



HAL
open science

Codeur vidéo scalable haute-fidélité SHVC modulable et parallèle

Ronan Parois

► **To cite this version:**

Ronan Parois. Codeur vidéo scalable haute-fidélité SHVC modulable et parallèle. Traitement du signal et de l'image [eess.SP]. INSA de Rennes, 2018. Français. NNT : 2018ISAR0016 . tel-02048319

HAL Id: tel-02048319

<https://theses.hal.science/tel-02048319>

Submitted on 25 Feb 2019

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

UNIVERSITE
BRETAGNE
LOIRE

THESE INSA Rennes
sous le sceau de l'Université Bretagne Loire
pour obtenir le titre de
DOCTEUR DE L'INSA RENNES
Spécialité : Signal, Vision, Image

présentée par

Ronan PAROIS

ECOLE DOCTORALE : *MathStic*
LABORATOIRE : *IETR*

Codeur vidéo scalable
haute-fidélité SHVC
modulable et parallèle

Thèse soutenue le 27.02.2018
devant le jury composé de :

Mme Anissa MOKRAOUI

Professeure à l'Université Paris 13 / Présidente

M. Marco MATTAVELLI

Professeur à l'EPFL, Lausanne (Suisse) / Rapporteur

M. Nouri MASMOUDI

Professeur à l'ENIS, Sfax (Tunisie) / Rapporteur

M. Mohamed AKIL

Professeur à l'ESIEE Paris / Examineur

M. Mickaël RAULET

Responsable Recherche chez ATEME Rennes / Co-encadrants

M. Wassim HAMIDOUCHE

Maître de Conférences à l'INSA de Rennes / Co-encadrants

M. Olivier DEFORGES

Professeur à l'INSA de Rennes / Directeur de thèse



Codeur vidéo scalable
haute-fidélité SHVC
modulable et parallèle

Ronan PAROIS



Table des matières

Glossaire	5
Remerciements	9
1 Introduction	11
1.1 Contexte	11
1.2 Objectifs et plan de la thèse	12
2 Le codage vidéo, état de l'art	13
2.1 Représentation d'un signal vidéo	13
2.1.1 Résolution et rapport de format d'images	13
2.1.2 Fréquence d'image	13
2.1.3 Représentation des couleurs et espaces colorimétriques	14
2.1.4 Représentation numérique, profondeur de données	14
2.1.5 Formats vidéo d'aujourd'hui et de demain	17
2.2 Métriques d'évaluation objectives	20
2.2.1 PSNR	20
2.2.2 SSIM	21
2.2.3 BD-BR et BD-PSNR	21
2.2.4 Métriques actuelles	21
2.3 Les outils de la compression vidéo	22
2.3.1 Répartition en blocs de pixel	22
2.3.2 Prédiction	22
2.3.3 Transformée	23
2.3.4 Quantification	23
2.3.5 Codage entropique	24
2.3.6 Filtrages	24
2.3.7 Syntaxes haut-niveau	24
2.4 Les standards de la compression vidéo	24
2.4.1 Évolution de la consommation vidéo et tendances	24
2.4.2 Nécessité d'un codage	24
2.4.3 Les standards de compression vidéo existants et à venir	25
2.5 Le standard HEVC	28
2.5.1 Répartition en blocs de pixels	28
2.5.2 Prédiction	29
2.5.3 Transformée et Quantification	31
2.5.4 Codage entropique	31
2.5.5 Filtrages Anti-blocs et filtrages SAO	32
2.5.6 Syntaxes haut-niveau	32
2.5.7 Outils de parallélisme	32
2.5.8 Structure de GOP et configuration de codage	33
2.5.9 Profile, tiers et level	34
2.6 Codage Scalable et SHVC	35

2.6.1	Codage scalable	35
2.6.2	Le standard SHVC	36
2.7	Régulation de débit et améliorations appliquées au standard SHVC	39
2.7.1	Principe de la régulation de débit	40
2.7.2	Travaux existants pour SHVC	41
2.8	Codeurs et décodeurs exploitant SHVC	42
2.8.1	SHM	42
2.8.2	Kvazaar	42
2.8.3	x265	42
2.8.4	openHEVC	43
2.8.5	Outils d'analyse existants	43
2.9	Cas d'usage du codage scalable	43
2.9.1	ATSC 3.0	44
2.9.2	3GPP SA4	45
3	La scalabilité spatiale appliquée à un codeur professionnel	49
3.1	Présentation de l'encodeur Atime	49
3.1.1	Architecture et parallélisme	50
3.1.2	Traitement multi-tâches des opérations	53
3.1.3	Décision de codage	54
3.1.4	Régulation de débit	56
3.2	Architecture de l'encodeur scalable SHVC	57
3.2.1	Sous-échantillonnage SHVC	59
3.2.2	Sur-échantillonnage SHVC	62
3.2.3	Prédiction inter-couches	64
3.3	Conduites des tests et outils de validation	64
3.4	Application de la prédiction inter-couches à différents types de slice	67
3.4.1	Analyse SHM des gains inter-couches en fonction du type de slice.	67
3.4.2	Application aux références I-slices	68
3.4.3	Application aux références P-slices	70
3.4.4	Application aux B-slices	74
4	Optimisation et parallélisation pour un codage temps-réel	79
4.1	Conduite des tests et outils de validation	80
4.2	Optimisations des vitesses de traitement appliquées aux filtrages	80
4.2.1	Algorithme des filtres d'interpolation	81
4.2.2	Parallélisme sur les données	82
4.2.3	Instructions SIMD	82
4.2.4	Résultats	85
4.3	Paramétrage du filtre de sous-échantillonnage avec 8 coefficients	86
4.3.1	Les différents paramètres de sous-échantillonnage	86
4.3.2	Expérimentations avec différents paramétrages du sous-échantillonnage	86
4.4	Amélioration de la vitesse de traitement en sélectionnant les outils dans la décision de codage	92
4.4.1	Décision de codage en configuration FILE	93
4.4.2	Décision de codage en configuration LIVE	94
5	Généricité de la méthode employée	103
5.1	Scalabilité spatiale sur plus de 2 couches	104
5.1.1	Adaptation de la signalisation pour un codage multi-couches	104
5.1.2	Adaptation de l'architecture pour un codage multi-couches	106
5.1.3	Application sur 3 couches spatiales	111
5.2	Scalabilité appliquée à d'autres niveaux de qualité	116
5.2.1	Application à la scalabilité du SNR	117
5.2.2	Application au codage multi-vues	122
5.2.3	Application à la scalabilité de la profondeur des données	122

5.2.4	Application à la scalabilité des codecs	124
5.2.5	Application à la scalabilité des espaces colorimétriques	125
6	Application dans un contexte de diffusion temps-réel	129
6.1	Caractéristiques des différents modules	130
6.1.1	Le module de management	130
6.1.2	Le module d'acquisition	131
6.1.3	Le module de codage	133
6.1.4	Le module de transmission	134
6.2	Agencement des modules pour un codage sur plateformes en temps-réel	136
6.3	Démonstrations du codeur SHVC	141
7	Conclusion	143
7.1	Travaux réalisés	143
7.2	Perspectives	144
	ANNEXES	151

Glossaire

- 3GP-DASH** Streaming 3GPP Dynamique et Adaptatif sur flux HTTP ("3GPP Dynamic Adaptive Streaming over HTTP"). 41, 43, 44, 133
- 3GPP** "3rd Generation Partnership Project". 41, 43
- AI** all-intra (ou intra-only). 31, 32, 53, 54, 65–69, 72, 74, 77, 86, 88, 89, 91–93, 95–99, 111, 113, 114, 117, 119
- AMVP** Prédiction des Vecteurs de Mouvement Avancée ("Advanced Motion Vector Prediction"). 28, 29, 62
- AOM** "Alliance for Open Media". 26
- API** "Application Programming Interface". 131
- ATSC** "Advanced Television Systems Committee". 10, 39, 41, 42, 128, 133, 139, 140, 142
- BT.2020** ITU-R BT.2020. 12, 15, 16, 123, 124
- BT.709** ITU-R BT.709. 12, 123, 124
- CABAC** Codeur Arithmétique à Contexte Adaptatif Binaire ("Context-Adaptive Binary Arithmetic Coding"). 29, 31
- CB** "Coding Block". 26, 27
- CCITT** Comité Consultatif International Télégraphique et Téléphonique. 23
- CGS** "Color Gamut Scalability". 9, 123, 124
- CIE** Comité Internationale de l'Éclairage. 12
- CTB** "Coding Tree Block". 26
- CTC** "Common Test Condition". 32, 41, 62, 63, 65, 66, 78, 95–97, 109, 117
- CTU** "Coding Tree Unit". 26, 27, 30, 31, 38, 40, 41
- CU** "Coding Units". 26, 27, 39, 45, 52, 53, 62, 69, 92, 93, 96, 132
- DCT** Transformée en Cosinus Discrète ("Discrete Cosine Transform"). 21, 24, 25, 29, 33
- DST** Transformée en Sinus Discrète ("Discrete Sine Transform"). 21, 25, 29
- DVB** "Digital Video Broadcasting". 10, 15, 133
- DWT** Transformée en Ondelettes Discrète ("Discrete Wavelet Transform"). 21
- ES** "Elementary Stream". 23, 132, 133
- filtres FIR** filtres à Réponse Impulsionnelle Finie. 35, 79, 80, 84
- GOP** Groupe d'images ("Group Of Pictures"). 31, 37–39, 47, 49, 62, 65–68
- H.262/MPEG-2** . 9, 24, 25, 33, 47, 123, 133
- H.264/MPEG-4 AVC** H.264/MPEG-4 "Advanced Video Coding". 9, 10, 16, 24–26, 29, 30, 33, 34, 40, 47, 123, 133

- HD** Haute Définition. 9–11, 18, 25, 92, 96, 125, 142
- HDR** "High Dynamic Range". 9, 15, 16, 18, 125, 140
- HEVC** "High Efficiency Video Coding". 9–11, 15, 16, 25–35, 38–41, 43, 44, 47, 49, 51, 53–55, 60–63, 65, 66, 68, 69, 72–76, 78, 90–92, 94, 95, 99, 104, 110, 117, 119–121, 123, 127, 131, 133, 140–142
- HFR** "High Frame Rate". 9, 12, 15, 16
- HLG** "Hybrid Log Gamma". 16
- HM** "HEVC Model". 40, 53, 69, 73, 75, 94
- IEC** Commission Electrotechnique Internationale. 9, 24
- IPS** Images par seconde. 11, 12, 16, 17, 40, 50, 63, 78, 83, 92, 95–97, 99, 101, 113, 119, 130, 131, 134, 138
- ISO** Organisation internationale de standardisation. 9, 24
- ITU** Union International des Télécommunications. 9, 23, 24, 43, 123, 129
- JCT-VC** "Joint Collaborative Team on Video Coding". 9, 25, 26, 71
- JVT** "Joint Video Team". 24
- LD-P** low-delay P. 32, 40, 53, 54, 65, 68–74, 77, 91–93, 95–99, 111, 113, 114, 117–119, 138
- MBMS** Service Multimedia de diffusion large et multi-bande ("Multimedia Broadcast/Multicast Service"). 41, 43, 90
- MMVC** Vidéo Conférence Multimédia Multi-utilisateurs ("Multiparty Multimedia Video Conferencing"). 41, 43
- MPEG** "Moving Picture Experts Group". 9, 24, 25, 123
- MPEG-2 TS** MPEG-2 "Transport Stream". 24, 133, 134
- MSE** Erreur Quadratique Moyenne ("Mean Square Error"). 18
- NALU** "Network Abstraction Layer Unit". 25, 30, 40, 123, 132, 139
- PB** "Prediction Block". 27
- PID** Paquet d'Identification ("Packet Identification"). 133
- PLP** Canal de la Couche Physique ("Physical Layer Pipe"). 43
- POC** "Picture Order Count". 48, 49, 62
- PPS** Set de Paramètre pour l'image ("Picture Parameter Set"). 30, 31, 34, 36, 37, 102, 117, 123
- PQ** "Perceptual Quantizer". 16
- projet WebM** projet WebM. 25, 26
- PS** MPEG-2 "Program Stream". 24
- PSNR** "Peak Signal to Noise Ratio". 18, 19, 63, 86, 93
- PU** "Prediction Unit". 26, 27, 29, 62, 66, 67
- QP** Paramètre de Quantification ("Quantization Parameter"). 29, 38, 39, 53–55, 62, 63, 69, 78, 117
- RA** random-access. 31, 32, 40, 43, 49, 50, 53–55, 65, 72–77, 91–93, 95–99, 101, 111, 113, 114, 117–119, 125, 138
- SAO** Décalage Adaptatif des Échantillons ("Sample Adaptive Offset"). 30, 31
- SDI** Interface numérique série ("Serial Digital Interface"). 129, 130, 139, 140
- SDR** "Standard Dynamic Range". 15, 18, 125, 140
- SEI** Information d'Amélioration Supplémentaire ("Supplemental Enhancement Information"). 16, 30, 140

- SHM** "Scalable HEVC Model". 40, 47, 59, 61, 63, 65, 67–74, 84, 85, 90–92, 99, 101, 127, 141
- SHVC** "Scalable High efficiency Video Coding". 9–11, 34–37, 39–45, 47, 53–56, 59, 61–63, 65, 71, 73, 75, 77, 83, 84, 90, 91, 95, 96, 100–103, 112, 114, 117–120, 123–125, 127–129, 133, 139–141
- SHVC-AE** "SHVC ATEME Encoder". 55–64, 66–78, 80, 83–102, 105–110, 112–125, 127–131, 133, 135–142, 153
- SIMD** "Single Instruction Multiple Data". 81, 83, 99
- SL-HDR1** . 16, 140
- SLE** Single Layer Encoder. 47–62, 67–75, 77, 86–93, 96, 97, 100, 104, 108, 110, 113, 116–119, 123, 125, 131, 141, 149
- SMPTE** "Society of Motion Picture and Television Engineers". 16, 129
- SNR** "Signal to Noise Ratio". 9, 24, 25, 33, 34, 39, 45, 47, 114–117, 119, 120, 123, 125, 141
- SPS** Set de Paramètre pour la Séquence ("Sequence Parameter Set"). 30, 34, 102, 117, 123
- SSE2** "Streaming SMID Extension 2". 83
- SSIM** "Structural SIMilarity". 18, 19
- SVC** "Scalable Video Coding". 10, 25, 33
- SVH** Système Visuel Humain. 12–14
- TB** "Transform Block". 27, 29
- TU** "Transform Unit". 27, 92
- UHD** Ultra Haute Définition. 9–11, 15, 18, 83, 92, 96, 125, 140, 142
- VBV** Buffer Virtuel de Vérification ("Virtual Buffer Verifier"). 38, 55
- VCEG** "Video Coding Experts Group". 9, 23–25
- VPS** Set de Paramètre pour la Vidéo ("Video Parameter Set"). 30, 34, 102, 103, 117, 120, 121, 123
- WHT** Transformée de Walsh-Hadamard ("Walsh-Hadamard Transform"). 25, 26
- WPP** "Wavefront Parallel Processing". 30, 31, 40, 41, 52

Remerciements

Je souhaite exprimer ma reconnaissance à Mickaël RAULET, responsable de l'équipe Recherche et Innovation chez ATEME Rennes, pour son encadrement et ses divers conseils au cours de ces trois années de thèse.

Je remercie également Olivier DEFORGES et Wassim HAMIDOUCHE pour leur encadrement. Leurs esprits critiques, les diverses méthodes de travail et leur accompagnement m'ont permis de m'améliorer tout au long de ces trois années.

Je remercie Marco MATTAVELLI et Nouri MASMOUDI pour avoir assumé la charge de rapporteur. Je remercie également Anissa MOKRAOUI et Mohamed AKIL pour avoir examiné mes travaux de thèse.

Je remercie à présent les diverses personnes qui ont pu croiser ma route au cours de cette thèse qu'ils viennent de la société ATEME ou bien de l'INSA de Rennes. Parmi mes collègues de chez ATEME, je remercie particulièrement Thomas GUIONNET et Eric TOULLEC pour leurs réponses à mes nombreuses questions mais également pour m'avoir donné l'occasion d'améliorer mes pâtisseries. Je remercie également Jérémy VIGOUR, Loic HABBRIAL, Pauline CASTETS et Guillaume PHILIPPE pour leur bonne humeur et pour chercher à rendre le quotidien chez ATEME plus chaleureux. Du côté de l'INSA de Rennes, je remercie Alexandre MERCAT, Karol DESNOS, Julien HEULOT, Jean-François NEZAN, Maxime PELCAT et Julien GOSSEAUME pour m'avoir donné la fibre de la recherche. Je remercie également Aurore GOUIN et Caroline COUILLARD pour leur efficacité et leur accompagnement dans toutes les démarches administratives respectivement côté INSA et ATEME.

Je remercie mes proches. Les caïls, Antoine et Camille, Gaëlle et Camille pour ces quelques virées dans Paris et les superbes souvenirs que je garde après ces trois années. Mes anciens colocataires expatriés, Fabien et Bertrand, pour m'avoir permis de m'aérer l'esprit. Côté Rennais, je remercie Yannick, Sophie, Antoine et Maxime pour avoir essayé tant bien que mal de me sortir.

Enfin, je remercie mes parents, Véronique et Jean-Yves, mes deux frères, Maël et Joris, et ma soeur, Lénaïc, car même s'ils n'ont pas toujours compris ce que je racontais, ils ont toujours été là pour me soutenir.

Chapitre 1

Introduction

1.1 Contexte

Les technologies de compression vidéo ont significativement évolué ces dernières années, notamment après l'entrée dans l'ère du numérique. Ces technologies sont développées par des groupes d'experts comme les "Video Coding Experts Group" (VCEG) et "Moving Picture Experts Group" (MPEG) issus des groupes de standardisation tels que l'Union International des Télécommunications (ITU), la Commission Electrotechnique Internationale (IEC) et l'Organisation internationale de standardisation (ISO). Parmi les standards de compression vidéo proposés ces dernières années, on trouve notamment, le standard (H.262/MPEG-2) en 1994, le standard H.264/MPEG-4 "Advanced Video Coding" (H.264/MPEG-4 AVC) en 2003 et le standard "High Efficiency Video Coding" (HEVC) [1] en fin d'année 2013.

L'amélioration des standards de compression a permis une amélioration globale de la qualité vidéo avec des technologies nécessitant davantage de débits comme les formats Haute Définition (HD) et Ultra Haute Définition (UHD), la technologie "High Dynamic Range" (HDR) ainsi que l'amélioration des fréquences avec le "High Frame Rate" (HFR). De plus, avec l'évolution matérielle des processeurs mono, multi et many-cœurs, davantage de plateformes sont susceptibles de demander un accès à un contenu vidéo (télévision, ordinateur, smartphone...). Les méthodes de diffusion ont donc également évolué avec des transmissions par câbles, par réseaux terrestres, par réseaux mobiles, par satellites, par Internet, par stockage local ou encore par une combinaison de ces différentes méthodes. Les améliorations technologiques ont donc ouvert la voie à de nouvelles tendances de consommation vidéo qui elles mêmes imposent de nouvelles contraintes aux nouvelles technologies.

Le standard HEVC a été développé par le groupe nommé "Joint Collaborative Team on Video Coding" (JCT-VC) avec pour objectif de réduire le débit de codage par rapport au H.264/MPEG-4 AVC de 50% pour une même qualité visuelle. De la même manière que pour les précédents standards, plusieurs extensions ont été ajoutées au standard HEVC comme par exemple les codages scalable, multi-vues et 3D. La scalabilité consiste à pouvoir répéter plusieurs fois une opération sur différentes échelles d'un même sujet. Dans le cas du codage vidéo, cela consiste à pouvoir coder ou décoder une même séquence avec différents niveaux de qualité. Chaque niveau de qualité correspond à une couche scalable et est défini selon les paramètres de codage d'une vidéo, comme par exemple :

- la résolution de l'image (largeur et hauteur) (i.e. scalabilité spatiale),
- la fréquence des images (i.e. scalabilité fréquentielle),
- la distorsion (i.e. scalabilité "Signal to Noise Ratio" (SNR)),
- la profondeur des pixels,
- l'espace de couleur (i.e. scalabilité sur les espaces colorimétriques ou "Color Gamut Scalability" (CGS)),
- le standard de compression employé (i.e. scalabilité sur les codecs).

L'extension scalable du standard HEVC porte le nom de "Scalable High efficiency Video Coding" (SHVC) et fut proposé en fin d'année 2014. Elle propose un codage scalable sous forme de couches permettant une scalabilité spatiale, fréquentielle (issue de la première version de HEVC), SNR, mais également sur la profondeur des pixels, le codec employé sur la couche de base et la CGS. L'extension SHVC permet également d'améliorer significativement le débit de codage par rapport à un codage HEVC équivalent sur l'ensemble des couches (i.e. simulcast). Cette amélioration est rendue possible grâce à l'utilisation d'une référence supplémentaire dans le mécanisme de prédiction inter-couches. L'extension SHVC améliore l'efficacité de codage de l'extension scalable du standard

H.264/MPEG-4 AVC, aussi appelé "Scalable Video Coding" (SVC) [2], de 50% tout en proposant une solution de codage moins intrusive. En effet, l'extension SVC définissait de nouveaux outils dans sa prédiction inter-couches, ce qui nécessitait une adaptation du mécanisme de codage par rapport au H.264/MPEG-4 AVC. Pour sa part, l'extension SHVC propose d'employer le même mécanisme de prédiction que le standard HEVC : les modifications apportées sont alors essentiellement syntaxiques.

Ces nouveaux standards permettent de réaliser des codages à la fois efficaces en terme de compression et rapides pour des contenus de grande résolution comme la HD ou l'UHD. La diffusion de tels contenus en condition temps-réel est alors rendu possible avec une qualité et des débits plus intéressants qu'avec les précédentes méthodes. De plus, l'extension SHVC permet également de répondre aux contraintes actuelles de transport. Les travaux réalisés durant cette thèse entre dans ce contexte.

1.2 Objectifs et plan de la thèse

L'objectif de cette thèse est de proposer des méthodes permettant de répondre à ces nouvelles contraintes en s'appuyant sur les méthodes de compression existantes. L'extension SHVC est une solution intéressante puisqu'elle permet une amélioration aussi bien sur l'efficacité du codage, mais aussi sur le transport des contenus. Par ailleurs, les organismes de standardisation "Advanced Television Systems Committee" (ATSC) et "Digital Video Broadcasting" (DVB), qui définissent des moyens de diffusion de la vidéo, considèrent l'extension SHVC comme une solution pour certains de leurs cas d'usage. Les travaux présentés dans cette thèse se focalisent sur des contraintes en terme d'efficacité de compression mais également en terme de vitesse de codage.

Le chapitre 2 présente l'état de l'art de la compression vidéo. Les caractéristiques et les nouvelles tendances des format vidéos sont d'abord présentés. Ensuite, les concepts et les standards de la compression vidéo employés sur ces dernières années sont présentés. Enfin, les outils spécifiques au codage SHVC et les cadres applicatifs pouvant avoir recours à ce type de compression sont exposés.

Dans le chapitre 3, cette thèse présente la méthode employée afin d'intégrer le codage SHVC pour une scalabilité en résolution dans un codeur professionnel. Cette méthode propose d'employer une architecture qui aligne les différentes couches afin qu'elles puissent être exécutées en parallèle tout en cherchant la meilleure efficacité de codage possible sur deux couches. Les outils de codage définis dans l'extension SHVC, sont analysés à travers différents types d'images afin de proposer un compromis entre efficacité de codage et complexité sur la prédiction inter-couches.

Ensuite, dans le chapitre 4, nous proposons une amélioration des performances à la fois en terme d'efficacité de codage et de vitesse de traitement sur le codeur SHVC proposé pour une scalabilité en résolution sur deux couches. Les améliorations en terme d'efficacité de codage sont obtenues en réalisant une optimisation sur le paramétrage du filtre de sous-échantillonnage nécessaire à une scalabilité en résolution. Le couche basse coûte plus cher en débit mais permet une prédiction inter-couches plus efficace sur la couche haute qui, en définitive, améliore le coût en débit global à qualité constante. De plus, les améliorations en terme de vitesse s'appuient sur un compromis entre distorsion et complexité. Plusieurs modes de qualité sont proposés avec une sélection sur les outils de codage définis dans le standard HEVC. Ces modes permettent de réaliser des codages en temps-réel sur différentes résolutions. Chaque couche se voit attribuer un de ces modes pour une scalabilité en résolution, ce qui permet une seconde amélioration de la prédiction inter-couches tout en conservant une vitesse de codage temps-réel.

Dans le chapitre 5, une analyse de l'architecture proposée ainsi que différentes adaptations présenteront la généralité de la méthode employée pour l'intégration du codage SHVC avec une scalabilité en résolution sur deux couches. Nous montrerons qu'elle permet d'ajouter d'autres couches sans modification de l'architecture ainsi que d'autres types de scalabilité. Les tests proposés montreront également que les performances en terme de vitesse et d'efficacité de codage sont conservés à la fois pour un nombre de couches supérieur à 2 mais également pour les autres types de scalabilité testés.

Le chapitre 6 présentera une intégration de la solution de codage SHVC dans un contexte de diffusion temps-réel. Ce contexte fait appel à une gestion par module des ressources d'acquisition, de codage et de transmission. L'intégration de la solution sur une plateforme multi-cœurs récente a permis de démontrer les performances temps-réel du codeur proposé lors de divers scénarios de démonstration pour une diffusion ATSC 3.0.

Enfin, cette thèse se termine par une conclusion globale sur les travaux effectués ainsi que par les directions futures de recherche.

Chapitre 2

Le codage vidéo, état de l'art

Introduction

Ce chapitre introduit les notions de compression vidéo nécessaire à la compréhension de cette thèse. Dans un premier temps, une définition de ce qui caractérise une vidéo est proposée. Ensuite, les différents outils, qui ont pu être utilisés au cours de cette thèse et permettant de qualifier une vidéo, sont présentés. La notion de codage vidéo est décrite au travers des standards de compression existants. Puis, le standard "High Efficiency Video Coding" (HEVC), ainsi que son extension scalable "Scalable High efficiency Video Coding" (SHVC), employés au cours de cette thèse sont détaillés. Enfin, un regard est porté sur les travaux d'étude ainsi que sur différents cadres applicatifs traitant du sujet de la compression vidéo scalable "Scalable High efficiency Video Coding" (SHVC).

2.1 Représentation d'un signal vidéo

Un signal vidéo est un signal constitué d'images qui se succèdent à une certaine vitesse. Une vidéo est donc caractérisée par cette vitesse aussi appelée fréquence d'images, ainsi que par les caractéristiques propres aux images qui la composent. Ces caractéristiques peuvent être sa dimension en résolution, son espace de couleur (ou colorimétrie) ou encore la quantité de données, aussi appelée profondeur, employée pour son élément de base : le pixel. Cette partie détaille ces différents éléments d'un signal vidéo.

2.1.1 Résolution et rapport de format d'images

La caractéristique de résolution définit la largeur et la hauteur d'une image en nombre de pixel. Certaines résolutions ont des noms propres comme la Haute Définition (HD) ou le Full HD correspondant respectivement à des dimensions de 1280×720 pixels (ou $1280 \times 720p$) et $1920 \times 1080p$. D'autres encore, comme le format UHD, peuvent définir une dimension, mais également d'autres caractéristiques de la vidéo qui peuvent varier selon le support et la zone géographique. Davantage d'informations sur ce format UHD sont données dans la section 2.1.5. On retiendra ici que le format UHD définit une résolution de $3840 \times 2160p$. Puisque la résolution est directement liée au nombre de pixel de l'image, l'augmentation de la résolution améliore le niveau de détail et constitue une amélioration de la qualité de la vidéo.

La caractéristique de rapport de format, quant à elle, définit le rapport de la largeur sur la hauteur de l'image. Historiquement, le premier rapport retenu pour le cinéma et la télévision était le $4/3$ ($1.33:1$). Aujourd'hui, le format $16/9$ ($1.78:1$) est celui principalement retenu pour la télévision. Pour le cinéma, la tendance est au format plus large ($1.85:1$).

2.1.2 Fréquence d'image

La caractéristique de fréquence d'image définit la vitesse d'affichage des images en Images par seconde (IPS). Historiquement, les premières fréquences d'images employées étaient de 16 IPS. Avec l'arrivée du cinéma parlant en 1927, la fréquence d'image a dû être rehaussée pour la reproduction de sons aigus. La fréquence de 24 IPS a

été choisie pour assurer un compromis entre la technique et les coûts de production et de distribution. Après l'entrée dans l'ère du numérique, ces coûts ont largement diminué et permettent désormais d'augmenter la fréquence d'image. En Europe et aux Etats-Unis la fréquence d'image pour la télévision passerait respectivement de 25 IPS à 50 IPS et de 30 IPS à 60 IPS. Il est même envisageable d'atteindre des fréquences de 100 IPS et 120 IPS comme cela a pu être montré dans le cadre du projet 4EVER-2, voir section 2.1.5. Pour ces fréquences supérieures à 60 IPS, on parle de HFR. Davantage d'éléments sur cette technologie sont présentés en section 2.1.5.

De plus, les images peuvent se succéder de façon progressive, les unes après les autres, ou bien suivant un entrelacement. L'affichage progressif utilise tous les pixels d'une image tandis que l'affichage entrelacé n'en utilise qu'une partie par image. L'entrelacement des images n'affiche que les lignes de pixel paires et impaires respectivement sur les images paires et impaires. Ainsi, une image dans une séquence entrelacée consomme deux fois moins de données par comparaison au mode progressif à la même fréquence d'image. L'entrelacement peut néanmoins ajouter des artefacts après la compression qui constituent un défaut ou une déformation de l'image pouvant gêner la visualisation.

2.1.3 Représentation des couleurs et espaces colorimétriques

Afin de représenter les couleurs, les images se basent sur un espace de couleur (ou colorimétrique) représentant des couleurs dans un système de synthèse. Aujourd'hui, il existe plusieurs espaces de couleur qui se définissent par rapport au standard CIE-XYZ proposé par le Comité Internationale de l'Éclairage (CIE). Ce standard représente de manière abstraite l'ensemble des couleurs visibles par le Système Visuel Humain (SVH). Ainsi, des espaces de couleurs tels que le ITU-R BT.709 (BT.709), le DCI-P3, le ITU-R BT.2020 (BT.2020) ou encore le CIE-RVB regroupent un sous-ensemble des couleurs visibles comme le montre le schéma sur la Figure 2.1. Cette illustration présente les différents espaces de couleur avec un blanc D65 correspondant à une exposition sous une lumière naturelle en plein jour en zone tempérée.

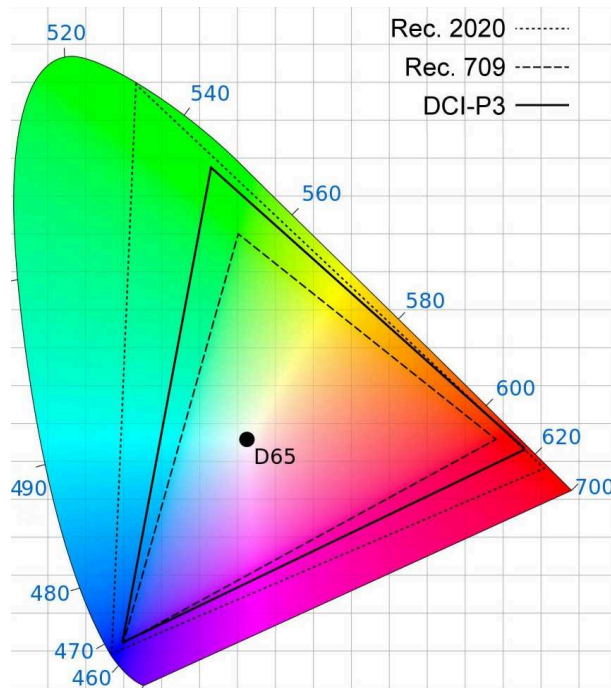


FIGURE 2.1 – Espaces de couleur BT.709, DCIP3, BT.2020 dans le domaine du visible à une luminosité D65

2.1.4 Représentation numérique, profondeur de données

Dans une image, la couleur est généralement représentée sur trois composantes. C'est le cas par exemple de la représentation des couleurs avec les composantes Rouge, Vert et Bleu (RVB) qui toutefois ne permet pas de

représenter l'ensemble du domaine visible. De plus, il a été observé que le SVH est plus sensible aux variations de lumière qu'aux variations de couleur, la représentation YUV exploite cette caractéristique. La composante Y représente la luminance et les composantes U et V, aussi appelées composantes chromatiques ou chrominances, représentent les couleurs. Une image en couleur est alors composée à partir des 3 images des composantes Y, U et V. Puisque le SVH est plus sensible aux informations de luminance, il est possible de réduire la dimension des images des chrominances en employant un échantillonnage sur ces composantes. Les échantillonnages couramment employés sont notamment 4:4:4, 4:2:2 et 4:2:0. Comme illustré sur la Figure 2.2, les échantillonnages 4:4:4, 4:2:2 et 4:2:0 utilisent sur chaque ligne 4 pixels pour Y et respectivement, 4 ou 2 pixels pour U et V sur les lignes paires et 4 ou 2 ou 0 pixels pour U et V sur les lignes impaires. Ainsi l'échantillonnage 4:2:2 consiste à diviser la largeur de l'image par deux pour les images de chrominance. L'échantillonnage 4:2:0 consiste, lui, à diviser la largeur et la hauteur de l'image par deux pour les images de chrominance. Cet échantillonnage constitue donc une première compression des données d'une image avec un compromis entre occupation mémoire et fidélité des couleurs.

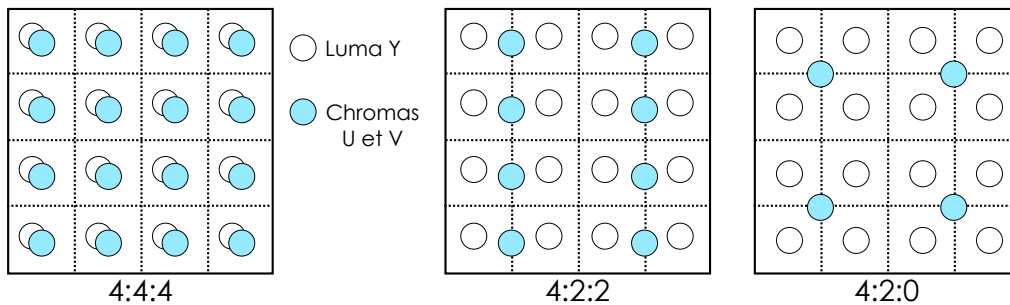


FIGURE 2.2 – Schéma des échantillonnages 4 : 4 : 4, 4 : 2 : 2 et 4 : 2 : 0. 4 : 4 : 4 : les tailles des images de luminance et de chrominance sont égales. 4 : 2 : 2 : la largeur des images de chrominance est 2 fois moins grande que celle de l'image de la luminance. 4 : 2 : 0 : la largeur et la hauteur des images de chrominance sont chacune d'elles 2 fois moins grandes que celles de l'image de la luminance.

De plus, chaque pixel est mesuré en nombre de bits qui constitue la profondeur des données. En général, on alloue 8 bits à un pixel, soit un total de 256 valeurs, pour représenter un échantillon sur une composante. Ainsi, un pixel d'une image en couleur requiert 24, 16 ou 12 bits respectivement avec un échantillonnage 4:4:4, 4:2:2 ou 4:2:0. Pour une profondeur de 8 bits par pixel, les 256 valeurs par composante ne suffisent pas toujours à représenter l'ensemble des contrastes et génèrent alors des dégradés visibles : on parle alors de *bandings* comme illustré sur la Figure 2.3. La quantité de données attribuée à chaque pixel permet de représenter un sous-ensemble des couleurs du spectre lumineux mais définit également un intervalle entre chaque niveau de couleur. Le banding apparaît lorsque cet intervalle est trop faible pour le contraste à afficher. Un moyen de réduire cet effet est d'augmenter la quantité de donnée attribuée aux pixels.

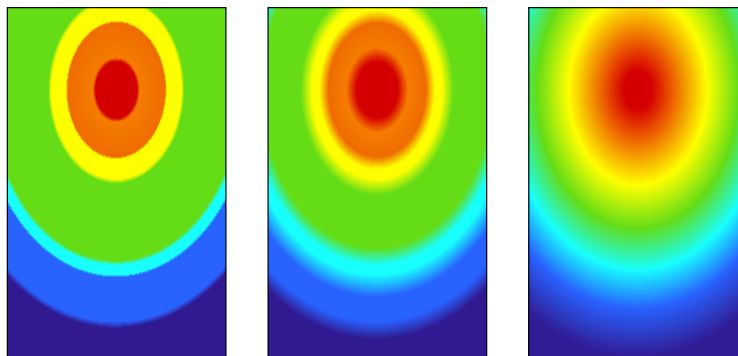


FIGURE 2.3 – Illustration de l'effet du banding sur des profondeurs de 6, 8 et 10 bits respectivement de gauche à droite

Il y a donc un réel intérêt, en terme de qualité vidéo, à augmenter la quantité de données allouer à chaque pixel

malgré le surcote en mémoire que cela engendre. Ainsi, la profondeur est également un paramètre ayant un effet direct sur la qualité vidéo.

Perception visuelle humaine

Le SVH peut être décrit sous la forme d'un réseau de capteur. Les yeux assurent la captation des signaux lumineux, les nerfs optiques font office de réseau de transfert, et enfin, le cerveau interprète ces signaux. C'est le cerveau qui est responsable de la reconnaissance des formes et des couleurs. Les yeux, quant à eux, captent des ondes sur une plage de longueur d'ondes comprises entre 380 nm et 780 nm, constituant le domaine de la lumière visible. Cette plage et les couleurs qui y sont associées sont représentées sur la Figure 2.4.

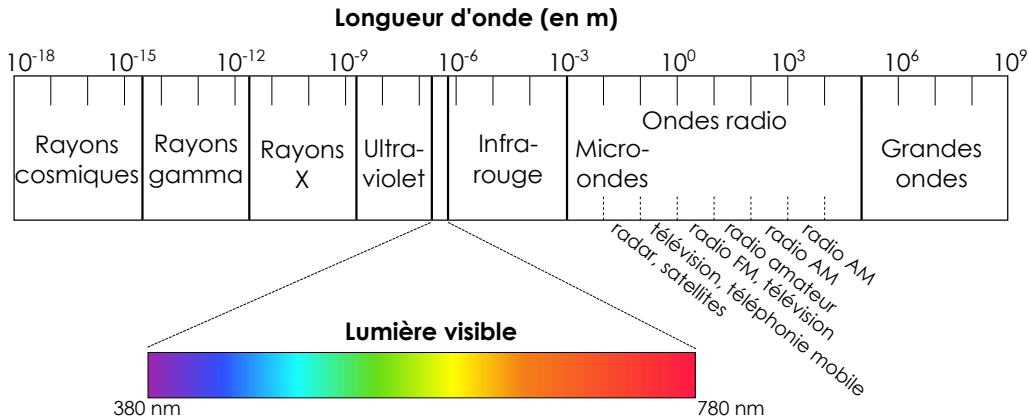


FIGURE 2.4 – Association des couleurs aux longueurs d'ondes des signaux lumineux.

Comme détaillé dans [3], l'œil humain permet de capter ces ondes lumineuses en utilisant des cellules appelées photo-récepteurs. Ces photo-récepteurs sont majoritairement les cônes et les bâtonnets. Les cônes permettent de distinguer les couleurs et fonctionnent efficacement dans un environnement lumineux. Les bâtonnets fonctionnent plus efficacement dans un environnement plus sombre et sont donc moins souvent utilisés de nos jours du fait des multiples sources lumineuses. L'œil humain dispose de trois types de cônes chacun plus ou moins sensibles à une partie du spectre lumineux : les cônes L (plus sensibles au rouge), les cônes M (plus sensibles au vert), et les cônes S (plus sensibles au bleu). La Figure 2.5 présente les pics de sensibilité des différents types de cônes. La couleur jaune, par exemple, stimulera les cônes M et L mais pas les cônes S. La perception des couleurs dépend de la combinaison de ces stimuli : l'œil humain est donc par définition tri-chromate. En combinant différents niveaux de bleu, de vert et de rouge, il est alors possible de générer un ensemble de couleurs perceptibles par l'œil humain, on parle alors de synthèse additive. Cette synthèse additive concerne tous les mélanges de lumière colorée.

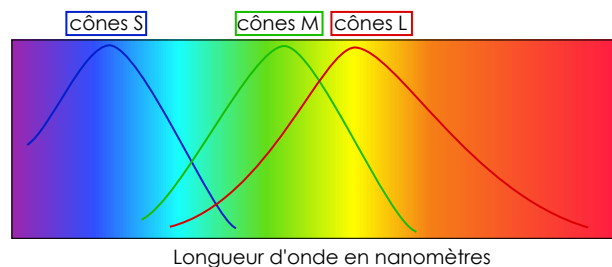


FIGURE 2.5 – Association des couleurs aux longueurs d'ondes des signaux lumineux.

La perception humaine est faiblement limitée en terme de distance, à moins d'un obstacle ou d'une trop faible intensité lumineuse. A titre d'exemple, l'élément le plus lointain visible à l'œil nu est la galaxie d'Andromède, située à 2.6 millions d'années-lumière de la Terre.

TABLE 2.1 – Format Haute Définition employé en diffusion et en stockage

Format UHD	Appellation selon la zone géographique		Résolutions	Format d'images
UHDTV1	Europe	UHD-1	3840×2160p	16/9
	Japon	Super Hi-Vision 4K		
UHDTV2	Europe	UHD-2	7680×4320p	16/9
	Japon	Super Hi-Vision 8K		
Ultra Blu-ray DCI 4K			3840×2160p	16/9
			4096×2160p	17/9

2.1.5 Formats vidéo d'aujourd'hui et de demain

Aujourd'hui, la tendance dans le monde de la diffusion est de proposer des contenus de qualité améliorée en augmentant les résolutions, la profondeur des données, les espaces de couleur, la fréquence d'image... Ces diverses pistes d'amélioration ont donné lieu à de nouveaux formats vidéos, à de nouvelles technologies permettant d'exploiter ces formats ainsi qu'à la formation de projet collaboratif entre entreprises et laboratoires afin d'améliorer l'expérience de consommation vidéo.

4EVER et 4EVER-2

Le projet 4EVER [4] a débuté en 2012 et s'est terminé en 2015. Il a réuni neuf partenaires industriels et académiques français. L'objectif était d'apporter une amélioration de la qualité d'expérience vidéo. Les travaux menés concernaient notamment l'introduction de la nouvelle résolution UHD ainsi que l'adoption du standard HEVC dans le monde de la diffusion. Au travers de plusieurs expérimentations, le projet 4EVER a permis de montrer, pour la première fois au monde, une chaîne opérationnelle temps-réel exploitant le standard HEVC sur des résolutions UHD de bout en bout : c'est à dire de la captation par caméra à l'affichage sur écran en passant par le codage et la transmission.

Le projet 4EVER-2 [5] est la suite du projet 4EVER, il a débuté en 2015 et s'est achevé en 2017. Les objectifs d'amélioration de la qualité d'expérience vidéo ont davantage ciblé des technologies innovantes et notamment les formats HDR et HFR. Le projet a donné lieu à plusieurs démonstrations de la chaîne temps-réel de bout en bout [6, 7, 8, 9].

UHD

Le format UHD se définit non seulement par la résolution 3840×2160p, mais aussi par d'autres aspects de la qualité vidéo selon le support employé. Ces autres paramètres peuvent être la fréquence d'images, la profondeur des pixels ou même l'espace de couleur. Dans le cas de la télévision, on parle d'UHDTV en Europe qui se décline en UHDTV1 et UHDTV2 et de "Super-Hi-Vision" au Japon. Pour le cinéma on parle de "Digital Cinema Initiatives" (DCI) 4K. La Table 2.1 présente les différentes appellations et caractéristiques des formats UHD par utilisation.

Le groupe de diffusion DVB, prévoit, à l'horizon 2020, la possibilité de diffuser des contenus à la résolution 8K comme illustré sur la road-map présentée en Figure 2.6.

HDR

La technologie HDR propose une amélioration de la dynamique de l'image. Comparée à la technologie "Standard Dynamic Range" (SDR) utilisée jusqu'à alors, le HDR permet de représenter et de mémoriser plusieurs niveaux d'intensité lumineuse au sein d'une même image comme illustré sur la Figure 2.7. Cette technologie s'appuie sur une utilisation des larges espaces de couleur tels que le BT.2020, une augmentation de la profondeur des données à un minimum de 10 bits par pixel par composante et des informations de signalisation servant à l'affichage.

La standardisation de la technologie HDR a permis de formaliser plusieurs spécifications au cours des dernières années définissant les technologies HDR parmi lesquelles on trouve les standards :

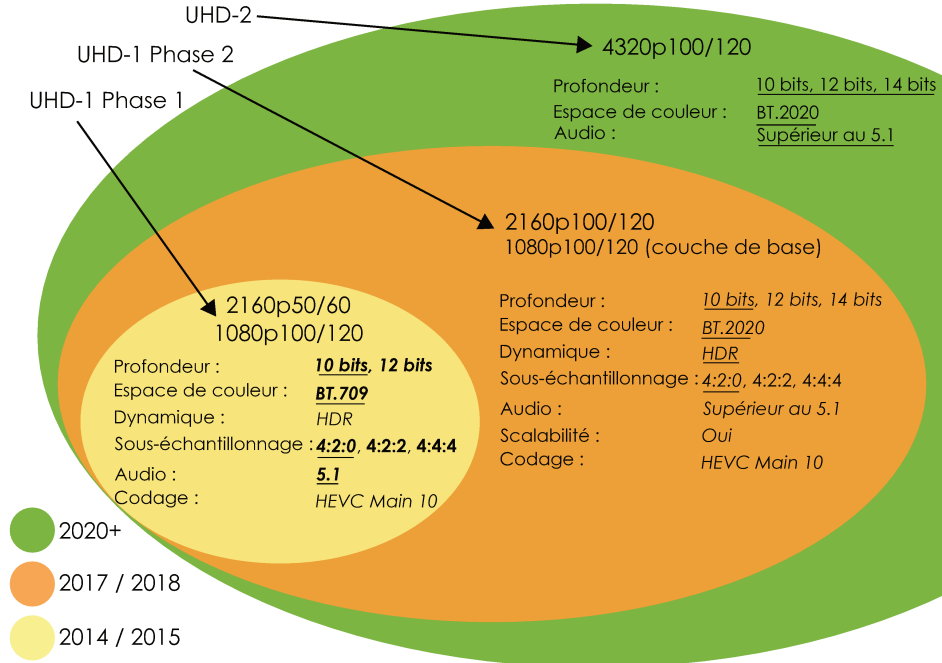


FIGURE 2.6 – Road-map de la diffusion UHD prévu par DVB

- "Perceptual Quantizer" (PQ). Il est publié par la "Society of Motion Picture and Television Engineers" (SMPTE) avec la spécification ST-2084. Cette dernière définit une fonction de transfert qui autorise l'affichage HDR pour un espace de couleur BT.2020 et une profondeur des pixels sur 10 bits.
- HDR10. Il exploite le standard PQ ainsi que la spécification SMPTE ST-2086 qui définit le transfert d'informations servant à la calibration des couleurs au niveau de l'affichage HDR. Ces informations sont présentées sous la forme de méta-données de tailles fixes.
- HDR10+. Ce standard, proposé par *Samsung* et *Amazon Video*, exploite et améliore le standard HDR10 grâce à une gestion des méta-données de taille variable. Cette gestion est définie dans la spécification SMPTE ST-2094-40.
- Dolby Vision. Il s'agit d'un format propriétaire défini par *Dolby* qui exploite la technologie PQ avec des méta-données qui peuvent être de tailles fixes (spécification SMPTE ST-2086) ou variable en s'appuyant sur la spécification SMPTE ST-2094-10. Par rapport au HDR10+, le standard Dolby Vision peut adapter le signal HDR à des télévisions avec un éclairage (mesuré en nits) moins important.
- "Hybrid Log Gamma" (HLG). HLG est un standard proposé par la "*British Broadcasting Corporation*" (BBC) et la "*Nippon Hoso Kyokai*" ("*Japan Broadcasting Corporation*") (NHK) qui définit une autre fonction de transfert que le PQ pour l'affichage HDR. Le standard HLG est défini dans la spécification ARIB STD-B67.
- SL-HDR1. Il s'agit d'un standard proposé par les sociétés *STMicroelectronics*, *Philips International* et *Technicolor* dans la spécification ETSI TS 103 433. Cette dernière s'appuie sur les spécifications SMPTE ST-2084, ST-2086, ST-2094-20 et ST-2094-30. Les méta-données sont transmises sur des messages Information d'Amélioration Supplémentaire ("Supplemental Enhancement Information") (SEI) permettant une utilisation des standards de compression H.264/MPEG-4 AVC et HEVC.

HFR

La technologie HFR consiste à augmenter la fréquence d'image au delà de 60 IPS. L'augmentation de la fréquence d'image permet d'améliorer la qualité vidéo puisqu'elle élimine l'effet de flou de mouvement comme illustré sur la Figure 2.8. Cet effet survient lors d'un mouvement provoqué par l'objet filmé ou par un mouvement de caméra. Le flou constitue une source d'erreurs de codage vidéo, il est donc intéressant de le réduire au moment de la captation. Néanmoins, le principal inconvénient de cette technologie repose sur la quantité de données à traiter



FIGURE 2.7 – Exemple de comparaison SDR vs HDR

qui impacte à la fois leur stockage et la vitesse de traitement. Ces contraintes concernent notamment le codage et le traitement en temps-réel.



FIGURE 2.8 – Exemple de captation à 100 IPS et 50 IPS. A 100 IPS, l'image captée contient moins de flou de mouvement et améliore le niveau de détail.

Vidéo à 360 degrés

La vidéo à 360 degrés, aussi appelée vidéo immersive, ou vidéo sphérique est une vidéo dans laquelle toutes les directions sont capturées en même temps. Pour leur capture, des réseaux de caméra ou bien une caméra dédiée à la capture de plusieurs directions sont employés afin d'obtenir plusieurs vues au même moment. Chacune des images capturées, propre à une direction, est alors assemblée aux autres en utilisant la méthode de *stitching*. Cette méthode consiste à analyser plusieurs images afin d'identifier les éléments communs aux différentes images et à les assembler en conséquence. Cette opération peut être réalisée au sein de la caméra ou bien par un ap-

pareil dédié. L'ensemble de la scène alors capturée est stocké dans une image de grande résolution, comme par exemple 3840×2160 p, ou plus grande encore, selon la résolution d'affichage souhaitée. Les résolutions d'affichage sont moins grandes et dépendent du système d'affichage : écran d'ordinateur, écran de smartphone, casque de réalité virtuelle... L'affichage de vidéo à 360 degrés est ensuite effectué sur des écrans interprétant une direction en fonction de la position d'un curseur (visualisation avec un ordinateur) ou bien selon un capteur gyroscopique (visualisation avec un smartphone ou un casque de réalité virtuelle). Il existe déjà aujourd'hui des services proposant une visualisation de vidéo à 360 degrés proposées par YouTube, Facebook ou Vimeo par exemple.

La vidéo à 360 degrés représente un intérêt par exemple pour des applications de vidéo conférence, de documentaire ou de publicité. La qualité vidéo de cette technologie dépend de la capacité à transmettre et à stocker de grande quantité de données. En effet, avec une résolution plus grande de l'image totale, la qualité de l'image visualisée pourra être améliorée. Il existe des méthodes permettant de réduire la bande passante lors de la transmission, par exemple, en adaptant la quantité de données en fonction des zones d'intérêt [10, 11, 12, 13].

L'expérience utilisateur peut être améliorée en utilisant des formats qui augmentent le paramétrage de certaines caractéristiques de l'image comme la résolution, la fréquence, la profondeur et les espaces de couleur. Cette amélioration a pu être évaluée lors de précédentes études sur les nouveaux formats proposés. Par exemple, les articles [14, 15, 16, 17] proposent d'étudier l'impact des formats HD et UHD sur la qualité perçue respectivement dans des contextes de transmission, de distorsion à haut et bas débit, sur l'attention portée par le regard et pour des contenus immersifs. Toutes ces études concluent que le format UHD présentent une amélioration de la qualité à haut débit par rapport au format HD. De même, les études [18, 19, 20] donnent une évaluation du format HDR par rapport au format SDR. Elles étudient respectivement l'impact de la comparaison par rapport à une sortie de caméra, le parcours du regard et l'impact sur la perception de la qualité objective et subjective. Ces études concluent à une nette préférence des utilisateurs pour le format HDR. En définitive, le rehaussement des paramètres des formats vidéo permet une amélioration de la qualité qui peut être chiffrée. Ce chiffrage ou cette évaluation de la qualité s'appuie sur des méthodes appelées *métriques* qui peuvent faire intervenir l'humain (subjective) ou bien une analyse mathématique (objective).

2.2 Métriques d'évaluation objectives

Afin de qualifier la qualité d'un codage vidéo, il est possible de faire appel à des métriques d'évaluation qui peuvent être objectives ou subjectives. Les métriques objectives qualifient une vidéo de façon mathématique en analysant les données du signal vidéo. Parmi ces métriques, on trouve notamment les métriques "Peak Signal to Noise Ratio" (PSNR) et "Structural SIMilarity" (SSIM). Les métriques subjectives, quant à elles, qualifient une vidéo en s'appuyant sur des réponses obtenues par des tests impliquant une notation par des testeurs humains. Ces métriques permettent alors de définir des outils psychovisuels qualifiant le codage en se basant sur les observations humaines. Elles permettent, par exemple, d'identifier des zones où se poseront le regard et où il peut être intéressant d'améliorer la qualité en augmentant le débit de codage.

2.2.1 PSNR

La métrique PSNR mesure la distorsion entre un signal et sa source. Cette métrique, quand elle est appliquée aux signaux vidéo, permet d'obtenir une bonne approche de la qualité d'une vidéo. Les formules ci après détaillent ce procédé basé sur l'utilisation de l'Erreur Quadratique Moyenne ("Mean Square Error") (MSE).

$$MSE = \frac{1}{w \times h} \times \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} (I(i; j) - K(i; j))^2 \quad (2.1)$$

$$PSNR = 20 \times \log_{10}(2^B - 1) - 10 \times \log_{10}(MSE) \quad (2.2)$$

Avec $I(i; j)$ et $K(i; j)$ respectivement les images source et reconstruite aux coordonnées $(i; j)$, avec w la largeur de l'image, h sa hauteur et B sa profondeur de données en nombre de bits.

Le PSNR est exprimé en dB, plus sa valeur est élevée, meilleure est la qualité. Puisque le PSNR est calculé sur un signal, il peut être calculé sur une image ou sur un ensemble d'image. Le calcul du PSNR ne prend pas effet sur des zones d'intérêt. Ceci peut être problématique dans certains cas, comme par exemple quand l'ensemble de l'image est flou ou bien quand le regard est attiré sur une zone spécifique de l'image.

2.2.2 SSIM

La métrique de qualité SSIM [21] mesure la différence entre deux images. Contrairement au PSNR, le SSIM mesure cette différence par zone structurale et non par pixel. La fenêtre de traitement alors définie peut se déplacer dans l'image. La formule de la valeur du SSIM est donnée ci après :

$$SSIM(x : y) = \frac{(2 \times \mu_x \times \mu_y + c_1) \times (2 \times \sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1) \times (\sigma_x^2 + \sigma_y^2 + c_2)} \quad (2.3)$$

$$c_1 = k_1 \times 2^B - 1 \text{ et } c_2 = k_2 \times 2^B - 1 \quad (2.4)$$

Les caractères μ_x et μ_y sont les moyennes des fenêtres x et y , σ_x^2 et σ_y^2 sont leurs variances et σ_{xy} est leur covariance. Enfin, c_1 et c_2 sont des paramètres calculés respectivement en fonction de la profondeur des données et de k_1 et k_2 . Ces deux derniers paramètres valent usuellement respectivement 0.01 et 0.03 mais peuvent être adaptés selon le cas d'usage.

2.2.3 BD-BR et BD-PSNR

La métrique de Bjontegaard [22] calcule la différence entre deux courbes débit/distorsion. Le débit correspond au débit de codage exprimé en kbits/s alors que la distorsion est exprimée à partir de la mesure du PSNR en dB. Cette méthode utilise des courbes débit/distorsion formées à partir de quatre points d'abscisses et d'ordonnées respectivement basées sur les débits de codage et les mesures du PSNR. Elle calcule l'interpolation des courbes à l'échelle logarithmique afin de calculer l'écart qui les sépare. Ce calcul permet d'obtenir, soit la différence moyenne des PSNR, notée BD-PSNR et exprimée en dB, soit la différence moyenne des débits, notée BD-BR et exprimée en pourcentage comme illustré sur la Figure 2.9. La différence ainsi exprimée permet d'évaluer respectivement les gains/pertes en qualité et les gains/pertes en débit d'une solution de codage par rapport à une autre.

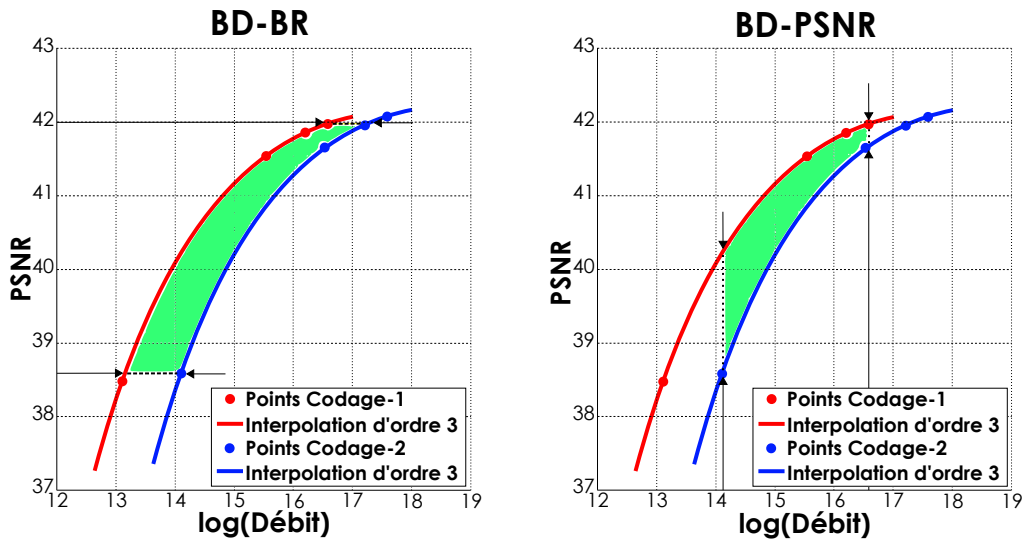


FIGURE 2.9 – Exemple de calcul du BD-BR et du BD-PSNR entre deux courbes débit/distorsion.

Dans la suite de cette thèse, cette métrique sera employée pour qualifier l'efficacité du codage des solutions proposées par rapport à d'autres codeurs. Les codages étant réalisés à qualité constante, la mesure BD-BR donnera une indication du gain ou de la perte en débit.

2.2.4 Métriques actuelles

Les métriques actuelles cherchent davantage à se rapprocher des métriques subjectives. Les métriques "Visual Quality Model for Variable Frame Delay" (VQM-VFD) [23] en open-source et "Video Multimethod Assessment Fusion" (VMAF) [24] développée par Netflix, proposent toutes deux de combiner plusieurs métriques de qualité

vidéo. Ces travaux cherchent à exploiter les avantages de plusieurs métriques en s'appuyant sur une méthode de machine learning. Le machine learning consiste à définir un comportement à partir de règles appliquées et affinées sur de très grande quantité de données. Pour cela, le programme est généralement exécuté dans une phase d'apprentissage durant laquelle il va définir son comportement à partir des règles fournies par le développeur et d'une évaluation/correction sur les données traitées. Ensuite, le programme pourra être utilisé afin de donner le meilleur comportement à suivre selon les précédentes expérimentations. Dans le cas des métriques de qualité, cela consiste à identifier quelle métrique donnera le meilleur résultat dans une certaine situation puis à employer cette métrique en tant que consigne pendant la phase de compression. La métrique alors produite est constituée d'un ensemble de métriques plus ou moins complexes et de règles définissant la situation dans laquelle elles doivent être employées.

Toutes ces métriques permettent d'évaluer la distorsion par rapport à la source provoquée par les outils de compression. Ces outils ont évolué au cours de ces dernières années à travers leur utilisation dans différents standards de compression.

2.3 Les outils de la compression vidéo

Le codage vidéo profite de la redondance intrinsèque des séquences afin de réduire la quantité de données une image après l'autre. Pour cela, l'algorithme de compression utilise des outils de prédiction et de compression d'énergie du signal. Les outils de prédiction permettent d'obtenir une image prédite à partir d'informations représentant un faible espace de stockage comme des vecteurs directionnels au sein d'une image ou des vecteurs de mouvement entre deux images respectivement pour des prédictions spatiales et temporelles. L'image prédite est alors comparée à l'image source pour en extraire un différentiel. Ce différentiel est compressé sous forme de coefficients grâce à l'opération de transformée. Les coefficients sont ensuite quantifiés permettant de conserver ou d'éliminer ceux ayant respectivement un poids important ou moindre. Toutes ces informations de codage sont ensuite envoyées vers un codeur entropique qui les ré-organise afin de compresser la signalisation. Ces informations peuvent également être employées pour reconstruire l'image compressée dans le but de l'utiliser dans la prédiction temporelle. L'ordre de traitement des images peut varier en fonction de la configuration du codage, on distingue alors l'ordre d'affichage et l'ordre de codage. L'ordre d'affichage des images correspond à la fois à l'ordre dans lequel les images doivent être affichées mais aussi à l'ordre dans lequel elles sont reçues par le codeur. L'ordre de codage, quant à lui, dépend de la configuration du codage et est déterminé au sein du codeur. Cet ordre de codage est notamment utile à la prédiction temporelle. La Figure 2.10 schématise un algorithme de compression réalisé sur les images suivant l'ordre de codage. L'image est d'abord répartie en blocs. Ces blocs sont prédits grâce aux prédictions spatiales et temporelles. La différence entre le bloc prédit et le bloc source est ensuite transformée puis quantifiée.

2.3.1 Répartition en blocs de pixel

La première étape du codage consiste à diviser l'image en un ensemble de blocs qui seront l'élément de base des autres opérations de codage. Chaque bloc contient une quantité spécifique de pixels généralement sous la forme d'un carré de taille $N \times N$. Une grande taille de ces blocs permettra de réaliser un traitement rapide tandis qu'une taille plus petite permettra de réaliser un traitement plus précis. Ces blocs peuvent être affinés dans les opérations de prédiction ainsi que dans l'opération de transformée. Les blocs et leurs affinements sont lus et traités de gauche à droite puis de haut en bas. Ce sens de lecture, aussi appelé ordre de balayage ou parcours en zig-zag, permet de définir les différentes dépendances entre blocs notamment lors de l'opération de prédiction.

2.3.2 Prédiction

La prédiction permet d'exploiter les corrélations temporelles et spatiales respectivement au travers des prédictions inter et intra images. Dans le cas de la prédiction intra, le bloc traité utilise les blocs proches et déjà codés en tant que référence. Quand aucun bloc n'est disponible, la prédiction intra définit aussi le moyen de palier le manque de référence. Dans le cas de la prédiction inter, les blocs prédits utilisent des blocs des images reconstruites (i.e. les images précédemment traitées) en tant que référence. Cette prédiction utilise des vecteurs de mouvement associant les coordonnées du bloc traité à celles du bloc de référence. Il est possible également d'employer deux blocs de référence pour la prédiction inter, on parle alors de bi-prédiction. De plus, les prédictions intra et inter

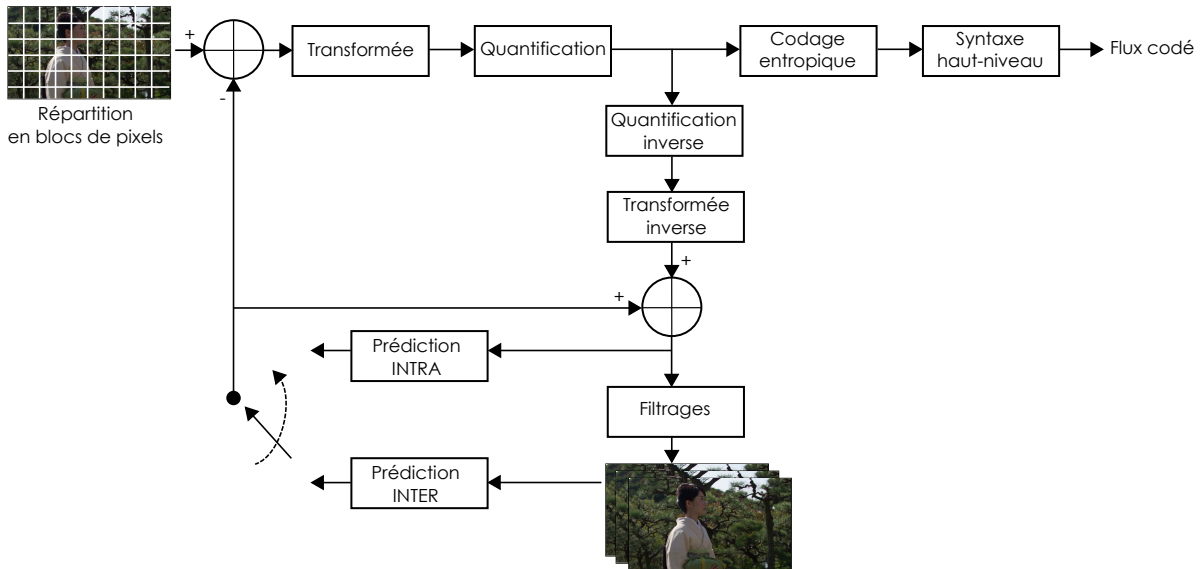


FIGURE 2.10 – Schéma de codage vidéo général.

peuvent s'appliquer à une partition du bloc traité afin d'affiner les détails. En terme de signalisation, la prédiction doit donc transmettre au moins :

- les coordonnées du bloc traité,
- le mode de prédiction employé (inter ou intra),
- les partitions éventuelles,
- un moyen de localiser le bloc prédicteur (direction, vecteurs de mouvement...)

La prédiction permet alors de produire un bloc prédit ou à terme une image prédite qui ressemble à l'image source malgré quelques différences. La différence entre les blocs sources et les blocs prédits permet d'obtenir les blocs résiduels ou résidus qui seront transformés puis quantifiés. La conservation des résidus permet d'améliorer le niveau de détail du codage.

2.3.3 Transformée

L'opération de transformée s'applique sur les résidus de la prédiction : elle consiste à compacter l'énergie du signal en réduisant sa corrélation. Pour cela, elle emploie des outils se basant généralement sur une transformée discrète avec des algorithmes du type Transformée en Cosinus Discrète ("Discrete Cosine Transform") (DCT), Transformée en Sinus Discrète ("Discrete Sine Transform") (DST) ou encore Transformée en Ondelettes Discrète ("Discrete Wavelet Transform") (DWT). Les algorithmes de transformée doivent être inversibles afin de ne pas ajouter d'altération de l'image au décodage. Le résultat de la transformée se présente sous la forme de coefficients.

2.3.4 Quantification

La quantification est appliquée aux coefficients de la transformée afin de réduire leur occupation mémoire. En effet, ces coefficients peuvent être nombreux et pour certains n'avoir qu'un impact mineur sur la précision du bloc traité. L'opération de quantification consiste à diviser et arrondir les coefficients afin de conserver les coefficients ayant le plus d'impact sur les détails. Cette opération n'est pas réversible pour tous les coefficients puisque les plus faibles peuvent être perdus durant le processus. Dans ce cas, la quantification est source de perte de données dans la compression, c'est pourquoi il est primordiale de réaliser une bonne prédiction pour un codage efficace.

2.3.5 Codage entropique

Le codage entropique utilise des statistiques sur un signal pour construire un code à longueur variable. Un code à longueur variable attribue les mots les plus courts aux symboles les plus fréquents. Pour cela, les éléments de la répartition des blocs, des prédictions, des transformées et des quantifications sont analysés et un code en fonction de leur fréquence leur est attribué. Cette opération permet de réduire l'occupation mémoire du flux codé sans perte d'information mais nécessite un temps de traitement long. De plus, cette opération requiert un certain nombre de bloc pour être efficace. Elle est généralement effectuée une fois que tous les blocs d'une image ont été prédits, transformés puis quantifiés.

2.3.6 Filtrages

Les différentes opérations de filtrages (anti-bloc, adaptatif des échantillons...) interviennent lors de la reconstruction de l'image traitée. Ils permettent d'affiner, voire d'améliorer, la qualité de l'image reconstruite en diminuant certains défauts visuels générés par les traitements précédents. Ils sont généralement responsables d'une part importante de la complexité du codage vidéo.

2.3.7 Syntaxes haut-niveau

La syntaxe haut-niveau regroupe les informations des différents outils employés pour le codage et constitue l'entête du flux codé (i.e. *bitstream*). Cette syntaxe sert à indiquer les différents outils et les différentes limitations choisies dans le codeur au moment du décodage. Plus le codeur est complexe, plus cette syntaxe sera longue.

Le signal vidéo compressé sera ainsi organisé en suivant la syntaxe haut-niveau. On trouvera notamment les informations donnant le comportement des différentes étapes de répartition en blocs, de prédiction, de transformée, de quantification, de codage entropique et de filtrage. De cette façon, le décodeur aura accès au même paramétrage des outils employés par le codeur. Le signal compressé contiendra également les résidus ré-organisés selon le paramétrage du codage entropique. L'ensemble des opérations du codage vidéo est composé de différents algorithmes plus ou moins complexes. Il en existe plusieurs plus ou moins efficaces et se combinant plus ou moins bien avec d'autres. Afin d'unifier le codage et le décodage vidéo à travers le monde, un processus de standardisation s'est développé ces dernières décennies.

