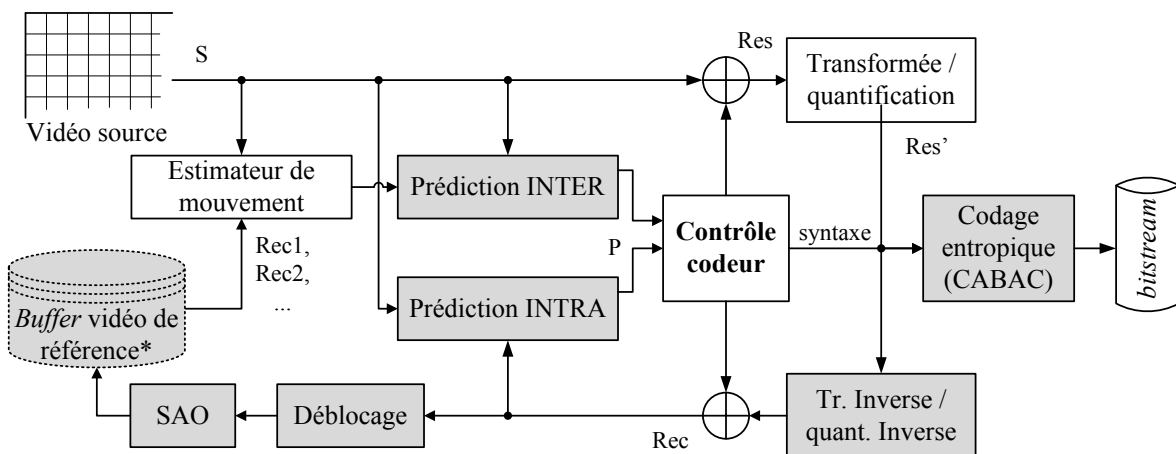


FIGURE 1.5 – Structure de codage hybride HEVC.

1.3.1 Partitionnement initial du signal vidéo

On utilise 3 types d'images dans un codeur hybride : les images I, les images P et les images B :

Les images I (Intra-frames) sont des images de référence, elles ne dépendent d'aucune autre image dans le codage. Elles ne peuvent donc utiliser que des prédictions dans le voisinage spatial causal, en l'occurrence, la prédiction Intra.

Les images P (P-frames) sont des images inter-prédites qui nécessitent des images précédentes de référence pour être reconstituées à partir de vecteurs de mouvement et de résidus. Elles peuvent être elles-mêmes des images de références pour d'autres prédictions.

Les images B (B-frames ou bi-prédicatives) sont aussi des images inter-prédites mais elles peuvent utiliser à la fois les images précédentes et suivantes comme références.

Afin de représenter une séquence vidéo dans un *bitstream*, des structures de codage, appelées *Group Of Pictures* (GOP), définissent l'ordre de codage des images relativement à l'ordre d'affichage, et les dépendances temporelles entre les images. Le délai maximum de décodage de chaque image en dépend.

Les principales structures de codage utilisées en codage vidéo sont :

All Intra (AI) (fig. 1.6 cfg.(a)) : Toutes les images sont codées indépendamment avec des images I. Cette structure de codage sans délai de décodage fait l'objet de la partie II.

Low delay (LD) (fig. 1.6 cfg.(b)) : Cette structure de codage est évaluée avec des images P en uni-prédiction ou avec des images B dans le cas de la bi-prédiction. Le codage n'introduit pas de délai de décodage dans la mesure où seules les images passées sont utilisées pour la prédiction. Ainsi, l'ordre de codage est le même que l'ordre séquentiel de la vidéo. Cette structure est typiquement utilisée dans les applications de conversation vidéo et de contribution, nécessitant une faible latence (*low-delay*). Nous ne traiterons pas cette structure dans ce manuscrit.

Random Access (RA) (fig. 1.6 cfg.(c)) : Cette structure de codage hiérarchique fournit la meilleure performance de compression. Elle impose cependant un délai de décodage. L'ordre de codage diffère de l'ordre d'affichage, comme les images qui précèdent dans l'ordre d'affichage sont codées après. Elle est massivement utilisée dans les applications de diffusion type *broadcast* ou *streaming* vidéo où le canal de transmission est uni-directionnel. Un décodage ne peut démarrer que sur des images I codées indépendamment, généralement insérées une fois par seconde. On les appelle *Random Access Point* (RAP). A noter que selon le niveau hiérarchique de l'image, un incrément de paramètre de quantification –ou *Quantization Parameter*– (QP) est imposé. Augmenter le QP des images ayant des références très proches permet de réduire significativement le débit de ces images sans impacter la qualité visuelle. Cette structure fait l'objet de la partie III.

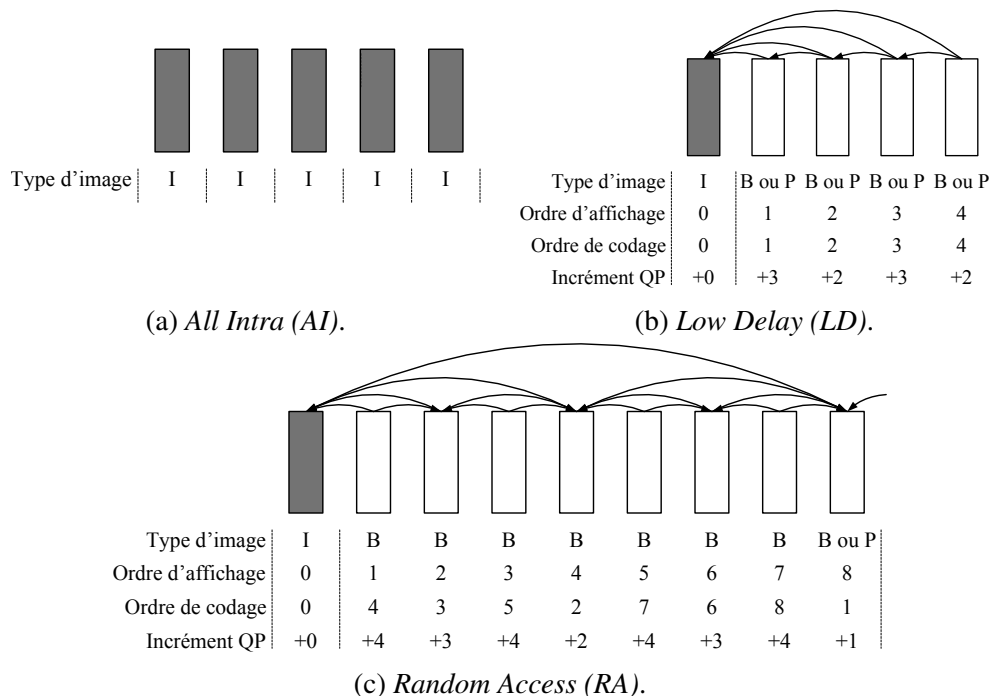


FIGURE 1.6 – Les 3 principales structures de codage utilisées en codage vidéo.

Chaque image codée peut être elle-même subdivisée en une ou plusieurs régions, décodables indépendamment. Généralement, chaque image est constituée d'une seule *slice*, mais selon les besoins de latence, le mode de parallélisme (voir la sec. 1.3.7) ou le dimensionnement des paquets employés lors de la transmission, une image peut être découpée en plusieurs *slices*. En HEVC, une *slice* correspond à une suite consécutive de *Coding Tree Units* (CTU)

dans l'ordre de codage *raster scan* (voir fig. 1.7), ou une suite de *Macroblocks* (MB) en H.264/AVC.

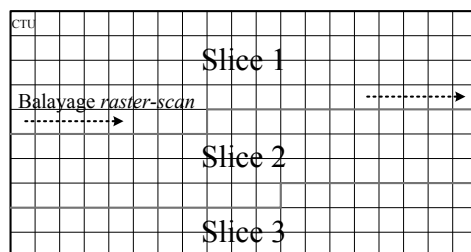


FIGURE 1.7 – Représentation des *slices*.

1.3.2 Structure de codage hiérarchique HEVC

La spécification HEVC distingue le concept de *block* et de *unit*. Alors que le *block* désigne le bloc d'échantillons d'un seul canal couleur (par exemple la luminance Y), le concept de *unit* correspond aux blocs co-localisés dans chaque composante YUV, soit un bloc de pixels.

Dans les précédents standards développés par l'ITU-T et l'ISO/IEC, les images d'une séquence vidéo étaient partitionnées en *Macroblocks* de tailles 16x16 pixels uniquement, ce qui était la taille maximum comme support d'estimation/compensation de mouvement. En HEVC, la plus grande unité de codage est le CTU de taille 64x64 pixels maximum. Le CTU est ensuite décomposé en 3 types d'unités de base :

Coding Units (CU)

Le CTU correspond à la racine d'un arbre quaternaire (*quad-tree*) d'unités de codage –ou *Coding Units*– (CU). Ainsi, le CTU est caractérisé par sa taille (au maximum 64x64 pixels) et sa profondeur maximum de découpe en CU : typiquement 4 profondeurs, sachant que le CU est un bloc 8x8 au minimum. Alors que les CTU sont codés dans l'ordre *raster-scan*, les CU qui le composent sont codés dans l'ordre *z-scan*, tel que représenté dans les fig. 1.8a et 1.8b. Cependant, cette structure flexible est plus exigeante en termes de ressources mémoire.

Prediction Units (PU)

Que ce soit en codage Intra ou Inter, chaque CU est lui-même découpé en une ou plusieurs unités de prédiction –ou *Prediction Units*– (PU) (fig. 1.8c). En cas de prédiction Intra (dans les images I, B ou P), le PU est de taille identique au CU (mode *INTRA_2Nx2N*), excepté pour la taille CU minimum, pouvant être re-partitionnée en 4 prédictions carrées (mode *INTRA_NxN*). En cas de prédiction Inter (images B et P uniquement), 8 types de partitionnements, carrés ou non carrés, peuvent s'appliquer au CU. HEVC introduit 4 nouvelles partitions, dites asymétriques (*Asymmetric Motion Partitions* (AMP)).

Transform Units (TU)

Un CU est également la racine d'un *quad-tree* d'unités de transformée –ou *Transform Units*– (TU). Le TU est une unité de base carrée décrivant les blocs de coefficients résiduels codés dans le *bitstream* (CBF = 1) ou non codés (CBF = 0). Ce *quad-tree* de TU est communément appelé *Residual Quad-Tree* (RQT). Au sein d'un CU, le partitionnement en TU est

indépendant du partitionnement en PU, comme le montre l'exemple de la fig. 1.8d. La taille minimale d'un TU est de 4x4 pixels et la taille maximale est de 32x32 pixels. Les transformées correspondantes sont définies par le standard (voir la section 1.3.6). La profondeur maximale du RQT est fixée au niveau séquence, elle est généralement fixée à 3.

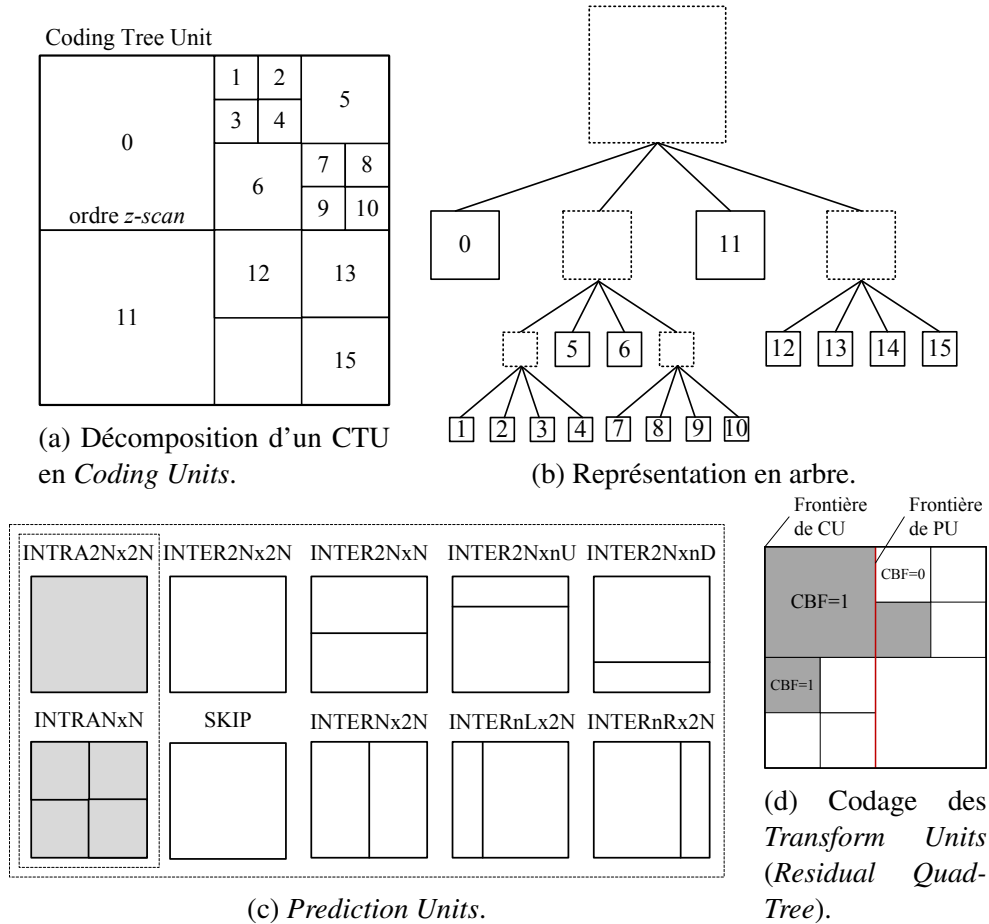


FIGURE 1.8 – Partitionnement d'un CTU en CU, PU et TU.

1.3.3 Généralités sur la prédiction

Le principe de la compression appliquée à la vidéo est d'utiliser les redondances spatiales et temporelles d'une séquence vidéo en réalisant des prédictions. La prédiction consiste à approcher un ensemble de pixels à partir d'informations causales, c'est-à-dire une zone de la séquence vidéo précédemment codée. Ainsi, comme le montre la fig. 1.5, un codeur fonctionne en boucle fermée, c'est-à-dire que les données déjà codées sont disponibles pour traiter les données courantes. C'est pourquoi, le codeur doit contenir en son sein un décodeur local (blocs grisés sur la fig. 1.5) pour obtenir les données causales. La prédiction nous permet d'obtenir un bloc résiduel $Res_c(x, y)$:

$$Res_c(x, y) = S(x, y) - P_c(x, y) \quad (1.3)$$

où $S(x, y)$ est le bloc source à la position (x, y) , $P_c(x, y)$ est le bloc de prédiction, soit une variable aléatoire parmi N variables aléatoires $\{P_{n-1}, P_{n-2}, \dots, P_{n-N}\}$ de l'environnement causal. Si $Res_c(x, y)$ est nul, la prédiction est parfaite. Aucun résidu ne sera transmis, seule la description de ce prédicteur sera transmise (mode, position). L'étape de reconstruction du

signal se traduit par :

$$Rec_c(x, y) = Res_c(x, y) + P_c(x, y) \quad (1.4)$$

Dans le cas d'une compression sans perte, on obtient $Rec_c(x, y) = S(x, y)$. Dans les applications de diffusion, la compression vidéo sans pertes ne permet pas d'atteindre un taux de compression suffisant. Un taux de compression plus élevé n'est possible que par la dégradation du signal résiduel $Res_c(x, y)$. Cette dégradation est contrôlée par une étape de quantification (voir sec. 1.3.6).

1.3.4 Prédiction Intra

Le codage des images I est basé uniquement sur la prédiction Intra. Cependant, la prédiction Intra peut être utilisée au sein des images Inter. Elle utilise les échantillons voisins reconstruits pour prédire les échantillons du bloc de prédiction –ou *Prediction Block*– (PB) courant. Comme le montrent les exemples de prédictions de la fig. 1.9, les échantillons causaux s'étendent sur deux fois la largeur et la hauteur du PU à prédire.

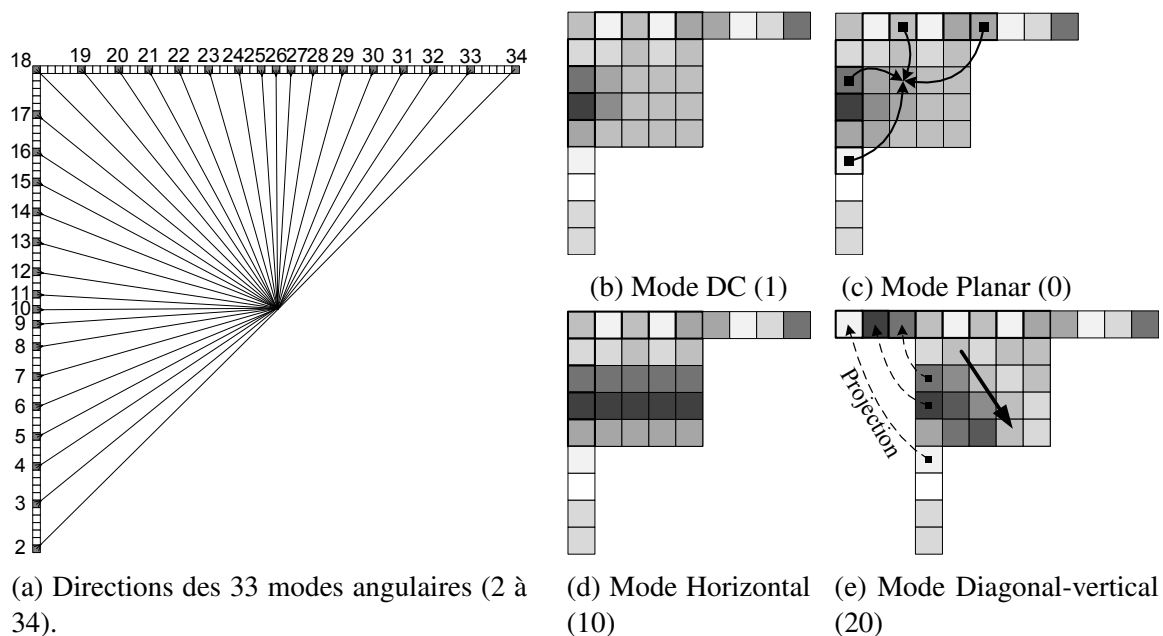


FIGURE 1.9 – Modes de prédictions Intra définis par HEVC pour les PB de luma.

Contrairement aux précédents standards, HEVC comporte plus de modes de prédictions que H.264/AVC, pour décrire plus précisément les structures directionnelles du signal. Ainsi, 35 modes de prédiction Intra sont définis pour la luma, chacun étant un motif de propagation des échantillons causaux dans le PB courant. Le mode DC (fig. 1.9b) constitue la prédiction du bloc courant en réalisant une simple moyenne des échantillons causaux. Le mode Planar réalise des interpolations à partir de 4 échantillons causaux comme représenté par la fig. 1.9c. Enfin, il existe 33 modes directionnels (fig. 1.9a) définissant une direction de propagation des échantillons causaux dans le PB. Le mode horizontal n°10 et le mode vertical-diagonal n°20 sont représentés en fig. 1.9d et 1.9e. Les échantillons causaux sont projetés sur un vecteur de référence 1D avant d'être propagés dans le PU à l'aide d'une interpolation au 1/32ième de pel. En H.264/AVC, seuls 9 modes de prédiction au maximum étaient disponibles, incluant le mode DC, les directions verticales, horizontales et diagonales. Pour prévenir des effets subjectifs indésirables de ces prédictions aux frontières de PU, et également réduire l'énergie du résidu produit, HEVC utilise des filtres adaptatifs à appliquer sur les échantillons causaux

de luma (voir le filtrage sur le mode DC en fig. 1.9b). En ce qui concerne la prédiction des PB de chroma, la signalisation de 5 modes est disponible, dont le DC, le Planar, la direction verticale, horizontale, et la copie du mode utilisé pour la luma (*DMchroma*). Si *DMchroma* décrit un mode redondant avec l'un des 4 autres modes, un mode alternatif diagonal n°34 est utilisé.

Les auteurs de [10] montrent que le codage Intra de HEVC atteint un gain en compression d'environ 22% par rapport à H.264/AVC.

1.3.5 Prédiction Inter et codage du mouvement

Compensation de mouvement

La prédiction Inter utilise les images déjà reconstruites comme références pour réaliser la prédiction compensée en mouvement. La prédiction idéale serait celle qui correspond au « mouvement réel » de l'objet vidéo considéré au cours du temps. La méthode du *flot optique* permet de répondre au problème en calculant le champ de déplacement visuel au cours du temps. Hors, calculer l'erreur de prédiction et la distorsion pour chaque déplacement dans l'image n'est pas envisageable en terme de complexité.

En codage vidéo, l'approche utilisée est celle du *block matching*, qui consiste à trouver le vecteur de mouvement $MV = [\Delta x, \Delta y]$ minimisant l'erreur de prédiction $R_{L,MV}$ entre le PU courant $S(x, y)$ et le bloc prédicteur $P_{L,MV}$ pointé par ce vecteur de mouvement –ou *Motion Vector*– (MV) dans l'image de référence. Cette recherche est réalisée dans une fenêtre de recherche limitée à $\pm L$ (typiquement $L = 64$ dans un codage de référence HEVC) autour du bloc co-localisé $Col(x, y)$ dans l'image de référence. Ainsi, l'important n'est pas de trouver le MV décrivant le « mouvement réel », mais celui étant la meilleure représentation au sens débit-distorsion (voir sec. 1.4.3).

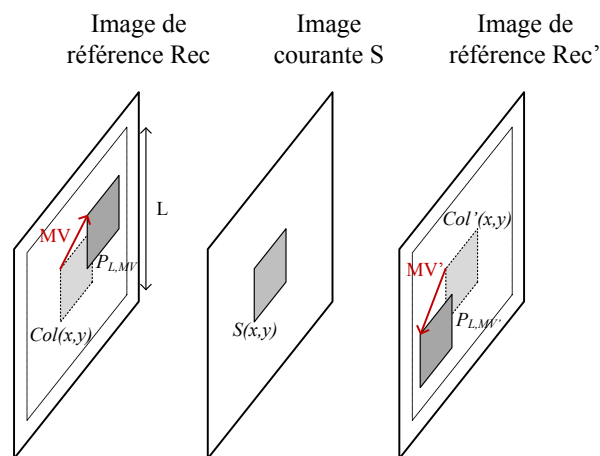


FIGURE 1.10 – Prédiction Inter compensée en mouvement.

La prédiction Inter peut-être décrite avec un seul MV pointant sur une image de référence (uni-prédiction) mais aussi par la combinaison de deux MV indépendants et pointant vers des images différentes (bi-prédiction), comme représenté par la fig. 1.10.

En HEVC, de la même manière qu'avec H.264/AVC, la recherche du meilleur MV est réalisée avec une précision sous-pixellique (jusqu'à 1/4 de pixels), à l'aide de filtres d'interpolations dédiés. Le mouvement est ainsi capturé plus efficacement. La luma utilise des filtres 7 et 8-tap et la chroma utilise un filtre 4-tap.

Le nombre de MV possibles est très important, c'est pourquoi on utilise des algorithmes

de parcours rapides. Dans le modèle de référence, la première étape d'estimation de mouvement sur les positions entières est décomposée de la manière suivante, décrite dans *Test Model* [11] :

1. Choix d'une position de départ
2. Recherche à l'aide d'un parcours optimisé, dit en "diamant".
3. Raffinement autour du meilleur candidat MV obtenu en 2.

Appliqué à du contenu réel, une forte cohérence des MV entre les PU adjacents apparaît. Les PU adjacents ont une forte probabilité d'appartenir au même objet vidéo, et donc de partager des paramètres de mouvement similaires. Par conséquent, les standards récents appliquent un codage du mouvement efficace, qui évite de transmettre des paramètres de mouvement similaires plusieurs fois de suite. Deux modes de transmission des paramètres de mouvement sont proposés avec HEVC : AMVP et MERGE.

Codage du mouvement AMVP

De la même manière que H.264/AVC, HEVC propose un codage du mouvement, appelé *Advanced Motion Vector Prediction* (AMVP), où les MV sont explicitement transmis par un vecteur de mouvement différentiel –ou *Motion Vector Difference*– (MVD). Le MVD (figure 1.11) est le vecteur différence entre le MV réel et le vecteur de mouvement prédicteur –ou *Motion Vector Predictor*– (MVP), obtenu à partir du PU candidat déjà codé dans l'environnement causal :

$$\text{MVD} = \begin{cases} \Delta x - \text{MVP}_x \\ \Delta y - \text{MVP}_y \end{cases}$$

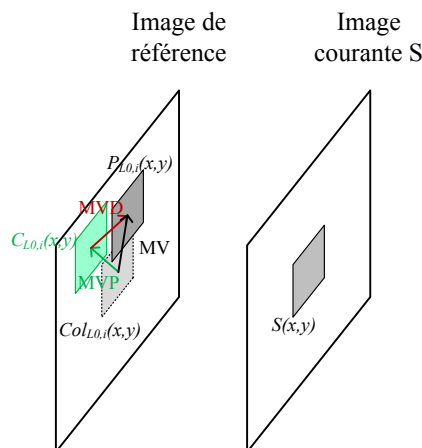


FIGURE 1.11 – Représentation du vecteur de mouvement différentiel (MVD).

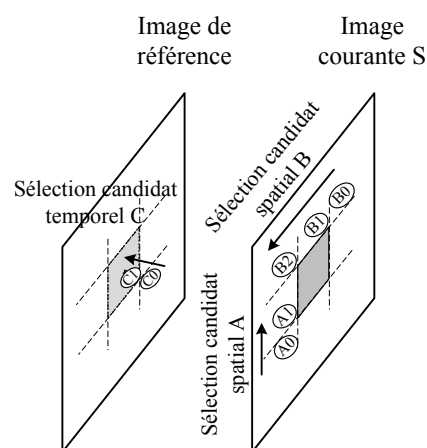


FIGURE 1.12 – Candidats spatiaux et temporels pour la construction de la liste AMVP.

Dans le cas de H.264/AVC, le prédicteur MVP est obtenu en réalisant simplement la médiane des MV des 3 PU adjacents. Avec cette approche implicite, aucune signalisation du MVP n'a besoin d'être transmise.

HEVC introduit avec AMVP, une compétition entre plusieurs candidats, ce qui nécessite de transmettre l'index du meilleur MVP parmi une liste de candidats potentiels. Cette liste est construite de la même manière dans le codeur et le décodeur à partir de nombreux prédicteurs spatiaux et temporels. Au final, seuls 2 éléments sont retenus :

1. Jusqu'à 2 candidats spatiaux sont choisis parmi 5 positions spatiales pré-définies. Comme représenté dans la fig. 1.12, le premier candidat spatial est sélectionné parmi les positions A0,A1, selon leur disponibilité. En parallèle, le deuxième candidat spatial est sélectionné parmi B0,B1,B2.
2. Jusqu'à 1 candidat temporel est choisi dans l'image de référence, à la position en bas à droite du PU co-localisé (position C0 dans la fig. 1.12). Si le PU à la position C0 n'est pas disponible (codé en Intra, ou hors du CTU co-localisé), c'est celui de la position C1 qui est choisi.
3. Après élimination des candidats redondants de la liste (pointant vers la même prédiction), la liste AMVP est soit tronquée, ou complétée par des candidats à vecteur nul, de manière à obtenir une liste AMVP de 2 candidats.

Comme indiqué dans le texte du standard [5], un PU utilisant le mode de codage du mouvement AMVP transmet les éléments de syntaxe suivants :

- soit la liste L0, soit la liste L1, soit les 2 sont utilisées (*inter_pred_idc*)
- l'index de l'image de référence utilisé pour chaque liste (*ref_idx_l0/l1*)
- l'index du candidat AMVP choisi pour chaque liste (*mvp_l0/l1*)
- une syntaxe spécifique pour la transmission du MVD associé à chaque liste (*abs_mvd_greater0/1_flag*, *abs_mvd_minus2*, *mvd_sign_flag*).

Codage du mouvement MERGE

Contrairement au mode AMVP, le mode MERGE n'autorise pas la transmission d'un vecteur différentiel, coûteux dans le processus d'estimation de mouvement. Ainsi, le mode MERGE autorise non plus 2, mais 5 candidats dans sa liste. Comme le codage AMVP, le codage MERGE cherche à déduire les paramètres de mouvement de ceux du voisinage spatio-temporel. Cependant, il y a une différence fondamentale avec le codage AMVP : alors qu'une liste AMVP contient les informations pour une seule liste de référence (L0 ou L1), la liste MERGE contient des candidats décrivant l'information de mouvement pour L0 et/ou L1. La liste de 5 candidats est construite à partir des mêmes positions spatiales et temporelles que AMVP, mais :

- Jusqu'à 4 candidats spatiaux sont choisis parmi 5 positions spatiales. Comme représenté dans la fig.1.13, les candidats sont sélectionnés dans l'ordre A1,B1,B0,A0,B2, selon leur disponibilité.
- Jusqu'à 1 candidat temporel est choisi dans l'image de référence, à la position en bas à droite du PU co-localisé (position C0 dans la fig.1.13). Si la position C0 n'est pas disponible (codée en Intra, ou hors du CTU co-localisé), la position C1 est choisie.
- Après élimination des candidats redondants (pointant vers la même prédiction), des candidats spécifiques peuvent compléter la liste MERGE afin d'obtenir une liste fixe de 5 candidats.

Ainsi, le candidat MERGE contenant toute l'information de mouvement, un PU codé avec MERGE transmet une syntaxe plus légère :

- un élément de syntaxe signalant le mode MERGE (*merge_flag=1*)
- l'index du meilleur candidat MERGE de la liste (*merge_idx*)

Selon [12], l'outil MERGE apporte un gain en taux de compression de 7.3%.

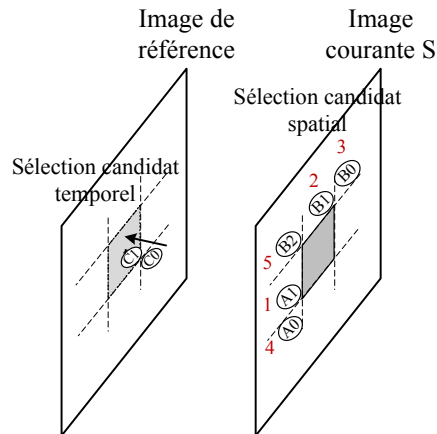


FIGURE 1.13 – Candidats spatiaux et temporels pour la construction de la liste MERGE.

Mode SKIP

Le mode SKIP est un mode MERGE particulier pour lequel aucun résidu n'est transmis. Le mode SKIP n'est possible que dans le cas où le CU courant est composé d'un seul PU (SKIP_2Nx2N). En plus des éléments de syntaxe du mode MERGE, l'élément `skip_flag=1` est transmis au tout début du CU pour indiquer qu'aucun résidu ne sera transmis pour ce CU. Le mode SKIP est massivement utilisé dans les images Inter grâce à sa très grande efficacité pour décrire des régions stationnaires ou homogènes en terme de mouvement.

1.3.6 Transformation et codage du résiduel

Comme représenté sur la fig. 1.5, la reconstruction du signal $Rec_c(x, y)$ dans le codeur exige de connaître $Res'_c(x, y)$, qui correspond au bloc $Res_c(x, y)$ transformé et quantifié. Côté décodeur, $Res'_c(x, y)$ est le résidu extrait du décodage CABAC. $Res_c(x, y)$ est obtenu après quantification et transformation inverses de $Res'_c(x, y)$.

Transformation

Afin de coder et comprimer efficacement le résidu de prédiction $Res_c(x, y)$, on le représente dans le domaine fréquentiel. La transformée en cosinus discrète –ou *Discrete Cosinus Transform*– (DCT) est largement utilisée dans le codage vidéo pour sa capacité à isoler l'information importante [13]. Selon les propriétés de la DCT, l'information importante de basse fréquence se concentre en haut à gauche du bloc transformé, alors que l'information de haute fréquence, moins importante pour l'œil humain, se disperse dans le reste du bloc. La DCT entière utilisée est peu coûteuse en calcul et produit des coefficients entiers. A l'instar des TU, les transformées DCT sont déclinées en tailles 4x4, 8x8, 16x16 et 32x32 [14]. HEVC introduit une transformée en sinus discrète –ou *Discrete Sinus Transform*– (DST) dans le cas spécifique des TU Intra de taille 4x4.

La transformation n'est pas un procédé de compression en tant que tel, mais un changement de représentation : le procédé est ainsi réversible, sans pertes.

Quantification

Suite à la transformation, les coefficients produits subissent une étape de quantification qui va cette fois introduire une erreur de quantification. L'objectif est de faire pas-

ser à 0 les coefficients peu importants. On parle alors de compression avec pertes où $Rec_c(x, y) \neq S(x, y)$. HEVC utilise une quantification scalaire contrôlée par un paramètre de quantification, appelé QP, prenant une valeur entre 0 et 51. Le QP agit sur le pas de quantification avec un rapport quasi-logarithmique (un incrément de 6 du QP double le pas de quantification, et donc réduit l'information résiduelle).

Dans le codeur de référence HEVC, une méthode appelée *Rate-Distortion Optimized Quantization* (RDOQ) [15] permet d'ajuster les niveaux de quantification des coefficients transformés au niveau TU. Cette méthode n'est pas un outil normatif mais permet d'obtenir un gain en compression d'environ 4%.

Codage entropique

HEVC spécifie une seule méthode de codage entropique : le *Context-Adaptive Binary Arithmetic Coding* (CABAC). C'est un outil de compression sans pertes utilisant les propriétés statistiques du signal pour représenter les symboles selon leur niveau de probabilité. Les symboles les plus représentés sont alors codés avec le nombre de *bits* le plus faible. Le codage CABAC est réalisé après l'étape de quantification, quand le signal a été réduit à un ensemble d'éléments de syntaxe décrivant : les propriétés des CTU (structure, quantification et paramètres SAO), des PU (modes de prédiction Intra ou paramètres de mouvement) et des TU (structure RQT et coefficients résiduels transformés et quantifiés).

La première étape du codeur entropique consiste à adapter les éléments de syntaxe non-binaires en symboles binaires (*bins*). Plusieurs techniques de "binarisation" sont utilisées : *Unary* (*U*), *Truncated Unary* (*TU*), *Exp-Golomb* (*EGk*), et *Fixed Length* (*FL*).

Certains éléments de syntaxe (peu significatifs ou présentant une entropie trop forte) ne sont pas soumis à la modélisation de contexte (mode *bypass*). Pour les autres, la probabilité de chaque (*bin*) du train binaire produit est déterminée selon un *context model*, dépendant des observations faites dans le voisinage proche. A chaque *bin* traité, le *context model* est actualisé pour s'adapter à la distribution de probabilités observée au cours du temps.

Enfin, le codeur arithmétique permet de comprimer les *bins* selon leur probabilité. En partant d'un intervalle donné, la probabilité de chaque *bin* est utilisée pour récursivement affiner l'intervalle et calculer un code binaire d'autant plus court que la probabilité du *bin* est élevé. Dans le cas *bypass*, le codage arithmétique est réalisé simplement avec des probabilités identiques de 50%, pour les valeurs 0 et 1.

D'après les auteurs de [16], le CABAC apporterait 5 à 9% de gain en compression par rapport à un codeur entropique *Context-Adaptive Variable Length Coding* (CAVLC) de H.264/AVC.

1.3.7 Autres outils HEVC

En plus des outils dédiés à la prédiction et au codage du résiduel, HEVC utilise des techniques de filtrage pour améliorer la qualité objective et subjective du signal reconstruit $Rec_c(x, y)$. C'est donc un signal reconstruit et filtré qui est présenté au *buffer* d'images. Nous présenterons également les outils dédiés au traitement parallèle de HEVC.

Deblocking filter

Le filtre anti-blocs ou *-deblocking filter-*, déjà utilisé dans H.264/AVC, permet de réduire les "effets de blocs" à la reconstruction du signal, en lissant de manière adaptative les échantillons de part et d'autres des frontières de PU et TU. Ce filtre a été légèrement modifié pour tenir compte de la nouvelle structure hiérarchique de HEVC et être plus adapté au traitement des vidéos de haute résolution. Sa complexité a été largement réduite car il n'est plus

réalisé sur les blocs de tailles 4x4 mais seulement sur une granularité 8x8. Selon [17], ce filtre permet d'améliorer le taux de compression de 1.2% à 3.6%, mais le gain subjectif peut être supérieur lorsque le contenu est fortement quantifié (bas débit) : les effets aux frontières de blocs peuvent s'avérer visuellement gênants.

SAO

Après application du *deblocking filter* sur le signal reconstruit, HEVC introduit un nouveau filtre : le *Sample Adaptive Offset* (SAO). Il permet de réduire la distorsion du bloc reconstruit en appliquant des corrections (*offsets*) sur les valeurs des pixels au sein d'un CTU. Une classification de l'activité directionnelle et des intensités du bloc est réalisée. Ainsi, 3 modes de transmission des paramètres SAO pour le CTU courant sont disponibles :

sao_type_idx=0 : La distorsion est optimale sans appliquer de correction aux pixels du CTU.

***sao_type_idx=1*, ou mode *Band Offset* (BO)** : Il s'agit d'appliquer des *offsets* sur les échantillons correspondant à 4 intervalles d'intensité (bandes) parmi 32 (en 8 bits, il y a 32 bandes de largeur 8). Le codeur transmet l'index de la première bande et 4 *offsets* à appliquer respectivement sur les échantillons correspondants aux 4 bandes. Concrètement, le mode BO permet de réduire les effets de bandes dus à la quantification sur les zones homogènes présentant des dégradés d'intensité.

***sao_type_idx=2*, ou mode *Edge Offset* (EO)** : Il s'agit d'appliquer des corrections aux échantillons selon leur activité directionnelle, mesurée par les gradients appliqués par rapport aux 8 échantillons connexes. Selon la direction choisie *sao_eo_class* (verticale, horizontale, diagonale 135° ou diagonale 45°), le codeur transmet ainsi *sao_eo_class* et 4 *offsets* à appliquer sur les échantillons selon le gradient mesuré dans la direction choisie. Concrètement, le mode EO permet d'améliorer la précision des zones de contours.

Selon [18], la SAO permet d'améliorer le taux de compression de 1.7% en RA, et de 9.2% avec la configuration de codage LD.

Outils pour le parallélisme

Lors du développement du standard HEVC, un effort particulier a été fait sur la prise en charge du parallélisme, pour pouvoir adresser les grandes résolutions à l'aide d'architectures de calcul multi-cœurs, multi-processeurs. Ainsi, HEVC va plus loin que le simple concept de *slices* (fig. 1.7) et introduit les notions de *tiles*, de *Wavefront Parallel Processing* (WPP) et de *dependent slices*.

Les *tiles* (fig. 1.14a) sont des régions décodables indépendamment, de la même manière que les *slices*, sauf qu'elles correspondent à un partitionnement rectangulaire de l'image, non contraint par l'ordre *raster-scan* des CTU.

Le WPP (fig. 1.14b) est un concept indépendant des *slices*. Il permet de paralléliser le codage ou le décodage d'une image au niveau des lignes de CTU. Dès que les 2 premiers CTU d'une rangée ont été codés, les contextes du codeur entropique CABAC sont disponibles pour que la rangée du dessous puisse démarrer un processus en parallèle. Ce concept est très utile quand il s'agit d'implémenter le codeur/décodeur sur des architectures massivement parallèles.

Le fonctionnement des *dependent slices* (fig. 1.14c) est similaire au WPP, sauf que les rangées de CTU sont des *slices*, transportées dans des *NALU* séparées. Cela permet ainsi d'adresser des applications à très faible latence (*ultra low delay*).

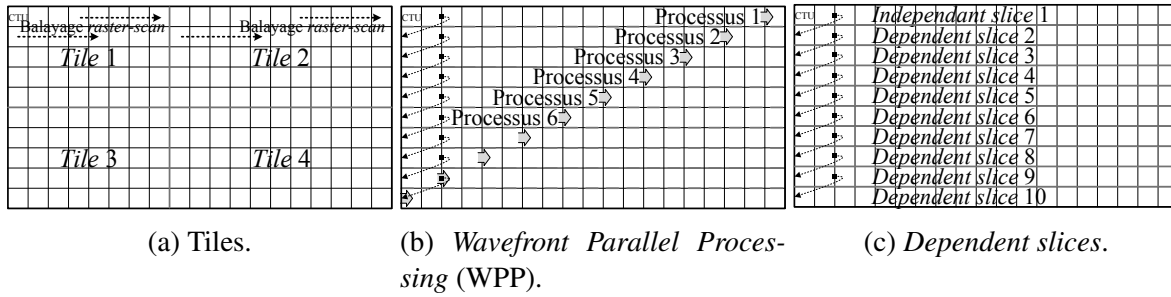


FIGURE 1.14 – Outils pour le parallélisme.

1.4 Contrôle du codeur et optimisation.

1.4.1 Optimisation au sens débit-distorsion

Comme tous les standards produits par l'ITU-T et l'ISO-IEC, seules la syntaxe du *bitstream* et la manière de le décoder sont imposées, ce qui ne garantit en aucun cas le taux de compression promis par le standard. L'objectif d'un codeur est avant tout de mettre en compétition les différents modes de codage proposés par le standard afin d'obtenir, au final, un ensemble d'éléments de syntaxe décrivant la solution optimale pour comprimer le signal d'entrée S . On cherche à minimiser une distorsion $D_c(S(x, y), Rec_c(x, y))$ entre le bloc $S(x, y)$ et le signal reconstruit $Rec_c(x, y)$, obtenu après prédiction, transformation, quantification, quantification inverse et transformation inverse, le tout en respectant une contrainte sur le débit codé dans la *bitstream* :

$$\operatorname{argmin}_{c \in C} D_c(S(x, y), Rec_c(x, y)) \text{ contraint par } R_c \leq R_E \quad (1.5)$$

avec c , le vecteur de paramètres décrivant les modes de codage appliqués au bloc $S(x, y)$, parmi l'ensemble des options de codage C disponibles pour ce bloc. R_c représente le nombre de bits produits avec les décisions c et R_E est la contrainte fixée sur le débit.

Concrètement, ce problème est résolu dans les codeurs par l'optimisation Lagrangienne [19, 20]. Cette fois, le terme de distorsion D_c est pondéré avec le terme de débit R_c par le multiplicateur de Lagrange λ . Ainsi, la solution J^* à ce problème de minimisation associant débit et distorsion s'écrit :

$$J^* = \operatorname{argmin}_{c \in C} D_c(S(x, y), Rec_c(x, y)) + \lambda \cdot R_c \quad (1.6)$$

avec λ , le multiplicateur de Lagrange qui est dépendant du paramètre QP utilisé, du type de décision évalué ainsi que de la structure de codage GOP.

L'optimisation Lagrangienne s'applique à tous les niveaux de décisions d'un codeur HEVC :

- à la construction du *quad-tree* de CU dans un CTU,
- au choix de la prédiction Intra ou Inter, du type de partitionnement dans le CU,
- au choix du mode de prédiction Intra,
- au choix de la liste de référence et des images de référence,
- au choix du type de codage de mouvement (AMVP ou MERGE), et intrinsèquement des meilleurs paramètres,
- à la sélection des meilleures tailles de transformées, de la transmission des résidus ou non,
- au choix du niveau de quantification.

1.4.2 Estimation des modes de codage

Différentes mesures de distorsion s'appliquent selon le type de décision à évaluer. Dans le modèle de référence HEVC [11], la plupart des mesures de distorsion sont réalisées par la somme des différences au carré –ou *Sum of Squared Differences*– (SSD) entre les échantillons $s(n, m)$ de la source S et les échantillons $rec(n, m)$ du bloc reconstruit Rec :

$$D_{SSD} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} |s(n, m) - rec(n, m)|^2 \quad (1.7)$$

Ainsi, pour choisir les meilleurs paramètres de codage d'un CU, on veut minimiser une fonction de coût J_{MODE} telle que :

$$J_{MODE} = D_{SSD} + \lambda_{MODE} \cdot R_{MODE} \quad (1.8)$$

Le multiplicateur de Lagrange λ_{MODE} est défini par :

$$\lambda_{MODE} = \alpha \cdot W_K \cdot 2^{((QP-12)/3.0)} \quad (1.9)$$

sachant que :

$$\left\{ \begin{array}{l} \alpha = 1 \text{ pour les images non utilisées comme références} \\ \text{(donc valable pour la configuration AI uniquement),} \\ \alpha = 1.0 - (0.05 \times \text{nombre d'images B}) \text{ pour les images référencées} \\ \text{(configurations LD et RA).} \end{array} \right.$$

avec W_K , un facteur dépendant de la configuration de codage (AI, LD ou RA) et de la position hiérarchique de l'image. Typiquement, pour une image I, $W_K = 0.57$ et pour une image B8 (soit la huitième image dans l'ordre d'affichage, et la première image dans l'ordre de codage, dans un GOP RA), $W_K = 0.442$.

R_{MODE} est le nombre de *bits* nécessaires pour coder la syntaxe et les résidus associés au CU.

1.4.3 Estimation des paramètres de mouvement

Pour l'estimation de mouvement avec précision entière, la distorsion D_{SAD} est mesurée comme une somme des différences absolues –ou *Sum of Absolute Differences*– (SAD) entre le bloc courant et le bloc de référence $P_{L,MV}$ décalé de $MV = [\Delta x, \Delta y]$ dans l'image de référence L :

$$D_{SAD} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} |s(n, m) - p_{L,MV}(n, m)| \quad (1.10)$$

Comme décrit en 1.3.5, une recherche rapide du meilleur vecteur MV est réalisée dans le codeur, évitant une recherche exhaustive dans la fenêtre de recherche.

En ce qui concerne le raffinement à la précision fractionnaire (au demi puis au quart de pixel), il a été montré que la SAD fonctionne mieux dans le domaine transformé de Hadamard, qui est une approximation de la DCT : on parle cette fois de *Sum of Absolute Transformed Differences* (SATD). Le bloc résiduel R est d'abord transformé par la matrice de Hadamard, puis la distorsion D_{SATD} est calculée :

$$R = H.(S(x, y) - P_{L,MV}(x, y)).H \quad (1.11)$$

$$D_{SATD} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} |r(n, m)| \quad (1.12)$$

La matrice de Hadamard H_N est orthogonale, de taille $2^N \times 2^N$ et constituée de valeurs ± 1 . Elle peut être récursivement déduite comme suit :

$$H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot (\sqrt{2})^{-2}, H_N = \begin{bmatrix} H_{N-1} & H_{N-1} \\ H_{N-1} & -H_{N-1} \end{bmatrix} \cdot (\sqrt{2})^{-N} \quad (1.13)$$

Le problème associé à l'estimation de mouvement consiste à minimiser la fonction J_{PRED} pour chaque liste L, premièrement en simple précision où le terme de distorsion D_{PRED} est une SAD, puis en précision fractionnaire, où D_{PRED} est une SATD :

$$J_{\text{PRED}} = D_{\text{PRED}} + \lambda_{\text{PRED}} \cdot R_{\text{PRED}} \quad (1.14)$$

sachant que : $\lambda_{\text{PRED}} = \sqrt{\lambda_{\text{MODE}}}$, avec λ_{MODE} défini en eq. 1.9, et que R_{PRED} est une estimation du nombre de bits nécessaires pour coder les paramètres de mouvement. Cette fonction de coût ne nécessite pas un codage puis un décodage complet, mais seulement de réaliser l'étape de prédiction, ce qui permet d'obtenir une estimation du coût débit-distorsion avec un faible coût calculatoire.

La bi-prédiction est, quant à elle, un problème d'optimisation jointe. L'algorithme est initialisé avec les meilleurs paramètres de mouvement obtenus pour la liste L0 et la liste L1, indépendamment. De manière itérative, un raffinement est réalisé, en terme de coût sur les paramètres de L0, en gardant ceux de L1 constant, puis inversement jusqu'à convergence.

1.4.4 Estimation des modes de prédiction Intra

Le modèle de référence HEVC utilise un schéma de pré-sélection rapide du meilleur mode de prédiction Intra de luminance [21]. Les 3 étapes sont représentées dans la fig. 1.15.

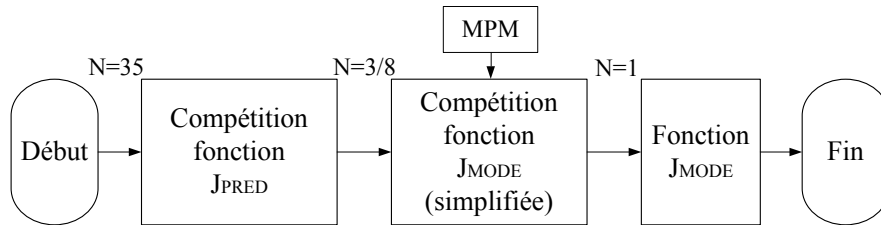


FIGURE 1.15 – Pré-sélection rapide du meilleur mode Intra employée dans le modèle de référence HEVC.

1. La première étape compare les 35 modes HEVC à l'aide de la fonction de coût à faible complexité J_{PRED} (eq. 1.14). D_{PRED} est la SATD (eq. 1.12) entre le PU source $S(x, y)$ et le bloc de prédiction $P(x, y)$ généré par le mode Intra.
2. Dans la seconde étape, la liste des N meilleurs candidats de la liste obtenue à l'étape 1 ($N = 3$ ou 8 selon la taille de PU) est complétée par les modes les plus probables –ou *Most Probable Modes*– (MPM). Les MPM sont composés à partir des modes Intra déjà codés dans les PU adjacents [22]. Cette fois, la fonction de coût J_{MODE} (eq. 1.8) est utilisée, en considérant uniquement la taille de TU égale à la taille CU.
3. La dernière étape évalue le meilleur mode obtenu dans la deuxième étape avec J_{MODE} et la profondeur du RQT standard (3 profondeurs en général), ce qui permet d'obtenir le coût optimal du PU au sens débit-distorsion.

Cette pré-sélection basée sur la SATD permet de réduire fortement le coût calculatoire de la décision Intra. Nos résultats de simulation présentés dans le tableau 1.1, montrent que ce schéma réduit la complexité de codage de 83.3% par rapport à une décision Intra entièrement

basée sur J_{MODE} pour les 35 modes de luma, et cela avec une perte de compression de 0.84% seulement en UHD. Les conditions d'expérimentation sont détaillées en section 2.1.

D'autres algorithmes rapides sont proposés dans la littérature. Ils sont présentés dans la section 3.2. Un algorithme de pré-sélection rapide, s'appuyant sur une passe de pré-codage de sous-résolution, est également développé dans ce manuscrit en section 4.3.

Séquences	<i>All Intra (AI)</i>	
	BD-rate	ΔT
<i>EBU_UHD</i>	0.84%	83.3%
<i>ClasseB_HD</i>	0.97%	82.7%

TABLEAU 1.1 – Performance de la pré-sélection rapide des modes Intra HEVC.

1.5 Conclusion

Ce chapitre explique les concepts fondamentaux d'un codeur vidéo et donne un état des lieux des outils de codage HEVC. Publié en 2013, HEVC a été développé au sein du groupe JCT-VC pour compresser les nouveaux formats vidéo en réduisant le débit de transmission par 2 par rapport au standard précédent.

Nous avons expliqué qu'un codeur vidéo doit résoudre un problème d'optimisation entre débit et distorsion. Au sein d'une boucle de codage, les modes de codage proposés par le standard sont mis en compétition afin de trouver la solution optimale pour le codage du *bits-tream*. HEVC augmente significativement le nombre de modes par rapport à H.264/AVC, ce qui améliore le taux de compression, mais cela devient un problème trop complexe pour une application temps réel (transcodage, *Live broadcast*,...) et de calcul contraint (circuits embarqués, sur batterie, à faible coût ...). La seule réponse à ce problème est de simplifier au maximum le procédé en perdant le moins possible en qualité de codage (ou taux de compression). C'est l'objectif de nos travaux.

Le chapitre suivant propose une analyse des aspects performance et complexité du standard HEVC, tout en pointant les opportunités de réduction de complexité.

Chapitre 2

Performance et complexité du codage HEVC

Dans ce chapitre, la première section 2.1 détaille le protocole utilisé pour générer les résultats de ce manuscrit. En particulier, nous expliquons les notions de réduction de complexité et de qualité de codage, sachant que l'objectif de cette thèse est de trouver le meilleur compromis entre les deux.

HEVC introduit une structure de codage plus flexible par rapport à H.264/AVC pour améliorer l'efficacité de codage. Grâce à des blocs de plus grandes tailles, et une augmentation du nombre d'outils permettant de prédire le signal, nous démontrerons l'efficacité de cette structure pour augmenter le taux de compression, notamment quand la résolution spatiale augmente.

Enfin, nous verrons que la recherche du partitionnement CU optimal a une très forte complexité calculatoire. Nous démontrerons que restreindre la compétition de modes aux tailles de blocs les plus probables permettrait de réduire significativement la complexité.

2.1 Méthodologie utilisée

Pour tous les résultats expérimentaux réalisés dans ce manuscrit, plusieurs jeux de séquences vidéos ont été utilisés, avec les configurations *All Intra* (AI) et *Random Access* (RA), ce qui correspond respectivement aux cas d'études des parties II et III.

Tout au long de ce manuscrit, nous montrerons la performance des algorithmes comme un compromis entre réduction de complexité et perte de compression par rapport à un codage de référence HEVC, réalisé avec le *Test Model HM*, version 11.0 [11].

2.1.1 Configurations de codage

Nous utilisons les configurations AI et RA de base ayant les critères suivants :

- Taille CTU : 64x64 pixels, profondeur de 4 (taille CU minimum : 8x8),
- Profondeur TU max. : 3,
- Transformées de tailles 4x4 à 32x32,
- RDOQ activée, filtre SAO activé,
- WPP activé,
- 4 codages par séquences à QP fixe : QP_I = 22, 27, 32, 37.

La configuration AI a pour spécificité :

- QP = QPI identique pour toutes les images,
- Taille de GOP : 1.

La configuration RA a pour spécificité :

- QP = QPI + 1 pour les images B de niveau hiérarchique 1, QP = QPI + 1 pour le niveau 2 et QP = QPI + 3 pour le niveau 3 (voir section 1.6(c)),
- Une image I (RAP) par seconde (soit une période Intra de 48 images pour une fréquence vidéo de 50Hz, 64 images pour 60Hz),
- Références multiples (jusqu'à 4 par image), références inter-GOP autorisées, sauf au travers des images RAP,
- Partitions Inter non carrées et asymétriques AMP activées,
- Liste MERGE de 5 candidats,
- Fenêtre de recherche de l'estimateur de mouvement $L = \pm 64$.

2.1.2 Séquences utilisées

Nos travaux s'adressent principalement aux formats vidéos UHD. Nous avons ainsi sélectionné un jeu de 10 vidéos sources fournies par l'EBU [23] (voir le lien : <https://tech.ebu.ch/Jahia/site/tech/cache/offonce/testsequences/uhd-1>).

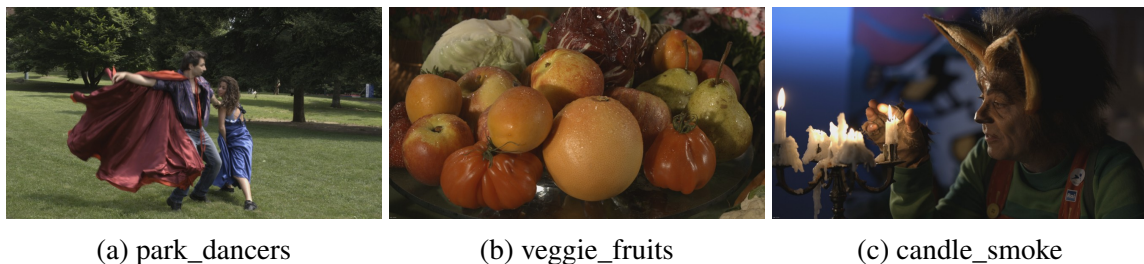


FIGURE 2.1 – Extraits de quelques séquences EBU.

Une description détaillée de ce jeu de séquences est présentée en Annexe A.1. Ces séquences représentent des scènes variées en termes de contenus, de caractéristiques spatiales et temporelles. La figure 2.2 le montre en représentant graphiquement le SI-TI de chaque séquence. La mesure SI-TI, définie en [24], permet de caractériser simplement l'information spatiale (SI) et l'information temporelle (TI) perçues d'une vidéo.

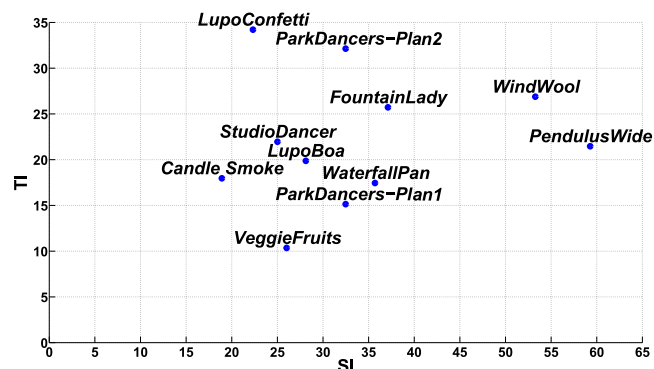


FIGURE 2.2 – SI-TI des séquences *EBU-UHD*.

Les séquences ont été capturées puis traitées dans les mêmes conditions :

- au format YUV 4 :2 :0 avec un *bitdepth* de 8 bits,
- résolution 3840x2160 pixels,
- 750 images chacune,
- 15 secondes chacune à une cadence de 50Hz,
- espace colorimétrique Rec.709.

Nous appellerons ce jeu de séquences : **EBU-UHD**.

Pour évaluer la performance de nos algorithmes au format HD, nous utiliserons le jeu de séquences constituant la classe B des séquences de test JCT-VC. La classe B est utilisée pour évaluer la performance de codage HDTV. Elle est d'ailleurs très utilisée par les contributeurs de l'état de l'art sur la réduction de complexité.

- YUV 4 :2 :0 avec un *bitdepth* de 8 bits,
- résolution 1920x1080 pixels,
- 10 secondes chacune à des cadences de 24, 50 et 60Hz,
- espace colorimétrique Rec.709.

Nous appellerons ce jeu de séquences : **ClasseB-HD**.

Afin d'étendre nos tests, notamment dans les expérimentations utilisant un apprentissage, nous utiliserons d'autres jeux de séquences au format UHD : cinq séquences SVT (**SVT-UHD**), décrites dans le document [25] et quinze séquences SJTU (**SJTU-UHD**), décrites dans le document [26].

2.1.3 Réduction de complexité

Afin de mesurer précisément la réduction de complexité obtenue, le modèle de référence et la version modifiée sont exécutés sur la même plateforme de calcul. Le codeur ne contient aucune forme de parallélisation ou d'optimisation dédiée à une architecture de calcul spécifique. Chaque exécution est donc purement séquentielle et allouée à un cœur calcul indépendant.

Ainsi, le temps d'exécution des processus permet de mesurer la réduction de complexité associée à l'algorithme, par rapport au codage de référence :

$$\Delta T = \left(1 - \frac{T_{\text{ALG}}}{T_{\text{REF}}}\right) \times 100\% \quad (2.1)$$

avec T_{REF} étant le temps de calcul nécessaire pour coder les séquences avec le codeur de référence et T_{ALG} , le temps de calcul pour le codeur modifié. Un ΔT positif indique une réduction de complexité. $\Delta T = 50\%$ est équivalent à une réduction de la complexité d'un facteur 2. Nos mesures sont réalisées sur un jeu de N séquences, avec 4 QP différents. Ainsi, la somme des temps de calculs de $4N$ codages est considérée pour calculer T_{REF} et pour T_{ALG} .

2.1.4 Qualité de codage

Pour comparer objectivement la qualité de codage entre 2 *bitstreams*, on peut représenter d'un côté la variation du débit ($\Delta \text{bitrate}$) et de l'autre, la variation ΔPSNR du rapport signal sur bruit –ou *Peak Signal to Noise Ratio*– (PSNR). Le ΔPSNR est défini par :

$$\Delta \text{PSNR} = \text{PSNR}_{\text{ALG}} - \text{PSNR}_{\text{REF}} \quad (2.2)$$

$$\text{PSNR} = \frac{1}{P} \sum_{p=0}^P 10 \log_{10} \frac{(2^{\text{bitdepth}} - 1)^2 \cdot L \cdot H}{\text{SSE}_p} \quad (2.3)$$

avec p , l'index de l'image dans la séquence de P images, SSE_p , la somme des différences au carré entre l'image p source et son image reconstruite. L et H sont respectivement la largeur et la hauteur de l'image. Bjøntegaard [27] propose une métrique permettant de combiner cette variation du débit et du PSNR. Cependant, elle nécessite plusieurs points de fonctionnement de débit, c'est pourquoi dans nos travaux, nous réalisons systématiquement 4 codages par séquences à QP = 22, 27, 32, 37. Ces valeurs de QP produisent une gamme de débit représentative des applications de diffusion. Les 2 courbes débit/qualité (correspondants aux 2 codeurs comparés) sont obtenues par interpolation du 3ème ordre :

$$\text{psnr} = a + b \cdot \text{bitrate} + c \cdot \text{bitrate}^2 + d \cdot \text{bitrate}^3 \quad (2.4)$$

La métrique de Bjøntegaard consiste à mesurer l'aire entre ces 2 courbes. Selon le sens choisi, on obtient la mesure *BD-PSNR* (figure 2.3a), qui est la différence de PSNR moyenne (en dB) sur l'intervalle de débit, ou la mesure *BD-rate* (figure 2.3b), correspondant à la différence de débit moyenne (en %) sur l'intervalle de PSNR.

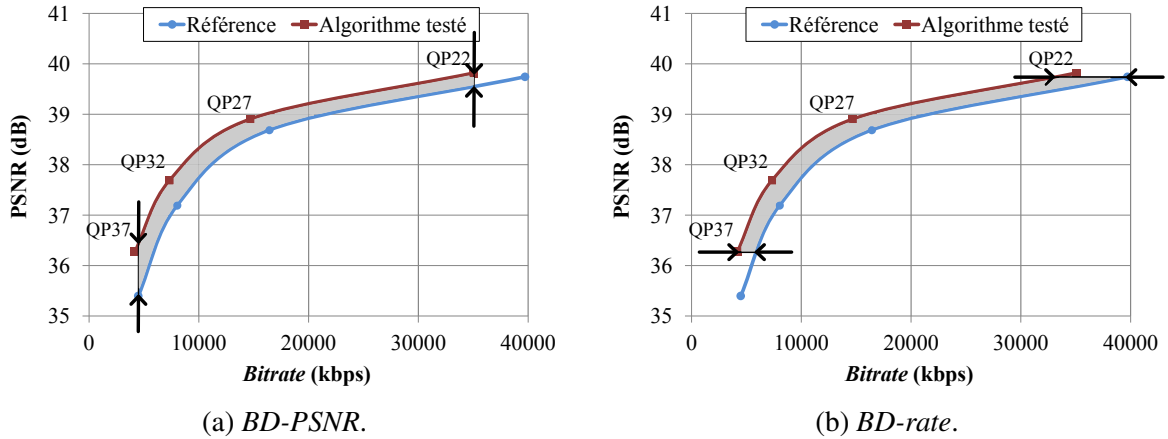


FIGURE 2.3 – Métrique de Bjøntegaard déterminée à partir de 2 courbes débit/PSNR.

Nous utiliserons le *BD-rate* dans le reste du manuscrit. Un *BD-rate* > 0% correspond à une perte de compression par rapport à la référence, un *BD-rate* < 0% signifie un gain de compression. Comme la mesure PSNR est calculée séparément pour chaque canal reconstruit Y, U ou V, le *BD-rate* est calculé à partir d'un PSNR pondéré entre les 3 canaux. Cette pondération est typiquement utilisée par les experts du domaine :

$$\text{PSNR} = \frac{6 \times \text{PSNR}_Y + \text{PSNR}_U + \text{PSNR}_V}{8} \quad (2.5)$$

La mesure de qualité étant réalisée sur un jeu de N séquences, la moyenne des *BD-rate* mesurés pour les N séquences est retenue dans les résultats de ce manuscrit. Le *BD-rate* est un bon indicateur de la qualité objective mais doit être complétée par une mesure subjective (observation d'utilisateurs experts et non-experts face à des écrans).

2.2 Performance de la structure de codage HEVC

Performance de codage des tailles CU 32x32 et 64x64

HEVC apporte une structure de codage *quad-tree* autorisant des tailles de blocs allant jusqu'à 64x64 pixels, alors que H.264/AVC limite le partitionnement à 16x16 pixels (*macro-blocks*). Le tableau 2.1 montre l'impact des CU de tailles 32x32 et 64x64 sur la performance de codage. Ainsi, le codage de référence est ici comparé à un codage restreint aux tailles CU de 8x8 et 16x16, dans le cas des configurations AI et RA. La réduction de complexité ΔT obtenue est donc négative, car les tailles 32x32 et 64x64 introduisent une complexité de codage supplémentaire.

Les CU de tailles 32x32 et 64x64 améliorent le taux de compression de 27.31% sur les contenus UHD en RA. Le gain est de seulement 15.50% sur les mêmes séquences réduites en HD (*EBU_réduitHD*) et de 13.57% avec la classe B. Pour le codage AI, le gain est moindre, mais tout de même statistiquement significatif : 11.79% de gain en UHD, contre 4.98% et 3.6% en HD. Les grandes tailles de CU augmentent la complexité de codage de 25 à 60% selon la configuration et la résolution.

Séquences	<i>All Intra (AI)</i>		<i>Random Access (RA)</i>	
	BD-rate	ΔT	BD-rate	ΔT
<i>EBU-UHD</i>	-11.79%	-24.2%	-27.31%	-48.6%
<i>EBU-réduitHD</i>	-4.98%	-25.5%	-15.50%	-44.1%
<i>ClasseB-HD</i>	-3.60%	-25.8%	-13.57%	-61.4%

TABEAU 2.1 – Performance de codage apportée par les tailles CU 32x32 et 64x64 dans HEVC.

Certaines séquences profitent plus que d'autres de la structure de codage flexible HEVC. Dans les résultats détaillés en Annexe B, la séquence *candle smoke* codée en RA atteint un *BD-rate* de plus de 50% alors que *waterfall pan* atteint seulement 14%. Cela dépend fortement de la nature de la séquence : *candle smoke* montre un personnage avec des détails fins (adaptés au codage avec des petits blocs) sur un fond uniforme (adapté au codage avec des grands blocs), alors que *waterfall pan* est une séquence d'extérieur constituée de textures fines. Le codage produit est essentiellement constitué de petits blocs, ainsi cette séquence tire moins partie du codage flexible HEVC que *candle smoke*.

Performance de codage des tailles TU 16x16 et 32x32

De la même manière, le tableau 2.2 montre l'impact des transformées (TU) de grandes tailles 16x16 et 32x32 sur la performance de codage. Que ce soit avec la configuration AI ou RA, le taux de compression est supérieur d'environ 20% en UHD, et de 9% pour les contenus HD. On note une complexité assez proche, ce qui s'explique par le partitionnement en TU des CU de grandes tailles (32x32 et 64x64). Un CU 64x64 est typiquement subdivisé en TU de tailles 32x32 et 16x16, ce qui est bien moins complexe à évaluer que 64 TU de taille 8x8.

Séquences	<i>All Intra (AI)</i>		<i>Random Access (RA)</i>	
	BD-rate	ΔT	BD-rate	ΔT
<i>EBU-UHD</i>	-20.10%	2.8%	-19.44%	-4%
<i>ClasseB-HD</i>	-8.96%	2.9%	-9.26%	-5%

TABEAU 2.2 – Performance de codage apportée par les tailles TU 16x16 et 32x32 dans HEVC.

Performance de codage des partitions AMP

Enfin, les nouvelles partitions dites AMP (voir section 1.3.2), permettent une découpe asymétrique des CU inter-prédits. Le tableau 2.3 montre un gain de compression peu élevé, que ce soit en UHD ou en HD, mais une augmentation de la complexité de codage. Nous verrons dans le chapitre 7 qu'il est préférable de ne pas utiliser cet outil si l'on veut implémenter un codeur rapide.

Séquences	Random Access (RA)	
	BD-rate	ΔT
<i>EBU-UHD</i>	-0.35%	-24.6%
<i>ClasseB-HD</i>	-0.57%	-20.7%

TABLEAU 2.3 – Performance de codage apportée par les partitions asymétriques (AMP) dans HEVC.

Les résultats de cette section montrent que la nouvelle structure flexible HEVC, basée sur l'utilisation de tailles de blocs étendues (CU jusqu'à 64x64 pixels et TU jusqu'à 32x32 pixels) est cohérente avec le codage des grandes résolutions, d'autant plus que la contribution au gain de compression augmente avec la résolution spatiale. Par contre, l'outil AMP, censé proposer un partitionnement plus fin du contenu, est peu efficace pour améliorer le taux de compression.

2.3 Construction de la structure de codage CU dans le modèle de référence

Le modèle de référence HEVC utilise une approche récursive afin de construire le *quad-tree* CU optimal de chaque CTU de l'image. Comme le montre la figure 2.4, le coût de codage de tous les CU possibles est évalué dans un parcours d'arbre en profondeur (index bleu). Pour un CTU de taille 64x64, cela revient à calculer le coût de codage de $1 + 4 + 16 + 64 = 85$ CU. En outre, chaque CU possède son propre mode de partitionnement PU, ses propres modes de prédictions associés et son propre RQT. Le *quad-tree* CU optimal est obtenu par le parcours remontant (index rouge) en comparant successivement le coût de codage des 4 CU fils et du CU parent. Cette décision récursive peut se formuler de la manière suivante : considérons que le bloc $S(x, y)$ peut être partitionné en $k = 4$ sous-ensembles $S_i(x, y)$ avec k ensembles de décisions c_i associés et les mesures de distorsions $D_{c,i}(x, y)$ associées, la décision de sub-diviser le CU courant s'écrit :

$$\sum_{i=0}^k \arg \min_{c_i \in C_i} D_{c,i}(S_i(x, y), Rec_{c,i}(x, y)) + \lambda.R_{c,i} < \arg \min_{c \in C} D_c(S(x, y), Rec_c(x, y)) + \lambda.R_c \quad (2.6)$$

Cette approche exhaustive permet d'obtenir la meilleure solution de codage mais est très consommatrice en ressources de calcul.

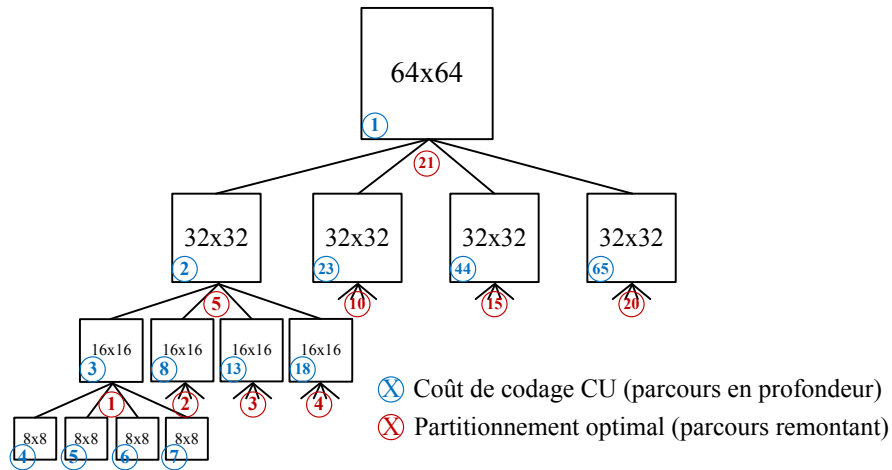


FIGURE 2.4 – Partitionnement CU : l’approche exhaustive du modèle de référence.

2.4 Réduction de complexité envisageable

Cette section met en évidence le potentiel de réduction de complexité pour un algorithme rapide qui chercherait à simplifier l’approche récursive.

2.4.1 Réduction de complexité maximum

Le premier scénario présenté dans le tableau 2.4 est le cas idéal où la boucle de codage n’évaluerait que le nombre minimal de CU, c’est à dire les CU de tailles 64x64 pixels uniquement. En UHD, la réduction de complexité est très élevée et atteint 87.4% en AI, 82.7% en RA. Elle est légèrement moins importante avec la résolution HD. La perte *BD-rate* est par contre très élevée car on s’interdit le codage flexible HEVC, ce qui se traduit par une augmentation très importante du débit nécessaire au codage des résidus.

Séquences	<i>All Intra (AI)</i>		<i>Random Access (RA)</i>	
	BD-rate	ΔT	BD-rate	ΔT
EBU-UHD	10.69%	87.4%	18.14%	82.7%
ClasseB-HD	15.95%	85.6%	32.07%	82.3%

TABLEAU 2.4 – Réduction de complexité maximum.

2.4.2 Réduction de complexité effective en évaluant les modes optimaux uniquement

Le deuxième scénario présenté dans le tableau 2.5 est le cas où la structure de codage CU, obtenue au préalable avec un codage de référence, est directement codée, sans évaluer les autres tailles de CU disponibles. On fournit donc le *quad-tree* CU de référence et on laisse la boucle de codage décider des meilleures partitions, modes de prédiction et transformées associés à ce *quad-tree*. Dans ce cas de figure, on obtient la même qualité de codage que la référence (*BD-rate*= 0%). La complexité du codage UHD Intra est réduite de 82.7%, celle du codage RA de 79.5%. La réduction de complexité est moins importante pour les formats HD. Ainsi, une approche cherchant à réduire le nombre de CU évalués par la boucle de codage aurait un potentiel d’autant plus élevé que la résolution spatiale est grande.

Séquences	All Intra (AI)		Random Access (RA)	
	BD-rate	ΔT	BD-rate	ΔT
EBU-UHD	0%	82.7%	0%	79.5%
ClasseB-HD	0%	68.6%	0%	77.2%

TABLEAU 2.5 – Réduction de complexité effective en évaluant le partitionnement CU optimal uniquement.

Dans le cas présenté dans le tableau 2.6, non seulement le codeur est contraint par le *quad-tree* CU optimal, mais également par le choix de partitionnement PU associé à chaque CU. Ce cas de figure réduit fortement la complexité en Inter car il n’y a plus besoin de comparer le coût de codage du CU pour les neuf cas de partitionnement : *INTER_2Nx2N*, *INTER_Nx2N*, *INTER_2NxN*, *INTER_AMP*, *INTRA_2Nx2N* et *INTRA_NxN*. En RA, la réduction de complexité est donc très importante : près de 95% en UHD.

Séquences	All Intra (AI)		Random Access (RA)	
	BD-rate	ΔT	BD-rate	ΔT
EBU-UHD	0%	84.2%	0%	94.5%
ClasseB-HD	0%	73.8%	0%	93.7%

TABLEAU 2.6 – Réduction de complexité effective en évaluant le partitionnement PU optimal uniquement.

Dans le cas AI, la complexité n’est réduite que légèrement par rapport au cas précédent (84.2% en UHD), car un CU Intra ne peut être constitué que d’un PU (*INTRA_2Nx2N*) sauf quand le CU concerné est de taille 8x8 : il peut également être subdivisé en 4 PU de tailles 4x4 (*INTRA_NxN*).

En allant plus loin, on peut aussi imposer le mode de prédiction Intra associé à chaque PU. Le tableau 2.7 montre une réduction de complexité de près de 96% en UHD et de plus de 95% en HD. Théoriquement, si la boucle de codage évaluait les 35 modes Intra avec J_{MODE} (eq. 1.8) pour chaque PU, la complexité serait réduite d’un rapport 35 par rapport aux scores Intra du tableau 2.6. Or, le HM intègre déjà une stratégie de pré-sélection rapide des modes Intra de luma (présentée en 1.4.4).

Séquences	All Intra (AI)	
	BD-rate	ΔT
EBU-UHD	0%	95.8%
ClasseB-HD	0%	95.2%

TABLEAU 2.7 – Réduction de complexité effective en évaluant le partitionnement PU optimal avec les modes Intra associés uniquement.

2.5 Conclusion

Bien que HEVC ne soit pas en rupture avec les codecs existants, il introduit une nouvelle structure de codage flexible ainsi qu’une incrémentation du nombre de modes de codage. Les

grandes tailles de blocs (CU jusqu'à 64x64 et TU jusqu'à 32x32 pixels) contribuent fortement à la performance de compression surtout quand il s'agit de traiter les nouveaux formats comme la UHD. Cette structure flexible entraîne cependant une augmentation de la complexité de codage liée à la compétition entre tous ces modes. Cependant, si l'on était capable de déterminer cette structure à l'avance, ou au moins prédire les modes les plus probables, on pourrait réduire considérablement la complexité de codage. La promesse est d'autant plus forte avec les formats UHD. C'est ce constat qui a motivé les approches développées dans cette thèse, pour le codage Intra (en partie II) et pour le codage Inter (en partie III).