2.4 Les standards de la compression vidéo

2.4.1 Évolution de la consommation vidéo et tendances

La consommation vidéo sur ces dernières années a connu un très important accroissement et promet de grandir encore davantage sur les prochaines années. Cette augmentation s'explique par la diversité des appareils pouvant demander un accès à un contenu vidéo (télévision, mobile, tablette, console de jeu...). L'augmentation de la consommation vidéo est aussi liée à la multiplication des médias de communication (télévision par liaison terrestre et satellite, par liaison internet, plateforme de diffusion de contenu sur internet...). En ce qui concerne les accès à la vidéo par internet, Cisco prévoit dans [25] une forte augmentation de leur consommation comme illustré sur la Figure 2.11. Les données de trafic sont exprimées en Petaoctets (Po), soit 10^{15} octets, par mois. La plage TV par Internet correspond aux données à destination d'un écran de télévision que ce soit par une set-top-box, par une connexion internet ou par une connexion ordinateur-télévision (i.e. PC-TV). La plage Vidéo inclut les contenus en accès streaming, les communications vidéo, les vidéos de sécurité personnelle et les vidéos transitant sur les réseaux mobiles (2G, 3G et 4G). On constate que Cisco prévoit que la consommation sur la plage Vidéo augmentera de 30% environ tous les ans.

2.4.2 Nécessité d'un codage

Dans ce contexte d'accroissement de la consommation vidéo, le stockage et la transmission de vidéo deviennent un véritable défi technologique. Plusieurs solutions définissent des standards de compression ainsi que des conteneurs pour permettre la transmission. Le conteneur est un format de fichier pouvant contenir divers types de données. Les spécifications du format d'un conteneur décrivent la façon dont sont organisées les données à l'intérieur

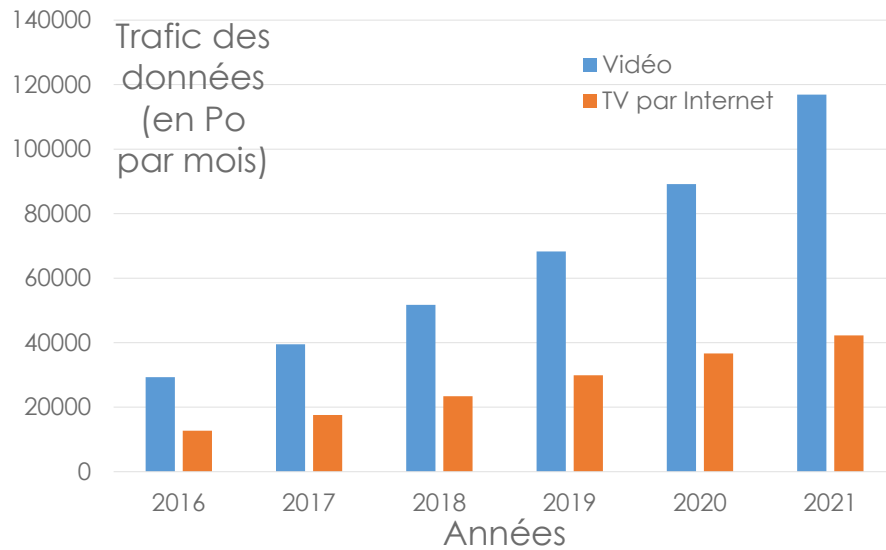


FIGURE 2.11 – Préviation de la consommation vidéo sur Internet par Cisco en Po/mois. TV par internet : données à destination d'un écran de télévision que ce soit par une set-top-box, par une connexion internet ou par une connexion PC-TV. Vidéo : contenus en accès streaming, communications vidéo, vidéo de sécurité personnelle et vidéo transitant sur les réseaux mobiles.

du fichier. Les conteneurs sont fortement utilisés dans le domaine du multimédia ; ils peuvent contenir des flux vidéo et/ou audio, en général compressés à l'aide de standards de compression. Des conteneurs plus avancés permettent également de stocker des sous-titres, des éléments de chapitrage, ainsi que d'autres informations concernant le média. Le standard de compression définit quant à lui des outils de codage ainsi qu'un format de fichier pouvant être interprété seulement par un décodeur. Ce format vidéo compressé, appelé "Elementary Stream" (ES), contient toutes les informations nécessaires à son interprétation. Il existe plusieurs types de standard de compression employés par les différents moyens de diffusion.

2.4.3 Les standards de compression vidéo existants et à venir

La compression vidéo consiste à réduire la quantité de données nécessaire à la représentation d'un signal vidéo. La compression peut être avec ou sans pertes selon le contexte d'utilisation. Dans le cas d'un codage avec pertes, il est nécessaire de réaliser un compromis entre la quantité de données, exprimée en débit, et la distorsion exprimée avec les métriques objectives. La Figure 2.12 présente une vue chronologique des différents standards de compression. Le processus de standardisation permet d'unifier les outils de codage et de standardiser la sortie des codeurs vidéo permettant à n'importe quel décodeur supportant le standard de décoder le bitstream correctement. Il s'agit donc d'un gage assurant une compatibilité entre codeurs et décodeurs à travers le monde.

En 1984, le Comité Consultatif International Télégraphique et Téléphonique (CCITT) publie le premier standard de compression vidéo numérique : le H.120 [26]. La première version du H.120 inclut des techniques de compression tels que la quantification scalaire et le codage entropique à longueur variable. En 1988, la deuxième version ajoute des outils de prédiction et de compensation de mouvement. En 1993, le CCITT a fusionné avec l'ITU et plus spécifiquement avec sa branche appelée UIT-T dédié à la normalisation. Le groupe spécifiquement responsable du codage vidéo est communément appelé VCEG. En 1988, le standard H.261 [27], successeur du H.120, fut ratifié par VCEG. Ce standard fut le premier à être largement employé en tant que technologie de compression vidéo. La majorité des techniques de compression définie dans le standard H.261 est encore employée aujourd'hui

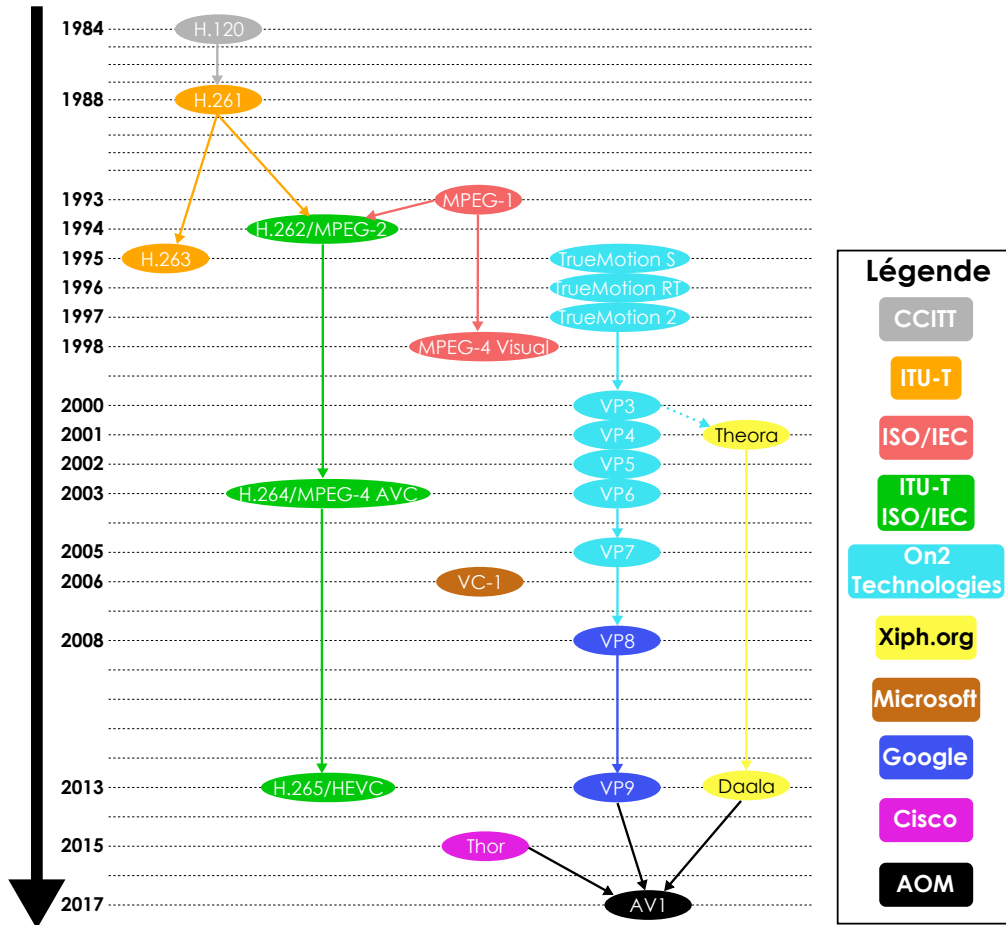


FIGURE 2.12 – Technologie et standard de compression vidéo de ces dernières années

dans les standards actuels. Parmi ces techniques on trouve notamment la compensation de mouvement par blocs de pixel, la DCT et le parcours des blocs de gauche à droite, puis de haut en bas (parcours en zig-zag). Toujours en 1988, le MPEG est créé par la collaboration des groupes ISO et IEC. Puis, en 1993, ce groupe ratifie le standard MPEG-1 basé lui aussi sur H.261 mais ajoutant des outils comme par exemple la bi-prédiction temporelle, l'estimation de mouvement demi-pixel ou encore le codage dans des structures en slices. Cette structure est présentée dans la Section 2.5.7.

En 1994, les groupes MPEG et VCEG proposent conjointement le standard H.262/MPEG-2. Ce standard hérite à la fois de H.261 et de MPEG-1. Il ajoute la notion de scalabilité dans le codage vidéo et l'entrelacement des images. Le standard MPEG-2 définit non seulement le codage vidéo dans H.262/MPEG-2 mais aussi des standards de conteneur et de codage audio. Le standard de conteneur définit le MPEG-2 "Transport Stream" (MPEG-2 TS) et le MPEG-2 "Program Stream" (PS). Le PS sert à multiplexer les codages vidéo et audio notamment. Le MPEG-2 TS sert à transmettre et stocker les PS et les codages audio et vidéo. Le standard MPEG-2 est notamment employé dans les DVD, la diffusion de la télévision par câble et par satellite. Les images sont divisées en blocs carrés de taille 16×16 appelés macroblocs. Ce standard propose également une extension scalable avec des scalabilités en résolution, en niveau de distorsion SNR et en fréquence.

En 1995 l'ITU propose, seul cette fois, le standard H.263 qui introduit notamment les types d'images P et B et le codage arithmétique entropique. De même, en 1998, MPEG propose le standard MPEG-4 Visual qui permet un codage sur des images de profondeur 10 et 12 bits par pixel et sur des contenus en image de synthèse. La création du groupe "Joint Video Team" (JVT) en 2001 rassemble les experts de MPEG et de VCEG dans le but de proposer un standard successeur à H.262/MPEG-2. En 2003, le standard H.264/MPEG-4 AVC est ratifié, par rapport au standard H.262/MPEG-2, il réduit de 50% le coût en débit pour un même niveau de qualité. Parmi toutes les fonctionnalités

et améliorations qu'il propose, on trouve :

- un filtre anti-blocs qui réduit les effets d'artefacts suite à la quantification des coefficients transformés DCT,
- un assouplissement des restrictions sur l'utilisation des références temporelles permettant une nette amélioration sur les vidéos à fort mouvement,
- des codeurs entropiques appliqués au bitstream,
- la définition des "Network Abstraction Layer Unit" (NALU) et des SPS et PPS permettant d'unifier la syntaxe des vidéos entre différents réseaux,
- la répartition des blocs en slices,
- ...

Le standard H.264/MPEG-4 AVC est plus simple d'utilisation dans les systèmes de réseaux comparés à ses prédecesseurs. C'est un des standards les plus utilisés dans la compression de vidéo au format HD. Les blocs de traitement sont de taille 16×16 pixels et sont appelés *macroblocks*. De plus, il est employé dans la diffusion de la télévision HD, le stockage sur HD DVD et Blu-ray ainsi que pour du streaming sur Internet "Real-time Transport Protocol" (RTP-IP). Une extension scalable à ce standard, appelée SVC a été définie en 2007. Cette extension a largement été employée pour des applications de vidéo conférences pour lesquelles le contenu doit être codé en temps-réel. L'extension SVC définit, comme dans l'extension scalable de H.262/MPEG-2, trois types de scalabilité : temporelle, spatiale et SNR. Il est également possible d'effectuer une combinaison de ces différents types de scalabilité. Malheureusement, cette extension n'a pas rencontré beaucoup de succès du fait de sa sortie tardive comparée au standard H.264/MPEG-4 AVC. De plus, la complexité de la prédiction est augmentée du fait de la prédiction inter-couches. En effet, celle-ci est non seulement différente des techniques de prédiction déjà définies dans le standard H.264/MPEG-4 AVC, mais doit aussi être signalée au niveau des blocs.

En 2010 le groupe JCT-VC rassemblant des membres de MPEG et de VCEG travaille sur un nouveau standard qui sera ratifié en 2013 sous le nom de HEVC. Ce standard sert de base aux travaux de cette thèse et est davantage détaillé dans la Section 2.5.

Parallèlement, d'autres groupes proposèrent également des standards de codage vidéo. En 1995, une société privée nommée On2 Technology propose le standard TrueMotion S qui s'adresse notamment aux vidéos 3D (i.e. hauteur \times largeur \times profondeur de champ). Deux améliorations appelées TrueMotion RT et TrueMotion 2 voient le jour respectivement en 1996 et 1997. En 2000, On2 Technology propose un standard de codage pour des scènes plus naturelles appelé VP3. Ce standard devient libre de droit permettant ainsi à l'organisation Xiph de proposer leur standard Theora en 2001. En 2001, 2002, 2005 et 2008, On2 Technology propose respectivement les standards VP4, VP5, VP7 et VP8. Le standard VP8 [28] est annoncé avec une efficacité de codage comparable au standard H.264/MPEG-4 AVC [29]. Il utilise une répartition des pixels pour la luminance et les chrominances respectivement dans des blocs de taille 16×16 et 8×8 . Ces blocs sont ensuite à nouveau divisés en blocs 4×4 pour les opérations de quantification et de transformée. La transformée de VP8 est basée sur une opération de DCT suivi d'une opération de transformée appelée Transformée de Walsh-Hadamard ("Walsh-Hadamard Transform") (WHT). La prédiction inter se base sur 3 images de référence :

- La première référence est l'image précédente dans l'ordre de codage.
- La deuxième est une image plus éloignée temporellement dans le passé appelée "golden frame". L'intérêt de cette image est de conserver des éléments qui bougent peu comme l'arrière-plan par exemple.
- La dernière référence est une image construite à partir de plusieurs sources appelées "alternate reference frame". Cette image n'est pas nécessairement affichée sur un écran puisque qu'elle peut soit être construite à partir de plusieurs images ou bien à partir d'un sous-ensemble de bloc issu d'un grand nombre d'images. Cela permet notamment d'utiliser des informations de codage issues des images futures dans l'ordre de codage.

En Février 2002, Google acquiert On2 Technology et rend tous ses standards libres de droit au sein du projet WebM. Ce projet cherche à proposer un format multimédia d'un conteneur ouvert principalement à destination d'un usage sur le web. En 2006, Microsoft propose le standard VC-1 qui devient un sérieux concurrent au H.264/MPEG-4 AVC de part son utilisation au sein des HD-DVD et des disques Blu-ray. En 2013, Google propose le standard VP9 qui apporte différentes améliorations au standard VP8. La division en blocs de pixels est étendue dans une structure appelée super-blocs à des rectangles de tailles 64×64 , 64×32 , 32×64 , 32×32 , 32×16 , 16×32 , 16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 et 4×4 . La prédiction spatiale dispose de 10 modes dont 6 angulaires. La prédiction temporelle se voit ajouter 2 nouveaux modes permettant soit d'utiliser un vecteur de mouvement nul, soit de définir un nouveau vecteur de mouvement. De même que dans VP8, les trois mêmes types d'images de référence peuvent être employés pour la prédiction temporelle. Le VP9 propose 3 types de transformée : une DCT, une DST

asymétrique et une WHT qui peuvent être combinées suivant le mode de prédiction. Afin de prévenir les artefacts des transformées, un filtre anti-bloc a aussi été ajouté. De plus, VP9 autorise la répartition des super-blocs dans des segments ayant des paramètres de codage différents. De larges régions divisant l'image en zones rectangulaires, appelées Tiles, peuvent être traitées indépendamment afin d'effectuer un parallélisme intra-image. Le VP9 intègre également la possibilité de réaliser des codages scalables sur les résolutions, les fréquences ou leur combinaison. Pour cela, il utilise la référence "alternate reference frame" qui permet de créer une référence à partir d'une autre image. Afin de réaliser une scalabilité spatiale, cette référence est sous-échantillonnée, employée en tant que référence temporelle pour les plus hautes résolutions puis signalée dans le bitstream comme expliqué dans [30]. A notre connaissance, il n'existe pas d'étude cherchant à exploiter et évaluer la scalabilité de ce standard.

En 2013, également, l'association de Mozilla Foundation et Xiph.org propose le standard Daala. Ce standard a pour objectif d'être gratuit et open-source. Il propose notamment des outils jusqu'alors peu exploités dans le domaine de la compression vidéo. Ces outils ont pour but de réduire les inconvénients de certaines opérations de filtrages et notamment le filtrage anti-bloc tout en augmentant peu la complexité. Pour ce faire, ils proposent de réaliser un filtrage anti-bloc simplifié en sortie de la transformée ainsi que l'opération inverse de ce filtrage avant la transformée. Le filtrage anti-bloc est une opération coûteuse à la fois en temps de traitement, mais aussi en signalisation et donc en débit. L'intérêt de la proposition est donc de réduire ces coûts avec des opérations de filtrage plus simples et qui nécessitent moins de signalisation. Les détails de cette proposition sont expliqués dans [31].

En 2015, Cisco propose le standard Thor en tant qu'alternative open-source à HEVC. Un des intérêts de proposer un tel standard pour une entreprise est de disposer d'une alternative à HEVC qui n'impose pas un paiement des licences. Le standard Thor se veut modérément complexe afin de faciliter son implémentation temps-réel dans des systèmes Software et Hardware. Il propose des outils équivalents aux standards H.264/MPEG-4 AVC et HEVC mais en quantité moins importante. Par exemple, il propose 10 modes de prédiction intra et 4 modes de prédiction inter. Le détail des outils proposés dans ce standard sont expliqués dans [32].

Enfin en 2017, le groupe "Alliance for Open Media" (AOM), composé entre autres des compagnies Amazon, Cisco, Google au travers du projet WebM, Intel Corporation, Microsoft, Mozilla Foundation, Netflix et Ateame, annonce son prochain standard AV1 qui prévoit une meilleure efficacité de codage que le standard HEVC. Ce nouveau standard s'appuie notamment sur le standard VP9 ainsi que des améliorations issues des standards Daala et Thor. Le standard AV1 est aujourd'hui en fin de développement, une première version du standard est prévue pour fin janvier 2018. Des discussions concernant un successeur, AV2, ont déjà commencé au sein du groupe AOM.

2.5 Le standard HEVC

Le standard de compression vidéo HEVC a été défini par le groupe JCT-VC. Sa première version a été finalisée en janvier 2013. Il propose des outils de codage permettant notamment de réduire le coût en débit de 50% par rapport au standard H.264/MPEG-4 AVC à même niveau de qualité.

2.5.1 Répartition en blocs de pixels

La répartition en blocs de pixel s'effectue dans des régions carrées de taille variable appelées "Coding Tree Unit" (CTU). Tout comme l'image est constituée de 3 composantes de couleur, une CTU est constituée de 3 "Coding Tree Block" (CTB). La taille $L \times L$ de la CTB de luminance peut être choisie par l'encodeur parmi $L = 16, 32$ ou 64 . Toutes les images codées sont divisées en séquences de CTU appelées Slices.

Ces slices disposent d'un entête qui permet de les coder/décoder indépendamment. Une image est alors constituée d'une ou de plusieurs slices dans lesquelles les CTU sont ordonnées dans l'ordre de lecture. Il en existe 3 types : les I-slices, P-slices et B-slices qui se définissent par rapport à leur mode de prédiction. Les slices de types I utilisent exclusivement une prédiction spatiale alors que les slices de types P et B peuvent utiliser des prédictions temporelles et spatiales. La prédiction temporelle s'effectue avec des listes de référence, les slices P et B disposent respectivement d'une et deux listes de référence. La deuxième liste de référence des B-slices permet d'effectuer une bi-prédiction.

Les CTU et les CTB, tels que définis dans le standard HEVC, peuvent être respectivement divisés en "Coding Units" (CU) et en "Coding Block" (CB). Cette sous-répartition des CTU en CU est illustrée sur la Figure 2.14 avec leur parcours standardisé de gauche à droite puis de haut en bas. Les CU servent de base aux traitements de prédiction, de quantification et de transformée pour lesquelles s'opère une sous-division en "Prediction Unit" (PU)

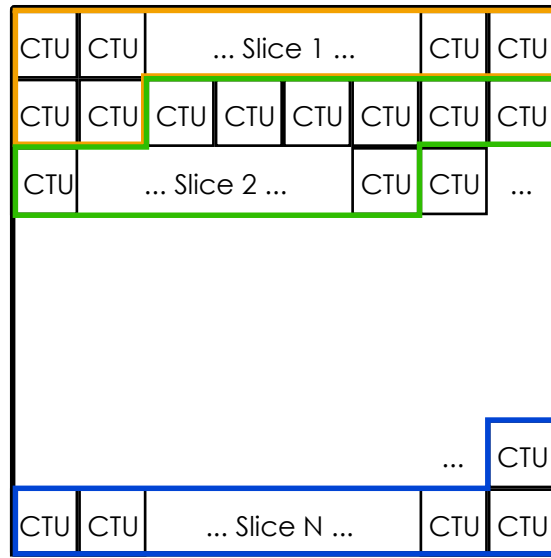


FIGURE 2.13 – Une image réparties en slices HEVC

et "Transform Unit" (TU). De la même façon que pour les CU et CB, les "Prediction Block" (PB) et les "Transform Block" (TB) sont les blocs par composante respectivement pour les PU et TU.

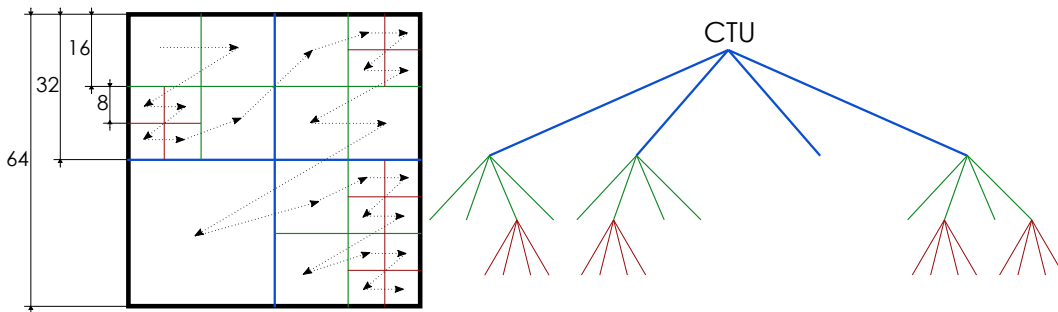


FIGURE 2.14 – Répartition arborescente des Coding Units HEVC

La décision du mode de prédiction est réalisée à l'échelle des CU parmi les modes définis dans les prédictions intra et inter. Chaque mode dispose d'une partition de la CU en PU comme illustré sur la Figure 2.15.

La sous-division des CU en TU est plus simple : elle reprend la répartition arborescente des CTU en CU. On parle alors de "Residual Quadtree Structure" (RQT). Cet arborescence des TU limite la taille maximale et minimale des blocs respectivement à 32x32 et 4x4 pixels.

2.5.2 Prédiction

La prédiction peut exploiter les corrélations spatiales ou temporelles respectivement avec les prédictions intra et inter. La prédiction intra de HEVC définit 35 modes : planar, DC et 33 modes directionnels. La prédiction intra est réalisée sur des PB de taille allant de 4x4 à 32x32 pixels. Afin d'alléger les calculs pour la recherche du meilleur mode intra, le standard HEVC définit des filtres permettant de pré-identifier les meilleurs modes. La prédiction intra est alors réalisée en extrapolant les valeurs des échantillons des blocs de référence.

Le mode DC consiste à prédire le PB en calculant la valeur moyenne des échantillons de référence. Le mode planar consiste à calculer la moyenne d'une prédiction horizontale et d'une prédiction verticale. Les modes directionnels

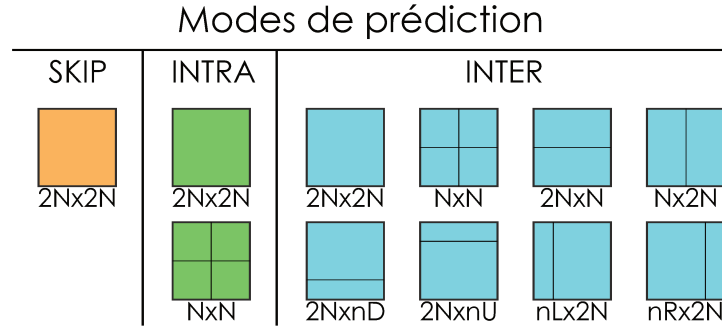


FIGURE 2.15 – Partition des Codings Units en Prediction Units

TABLE 2.2 – Coefficients de l'interpolation des échantillons fractionnels pour la luminance (Y) et les chrominances (U et V).

Indice i		-3	-2	-1	0	1	2	3	4
Y	hFilter[i]	-1	4	-11	40	40	-11	4	1
	qFilter[i]	-1	4	-10	58	17	-5	1	N/A
U et V	filter1[i]	N/A	N/A	-2	58	10	-2	N/A	N/A
	filter2[i]	N/A	N/A	-4	54	16	-4	N/A	N/A
	filter3[i]	N/A	N/A	-6	46	28	-4	N/A	N/A
	filter4[i]	N/A	N/A	-4	36	36	-4	N/A	N/A

consistent à réaliser une interpolation de la ou des directions qui ont été identifiées.

La prédiction inter utilise les échantillons des images reconstruites précédemment codées. Pour cela, ces images de référence sont identifiées par un paramètre appelé indice de référence. Les blocs de référence associés à la prédiction sont localisés avec des vecteurs de mouvement exprimant la différence entre la position du bloc de référence et la position du bloc courant. Si les coordonnées du vecteur sont entières, alors le bloc est directement déduit depuis la position courante. En revanche si les coordonnées sont fractionnelles, l'image de référence doit alors être ré-échantillonnée avec des filtres 4, 7 et 8 coefficients respectivement pour le filtrage sur les chrominances, sur la luminance en quart de pixel et sur la luminance en demi-pixel. Ce filtrage permet alors d'atteindre une précision dite au quart de pixel sur la prédiction inter. Les coefficients du filtre sont présentés dans le Tableau 2.2.

Les blocs de référence, aussi appelés blocs prédicteurs, sont signalés en indiquant les deux coordonnées du vecteur de mouvement ainsi que l'indice de référence respectivement sous la forme $(\Delta x_i, \Delta y_i, \Delta t_i)$ avec i indiquant la liste employée. Cette liste est utile pour identifier s'il s'agit d'une uni-prédiction (P-slices) ou bien d'une bi-prédiction (B-slices). Le standard HEVC propose trois modes de prédiction inter : Prédiction des Vecteurs de Mouvement Avancée ("Advanced Motion Vector Prediction") (AMVP), Merge et Skip.

Le mode AMVP permet de mettre en compétition plusieurs références possibles, alors appelés candidats. Ces candidats sont des blocs alentours à la position équivalente du bloc courant dans l'image de référence. Comme illustré sur la Figure 2.16, les candidats sont au nombre de 6 et identifiés par 5 candidats alentours A0, A1, B0, B1, B2 et un candidat temporel aussi appelé collocalisé. Ces candidats sont associés à des vecteurs de mouvement appelés prédicteurs ou "Motion Vector Predictor" (MVP). Ces MVP sont utilisés tels quels ou bien affinés à partir des précédents vecteurs de mouvement employés dans la prédiction et stockés dans une liste de prédicteurs. Cette liste présente à la fois l'intérêt de mettre en compétition plusieurs prédicteurs, mais aussi de pouvoir les identifier plus simplement sans nécessiter plus de signalisation. Seul le vecteur de mouvement différentiel, ou "Motion Vector Difference" (MVD), entre la position du bloc de référence et le prédicteur est signalé :

$$MVD_x = \Delta x - MVP_x \quad (2.5)$$

$$MVD_y = \Delta y - MVP_y \quad (2.6)$$

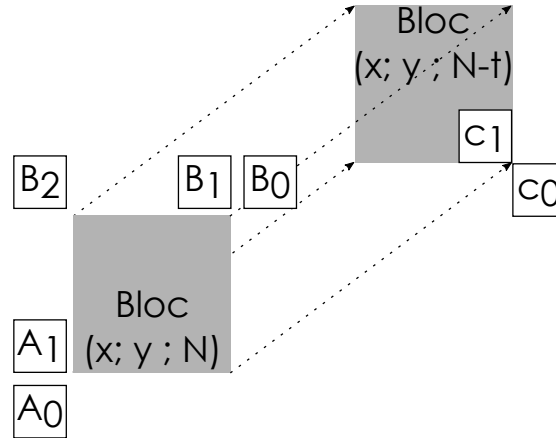


FIGURE 2.16 – Blocs candidats dans la prédiction inter HEVC.

En plus du mode AMVP, le standard HEVC définit le mode Merge. Ce mode permet de réduire la redondance dans la signalisation des vecteurs de mouvement. En effet, le mode Merge utilise le même système de choix de candidat que le mode AMVP mais signale que la prédiction du bloc courant sera effectuée directement à partir des valeurs des vecteurs de mouvement des prédicteurs. Lors de l'identification des candidats possibles, le mode Merge permet d'en ajouter de nouveau en combinant les prédicteurs à d'autres références puis en ajoutant des vecteurs nuls pour les références dans l'ordre de proximité temporelle. De cette manière, de nouveaux candidats peuvent être générés.

Enfin, un prédicteur utilisant le mode Merge peut être signalé en Skip qui requiert moins d'information si les contraintes suivantes sont respectées :

- la partition des PU est $2N \times 2N$,
- le mode de prédiction est le mode Merge,
- il n'y a pas de résidus signalés dans le bitstream.

2.5.3 Transformée et Quantification

L'opération de transformée est réalisée sur les TB avec des tailles admissibles de 4×4 , 8×8 , 16×16 et 32×32 pixels. Le standard HEVC définit une opération de DCT afin de calculer les coefficients résiduels à partir de la différence entre l'image source et l'image prédite. Dans le cas d'une prédiction intra sur des blocs de tailles 4×4 , l'opération de DCT est remplacée par une opération de DST. Une fois les coefficients obtenus, une opération de quantification contrôlée par un Paramètre de Quantification ("Quantization Parameter") (QP) leur est appliquée. Ce QP peut prendre des valeurs entières comprises entre 0 et 51 comme dans le standard H.264/MPEG-4 AVC.

2.5.4 Codage entropique

Le codage entropique du standard HEVC est réalisé en utilisant une fonction de Codeur Arithmétique à Contexte Adaptatif Binaire ("Context-Adaptive Binary Arithmetic Coding") (CABAC). Les éléments de syntaxe sont tout d'abord convertis sous forme de symbole binaire qui peuvent être de 5 types différents. Ces symboles sont ensuite évalués et mesurés statistiquement afin d'associer une probabilité de présence à chaque symbole binaire. Enfin, chaque symbole binaire est compressé sur une quantité de donnée qui est calculée en fonction de sa probabilité. Une prédiction produisant peu de résidu et peu de signalisation permettra de générer une plus faible quantité de données. C'est pourquoi la prédiction intra, qui produit davantage de données, requiert davantage de temps de codage entropique par rapport à une prédiction inter.

2.5.5 Filtrages Anti-blocs et filtrages SAO

Dans le but d'améliorer la qualité des images reconstruites, HEVC définit deux filtres : Anti-blocs et Décalage Adaptatif des Échantillons ("Sample Adaptive Offset") (SAO). Le filtre Anti-bloc est issu de H.264/MPEG-4 AVC avec toutefois des améliorations permettant un parallélisme de traitement. Le filtre SAO est, quant à lui, défini pour la première fois dans le standard HEVC. Ces deux filtres peuvent être activés pour un même codage. Le filtre SAO classe les échantillons de l'image reconstruite dans deux catégories. Cette distinction permet d'appliquer un décalage positif ou négatif plus ou moins important sur les échantillons des frontières au sein ou en bordure de l'image. Les paramètres de décalage sont ensuite signalés dans le bitstream au niveau des CTU. Les filtres Anti-blocs et SAO permettent donc de réduire les effets de blocs sur l'image reconstruite.

2.5.6 Syntaxes haut-niveau

Le standard HEVC définit un nouveau type d'entête de bitstream constitué d'un ensemble de paramètres caractérisant la vidéo : le Set de Paramètre pour la Vidéo ("Video Parameter Set") (VPS). De plus, l'entête est également constitué d'un Set de Paramètre pour la Séquence ("Sequence Parameter Set") (SPS) et d'un Set de Paramètre pour l'image ("Picture Parameter Set") (PPS) qui étaient déjà définis dans H.264/MPEG-4 AVC. L'ensemble des VPS, SPS et PPS constitue les "parameters set". Les éléments de syntaxe haut-niveau sont contenus dans des structures appelées NALU qui sont des paquets de données mis en forme de façon pratique pour la diffusion, le stockage, le streaming... Il existe 31 types de NALU dont 3 sont les parameters set.

Les entêtes de slice font partie des éléments de la syntaxe haut niveau, ils sont transmis juste avant les informations de codage par bloc. De plus, il est possible d'ajouter des informations pouvant améliorer, par exemple, le codage ou l'affichage en utilisant des messages de type SEI. Ces messages SEI peuvent être utilisés en préfixe ou en suffixe de la slice concernée. De plus, certains messages SEI peuvent servir à plusieurs slices et peuvent être désactivés en envoyant un autre message SEI correspondant. La Figure 2.17 présente le schéma d'un bitstream HEVC.

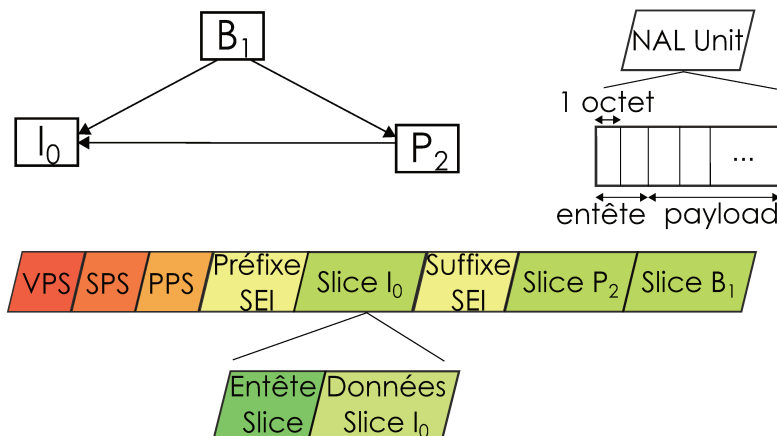


FIGURE 2.17 – Schéma de la constitution d'un bitstream HEVC

2.5.7 Outils de parallélisme

Le standard HEVC définit également des outils permettant d'améliorer les vitesses de traitement du parcours des blocs comme illustré sur la Figure 2.18. Les slices, tels qu'ils étaient définis dans les précédents standards de compression, permettaient déjà de réaliser un parallélisme sur le traitement des blocs. Toutefois, malgré les opérations de filtrage, cette répartition en slices pouvait provoquer des distorsions sur leurs frontières. Le standard HEVC propose deux outils supplémentaires permettant un parallélisme : les Tiles et le parcours "Wavefront Parallel Processing" (WPP).

Les Tiles sont des régions rectangulaires au sein d'une image qui peuvent être considérées comme indépendantes vis à vis de la prédiction et de la transformée. Chaque Tiles peut alors être traitée en parallèle des autres avec

une signalisation dans les PPS. Toutefois, les Tiles réduisent la corrélation spatiale et requiert une ré-initialisation du CABAC à chaque nouvelle Tile. L'utilisation des Tiles permet donc d'améliorer significativement la vitesse de codage pour une éventuelle perte en terme d'efficacité de codage ce qui constitue un compromis entre complexité et distorsion. Les filtrages anti-blocs et SAO ne peuvent être employés au sein d'une slice ou d'une Tile ce qui requiert une gestion particulière pour ces opérations.

Le WPP définit un parcours des blocs autorisant un parallélisme sur le lignes de CTU. Une ligne de CTU peut être traitée en parallèle des autres une fois que la ligne qui la précède à terminer le traitement sur ces deux premiers CTU. Ce mécanisme n'a pas d'influence sur la prédiction ni sur l'efficacité de codage mais en revanche impacte le CABAC qui doit initialiser un contexte à chaque début de ligne. En pratique, le standard HEVC n'autorise pas l'utilisation combinée des Tiles et du WPP même si ces deux méthodes peuvent se combiner en théorie.

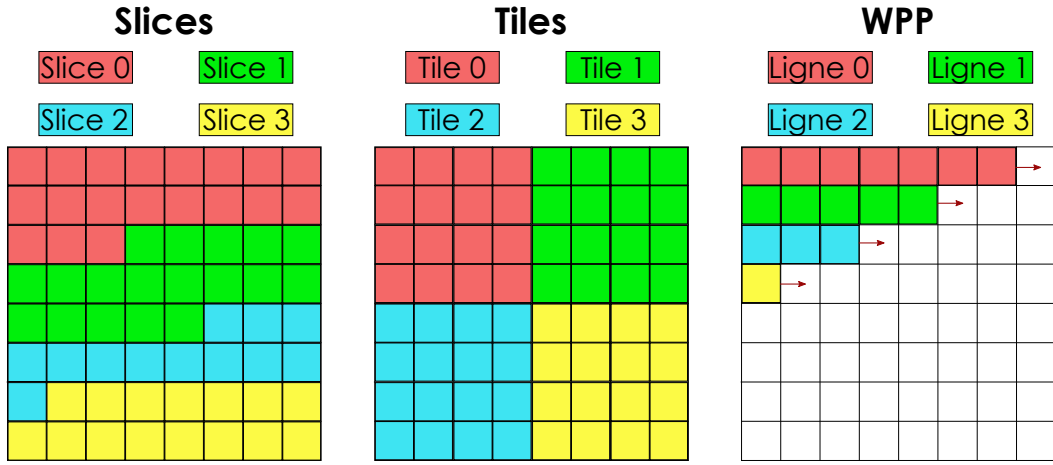


FIGURE 2.18 – Illustration des parallélismes sur le parcours des blocs définis dans HEVC.

2.5.8 Structure de GOP et configuration de codage

Une structure de Groupe d'images ("Group Of Pictures") (GOP) permet d'attribuer un rôle à certaines images. En effet, certaines images peuvent servir exclusivement de référence pour que d'autres puissent être prédites. Pour cela, les types d'images (comme les types de slices) suivant sont définis :

- les images I qui n'utilisent que la prédiction intra,
- les images P qui peuvent utiliser les prédictions intra et inter,
- les images B qui peuvent utiliser les prédictions intra, inter et la bi-prédiction.

Les images I servent de référence aux autres images et permettent un rafraichissement en cas de perte de données à la transmission. Les images P et B peuvent toutes deux utiliser une référence temporelle. Les images P stockent ces images de référence dans une liste tandis que les images B peuvent les stocker dans deux listes. Ce sont ces deux listes qui permettent aux images B de réaliser une bi-prédiction. Dans ce cas, les références de chacune des listes sont pondérées par des poids annoncés dans l'entête de PPS et signalés dans l'entête de slice.

Les images I, P et B sont agencées dans une structure de GOP. Les structures de GOP couramment employées sont les configurations :

- images I seulement all-intra (ou intra-only) (AI) illustré sur la Figure 2.19,
- faible latence (low-delay)(LD) illustré sur la Figure 2.20 avec la prédiction temporelle réalisée sur des images de type P ou B,
- random-access (RA) illustré sur la Figure 2.21.

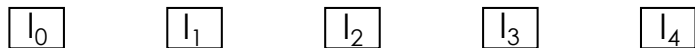


FIGURE 2.19 – Schéma de la configuration AI.

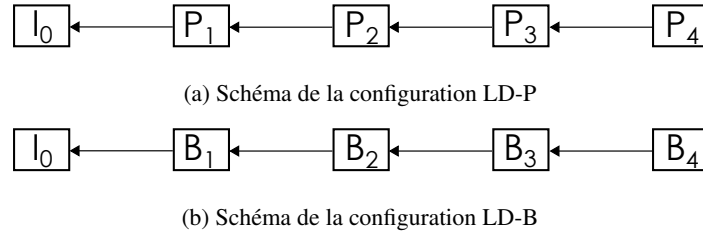


FIGURE 2.20 – Schéma des configurations LD.

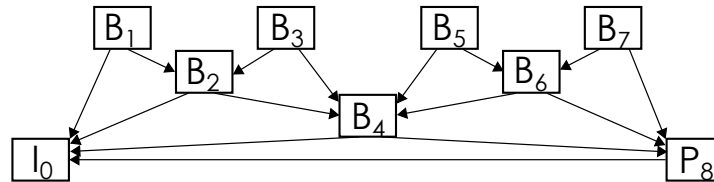


FIGURE 2.21 – Schéma de la configuration random-access

La configuration AI n'emploie que des images de type I. Les images sont alors traitées dans l'ordre d'affichage. Cette configuration nécessite une quantité de débit importante mais permet d'avoir un taux de rafraîchissement très rapide. Elle diminue ainsi les risques de dégradations du flux après transmission.

La configuration LD emploie seulement deux types d'images, I et P ou I et B. Elle traite les images dans l'ordre d'affichage. Dans le cas d'utilisation des images I et P on appellera la configuration low-delay P (LD-P). On appellera low-delay B (LD-B) la configuration LD employant les images I et B. Elle permet de produire un bitstream rapidement (faible latence) tout en réduisant la consommation de débit par comparaison à la configuration AI. Néanmoins, le coût en débit reste assez élevé. Les applications qui ont recours à cette configuration servent généralement à la communication comme la vidéo conférence par exemple.

La configuration RA peut employer tous les types d'images. Contrairement aux autres configurations, elle permet aux images B d'employer une bi-prédiction. Le coût en débit est donc bien moins important par comparaison aux configurations AI, LD-P et LD-B. La configuration RA, telle que proposée dans les "Common Test Condition" (CTC)[33], définit des périodes pour chaque type d'image :

- une seconde pour les images I,
- huit images pour les image P,
- le reste du bitstream pour les images B.

Ces images sont alors réparties hiérarchiquement. Ainsi, comme on peut l'observer sur la Figure 2.21, une image sur deux ne sert pas de référence temporelle. La configuration RA est largement employée dans la diffusion de contenu vidéo.

2.5.9 Profile, tiers et level

Les profils, les tiers et les levels spécifient des points de conformité afin d'assurer une inter-opérabilité entre les applications utilisant le standard. Les profils définissent un ensemble d'outils de codage ou d'algorithmes susceptibles d'être employés. Les niveaux fixent des contraintes sur certains paramètres du bitstream. Les tiers sont des sous-divisions des niveaux correspondant à des cas généraux ou spécifiques. Les profils, tiers et levels concernent plus généralement le décodeur puisqu'ils définissent une conformité du bitstream. Le codeur, quant à lui, doit juste s'assurer de fournir un bitstream conforme au standard.

Le standard HEVC définit 3 profils "*Main*", "*Main 10*" et "*Main Still Picture*" dans sa première version en 2013. Dans la deuxième version sortie en 2014, 21 profils d'extension sur les échantillonnages des couleurs, 2 profils scalables et 1 profil multi-vue sont ajoutés. En 2015, la troisième version ajoute le profil 3D *Main*. Enfin la quatrième version sortie en 2016 ajoute 7 profils d'affichage sur écran et 4 profils scalables.

2.6 Codage Scalable et SHVC

2.6.1 Codage scalable

Principe du codage scalable

Le codage scalable consiste à coder la même source à différents niveaux de qualité et de ne former qu'un unique flux. Les niveaux de qualité considérés en scalabilité regroupent entre autres la résolution, le rapport SNR et la fréquence d'image. On parle alors respectivement de scalabilité spatiale, SNR et temporelle. Chaque niveau de qualité définit une couche de codage généralement ordonnée de la couche du niveau de qualité le plus bas vers le plus haut.

La couche de base (i.e. "base layer", BL) est la couche de niveau de qualité le plus bas, elle sert de référence aux couches supérieures. Les couches supérieures, aussi appelées couches de rehaussement (i.e. "enhancement layers", EL), sont des couches qui peuvent utiliser des informations de codage des couches inférieures. Ces couches inférieures regroupent la BL et les EL d'un niveau de qualité moindre. La BL est donc une couche de référence exclusivement alors que l'EL du plus haut niveau de qualité est une couche exclusivement prédite et les EL intermédiaires sont à la fois des couches de référence et des couches prédites. L'intérêt du codage scalable réside dans cette prédiction inter-couches. En effet, elle permet aux couches de niveau de qualité les plus élevés d'améliorer significativement leur efficacité de codage par rapport à un codage simulcast (contraction de "simultaneous broadcast"), on parle alors de gains inter-couches. Un codage simulcast consiste à coder une même vidéo source à plusieurs niveaux de qualité. Contrairement au codage scalable, chaque niveau de qualité constitue un codage indépendant des autres.

Précédentes extensions scalables

Les standards précédents et contemporains au HEVC ont également défini des moyens permettant un codage scalable. Dans le cas du standard VP9, comme expliqué dans [30], la couche de base est construite à partir des images de référence de type "alternate reference frame". Ces images peuvent être sous-échantillonnées, codées avec un débit ou une qualité moindre ou employées pour seulement certaines images afin de réaliser des scalabilités respectivement de type spatiale, SNR et temporelle. A notre connaissance, il n'existe pas d'étude cherchant à exploiter et évaluer la scalabilité sur le standard VP9.

Le standard H.262/MPEG-2 intègre un mécanisme permettant de réaliser une scalabilité sur les résolutions, niveaux de qualité SNR et fréquences en s'appuyant sur la granularité de la quantification des coefficients transformés. Les coefficients sur la couche de base sont d'abord quantifiés avec une première quantification qui conserve peu de détails. Ces coefficients quantifiés sont transmis au codeur à taille variable (VLC) défini dans le standard. Parallèlement à ce codage, une copie de ces coefficients sont dé-quantifiés avant d'être de nouveau quantifiés avec, cette fois-ci, une conservation plus fine des détails. Cette quantification correspond à la couche de rehaussement et permet une scalabilité SNR. Pour la scalabilité spatiale, le même processus est employé avec l'ajout d'une compensation de mouvement sur une image sous-échantillonnée sur la couche de base. Une transformée de type DCT de taille réduite lui est appliquée suivie de la quantification et du codage VLC. Enfin, la scalabilité temporelle consiste à ajouter un module de démultiplexage temporel. Ce module transmet une moitié des images à la couche de base et l'autre moitié à la couche de rehaussement. La couche de rehaussement peut employer les images reconstruites de la couche de base dans sa compensation de mouvement. La syntaxe des deux codages séparés est ensuite multiplexée avant d'être transmise. Du côté du décodeur, un démultiplexeur équivalent se charge d'abord d'identifier les couches de base et de rehaussement avant de les décoder séparément.

Le standard H.264/MPEG-4 AVC (2003) définit une extension scalable appelée "Scalable Video Coding" (SVC) (2007) 4 ans après sa standardisation. Elle permet une scalabilité sur les résolutions, niveaux de qualité SNR et fréquences. Pour cela elle définit un codage par couche avec différentes caractéristiques de l'image pour chacune des couches. Les blocs de la couche de rehaussement peuvent alors utiliser une référence temporelle issue de la couche de base dans les mécanismes de prédiction et de transformée. Ainsi, un nouveau type de macroblock spécifique à la prédiction inter-couches est ajouté. De plus, une prédiction inter-couches sur les résidus peut également être effectuée quelque soit le type de macroblock employé dans la prédiction intra ou inter.

Toutes ces extensions ont pour point communs de définir des outils de prédiction spécifique à la prédiction inter-couches. L'extension scalable du standard HEVC n'entre pas dans cette catégorie.

TABLE 2.3 – Les différents types de scalabilité supportés par le standard SHVC.

Type de scalabilité	Exemple de BL	Exemple d'EL
Temporelle	25 fps	50 fps
Spatiale	1920×1080 (HD)	3840×2160 (UHD)
SNR	30 dB	26 dB
Codec	H.264/AVC	HEVC
Profondeur des pixels	8 bits par pixels	10 bits par pixels
Espace de couleur	BT.709	BT.2020

2.6.2 Le standard SHVC

Standardisation

L'extension scalable du standard SHVC a été publiée dans la deuxième extension du standard en octobre 2014 dans l'annexe H. Cette extension propose notamment deux profils scalables : "Scalable Main" et "Scalable Main 10". Le développement de SHVC aura duré 2 ans entre octobre 2012 et juin 2014. Cette période est relativement courte si on la compare à l'extension scalable du standard H.264/MPEG-4 AVC qui a duré 4 ans (entre 2003 et 2007). De plus, l'extension SHVC propose de nouveaux types de scalabilité.

Les différents types de scalabilité

L'extension SHVC propose une scalabilité sur des niveaux de qualité "classique" : résolution, fréquence et qualité SNR mais aussi sur des niveaux de qualité moins souvent proposés : profondeur des pixels, codec, espace de couleur. La Table 2.3 présente ces différents types de scalabilité supportés par SHVC avec quelques exemples. En ce qui concerne la scalabilité temporelle, celle-ci était supportée dans la première version du standard HEVC.

La prédiction inter-couche

Le codage SHVC se veut le moins intrusif possible vis-à-vis du codage HEVC. Pour cela, un codeur SHVC emploie les mêmes mécanismes de prédiction pour chacune de ses couches. Quand une couche est prédite, ses images utilisent, en plus des informations de codage dont elles disposent déjà, des informations provenant d'une image de référence issues d'une couche plus basse. L'image de référence inter-couches est repérée de la même manière que les images de référence HEVC : par un indice présent dans les listes de référence. Cet indice est alors, généralement, le dernier élément de la liste. Cela permet d'employer les mêmes modes de prédiction que dans HEVC et de ne pas ajouter de modifications à la prédiction des blocs. La Figure 2.22 illustre cet ajout de référence dans les listes L0 et L1. La position de la référence inter-couches dans une liste peut être différente mais il est alors nécessaire de la signaler en modifiant l'intégralité de la liste concernée. Les informations de codage apportées par ce prédicteur sont notamment la texture de l'image reconstruite et les vecteurs de mouvement. L'information de texture est employée avec un vecteur de mouvement nul, quelque soit le mode de prédiction choisi (merge, skip ou amvp). Les vecteurs de mouvement issus de la couche de référence sont utilisés en tant que vecteurs collocalisés et ne peuvent être employés que pour des blocs prédits temporellement. Par conséquent, les images I, P et B peuvent toutes fournir des informations de texture mais seuls les images P et B peuvent fournir des vecteurs de mouvement.

Du point de vue de la signalisation, la prédiction par bloc est la même qu'en HEVC. L'image de référence inter-couches est signalée en tant que référence de longue durée afin que les vecteurs de mouvement ne soient pas modifiés dans le cadre de la mise à l'échelle en fonction de la distance temporelle. La syntaxe des entêtes subit quelques modifications dans une suite d'éléments, appelé extension, qui ne sont pas pris en compte dans un codage HEVC permettant ainsi une rétro-compatibilité avec la première version du standard. Le VPS reste commun à toutes les couches et ajoute une description de la relation entre chacune des couches, le type de prédiction inter-couches (texture et/ou vecteur de mouvement) et le type de scalabilité. Les autres éléments de syntaxe haut niveau (SPS, PPS, entête de slice...) sont propres à chaque couche et repérés par un identifiant de couche : `nuh_layer_id`. L'utilisation de la prédiction inter-couches est signalée dans l'entête de slice dans un champ à part des autres références de prédiction temporelle.

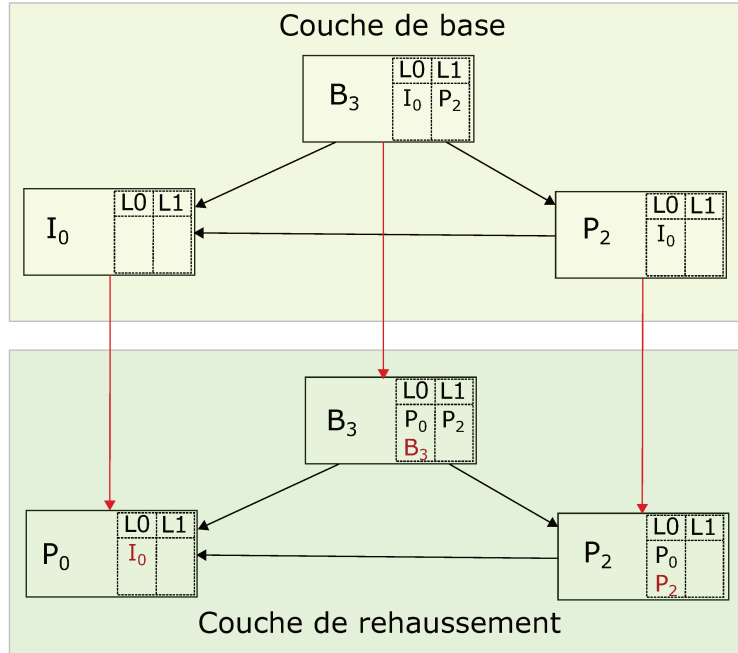


FIGURE 2.22 – Exemple de GOP avec prédiction inter-couche

Dans le cas de certains types de scalabilité, en résolution et sur les espaces de couleur notamment, il peut être nécessaire d'adapter le format des données de prédiction à la couche prédite. Pour cela, des opérations de filtrage, sur-échantillonnage et conversion d'espace, sont définies dans le standard SHVC.

Sur-échantillonnage des textures pour la scalabilité spatiale

Le filtre de sur-échantillonnage défini dans l'extension scalable du standard HEVC permet d'adapter les échantillons des images reconstruites de résolution basse vers une résolution plus élevée. Cela permet, entre autre, d'utiliser les images reconstruites des couches de référence dans la prédiction inter-couches en scalabilité spatiale. L'opération de sur-échantillonnage est une opération normative puisqu'elle est définie dans l'extension du standard et doit être rigoureusement la même entre un codeur et un décodeur. Cette opération emploie, pour la composante de luminance, des filtres à Réponse Impulsionnelle Finie (filtres FIR) [34] sur 16 phases et 8 coefficients tandis que les chrominances emploient 16 phases et 4 coefficients. Les filtres FIR donnent des réponses impulsionnelles stables et de durée finie qui dépend du nombre de coefficients du filtre. Ils effectuent un algorithme d'interpolation d'ordre N avec la formule suivante :

$$s[n] = \sum_{i=-\lfloor N/2 \rfloor - 1}^{\lfloor N/2 \rfloor} c_i \times p[n+i] \quad (2.7)$$

Les coefficients du filtre de sur-échantillonnage définis dans l'extension SHVC sont décrits dans la Table 2.4. Ces coefficients correspondent à ceux du filtre de la compensation de mouvement défini dans la première version du standard. Toutefois, le sur-échantillonnage est défini avec une précision d'un seizième de pixels alors que la compensation de mouvement est définie avec une précision d'un quart et d'un huitième de pixel respectivement sur la luminance et les chrominances. C'est pourquoi 6 lignes de coefficient ont été ajoutées aux filtres de compensation de mouvement.

Le filtre de sur-échantillonnage permet un accroissement des résolutions des images avec un ratio supérieur à 1 pour la largeur d'une part et la hauteur d'autre part. Il est, par conséquent, appliqué une première fois dans le sens horizontal puis une seconde fois dans le sens vertical. Le choix de la phase à appliquer est le même calcul que pour l'interpolation dans la compensation de mouvement. Par exemple, pour un ratio $2 \times$, il faut employer les coefficients des phases 0 et 8 alternativement pour chaque ligne de pixel de l'image de destination. De plus, il est

TABLE 2.4 – Coefficients du filtre de sur-échantillonnage pour les composantes de luminance (Luma) et de chrominances (Chroma)

Phase p	Luma $f_L[p,i]$								Chroma $f_L[p,k]$			
	i=0	i=1	i=2	i=3	i=4	i=5	i=6	i=7	k=0	k=1	k=2	k=3
0	0	0	0	64	0	0	0	0	0	64	0	0
1	0	1	-3	63	4	-2	1	0	-2	62	4	0
2	-1	2	-5	62	8	-3	1	0	-2	58	10	-2
3	-1	3	-8	60	13	-4	1	0	-4	56	14	-2
4	-1	4	-10	58	17	-5	1	0	-4	54	16	-2
5	-1	4	-11	52	26	-8	3	-1	-6	52	20	-2
6	-1	3	-9	47	31	-10	4	-1	-6	46	28	-4
7	-1	4	-11	45	34	-10	4	-1	-4	42	36	-4
8	-1	4	-11	40	40	-11	4	-1	-4	36	36	-4
9	-1	4	-10	34	45	-11	4	-1	-4	30	42	-4
10	-1	4	-10	31	47	-9	3	-1	-4	28	46	-6
11	-1	3	-8	26	52	-11	4	-1	-2	20	52	-6
12	0	1	-5	17	58	-10	4	-1	-2	16	54	-4
13	0	1	-4	13	60	-8	3	-1	-2	14	56	-4
14	0	1	-3	8	62	-5	2	-1	-2	10	58	-2
15	0	1	-2	4	63	-3	1	0	-2	4	62	-2

possible de n'appliquer ce filtre que sur une fenêtre réduite de l'image source grâce à un paramètre de décalage qui doit être signalé dans le PPS.

Sur-échantillonnage des vecteurs de mouvement pour la scalabilité spatiale

Dans l'extension SHVC, la prédiction inter-couches peut aussi bien s'appliquer aux vecteurs de mouvement. Cette information de codage supplémentaire permet d'améliorer davantage l'efficacité de codage des couches prédites. Les vecteurs de mouvement de la couche de référence peuvent être employés dans la prédiction temporelle en tant que vecteurs collocalisés. Ces vecteurs sont conservés et utilisés sous forme de bloc de taille 16×16 pixels. La première étape du traitement consiste à déterminer la position du bloc dans la couche de référence à partir de la position courante dans la couche prédite. Le calcul de cette position prend en paramètres :

- la position du bloc dans la couche de référence,
- le ratio entre les résolutions (largeur et hauteur séparément),
- les éventuels décalages.

Une fois cette position obtenue, le vecteur considéré est alors mis à l'échelle de la couche prédite avec le ratio entre les résolutions. Ce traitement de mise à l'échelle permet d'utiliser les vecteurs de mouvement issus de la couche de référence sans modifier le fonctionnement de la prédiction employant les vecteurs collocalisés.

Cette opération de prédiction est optionnelle et peut être désactivée dans la syntaxe. De plus, cette opération ne peut en aucun cas être effectuée dans le cas où la scalabilité de codec est employée.

Conversion des espaces de couleurs pour la scalabilité colorimétrique

Dans le cas d'une scalabilité sur les espaces de couleur (ou colorimétrique), la conversion d'un espace à un autre est réalisée via une table de conversion 3D ("look-up table", LUT). La table de conversion permet de mettre en relation la position d'une couleur depuis un espace de couleur vers un autre espace. La conversion s'applique à l'image reconstruite employée dans la prédiction inter-couches. Dans le cas où la scalabilité entre deux couches est à la fois spatiale et sur les espaces de couleurs, la conversion des espaces de couleur est réalisée avant le sur-échantillonnage parce qu'elle représente un temps de traitement plus important à même résolution.

Le standard SHVC définit un cube selon les axes des composantes de luminance et des deux chrominances. La composante de luminance est partitionnée en un maximum de 8 régions uniformes tandis que les chrominances sont partitionnées en un maximum de 2 régions chacune non-uniforme, comme illustré sur la Figure 2.23. Pour chaque

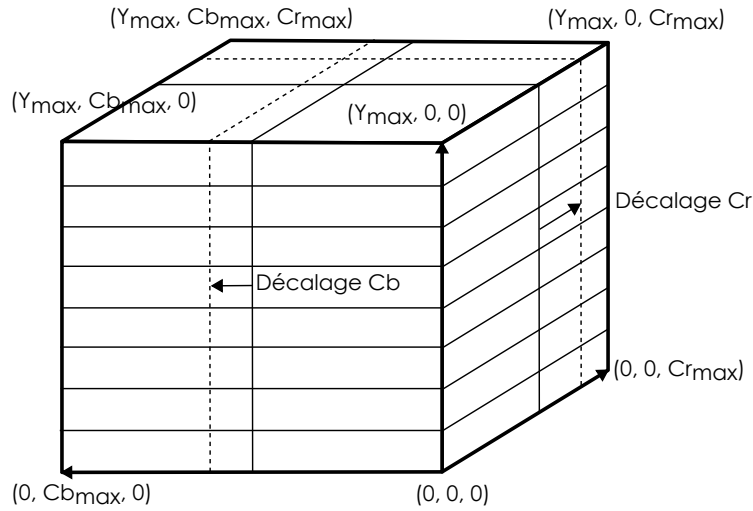


FIGURE 2.23 – Illustration d'une LUT-3D 8×2×2

partition, la conversion d'un espace de couleur à un autre est réalisée par l'équation suivante :

$$\begin{bmatrix} Y_{EL} \\ U_{EL} \\ V_{EL} \end{bmatrix} = \begin{bmatrix} a_Y & b_Y & c_Y \\ a_U & b_U & c_U \\ a_V & b_V & c_V \end{bmatrix} * \begin{bmatrix} Y_{BL} \\ U_{BL} \\ V_{BL} \end{bmatrix} + \begin{bmatrix} o_Y \\ o_U \\ o_V \end{bmatrix}$$

(Y_{EL}, U_{EL}, V_{EL}) sont les échantillons de la couche de rehaussement, (Y_{RL}, U_{RL}, V_{RL}) sont les échantillons de la couche de référence, a_i, b_i, c_i et o_i avec $i = Y, U$ ou V sont les coefficients et les décalages de la conversion d'espace de couleur. Pour chaque partitions, ces coefficients et décalages sont signalés dans le PPS.

Dans l'extension SHVC, la complexité du codage est augmentée par l'ajout des différentes opérations de conversion des informations de codage :

- sur-échantillonnage des textures,
- sur-échantillonnage des vecteurs de mouvement,
- conversion des espaces de couleur,
- utilisation de ces informations de codage dans la prédiction inter-couches.

Cette complexité permet, via la prédiction inter-couches, d'améliorer l'efficacité de codage de la couche de rehaussement et par conséquent sa qualité à débit constant. Cette amélioration de la qualité peut être rehaussée grâce à une régulation du débit bénéficiant soit à la couche de base, soit à la couche de rehaussement, soit à un équilibre entre les deux couches. Cette régulation du débit constitue un mécanisme complexe mais permet de tirer un meilleur profit des outils de codage dans des configurations à débit constant.

2.7 Régulation de débit et améliorations appliquées au standard SHVC

Afin d'améliorer la qualité vidéo à débit constant, les différents outils de codage doivent être évalués scrupuleusement afin d'identifier le meilleur compromis entre coût en débit et fidélité vis à vis de la source. Dans le cas du codage à débit constant, celui ci est généralement attribué à un ensemble d'image (une image, un GOP ou la séquence entière) ce qui peut influencer le choix des outils de codage sur une durée plus ou moins longue. En effet, si l'intégralité du débit est attribué aux premières images, les dernières seront de moins bonne qualité. Inversement, si une trop grande quantité du débit n'est pas attribuée aux premières images, la qualité des images suivantes sera visuellement supérieure. Dans les deux cas, une transition entre bonne et mauvaise qualité d'image prendra effet ce qui constitue une gêne visuelle qui doit être évitée. Pour cela, des mécanismes permettent une régulation du débit afin d'assurer une cohérence en terme de qualité entre les images.

2.7.1 Principe de la régulation de débit

La régulation de débit est un mécanisme intervenant entre la sélection des outils (ou décision) de codage et la transmission des données. D'un côté la décision permet de mesurer le coût des différents outils de compression (prédiction, transformation, ...) et de l'autre la transmission des données doit être réalisée de sorte que l'intégralité des données puissent être transmises. Les contraintes imposées à la régulation de débit dépendent de l'application et peuvent être par exemple :

- un objectif sur le débit,
- une contrainte sur l'occupation mémoire d'un outils de transfert,
- appliquées spécifiquement à une granularité de la vidéo (GOP, image, CTU),
- ou encore un niveau de qualité spécifique mesuré avec des métriques objectives ou psychovisuelles (voir Section 2.2).

Un algorithme de régulation de débit est généralement composé de 3 étapes supervisant un codeur comme illustré sur la Figure 2.24. La première étape consiste à allouer une certaine quantité de données à la granularité ciblée en fonction des différentes contraintes de la régulation de débit. Cette étape peut être optimisée, par exemple, en identifiant avant le codage les zones les plus complexes d'une image afin d'y attribuer plus de données. Une fois qu'une quantité de données est allouée à l'image, la deuxième étape consiste à estimer un modèle permettant de contrôler certains paramètres de codage comme par exemple le QP ou le lagrangien λ . Enfin, la troisième étape consiste à sélectionner ces paramètres qui serviront à définir un coût dans le codage.

De plus, puisque le comportement d'un codage est difficile à prévoir, la régulation du débit doit analyser ses propres décisions. A la fin du codage à la granularité (i.e. taille de bloc) concernée, des informations sont alors remontées vers les 3 étapes afin d'affiner le comportement futur et de corriger les éventuelles mauvaises estimations. Ces informations peuvent être, par exemple, le débit réellement alloué au codage, la qualité atteinte, des statistiques sur la prédiction...

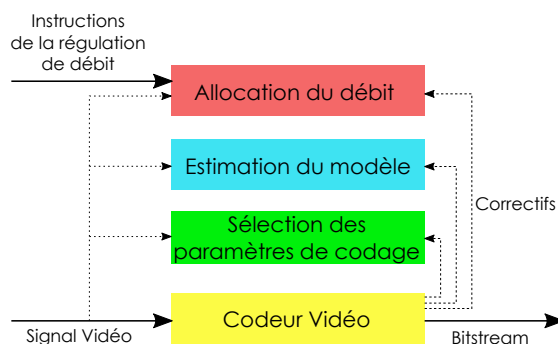


FIGURE 2.24 – Étapes de la régulation de débit

Dans le cas d'une contrainte de débit constant (i.e "Constant Bit-Rate" (CBR)), par exemple, la transmission des données est réalisée sur le principe du "sceau percé" (leak bucket) qui consiste à recevoir une quantité variables de données et d'en émettre une quantité constante. Dans le cas d'un codeur HEVC, cette transmission est réalisée par un buffer appelé Buffer Virtuel de Vérification ("Virtual Buffer Verifier") (VBV). Les données traitées sont les informations de codage à transmettre. La principale contrainte du VBV réside dans le fait que ces données sont inégales en taille selon leur position dans le GOP. Il est alors nécessaire de s'assurer que les informations de codage ne dépassent pas un certain seuil selon leur position dans le GOP et selon les contraintes de débit et de vitesse de codage.

Le rôle de la régulation de débit est donc d'assurer que les informations de codage ne dépassent pas ce seuil fixé préalablement. Pour cela, des métriques sont employées dans la décision de codage afin de mesurer le coût de l'utilisation des différents outils de compression. La régulation de débit peut intervenir à différentes granularités comme par exemple au niveau du GOP, de l'image, d'une région d'intérêt ou d'un bloc de pixel.

Parmi les différents modèles existants, la régulation de débit peut être effectuée dans les domaines R-Q, R- ρ et R- λ . Le domaine R-Q consiste à modéliser le débit "R" en fonction du QP. Le principal inconvénient de ce domaine est qu'il peut être difficile à modéliser et, par conséquent, introduire de la complexité de traitement. Le domaine R- ρ consiste à modéliser une relation linéaire entre le débit R et les coefficients résiduels non nuls. Ce domaine

n'est plus vraiment adapté aux standards de compression récents pour deux raisons essentielles :

- ils emploient généralement de longues entêtes qui ne sont pas prise en compte dans le domaine R- ρ ,
- les améliorations sur les méthodes de prédiction réduisent significativement la quantité de coefficients résiduels rendant la régulation moins efficace.

Enfin, le domaine R- λ met en relation le débit avec le Lagrangien λ . Ce paramètre est tout d'abord estimé puis sert à mettre à jour le QP avec la formule suivante :

$$QP = 4.2005 \times \ln(\lambda) + 13.7122 \quad (2.8)$$

Ce modèle R- λ est employé en grande majorité par les codeurs actuels.

En ce qui concerne la régulation du débit pour les prédictions inter-couches, celle ci a pu être étudiée au cours de précédents travaux portant sur le codage SHVC.

2.7.2 Travaux existants pour SHVC

L'extension SHVC présente un intérêt concret pour la diffusion et le stockage de contenus vidéo. C'est pourquoi plusieurs travaux proposent des améliorations de l'efficacité de codage ou des méthodes destinées à répondre à des contraintes pour des applications comme la vidéo surveillance, l'analyse médicale ou la diffusion.

Les auteurs de [35] décrivent le fonctionnement d'un codage SHVC et qualifient sa prédiction inter-couches. Cette prédiction pour des scalabilités spatiale et SNR permet en moyenne d'économiser respectivement 26.1% et 31.5% du débit à qualité constante en configuration #glsra.

Dans [36], les auteurs proposent une régulation de débit au niveau des GOP en se basant sur la complexité par images dans des codages SHVC. Cette complexité est évaluée à partir de la différence moyenne absolue prédite, de l'occupation mémoire et du niveau hiérarchique. La régulation du débit est ensuite effectuée dans le domaine R- λ .

[37] propose un algorithme de régulation de débit basé sur le domaine R- λ avec un paramétrage dédié pour les différents type de scalabilité.

[38] propose un mécanisme de régulation du débit pour des codages SHVC en scalabilité spatiale et SNR en 3 étapes :

1. d'abord, une estimation du coût en débit de la première image sur les couches de base et de rehaussement avec un compromis entre distorsion et débit,
2. ensuite, une estimation du débit par image pour les prédictions intra et inter-couches,
3. enfin, une adaptation de l'algorithme de régulation du débit dans le domaine R- λ proposé par les mêmes auteurs dans des travaux précédents dédiés au codage HEVC [39].

[40] propose une réduction de la complexité de la prédiction. Cette réduction s'appuie sur la définition d'un parcours de recherche des modes de prédiction (directions intra, merge inter, skip inter et amvp inter) pour chaque CU. Une première étude analysant les meilleurs modes de prédiction sur certains types de CU pour plusieurs séquences a servi à identifier un mode de prédiction optimal pour certaines catégories de CU. Chaque CU est classée dans une catégorie en fonction de ses prédicteurs potentiels qui dépendent du type d'image, de la position du bloc, de sa taille et de sa partition. Selon la probabilité du bloc d'appartenir à une des catégories ciblées, un chemin de prédiction est alors suivi par ce bloc. Cette méthode montre une amélioration de la vitesse de codage de 59.66% pour une perte d'efficacité de codage de 2.16% en BD-BR par rapport au codage ne l'employant pas.

[41] propose une comparaison du codage SHVC pour des compressions sans pertes. Cette étude cible des applications de type médicale, de vidéo-surveillance et de réalité augmentée pour lesquelles le codage sans perte peut être nécessaire. Elle propose une méthode d'un codage autorisant les pertes sur la couche de base et sans perte sur la couche de rehaussement. Cette dernière bénéficie de la prédiction inter-couches mais ne produit qu'une partie des images de la séquence source tandis que la couche de base code l'ensemble des images.

[42] propose un méthode combinant le codage SHVC et le transport "Layered Division Multiplexing" (LDM) pour une diffusion de type ATSC 3.0 avec une scalabilité SNR sur deux couches. Le transport LDM permet de transmettre plusieurs services à travers un unique canal de diffusion. Ces services correspondent généralement à différents formats (résolution, fréquence, profondeur...) ou paramètres de codage (qualité SNR, débit, codec...) d'un même contenu vidéo. Cette combinaison permet de profiter de la prédiction inter-couches et de la robustesse du LDM afin d'assurer un transport fiable des contenus vidéo. Parallèlement, [43] propose une étude sur cette même méthode de transport avec une scalabilité sur 3 couches spatiales permettant de réduire le coût de la bande-passante de 23% à 43% par rapport au codage simulcast.

[44] propose une amélioration du décodeur openHEVC, présenté dans la partie 2.8.4, afin de permettre un décodage sur des flux employant plusieurs codec. Le décodeur effectue un traitement sur des flux avec une syntaxe SHVC dont les NALU des couches de base et de rehaussement sont codées respectivement à partir des standards de compression H.264/MPEG-4 AVC et HEVC.

Toutes ces études montrent l'intérêt du codage SHVC au travers de différentes applications. Il existe certains outils permettant d'obtenir des codages SHVC mais aucun, avant cette thèse, ne permettait de coder à la fois rapide et efficace en terme de compression.

2.8 Codeurs et décodeurs exploitant SHVC

2.8.1 SHM

Le "Scalable HEVC Model" (SHM) est un logiciel de référence du standard SHVC. Au même titre que la version non scalable nommée "HEVC Model" (HM), celui-ci propose une solution de codage et de décodage. La principale raison d'être de ce logiciel est de fournir une solution permettant de tester les différents outils de codage du standard SHM sans objectif d'amélioration des performances de vitesse ou de complexité. Le logiciel SHM emploie de multiples instances de HM exécutées séquentiellement en tant que couche scalable comme illustrée en Figure 2.25.

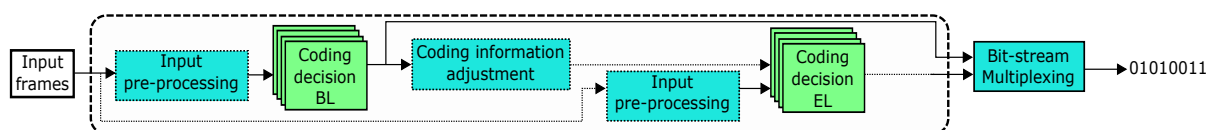


FIGURE 2.25 – Agencement séquentiel des couches scalables

Cette solution d'implémentation n'est pas la plus optimale puisqu'elle augmente non seulement la durée de sortie de la première image codée (latence) mais aussi la durée de traitement entre chaque image codée (cadence).

2.8.2 Kvazaar

Le codeur Kvazaar propose un codage HEVC pour des modes de prédiction intra essentiellement. Il est développé dans un projet académique open-source [45] au sein du groupe "Ultra Video Group" composé de membres de l'équipe *Pervasive Computing* de l'Université de Technologie de Tampere en Finlande. Il permet des codages HEVC dans des modes rapides ou efficaces en terme de compression. Les modes rapides permettent dans [46] et [47] des codages en temps réel respectivement sur des séquences $3840 \times 2160p$ et $1920 \times 1080p$ à 30 IPS. Dans [46], ces performances sont possibles grâce au portage de la prédiction intra sur FPGA. Toutefois, ce codeur n'emploie pas ou peu de prédiction inter ce qui limite son efficacité de codage dans des configurations LD-P, LD-B et RA. Enfin, le codeur Kvazaar propose un bitstream SHVC ne faisant pas ou peu intervenir une prédiction inter-couches. On notera alors Kvazaar-HEVC et Kvazaar-SHVC les codages obtenus avec le codeur Kvazaar respectivement avec les standards HEVC et SHVC.

2.8.3 x265

x265 [48] est une bibliothèque open-source développée notamment par MulticoreWare. La solution x265 supporte le codage HEVC mais pas son extension scalable. Il reste toutefois intéressant pour ses performances puisqu'il propose plusieurs modes qui permettent de réaliser des codages :

- fidèles à la source et cherchant la meilleure efficacité de codage possible,
- rapides en effectuant un compromis entre distorsion et vitesse au niveau des outils de codage.

En effet, l'objectif de x265 est de fournir la meilleure efficacité de codage ainsi que les meilleures performances de vitesse possibles sur de multiples plateformes. Pour cela, il propose des optimisations bas niveau qui peuvent être interprétées par plusieurs machines via plusieurs langages d'implémentation. De plus, il intègre un parallélisme sur les images avec un parcours des CTU basé sur le WPP permettant un parallélisme générique [49].

2.8.4 openHEVC

OpenHEVC propose une solution développée en open-source pour des décodages HEVC. Il est codé en langage C et exploite la librairie FFmpeg [50]. Le code source est largement optimisé en employant des langages faisant appel à des outils bas niveau. Le décodage est réalisé en traitant les lignes de manière indépendante avec un mécanisme de wavefront différent du WPP. Puis, sur chaque ligne, toutes les CTU sont traitées successivement. Le décodeur openHEVC optimise aussi sa gestion mémoire en utilisant une mémoire locale au niveau des CTU et une mémoire globale au niveau des images et de la vidéo plus généralement. Il permet également de décoder des bitstreams SHVC en temps-réel [51].

2.8.5 Outils d'analyse existants

Les outils d'analyse peuvent permettre d'analyser un bitstream pour plusieurs raisons. Ils peuvent, par exemple, servir à vérifier la conformité par rapport à la syntaxe du standard, ou bien permettre d'analyser les types de prédiction ou de transformée au niveau des blocs. Il existe des outils supportant le standard HEVC mais aucun supportant son extension scalable. Au cours de cette thèse, il a donc été nécessaire de créer un de ces outils afin de pouvoir vérifier plus efficacement et plus rapidement le codage SHVC. En utilisant un décodeur SHVC, les données de codage sont extraites et stockées dans un fichier externe. Ce fichier est ensuite analysé et ajoute une grille colorée sur la vidéo reconstruite au décodage, comme illustré sur la Figure 2.26. Chaque couleur est associée à un type de prédiction comme suit :

- Intra : Vert
- Skip inter-layer : orange
- Merge ou AMVP inter-layer : bleu clair
- Skip HEVC : rouge
- Merge ou AMVP HEVC : violet

De plus, des statistiques sur les types de prédiction peuvent être affichées en console. Le décodeur choisi pour le test est openHEVC puisque son code est open-source et qu'il propose un décodage très rapide même avec l'extension SHVC. L'analyseur ainsi créé permet donc d'analyser les données de prédictions des bitstreams SHVC et HEVC de façon minimaliste mais efficace.



FIGURE 2.26 – Schéma de l'analyseur SHVC.

Les différents outils d'analyse existants ou créés permettent de valider la conformité d'un flux SHVC, mais également d'analyser la réponse sous différentes contraintes. Ces contraintes définissent un contexte de codage ainsi qu'un paramétrage d'un codeur ou d'un décodeur. Elles peuvent être issues de recommandations d'un organisme de standardisation, comme c'est le cas des CTC [33], ou bien extraites de cas d'usage pratiques. Ces cas d'usage peuvent être liés au stockage ou à la diffusion par exemple.

2.9 Cas d'usage du codage scalable

Cette thèse s'inscrit dans un cadre industriel (CIFRE) en collaboration avec la société Ateame pour laquelle la diffusion de contenus vidéo constitue le cœur de métier. Cette diffusion peut être réalisée en suivant différentes recommandations issues des standards ou des protocoles de diffusion tels que l'ATSC 3.0, le "3rd Generation Partnership Project" (3GPP), le Vidéo Conférence Multimédia Multi-utilisateurs ("Multiparty Multimedia Video Conferencing") (MMVC), le Service Multimedia de diffusion large et multi-bande ("Multimedia Broadcast/Multicast Service") (MBMS) ou le Streaming 3GPP Dynamique et Adaptatif sur flux HTTP ("3GPP Dynamic Adaptive Streaming over HTTP") (3GP-DASH).

TABLE 2.5 – Conditions de tests pour les différents scénarios proposés par l'ATSC 3.0.

Scénario	Paramètre	PLP-1	PLP-2	PLP-3
A	Résolution	2160p	1080p	Audio/misc.
	Débit de codage (Mbps)	15.1	5.0	0.47
B	Résolution	1080p	720p	Audio/misc.
	Débit de codage (Mbps)	5.0	3.46	0.3
C	Résolution	1080p	540p	Audio/misc.
	Débit de codage (Mbps)	5.0	1.7	0.34
D	Résolution	2160p	1080p	Audio/misc.
	Débit de codage (Mbps)	4.5	2.75	0.3

2.9.1 ATSC 3.0

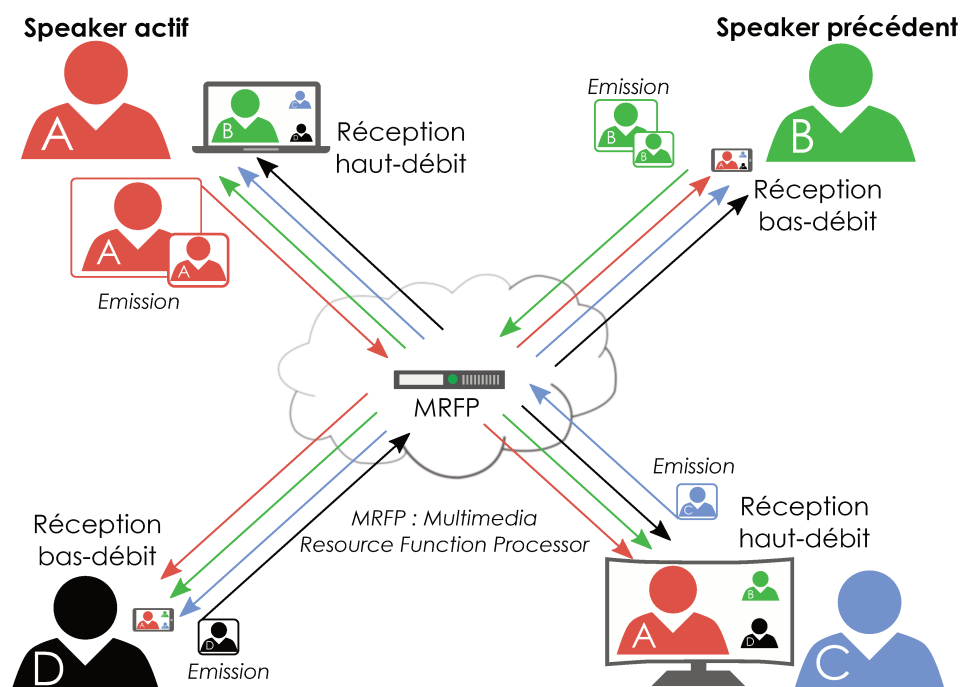


FIGURE 2.27 – Schéma du cas d'usage MMVC avec 4 utilisateurs. A est le "speaker" actif, B est le précédent, C et D écoutent. A envoie une capture en haute et moyenne résolution pour l'affichage principal des autres participants respectivement à haut et bas débit. B envoie une moyenne et une petite résolution qui seront respectivement l'affichage principal pour A et une vignette pour les autres. C et D envoient une petite résolution pour un affichage sous forme de vignette.

L'ATSC 3.0 est un standard de diffusion cherchant à répondre à l'accroissement de consommation de vidéo d'aujourd'hui. Les besoins identifiés sont notamment la possibilité de regarder des vidéos sur demande, quelque soit l'appareil employé et quelque soit le lieu. Les moyens d'accès aux contenus vidéo peuvent être effectués par câble, par liaison terrestre ou satellite, par Internet, par stockage local ou encore par une combinaison de ces moyens. L'ATSC 3.0 considère le standard SHVC comme un candidat pour sa solution de codage vidéo [52]. Le standard de compression SHVC a donc été évalué dans différentes applications possibles de l'ATSC 3.0 [53] définies à travers les scénarios suivants :

- Scénario A : Les récepteurs de première catégorie sont immobiles dans la zone couverte par la diffusion ATSC 1.0. Les récepteurs de seconde catégorie sont immobiles et ne sont pas présents dans cette zone.
- Scénario B : Les récepteurs de première catégorie sont mobiles à une vitesse de piéton. Les récepteurs de

- seconde catégorie sont immobiles.
- Scénario C : Les récepteurs de première catégorie se déplacent à une vitesse relativement rapide. Les récepteurs de seconde catégorie sont immobiles.
- Scénario D : Les récepteurs de première catégorie sont en intérieur et peuvent se déplacer. Les récepteurs de seconde catégorie sont en intérieur et immobiles.

Dans chaque scénario, un Canal de la Couche Physique ("Physical Layer Pipe") (PLP) est défini par couche scalable et un autre pour l'audio. Les conditions de tests sont présentées dans la Table 2.5. Les tests réalisés dans [53] ont permis de montrer un gain en débit de codage de 40% à 47% entre les codage SHVC et HEVC simulcast en configuration RA.

2.9.2 3GPP SA4

Le SA4 est le quatrième groupe de travail concernant les aspects système du 3GPP. Le 3GPP est une coopération entre organismes de standardisation en télécommunications qui produit et publie des spécifications techniques pour les réseaux mobiles de troisième et quatrième génération (3G et 4G). L'ITU en fait notamment partie. 3GPP SA4 a permis de définir la manière de gérer les différents services proposés par 3GPP. Parmi ces services, on trouve notamment le MMVC et la téléprésence, le MBMS et le 3GP-DASH.

MMVC

Le MMVC, illustré en Figure 2.27, considère une vidéo conférence avec plusieurs utilisateurs ayant différentes capacités de réception. Dans l'exemple, deux cas sont utilisés : une réception haut-débit et une réception bas-débit. L'utilisateur qui parle, ou "*speaker*" *actif*, occupe la majorité de l'écran des utilisateurs qui écoutent. Les utilisateurs qui écoutent sont affichés dans des vignettes à l'exception du "*speaker*" *précédent* qui occupe la majorité de l'écran du "*speaker*" actif. Dans l'exemple, le "*speaker*" actif dispose d'un accès haut-débit, il peut donc transmettre sa vidéo en haute-résolution et en moyenne résolution. La haute résolution est affichée pour les utilisateurs disposant d'un accès haut-débit tandis que c'est la basse résolution qui est affichée pour les utilisateurs ayant un accès bas-débit. Le "*speaker*" précédent dispose d'un accès bas-débit, par conséquent, il transmet sa vidéo en basse résolution pour les vignettes et en moyenne résolution pour l'affichage sur l'écran du "*speaker*" actif. Les autres utilisateurs transmettent leur vidéo en basse résolution pour les vignettes. Dans ce cas d'usage, les accès haut débit doivent pouvoir transmettre leur vidéo en haute, moyenne et basse résolution. De même, les accès bas-débit doivent pouvoir transmettre leur vidéo en moyenne et basse résolution. Le codage et décodage vidéo en scalabilité spatiale permet d'économiser de la bande passante. Des expérimentations conduites dans [54] ont montré que les utilisateurs transmettant plus d'une résolution ou recevant seulement la haute résolution percevaient respectivement un gain ou une perte de bande passante avec des codages SHVC par rapport aux codages HEVC simulcast.

MBMS

Le MBMS propose plusieurs services vidéos avec des résolutions différentes. La Figure 2.28 illustre ce cas avec deux services : *normal* et *premium*. Le service normal transmet une résolution moyenne tandis que le service premium transmet une haute résolution. Chaque équipement dispose d'un abonnement à l'un de ces services en fonction de ses capacités d'affichage, de décodage, du coût ou encore de son accès réseau. Chaque équipement est intégré à un réseau qui est relié à un centre de diffusion supportant le MBMS. Ce centre de diffusion récupère les vidéos sources chez un fournisseur de contenus.

Les différents services du MBMS peuvent correspondre aux différentes couches scalables en résolution d'un codage SHVC. Les équipements du service normal décodent la couche de base tandis que les équipements du service premium décodent les couches de base et de rehaussement afin de bénéficier de la prédiction inter-couches. Des expérimentations menées dans [54] et résumées dans [55] ont permis de montrer un gain moyen de 32.9% de bande-passante du codage SHVC comparé au codage simulcast HEVC en configuration RA avec des résolutions de 1280×720p et 1920×1080p respectivement pour les services normal et premium. Il n'y a pas de différence entre les codages SHVC et HEVC du service normal. En revanche, les codages du service premium ont à la fois une qualité vidéo et une complexité de décodage supérieure en SHVC par comparaison au HEVC simulcast.

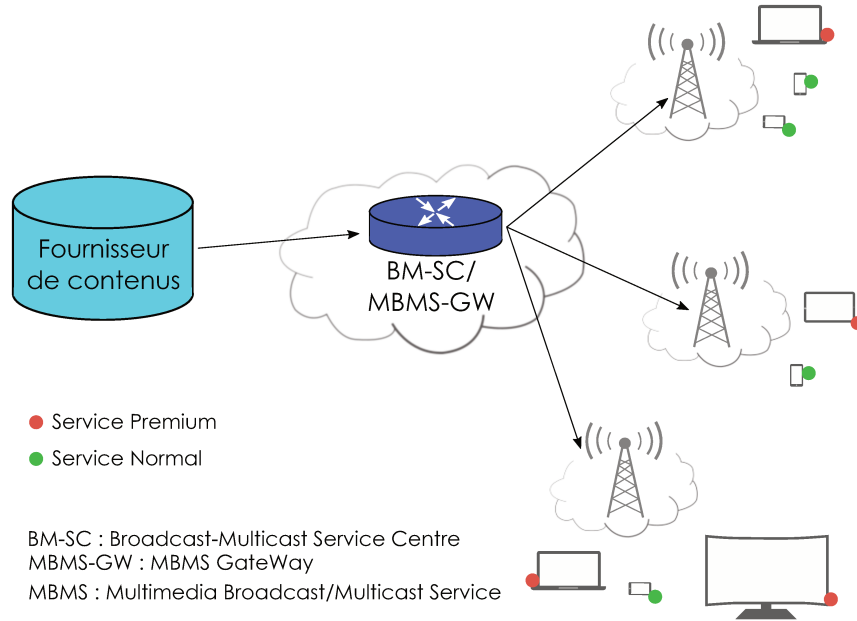


FIGURE 2.28 – Schéma du cas d'usage MBMS avec deux services.

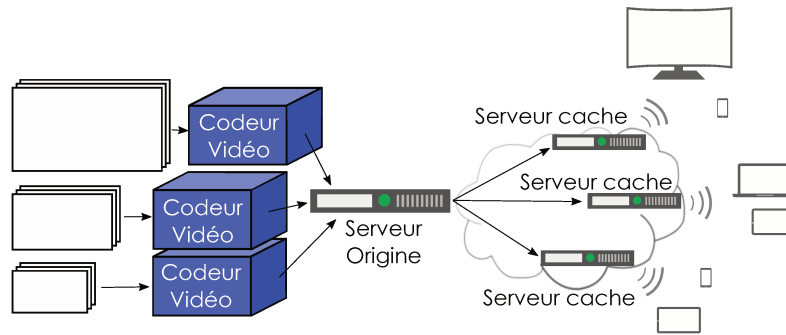


FIGURE 2.29 – Schéma du cas d'usage 3GP-DASH avec 3 résolutions et un niveau de qualité.

3GP-DASH

Le 3GP-DASH consiste à laisser le choix à l'utilisateur de la résolution ainsi que du niveau de qualité de la vidéo affichée. Ainsi, dans une telle application, une même vidéo est transmise sous plusieurs résolutions, chacune de ces résolutions étant codée avec différents niveaux de qualité. La Figure 2.29 schématise le service 3GP-DASH.

L'exemple évalué dans [54] considère 3 résolutions $640 \times 360p$, $1280 \times 720p$ et $1920 \times 1080p$ et un seul niveau de qualité. Les utilisateurs peuvent donc choisir entre ces 3 résolutions différentes. Avec le codage SHVC, les couches scalables correspondantes aux différentes résolutions demandées sont transmises avec leurs couches de référence si elles en ont. La comparaison entre codages SHVC et simulcast HEVC montre que la bande passante économise entre 9.2% et 10.5% et entre 23.3% et 23.6% du débit respectivement lorsque les résolutions $640 \times 360p$ - $1280 \times 720p$ et $640 \times 360p$ - $1280 \times 720p$ - $1920 \times 1080p$ sont transmises.

Toutes ces applications montrent l'intérêt d'un codage SHVC dans le monde de la diffusion et de la communication. Il existe encore d'autres types d'application dans différents domaines pouvant bénéficier du codage SHVC. Parmi eux, on trouve par exemple les domaines médicaux et de la surveillance.

Conclusion

Les notions abordées dans cette partie permettent d'évaluer la complexité du codage vidéo scalable. En effet, les données manipulées dans ce traitement peuvent être caractérisées sous plusieurs aspects :

- résolution,
- fréquence,
- SNR
- profondeur des pixels,
- espace de couleur.

Chacun d'eux peut alors constituer un type de scalabilité dans un codage SHVC. De plus, l'algorithme de compression vidéo SHVC, bien qu'efficace, présente une complexité de traitement non négligeable au travers des opérations suivantes :

- attribution d'un type I, P ou B à chacune des images,
- séparation des images en CU de taille variable,
- prédiction spatiale ou temporelle sur ces CU,
- conversion des informations de codage inter-couches,
- prédiction inter-couches,
- transformée et quantification sur les CU,
- codage entropique,
- reconstruction de l'image,
- construction de la syntaxe.

Comme nous avons pu le voir, le codage ainsi effectué présente un intérêt certain dans le cadre de la diffusion vidéo avec toutefois des contraintes en terme de débit, de format d'image et de vitesse de traitement. Au début de cette thèse, il n'existait pas de solution permettant de répondre à de telles contraintes. C'est pourquoi cette thèse propose une solution permettant de réaliser un codage SHVC en mesure de répondre à ces différentes contraintes.

Chapitre 3

La scalabilité spatiale appliquée à un codeur professionnel

Introduction

L'extension "Scalable High efficiency Video Coding" (SHVC) est une solution de codage intéressante permettant de répondre aux contraintes des dernières évolutions pour la production et la diffusion de vidéos. Elle s'intègre au standard "High Efficiency Video Coding" (HEVC) de façon moins intrusive que les extensions scalables précédentes. Les encodages SHVC sont réalisés dans différentes couches, chacune correspondant à un niveau de qualité qui peut s'exprimer la résolution, la qualité "Signal to Noise Ratio" (SNR), la profondeur des pixels, l'espace de couleur... Ces couches sont traitées les unes après les autres dans le codeur de référence de l'état de l'art : le "Scalable HEVC Model" (SHM) [56]. Ce chapitre propose une méthode permettant de traiter les différentes couches scalables simultanément pour une scalabilité en résolution (i.e. spatiale) sur deux couches. Le codeur réalisera un codage SHVC cherchant la meilleure efficacité de codage (ou FILE) avec une couche de base et une couche de rehaussement comme illustré en Figure 3.1.

La méthode proposée est intégrée dans un codeur professionnel développé par Ateme. Elle s'appuie sur l'architecture en pipeline de ce codeur et permet de bénéficier de ses différents atouts tel que le codage en temps-réel et une régulation du débit. L'architecture en pipeline permet d'aligner les différentes couches afin de réduire au maximum le délai entre chacune d'elles. Les couches scalables sont donc traitées simultanément avec un léger décalage à l'initialisation. Ce décalage provoque une augmentation de la latence de codage, c'est pourquoi cet écart doit être le plus court possible.

De plus, le codeur SHVC proposé réalise un compromis sur la prédiction inter-couches à l'échelle des images. Ce compromis, réalisé sur certains types d'image, permet de diminuer le surcoût en temps de traitement introduit principalement par l'ajout d'une couche scalable et notamment du filtre de sur-échantillonnage.

Tout d'abord, ce chapitre présente le codeur développé par Ateme et ses spécificités, notamment sur le pipeline de codage et sur les multiples niveaux de parallélisme qu'il emploie. Ensuite, l'architecture scalable proposée et les solutions d'alignement et de synchronisation des couches sont exposées. Enfin, les différents compromis au niveau des types de slice pour lesquels s'applique une prédiction inter-couches sont présentés et évalués sur différentes configurations de GOP. Les résultats présentés donnent les performances de la méthode proposée en terme d'efficacité de codage débit/distorsion et de vitesse de traitement.

3.1 Présentation de l'encodeur Ateme

L'encodeur professionnel utilisé lors des développements est un encodeur multi-supports développé par la société Ateme [57]. Cet encodeur appelé Single Layer Encoder (SLE) supporte notamment les encodages des différents standards de compression vidéo H.262/MPEG-2, H.264/MPEG-4 AVC, HEVC et AV1. Pour l'ensemble de ces standards, l'encodeur est en mesure de réaliser des encodages efficaces grâce à son architecture développée sous la forme d'un pipeline. De plus, il dispose de plusieurs modes de qualité permettant de réaliser au choix des encodages temps-réel, rapides ou cherchant la meilleure efficacité de codage possible. Ces modes de qualité réalisent

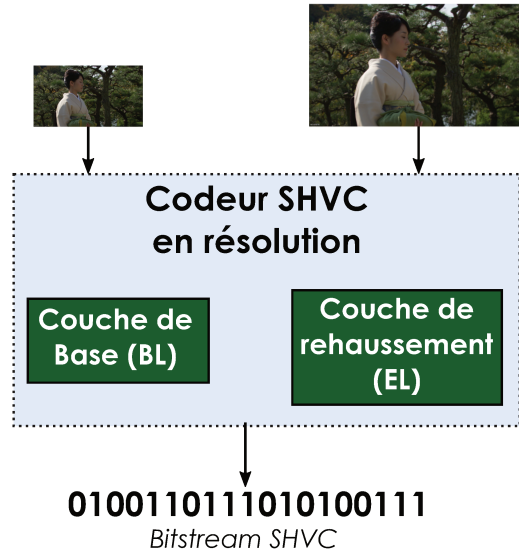


FIGURE 3.1 – Schéma du codeur SHVC FILE visé en scalabilité spatiale sur deux couches.

alors un premier compromis entre vitesse et efficacité de codage. L'architecture en pipeline permet d'intégrer un parallélisme entre les images. De plus, l'étape de décision de codage du SLE intègre de multiples parallélismes sur différents niveaux : intra-images, inter-blocs et inter-opérations. Enfin, le SLE dispose d'un module de régulation de débit permettant, par exemple, de réaliser des codages à débit constant.

3.1.1 Architecture et parallélisme

L'architecture du codeur SLE se présente sous la forme d'un pipeline d'étapes de codage illustré sur la Figure 3.2. Les différentes étapes de pipeline illustrées sur cette figure constituent les éléments clés du codage du SLE. Ces étapes détaillées dans cette même section sont :

- l'acquisition,
- le pré-traitement sur les images non compressées,
- l'attribution du "Picture Order Count" (POC),
- la pré-estimation du mouvement,
- la pré-estimation du débit et son attente de 5 secondes d'images,
- la décision de codage.

L'étape d'acquisition consiste à lire les images dans l'ordre d'affichage, de les conserver en mémoire et de les intégrer dans le pipeline. Les images alors acquises sont, au besoin, converties au format d'images planaires YUV 4:2:0 ou 4:2:2 selon la configuration de codage employée. Cette éventuelle conversion est alors réalisée au sein de l'étape de pré-traitement. Cette étape permet alors de réaliser une ou plusieurs opérations de pré-traitement appliquées aux images non compressées. Ces opérations peuvent servir :

- à affecter la qualité du codage : débruitage, estimation du flou de mouvement...
- à adapter la source au format de codage souhaité : ré-échantillonnage des composantes, entrelacement et dés-entrelacement des images...
- à détecter ou ajouter des informations sur l'image : logo, bandes noires...

La plupart des opérations de pré-traitement réalisent des filtrages sur une partie ou sur l'ensemble des échantillons de l'image. Elles peuvent nécessiter un temps de traitement important. Pour réduire cette durée, des optimisations bas niveau (langage assembleur essentiellement) ainsi qu'un parallélisme sur les données sont employés. Quand elles sont parallélisées, ces opérations se voient attribuées plusieurs tâches de traitement qui seront mises en concurrence sur les différents processeurs (ou cœurs). Dans le cas où il n'est pas nécessaire de réaliser un pré-traitement sur les images sources, cette étape de traitement est retirée du pipeline de codage.

L'étape de pré-estimation de mouvement sert à anticiper la prédiction inter. Cette étape consiste à estimer le coût des prédictions temporelles selon les images de référence. À la fin de cette estimation, les meilleures références

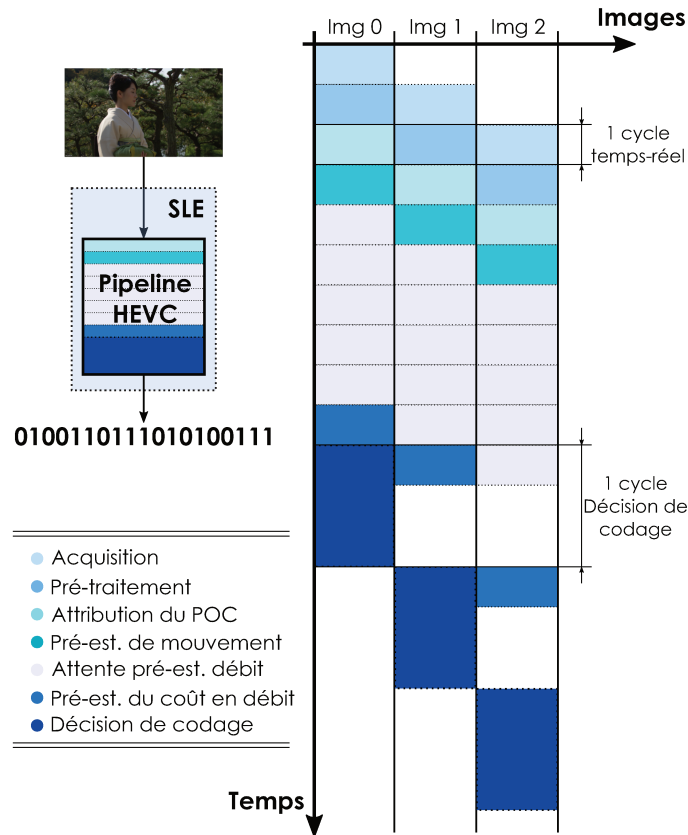


FIGURE 3.2 – Schéma du pipeline du codeur HEVC AteME (ou Single Layer Encoder (SLE)). Il permet d'effectuer un parallélisme inter-images pour chaque opération (Acquisition, pré-traitement...)

retenues disposeront d'un traitement privilégié dans la décision de codage. En effet, ces références pourront être les seules considérées dans la décision pour des configurations d'encodage rapides ou temps-réel. Dans le cas où l'image ne dispose pas de référence, comme c'est le cas de la première image de la vidéo ou des images de type I par exemple, la pré-estimation de mouvement est remplacée par une étape d'attente. Cette étape permet alors de conserver une même taille de pipeline entre les différents types d'images I, P et B.

L'étape de construction du GOP permet de faire passer le traitement des images de l'ordre d'affichage à l'ordre de codage. Pour cela, deux identifiants sont définis : l'indice d'image et le POC correspondants respectivement à l'identifiant de l'image dans l'ordre d'affichage et de codage. L'indice d'image est attribué dès l'acquisition tandis que le POC est attribué et calculé pendant la construction du GOP. Cette étape peut prendre plus ou moins de temps selon la structure de GOP employée. Il est en effet nécessaire de préalablement valider la présence de toutes les images d'une structure de GOP avant de laisser la main aux étapes nécessitant l'ordre de codage. Pour cela, il est possible de mettre l'image en attente en créant des étapes de pipeline vides. Ces étapes ne réalisent aucun traitement et servent uniquement à accumuler une quantité d'image suffisante. Ainsi, l'étape de construction de GOP peut s'étendre sur plusieurs cycles selon la position de l'image et la structure de GOP employée. La construction du GOP est une étape nécessaire au codage mais il ne s'agit donc pas à proprement parlé d'une étape du pipeline. Dans le cas d'un codage random-access (RA) illustré en Figure 3.3, la construction du GOP nécessitera 1 et 9 cycles respectivement pour la dernière et la première image (I_0 et P_8) du GOP dans l'ordre d'affichage.

L'étape de pré-estimation du coût en débit permet de réaliser une première estimation avant la décision de codage. Les informations issues de cette pré-estimation pourront être employées afin d'affiner l'efficacité du codage. Cette étape contribue à identifier les surcoûts en consommation de débit provoqués par des changements de scènes par exemple. Afin de pouvoir compenser ces coûts imprévus en débit, la régulation du débit est appliquée sur des séquences d'images. La durée de ces séquences a été fixée à 5 secondes ce qui représente un compromis entre l'efficacité de la régulation du débit et la latence du codage. Un coût élevé en débit sur quelques images pourra

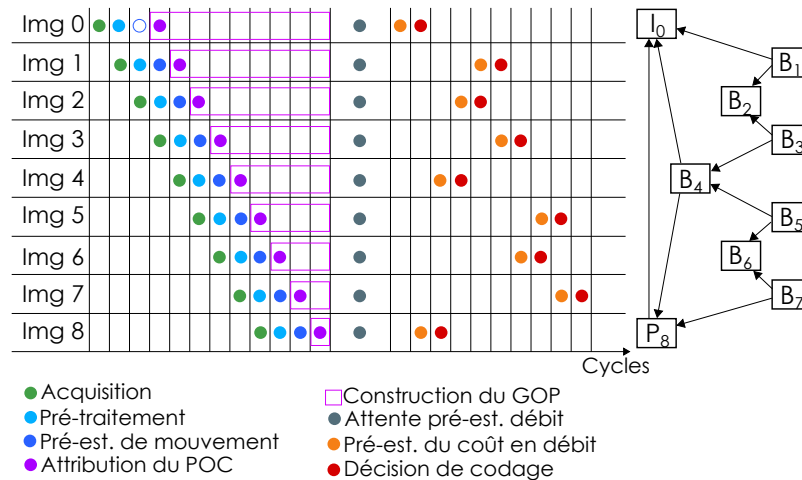


FIGURE 3.3 – Ordonnancement des étapes de pipeline avec une structure de GOP random-access (RA).

TABLE 3.1 – Durée d'un cycle de pipeline temps-réel en fonction de la fréquence d'image.

Fréquence d'image (IPS)	24	25	30	50	60	100	120
Durée d'un cycle (ms)	41.67	40	33.33	20	16.67	10	8.33

alors être compensé sur les autres images de cette période. La régulation du débit final sera plus efficace avec une vue d'ensemble sur le coût de chaque image. Une durée plus longue améliorerait significativement son efficacité mais augmenterait également la latence de codage. De plus, chaque image doit alors être conservée afin de réaliser cette pré-estimation du débit sur une séquence d'image. Cette conservation des images représente un coût important en mémoire.

Enfin, toutes les étapes de pipeline, hormis la décision de codage, respectent une contrainte temps-réel même si celle-ci n'est pas imposée au codage global. Ainsi, chaque étape de pipeline avant la décision de codage est exécutée avec une durée dépendant de la fréquence d'images. Le calcul de la taille du pipeline s'exprime alors ainsi :

$$\text{Taille du pipeline} = 1 + 5000 (\text{ms}) / \text{Durée d'un cycle (ms)}$$

Le '1' ajouté au total correspond à l'étape de décision de codage tandis que le 5000 ms correspond au délai d'attente requis par l'étape de pré-estimation du débit. La durée d'un cycle correspond à la contrainte temps-réel dépendant de la fréquence des images avec la relation :

$$\text{Durée d'un cycle (ms)} = 1000 / \text{Fréquence d'images (IPS)}$$

Ainsi chaque étape du pipeline de codage du SLE, hormis la décision de codage, doit respecter la contrainte temps-réel quelque soit le cas d'utilisation du codeur. La Table 3.1 présente la durée maximale des cycles en fonction des fréquences d'image pour un codage temps-réel.

Les différentes étapes de pipeline de chaque image sont exécutées en parallèle. Ainsi, le codeur SLE dispose d'un premier parallélisme au niveau de la séquence des images. Le parallélisme entre les images permet alors, une fois la latence de codage écoulee, d'exécuter la décision de codage à chaque cycle comme illustré sur les Figures 3.2 et 3.3. Ce parallélisme entre les étapes de pipeline est illustré sous la forme d'un réseau de pétri avec deux étapes en Figure 3.4. L'étape située plus haut dans le pipeline doit être réalisée une première fois avant d'autoriser l'exécution de l'étape suivante. L'annexe B présente le cheminement des différentes opérations permettant le traitement en parallèle de deux étapes du pipeline. Afin de traiter les étapes de pipeline en parallèle, chacune d'elle est intégrée dans une tâche de traitement.

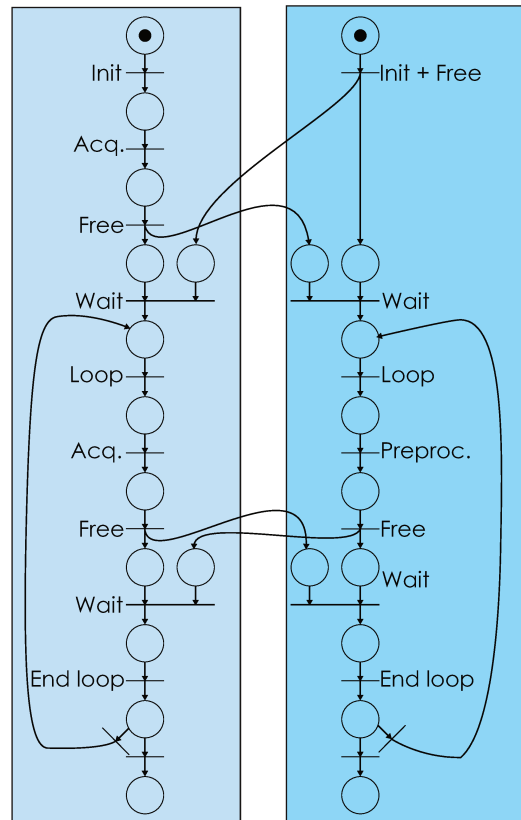


FIGURE 3.4 – Schéma du réseau de pétri pour la synchronisation inter-étapes du SLE avec 2 étapes de pipeline. Exemple avec l'acquisition (Acq.) et le pré-traitement des images sources (Preproc.)

3.1.2 Traitement multi-tâches des opérations

Une tâche ou un *thread* englobe une à plusieurs opérations de traitement qui seront attribuées à un cœur. Une tâche permet donc ici d'exécuter une ou plusieurs opérations en parallèle des autres tâches. L'intégration des opérations de traitement au sein de tâches est réalisée par des instructions de type POSIX [58]. Les instructions de type POSIX (pour "Portable Operating System Interface") font partie d'une famille d'instructions permettant de maintenir une compatibilité entre différents systèmes d'exploitation. Elles permettent une gestion des interfaces utilisateurs et des interfaces logicielles. L'emploi de ce type d'instruction permet au SLE d'être exécuté sur n'importe quel type de système d'exploitation.

L'ensemble des tâches est constitué d'une tâche de management et de plusieurs tâches de traitement. La tâche de management gère la synchronisation entre les tâches de traitement ainsi que leur ordonnancement en fonction de leur priorité. Les tâches de traitement constituent les différentes opérations du codage réparties au sein des différentes étapes de pipeline du SLE. Une étape de pipeline peut être constituée d'une ou de plusieurs tâches de traitement. Ces tâches peuvent également avoir des dépendances d'ordonnancement au sein d'une même étape de pipeline. Dans ce cas, une tâche de traitement principal et prioritaire est créée au sein de l'étape de pipeline afin de gérer les différentes dépendances, ainsi que le parallélisme entre les autres tâches de traitement. Le codeur SLE intègre donc un deuxième niveau de parallélisme entre les tâches de traitement d'une même étape de pipeline.

A la fin de chaque étape de pipeline, toutes les tâches de traitement émettent un signal de synchronisation à destination de la tâche de management. Une fois que tous les signaux sont reçus, la tâche de management peut alors lancer le prochain cycle de pipeline. Cette architecture en pipeline permet notamment d'obtenir les informations de codage pour une image entière à chaque cycle au coût d'une certaine latence de codage. Cette latence correspond à la durée du pipeline entier sur la première image, elle dure environ 5 secondes en HEVC. La durée maximale de traitement de chaque étape de pipeline (ou cadence) dépend de la durée de la plus longue opération, à savoir la décision de codage. Selon le contexte d'utilisation, cette dernière peut respecter la contrainte temps-réel en réalisant

des compromis entre vitesse et efficacité de codage. Ces compromis sont présentés dans le chapitre suivant. Dans ce chapitre, la décision de codage cherche à atteindre la meilleure efficacité de codage possible sans prendre en compte la notion de temps de traitement, on parlera de codage FILE. Toutes les autres étapes sont exécutées en respectant la contrainte temps-réel.

3.1.3 Décision de codage

L'étape de décision de codage est l'opération de loin la plus complexe du pipeline. De part sa complexité, davantage d'efforts sur son accélération ont été mis en œuvre. Cette décision, réalisée au niveau des CU, est constituée séquentiellement des prédictions inter et intra, du calcul des résidus, de la transformée, de la reconstruction de l'image et de la pré-signalisation des données de la CU. Cette dernière pré-signalisation est ensuite récupérée au moment du traitement de CABAC qui est réalisé à la toute fin de la décision de codage. Plus précisément, le traitement de CABAC est effectué au sein de l'étape de décision de codage, mais seulement une fois que la dernière CU ait été traitée.

Le traitement des CU est réalisé de gauche à droite puis de haut en bas. Chaque CU dépend des CU voisines de gauche, du haut, d'en haut à gauche et d'en haut à droite, comme illustré sur la Figure 3.5a. Au moins une de ces dépendances doit être respectée. Dans le cas d'un début de ligne, le bloc de gauche peut être remplacé par le bloc terminant la ligne précédente. Cependant, ces dépendances ne traversent pas les slices comme illustré sur la Figure 3.5a. Le trait rouge représente une frontière de slice, le bloc jaune est le bloc courant et les blocs verts sont les blocs dont il dépend et qui doivent être traités précédemment.

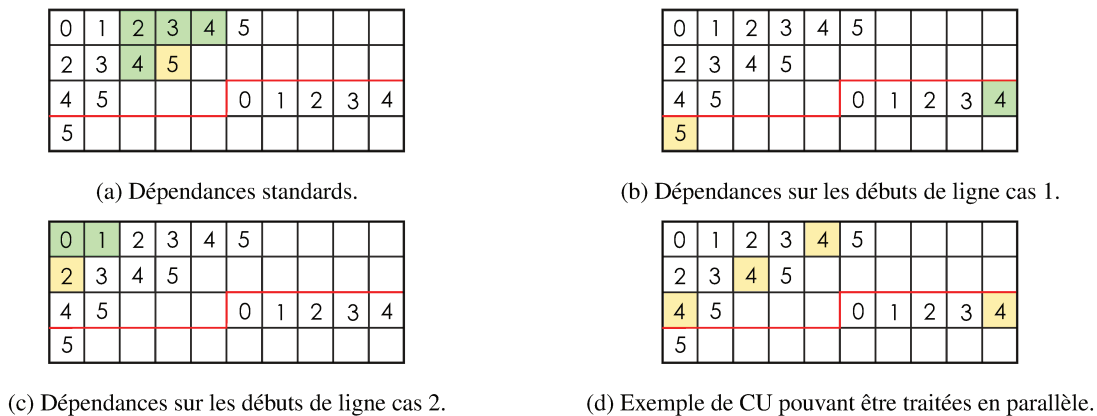


FIGURE 3.5 – Dépendances entre les "Coding Units" et ordre de traitement dans le SLE. En jaune, le bloc courant. En vert, les blocs lui servant de référence. En rouge, les frontières de slice.

En début de slice, la CU n'a pas de dépendance. En début de ligne en revanche, la CU dépend de ses voisines. Par exemple, sur la Figure 3.5b, au niveau de la seconde slice, le premier bloc de la deuxième ligne numéroté 5 n'a qu'un seul voisin, il s'agit du bloc terminant la ligne précédente. Il faudra donc traiter ce bloc 5 après le bloc 4 afin de respecter les dépendances. Dans le premier slice, le premier bloc de la deuxième ligne doit attendre le dernier bloc de la première ligne. Toutefois, le SLE autorise cette dépendance à ne pas être prise en compte sur les bords quand au moins deux blocs voisins sont disponibles, comme illustré sur la Figure 3.5c. Ainsi, le premier bloc de la deuxième ligne de du premier slice pourra être traité en deuxième instance. Cela permet de réaliser un parcours des lignes de CU de façon indépendante. Le parallélisme par CU induit par ce parcours est schématisé en Figure 3.5d. En revanche, ce parcours n'est pas le WPP (voir section 2.5.7). Il n'est donc pas signalé dans la syntaxe et ne pourra pas être employé tel quel par un décodeur. Cela empêche l'écriture du bitstream à la fin de chaque bloc. C'est pourquoi une pré-signalisation est réalisée à la fin du traitement de chaque bloc. Cette pré-signalisation conserve les informations de codage avant de les transmettre à l'écriture finale du bitstream. Il faudra attendre la fin du traitement de tous les CU de la slice courante avant de débiter l'écriture du bitstream.

La Table 3.2 présente les nombres maximum de blocs qui peuvent être traités en parallèle avec ce parcours des lignes pour différentes résolutions. Pratiquement, seule la colonne des CU 64×64 dans la Table 3.2 est conservée. La raison de ce choix est liée à la gestion mémoire. La mémoire cache sollicitée sur des blocs de taille inférieure

TABLE 3.2 – Nombres maximum de blocs qui peuvent être traités en parallèle selon le parcours des lignes employé dans le SLE par résolution et par taille de "Coding Units"

Résolution	Nombres maximum de blocs pouvant être traités en parallèle			
	CU 64×64	CU 32×32	CU 16×16	CU 8×8
960×540	7	15	30	60
1280×720	10	20	40	80
1920×1080	15	30	60	120
3840×2160	30	60	120	240
7680×4320	60	120	240	480

à 32×32 ne sera pas très efficace et devra être chargée plus régulièrement. Cette perte de performance est évitée en employant uniquement des blocs de taille plus grande. Ainsi, lorsque la taille des CU est fixée à 16×16 par exemple, les blocs sont traités par groupe de 4.

L'opération de CABAC est réalisée dans un thread mono-core une fois que tous les blocs d'une même slice ont été codés. Pour ce faire, à la fin de chaque codage de blocs, un bitstream intermédiaire est créé. Ce bitstream contient les informations nécessaires à la création du bitstream final pendant toute la durée de la décision de codage. Il regroupe ainsi la représentation binaire des éléments de la syntaxe et les contextes CABAC. Dans les cas où la représentation binaire ne peut être envoyée, comme par exemple dans le cas du QP, c'est directement sa valeur qui est conservée. La sauvegarde de ces informations occupe une quantité très importante de mémoire et requiert une bonne gestion de celle-ci. De plus, tous les accès mémoire sont doublés par rapport à une génération immédiate du bitstream. Les données doivent d'abord être écrites pour la création du bitstream intermédiaire lors de la pré-signalisation. Puis, elles sont écrites à nouveau lors de la génération du bitstream final.

Le codeur SLE intègre alors dans sa décision de codage un troisième niveau de parallélisme sur le parcours des blocs. De plus, puisque le parcours des blocs n'est pas signalé dans la syntaxe, la séparation des images en Tiles indépendantes reste possible. Cette répartition permet d'augmenter davantage le nombre de threads, mais cela peut également créer des artefacts de codage sur les frontières de Tiles. Pour palier à ce problème, le filtre anti-blocs défini dans le standard HEVC est employé. Ce filtre est exécuté pour une ligne entière une fois toutes les CU d'une ligne codées. Un quatrième niveau de parallélisme au niveau des images est alors possible avec la répartition en Tiles et en Slices (voir section 2.5.7. Un cinquième niveau de parallélisme sur les opérations de filtrage peut également être employés dans le SLE. Ces multiples parallélismes permettent de réaliser des encodages HEVC très rapides avec un impact minime sur l'efficacité de codage. Différents modes de qualité sont alors définis. Dans ce chapitre, seul le mode de qualité cherchant à approcher la meilleure efficacité de codage est employé : il s'agit du mode de qualité *FILE*. Les autres de modes de qualité ainsi que les compromis entre vitesse et efficacité de codage sont présentés dans le chapitre suivant en section 4.4. On peut évaluer l'efficacité de codage du SLE en le comparant à d'autres solutions d'encodages HEVC. La Table 3.3 compare notamment le SLE aux encodeurs HM, x265 et Kvazaar avec la métrique BD-BR à qualité constante. On constate que le codeur Kvazaar fournit des encodages qui ne sont pas au niveau du SLE : en moyenne 1342.7% d'écart BD-BR en configuration RA. Ce codeur est toutefois encore en développement et pourrait améliorer son efficacité de codage notamment en intégrant des mécanismes de prédiction intra et inter améliorés. Le SLE, quant à lui, est employé dans une configuration réalisant un léger compromis entre l'efficacité de codage et la vitesse de traitement. En effet, la configuration FILE du SLE employée ici utilise seulement 19 directions pour la prédiction intra et ne réalise pas de décision inter sur les partitions asymétriques : $2N \times nD$, $2n \times nU$, $nL \times 2N$ et $nR \times 2N$ (voir Figure 2.15). Ces restrictions sur les outils de codage expliquent les écarts entre le SLE et le HM. De plus, la version du SLE employé date de 2015 tandis que la version du x265 date de 2017 : version 2.6. Entre 2015 et 2017, plusieurs évolutions du code du SLE ont permis d'améliorer son efficacité de codage. Malheureusement, faute de temps, ces modifications n'ont pas pu être mise en place dans le codeur SHVC réalisé durant cette thèse. Les codeurs x265 et HM sont plus efficaces, en terme de compression, que le SLE dans les configurations AI et LD-P. La Figure 3.6 illustre ce phénomène avec la courbe débit/distorsion sur la séquence B1 en configuration AI. En apparence, le SLE fournit une efficacité de codage équivalente au x265 en configuration RA. La Figure 3.7 présente la courbe débit/distorsion sur la séquence B1 en configuration RA qui illustre cette différence : en terme de compression, le HM est plus efficace que le SLE lui-même plus efficace que le x265. En se basant sur les résultats de la Table 3.3, le x265 semble plus efficace pour les séquences A1, A2 et B5 en configuration RA. Toutefois, pour ces 3 séquences, les courbes débit/distorsion se croisent, c'est pourquoi il est

TABLE 3.3 – Comparaison BD-BR de SLE par rapport à d’autres codeurs HEVC. Par ordre décroissant d’efficacité de codage : HM, x265, SLE, Kvazaar. Encodeur évalué vs Encodeur de référence.

Séquences	BD-BR SLE vs HM			BD-BR SLE vs x265			BD-BR Kvazaar vs SLE		
	AI	LD-P	RA	AI	LD-P	RA	AI	LD-P	RA
A1	9.2%	23.5%	31.6%	9.3%	10.9%	7.6%	386,6%	420.4%	628.3%
A2	9.0%	12.0%	19.4%	8.8%	9.8%	1.0%	389,1%	488.0%	695.1%
B1	13.6%	11.8%	32.2%	15.7%	10.3%	-6.9%	806,2%	943.5%	1418.6%
B2	7.9%	17.9%	28.0%	8.2%	9.9%	-3.8%	876,4%	950.3%	1448.7%
B3	9.0%	22.1%	25.3%	9.0%	11.9%	-7.5%	873,9%	954.3%	1839.3%
B4	12.2%	17.2%	25.6%	13.1%	14.2%	-3.1%	844,9%	979.2%	1505.6%
B5	8.5%	38.7%	50.9%	9.4%	18.6%	23.0%	868,4%	876.2%	1863.2%
Moyenne	9.9%	20.5%	30.4%	10.5%	12.2%	1.5%	720.8%	801.7%	1342.7%

difficile de statuer seulement à partir de la métrique BD-BR. En général, le SLE est plus efficace que le x265 à bas débit, mais cette tendance s’inverse à plus haut débit. Le codeur x265 propose certes des codages efficaces mais il n’a pas été retenu pour l’implémentation du codage SHVC durant cette thèse. Ce choix repose essentiellement sur la politique de développement open-source proposé par x265. Cette thèse étant réalisée dans un cadre industriel (CIFRE), il n’est pas intéressant pour la société Ateme de proposer un code open-source. Le codeur HEVC retenu est donc le SLE dans sa version datant de 2015.

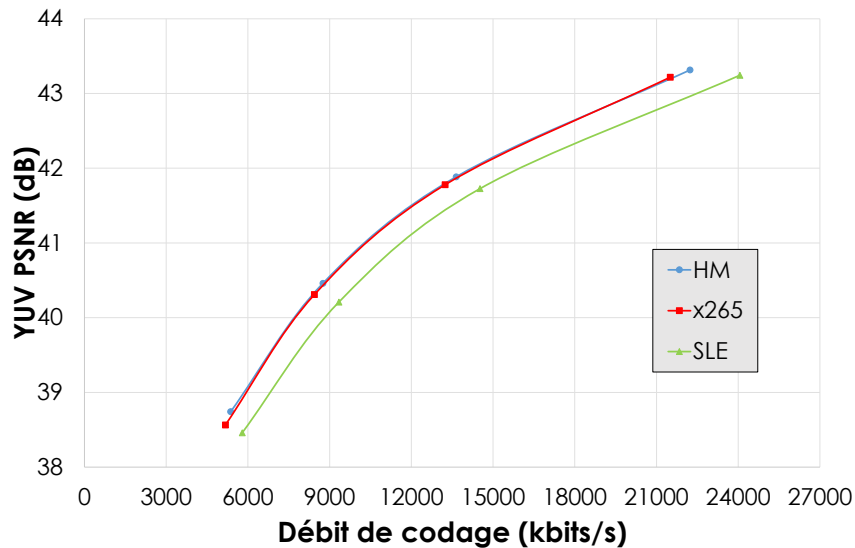


FIGURE 3.6 – Courbes débit/distorsion des encodages obtenus avec les codeurs HM, x265 et SLE en configuration AI sur la séquence B1. HM et x265 sont équivalents et plus efficaces que le SLE dans sa version de 2015.

3.1.4 Régulation de débit

Le SLE intègre également un module de régulation de débit permettant de réaliser des encodages à facteur de quantification constant (QP constant) et à débit constant (CBR). Le mécanisme de codage CBR s’appuie sur la régulation du débit dans le domaine $R-\lambda$ dont le principe est expliqué en 2.7.1. Cette régulation s’opère au niveau des images et apparaît avant et pendant la décision de codage. En effet, la régulation de débit du codeur

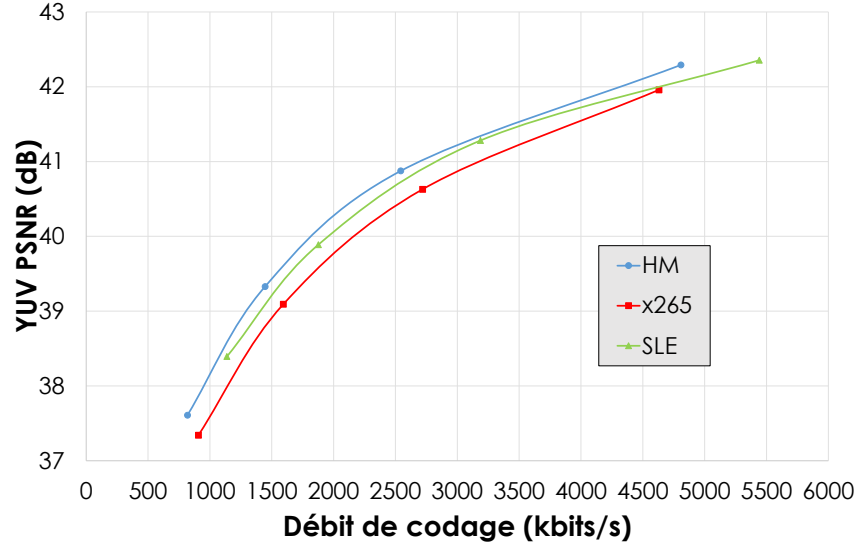


FIGURE 3.7 – Courbes débit/distorsion des encodages obtenus avec les codeurs HM, x265 et SLE en configuration RA sur la séquence B1. HM est plus efficace que le SLE dans sa version de 2015, lui-même plus efficace que x265.

SLE est tout d'abord simulée sur une période de 5 secondes en dehors de l'étape de décision de codage. Comme vu précédemment, cette pré-estimation du débit introduit une latence supplémentaire au pipeline de codage. Une première cible de débit par image est attribuée à chaque image. Une mauvaise estimation peut être causée par une sous-évaluation ou par une sur-évaluation du coût de la signalisation. Dans chacun de ces cas, les données fournies au VBV feront entrer respectivement en famine ou en engorgement. Dans le cas où le débit est sous-évalué, la contrainte vis-à-vis du VBV sera respectée mais la qualité pourra être augmentée (famine du VBV). Ainsi, la valeur du QP attribuée aux images de référence dans le pré-traitement sera rehaussée dans la décision de codage. Dans le cas où le débit est sur-évalué, le VBV est engorgé. La valeur du QP attribuée aux images prédites dans le pré-traitement sera alors abaissée dans la décision de codage. Par la suite, cette évaluation du coût en débit sera à nouveau réalisée une fois l'image entièrement codée. La contrainte en débit, alors ajustée une seconde fois, sera prise en compte pour les prochains encodages sur une période de 5 secondes. Cette période permet de lisser la qualité de la vidéo et de conserver une qualité uniforme. Dans le cas de changement de scène détecté, cette période est arrêtée et initialisée à partir du début de la nouvelle scène.

Le codeur SLE est alors en mesure de fournir des encodages HEVC intéressants. Ce codeur sert de base au développement du codeur SHVC proposé et réalisé au cours de cette thèse.

3.2 Architecture de l'encodeur scalable SHVC

Le codage scalable permet de répondre à une grande partie des nouvelles contraintes de création et d'usage vidéo. Néanmoins, ce type de codage demeure complexe. Une solution permettant de réaliser des encodages scalables doit donc respecter certaines contraintes. Elle doit notamment être la moins intrusive possible à la fois en :

- altérant le moins possible le codage,
- s'intégrant efficacement à l'architecture en pipeline,
- introduisant le moins possible de délai,
- introduisant le moins possible de stockage en mémoire.

La solution de codage SHVC que nous proposons dans [59], appelée "SHVC ATEME Encoder" (SHVC-AE), est composée de deux codeurs SLE : l'un correspond à la couche de base tandis que l'autre correspond à la couche

de rehaussement. Chacune de ces instances de SLE réalise un codage sur une résolution différente de l'autre instance. Ainsi, chaque couche traitera une résolution différente pour une scalabilité spatiale sur 2 couches. De plus, les données de prédiction inter-couches doivent être re-dimensionnées avant d'être utilisées par la couche de rehaussement. Les deux instances de SLE réalisent leur encodage respectif simultanément et se voient chacune attribuer un nombre de cœurs. Le nombre de cœurs est fixé dans le code et peut être :

- égale sur chaque couche,
- plus importante sur la couche de rehaussement,
- partagé entre toutes les couches.

Dans le cas où un nombre de cœurs est attribué à une couche, cette dernière ne peut utiliser que cette quantité de ressources (allocation statique des ressources). Quand les cœurs sont partagés entre toutes les couches, les différentes opérations de codage sont alors mises en concurrence (allocation dynamique des ressources). Afin de pouvoir échanger les informations de codage nécessaires à la prédiction inter-couches, une synchronisation entre les couches est nécessaire. Cette synchronisation est réalisée à l'échelle du pipeline des codeurs SLE. Au lieu d'attribuer une tâche de management à chaque instance de SLE, une unique tâche est créée pour synchroniser les couches à la fois entre elles et dans leurs traitements. Le mécanisme de synchronisation est alors le même que dans le SLE, à ceci près qu'il est étendu à toutes les couches scalables. Le cycle de pipeline suivant commence une fois que tous les signaux de synchronisation de toutes les couches sont reçus par la tâche de management. Par conséquent, la cadence du codeur SHVC proposée est relative à la durée de la tâche de traitement la plus longue des deux couches.

L'acquisition du codeur SHVC-AE peut employer une ou deux sources vidéos non compressées en entrée comme illustré respectivement sur les Figures 3.9 et 3.10. Quand le SHVC-AE acquiert une unique source vidéo (Figure 3.9), une étape de sous-échantillonnage doit être réalisée sur la couche de base pour un codage scalable en résolution. Dans le SHVC-AE, cette étape est spécifique aux couches servant de référence. Elle est ajoutée au pipeline de la couche base et exécutée juste après la réception de l'image à coder. En revanche, quand le SHVC-AE acquiert deux sources vidéos, chacune d'elles correspond aux couches de base et de rehaussement. Dans ce cas, le SHVC-AE n'exécute pas l'étape de sous-échantillonnage, comme illustré sur la Figure 3.10, qui doit alors être effectuée au préalable en dehors du processus de codage. Ce cas d'usage multi-sources ne permet pas d'intégrer le SHVC-AE dans un système de diffusion muni d'une caméra fournissant un unique flux vidéo. En revanche, pour des comparaisons entre codeurs SHVC, il permet d'assurer l'utilisation des mêmes sources pour d'une part la couche de base et d'autre part la couche rehaussement.

Ensuite, afin de bénéficier de la prédiction inter-couches en scalabilité spatiale, il est nécessaire d'adapter les informations de codage transmises à la couche de rehaussement. Pour cela, une opération de sur-échantillonnage définie dans le standard SHVC [60] est réalisée sur les textures et les vecteurs de mouvement produits par la couche de base. Cette opération est réalisée une fois que les informations de codage de la couche de base sont prêtes, c'est-à-dire à la fin de sa décision de codage, comme illustré sur la Figure 3.10. De la même manière que pour le filtre de sous-échantillonnage, l'opération de sur-échantillonnage est intégrée dans sa propre étape de pipeline. Celle-ci peut-être effectuée sur la couche de rehaussement ou sur la couche de base. Dans les deux cas, cette opération requiert d'être exécutée :

- après la décision de codage de la couche de base,
- avant la décision de codage de la couche de rehaussement.

Pour cela, les pipelines des couches de base et de rehaussement doivent être synchronisés. Chaque étape de pipeline du SLE est déjà synchronisée avec les autres étapes de ce même pipeline. Cette synchronisation est pilotée par un thread de management. Ce thread est partagé entre les couches de base et de rehaussement permettant un alignement des pipelines de ces deux couches. Ainsi, chaque étape de pipeline est également synchronisée avec celles des autres instances de SLE. C'est pourquoi la synchronisation entre les étapes de codage des couches de base et de rehaussement, et l'étape de sur-échantillonnage peut être réalisée en effectuant un décalage entre les pipelines des couches. Cette solution permet non seulement de conserver l'enchaînement des différentes étapes du pipeline initial, mais aussi de ne pas ajouter de communication entre les couches. En effet, il aurait été possible de réaliser une synchronisation communicante entre les couches à partir d'outils tels que les mutex ou les sémaphores. Ces outils permettent de restreindre l'accès à une mémoire partagée. A chaque fois qu'un accès à cette mémoire est nécessaire pour une tâche donnée, un signal est transmis aux autres tâches. Ce signal leur indique qu'elles ne peuvent pas accéder à la mémoire partagée avant qu'un second signal ne soit émis. Une fois que le traitement sur la mémoire partagée est achevé, un second signal est émis afin d'indiquer que la mémoire est à présent libre d'accès. Dans le cas du sur-échantillonnage, la mémoire partagée entre les différentes couches serait constituée de l'image

reconstruite et des vecteurs de mouvement provenant de la couche de base. Les tâches y faisant appel seraient les décisions de codage des couches de base et de rehaussement. La décision de la couche de rehaussement pourrait alors être bloquée pendant la durée de la décision de la couche de base. Non seulement ces solutions obligent une interaction entre les différentes instances de SLE, mais elles rendraient également la synchronisation des pipelines plus complexe. Le décalage des étapes entre les couches, quant à lui, ne fait intervenir que le thread de management. Le décalage peut alors être effectué :

- au plus tôt, après l'acquisition,
- au plus tard, avant la décision de codage de la couche de base,
- ou entre l'acquisition et la décision de codage de la couche de base.

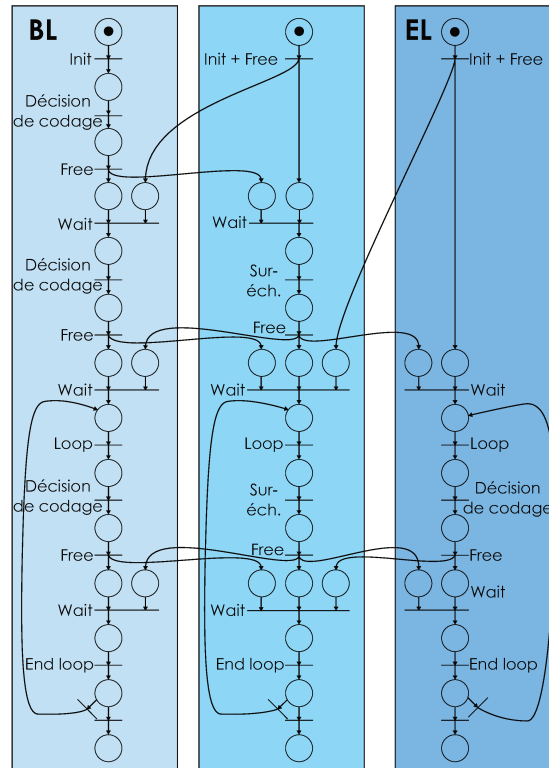


FIGURE 3.8 – Réseau de pétri de la synchronisation entre la décision de codage de la couche de base, le sur-échantillonnage et la décision de codage de la couche de rehaussement dans le SHVC-AE

La Figure 3.8 représente le réseau de pétri de la synchronisation entre la décision de codage de la couche de base, le sur-échantillonnage et la décision de codage de la couche de rehaussement dans le SHVC-AE. Le cheminement des jetons est expliqué en annexe C. Dans le SHVC-AE, il a été choisi de démarrer la couche de rehaussement deux cycles de pipeline après la couche de base afin de synchroniser les pipelines dès le début du traitement. Ainsi, la couche de rehaussement est mise en attente dès lors que l'acquisition est terminée. Cette attente prend fin une fois que le nombre de cycles nécessaires à la synchronisation inter-couches est atteint. Contrairement aux autres possibilités, ce choix de positionnement du décalage permet de ne mettre en attente que l'image source. Dans les autres cas, il faudrait mettre en attente non seulement l'image source, mais également les données des étapes du pipeline déjà exécutées. Par conséquent, par comparaison avec le SLE, le SHVC-AE est plus long de seulement 3 cycles de pipeline. La latence de la solution de codage SHVC proposée est donc plus importante de 3 cycles comparée à celle du SLE.

3.2.1 Sous-échantillonnage SHVC

L'opération de sous-échantillonnage est réalisée dans le cadre de la scalabilité spatiale sur les couches servant de référence : c'est le cas pour la couche de base ou pour les couches de rehaussement utilisées comme référence.