Deuxième partie

Méthodes de réduction de complexité appliquées au codage des images Intra

Chapitre 3

Approches conventionnelles pour la réduction de complexité

Nous avons vu au sein des deux premiers chapitres que pour coder un *bitstream* HEVC, une masse de calculs considérable est réalisée car on dispose de nombreux outils de codage permettant de réduire l'erreur de prédiction, et par conséquent, la taille du *bitstream*. Afin de limiter ces calculs, de nombreuses techniques ont été développées pour le standard H.264/AVC, et plus récemment pour HEVC, afin d'identifier les modes de codage les plus probables et ainsi limiter la complexité en ne traitant pas les autres. Ces techniques sont axées sur trois stratégies principales :

Le choix du meilleur partitionnement : En Intra, la notion de CU et de PU est presque la même dans la mesure où un CU n'est constitué que d'un seul PU, sauf pour la plus petite taille CU, qui peut être décomposée en quatre PU (INTRA_NxN). Le partitionnement HEVC dispose de tailles de blocs plus étendues que celui de H.264/AVC. Comme évoqué dans la section 2.3, un CTU de taille 64x64 revient à calculer le coût de codage de 85 CU, et pour chacun, de tous les modes de prédiction et tailles de TU possibles. En limitant les calculs aux CU les plus probables, le potentiel de réduction de complexité est très important. De nombreux travaux récents [28–36] s'appuient sur les statistiques de codage et/ou la pré-analyse du contenu pour atteindre cet objectif.

Le choix du mode de prédiction : La littérature comporte plusieurs travaux cherchant à optimiser la prédiction Intra de H.264/AVC, et plus récemment celle de HEVC. Pour l'un, quatre ou neuf modes sont à tester selon la taille de bloc, pour prédire le *macro-block*. Pour l'autre, ce sont pas moins de 35 modes pour la luminance. Ce problème est souvent traité par l'analyse de l'activité directionnelle de la texture afin de déterminer les modes les plus probables. Cet aspect est souvent traité conjointement avec les solutions de partitionnement rapide.

La représentation du résiduel : La structure *quad-tree* TU au sein d'un CU (RQT) contribue à une part importante de la complexité de codage. En H.264/AVC, elle consistait à faire un choix entre une représentation avec des transformées 4x4 ou 8x8, avec ou sans coefficients codés. En HEVC, avec la nouvelle structure flexible, des tailles de transformées allant de 4x4 à 32x32 peuvent cohabiter dans un même CU, avec ou sans coefficients codés. Ce partitionnement doit être re-calculé pour chaque PU et chaque mode de prédiction associé. L'état de l'art propose des solutions pour simplifier ce processus de partitionnement et déterminer à l'avance les TU les plus probables.

3.1 Réduction de la complexité appliquée au partitionnement *quad-tree*

3.1.1 Approches basées sur les statistiques de codage

Les méthodes suivantes utilisent des heuristiques basées sur des statistiques de codage, pour identifier les tailles de blocs les plus probables et ainsi ne pas traiter les autres.

Shen et Zhang [28] proposent d'éliminer certaines tailles de CU peu probables en fonction des tailles de CU déjà codées dans le voisinage spatial. Les auteurs montrent que les CU adjacents sont fortement corrélés en termes de tailles. Ainsi, l'algorithme proposé permet de ne pas traiter les tailles de CU rarement utilisées dans le voisinage.

Kim et Choe [29] proposent d'utiliser la distribution statistique des coûts de codage CU pour déterminer si le CU courant doit être re-décomposé (*SPLIT*) ou non (*NON-SPLIT*) dans le procédé de partitionnement. Si le coût de codage du CU courant est supérieur à un seuil pré-défini, le CU courant est re-décomposé (*SPLIT*), sinon le partitionnement s'arrête (*NON-SPLIT*). Ainsi, on ne traite pas les CU fils pour réduire la complexité de codage. Pour déterminer les seuils de décision appropriés, une phase d'apprentissage calcule la distribution des coûts correspondants aux cas *SPLIT* et *NON-SPLIT*, comme dans le schéma de la fig. 3.1. Les auteurs choisissent le seuil selon la proportion de CU classés *NON-SPLIT* au lieu de *SPLIT*, appelé *False Termination Rate (FTR)*. Pour un *FTR* de 5%, l'algorithme permet de réduire la complexité de 24% contre une perte de qualité de 0.8%.

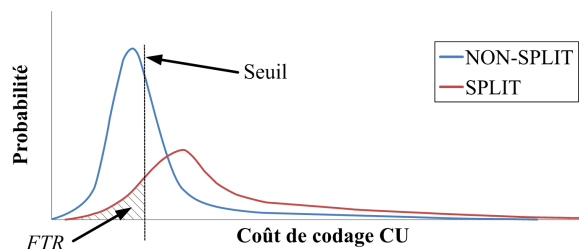


FIGURE 3.1 – Distribution statistique des classes *SPLIT* et *NON-SPLIT* en fonction du coût de codage CU.

Ce problème de classification binaire *SPLIT/NON-SPLIT* est très largement développé dans l'état de l'art. Les auteurs de [30] mais aussi de [31] abordent ce problème en mettant en œuvre une décision Bayésienne. Dans [30], la décision *SPLIT* est basée sur les informations de coût de codage J_{PRED} , alors que la décision *NON-SPLIT* est basée sur J_{MODE} . La complexité est divisée par deux contre une perte *BD-rate* de 0.6%.

3.1.2 Approches basées sur la pré-analyse du contenu

L'analyse de la texture vidéo peut également apporter de l'information sur les tailles de blocs les plus probables. Tian et Goto [32] mesurent la complexité de la texture via la variance calculée à partir d'une sous-résolution 16x16 du CTU. Des seuils permettent de ne pas traiter les PU les moins probables. La première image de la séquence est traitée avec un codage de référence pour réaliser l'apprentissage des seuils. Les complexités minimum et maximum observées dans cette image permettent de déterminer les seuils à appliquer sur les images suivantes.

Les auteurs de [33] utilisent l'entropie du signal pour décider de poursuivre le partitionnement CU ou non (classification *SPLIT/NON-SPLIT*). Les auteurs préconisent de filtrer la

texture au préalable pour ne pas impacter la mesure de l'entropie par le bruit de la vidéo source. L'algorithme permet de réduire la complexité de 62% contre une perte de 3.7%. Ces types d'approches s'apparentent à notre contribution du chapitre 5 utilisant un critère de variance.

3.1.3 Re-configuration de l'approche récursive du partitionnement

Dans le codage de référence, c'est une approche récursive qui est utilisée pour déterminer le partitionnement optimal (section 2.3). En s'appuyant sur des conditions sur les données de codage ou la pré-analyse du contenu, il peut être judicieux de ne plus démarrer le codage d'un CTU par la racine, mais par les tailles de CU intermédiaires. Ainsi, un critère de complexité permet de juger si le bloc courant peut être re-décomposé (*split*) et un critère d'homogénéité permet de juger si quatre blocs adjacents peuvent être fusionnés (*merge*). Chen et Sun [34] envisagent ce type d'approche : appelée *merge-and-split*, en démarrant le codage par la taille CU 32x32 (figure 3.2). Le CU est considéré homogène si son mode de prédiction est DC ou si l'histogramme des directions des gradients révèle une direction dominante. Dans le cas contraire, le *split* est réalisé. Si quatre CU adjacents sont homogènes et codés avec le même mode, le *merge* est réalisé.

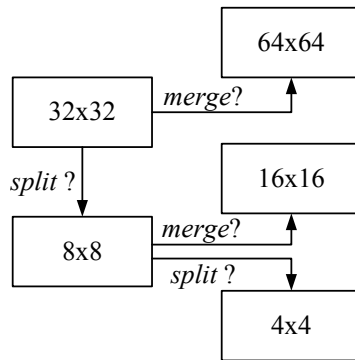


FIGURE 3.2 – Approche *merge-and-split* de Chen et Sun [34].

D'une autre manière, les auteurs de [35] proposent une approche remontante (*bottom-up*). En commençant la procédure de partitionnement par les CU 8x8, les auteurs montrent qu'il est possible d'éliminer les plus grandes tailles de CU en fonction de la structure de leurs sous-arbres CU, déjà construits. Cette méthode simple permet de réduire la complexité de codage de 38%. Une approche *bottom-up* est également introduite par Khan et al [36], non pas à partir d'informations de codage mais à partir d'un critère de variance et de moyenne du bloc de luminance. En partant de tous les blocs 4x4, et à l'aide de seuils pré-définis, les deux partitionnements les plus probables sont générés pour le codage. Cela permet de réduire la complexité de codage de 44%.

3.2 Réduction de la complexité appliquée au choix du meilleur mode de prédiction

HEVC porte le nombre de modes à 35 pour la luma. Nous avons montré en section 1.4.4 que l'approche du modèle de référence basée sur le coût SATD [21] (fonction J_{PRED}) est efficace pour réduire la complexité. La pré-sélection J_{PRED} permet de limiter l'appel à la fonction J_{MODE} , coûteuse, car elle nécessite les étapes de transformation, de quantification,

de codage entropique et de décodage. Selon [37], l'étape de pré-sélection J_{PRED} représente environ 19% de la complexité de la prédiction Intra, la compétition entre les N meilleurs modes J_{PRED} avec la fonction J_{MODE} compte pour 56% et enfin, le calcul de J_{MODE} sur le candidat final compte pour 23% de la complexité. La plupart des méthodes de l'état de l'art cherchent à limiter le nombre de candidats à évaluer avec la fonction J_{MODE} .

3.2.1 Pré-sélection basée sur l'activité directionnelle de la texture

La pré-sélection des modes de prédiction en analysant l'activité directionnelle de la texture est un sujet qui fut particulièrement traité pour H.264/AVC. Les auteurs en [38] partent du principe que la prédiction Intra est celle où les pixels dans la direction de la prédiction ont des valeurs similaires. Ainsi, en réalisant l'histogramme des directions des gradients dans le bloc à l'aide de filtres de Sobel (figure 3.3) appliqués sur chaque pixels du PU, on peut estimer la direction privilégiée et ainsi réduire le nombre de candidats J_{MODE} . Le mode DC est inclus systématiquement et ainsi, 4 candidats sur 9 sont retenus pour les prédictions 4x4 et 2 sur 4 pour les prédictions 16x16, ce qui réduit de 60% la complexité du codage Intra.

-1	0	1	1	2	1
-2	0	2	0	0	0
-1	0	1	-1	-2	-1

FIGURE 3.3 – Filtres 3x3 de Sobel dans les directions verticales et horizontales.

Cette approche basée sur les gradients a été appliquée aux modes HEVC dans la contribution de [39]. L'approche permet de réduire de 20% la complexité par rapport au codage de référence. Ce chiffre paraît peu élevé, mais s'ajoute à la stratégie de pré-sélection du modèle de référence (gain mesuré en section 1.4.4). Réalisé avant la prédiction, l'histogramme des directions des gradients est calculé, ce qui permet de réduire la taille de la liste des candidats J_{PRED} à un nombre compris entre 1 et 10 puis la liste des candidats évalués par J_{MODE} à un nombre compris entre 1 et 5 modes.

A l'instar du gradient, une mesure de l'énergie dans différentes directions du bloc courant est proposée dans la contribution de Fang et Chang [40]. Une liste de candidats J_{PRED} restreinte aux directions les plus probables est ainsi construite. La complexité de la compétition J_{PRED} est ainsi réduite de 68%.

3.2.2 Autres méthodes

Dans le modèle de référence, les N meilleurs candidats J_{PRED} sont sélectionnés pour le calcul de J_{MODE} sans autre considération. Il est possible d'analyser les statistiques générées par cette liste pour construire une liste de candidats J_{MODE} encore plus réduite. Les auteurs de [37] mais aussi de [41] comparent le coût du premier candidat J_{PRED} avec celui du deuxième candidat. Si la différence est élevée, les auteurs montrent que le premier candidat a une forte probabilité d'être choisi comme le mode optimal. Ainsi, on peut réduire le nombre de candidats J_{MODE} à 1 dans certains cas.

Siu et Wang [42] ajustent le nombre de modes disponibles selon les caractéristiques des pixels causaux du PU courant. Selon la variance de ces pixels causaux, le nombre de modes disponibles est limité à 1, 19 ou 35. Si la variance est faible, aucune compétition de modes

n'est réalisée, seul le mode *Planar* est codé. Dans le même temps, les auteurs proposent de réduire la signalisation des modes de prédiction selon ce principe, mais cela sort du cadre normatif HEVC.

Les auteurs de [43] utilisent les gradients pour simplifier à la fois le partitionnement et le choix des modes de prédiction. En appliquant 4 filtres gradients directionnels sur chaque bloc 4x4 du CTU, une mesure de complexité du PU est calculée, ce qui permet de choisir le partitionnement CU à l'avance. De plus, pour simplifier le choix des modes Intra, la liste de candidats J_{PRED} est construite avec 8 modes angulaires autour de la direction gradient la plus représentée, et 8 modes angulaires autour de la deuxième direction gradient la plus représentée. Ainsi, la complexité est réduite d'un facteur supérieur à 2, mais au prix d'une perte *BD-rate* plus élevée (supérieure à 5%).

HEVC est basé sur une structure hiérarchique. Certains travaux utilisent les corrélations entre les différentes tailles de blocs afin de réduire la complexité de codage. Kim et Yang [44] réduisent le nombre de candidats J_{MODE} tout en incluant le mode de prédiction utilisé pour le PU parent, qui a une forte probabilité d'être optimal pour le PU courant. De plus, la condition suivante est considérée suffisante pour stopper le partitionnement PU : le mode de prédiction optimal du PU courant est le même que celui du PU parent, et le PU courant n'est constitué que d'un seul TU.

3.3 Réduction de complexité appliquée à la représentation des résidus

Dans le codeur HEVC de référence, la moindre modification des paramètres de codage va modifier l'erreur de prédiction et donc avoir un impact sur la représentation optimale des résidus. Ainsi, la procédure de partitionnement TU doit être réalisée pour chaque combinaison taille de PU / mode de prédiction dans un CU. Par défaut, la profondeur RQT maximale est configurée à 3.

3.3.1 *Zero-blocks*

L'état de l'art propose des algorithmes rapides basés sur la détection de *zero-blocks*. Pour chaque TU évalué, 2 options existent : coder les résidus ($\text{CBF} = 1$) ou ne pas transmettre les résidus ($\text{CBF} = 0$). Les coefficients DCT d'un TU ayant peu d'énergie sont souvent quantifiés à zéro, surtout à QP élevé. C'est le cas dans les régions homogènes ou dans le cas d'une activité temporelle simple. Pour accélérer le traitement du partitionnement TU, des conditions suffisantes permettent de qualifier à l'avance des TU comme "*zero-blocks*".

Dans l'algorithme de [45], les tailles de TU à évaluer ou à ne pas évaluer par la boucle de codage sont pré-déterminées selon la valeur SAD du résidu après prédiction. L'algorithme présenté dans [46] met en relation le choix de la taille de TU avec le nombre de coefficients DCT non nuls produit. Il s'avère que le TU racine du RQT a une forte probabilité d'être choisi quand ce nombre est faible. La condition suffisante pour ne pas évaluer les sous-partitions TU est établie quand le nombre de coefficients produit est inférieur à un seuil. Un seuil à 3 coefficients pour toutes les tailles de TU permet de réduire de 61% la complexité du partitionnement TU, que ce soit dans les images Intra ou Inter.

3.3.2 Re-configuration de l'approche récursive du partitionnement

L'algorithme de Zhao et Zhang [47], appliqué indifféremment sur les images Intra et Inter, propose une approche de type *merge-and-split* pour obtenir le RQT optimal. Dans un

CU de taille 32x32, le processus commence par traiter les TU de tailles 16x16. Quand le TU 16x16 est qualifié de *zero-block*, aucune subdivision n'est jugée nécessaire. Les auteurs montrent que la plupart des sous-partitions d'un *zero-block* donnent des *zero-blocks*. Quand le TU 16x16 n'est pas qualifié de *zero-block*, des conditions sont définies pour l'évaluation des sous-partitions 8x8, ainsi que la fusion des 4 TU 16x16 vers un TU 32x32. En configuration RA, la complexité du partitionnement TU est réduite par 2, ce qui est bien moindre ramené à la complexité de codage dans son ensemble. Une approche de type *merge-and-split* est très adaptée au traitement rapide d'une structure de type *quad-tree* si des conditions efficaces sont utilisées pour la fusion (*merge*) et la division (*split*). Adapté aux implémentations logicielles, ce type d'approche serait cependant peu envisagé dans une implémentation sur circuit dédié (*hardware*).

La complexité du partitionnement TU est dépendante de la profondeur maximale du RQT configurée dans le codeur. Les auteurs de [48] proposent simplement de moduler cette profondeur en fonction de la taille PU. Ils montrent qu'une profondeur réduite à 1 pour les tailles PU supérieures à 8x8 aurait un impact quasiment nul sur la réduction de complexité.

3.4 Conclusion

Ce chapitre donne un aperçu des techniques utilisées pour réduire la complexité du codage HEVC Intra. On peut ainsi agir sur la complexité avec 3 niveaux de décision : le choix des tailles de blocs, des modes Intra et des TU.

Pour le traitement rapide des tailles de blocs, les approches de l'état de l'art exploitent soit les statistiques de codage (du voisinage et/ou du CU courant), soit l'analyse de la texture source. L'exploitation de statistiques nécessite au préalable un apprentissage des paramètres pour déterminer les seuils de décision appropriés. La performance de ces algorithmes est très dépendante du choix des paramètres. Ainsi, elle peut se dégrader fortement sur un contenu qui diffère de l'ensemble d'apprentissage. Pour pallier ce problème, certains travaux intègrent un apprentissage en ligne (*online*), ce qui nécessite de coder périodiquement des images de référence (non accélérées), pour mettre à jour les paramètres.

Pour la sélection rapide du mode de prédiction, l'analyse de l'activité directionnelle du bloc courant peut donner une indication sur le meilleur candidat. Le modèle de référence intégrant déjà une technique de pré-sélection rapide, les algorithmes dédiés à HEVC cherchent à réduire le nombre de candidats J_{PRED} et le nombre de candidats à évaluer avec J_{MODE} , le calcul du coup J_{MODE} étant bien plus calculatoire que celui de J_{PRED} . Les approches présentées atteignent une réduction de complexité assez limitée par rapport au modèle de référence, malgré une perte de qualité avoisinant souvent les 5%.

Pour le traitement des TU, les approches présentées exploitent l'énergie des résidus pour limiter le nombre de TU à évaluer. Ces algorithmes produisent une perte *BD-rate* très faible, mais également une réduction de complexité limitée. Même si nos contributions ne concernent pas le partitionnement TU, ces approches de l'état de l'art peuvent tout à fait venir en complément, pour encore améliorer la réduction de complexité.

Enfin, la performance objective des algorithmes de l'état de l'art est généralement évaluée sur des séquences vidéos de petites résolutions (HD et moins), notamment sur les classes de séquences du groupe *JCT-VC*. Malheureusement, le format UHD est rarement traité. Pourtant, nous avons montré dans le chapitre 2 que HEVC obtient une excellente performance de compression et un fort potentiel de réduction de complexité avec le format UHD.

Chapitre 4

Inférence de modes dans un codage basé pré-codage de sous-résolution

Dans ce chapitre, nous présentons notre première contribution qui consiste à inférer les modes les plus probables à partir d'une passe de pré-codage de sous-résolution. Notre intuition est que le codage de la version réduite en HD du contenu vidéo UHD permet de fournir un codage très proche du codage UHD. La figure 4.1 montre que le codage de la version HD est 4.5 fois moins complexe que le codage UHD de référence. Ainsi, nous proposons dans ce chapitre une architecture de codage à deux passes, constituée d'un pré-codage HD puis d'un codage UHD à très faible complexité, ne calculant qu'un nombre réduit de modes de codage, en fonction des modes codés dans la première passe. Cette approche est moins complexe que le codage UHD seul.

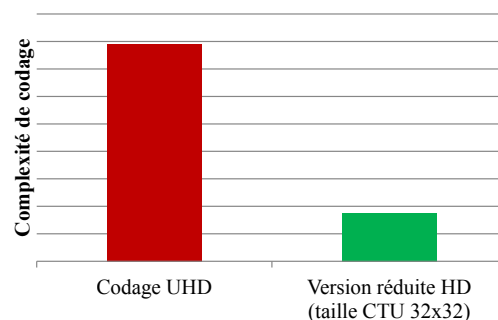


FIGURE 4.1 – Complexités du codage UHD et du codage de la version réduite au format HD.

Peu d'approches utilisant un pré-codage de sous-résolution existent dans la littérature. Seuls Dai et Zhu [49] proposent, pour MPEG-4/AVC, de pré-coder le signal vidéo, réduit de moitié en hauteur et en largeur. En fonction du mode sélectionné pour le *macroblock* pré-codé, seul un sous-ensemble de modes est calculé pour le *macroblock* co-localisé dans le codage final. Les auteurs montrent une réduction de complexité par deux sur le codage des images inter-prédites.

A titre informatif, nous pouvons également faire le lien entre l'approche développée dans ce chapitre et l'approche proposée au cours de la standardisation SHVC par Zhao et Si [50] : l'*Inter-layer Intra Mode Coding*. Il s'agit d'un outil de codage permettant de réduire le coût de syntaxe du flux scalable, en utilisant les corrélations entre les modes de codage de la couche de base (BL) et ceux de la couche d'amélioration (EL). Cet outil n'a cependant pas pour objectif de réduire la complexité, mais bien d'améliorer la performance de compression par rapport aux possibilités du standard.

Enfin, des approches d'inférence de modes sont fréquentes dans le domaine du transcodage. Il s'agit, par exemple, d'inférer les modes HEVC à partir de la syntaxe décodée d'un *bitstream* H.264/AVC [51].

4.1 Présentation de l'architecture de test

4.1.1 Principe

La figure 4.2 présente notre architecture basée sur le pré-codage. La vidéo source UHD est premièrement réduite par deux en largeur et en hauteur (rapport de résolution dyadique), au format HD. Ensuite, un codage de ce contenu HD est réalisé afin d'obtenir un premier partitionnement HEVC avec les modes de codage associés. Ce codage est seulement contraint par une taille de CTU de 32x32 au lieu de 64x64 pour le codage UHD.

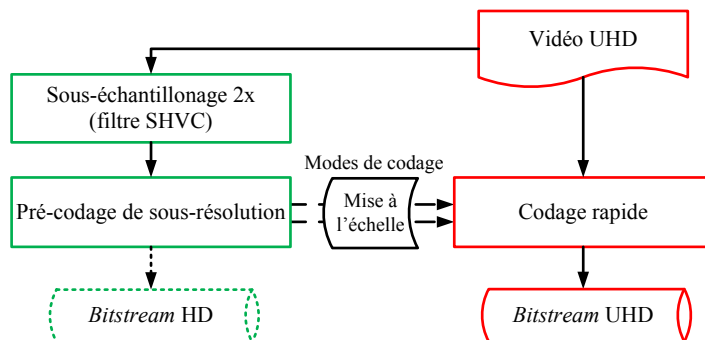


FIGURE 4.2 – Architecture de codage basé pré-codage de sous-résolution.

Enfin, les informations obtenues à partir de ce codage de sous-résolution sont mises à l'échelle (relation dyadique) puis utilisées pour inférer les modes UHD les plus probables. La figure 4.3 illustre toutes les tailles de PU disponibles pour le codage HD en configuration AI et leurs correspondances possibles dans le codage UHD. La profondeur 0 correspond à la taille de CU 32x32 en HD et à la taille de CU 64x64 en UHD. Plusieurs stratégies d'inférence sont proposées dans la section 4.2.2.

4.1.2 Filtre de sous-échantillonnage

Pour réaliser le pré-codage de sous-résolution HD, les images du contenu UHD sont réduites à l'aide du filtre utilisé dans la norme SHVC [52]. Les échantillons de luma et de chroma sont traités par des filtres 2D séparables à 12 coefficients, comme présentés ci-dessous :

- Sur les échantillons de luma, horizontalement et verticalement :

2	-3	-9	6	39	58	39	6	-9	-3	2	0
---	----	----	---	----	----	----	---	----	----	---	---

- Sur les échantillons de chroma, horizontalement :

2	-3	-9	6	39	58	39	6	-9	-3	2	0
---	----	----	---	----	----	----	---	----	----	---	---

- Sur les échantillons de chroma, verticalement :

1	-1	-8	-1	31	57	47	13	-7	-5	1	0
---	----	----	----	----	----	----	----	----	----	---	---

Ces filtres s'avèrent très efficaces pour sous-échantillonner le contenu UHD. Dans le tableau 4.1, nous comparons le pré-codage HD de la vidéo réduite avec le filtre SHVC au pré-codage HD de la vidéo réduite avec un simple filtre bi-linéaire. Nous obtenons un taux

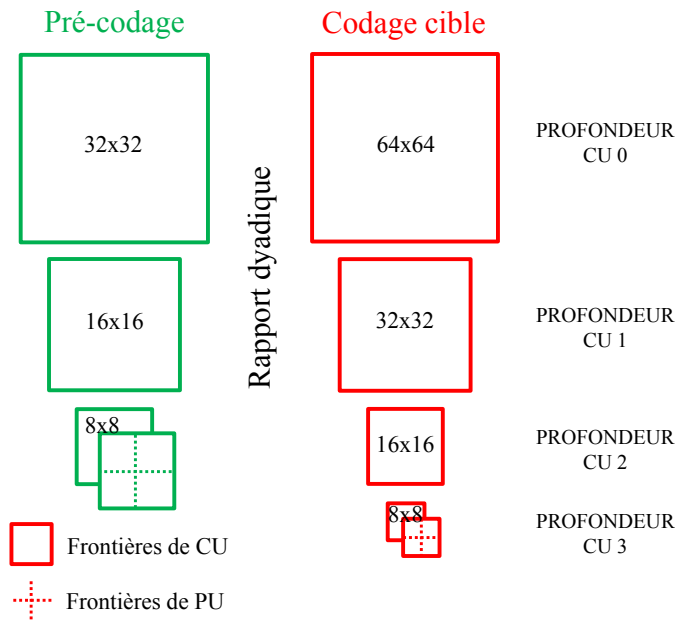


FIGURE 4.3 – Modes de prédictions AI disponibles pour le pré-codage HD et pour le codage UHD.

Séquences	All Intra (AI)	
	BD-rate	ΔT
<i>EBU-réduitHD</i>	-20%	0%

TABLEAU 4.1 – Performance du pré-codage des séquences UHD réduites en HD avec le filtre SHVC par rapport au filtre bi-linéaire.

de compression amélioré de 20%. A même QP, nous constatons une forte diminution du débit, et une augmentation du PSNR.

Les figures 4.4 et 4.5 illustrent les partitionnements HD obtenus avec les deux filtrages : SHVC et bi-linéaire. Surtout dans le bas de l'image, on voit que le partitionnement est moins dense avec le filtrage SHVC qu'avec le filtrage bi-linéaire, ce qui suggère que le contenu est moins complexe à coder. Le filtrage bi-linéaire appliqué aux textures fines d'herbe et de végétation produit un contenu bruité, ce qui explique cet écart de performance. Le bruit produit un "sur-partitionnement" et conséquemment un sur-coût de syntaxe et de résidus.



FIGURE 4.4 – Partitionnement HD (QP = 27) produit après filtrage bi-linéaire.



FIGURE 4.5 – Partitionnement HD (QP = 27) produit après filtrage SHVC.

4.2 Inférence du partitionnement

4.2.1 Analyse des partitionnements HD et UHD

Nous voulons déterminer la meilleure stratégie d'inférence pour le codage UHD, c'est à dire celle qui va réduire au maximum le nombre de modes candidats dans la région co-localisée à la partition HD, tout en maintenant le taux de compression. Nous nous intéressons seulement aux tailles de PU (partitions) dans cette section. Nous réalisons, d'une part, un pré-codage HD de référence, et d'autre part, un codage UHD de référence, à QP identique. Ainsi, nous pouvons comparer les deux partitionnements produits et déterminer statistiquement quels sont les partitionnements UHD les plus probables sachant la partition HD.

La figure 4.6 montre les différents cas de partitionnements pouvant se présenter dans la région UHD co-localisée à une partition HD. La figure 4.7 montre les histogrammes du partitionnement UHD sachant la partition HD. Le jeu de séquences *EBU-UHD* est codé à QP = 32 dans cet exemple.

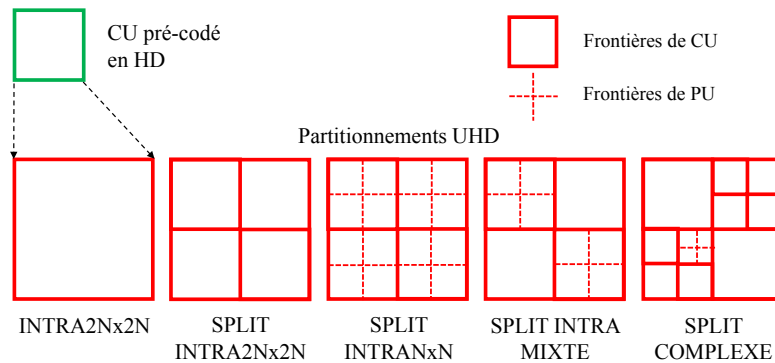


FIGURE 4.6 – Partitionnements PU possibles de la région UHD co-localisé au PU HD.

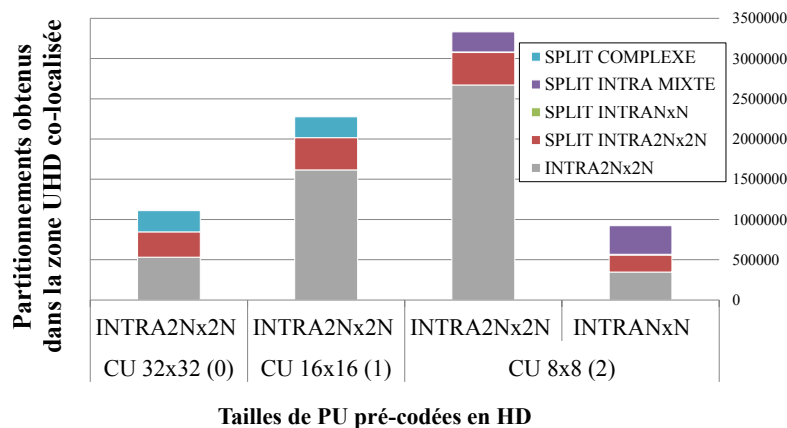


FIGURE 4.7 – Histogramme du partitionnement PU UHD sachant la taille de PU HD co-localisée (QP = 32).

Typiquement, une partition HD peut être transformée en un seul PU UHD (INTRA2Nx2N), ou bien être décomposé en quatre PU (SPLIT INTRA2Nx2N), voire être décomposé encore plus finement avec plus de 4 PU (SPLIT COMPLEXE). Lorsqu'on a un CU HD de taille 8x8, il peut être constitué d'un PU 8x8 (INTRA2Nx2N) ou de quatre PU 4x4 (INTRANxN). Dans la zone UHD co-localisée, cela peut se traduire également par 4 CU indépendants de tailles 8x8, eux-mêmes constitués de PU 8x8 (SPLIT INTRA2Nx2N), de PU 4x4 (SPLIT INTRANxN) ou des deux (SPLIT INTRA MIXTE).

La figure 4.7 nous montre que la majorité des partitions HD sont mises à l'échelle dans le

partitionnement UHD (cas *INTRA2Nx2N*). Pour les partitions HD de tailles 32x32, 16x16 et 8x8, respectivement 48%, 71% et 80% deviennent un PU UHD unique. Ainsi, plus la partition HD est petite, moins elle sera re-décomposée en UHD. Dans le cas où le CU HD est de taille 8x8, mais composé de quatre PU 4x4 (*INTRANxN*), la région UHD est majoritairement codée avec quatre CU 8x8 indépendants, constitués de prédictions 8x8 et/ou 4x4 (cas *SPLIT INTRA2Nx2N*, *SPLIT INTRANxN* et *SPLIT INTRA MIXTE*), pour une probabilité de 62%.

4.2.2 Stratégies d'inférence du partitionnement

A partir des observations précédentes, nous proposons plusieurs stratégies d'inférence du partitionnement UHD à partir du partitionnement HD. Les résultats expérimentaux correspondants sont détaillés en section 4.2.3.

Premièrement, la stratégie *CU_UP* représentée dans la figure 4.8, considère la simple mise à l'échelle du partitionnement CU HD. Par exemple, si le pré-codage HD produit un CU de taille 16x16, le codage UHD calculera uniquement un CU de taille 32x32. Seule exception, comme évoqué dans la section précédente, lorsqu'un CU HD de taille 8x8 est composé de quatre PU 4x4 (mode *INTRANxN*), il est préférable de re-décomposer en quatre CU 8x8 indépendants. Pour chacun, on compare le coût d'un PU 8x8 (*INTRA2Nx2N*) avec le coût de quatre PU 4x4 (*INTRANxN*).

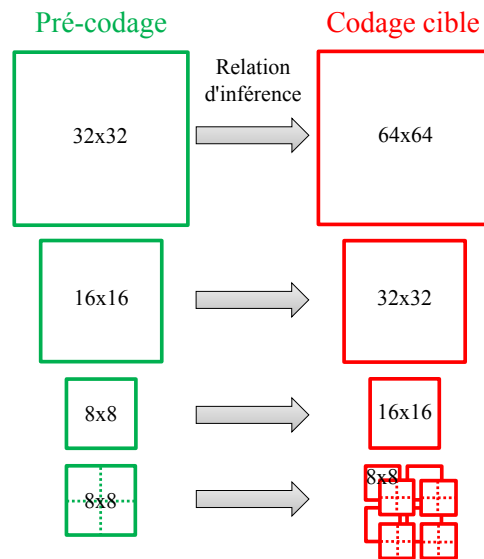


FIGURE 4.8 – Stratégie d'inférence *CU_UP*.

De la même manière, la stratégie *PU_UP* (figure 4.9) met à l'échelle le partitionnement HD. Cependant, les prédictions 4x4 (*INTRANxN*) sont omises dans le codage UHD. C'est la stratégie la plus simple car le codeur UHD n'a plus aucun degré de liberté dans le choix du partitionnement UHD.

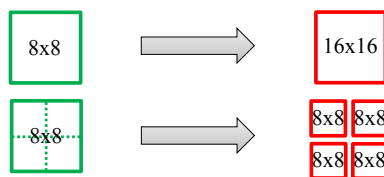


FIGURE 4.9 – Stratégie d'inférence *PU_UP*.

Pour augmenter les chances d'obtenir le mode UHD optimal, nous pouvons également consi-

dérer une re-découpe en quatre CU de la région UHD (SPLIT). D'après la figure 4.7, cela porte les chances d'obtenir le mode UHD optimal à 76%, 89% et 100% pour les profondeurs CU 0,1 et 2, respectivement. Cette stratégie, que nous appelons *CU_UP_SPLIT* (figure 4.10), impose au codeur UHD d'évaluer le coût de codage du CU mis à l'échelle, ainsi que les quatre CU fils, systématiquement.

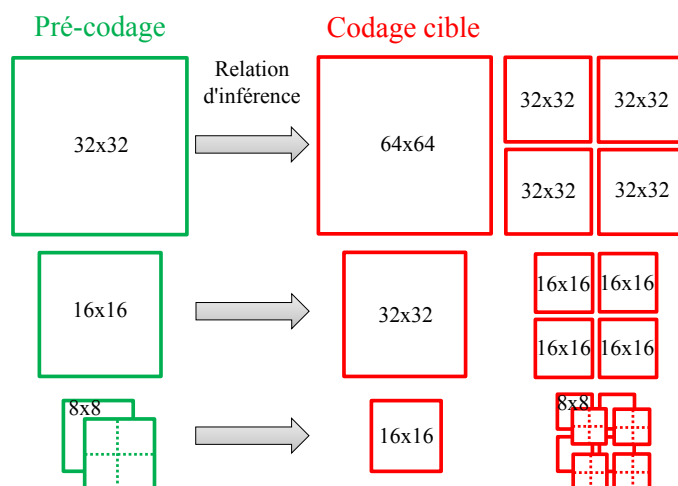


FIGURE 4.10 – Stratégie d'inférence *CU_UP_SPLIT*.

Enfin, dans certains cas non considérés en section 4.2.1, le codage UHD de référence trouve un coût débit-distorsion plus favorable en sélectionnant une partition UHD plus grande que la mise à l'échelle de la partition HD (exemples représentés en violet dans la figure 4.11). Dans ce cas, une partition UHD correspond à plusieurs partitions dans la région HD co-localisée. La stratégie d'inférence *CU_UP_MERGE&SPLIT* évalue systématiquement le CU mis à l'échelle, les quatre CU fils, et également la fusion en un CU parent, lorsque le partitionnement HD présente quatre partitions adjacentes de mêmes tailles. Avec cette stratégie, le codeur UHD a plus de degrés de liberté : d'un côté, il peut obtenir un coût de codage plus favorable en partitionnant davantage les zones très complexes (des régions où le sous-échantillonnage de la vidéo en HD a potentiellement éliminé de l'information de hautes fréquences), et d'un autre côté, il pourra obtenir un coût de codage plus favorable en sélectionnant une partition plus grande, dans des régions plus homogènes.

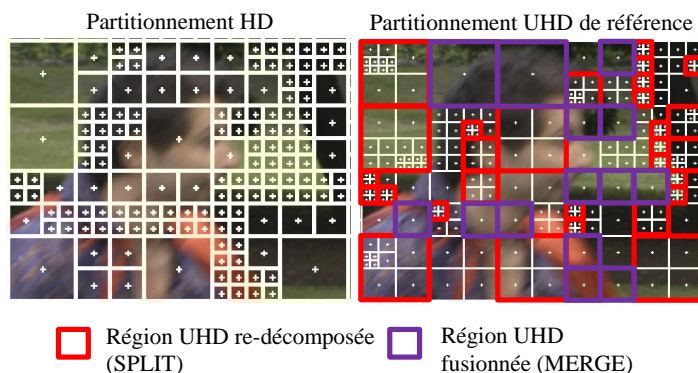


FIGURE 4.11 – Régions UHD fusionnées (MERGE) ou re-décomposées (SPLIT) dans le codeur de référence par rapport au partitionnement HD (extrait de la séquence *park_dancers* à QP = 32).

4.2.3 Résultats expérimentaux

Les résultats présentés en figure 4.12 sont réalisés selon les conditions expérimentales définies en section 2.1 avec la configuration *All Intra* (AI) et le jeu de séquences *EBU-UHD*. La réduction de complexité est représentée en pourcentage relatif au codage de référence (100%), pour distinguer la contribution du pré-codage et du codage rapide UHD. Les deux passes de codage sont réalisées à QP identique. La perte *BD-rate* moyenne est représentée ainsi que les valeurs minimales et maximales. Les résultats par séquence sont détaillés en Annexe C.