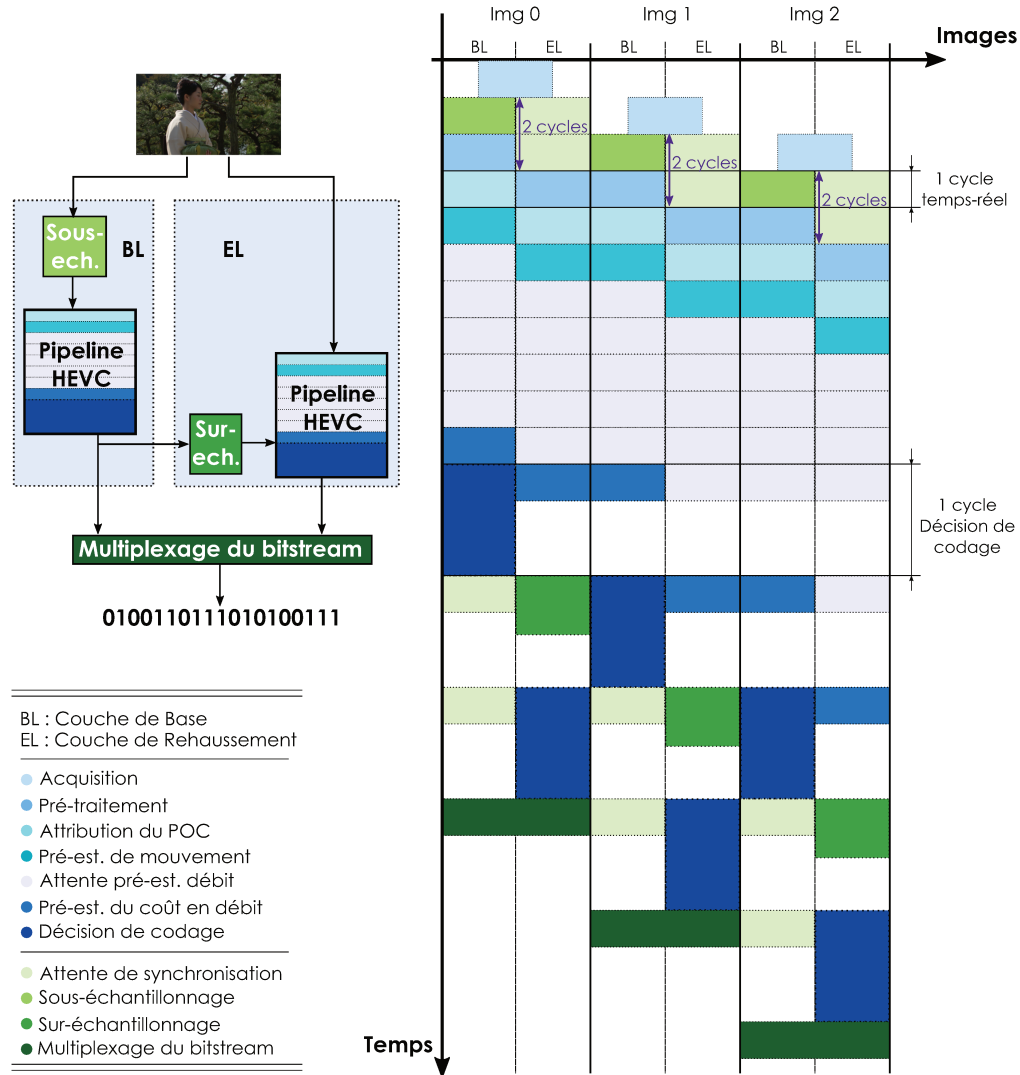


FIGURE 3.9 – Schéma du pipeline de codage de l'encodeur scalable avec une seule source vidéo pour les couches de base (BL), de rehaussement (EL) et le sur-échantillonnage sur l'EL.

Cette opération prend effet sur la vidéo non compressée et requiert donc un temps de traitement important. Elle peut être réalisée en dehors de la chaîne de codage, dans ce cas aucune étape de pipeline n'est ajoutée. Elle peut aussi être réalisée au sein de la chaîne de codage où elle est intégrée au pipeline. Dans le SHVC-AE, l'opération de sous-échantillonnage est située au début du pipeline, juste après l'acquisition et avant les pré-traitements éventuels de l'image comme illustré sur la Figure 3.10. Ce choix d'implémentation permet de s'approcher au plus près du comportement d'un codeur scalable avec deux sources. En effet, en réalisant cette opération au plus tôt, l'image source est alors altérée le moins possible par d'éventuels traitements. L'opération de sous-échantillonnage dispose de son propre étage de pipeline permettant ainsi de l'exécuter en parallèle des étapes de pipeline. Enfin, cette opération est réalisée sur la couche qui en a l'utilité, à savoir, la couche de base uniquement dans le cas d'un codeur scalable sur deux couches.

Dans le SHVC-AE, il est possible de n'utiliser qu'une vidéo source ou bien autant de sources qu'il y a de couches. Ces multiples sources sont alors préalablement sous-échantillonnées. Dans le premier cas, une seule source, le pipeline de codage est augmenté d'une étape : le sous-échantillonnage. L'acquisition de la source est réalisé une seule fois dans un thread commun à toutes les couches scalables, comme schématisé sur la Figure 3.9. Ensuite, la couche de base peut réaliser son opération de sous-échantillonnage tandis que la couche de rehaussement place ses images sources en attente afin de déphaser les pipelines des SLE. Dans le second cas, plusieurs

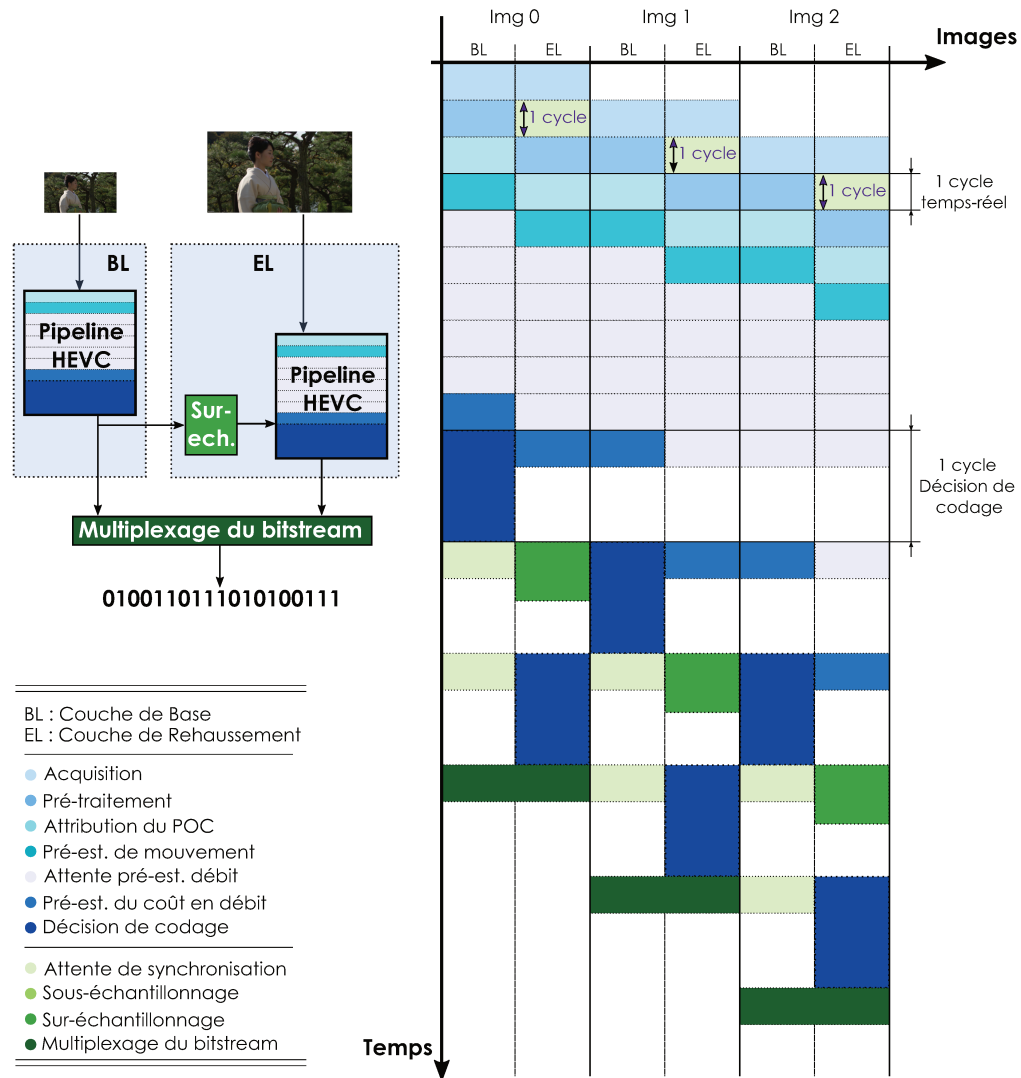


FIGURE 3.10 – Schéma du pipeline de codage de l’encodeur scalable avec plusieurs sources vidéo pour les couches de base (BL), de rehaussement (EL) et le sur-échantillonnage sur l’EL.

sources déjà aux formats souhaités, la couche de base n’a pas à réaliser d’opération de sous-échantillonnage. Les contraintes sur les pipelines des SLE ne dépendent plus que de la prédiction inter-couches. Chaque couche réalise alors sa propre opération d’acquisition simultanément comme illustré sur la Figure 3.10. Ensuite, la couche de base peut poursuivre son pipeline alors non modifié tandis que les images sources de la couche de rehaussement sont placées en attente pour au moins un cycle.

De plus, l’opération de sous-échantillonnage n’est pas définie dans le standard SHVC. En effet, cette opération n’est pas nécessaire dans le décodage. Le codeur SHM emploie un filtrage avec 12 coefficients issus des recommandations [61]. Cette recommandation fournit un paramétrage pour un filtre de sous-échantillonnage appliqué aux composantes chromatiques dans le cas d’une conversion de l’échantillonnage de l’image : 4:4:4 vers 4:2:2 et 4:2:2 vers 4:2:0. A partir de ce filtre, le SHM emploie alors une source pré-sous-échantillonnée pour le codage de sa couche de base. Dans ce chapitre, les codages employant le SHVC-AE ne réalisent pas d’opération de sous-échantillonnage. De manière identique au SHM, le SHVC-AE emploie la même source pré-sous-échantillonnée sur la couche de base. Le chapitre suivant se place en condition d’utilisation temps-réel et propose une optimisation de ce filtrage pour répondre à une telle contrainte, mais également un paramétrage permettant d’améliorer l’efficacité de codage global.

3.2.2 Sur-échantillonnage SHVC

L'opération de sur-échantillonnage nécessaire à la prédiction inter-couches en scalabilité spatiale peut être réalisée sur la couche de rehaussement ou sur la couche de base. Les Figures 3.9 et 3.11 illustrent respectivement ces deux possibilités. Le sur-échantillonnage est nécessairement réalisé avant la décision de codage de la couche de rehaussement et après celle de la couche de base. Le choix du placement de cette opération sur une couche en particulier peut dépendre de différents besoins comme l'occupation des ressources matérielles ou la flexibilité algorithmique.

En choisissant de réaliser le sur-échantillonnage sur la couche de base, comme illustré sur la Figure 3.11, l'occupation des ressources matérielles sera améliorée. En effet, la couche de base traite et transmet une quantité de données moins grande par comparaison à la couche de rehaussement. La couche de rehaussement ainsi que la couche de base emploient toutes deux les mêmes outils de codage issus du standard HEVC. Puisque la couche de rehaussement ajoute à ces outils la prédiction inter-couches, la décision de codage sur la couche de base sera réalisée plus rapidement que sur la couche de rehaussement. Ainsi, dans le cas d'une répartition équivalente des ressources matérielles entre la couche de base et la couche de rehaussement, il peut être intéressant de réaliser l'opération de sur-échantillonnage sur la couche de base afin d'équilibrer la charge de ses ressources matérielles.

En réalisant l'opération de sur-échantillonnage sur la couche de rehaussement, comme illustré sur la Figure 3.9, c'est la flexibilité algorithmique qui sera améliorée. En effet, le sous-échantillonnage étant effectué sur la couche de référence et le sur-échantillonnage sur la couche prédite, chaque codeur SLE se voit ajouter une étape de pipeline dont les futures dépendances sont sur le même pipeline. Cela permet de définir, du point de vue algorithmique, les couches de référence et les couches prédites comme disposant respectivement d'une étape de sous-échantillonnage et d'une étape de sur-échantillonnage. Cette définition permet d'ajouter des couches de rehaussement supplémentaires de façon plus cohérente. Ce placement est donc moins intrusif et plus flexible en ce qui concerne l'ajout de couches supplémentaires, mais également plus stressant sur les ressources matérielles attribuées à la couche de rehaussement. Dans ce cas, la répartition des cœurs sur les différentes couches sera plus efficace en attribuant davantage de ressources à la couche de rehaussement.

Dans le cas du codeur SHVC-AE, l'opération de sur-échantillonnage est réalisée par défaut sur la couche de rehaussement, juste avant la décision de codage faisant intervenir la prédiction inter-couches. Il est toutefois possible de réaliser cette opération sur la couche de base. Le positionnement du sur-échantillonnage dépend alors d'un paramètre fourni à l'initialisation du codage.

Par défaut, l'information de texture, sous la forme d'une image reconstruite, et les vecteurs de mouvement sont copiés depuis la couche de base vers la couche de rehaussement. Pour cela, deux buffers partagés entre les couches sont ajoutés, l'un pour l'image reconstruite, l'autre pour les vecteurs de mouvement. Les buffers doivent conserver ces informations jusqu'à ce que la couche de rehaussement les emploie dans la prédiction inter-couches. Ainsi, ces buffers doivent tenir compte du déphasage entre les pipelines des couches. C'est pourquoi il est nécessaire d'instancier une table de buffers pouvant contenir 3 éléments au minimum. En effet, 3 étapes de pipeline sont susceptibles de nécessiter un accès aux buffers :

- la décision de codage de la couche de base sur une image N pour y sauvegarder les informations de codage,
- le sur-échantillonnage sur l'image $N-1$ pour adapter les informations de codage au format de la couche de rehaussement,
- la décision de codage de la couche de rehaussement sur l'image $N-2$ pour utiliser les informations de codage dans la prédiction inter-couches.

Les données de prédiction inter-couches sont alors conservées sur au plus 3 cycles de pipeline. Le SHVC-AE permet donc d'économiser une quantité importante de mémoire. Avec une implémentation séquentielle, il aurait fallu conserver ces informations pendant l'intégralité du pipeline de la couche de rehaussement.

En terme de complexité, le sur-échantillonnage sur les textures consiste à réaliser les opérations suivantes :

- calculer les coordonnées du pixel dans la couche de base à partir des coordonnées du pixel dans la couche de rehaussement,
- identifier les coefficients à appliquer,
- réaliser l'opération de filtrage à partir des coefficients standardisés d'abord horizontalement puis verticalement.

Ce processus est standardisé dans la première extension du standard HEVC [60]. Davantage d'informations sur la complexité du sur-échantillonnage sur les textures sont données en Section 4.2.

Le sur-échantillonnage sur les vecteurs de mouvement est similaire au filtrage sur les textures. Toutefois, celui-

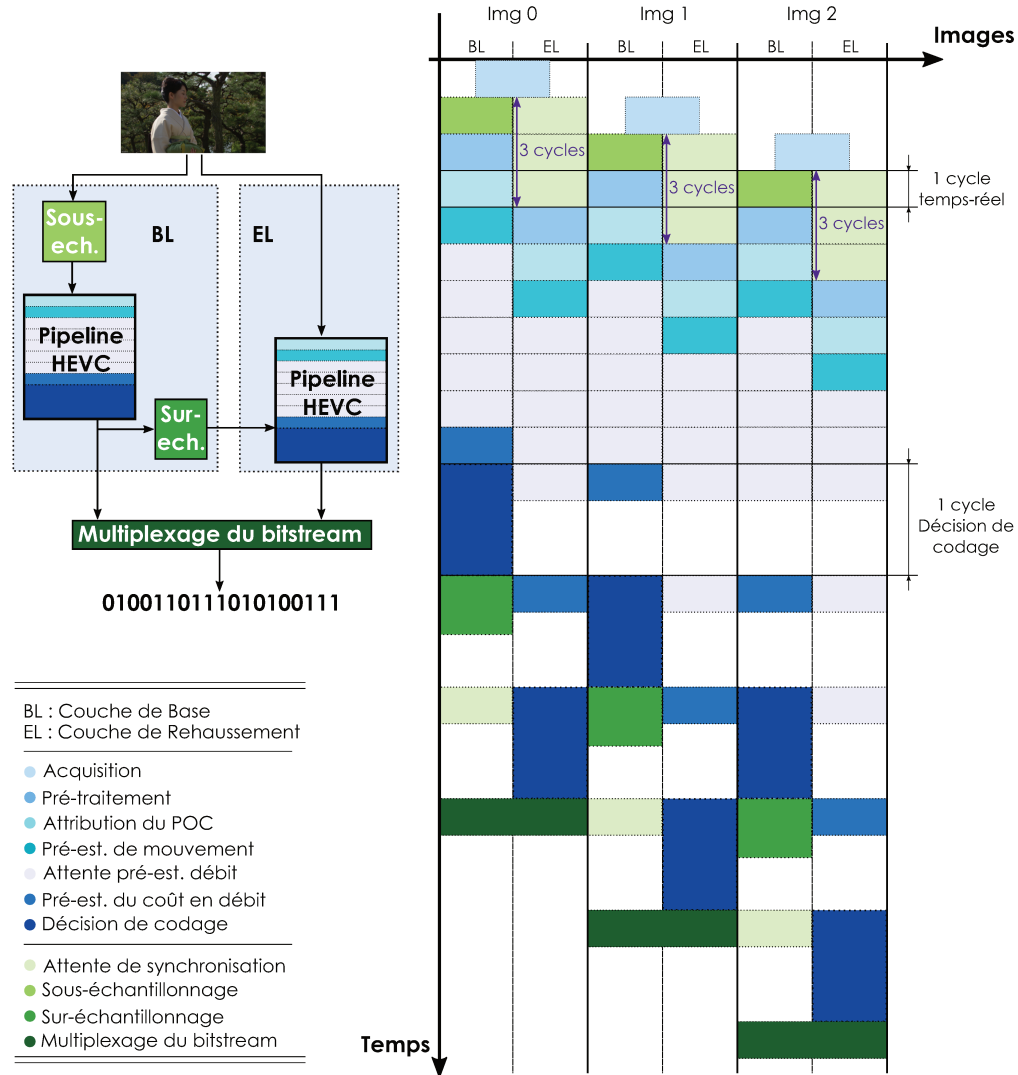


FIGURE 3.11 – Schéma du pipeline de codage de l’encodeur scalable avec le sur-échantillonnage sur la couche de base.

ci n’emploie pas de table coefficients mais deux facteurs de multiplication : un pour l’axe horizontal, l’autre pour l’axe vertical. Ces facteurs sont calculés à partir des résolutions des images des couches de base et de rehaussement. Ensuite, de la même manière que pour le sur-échantillonnage sur les textures, la position du vecteur de mouvement dans la couche de base est calculée à partir de sa position dans la couche de rehaussement. Ce calcul est aussi standardisé dans l’extension SHVC [60]. Afin de pouvoir réaliser l’opération de sur-échantillonnage sur les vecteurs de mouvement, il est nécessaire de disposer de l’ensemble de ces vecteurs. Dans le codeur SHM, les vecteurs de mouvement sont conservés pour chaque pixel jusqu’à ce que l’image ne serve plus de référence temporelle. Dans le SLE, les vecteurs de mouvement qui se destinent à une utilisation sous forme de prédicteur temporel sont conservés directement sous leur forme colocalisée, c’est à dire par bloc de taille 16×16 . L’ensemble des vecteurs de mouvement alors conservés ne suffit pas à l’opération de sur-échantillonnage. Le SHVC-AE emploie donc une sauvegarde d’un ensemble suffisant de ces vecteurs. Dans les codages HEVC et SHVC, la dimension minimum d’un bloc prédit est de taille 4×4 . C’est pourquoi il est possible de se contenter d’une sauvegarde des vecteurs de mouvement par bloc de taille 4×4 . Ainsi, la quantité de données sauvegardées correspond à une image de taille 4 fois moins large et 4 fois moins haute que l’image source. A partir de ces vecteurs, le sur-échantillonnage sur les vecteurs de mouvement est exécuté séquentiellement après le sur-échantillonnage des textures avant de réaliser la prédiction inter-couches.

3.2.3 Prédiction inter-couches

La prédiction inter-couches du standard SHVC peut faire intervenir deux éléments issus de la couche de référence : les vecteurs de mouvement et les informations de texture de l'image reconstruite. Les informations de texture peuvent intervenir dans les différents modes de prédiction inter : Skip, Merge et AMVP en tant que blocs de référence candidats à la prédiction. Les vecteurs de mouvement peuvent être employés en tant que prédicteur temporel, aussi appelé collocalisé. La prédiction inter-couches n'ajoute pas d'outil de codage au standard HEVC qui serait propre à la scalabilité. Elle consiste plutôt à employer les outils de prédiction du standard HEVC sur une ou plusieurs images de référence supplémentaires provenant d'une couche scalable inférieure, aussi appelée couche de référence.

Dans le codeur SHVC-AE, cette prédiction inter-couches emploie une seule référence supplémentaire. Puisque toutes les couches scalables utilisent la même structure de GOP, l'image de référence inter-couches correspond à l'image en cours de codage. Ainsi, les deux couches ont le même aspect visuel, le même POC et le même indice d'image mais appartiennent chacune à des couches différentes.

Cette unique référence supplémentaire représente toutefois un coût important en terme de temps de traitement puisqu'elle augmente non seulement la taille du pipeline avec le sur-échantillonnage, mais aussi le temps de traitement de la décision de codage de la couche de rehaussement. En effet, la référence inter-couches est évaluée comme les autres références, c'est à dire que la référence est employée avec tous les outils de prédiction temporelle, à toutes les granularités (i.e. tailles) de PU et avec toutes les partitions. Le coût de la prédiction inter-couches est ensuite comparé au coût des prédictions des autres références puis des autres modes de prédiction.

La prédiction inter-couches sur les textures consiste alors à employer la référence inter-couches avec des vecteurs de mouvement nuls : (0; 0). Ces vecteurs nuls permettent alors d'employer une prédiction temporelle sur le bloc de même coordonnées spatiales issu de la couche de référence. En plus du vecteur de mouvement, la prédiction temporelle requiert un indice servant à identifier le prédicteur dans sa liste de référence (L0 ou L1). Dans le cas de la prédiction inter-couches, cette indice est utilisé de la même manière et sert à identifier la référence inter-couches (i.e. l'image de référence issue de la couche de base) parmi les autres références de la liste (L0 ou L1). Ensuite, les vecteurs sur-échantillonnés peuvent être employés dans la décision de codage de la couche de rehaussement en temps que prédicteur temporel pour n'importe quelle image de référence contenu dans la liste (L0 ou L1). Toutefois, le standard SHVC indique qu'une référence inter-couches doit être utilisée en tant que référence à long-terme. Dans le codage HEVC, la principale différence entre les références à long-termes et à court-termes réside dans l'utilisation de leurs vecteurs de mouvement en tant que prédicteurs temporels. Les vecteurs de mouvement des références à court-termes peuvent être re-dimensionnés tandis que ceux des références à long-termes ne doivent jamais l'être. Le SLE n'emploie pas de référence à long-terme et n'a pas de mécanisme de prédiction permettant de le prendre en compte. C'est pourquoi il est nécessaire d'assurer la gestion de ce type de référence dans la décision de codage du SHVC-AE. Le standard HEVC autorise l'emploi d'un vecteur collocalisé (i.e. prédicteur temporel) si :

- le bloc 16×16 servant de référence collocalisée n'est pas un bloc intra,
- un vecteur est présent sur la liste des références collocalisées (L0 ou L1),
- l'image de référence et l'image de référence collocalisée sont ou ne sont pas toutes deux des références à long-termes.

Cette dernière condition a été ajoutée dans le SHVC-AE pour les besoins d'utilisation des vecteurs de mouvement issus de la couche de base.

Afin d'accélérer la vitesse de traitement dans la décision de codage, il est possible de sélectionner un certain nombre d'outils de prédiction, de n'employer que certaines tailles de CU et de partitions. L'efficacité de codage est affectée par ces modifications mais le compromis entre vitesse de traitement et qualité reste intéressant. Cette méthode est connue et fréquemment utilisée dans les codeurs vidéo. Par exemple, les codeurs x265 [62] et Kvaazaar [45] définissent des tables qui listent les différents outils employés dans la décision de codage. Toutefois, afin d'accélérer la vitesse de traitement de la prédiction inter-couches, il a été choisi de réaliser un compromis non pas sur la décision de codage mais sur les types d'images servant de références inter-couches.

3.3 Conduites des tests et outils de validation

Les diverses expérimentations qui vont suivre ont été menées en respectant les "Common Test Condition" (CTC) [33]. Les séquences considérées ainsi que les différents Paramètre de Quantification ("Quantization Para-

TABLE 3.4 – Caractéristiques des séquences considérées.

Séquences	Nom	Résolution	Fréquence (IPS)	Nombres d'images
A1	Traffic	2560×1600	30	150
A2	PeopleOnStreet	2560×1600	30	150
B1	Kimono1	1920×1080	24	240
B2	ParksScene	1920×1080	24	240
B3	Cactus	1920×1080	50	500
B4	BasketballDrive	1920×1080	50	500
B5	BQTerrace	1920×1080	60	600

TABLE 3.5 – Couples des QP employés dans les encodages sur les couches de base (BL) et de rehaussement (EL) issus des CTC [33].

QP de la couche de base	22	26	30	34	22	26	30	34
QP de la couche de rehaussement	22	26	30	34	24	28	32	36

meter") (QP) employés sont respectivement présentés dans les Tables 3.4 et 3.5. Les codeurs SHVC sont employés avec en scalabilité spatiale $2\times$, c'est à dire que les couches de base réalisent un codage sur des images deux fois moins larges et deux fois moins hautes que les couches de rehaussement. Tous les codeurs SHVC considérés reçoivent deux sources qui correspondent aux couches de base et de rehaussement. Ces sources sont issues du pré-sous-échantillonnage proposé par le SHM. Le SHVC-AE réalise le sur-échantillonnage sur la couche de base. La configuration de ce codeur est donc celle présentée en Figure 3.12. Afin d'évaluer l'efficacité de codage d'un codeur par rapport à un autre, la métrique BD-BR est employée. Le gain en terme d'efficacité de codage scalable est présenté en comparant les mesures de débit et de niveaux PSNR de la couche de rehaussement à celles d'un codage HEVC équivalent (i.e. à même résolution et sous les mêmes conditions de tests). Dans ce cas, les encodages HEVC ou mono-couche servent de référence dans la métrique BD-BR. Les accélérations sont, quant à elles, présentées sous forme du rapport des vitesses de codage. Les résultats d'encodages présentés dans ce chapitre ont tous été obtenus sur la plateforme Intel Xeon E5-4627V2. Les caractéristiques de cette plateforme sont présentées dans la Table 3.6 Les codeurs SHM et Kvazaar-SHVC utilisent un seul cœur. Le codeur SHVC-AE met toutes les étapes de pipeline et toutes les tâches de traitement en concurrence sur les 32 cœurs disponibles.

TABLE 3.6 – Caractéristiques de la plateforme employée pour les encodages de ce chapitre.

Caractéristiques	E5-4627V2
Nombre de CPU	4
Nombre de cœurs par CPU	8
Fréquence des cœurs	3.30 GHz
Taille de cache par cœur	16 Mo
Mémoire RAM partagée entre les cœurs	64 Go
Compatible SSE2	oui

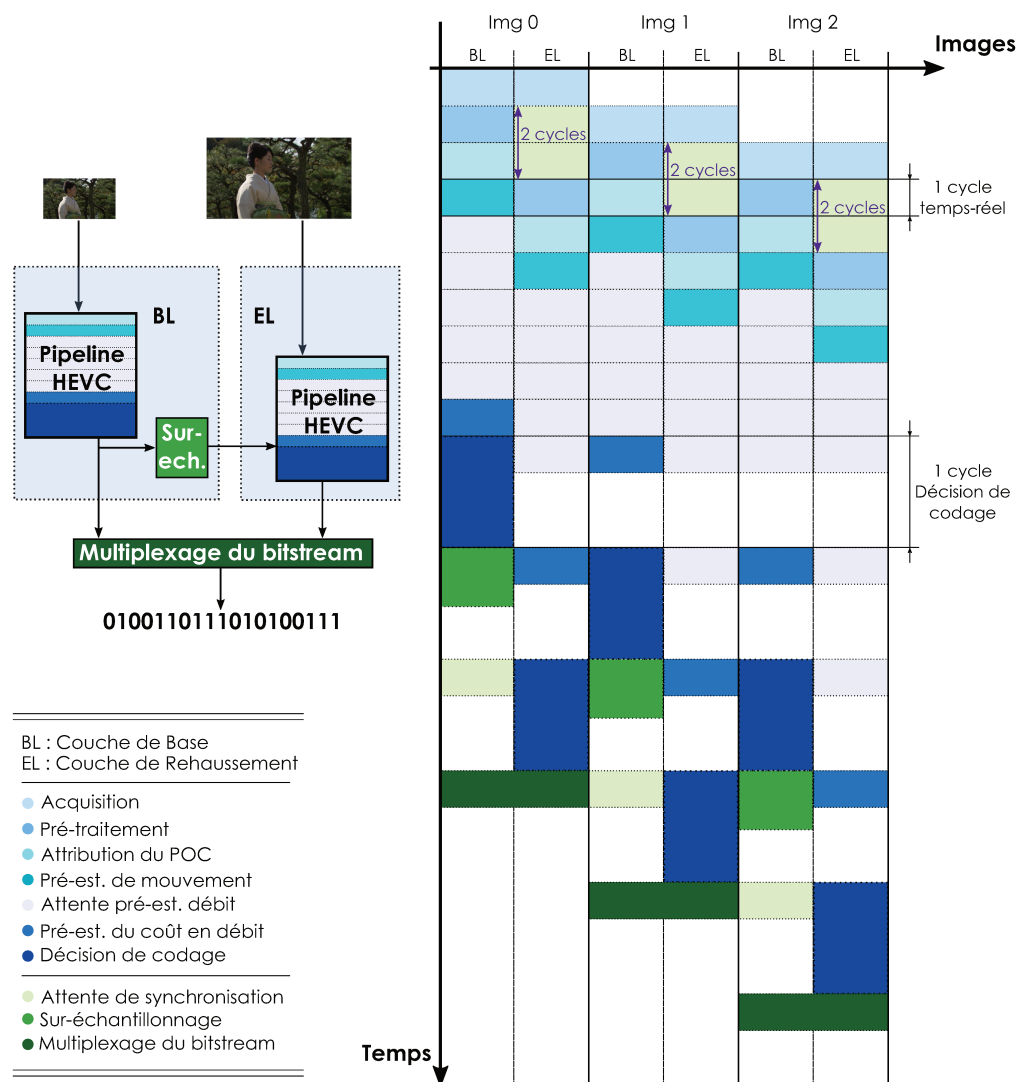


FIGURE 3.12 – Schéma du pipeline de codage du SHVC-AE employé durant les tests. La couche de base acquiert la source pré-sous-échantillonnée et ne réalise pas de sous-échantillonnage. Le sur-échantillonnage est effectué sur la couche de base.

TABLE 3.7 – Gains en débit de SHM par rapport à HM avec une prédiction inter-couches (ILP) sur les slices I, P et B (I-P-B), sur les slices I et P (I-P) et sur les slices I seulement en configuration RA.

Séquences	Gains débit ILP sur I-P-B-slices	Gains débit ILP sur I-P-slices	Gains débit ILP sur I-slices
A1	-26.8%	-24.1%	-18.0%
A2	-45.1%	-20.6%	-7.7%
B1	-42.7%	-29.34%	-14.5%
B2	-27.7%	-23.0%	-15.8%
B3	-28.0%	-18.6%	-6.6%
B4	-31.5%	-14.6%	-2.9%
B5	-8.2%	-7.0%	-1.9%
Moyenne	-30.0%	-19.61%	-9.63%
Pourcentage gains possibles	100%	69.38%	33.03%

3.4 Application de la prédiction inter-couches à différents types de slice

La prédiction inter-couches dans l'extension SHVC emploie dans sa prédiction temporelle des références supplémentaires issues de la couche de base. Le codage de ces références est basé sur le codage HEVC avec différents outils susceptibles d'être employés en fonction du type de l'image (I, P ou B). Ces images de référence inter-couches proposent donc des informations de codage soit exclusivement basées sur une prédiction spatiale, soit basées sur des prédictions spatiales et temporelles. Les informations de codage employées dans la prédiction inter-couches sont donc variées d'un type d'image à un autre. C'est pourquoi une analyse sur la prédiction inter-couches au niveau de ces types d'images est intéressante.

3.4.1 Analyse SHM des gains inter-couches en fonction du type de slice.

Parmi les 3 types de slice I, P et B, chacun peut servir de référence pour la prédiction inter-couches. Il est donc également envisageable de désactiver la prédiction inter-couches sur certains types de slice. L'intérêt de ce procédé est de réduire la complexité du codeur SHVC tout en limitant la perte en débit/distorsion. En effet, la scalabilité spatiale introduit non seulement la mécanique de prédiction inter-couches, mais aussi les opérations de sous-échantillonnage et de sur-échantillonnage. La prédiction inter-couches introduit une image de référence supplémentaire, allongeant ainsi le temps de traitement de la décision de codage. Les deux filtres, quant à eux, interviennent sur des données non compressées :

- l'image initiale pour le sous-échantillonnage,
- l'image reconstruite et les vecteurs de mouvement issus de la couche de base pour le sur-échantillonnage.

L'introduction de ces filtres augmente davantage le temps de traitement d'une image. Ainsi cette double complexité supplémentaire peut être réduite en désactivant à la fois la prédiction inter-couches et l'opération de sur-échantillonnage sur certaines slices. La Table 3.7 présente les gains en débit obtenus avec le codeur SHM en activant la prédiction inter-couches sur toutes les slices, sur les slices I et P seulement et sur les slices I seulement en configuration RA. On constate que les slices I, P et B sont chacune responsables d'environ un tiers du total de gain en débit possible apporté par la prédiction inter-couches. Ces résultats peuvent être pondérés par le nombre d'un même type de slice présent dans un flux codé.

La Table 3.8 présente la part de chaque type de slice selon les configurations de GOP et les fréquences d'images. Dans les conditions de tests issues des CTC [33], on remarque que les slices de type B représentent une part du nombre total d'images bien plus importante que les slices de type I ou P. En particulier, dans la configuration RA qui est largement employée en diffusion, l'emploi de la prédiction inter-couches serait responsable d'une importante perte de performances sur la vitesse de traitement.

Il est alors envisageable de réaliser un compromis entre le gain en débit de la prédiction inter-couches et la vitesse de codage au niveau des types de slice en configuration RA. Les configurations de GOP all-intra (ou intra-only) (AI) et low-delay P (LD-P) serviront de cas d'étude permettant d'isoler certains types de slice : d'abord I puis I et P. Il est également possible de réaliser un compromis entre le gain en débit de la prédiction inter-couches et la vitesse de codage au niveau des types de slice pour les autres configurations (AI, LD-P et LD-B).

TABLE 3.8 – Pourcentages de présence des différents types de slice par configuration issus des CTC [33] pour des séquences de 10 secondes codées en HEVC mono-couche

Configuration de GOP	Pourcentage d'I-slices	Pourcentage de P-slices	Pourcentage de B-slices
RA 24 fps	4.17%	8.33%	87.50%
RA 30 fps	3.33%	9.33%	87.33%
RA 50 fps	2.20%	10.40%	87.40%
RA 60 fps	1.67%	10.83%	87.50%
LD-P 24 fps	25%	75%	-
LD-P 30 fps	25.33%	74.67%	-
LD-P 50 fps	25%	75%	-
LD-P 60 fps	25%	75%	-
AI 24 fps	100%	-	-
AI 30 fps	100%	-	-
AI 50 fps	100%	-	-
AI 60 fps	100%	-	-
LD-B 24 fps	25%	-	75%
LD-B 30 fps	25.33%	-	74.67%
LD-B 50 fps	25%	-	75%
LD-B 60 fps	25%	-	75%
Moyenne quand présent	38.25%	42.32%	81.18%

3.4.2 Application aux références I-slices

Les I-slices n'exploitent que la corrélation spatiale dans leurs prédictions. Utilisées en tant que référence de prédiction, elles ne peuvent alors fournir que l'information de texture, c'est-à-dire que seule l'image reconstruite est employée dans cette prédiction. Par conséquent, pour la prédiction inter-couches à partir d'une référence I-slice, seule la texture peut être prise en compte dans la décision de codage. En terme de complexité, l'utilisation des I-slices en tant que références inter-couches se traduit par un sur-échantillonnage sur les textures ainsi que l'ajout d'une référence sur les images de la couche de rehaussement. Puisque les structures de GOP sont alignées entre les couches de base et de rehaussement, l'utilisation d'une référence inter-couches de type I impose à la couche de rehaussement de signaler son image en type P. En effet, cette image sur la couche de rehaussement dispose alors d'au moins une référence temporelle et ne peut plus être de type I. Pour cela il est nécessaire d'adapter la construction du GOP de la couche de rehaussement en précisant que les slices employant une référence de type I doivent être signalées en slice de type P. Ensuite, il faut ajouter l'ensemble du mécanisme de prédiction temporelle à ces slices afin de pouvoir utiliser la prédiction inter-couches. Finalement, il faut aussi comparer les coûts de la prédiction intra et de la prédiction inter-couches sur chaque granularité de PU et pour chaque partition.

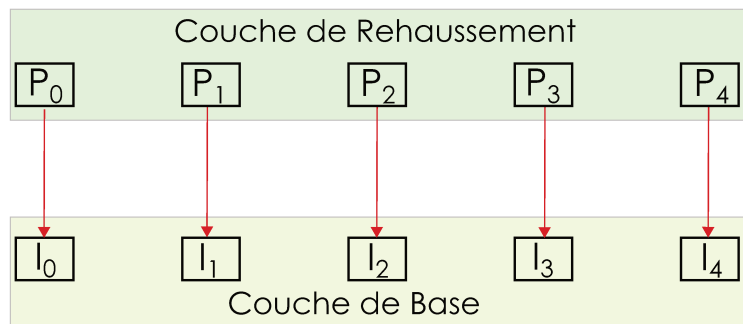


FIGURE 3.13 – Dépendances de GOP en AI du codeur SHVC-AE

Le processus de prédiction employé est alors le même que pour une prédiction temporelle classique issue du standard HEVC, à ceci près que les vecteurs de mouvement ne sont pas calculés. En effet, le vecteur de la

TABLE 3.9 – Comparaison BD-BR des codeurs SHVC par rapport aux codeurs HEVC sur la couche de rehaussement équivalente en configuration AI

Séquences	BD-BR SHVC-AE vs SLE	BD-BR SHM vs HM	BD-BR Kvazaar-SHVC vs Kvazaar HEVC	BD-BR SHVC-AE vs SHM
A1	-42.6%	-40.2%	-17.2%	3.9%
A2	-46.7%	-44.1%	-18.8%	3.2%
B1	-50.5%	-48.7%	-20.7%	7.6%
B2	-35.8%	-34.2%	-15.0%	5.1%
B3	-31.9%	-31.1%	-11.7%	7.1%
B4	-26.1%	-25.7%	-7.5%	10.7%
B5	-18.4%	-17.6%	-5.5%	7.5%
Moyenne	-36.0%	-34.5%	-13.8%	6.4%

prédiction inter-couches sur les textures est défini à (0; 0) afin de faire correspondre le bloc PU courant avec le bloc de référence sur-échantillonné à la même position.

Afin de caractériser le gain de la prédiction inter-couches, nous utilisons dans nos tests la structure de GOP AI qui ne fait intervenir que des slices de type I sur la couche de base comme illustré sur la Figure 3.13. La Table 3.9 présente les gains de la prédiction inter-couches des différents codeurs SHVC employés dans une configuration AI en scalabilité spatiale avec des images sur la couche de base deux fois moins haute et deux fois moins large que sur la couche de rehaussement. Ces expérimentations conduites dans [63] montrent que le SHVC-AE propose des gains équivalents au codeur de référence SHM. Cette expérimentation permet d'affirmer que la prédiction inter-couches employant les textures est équivalente à celle de l'état de l'art. On constate également que le codeur Kvazaar, encore en cours de développement, propose des gains de prédiction inter-couches moins importants.

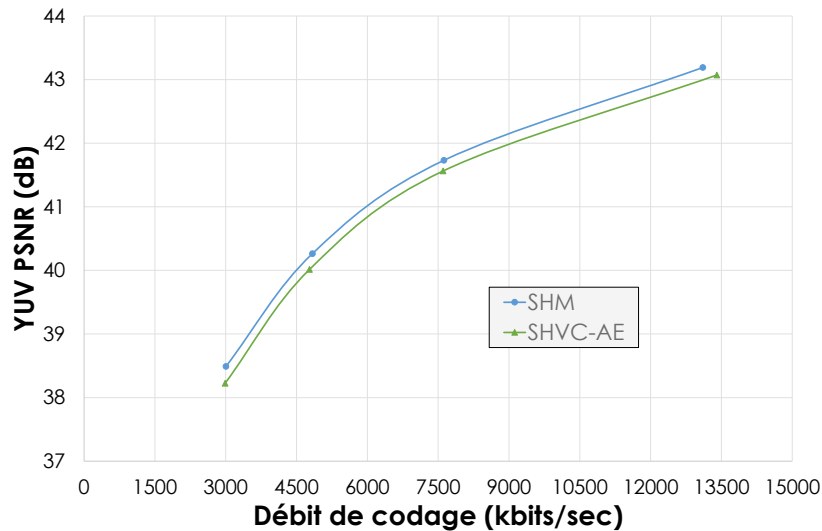


FIGURE 3.14 – Courbes débit/distorsion des codeurs SHM et SHVC-AE sur la séquence B1 en configuration AI.

En terme d'efficacité de codage, la Table 3.9 présente dans sa dernière colonne le rapport BD-BR des encodages de SHVC-AE par rapport au SHM. De plus, la Figure 3.14 illustre l'écart entre ces codeurs avec les courbes débit/distorsion des codeurs SHVC-AE et SHM sur la séquence B3. On constate que cet écart est faible mais que l'efficacité de codage est meilleure sur le SHM. Ceci s'explique par les choix réalisés dans la décision de codage. En effet, la décision de codage sur les images de type I est à peu près équivalente entre le SHVC-AE et

TABLE 3.10 – Rapport des vitesses de codage SHVC-AE / SLE et SHVC-AE / SHM en configuration AI

Séquences	Rapport SHVC-AE / SLE	Rapport SHVC-AE / SHM
A1	0.43	51.9
A2	0.39	52.8
B1	0.34	26.0
B2	0.40	39.5
B3	0.46	31.0
B4	0.55	30.4
B5	0.51	31.1
Moyenne	0.44	37.5

le SHM. Néanmoins, la décision de codage HEVC du codeur SHVC-AE emploie seulement 19 directions pour la prédiction intra et ne réalise pas de décision inter sur les partitions $2N \times nD$, $2n \times nU$, $nL \times 2N$ et $nR \times 2N$. Ce sont principalement ces choix de prédiction qui sont responsables de l'écart entre le SHM et le SHVC-AE en configuration AI. Toutefois, ce compromis sur la prédiction HEVC permet au SLE et au SHVC-AE de réaliser des encodages beaucoup plus rapides.

Afin d'évaluer le coût de la prédiction inter-couches en terme de vitesse de codage, la Table 3.10 présente le rapport des vitesses de codage des codeurs SHVC-AE sur SLE en configuration AI. On constate que ce coût est important dans cette configuration puisque le codage scalable est ralenti, en moyenne, de 44% par rapport à la vitesse de codage du SLE. Ce coût s'explique non seulement par l'ajout du mécanisme de prédiction temporelle sur les références inter-couches, mais aussi par l'ajout de l'opération de sur-échantillonnage de la couche de base. En effet, l'ajout d'une image référence augmente significativement le temps de traitement de la décision de codage. Ensuite, l'opération de sur-échantillonnage parcourt l'ensemble des pixels d'une image au format de la couche de rehaussement. Non seulement cette opération requiert un important temps de traitement, mais elle est également exécutée sur toutes les images reconstruites de la couche de base dans la configuration AI. La troisième colonne de la Table 3.10 donne le rapport entre temps d'encodage SHM et SHVC-AE, qui est en moyenne 37 fois plus rapide que le SHM. Cette accélération est due aux multiples parallélismes et aux optimisations bas-niveau présents dans le SHVC-AE employé avec la plateforme considérée (voir Table 3.6). Les 2 couches spatiales du SHVC-AE sont mises en concurrence sur les 32 cœurs tandis que le SHM réalise un traitement sur un cœur uniquement.

Les résultats obtenus dans la configuration AI ne permettent pas d'identifier si l'ajout des opérations de sur-échantillonnage et de la prédiction temporelle à partir d'images de référence inter-couches de type I sont responsables dans des proportions équivalentes ou inégales de la perte de vitesse. Les tests en configuration LD-P présentés dans la section suivante permettent d'affirmer que c'est l'ajout de la prédiction temporelle qui impacte le plus la vitesse de codage.

3.4.3 Application aux références P-slices

L'utilisation des P-slices en tant que référence permet d'exploiter les informations de texture de l'image reconstruite ainsi que les vecteurs de mouvement employés dans la prédiction temporelle. Afin de pouvoir exploiter ces informations dans la prédiction inter-couches, il est là aussi nécessaire de sur-échantillonner ces données. Les vecteurs de mouvement sur-échantillonnés sont alors stockés par bloc de taille 16×16 afin d'être employés en tant que prédicteurs collocalisés. De plus, puisque les P-slices disposent déjà d'une première référence provenant de la même couche, il est possible d'employer la référence inter-couches afin de réaliser une bi-prédiction. Dans ce cas, une image de la couche de rehaussement employant une référence inter-couches de type P devra être signalée en tant que B-slice comme illustré sur la Figure 3.15. Le SHVC-AE réalise cette bi-prédiction et placent la référence inter-couches à la fois sur la liste L0 et sur la liste L1. De plus, le mécanisme de bi-prédiction est ajouté aux outils de codage. Enfin, les vecteurs collocalisés sont autorisées sur les couches de base et de rehaussement mais n'utilisent que les références temporelles issues de la même couche. Le sur-échantillonnage et l'utilisation des vecteurs de mouvement issus de la couche de base ajoute un temps de traitement et une occupation mémoire non négligeable. Le fait de ne pas utiliser ces vecteurs de mouvement constitue un compromis qui pénalise l'efficacité de codage mais qui bénéficie à la fois à la vitesse de traitement et à l'occupation mémoire.

Afin d'évaluer ce compromis sur la prédiction inter-couches, nous utilisons la structure de GOP LD-P qui ne

fait intervenir que des images de type I et P comme le montre le schéma en Figure 3.15. La Table 3.11 présente l'amélioration du codage de la couche de rehaussement par rapport à un codage HEVC équivalent. On constate que le SHVC-AE ne permet pas d'obtenir la même efficacité de codage que le SHM. Néanmoins, les gains de la prédiction inter-couches demeurent intéressants puisqu'ils permettent d'économiser en moyenne 33.3% du débit à qualité constante. Dans [64], nous proposons un autre compromis sur les outils de codage : la prédiction inter-couches employait uniquement les informations de textures. De plus, les vecteurs collocalisés sur :

- la couche de base étaient utilisés dans la décision,
- la couche de rehaussement, qu'ils proviennent d'une référence temporelle HEVC (i.e. même couche) ou bien des vecteurs de mouvement issus de la couche de base, n'étaient pas utilisés.

Ces expérimentations, présentées dans la deuxième colonne (0-Col) de la Table 3.11, ont permis de montrer qu'un compromis sur les outils de codage au niveau de la prédiction inter-couches permet malgré tout de conserver une amélioration significative du codage de la couche de rehaussement par rapport à un codage HEVC équivalent : 27.3% du débit économisé par rapport au codage HEVC équivalent en moyenne.

Dans la Table 3.11, les vecteurs collocalisés issus de la même couche sont autorisés pour les colonnes H-Col, ce qui améliore l'efficacité de codage de la couche de rehaussement par rapport aux compromis proposés dans [64]. En revanche, les vecteurs de mouvement issus de la couche de base ne sont pas employés ici comme illustré sur la Figure 3.16. La différence en terme de gains inter-couches (colonnes 3 et 4) entre le SHM et le SHVC-AE repose sur les compromis effectués dans la décision de codage du SLE ainsi que sa version datant de 2015. En effet, la prédiction inter-couches utilise les mécanismes de prédiction temporelle issus du standard HEVC. Ceux-ci, d'une part, sont limités par l'absence des partitions asymétriques et, d'autre part, ne disposent pas des améliorations récentes sur la prédiction temporelle du SLE. Enfin, le codeur scalable Kvazaar étant en cours de développement, et notamment sur la prédiction temporelle, la prédiction inter-couches n'est pas efficace dans la configuration LD-P.

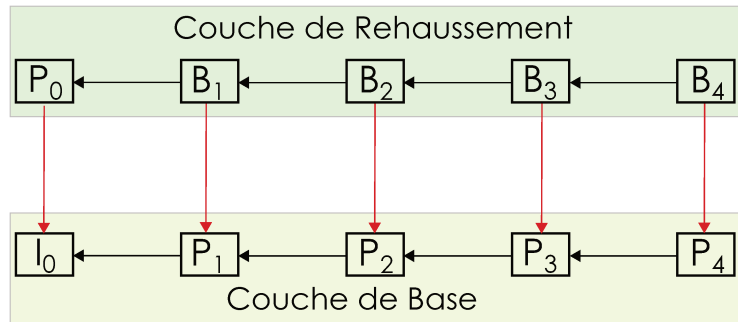


FIGURE 3.15 – Dépendances de GOP en LD-P du codeur SHVC-AE

En terme d'efficacité de codage, la Table 3.11 présente en sa dernière colonne le rapport BD-BR des encodages de SHVC-AE par rapport au SHM en configuration LD-P. La Figure 3.18 illustre cette même différence avec les courbes débit/distorsion des codeurs SHVC-AE et SHM sur la séquence B4. L'écart entre les deux solutions est plus important qu'en configuration AI. Cette augmentation s'explique par les compromis réalisés dans la décision de codage et par la version du SLE datant de 2015 et ne fournissant pas une efficacité de codage HEVC équivalente au HM. La Table 3.3 montre cet écart. Le codage de la couche de base du SHVC-AE étant moins efficace que celui du SHM, la prédiction inter-couches bénéficie moins efficacement des informations de codage qui lui sont fournies.

La Figure 3.17 schématise la prédiction inter-couches du SHVC-AE faisant intervenir les vecteurs de mouvement issus de la couche de base. La Table 3.12 présente une comparaison BD-BR des résultats de codage sur les couches de rehaussement du SHVC-AE par rapport au SLE avec et sans l'utilisation des vecteurs de mouvement issus de la couche de base. On constate que l'utilisation de ces vecteurs de mouvement dans la prédiction inter-couches du SHVC-AE est moins efficace que le codage avec une prédiction inter-couches basée uniquement sur les textures. Dans le SHVC-AE, les prédicteurs temporels sont donc plus intéressants quand ils sont issus de la même couche plutôt que d'une autre. Une explication pourrait venir de la gestion des tables de quantification employé dans le SHVC-AE. Dans le SLE, les QP par CU ne sont pas calculés mais lus dans une table de quantification. Cette table de quantification permet d'accélérer la décision de codage plutôt que de calculer le QP pour chaque CU. Elle prend en compte l'éloignement de l'image de référence. Le SHM gère cette notion d'éloignement pour la prédiction inter-couches mais ce n'est pas le cas du SHVC-AE. C'est pourquoi la décision de codage employant les

TABLE 3.11 – Comparaison BD-BR des codeurs SHVC par rapport aux codeurs HEVC sur la couche de rehaussement équivalente en configuration LD-P. Les prédictions inter-couches emploient seulement les textures des codeurs SHM, SHVC-AE et Kvazaar. 0-Col : les vecteurs collocalisés ne sont pas employés sur la couche de rehaussement. H-Col : les vecteurs collocalisés issus de la même couche (prédiction HEVC) sont autorisés sur toutes les couches.

Séquences	BD-BR SHVC-AE vs SLE		BD-BR SHM vs HM	BD-BR Kvazaar-SHVC vs BD-BR Kvazaar HEVC	BD-BR SHVC-AE vs SHM
	0-Col	H-Col	H-Col	H-Col	H-Col
	A1	-31.1%	-39.0%	-69.3%	20.4%
A2	-36.5%	-41.4%	-49.1%	-24.9%	21.0%
B1	-38.5%	-44.1%	-54.9%	-22.0%	31.9%
B2	-25.6%	-33.1%	-59.8%	33.1%	84.2%
B3	-24.1%	-30.2%	-59.2%	14.9%	93.7%
B4	-21.2%	-25.7%	-38.1%	-0.5%	30.4%
B5	-13.8%	-19.3%	-44.5%	23.5%	101.9%
Moyenne	-27.3%	-33.3%	-53.6%	6.4%	69.6%

TABLE 3.12 – Comparaison BD-BR des codages SHVC-AE par rapport au SLE sur la couche de rehaussement avec (Col-ILP) et sans (NoCol-ILP) les vecteurs de mouvement issus de la couche de base (BL) en configuration LD-P

Séquences	BD-BR SHVC-AE vs SLE		BD-BR SHVC-AE Col-ILP vs NoCol-ILP
	NoCol-ILP	Col-ILP	
	A1	-39.0%	
A2	-41.4%	-36.9%	8.7%
B1	-44.1%	-38.8%	10.5%
B2	-33.1%	-26.2%	11.5%
B3	-30.2%	-24.7%	8.9%
B4	-25.7%	-23.4%	3.6%
B5	-19.3%	-17.7%	2.6%
Moyenne	-33.3%	-28.5%	8.4%

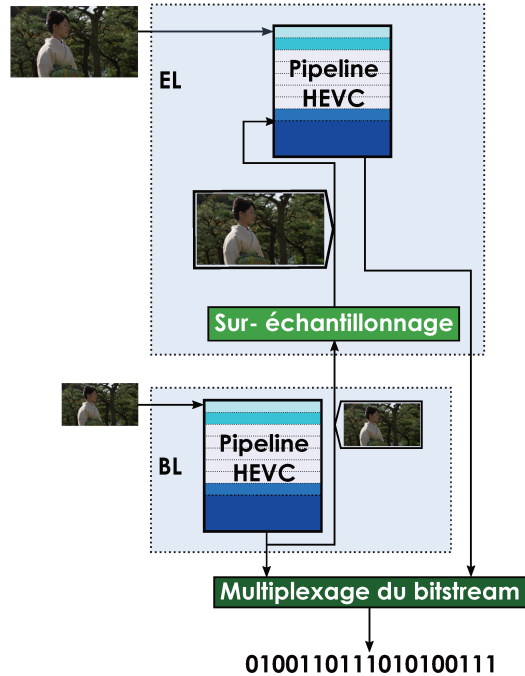


FIGURE 3.16 – Schéma du SHVC-AE sans l'utilisation des vecteurs de mouvement issus de la couche de base (BL) dans la couche de rehaussement (EL).

vecteurs de mouvement issus de la couche de base est moins efficace que sans ces vecteurs de mouvement. Cette incohérence du point de vue des outils de codage nécessiterait une étude plus approfondie sur l'estimation du coût des prédicteurs inter-couches.

De plus, le SHM recommande de ne pas employer les vecteurs de mouvement issus de la couche de base en tant que prédicteur temporel dans la couche de rehaussement. En effet, lors de plusieurs études [65, 66, 67] durant le processus de standardisation de l'extension SHVC sur cet outil de prédiction, le groupe JCT-VC a conclu que l'amélioration en débit ne valait pas le coût en vitesse et en mémoire. Par défaut, le SHM n'emploie pas les vecteurs de mouvement issus de la couche de base dans la décision de codage de la couche de rehaussement.

Pour toutes ces raisons, il est préférable de ne pas employer les vecteurs de mouvement issus de la couche de base dans le SHVC-AE en configuration LD-P. Si le compromis réalisé sur les slices de type P réduit l'efficacité de codage par rapport au SHM, la vitesse de traitement est, quant à elle, bien supérieure.

La Table 3.13 présente le rapport des vitesses de codage du SHVC-AE par rapport au SLE en configuration LD-P. Les couches de base et de rehaussement sont mises en concurrence sur les cœurs de la plateforme employée. On constate que le coût de la prédiction inter-couches sur les images P compense très largement celui des images

TABLE 3.13 – Rapport des vitesses de codage SHVC-AE / SLE et SHVC-AE / SHM en configuration LD-P avec le sur-échantillonnage sur les textures et les vecteurs de mouvement issus de la couche de base.

Séquences	Rapport SHVC-AE / SLE	Rapport SHVC-AE / SHM
A1	0.88	63.9
A2	0.89	91.1
B1	0.94	64.6
B2	0.88	59.4
B3	0.91	62.0
B4	0.92	67.0
B5	0.90	60.1
Moyenne	0.90	66.9

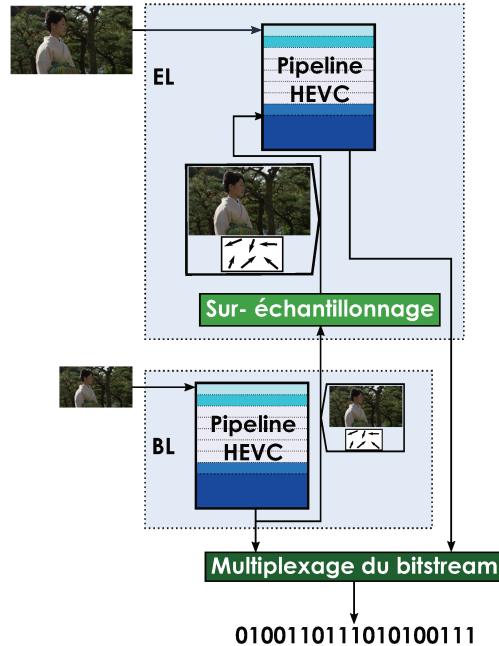


FIGURE 3.17 – Schéma du SHVC-AE avec l'utilisation des vecteurs de mouvement issus de la couche de base (BL) dans la couche de rehaussement (EL).

I. En effet, le temps de traitement du SHVC-AE en configuration LD-P est beaucoup plus proche de celui du SLE. Cela s'explique par la présence moins importante d'images I. De plus, de façon identique à la configuration AI, toutes les images reconstruites de la couche de base sont sur-échantillonnées dans la configuration LD-P. Chaque image de la couche de rehaussement ajoute également une référence inter-couches aux références déjà présentes. Le temps de traitement de sur-échantillonnage dépend essentiellement de la quantité d'information à traiter (résolution, profondeur des données...) Le type d'image n'a donc aucune influence sur ce dernier. On peut donc conclure que l'ajout du sur-échantillonnage, ainsi que de la référence inter-couches dans une décision temporelle, impacte certes la vitesse de codage, mais bien moins significativement que l'ajout d'une prédiction temporelle sur les images employant une référence inter-couches de type I. Puisque la configuration LD-P emploie, sur sa couche de base, moins d'images de type I que la configuration AI et que l'écart avec le SLE est moins important, on peut déduire que la prédiction inter-couches employant des images de référence de type P est moins intrusive. Dans ce cas, on peut déjà avancer que l'ajout d'une prédiction inter-couches employant des images de type B aurait un impact similaire à l'ajout sur les références de type P. La seconde colonne de la Table 3.13 présente le rapport des vitesses de codage entre le SHVC-AE et le SHM. La vitesse de codage du SHVC-AE est bien supérieure notamment grâce à l'utilisation du threading et des opérations bas-niveau sur la plateforme considérée (voir Table 3.6). La section suivante présente des résultats en configuration RA et permet de vérifier l'impact de la prédiction inter-couches sur les images de type B.

3.4.4 Application aux B-slices

La configuration RA est une configuration communément employée dans la diffusion de vidéos. Elle fait intervenir majoritairement les images de type B. Dans le standard HEVC, ces images peuvent utiliser la prédiction intra, la prédiction inter ainsi que la bi-prédiction. Ainsi, le temps de traitement accordé à la prédiction est plus important que pour les images I et P. Par conséquent, elles bénéficient d'une meilleure corrélation temporelle et coûtent généralement moins chers en débit pour une complexité supérieure. L'ajout de la prédiction inter-couches sur ces images, dans la couche de rehaussement, augmenterait encore davantage la complexité de leur codage. De plus, les images B sont très majoritairement présentes dans le bitstream en configuration RA, voir Table 3.8, et bénéficient unitairement bien moins efficacement de la prédiction inter-couches. En effet, comme vu dans la section 3.4.1, les images employant une référence inter-couches de type B sont très largement majoritaires, de 87.33%

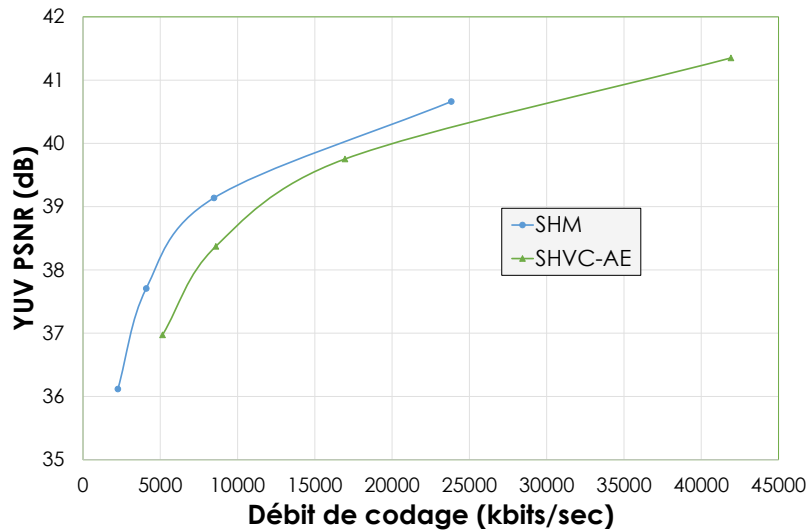


FIGURE 3.18 – Courbes débit/distorsion des codeurs SHM et SHVC-AE sur la séquence B4 en configuration LD-P.

à 87.5% en configuration RA et bénéficient d'un tiers de l'amélioration issue de la prédiction inter-couches (voir Table 3.7). Les références inter-couches de type I et P, quant à elles, bénéficient de la même amélioration, un tiers chacune, mais sont bien moins présentes respectivement de 1.67% à 4.17% et de 8.33% à 10.83% des images d'un flux RA. L'ajout de la prédiction inter-couches employant des références de type B, augmenterait la complexité du codage pour une amélioration unitaire moins importante que sur les images I et P. Enfin, les images de type B de la couche de rehaussement profitent déjà indirectement de la prédiction inter-couches. En effet, les images de type I et P bénéficient de la prédiction inter-couches sur les textures. Leurs images reconstruites sont donc déjà améliorées et améliorent, de facto, les images les employant en tant que référence. C'est pourquoi il a été choisi de réaliser un compromis sur l'emploi de la prédiction inter-couches sur les images de type B. De fait, comme illustré sur la Figure 3.19, la prédiction inter-couches n'est effectuée que sur les images de référence. Celles-ci incluent les images de type I et de type P ainsi que les images de type B qui servent de référence dans une prédiction HEVC. Ainsi, la prédiction inter-couches est effectuée pour une image sur deux dans un bitstream RA.

Afin d'évaluer les gains permis par ce compromis, les tests sont réalisés en configuration RA sur les codeurs HEVC HM et SLE ainsi que sur les codeurs SHVC SHM et SHVC-AE. La Figure 3.20 illustre les courbes débit/distorsion de ces codeurs sur la séquence B2. On constate que le codeur SHVC-AE, en terme d'efficacité de codage, n'est pas au niveau du SHM. Cette différence est principalement due aux compromis réalisés sur la prédiction inter-couches sur les images de type P et B. Toutefois cet écart est également causé par l'efficacité du codage HEVC. En effet, la Table 3.3 montre la différence de codage HEVC entre le SLE et le HM. Cet écart est non seulement présent sur le codage de la couche de base, mais également sur celui de la couche de rehaussement. La couche de rehaussement du SHVC-AE subit alors la double peine du manque d'efficacité de la prédiction HEVC et de la prédiction inter-couches qui emploient une couche de base moins efficace.

La Table 3.14 présente les différences BD-BR des encodages des couches de rehaussement réalisés par les codeurs scalables SHVC-AE et SHM respectivement par rapport aux codeurs HEVC équivalents SLE et HM. L'écart entre les différences BD-BR s'explique par le compromis réalisé sur la prédiction inter-couches. Néanmoins, on peut constater que cet écart n'est pas si important que celui de la configuration LD-P. En effet, le compromis permet malgré tout de réaliser les trois quarts du gain inter-couches théorique. Le quart restant est alors dû à la prédiction inter-couches sur les images de type B ne servant pas de référence et au manque d'efficacité du codage HEVC de la version du SLE datant de 2015.

La dernière colonne de la Table 3.14 présente la différence BD-BR des encodages du SHVC-AE par rapport au SHM. On constate que le SHVC-AE propose une efficacité de codage moins importante que le SHM d'en

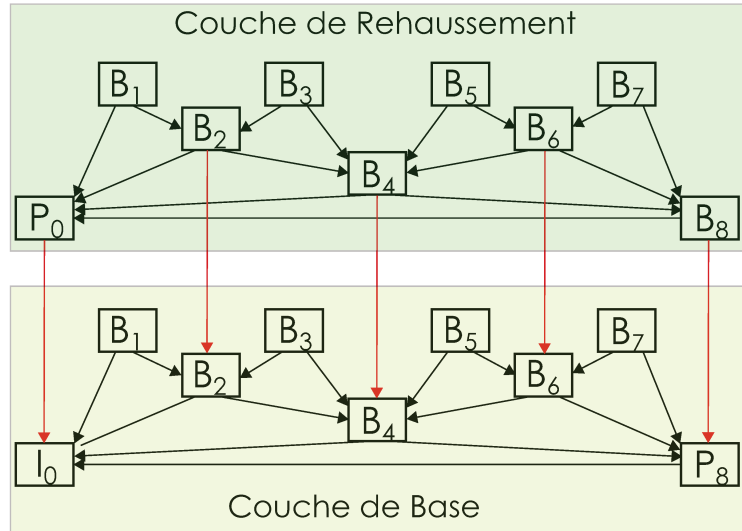


FIGURE 3.19 – Dépendances de GOP en RA du codeur SHVC-AE

TABLE 3.14 – Comparaison BD-BR des codeurs SHVC par rapport aux codeurs HEVC sur la couche de rehaussement équivalente en configuration RA

Séquences	BD-BR SHVC-AE	BD-BR SHM	BD-BR SHVC-AE
	vs SLE	vs HM	vs SHM
A1	-19.1%	-21.7%	31.0%
A2	-27.7%	-36.6%	39.2%
B1	-30.2%	-38.1%	43.4%
B2	-17.1%	-22.5%	33.1%
B3	-15.0%	-22.3%	35.9%
B4	-12.5%	-24.8%	45.8%
B5	-8.2%	-6.3%	59.2%
Moyenne	-18.5%	-24.6%	41.1%

moyenne 41.1%. Cet écart est moins important que dans la configuration LD-P. Il s'explique lui aussi par le manque d'efficacité de codage de la version du SLE, mais également par le compromis réalisé sur les références inter-couches de type B qui ne sont employées que si elles servent également de référence dans le codage HEVC de la couche de base. En revanche, ce compromis permet d'améliorer significativement la vitesse de codage.

La Table 3.15 présente le rapport des vitesses de codage du SHVC-AE par rapport au SLE en configuration RA. On constate que la vitesse de codage du SHVC-AE est très proche de celle du SLE. Ceci s'explique non seulement par l'architecture permettant de traiter les couches de base et de rehaussement simultanément, mais aussi par les compromis réalisés dans la prédiction inter-couches. En effet, la prédiction inter-couches est réalisée une image sur deux dans la configuration RA uniquement sur les informations de textures. Cela permet d'économiser le temps de traitement de la décision de codage de la moitié des images n'employant pas la prédiction inter-couches. L'impact sur les vitesses de codage provoqué par l'ajout d'une prédiction temporelle sur les images employant une référence inter-couches de type I n'est alors plus autant significatif que pour la configuration AI. Le compromis sur la décision de codage est moins intrusif sur les images de référence P et B par rapport aux images de référence I. L'impact de la prédiction inter-couches sur ces images est ici presque insignifiant. La seconde colonne de la Table 3.15 présente le rapport des vitesses de codage entre le SHVC-AE et le SHM. Elle sont en moyenne 66,5 fois plus rapides pour le SHVC-AE qui utilise 32 coeurs contre le SHM qui en utilisent un seulement. Cette accélération est due aux divers parallélismes et optimisations bas-niveau présents dans le SHVC-AE.

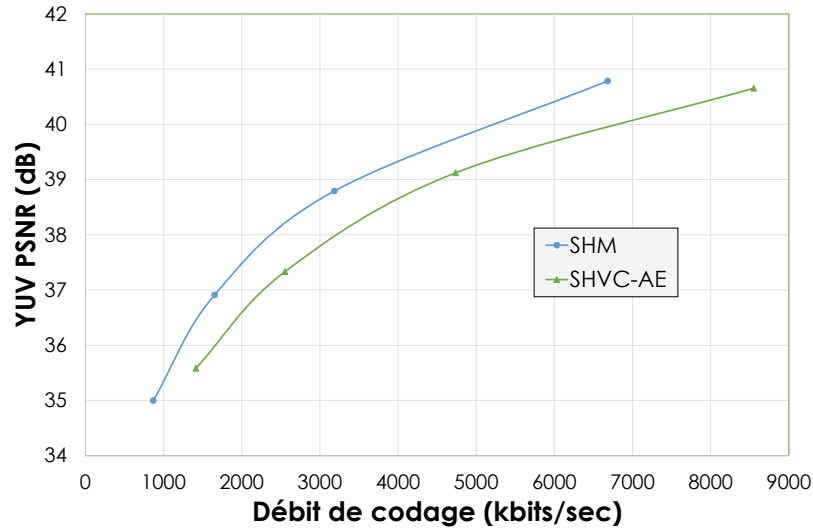


FIGURE 3.20 – Courbes débit/distorsion des codeurs SHM et SHVC-AE sur la séquence B2 en configuration RA.