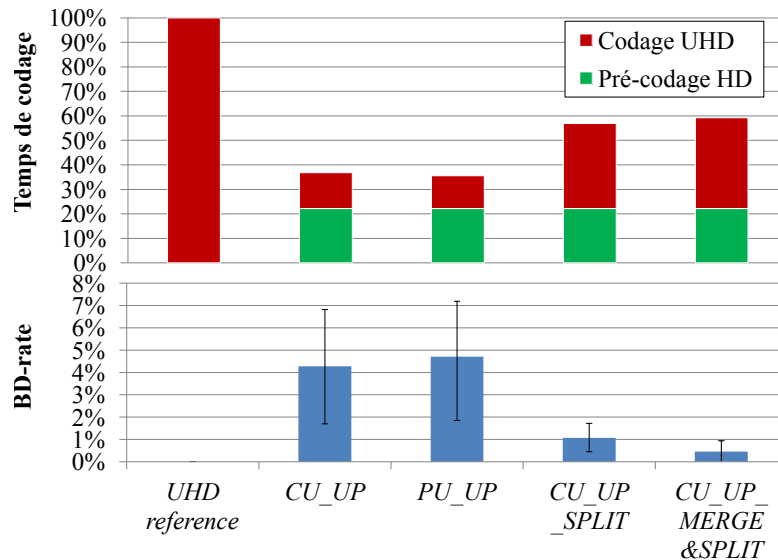


FIGURE 4.12 – Performance des stratégies d’inférence du partitionnement par rapport au codage UHD de référence.

Analyse globale

La figure 4.12 nous indique que les stratégies d’inférence *CU_UP* et *PU_UP* sont les moins complexes, en réduisant la complexité de 63.2% et 64.4%, respectivement. Le taux de compression est réduit de 4.3% et de 4.7%, respectivement. Ce niveau de perte reste acceptable dans un codeur rapide. En effet, la différence de qualité subjective entre le signal reconstruit et celui du codage de référence reste négligeable.

La stratégie *CU_UP_SPLIT*, qui évalue systématiquement les 4 CU fils du CU mis à l’échelle, permet de ramener la perte *BD-rate* à seulement 1.1%. En envisageant la fusion de 4 blocs, *CU_UP_MERGE&SPLIT* ramène la perte sous les 0.5%, en moyenne. Seulement, les réductions de complexité sont bien moins favorables : 43.1% pour *CU_UP_SPLIT*, et 40.8% pour *CU_UP_MERGE&SPLIT*.

D’une part, en ré-utilisant simplement le partitionnement HD mis à l’échelle (*CU_UP* et *PU_UP*), on favorise la réduction de complexité plutôt que la qualité de codage. D’autre part, en raffinant davantage la décision UHD (*CU_UP_SPLIT* et *CU_UP_MERGE&SPLIT*), on privilégie la qualité de codage plutôt que la réduction de complexité. Toutes les régions UHD ne nécessitent pas d’être re-décomposées par rapport à la HD, ainsi il serait important de trouver un critère efficace pour sélectionner le test *SPLIT*, uniquement lorsque c’est nécessaire. Ainsi, on pourrait par exemple obtenir une stratégie moins complexe que *CU_UP_SPLIT* et de meilleure qualité que *CU_UP* ou *PU_UP*.

Analyse par séquences

Les résultats par séquence (en Annexe C) montrent que la plus forte perte *BD-rate* est attribuée à la séquence *studio_dancer* et la plus faible aux séquences *park_dancers* et *waterfall_pan*, pour toutes les stratégies évaluées. *Park_dancers* et *waterfall_pan* sont deux séquences d'extérieur, comportant principalement des textures fines et peu structurées (herbe, végétation). Ainsi, ces séquences sont essentiellement codées avec des PU de tailles réduites et donc la décision contrainte au niveau du codeur UHD a moins d'impact sur la performance *BD-rate*. *Studio_dancer* est une séquence d'intérieur constituée à la fois de contours d'objets et de zones homogènes ou de gradients faibles. Ainsi, toutes les tailles de blocs possibles sont utilisées mais cela se traduit aussi par plus de mauvais choix lors de l'inférence depuis le partitionnement HD.

Variation du QP entre les deux passes de codage

Jusque là, nous avons choisi un QP identique pour le pré-codage HD et le codage UHD. Cependant, nous pouvons nous interroger si cette stratégie est optimale pour l'inférence du partitionnement *PU_UP*. Appliquer un QP plus faible au codage HD produirait plus de petits blocs, ce qui pourrait approcher davantage le partitionnement UHD. Dans les résultats expérimentaux en figure 4.13, nous choisissons d'appliquer un ΔQP compris entre 0 et -10 au pré-codage HD. Le codage UHD, quant à lui, est réalisé avec la stratégie d'inférence *PU_UP*. Nous ne traitons pas les cas $\Delta QP > 0$, car une quantification plus forte du codage HD produirait un partitionnement trop différent du codage UHD.

Lorsqu'un QP plus faible est appliqué sur le pré-codage HD, la perte *BD-rate* du codage UHD augmente, ainsi que la complexité des deux passes de codage. Le pas de quantification influe forcément sur la complexité d'un codage, ce qui explique l'augmentation de complexité du pré-codage HD. Quant au codage UHD, utilisant l'inférence *PU_UP*, plus le ΔQP diminue, plus le partitionnement traité par le codeur UHD sera constitué de petits blocs. Néanmoins, les résultats *BD-rate* nous montrent qu'un codage HD moins quantifié ne produira pas un partitionnement plus proche du codage UHD. Il est donc préférable de maintenir le même QP entre les deux codages.

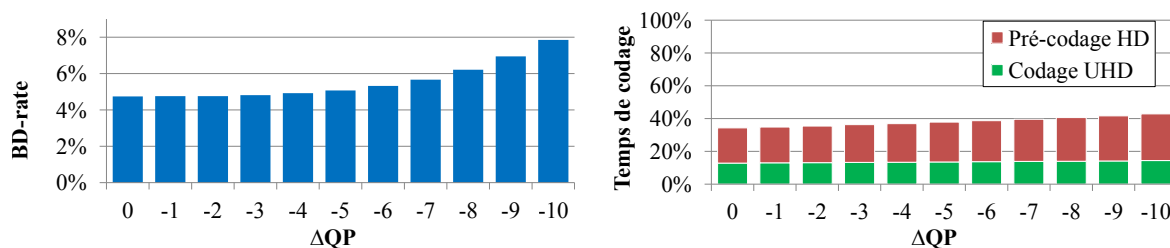


FIGURE 4.13 – Variation du QP entre la passe de pré-codage HD et la passe de codage UHD.

4.3 Inférence du mode de prédiction

Dans la section précédente, nous avons seulement contraint l'aspect partitionnement dans le codage UHD. La vidéo UHD et sa version réduite HD présentent une corrélation forte, notamment de l'activité directionnelle. Ainsi, le mode de prédiction sélectionné dans le pré-codage HD pourrait être un bon *a priori* sur le mode co-localisée en UHD. Nous proposons dans cette section un schéma de pré-sélection du mode de prédiction basé sur le mode HD.

La complexité est ainsi réduite par rapport à la stratégie *PU_UP* tout en maintenant la qualité de reconstruction.

4.3.1 Analyse des corrélations entre les modes HD et UHD

Dans le tableau 4.2, nous avons représenté certaines statistiques de codage entre les modes HD et les modes obtenus en UHD, dans le PU co-localisé. Nous utilisons pour cela la stratégie d'inférence *PU_UP* pour le codage UHD afin d'avoir un partitionnement identique. Nous ne modifions pas la pré-sélection rapide du mode de prédiction UHD (section 1.4.4). Le QP est identique entre les deux codages, et les statistiques sont obtenues par le codage du jeu de séquences *EBU-UHD*. La figure 4.14 montre la distribution des 35 modes de luminance dans le codage UHD.

Analyse des résultats observés

Le tableau 4.2 montre que seuls 13% des modes de prédiction UHD sont identiques au mode HD dans le PU co-localisé. Ainsi, une simple inférence de ce mode, sans compétition avec d'autres candidats, est insuffisante.

D'après la figure 4.14, les modes UHD les plus utilisés sont le Planar (30%) et le DC (18%). Or, dans 56% des cas, si le mode HD est DC ou Planar, le mode UHD sera soit DC soit Planar. Ces deux modes sont donc des candidats essentiels dans une stratégie d'inférence voulant limiter le nombre de candidats.

De plus, 54% des modes HD angulaires deviennent un des 33 modes UHD angulaires, les 46% restants deviennent DC ou Planar. Les 33 modes angulaires totalisent 52% des modes UHD et sont équitablement répartis, à part les modes Vertical (26) et Horizontal (10) qui comptent pour 5% chacun. Lorsque le mode HD est angulaire, la probabilité est inférieure à 10% d'obtenir exactement le même mode en UHD. Par contre, en considérant les 2 modes angulaires adjacents, la probabilité passe à 19%. Cela montre que l'angle optimal UHD a une certaine cohérence avec celui de la HD.

mode HD	mode UHD optimal	probabilité
-	mode HD	13%
-	mode HD ou un des 2 modes adjacents si angulaire	19%
Angulaire	Angulaire	54%
DC	mode HD	29%
Planar	mode HD	35%
DC ou Planar	DC ou Planar	56%
-	mode HD ou un des 2 modes adjacents ou DC ou Planar	60%
-	MPM haut ou gauche	56%
-	mode HD ou un des 2 modes adjacents ou DC ou Planar ou MPM	73%
-	J _{PRED1}	44%
-	J _{PRED1} ou J _{PRED2}	59%
-	J _{PRED1} ou MPM gauche ou haut	67%
-	J _{PRED1} ou J _{PRED2} ou MPM gauche ou haut	76%

TABLEAU 4.2 – Corrélation entre les modes HD et UHD.

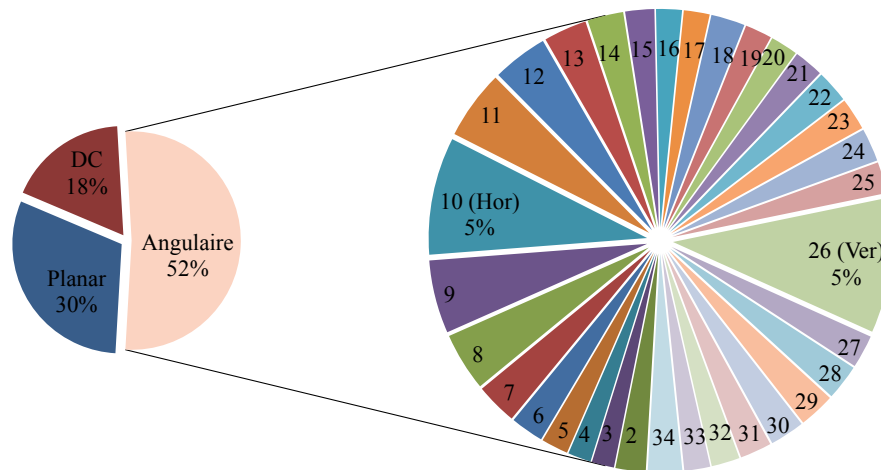


FIGURE 4.14 – Distribution des modes de luminance dans le codage UHD (PU_{UP}).

La stratégie utilisée dans le modèle de référence compare les 35 modes de luminance avec la fonction J_{PRED} (éq. 1.14), puis compare les 3 ou 8 meilleurs d’entre eux, ainsi que les MPM avec la fonction J_{MODE} (éq. 1.8). D’après nos statistiques, le meilleur mode J_{PRED} parmi 35 n’a que 44% de chances d’être le mode optimal. En le comparant aux MPM avec la fonction J_{MODE} , on obtient une probabilité de 67%. Avec le deuxième meilleur candidat, $J_{\text{PRED}2}$ ($N = 2$), la probabilité atteint 76%.

En outre, le mode UHD optimal a 60% de chances d’être l’un des modes suivants : le mode HD, les deux modes angulaires adjacents, le mode DC ou Planar. Si l’on ajoute les MPM, on atteint 73%. Avec ce sous-ensemble de candidats, on peut donc envisager une stratégie de pré-sélection plus efficace pour le mode de luminance.

Concernant le mode de chroma, la probabilité d’obtenir un mode identique au PU co-localisé en HD est de 85%. Ce chiffre est élevé, parce que le mode *DM chroma*, qui copie le mode de luma, est présent dans 82% des partitions UHD (voir la figure 4.15). Les autres modes de chroma sont répartis dans les 18% restantes. Pour la chroma, une stratégie d’inférence du mode de prédiction n’est pas envisageable, car son choix est trop dépendant du mode choisi pour la luma. Toute contrainte sur le choix du mode de chroma a montré des pertes de compression élevées. Ainsi, nous préférons maintenir une compétition entre 5 candidats avec la fonction J_{MODE} .

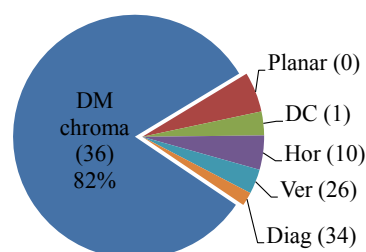


FIGURE 4.15 – Distribution des modes de chrominance dans le codage UHD (PU_{UP}).

4.3.2 Stratégies de pré-sélection du mode UHD

A partir des statistiques obtenues, nous envisageons plusieurs stratégies d’inférence du mode de prédiction utilisant le mode HD comme *a priori* sur le mode UHD. Le tableau 4.3

résume ces stratégies. L'objectif est de réduire au minimum le nombre de candidats J_{PRED} et J_{MODE} . L'inférence du partitionnement utilisée reste *PU_UP*.

Stratégie 1 : Même si les statistiques (figure 4.2) montrent que dans seulement 13% des cas, le mode UHD est identique au mode HD, nous évaluons la performance de l'inférence du mode de *luma*. Aucun degré de liberté n'est laissé au codeur UHD pour le choix du mode, le coût J_{MODE} est donc calculé une seule fois par PU.

Stratégie 2 : nous définissons un jeu de candidats J_{PRED} , représentant 60% de chances de trouver le mode UHD optimal : le mode HD, les deux modes angulaires adjacents, DC et Planar. Le coût J_{MODE} n'est calculé qu'une fois par PU, sur le meilleur candidat $J_{\text{PRED}1}$.

Stratégie 3 : nous comparons le meilleur candidat $J_{\text{PRED}1}$ avec les MPM. Entre 1 et 3 modes par PU sont ainsi évalués avec J_{MODE} .

Stratégie 4 : nous ajoutons le deuxième meilleur candidat $J_{\text{PRED}2}$, ce qui réduit la probabilité de manquer le mode optimal lors de la compétition entre les candidats J_{PRED} .

Stratégie	Candidats J_{PRED}	Candidats J_{MODE}
1	-	mode HD
2	mode HD 2 modes angulaires adjacents DC + Planar	$J_{\text{PRED}1}$
3	mode HD 2 modes angulaires adjacents DC + Planar	$J_{\text{PRED}1}$ + MPM(s)
4	mode HD 2 modes angulaires adjacents DC + Planar	$J_{\text{PRED}1}$ + $J_{\text{PRED}2}$ + MPM(s)

TABLEAU 4.3 – Stratégies de pré-sélection du mode de prédiction.

4.3.3 Résultats expérimentaux

Les résultats expérimentaux sont présentés dans le tableau 4.4. La stratégie la plus contrainte, qui évalue uniquement le mode HD de *luma*, produit une perte élevée : 11%, mais la réduction de complexité est supérieure à 70%.

Stratégie	BD-rate	ΔT
1	11%	70.5%
2	7.80%	70.1%
3	5.90%	67.3%
4	4.70%	66.5%

TABLEAU 4.4 – Performance des stratégies de pré-sélection du mode de prédiction UHD.

La stratégie 2 réduit le *BD-rate* à 7.8% en comparant une liste restreinte de candidats (entre 2 et 5) avec la fonction de coût à faible complexité J_{PRED} . L'impact sur la complexité est négligeable par rapport à la stratégie 1. Augmenter le nombre de candidats J_{PRED} ne permet

pas d’obtenir un *BD-rate* plus favorable, si l’on choisit seulement le meilleur candidat J_{PRED} comme mode optimal. Cela peut même dégrader la performance. Pour améliorer le taux de compression, il est nécessaire d’introduire une compétition basée sur le coût J_{MODE} , plus précis que le coût J_{PRED} , mais aussi plus coûteux.

Grâce à la forte corrélation spatiale entre le PU courant et les PU adjacents, les modes les plus probables (MPM) sont les plus indiqués pour construire une liste de candidats J_{MODE} . En comparant le meilleur candidat $J_{\text{PRED}1}$ avec les MPM, le *BD-rate* descend à 5.9% contre une réduction de complexité de 67.3%. Enfin, en ajoutant le deuxième meilleur candidat $J_{\text{PRED}2}$, la perte *BD-rate* est de 4.7% avec une complexité réduite de 66.5%, soit un rapport 3. Le calcul de J_{MODE} est coûteux, c’est pourquoi, il est important de sélectionner un jeu de candidats le plus petit possible. Avec la stratégie 4, le coût J_{MODE} est calculé $2 + \text{MPM}$ fois, soit entre 2 et 4 fois. En comparaison, le modèle de référence le calcule $3 + \text{MPM}$ ou $8 + \text{MPM}$ fois selon la taille de PU traitée (soit entre 3 et 10 fois).

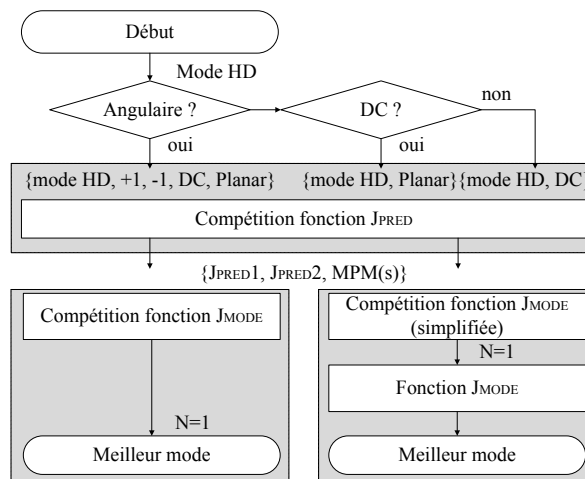


FIGURE 4.16 – Schémas de pré-sélection rapide du mode de prédiction : stratégie 4 (à gauche) et sa variante (à droite).

La stratégie 4 permet d’obtenir la même perte *BD-rate* que l’inférence *PU_UP*, mais à moindre coût calculatoire. Nous suggérons également une variante dans laquelle les candidats J_{MODE} sont évalués en considérant uniquement la taille TU égale à la taille PU, ce qui est réalisé dans le schéma de référence. En troisième et dernière étape, le coût J_{MODE} est calculé sur le meilleur candidat, avec 3 niveaux de profondeur de TU, pour profiter de la représentation flexible des résidus. La figure 4.16 illustre la différence entre la stratégie 4 et sa variante. Comme le montrent les résultats de la figure 4.17, la variante permet de réduire davantage la complexité avec une perte *BD-rate* dépassant légèrement les 5% en moyenne. Globalement, la réduction de complexité apportée par cette optimisation sur le mode de prédiction peut sembler limitée. Cependant, si l’on considère le codage UHD seul (en rouge), la complexité UHD est réduite d’un facteur 8.8 avec la stratégie 4, et d’un facteur 11.9 avec sa variante, contre 5.5 sans optimisation (stratégie *PU_UP*). Les résultats détaillés, par séquence, sont présentés en annexe C. Cette solution a fait l’objet d’une publication scientifique en [53].

Afin de démontrer que notre algorithme n’impacte pas la qualité de codage, nous avons réalisé des observations subjectives (2 observateurs experts) devant un écran UHD de grande taille (84 pouces). Après décodage des flux UHD HEVC produits par le codage de référence, d’une part, et par notre algorithme, d’autre part, à des débits identiques, nous n’avons pas perçu de différences de qualité. Cela montre que les 5% de perte *BD-rate* sont tout à fait acceptables dans la conception d’un codeur rapide.

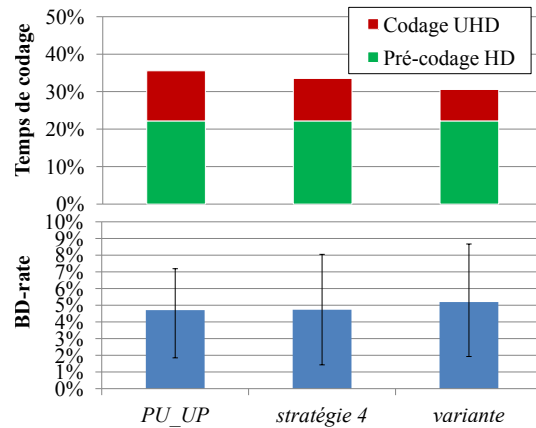


FIGURE 4.17 – Performance de la pré-sélection rapide du mode de prédiction par rapport au codage UHD de référence.

4.4 Conclusion

Dans ce chapitre, un algorithme rapide pour le codage HEVC Intra est présenté. La vidéo de sous-résolution obtenue à partir de la vidéo source est pré-codée de manière à déterminer les modes les plus probables pour le codage final. Nous avons montré qu’avec la résolution UHD, nous pouvons réduire la complexité d’un facteur 3 contre une perte de compression limitée à 5%. Cependant, la réduction de complexité est limitée par le coût du codage de sous-résolution, que l’on n’a pas cherché à simplifier dans le contexte de cette étude.

Dans beaucoup d’applications liées au *streaming* de vidéos, le contenu vidéo source est proposé en plusieurs codages de différentes résolutions, formats et qualités. Ainsi l’utilisateur peut adresser à tout instant, le flux le plus adapté à son type d’appareil et à la bande passante réseau disponible. Sur le principe de l’inférence de modes, on peut tout à fait envisager un codage "pivot" HD permettant de proposer plusieurs codages à faible complexité. Notre algorithme délivre un codage UHD jusqu’à 11.9 fois plus rapide que le codage de référence, à partir de ce codage pivot.

L’état de l’art contient de nombreux travaux liés au partitionnement rapide, généralement basés sur l’apprentissage de paramètres statistiques. La performance est très dépendante des seuils choisis, et souvent du QP et de la nature des séquences traitées. Notre algorithme ne dépend d’aucun paramètre et la performance reste globalement constante sur tous les QP et séquences. La principale source d’erreur de notre algorithme est intrinsèquement liée à l’erreur de texture entre la vidéo UHD et la vidéo HD. Le pré-codage est réalisé sur un contenu ayant perdu de l’information (hautes fréquences) par rapport à la vidéo de pleine taille. Cela peut se traduire par un partitionnement UHD légèrement insuffisant lorsque l’on met simplement à l’échelle le partitionnement HD. Même si ce n’est pas toujours la solution optimale, le codage flexible des résidus (TU) permet de compenser en grande partie les erreurs. Pour la sélection du mode de prédiction, le schéma de pré-sélection rapide que nous proposons inclut le mode HD co-localisé dans les candidats les plus probables, ce qui est inédit par rapport aux approches de l’état de l’art. Nous réduisons ainsi le nombre de modes à évaluer avec la fonction J_{MODE} , et cela n’introduit pas de perte de qualité supplémentaire par rapport à la stratégie *PU_UP*.

Chapitre 5

Codage accéléré par la pré-analyse locale du contenu

Dans ce chapitre, nous nous intéressons à l'aspect caractérisation du contenu vidéo, ou pré-analyse. Cette fois, nous nous affranchissons de la passe de pré-codage. Nous nous intéressons à l'aspect partitionnement PU uniquement. Comme l'illustre la figure 5.1, notre contribution s'articule en deux phases : une phase d'apprentissage qui réalise une analyse statistique du signal local, pour déterminer des seuils de décision ; une phase de production dans laquelle les seuils de décision sont utilisés dans notre algorithme de partitionnement rapide. Ainsi, la boucle de codage n'évalue que les tailles de PU les plus probables, ce qui permet d'obtenir un codage à faible complexité.

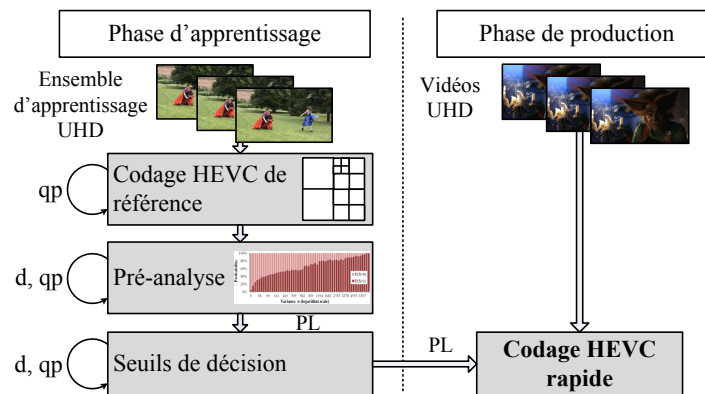


FIGURE 5.1 – Architecture de notre solution de codage accéléré par la pré-analyse.

Ce chapitre s'organise de la façon suivante : en premier lieu, nous déterminons le meilleur opérateur de pré-analyse pour caractériser le partitionnement HEVC UHD. Ensuite, nous expliquons l'apprentissage des seuils de décision, puis nous exposons notre algorithme de partitionnement rapide ainsi que les résultats expérimentaux associés. Enfin, nous proposons d'intégrer la phase d'apprentissage, non plus hors ligne, mais en ligne dans le codage rapide, afin d'enrichir l'apprentissage au cours de codage.

5.1 Motivations

Formulation du problème

Dans l'état de l'art (chapitre 3), les travaux traitant le partitionnement rapide cherchent souvent à résoudre le problème de classification binaire SPLIT/NON-SPLIT. Nous le formu-

ons de la manière suivante :

Soit S , une variable aléatoire suivant une loi de Bernoulli de paramètre p :

$$p(S|x, d, QP) = \begin{cases} p & \text{si } S = \text{SPLIT} \\ 1 - p & \text{si } S = \text{NON-SPLIT} \end{cases} \quad (5.1)$$

avec p , la probabilité qu'un PU soit re-décomposé dans le partitionnement de référence HEVC, sachant la caractéristique locale du signal x , la profondeur PU $d \in \{0, 1, 2, 3\}$ et le paramètre de quantification QP.

Choix de l'opérateur de pré-analyse

Notre intuition est que le codeur va préférer utiliser des petites tailles de blocs lorsque la région considérée est complexe. A l'inverse, un codeur va préférer utiliser des grandes tailles de blocs lorsque la région est homogène. Ainsi, nous avons évalué plusieurs opérateurs de pré-analyse pouvant caractériser la complexité spatiale d'un bloc : la variance σ , l'entropie E (utilisée dans les travaux de [33]), le gradient G et les tenseurs de structure C . Ces opérateurs sont expliqués en annexe D.1.

A partir d'un codage UHD de référence, un jeu de données (ou *dataset*) a été construit, associant les mesures de pré-analyse sur chaque PU évalué à la classe SPLIT ou NON-SPLIT. SPLIT signifie que le PU courant a été sub-divisé dans le codage optimal, alors que NON-SPLIT signifie que le PU courant a été sélectionné dans le codage optimal. A l'aide d'une application dédiée à l'apprentissage automatique (ou *Machine Learning*), *Weka* [54], nous avons calculé l'information mutuelle IM (ou gain en information) apporté par chacun de ces opérateurs. Ainsi, on peut quantifier la dépendance statistique entre la classe et l'opérateur :

$$IM(\text{classe}|\text{opérateur}) = H(\text{classe}) - H(\text{classe}|\text{opérateur}) \quad (5.2)$$

Le tableau 5.1 reporte les résultats obtenus pour chaque opérateur, sur un *dataset* généré à partir d'un codage à QP = 27 du jeu de séquences *EBU-UHD*.

attribut	IM (classe attribut)
variance σ	0.126
entropie E [33]	0.098
gradients G	0.067
tenseurs de structure C [55]	0.044

TABLEAU 5.1 – Comparaison des opérateurs de pré-analyse selon leur information mutuelle par rapport aux classes SPLIT et NON-SPLIT.

La variance σ est calculée sur le bloc de luminance et surpasse tous les opérateurs pour associer les caractéristiques locales de la texture au partitionnement HEVC. Nous la choisissons comme seul opérateur de pré-analyse pour notre algorithme.

$$\sigma = \frac{1}{N} \sum_{i=0}^{N-1} (y[i] - \bar{y})^2 \quad (5.3)$$

avec $y[i]$, l'échantillon de luminance à la position i et \bar{y} , la moyenne des échantillons de luminance dans le bloc de taille N .

Nous n'avons pas envisagé de combiner plusieurs opérateurs de pré-analyse dans un même système, ce qui signifierait concevoir un classifieur complexe avec les techniques du *Machine Learning* (arbres de décisions, *Support Vector Machines*, ...). En outre, le calcul de tous les opérateurs impacterait fortement le temps de calcul. Le système serait peut-être plus précis dans la classification mais au risque d'être trop spécifique aux données d'apprentissage. Notre objectif reste de réduire au maximum la complexité de codage, avec des seuils de décision génériques, car un codeur optimisé doit maintenir une qualité de codage constante, peu importe l'évolution du contenu au cours du temps.

Probabilités conditionnelles à la variance

Afin d'analyser cette dépendance statistique entre les classes et l'opérateur σ , nous établissons les probabilités conditionnelles d'une classe sachant la variance, lors du codage de référence. Nous voulons construire un algorithme dépendant de la taille de bloc et du QP, ainsi nous construisons les histogrammes $H_{\text{SPLIT}}^{d,\text{QP}}$ et $H_{\text{NON-SPLIT}}^{d,\text{QP}}$ pour chaque d et QP. Les probabilités conditionnelles s'écrivent :

$$p(S = \text{SPLIT} | \sigma_b, d, \text{QP}) = \frac{H_{\text{SPLIT}}^{d,\text{QP}}(\sigma_b)}{H_{\text{SPLIT}}^{d,\text{QP}}(\sigma_b) + H_{\text{NON-SPLIT}}^{d,\text{QP}}(\sigma_b)} \quad (5.4)$$

avec σ_b , un *bin* de variance. La dispersion des valeurs de variance est telle qu'il est préférable d'échantillonner en valeurs entières sur une échelle logarithmique. Nous choisissons $\sigma_b = \text{Entier}(50 \cdot \ln(0.01\sigma + 1))$, ce qui permet de représenter σ sur 256 *bins*.

La figure 5.2 montre un exemple pour les PU de tailles 32x32 codées à QP = 32. On constate que la probabilité de la classe SPLIT augmente avec la variance. Inversement la probabilité de la classe NON-SPLIT diminue avec la variance. Ainsi, il est envisageable de pré-sélectionner les modes les plus probables ou à l'inverse, écarter les modes moins probables selon ces valeurs de probabilité et la variance mesurée pour le PU courant.

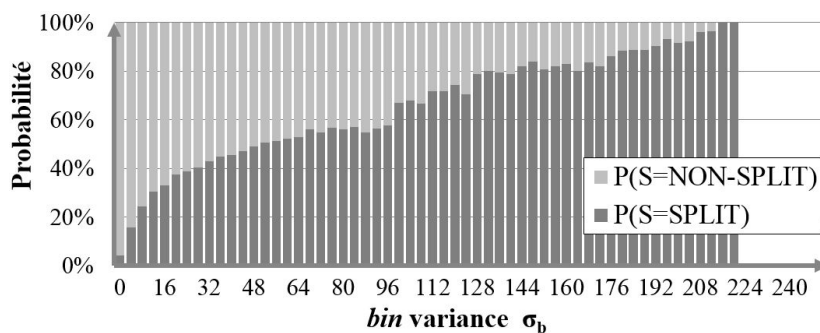


FIGURE 5.2 – Probabilités conditionnelles de SPLIT sachant σ_b pour les PU 32x32 ($d = 1$) à QP = 32.

5.2 Algorithme de partitionnement rapide à 2 classes

5.2.1 Apprentissage des seuils

La phase d'apprentissage de l'algorithme nécessite de traiter un jeu de séquences (ensemble d'apprentissage) avec le codage de référence. Elle consiste à calculer des seuils de décision Th permettant de classer les PU (NON-SPLIT ou SPLIT) selon leur variance. Th

est déterminé pour chaque profondeur PU $d \in \{0, 1, 2, 3\}$, chaque QP et chaque PL. PL est un facteur d'ajustement, compris entre 0.5 et 1, définissant la précision que l'on veut donner à l'algorithme de partitionnement rapide, la précision étant le taux de NON-SPLIT bien classés. Ce paramètre est fourni à l'algorithme de partitionnement rapide pour ajuster son point de fonctionnement (compromis réduction de complexité / perte de compression). Lors de l'apprentissage, il faut donc calculer les seuils pour chaque PL que l'on souhaite utiliser dans le codeur de production.

Comme en section 5.1, les probabilités conditionnelles de la classe SPLIT sachant σ_b sont établies pour chaque d et QP. Les seuils de décision Th , pour chaque d , QP et PL, sont déterminés selon la condition suivante :

$$p(S = \text{SPLIT} | \sigma_b, d, \text{QP}) = (1 - \text{PL}) \quad (5.5)$$

Lorsque $\text{PL} = 1$, $Th = 0$ donc aucun PU ne peut être classé NON-SPLIT. Si $\text{PL} = 0.6$, la figure 5.2 nous montre que Th prendra la valeur $\sigma_b = 28$ lorsque $d = 1$ et $\text{QP} = 32$. Cependant, jusqu'à 40% des partitions classées NON-SPLIT seront potentiellement mal classées. L'annexe D.2 liste les seuils que nous avons sélectionné pour notre algorithme.

5.2.2 Partitionnement rapide

Le schéma en figure 5.3 montre la procédure employée dans l'algorithme de partitionnement. En partant du CTU (profondeur $d = 0$), la variance σ_b est calculée. Si $\sigma_b < Th$, alors le PU courant est classé NON-SPLIT, ce qui correspond à une faible probabilité de sélectionner des PU de plus petites tailles dans le partitionnement de référence. Dans ce cas, les PU de plus petites tailles ne sont pas évalués. A l'inverse, si $\sigma_b \geq Th$, le PU courant est classé SPLIT. Il est évalué par la boucle de codage, puis la procédure est répétée sur les 4 PU fils. Ainsi, le partitionnement HEVC se poursuit tant que le choix SPLIT est sélectionné. A la profondeur $d = 3$, le choix SPLIT signifie que le CU de taille 8×8 peut être évalué avec 4 prédictions 4×4 (mode *INTRANxN*).

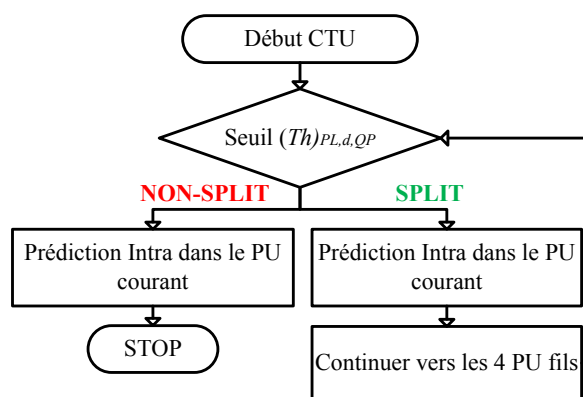


FIGURE 5.3 – Procédure de partitionnement rapide.

5.2.3 Résultats expérimentaux

Les résultats expérimentaux associés à l'algorithme de partitionnement rapide sont présentés dans la figure 5.4. Le jeu de séquences *EBU-UHD* est d'abord codé avec le modèle de référence, en configuration AI, pour générer les probabilités conditionnelles et les seuils de décision. Comme nous l'avons évoqué dans la section 2.1.2, ces séquences UHD ont

des caractéristiques spatio-temporelles très variées, ce qui nous a semblé adapté pour obtenir des seuils de décision les plus génériques possibles. Ensuite, nous traitons le même jeu de séquences avec l'algorithme de partitionnement rapide, pour plusieurs valeurs de PL (de 0.5 à 0.95). Traiter le même jeu de séquences pour la phase d'apprentissage et la phase de production permet de constater les points de fonctionnement de l'algorithme dans les conditions optimales. Ici, nous représentons le compromis réduction de complexité / perte *BD-rate* graphiquement, la position $\Delta T = 0\%$ / *BD-rate* = 0% correspondant au codage de référence UHD. $\Delta T > 0\%$ / *BD-rate* > 0% correspond à un algorithme qui a une complexité réduite, mais produit une perte de compression par rapport à la référence.

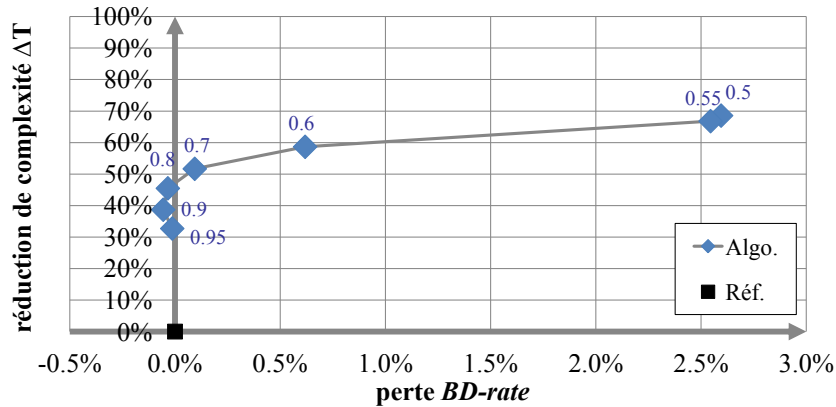
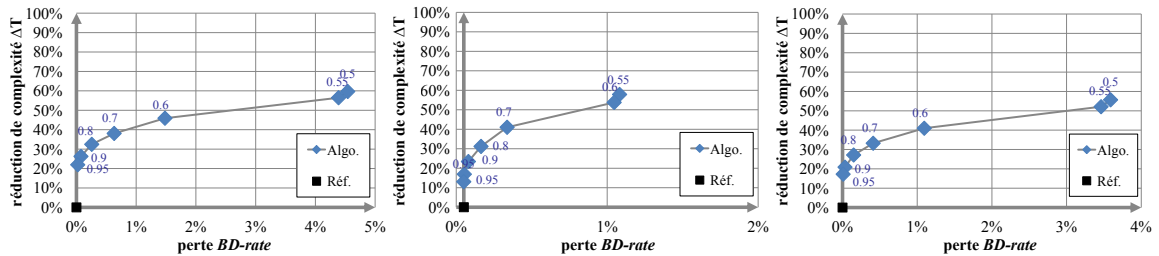


FIGURE 5.4 – Performance de notre algorithme de partitionnement rapide, pour plusieurs valeurs de PL, par rapport au codage UHD de référence (jeu de séquences *EBU-UHD*).

Pour validation de l'algorithme, les mêmes seuils de décision sont appliqués sur deux autres jeux de séquences UHD et un jeu de séquences HD dans la figure 5.5.



(a) Jeu de séquences *SJTU-UHD* (b) Jeu de séquences *SVT-UHD* (c) Jeu de séquences *classeB-HD*

FIGURE 5.5 – Performance de notre algorithme de partitionnement rapide, appliqué à d'autres jeux de séquences (validation).

Sur la figure 5.4, nous observons un *BD-rate* légèrement négatif lorsque $PL > 0.7$, soit un gain de compression par rapport à la référence. Il est difficile d'envisager que notre algorithme améliore le taux de compression, mais nous pouvons expliquer ce résultat par les scores $\Delta PSNR$ (éq. 2.2), de signes parfois différents entre le canal de luma et les canaux de chroma. Ce biais est dû au calcul de J_{MODE} (éq. 1.8) au sein du modèle de référence. La pondération entre luma et chroma choisie dans le calcul du terme de distorsion D_{SSD} de l'éq. 1.8 est la suivante : $D_{SSD} = SSD_Y + 2.SSD_U + 2.SSD_V$. Les échantillons de chroma sont ainsi favorisés par rapport à la luma. Ainsi, il arrive que les $\Delta PSNR$ soient de signes différents entre le canal Y et les canaux U,V. Dans ce cas, la pondération employée pour le PSNR employée pour le calcul du *BD-rate* (éq. 2.5) n'est pas idéale dans tous les cas de figure. Pour les résultats de validation en figure 5.5, on ne constate pas ce phénomène.

On peut retenir que l’algorithme réduit la complexité sans sacrifier la qualité *BD-rate*, et le paramètre PL offre différents points de fonctionnement. D’une part, la perte *BD-rate* est nulle quand $PL > 0.7$. A $PL = 0.95$, on tolère jusqu’à 5% de mauvaises classifications NON-SPLIT. Dans ce cas, la réduction de complexité est déjà de 32.7%. D’autre part, avec $PL = 0.5$, la réduction de complexité atteint 68.6% contre environ 2.6% de perte *BD-rate*. Dans ce cas, on tolère jusqu’à 50% de mauvaises classifications NON-SPLIT.

En figure 5.5, le codage rapide des jeux de séquences *SVT-UHD*, *SJTU-UHD* et *classeB-HD* produit une réduction de complexité légèrement inférieure mais tout de même très significative (proche de 60% pour $PL = 0.5$). La perte *BD-rate* varie selon le jeu de séquences, mais elle se maintient à un niveau très faible dans tous les cas. Les seuils de décision que nous avons obtenus à partir de la base d’apprentissage *EBU-UHD* sont suffisamment génériques pour traiter les contenus UHD, et même les résolutions inférieures, avec une réduction de complexité importante.

5.3 Algorithme de partitionnement rapide à 3 classes

5.3.1 Apprentissage des seuils

L’algorithme à 2 classes permet d’éliminer les plus petites tailles de blocs du processus de partitionnement, lorsque les PU sont de faible variance et classés NON-SPLIT. Or, plus la variance est élevée dans le PU courant, plus la probabilité de SPLIT tend vers 100%. Nous proposons de classer les PU comme SPLIT lorsque σ_b est très élevé, ce qui permet de ne pas traiter ces tailles de PU inutilement. De plus, nous définissons une classe intermédiaire NON-DÉFINI, pour laquelle il n’est pas préférable d’éliminer des tailles de PU du partitionnement. Cet algorithme a fait l’objet d’une publication scientifique en [56].

Cette fois-ci, lors de l’apprentissage, nous déterminons un couple de seuils sur la variance σ_b : Th_L et Th_H . A partir des probabilités conditionnelles, pour chaque profondeur PU $d \in \{0, 1, 2, 3\}$, chaque QP et chaque PL, Th_L est déterminé selon la condition de l’éq. 5.5 et Th_H est déterminé selon la condition :

$$p(S = \text{SPLIT} | \sigma_b, d, \text{QP}) = PL \quad (5.6)$$

5.3.2 Partitionnement rapide

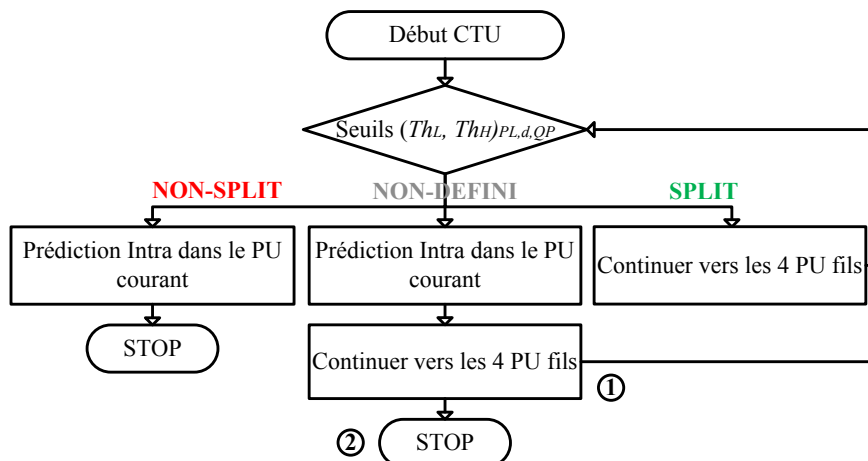


FIGURE 5.6 – Procédure de partitionnement rapide à 3 classes (variantes 1 et 2).

Le schéma en figure 5.6 montre la procédure employée dans l’algorithme de partitionnement. De la même manière qu’en section 5.2.2, la variance σ_b est calculée dans le PU courant, en commençant par la taille CTU.

Si $\sigma_b < Th_L$, alors le PU courant est classé NON-SPLIT et les PU de plus petites tailles ne sont pas évalués. A l’opposé, si $\sigma_b \geq Th_H$, le PU est classé SPLIT, car il a une forte probabilité d’être sub-divisé dans le partitionnement de référence. Cette fois-ci, le PU courant n’est pas évalué par la boucle de codage, la procédure est directement poursuivie dans les 4 PU fils. De cette manière, on donne à l’algorithme la possibilité d’éliminer les grandes tailles de PU du codage, notamment dans les régions complexes.

Enfin, si $Th_L \leq \sigma_b < Th_H$, le PU est courant est classé NON-DÉFINI. La probabilité de SPLIT correspondante est intermédiaire. Dans une première variante de l’algorithme, on ne contraint pas le codeur lors du choix NON-DÉFINI. Le PU courant est évalué, et la procédure se poursuit dans les 4 PU fils. Dans une deuxième variante, plus contrainte, le PU courant est évalué, ainsi que les 4 PU fils, puis la procédure de partitionnement s’arrête.

5.3.3 Résultats expérimentaux

Les résultats expérimentaux associés aux deux variantes sont présentés dans la figure 5.7. Dans ce cas, nous codons le même jeu de séquences (*EBU-UHD*) qui a été utilisé pour l’apprentissage des seuils (détails en annexe D.2). Dans la figure 5.8, nous appliquons ces mêmes seuils sur deux autres jeux de séquences UHD et HD.

La variante 1 offre des points de fonctionnement intéressants et une performance supérieure à l’algorithme basé sur 2 classes. A $PL = 0.9$, on obtient une *BD-rate* nul et une réduction de complexité de 45%, ce qui montre que le seuillage sur la variance est particulièrement précis. A l’inverse, si l’on réduit PL , le nombre de classifications SPLIT et NON-SPLIT incorrectes augmente. Avec $PL = 0.5$, on obtient une très forte réduction de complexité $\Delta T = 83.6\%$, au prix d’une perte *BD-rate* de 6.6%. La performance est très similaire lorsque l’on applique les mêmes seuils sur d’autres séquences HD et UHD, ΔT est d’environ 80% pour $PL = 0.5$ et le *BD-rate* se maintient à des niveaux acceptables.

La variante 2, qui contraint le choix intermédiaire (NON-DÉFINI) à une seule découpe en 4 PU ne permet pas d’obtenir les mêmes points de fonctionnement. Avec $PL = 0.5$, la performance est identique entre les 2 variantes car $Th_L = Th_H$. A PL élevé, l’algorithme commet des erreurs avec la décision NON-DÉFINI, alors qu’à PL faible, les erreurs sont commises sur les choix SPLIT et NON-SPLIT. Cependant, on obtient un très bon point de fonctionnement avec $PL = 0.6$: $\Delta T = 83\%$ contre une perte légèrement supérieure à 5%.

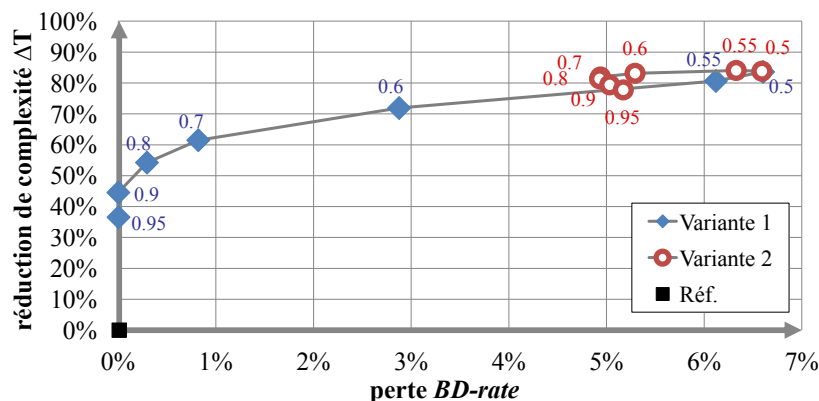


FIGURE 5.7 – Performance de notre algorithme de partitionnement rapide à 3 classes, en fonction de PL , par rapport au codage UHD de référence (jeu de séquences *EBU-UHD*).

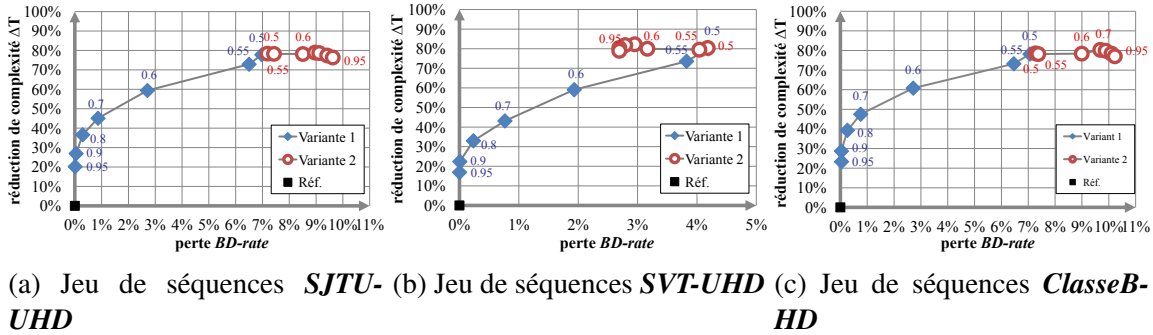


FIGURE 5.8 – Performance de notre algorithme de partitionnement rapide, appliqué à d’autres jeux de séquences (validation).

Néanmoins, nous ne retenons pas cette variante 2, car elle dégrade fortement la performance sur d’autres jeux de séquences. Comme on peut le voir avec les résultats complémentaires en figure 5.8, la perte *BD-rate* augmente avec PL. La contrainte sur la classe NON-DÉFINI a tendance à stopper trop rapidement le partitionnement. Ainsi, la réduction de complexité est importante, mais le codage peut produire un « sous-partitionnement » par rapport au partitionnement optimal, et donc une faible qualité de codage. Nous préférons donc la variante 1, car elle maintient une bonne performance et des points de fonctionnements équivalents sur les autres jeux de séquences HD et UHD (figure 5.8). Les résultats par séquence associés à l’algorithme à 3 classes sont reportés en annexe D.3. Nous notons dans ces résultats une variabilité importante de ΔT pour PL = 0.9 et une variabilité importante du *BD-rate* pour PL = 0.5, en fonction des séquences traitées, ce qui suggère que la performance de l’algorithme reste assez dépendante de la nature du contenu traité et des seuils choisis.

5.4 Analyse du comportement de l’algorithme

Pourcentages de PU éliminés de la procédure de partitionnement.

Pour mieux comprendre le comportement de l’algorithme rapide, nous avons mesuré le pourcentage de PU éliminés par l’algorithme, par QP et par tailles de PU sur un exemple de séquence UHD : *wind_wool*. Le tableau 5.2 montre les statistiques associées à l’algorithme à 2 classes (figure 5.3), et le tableau 5.3 pour l’algorithme à 3 classes (figure 5.6). Avec PL = 0.9 et PL = 0.6, les deux stratégies réussissent à éliminer un pourcentage élevé de PU. Avec l’algorithme à 2 classes, la taille de PU 64x64 n’est jamais éliminée, car la classe SPLIT traite systématiquement le PU courant. Pour les autres tailles de PU, le pourcentage augmente avec la profondeur PU, car un PU classé NON-SPLIT interdit le traitement de tous les PU fils. Il est bien évident que plus le jugement NON-SPLIT intervient sur une taille de PU importante, plus l’impact sur la complexité est important. De plus, le pourcentage de PU éliminés augmente avec le QP. Un QP élevé signifie une plus forte probabilité de coder des grandes tailles de PU, par conséquent les seuils établis sont croissants avec le QP, ce qui augmente le pourcentage de PU classés NON-SPLIT.

Avec l’algorithme à 3 classes, les grandes tailles de PU peuvent être éliminées par le choix SPLIT. On voit que la taille 64x64 est souvent éliminée, notamment à faible QP. Globalement, tous les pourcentages sont plus élevés qu’avec la version à 2 classes. Ainsi, l’algorithme permet d’obtenir une réduction de complexité plus élevée, mais aussi une variation un peu moindre selon le QP.

Séquence	PL	QP	64x64	32x32	16x16	8x8	4x4	$\Delta bitrate$	$\Delta PSNR$	$BD-rate$	ΔT
<i>wind_wool</i>	0.9	22	0.00%	0.04%	0.12%	57.32%	71.87%	-0.263%	-0.001dB	0.031%	36.7%
		27	0.00%	0.04%	41.36%	72.01%	87.67%	0.009%	-0.001dB		47.4%
		32	0.00%	0.04%	41.36%	77.62%	91.50%	-0.042%	-0.002dB		57.6%
		37	0.00%	27.30%	57.27%	87.11%	96.99%	-0.065%	-0.005dB		68.3%
<i>wind_wool</i>	0.6	22	0.00%	0.04%	41.36%	72.01%	91.43%	-0.912%	-0.016dB	2.22%	54.5%
		27	0.00%	27.30%	64.02%	87.11%	96.99%	0.806%	-0.011dB		66.9%
		32	0.00%	27.30%	77.27%	93.25%	99.03%	1.583%	-0.026dB		72.5%
		37	0.00%	39.99%	85.66%	96.64%	99.63%	1.682%	-0.048dB		77.1%

TABLEAU 5.2 – Pourcentages de PU éliminés du partitionnement avec l’algorithme à 2 classes.

Séquence	PL	QP	64x64	32x32	16x16	8x8	4x4	$\Delta bitrate$	$\Delta PSNR$	$BD-rate$	ΔT
<i>wind_wool</i>	0.9	22	59.83%	0.30%	0.00%	57.56%	71.95%	-0.207%	-0.001dB	0.04%	46.8%
		27	21.82%	0.13%	41.54%	72.07%	87.40%	-0.010%	-0.001dB		56.5%
		32	7.15%	0.13%	41.54%	77.69%	91.59%	-0.041%	-0.002dB		61.5%
		37	6.48%	27.67%	57.52%	87.25%	97.05%	-0.050%	-0.004dB		71.1%
<i>wind_wool</i>	0.6	22	100.00%	30.88%	48.95%	72.28%	91.52%	2.595%	-0.019dB	3.92%	71.3%
		27	72.52%	36.15%	67.07%	87.51%	87.51%	1.908%	-0.019dB		77.5%
		32	59.83%	31.28%	79.09%	93.55%	99.05%	2.478%	-0.036dB		80.9%
		37	46.07%	43.97%	86.55%	96.73%	99.64%	2.641%	-0.055dB		83.5%

TABLEAU 5.3 – Pourcentages de PU éliminés du partitionnement avec l’algorithme à 3 classes (variante 1).

Comparaison avec le partitionnement de référence.

La figure 5.9 montre les partitionnements UHD obtenus sur une région de la séquence *wind_wool*. Il y a de bonnes similitudes entre les partitionnements obtenus avec le jugement à 2 classes, à 3 classes, et le codage de référence. Cependant, l’algorithme concentre davantage les petits blocs sur les zones de contours, quand la référence montre une plus grande diversité de tailles PU sur des zones homogènes ou à faibles gradients.

On remarque également que l’algorithme à 3 classes génère plus de partitions de tailles intermédiaires que l’algorithme à 2 classes. Dans cet exemple, l’algorithme à 3 classes a moins produit de PU de tailles 64x64 que l’algorithme à 2 classes. Cela est dû à la classe SPLIT, qui a souvent choisi d’éliminer la taille 64x64. Au final, le partitionnement est un peu moins diversifié et aussi moins proche du partitionnement de référence qu’avec l’algorithme à 2 classes. Cependant, cela ne le rend pas moins performant.

Sensibilité des seuils sur la variance.

Dans cette section, pour démontrer l’intérêt de l’apprentissage des seuils, on se place dans le pire des cas où les seuils de décision ne sont pas appris mais tirés aléatoirement. La figure 5.10 montre les résultats expérimentaux associés à un tirage aléatoire des seuils sur σ_b . Dans un cas, les seuils sont tirés aléatoirement une fois au début du codage de la séquence. Dans le deuxième cas, les seuils Th_L et Th_H sont tirés aléatoirement à chaque PU traité. La seule condition imposée est que $Th_H \geq Th_L$.

Avec un tirage au début de la séquence, nous voyons que l’algorithme produit une perte de compression très élevée (10% en moyenne) et atteint quasiment la réduction de complexité maximum mise en évidence dans le chap. 2, dans les résultats du tableau. 2.4. Cela signifie que le seuil Th_L , associé à $d = 0$, prend une valeur trop élevée, ce qui stoppe le partitionnement systématiquement dès la taille de PU 64x64. Avec un tirage aléatoire sur chaque PU, la performance moyenne est quasiment identique sur chaque test. La perte $BD-rate$ moyenne est assez proche de celle du point de fonctionnement de notre algorithme à $PL = 0.5$, avec une complexité supérieure. La conséquence de ce tirage aléatoire est que, statistiquement, 50%

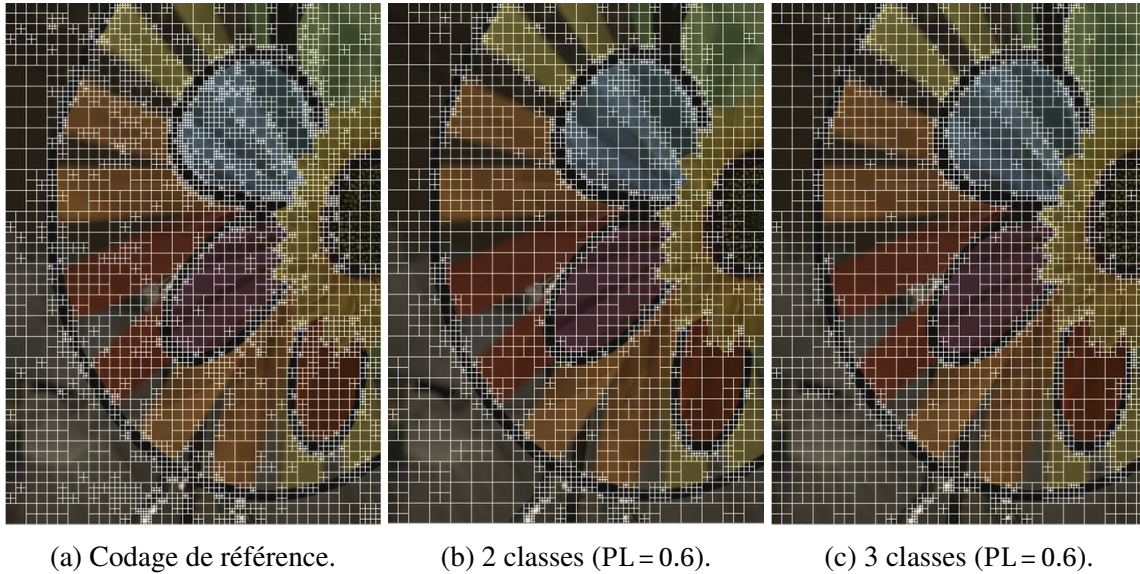


FIGURE 5.9 – Partitionnements UHD d’une région de la séquence *wind_wool* à QP = 27 .

des PU sont mal classés, et le partitionnement obtenu sera très aléatoire au vu des caractéristiques locales du signal. Ces résultats expérimentaux montrent l’importance de l’apprentissage des seuils dans notre algorithme.

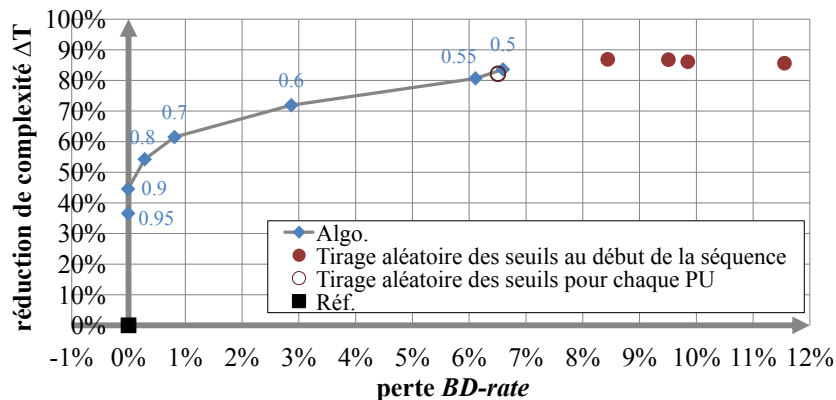


FIGURE 5.10 – Sensibilité des seuils sur la variance (jeu de séquences *EBU-UHD*).

5.5 Autres optimisations possibles

5.5.1 Simplification du choix des modes de prédiction et des TU

L’algorithme présenté dans les sections précédentes n’agit que sur la sélection des tailles de blocs. Pour réduire davantage la complexité, on pourrait agir en simultanément sur le choix du mode de prédiction, ainsi que sur le choix des TU. Nous ne pouvons pas nous servir du mode de prédiction obtenu dans un pré-codage de sous-résolution, comme dans la solution développée dans le chapitre 4. Ainsi, nous proposons une solution simple : réduire le nombre de candidats évalués avec la fonction J_{MODE} . Par défaut, les MPM sont comparés avec les 3 ou 8 meilleurs candidats J_{PRED} . Nous proposons de comparer un seul candidat aux MPM (optimisation 1 + MPM). D’autre part, nous proposons de restreindre les tailles de TU à la taille du CU (optimisation RQT1).

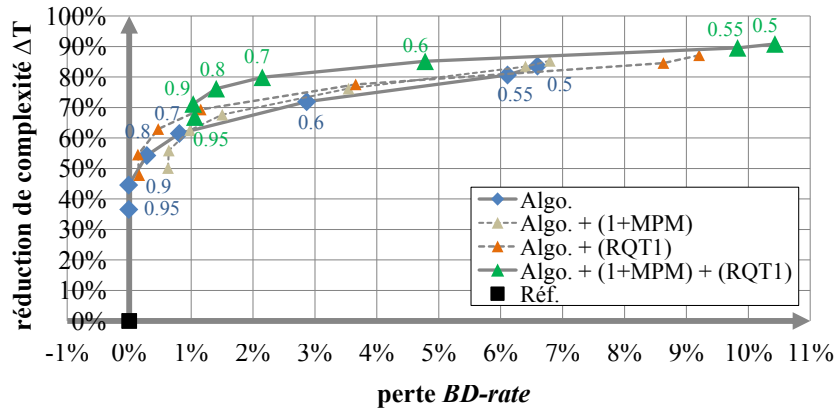


FIGURE 5.11 – Performance de l'algorithme de partitionnement rapide à 3 classes associé à des simplifications (jeu de séquences *EBU-UHD*).

Dans les résultats de la figure 5.11, les deux optimisations sont combinées avec l'algorithme à 3 classes. L'optimisation 1 + MPM considérée seule a peu d'impact sur la réduction de la complexité et le *BD-rate*. L'optimisation RQT1 permet une réduction de complexité bien plus importante mais impacte plus le taux de compression. En effet, le partitionnement rapide ne produit pas toujours un partitionnement optimal. La représentation flexible des TU permet de compenser en grande partie les erreurs de partitionnement par la transmission de TU de plus petites tailles, et/ou de plus de résidus. Il faut manier avec précaution ce genre d'approches qui restreint la plupart des degrés de libertés du codeur.

Néanmoins, les deux optimisations combinées avec l'algorithme de partitionnement rapide (en vert) permettent d'obtenir un très bon compromis lorsque PL est compris entre 0.6 et 0.95. En dessous de PL = 0.6, le taux de compression est trop impacté. Avec PL = 0.7 par exemple, la réduction de complexité est de 80% contre une perte *BD-rate* de 2.1% en moyenne. Avec PL = 0.6, la réduction de complexité atteint 85% contre une perte *BD-rate* de 4.8% en moyenne. En comparaison, la solution de pré-codage présentée dans le chapitre 4 réduit la complexité de 66% pour ce même niveau de *BD-rate*.

5.5.2 Combinaison avec l'algorithme de pré-codage

Dans le chapitre 4, nous avons étudié des solutions d'inférence de modes à partir d'un pré-codage de sous-résolution. Inférer le partitionnement HD dans le codage UHD (stratégie *PU_UP*) permet d'obtenir une réduction de complexité élevée. Envisager une re-découpe de ce partitionnement systématiquement (stratégie *CU_UP_SPLIT*) produit un *BD-rate* très faible mais au prix d'une complexité élevée. Un critère permettant de conditionner cette re-découpe permettrait d'obtenir un compromis réduction de complexité / perte de compression plus intéressant. La pré-analyse sur la variance pourrait être un bon critère, c'est pourquoi, nous suggérons d'inférer le partitionnement pré-codé, puis de conditionner le choix *SPLIT* par la pré-analyse (algorithme à 3 classes). Ainsi, le partitionnement rapide ne débute plus au niveau du CTU, mais à partir de la taille PU inférée depuis le pré-codage HD.

Dans les résultats expérimentaux en figure 5.12, la pré-analyse est appliquée uniquement sur le PU inféré à partir du partitionnement HD. Ce PU est évalué dans les cas NON-SPLIT et NON-DÉFINI. Les 4 PU fils sont évalués dans les cas NON-DÉFINI et SPLIT. L'algorithme utilise deux passes de codage (pré-codage HD et codage UHD), nous représentons la contribution des deux passes à la complexité, mais aussi la contribution du codage UHD seul.

La réduction de complexité est nettement plus faible qu'avec l'algorithme de pré-analyse seul : elle atteint 50% pour une perte de 3.3%. La passe de pré-codage HD impacte forte-

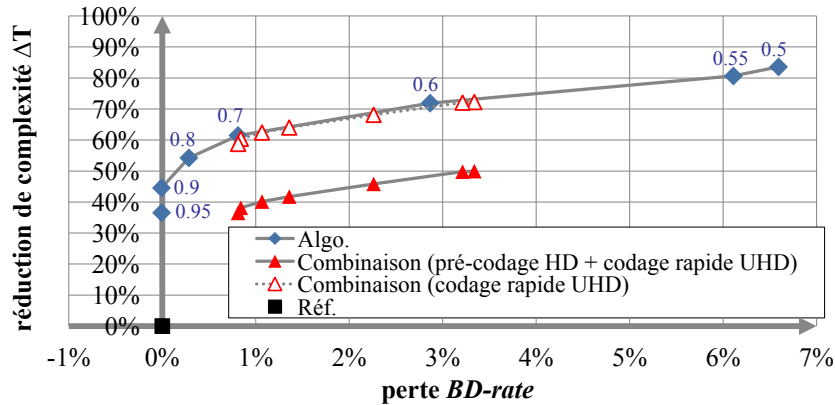


FIGURE 5.12 – Performance de l’algorithme de partitionnement rapide à 3 classes combiné à l’algorithme de pré-codage de sous-résolution (jeu de séquences *EBU-UHD*).

ment la complexité, car lorsque l’on considère uniquement le codage UHD, on approche la courbe de performance de la pré-analyse seule, avec des points de fonctionnement plus resserrés.

Cette approche est tout de même intéressante à considérer dans la conception d’un codeur optimisé. L’inconvénient de l’algorithme de pré-analyse est d’être dépendant de seuils appris, ce qui n’interdit pas des dérives si le contenu change fortement de nature par rapport aux données d’apprentissage. En outre, même si mettre à l’échelle le partitionnement HD produit un sous-partitionnement UHD, cette donnée de codage reste très cohérente avec le codage UHD. Ainsi, cette manière de combiner les deux algorithmes assure au système de ne jamais introduire de dérives importantes dans le codage, et permet donc d’assurer une qualité de codage élevée au cours du temps. Cet aspect est rarement mis en avant dans les contributions de l’état de l’art. Cette approche mérite d’être considérée en particulier pour les applications nécessitant la production de flux à différentes résolutions (WebTV/OTT).

5.6 Apprentissage *online*

5.6.1 Principe

Jusque là, l’apprentissage des seuils sur la variance se faisait hors-ligne, sur un ensemble d’apprentissage donné. Les seuils obtenus fonctionnent correctement sur la plupart des séquences testées.

Cependant, dans un codeur optimisé pour la diffusion, il est important de ne pas être spécifique et dépendant de règles arbitraires, sachant que le contenu traité évolue au cours du temps en termes de caractéristiques. Dans cette section, nous proposons d’intégrer la phase d’apprentissage sur certaines images de la séquence, afin de mettre à jour les seuils. On parle d’apprentissage en ligne, ou *online*, en anglais.

Les séquences traitées ne contiennent pas de transitions ou de changements de scènes particuliers, ainsi nous traitons une image tous les N images avec le codage de référence, comme représenté par la figure 5.13. Les autres images sont codées avec l’algorithme de partitionnement rapide et les seuils actualisés.

Chaque image d’apprentissage réalise un partitionnement complet de chaque CTU, ce qui nous permet de construire des histogrammes H_{SPLIT}^d et $H_{\text{NON-SPLIT}}^d$, et ainsi de déduire les probabilités conditionnelles $p(S = \text{SPLIT} | \sigma_b, d)$ de la classe *SPLIT*.

De la même manière qu’en section 5.3, les seuils Th_L et Th_H sur les *bins* de variance σ_b sont

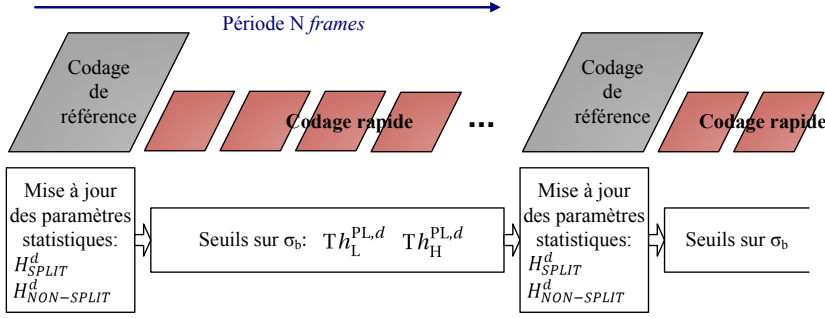


FIGURE 5.13 – Apprentissage *online* des seuils sur la variance.

déterminés selon les conditions respectives, sachant $PL \in [0.5 : 1]$:

$$\frac{H_{SPLIT}^d(\sigma_b)}{H_{SPLIT}^d(\sigma_b) + H_{NON-SPLIT}^d(\sigma_b)} = p(S = SPLIT | \sigma_b, d) = (1 - PL) \quad p(S = SPLIT | \sigma_b, d) = PL \quad (5.7)$$

Cependant, une seule image d'apprentissage, même de résolution UHD, produit des histogrammes très bruités et incomplets, qui ne permettent pas de sélectionner convenablement les seuils sur σ_b , comme le montre l'exemple de la figure 5.14a. Ainsi, pour faciliter la décision sur les seuils, nous proposons de régulariser les valeurs des histogrammes à partir des 10 valeurs voisines (noyau [1 1 1 2 3 15 3 2 1 1]). L'exemple en figure 5.14b illustre cette régularisation. Avec $PL = 50\%$, les seuils $Th_L^{PL,d}$ et $Th_H^{PL,d}$ sont identiques, et la valeur sur σ_b correspond au point de croisement entre H_{SPLIT}^d et $H_{NON-SPLIT}^d$, soit $\sigma_b = 24$.

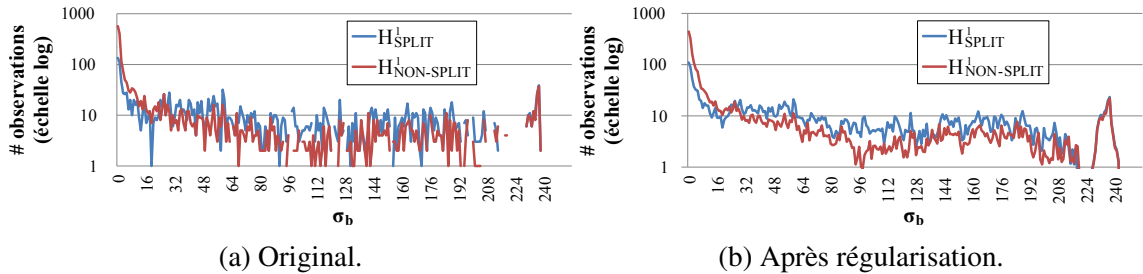


FIGURE 5.14 – Histogrammes conditionnels à σ_b pour les PU de tailles 32x32 (1ère image de la séquence *candle_smoke* codée à QP = 32).

Quand il n'est pas possible de déterminer les seuils à partir des probabilités conditionnelles, par exemple quand $\forall \sigma_b : H_{SPLIT}^d < H_{NON-SPLIT}^d$ (pas d'intersection entre H_{SPLIT}^d et $H_{NON-SPLIT}^d$), alors on utilise, par défaut, ceux de la base de seuils apprise hors-ligne.

5.6.2 Résultats expérimentaux

Les simulations ont été menées sur différents jeux de séquences, sur trois jeux de séquences UHD (*EBU-UHD*, *SVT-UHD* et *SJTU-UHD*) et sur un jeu de séquences de résolution HD (*ClasseB-HD*). Dans un cas, l'apprentissage est réalisé toutes les 24 images ; dans un autre, l'apprentissage est réalisé uniquement sur la 1ère image de la séquence. La comparaison est faite avec la version hors-ligne, utilisant la base de seuils apprise sur *EBU-UHD*. Dans tous les cas, c'est l'algorithme de partitionnement rapide à 3 classes qui est utilisé. Les résultats expérimentaux en figure 5.15 montrent globalement une réduction de complexité moins importante par rapport à la version hors-ligne. Le coût additionnel des *images*

traitées avec le codage de référence n'est pas la raison principale, car la différence de complexité est plutôt minimale entre la version $N = 24$ et la version avec apprentissage sur la 1ère image. Excepté le jeu de séquences *SVT-UHD*, l'apprentissage en ligne permet un léger gain de compression. Deux explications peuvent justifier ces écarts de complexité et de taux de compression :

- Le manque d'échantillons obtenus lors de l'apprentissage et leur manque de diversité génère des seuils imprécis. Trop de PU sont classés NON-DÉFINI, alors qu'ils pourraient être classés NON-SPLIT ou SPLIT. Les PU les moins probables ne sont donc pas tous éliminés. Le *BD-rate* et ΔT sont ainsi réduits par rapport à la version hors-ligne.
- L'image d'apprentissage est riche en observations, les seuils obtenus sont ainsi précis et très adaptés à la séquence. Cependant, l'algorithme est moins tolérant et ne classe pas suffisamment de PU comme NON-SPLIT ou SPLIT. Le *BD-rate* et ΔT sont ainsi réduits par rapport à la version hors-ligne.

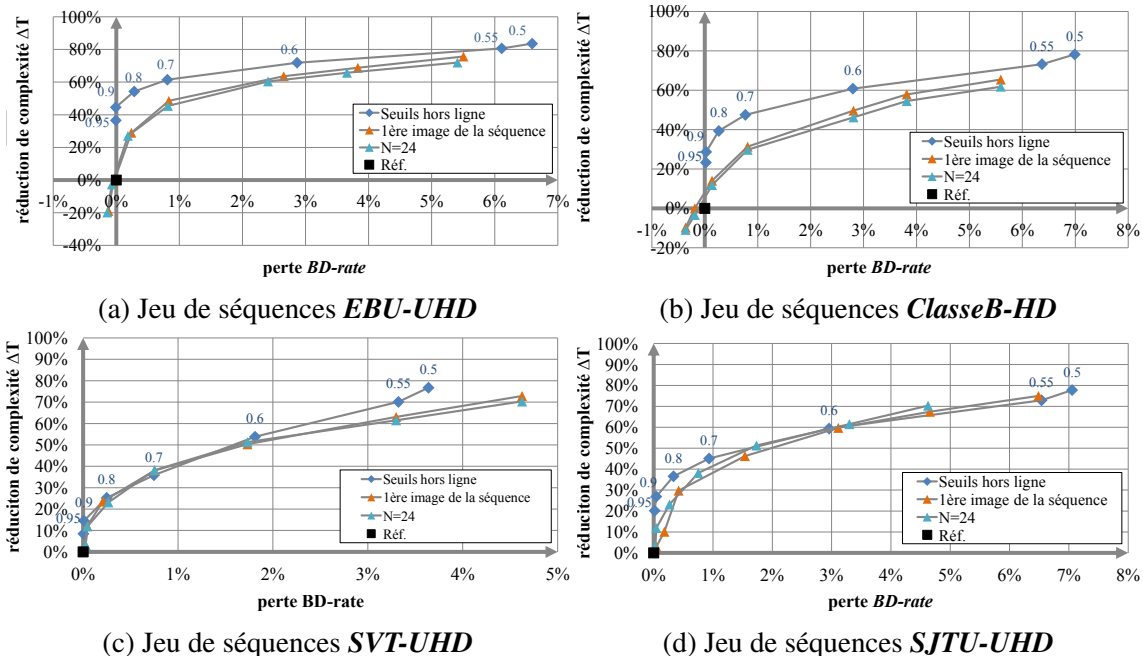


FIGURE 5.15 – Performance de l'algorithme 3 classes avec apprentissage *online*, en fonction de PL, par rapport au codage UHD de référence.