TABLE 3.15 – Rapport des vitesses de codage SHVC-AE / SLE et SHVC-AE / SHM en configuration RA

Séquences	Rapport SHVC-AE / SLE	Rapport SHVC-AE / SHM
A1	0.96	70.6
A2	0.99	102.0
B1	0.98	60.6
B2	0.97	54.5
B3	0.98	58.7
B4	0.99	62.8
B5	0.98	56.6
Moyenne	0.98	66.5

Conclusion

La méthode présentée dans ce chapitre permet de réaliser des encodages SHVC sur deux couches scalables en résolutions, c'est à dire traitant chacune des images de résolution différentes. La méthode proposée s'appuie sur le codeur SLE développé par Ateame, et notamment sur son architecture en pipeline. Ce permet de réaliser des codages HEVC efficaces bien que la version employée, datant de 2015, ne soit pas au niveau des codeurs HM et x265. Il permet toutefois de bénéficier d'une vitesse de codage supérieure aux autres codeurs grâce à plusieurs niveaux de parallélisme. Le codeur SHVC-AE proposé utilise deux instances de SLE correspondant aux couches scalables. Ces couches sont traitées simultanément grâce à un alignement de leurs étapes de pipeline respectives. Cette alignement requiert toutefois un léger décalage de 3 cycles de pipeline minimum. La méthode proposée permet de tirer profit des différentes outils ainsi que des performances temps-réel du SLE.

De plus, les différents compromis réalisés sur la prédiction inter-couches permettent de diminuer l'impact du filtre de sur-échantillonnage sur le temps de traitement. Ces compromis permettent également de compenser la complexité ajoutée par la prédiction inter-couches notamment sur les images employant une référence de type I. Le choix de ne pas employer les vecteurs de mouvement issus de la couche de base permet d'économiser de l'espace mémoire et de réduire le temps de traitement de l'étape de sur-échantillonnage. La prédiction inter-couches n'est pas réalisée sur les images qui ne servent pas de référence temporelle HEVC dans le codage de la couche de

base. Ainsi, en configuration RA, une image sur deux effectue une prédiction inter-couches sur les informations de texture. Ce choix permet de rendre le codage moins complexe pour une image sur deux améliorant la vitesse de traitement. Les résultats de codage proposés cherchent à atteindre la meilleure efficacité de codage avec des temps de traitement raisonnables.

Le chapitre suivant propose des améliorations du SHVC-AE afin de permettre son utilisation dans un contexte temps-réel. En effet, bien que la solution proposée dans ce chapitre soit peu intrusive, elle ne permet pas de répondre à une contrainte temps-réel. La réponse à une telle contrainte est rendue possible grâce à diverses optimisations sur les opérations de sous-échantillonnage et de sur-échantillonnage, ainsi que par une sélection des outils de codage HEVC sur les couches de base et de rehaussement.

Chapitre 4

Optimisation et parallélisation pour un codage temps-réel

Introduction

La solution de codage scalable présentée dans le chapitre précédent permet de réaliser des encodages en scalabilité spatiale en cherchant la meilleure efficacité de codage sur deux couches. Les compromis réalisés au sein de la décision de codage permettent de produire des encodages SHVC efficaces en terme d'efficacité de compression. En terme de vitesse de traitement, il a été montré que l'ajout des fonctionnalités scalables impacte peu la vitesse de traitement du SLE pour les configurations low-delay P (LD-P) et random-access (RA). La prédiction inter-couches en configuration all-intra (ou intra-only) (AI) présente un coût plus important en terme de complexité puisqu'il implique l'ajout du mécanisme de prédiction temporelle sur des images initialement de type I. Toutefois, le codage réalisé ne permet pas de répondre à des contraintes de vitesse ou temps-réel.

En s'appuyant sur le SHVC-AE proposé, ce chapitre propose diverses optimisations visant à permettre un traitement temps-réel du codeur SHVC-AE (LIVE). Dans un tel contexte, le codeur reçoit une unique source et chaque couche réalise un traitement temps-réel comme illustré en Figure 4.1. Le filtre de sous-échantillonnage permet de fournir une source aux couches les plus basses pour une scalabilité en résolution.

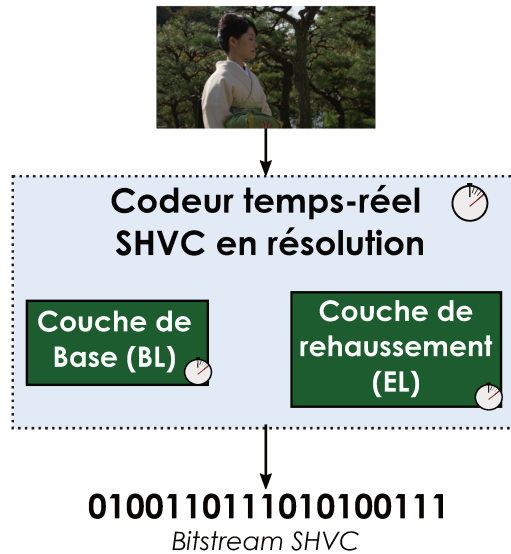


FIGURE 4.1 – Schéma du codeur SHVC LIVE visé en scalabilité spatiale sur deux couches pour des codages rapides et temps-réel.

TABLE 4.1 – Caractéristiques des séquences considérées.

Séquences	Nom	Résolution	Fréquence (IPS)	Nombres d'images
UHD1	PeopleOnStreet	3840×2160	30	150
A1	Traffic	2560×1600	30	150
A2	PeopleOnStreet	2560×1600	30	150
B1	Kimono1	1920×1080	24	240
B2	ParksScene	1920×1080	24	240
B3	Cactus	1920×1080	50	500
B4	BasketballDrive	1920×1080	50	500
B5	BQTerrace	1920×1080	60	600

Les optimisations proposées dans ce chapitre concernent notamment :

- les opérations de filtrage nécessaires à la prédiction inter-couches en scalabilité spatiale (sous-échantillonnage et sur-échantillonnage) afin d'accélérer leur traitement pour permettre de respecter la contrainte temps-réel,
- le paramétrage du filtre de sous-échantillonnage afin de minimiser le coût en débit des deux couches scalables à qualité constante,
- les outils de codage issus du standard HEVC afin de pouvoir réaliser un compromis entre efficacité de codage et vitesse de traitement.

Les opérations de filtrage sont accélérées en ajoutant deux niveaux de parallélisme sur les données ainsi qu'une utilisation d'opérations optimisées dans un langage bas niveau. Ensuite, une régulation du débit global est proposée en modifiant le paramétrage du filtre de sous-échantillonnage. Cette solution permet de réaliser une régulation du débit sans ajouter de complexité autre que l'opération de filtrage. Ce paramétrage permet également de réduire le nombre de coefficients afin de bénéficier des améliorations en vitesse de traitement des filtres. De plus, une sélection de différents outils de codage issus du standard HEVC est réalisée afin de définir plusieurs modes de qualité. Ces modes permettent de réaliser un compromis entre vitesse de codage et efficacité de codage globale en fonction de la résolution de l'image source. Enfin, une combinaison des différents modes de qualité proposés adaptés aux formats des différentes couches permet d'améliorer la prédiction inter-couches et ainsi d'optimiser le débit de codage global pour des traitements temps-réel.

4.1 Conduite des tests et outils de validation

Les expérimentations sont menées en suivant les CTC [33] sur la plateforme Intel Xeon E5-4627V2. Les conditions de test sont similaires au chapitre précédent (voir 3.3). Toutefois, nous proposons ici d'ajouter la séquence UHD1. Cette séquence permet d'analyser le comportement, notamment en terme de vitesse de codage, des améliorations proposées au SHVC-AE sur des séquences de résolution 3840×2160p. Les séquences considérées ainsi que les différents QP employés sont respectivement présentés dans les Tables 4.1 et 3.5.

Afin d'évaluer l'efficacité de codage d'un codeur par rapport à un autre la métrique BD-BR est employée. Dans le cas de la comparaison entre des codages scalables et mono-couche, les codages mono-couche servent de référence de débit. Dans ce cas, la référence en terme de qualité est celle de la couche de rehaussement.

Les vitesses de codage sont exprimées en IPS tandis que les mesures de vitesses des opérations sont exprimées en *ms*. Les accélérations sont, quant à elles, présentées sous forme du rapport des vitesses. La plateforme employée au cours des expérimentations est la plateforme Intel Xeon E5-4627V2 présentée en Table 3.6. Les mesures de vitesses de codage sont réalisées sur cette même plateforme.

4.2 Optimisations des vitesses de traitement appliquées aux filtrages

Dans le cas de la scalabilité spatiale, la prédiction inter-couches au niveau de l'encodeur requiert deux opérations de filtrage supplémentaires : un sous-échantillonnage et un sur-échantillonnage. Le sous-échantillonnage permet aux couches de référence de réaliser des encodages sur des résolutions plus basses que celle de l'image acquise. Le sur-échantillonnage permet d'adapter les informations de codage issues des couches de référence au format des couches prédites. Ces deux opérations supplémentaires sont toutes deux réalisées sur des images non compressées

et requièrent un temps de traitement important. Dans le cas d'un codage temps-réel, il devient nécessaire de réduire ce temps de traitement en optimisant ces opérations.

4.2.1 Algorithme des filtres d'interpolation

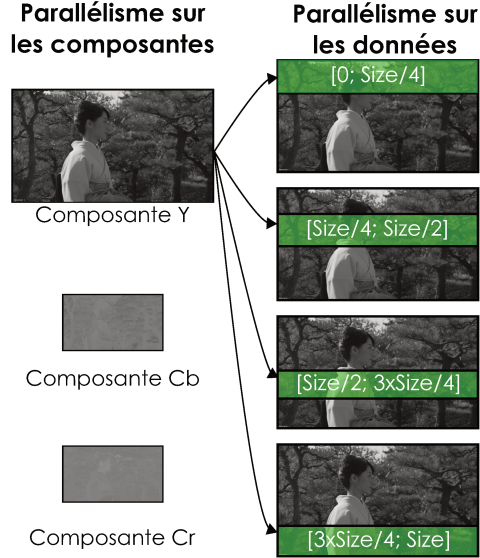


FIGURE 4.2 – Parallélismes employés sur les opérations de sur-échantillonnage et de sous-échantillonnage.

Les opérations de filtrage de sur-échantillonnage et de sous-échantillonnage sont basées sur des filtres à Réponse Impulsionnelle Finie (filtres FIR) qui sont des filtres dont la réponse impulsionnelle est stable et d'une durée finie. Comme expliqué dans [34], les filtres FIR sont généralement réalisés en opérant un algorithme d'interpolation d'ordre N avec la formule suivante :

$$s[n] = \sum_{i=-\lfloor N/2 \rfloor}^{\lfloor N/2 \rfloor} c_i \cdot p[n+i] \quad (4.1)$$

Avec c_i les coefficients du filtre, $s[n]$ l'échantillon à calculer à la position n et $p[n+i]$ les échantillons de la source centrés autour de la position n . Chaque échantillon $s[n]$ peut être traité indépendamment des autres puisqu'il ne dépend pas des échantillons précédents. Cette indépendance permet de traiter les échantillons $s[n]$ en parallèle les uns des autres sous conditions d'un accès partagé aux échantillons $p[n]$.

De plus, l'algorithme d'interpolation réalise successivement N multiplications puis $N - 1$ additions constituant un produit de convolution. Les échantillons de l'image sont contenus dans un espace à deux dimensions. En supposant un filtre séparable, le produit de convolution est alors appliqué deux fois pour chaque filtre : d'abord horizontalement puis verticalement. Les opérations réalisées ne sont donc pas très complexes mais doivent être appliquées sur tous les pixels d'une image :

- $N \times Wd \times Hs + N \times Wd \times Hd$ multiplications,
- $(N - 1) \times Wd \times Hd + (N - 1) \times Wd \times Hd$ additions.

Ws , Wd et Hd sont respectivement la largeur de l'image source, la largeur et la hauteur de l'image de destination en nombre de pixels.

Dans le cas du sous-échantillonnage, le filtrage est de type passe-bas. Par conséquent, des informations contenues dans les hautes fréquences sont susceptibles de ne plus apparaître dans l'image sous-échantillonnée. L'interpolation emploie plus d'échantillons qu'elle n'en crée. Pour un sous-échantillonnage divisant les nombres d'échantillons par 2, l'équation du produit de convolution est :

$$s[n] = \sum_{i=-\lfloor N/2 \rfloor}^{\lfloor N/2 \rfloor} c_i \cdot p[2 \times n - i] \quad (4.2)$$

La progression sur les échantillons de la source est donc plus rapide que sur celles des échantillons produits.

Dans le cas du sur-échantillonnage, le filtrage est de type passe-tout. L'interpolation crée davantage d'échantillons qu'elle n'en emploie dans sa source. Pour un sur-échantillonnage multipliant le nombre d'échantillons par 2, l'équation du produit de convolution est décomposée en :

$$s[2 \times n] = \sum_{i=[N/2]-1}^{[N/2]} c_i \times p[n-i] \quad (4.3)$$

$$s[2 \times n + 1] = \sum_{i=[N/2]-1}^{[N/2]} c_i \times p[n+1-i] \quad (4.4)$$

La progression sur les échantillons produits est donc plus rapide que sur celle des échantillons de la source.

4.2.2 Parallélisme sur les données

Les filtres de sur-échantillonnage et de sous-échantillonnage étant des filtres FIR, il est possible de traiter chaque pixel indépendamment des autres. Ce choix de parallélisme nécessiterait un très grand nombre de cœurs mais demeure possible. En effet, certaines architectures matérielles, aussi appelées *manycores*, disposent d'un très grand nombre de cœurs de calcul (supérieur à 100). C'est le cas par exemple du MPPA [68] de Kalray ou encore du Xeon Phi [69] d'Intel. Les cœurs employés dans ce genre d'architecture, disposent généralement d'une fréquence de traitement moins élevée afin de réduire la consommation d'énergie globale. Ce type d'architecture pourrait alors réaliser un traitement sur chaque pixel en parallèle. Néanmoins, les performances mono-cœur ne permettraient pas de traitements séquentiels rapides. Il est plus intéressant d'utiliser des plateformes multicœurs qui emploient moins de cœurs mais qui disposent d'une fréquence de traitement plus élevée pour chacun d'eux. Ces architectures multicœurs permettent d'employer une granularité de parallélisme moins importante. En effet, un parallélisme sur tous les pixels impliquerait un nombre de tâches largement supérieur au nombre de cœurs d'un CPU multicœurs. La majorité des tâches serait alors exécutée successivement sur les différents cœurs. De plus, entre chaque tâche, un temps de transition non nul s'ajouterait au temps de traitement total de l'opération. Afin de dépendre le moins possible de la vitesse de transition entre deux tâches, il est préférable d'employer moins de parallélisme et davantage de traitements séquentiels. Dans le SHVC-AE, il a été choisi de séparer le traitement des filtres sur 4 régions de l'image de tailles approximativement égales afin de les traiter en parallèle. Dans le cas où la hauteur de l'image n'est pas multiple de 4, les 3 premières régions sont arrondies au nombre de pixel inférieur tandis que la quatrième et dernière région traitera le reste de lignes de pixel de l'image. Cette répartition des données est alors employée uniquement sur les filtres de sous-échantillonnage et de sur-échantillonnage.

De plus, un autre niveau de parallélisme peut être employé. En effet, chaque image est constituée de trois composantes indépendantes. Ces dernières peuvent également être traitées en parallèle les unes des autres. La Figure 4.2 illustre les deux dimensions de parallélisme employées dans les deux opérations de filtrage. Le SHVC-AE intègre donc deux niveaux de parallélisme sur les opérations de sous-échantillonnage et de sur-échantillonnage : sur les composantes d'abord puis sur les données.

En plus de séparer le traitement sur différents processeurs, il est possible de réduire le temps de traitement des filtres en réduisant le temps de traitement de l'opération d'interpolation. Cette accélération est possible en employant un langage de programmation bas-niveau sur le produit de convolution.

4.2.3 Instructions SIMD

Afin d'améliorer la vitesse de traitement d'un programme informatique, il est possible d'employer des langages bas niveau. Ces langages, plus proches de la machine, permettent notamment d'accéder aux opérateurs logiques. Les optimisations bas niveau permettent entre autre :

- de réaliser plusieurs fois la même opération logique en moins de cycles en exploitant toute la largeur des registres disponibles (128, 256, 512 bits),
- de réduire le nombre de cycles d'une succession d'opérations logiques lentes en la remplaçant par une plus grande succession d'autres opérations logiques unitairement plus rapides,
- de configurer la mémoire en réalisant un alignement des données afin de réduire le nombre de demandes d'accès mémoire...

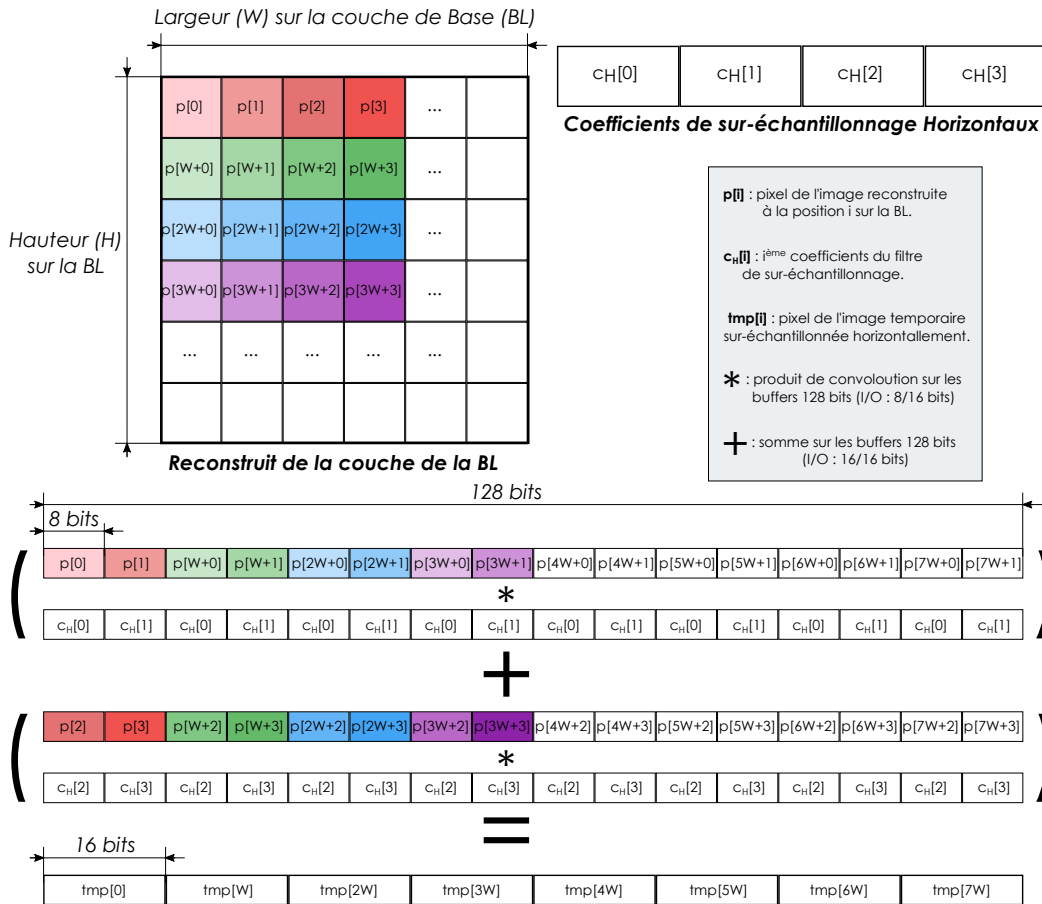


FIGURE 4.3 – Schéma du produit de convolution optimisé sur le filtre horizontal.

Le langage assembleur, par exemple, permet de réaliser ces optimisations. Cependant ce langage devient rapidement complexe d'utilisation dès lors que le programme rédigé augmente en complexité. Afin de réduire la complexité du code source, il est possible de faire appel à des opérations pré-optimisées en conservant un langage de programmation plus haut niveau. Certains processeurs mettent à disposition des jeux d'instructions permettant de faire appel à ce genre d'opérations. Ces instructions donnent alors accès à des opérations rapides et optimisées sans changer de langage de programmation. C'est notamment le cas des instructions "Single Instruction Multiple Data" (SIMD).

Les instructions SIMD permettent d'exécuter en parallèle un même jeu d'instructions sur différents buffers de données. Ces instructions permettent alors de réduire la quantité de module logique "hardware" nécessaire à ces multiples opérations. On trouve ces instructions dans des bibliothèques proposées par les concepteurs de processeurs comme Intel ou ARM/Neon. Parmi ces instructions optimisées, on trouve notamment l'opération de produit de convolution qui consiste à réaliser la somme de plusieurs produits. Cette opération étant la base du traitement des filtres de sur-échantillonnage et de sous-échantillonnage, il est alors possible d'optimiser à bas niveau les opérations de filtrage.

Pour un codage 8 bits, les échantillons sont stockés sur un octet. Les 8 échantillons nécessaires au filtre 8-coefficients sont alors alignés dans une zone mémoire appelée buffer de tailles 128 bits. De même les 8 coefficients sont stockés et alignés dans un buffer 128 bits. Ensuite, les coefficients c_H et c_V sont respectivement multipliés aux échantillons $p[n]$ et $p'[n]$. Ces produits sont alors sommés pour obtenir respectivement les échantillons filtrés $p'[n]$ et $s[n]$ qui correspondent respectivement aux filtrages horizontal et vertical. Les Figures 4.3 et 4.4 schématisent les produits de convolution respectivement appliqués au filtrage horizontal et vertical sur les composantes chromatiques U et V. Les schémas équivalents pour la composante Y sont présentés en Annexe D. Le produit de convolution sur les 8 échantillons est alors réalisé avec le découpage suivant sur la composante Y (ou sur les composantes U et V)

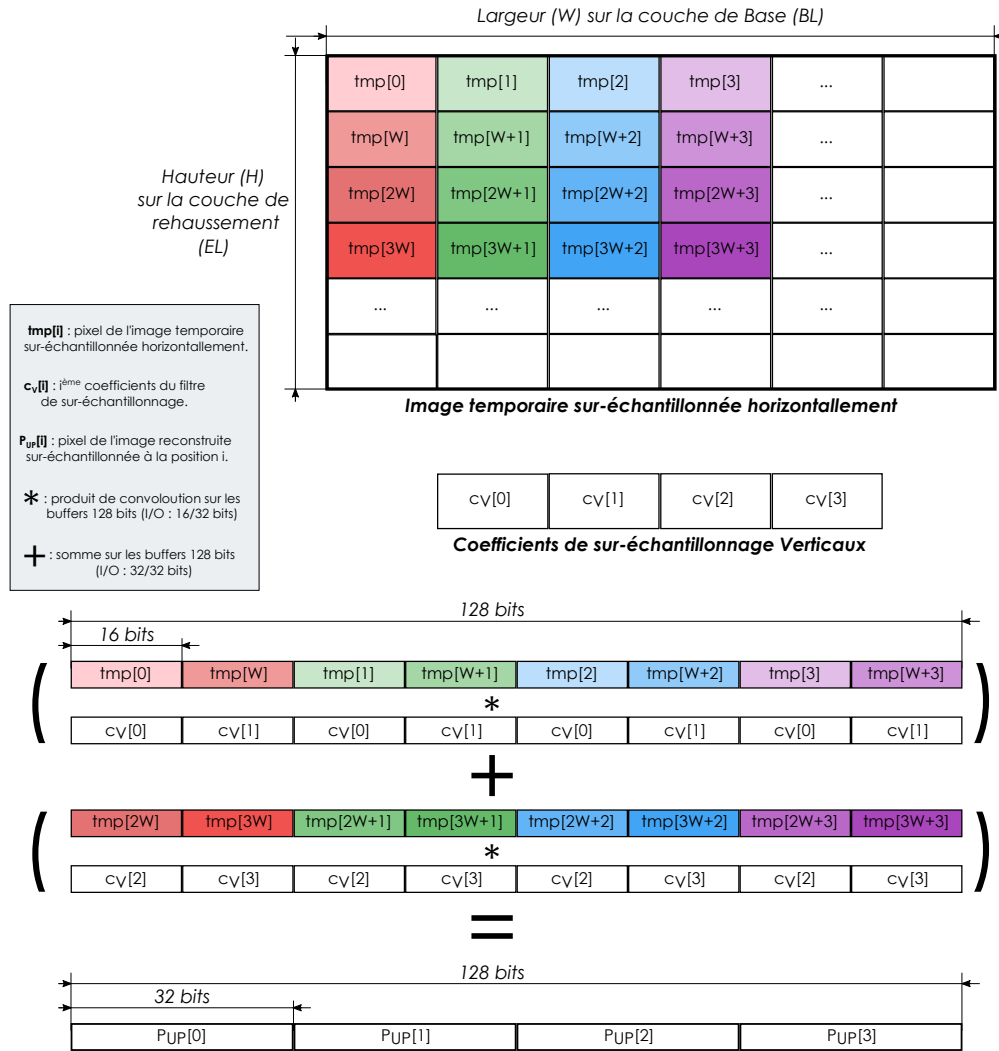


FIGURE 4.4 – Schéma du produit de convolution optimisé sur le filtre vertical.

avec $p[i]$ les échantillons à filtrer et $c[i]$ les coefficients :

- 4 (ou 2) produits de convolution : $p[i] \times c[i] + p[i + 1] \times c[i + 1]$
- 3 (ou 1) additions sur les résultats des produits de convolution.

On se place dans le cas du codage avec des pixels sur 8 bits. Les échantillons et les coefficients fournis au filtrage horizontal sont stockés sur 8 bits. Il est possible de stocker jusqu'à 16 échantillons de taille 8 bits dans le buffer 128 bits. Le produit de convolution donne alors un échantillon temporaire sur 16 bits. Ainsi, il est possible de produire 8 échantillons temporaires avec un seul buffer 128 bits. Pour cela, chaque buffer est séparé en 8 régions. Ces régions sont de tailles 16 bits et chacune d'elles correspond à un échantillon temporaire à produire. Dans les Figures 4.3 et 4.4, les régions sont représentées avec les nuances de couleurs. Afin de stocker les 8 échantillons sources issus de la couche de base, ceux-ci sont alors répartis sur 4 et 2 buffers 128 bits respectivement pour la composante Y et les composantes U et V. Les 8 et 4 coefficients sont répartis de la même manière respectivement pour la composante Y et les composantes U et V.

Les échantillons temporaires issus du filtrage horizontal sont conservés dans une image temporaire de dimension :

- Largeur : égale à la largeur de l'image de la couche de base,
- Hauteur : égale à la hauteur de l'image de la couche de rehaussement.

Ensuite, l'image contenant ces échantillons temporaires stockés sur 16 bits est fournie au filtrage vertical. Afin

TABLE 4.2 – Temps de traitement des opérations de sous-échantillonnage et de sur-échantillonnage exprimés en millisecondes (ms)

Séq.	Contrainte temps-réel	Sous-échantillonnage		Sur-échantillonnage	
		sans optimisation	avec optimisation	sans optimisation	avec optimisation
UHD1	33.3	26.8	7.3	43.2	6.1
A1 A2	33.3	10.9	1.5	19.7	2.8
B1 B2	41.7	4.8	0.8	10.0	1.4
B3 B4	20.0	4.8	0.8	10.0	1.4
B5	16.7	4.8	0.8	10.0	1.4

d'aligner les opérations sur les buffers 128 bits, les coefficients du filtrage sont eux aussi stockés sur 16 bits pour un codage 8 bits. Le produit de convolution donnera alors un résultat stocké sur 32 bits. Il est alors possible de produire 4 échantillons filtrés. De la même manière que pour le filtrage horizontal, les échantillons temporaires sont répartis en régions. Ces régions de taille 32 bits peuvent contenir 2 échantillons temporaires et fixent le nombre de buffers 128 bits nécessaire à 4 et 2 respectivement pour la composante Y et les composantes U et V. Autant de buffers contenant les échantillons sont alloués. Enfin, les échantillons filtrés sont reformatés pour un codage 8 bits avant d'être conservés dans ce même format. Le reformatage est standardisé et défini par le calcul suivant avec $ech_{8\text{ bits}}$ l'échantillon qui sera employé dans la prédiction inter-couches et $ech_{\text{filtré}}$ l'échantillon obtenu après le filtrage vertical :

$$ech_{8\text{ bits}} = \frac{ech_{\text{filtré}} + 2^{(20-\text{profondeur}-1)}}{2^{(20-\text{profondeur})}}$$

Dans le cas du codeur SHVC-AE, ces instructions SIMD sont issues de la librairie "Streaming SIMD Extension 2" (SSE2) [70]. Cette librairie est composée de 144 instructions exploitables par des microprocesseurs x86. Elle permet de réaliser des opérations sur des entiers de 128 bits ainsi que sur des flottants en simple (32 bits) et double (64 bits) précision. De plus, le filtre de sur-échantillonnage est déjà défini sur 8 coefficients dans le standard SHVC permettant d'intégrer ces optimisations sur 8 coefficients. Le filtre de sous-échantillonnage, quant à lui, n'est pas défini dans le standard SHVC. Le paramétrage de ce filtre sur 8 coefficients est présenté dans la Section 4.3.

4.2.4 Résultats

Les trois niveaux de parallélisme sur les composantes, les données et les opérations bas niveau permettent une amélioration significative du temps de traitement des opérations de filtrage. De plus, ces optimisations n'altèrent pas les résultats produits. La Table 4.2 présente les performances ainsi obtenues sur les filtres de sous-échantillonnage et de sur-échantillonnage. La contraintes temps-réel sur une image est calculée comme suit :

$$\text{Contrainte temps-réel sur une image (ms)} = 1000 \times \frac{1 (\text{image})}{\text{Fréquence d'image (IPS)}} \quad (4.5)$$

On constate que les optimisations sont nécessaires uniquement dans le cas de vidéos de résolution 3840×2160p. Néanmoins, pour les résolutions plus basses, ces optimisations restent intéressantes puisqu'elles permettent de réduire significativement le temps d'exécution des filtres. Ainsi, non seulement la qualité des images ré-échantillonnées n'est pas altérée, mais en plus la complexité du codage en scalabilité spatiale est à la fois diminuée et moins intrusive.

De plus, les améliorations des temps de traitement des filtres dépendent essentiellement de la résolution de l'image tandis que la contrainte temps-réel dépend uniquement de la fréquence d'images. Ces optimisations pourraient alors permettre de réaliser les opérations de filtrage en respectant la contrainte temps-réel sur des séquences de résolution 3840×2160 à 120 IPS. En effet, la contrainte dans ce cas s'élève à 8.3 ms par image alors que les mesures sur les opérations de filtrage sur les résolutions 3840×2160p sont en moyenne inférieures.

Dans le cas de traitement de séquences avec des résolutions inférieures à l'Ultra Haute Définition (UHD) (3840×2160p), ces optimisations restent intéressantes puisqu'elles permettent de réduire la durée des tâches de filtrage sans altérer le résultat de l'opération. Ces tâches sont ainsi exécutées plus rapidement et réduisent l'occupation des ressources matérielles. Le démarrage des autres tâches peut alors être effectué plus rapidement.

En ce qui concerne le filtre de sous-échantillonnage, les optimisations apportées sur les 3 niveaux de parallélisme permettent d'assurer son traitement en temps-réel pour un filtrage employant 8 coefficients. Le paramétrage de ce filtre n'est pas défini dans l'extension SHVC. Par conséquent, il est nécessaire de définir un paramétrage permettant de réaliser un codage efficace sur la couche de base afin d'améliorer l'efficacité de la prédiction inter-couches.

4.3 Paramétrage du filtre de sous-échantillonnage avec 8 coefficients

Le filtre de sous-échantillonnage n'est pas défini dans le standard SHVC puisqu'il n'est pas présent dans le décodage. Le paramétrage de ce filtrage peut donc être choisi arbitrairement par le codeur. Le codeur de référence SHM emploie un pré-sous-échantillonnage avec 12 coefficients. Le paramétrage utilisé est issu de la recommandation [61] qui propose un filtre de sous-échantillonnage appliqué aux composantes chromatiques pour des conversions 4:4:4 vers 4:2:2 et 4:2:2 vers 4:2:0. Dans ce chapitre, le sous-échantillonnage est intégré au pipeline de codage du SHVC-AE afin de pouvoir être employé en condition de diffusion temps-réel avec une unique source. Le paramétrage du filtre de sous-échantillonnage peut affecter significativement l'efficacité de codage de la prédiction inter-couches. En effet, puisqu'il définit la source des couches de référence, le codage, et notamment l'image reconstruite de ces dernières, sera affecté. Le choix du paramétrage peut donc permettre à la fois d'améliorer l'efficacité de la prédiction inter-couches en améliorant les images reconstruites des couches de référence, mais aussi de réduire le coût en débit de la couche de base.

4.3.1 Les différents paramètres de sous-échantillonnage

Le paramétrage du filtre de sous-échantillonnage employé dans le SHM est issu des recommandations de [61]. Ce sous-échantillonnage se base sur un filtre FIR avec ajustement de phase dans une fenêtre Kaiser demi-bande. Par exemple, son paramétrage avec un ratio 2 est exprimé comme suit :

- Paramètre Kaiser (K) : 2.75
- Décalage de la phase (p) : 0
- Nombre de lobes secondaires plus un (N) : 3
- Ratio de sous-échantillonnage (D) : 2

Le calcul du nombre de coefficients dépend des paramètres N et D uniquement. Ce nombre de coefficient est défini par l'équation suivante :

$$\text{Nombre de coefficients} = 2 \times N \times D \quad (4.6)$$

Le filtre employé dans le SHM est donc un filtre sur 12 coefficients. On appellera ce filtre *SoEc-SHM*. Dans le cas du codeur SHVC-AE, on cherche à trouver un paramétrage avec 8 coefficients afin de bénéficier des performances des filtres optimisées dans la Section 4.2 pour un codage sur deux couches avec des résolutions deux fois moins larges et deux fois moins hautes sur la couche de base par rapport à la couche de rehaussement (i.e. $D = 2$). On constate alors que seul le nombre de lobes secondaires peut être changé, N doit donc prendre la valeur 2. Ceci aura pour effet de ralentir la pente de la réponse impulsionnelle. Le seul paramètre pouvant permettre de compenser ce décalage est le paramètre Kaiser K . En effet, afin de ne pas sévèrement dégrader l'efficacité de la prédiction inter-couches, le décalage de phase doit être le même entre le sous-échantillonnage et le sur-échantillonnage, soit 0 ici. Dans le reste de cette section, on cherchera à identifier un filtrage qui permettra d'obtenir une efficacité de codage équivalente ou meilleure que le paramétrage SoEch-SHM avec un paramétrage du filtre comme suit :

- Paramètre Kaiser (K) : à identifier
- Décalage de la phase (p) : 0
- Nombre de lobes secondaires plus un (N) : 2
- Ratio de sous-échantillonnage (D) : 2

4.3.2 Expérimentations avec différents paramétrages du sous-échantillonnage

L'objectif des expérimentations présentées dans cette partie est d'identifier une valeur de K permettant au filtre de sous-échantillonnage d'être efficace. Cette efficacité s'exprime au travers d'un compromis réalisé entre les différents effets provoqués par le sous-échantillonnage suivant :

- la distorsion de l'image source,
- l'efficacité de la prédiction inter-couches sur les textures,
- le coût en débit total par rapport à une application simulcast.

Afin de pouvoir bénéficier des optimisations de la partie précédente, il est nécessaire de choisir une valeur pour le paramètre K . Ce paramétrage doit permettre de donner une représentation fidèle de l'image source. Dans les expérimentations menées, ce paramètre est compris dans la plage des valeurs commençant à 1.0 et se terminant à 4.5, avec un pas de 0.25 entre chaque pour 15 valeurs de K testées. Ces différents paramétrages appliqués au SHVC-AE sont appelés "K- n ", avec n la valeur du paramètre K . La Figure 4.5, présente les réponses impulsionnelles des différents paramétrages employés dans le SHM sur 12 coefficients, ainsi que dans le SHVC-AE sur K-1.0, K-2.0 et K-4.5. Les autres réponses impulsionnelles sont comprises entre les deux courbes K-1.0 et K-4.5 comme c'est le cas de K-2.0 par exemple. Les vues de tête et de queue de ces réponses impulsionnelles sont présentées en Annexe E. On constate que les réponses impulsionnelles au niveau des basses fréquences accusent quelques distorsions pouvant affecter la qualité de l'image sous-échantillonnée. Cette distorsion est moins importante pour une valeur de K élevée. On constate également que les filtres K- n considérés conservent davantage de hautes fréquences par rapport au SoEc-SHM. Ce comportement peut être bénéfique pour la prédiction inter-couches puisqu'il permet de conserver davantage d'informations dans l'image source. Toutefois, les filtres K- n créent des amplitudes plus grandes sur les lobes secondaires. Ces deux comportements dans les hautes fréquences peuvent être responsables d'un effet visuel indésirable appelé "repliement de spectre", ou "spatial aliasing", qu'on appellera simplement "aliasing". Cet effet, quand il est appliqué à la vidéo, consiste en une distorsion des contrastes sur les bordures d'un objet comme illustré sur la Figure 4.6. Il peut prendre effet indépendamment sur les différentes composantes. Il produit une déformation des lignes sur les contours pour la luminance et un effet de coulure pour les composantes chromatiques. Cet effet est difficilement mesurable avec des métriques objectives, il est alors nécessaire de recourir à une observation visuelle de l'image produite ou bien à des métriques subjectives dédiées à ce phénomène. Dans le cas du sous-échantillonnage sur le SHVC-AE, l'aliasing est étudié par observations visuelles.

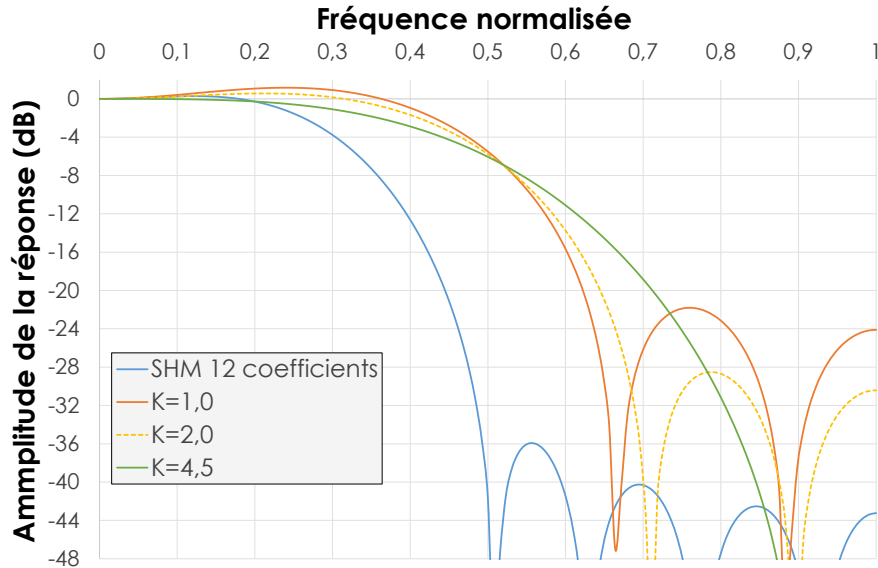


FIGURE 4.5 – Réponse impulsionnelle des différents paramétrages du filtre de sous-échantillonnage : vue globale.

L'aliasing provoqué par les filtres K- n est illustré sur la Figure 4.7. Les zones d'intérêt en rouge et en vert sur la Figure 4.7a identifient des zones susceptibles de produire de l'aliasing. On observe que dans la zone rouge, les filtres K-1.0 et K-4.5 récupèrent davantage d'informations par comparaison au SoEc-SHM. Ceci est notamment dû à la conservation de davantage d'informations dans les hautes fréquences sur les filtres K- n . De plus, la zone verte

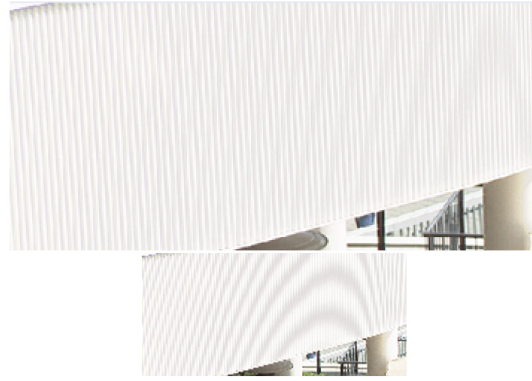


FIGURE 4.6 – Exemple d’effet d’aliasing sur les contours dans les images. (a) Image source. (b) Image sous-échantillonnée.

TABLE 4.3 – Comparaison BD-BR de la couche de rehaussement avec les différents paramètres de filtres de sous-échantillonnage sur SHVC-AE par rapport au SLE en configuration AI.

Séquences	SoEc-SHM	K-1	K-1.25	K-1.5	K-1.75	K-2	K-2.25	K-2.5
UHD1	-45.2%	-45.5%	-46.4%	-47.0%	-47.4%	-47.5%	-47.3%	-46.9%
A1	-42.6%	-44.6%	-45.2%	-45.5%	-45.6%	-45.5%	-45.1%	-44.6%
A2	-46.7%	-45.4%	-46.4%	-47.2%	-47.7%	-48.0%	-48.0%	-47.7%
B1	-50.5%	-52.1%	-52.1%	-52.0%	-51.8%	-51.4%	-51.0%	-50.4%
B2	-35.8%	-40.7%	-40.8%	-40.7%	-40.5%	-40.0%	-39.5%	-38.8%
B3	-31.9%	-34.5%	-34.7%	-34.8%	-34.6%	-34.4%	-34.0%	-33.5%
B4	-26.1%	-26.6%	-26.8%	-26.9%	-26.9%	-26.7%	-26.6%	-26.2%
B5	-18.4%	-20.6%	-20.9%	-21.1%	-21.1%	-21.1%	-20.9%	-20.6%
Moyenne	-36.0%	-37.8%	-38.3%	-38.3%	-38.3%	-38.1%	-37.9%	-37.4%

montre un léger aliasing mais qui reste toutefois difficile à percevoir. Celui-ci est plus prononcé sur les filtrages avec un paramètre K proche de 4.5. En effet, avec une valeur plus élevée de ce paramètre, le risque d’aliasing augmente puisque davantage de hautes fréquences sont conservées. On peut conclure ici que les différents paramètres de filtrage considérés sont certes susceptibles de produire un effet d’aliasing, mais toutefois minime et perceptible sous condition d’un fort agrandissement de l’image. Le phénomène de distorsion provoqué par les filtres $K-n$ est alors jugé non gênant pour la visualisation. On notera tout de même que les filtres avec un paramètre K plus faible sont plus intéressants puisqu’ils réduisent le risque de production d’aliasing.

Les Tables 4.3 et 4.4 présentent les comparaisons BD-BR du SHVC-AE sur la couche de rehaussement par rapport au SLE équivalent avec les différents filtres de sous-échantillonnage $K-n$. Les mesures de débit et de PSNR sont réalisées en configuration AI avec un ratio 2 entre les deux couches (i.e. résolutions d’images de la couche de base deux fois moins hautes et deux fois moins larges par rapport à la couche de rehaussement). On constate que les filtrages $K-1$ à $K-3$ permettent d’obtenir une efficacité de la prédiction inter-couches supérieure aux codages réalisés avec le filtre SoEc-SHM. Cela s’explique par la création d’une image reconstruite sur la couche de base plus fidèle à l’image sous-échantillonnée. Ainsi, les textures employées dans la prédiction inter-couches sont plus intéressantes avec ces filtrages $K-1$ à $K-3$ par rapport au SoEc-SHM. Néanmoins, cette amélioration de la qualité du reconstruit de la couche de base peut être due à un coût en débit plus élevé sur cette couche. Dans ce cas, les mesures en terme d’efficacité de codage sur la couche de base sont également nécessaires afin de valider l’intérêt du paramétrage du sous-échantillonnage.

Les Tables 4.5 et 4.6 présentent la comparaison BD-BR des encodages réalisés sur la couche de base du SHVC-AE avec les filtres $K-n$ par rapport à l’encodage réalisé avec le filtre SoEc-SHM sur le même codeur. On constate que le coût en débit à qualité constante sur la couche de base avec les paramètres compris entre $K-1.0$ et $K-3.75$ est plus important qu’avec le filtre SoEc-SHM. Par conséquent, les images produites avec ces filtres sont plus complexes à coder sur la couche de base. La variation du paramètre K permet donc d’effectuer une régulation de



(a) Image source.



(b) Sous-échantillonnage avec SoEc-SHM.



(c) Sous-échantillonnage avec K-1.0.



(d) Sous-échantillonnage avec K-4.5.

FIGURE 4.7 – Zoom sur les captures d’images de la séquence B5 source et sous-échantillonnée avec les filtres de SHM, K-1.0 et K-4.5.

débit entre les couches de base et de rehaussement. L’opération de filtrage est cependant moins complexe qu’une régulation de débit.

On constate alors que les paramétrages proposés sur le sous-échantillonnage, à qualité constante, réduisent le coût en débit de la couche de rehaussement mais augmente celui de la couche de base. Une évaluation du coût global en débit (couche de base + couche de rehaussement) est alors nécessaire afin de vérifier si le paramétrage de sous-échantillonnage avec 8 coefficients améliore ou non l’efficacité de codage du SHVC-AE.

Les Tables 4.7 et 4.8 présentent la comparaison BD-BR des encodages des couches de base et de rehaussement du SHVC-AE par rapport aux encodages simulcast équivalents réalisés par le SLE. Ces encodages simulcast prennent les mêmes paramètres de filtrage $K-n$ considérés. On constate que le gain total par rapport au filtre SoEc-SHM est plus intéressant entre K-1.25 et K-3.25. Pour ces paramètres, le surcote en débit de la couche de base est compensé par l’amélioration de l’efficacité de codage de la couche de rehaussement. On remarque également que ces gains ont un optimum en K-2.0 et K-2.25. Ces deux valeurs de K permettent donc de maximiser l’efficacité de codage en minimisant le coût en débit à qualité constante. L’allure des gains obtenus est représentée sur la courbe en Figure 4.8. Il est plus intéressant de sélectionner un optimum sur la pente descendante. En effet, sur cette partie de la courbe, les paramètres de filtrage récupèrent légèrement moins d’informations issues des hautes fréquences. Ils permettent alors, à efficacité de codage équivalente, de réduire le risque d’aliasing. Le filtre K-2.0 est le filtre de sous-échantillonnage sélectionné dans le SHVC-AE, car il présente moins de risque de produire des effets d’aliasing par rapport au filtre K-2.25.

TABLE 4.4 – Comparaison BD-BR de la couche de rehaussement avec les différents paramètres de filtre de sous-échantillonnage sur SHVC-AE par rapport au SLE en configuration AI.

Séquences	K-2.75	K-3	K-3.25	K-3.5	K-3.75	K-4	K-4.25	K-4.5
UHD1	-46.4%	-45.7%	-45.1%	-44.3%	-43.7%	-43.0%	-42.3%	-41.8%
A1	-44.0%	-43.3%	-42.6%	-41.8%	-41.2%	-40.5%	-39.9%	-39.4%
A2	-47.5%	-46.6%	-46.0%	-45.3%	-44.7%	-44.0%	-43.4%	-42.9%
B1	-49.9%	-49.3%	-48.8%	-48.3%	-47.8%	-47.4%	-46.9%	-46.6%
B2	-38.2%	-37.5%	-36.8%	-36.1%	-35.5%	-35.0%	-34.4%	-34.0%
B3	-33.0%	-32.4%	-31.8%	-31.3%	-30.7%	-30.3%	-29.8%	-29.4%
B4	-25.9%	-25.6%	-25.2%	-24.8%	-24.5%	-24.1%	-23.8%	-23.6%
B5	-20.3%	-19.9%	-19.2%	-19.1%	-18.7%	-18.4%	-18.0%	-17.8%
Moyenne	-36.9%	-36.4%	-35.8%	-35.2%	-34.7%	-34.2%	-33.8%	-33.4%

TABLE 4.5 – Comparaison BD-BR sur la couche de base avec les différents paramètres de filtre de sous-échantillonnage sur SHVC-AE par rapport au filtrage SoEc-SHM en configuration AI.

Séquences	SoEc-SHM	K-1	K-1.25	K-1.5	K-1.75	K-2	K-2.25	K-2.5
UHD1	-	23.8%	20.8%	17.9%	14.9%	12.0%	9.7%	7.1%
A1	-	27.1%	23.9%	20.6%	17.4%	14.2%	11.7%	8.9%
A2	-	25.1%	21.9%	18.8%	15.6%	12.5%	10.1%	7.4%
B1	-	28.4%	25.0%	21.5%	18.2%	14.8%	12.3%	9.4%
B2	-	34.0%	30.0%	26.0%	22.1%	18.3%	15.0%	11.8%
B3	-	31.0%	27.4%	23.6%	20.1%	16.6%	13.6%	10.7%
B4	-	33.8%	29.9%	25.8%	22.1%	18.4%	15.3%	12.3%
B5	-	31.8%	28.1%	24.4%	20.8%	17.5%	14.5%	11.6%
Moyenne	-	30.2%	26.6%	23.0%	19.5%	16.0%	13.2%	10.3%

TABLE 4.6 – Comparaison BD-BR sur la couche de base avec les différents paramètres de filtre de sous-échantillonnage sur SHVC-AE par rapport au filtrage SoEc-SHM en configuration AI.

Séquences	K-2.75	K-3	K-3.25	K-3.5	K-3.75	K-4	K-4.25	K-4.5
UHD1	5.2%	3.3%	1.8%	0.1%	-1.0%	-2.2%	-3.3%	-4.0%
A1	6.9%	4.8%	3.1%	1.4%	0.1%	-1.2%	-2.4%	-3.1%
A2	5.4%	3.4%	1.8%	0.1%	-1.2%	-2.4%	-3.5%	-4.2%
B1	7.4%	5.2%	3.6%	1.7%	0.4%	-0.9%	-1.9%	-2.7%
B2	9.1%	6.6%	4.5%	2.4%	0.7%	-0.8%	-2.2%	-3.3%
B3	8.4%	6.1%	4.2%	2.3%	0.8%	-0.6%	-1.7%	-2.7%
B4	9.9%	7.7%	5.8%	3.8%	2.4%	1.0%	-0.1%	-1.0%
B5	9.3%	7.1%	5.2%	3.4%	1.9%	0.5%	-0.6%	-1.5%
Moyenne	8.1%	5.8%	4.0%	2.2%	0.7%	-0.6%	-1.8%	-2.6%

TABLE 4.7 – Comparaison BD-BR sur les couches de base et de rehaussement avec les différents paramètres de filtre de sous-échantillonnage sur SHVC-AE par rapport au codage simulcast SLE en configuration AI.

Séquences	SoEc-SHM	K-1	K-1.25	K-1.5	K-1.75	K-2	K-2.25	K-2.5
UHD1	-26.9%	-25.4%	-26.2%	-26.8%	-27.3%	-27.6%	-27.7%	-27.7%
A1	-24.9%	-24.6%	-25.1%	-25.6%	-25.9%	-26.0%	-26.1%	-25.9%
A2	-27.4%	-24.8%	-25.7%	-26.4%	-27.0%	-27.4%	-27.6%	-27.6%
B1	-29.6%	-29.0%	-29.3%	-29.5%	-29.6%	-29.6%	-29.6%	-29.5%
B2	-21.9%	-23.2%	-23.6%	-23.8%	-23.9%	-23.8%	-23.7%	-23.5%
B3	-19.5%	-19.8%	-20.1%	-20.3%	-20.4%	-20.5%	-20.4%	-20.2%
B4	-16.3%	-15.8%	-16.0%	-16.2%	-16.4%	-16.4%	-16.4%	-16.3%
B5	-12.3%	-12.9%	-13.2%	-13.5%	-13.6%	-13.7%	-13.7%	-13.5%
Moyenne	-21.7%	-21.4%	-21.9%	-22.2%	-22.4%	-22.5%	-22.5%	-22.4%

TABLE 4.8 – Comparaison BD-BR sur les couches de base et de rehaussement avec les différents paramètres de filtre de sous-échantillonnage sur SHVC-AE par rapport au codage simulcast SLE en configuration AI.

Séquences	K-2.75	K-3	K-3.25	K-3.5	K-3.75	K-4	K-4.25	K-4.5
UHD1	-27.5%	-27.3%	-27.0%	-26.7%	-26.4%	-26.1%	-25.7%	-25.5%
A1	-25.7%	-25.5%	-25.2%	-24.9%	-24.5%	-24.2%	-23.9%	-23.7%
A2	-27.6%	-27.4%	-27.2%	-26.9%	-26.6%	-26.3%	-26.0%	-25.7%
B1	-29.3%	-29.2%	-29.0%	-28.8%	-28.6%	-28.4%	-28.2%	-28.1%
B2	-23.3%	-23.0%	-22.7%	-22.4%	-22.1%	-21.8%	-21.5%	-21.3%
B3	-20.0%	-19.8%	-19.6%	-19.3%	-19.0%	-18.8%	-18.6%	-18.4%
B4	-16.2%	-16.0%	-15.9%	-15.7%	-15.5%	-15.4%	-15.2%	-15.1%
B5	-13.4%	-13.2%	-13.0%	-12.8%	-12.6%	-12.4%	-12.2%	-12.0%
Moyenne	-22.2%	-22.0%	-21.8%	-21.5%	-21.3%	-21.0%	-20.8%	-20.6%

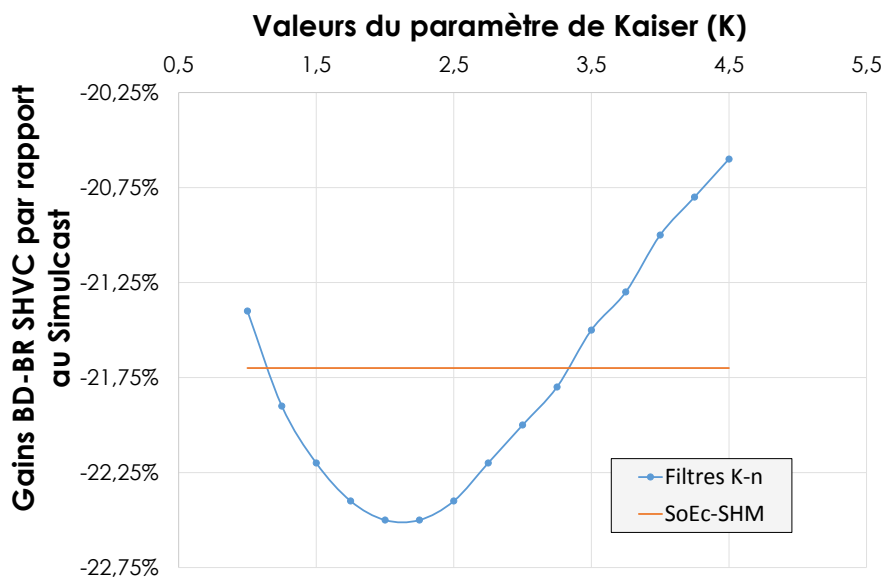


FIGURE 4.8 – Gains BD-BR entre SHVC-AE et SLE en simulcast sur le coût total en débit avec les différents filtres considérés.

Le paramétrage du filtre de sous-échantillonnage permet d'améliorer l'efficacité de codage global du codage SHVC. Il est alors possible de réaliser une régulation de débit entre les différentes couches en scalabilité spatiale. Cette régulation permet alors d'améliorer l'efficacité de codage par rapport à l'utilisation du pré-sous-échantillonnage sur 12 coefficients proposé par le SHM. Cette régulation est toutefois moins complexe qu'une régulation de débit classique. Cependant, la sélection d'un optimum pour ce paramétrage dépend essentiellement du cadre d'application. En effet, le choix réalisé ici optimise le coût global pour des résolutions deux fois moins larges et deux fois moins hautes sur la couche de base par rapport à la couche de rehaussement. Des applications comme le MBMS (voir section 2.9 peuvent préférer une amélioration de qualité de la couche de rehaussement pour leur service "premium" au détriment de l'efficacité de codage de la couche de base. Il serait alors intéressant de proposer un paramétrage du filtre de sous-échantillonnage dynamique. De la même manière que pour la qualité ou bien pour le débit de codage, une dynamique sur ce paramétrage permettrait de répondre à divers cas pratiques d'utilisation d'un codeur SHVC.

Avec le paramétrage du filtre de sous-échantillonnage intégré au pipeline de codage du SHVC-AE, il est possible d'envisager une utilisation du codeur proposé dans un contexte temps-réel qui ne peut fournir qu'une source. Toutefois, les vitesses obtenues dans le chapitre 3 et jusqu'à présent sur la décision de codage ne permettent d'atteindre une telle vitesse de traitement. Il devient alors nécessaire d'accélérer cette décision en réduisant sa complexité par exemple en sélectionnant les outils issus du standard HEVC dans la décision de codage.

4.4 Amélioration de la vitesse de traitement en sélectionnant les outils dans la décision de codage

Afin d'améliorer la vitesse de traitement du codeur SHVC-AE, il est possible d'optimiser chaque opération à toutes les étapes du pipeline. De la même manière que pour les opérations de filtrage, ces opérations peuvent employer un langage bas niveau. Cependant, la complexité de la rédaction d'un tel programme serait alors très élevée et rendrait le programme difficilement modulable. Un autre moyen efficace d'améliorer la vitesse de traitement du codeur est de réaliser une sélection des outils de codage. Cette méthode permet de réduire la complexité d'un

codeur au détriment de son efficacité de compression. Dans ce cas, il est alors nécessaire de réaliser un compromis entre la vitesse et l'efficacité du codage. En ce qui concerne le SHVC, il est possible de sélectionner des outils de codage parmi ceux définis dans le standard HEVC ainsi que parmi ceux définis dans l'extension SHVC. Les compromis réalisés sur les outils de prédiction propres à la prédiction inter-couches sont déjà présentés dans le chapitre 3. Ce chapitre présente les compromis réalisés sur les outils issus du standard HEVC. La configuration du codeur cherchant à réaliser la meilleure efficacité de codage, employée dans le chapitre précédent, est appelée FILE. La configuration du codeur cherchant à réaliser un traitement plus rapide voir temps-réel sera appelée LIVE.

4.4.1 Décision de codage en configuration FILE

Afin d'obtenir une bonne efficacité de codage avec la configuration FILE, la plupart des outils définis dans le standard HEVC sont employés. Un compromis sur certains d'entre eux a toutefois été réalisé afin d'obtenir une vitesse de traitement raisonnable. Ainsi, la prédiction intra emploie 19 directions sur les 33 modes directionnels disponibles et les partitions asymétriques (voir Figure 2.15 de la prédiction inter ne sont pas employées. La Table 4.10 présente ces quelques restrictions sur le mode de qualité FILE. La Figure 4.9 présente les courbes débit/distorsion des codeurs SHVC-AE et SHM sur la séquence B4 en configuration RA. L'écart entre ces deux codeurs est causé par les choix réalisés sur les outils de codage ainsi que par la version du SLE datant de 2015 et n'intégrant pas les meilleures décisions de codage en prédiction inter. On peut toutefois constater que l'efficacité de codage du SHVC-AE en mode de qualité FILE est comparable à celle du codeur SHM.

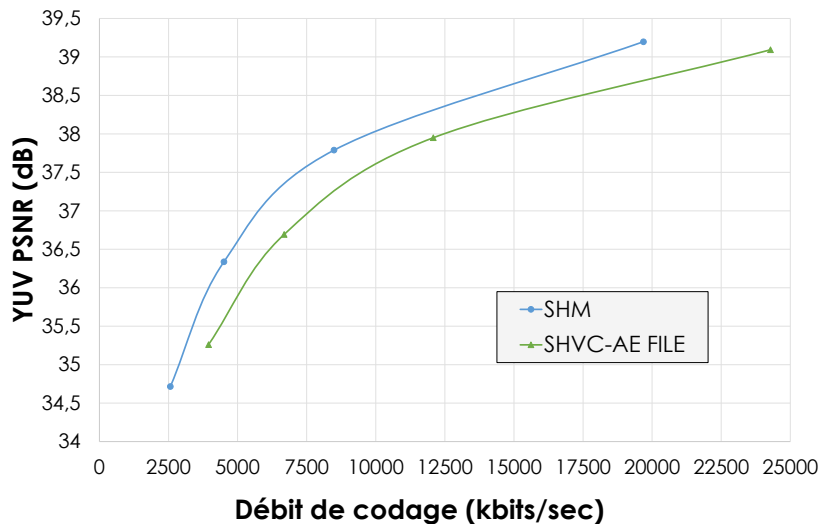


FIGURE 4.9 – Courbes débit/distorsion des codeurs SHVC-AE FILE et SHM sur les couches de base et de rehaussement sur la séquence B4 en configuration RA.

La Table 4.9 présente les gains en débit de la prédiction inter-couches, exprimés en BD-BR, du SHVC-AE par rapport au SLE dans la configuration FILE. On peut constater une amélioration de la prédiction inter-couches dans les configurations AI, LD-P et RA par rapport aux résultats présentés respectivement dans les Tables 3.9, 3.11 et 3.14 du chapitre précédent. Cette amélioration est due à l'emploi du sous-échantillonnage sur 8 coefficients et du compromis réalisé entre distorsion et efficacité de codage globale proposé sur son paramétrage. Néanmoins, cette amélioration ne permet pas d'égaliser l'efficacité de codage du codeur SHM. Cet écart avec le SHM est toutefois compensé par les performances du SHVC-AE en mode FILE en terme de vitesse de codage.

La table 4.9 présente dans ses dernières colonnes les vitesses de codage du SHVC-AE en mode FILE ainsi que les rapports des vitesses de codage du SHVC-AE FILE par rapport au SHM. Le traitement est réalisé sur la plateforme E5-4627V2 avec une mise en concurrence des tâches sur les 32 cœurs. On constate que les vitesses

TABLE 4.9 – Écart en BD-BR du SHVC-AE en mode FILE par rapport au SLE sur les couches de rehaussement. Vitesses de codage du SHVC-AE. Rapport des vitesses de codage du SHVC-AE / SHM.

Séquences	BD-BR SHVC-AE / SLE sur les couches de rehaussement			Vitesses de codage (IPS)			Rapport des vitesses SHVC-AE / SHM		
	AI	LD-P	RA	AI	LD-P	RA	AI	LD-P	RA
UHD1	-49.5%	-40.9%	-31.5%	0.46	0.33	0.33			
A1	-42.6%	-39.0%	-19.1%	0.87	0.60	0.64	51.9	63.9	70.6
A2	-46.7%	-41.4%	-27.7%	0.88	0.65	0.64	52.8	91.1	102.0
B1	-50.5%	-44.1%	-30.2%	1.55	1.15	1.19	26.0	64.6	60.6
B2	-35.8%	-33.1%	-17.1%	1.49	1.06	1.08	39.5	59.4	54.5
B3	-31.9%	-30.2%	-15.0%	1.50	1.11	1.14	31.0	62.0	58.7
B4	-26.1%	-25.7%	-12.5%	1.53	1.10	1.11	30.4	67.0	62.8
B5	-18.4%	-19.3%	-8.2%	1.51	1.07	1.05	31.1	60.1	56.6
Moyenne	-36.0%	-33.3%	-18.5%	1.33	0.96	0.98	37.5	66.9	66.5

sont bien supérieures au SHM. En effet, les compromis sur les outils dans la décision permettent d’accélérer la vitesse de codage du SHVC-AE par rapport au SHM. De plus, afin d’améliorer la vitesse de traitement du mode de qualité FILE, celui-ci emploie plusieurs optimisations bas-niveau permettant d’accélérer le traitement de certaines opérations. Ces optimisations prennent notamment effet sur les opérations ayant lieu au niveau pixel. Parmi ces opérations, on trouve par exemple le filtrage anti-blocs, les opérations de transformations aux différentes tailles de TU et les opérations de compensation de mouvement. Ces diverses optimisations permettent d’accélérer le traitement de certaines opérations mais ne garantissent pas des performances temps-réel. En effet, la prédiction emploie de nombreuses tailles de CU. La recherche sur toutes les tailles de blocs en comparant tous les outils de prédiction requiert un temps de traitement important. Une sélection sur une partie de ces outils et sur une partie des blocs permet d’accélérer encore davantage la vitesse de codage au détriment de l’efficacité de compression.

4.4.2 Décision de codage en configuration LIVE

La configuration LIVE cherche à réaliser un codage sous une contrainte temps-réel pour des formats HD et UHD. Le temps de traitement du codage vidéo dépend non seulement de la complexité du codeur mais aussi de la quantité de données à traiter qui dépend du format vidéo employé : résolution, fréquence, profondeur... Afin de réduire le temps de traitement, la complexité du codeur SHVC-AE a été réduite en utilisant seulement une partie des outils de codage. Les outils subissant ces restrictions dépendent du format de la vidéo et notamment de la résolution avec une fréquence et une profondeur maximale respectivement fixées à 60 IPS et 10 bits par pixel. Cette fréquence et cette profondeur sont fixées à partir des tendances des codages pour la diffusion. En fixant une valeur maximale pour ces caractéristiques, seul la résolution constitue une source de complexité pour le codage en temps-réel. La Table 4.10 résume les différents outils de codage employés dans la décision pour le traitement temps-réel. Le schéma du pipeline du SHVC-AE employé durant les tests est présenté en Figure 4.10. La décision de codage et les filtres de sous-échantillonnage et de sur-échantillonnage sont accélérés. Contrairement aux conditions de test du chapitre précédent (voir Figure 3.12, ces dernières étapes de pipeline durent moins longtemps. Puisque le temps de traitement est lié à la quantité d’informations à coder, la configuration LIVE peut être déclinée en fonction de la résolution de la séquence traitée. On nommera ainsi les configurations :

- LIVE SD pour les encodages LIVE sur des résolutions inférieures ou égales à $960 \times 540p$,
- LIVE HD pour les encodages LIVE sur des résolutions strictement supérieures à $960 \times 540p$ et inférieures à $2560 \times 1600p$,
- LIVE UHD pour les encodages LIVE sur des résolutions strictement supérieures à $2560 \times 1600p$,

Ces différents modes de qualité LIVEs sont appliqués de la même manière aux différentes couches du SHVC-AE. De plus, il est également possible d’appliquer ces modes LIVEs au SLE puisqu’elles consistent à optimiser les outils issus du standard HEVC.

Les différents modes de qualité considérés peuvent également employer un découpage sur les Tiles. Ce découpage permet de diviser la décision de codage en plusieurs tâches indépendantes. Ces décisions sont alors traitées en parallèle. Tous les modes de qualité LIVEs emploient 16 Tiles par image réparties en 4×4 , c’est à dire 4 Tiles sur

TABLE 4.10 – Outils de codage employés dans les décisions des modes FILE et LIVE appliqués à différentes résolutions.

Outils de codage	FILE	LIVE SD $\leq 960 \times 540p$	LIVE HD $> 960 \times 540p$ et $\geq 2560 \times 1600p$	LIVE UHD $> 2560 \times 1600$
Taille des CU	64×64, 32×32, 16×16, 8×8	64×64, 32×32, 16×16, 8×8	32×32, 16×16, 8×8	32×32, 16×16
Partition des PU	2N×2N, 2N×N, N×2N, N×N	2N×2N, 2N×N, N×2N, N×N	2N×2N	2N×2N
Taille des TU	toutes	toutes	16×16, 8×8	16×16, 8×8
Nombre de références temporelles par liste (L0 et L1)	8	2	1	1
Prédicteur INTRA	DC, Planar, 27 directions	DC, Planar, 9 directions	DC, Planar	DC, Planar
Nombre de Prédicteurs INTER	11	7	3	3
Split INTER	16×8, 8×16, 8×8	16×8, 8×16, 8×8	non	non

TABLE 4.11 – Comparaisons BD-BR couches de rehaussement SHVC-AE par rapport au SLE équivalent sur dans les modes de qualité LIVE-SD, LIVE-HD et LIVE-UHD en configuration AI, LD-P et RA.

Séquences	BD-BR SHVC-AE par rapport au SLE sur les couches de rehaussement								
	LIVE-SD			LIVE-HD			LIVE-UHD		
	AI	LD-P	RA	AI	LD-P	RA	AI	LD-P	RA
UHD1	-51.0%	-43.7%	-33.4%	-55.3%	-44.4%	-33.6%	-53.9%	-43.3%	-33.5%
A1	-48.5%	-40.3%	-24.2%	-50.0%	-40.3%	-21.6%	-49.8%	-36.7%	-18.9%
A2	-52.2%	-44.6%	-33.9%	-55.5%	-44.4%	-33.7%	-53.9%	-42.1%	-32.5%
B1	-53.7%	-43.3%	-34.5%	-55.8%	-43.8%	-34.3%	-51.1%	-35.3%	-30.2%
B2	-40.2%	-32.6%	-18.4%	-42.8%	-33.5%	-16.7%	-40.9%	-28.2%	-13.3%
B3	-37.1%	-31.1%	-17.9%	-42.5%	-34.2%	-16.9%	-40.7%	-32.7%	-17.7%
B4	-30.7%	-26.7%	-17.9%	-38.6%	-31.4%	-19.0%	-34.6%	-24.6%	-15.5%
B5	-22.8%	-15.8%	-1.2%	-30.8%	-21.6%	-2.0%	-29.5%	-20.0%	-1.8%
Moyenne	-40.7%	-33.5%	-21.1%	-45.1%	-35.6%	-20.6%	-42.9%	-31.4%	-18.6%

les lignes et 4 Tiles sur les colonnes. Le mode FILE, quant à lui, emploie une répartition de Tiles 1×1.