5.7 Conclusion

Dans ce chapitre, nous proposons un algorithme de codage Intra à faible complexité. En caractérisant le contenu vidéo à l'aide d'un simple opérateur de variance, notre algorithme n'évalue que les tailles de PU les plus probables. Notre approche élimine aussi bien les petites tailles de PU que les grandes tailles de PU du processus de partitionnement. La performance de l'algorithme est ainsi moins dépendante du contenu et du QP que les approches de l'état de l'art (section 3.1). De plus, on peut choisir le point de fonctionnement souhaité (compromis réduction de complexité / perte de compression) à l'aide d'un seul paramètre. La réduction de complexité peut encore être améliorée en contraignant le choix des modes de prédictions et les tailles de transformées. Par exemple, nous obtenons un point de fonctionnement montrant une réduction de complexité de 83.6% contre une perte de compression limitée à 5%. De la même manière qu'avec l'algorithme de pré-codage, la qualité subjective n'est pas impactée à ce niveau de perte de compression.

Néanmoins, l'algorithme est basé sur des seuils et donc lié à un ensemble d'apprentissage donné. Pour tous les jeux de séquences évalués, la performance est restée similaire en utilisant une seule base de seuils, mais rien ne garantit l'adaptation des paramètres au cours du temps sur un codeur de diffusion. Au prix d'une réduction de complexité moins favorable, nous suggérons deux architectures permettant d'assurer une qualité de codage élevée et constante au cours du temps.

La première solution est de combiner la pré-analyse à l'algorithme de pré-codage de sous-résolution. Le pré-codage HD fournit un *a priori* cohérent avec le codage UHD, en toute circonstance. La pré-analyse apporte une information supplémentaire, en ajustant ce partitionnement aux caractéristiques locales du signal UHD, ce qui améliore la qualité de codage. La deuxième solution est de réaliser l'apprentissage des seuils sur la variance *online*, afin de mettre à jour les seuils au cours du temps. Partant de là, un scénario de codage sur le *Cloud* pourrait être envisagé. Ayant à disposition une ressource de calcul flexible, on pourrait étendre la ressource sur un codage en mode apprentissage. Ainsi, on peut enrichir l'apprentissage séquentiellement, au tour à tour, ou sur demande, plusieurs codeurs de diffusion à faible complexité.

Troisième partie

Extension à un contexte de diffusion vidéo (images inter-prédites)

Chapitre 6

Approches conventionnelles pour la réduction de complexité

Nous avons vu dans le chapitre 3 des méthodes de réduction de complexité appliquées aux images Intra. Pour réduire la complexité, il faut identifier un nombre réduit de modes à tester (tailles de blocs de codage, modes de prédiction et tailles de transformées). Cependant, les images inter-prédites (images B ou P) sont beaucoup plus complexes à coder que les images intra-prédites. Premièrement, les CU peuvent être aussi bien traités avec la prédiction Intra que Inter. Ensuite, la prédiction Inter peut être représentée à travers un ou plusieurs PU au sein d'un CU. Un PU peut coder le mouvement selon un mode explicite, appelé AMVP ou implicite, appelé MERGE, et utiliser une ou deux images de référence parmi deux listes de référence. En outre, l'estimation de mouvement, utilisée pour le codage explicite du mouvement AMVP, représente une charge de calcul importante dans un codeur. De nombreux travaux cherchent à accélérer ce processus, mais nous ne traitons pas cet aspect dans cette thèse. Réduire le nombre de modes de codage Inter réduit incidemment la complexité liée à l'estimation de mouvement.

Dans cette partie, nous présentons des terminaisons rapides utilisées pour réduire le nombre de modes de codage à tester, et nous verrons que les approches de l'état de l'art exploitent essentiellement les statistiques de codage du voisinage spatio-temporel.

6.1 Types de terminaisons rapides

La figure 6.1 montre les étapes de recherche des prédictions Inter et Intra nécessaires pour calculer un CU d'une image B ou P. Pour la prédiction Inter, les partitionnements PU carrés, non-carrés et asymétriques sont à évaluer en utilisant les codages de mouvement MERGE et AMVP, et les images de référence disponibles. Pour la partition carrée (INTER2Nx2N), on distingue le mode SKIP (codage MERGE sans transmission de résidus). Ensuite, la prédiction Intra est calculée, ainsi que le codage PCM. Le codage PCM est un codage sans perte, où le CU est directement codé dans le *bitstream* sans passer par les étapes de prédiction, de transformation et de quantification. Dans certains cas, le contenu peut être fortement bruité, et le codage du résidu transformé produit un débit plus élevé qu'un codage sans perte. Dans le même temps, nous avons positionné sur la figure 6.1 les terminaisons rapides typiquement utilisées pour réduire la complexité du codage.

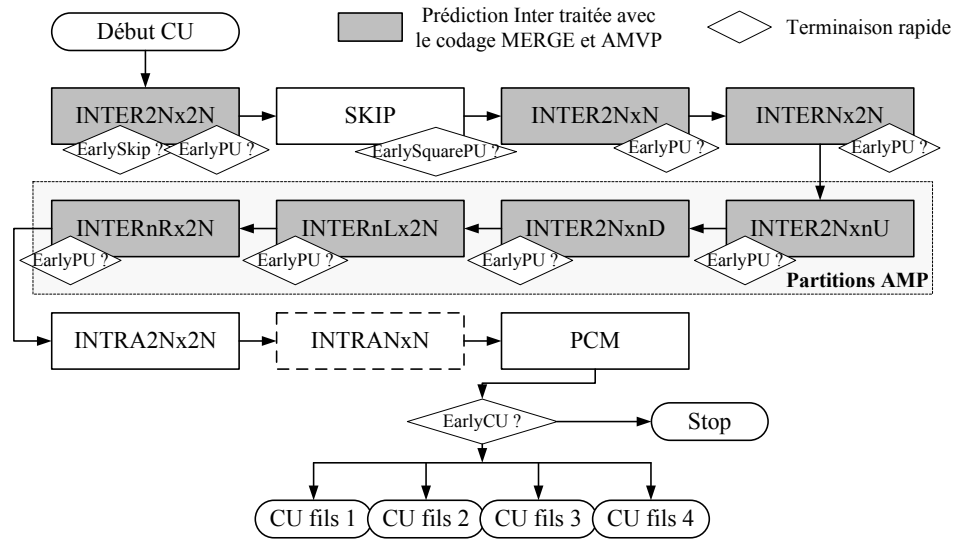


FIGURE 6.1 – Simplifications optionnelles dans le logiciel de référence.

EarlySkip

Dans le modèle de référence, il est possible d’activer une terminaison rapide, appelée *EarlySkip*. Le mode SKIP est généralement optimal lorsque la région est stationnaire, et donc identique à la région dans l’image de référence. Identifier ce mode à l’avance permet de ne pas traiter les autres modes de prédiction pour le CU courant, et ainsi éviter des calculs d’estimation de mouvement inutiles. Dans le modèle de référence, les conditions de stationnarité pour l’*EarlySkip* sont définies par les auteurs de [57] :

- $MVD = 0$
- Aucun coefficient produit ($CBF = 0$) avec le mode INTER2Nx2N

Cette terminaison rapide permettrait de réduire la complexité de 33% [57].

Il existe de nombreux algorithmes d’*EarlySkip* dans la littérature, pour H.264/AVC et pour HEVC. Récemment, Lee et al. [58] ont proposé une version améliorée de l’*EarlySkip*, ne produisant aucune perte *BD-rate*. La condition d’*EarlySkip* est un seuil de décision sur l’erreur SSD entre le PU courant et le meilleur candidat MERGE2Nx2N. Si la SSD est inférieure à ce seuil, dépendant du QP, le mode SKIP est hautement probable et les autres prédictions ne sont pas testées. L’avantage de cette condition par rapport à celle de [57] est qu’il n’est pas nécessaire de réaliser d’estimation de mouvement (AMVP2Nx2N) pour décider l’*EarlySkip*. La réduction de complexité obtenue est similaire à celle de [57].

Pour Pan et al. [59], si le CU est un *zero-block* (voir la section 3.3.1), et si pour le mode INTER2Nx2N, la distance euclidienne entre le point de départ de l’estimation de mouvement et la position optimale est nulle, le CU a de fortes chances d’appartenir à un objet vidéo uniforme, et donc d’utiliser le mode SKIP. Les autres modes ne sont pas traités dans ce cas. Pour les profondeurs CU 1, 2 et 3, les auteurs montrent que si le CU racine (profondeur 0) répond aux conditions précédentes, cela est suffisant pour juger ces CU comme SKIP. En évitant de nombreuses estimations du mouvement, cet algorithme permet de réduire la complexité de 59% contre une perte de compression faible.

EarlyPU

Nous parlons de terminaison *EarlyPU* lorsque la condition définie sur un type de PU est suffisante pour ne pas traiter les types de PU restants. Dans le modèle de référence, il est

possible d'activer le mode *CBF fast* : lorsque le type de PU Inter (INTER2NX2N ou l'un des types non-carrés) ne produit pas de résidus ($CBF = 0$), la prédiction est jugée optimale pour le CU et on ne traite plus les autres types de PU.

Une variante, que nous appelons *EarlySquarePU*, permet de ne pas traiter les cas où le CU est composé de plusieurs PU (INTER2NxN, INTERNx2N et partitions asymétriques).

Correa et al. [60] utilisent un apprentissage pour construire un arbre de décision afin de décider si le CU peut être traité avec plusieurs prédictions Inter ou non. La méthode utilise un ensemble de neuf attributs pour la décision et permet de réduire la complexité de 42%.

Vanne et al. [61] ont mené une étude statistique sur l'usage des PU non-carrés et asymétriques. Ils proposent de ne pas traiter les partitions Inter non-carrées (INTER2NxN et INTERNx2N) en fonction des coûts de codage (*RD-costs*) obtenus pour les modes SKIP et MERGE, de la manière d'un *EarlySkip*. D'autre part, les auteurs proposent de limiter l'usage des partitions non-carrées et AMP selon la profondeur CU traitée et le paramètre QP de codage. L'approche permet de réduire la complexité de 50%.

EarlyCU

La terminaison rapide *EarlyCU* consiste à déterminer si le CU courant est le plus probable. Ainsi, si des conditions suffisantes sont atteintes, il n'est pas nécessaire d'évaluer les CU fils (*NON-SPLIT*). Selon les auteurs de [62], il y a très peu de chances que des CU de petites tailles soient optimaux pour le codage si le mode SKIP est optimal pour le CU courant. Dans le modèle de référence, cette condition simple peut-être activée, et permet de réduire la complexité de 42% en ne traitant pas les sous-arbres CU. L'*EarlyCU* est particulièrement adaptée à la structure de codage flexible de HEVC. De nombreux travaux récents ont été développés pour HEVC et pour les modes Inter. Nous en détaillerons plusieurs dans les deux sections suivantes.

6.2 Approches basées sur la corrélation du voisinage spatio-temporel

Shen et al. [63] ont proposé l'une des premières contributions de réduction de complexité cherchant à limiter le nombre de CU traités dans les images inter-prédites. Pour cela, ils utilisent les informations de codage des CU voisins : trois voisins spatiaux (L, U-L et U) et le CU co-localisé dans l'image précédemment codée (voir la figure 6.2). En fonction des tailles de CU codées dans ce voisinage, certaines tailles de CU les moins probables ne sont pas traitées par le codeur. Cette technique seule permet de réduire la complexité de 30%. Ensuite, les auteurs suggèrent trois conditions d'*EarlyCU* pour limiter la complexité : l'une mesure l'homogénéité des vecteurs de mouvement avec les voisins spatiaux, une autre compare le *RD-cost* du CU à un seuil déterminé à partir des *RD-costs* des CU voisins. La troisième condition d'*EarlyCU* exige que le mode optimal du CU courant et de son CU parent soient le mode SKIP. L'ensemble de ces méthodes permet de réduire la complexité de 42%. Dans la même veine, Leng et al. [64] restreignent le nombre de CU évalués à partir des données de codage du voisinage, en incluant également le CU voisin en haut à droite U-R (voir la figure 6.2).

Récemment, Zhao et al. [65] ont proposé un algorithme plus performant, permettant de réduire la complexité de 53%. Dans une première étape de l'algorithme, les auteurs jugent que la structure du CTU co-localisé dans l'image précédente est très proche de celle du CTU courant. Ainsi, les tailles de CU observées dans le CTU co-localisé permettent de prédire la

complexité du bloc courant (simple, intermédiaire ou complexe), et en fonction, de ne pas traiter certaines tailles de CU peu probables.

Évidemment, ce genre d’algorithme produit des erreurs et il se peut qu’une taille de bloc optimale ne soit pas traitée. Ces erreurs peuvent s’accumuler au cours du codage et dégrader la qualité de codage. En outre, un changement de scène brutal peut aussi dégrader la qualité de codage. Les auteurs de [65] suggèrent de réaliser un codage de référence sur une image toutes les N images, afin de maintenir la performance de codage au cours du temps.

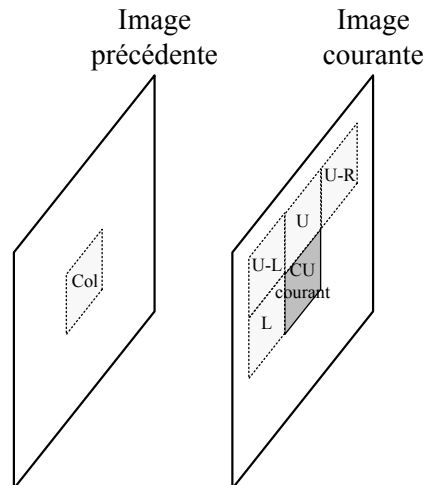


FIGURE 6.2 – Voisinage spatio-temporel typiquement utilisé pour la réduction de complexité.

6.3 Approches basées sur les caractéristiques de codage du bloc courant

Les algorithmes suivants utilisent les caractéristiques de codage du bloc courant, mais ont tout de même besoin des données précédemment codées pour déterminer les seuils de décision à utiliser.

La deuxième piste explorée dans l’algorithme de Zhao et al. [65] nécessite le traitement du mode INTER2Nx2N, et donc la réalisation de l’estimation de mouvement. Le résidu produit au sein du CU est divisé en 16 sous-blocs pour lesquels la variance est calculée. La distribution spatiale de l’énergie au sein du résidu peut être une bonne indication sur le partitionnement optimal. Ainsi, selon la distribution de ces variances, des conditions d’*EarlyPU* et d’*EarlyCU* sont définies.

D’une autre manière, Hou et Xue [66] s’intéressent au coût de codage optimal du CU courant. Ils proposent de comparer ce *RD-cost* à un seuil de décision. Si il est inférieur à ce seuil, basé sur la distribution des *RD-costs* dans l’image précédente, les CU de plus petites tailles ne sont pas traitées, et la complexité est réduite de 30% en moyenne. Dans la même veine, Cassa et al. [67] stoppent le partitionnement CU en fonction d’un seuil sur le *RD-cost*. Mais en plus, les auteurs introduisent le *TopSkip* : une taille de CU maximale pour le CTU est choisie en fonction de la taille de CU maximale observée dans le CTU co-localisée dans l’image précédente. Ainsi, on évite de traiter les plus grandes tailles CU lorsque c’est inutile, ce qui porte la réduction de complexité à 45%.

Enfin, les auteurs de [68] proposent un algorithme riche combinant des terminaisons *EarlySkip* et *EarlyCU*. Pour cela, ils traitent le mode INTER2Nx2N afin d’obtenir les informations de codage nécessaires pour estimer la complexité spatio-temporelle du CU : d’un côté, la complexité spatiale est déterminée de façon inédite, à partir des paramètres SAO, de l’autre, la complexité temporelle est estimée à partir des MV, des tailles de TU et des valeurs CBF.

La réduction de complexité approche les 50%.

Dans la contribution de Yu et al. [69], c'est l'erreur quadratique moyenne (MSE) sur le résidu de la prédiction INTER2Nx2N qui est calculée. Si cette MSE est inférieure à un seuil, appris sur un ensemble d'apprentissage, il y a peu de chances que le bloc soit codé avec des plus petites tailles de CU, on applique donc une terminaison rapide de type *EarlyCU*.

Enfin, il peut être envisagé d'utiliser la plupart des statistiques évoquées dans ce chapitre et d'y appliquer les techniques du *machine learning*. Shen et Yu [70] proposent de construire un classifieur de type SVM (*Support Vector Machine*) pour prédire si le CU doit être redécomposé (SPLIT) ou non (NON-SPLIT). L'idée est de déterminer un hyperplan qui maximise la distance entre les échantillons des deux classes binaires SPLIT et NON-SPLIT, en utilisant comme attributs, des statistiques telles que la corrélation de tailles CU du voisinage spatio-temporel, l'énergie de l'erreur de prédiction, l'activité des MV ou le *RD-cost* associé à un mode de prédiction. La réduction de complexité obtenue est de 45%. Ces approches sont complexes, ajoutent de la complexité liée au calcul de tous les attributs, et sont très dépendantes de la base d'apprentissage utilisée.

6.4 Conclusion

Dans ce chapitre, nous avons expliqué les terminaisons rapides typiquement utilisées pour réduire la complexité du codage des images inter-prédites. Généralement, la performance de ces terminaisons est très dépendante de la complexité spatio-temporelle du contenu vidéo, mais aussi du paramètre QP. Avec l'*EarlySkip*, plus le contenu est stationnaire et plus le QP est fort, plus le mode SKIP a de chances d'être choisi comme le mode optimal, et donc plus la complexité sera réduite. Avec l'*EarlyCU*, c'est le nombre de CU traités qui va diminuer. Avec l'*EarlyPU*, les conditions suffisantes seront plus souvent atteintes, ce qui aura pour effet de réduire le nombre de partitionnement PU traités.

On remarque que les conditions suffisantes pour activer une terminaison rapide sont souvent basées sur des caractéristiques de codage obtenues sur le bloc courant (vecteurs de mouvement, coûts de codage, énergie de l'erreur de prédiction). Il est par conséquent nécessaire de calculer au moins un mode de prédiction (SKIP, MERGE2Nx2N, voire l'AMVP2Nx2N) avant de prendre une décision.

En outre, une vidéo présente souvent des régions homogènes, constituant un même objet vidéo, ou un même arrière-plan. De nombreux travaux récents sur HEVC proposent d'analyser la corrélation des informations de codage du voisinage spatio-temporel afin de contraindre le codage du bloc courant à un sous-ensemble de modes. Cependant, ces approches peuvent engendrer une dégradation de la qualité de codage dès que ces régions sont complexes et avec un mouvement non homogène. D'autres travaux utilisent les données de codage du bloc courant, mais instaurent une dépendance forte avec les images précédentes car elles servent à fixer les seuils de décision. Ces approches risquent d'accumuler des erreurs au cours du codage. De plus, se baser sur l'information précédemment codée complique l'implémentation sur des architectures de calcul composées de multiples unités d'exécution parallèles.

Enfin, il est également possible d'accélérer le traitement des TU pour réduire la complexité. Nous avons exposé dans le chapitre 3, section 3.3, des méthodes utilisées indifféremment sur les images I, B et P.

Avec la prédiction Inter, il est difficile de corréliser les caractéristiques spatiales et/ou temporelle avec les modes de prédiction optimaux, car le mode optimal est fortement lié au codage obtenu dans les images précédentes. Il n'existe pas d'approches basées purement sur la pré-analyse spatiale du contenu vidéo, comme nous l'avons montré pour les images Intra. Nous ne traiterons pas dans cette partie la pré-analyse du contenu. Nous préférons traiter

l'inférence de modes à partir d'un codage de sous-résolution. Avec notre méthode, nous ne sommes pas dépendant d'un apprentissage réalisé dans le voisinage spatio-temporel, ce qui garantit une performance de codage constante et facilite la gestion mémoire et l'implémentation sur les architectures parallèles.

Chapitre 7

Codage rapide optimisé pour la diffusion au format UHD

Ce chapitre propose une configuration de codage HEVC optimisée pour la diffusion au format UHD. Elle nous servira de référence de codage optimisée afin d'estimer l'apport de nos travaux sur un système de compression réellement déployé. Nous utilisons la structure de codage *Random Access*, offrant le meilleur taux de compression par l'utilisation d'images B hiérarchiques (voir fig. 1.6c). A partir de maintenant, nous ne considérons que les images B. Pour atteindre le meilleur compromis entre réduction de complexité et perte de compression, une implémentation optimisée va limiter au maximum la combinatoire de modes appliqués à la boucle de codage, et ainsi limiter l'appel à la fonction de coût J_{MODE} . Pour cela, nous évaluons plusieurs simplifications du codage de référence HEVC : des terminaisons rapides de la procédure de partitionnement HEVC, des restrictions d'outils de codage, et une limitation du nombre d'images de référence.

7.1 Simplification des outils de codage HEVC

Dans cette section, nous évaluons des terminaisons rapides et des restrictions d'outils du codage HEVC.

Partitions non-carrées

Nous avons montré en section 2.2, que les partitions asymétriques n'étaient pas efficaces pour réduire le débit, au vu de leur complexité. « AMP off » désactive ces partitions du codage. Comme indiqué dans le tableau 7.1, nous obtenons une réduction de complexité de 21.9% en UHD contre une perte de compression de 0.3% seulement.

Nous proposons également d'éliminer les plus petites partitions non-carrées *INTER8x4* et *INTER4x8*, rarement traitées dans les codeurs temps réel, car l'impact sur la qualité est minime. Nous observons une perte *BD-rate* de seulement 0.1% en UHD. Il est même envisageable d'éliminer toutes les PU non-carrés *INTER2NxN* et *INTERNx2N* du codage (« $2NxN-Nx2N$ off »), peu importe la profondeur CU. La perte de compression est de seulement 0.9% en UHD pour une complexité réduite de 57%.

Les résultats montrent également que le compromis réduction de complexité / perte de compression de ces restrictions est plus favorable avec le format UHD qu'avec le format HD.

Pré-sélection des modes de prédiction Intra

Pour le traitement de la prédiction Intra dans les images B, la simplification « INTRA 1+MPM » limite l'appel à la fonction J_{MODE} pour le choix du mode de prédiction. Seul le meilleur candidat J_{PRED} est comparé aux MPM. L'impact sur la performance est assez faible, car le calcul des prédictions Intra est bien moins complexe que celui des prédictions Inter. Pour les canaux de chroma, comme le mode DM (copie du mode de luma) est dominant, la simplification « chroma DM seul » contraint le codage à ce mode uniquement. La perte *BD-rate* est cependant trop élevée pour un ΔT faible.

Choix des transformées

Nous proposons deux simplifications du choix des tailles de TU : soit limiter les tailles de TU à la taille CU (« RQT1 »), soit utiliser le principe d'héritage des *zeros-blocks* défendu par les auteurs de [47] (« RQT-ZBI »). Dès qu'un TU ne produit aucun coefficient ($CBF = 0$), la procédure de partitionnement TU s'arrête. Dans les deux cas, la réduction de complexité est proche de 25% en UHD, contre 0.6% de perte de compression.

Terminaisons rapides

Nous avons évoqué les terminaisons rapides *EarlyCU* (ECU), *EarlySkip* (ESD) et *CBF-fast* (CFM) dans l'état de l'art en section 6.1 (activables dans le logiciel de référence). Même si elles dépendent fortement de la complexité du contenu vidéo et du QP, l'efficacité de ces trois terminaisons rapides est très importante avec le format UHD : 70% de réduction de complexité en moyenne contre seulement 0.5% de perte de compression.

Autres outils

La désactivation du filtre SAO réduit le taux de compression de 1.4% en UHD contre une faible réduction de complexité. L'outil de quantification optimisée RDOQ est quant à lui très efficace pour améliorer le taux de compression. Nous ne retenons pas ces deux simplifications.

		Simplifications									
		AMP off	8x4-4x8 off	2NxN-Nx2N off	INTRA 1+MPM	chroma DM seul	RQT1	RQT-ZBI	ECU-ESD-CFM on	SAO off	RDOQ off
<i>EBU-UHD</i>	<i>BD-rate</i> ΔT	0.30% 21.9%	0.10% 27.0%	0.90% 57.4%	0.10% 9.9%	2.40% 7.4%	0.60% 23.1%	0.60% 25.9%	0.50% 69.8%	1.40% 5.7%	6.60% 16.7%
<i>ClasseB-HD</i>	<i>BD-rate</i> ΔT	0.60% 13.0%	0.40% 22.5%	1.30% 50.9%	0.20% 4.8%	1.20% 1.9%	1.00% 21.3%	1.00% 18.7%	1.30% 62.4%	2.20% 9.1%	5.90% 15.3%

TABLEAU 7.1 – Performance des simplifications du logiciel de référence HEVC.

Au vu des résultats obtenus, nous choisissons le sous-ensemble suivant pour former une configuration de codage rapide pour les contenus UHD :

- AMP off
- 2NxN-Nx2N off
- INTRA 1+MPM
- RQT1
- ECU / ESD / CFM on

Les résultats expérimentaux reportés dans le tableau 7.2 montrent que la complexité en UHD est réduite de 87% contre une perte de compression de 3.3%. Cette perte augmente significativement sur une résolution plus faible, ce qui prouve que notre solution est très adaptée au codage rapide des vidéos UHD.

Séquences	Random Access (RA)	
	BD-rate	ΔT
EBU-UHD	3.30%	87.1%
ClasseB-HD	6.08%	86.3%

TABLEAU 7.2 – Performance de notre configuration de codage *Random Access* optimisée.

7.2 Limitation du nombre d'images de référence

Les expérimentations avec la structure de codage RA ont été réalisées avec le GOP de référence et les listes d'images de références associées. Comme représenté dans la figure 7.1, une image peut utiliser jusqu'à 4 références par liste (L0 et L1) pour la prédiction Inter. Pour être plus proche des implémentations temps réel des systèmes de diffusion, nous suggérons de restreindre les prédictions à une référence par liste. Néanmoins, l'image B8 (soit la huitième image dans l'ordre d'affichage, et la première image dans l'ordre de codage) doit maintenir une deuxième référence dans le GOP précédent, pour maintenir un bon rendu temporel et un taux de compression élevé dans les séquences de *panning*. Cette « référence inter-GOP » est activée sauf pour l'image B8 suivant une image clé, ou *Clean Random Access* (CRA). Ainsi, le "zapping" de programme reste possible à partir des CRA. Le GOP optimisé que nous suggérons est représenté par la figure 7.2. Les valeurs indiquées dans les imagerettes du GOP représentent le nombre de références disponibles par liste, et entre parenthèse, le nombre de références en cas de présence d'une CRA au début du GOP.

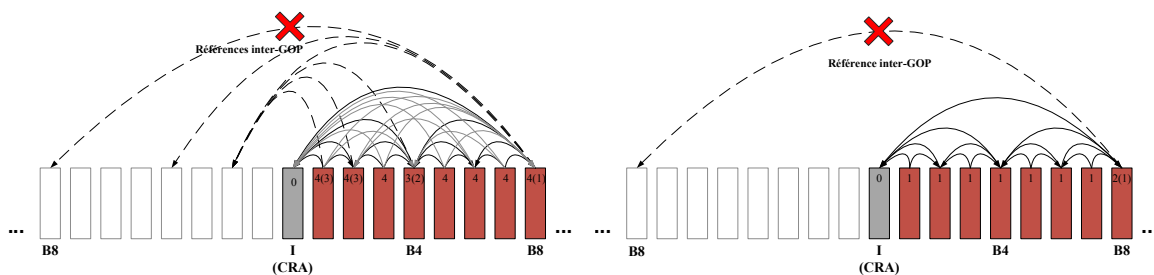


FIGURE 7.1 – Références utilisées dans le GOP *Random Access* de référence.

FIGURE 7.2 – Références utilisées dans le GOP *Random Access* optimisé pour la diffusion.

Combiné avec notre configuration de codage de la section précédente, nous obtenons les résultats suivants :

Séquences	Random Access (RA)	
	BD-rate	ΔT
EBU-UHD	4.56%	91.0%
ClasseB-HD	7.51%	90.3%

TABLEAU 7.3 – Performance de la configuration de codage RA optimisée et du GOP simplifié.

Même si nous avons fortement réduit le nombre d'images de référence (souvent de quatre à une), la réduction de complexité atteint 91% en UHD, ce qui correspond à une réduction de 33% seulement par rapport à la configuration de codage de la section précédente. Ceci s'explique par une optimisation des listes de référence, implémentée d'origine dans le modèle de référence. Lorsqu'une même image de référence est utilisée pour les listes L0 et L1, l'estimation de mouvement n'est réalisée que pour la liste L0, et pas pour la liste L1.

7.3 Conclusion

En utilisant simultanément des terminaisons rapides, des restrictions sur les modes de codage HEVC, et en limitant le nombre d'images de références utilisées pour la prédiction temporelle, nous obtenons ainsi un codage *Random Access* réduit de 91% en complexité par rapport au codage de référence. En UHD seulement, elle produit une perte de compression très acceptable (inférieure à 5%) et une qualité subjective équivalente au codage de référence. Nous appliquerons cette version optimisée dans le chapitre suivant, section 8.4.

Chapitre 8

Inférence de modes dans un codage basé pré-codage de sous-résolution

Dans ce chapitre, nous présentons des solutions rapides de codage adaptées au traitement des images inter-prédites B dans une structure de codage de type *Random Access*. A l’instar des contributions présentées dans le chapitre 4, la vidéo UHD est pré-codée dans une sous-résolution HD, moins complexe à coder. A partir des modes de codage obtenus, le codage UHD est accéléré en traitant uniquement les modes les plus probables. Nous reprenons ainsi l’architecture générale de la figure 4.2. La figure 4.3 illustre toutes les partitions PU disponibles pour le codage des images Intra HD et des images UHD. En codage Inter, il faut considérer pour chaque taille de CU, les partitions suivantes :

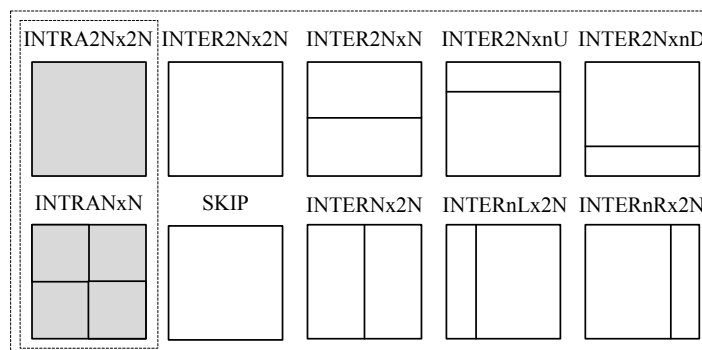


FIGURE 8.1 – Partitions et types de prédictions pour le codage d’un CU.

Nous rappelons que chaque PU inter-prédit (représentés avec un fond blanc) peut utiliser deux types de codage du mouvement : l’AMVP ou le MERGE. Chacun peut être utilisé en mono-prédiction (vers une image de référence de la liste L0 ou L1) ou en bi-prédiction. De plus, la partition carrée $INTER2N \times 2N$ peut être codée en MERGE sans transmettre de résidus : c’est le mode SKIP. Comme en Intra, nous utilisons une relation dyadique entre les partitions HD et UHD. La profondeur 0 correspond à la taille de CU 32×32 en HD et à la taille de CU 64×64 en UHD.

8.1 Analyse des partitionnements et des types de prédictions HD et UHD

Afin de déterminer la meilleure stratégie d’inférence pour les images Inter, nous établissons les probabilités conditionnelles du partitionnement UHD sachant la partition HD. Il

s'agit de réaliser d'une part le pré-codage de référence, et d'autre part le codage UHD de référence, puis d'analyser les corrélations entre les deux. Le jeu de séquences *EBU-UHD* et sa version réduite en HD sont codés à QP = 32 pour obtenir les résultats de la figure 8.2. La figure 8.3 représente la distribution des modes de codage de la partition HD et de ceux du partitionnement UHD co-localisé à la partition HD, en fonction du QP.

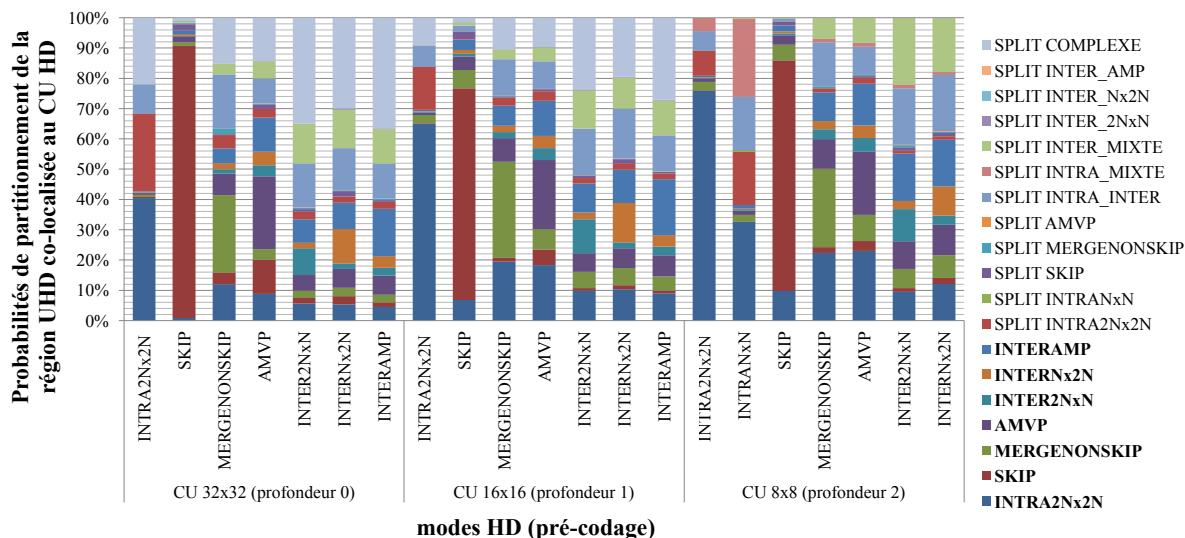


FIGURE 8.2 – Histogramme conditionnel du partitionnement UHD sachant la partition HD co-localisée (QP = 32), dans les images B.

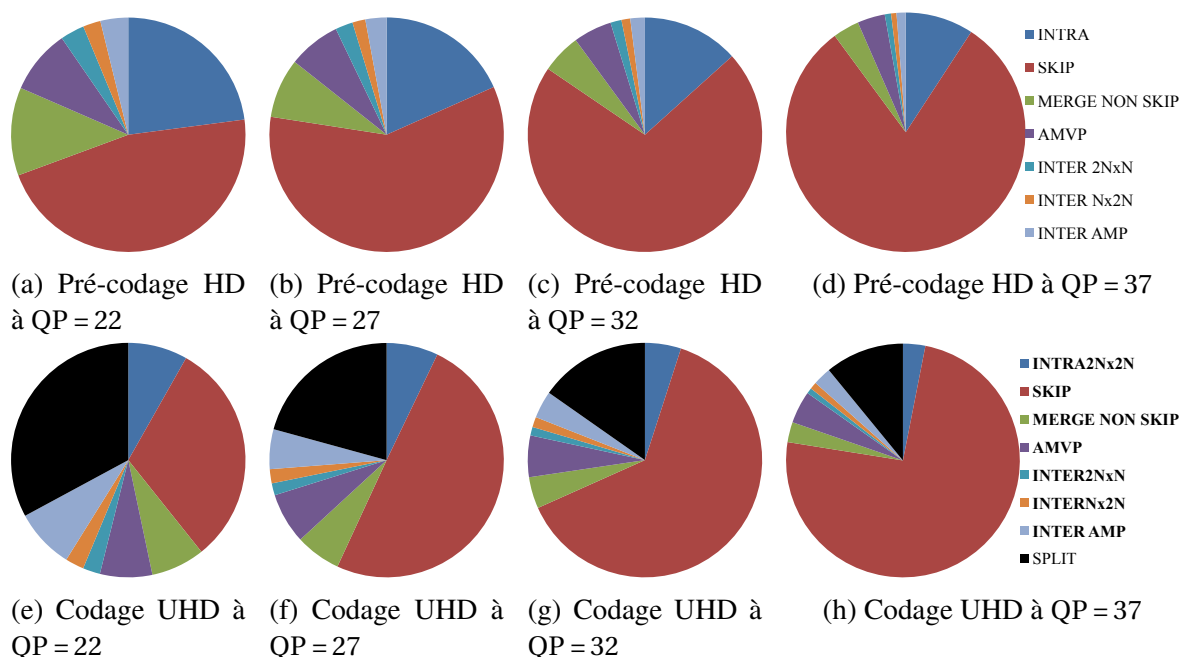


FIGURE 8.3 – Distribution des modes dans les partitions HD ((a) à (d)) et dans les partitionnements UHD co-localisé à la partition HD ((e) à (h)), dans les images B.

Cette fois, nous considérons les partitionnements non-carrés (INTER2NxN, INTERNx2N et INTERAMP), et les types de codage du mouvement SKIP, MERGENONSKIP et AMVP, lorsqu'il s'agit de PU Inter carrés (INTER2Nx2N). Les modes représentés en gras sur la légende de la figure 8.2 sont ceux pour lesquels la taille de CU est identique entre la HD et la UHD (NON-SPLIT). Les autres signifient que la région UHD co-localisée au CU HD est constituée de plusieurs CU (SPLIT). La catégorie noire en figure 8.3 illustre tous les cas de

SPLIT en UHD.

Ces statistiques apportent de nombreuses informations importantes, notamment en ce qui concerne le mode SKIP.

- En premier lieu, nous observons une forte proportion de SKIP en HD comme en UHD. L'utilisation massive du mode SKIP reflète d'ailleurs l'efficacité de HEVC pour décrire les régions stationnaires et homogènes en mouvement. Le cas SPLIT peut bien sûr produire des sous-partitions SKIP. Enfin, le mode SKIP est très fortement corrélé en UHD : plus de 90% pour la profondeur 0. Pour le reste, il y a très peu de cas SPLIT lorsque la partition HD est SKIP. Cette information essentielle va nous servir à construire une stratégie d'inférence efficace.
- La prédiction Intra représente entre 10 et 20% des blocs codés en HD. Elle est cependant moins représentée en UHD au profit du SPLIT, souvent constitué de quatre partitions Intra (SPLIT INTRA2Nx2N).
- Les modes MERGE et AMVP représentent environ 10 à 20% des blocs codés en HD. En UHD, ils se traduisent pour moitié par une seule partition carrée (Intra et Inter), et l'autre moitié par un partitionnement non-carré ou un SPLIT.
- Les partitions non-carrées (INTER2NxN, INTERNx2N et INTERAMP) en HD deviennent majoritairement un partitionnement non-carré ou un SPLIT en quatre CU et plus.

L'ensemble de ces observations nous serviront pour construire notre solution optimisée de codage en section 8.3.

8.2 Inférence du partitionnement

8.2.1 Stratégies d'inférence du partitionnement

En Intra, dans le chapitre 4, nous avons établi des stratégies d'inférence du partitionnement. Nous avons retenu *PU_UP* pour sa faible complexité. Dans la mesure où les possibilités de partitionnements et types de prédictions par CU sont plus importantes dans les images Inter, nous n'envisageons pas la stratégie *PU_UP*. Cependant, nous évaluons la stratégie *CU_UP*, pour laquelle le codeur UHD évalue uniquement le partitionnement CU HD mis à l'échelle. Ainsi, la boucle de codage UHD doit choisir entre les partitionnements de la figure 8.1 pour chaque CU. Seul cas particulier, lorsque l'on a un CU HD de type INTRANxN, donc composé de quatre prédictions Intra 4x4 : le codeur UHD va simplement évaluer quatre CU de tailles 8x8 avec le type de prédiction Intra.

La stratégie *CU_UP_SPLIT* permet au codeur d'évaluer les 4 CU fils du CU HD mis à l'échelle et ainsi de maximiser les chances d'obtenir le partitionnement optimal. La stratégie *CU_UP_MERGE&SPLIT* évalue également la fusion de quatre CU HD mis à l'échelle, à condition que les quatre CU soient de même nature.

En complément des statistiques de la section précédente, nous avons montré que seuls 6% des CU, pré-codés avec le type de prédiction Intra, basculent sur un type de prédiction Inter en UHD. Inversement, 13% des CU HD Inter basculent vers le type Intra en UHD. Grâce à cette observation, nous proposons une stratégie *PRED_UP_INTRA* basée sur *CU_UP* mais où seules les prédictions Intra sont évaluées en UHD lorsque le type Intra est utilisé en HD.

8.2.2 Résultats expérimentaux

Les résultats présentés en figure 8.4 sont réalisés selon les conditions expérimentales définies en section 2.1 avec la configuration RA et le jeu de séquences *EBU-UHD*. Nous avons traité seulement les images B avec le codage rapide. La complexité des images I (CRA) est marginale par rapport aux images B, et il n’y en a qu’une par seconde de vidéo. Par exemple, pour une vidéo de fréquence 50Hz, l’image CRA est insérée à chaque période Intra, soit une fois toutes les 48 images. Nous préférons maintenir un codage de référence sur ces images, le codage des images B étant fortement influencé par leur qualité de codage. Les résultats par séquence sont détaillés en Annexe E.1.

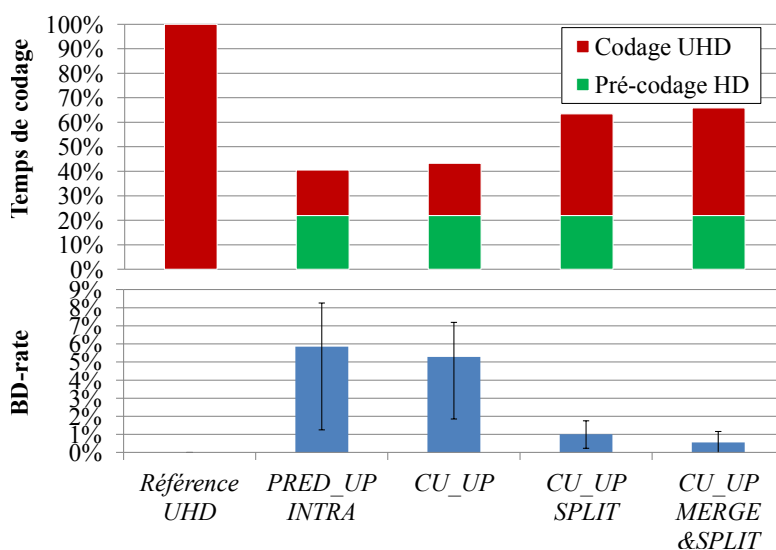


FIGURE 8.4 – Performance des stratégies d’inférence du partitionnement UHD par rapport au codage UHD de référence.

Tout d’abord, comme pour les images Intra, le codage de version HD (en vert) est 4.5 fois moins complexe que le codage UHD de référence.

Les stratégies d’inférence *CU_UP* et *PRED_UP_INTRA* sont les moins complexes, en réduisant la complexité de 56.8% et 59.5%, respectivement. Le *BD-rate* est réduit de 5.3% et de 5.8%, respectivement. La performance est assez comparable, cependant, nous remarquons sur les résultats par séquence que la perte *BD-rate* présente un écart-type plus important avec *PRED_UP_INTRA*.

Comme avec le codage Intra, *CU_UP_SPLIT* et *CU_UP_MERGE&SPLIT* favorisent davantage la qualité de reconstruction plutôt que la réduction de complexité. Les pertes *BD-rate* sont de 1% et 0.5%, respectivement. Les complexités sont réduites de 36.6% et 34.2%, respectivement. Toutes les régions UHD ne nécessitent pas d’être re-décomposées par rapport à la HD. Nous proposons dans la section suivante de conditionner le *SPLIT* selon le mode de prédiction de la partition HD pour obtenir, à la fois, une plus faible complexité que *CU_UP_SPLIT* et moins de perte de compression que *CU_UP*.

8.3 Inférence conditionnelle aux types de prédictions

8.3.1 Condition sur le mode SKIP

En section 8.1, nous avons observé une forte représentation du mode *SKIP* dans les codages HD et UHD, et une forte corrélation entre les deux codages. De plus, un *SPLIT* dans

la région UHD est peu probable lorsque la partition HD est SKIP. Ainsi, nous proposons une stratégie basée sur *CU_UP_SPLIT*, mais pour laquelle le *SPLIT* est conditionné par l'absence du mode *SKIP* dans la partition HD co-localisée. Nous l'appelons *NONSPPLIT_SKIP*. Dans une deuxième variante : *NONSPPLIT_SKIP_2*, le codeur UHD n'évalue que les partitions carrées (Intra et Inter) lorsque la partition HD est *SKIP*. Malgré les probabilités élevées d'obtenir le mode *SKIP* en UHD, contraindre le codage au seul mode *SKIP* dégrade très fortement la qualité. Nous avons écarté cette possibilité. Les résultats expérimentaux sont reportés en section 8.3.3.

8.3.2 Stratégie d'inférence optimisée

Enfin, nous proposons une stratégie d'inférence utilisant les observations faites en section 8.1. Nous l'appelons *OPTIMAL_UP*. Par rapport à l'histogramme conditionnel représenté en figure 8.2, nous avons construit un ensemble de règles dépendantes des types de prédictions, des partitionnements et des tailles de blocs obtenues avec le pré-codage de sous-résolution. Ces règles sont représentées par la table de correspondance 8.1, les cases grisées représentant les modes évalués par la boucle de codage UHD.

			NON-SPLIT						SPLIT																	
			INTRA2Nx2N	SKIP	INTER2Nx2N	INTER2NxN	INTERNx2N	INTERAMP	INTRA2Nx2N	INTRANxN	SKIP	INTER2Nx2N	INTER_2NxN	INTER_Nx2N	INTER_AMP											
Pré-codage HD	32x32 (0)	INTRA2Nx2N																								
		SKIP																								
		MERGENONSKIP																								
		AMVP																								
		INTER2NxN																								
		INTERNx2N																								
		INTERAMP																								
	16x16 (1)	INTRA2Nx2N																								
		SKIP																								
		MERGENONSKIP																								
		AMVP																								
		INTER2NxN																								
		INTERNx2N																								
		INTERAMP																								
	8x8 (2)	INTRA2Nx2N																								
		INTRANxN																								
		SKIP																								
		MERGENONSKIP																								
		AMVP																								
		INTER2NxN																								
		INTERNx2N																								

TABLEAU 8.1 – Table de correspondance associée à la stratégie d'inférence *OPTIMAL_UP*, pour les images B.

Ces règles ont été déterminées de manière à obtenir le mode optimal UHD avec une probabilité la plus élevée possible, mais en traitant le moins possible de modes UHD.

Les règles d'inférence sont les suivantes :

- Lorsque la partition HD est codée *INTRA2Nx2N*, nous choisissons d'évaluer systématiquement la région UHD co-localisée avec le mode *INTRA2Nx2N* ainsi que le mode *INTRA2Nx2N* dans les 4 CU fils. A profondeur 0, nous évaluons également tous les modes des 4 CU fils. Les probabilités conditionnelles en figure 8.2 montrent que la classe *SPLIT_COMPLEXE* est très représentée. Cela veut dire que le codage optimal produit un partitionnement de plus de quatre CU dans la région co-localisée. Pour limiter la complexité, nous préférons limiter la procédure à un seul SPLIT. Selon les probabilités conditionnelles, les chances de succès pour les profondeurs 0, 1 et 2, sont de 75.8%, 81% et 86.4%, respectivement. A profondeur 2, la partition HD peut être de type *INTRANxN*, donc composée de prédictions 4x4. Nous laissons le codeur UHD traiter tous les modes Intra (*INTRA2Nx2N* et *INTRANxN*) dans les quatre CU fils. C'est d'ailleurs dans ce cas essentiellement qu'apparaît l'*INTRANxN* sur les probabilités conditionnelles.
- Lorsque la partition HD est codée *SKIP*, nous traitons le CU mis à l'échelle avec des partitions carrées uniquement (Intra ou Inter), comme avec la stratégie *NONSPLIT_SKIP_2*. Les probabilités de succès sont cette fois de 93.3%, 86.3% et 93.5%, respectivement.
- Lorsque la partition HD est carrée et *MERGENONSKIP*, nous traitons le CU mis à l'échelle avec des partitions carrées (Intra ou Inter), mais également avec le SPLIT (tous les modes possibles dans les quatre CU fils). Les probabilités de succès sont de 73.7%, 76.6% et 80%, respectivement.
- Lorsque la partition HD est carrée et codée *AMVP*, nous traitons tous les modes du CU mis à l'échelle ainsi que le SPLIT. L'*AMVP* traduit souvent une certaine complexité du mouvement, ou une frontière entre deux objets vidéos. Cependant, on remarque sur les probabilités conditionnelles qu'aucun mode n'est réellement dominant. Ainsi, on traite le CU mis à l'échelle et le SPLIT, les probabilités de succès sont de 86.2%, 90.6% et 100%, respectivement.
- Lorsque la partition HD est non-carrée, les chances d'obtenir un partitionnement UHD de même orientation sont plus élevées que d'obtenir une des versions perpendiculaires. Par exemple, une partition *INTERNx2N* en HD a plus de chances de devenir une partition *INTERNx2N*, *INTERnRx2N* ou *INTERnLx2N* en UHD, que de devenir *INTER2NxN*, *INTER2NxN* ou *INTER2NxN*. Ainsi, dans notre solution, lorsqu'une partition HD est de type *INTERNx2N* ou *INTER2NxN*, on traite les trois partitionnements Inter non-carrés, ainsi que le SPLIT. Lorsqu'il s'agit des partitions AMP, on calcule seulement le même partitionnement en UHD ainsi que le SPLIT. La classe *SPLIT_COMPLEXE* est très présente sur les probabilités conditionnelles mais encore une fois, pour limiter la complexité nous limitons la procédure de partitionnement au seul SPLIT.

Les résultats expérimentaux sont reportés en section 8.3.3.

8.3.3 Résultats expérimentaux

Les résultats présentés en figure 8.5 sont réalisés selon les conditions expérimentales définies en section 2.1 avec la configuration RA et le jeu de séquences *EBU-UHD*. Seules les images B sont traitées avec notre algorithme. Les résultats par séquence sont détaillés en

Annexe E.2.

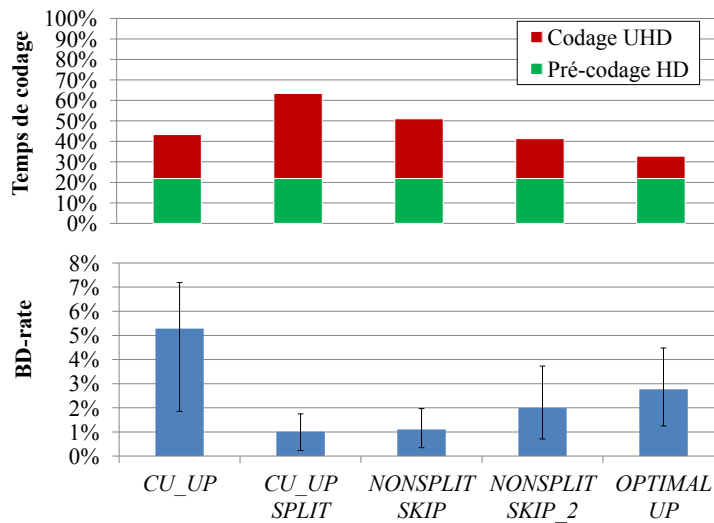


FIGURE 8.5 – Performance des stratégies d’inférence conditionnelle UHD par rapport au codage UHD de référence.

La stratégie *NONSPLIT_SKIP* réussit parfaitement à réduire la complexité par rapport à *CU_UP_SPLIT* (de 36.6% à 49%), sans pour autant augmenter la perte de compression (environ 1%). Cependant, la variante *NONSPLIT_SKIP_2* est encore plus performante car elle est moins complexe que *CU_UP*, et génère une perte de compression de seulement 2%, contre 5.3% pour *CU_UP*.

Enfin, la solution que nous retenons est la stratégie *OPTIMAL_UP*, car elle produit le meilleur compromis réduction de complexité / perte de compression. Contre 2.8% de perte *BD-rate*, la complexité est réduite de 67.2%, ce qui correspond à une complexité réduite d’un facteur 3. Autre chiffre marquant : la passe de codage UHD, considérée seule, est 9.2 fois moins complexe que le codage de référence UHD, ce qui prouve l’efficacité des règles d’inférence établies dans ce chapitre.

8.4 Inférence du partitionnement et des types de prédiction, combinée à une configuration de codage rapide.

8.4.1 Principe

Nous avons défini dans le chapitre 7 une configuration de codage rapide basée sur des simplifications du modèle de référence, imitant les systèmes de compression temps réel. Nous avons mesuré une réduction de complexité de 91% en UHD. Nous voulons estimer le bénéfice de notre approche d’inférence en sus de cette configuration optimisée, contraignant déjà fortement les degrés de liberté du codage HEVC. Nous traitons les deux passes de codage (HD et UHD) avec la configuration de codage rapide et nous appliquons les règles d’inférence, définies dans les sections précédentes, sur le codeur UHD. En ce qui concerne la stratégie d’inférence *OPTIMAL_UP*, les règles utilisées sont toujours celles de la table de correspondance 8.1 mais en ignorant bien sûr les partitionnements non-carrés.

8.4.2 Résultats expérimentaux

Les résultats expérimentaux sont reportés en figure 8.6. Les résultats par séquence sont détaillés en Annexe E.3.

Dans le cas de la stratégie *CU_UP*, la perte *BD-rate* est très importante (9%). Alors que cette stratégie produisait seulement 5% de perte *BD-rate* environ pour le codage non optimisé, ici le pré-codage HD est très contraint et le codage UHD n'a quasiment plus de degrés de liberté : seulement le choix de la prédiction Intra ou Inter. Seules les partitions carrées, et les tailles de TU identiques à la taille CU sont autorisées. Aucun raffinement du partitionnement ne peut venir compenser les erreurs ou tout au moins les choix de codage non optimaux apportés par le partitionnement HD. De plus, la réduction de complexité est bien moins importante lorsque l'on se compare à cette référence optimisée (32.8%).

Avec la stratégie *CU_UP_SPLIT*, le *SPLIT* apporte un degré de liberté supplémentaire au codage UHD, ainsi la perte de compression est ramenée à seulement 2.2%, mais le codage des deux flux HD et UHD est seulement 15.3% moins complexe que le codage de référence UHD.

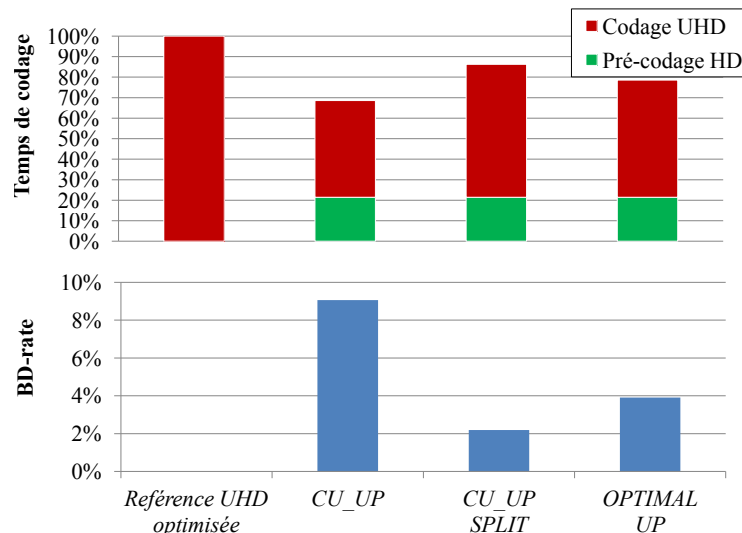


FIGURE 8.6 – Performance des stratégies d'inférence du partitionnement UHD par rapport à la référence de codage optimisée.

Notre stratégie d'inférence *OPTIMAL_UP* produit une perte *BD-rate* de 3.9% et une complexité réduite de 23%. Le bénéfice en réduction de complexité est moins important par rapport à la stratégie *CU_UP*, mais cela reste le meilleur compromis réduction de complexité / perte de compression.

8.5 Conclusion

Dans ce chapitre, nous proposons une solution rapide de codage des images inter-prédites au format UHD. Pour cela, nous utilisons le partitionnement et les types de prédictions obtenus en codant une version réduite de la vidéo source. Avec la stratégie *OPTIMAL_UP*, nous obtenons le meilleur compromis réduction de complexité / perte de compression : la complexité est réduite d'un facteur 3 contre une perte inférieure à 3%, par rapport à un codage de référence. Au niveau de la qualité subjective, l'algorithme ne dégrade pas la qualité vidéo lorsque l'on se compare, à même débit, au codage de référence devant un écran UHD. Contrairement aux algorithmes de la littérature (voir le chapitre 6), notre solution n'est pas

dépendante des informations de codage du voisinage et/ou des images précédentes. Ainsi, la présence de transitions de scènes, ou de fondus enchainés ne provoquera pas de dégradations de la performance. Le traitement sur des architectures massivement parallèles est aussi facilité : il est plus facile de tirer partie d'un outil comme le WPP (section 1.3.7). L'approche n'est pas non plus basée sur des terminaisons rapides, pour lesquelles la réduction de complexité va varier en fonction du QP et de la complexité du contenu.

Pour les images Intra, la meilleure solution était de traiter simplement la mise à l'échelle du partitionnement HD, ce qui entraînait un léger sous-partitionnement en UHD. Notre solution pour les images Inter tient compte des types de prédictions HD et de leurs propriétés. Certains modes nécessitent de traiter des partitions plus fines dans la région UHD co-localisée alors que le mode SKIP, massivement utilisé, ne nécessite pas de re-décomposer la région UHD.

Nous avons également estimé la performance de notre approche dans un système de compression réel, présentant déjà des simplifications du codage. Pour cela, nous avons utilisé la stratégie d'inférence en appliquant les simplifications du chapitre 7 sur les deux passes de codage. Les simplifications limitent déjà fortement les degrés de liberté du codage. Utiliser en sus une stratégie d'inférence dans le codeur UHD les réduit encore plus. La stratégie *OPTIMAL_UP* maintient une perte *BD-rate* acceptable (3.9%) et réduit la complexité d'un peu plus de 20% par rapport à un codage présentant des simplifications. Au final, la stratégie d'inférence *OPTIMAL_UP*, combinée avec les simplifications du codage, permet un codage 13.4 fois plus rapide que le codage de référence pour traiter la UHD. Le codage UHD seul est quant à lui 18 fois moins complexe, ce qui est un résultat très prometteur pour développer un système de compression, où plusieurs codages sont produits à faible complexité à partir d'un codage pivot HD.

Conclusion générale

Rappel des objectifs

L'objectif de cette thèse était de proposer des algorithmes rapides pour le codage vidéo avec le nouveau standard HEVC. Ces travaux s'inscrivent dans l'évolution de la vidéo numérique vers les formats à très haute définition UHD, nécessitant un taux de compression plus élevé, mais intrinsèquement, un coût calcul plus important.

Pour référence, nous considérons un codeur HEVC maximisant le compromis débit-distorsion, c'est-à-dire paramétré pour tester et comparer tous les degrés de liberté offerts par le standard. Cela n'est pas réaliste en terme de complexité si l'on veut traiter des formats UHD. Nous cherchons dans nos travaux à limiter ces tests tout en maintenant un taux de compression proche du codage de référence.

Deux axes de recherche ont été définis initialement, le premier utilise un pré-codage de la vidéo sous-échantillonnée. Le pré-codage permettrait d'extrapoler les meilleurs modes de codage pour coder la pleine résolution. Le deuxième axe défini envisage de coupler les caractéristiques locales du signal vidéo avec les modes de codage HEVC optimaux. Ainsi, nous pouvons apprendre une base de connaissance permettant de pré-sélectionner les modes de codage à tester et ainsi réduire la complexité.

Synthèse des travaux et des résultats obtenus

Dans cette thèse, nous avons traité en premier lieu le codage des images UHD en Intra, à travers le GOP *All Intra*, puis le codage des images UHD en Inter, à travers les images B du GOP *Random Access*.

Réduction de complexité du codage des images Intra

Pour la réduction de complexité des images Intra, nous avons développé une approche (chapitre 4) dans laquelle la vidéo UHD réduite en HD est pré-codée, puis les modes de codage obtenus permettent d'inférer les modes les plus probables pour le codage UHD. Dans notre méthode, seul le partitionnement HD mis à l'échelle est traité dans le codage UHD. Pour les modes de prédiction Intra, nous avons mis en œuvre une stratégie efficace, utilisant un nombre restreint de candidats, le mode utilisé dans la partition co-localisée dans le codage HD étant inclus comme candidat.

- Au format UHD, notre méthode permet de réduire la complexité d'un facteur 3, en considérant le codage des deux flux HD et UHD par rapport à un codage UHD unique, pour une perte de compression d'environ 5% en débit à PSNR équivalent. Si l'on possède déjà le flux HD, le temps de calcul associé au codage du flux UHD est réduit d'un facteur 11.9.
- L'intérêt de cette approche est d'offrir des performances stables assez indépendantes des contenus vidéos et de la quantification, ce qui n'est généralement pas le cas des

approches traditionnelles de réduction de complexité d'un codeur (voir le chapitre 3).

Dans un second temps, nous avons développé une approche (chapitre 5) utilisant les caractéristiques locales de la vidéo source pour simplifier la procédure de partitionnement PU. Cette approche ne nécessite pas de pré-codage, mais nécessite une phase d'apprentissage. Réalisée hors-ligne ou en ligne, elle permet de déterminer des seuils de décision sur la variance du bloc de luminance. Lors du codage, ces seuils permettent de sélectionner les tailles de PU les plus probables et de ne pas traiter les moins probables. Les résultats obtenus avec cette approche sont satisfaisants :

- Avec un seul paramètre d'entrée, notre algorithme de partitionnement rapide permet de viser plusieurs points de fonctionnement (compromis réduction de complexité / perte de compression).
- Sans aucune perte de compression, la complexité est réduite de 45%. Avec une perte de compression de 6.6%, elle atteint 83.6%, soit un temps de calcul réduit d'un facteur 6.
- Pour réduire davantage la complexité, on peut combiner l'algorithme avec des simplifications sur le choix des modes de prédiction et des tailles de TU.

En appliquant les seuils de décisions à des séquences différentes de l'ensemble d'apprentissage, les performances sont maintenues, ce qui montre que les seuils de décision peuvent être suffisamment génériques pour être utilisés avec la plupart des contenus. Cependant, pour assurer l'adaptation des paramètres au cours du temps, nous suggérons tout de même de réaliser l'apprentissage en ligne (*online*), pour enrichir l'apprentissage au cours du codage, ou bien de combiner l'approche avec l'algorithme de pré-codage, qui ne dépend pas d'un apprentissage.

Réduction de complexité du codage des images Inter

Pour la réduction de complexité des images Inter, nous avons étudié l'inférence de modes à partir d'un codage en sous-résolution. Cette approche a fourni des résultats satisfaisants :

- En mettant en œuvre une inférence de partitionnement conditionnée aux types de prédictions, nous avons pu réduire par un facteur 3 le temps de calcul associé au codage des deux flux HEVC HD et UHD par rapport à un codage UHD unique, pour une perte de compression limitée à 2.8% (un facteur 9.2 pour le codage du flux UHD dès lors qu'on possède le flux HD),
- Comme avec les images Intra, cette approche offre des performances stables assez indépendantes des contenus vidéos et de la quantification, ce qui n'est généralement pas le cas des approches traditionnelles de réduction de complexité d'un codeur (voir le chapitre 6).
- Cette approche peut être combinée aux autres stratégies de réduction de complexité mises en œuvre dans les codeurs de diffusion (restrictions d'outils, terminaisons rapides). La réduction de complexité complémentaire fournie par notre approche est bien sûr moindre par rapport à un codage optimisé mais reste significative : 23% si l'on considère les 2 flux HD et UHD produits et 44% sur le flux UHD, pour une perte de compression limitée à 4% et non visible.

Intérêt industriel et applications possibles

Les stratégies de réduction de complexité mises en œuvre au cours de cette thèse pour le codage Intra et Inter offrent des perspectives intéressantes pour le développement de codeurs

HEVC UHD plus économes en ressources de calculs mais ne sacrifiant pas la qualité de compression. Elles sont particulièrement adaptées au domaine de la WebTV/OTT qui prend une part croissante dans la diffusion de la vidéo et pour lequel le signal vidéo est codé à des résolutions multiples pour adresser des réseaux et des terminaux de capacités variées.

Actuellement, il existe des codeurs UHD HEVC temps réel sur le marché, essentiellement sur plateforme PC, donc dédiés à la WebTV/OTT. Cependant, pour offrir des performances proches des 50% de gain en taux de compression par rapport à H.264/AVC, il est nécessaire d'utiliser un nombre important de coeurs de calcul, et ainsi des équipements très coûteux. A l'inverse, les architectures à faible coût (utilisant moins de coeurs de calcul) font l'impasse sur de nombreux outils HEVC et produisent ainsi une compression moins performante. Cette thèse est donc en phase avec des intérêts industriels à court terme, car nous proposons des solutions pour réduire le coût des équipements en maintenant une qualité de codage élevée. Les solutions pour le *broadcast* (IPTV / terrestre / satellite) sont traditionnellement présentes sur des architectures *hardware* (circuits dédiés). Ces architectures dédiées utilisent massivement la parallélisation et ne sont pas forcément adaptées à des terminaisons rapides telles que celles proposées dans le chapitre 5. Pour ces solutions, l'enjeu industriel est à plus long terme dans la mesure où le renouvellement matériel est dépendant des normes en vigueur dans le pays et donc au renouvellement du parc de décodeurs.

Perspectives pour de futurs travaux

Les travaux de cette thèse offrent de nombreuses perspectives d'études complémentaires.

Amélioration des stratégies d'inférence de modes Les travaux menés ont essentiellement cherché à réduire la complexité liée aux tailles de blocs et aux types de prédictions. Pour améliorer la performance de notre algorithme d'inférence à partir d'un codage HD, nous pourrions étendre les analyses statistiques sur d'autres modes de codage, comme les images de référence, les tailles de transformées, les MV, les candidats au mode MERGE et AMVP, afin de réduire au minimum les tests réalisés par le codeur UHD.

Ajustement local de quantification La performance de codage a été optimisée et mesurée selon une métrique objective (le *BD-rate*). Cependant, les systèmes réels sont développés pour maximiser la qualité subjective. La technique la plus répandue est d'utiliser un ajustement local de quantification (ALQ), c'est-à-dire d'appliquer un QP variable pour chaque CU codé dans le *bitstream*. L'œil est moins sensible à une plus forte quantification dans les zones de mouvement et de faibles gradients. A l'inverse, il est préférable de maximiser le débit alloué aux zones de détails et de faible activité temporelle. Ainsi, il faudrait appliquer l'ALQ dans nos algorithmes, et en particulier avec l'inférence de modes, de déterminer la meilleure stratégie d'ALQ pour chaque passe de codage.

Codage pivot Dans le chapitre 4, nous avons introduit l'idée d'un codage "pivot" permettant, sur le principe de l'inférence de modes, de proposer plusieurs codages à faible complexité, pour les systèmes de WebTV/OTT. Dans cette thèse, nous avons uniquement étudié la conversion HD vers UHD, qui permet de réduire fortement le coût de codage UHD. Un système WebTV/OTT typique chercherait à décliner la vidéo source en plusieurs flux allant de la résolution native UHD à la résolution *Standard Definition* (SD), avec différents débits, fréquences et standards de compression. Dans ce cas, le choix d'une résolution intermédiaire HD comme format "pivot" est le plus adapté, car il permet d'inférer vers les résolutions les plus proches, donc partageant le plus d'informations. Pour cela, des stratégies d'inférence complémentaires doivent

être développées, notamment pour inférer le codage de la résolution inférieure ou d'un débit de codage inférieur.

Apprentissage sur une ressource flexible La mise en œuvre de l'algorithme du chapitre 5 nécessite d'apprendre (hors-ligne ou en ligne) une base de connaissance (des seuils de décision). Les nouvelles plateformes virtualisées *Cloud* offre une grande flexibilité dans l'usage de la ressource de calcul. C'est une opportunité pour intégrer des systèmes d'apprentissage dans des produits à vocation temps réel.

Implémentation temps réel Nos mesures ont été réalisées sur le modèle de référence. Notre contribution du chapitre 8, aucune optimisation de code spécifique à une architecture de calcul n'a été réalisée. Nos algorithmes ont été appliqués au logiciel de référence, écrit en C, sans parallélisation et autres optimisations spécifiques. Il serait intéressant d'appliquer nos contributions à des systèmes réels (sur architecture PC ou circuits dédiés), déjà fortement optimisés car contraints par le traitement en temps réel.

Bibliographie

- [1] T. WIEGAND, G. SULLIVAN et A. LUTHRA : Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification. Rapport technique JVT-G050r1, Joint Video Team (JVT), 8th meeting, Geneva, CH, mai 2013. 6
- [2] ITU-T/ISO/IEC JCT ON VIDEO CODING : Terms of reference of the joint collaborative team on video coding standard development. Rapport technique, 39th meeting : Kyoto, JP, janvier 2010. 7
- [3] VCEG : Joint Call for Proposals on Video Compression Technology. Rapport technique VCEG-AM91, VCEG, 39th meeting, Kyoto, JP, janvier 2010. 7
- [4] I. AMONOU, N. CAMMAS, G. CLARE, J. JUNG, L. NOBLET et AL. : Description of video coding technology proposal by NTT, NTT DOCOMO, Orange Labs, Panasonic and Technicolor. Rapport technique JCTVC-A114, JCT-VC, 1st meeting, Dresden, DE, avril 2010. 7
- [5] ITU-T : Recommendation H.265 : High Efficiency Video Coding. Rapport technique version 1, ITU-T, avril 2013. 7, 10, 18
- [6] T. TAN, M. MRAK, V. BARONCINI et N. RAMZAN : HEVC verification test results. Rapport technique JCTVC-Q0204, JCT-VC, 17th meeting : Valencia, ES, avril 2014. 7
- [7] ITU-T : Recommendation H.265 : High Efficiency Video Coding. Rapport technique version 2, ITU-T, octobre 2014. 7
- [8] ITU-T : Recommendation H.265 : High Efficiency Video Coding. Rapport technique version 3, ITU-T, avril 2015. 7
- [9] ITU-R REC. : Parameter values for the HDTV standards for production and international programme exchange. Rapport technique ITU-REC-BT.709-6, juin 2015. 10
- [10] J. LAINEMA, F. BOSSEN, W.-J. HAN, J. MIN et K. UGUR : Intra Coding of the HEVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, décembre 2012. 16
- [11] I.-K. KIM : High Efficiency Video Coding (HEVC) Test Model 11 (HM11) Encoder Description. Rapport technique JCTVC-M1002, JCT-VC, 13th meeting : Incheon, KR, avril 2013. 17, 23, 26
- [12] P. HELLE, S. OUDIN, B. BROSS, D. MARPE, M.-O. BICI, K. UGUR, J. JUNG, G. CLARE et T. WIEGAND : Block Merging for Quadtree-Based Partitioning in HEVC. *IEEE Transactions on Circuits and Systems for Video Technology*, décembre 2012. 18

- [13] K. R. RAO et P. YIP : *Discrete Cosine Transform : Algorithms, Advantages, Applications*. Academic Press Professional, Inc., San Diego, CA, USA, 1990. 19
- [14] A. FULDSETH, G. BJØNTEGAARD, M. BUDAGAVI et V. SZE : CE10 : Core transform design for HEVC. Rapport technique JCTVC-G495, JCT-VC, 7th meeting : Geneva, CH, novembre 2011. 19
- [15] J. STANKOWSKI, C. KORZENIEWSKI, M. DOMANSKI et T. GRAJEK : Rate-distortion optimized quantization in HEVC : Performance limitations. *In Picture Coding Symposium (PCS)*, mai 2015. 20
- [16] V. SZE et M. BUDAGAVI : High Throughput CABAC Entropy Coding in HEVC. *IEEE Transactions on Circuits and Systems for Video Technology*, décembre 2012. 20
- [17] A. NORKIN, G. BJØNTEGAARD, A. FULDSETH, M. NARROSCHE et M. et al IKEDA : HEVC Deblocking Filter. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), décembre 2012. 21
- [18] C.-M. FU, E. ALSHINA, A. ALSHIN, Y.-W. HUANG, C.-Y. CHEN, C.-Y. TSAI, C.-W. HSU, S.-M. L., J.-H. PARK et W.-J. HAN : Sample Adaptive Offset in the HEVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, décembre 2012. 21
- [19] A. ORTEGA et K. RAMCHANDRAN : Rate-distortion methods for image and video compression. *Signal Processing Magazine*, novembre 1998. 22
- [20] G.J. SULLIVAN et T. WIEGAND : Rate-distortion optimization for video compression. *Signal Processing Magazine*, novembre 1998. 22
- [21] Y. PIAO, J. MIN et J. CHEN : Encoder improvement of unified intra prediction. Rapport technique JCTVC-C207, JCT-VC, 3rd meeting : Guangzhou, CN, octobre 2010. 24, 38
- [22] L. ZHAO, L. ZHANG, S. MA et D. ZHAO : Fast mode decision algorithm for intra prediction in HEVC. *In Visual Communications and Image Processing (VCIP)*, 2011. 24
- [23] EUROPEAN BROADCASTING UNION : UHD-1 test sequences. <https://tech.ebu.ch/Jahia/site/tech/cache/offonce/testsequences/uhd-1>. 27
- [24] ITU-T : Subjective video quality assessment methods for multimedia applications. Rapport technique, ITU-T, 2008. 27
- [25] L. HAGLUND : The SVT High Definition Multi Format Test Set. Rapport technique, 2006. 28
- [26] L. SONG, X. TANG, W. ZHANG, X. YANG et P. XIA : The SJTU 4k video sequence dataset. *In QoMEX*, 2013. 28
- [27] G. BJØNTEGAARD : Calculation of average PSNR differences between RD-curves. *In Proc. ITU-T SG16/Q.6 VCEG 13th Meeting*, Austin, TX, avril 2001. VCEG-M33. 29
- [28] L. SHEN, Z. ZHANG et P. AN : Fast CU size decision and mode decision algorithm for HEVC intra coding. *IEEE Transactions on Consumer Electronics*, 2013. 36, 37

- [29] J. KIM, Y. CHOE et Y.-G. KIM : Fast Coding Unit size decision algorithm for intra coding in HEVC. *In International Conference on Consumer Electronics (ICCE)*, janvier 2013. 37
- [30] S. CHO et M. KIM : Fast CU Splitting and Pruning for Suboptimal CU Partitioning in HEVC Intra Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2013. 37
- [31] X. SHEN, L. YU et J. CHEN : Fast coding unit size selection for HEVC based on Bayesian decision rule. *In Picture Coding Symposium (PCS)*, mai 2012. 37
- [32] G. TIAN et S. GOTO : Content adaptive prediction unit size decision algorithm for HEVC intra coding. *In Picture Coding Symposium (PCS)*, mai 2012. 37
- [33] M. ZHANG, J. QU et H. BAI : Entropy-Based Fast Largest Coding Unit Partition Algorithm in High-Efficiency Video Coding. *Entropy*, juin 2013. 37, 56, 104
- [34] G. CHEN, L. SUN, Z. LIU et T. IKENAGA : Fast mode and depth decision HEVC intra prediction based on edge detection and partitioning reconfiguration. *In International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS)*, novembre 2013. 38, 109
- [35] H. HUANG, Y. ZHAO, C. LIN et H. BAI : Fast bottom-up pruning for HEVC intraframe coding. *In Visual Communications and Image Processing (VCIP)*, novembre 2013. 38
- [36] M.U.K. KHAN, M. SHAFIQUE et J. HENKEL : An adaptive complexity reduction scheme with fast prediction unit decision for HEVC Intra encoding. *In International Conference on Image Processing (ICIP)*, 2013. 36, 38
- [37] Y. KIM, D. JUN, S.-h. JUNG, J. S. CHOI et J. KIM : A Fast Intra-Prediction Method in HEVC Using Rate-Distortion Estimation Based on Hadamard Transform. *ETRI Journal*, avril 2013. 39
- [38] Feng PAN, Xiao LIN, S. RAHARDJA, K.P. LIM, Z.G. LI, Dajun WU et Si WU : Fast mode decision algorithm for intraprediction in H.264/AVC video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, juillet 2005. 39
- [39] W. JIANG, H. MA et Y. CHEN : Gradient based fast mode decision algorithm for intra prediction in HEVC. *In International Conference on Consumer Electronics, Communications and Networks (CECNet)*, avril 2012. 39
- [40] C.-M. FANG, Y.-T. CHANG et W.-H. CHUNG : Fast intra mode decision for HEVC based on direction energy distribution. *In International Symposium on Consumer Electronics (ISCE)*, 2013. 39
- [41] S. YAN, L. HONG, W. HE et Q. WANG : Group-Based Fast Mode Decision Algorithm for Intra Prediction in HEVC. *In International Conference on Signal Image Technology and Internet Based Systems (SITIS)*, 2012. 39
- [42] W.C. SIU et L.L. WANG : Novel Adaptive Algorithm for Intra Prediction with Compromised Modes Skipping and Signaling Processes in HEVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 2013. 39