Le gain en débit du SHVC-AE par rapport au SLE est présenté dans la Table 4.11 dans les différents modes de qualité LIVES. Ces modes de qualité sont appliqués aussi bien au SHVC-AE qu'au SLE. On constate que la tendance des gains de la prédiction inter-couches est conservée. En effet, ces gains sont à peu près du même ordre de grandeur entre les modes FILE, LIVE-SD, LIVE-HD et LIVE-UHD : environ 40%, plus de 30% et environ 20% respectivement pour les configurations AI, LD-P et RA. Quelques différences peuvent toutefois être relevées entre ces gains : par exemple -45.1% et -40.7% en configuration AI respectivement pour les modes LIVE-SD et LIVE-HD. Ces écarts ne peuvent pas être directement évalués avec ces résultats puisque les références BD-BR employées sont différentes : SLE en mode de qualité LIVE-SD, LIVE-HD et LIVE-UHD.

La Table 4.12 présente la comparaison en terme de BD-BR des modes LIVES proposés par rapport à une référence en mode FILE sur les débits des couches de base et de rehaussement. La référence en terme de qualité PSNR est issue de la couche de rehaussement du SHVC-AE. La référence BD-BR est alors la même pour chaque comparaison : la couche de rehaussement du SHVC-AE en mode FILE. L'efficacité de codage est significativement moins importante sur les modes LIVES par rapport à celle du mode FILE puisque qu'elles emploient moins d'outils de codage. De plus, l'utilisation des Tiles provoque aussi une perte d'efficacité de codage comme expliqué dans [71]. L'écart entre les différents modes LIVES est également important : environ 50% d'écart entre les modes LIVE-SD et LIVE-UHD en configuration RA. On constate que ces écarts sont plus importants sur les configurations plus

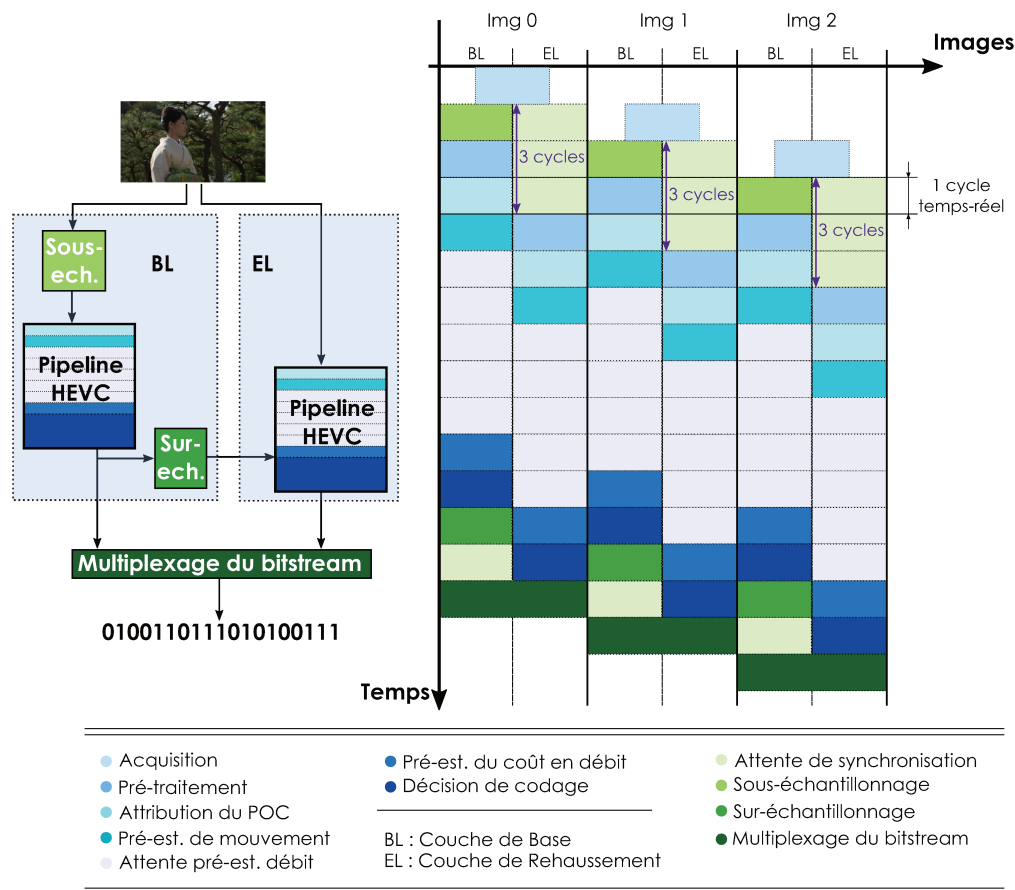


FIGURE 4.10 – Schéma de pipeline du SHVC-AE employé en configuration LIVE. Les filtres sous-échantillonnage et de sur-échantillonnage sont réalisés sur la couche de base. La décision de codage et les filtres prennent autant de temps que les autres étapes du pipeline.

complexe qui emploient en conséquence davantage de prédiction temporelle. Les figures 4.11 et 4.12 présentent les débits d’un codage HM dans les mêmes conditions respectivement pour les séquences B2 et B5. Ces courbes sont obtenues à partir d’un outils d’analyse de flux HEVC fourni par Tektronix [72]. La séquence B2 requiert beaucoup de données pour ses images I mais beaucoup moins pour les images B. La séquence B5 requiert beaucoup de données pour les images I également et ses images B requiert davantage de données par rapport à la séquence B2. Par conséquent, on peut donc dire que la séquence B5 est plus complexe que la séquence B2 puisqu’elle requiert davantage de débit pour un même type d’image. La séquence B5 a donc tout intérêt à employer davantage la prédiction temporelle puisque qu’elle présente un coût en débit inférieur à la prédiction intra.

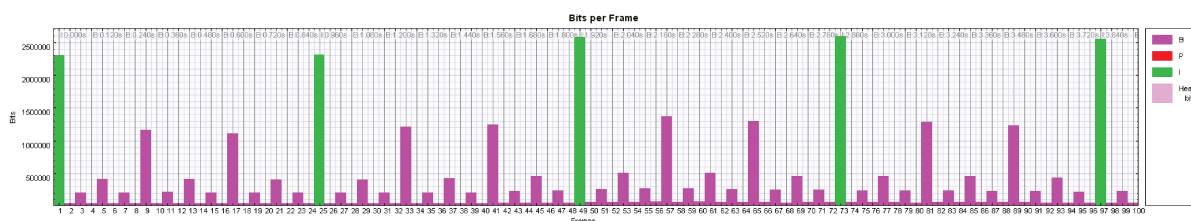


FIGURE 4.11 – Courbes des débits par images sur la séquence B2 avec un analyseur de Tektronix [72]

Par ailleurs, la sélection des outils de codage affecte tout particulièrement la prédiction temporelle. La prédiction temporelle repose en partie sur l’efficacité de la prédiction spatiale. En effet, la première image de référence est une

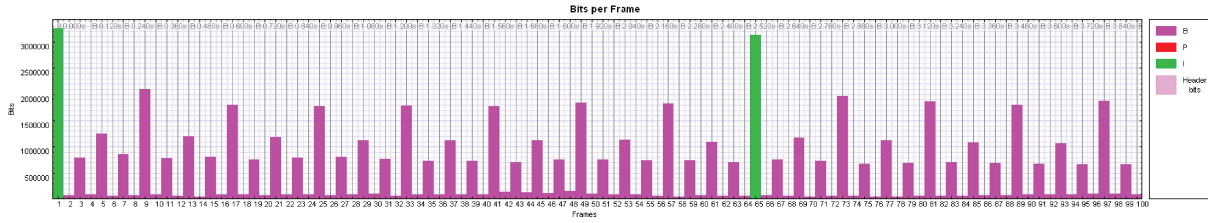


FIGURE 4.12 – Courbes des débits par images sur la séquence B5 avec un analyseur de Tektronix [72]

TABLE 4.12 – Comparaison BD-BR des différents modes de qualité FILE et LIVE appliqués aux séquences des CTC sur le SHVC-AE avec le mode FILE en référence sur les couches de base (BL) et de rehaussement (EL).

Séquences	BD-BR SHVC-AE sur les BL et EL dans les modes								
	LIVE-SD / FILE			LIVE-HD / FILE			LIVE-UHD / FILE		
	AI	LD-P	RA	AI	LD-P	RA	AI	LD-P	RA
UHD1	4.2%	6.4%	10.8%	13.9%	22.3%	28.1%	24.9%	39.4%	49.2%
A1	2.8%	5.7%	11.7%	11.6%	17.3%	30.2%	23.3%	33.3%	50.2%
A2	4.7%	6.9%	11.4%	15.9%	24.8%	29.7%	28.4%	44.6%	54.0%
B1	3.5%	7.5%	14.2%	11.6%	15.0%	25.5%	24.8%	38.2%	49.2%
B2	2.0%	5.1%	12.4%	6.6%	12.3%	26.5%	14.6%	26.6%	45.5%
B3	3.6%	7.2%	17.0%	12.8%	18.3%	38.1%	23.7%	37.4%	66.8%
B4	4.7%	10.1%	18.8%	15.9%	20.9%	33.0%	41.0%	58.8%	69.2%
B5	2.5%	9.5%	29.6%	12.1%	18.7%	42.6%	20.7%	39.8%	98.6%
Moyenne	3.4%	7.4%	16.4%	12.3%	18.2%	32.2%	25.2%	39.8%	61.9%

image I qui emploie exclusivement la prédiction spatiale. Dans le cas où cette référence est dégradée, son utilisation dans la prédiction temporelle affectera la qualité de toutes les images prédites. Il est alors primordiale de conserver une prédiction spatiale efficace afin que la prédiction temporelle puisse en bénéficier. C'est pourquoi la réduction de la complexité sur la prédiction repose majoritairement sur la prédiction temporelle. On peut d'ailleurs constater que, dans le SHVC-AE avec les modes de qualité proposés, la réduction de la complexité sur la prédiction spatiale seule (mode AI) diminue l'efficacité de codage moyenne de 3.4% à 25.2%. Ces résultats doivent être pondérés par les performances en terme de vitesse des codages obtenus avec ces différents modes de qualité.

La Table 4.13 présente les rapports des vitesses de codage des modes LIVEs par rapport au mode FILE. On constate que là où les écarts en terme d'efficacité de codage étaient importants par rapport au mode FILE, ils le sont tout autant en terme de vitesse de codage. En effet, la sélection des outils de codage proposée permet d'améliorer significativement la vitesse du codage du SHVC-AE. Le codage SHVC peut alors être effectué en moyenne au moins 13,5 ou au mieux 128,8 fois plus rapidement par rapport au mode qualité FILE. Les modes de qualité LIVE-SD, LIVE-HD et LIVE-UHD permettent alors de réaliser un compromis entre l'efficacité de codage d'une part et la vitesse de codage d'autre part. Ces modes de qualité permettent au SHVC-AE de respecter la contrainte temps-réel pour des résolutions données.

Les vitesses de codage sur la couche de base et sur la couche de rehaussement sont présentées respectivement dans les Tables 4.14 et 4.15 en IPS. On portera une attention particulière aux vitesses de codage sur la couche de rehaussement. En effet, la quantité de données à traiter y est plus importante que sur la couche de base. De plus, l'utilisation de la prédiction inter-couches augmente le temps de traitement par rapport à un codage HEVC équivalent. On constate que les vitesses de codage des couches de rehaussement peuvent respecter les contraintes temps-réel en configuration de diffusion pour les modes LIVE-HD et LIVE-UHD sur les séquences A et B. Le traitement de la séquence UHD1 sous la contrainte temps-réel requiert l'emploi du mode de qualité LIVE-UHD sur la couche de rehaussement. D'un autre côté, la couche de base est en mesure de répondre à la contrainte temps-réel en configuration de diffusion à partir du mode de qualité LIVE-SD pour les séquences A et B et à partir du mode LIVE-HD pour la séquence UHD1. On notera tout de même que la contrainte temps-réel n'est pas respectée en configuration AI sur la séquence B5. La configuration AI n'est pas employée dans un contexte de diffusion, c'est pourquoi le respect de la contrainte temps-réel dans cette configuration est moins problématique que pour les

TABLE 4.13 – Comparaison des vitesses de codage des différents modes de qualité FILE et LIVE appliqués aux séquences des CTC sur le SHVC-AE avec le mode FILE en référence sur les couches de rehaussement.

Séquences	Rapport des vitesses SHVC-AE / SLE sur les couches de rehaussement								
	LIVE SD / FILE			LIVE HD / FILE			LIVE UHD / FILE		
	AI	LD-P	RA	AI	LD-P	RA	AI	LD-P	RA
UHD1	13.4	14.2	21.3	49.1	45.8	47.9	108.7	90.6	90.3
A1	13.2	21.1	28.7	47.9	66.6	62.9	108.2	125.9	114.1
A2	13.3	22.0	35.8	49.0	63.3	72.9	109.0	122.2	123.4
B1	14.0	21.5	31.7	46.6	61.7	63.5	123.1	122.1	126.3
B2	13.3	22.1	33.5	43.5	61.6	69.7	115.7	127.0	133.1
B3	13.6	21.6	33.1	45.4	62.8	66.2	118.5	127.3	133.4
B4	13.8	22.3	32.3	45.3	63.1	68.8	118.9	126.9	132.9
B5	13.3	21.5	37.5	43.6	61.0	75.4	107.6	123.8	138.7
Moyenne	13.5	21.7	33.2	45.9	62.9	68.5	114.4	125.0	128.8

TABLE 4.14 – Vitesses de codage des différents modes de qualités LIVEs appliqués aux séquences des CTC sur le SHVC-AE sur la couche de base.

Séquences	Vitesses(IPS) LIVE SD			Vitesses(IPS) LIVE HD			Vitesses(IPS) LIVE UHD		
	AI	LD-P	RA	AI	LD-P	RA	AI	LD-P	RA
UHD1 (30 IPS)	20.6	21.2	29.2	43.7	43.3	47.4	113.7	102.5	99.8
A1 (30 IPS)	33.0	32.6	43.2	81.8	79.9	79.0	81.6	80.4	101.9
A2 (30 IPS)	33.3	42.5	58.4	83.0	82.5	89.9	81.5	82.3	105.5
B1 (24 IPS)	53.2	60.5	79.6	151.5	150.5	161.4	155.8	157.9	204.0
B2 (24 IPS)	50.5	61.1	80.9	138.2	151.9	160.1	157.8	160.4	204.7
B3 (50 IPS)	52.0	55.6	76.6	152.4	154.4	168.5	157.2	162.0	213.2
B4 (50 IPS)	51.8	55.6	74.2	149.7	156.0	164.6	156.5	159.0	207.9
B5 (60 IPS)	50.9	64.2	92.1	148.7	153.2	168.8	158.4	162.3	215.4
Moyenne	46.4	53.1	72.1	129.3	132.6	141.7	135.5	137.8	178.9

configurations de diffusion LD-P et RA. De plus, le codage dans la configuration AI n'a pas le même comportement en vitesse selon les modes de qualité. On constate, par exemple, que celui-ci s'exécute moins et plus rapidement que les configurations LD-P et RA respectivement dans les modes de qualité LIVE-SD et LIVE-UHD. Cette différence de comportement s'explique par la différence entre les prédictions spatiales employées dans ces modes de qualité. En effet, la prédiction spatiale, bien que généralement plus rapide que la prédiction temporelle, augmente davantage le temps de traitement du CABAC. Quand cette prédiction spatiale est très limitée, notamment par les tailles de CU employées, son impact sur le traitement du CABAC est alors moins important. Dans le cas du mode LIVE-UHD, la prédiction spatiale emploie seulement une taille minimale de CU de 16×16 contre 8×8 pour les autres modes de qualité. Généralement, le codage entropique consomme davantage de temps lorsqu'il doit traiter davantage de données. Le CABAC a donc plus de facilité à traiter moins de blocs de grandes tailles que plus de blocs de petites tailles. La prédiction spatiale génère alors une consommation du temps de traitement du CABAC moins importante dans le mode de qualité LIVE-UHD. C'est la raison pour laquelle le mode LIVE-UHD a un comportement, en terme de vitesse de codage, différent des autres modes de qualité en configuration AI.

Enfin, il est possible d'employer différents modes de qualité sur les couches de base et de rehaussement. En effet, la scalabilité spatiale nécessite un temps de traitement moins important sur la couche de base. Par conséquent, l'emploi d'un mode de qualité moins rapide sur la couche de base permettrait d'améliorer l'efficacité de codage de cette couche et de réduire le coût en débit global du codage SHVC. On nommera alors les modes de qualité :

- LIVE-HD+ : ce mode combine les modes de qualité LIVE-SD et LIVE-HD respectivement sur les couches de base et de rehaussement. Il permet un codage temps-réel sur des séquences HD de résolution $1920 \times 1080p$.
- LIVE-UHD+ : ce mode combine les modes de qualité LIVE-HD et LIVE-UHD respectivement sur les couches de base et de rehaussement. Il permet un codage temps-réel sur des séquences UHD de résolution $3840 \times 2160p$.

TABLE 4.15 – Vitesses de codage des différents modes de qualité LIVEs appliqués aux séquences des CTC sur le SHVC-AE sur la couche de rehaussement.

Séquences	Vitesses(IPS) LIVE SD			Vitesses(IPS) LIVE HD			Vitesses(IPS) LIVE UHD		
	AI	LD-P	RA	AI	LD-P	RA	AI	LD-P	RA
UHD1 (30 IPS)	6.2	6.6	9.8	22.7	21.1	22.1	50.1	41.8	41.6
A1 (30 IPS)	11.5	12.6	18.4	41.9	40.0	40.3	94.5	75.5	73.1
A2 (30 IPS)	11.7	14.2	23.0	43.0	41.0	46.8	95.6	79.2	79.3
B1 (24 IPS)	21.7	24.8	37.7	72.0	71.2	75.4	190.3	140.8	149.9
B2 (24 IPS)	19.8	23.5	36.3	65.0	65.3	75.5	172.6	134.7	144.1
B3 (50 IPS)	20.5	23.9	37.6	68.3	69.5	75.1	178.1	140.9	151.4
B4 (50 IPS)	21.0	24.5	35.9	69.3	69.3	76.5	181.7	139.3	147.8
B5 (60 IPS)	20.1	23.0	39.4	65.9	65.1	79.2	162.7	132.1	145.8
Moyenne	18.1	20.9	32.6	60.8	60.2	67.0	153.7	120.3	127.4

TABLE 4.16 – Gain en débit (BD-BR) sur la couche de rehaussement du SHVC-AE avec les modes de qualité LIVE-HD+ et LIVE-UHD+ par rapport aux codages sur le SLE équivalent.

Séquences	BD-BR SHVC-AE par rapport au SLE sur les couches de rehaussement					
	LIVE-HD+ SHVC-AE / SLE			LIVE-UHD+ SHVC-AE / SLE		
	AI	LD-P	RA	AI	LD-P	RA
UHD1	-58.3%	-56.8%	-37.4%	-55.8%	-45.3%	-35.2%
A1	-54.6%	-45.4%	-26.1%	-51.4%	-37.9%	-19.8%
A2	-60.0%	-50.9%	-38.0%	-56.4%	-44.8%	-35.1%
B1	-59.0%	-47.7%	-37.2%	-55.9%	-38.3%	-29.3%
B2	-46.2%	-37.3%	-19.7%	-42.4%	-28.9%	-13.8%
B3	-45.9%	-38.1%	-20.3%	-43.1%	-34.4%	-18.3%
B4	-41.6%	-34.9%	-21.6%	-39.5%	-29.0%	-17.8%
B5	-33.1%	-24.0%	-3.8%	-31.0%	-21.2%	-3.4%
Moyenne	-48.6%	-39.7%	-23.8%	-45.7%	-33.5%	-19.7%

La Table 4.16 présente l'efficacité de codage en terme de BD-BR sur le SHVC-AE dans les modes LIVE-HD+ et LIVE-UHD+ par rapport au SLE sur la couche de rehaussement équivalente. On constate que les gains en débit de la prédiction inter-couches sont plus importants avec les modes LIVE-HD+ et LIVE-UHD+ comparés respectivement aux modes LIVE-HD et LIVE-UHD. En effet, la référence inter-couches issue de la couche de base est de meilleure qualité objective comparée aux références de la couche de rehaussement. Ainsi, la prédiction inter-couches peut employer une référence de meilleure qualité. Les modes de qualité LIVE-HD+ et LIVE-UHD+ réalisent alors une prédiction inter-couches plus efficace par comparaison aux modes LIVE-HD et LIVE-UHD. Cette amélioration de la prédiction inter-couches combinée à l'amélioration de l'efficacité de codage de la couche de base permettent une double amélioration de l'efficacité de codage globale.

Comme dans le chapitre précédent, les tâches de traitement des couches de base et de rehaussement sont mises en concurrence. Elles disposent ici des 32 cœurs de la plateforme E5-4627V2. La Figure 4.13 montre les courbes débit/distorsion des couches de rehaussement des modes LIVE-HD et LIVE-HD+ sur la séquence A2. L'écart illustré ici montre l'intérêt d'employer un mode de qualité proposant une meilleure efficacité de codage sur la couche de base. La Table 4.17 présente cet écart avec la mesure BD-BR sur les couches de base et de rehaussement entre les codages des modes LIVE-HD+ et LIVE-HD ainsi qu'entre ceux des modes LIVE-UHD+ et LIVE-UHD avec respectivement les modes LIVE-HD et LIVE-UHD en référence. Ces améliorations en terme d'efficacité de codage n'impactent pas la vitesse de codage globale puisque les deux couches réalisent leurs traitements simultanément. De plus, les modes de qualité LIVE-HD+ et LIVE-UHD+ permettent de respecter la contrainte temps-réel respectivement sur le codage des séquences de résolution 1920×1080p et 3840×2160p.

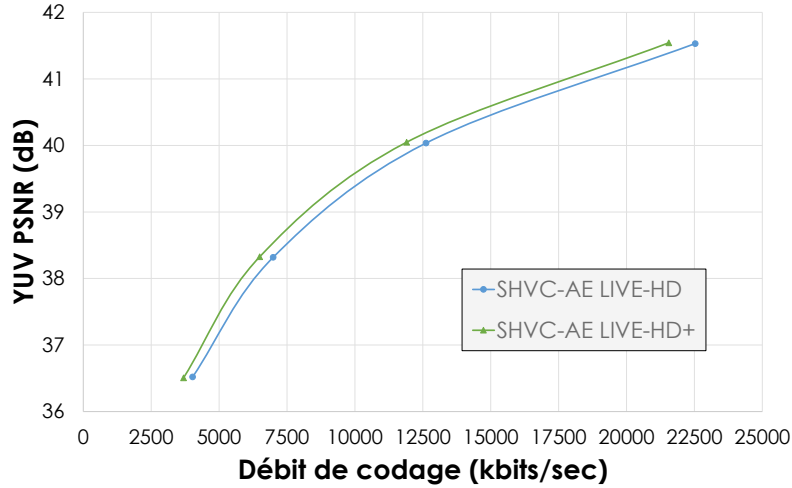


FIGURE 4.13 – Courbes débit/distorsion des modes LIVE-HD+ et LIVE-HD sur les couches de base et de rehaussement sur la séquence A2 en configuration RA.

TABLE 4.17 – Écart en BD-rate des modes de qualité LIVE-HD+ et LIVE-UHD+ respectivement par rapport aux modes LIVE-HD et LIVE-UHD sur les couches de base (BL) et de rehaussement(EL).

Séquences	BD-BR SHVC-AE sur les BL et EL dans les modes LIVE-HD+ / LIVE-HD / LIVE-UHD+ / LIVE-UHD					
	AI	LD-P	RA	AI	LD-P	RA
UHD1	-7.7%	-11.0%	-9.2%	-5.1%	-7.6%	-8.5%
A1	-6.9%	-7.0%	-7.6%	-5.2%	-5.6%	-6.3%
A2	-9.0%	-12.0%	-9.5%	-5.7%	-8.4%	-9.4%
B1	-5.9%	-4.2%	-5.4%	-8.1%	-8.4%	-6.0%
B2	-3.4%	-2.8%	-3.2%	-3.1%	-4.4%	-5.9%
B3	-6.0%	-5.4%	-5.8%	-4.5%	-6.0%	7.2%
B4	-6.6%	-5.6%	-5.5%	-11.8%	-10.8%	-7.9%
B5	-5.6%	-4.7%	-3.5%	-3.5%	-3.9%	-5.3%
Moyenne	-6.2%	-6.0%	-5.8%	-6.0%	-6.8%	-6.9%

TABLE 4.18 – Vitesses de codage des modes de qualité LIVE-HD+ et LIVE-UHD+ sur les couches de base (BL) et de rehaussement (EL).

Séquences	Vitesses (IPS) en mode LIVE-HD+						Vitesses (IPS) en mode LIVE-UHD+					
	BL			EL			BL			EL		
	AI	LD-P	RA	AI	LD-P	RA	AI	LD-P	RA	AI	LD-P	RA
UHD1 (30 IPS)	20.6	21.3	29.3	22.9	21.4	22.2	43.6	43.8	47.2	50.1	42.4	41.6
A1 (30 IPS)	33.0	32.7	43.1	42.3	40.5	40.6	80.1	79.7	79.2	94.4	75.5	73.0
A2 (30 IPS)	33.4	42.5	58.6	43.2	41.2	47.0	81.2	82.4	91.0	94.2	78.9	79.0
B1 (24 IPS)	53.7	60.8	80.0	72.3	71.9	75.6	149.0	149.6	161.0	189.1	140.1	148.7
B2 (24 IPS)	51.0	61.7	81.5	65.3	65.6	75.3	138.4	150.3	160.5	169.3	134.0	142.6
B3 (50 IPS)	52.2	56.0	76.9	68.8	69.6	75.0	150.7	153.4	168.4	178.5	140.0	150.5
B4 (50 IPS)	52.2	55.7	74.5	69.8	69.7	76.5	147.1	153.2	165.7	180.3	139.4	147.2
B5 (60 IPS)	51.2	64.8	92.6	66.3	65.2	79.1	146.1	152.3	168.8	163.7	131.8	144.7
Moyenne	46.7	53.5	72.4	61.1	60.5	67.0	127.5	131.6	142.1	152.8	120.0	126.5

La Table 4.18 présente les vitesses de codage obtenues avec les modes de qualité LIVE-HD+ et LIVE-UHD+ en configuration AI, LD-P et RA avec le SHVC-AE. On constate que le SHVC-AE est en mesure de répondre aux contraintes temps-réel dans une configuration de diffusion. La configuration AI ne permet pas le codage temps-réel sur certaines séquences, notamment B5. Néanmoins, les configurations LD-P et RA, plus généralement employées en diffusion, présentent des performances temps-réel sur les séquences A et B. La séquence UHD1 requiert l'emploi du mode de qualité LIVE-UHD afin d'être traitée en temps-réel.

Les différents compromis réalisés sur les outils de codage issus du standard HEVC permettent de réaliser des encodages temps-réel. De plus, les modes de qualité LIVE-HD+ et LIVE-UHD+ permettent d'adapter la complexité du codage aux différentes couches scalables en résolution. Les compromis proposés entre efficacité de codage et vitesse de traitement permettent au SHVC-AE d'être employé dans un contexte de diffusion temps-réel.

Conclusion

Les différentes optimisations proposées dans ce chapitre sur les opérations de filtrage de sous-échantillonnage et de sur-échantillonnage ont permis de réduire l'impact de la prédiction inter-couches sur le temps de traitement. Ces optimisations permettent, par ailleurs, d'envisager le traitement temps-réel. En effet, les parallélismes sur les composantes et sur les données combinés à l'utilisation d'instructions SIMD permettent de répondre à une contrainte temps-réel sur les opérations de filtrage pour des séquences $3840 \times 2160p$ à 120 IPS au maximum.

De plus, le choix du paramètre de Kaiser (K) du filtre de sous-échantillonnage avec 8 coefficients permet de réaliser une régulation du débit entre les couches de base et de rehaussement à moindre complexité. En effet, la variation de ce paramètre a permis de montrer que le coût en débit global à qualité constante pouvait être optimisé en réalisant un compromis entre distorsion et efficacité de codage. La perception de cette distorsion, résultante d'un effet d'aliasing, est jugée suffisamment faible pour ne pas représenter une gêne visuelle dans les conditions des expérimentations effectuées. L'efficacité de codage globale, de son côté, peut être améliorée de 0.8% en employant un bon paramétrage du filtre de sous-échantillonnage sur 8 coefficients par comparaison au filtrage employé dans le SHM sur 12 coefficients pour une scalabilité spatiale avec des résolutions deux fois moins hautes et deux fois moins larges sur la couche de base par rapport à la couche de rehaussement.

Ensuite, les différents modes de qualité proposés permettent de réaliser un compromis entre la vitesse et l'efficacité du codage. Ces modes emploient plus ou moins d'outils de codage issus du standard HEVC en fonction de la quantité de données à traiter. Les modes de qualité proposés permettent alors d'effectuer des encodages en condition temps-réel sur des images de résolution $3840 \times 2160p$. De plus, dans le cas de la scalabilité spatiale, ces modes de qualité peuvent être différents entre les couches de base et de rehaussement. Cette différence de mode entre les couches permet alors d'améliorer l'efficacité de codage globale : non seulement l'efficacité de codage de la couche de base est augmentée, mais la prédiction inter-couches est, elle aussi, améliorée puisqu'elle bénéficie d'une amélioration sur les informations de codage issues de la couche de référence. Cette double amélioration en terme d'efficacité de codage est par ailleurs réalisée pour des codages respectant la contrainte temps-réel.

Enfin, l'architecture du SHVC-AE permet de bénéficier de l'efficacité et de la vitesse de codage du SLE pour une scalabilité en résolution sur deux couches. Cette architecture est modulable et peut également être utilisée pour des codages sur davantage de couches et avec d'autres types de scalabilité. Le chapitre suivant propose un affinement de l'architecture employée dans le SHVC-AE afin de permettre des codages SHVC multi-couches et multi-scalabilités.

Chapitre 5

Généricité de la méthode employée

Introduction

Le codeur SHVC-AE permet de réaliser des codages SHVC dans différents modes de qualité qui donnent les possibilités suivantes :

- effectuer un codage SHVC cherchant à atteindre la meilleure efficacité de codage possible (FILE),
- effectuer un codage rapide permettant le codage temps-réel sur différentes résolutions (LIVE-SD, LIVE-HD et LIVE-UHD),
- effectuer un codage rapide et tenant compte de la résolution de l'image sur la couche de base afin d'améliorer l'efficacité de la prédiction inter-couches (LIVE-HD+, LIVE-UHD+).

Tous ces modes de qualité emploient une décision de codage améliorée par rapport à un codage simulcast grâce à la prédiction inter-couches. Cette prédiction emploie des images de référence de type I, P et B. Les images B servant de référence sont seulement celles qui servent également de référence temporelle dans la couche de base. La prédiction inter-couches en configuration random-access (RA) est par conséquent employée pour une image sur deux. De plus, les informations de codage exploitées par cette prédiction sont uniquement les textures de l'image reconstruite. Les vecteurs de mouvement issus de la couche de base présentent une faible amélioration de l'efficacité de compression pour un coût en mémoire important et une complexité augmentée dans le codeur de référence SHM. Dans le codeur proposé, SHVC-AE, l'utilisation de ces vecteurs de mouvement n'améliore pas l'efficacité de codage de la prédiction inter-couches, au contraire, dans l'état actuel, elle dégrade son efficacité de codage. Enfin, l'architecture employée dans le SHVC-AE permet de réaliser un parallélisme entre les traitements des images. La latence de codage entre les différentes couches est alors très faible. Celle-ci coûte seulement 3 cycles de pipeline, soit, par exemple, moins de 50 ms (i.e. $3 \text{ cycles} \times \frac{1}{60 \text{ IPS}}$) pour des codages temps-réel sur des séquences de fréquence d'image à 60 IPS.

L'objectif de ce chapitre est de montrer que l'architecture employée permet de réaliser des codages SHVC générique. Jusqu'alors, le SHVC-AE réalisait des codages uniquement en scalabilité spatiale sur deux couches. La généricité du codage scalable s'exprime alors sur les deux caractéristiques suivantes :

- le nombre de couche de rehaussement (codage multi-couches),
- le type de scalabilité.

C'est pourquoi nous proposons dans ce chapitre d'analyser le comportement théorique de l'architecture employée. Nous montrerons qu'elle permet de réaliser un codage multi-couches avec des résultats sur des codage à 3 couches en scalabilité spatiale. De plus, nous montrerons que l'architecture employée permet d'intégrer d'autres types de scalabilité au SHVC-AE. Les types de scalabilité étudiés sont notamment :

- la scalabilité sur la qualité (SNR),
- la scalabilité sur les vues (MV-HEVC),
- la scalabilité sur la profondeur des données (en bits par pixels),
- la scalabilité sur les codecs,
- la scalabilité sur les espaces colorimétriques (CGS).

La Figure 5.1 illustre le codage multi-couches et multi-scalabilité visé dans ce chapitre.

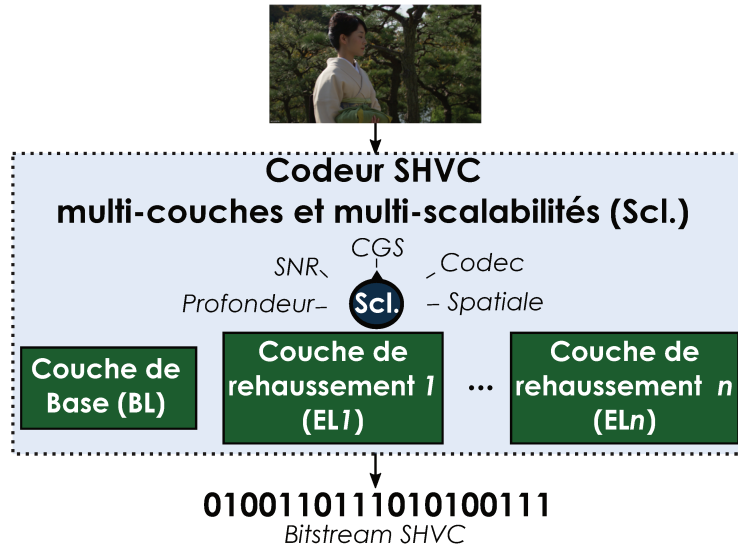


FIGURE 5.1 – Schéma du codeur SHVC multi-couches et multi-scalabilités visé.

5.1 Scalabilité spatiale sur plus de 2 couches

Le codeur SHVC-AE est en mesure de répondre à des contraintes d'efficacité de codage et de rapidité de traitement pour une scalabilité en résolution sur deux couches. L'architecture employée pour le codage SHVC permet de réaliser un codage simultané sur ces deux couches. Cette architecture peut être adaptée pour réaliser un codage sur davantage de couches. En terme de signalisation, certains paramètres du VPS et entête de slice concernant les dépendances entre les différentes couches doivent être configurés. La dépendance entre les couches est simple pour deux couches : la couche de base prédit la couche de rehaussement. Quand plus de deux couches sont employées, il est alors nécessaire de signaler les dépendances entre les différentes couches. L'objectif de cette section est de montrer que le SHVC-AE est en mesure de réaliser un codage multi-couches en scalabilité spatiale sans modifier son architecture.

5.1.1 Adaptation de la signalisation pour un codage multi-couches

Le codage multi-couches n'altère pas la structure de la syntaxe employée. L'ordre des éléments de syntaxe pour N couches est alors défini comme suit :

- 1) un VPS pour toutes les couches,
- 2) les SPS de chaque couche,
- 3) le PPS de la première couche,
- 4) l'entête de slice de la première image sur la première couche,
- ...
- $5 + 2 \times (N-2)$ le PPS de la dernière couche,
- $5 + 2 \times (N-2) + 1$ l'entête de slice de la première image sur la dernière couche,
- $5 + 2 \times (N-1)$ l'entête de slice de la deuxième image sur la première couche,
- $5 + 2 \times (N-1) + 1$ l'entête de slice de la deuxième image sur la deuxième couche,
- ...
- $5 + 2N$ l'entête de slice de la deuxième image sur la dernière couche,
- ...

Bien que l'ordonnancement des éléments de syntaxe ne change pas, la signalisation du codage multi-couches requiert une gestion plus précise des dépendances inter-couches. En effet, dans le cas d'un codeur SHVC avec deux couches, la couche de base prédit la couche de rehaussement. L'extension SHVC [60] précise qu'il ne peut y avoir qu'une seule référence inter-couches par couche pour des scalabilités spatiale ou sur les espaces de couleur. Cette restriction permet de fixer le nombre de sur-échantillonnage à 1 pour chaque image utilisant la prédiction inter-couches. Le nombre de références inter-couches employées est signalée dans l'entête de slice avec le champ

$inter_layer_pred_layer_idc[i]$, il vaut 1 pour des scalabilités spatiale et colorimétrique et peut être plus élevé pour les autres types de scalabilité. Ces références, issues du codage dans une autre couche, sont signalées dans le VPS au travers des dépendances inter-couches. Dans le cas d'un codage SHVC en scalabilité spatiale sur plus de deux couches, il ne peut alors y avoir qu'une seule couche de référence par couche prédite.

Les dépendances entre les différentes couches sont signalées dans le VPS. Dans un premier temps, une table contenant l'ensemble des dépendances directes et indirectes des prédictions inter-couches est signalée à partir de deux indices i et j correspondant aux indices de la couche courante et d'une autre couche. Une dépendance directe signifie que la couche i emploie une prédiction inter-couches avec une référence provenant de la couche j . Une dépendance indirecte signifie que la couche i emploie une référence inter-couches ne provenant pas de la couche j mais d'une tierce couche qui utilise une référence inter-couches, directement ou indirectement, depuis la couche j . Par exemple, dans la Figure 5.2, la 3^{ème} couche de rehaussement emploie une prédiction inter-couches avec pour référence la 2^{ème} couche de rehaussement qui en emploie une autre avec pour référence la 1^{ère} couche de rehaussement elle-même employant la couche de base en tant que référence inter-couches. Dans cet exemple, la 3^{ème} couche de rehaussement possède une dépendance directe envers la 2^{ème} couche de rehaussement et des dépendances indirectes envers la 1^{ère} couche de rehaussement et la couche de base. Pour une couche donnée, la table des dépendances permet de déduire le nombre de couches utiles et nécessaires à la prédiction inter-couches. Le champ correspondant à cette signalisation dans l'extension SHVCest le $layer_id_included_flag[i][j]$ avec i l'indice de la couche prédite et j l'indice de la couche de référence.

Ensuite, les dépendances directes entre couches sont signalées avec le champ $direct_dependency_flag[i][j]$. Ce champ indique si la i -ème couche emploie directement la j -ème couche en tant que référence inter-couches. Il permet notamment de déduire le nombre de références inter-couches employées pour chaque couche de rehaussement. Il permet également de déduire la présence ou non de couches indépendantes. Les couches indépendantes sont des couches de rehaussement qui n'emploient pas de référence inter-couches, pas même la couche de base. Enfin, après vérification de la valeur du champ $direct_dependency_flag[i][j]$, le type de prédiction inter-couches employé sur l'ensemble de la séquence est signalé. Le champ correspondant, $direct_dependency_type[i][j]$, permet de définir si la i -ème couche emploie la j -ème couche en tant que référence pour au choix :

- les informations de texture seulement,
- les vecteurs de mouvement seulement,
- les informations de texture et les vecteurs de mouvement.

Cette information sur le type de prédiction inter-couches permet alors d'activer ou non l'emploi de l'opération de sur-échantillonnage sur les textures et/ou sur les vecteurs de mouvement issus de la couche de référence. Tous les éléments de syntaxe précédemment cités permettent, pour une couche de rehaussement donnée, d'identifier :

- la couche de référence employée pour la prédiction inter-couches,
- le type de prédiction inter-couches employé,
- les différentes couches directement et indirectement nécessaires à la prédiction inter-couches,
- le nombre d'images de référence employées dans la prédiction inter-couches.

Il est nécessaire de définir les différentes dépendances ainsi que le type de scalabilité à l'initialisation du codage scalable. Ceci permet alors d'adapter la configuration de l'architecture scalable au besoin du codage.

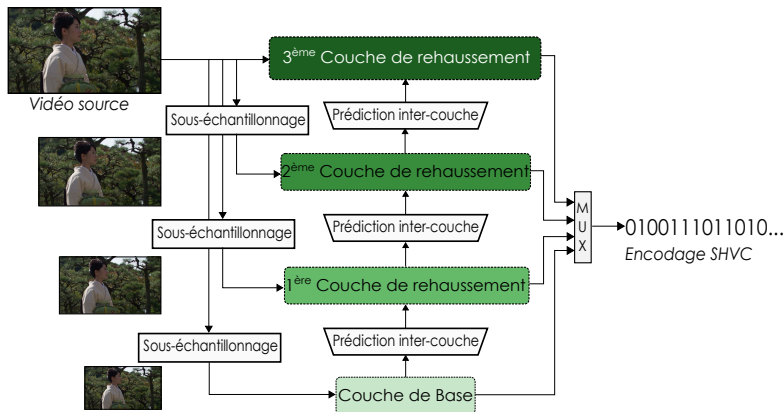


FIGURE 5.2 – Schéma d'un codeur SHVC à 4 couches spatiales.

TABLE 5.1 – Opérations requises pour les différents types de couche en scalabilité spatiale pour des codages FILE et LIVE.

Opération nécessaire	Couche de base		Couche de rehaussement intermédiaire		Couche de rehaussement finale	
	FILE	LIVE	FILE	LIVE	FILE	LIVE
Sous-échantillonnage	oui si pas de source correspondante	oui	oui si pas de source correspondante	oui	non	non
Sur-échantillonnage	non	non	oui	oui	oui	oui
Prédiction inter-couches	non	non	oui	oui	oui	oui
Multiplexage du bitstream	oui					

5.1.2 Adaptation de l'architecture pour un codage multi-couches

Afin de réaliser un codage multi-couches, chaque couche correspond à une instance d'un SLE. Comme vu dans le chapitre 3, ces couches réalisent leurs étapes de pipeline respectives simultanément mais avec un décalage à l'initialisation. Ce décalage repose sur les besoins de la prédiction inter-couches. Dans le cas du codage multi-couches en scalabilité spatiale, il est possible d'identifier 3 types de couche :

- la couche de base,
- les couches de rehaussement intermédiaires,
- la couche de rehaussement finale.

Comme explicité dans la Table 5.1, la couche de base et les couches de rehaussement intermédiaires requièrent chacune l'ajout d'une étape de sous-échantillonnage pour un codage LIVE. Pour un codage FILE, l'étape de sous-échantillonnage peut être effectuée en dehors de la chaîne de codage. Dans ce cas, le codeur reçoit autant de sources vidéos qu'il y a de couches. La couche de base réalise alors un codage HEVC. Ensuite, les couches de rehaussement intermédiaires et finale requièrent une étape de sur-échantillonnage pour leurs prédictions inter-couches. Pour un codage LIVE, les couches de rehaussement intermédiaires requièrent également l'ajout d'une étape de sous-échantillonnage.

On peut alors déduire, pour les différents types de couche, le nombre d'étapes de pipeline ajouté pour la prédiction inter-couches. On peut également déduire le décalage nécessaire à la synchronisation inter-couches. Cette synchronisation prend effet entre la fin du sur-échantillonnage d'une image de référence et la décision de codage de la couche de rehaussement qui l'utilise comme référence. La synchronisation se traduit par un décalage entre les instances de SLE au début du codage. Le décalage est directement lié au nombre d'étapes de pipeline réalisées sur les différentes couches. Ainsi, on peut distinguer les cas où le sur-échantillonnage est exécuté au niveau de la couche de référence (i.e. cas 1) ou bien au niveau de la couche prédite (i.e. cas 2).

Placement du sur-échantillonnage pour des codages FILE

Dans le cas du codage FILE, les Figures 5.3 et 5.4 illustrent respectivement l'exécution du sur-échantillonnage sur les couches de référence ou sur les couches prédites.

Dans les deux cas considérés, le nombre d'étapes de traitement entre la première acquisition d'une image et l'étape de multiplexage du bitstream est le même. La différence entre les cas 1 et 2 réside dans la durée d'attente entre les différentes couches à la fin du pipeline. L'unique contrainte est de réaliser la décision de codage d'une couche de rehaussement après le sur-échantillonnage dont dépend sa prédiction inter-couches. Ceci correspond à réaliser une synchronisation entre la fin d'une décision de codage sur une couche n pour une image i et à la fois le début de la décision de codage sur la couche n pour l'image $i+1$ et le début du sur-échantillonnage pour l'image i . En plaçant l'étape de sur-échantillonnage sur les couches prédites, le nombre d'étapes de pipeline des couches de rehaussement intermédiaires est plus important d'une étape par rapport à celui de la couche de base. Dans le cas 1, comme illustré en Figure 5.5, il en résulte plusieurs cycles d'attente avant le multiplexage du bitstream : 4 cycles pour la couche de base avec 4 couches. Dans le cas 2, comme illustré sur la Figure 5.6, le nombre d'étapes d'attentes est réduit à 2 cycles pour la couche de base avec 4 couches. Dans les deux cas, le décalage au début du pipeline s'élève à un cycle seulement entre une couche de référence et sa couche prédite. Pour le codage FILE,

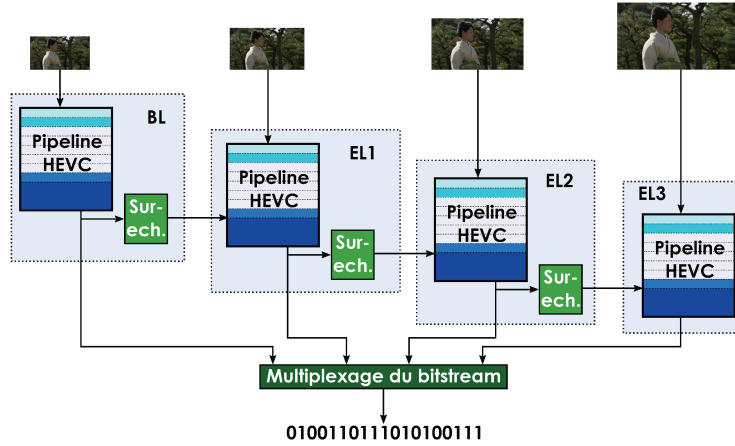


FIGURE 5.3 – Schéma du SHVC-AE en configuration FILE avec une couche de base (BL) et 3 couches de rehaussement (EL1, EL2, EL3). Le sur-échantillonnage est effectué sur les couches de référence.

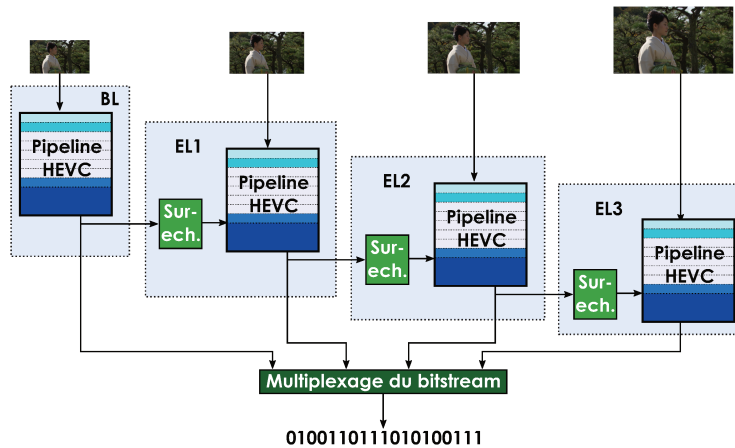


FIGURE 5.4 – Schéma du SHVC-AE en configuration FILE avec une couche de base (BL) et 3 couches de rehaussement (EL1, EL2, EL3). Le sur-échantillonnage est effectué sur les couches prédites.

l'exécution du sur-échantillonnage sur les couches de référence (cas 2) permet d'être plus rapide d'un cycle par rapport à l'exécution du sur-échantillonnage sur les couches prédites (cas 1).

Placement du sur-échantillonnage pour des codages LIVE

Dans le cas du codage LIVE, une étape de sous-échantillonnage vient s'ajouter au pipeline des couches de référence. Les Figures 5.7 et 5.8 illustrent respectivement l'exécution du sur-échantillonnage sur les couches de référence ou sur les couches prédites.

Pour les mêmes raisons que le codage FILE, l'exécution du sur-échantillonnage sur les couches de référence (cas 2) permet d'être plus rapide d'un cycle par rapport à l'exécution du sur-échantillonnage sur les couches prédites (cas 1). Toutefois, l'ajout du sous-échantillonnage sur les couches de référence requiert une étape d'attente supplémentaire sur la couche de rehaussement finale. Ce décalage s'explique par l'absence de sous-échantillonnage sur cette couche. En effet, quelque soit le placement du sur-échantillonnage, la taille du pipeline de cette couche est toujours moins importante que celle des couches de rehaussement intermédiaires. Une étape d'attente supplémentaire est donc nécessaire sur la couche de rehaussement finale afin d'attendre l'exécution du sous-échantillonnage sur sa couche de référence. Ce décalage peut alors être exprimé avec les formules suivantes respectivement avec le sur-échantillonnage sur les couches de rehaussement intermédiaires (EL-I) et sur la couche de rehaussement finale

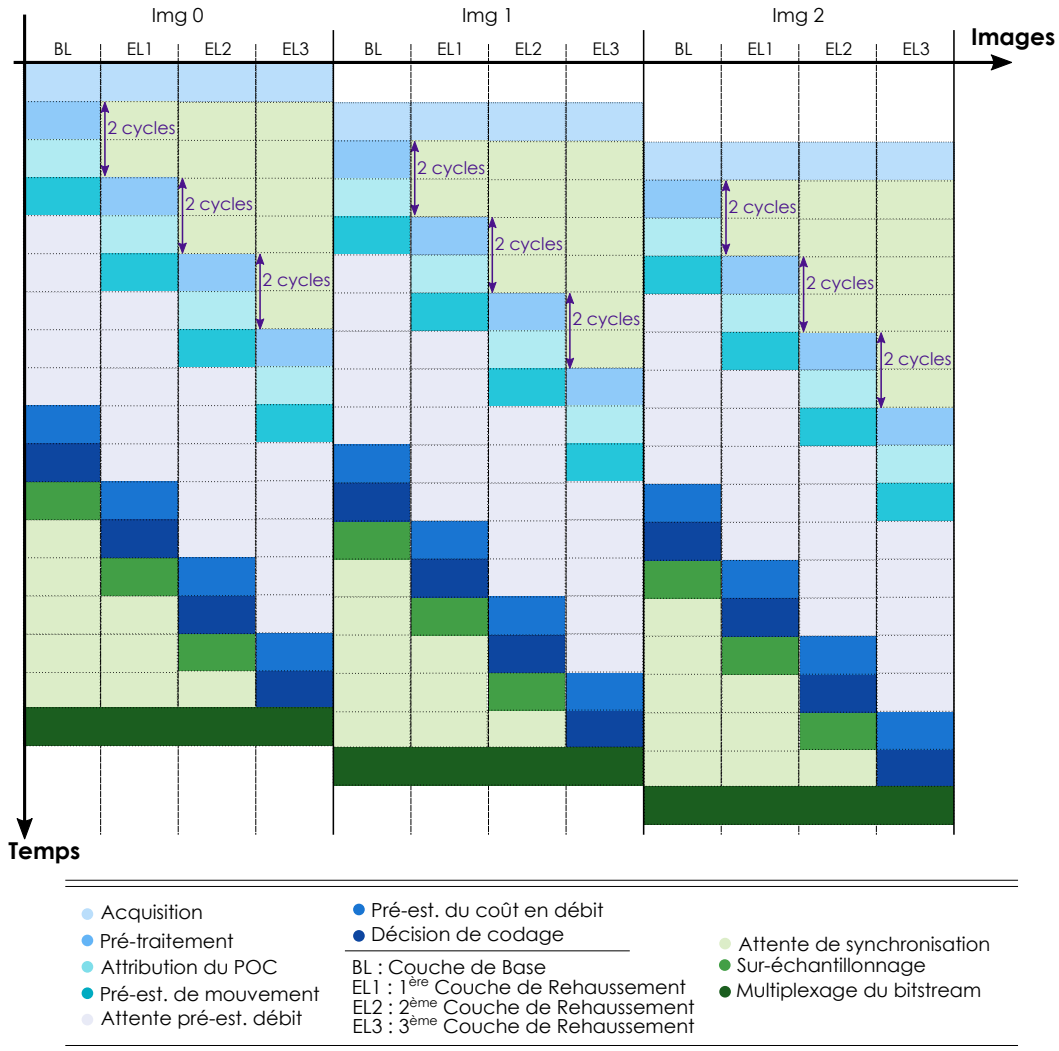


FIGURE 5.5 – Pipeline du SHVC-AE en mode FILE avec 4 couches et l’étape de sur-échantillonnage effectuée sur les couches de référence. Schéma de l’architecture correspondante en Figure 5.3

(EL-F) :

$$(EL - I) \text{ Décalage} = 2 \times \text{Indice de couche} - 1 \tag{5.1}$$

$$(EL - F) \text{ Décalage} = 2 \times \text{Indice de couche} \tag{5.2}$$

La valeur de l’Indice de couche vaut 0 pour la couche de base, 1 pour la première couche de rehaussement... N-1 pour la couche de rehaussement finale.

Placement du sous-échantillonnage pour des codages LIVE

En ce qui concerne le placement de l’étape de sous-échantillonnage, il existe plusieurs solutions :

- effectuer le sous-échantillonnage en amont du pipeline de codage (voir Figures 5.3 et 5.3),
- effectuer le sous-échantillonnage uniquement à partir de l’image source acquise (voir Figures 5.7 et 5.8),
- effectuer le sous-échantillonnage à partir de l’image source d’une autre couche (voir Figure 5.11).

La réalisation du sous-échantillonnage à partir de l’image source acquise uniquement revient à ajouter une étape de sous-échantillonnage pour chaque couche de référence. Dans ce cas, le SHVC-AE reçoit une consigne sur les résolutions des couches de référence. Sinon, la réalisation du sous-échantillonnage à partir de l’image source d’une autre couche revient à cascader les opérations de sous-échantillonnage. Ainsi, l’image source de la couche de base

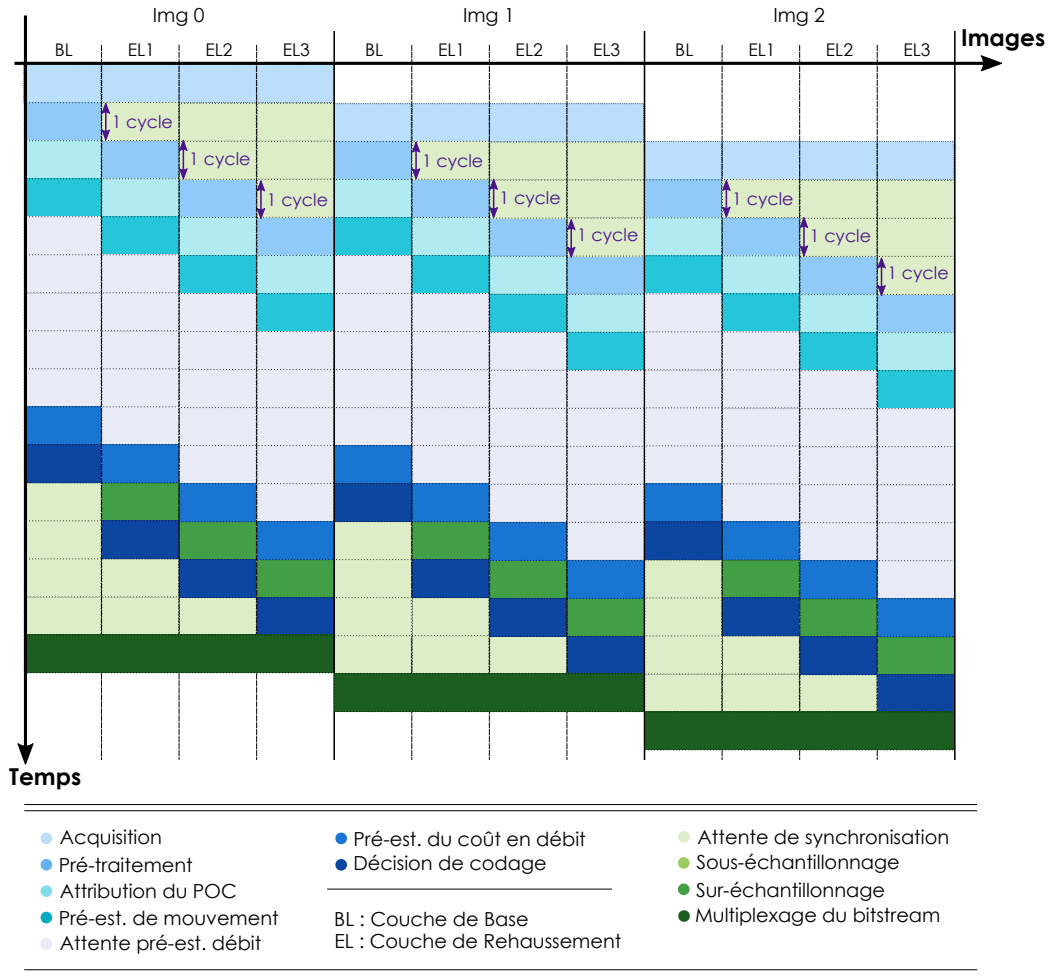


FIGURE 5.6 – Pipeline du SHVC-AE en mode FILE avec 4 couches et l’étape de sur-échantillonnage effectuée sur les couches prédites. Schéma de l’architecture correspondante en Figure 5.4

sera filtrée par autant d’étapes de sous-échantillonnage qu’il y a de couches de rehaussement intermédiaires. Dans certains cas, cette cascade de sous-échantillonnage peut présenter un intérêt du point de vue de l’efficacité du filtrage. En revanche, elle pose un soucis de latence du point de vue de l’architecture. En effet, le sous-échantillonnage de la couche de base sera retardé d’autant de cycle qu’il y a de couches de rehaussement intermédiaires. Comme illustré sur la Figure 5.12 avec 4 couches, la couche de base augmente son pipeline de 2 étapes d’attentes. De plus, ce retard est également ajouté à toutes les autres couches. Par conséquent, la latence globale du codage avec les sous-échantillonnages cascades est augmenté de 2 cycles par rapport au codage avec les sous-échantillonnages sur l’image acquise uniquement.

La solution par défaut retenue dans le SHVC-AE consiste à réaliser un sur-échantillonnage sur les couches prédites. De plus, les sous-échantillonnages sont réalisés à partir de l’image acquise (i.e. image source) uniquement. Les configurations FILE et LIVE par défaut correspondent alors respectivement aux Figures 5.3 et 5.7.

L’ajout de nouvelles couches de rehaussement modifie le pipeline du SHVC-AE sans altérer le principe de l’architecture employée. Ce principe repose sur un décalage entre couches afin de synchroniser leurs pipelines. Ce décalage induit une latence entre chaque couche qui requiert un stockage des informations de codage dans des buffers qui doivent être adaptés.

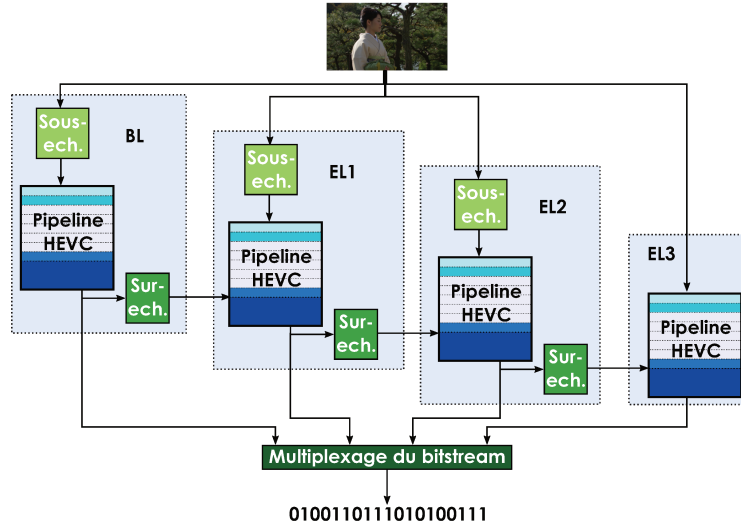


FIGURE 5.7 – Schéma du SHVC-AE en configuration LIVE avec une couche de base (BL) et 3 couches de rehaussement (EL1, EL2, EL3). Le sur-échantillonnage est effectué sur les couches de référence (cas 1).

TABLE 5.2 – Tailles des buffers partagés entre les couches de référence et prédites pour les cas d'usage présentés en Figure 5.13 avec un pipeline de codage tel que présenté en Figures 5.3, 5.4, 5.7 et 5.8.

Couches de rehaussement	Taille des buffers partagés					
	5.13a	5.13b	5.13c	5.13d	5.13e	5.13f
1280×720p	2	2	2	2	2	2
1920×1080p	4	2	4	4	2	2
3840×2160p	6	6	4	2	2	4

Stockage des informations de codage pour la prédiction inter-couches

En plus du stockage de l'image source requis par la synchronisation entre les instances de SLE, le codage multi-couches requiert également un stockage supplémentaire pour stocker les textures et les vecteurs de mouvement sur-échantillonnés. Le mécanisme employé avec deux couches consiste à stocker ces informations de codage dans des buffers partagés entre les couches de base et de rehaussement. Ce buffer doit conserver ces informations au moins deux cycles et doit donc pouvoir contenir, au minimum, deux images. Cette quantité correspond à l'écart maximum en nombre de cycles de pipelines entre la fin de la décision de codage de la couche de référence et le début de celle de la couche prédite. Dans le cas du codage multi-couches, cet écart n'est pas toujours de 2 cycles. L'ensemble de figures 5.13 illustre différentes dépendances possible pour une prédiction inter-couches en scalabilité spatiale avec 4 couches et sans couches de rehaussement indépendantes.

Pour chacun de ces cas d'usage, les besoins en terme de buffers de stockage des vecteurs et des images reconstruites ne sont pas les mêmes. La Table 5.2 donne les différentes tailles de buffers minimum requis pour ces différents cas d'usage.

Avec l'architecture en pipeline employée, on constate qu'il est plus intéressant, d'un point de vue gestion mémoire, de rapprocher les couches réalisant une prédiction inter-couches. Le cas d'usage présenté sur la Figure 5.13e donne une gestion mémoire plus intéressante que les autres puisque les prédictions inter-couches requièrent des buffers moins grand. Toutefois, il est possible d'améliorer la gestion mémoire des autres cas d'usage. En effet, il est possible d'aligner le codage des couches qui n'ont aucune dépendances directes ou indirectes. Si on prend l'exemple du cas présenté sur la Figure 5.13a, les couches de rehaussement ne dépendent que de la couche de base. Il serait alors possible de démarrer le pipeline de codage de toutes les couches de rehaussement simultanément. La taille des buffers de stockage pour les informations de codage pourrait alors être réduit. Le SHVC-AE multi-couches n'est pas optimal de ce point de vue. Chaque couche de rehaussement est décalée par rapport à la précédente couche, qu'elle soit sa couche de référence ou non. Un réglage dynamique de ce décalage est envisa-

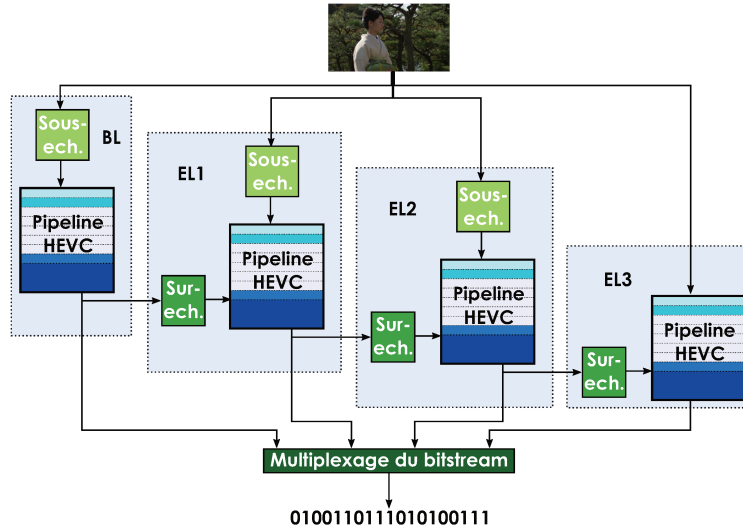


FIGURE 5.8 – Schéma du SHVC-AE en configuration LIVE avec une couche de base (BL) et 3 couches de rehaussement (EL1, EL2, EL3). Le sur-échantillonnage est effectué sur les couches prédites (cas 2).

geable. Toutefois, l'objectif principal durant ces travaux était de vérifier la faisabilité du codage multi-couches à partir du SHVC-AE.

En définitive, le codage multi-couches appliqué au SHVC-AE en scalabilité spatiale effectue :

- un codage dont les dépendances entre couches sont données à l'initialisation,
- un sous-échantillonnage par couche à partir de l'image source acquise ou bien une acquisition de plusieurs images sources dont certaines pré-sous-échantillonnées,
- un sur-échantillonnage sur les couches prédites,
- un stockage des informations de codage (textures et vecteurs de mouvement) sur les couches prédites dépendant du décalage total avec sa couche de référence.

Enfin, le codage multi-couches emploie les mêmes outils que le codage à deux couches pour les décisions. La prédiction inter-couches peut employer une image reconstruite et les vecteurs de mouvement issus de la couche de référence. L'image reconstruite provient du codage de la couche de référence pour la même image source. Quand ils sont exploités, les vecteurs de mouvement issus de la couche de référence sont utilisés en tant que prédicteur temporel (ou vecteur collocalisé). Par défaut, le SHVC-AE emploie uniquement les textures pour la prédiction inter-couches. De plus, la régulation du débit est réalisée indépendamment par couche. Dans le cas du codage à débit constant :

- soit la consigne de débit global est divisée entre les différentes couches,
- soit une consigne par couche est fournie.

Chaque couche dispose alors d'une consigne en débit qu'elle peut suivre indépendamment du codage des autres couches. Dans le cas d'une consigne de débit global, le débit de chaque couche est calculé par la formule suivante :

$$\text{Débit de la couche } idx = \text{Débit global} \times \frac{idx + 1}{\sum_{j=0}^{\text{Nombre de couches} - 1} j} \quad (5.3)$$

Avec idx l'indice de la couche de valeur 0 pour la couche de base, 1 pour la première couche de rehaussement... Et Nombre de couches - 1 pour la couche de rehaussement finale.

5.1.3 Application sur 3 couches spatiales

Les conditions de test sont les mêmes que dans les chapitres 3 et 4. Les encodages sont réalisés sur les séquences B1 à B5. Les couches de base (BL), de rehaussement intermédiaire (EL1) et de rehaussement finale (EL2) réalisent des encodages respectivement sur des séquences de résolution $960 \times 540p$, $1280 \times 720p$ et $1920 \times 1080p$. Dans les CTC [33], les séquences A1 et A2 ne permettent pas un ratio 1.5 employé entre EL1 et EL2. Ces dernières séquences

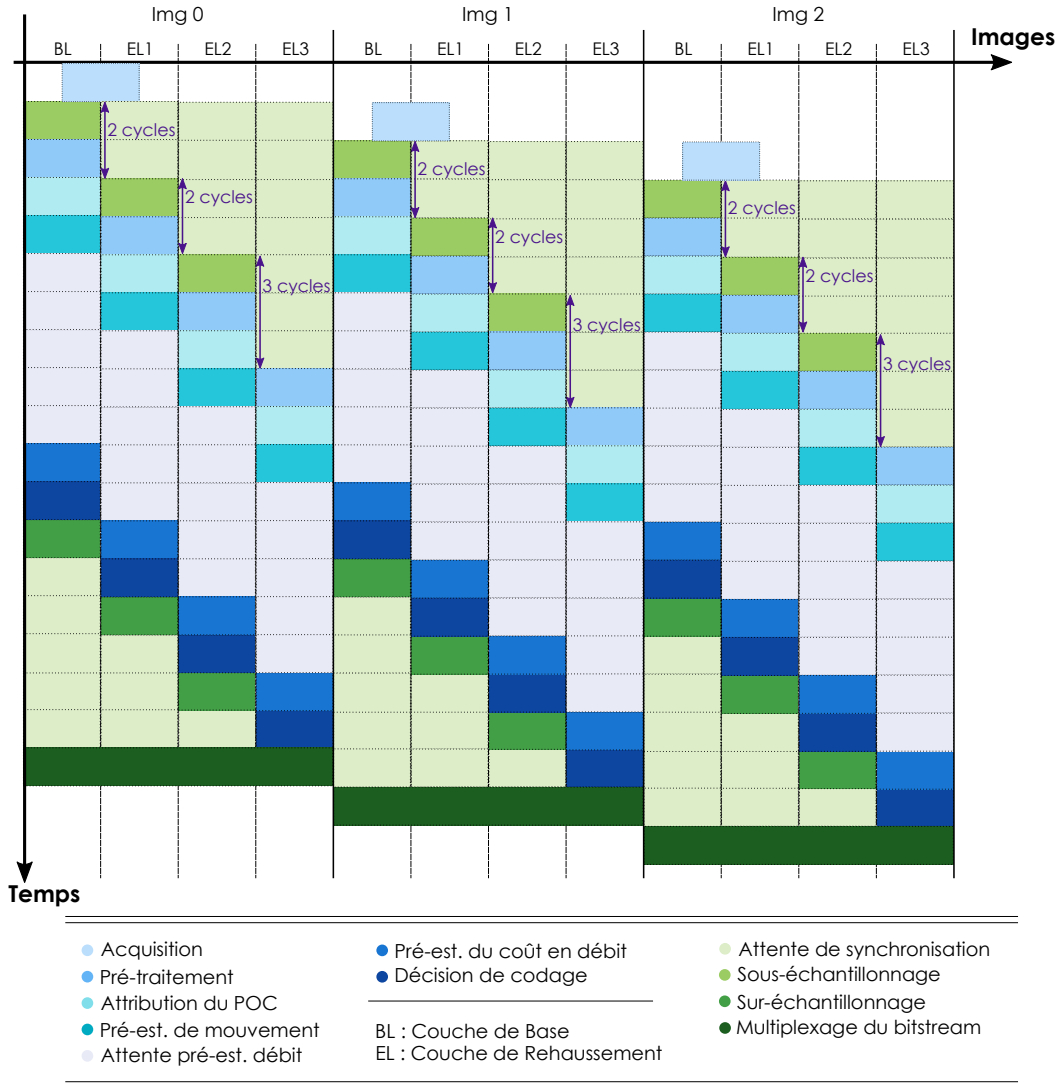


FIGURE 5.9 – Pipeline du SHVC-AE en mode LIVE avec 4 couches et l’étape de sur-échantillonnage effectuée sur les couches de référence. Schéma de l’architecture correspondante en Figure 5.7 (cas 1).

ne sont donc pas utilisées dans ces tests. Les encodages sont réalisés sur la plateforme Intel Xeon E5-4627V2 décrit en Table 3.6. Les encodages sont réalisés dans les modes de qualité FILE et LIVE-HD.

Les expérimentations conduites sur le codage multi-couches sont réalisées sur 3 couches spatiales. Ces 3 couches peuvent être interfacées selon deux configurations de dépendance inter-couches présentées en Figures 5.14 et 5.15. Dans le premier cas, la couche de rehaussement intermédiaire est d’abord prédite par la couche de base puis sert de référence à la couche de rehaussement finale. Cette première configuration des dépendances inter-couches sera nommée BL–EL1, EL1–EL2. Dans le second cas, la couche de base uniquement sert de référence inter-couches. Cette seconde configuration des dépendances inter-couches sera nommée BL–EL1, BL–EL2.

Les résultats présentés dans cette partie cherchent à quantifier l’efficacité de codage du SHVC-AE ainsi qu’à évaluer sa vitesse de codage. Pour cela les codages du SHVC-AE sont comparés aux codages équivalent simulcast réalisés sur le SLE avec la métrique BD-BR. La référence en qualité est celle de la couche de rehaussement finale.

Les Table 5.3 et 5.4 présentent les comparaisons BD-BR des encodages pour les dépendances BL–EL1, BL–EL2 et BL–EL1, EL1–EL2 sur l’EL2 et l’EL2 HEVC équivalente respectivement pour les modes de qualité FILE et LIVE-HD. La prédiction inter-couches est plus efficace avec la dépendance BL–EL1, EL1–EL2. En effet, les informations de codage issues de l’EL1 sont plus proches en terme de résolution de l’EL2 par rapport à la BL. Ce

TABLE 5.3 – BD-BR entre la couche de rehaussement finale (EL2) par rapport au HEVC équivalent en mode de qualité FILE dans les configurations all-intra (ou intra-only) (AI), low-delay P (LD-P) et RA. La référence de qualité est celle de l'EL2. BL–EL1, BL–EL2 (Figure 5.14) : la couche de base (BL) prédit les couches de rehaussement EL1 et EL2. BL–EL1, EL1–EL2 (Figure 5.15) : la BL prédit l'EL1 puis l'EL1 prédit l'EL2.

Séquences	BL–EL1, BL–EL2			BL–EL1, EL1–EL2		
	AI	LD	RA	AI	LD	RA
B1	-50.5%	-44.1%	-30.2%	-46.2%	-34.9%	-27.0%
B2	-35.8%	-33.1%	-17.1%	-46.1%	-36.1%	-19.1%
B3	-31.9%	-30.2%	-15.0%	-46.0%	-36.8%	-21.0%
B4	-26.1%	-25.7%	-12.5%	-38.0%	-31.3%	-17.9%
B5	-18.4%	-19.3%	-8.2%	-38.5%	-26.8%	-7.5%
Moyenne	-32.5%	-30.5%	-16.6%	-43.0%	-33.2%	-18.5%

TABLE 5.4 – Idem Table 5.3 mais en LIVE-HD. BL–EL1, BL–EL2 (Figure 5.14) : la couche de base (BL) prédit les couches de rehaussement EL1 et EL2. BL–EL1, EL1–EL2 (Figure 5.15) : la BL prédit l'EL1 puis l'EL1 prédit l'EL2.

Séquences	BL–EL1, BL–EL2			BL–EL1, EL1–EL2		
	AI	LD	RA	AI	LD	RA
B1	-43.4%	-30.3%	-20.1%	-50.6%	-36.7%	-25.0%
B2	-34.5%	-26.2%	-9.5%	-45.9%	-35.8%	-18.7%
B3	-33.7%	-26.2%	-10.1%	-46.0%	-36.3%	-18.8%
B4	-29.2%	-22.8%	-8.1%	-39.2%	-31.0%	-14.9%
B5	-24.7%	-16.9%	-0.0%	-38.4%	-26.7%	-8.4%
Moyenne	-33.1%	-24.5%	-9.6%	-44.0%	-33.3%	-17.1%

TABLE 5.5 – BD-BR entre la somme des débits des couches (EL2+EL1+BL) par rapport à la somme des débits HEVC équivalents (EL2+EL1+BL) en mode de qualité FILE dans les configurations AI, LD-P et RA au niveau de qualité de l'EL2. La référence de qualité est celle de l'EL2. BL–EL1, BL–EL2 (Figure 5.14) : la couche de base (BL) prédit les couches de rehaussement EL1 et EL2. BL–EL1, EL1–EL2 (Figure 5.15) : la BL prédit l'EL1 puis l'EL1 prédit l'EL2.

Séquences	BL-EL1 BL-EL2			BL-EL1 EL1-EL2		
	AI	LD	RA	AI	LD	RA
B1	-38.9%	-32.6%	-25.8%	-42.6%	-35.2%	-28.2%
B2	-35.8%	-28.8%	-15.9%	-42.0%	-31.4%	-19.5%
B3	-35.2%	-29.7%	-17.3%	-41.5%	-34.0%	-21.4%
B4	-32.3%	-27.8%	-16.0%	-37.6%	-30.5%	-15.6%
B5	-28.7%	-20.8%	-4.5%	-35.9%	-23.4%	-4.5%
Moyenne	-34.2%	-27.9%	-15.9%	-39.9%	-30.9%	-17.8%

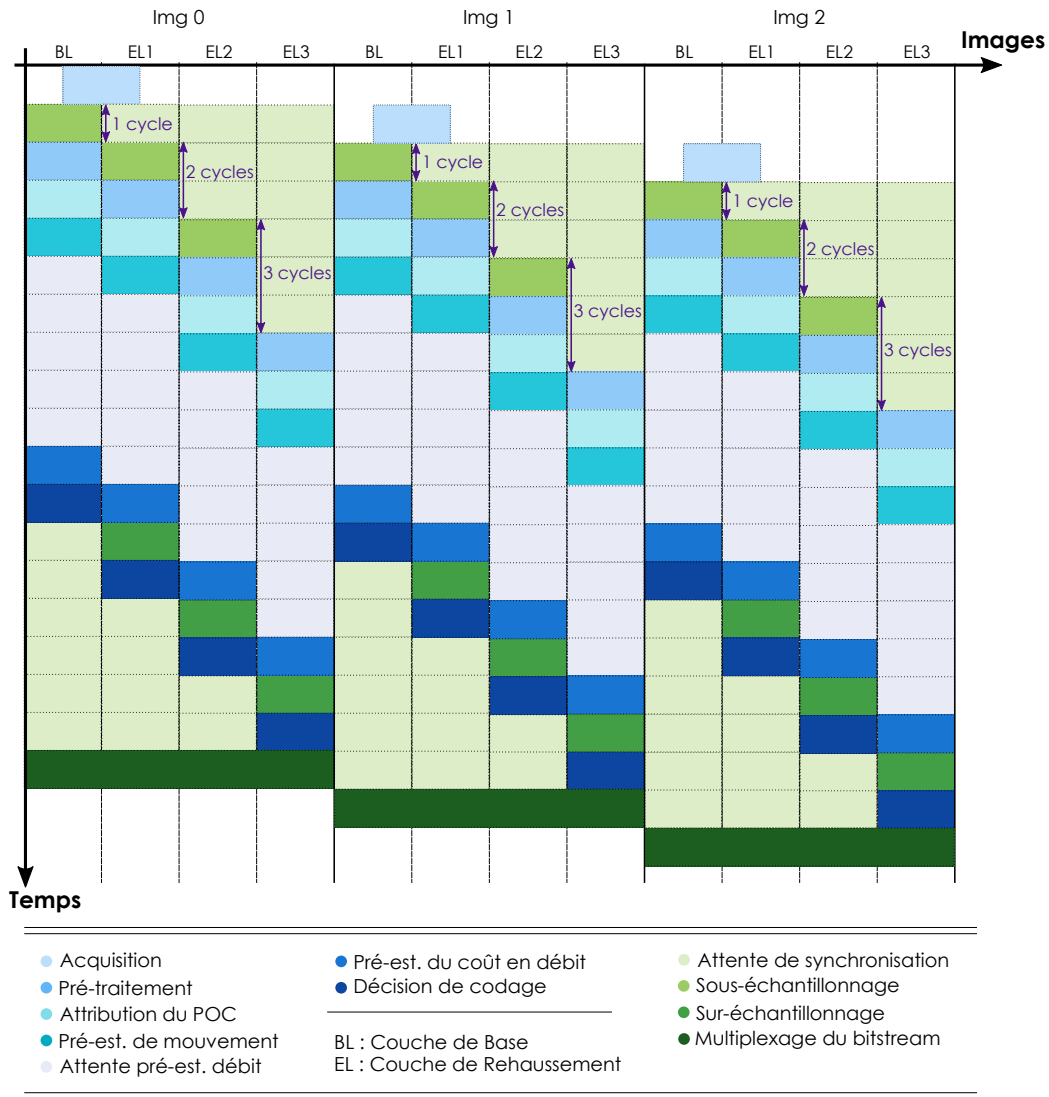


FIGURE 5.10 – Pipeline du SHVC-AE en mode LIVE avec 4 couches et l’étape de sur-échantillonnage effectuée sur les couches prédites. Schéma de l’architecture correspondante en Figure 5.8 (cas 2).

constat est vérifié dans les expérimentations sur deux couches menées dans [35]. De plus, pour les dépendances BL–EL1, BL–EL2, on s’attendrait à obtenir une amélioration de la couche de rehaussement équivalente à celle obtenue dans le chapitre 3. Le codage multi-couches augmente le décalage entre la couche de base et la couche de rehaussement finale par rapport aux encodages sur deux couches. Ce décalage n’est pas géré sur la fin du codage. Par conséquent, le codage multi-couches traite moins d’images sur l’EL2 par rapport à la couche de rehaussement dans un codage deux couches. L’absence des dernières images change les valeurs de débit et de PSNR sur cette couche. C’est pourquoi l’amélioration de codage sur l’EL2 semble différente même si, en réalité, le codage sur la couche de rehaussement finale est le même pour les configurations deux couches et BL–EL1, BL–EL2.

Les Table 5.5 et 5.6 présentent les comparaisons BD-BR des encodages avec les dépendances considérées sur la somme des débits des couches de base et de rehaussement entre le codage SHVC et le codage simulcast équivalent respectivement pour les modes de qualité FILE et LIVE-HD. Ces résultats permettent alors d’estimer l’économie en débit du codage SHVC par rapport au simulcast. On constate que cette amélioration est plus importante pour la dépendance BL–EL1, EL1–EL2 pour la même raison de proximité des résolutions entre l’EL2 et sa couche de référence. Pour les 3 résolutions 960×540p, 1280×720p et 1920×1080p, par comparaison au codage simulcast, le SHVC-AE permet d’économiser en moyenne 15.9% et 17.8% du débit en mode de qualité FILE et 15.8% et 18.8%

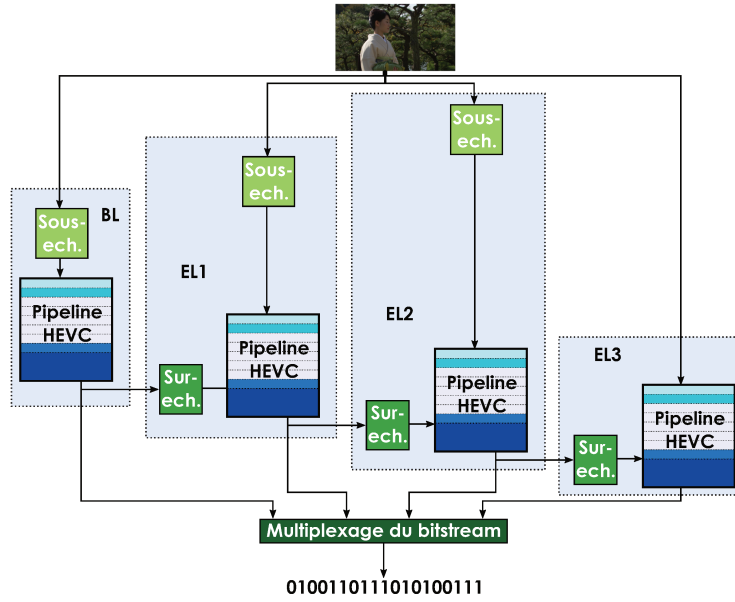


FIGURE 5.11 – Schéma du SHVC-AE en configuration LIVE avec une couche de base (BL) et 3 couches de rehaussement (EL1, EL2, EL3). Le sous-échantillonnage est réalisé en cascade entre les différentes couches de référence.

TABLE 5.6 – Idem Table5.5 mais en LIVE-HD. BL–EL1, BL–EL2 (Figure 5.14) : la couche de base (BL) prédit les couches de rehaussement EL1 et EL2. BL–EL1, EL1–EL2 (Figure 5.15) : la BL prédit l’EL1 puis l’EL1 prédit l’EL2.

Séquences	BL-EL1 BL-EL2			BL-EL1 EL1-EL2		
	AI	LD	RA	AI	LD	RA
B1	-43.2%	-35.1%	-26.4%	-46.2%	-37.3%	-28.2%
B2	-36.5%	-29.7%	-16.7%	-42.1%	-32.9%	-20.4%
B3	-35.7%	-29.8%	-16.9%	-41.7%	-34.1%	-20.7%
B4	-33.0%	-27.5%	-14.9%	-37.9%	-30.5%	-17.5%
B5	-28.6%	-20.3%	-4.2%	-35.7%	-23.8%	-6.9%
Moyenne	-35.4%	-28.5%	-15.8%	-40.7%	-31.7%	-18.8%

du débit en mode de qualité LIVE-HD respectivement pour les dépendances BL–EL1, BL–EL2 et BL–EL1, EL1–EL2. Les Tables 5.7 et 5.8 présentent les vitesses de codage multi-couches du SHVC-AE avec les dépendances BL–EL1, BL–EL2 et BL–EL1, EL1–EL2. On constate que le mode de qualité LIVE-HD, permet de répondre à la contrainte temps-réel sur les codages multi-couches. Le mode de qualité FILE, quant à lui, ne permet pas de répondre à la contrainte temps-réel mais propose des vitesses de codage intéressantes pour un codage cherchant la meilleure efficacité de codage. Les vitesses de codage avec les différentes dépendances sont similaires. On note toutefois que les codages avec la dépendance BL–EL1, EL1–EL2 sont légèrement plus rapides que ceux de la dépendance BL–EL1, BL–EL2.

De plus, les Tables 5.7 et 5.8 présentent également les rapports de vitesse de codage SHVC par rapport aux codages simulcast équivalents. De la même manière que pour les codages à 2 couches présentés dans le chapitre 3, on constate que dans le mode de qualité FILE le coût de la prédiction inter-couches est plus important en mode configuration AI et LD-P par rapport au RA. Les codages AI et LD-P réalisent une prédiction inter-couches ainsi qu’un sur-échantillonnage sur les textures pour l’ensemble des images traitées. Le codage RA, quant à lui, réalise une prédiction inter-couches seulement pour une image sur deux : sur une couche de référence, les images B ne servant pas de référence temporelle ne servent pas non plus de références inter-couches. Puisque le SHVC-AE en RA emploie moins souvent une prédiction inter-couches, celles-ci influencent moins les performances de vitesse

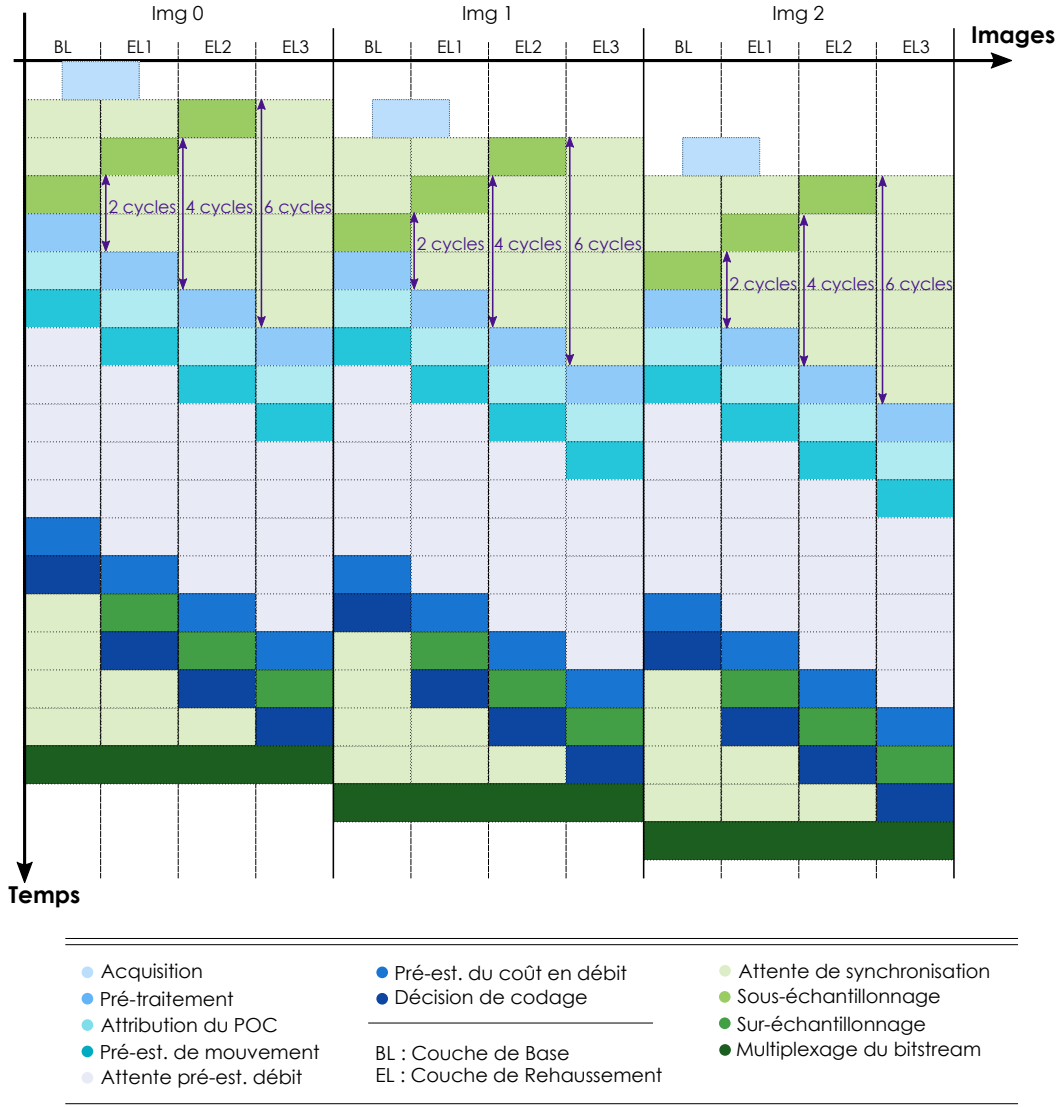


FIGURE 5.12 – Pipeline du SHVC-AE en mode LIVE avec 4 couches et les étapes de sous-échantillonnage réalisées en cascade. Schéma de l’architecture correspondante en Figure 5.11.

par rapport aux configurations AI et LD-P. Enfin, dans le mode de qualité LIVE-HD, la tendance est la même pour les configurations AI, LD-P et RA. On note toutefois que la prédiction inter-couches en configuration AI dégrade les performances de vitesse par rapport au codage FILE. En effet, la prédiction inter-couches utilisent la même mécanique que la prédiction temporelle (ou inter). Dans le mode LIVE-HD, un compromis sur les outils de codage est réalisé pour cette prédiction : elle est plus rapide mais moins efficace en terme de compression. C’est pourquoi, la prédiction inter-couches est effectuée plus rapidement en mode LIVE-HD qu’en mode FILE.

Le codeur SHVC-AE permet donc de réaliser des codages multi-couches en scalabilité sur les résolutions. Toutefois, l’architecture employée est suffisamment modulable pour supporter d’autres types de scalabilité.

5.2 Scalabilité appliquée à d’autres niveaux de qualité

Les types de scalabilité supportés par le standard SHVC ne sont pas restreints à la scalabilité spatiale. Par exemple, il peut également servir à réaliser des codages scalables sur différent(s) :

- niveaux de qualité (SNR),

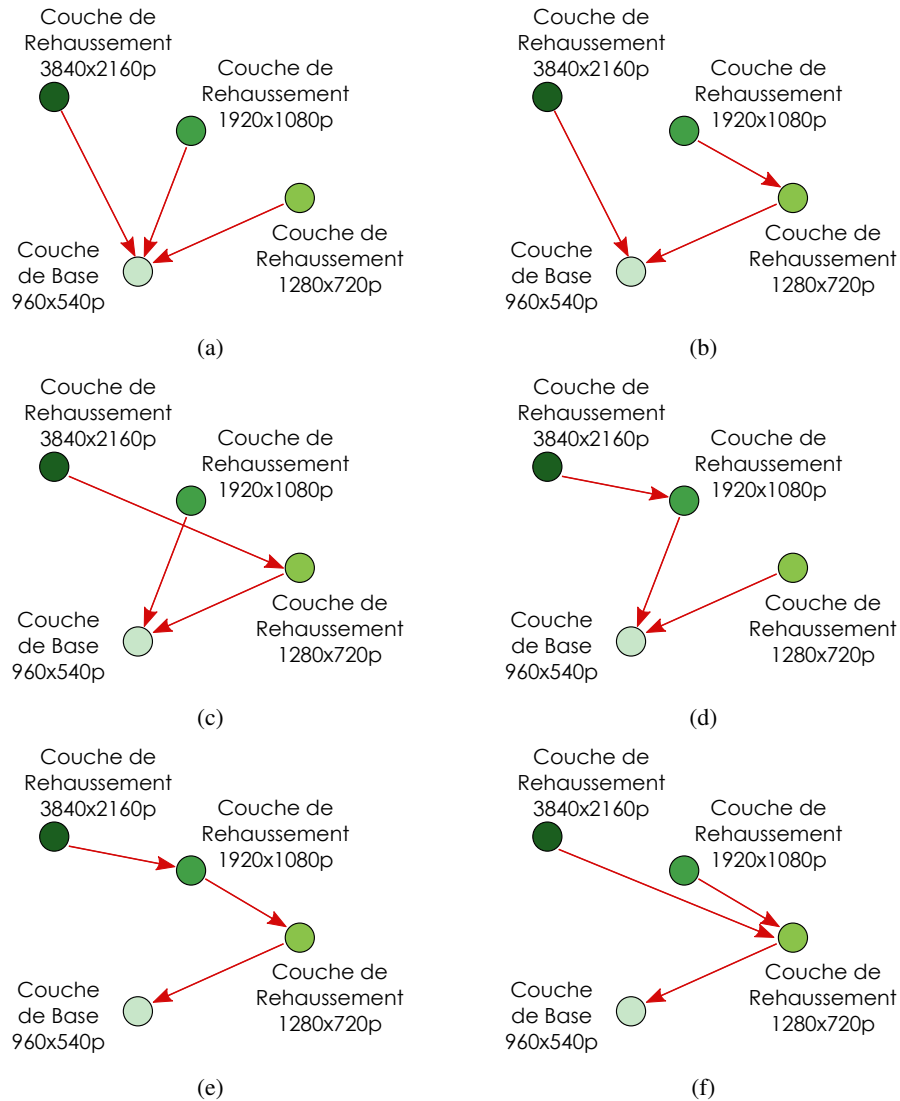


FIGURE 5.13 – Schémas de prédiction inter-couches avec 4 couches en scalabilité spatiale.

- profondeurs des données,
- standards de codage pour la couche de base,
- espaces colorimétrique.

5.2.1 Application à la scalabilité du SNR

Dans le cas de la scalabilité SNR, c'est la qualité mesurée à partir d'une métrique objective qui constitue la différence entre les couches. Cette qualité peut, par exemple, se traduire par une consigne en niveau de qualité ou bien en débit de codage respectivement pour des codages à qualité constante ou à débit constant.

Ce type de scalabilité ne requiert qu'une seule et unique source vidéo. Par conséquent, il n'est pas nécessaire d'ajouter une étape de pré-traitement sur les données sources. L'étape de sous-échantillonnage est alors retiré du pipeline. Le retrait de cette étape ne demande pas de nouvelle synchronisation puisque il est déjà géré quand le codeur reçoit plusieurs sources. De plus, le format d'images entre les différentes couches étant le même, il n'est pas non plus nécessaire d'ajouter une étape d'adaptation des informations de codage (textures et vecteurs de mouvement). Pour cela, deux choix d'architecture sont possibles :

1. la décision de codage de la couche de base est directement suivie de la décision de la couche de rehausse-

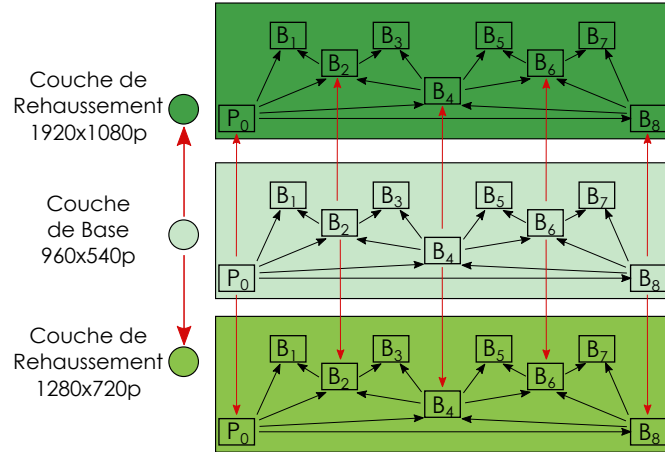


FIGURE 5.14 – Schéma de la dépendance inter-couches BL–EL1, BL–EL2 pour un codage sur 3 couches en configuration RA . La couche de base (BL 960×540p) prédit à la fois la couche de rehaussement intermédiaire (EL1 1280×720p) et la couche de rehaussement finale (EL2 1920×1080p).

TABLE 5.7 – Vitesse de codage (VdC) en IPS et rapport des vitesses (RdV) en mode de qualité FILE dans les configurations AI, LD-P et RA.

Séquences	BL–EL1, BL–EL2						BL–EL1, EL1–EL2					
	VdC du SHVC-AE			RdV SHVC-AE / SLE			VdC du SHVC-AE			RdV SHVC-AE / SLE		
	AI	LD	RA	AI	LD	RA	AI	LD	RA	AI	LD	RA
B1	1.39	0.91	0.98	0.51	0.72	0.96	1.47	0.96	1.01	0.54	0.76	0.99
B2	1.40	0.88	0.92	0.53	0.68	0.90	1.42	0.91	0.95	0.53	0.71	0.93
B3	1.39	0.90	0.94	0.52	0.71	0.93	1.42	0.95	0.99	0.53	0.75	0.97
B4	1.46	0.89	0.91	0.55	0.71	0.92	1.46	0.94	0.96	0.55	0.75	0.96
B5	1.43	0.88	0.90	0.53	0.70	0.90	1.43	0.94	0.95	0.53	0.75	0.95
Moyenne	1.41	0.89	0.93	0.53	0.70	0.92	1.44	0.94	0.97	0.54	0.74	0.96

ment,

- la décision de la couche de base est suivie d'une étape d'attente remplaçant l'étape de sur-échantillonnage avant de réaliser la décision de la couche de rehaussement.

Le premier cas permet de réaliser un codage avec une latence minimale d'un cycle mais requiert un retrait de l'étape correspondant au sur-échantillonnage. Le second cas ne modifie pas l'architecture actuelle du SHVC-AE mais remplace l'étape de sur-échantillonnage par une étape de recopie des informations de codage depuis la couche de base vers la couche de rehaussement. Bien que la latence de codage ne soit pas optimale dans le second cas (2 cycles), c'est cette dernière solution qui a été retenue pour effectuer des codages en scalabilité SNR dans le SHVC-AE. Elle permet de conserver une étape de pipeline correspondant au sur-échantillonnage afin de pouvoir combiner différents types de scalabilité sans modification du pipeline du codeur.

La Figure 5.16 schématise le pipeline de la solution retenue dans le mode de qualité LIVE. Le décalage entre les deux instances de SLE est alors de 2 cycles de pipeline. Ce choix d'architecture autorise d'éventuelles combinaisons avec d'autres types de scalabilité. Par exemple, dans le cas d'une scalabilité spatiale et SNR combinée, les étapes de sous-échantillonnage et de sur-échantillonnage auraient déjà une étape de pipeline réservée. Ainsi, il ne serait pas nécessaire de modifier l'architecture du SHVC-AE.

En terme de signalisation, le bitstream est construit de la même manière que pour le codage en scalabilité spatiale :

- VPS commun à toutes les couches,
- SPS par couches,
- PPS par couches,

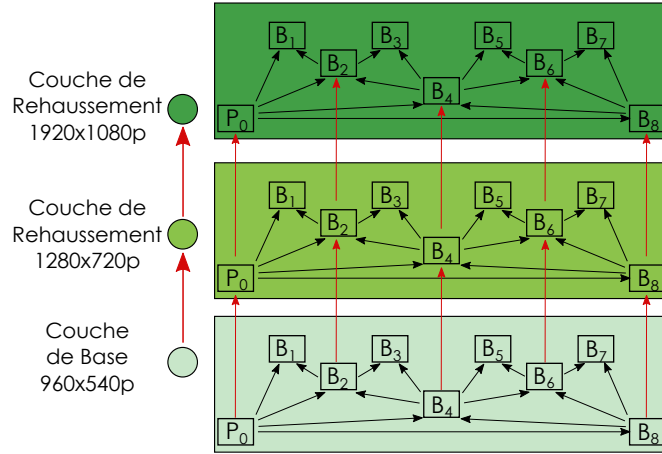


FIGURE 5.15 – Schéma de la dépendance inter-couches BL–EL1, EL1–EL2 pour un codage sur 3 couches en configuration RA. La couche de base (BL 960×540p) prédit la couche de rehaussement intermédiaire (EL1 1280×720p) puis l’EL1 prédit la couche de rehaussement finale (EL2 1920×1080p).

TABLE 5.8 – Vitesse de codage (VdC) en IPS et rapport des vitesses (RdV) en mode de qualité LIVE-HD dans les configurations AI, LD-P et RA.

Séquences	BL–EL1, BL–EL2						BL–EL1, EL1–EL2					
	VdC du SHVC-AE			RdV SHVC-AE / SLE			VdC du SHVC-AE			RdV SHVC-AE / SLE		
	AI	LD	RA	AI	LD	RA	AI	LD	RA	AI	LD	RA
B1	62.6	60.9	65.1	0.79	0.80	0.93	63.6	61.0	64.9	0.81	0.80	0.93
B2	56.8	56.6	64.6	0.81	0.79	0.93	57.9	59.9	64.2	0.82	0.80	0.92
B3	57.8	59.2	66.2	0.80	0.78	0.95	58.9	59.6	66.7	0.82	0.78	0.96
B4	60.2	60.0	67.4	0.79	0.77	0.92	61.2	60.7	67.8	0.80	0.78	0.93
B5	56.0	54.9	67.2	0.80	0.78	0.90	57.2	55.4	67.9	0.82	0.78	0.91
Moyenne	58.7	58.3	66.1	0.80	0.78	0.93	59.8	59.3	66.3	0.81	0.79	0.93

— entête de slice par couche...

Il n’y a pas d’éléments de syntaxe à ajouter ou à supprimer par rapport au codage en scalabilité spatiale. De plus, la scalabilité SNR emploie un système de dépendance entre couche. Le paramétrage des champs propres à l’extension SHVC ne sont pas non plus modifiés par rapport au codage en scalabilité spatiale.

Les expérimentations ont été menées en suivant les recommandations des CTC[33] avec les séquences présentées en Table 3.4. Les codages sont réalisés à QP constants. Les QP employés pour les couches de base et de rehaussement sont présentés en Table 5.9. Les CTC[33] ne testent pas les séquences A1 et A2 pour la configuration AI en scalabilité SNR.

Les Tables 5.10 et 5.11 présentent les comparaisons BD-BR des encodages SHVC-AE par rapport aux encodages SLE équivalents respectivement avec les modes de qualité FILE et LIVE-HD. La qualité de référence est celle de la couche de rehaussement. On constate que le SHVC-AE permet d’améliorer le débit de codage de la couche de rehaussement d’en moyenne 30.8% et 26.9% en configuration RA respectivement pour les modes de qualité FILE et LIVE-HD. Cette amélioration permet alors d’économiser, en moyenne, respectivement 15.9% et 13.6% du débit total par rapport à un codage simulcast. Les configurations AI et LD-P requièrent des débits de codage plus importants que la configuration RA. On constate que la prédiction inter-couches profite davantage à

TABLE 5.9

QP de la couche de base	26	30	34	38	26	30	34	38
QP de la couche de rehaussement	20	24	28	32	22	26	30	34

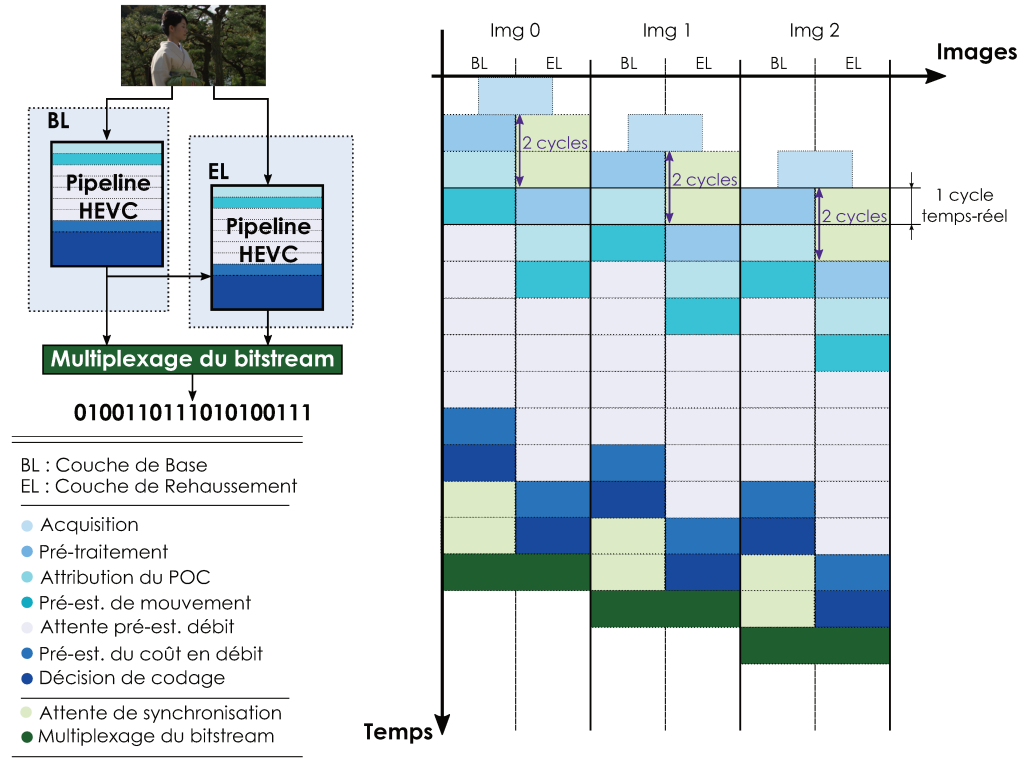


FIGURE 5.16 – Schéma du pipeline du SHVC-AE en scalabilité SNR sur 2 couches. Le sur-échantillonnage sur la couche de base (BL) est remplacé par une étape d’attente servant à la recopie des informations de codage de la prédiction inter-couches.

ces configurations en partie pour cette raison. L’autre raison concerne le compromis réalisé sur la prédiction inter-couches. La configuration RA emploie une prédiction inter-couches sur la moitié des images traitées. Les images concernées sont uniquement les images (I, P et B) qui servent de référence temporelle dans le codage HEVC. Les images B qui ne servent pas de référence temporelle profitent indirectement de la prédiction inter-couches puisque leurs images de référence sont enrichies par celle-ci. De plus, ce compromis sur la prédiction inter-couches permet de réaliser des codages SHVC avec des vitesses de codage proches de celles des codages HEVC équivalents.

La Table 5.12 présente les vitesses de codage du SHVC-AE dans les modes de qualité FILE et LIVE-HD pour des configurations LD-P et RA avec une scalabilité SNR. Les encodages ont été réalisés sur la plateforme Intel Xeon E5-4627V2 présentée en Table 3.6. Les codages FILE n’utilisent pas de répartition en Tiles alors que les codages LIVE-HD emploient une répartition 4×4 . Les optimisations bas-niveau issues du SLE sont employées. On constate que le mode de qualité LIVE-HD permet de répondre à une contrainte de vitesse temps-réel. Bien que le codage FILE cherche la meilleure efficacité de codage sans prendre en compte la vitesse de codage, en scalabilité spatiale, il permet malgré tout de réaliser des codages efficaces à une vitesse raisonnable d’en moyenne (voir Table 4.9) :

- 1.33 IPS en AI,
- 0.96 IPS en LD-P,
- 0.98 IPS en RA.

Pour cela, un compromis entre distorsion et vitesse de codage sur certains outils définis dans HEVC est utilisé, voir Table 4.10. Enfin, on constate que le SHVC-AE permet de réaliser des codages SHVC en scalabilité SNR avec une vitesse de codage proche de celle des codages HEVC équivalents. En effet, en configuration RA, les modes de qualité FILE et LIVE-HD sont respectivement 0.82 et 0.92 fois plus lents que le SLE dans ces mêmes modes de qualité. Ainsi, la vitesse de codage est faiblement affectée par la prédiction inter-couches du SHVC-AE.

TABLE 5.10 – Comparaison BD-BR du SHVC-AE par rapport au SLE en qualité FILE sur les débits de la couche de rehaussement (EL) et sur la somme des débits SHVC(BL+EL) et simulcast(BL+EL) équivalents en configuration LD-P et RA en scalabilité SNR.

Séquences	BD-BR SHVC-AE vs SLE (sur les EL)			BD-BR SHVC-AE vs SLE (sur les EL+BL)		
	AI	LD	RA	AI	LD	RA
A1	-	-44.2%	-34.4%	-	-25.3%	-17.9%
A2	-	-49.7%	-37.2%	-	-27.7%	-19.7%
B1	-54.2%	-43.1%	-31.2%	-30.3%	-24.2%	-14.8%
B2	-52.3%	-42.4%	-30.4%	-28.8%	-24.4%	-14.7%
B3	-50.8%	-39.9%	-28.5%	-28.8%	-24.1%	-14.2%
B4	-45.3%	-39.4%	-27.5%	-26.6%	-23.5%	-13.2%
B5	-49.0%	-38.1%	-26.1%	-28.3%	-24.7%	-16.5%
Moyenne	-50.3%	-42.4%	-30.8%	-28.6%	-24.8%	-15.9%

TABLE 5.11 – Comparaison BD-BR du SHVC-AE par rapport au SLE en qualité LIVE-HD sur les débits de la couche de rehaussement (EL) et sur la somme des débits SHVC(BL+EL) et simulcast(BL+EL) équivalents en configuration LD-P et RA en scalabilité SNR.

Séquences	BD-BR SHVC-AE vs SLE (sur les EL)			BD-BR SHVC-AE vs SLE (sur les EL+BL)		
	AI	LD	RA	AI	LD	RA
A1	-	-42.4%	-30.9%	-	-25.0%	-16.1%
A2	-	-49.7%	-36.6%	-	-29.2%	-20.6%
B1	-54.0%	-41.1%	-28.5%	-30.5%	-23.3%	-13.3%
B2	-50.3%	-38.7%	-21.2%	-29.5%	-23.5%	-8.2%
B3	-52.4%	-38.7%	-25.3%	-30.9%	-23.6%	-12.4%
B4	-50.1%	-40.0%	-26.7%	-29.4%	-24.1%	-12.7%
B5	-52.0%	-36.0%	-19.3%	-31.1%	-23.9%	-11.8%
Moyenne	-51.8%	-40.9%	-26.9%	-30.3%	-24.7%	-13.6%

TABLE 5.12 – Vitesse de codage (VdC) en IPS et rapport des vitesses (RdV) en modes de qualité FILE et LIVE-HD dans les configurations LD-P et RA en scalabilité SNR.

Séquences	FILE				LIVE-HD			
	VdC du SHVC-AE		RdV SHVC-AE / SLE		VdC du SHVC-AE		RdV SHVC-AE / SLE	
	LD	RA	LD	RA	LD	RA	LD	RA
A1	0.50	0.58	0.66	0.86	52.2	50.8	0.79	0.86
A2	0.51	0.55	0.70	0.89	46.5	40.4	0.84	0.92
B1	0.93	1.02	0.71	0.90	79.9	73.7	0.84	0.94
B2	0.87	0.96	0.67	0.87	75.2	76.2	0.84	0.93
B3	0.89	0.97	0.68	0.88	80.9	71.9	0.82	0.95
B4	0.91	0.96	0.71	0.90	82.9	77.0	0.81	0.93
B5	0.87	0.92	0.68	0.88	76.5	76.5	0.81	0.91
Moyenne	0.78	0.85	0.82	0.88	70.6	67.1	0.82	0.92

5.2.2 Application au codage multi-vues

Le codage multi-vues consiste à employer plusieurs sources capturées sous différents angles de vue. Les deux paramètres à prendre en compte dans le codage multi-vues sont notamment le nombre de vue et l'écart entre ces vues. Le nombre de vue, et par conséquent le nombre de sources, détermine le nombre de codeurs nécessaires. L'écart entre les vues, aussi appelé déphasage, est tiré de l'écart entre les caméras. Ce déphasage permet de définir des cartes de profondeur. Ces cartes permettent alors d'extraire la position des différents objets d'une image sur différents plans. Il est alors possible de déterminer l'éloignement d'un objet par rapport aux caméras ainsi qu'aux autres objets afin de réaliser un codage 3D. Dans cette partie, on ne s'intéresse pas au codage 3D, seulement au codage multi-vues.

Le codage multi-vues HEVC (MV-HEVC) est défini dans la première extension du standard [60]. Celui-ci définit un codage similaire au codage SHVC. Cependant, contrairement au codage SHVC, le codage MV-HEVC ne requiert ni pré-traitement tel que le sous-échantillonnage, ni conversion des données de codage tel que le sur-échantillonnage. Il s'approche donc plus d'une scalabilité SNR avec toutefois la nécessité d'acquérir plusieurs sources. De plus, le MV-HEVC ne définit pas de limitation sur les vecteurs de mouvement employés dans la prédiction inter-couches. Ceux-ci ne sont pas figés à la valeur (0,0) comme c'est le cas en SHVC. Enfin, il est nécessaire d'adapter certains paramètres de l'extension du VPS tel le *vps_scalability_mask_flag* qui donne le type de scalabilité employé. Ce paramètre permet alors de fournir non pas les dépendances entre couches comme pour le codage SHVC mais l'ordre de codage des vues. Ensuite, en remplacement des dépendances inter-couches, le paramètre *scalabilityId_mapping* vient compléter cet ordonnancement. Le [73] précise les subtilités de signalisation entre les codages SHVC et MV-HEVC. Parmi les études les plus récentes, [74] propose une régulation du débit basée sur la quantité de choix du mode AMVP en prédiction inter-couches et [75] propose un ajustement de la régulation du débit basé sur une étude statistique du codage MV-HEVC.

Dans le SHVC-AE, l'acquisition des différentes vues serait réalisée indépendamment pour chaque couche. Les différentes sources étant des images de même résolution, il n'est pas non plus nécessaire de réaliser un traitement sur les données de codage que sont les textures de l'image reconstruite et les vecteurs de mouvement issus de la vue de référence. Le SHVC-AE est donc en mesure de réaliser un codage multi-vues puisqu'il est capable d'acquérir différentes sources pour chacune des couches et qu'il effectue une prédiction inter-couches sur les mêmes informations de codage. De la même manière que pour la scalabilité SNR, il est nécessaire de ne pas réaliser de traitement sur les données de codage. Les étapes de sous-échantillonnage et de sur-échantillonnage sont remplacées par des étapes d'attente. Ce choix d'architecture, schématisée en Figure 5.17, ne présente pas la meilleure latence de codage mais permet de ne pas modifier la synchronisation entre les couches. Elle permettrait non seulement de ne pas altérer le code du SHVC-AE, mais également d'autoriser la combinaison avec d'autres types de scalabilité.

Du point de vue de l'efficacité de codage, la prédiction inter-couches permet une réduction de la consommation du débit à niveau de qualité constant. De la même manière que pour la prédiction inter-couches, la prédiction inter-vues utilise une référence supplémentaire identifiée par un nouvel indice de référence. Le mécanisme de prédiction inter-vues est alors le même que pour une prédiction temporelle dans un codage HEVC. Elle permet alors d'améliorer significativement l'efficacité de codage sur les vues prédites. En faisant appel au codage multi-couches présentés dans la section 5.1, deux cas d'usage multi-vues peuvent être évalués :

- 2 vues : la vue de gauche prédit la vue de droite,
- 3 vues : la vue centrale prédit les vues de gauche et de droite.

Pour un codage sur 2 vues, les vues de gauche et de droite correspondraient respectivement aux couches de base et de rehaussement du SHVC-AE. Pour un codage sur 3 vues, les vues du centre, de gauche et de droite correspondraient respectivement aux couches de base, de rehaussement intermédiaire et de rehaussement finale du SHVC-AE avec la dépendance BL-EL1, BL-EL2.

L'architecture du SHVC-AE lui permet de réaliser des codages multi-vues. Quelques modifications concernant la syntaxe permettraient de réaliser un bitstream compatible au codage MV-HEVC. Enfin, le calcul des vecteurs de mouvement à la place des vecteurs nuls améliorerait l'efficacité de ce codage.

5.2.3 Application à la scalabilité de la profondeur des données

Le codage en scalabilité sur la profondeur des données consiste à attribuer à chaque couche différentes tailles de pixel. Cette taille s'exprime en nombre de bits par pixel (i.e. bpp). Une taille plus importante des pixels permet un meilleur rendu des contrastes sur les images. La taille des pixels par couche est signalée dans le VPS séparément pour les composantes de luminance et de chrominance respectivement avec les champs *bit_depth_vps_luma_minus8*

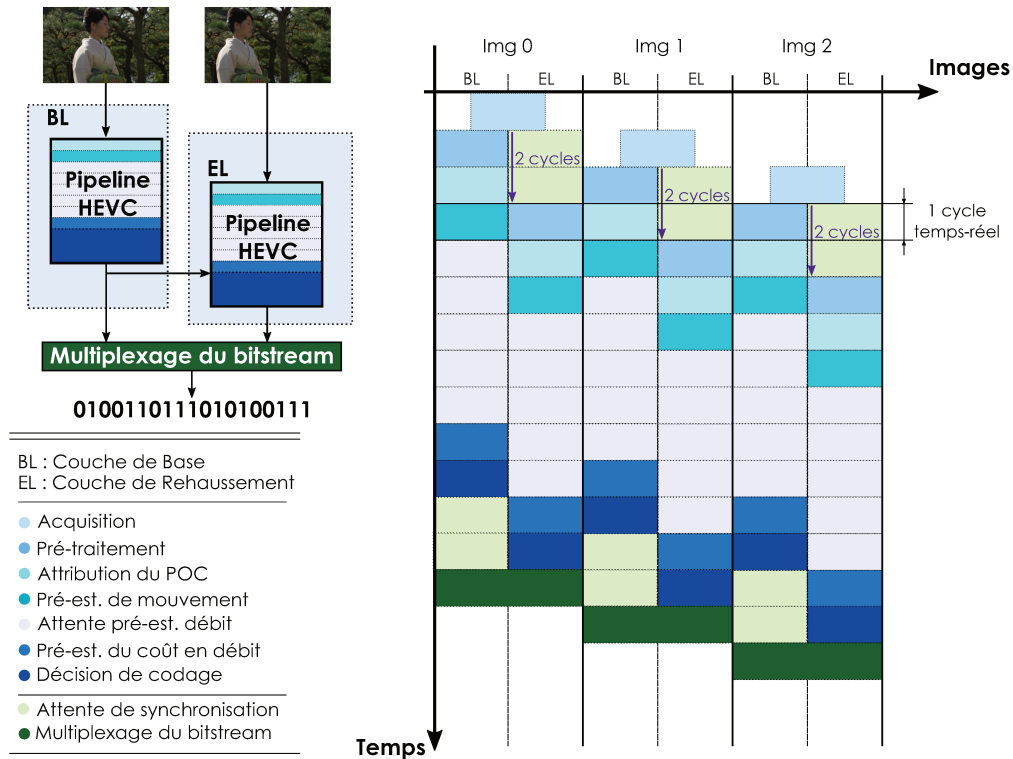


FIGURE 5.17 – Pipeline du SHVC-AE pour un codage multi-vues. Le sur-échantillonnage sur la couche de base (BL) est remplacé par une étape d’attente.

et $bit_depth_yps_chroma_minus8$. Le reste de la signalisation correspond à une scalabilité spatiale avec une conversion sur les valeurs de chaque pixel d’une profondeur à une autre.

De point de vue des informations de codage, les besoins pour une prédiction inter-couches avec une telle scalabilité sont comparables à ceux de la scalabilité spatiale. En effet, les données sources sont différentes sur chaque couche et il est nécessaire d’adapter les informations de codage aux couches de rehaussement. En ce qui concerne la réception des données sources, deux cas sont possibles : la réception d’autant de sources qu’il y a de couches ou bien la réception d’une seule source. Tout d’abord, dans le cas où plusieurs sources sont reçues, seule une étape de conversion des informations de codage doit être ajoutée. Dans ce cas, le décalage entre les couches s’élève à 1 ou 2 cycles selon que la conversion des données soit effectuée respectivement sur la couche prédite ou sur la couche de référence. Ce décalage correspond à la synchronisation entre la fin de la décision de codage sur la couche de référence et le début de la conversion des informations de codage. La Table 5.13 résume les différents décalages minimum possibles entre les couches pour les différents cas identifiés.

Ensuite, dans le cas où une seule source est réceptionnée, une première étape de conversion des données est nécessaire en début du pipeline. De la même manière que pour le sous-échantillonnage en scalabilité spatiale, cette étape de conversion sur la profondeur des pixels peut constituer une étape de pipeline supplémentaire. Cette opération de conversion peut être soit réalisée après l’acquisition des données, soit réalisée au moment de l’acquisition de la source. Dans le premier cas, une seule lecture de la source est requise et une étape de conversion sur les tailles de pixels est ajoutée à la position du sous-échantillonnage. Dans le second cas, les données de l’image source sont directement converties avec la bonne profondeur au moment de l’acquisition. Ce dernier cas requiert d’instancier autant d’étapes d’acquisitions qu’il y a de couches afin de réaliser plusieurs lectures sur le même fichier mais avec différentes cibles de profondeur des pixels. La conversion est alors réalisée au niveau de l’étape d’acquisition.

Enfin, la conversion des données sur les informations de codage ne prend effet que sur les textures. L’opération consiste à convertir les valeurs de chaque pixel d’une profondeur à une autre. Pour cela, un parcours sur l’ensemble des pixels de chaque composante est réalisé avant de multiplier la valeur du pixel par une constante. Cette constante est calculée selon la formule suivante avec $BitdepthDst$ et $BitdepthSrc$ respectivement la taille des pixels sur la

TABLE 5.13 – Décalage entre la couche de référence et

- la première couche de rehaussement intermédiaire (P-EL-I),
- les autres couches de rehaussement intermédiaires (A-EL-I),
- la couche de rehaussement finale (EL-F)

en scalabilité sur la profondeur des données pour une conversion des informations de codage

- sur les couches prédites (Conv-ILP-p),
- sur les couches de référence (Conv-ILP-ref)

dans les 3 cas :

1. plusieurs sources pré-converties (N-Src),
2. une seule source et une étape de conversion ajoutée au pipeline (1Src-Pp.Conv),
3. une seule source lue avec les conversions intégrées aux acquisitions (1Src-Acq.Conv).

Couches identifiées	Conv-ILP-p			Conv-ILP-ref		
	N-Src	Une source		N-Src	Une source	
		1Src-Pp.Conv	1Src-Acq.Conv		1Src-Pp.Conv	1Src-Acq.Conv
P-EL-I	1	1	1	2	2	2
A-EL-I	2	2	2	2	2	2
EL-F	2	3	2	2	3	2

couche prédite et sur la couche de référence :

$$\text{Constante de conversion} = 2^{\text{BitdepthDst} - \text{BitdepthSrc}}$$

La prédiction inter-couches fonctionnent ensuite de la même manière qu'en scalabilité spatiale : l'image issue de la couche de référence est employée dans la prédiction de l'image correspondante sur la couche prédite. En définitive, l'architecture du SHVC-AE permettrait de réaliser un codage scalable sur la profondeur des données. Il suffirait de remplacer les étapes de sous-échantillonnage et de sur-échantillonnage par les étapes de conversion des tailles de pixels correspondantes.

Enfin, il est à noter que les opérations de sous-échantillonnage et de sur-échantillonnage permettent de réaliser la multiplication par la constante de conversion ce qui permet de combiner les scalabilités sur les résolutions et sur les profondeurs. Les étapes de conversion des tailles des pixels sont alors confondues avec les opérations de sous-échantillonnage et de sur-échantillonnage. Le codage en scalabilité sur la profondeur des données peut alors être combiné avec le codage en scalabilité spatiale et en scalabilité SHVC-AE sans modifier l'architecture du SHVC-AE. Dans ce cas, seul la multiplication par la constante de conversion doit être ajoutée au sur-échantillonnage.

5.2.4 Application à la scalabilité des codecs

L'extension SHVC autorise l'utilisation d'un standard de compression différent sur la couche de base. La scalabilité concernée, alors appelée scalabilité des codecs, requiert toutefois que le codec employé sur la couche de base soit issu de la standardisation ITU comme H.262/MPEG-2 ou H.264/MPEG-4 AVC par exemple.

La signalisation d'un bitstream employant une telle scalabilité ne diffère pas dans sa syntaxe haut-niveau. En effet, la succession des "*parameters sets*" ne change pas : VPS, SPS, PPS... Ce sont les NALU issus de la couche de base qui change. Puisqu'elles emploient un standard de compression différent du HEVC, la signalisation au sein de cette NALU doit correspondre à ce standard. Les couches de rehaussement ne sont pas modifiées.

Dans le cas du SHVC-AE, l'instance de SLE correspondant à la couche de base emploierait alors un standard de compression différent du HEVC. La signalisation des *parameters sets* est basée sur le codec employé dans cette couche de base. Afin de permettre un codage en scalabilité des codecs, il faudrait d'abord signaler les *parameters sets* issus du HEVC avant de signaler les différentes NALU contenant les informations de codage. Cette modification ne prend pas effet sur l'architecture du pipeline mais requiert des modifications de la syntaxe sur la couche de base. En effet, dans le SHVC-AE, c'est la couche de base qui génère le VPS, les instances de SLE des autres couches n'en génèrent pas. Le VPS a été défini pour la première fois dans le standard HEVC[60], les autres standards de compression des groupes MPEG et ITU n'en génèrent pas. Pour une scalabilité sur les codecs, il serait

alors nécessaire de faire en sorte que la couche de base génère un VPS même si son standard ne le définit pas. Par conséquent, la syntaxe haut-niveau serait détacher du type de codec employé.

Du point de vue de l'architecture de codage, celle ci est la même que pour la scalabilité SNR. En effet, les données sources et les informations de codage conservent le même format. La synchronisation entre les couches serait alors pilotée par la fin de la décision de codage sur la couche de base et le début de l'étape de recopie des informations de codage. Mis à part la décision de codage, la structure du code est la même dans le SLE pour un codage HEVC ou pour un autre standard de compression. Par conséquent, les buffers de transfert des informations de codage ajoutés pour la prédiction inter-couches dans le SHVC-AE peuvent déjà être utilisés à partir de la couche de base, et ce, quelque soit le codec employé.

La prédiction inter-couches est employée de la même manière que pour les autres types de scalabilité. L'architecture du SHVC-AE est alors la même que pour un codage en scalabilité SNR. De plus, puisque les données sources et les informations de codage conservent le même format, la scalabilité des codecs peut être employée en combinaison avec les scalabilités spatiale, SNR et sur la profondeur des données.

5.2.5 Application à la scalabilité des espaces colorimétriques

La scalabilité sur les espaces colorimétriques (ou "Color Gamut Scalability" (CGS)) consiste à coder une même séquence avec différents espaces de couleur sur chaque couche. Par exemple, un codage en CGS pourrait utiliser les espaces de couleur BT.709, DCI-P3 et BT.2020 respectivement sur les couches de base, de rehaussement intermédiaire et de rehaussement finale. Une telle scalabilité requiert une conversion d'un espace de couleur à un autre. Cette conversion des données depuis de la couche de référence vers la couche prédite est définie dans l'extension SHVC. Afin de signaler cette conversion, le champ *colour_mapping_enabled_flag* doit être mis à 1 dans le PPS. Ensuite, les caractéristiques de la conversion sont ajoutées. Tous ces éléments de syntaxe sont présents dans les champs du PPS relatifs à l'extension SHVC. Le reste de la syntaxe haut-niveau (VPS, SPS et entête de slice) est similaire au codage en scalabilité spatiale.

De plus, pour une scalabilité CGS, l'architecture du SHVC-AE serait similaire à un usage en scalabilité spatiale. En effet, la scalabilité CGS requiert non seulement une étape de conversion des données entre les couches de référence et les couches prédites mais également une conversion sur l'image source à inclure soit :

- dans un pré-traitement intégré au pipeline de codage équivalent au sous-échantillonnage,
- dans un pré-traitement préalable à l'acquisition équivalent au pré-sous-échantillonnage.

Dans ce dernier cas, il est nécessaire de fournir plusieurs sources au codeur. La problématique concernant ce pré-traitement est d'ailleurs la même que pour le sous-échantillonnage en scalabilité spatiale : cette conversion n'est pas définie dans l'extension SHVC. Dans [76], le codage est réalisé sur deux couches et avec deux sources. Les espaces de couleur pour les sources des couches de base et de rehaussement sont respectivement BT.709 et BT.2020. La source BT.709 correspond au format de capture de la vidéo en sortie de caméra. La source BT.2020 est obtenue en convertissant la source BT.709 en amont du codage. L'architecture ainsi que la structure du pipeline de codage employée en scalabilité sur les espaces de couleur serait donc la même que pour la scalabilité spatiale à l'exception des opérations de conversion des données.

En terme de complexité, le codage en scalabilité sur les espaces de couleur est similaire au codage en scalabilité spatiale pour la prédiction inter-couches. En ce qui concerne la conversion de données de la couche de référence vers la couche prédite, celle ci s'appuie sur une "lookup table" suivant les 3 composantes (i.e. LUT-3D). La Figure 5.18 illustre une LUT-3D dans le domaine Rouge-Vert-Bleu (RVB). Cette LUT-3D associe une couleur à des coordonnées selon 3 composantes dans les différents espaces de couleur. L'espace de couleur initial est alors divisé spatialement en plusieurs cubes appelés octants.

La conversion d'un espace de couleur à un autre qui s'appuie sur l'utilisation d'une LUT-3D, aussi appelée interpolation tri-linéaire, consiste donc à :

- déterminer 3 les octants $(y_i, u_j, v_k)_{i=0,1, j=0,1, k=0,1}$ à partir des 3 composantes (y, u, v) d'une couleur dans l'espace de couleur initial,
- associer les 3 octants dans la LUT-3D : $(LUT_Y[y_i][u_j][v_k], LUT_U[y_i][u_j][v_k], LUT_V[y_i][u_j][v_k])_{i=0,1, j=0,1, k=0,1}$ afin d'obtenir les largeurs des cubes $(s_i(y), s_j(u), s_k(v))$,
- effectuer une interpolation dans l'espace de couleur de destination :

$$Y = \frac{\sum_{i=0}^1 \sum_{j=0}^1 \sum_{k=0}^1 s_i(y) \times s_j(u) \times s_k(v) \times LUT_Y[y_i][u_j][v_k]}{(y_1 - y_0) \times (u_1 - u_0) \times (v_1 - v_0)} \quad (5.4)$$

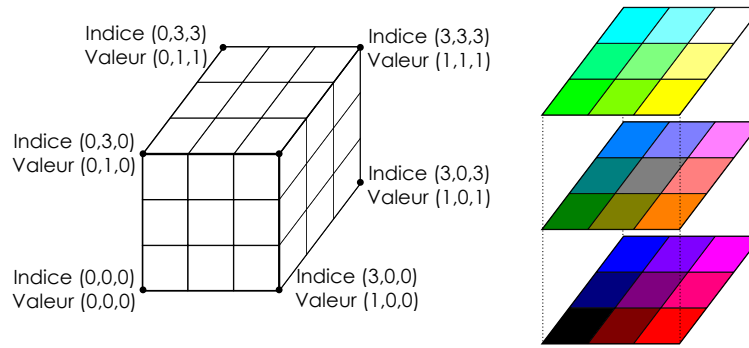


FIGURE 5.18 – Exemple d'une LUT-3D dans l'espace de couleur Rouge-Vert-Bleu (RVB) avec 3 octants par composante.

Dans le cas d'une combinaison des scalabilités sur les résolutions et sur les espaces de couleur, cette interpolation tri-linéaire doit être effectuée avant le sur-échantillonnage. En effet, cette interpolation est plus complexe que le sur-échantillonnage et sera donc effectuée plus rapidement sur des résolutions plus petites. Cette recommandation est normalisée dans l'extension SHVC[60].

Afin de réaliser un codage en scalabilité CGS, le SHVC-AE devrait alors :

- remplacer le sous-échantillonnage par une conversion non standardisée suivant une LUT-3D ou recevoir une source préalablement convertie pour chaque couche,
- remplacer le sur-échantillonnage par l'interpolation tri-linéaire standardisée avec la LUT-3D du standard SHVC.

Par conséquent, l'architecture du SHVC-AE permet de réaliser une scalabilité CGS. Pour une conformité du bitstream il faudrait ajouter l'interpolation tri-linéaire ainsi que les éléments de syntaxe propre à la scalabilité CGS.

Conclusion

En définitive, nous avons montré dans ce chapitre que le codeur SHVC-AE permet à la fois de réaliser des codages multi-couches et multi-scalabilités. L'architecture employée permet alors de réaliser des codages SHVC génériques. La flexibilité de l'architecture employée permet de présenter des résultats de codage intéressants pour un codage sur 3 couches et pour une scalabilité SNR. Les tests dans le mode de qualité FILE ont montré 18.5% et 30.5% d'amélioration du débit à qualité constante sur la plus haute couche de rehaussement en configuration RA respectivement pour les codages à 3 couches et en scalabilité SNR. Ils montrent également les performances du SHVC-AE en terme de vitesse de codage prouvant que la prédiction inter-couches peut améliorer l'efficacité de codage en affectant peu la latence de codage. Par exemple, en configuration RA en mode de qualité LIVE-HD, le SHVC-AE est seulement 0.93 et 0.92 fois moins rapide que le SLE respectivement pour les codages multi-couches BL-EL1, EL1-EL2 et le codage en scalabilité SNR.

De plus, l'architecture employée est suffisamment modulable pour permettre le codage avec d'autres types de scalabilité comme sur la profondeur des données, sur le codec de la couche de base ou sur les espaces de couleur. Enfin, les différents types de scalabilité étudiés peuvent être combinés pour différents types d'applications. Par exemple, la combinaison des scalabilités sur la profondeur des données et sur les espaces de couleur permettrait de proposer une couche de base "Standard Dynamic Range" (SDR) et une couche de rehaussement "High Dynamic Range" (HDR).

Enfin, le codeur SHVC-AE dispose de nombreuses améliorations qui le prépare à une utilisation dans un environnement de diffusion :

- architecture modulable,
- parallélisme sur plusieurs niveaux : inter-couches, inter-images, intra-images, inter-blocs, inter-opérations.
- efficacité de codage optimisés pour un codage scalable en résolution sur deux couches,
- modes de qualité permettant le codage en temps-réel de séquences HD et UHD.

Le chapitre suivant présente l'intégration du SHVC-AE dans un tel environnement. Pour cela, le SHVC-AE est associé à d'autres modules permettant la gestion des entrées/sorties du codage ainsi qu'une attribution des res-

sources matérielles permettant de garantir des performances temps-réel sur des séquences au format UHD.

Chapitre 6

Application dans un contexte de diffusion temps-réel

Introduction

Dans les chapitres précédents, nous avons montré les performances du codeur SHVC-AE en terme :

- d'efficacité de codage par rapport au codeur SHM, mais aussi par rapport au codage HEVC équivalent,
- de vitesse de codage sur des plateformes multi-cœurs permettant de montrer l'intérêt des différents niveaux de parallélisme du codeur (inter-couches, inter-images, intra-images, inter-blocs et sur les opérations bas-niveau),
- de latence vis à vis d'un codeur HEVC équivalent,
- de flexibilité de l'architecture qui permet de réaliser des codages sur plus de deux couches et avec différents types de scalabilité (spatiale, SNR, profondeur des données, codec, espaces de couleur...)

De plus, différents modes de qualité ont été définis. Ces modes de qualité permettent de réaliser :

- un codage SHVC cherchant à atteindre la meilleure efficacité de codage possible (FILE),
- un codage rapide permettant le codage temps-réel sur différentes résolutions (LIVE-SD, LIVE-HD et LIVE-UHD),
- un codage rapide et tenant compte de la complexité du codage sur la couche de base afin d'améliorer l'efficacité de la prédiction inter-couches (LIVE-HD+, LIVE-UHD+).

Les différents compromis réalisés sur les outils de codage dans les modes de qualité LIVE permettent d'assurer une vitesse de codage en condition temps-réel au détriment d'une diminution de l'efficacité de codage. De plus, des compromis entre complexité et efficacité de codage sont réalisés sur les outils de la prédiction inter-couches :

- les textures des images reconstruites sont utilisées dans la prédiction inter-couches,
- les vecteurs de mouvement issus des couches de référence ne sont pas utilisés dans la prédiction inter-couches,
- les références inter-couches sont seulement les images employées en tant que référence temporelle dans le codage HEVC.

En plus des compromis proposés, les différentes optimisations intégrées au SHVC-AE, tant au niveau des opérations de filtrage qu'au niveau des outils employés dans la décision, accélèrent significativement sa vitesse de codage. Le SHVC-AE est par conséquent une solution logicielle capable de réaliser des codages SHVC en condition temps-réel. Toutefois, l'exploitation du codeur dans une telle configuration requiert quelques ajustements aux niveaux des entrées/sorties ainsi qu'au niveau de la mise en concurrence des tâches de traitement entre la couche de base et la couche de rehaussement.

Dans ce chapitre, nous proposons une intégration du codeur SHVC-AE en scalabilité spatiale sur des plateformes multi-processeurs (ou multi-cœurs) dans un contexte de diffusion temps-réel. Au cours de cette thèse, cette intégration du SHVC-AE a permis d'accélérer significativement le temps de développement du code. En effet, la participation à diverses démonstrations a nécessité des efforts supplémentaires afin de livrer une solution fonctionnelle dans les temps. Sur ces périodes, certaines parties du code du SHVC-AE ont évolué rapidement, comme par exemple les optimisations des filtres, l'utilisation de la prédiction inter-couches sur les textures ou encore la

génération d'un bitstream SHVC conforme au standard.

Dans ce chapitre, l'objectif est de proposer une solution de codage pouvant s'intégrer dans une chaîne de diffusion. La chaîne de diffusion étudiée s'appuie sur le standard de diffusion ATSC 3.0. La Figure 6.1 illustre de façon simpliste les éléments qui constituent une telle chaîne de diffusion. D'abord, le codeur réceptionne des images issues d'une captation par caméra. Une fois le codage effectué, le bitstream est transmis à travers un canal de diffusion avant d'être réceptionné par de multiples appareils en mesure d'afficher la couche de base ou, pour les plus adaptés, la couche de rehaussement. Afin d'intégrer le SHVC-AE dans une telle chaîne, il est nécessaire de définir son comportement sur les entrées/sorties du codeur. Les différentes opérations associées à ces comportements sont réparties sur des modules : un module de management pilote les modules d'acquisition, de codage et de transmission. Enfin, chacun de ces modules se voit attribuer une quantité de ressource matérielle permettant de mener leurs opérations en temps-réel.