- [43] Y. ZHANG, Z. LI et B. LI : Gradient-based fast decision for intra prediction in HEVC. *In Visual Communications and Image Processing (VCIP)*, novembre 2012. 40
- [44] J. KIM, J. YANG, H. LEE et B. JEON : Fast intra mode decision of HEVC based on hierarchical structure. *In International Conference on Information, Communications and Signal Processing (ICICS)*, 2011. 40
- [45] J. SU, K. NITTA, M. IKEDA et A. SHIMIZU : Residue role assignment based transform partition predetermination on HEVC. *In International Conference on Image Processing (ICIP)*, septembre 2013. 40
- [46] K. CHOI et E.S. JANG : Early TU decision method for fast video encoding in high efficiency video coding. *Electronics Letters*, 2012. 40
- [47] S.-W. TENG, H.-M. HANG et Y.-F. CHEN : Fast mode decision algorithm for Residual Quadtree coding in HEVC. *In Visual Communications and Image Processing (VCIP)*, novembre 2011. 40, 79
- [48] Y. ZHANG, Z. LI, M. ZHAO et B. LI : Fast residual quad-tree coding for the emerging high efficiency video coding standard. *Communications, China*, octobre 2013. 41
- [49] Q. DAI, D. ZHU et R. DING : Fast mode decision for inter prediction in H.264. *In International Conference on Image Processing (ICIP)*, octobre 2004. 42
- [50] Z. ZHAO, J. SI, J. OSTERMANN et W. LI : Inter-layer intra mode coding for the scalable extension of HEVC. *In International Symposium on Circuits and Systems (ISCAS)*, mai 2013. 42
- [51] E. PEIXOTO, B. MACCHIAVELLO, E. M. HUNG, A. ZAGHETTO, T. SHANABLEH et E. IZQUIERDO : An h.264/avc to hevc video transcoder based on mode mapping. *In International Conference on Image Processing (ICIP)*, sept 2013. 43
- [52] J. CHEN, J. BOYCE, Y. YAN et M. M. HANNUKSELA : Scalable HEVC (SHVC) Test Model 8 (SHM 8). Rapport technique JCTVC-S1007, JCT-VC, 19th Meeting : Strasbourg, FR, octobre 2014. 43
- [53] N. DHOLLANDE, O. LE MEUR et C. GUILLEMOT : HEVC Intra coding of Ultra HD video with reduced complexity. *In International Conference on Image Processing (ICIP)*, Paris, octobre 2014. 53
- [54] University of Waikato MACHINE LEARNING GROUP : Weka 3 : Data mining software in java. <http://www.cs.waikato.ac.nz/ml/weka/>. 56
- [55] O. LE MEUR, J. GAUTIER et C. GUILLEMOT : Exemplar-based inpainting based on local geometry. *In International Conference on Image Processing (ICIP)*, 2011. 56, 104
- [56] N. DHOLLANDE, X. DUCLOUX, O. LE MEUR et C. GUILLEMOT : Fast block partitioning method in HEVC Intra coding for UHD video. *In International Conference on Consumer Electronics (ICCE-Berlin)*, Berlin, 2015. 60
- [57] J. KIM, J. YANG, K. WON et B. JEON : Early determination of mode decision for hevc. *In Picture Coding Symposium (PCS)*, mai 2012. 73

- [58] H. LEE, H. J. SHIM, Y. PARK et B. JEON : Early Skip Mode Decision for HEVC Encoder With Emphasis on Coding Quality. *IEEE Transactions on Broadcasting*, 2015. 73
- [59] Z. PAN, S. KWONG, M. SUN et J. LEI : Early MERGE Mode Decision Based on Motion Estimation and Hierarchical Depth Correlation for HEVC. *IEEE Transactions on Broadcasting*, juin 2014. 73
- [60] G. CORREA, P. ASSUNCAO, L. AGOSTINI et L. A. CRUZ : A method for early-splitting of HEVC inter blocks based on decision trees. *In European Signal Processing Conference (EUSIPCO)*. IEEE, 2014. 74
- [61] J. VANNE, M. VIITANEN et T. D. HAMALAINEN : Efficient mode decision schemes for HEVC inter prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2014. 74
- [62] K. CHOI, S. H. PARK et E-S. JANG : Coding tree pruning based CU early termination. Rapport technique JCTVC-F092, JCT-VC, 6th meeting : Torino, IT, juillet 2011. 74
- [63] L. SHEN, Z. ZHANG et Z. LIU : Effective CU Size Decision for HEVC Intracoding. *IEEE Transactions on Image Processing*, octobre 2014. 74
- [64] J. LENG, L. SUN, T. IKENAGA et S. SAKAIDA : Content Based Hierarchical Fast Coding Unit Decision Algorithm for HEVC. *In International Conference on Multimedia and Signal Processing*, Washington, DC, USA, 2011. 74
- [65] W. ZHAO, T. ONOYE et T. SONG : Hierarchical Structure-Based Fast Mode Decision for H.265/HEVC. *IEEE Transactions on Circuits and Systems for Video Technology*, octobre 2015. 74, 75
- [66] X. HOU et Y. XUE : Fast coding unit partitioning algorithm for HEVC. *In International Conference on Consumer Electronics (ICCE)*, janvier 2014. 75
- [67] M. B. CASSA, M. NACCARI et F. PEREIRA : Fast rate distortion optimization for the emerging HEVC standard. *In Picture Coding Symposium (PCS)*, 2012. 75
- [68] S. AHN, B. LEE et M. KIM : A Novel Fast CU Encoding Scheme Based on Spatio-temporal Encoding Parameters for HEVC Inter Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, mars 2015. 75
- [69] Q. YU, X. ZHANG, S. WANG et S. MA : Early termination of coding unit splitting for HEVC. *In Asia-Pacific Signal Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2012. 76
- [70] X. SHEN et L. YU : CU splitting early termination based on weighted SVM. *EURASIP Journal on Image and Video Processing*, janvier 2013. 76

Annexes

A Séquences utilisées

A.1 EBU-UHD

nom	format	description	capture
<i>candle smoke</i>	3840x2160, 750 images, 50 Hz, 8 bits	Plan fixe sur un personnage avec textures complexes (cheveux). Fumée.	
<i>fountain lady</i>	"	Femme en mouvement près d'une fontaine d'eau. Jets d'eau.	
<i>lupo boa</i>	"	Plan fixe sur un personnage en mouvement avec un boa (écharpe).	
<i>lupo confetti</i>	"	Plan fixe sur un personnage en mouvement. Confettis et cotillons.	
<i>park dancers</i>	"	2 danseurs dans un parc. Fine textures de végétation. 2 plans de caméra.	
<i>pendulus wide</i>	"	Plan fixe sur un présentoir en rotation avec des textures fines et un pendule.	
<i>studio dancer</i>	"	2 danseurs en mouvement dans un studio. Arrière-plan flou.	
<i>veggie fruits</i>	"	Plan fixe sur une corbeille de fruit en rotation.	
<i>waterfall pan</i>	"	Grand <i>panning</i> sur un plan d'eau, la végétation et une cascade d'eau.	
<i>wind wool</i>	"	Plan en mouvement sur des éoliennes, et des objets texturés (cordes, tissus).	

TABLEAU 2 – Jeu de séquences *EBU-UHD*

A.2 ClasseB-HD






nom	format		description	capture
<i>BasketballDrive</i>	1920x1080, images, 50 Hz, 8 bits	500	Scène de basket en salle. Importants mouvements de caméra.	
<i>BQTerrace</i>	1920x1080, images, 60 Hz, 8 bits	600	Grand <i>panning</i> sur un pont en ville. Grilles générant de l'aliasing.	
<i>Cactus</i>	1920x1080, images, 50 Hz, 8 bits	500	Plan fixe sur différents objets en mouvement (plantes, pendule, cartes).	
<i>Kimono</i>	1920x1080, images, 24 Hz, 8 bits	240	Suivi d'une femme en kimono s'approchant d'une maison.	
<i>ParkScene</i>	1920x1080, images, 24 Hz, 8 bits	240	Suivi de cyclistes évoluant dans un parc	

TABLEAU 3 – Jeu de séquences *ClasseB-HD*

B Performance HEVC

		<i>All Intra (AI)</i>		<i>Random Access (RA)</i>	
Séquences		BD-rate	ΔT	BD-rate	ΔT
EBU-UHD	candle_smoke	-21.08%	-24.0%	-50.47%	-38.7%
	fountain_lady	-13.72%	-21.5%	-28.88%	-48.3%
	lupo_boa	-14.85%	-24.1%	-23.75%	-51.5%
	lupo_confetti	-17.65%	-22.7%	-28.00%	-56.1%
	park_dancers	-5.41%	-18.6%	-16.72%	-48.8%
	pendulus_wide	-3.75%	-25.9%	-12.79%	-47.4%
	studio_dancer	-17.67%	-25.8%	-33.27%	-55.2%
	veggie_fruits	-8.64%	-26.0%	-30.70%	-48.0%
	waterfall_pan	-4.36%	-25.5%	-14.07%	-46.1%
	wind_wool	-10.77%	-27.9%	-34.47%	-45.8%
	Moyenne	-11.79%	-24.3%	-27.31%	-49.2%

TABLEAU 4 – Performance de codage apportée par les tailles CU 32x32 et 64x64 dans HEVC : résultats par séquence.

C Codage rapide des images I basé sur le pré-codage de sous-résolution

Séquences	CU_UP		PU_UP		CU_UP_SPLIT		CU_UP_MERGE&SPLIT		
	BD-rate	ΔT	BD-rate	ΔT	BD-rate	ΔT	BD-rate	ΔT	
<i>EBU-UHD</i>	candle_smoke	5.88%	65.6%	6.32%	66.0%	1.13%	48.8%	0.62%	47.1%
	fountain_lady	2.61%	64.3%	2.67%	64.3%	1.02%	46.1%	0.05%	43.9%
	lupo_boa	4.12%	65.0%	4.39%	65.6%	0.97%	46.9%	0.26%	44.9%
	lupo_confetti	5.06%	65.3%	5.20%	65.3%	1.05%	48.9%	0.49%	46.6%
	park_dancers	1.70%	62.1%	1.85%	63.7%	0.62%	42.4%	-0.09%	41.1%
	pendulus_wide	3.66%	60.2%	4.64%	63.2%	1.28%	36.9%	0.94%	34.8%
	studio_dancer	6.82%	64.5%	7.19%	65.3%	1.72%	45.2%	0.87%	42.9%
	veggie_fruits	4.83%	61.4%	5.74%	63.2%	1.05%	40.7%	0.61%	38.5%
	waterfall_pan	1.91%	60.5%	2.21%	62.6%	0.45%	38.3%	-0.08%	35.5%
	wind_wool	6.21%	64.6%	6.91%	64.3%	1.43%	44.1%	0.89%	40.8%
	Moyenne	4.28%	63.2%	4.71%	64.4%	1.07%	43.0%	0.46%	40.8%

TABLEAU 5 – Inférence du partitionnement (images I) : résultats par séquence.

Séquences	Stratégie 4		variante		
	BD-rate	ΔT	BD-rate	ΔT	
<i>EBU-UHD</i>	candle_smoke	7.47%	68.1%	7.48%	69.7%
	fountain_lady	3.02%	65.6%	3.14%	68.2%
	lupo_boa	4.99%	67.4%	5.08%	70.3%
	lupo_confetti	5.80%	67.5%	5.83%	69.6%
	park_dancers	1.67%	66.4%	1.93%	68.8%
	pendulus_wide	3.75%	65.6%	4.57%	68.6%
	studio_dancer	8.05%	66.0%	8.68%	70.4%
	veggie_fruits	4.37%	65.5%	5.39%	69.3%
	waterfall_pan	1.43%	65.1%	2.06%	69.1%
	wind_wool	6.71%	66.5%	7.70%	68.5%
	Moyenne	4.73%	66.4%	5.19%	69.4%

TABLEAU 6 – Inférence du mode de prédiction (images I) : résultats par séquence.

D Codage rapide des images I basé sur la pré-analyse

D.1 Opérateurs de pré-analyse

Variance

La variance σ est calculée sur le bloc de luminance.

$$\sigma = \frac{1}{N} \sum_{i=0}^{N-1} (y[i] - \tilde{y})^2 \quad (1)$$

$y[i]$ est l'intensité à la position i .

\tilde{y} est la moyenne des intensités dans le bloc de N échantillons.

Entropie

Selon les travaux de [33], l'entropie est calculée sur le bloc de luminance après filtrage anti-bruit des échantillons sources. Les intensités sont d'abord quantifiées sur un pas de 8, ce qui ramène les valeurs possibles à 32 niveaux. L'entropie E pour le bloc étudié est exprimée comme suit :

$$E = \sum_{i=0}^{i=31} p_i \log_2 p_i \quad (2)$$

avec p_i , la probabilité d'une intensité i parmi 32 niveaux dans le bloc.

Magnitude moyenne des gradients

Les gradients horizontaux et verticaux g_x et g_y sont calculés avec les filtres de Sobel en figure 3.3. Nous considérons la moyenne des gradients ∇_i dans le bloc, pour représenter la complexité G du bloc étudié :

$$\nabla_i = \sqrt{g_x^2 + g_y^2} \quad (3)$$

Cohérence norme des tenseurs de structure

Le tenseur de structure t décrit les orientations et les magnitudes de l'image et est obtenu à partir du gradient ∇_i .

$$t = (\nabla_i \nabla_i^T) * G_\sigma \quad (4)$$

Il fournit une meilleure description des structures locales de l'image que le gradient et la régularisation par filtrage Gaussien 3×3 G_σ , permet d'améliorer la robustesse au bruit.

Nous définissons ensuite un terme de cohérence norme c [55], basé sur une différence des valeurs propres λ_1 et λ_2 de t , afin de mesurer le degré d'anisotropie. La moyenne des cohérences normes c nous donne la complexité C du bloc.

$$c = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2 \quad (5)$$

D.2 Seuils de décision

Th/Th_L	$d=0$	$d=1$	$d=2$	$d=3$	Th/Th_L	$d=0$	$d=1$	$d=2$	$d=3$	Th/Th_L	$d=0$	$d=1$	$d=2$	$d=3$
QP = 22	0	0	0	4	QP = 22	0	0	4	4	QP = 22	0	0	4	8
QP = 27	0	0	4	8	QP = 27	0	4	8	12	QP = 27	0	4	12	16
QP = 32	0	4	8	16	QP = 32	0	4	12	20	QP = 32	4	8	20	40
QP = 37	0	8	20	40	QP = 37	0	4	28	52	QP = 37	4	16	44	76

Th_H	$d=0$	$d=1$	$d=2$	$d=3$	Th_H	$d=0$	$d=1$	$d=2$	$d=3$	Th_H	$d=0$	$d=1$	$d=2$	$d=3$
QP = 22	56	200	255	255	QP = 22	8	176	255	255	QP = 22	4	100	176	196
QP = 27	96	196	224	255	QP = 27	52	196	220	255	QP = 27	12	128	144	255
QP = 32	108	208	224	255	QP = 32	96	196	220	255	QP = 32	32	144	192	255
QP = 37	116	216	224	255	QP = 37	100	188	220	255	QP = 37	52	180	204	255

(a) PL = 0.95					(b) PL = 0.9					(c) PL = 0.8				
Th/Th_L	$d=0$	$d=1$	$d=2$	$d=3$	Th/Th_L	$d=0$	$d=1$	$d=2$	$d=3$	Th/Th_L	$d=0$	$d=1$	$d=2$	$d=3$
QP = 22	0	0	8	12	QP = 22	0	4	8	20	QP = 22	1	5	12	32
QP = 27	0	8	16	28	QP = 27	4	12	28	52	QP = 27	4	61	73	117
QP = 32	4	12	36	60	QP = 32	4	28	52	96	QP = 32	15	73	103	138
QP = 37	4	28	60	76	QP = 37	8	48	80	132	QP = 37	10	76	115	156

Th_H	$d=0$	$d=1$	$d=2$	$d=3$	Th_H	$d=0$	$d=1$	$d=2$	$d=3$	Th_H	$d=0$	$d=1$	$d=2$	$d=3$
QP = 22	4	68	184	188	QP = 22	0	16	48	152	QP = 22	1	8	34	122
QP = 27	8	112	124	255	QP = 27	4	68	88	144	QP = 27	4	61	73	138
QP = 32	16	112	144	255	QP = 32	8	100	108	156	QP = 32	15	73	103	138
QP = 37	28	124	180	255	QP = 37	16	100	140	255	QP = 37	14	105	151	210

(d) PL = 0.7					(e) PL = 0.6					(f) PL = 0.55				
Th/Th_L	$d=0$	$d=1$	$d=2$	$d=3$	Th/Th_L	$d=0$	$d=1$	$d=2$	$d=3$	Th/Th_L	$d=0$	$d=1$	$d=2$	$d=3$
QP = 22	0	0	8	12	QP = 22	0	4	8	20	QP = 22	1	5	12	32
QP = 27	0	8	16	28	QP = 27	4	12	28	52	QP = 27	4	61	73	117
QP = 32	4	12	36	60	QP = 32	4	28	52	96	QP = 32	15	73	103	138
QP = 37	4	28	60	76	QP = 37	8	48	80	132	QP = 37	10	76	115	156

FIGURE 7 – Seuils de décision appris à partir du jeu de séquence *EBU-UHD* et utilisés par l’algorithme de partitionnement rapide (à 2 et 3 classes).

D.3 Résultats expérimentaux : algorithme à 3 classes

Séquences	PL = 0.9		PL = 0.6		PL = 0.5	
	BD-rate	ΔT	BD-rate	ΔT	BD-rate	ΔT
<i>EBU-UHD</i> candle_smoke	-0.05%	61.3%	2.62%	80.2%	6.42%	85.0%
fountain_lady	-0.15%	53.2%	3.23%	77.0%	5.94%	84.8%
lupo_boa	-0.03%	52.7%	2.28%	76.7%	5.87%	84.6%
lupo_confetti	0.03%	57.8%	3.67%	78.7%	7.70%	84.7%
park_dancers	0.01%	29.3%	0.87%	70.7%	2.15%	84.4%
pendulus_wide	0.02%	37.3%	2.82%	68.4%	5.76%	81.8%
studio_dancer	0.10%	59.2%	5.43%	78.6%	12.31%	84.7%
veggie_fruits	-0.03%	52.8%	2.58%	76.7%	7.28%	84.7%
waterfall_pan	-0.03%	34.1%	1.25%	69.7%	3.38%	84.2%
wind_wool	0.04%	60.9%	3.92%	79.2%	9.14%	85.0%
Moyenne	-0.01%	44.6%	2.87%	71.9%	6.59%	83.6%

TABEAU 7 – Algorithme de partitionnement rapide à 3 classes : résultats par séquence.

E Codage rapide des images B basé sur le pré-codage de sous-résolution

E.1 Résultats expérimentaux : inférence du partitionnement

Séquences		PRED_UP_INTRA		CU_UP		CU_UP_SPLIT		CU_UP_MERGE&SPLIT	
		BD-rate	ΔT	BD-rate	ΔT	BD-rate	ΔT	BD-rate	ΔT
<i>EBU-UHD</i>	candle_smoke	4.50%	53.3%	3.73%	52.9%	0.51%	32.0%	-0.06%	31.3%
	fountain_lady	4.51%	53.7%	3.95%	51.4%	1.11%	29.7%	0.43%	27.4%
	lupo_boa	8.93%	59.4%	8.27%	54.8%	1.34%	31.7%	0.83%	29.0%
	lupo_confetti	6.54%	62.9%	5.87%	55.7%	1.12%	32.8%	0.59%	29.4%
	park_dancers	3.29%	60.1%	2.99%	58.6%	0.40%	40.8%	0.14%	38.6%
	pendulus_wide	7.38%	58.6%	7.19%	57.8%	1.17%	38.6%	0.91%	34.9%
	studio_dancer	8.30%	61.5%	7.37%	58.2%	1.75%	38.2%	1.11%	36.0%
	veggie_fruits	5.74%	59.2%	5.16%	58.9%	0.81%	41.2%	0.45%	39.5%
	waterfall_pan	1.31%	62.6%	1.25%	60.7%	0.23%	42.7%	-0.07%	41.2%
	wind_wool	8.04%	62.8%	7.08%	57.8%	1.71%	36.8%	1.16%	34.5%
Moyenne		5.85%	59.5%	5.28%	56.8%	1.01%	36.6%	0.55%	34.2%

TABLEAU 8 – Inférence du partitionnement (images B) : résultats par séquence.

E.2 Résultats expérimentaux : inférence conditionnelle aux types de prédictions

Séquences		NONSPLIT_SKIP		NONSPLIT_SKIP_2		OPTIMAL_UP	
		BD-rate	ΔT	BD-rate	ΔT	BD-rate	ΔT
<i>EBU-UHD</i>	candle_smoke	0.96%	47.7%	1.92%	59.5%	1.98%	66.7%
	fountain_lady	1.20%	43.1%	1.73%	56.2%	2.21%	64.1%
	lupo_boa	1.71%	44.2%	3.73%	53.9%	4.48%	62.1%
	lupo_confetti	0.78%	43.1%	1.24%	49.8%	2.03%	64.5%
	park_dancers	0.35%	53.3%	0.71%	62.5%	1.25%	71.1%
	pendulus_wide	1.97%	50.9%	3.56%	61.0%	4.37%	66.8%
	studio_dancer	1.27%	49.7%	2.07%	58.4%	3.68%	67.4%
	veggie_fruits	1.27%	54.3%	2.57%	63.5%	3.48%	71.9%
	waterfall_pan	0.53%	54.6%	0.88%	64.5%	1.25%	72.3%
	wind_wool	0.96%	47.8%	1.52%	56.9%	2.99%	64.7%
Moyenne		1.10%	49.0%	1.99%	58.7%	2.77%	67.3%

TABLEAU 9 – Inférence conditionnelle aux types de prédictions (images B) : résultats par séquence.

E.3 Résultats expérimentaux : configuration de codage optimisé

Séquences		CU_UP		CU_UP_SPLIT		OPTIMAL_UP	
		BD-rate	ΔT	BD-rate	ΔT	BD-rate	ΔT
<i>EBU-UHD</i>	candle_smoke	7.32%	15.7%	1.64%	2.2%	3.59%	7.6%
	fountain_lady	5.31%	29.9%	1.46%	13.1%	2.42%	20.3%
	lupo_boa	12.41%	43.5%	2.95%	24.0%	5.43%	33.3%
	lupo_confetti	7.00%	48.6%	1.64%	30.3%	1.95%	39.4%
	park_dancers	5.40%	26.8%	1.36%	10.5%	2.75%	19.2%
	pendulus_wide	16.53%	31.2%	3.16%	10.9%	7.45%	17.8%
	studio_dancer	9.50%	38.0%	2.87%	18.9%	3.95%	26.2%
	veggie_fruits	13.23%	19.7%	2.76%	4.4%	6.17%	10.1%
	waterfall_pan	2.22%	33.4%	0.92%	15.9%	1.47%	24.6%
	wind_wool	11.69%	42.6%	3.12%	22.9%	3.89%	32.4%
	Moyenne	9.06%	32.8%	2.19%	15.3%	3.91%	23.0%

TABLEAU 10 – Inférence conditionnelle aux types de prédictions (images B) : résultats par séquence.

Liste des figures

1.1	Déroulement de la standardisation HEVC.	7
1.2	Structure d'une vidéo.	8
1.3	Schéma général d'un système de transmission vidéo.	9
1.4	Sous-échantillonnage YUV 4 :2 :0.	10
1.5	Structure de codage hybride HEVC.	11
1.6	Les 3 principales structures de codage utilisées en codage vidéo.	12
1.7	Représentation des <i>slices</i>	13
1.8	Partitionnement d'un CTU en CU, PU et TU.	14
1.9	Modes de prédictions Intra définis par HEVC pour les PB de luma.	15
1.10	Prédiction Inter compensée en mouvement.	16
1.11	Représentation du vecteur de mouvement différentiel (MVD).	17
1.12	Candidats spatiaux et temporels pour la construction de la liste AMVP.	17
1.13	Candidats spatiaux et temporels pour la construction de la liste MERGE.	19
1.14	Outils pour le parallélisme.	22
1.15	Pré-sélection rapide du meilleur mode Intra employée dans le modèle de référence HEVC.	24
2.1	Extraits de quelques séquences EBU.	27
2.2	SI-TI des séquences <i>EBU-UHD</i>	27
2.3	Métrique de Bjøntegaard déterminée à partir de 2 courbes débit/PSNR.	29
2.4	Partitionnement CU : l'approche exhaustive du modèle de référence.	32
3.1	Distribution statistique des classes <i>SPLIT</i> et <i>NON-SPLIT</i> en fonction du coût de codage CU.	37
3.2	Approche <i>merge-and-split</i> de Chen et Sun [34].	38
3.3	Filtres 3x3 de Sobel dans les directions verticales et horizontales.	39
4.1	Complexités du codage UHD et du codage de la version réduite au format HD.	42
4.2	Architecture de codage basé pré-codage de sous-résolution.	43
4.3	Modes de prédictions AI disponibles pour le pré-codage HD et pour le codage UHD.	44
4.4	Partitionnement HD (QP = 27) produit après filtrage bi-linéaire.	44
4.5	Partitionnement HD (QP = 27) produit après filtrage SHVC.	44
4.6	Partitionnements PU possibles de la région UHD co-localisé au PU HD.	45
4.7	Histogramme du partitionnement PU UHD sachant la taille de PU HD co-localisée (QP = 32).	45
4.8	Stratégie d'inférence <i>CU_UP</i>	46
4.9	Stratégie d'inférence <i>PU_UP</i>	46
4.10	Stratégie d'inférence <i>CU_UP_SPLIT</i>	47
4.11	Régions UHD fusionnées (MERGE) ou re-décomposées (SPLIT) dans le codeur de référence par rapport au partitionnement HD (extrait de la séquence <i>park_dancers</i> à QP = 32).	47
4.12	Performance des stratégies d'inférence du partitionnement par rapport au codage UHD de référence.	48

4.13	Variation du QP entre la passe de pré-codage HD et la passe de codage UHD.	49
4.14	Distribution des modes de luminance dans le codage UHD (<i>PU_UP</i>).	51
4.15	Distribution des modes de chrominance dans le codage UHD (<i>PU_UP</i>).	51
4.16	Schémas de pré-sélection rapide du mode de prédiction : stratégie 4 (à gauche) et sa variante (à droite).	53
4.17	Performance de la pré-sélection rapide du mode de prédiction par rapport au codage UHD de référence.	54
5.1	Architecture de notre solution de codage accéléré par la pré-analyse.	55
5.2	Probabilités conditionnelles de SPLIT sachant σ_b pour les PU 32x32 ($d = 1$) à QP = 32.	57
5.3	Procédure de partitionnement rapide.	58
5.4	Performance de notre algorithme de partitionnement rapide, pour plusieurs valeurs de PL, par rapport au codage UHD de référence (jeu de séquences <i>EBU-UHD</i>).	59
5.5	Performance de notre algorithme de partitionnement rapide, appliqué à d'autres jeux de séquences (validation).	59
5.6	Procédure de partitionnement rapide à 3 classes (variantes 1 et 2).	60
5.7	Performance de notre algorithme de partitionnement rapide à 3 classes, en fonction de PL, par rapport au codage UHD de référence (jeu de séquences <i>EBU-UHD</i>).	61
5.8	Performance de notre algorithme de partitionnement rapide, appliqué à d'autres jeux de séquences (validation).	62
5.9	Partitionnements UHD d'une région de la séquence <i>wind_wool</i> à QP = 27.	64
5.10	Sensibilité des seuils sur la variance (jeu de séquences <i>EBU-UHD</i>).	64
5.11	Performance de l'algorithme de partitionnement rapide à 3 classes associé à des simplifications (jeu de séquences <i>EBU-UHD</i>).	65
5.12	Performance de l'algorithme de partitionnement rapide à 3 classes combiné à l'algorithme de pré-codage de sous-résolution (jeu de séquences <i>EBU-UHD</i>).	66
5.13	Apprentissage <i>online</i> des seuils sur la variance.	67
5.14	Histogrammes conditionnels à σ_b pour les PU de tailles 32x32 (1ère image de la séquence <i>candle_smoke</i> codée à QP = 32).	67
5.15	Performance de l'algorithme 3 classes avec apprentissage <i>online</i> , en fonction de PL, par rapport au codage UHD de référence.	68
6.1	Simplifications optionnelles dans le logiciel de référence.	73
6.2	Voisinage spatio-temporel typiquement utilisé pour la réduction de complexité.	75
7.1	Références utilisées dans le GOP <i>Random Access</i> de référence.	80
7.2	Références utilisées dans le GOP <i>Random Access</i> optimisé pour la diffusion.	80
8.1	Partitions et types de prédictions pour le codage d'un CU.	82
8.2	Histogramme conditionnel du partitionnement UHD sachant la partition HD co-localisée (QP = 32), dans les images B.	83
8.3	Distribution des modes dans les partitions HD ((a) à (d)) et dans les partitionnements UHD co-localisé à la partition HD ((e) à (h)), dans les images B.	83
8.4	Performance des stratégies d'inférence du partitionnement UHD par rapport au codage UHD de référence.	85
8.5	Performance des stratégies d'inférence conditionnelle UHD par rapport au codage UHD de référence.	88
8.6	Performance des stratégies d'inférence du partitionnement UHD par rapport à la référence de codage optimisée.	89
7	Seuils de décision appris à partir du jeu de séquence <i>EBU-UHD</i> et utilisés par l'algorithme de partitionnement rapide (à 2 et 3 classes).	105

Liste des tableaux

1.1	Performance de la pré-sélection rapide des modes Intra HEVC.	25
2.1	Performance de codage apportée par les tailles CU 32x32 et 64x64 dans HEVC.	30
2.2	Performance de codage apportée par les tailles TU 16x16 et 32x32 dans HEVC.	30
2.3	Performance de codage apportée par les partitions asymétriques (AMP) dans HEVC.	31
2.4	Réduction de complexité maximum.	32
2.5	Réduction de complexité effective en évaluant le partitionnement CU optimal unique- ment.	33
2.6	Réduction de complexité effective en évaluant le partitionnement PU optimal unique- ment.	33
2.7	Réduction de complexité effective en évaluant le partitionnement PU optimal avec les modes Intra associés uniquement.	33
4.1	Performance du pré-codage des séquences UHD réduites en HD avec le filtre SHVC par rapport au filtre bi-linéaire.	44
4.2	Corrélation entre les modes HD et UHD.	50
4.3	Stratégies de pré-sélection du mode de prédiction.	52
4.4	Performance des stratégies de pré-sélection du mode de prédiction UHD.	52
5.1	Comparaison des opérateurs de pré-analyse selon leur information mutuelle par rap- port aux classes SPLIT et NON-SPLIT.	56
5.2	Pourcentages de PU éliminés du partitionnement avec l’algorithme à 2 classes.	63
5.3	Pourcentages de PU éliminés du partitionnement avec l’algorithme à 3 classes (va- riante 1).	63
7.1	Performance des simplifications du logiciel de référence HEVC.	79
7.2	Performance de notre configuration de codage <i>Random Access</i> optimisée.	80
7.3	Performance de la configuration de codage RA optimisée et du GOP simplifié.	80
8.1	Table de correspondance associée à la stratégie d’inférence <i>OPTIMAL_UP</i> , pour les images B.	86
2	Jeu de séquences <i>EBU-UHD</i>	101
3	Jeu de séquences <i>ClasseB-HD</i>	102
4	Performance de codage apportée par les tailles CU 32x32 et 64x64 dans HEVC : résultats par séquence.	102
5	Inférence du partitionnement (images I) : résultats par séquence.	103
6	Inférence du mode de prédiction (images I) : résultats par séquence.	103
7	Algorithme de partitionnement rapide à 3 classes : résultats par séquence.	105
8	Inférence du partitionnement (images B) : résultats par séquence.	106
9	Inférence conditionnelle aux types de prédictions (images B) : résultats par séquence.	106
10	Inférence conditionnelle aux types de prédictions (images B) : résultats par séquence.	107

Publications

Conférences Internationales avec comité de lecture et publication des actes

N. Dhollande, O. Le Meur, and C. Guillemot. HEVC Intra coding of Ultra HD video with reduced complexity. In *2014 IEEE International Conference on Image Processing (ICIP)*, Paris, October 2014.

N. Dhollande, X. Ducloux, O. Le Meur, and C. Guillemot. Fast block partitioning method in HEVC Intra coding for UHD video. In *2014 IEEE International Conference on Consumer Electronics (ICCE)*, Berlin, 2015.

Rapport d'étude

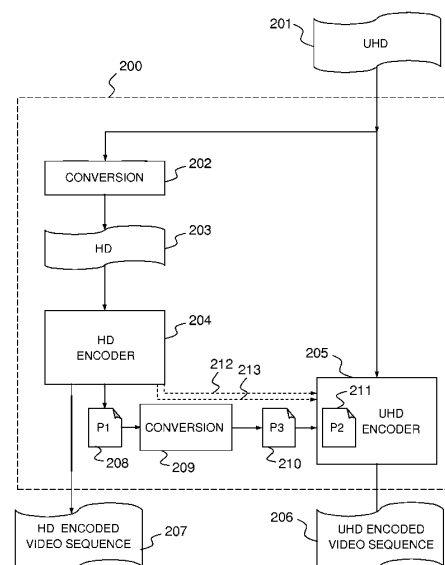
Livable ANR 2015 : *HEVC encoder optimization*

Livable ANR 2016 : *Optimisation du Codage HEVC Inter pour Vidéos UHD*

Brevet d'invention

WO2015121112 (A1) - AN APPARATUS FOR ENCODING AN ULTRA-HIGH DEFINITION VIDEO SEQUENCE. Dépositant : France Brevets. Auteurs : N. Dhollande, X. Ducloux.

An apparatus (200) for the encoding of an original video sequence (201), comprising : a conversion module (202) which converts the original video sequence (201) into a dyadic sub-sampled video sequence (203); a first encoding module (204) providing a first partition (208) of the images of the dyadic sub-sampled video sequence (203), said first partition (208) comprising at least one coding block; a second encoding module (205) for encoding the original video sequence, said module determining a second partition (211) of the images of the original video sequence (201), the second partition (211) being derived from a third partition (210) corresponding to the first partition (208) which is adjusted (209) to the resolution of the original video sequence (201).



Résumé

La compression vidéo HEVC standardisée en 2013 offre des gains de compression dépassant les 50% par rapport au standard de compression précédent MPEG4-AVC/H.264. Ces gains de compression se paient par une augmentation très importante de la complexité de codage. Si on ajoute à cela l'augmentation de complexité générée par l'accroissement de résolution et de fréquence image du signal vidéo d'entrée pour passer de la Haute Définition (HD) à l'Ultra Haute Définition (UHD), on comprend vite l'intérêt des techniques de réduction de complexité pour le développement de codeurs économiquement viables. En premier lieu, un effort particulier a été réalisé pour réduire la complexité des images Intra. Nous proposons une méthode d'inférence des modes de codage à partir d'un pré-codage d'une version réduite en HD de la vidéo UHD. Ensuite, nous proposons une méthode de partitionnement rapide basée sur la pré-analyse du contenu. La première méthode offre une réduction de complexité d'un facteur 3 et la deuxième, d'un facteur 6, contre une perte de compression proche de 5%. En second lieu, nous avons traité le codage des images Inter. En mettant en œuvre une solution d'inférence des modes de codage UHD à partir d'un pré-codage au format HD, la complexité de codage est réduite d'un facteur 3 en considérant les 2 flux produits et d'un facteur 9.2 sur le seul flux UHD, pour une perte en compression proche de 3%. Appliqué à une configuration de codage proche d'un système réellement déployé, l'apport de notre algorithme reste intéressant puisqu'il réduit la complexité de codage du flux UHD d'un facteur proche de 2 pour une perte de compression limitée à 4%. Les stratégies de réduction de complexité mises en œuvre au cours de cette thèse pour le codage Intra et Inter offrent des perspectives intéressantes pour le développement de codeurs HEVC UHD plus économes en ressources de calculs. Elles sont particulièrement adaptées au domaine de la WebTV/OTT qui prend une part croissante dans la diffusion de la vidéo et pour lequel le signal vidéo est codé à des résolutions multiples pour adresser des réseaux et des terminaux de capacités variées.

Abstract

The High Efficiency Video Coding (HEVC) standard was released in 2013 which reduced network bandwidth by a factor of 2 compared to the prior standard H.264/AVC. These gains are achieved by a very significant increase in the encoding complexity. Especially with the industrial demand to shift in format from High Definition (HD) to Ultra High Definition (UHD), one can understand the relevance of complexity reduction techniques to develop cost-effective encoders. In our first contribution, we attempted new strategies to reduce the encoding complexity of Intra-pictures. We proposed a method with inference rules on the coding modes from the modes obtained with pre-encoding of the UHD video down-sampled in HD. We, then, proposed a fast partitioning method based on a pre-analysis of the content. The first method reduced the complexity by a factor of 3x and the second one, by a factor of 6, with a loss of compression efficiency of 5%. As a second contribution, we addressed the Inter-pictures. By implementing inference rules in the UHD encoder, from a HD pre-encoding pass, the encoding complexity is reduced by a factor of 3x when both HD and UHD encodings are considered, and by 9.2x on just the UHD encoding, with a loss of compression efficiency of 3%. Combined with an encoding configuration imitating a real system, our approach reduces the complexity by a factor of close to 2x with 4% of loss. These strategies built during this thesis offer encouraging prospects for implementation of low complexity HEVC UHD encoders. They are fully adapted to the WebTV/OTT segment that is playing a growing part in the video delivery, in which the video signal is encoded with different resolution to reach heterogeneous devices and network capacities.