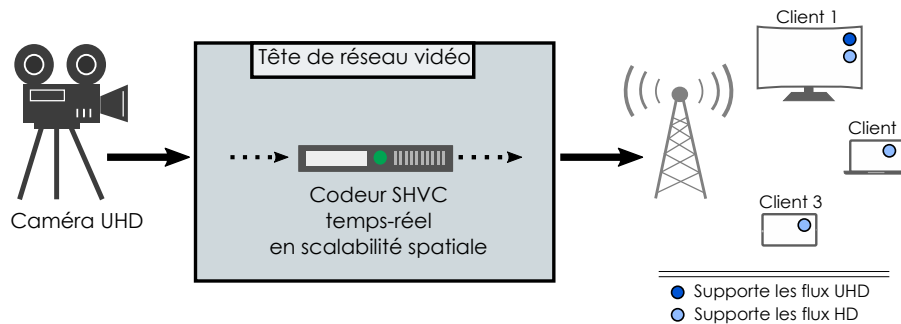


FIGURE 6.1 – Intégration du codeur SHVC dans une chaîne de diffusion vidéo Ultra Haute Définition (UHD) en temps-réel.

6.1 Caractéristiques des différents modules

Les différents modules employés pour l'intégration du SHVC-AE dans une chaîne de diffusion temps-réel sont :

- le module de management,
- le module d'acquisition,
- le module de codage,
- le module de transmission.

Le module de management permet de piloter les autres modules. Il sert à synchroniser ces derniers et à donner des consignes d'ordonnancement des opérations. Le module d'acquisition permet de recevoir les images en temps-réel. Il sert d'interface d'entrée avec l'extérieur de la plateforme de codage. Les images reçues sont ensuite fournies au module de codage. Le module de codage réalise un codage SHVC. Il fait appel au codeur SHVC-AE dans ses modes de configuration permettant la création d'un bitstream en temps-réel. Enfin, le module de transmission met en forme le bitstream avant de le transmettre sur un canal IP. Ce module sert d'interface de sortie à la plateforme de codage. Chacun des modules se voit attribuer un nombre de threads permettant de réaliser leurs opérations respectives en temps-réel. Ces threads seront, par la suite, attribués à un certain nombre de cœurs eux même répartis sur les différents CPU.

6.1.1 Le module de management

Le module de management est responsable de l'initialisation des différents modules de traitement tels que le codage, l'acquisition et la transmission sur canal IP. Les différents modules de traitement gérés par le module de management sont présentés sur la Figure 6.2. Une fois l'initialisation et le démarrage des modules de traitement effectués, le module de management assure la synchronisation inter-modules au niveau du pipeline en réceptionnant des signaux de fin de traitement. Une fois tous les signaux reçus, le cycle de pipeline suivant est démarré. Les opérations réalisées sur le module de management sont primordiales mais de faible complexité. Elles sont toutes réalisées dans un unique thread qui sera attribué.

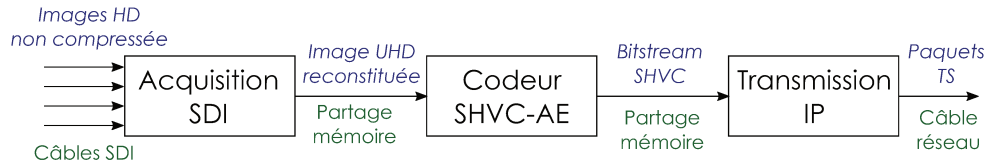


FIGURE 6.2 – Schéma des différents modules de traitement pilotés par le module de management dans l'intégration du codeur SHVC-AE dans un système de diffusion temps-réel.

6.1.2 Le module d'acquisition

Un des problèmes les plus importants dans le codage vidéo en temps-réel est issu de l'acquisition des données sources. En effet, la transmission d'un contenu vidéo non compressé nécessite de pouvoir émettre ou recevoir énormément de données. L'échantillonnage 4:2:2 est usuellement apprécié pour transmettre des vidéos non compressées. En effet, il constitue un compromis entre la fidélité vis à vis de l'image source et la quantité de données à transmettre. On peut estimer le débit minimum nécessaire à la transmission de vidéo de profondeur 10 bits par pixel, avec un échantillonnage 4:2:2 pour des résolutions 1080p25 et 2160p50 respectivement à 1.037 Gbps et 8.294 Gbps.

$$\text{Débit}_{1080p25} = 1920 \times 1080 \times 10 \times 25 \times 2 = 1036800000 \text{bps} = 1.037 \text{Gbps}$$

$$\text{Débit}_{2160p50} = 3840 \times 2160 \times 10 \times 50 \times 2 = 8294400000 \text{bps} = 8.294 \text{Gbps}$$

Ces contraintes requièrent des moyens de transmission rapides et fiables pour une transmission temps-réel qui n'impacterait pas le temps de codage vidéo.

Une première solution est d'employer le transcodage vidéo. Le transcodage consiste à transmettre un flux déjà compressé afin de le coder de nouveau. La quantité de données transmises est déjà réduite et requiert une bande passante moins importante. Le nouveau codage permet alors de modifier certains paramètres de codage comme par exemple le codec, le débit, le format de la vidéo... Il peut également permettre d'enrichir le codage en ajoutant des opérations comme par exemple un débruitage, une détection de scène... Pour cela le transcodage réalise successivement un décodage sur le flux réceptionné suivi d'un encodage sur le flux reconstruit du décodage comme illustré sur la Figure 6.3. Le principal problème de cette solution réside dans le fait que la vidéo soit déjà codée. En effet, en admettant que le décodage et le nouveau codage soient réalisés sans pertes, le premier codage détermine la qualité maximale de la vidéo reconstruite finale. Tous les efforts réalisés par l'opération de transcodage repose alors sur une ressource qu'elle ne peut pas maîtriser : la qualité de la source reçue. De plus, en considérant à présent que le décodage et le codage ne soient pas réalisés sans pertes, les distorsions occasionnées pourraient aggraver davantage des distorsions déjà présentes dans le premier codage. Enfin, la réception et surtout le décodage constituent une latence significative avant le démarrage du codage SHVC. Afin d'éviter ces problèmes de qualité vidéo et de latence, il est nécessaire d'employer directement les images non compressées.



FIGURE 6.3 – Schéma d'une chaîne de transcodage vidéo.

Dans ce contexte, des groupes comme la "Society of Motion Picture and Television Engineers" (SMPTE), l'Union européenne de radio-télévision (UER ou EBU), l'Union International des Télécommunications (ITU) ou encore le "Digital Cinema Initiatives" (DCI) proposent une normalisation des formats vidéos et des protocoles de transport afin de faciliter les échanges. C'est notamment le cas du standard Interface numérique série ("Serial Digital Interface") (SDI) proposé par la SMPTE [77]. Ce standard, défini en 1989 dans la norme SMPTE ST

TABLE 6.1 – Évolution du standard SDI sur les dernières années. Les formats d’images sont présentés à partir de leur largeur, entrelacement (progressif (p) ou entrelacée (i)), et fréquence d’image.

Nom	Standard SDI	Date	Formats d’images largeur / fréquence	Échantillonnage / profondeur	Débit maximum
SD-SDI	SMPTE-259M [78]	1989	480i / 59.94, 50	4:2:2/10 bits	270 Mbps
	SMPTE 344M [79]		576i / 59.94, 50		360 Mbps
HD-SDI	SMPTE ST 292 [80]	1998	480p, 576p / 30, 25	4:2:2/10 bits	540 Mbps
			720p / 60, 50		1.485 Gbps
			720p / 30, 25, 24 1080p / 30, 25, 24		
3G-SDI	SMPTE ST 424 [81]	2006	1080p / 60, 50	4:2:2/10 bits	1.970 Gbps
Quad 3G-SDI	SMPTE ST 425-5 [82]	2012	2160p / 60, 50	4:2:2/10 bits	12 Gbps
6G UHD-SDI	SMPTE ST 2081-10 [83]	2014	2160p / 30, 25, 24	4:2:2/10 bits	6Gbps
Dual 6G UHD-SDI	SMPTE ST 2081-11 [84]	2014	2160p / 60, 50	4:2:2/10 bits	12 Gbps
12G UHD-SDI	SMPTE ST 2082-10 [85]	2014	2160p / 60, 50	4:2:2/10 bits	12 Gbps
			2160p / 30, 25, 24	4:4:4/12 bits	

259 [78], propose d’abord un protocole de diffusion ou de transport de vidéo au format SD. Le support des autres formats suivirent au cours des années suivantes comme présenté dans la Table 6.1. Les données du signal vidéo ne sont pas compressées mais peuvent être échantillonnées et quantifiées. Les standards les plus récents permettent le transport de vidéos échantillonnées et quantifiées en 4:2:2 10 bits et 4:4:4 12 bits. Les données Interface numérique série ("Serial Digital Interface") (SDI) sont transmises sur un ou plusieurs câbles coaxiaux d’impédance 75Ω . Ces câbles disposent d’un connecteur "Bayonet Neill–Concelman" (BNC) ou d’une tête à fibre optique.

Entre le format en sortie de caméra et le format d’entrée du codeur, il peut être nécessaire de réaliser certaines opérations afin que le signal transmis soit conforme au standard SDI. Les opérations de quantification et de ré-échantillonnage en font partie. Ces opérations sont généralement pilotées par la carte d’acquisition. Afin de répondre à la contrainte temps-réel, ces opérations doivent être optimisées ou réalisées sur des architectures "hardware".



FIGURE 6.4 – Photo de la carte d’acquisition DTA-2174 employée pour les démonstrations.

Dans le cas de la démonstration temps-réel du codeur SHVC-AE, le plus grand format de vidéo de l’application proposée dispose des caractéristiques suivantes :

- une résolution de $3840 \times 2160p$,
- une fréquence de 30 IPS,
- une profondeur de 10 bits par pixel,
- un échantillonnage 4:2:2 planaire.

Pour réaliser la réception d’un tel format de vidéo, la carte d’acquisition Quad 3G-SDI du fabricant Dektec : la DTA-2174 [86] a été choisie. Cette carte, illustrée sur la Figure 6.4, permet de recevoir simultanément quatre vidéos de résolution $1920 \times 1080p$ à 60 IPS. Il est alors possible de recombinaison ces quatre images (ou *quadrants*)

pour recréer une vidéo de résolution $3840 \times 2160p$ à 60Hz ou bien $1920 \times 1080p$ à 240 IPS ou encore $1920 \times 1080p$ à 60 IPS en 3D. Dans le cadre de l'application proposée, l'acquisition des quatre quadrants $1920 \times 1080p$ à 30 IPS est suffisante. Avant l'assemblage des quadrants, il est nécessaire de réaliser un ré-échantillonnage sur les images. Le SHVC-AE requiert un format 4:2:2 planaire avec 10 bits par pixel. La transmission sur la DTA-2174, quant à elle, traite des images avec un format V210. Le format V210 correspond à un format 4:2:2 avec 10 bits par pixel mais dont les échantillons sont transmis par paquet et non par plan. La Figure 6.5 illustre la distinction entre le format V210 et 4:2:2 planaire avec 10 bits par pixel.

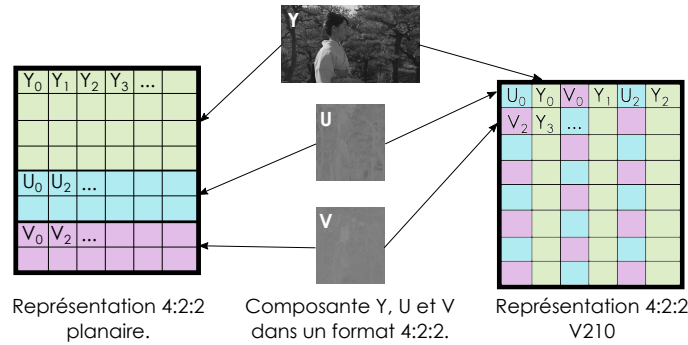


FIGURE 6.5 – Représentations planaire et en V210 d'une image au format 4 :2 :2 avec 10 bits par échantillon.

La reconstitution d'une grande image à partir des quatre plus petites ainsi que le ré-échantillonnage nécessitent un recours aux outils proposés par le constructeur et définis dans une interface de programmation applicative ("Application Programming Interface" (API)). L'utilisation de cette "Application Programming Interface" (API) nécessite alors d'allouer des ressources matérielles et de constituer des threads pour les différents traitements. La réception de l'image $3840 \times 2160p$ requiert d'attribuer un thread par quadrant d'image $1920 \times 1080p$ reçu. Le ré-échantillonnage puis la reconstitution de l'image de résolution $3840 \times 2160p$ requiert un thread chacun. L'ensemble du module d'acquisition requiert alors 6 threads. Afin d'assurer le traitement temps-réel du module d'acquisition, il est nécessaire d'attribuer les 6 threads à un certain nombre de cœurs. Sur les plateformes considérées et présentées dans la Section 6.2, un unique cœur suffit à assurer l'acquisition des images $1920 \times 1080p$ à 30 IPS sur chaque quadrant en temps-réel. Pour des fréquences d'images supérieures, il est nécessaire d'attribuer au moins 3 cœurs. Dans le cadre de l'application proposée, un seul cœur sera donc suffisant pour le module d'acquisition.

Une fois l'image reçue via la carte d'acquisition, celle-ci est transmise au module de management via un buffer partagé. Le module de management pourra alors transmettre l'image de résolution $3840 \times 2160p$ au module de codage.

6.1.3 Le module de codage

Le module de codage utilise l'architecture du SHVC-AE : il instancie les deux SLE correspondants aux couches de base et de rehaussement. Ces deux couches reçoivent une copie de l'image non compressée. Le pipeline de codage employé dans le module de codage est présenté en Figure 4.10. Il effectue :

- un sous-échantillonnage sur la couche de base,
- un sur-échantillonnage sur la couche de rehaussement,
- une prédiction inter-couches à partir des textures des images reconstruites issues des références temporelles HEVC de la couche de base.

La couche de base démarre son traitement dès que l'image est reçue tandis que la couche de rehaussement met l'image reçue en attente pendant 3 cycles de pipeline. La couche de base exécute le sur-échantillonnage sur les textures une fois sa décision de codage achevée. Le début de la décision de codage de la couche de rehaussement est synchronisé avec la fin du sur-échantillonnage, d'où le décalage de 3 cycles de pipeline.

Les étapes de sous-échantillonnage et de sur-échantillonnage emploient les optimisations présentées en Section 4.2. Pour ces opérations, les données sont divisées en 4 et 2 régions respectivement pour les composantes de luminance (Y) et de chrominances (U et V) avec un thread attribué par région. Les filtres de sous-échantillonnage et de sur-échantillonnage requièrent tous deux 8 threads.

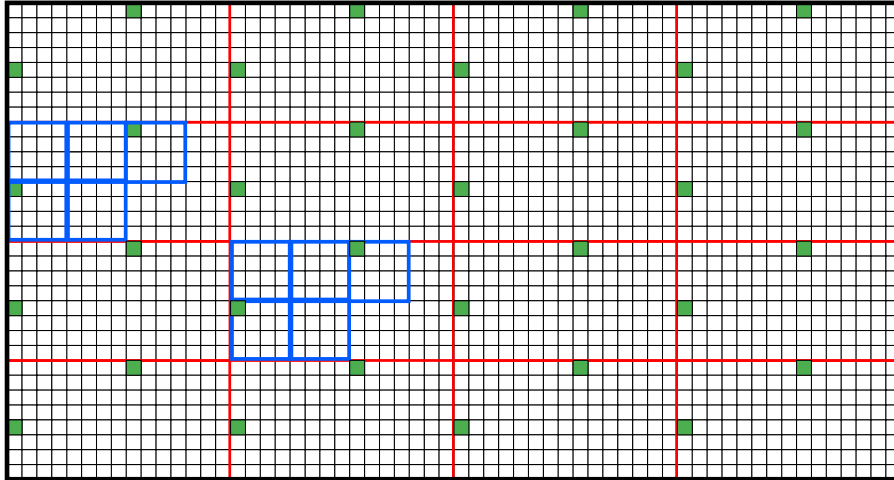


FIGURE 6.6 – Parallélisme sur les CU de taille 16×16 pixels. En rouge, les 4×4 Tiles. En bleu, les régions 64×64 pixels correspondant à un thread chacune. En vert, les CU traitées au même moment.

L'étape de décision de codage emploie un parallélisme sur les Tiles mais également sur son parcours des "Coding Units" (CU). Ce parcours des blocs n'est pas signalé et peut être combiné avec la répartition en Tiles. Le nombre de blocs, et donc de threads, maximum pouvant être traité en parallèle est présenté dans la Table 3.2. La répartition en Tiles 4×4 (4 Tiles sur les lignes et 4 Tiles sur les colonnes) permet d'avoir au minimum 16 lignes (ou 16 threads) exécutées en parallèle. Le mode de qualité employé, LIVE-UHD, définit une taille de CU minimum de 16×16 pixels (voir Table 4.10). Toutefois, chaque traitement de CU est intégré dans un thread contenant 4 CU. Le parcours des blocs ligne par ligne (i.e. "wavefront") est alors effectué sur ces ensembles de 4 CU. Par conséquent, sur des séquences sources de résolution $3840 \times 2160p$, pour chaque Tile, alors de résolution $960 \times 540p$, le nombre maximum de threads sur les couches de base et de rehaussement s'élève respectivement à 3 et 7. La Figure 6.6 illustre ce parallélisme de traitement sur les CU. Au total, pour des résolutions $3840 \times 2160p$, les couches de base et de rehaussement peuvent traiter respectivement jusqu'à 48 et 112 CU en même temps.

Enfin, les autres étapes du pipeline requièrent un thread chacune. Le nombre total de threads exécuté dans le module de codage en parallèle s'élève à 182. La formule suivante détaille son calcul avec :

- $NbThreads_{Total}$ le nombre de threads total,
- $NbThreads_{Sur-éch.}$ le nombre de threads du sur-échantillonnage (ici 8),
- $NbThreads_{Sous-éch.}$ le nombre de threads du sous-échantillonnage (ici 8),
- $NbCouches$ le nombre de couches (ici 2),
- $NbThreads_{AutresEtages}$ le nombre de threads des autres étapes du pipeline (ici 4),
- $NbThreads_{Déc. \text{ codage}BL}$ le nombre de threads de la décision de codage sur la couche de base (ici 48),
- $NbThreads_{Déc. \text{ codage}EL}$ le nombre de threads de la décision de codage sur la couche de rehaussement (ici 112).

$$\begin{aligned}
 NbThreads_{Total} = & NbThreads_{Sur-éch.} + NbThreads_{Sous-éch.} \\
 & + NbCouches \times NbThreads_{AutresEtages} \\
 & + NbThreads_{Déc. \text{ codage}BL} \\
 & + NbThreads_{Déc. \text{ codage}EL}
 \end{aligned} \tag{6.1}$$

Une fois la décision de codage effectuée sur la couche de rehaussement, le multiplexage du bitstream est effectué par le module de transmission.

6.1.4 Le module de transmission

Le module de transmission gère l'interface de sortie du codage temps-réel. Ce module effectue tout d'abord l'opération de multiplexage du bitstream afin d'agencer les différentes "Network Abstraction Layer Unit" (NALU) dans le bon ordre : vps, sps, pps, entête de slice... Ensuite, le bitstream (ou "Elementary Stream" (ES)) de la vidéo

est placé dans un conteneur avant de l'envoyer sur la liaison IP. L'ES vidéo contient l'ensemble de la syntaxe SHVC. De la même manière que pour la compression vidéo, l'encapsulation de l'ES dans un conteneur est standardisée. Elle s'appuie sur le standard MPEG-2 définissant notamment les paquets MPEG-2 "Transport Stream" (MPEG-2 TS) qui sont constitués d'une entête et d'une "payload" comme schématisé sur la Figure 6.7. L'entête du MPEG-2 TS permet d'identifier les différents types d'ES contenus dans sa "payload". L'élément de syntaxe permettant cette identification est le Paquet d'Identification ("Packet Identification") (PID). La correspondance entre un ES et son Paquet d'Identification ("Packet Identification") (PID) se trouve dans une table qui doit être transmise en tout début de transmission. Enfin, la "payload" contient un ES correspondant à un codage vidéo ou un codage audio ou une piste de sous-titres...

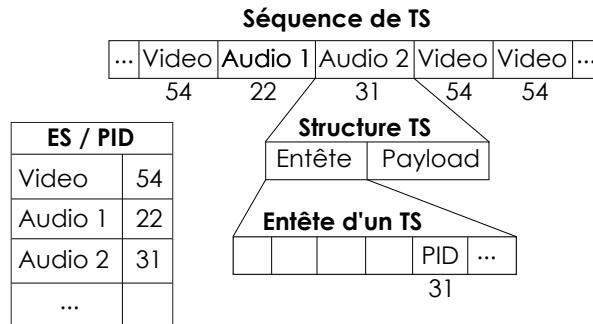


FIGURE 6.7 – Schéma de la structure des paquets TS avec sa "payload" qui contient notamment l'ES du codage vidéo.

En ce qui concerne la mise en application, il existe deux principaux standards de diffusion s'appuyant sur le standard MPEG-2 TS appelés "Digital Video Broadcasting" (DVB) et "Advanced Television Systems Committee" (ATSC). Tous deux proposent des solutions de diffusion pour la télévision en s'appuyant sur les différents standards techniques existants qu'ils concernent la transmission (MPEG-2 TS) ou la compression (H.262/MPEG-2, H.264/MPEG-4 AVC, HEVC). Les recommandations DVB sont utilisées en Afrique, en Europe, en Océanie, au Moyen-Orient et en Asie du sud. Les recommandations ATSC sont majoritairement utilisées en Amérique du nord et en Corée du Sud. Dans le cas du codage SHVC, les groupes DVB et ATSC ne recommandent pas la même utilisation du PID. En effet, le groupe DVB recommande d'employer un PID de vidéo différent par couche scalable tandis que les membres de l'ATSC recommandent d'employer un PID pour la vidéo dans son ensemble. Les recommandations DVB impliquent donc de contenir la couche de base dans un ES différent de celui de la couche de rehaussement. Les recommandations ATSC, quant à elle, impliquent d'employer un unique ES pour les couches de base et de rehaussement.

En ce qui concerne la chaîne de transmission ATSC illustrée sur la Figure 6.8, celle-ci intègre un codeur temps-réel qui supporte les standards de compression HEVC et SHVC. Elle reçoit des séquences non compressées depuis :

- un diffuseur (ou streamer) sur lequel la séquence est lue depuis un appareil de stockage,
- ou une caméra captant en temps-réel la séquence à diffuser.

Ensuite, l'ES est transmis sur liaison IP avec un protocole Streaming 3GPP Dynamique et Adaptatif sur flux HTTP ("3GPP Dynamic Adaptive Streaming over HTTP") (3GP-DASH) (voir section 2.9 jusqu'à "Guide Builder". Ce *Guide Builder* sépare les différents flux afin de les intégrer dans différents services. Il transmet ces services au modulateur accompagnés d'une signalisation qui décrit et qualifie les services. Ces descriptions permettent au modulateur de savoir quel traitement, protocole et destination correspond à chaque service. Le modulateur se charge alors de la mise en forme pour le transport sur de plus grande distance en passant par les canaux terrestre, filaire ou satellite. Enfin, les différents appareils d'affichage réceptionnent les services qu'ils supportent et affiche la vidéo décodée.

Dans le cas du codeur SHVC-AE, par défaut, l'ES de la vidéo ne correspond qu'à un seul PID afin de suivre les recommandations du standard ATSC 3.0. Néanmoins, le codeur SHVC-AE permet également d'attribuer un PID différent à chaque couche scalable pour être conforme aux recommandations DVB. Ce choix d'implémentation s'explique par l'utilisation du codeur SHVC-AE en situation de diffusion temps-réel au cours de plusieurs démonstrations ATSC 3.0.

Le module de transmission génère un fichier MPEG-2 TS (ATSC 3.0 ou DVB) à partir du bitstream SHVC. Ce

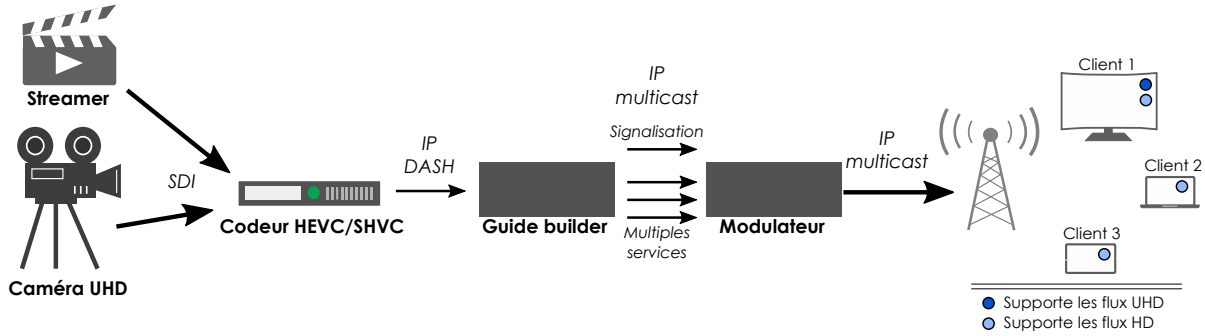


FIGURE 6.8 – Schéma de la chaîne de diffusion ATSC 3.0.

TABLE 6.2 – Caractéristiques notables des plateformes considérées.

Caractéristiques	E5-4627V2	E5-4627V3
Nombre de CPU	4	4
Nombre de cœurs par CPU	8	10
Fréquence des cœurs	3.30 GHz	2.60 GHz
Taille de cache par cœurs	16 Mo	25 Mo
Mémoire RAM partagée entre les cœurs	64 Go	128 Go
Compatible SSE2	oui	oui

fichier MPEG-2 TS est ensuite transmis sur une liaison IP. L'ensemble des opérations du module de transmission peut être effectué sur un unique thread.

Les différents threads attribués aux modules de management, d'acquisition, de codage et de transmission doivent alors être répartis sur les cœurs des CPU présents sur les plateformes considérées.

6.2 Agencement des modules pour un codage sur plateformes en temps-réel

Afin de pouvoir réaliser l'intégration des différents modules dans la plateforme de codage, ceux-ci se voient attribuer une quantité de ressources matérielles. Les ressources attribuées doivent permettre d'exécuter l'ensemble des threads de ces modules. Les nombres de threads par module sont les suivants :

- 1 thread pour le module de management,
- 6 threads pour le module d'acquisition,
- 182 threads pour le module de codage,
- 1 thread pour le module de transmission.

La quantité de ressources matérielles s'expriment en nombre de cœurs.

Les plateformes considérées pour l'implémentation temps-réel sont les plateformes Intel Xeon E5-4627V2 et Intel Xeon E5-4627V3. Les caractéristiques de ces plateformes sont présentées dans la Table 6.2. Les principales différences entre ces deux plateformes sont le nombre de cœurs par CPU, respectivement 8 et 10 cœurs, et la quantité de mémoire vive, respectivement 64 Go et 128 Go.

Le nombre de cœur attribué à un module dépend de l'application souhaitée. Nous souhaitons pouvoir coder des séquences de résolution $3840 \times 2160p$ à 30 IPS avec 10 bits par pixel en temps-réel. Les threads du module d'acquisition peuvent se contenter d'un cœur. Le module de transmission IP requiert 1 cœur pour fonctionner en temps-réel. Toutefois, les opérations de ce module n'étant pas les plus complexes, celui-ci n'exploite pas au mieux les performances du cœur qui lui est attribué. De plus, le module de management ne nécessite pas, lui non plus, une utilisation intense des ressources matérielles. En effet, après une phase d'initialisation, le module de management n'est opérationnel qu'entre une fin d'exécution d'un cycle de pipeline et le début du cycle suivant. Le module de management ne fonctionne alors que par impulsion espacée de la durée d'un cycle de pipeline. C'est pourquoi, nous avons choisi de partager un unique cœur entre les modules de management et de transmission IP. Les différents

threads de ces modules seront mis en concurrence sur ce cœur. Enfin, le module de codage peut bénéficier du reste des cœurs, à savoir 30 et 38 cœurs respectivement sur les plateformes E5-4627V2 et E5-4627V3. Ce nombre de cœur peut alors être :

- partagé pour tous les threads des couches de base et de rehaussement, voir Figure 6.9,
- réparti également entre la couche de base et la couche de rehaussement, voir Figure 6.10,
- réparti entre les couches avec plus de cœurs sur la couche de base, voir Figure 6.11,
- réparti entre les couches avec plus de cœurs sur la couche de rehaussement, voir Figure 6.12.

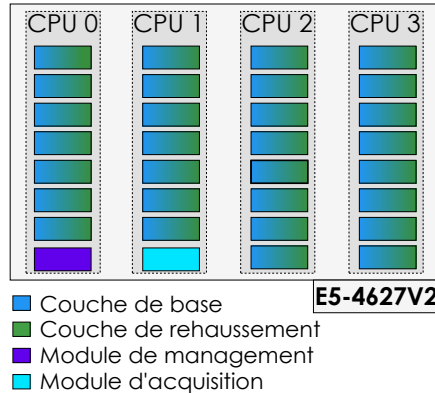


FIGURE 6.9 – Répartition des cœurs avec mise en concurrence des threads des couches de base et de rehaussement sur la plateforme Intel Xeon E5-4627V2. Simple à utiliser mais nécessite un gestionnaire de tâches complexe avec des pré-requis sur la durée des tâches.

Une répartition employant une mise en concurrence des opérations de codage des couches de base et de rehaussement permet de laisser le gestionnaire de tâches gérer l’occupation des cœurs. La Figure 6.9 schématise cette répartition. Néanmoins, la gestion des tâches nécessite de connaître la taille(ou la durée) des tâches afin de ne pas engendrer d’attente inter-threads. En effet, afin de ne pas être dans une situation où un seul cœur réalise une opération longue tandis que les autres cœurs n’ont plus d’opérations à réaliser, la durée des tâches, ou du moins une estimation de cette durée, est nécessaire afin d’ordonner les différents threads.

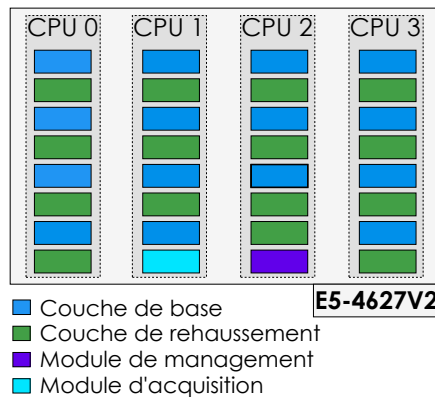


FIGURE 6.10 – Répartition d’un même nombre de cœurs sur les couches de base et de rehaussement sur plateforme Intel Xeon E5-4627V2. Considère une complexité équivalente entre couche de base et couche de rehaussement. Ce n’est pas le cas en scalabilité spatiale.

Une répartition employant le même nombre de cœurs sur les couches de base et de rehaussement, comme illustrée sur la Figure 6.10, considèrent une complexité équivalente entre ces couches. Dans le cas du SHVC-AE, qui emploie une scalabilité spatiale, les complexités de ces deux couches ne sont pas équivalentes. La couche de base peut alors employer un mode de qualité supérieur à celui de la couche de rehaussement. Toutefois, le codage en mode de qualité LIVE-UHD sur la couche de rehaussement avec seulement la moitié des cœurs de la plateforme

ne suffit pas à garantir la tenue des performances temps-réel. Il serait nécessaire d'accepter un compromis entre vitesse et distorsion permettant un codage encore plus rapide.

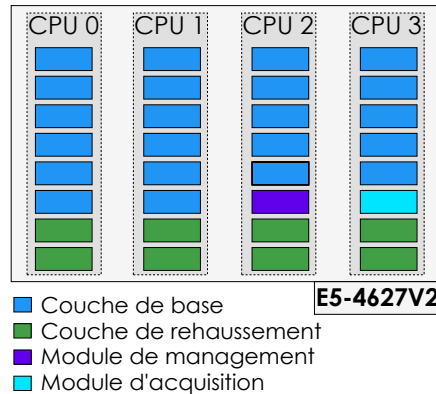


FIGURE 6.11 – Répartition employant un plus grand nombre de cœurs sur la couche de base sur plateforme Intel Xeon E5-4627V2. Considère une complexité sur la couche de base supérieure ou équivalente à celle de la couche de rehaussement. Ce n'est pas le cas en scalabilité spatiale.

De la même manière, une répartition employant un plus grand nombre de cœurs sur la couche de base, illustrée sur la Figure 6.11, permet d'avoir une complexité supérieure du traitement sur la couche de base par rapport à celle de la couche de rehaussement. Dans le cas du SHVC-AE, une utilisation d'un mode de qualité supérieur sur la couche de base est, là aussi, nécessaire pour ne pas laisser de ressource en famine de traitement. Néanmoins, cette répartition nécessite également une diminution de la complexité du traitement en LIVE-UHD sur la couche de rehaussement. Ceci n'est alors possible qu'au coût d'une dégradation significative du rapport entre qualité et débit sur cette couche.

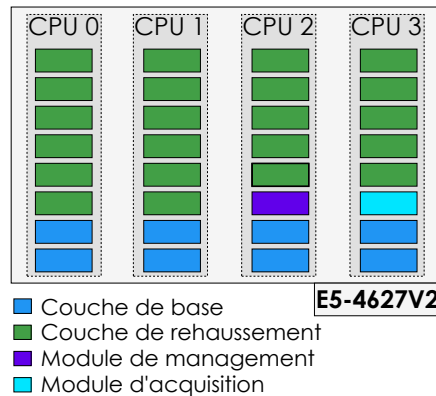


FIGURE 6.12 – Répartition employant un plus grand nombre de cœurs sur la couche de rehaussement sur plateforme Intel Xeon E5-4627V2. Considère une complexité sur la couche de base inférieure à celle de la couche de rehaussement. Cette répartition correspond à la scalabilité spatiale.

Enfin, une répartition des cœurs plus généreuse envers la couche de rehaussement, comme illustrée sur la Figure 6.12, considère une complexité supérieure sur le traitement de la couche de rehaussement par rapport à la couche de base. Cette répartition s'approche davantage du comportement du codage scalable en résolution puisque pour un même mode de qualité et à vitesse de codage, la couche de rehaussement requiert un plus grand nombre de ressources que la couche de base.

Cependant, la répartition des modules sur les différents cœurs nécessite également une approche au niveau de la gestion mémoire des CPU. En effet, la communication entre des cœurs présents sur différents CPU (inter-CPU) est plus longue que la communication entre des cœurs présents sur le même CPU (intra-CPU). Sur les plateformes considérées, la communication intra-CPU repose notamment sur une mémoire cache partagée comme schématisé

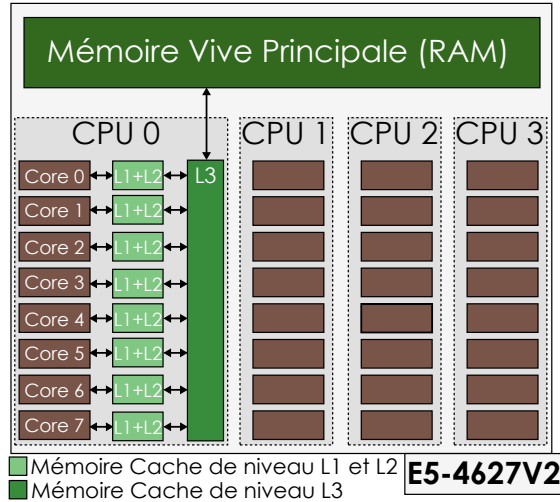


FIGURE 6.13 – Schéma de l’architecture du Intel Xeon E5-4627V2. La communication intra-CPU passe par les caches L1, L2 et L3 avec des temps d’accès courts. La communication inter-CPU passe par la mémoire vive avec des temps d’accès plus longs.

sur la Figure 6.13. Cette mémoire cache partagée est accessible à travers 3 niveaux de caches : L1, L2 et L3. Les niveaux de caches L1 et L2 sont propres à chaque cœur. Le niveau L3 est accessible plus rapidement par les cœurs par comparaison à un accès direct à la mémoire vive principale. La communication inter-CPU, quant à elle, requiert un accès à la mémoire vive principale. La mémoire vive n’est pas présente sur les CPU et donc requiert des bus dédiés pour y accéder. Ces accès via les bus sont plus lents que les accès aux différentes caches. La communication intra-CPU est alors plus rapide que la communication inter-CPU. Dans ce contexte, il est plus intéressant d’employer la communication inter-CPU pour des échanges d’une grande quantité de données, mais également d’éviter d’y avoir recours pour de multiples échanges d’une faible quantité de données.

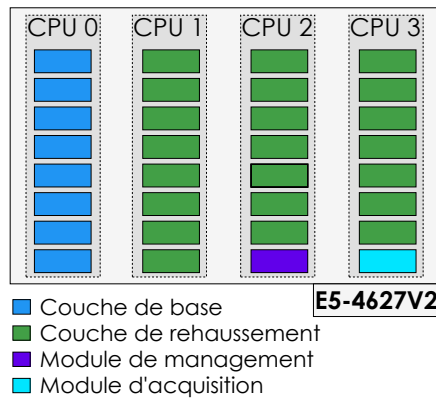


FIGURE 6.14 – Attribution des cœurs aux différentes opérations du codage SHVC sur plateforme Intel Xeon E5-4627V2 avec le mode de qualité LIVE-UHD.

La Figure 6.14 illustre la répartition des différents modules sur la plateforme Intel Xeon E5-4627V2 employé pour les démonstrations temps-réel du codeur SHVC-AE. Davantage de cœurs sont attribués à la couche de rehaussement pour tenir compte de l’écart en terme de quantité de données à traiter par rapport à la couche de base. De plus, cette répartition tient compte des communications entre cœurs. Ainsi, la couche de base dispose d’un CPU entier, soit 8 cœurs, alors que la couche de rehaussement dispose des cœurs restant sur les trois autres CPU, soit 22 cœurs. Cette répartition permet d’éviter au maximum l’utilisation des transferts inter-CPU pour des données de codage intermédiaires sur la couche de base. La couche de base emploie alors la communication inter-CPU uniquement pour accéder à l’image source et pour transmettre les informations de codage respectivement en début

et en fin de pipeline. Ces informations de codage sont d'ailleurs transmises à la fois à la couche de rehaussement pour la prédiction inter-couches et au module de management en vue de la transmission du bitstream. La couche de rehaussement, quant à elle, dispose de cœurs répartis sur plusieurs CPU. Le temps de codage de cette couche est ainsi affecté par la gestion mémoire. Ceci limite alors l'utilisation du SHVC-AE à une utilisation temps-réel sur des séquences au format $3840 \times 2160p$ à 30 IPS avec 10 bits par pixel. Cette répartition des ressources permet l'utilisation du SHVC-AE en mode de qualité LIVE-UHD et non pas LIVE-UHD+. La complexité du codage sur la couche de base étant augmentée dans le mode LIVE-UHD+, il est nécessaire de lui attribuer davantage de ressources matérielles. Le nombre de cœurs employé dans la couche de base permet tout juste de tenir les performances temps-réel dans le mode LIVE-UHD sur la plateforme E5-4627V2. Afin de pouvoir respecter la contrainte temps-réel sur la couche de base en mode de qualité LIVE-UHD+, il est nécessaire d'employer la plateforme E5-4627V3. Cette dernière dispose de 2 cœurs supplémentaire par CPU.

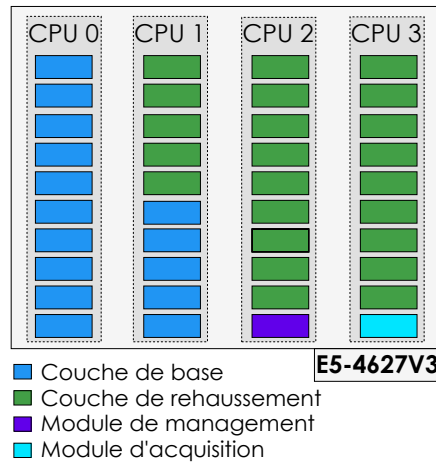


FIGURE 6.15 – Attribution des cœurs aux différentes opérations du codage SHVC sur la plateforme Intel Xeon E5-4627V3 avec le mode de qualité LIVE-UHD+.

La Figure 6.15 illustre la répartition employée sur la plateforme E5-4627V3 pour une codage temps-réel en mode de qualité LIVE-UHD+. Afin de gérer la complexité de ce mode sur la couche de base, la moitié des cœurs d'un CPU lui est attribué en supplément du CPU dont elle dispose déjà. Ces 5 cœurs supplémentaires permettent également de compenser les pertes dues aux transferts inter-CPU pour accéder à des données de codage intermédiaires. De plus, le mode de qualité LIVE-UHD+ dans les configurations low-delay P (LD-P) et random-access (RA) permet une vitesse de codage suffisamment élevée sur les couches de base et de rehaussement pour que ces pertes soient compensées. En définitive, les ressources attribuées aux différents modules sont :

- 1 cœur partagé entre les 2 threads des modules de management et de transmission,
- 1 cœur pour les 5 threads du module d'acquisition,
- 30 ou 38 cœurs pour les 182 threads du module de codage respectivement avec les plateformes E5-4627V2 et E5-4627V3. Le module de codage est lui-même réparti en :
 - 8 ou 15 cœurs pour la couche de base,
 - 22 ou 23 cœurs pour la couche de rehaussement.

Les occupations moyennes des cœurs sont présentées dans la Tables6.3. Les mesures de ces occupations sont obtenues en calculant la moyenne des occupations sur 10 captures avec l'outil *htop*. *htop* est un moniteur système pour les systèmes d'exploitation Unix. Il permet notamment d'obtenir une mesure sur l'occupation des cœurs mais également sur l'occupation mémoire et le nombre de threads en cours d'exécution. La plupart du temps, les modules de management et de transmission occupent très peu leur cœur commun. Toutefois, quand ces deux modules doivent réaliser leurs opérations en même temps, l'occupation du cœur peut monter jusqu'à 95% de charge. Le module d'acquisition des images occupent en moyenne 60% et 85% de la charge de son cœur respectivement pour des profondeurs de 8 et 10 bits par pixel. La couche de base utilisent peu les transferts inter-CPU, c'est pourquoi l'occupation de ses cœurs, à un instant donné, varie seulement de 10 % en moyenne. En revanche, les cœurs attribués à la couche de rehaussement utilisent régulièrement les transferts inter-CPU, ce qui explique les variations de charge sur ses cœurs de 35% en moyenne.

TABLE 6.3 – Charges moyennes des cœurs (en pourcentage) sur les plateformes E5-4627V2 et E5-4627V3 respectivement pour le mode de qualité LIVE-UHD et LIVE-UHD+. Mesures des charges obtenues avec 10 captures *htop*.

Modules	E5-4627V2		E5-4627V3	
	LIVE-UHD	LIVE-UHD	LIVE-UHD	LIVE-UHD+
Management + Transmission	3% à faible occupation, 95% à forte occupation.			
Acquisition	60% et 85% en 8 bits et 10 bits			
Codage couche de base $\pm 10\%$	85%	65%	75%	
Codage couche de rehaussement $\pm 35\%$	65%	55%	60%	

Pour finir, le multiplexage des NALU des différentes couches est réalisé dans le module de transmission IP. Les NALU issues de la couche de base sont stockées durant deux cycles de pipeline en attendant celles de la couche de rehaussement. Une fois toutes les NALU d'une même image disponibles, celles-ci sont alors multiplexées et ré-ordonnées afin de rendre le bitstream conforme au standard SHVC. Ce bitstream SHVC est alors transmis par liaison IP via le module de transmission. Cette opération étant peu coûteuse, le module de transmission utilise le même cœur que le module de management.

6.3 Démonstrations du codeur SHVC

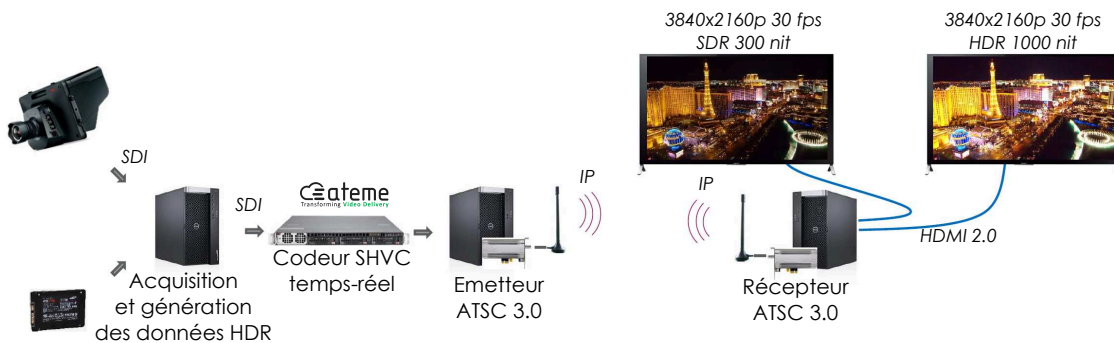


FIGURE 6.16 – Schéma de la démonstration ATSC 3.0, SHVC, HDR conjointe Technicolor-Ateme.

Le codeur SHVC-AE dans sa configuration LIVE avec les diverses optimisations proposées et exploitant une acquisition sur liaison SDI a pu démontrer ses performances temps-réel lors de plusieurs meetings ATSC [87, 88], lors de salon comme les "National Association of Broadcasters (NAB) Show" 2016 et 2017 ou "l'International Broadcasting Convention (IBC)" 2016 ainsi que lors d'une session de démonstration en conférence [89]. Ces différentes démonstrations ont permis de faire rapidement évoluer le code du SHVC-AE au cours de cette thèse. En effet, ces démonstrations étant datées et nécessitant des contraintes bien précises, elles ont nécessité un effort supplémentaire sur certaines périodes de la thèse. Pendant ces périodes de développement intensif, les travaux de thèse étaient entièrement consacrés au développement du SHVC-AE. Les contraintes concernées sont par exemple :

- la production d'un bitstream conforme au standard SHVC,
- le codage en temps-réel sur des séquences 3840×2160p,
- le codage avec des profondeurs sur les pixels de 8 et 10 bits.

Par la suite, ces développements ont pu être améliorés et généralisés afin de pouvoir répondre à d'autres types de contraintes :

- la recherche d'une meilleure efficacité de codage,

- la diminution de la latence globale,
- l'utilisation de la prédiction inter-couches sur plusieurs types d'image,
- l'amélioration du paramétrage du filtre de sous-échantillonnage.

La démonstration ATSC consiste à filmer une scène avec une caméra UHD (de résolution 3840×2160p) ou bien de lire une séquence UHD à partir d'un émetteur (ou "*streamer*"). La vidéo non compressée est émise vers le codeur sur une liaison SDI avec 4 quadrants. Le codeur réalise son codage scalable et émet un signal compressé vers un émetteur ATSC 3.0 via liaison IP. Plus loin, un récepteur ATSC 3.0 récupère ces données et les transmet à un ordinateur (PC) via une liaison IP. Le PC décode et joue la séquence décodée sur un ou plusieurs écrans supportant le format UHD. L'utilisation du PC compense l'absence de plateforme existante (décodeur embarqué ou téléviseur) permettant de réaliser le décodage du bitstream SHVC. Pour cela, il utilise le logiciel gpac [90] avec la solution de décodage openHEVC proposé dans [51]. Le logiciel gpac est composé de plusieurs outils dédiés aux transports et à la diffusion de vidéo. Le décodeur openHEVC est un décodeur open source permettant un décodage en temps-réel de flux HEVC et SHVC.

De plus, avec la coopération de Technicolor, un module d'acquisition des données "High Dynamic Range" (HDR) de type (SL-HDR1) a pu être placé entre la caméra et le codeur. Ce module génère des informations HDR transmises sur liaison SDI via des paquets appelés "ancillary data". Le codeur, lors de la réception des images avec la carte d'acquisition, traite ces données en parallèle du codage. Ces données sont extraites puis transmises sous forme de message Information d'Amélioration Supplémentaire ("Supplemental Enhancement Information") (SEI). Ce n'est qu'au niveau de l'affichage (téléviseur) que ces données sont exploitées pour un rendu HDR de la vidéo sur toutes les couches scalables. Il est possible de ne pas prendre en compte ces données, auquel cas, la vidéo sera affichée en "Standard Dynamic Range" (SDR). Cette configuration permet alors au SHVC-AE de réaliser des encodages temps-réel sur des séquences HDR sur les couches de base et de rehaussement.

Conclusion

En conclusion, ce chapitre a présenté l'intégration de la solution logicielle SHVC-AE dans une chaîne de diffusion en condition temps-réel. L'intégration par module permet d'abstraire les opérations d'interface des opérations de codage. Le module de management pilote les modules d'acquisition, de codage et de transmission afin de synchroniser leurs opérations respectives. Le module d'acquisition s'appuie sur l'utilisation d'une carte d'acquisition sur liaison SDI qui permet la réception d'images de résolution 3840×2160p en temps-réel. Le module de codage fait appel au SHVC-AE afin de coder l'image réceptionnée avant de la transmettre. Enfin, le module de transmission permet de mettre en forme et de diffuser le bitstream SHVC sur des liaisons IP. Plusieurs stratégies d'attribution des ressources matérielles ont été présentées, chacune effectuant des compromis entre simplicité d'intégration et gestion de la complexité du codage par couche. La stratégie retenue prend en compte à la fois la complexité de chaque module mais également les temps de transfert des données inter-CPU et intra-CPU. Cette stratégie a permis d'intégrer le codeur SHVC-AE dans une plateforme de codage à destination d'une chaîne de diffusion de type ATSC 3.0. Les différentes démonstrations de cette plateforme de codage ont permis à la fois de promouvoir le codeur SHVC-AE mais également de démontrer l'intérêt de la solution de codage scalable SHVC dans le monde de la diffusion vidéo.

Chapitre 7

Conclusion

7.1 Travaux réalisés

Dans une première partie, nous avons présenté les notions de codage vidéo ainsi que les tendances actuelles dans le monde de la diffusion vidéo. Les différentes contraintes, dans un tel contexte, requièrent des moyens permettant au codage d'être à la fois :

- efficace en terme de compression,
- capable d'être effectué rapidement voire en temps-réel,
- transposable pour des transmissions vers de multiples appareils.

Le codage SHVC permet de faire cela grâce à un codage sur plusieurs couches. Chaque couche correspond à un format d'image avec des caractéristiques ciblées : résolution, fréquence, qualité "Signal to Noise Ratio" (SNR), profondeur des pixels, espace des couleurs... Une prédiction inter-couches permet d'améliorer l'efficacité de codage par rapport à un codage simulcast, c'est à dire effectuant autant de codages HEVC qu'il y a de couches scalables. Le codage SHVC présente alors un intérêt pour la diffusion. Celle ci a donc défini plusieurs scénarios utilisant ce standard à partir des contraintes de diffusion actuelles et futures.

Nous présentons, dans une deuxième partie, notre méthode d'intégration du codage SHVC dans une configuration cherchant la meilleure efficacité de codage possible (i.e. FILE) avec deux couches scalables en résolution (i.e. spatiale). Cette méthode, proposée dans [59], s'appuie sur l'architecture d'un codeur professionnel, le Single Layer Encoder (SLE), afin de réduire au maximum la latence inter-couches. Chaque couche intègre une architecture en pipeline permettant de réaliser un parallélisme inter-images, intra-images, inter-blocs et inter-opérations. Le parallélisme inter-couches proposé requiert une synchronisation qui est effectuée au niveau du pipeline. Les couches sont donc traitées en parallèle avec un décalage à l'initialisation afin de permettre la prédiction inter-couches pour une scalabilité spatiale. Le codeur proposé dans [63], appelé "SHVC ATEME Encoder" (SHVC-AE), dans sa configuration FILE, montre des résultats permettant de fournir des encodages comparables à l'état de l'art tout en réduisant très significativement le temps de traitement. Le codeur SHVC-AE réalise un compromis entre distorsion et complexité en employant une prédiction inter-couches sur les textures des images de référence de type I, P et B qui servent également de référence dans le codage HEVC. Les vecteurs de mouvement issus de la couche de base ne permettent pas d'améliorer l'efficacité de codage et ne sont pas employés dans le SHVC-AE. Dans cette configuration FILE, l'efficacité de codage du SHVC-AE est légèrement inférieure à celle du codeur de référence, le "Scalable HEVC Model" (SHM), mais sa vitesse de codage lui est largement supérieure.

Dans une troisième partie, nous présentons des améliorations du SHVC-AE en scalabilité spatiale sur deux couches afin d'accélérer la vitesse de codage. Une première amélioration consiste à définir un paramétrage du filtre de sous-échantillonnage nécessaire à la couche de base. Ce paramétrage tire profit de la prédiction inter-couches en augmentant la complexité du codage de la couche de base afin de fournir de meilleures informations de codage à la couche de rehaussement. Il en résulte un accroissement du débit de codage sur la couche de base pour une plus grande réduction du débit sur la couche de rehaussement. Le paramétrage du filtre de sous-échantillonnage permet alors d'améliorer le débit global. L'intégration du filtre de sous-échantillonnage au pipeline de codage du SHVC-AE permet son utilisation dans une configuration temps-réel (LIVE). De plus, plusieurs modes de qualité permettant un codage SHVC en temps-réel, proposés dans [64], sont définis pour des résolutions ciblées :

- LIVE-SD pour des formats Simple Définition (SD),

- LIVE-HD pour des formats Haute Définition (HD),
- et LIVE-UHD pour des formats Ultra Haute Définition (UHD).

L'accélération du codage repose sur un compromis entre distorsion et complexité appliqué au niveau des outils de codage issus du standard HEVC. Ces modes de qualité peuvent être appliqués indépendamment à chaque couche permettant la définition des modes LIVE-HD+ et LIVE-UHD+ composés respectivement des modes LIVE-SD et LIVE-HD sur la couche de base et des modes LIVE-HD et LIVE-UHD sur la couche de rehaussement. Ces modes LIVE-HD+ et LIVE-UHD+ tirent un meilleur profit de la prédiction inter-couches en améliorant l'efficacité de codage sur la couche de base par rapport aux modes de qualité précédents. Par conséquent, ces deux nouveaux mode améliorent l'efficacité de codage du SHVC-AE tout en conservant un codage en temps-réel.

Nous présentons dans une quatrième partie, le potentiel modulable et générique de l'architecture employée dans le SHVC-AE. Nous affinons l'architecture employée en définissant les couches de base, de rehaussement intermédiaires et de rehaussement finale pour des codages multi-couches. Le SHVC-AE permet alors de réaliser des codages sur 3 couches scalables en résolution avec différentes dépendances tout en conservant son efficacité de codage ainsi que sa vitesse de traitement. Ensuite, nous démontrons la possibilité d'intégrer d'autres types de scalabilité au codeur SHVC-AE sans modifier son architecture.

Enfin, dans une cinquième partie, nous intégrons le SHVC-AE dans une chaîne de diffusion de type ATSC 3.0. Cette intégration est effectuée grâce à des modules d'interface entrées/sorties capable de réceptionner une image ou de transmettre le bitstream en temps-réel. Les différents modules de management, d'acquisition, de codage et de transmission effectuent leurs opérations dans des threads. Ces threads sont exécutés sur des cœurs attribués aux différents modules en prenant compte de leurs complexités ainsi que des temps de transferts entre les différents cœurs. La solution de codage alors proposée a été montrée plusieurs fois lors de salons, de meetings "Advanced Television Systems Committee" (ATSC) et de conférences de portées internationales.

7.2 Perspectives

Parmi les perspectives possibles, il est possible de proposer des améliorations sur l'efficacité du codage, les outils de codage employés, les performances de vitesse et l'intégration dans d'autres contextes de diffusion.

En terme d'amélioration d'efficacité de codage, il est envisageable d'ajouter une régulation du débit sur la prédiction inter-couches pour un codage à débit constant ou variable. Le SHVC-AE emploie une régulation du débit basée uniquement sur le codage HEVC. On trouve dans l'état de l'art des moyens permettant d'améliorer l'efficacité de codage avec une régulation du débit sur la prédiction inter-couches. L'ajout de tels outils est envisageable et constituerait une amélioration non négligeable pour des codages en condition temps-réel. Néanmoins, la régulation du débit représente un coût important en terme de complexité. C'est pourquoi un compromis entre distorsion et vitesse sur une régulation de débit temps-réel dédiée à la prédiction inter-couches pourrait constituer une étude intéressante et pertinente dans un contexte de diffusion.

Le SHVC-AE a prouvé qu'il pouvait être intégré dans une solution de codage faisant intervenir des plateformes matérielles. Cette solution de codage utilise une répartition en modules. D'autres applications utilisent ce genre de répartition comme notamment les solutions virtuelles. Les solutions virtuelles consistent à s'abstraire des notions matérielles en employant des systèmes qui mutualisent les ressources. Concrètement, un système virtuel s'appuie sur une ou plusieurs ressources matérielles en organisant le traitement attribué à chacune afin que l'utilisateur n'ait pas à intervenir au niveau des plateformes. Il permet également de s'abstraire des systèmes d'exploitation en apportant tous les outils nécessaires au bon fonctionnement du traitement souhaité. Dans le cas du codage vidéo, cette discipline est récente et permet de réaliser un système de diffusion sans achat de plateforme ou de ressource matérielle spécifique : les parties logicielles et virtuelles seulement sont suffisantes. La société Ateame propose des solutions de codage virtuelles en s'appuyant sur un mécanisme par module. Cette méthodologie correspond à celle employée dans le chapitre 6 et permet d'attribuer une fonctionnalité par module et une hiérarchie entre eux pour les différentes opérations d'un système de diffusion. La vitesse de traitement ne dépend plus seulement de la vitesse de chaque module mais également de leur façon de communiquer entre eux. Néanmoins, de manière générale, il reste des progrès à faire sur la communication entre ressources afin que la virtualisation puisse permettre d'assurer une réponse à des contraintes temps-réel.

Bibliographie

- [1] ITU-T, *High Efficiency Video Coding*, 2013.
- [2] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, “Overview of the Scalable Video Coding Extension of the H.264/AVC Standard,” *IEEE Transactions on Circuits and Systems for video Technology*, pp. 1103–1120, september 2007.
- [3] Edwin H. Land, “The Retinex Theory of Color Vision,” *Scientific American*, vol. 237, no. 6, pp. 108–128, december 1977.
- [4] 4EVER, “<http://www.4ever-project.com>,” .
- [5] 4EVER-2, “<http://www.4ever-2.com>,” .
- [6] Klimberg, Nathalie, “HDR et HFR : le consortium 4EVER-2 récompensé par un NAB TECHNOLOGY Innovation Award !,” *Mediakwest* - <http://www.mediakwest.com/communaute/item/nab-2016-le-consortium-4ever-2-presente-ses-recherches-sur-le-hdr-et-le-hfr-uhd.html>, april 2016.
- [7] Conseil Supérieur de l’Audiovisuel, République Française, “Présentation du standard d’image en Ultra-Haute Définition (UHD) aux membres du Conseil Supérieur de l’Audiovisuel,” <http://www.csa.fr/Etudes-et-publications/Les-dossiers-d-actualite/Presentation-du-standard-d-image-en-Ultra-Haute-Definition-UHD-aux-membres-du-Conseil-Superieur-de-l-Audiovisuel>, march 2017.
- [8] Marinoni, Fabrice, “4EVER-2, un workflow complet en High Frame Rate,” *Mediakwest* - <http://www.mediakwest.com/broadcast/item/mk22-4ever-2-un-workflow-complet-en-high-frame-rate.html>, august 2017.
- [9] thefuture.tv, “4EVER-2 PROVES HDR, HFR AND NEXTGEN AUDIO TECHNOLOGIES ARE OPERATIONAL AND NECESSARY !,” <http://www.thefuture.tv/?p=3032>, september 2017.
- [10] A. TaghaviNasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, “Adaptive 360-degree video streaming using layered video coding,” *IEEE Virtual Reality (VR)*, march 2017.
- [11] Mohammad Hosseini and Viswanathan Swaminatha, “Adaptive 360 VR Video Streaming : Divide and Conquer !,” *IEEE International Symposium on Multimedia (ISM)*, december 2016.
- [12] M. Budagavi, J. Furton, G. Jin, A. Saxena, J. Wilkinson, and A. Dickerson, “360 degrees video coding using region adaptive smoothing,” *IEEE International Conference on Image Processing (ICIP)*, september 2015.
- [13] Raulet, Mickaël and Toullec, Eric and Grégory, Lucas and Le Feuvre, Jean, “Tile-based high resolution VR contents delivery ,” october 2017.
- [14] K. Berger, Y. Koudota, M. Barkowsky, and P. Le Callet, “Subjective quality assessment comparing uhd and hd resolution in hevc transmission chains,” pp. 1–6, May 2015.
- [15] P. A. Kara, W. Robitza, A. Raake, and M. G. Martini, “The label knows better : The impact of labeling effects on perceived quality of hd and uhd video streaming,” pp. 1–6, May 2017.
- [16] Hiromi Nemoto, Philippe Hanhart, Pavel Korshunov, and Touradj Ebrahimi, “Impact of ultra high definition on visual attention,” pp. 247–256, 2014.
- [17] M. Cheon and J. S. Lee, “Subjective and objective quality assessment of compressed 4k uhd videos for immersive experience,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2017.
- [18] L. Krasula, M. Narwaria, K. Fliegel, and P. Le Callet, “Preference of experience in image tone-mapping : Dataset and framework for objective measures comparison,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 1, pp. 64–74, Feb 2017.

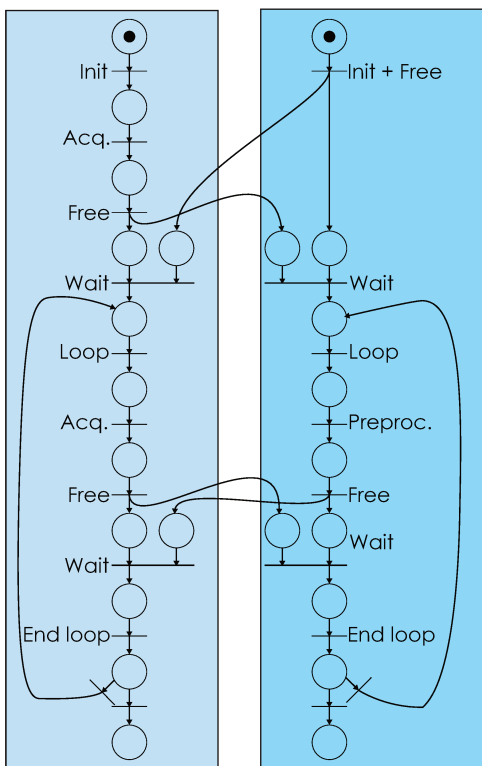
- [19] E. Nasiopoulos, Y. Dong, and A. Kingstone, "Evaluation of high dynamic range content viewing experience using eye-tracking data," pp. 13–17, Aug 2014.
- [20] P. Hanhart, M. Řeřábek, and T. Ebrahimi, "Subjective and objective evaluation of hdr video coding technologies," pp. 1–6, June 2016.
- [21] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Seikh, and Eero P. Simoncelli, "Image Quality Assessment : From Error Visibility to Structural Similarity," *IEEE Signal Processing Society*, vol. 13, no. 4, pp. 600–612, april 2004.
- [22] Gisle Bjontegaard, "Calculation of Average PSNR Differences between RD-curves," *ITU-T video Coding Experts Group document VCEG-M33*, march 2001.
- [23] Margaret H. Pinson, Lark Kwon Choi, and Alan Conrad Bovik, "Temporal Video Quality Model Accounting for Variable Frame Delay Distortions," *IEEE Transactions On Broadcasting*, vol. 60, no. 4, pp. 637–649, december 2014.
- [24] NETFLIX, "<https://medium.com/netflix-techblog/toward-a-practical-perceptual-video-quality-metric-653f208b9652>."
- [25] Cisco, "Cisco Visual Networking Index : Forecast and Methodology, 2016–2021," june 2017.
- [26] CCIT, "Recommendation H.120 : Codecs for Videoconferencing Using Primary Digital Group Transmission.," .
- [27] ITU-T, "Recommendation H.261 : Video Codec for Audiovisual Services at px64 kbits.," .
- [28] Jim Bankoski, Paul Wilkins, and Yaowu Xu, "Technical Overview of VP8, an Open-Source Video Codec for the Web," *IEEE International Conference on Multimedia and Expo (ICME)*, july 2011.
- [29] Christian Feller, Juergen Wuenschmann, Thorsten Roll, and Albrecht Rothermel, "The VP8 Video Codec - Overview and Comparison to H.264/AVC," *IEEE International Conference on Consumer Electronics - Berlin (ICCE - Berlin)*, september 2011.
- [30] Debargha Mukherjee, Jim Bankoski, Adrian Grange, Jingning Han, John Koleszar, Paul Wilkins, Yaowu Xu, and Ronald Bultje, "The latest open-source video codec VP9 - An overview and preliminary results," *SMPTE Motion Imaging Journal*, february 2015.
- [31] J. M. Valin, T. B. Terriberry, N. E. Egge, T. Daede, Y. Cho, C. Montgomery, and M. Bebenita, "Daala : Building a next-generation video codec from unconventional technology," *IEEE 18th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, Sept 2016.
- [32] G. Bjøntegaard, T. Davies, A. Fuldseth, and S. Midtskogen, "The thor video codec," *Data Compression Conference (DCC)*, pp. 476–485, March 2016.
- [33] Vadim Seregin and Yong He, "Common SHM test conditions and software reference configurations," *Joint Collaborative Team on Video Coding, JCTVC-Q1009, Valencia, ES*, march - april 2014.
- [34] Douglas F. Elliot, *Handbook of Digital Signal Processing Engineering Applications*, Academic Press, 1987.
- [35] Jill M. Boyce, Yan Ye, Jianle Chen, and Adarsh K. Ramasubramonian, "Overview of SHVC : Scalable Extensions of the High Efficiency Video Coding Standard," *IEEE Transaction on Circuits and Systems for Video Technology*, pp. 20–34, july 2015.
- [36] Shih-Hsuan Yang and Phuong Binh Vo, "Adaptive Bit Allocation for Consistent Video Quality in Scalable High Efficiency Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, july 2017.
- [37] Li Li, Bin Li, Dong Liu, and Houqiang Li, " λ -Domain Rate Control Algorithm for HEVC Scalable Extension," *IEEE Transactions on Multimedia*, vol. 18, no. 10, pp. 2023–2039, october 2016.
- [38] T. Biatek, W. Hamidouche, J. F. Travers, and O. Deforges, "Optimal bitrate allocation in the scalable hevc extension for the deployment of uhd services," *IEEE Transactions on Broadcasting*, vol. 62, no. 4, pp. 826–841, Dec 2016.
- [39] B. Li, H. Li, L. Li, and J. Zhang, " λ domain rate control algorithm for high efficiency video coding," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3841–3854, Sept 2014.

- [40] H. R. Tohidypour, H. Bashashati, M. T. Pourazad, and P. Nasiopoulos, "Online-learning-based mode prediction method for quality scalable extension of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 10, pp. 2204–2215, Oct 2017.
- [41] A. Heindel, E. Wige, and A. Kaup, "Low-complexity enhancement layer compression for scalable lossless video coding based on hevc," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 8, pp. 1749–1760, Aug 2017.
- [42] J. y. Lee, S. I. Park, S. Kwon, B. M. Lim, H. M. Kim, N. Hur, A. Pesin, J. C. Chevet, J. Llach, A. J. Stein, S. Jeon, and Y. Wu, "Efficient transmission of multiple broadcasting services using ldm and shvc," *IEEE Transactions on Broadcasting*, vol. PP, no. 99, pp. 1–11, 2017.
- [43] K. Misra, A. Segall, J. Zhao, S. H. Kim, J. Llach, A. Stein, J. Stewart, Hendry, Y. K. Wang, Y. Ye, and Y. He, "Spatially scalable hevc for layered division multiplexing in broadcast," in *2017 Data Compression Conference (DCC)*, April 2017, pp. 3–12.
- [44] P. L. Cabarat, W. Hamidouche, and O. Déforges, "Real-time and parallel shvc hybrid codec avc to hevc decoder," pp. 3046–3050, March 2017.
- [45] "Kvazaar git," <https://github.com/ultravideo/kvazaar/tree/Scalable>.
- [46] P. Sjövall, V. Viitamäki, A. Oinonen, J. Vanne, T. D. Hämäläinen, and A. Kulmala, "Kvazaar 4k hevc intra encoder on fpga accelerated airframe server," pp. 1–6, Oct 2017.
- [47] A. Lemmetti, A. Koivula, M. Viitanen, J. Vanne, and T. D. Hämäläinen, "Avx2-optimized kvazaar hevc intra encoder," pp. 549–553, Sept 2016.
- [48] "x265," <http://x265.org/>.
- [49] "Documentation x265 - threading," 2015, <http://x265.readthedocs.io/en/default/threading.html>.
- [50] "FFmpeg : Open source and cross-platform multimedia library," <http://www.ffmpeg.org>.
- [51] Wassim Hamidouche, Mickaël Raulet, and Olivier Deforges, "4K Real-Time and Parallel Software Video Decoder for Multilayer HEVC Extensions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 169–180, january 2016.
- [52] "A/341 : ATSC Candidate Standard : Video - HEVC," Approved 24 November 2015. Updated 3 January 2017. The CS period ends 30 January 2017.
- [53] ATSC, "S34-1-169r1-Summary of SHVC Testing," *ATSC 3.0 Documents*, may 2015.
- [54] "Study on Video Enhancements in 3GPP Multimedia Services," *3GPP TR-T12-26.987 V13.0.0*, december 2015.
- [55] Yan Ye Yong He Ye-Kui and Wang Hendry, "SHVC, the Scalable Extensions of HEVC, and Its Applications," *ZTE Communications*, vol. 1, 2016.
- [56] "SHM reference software version 10.0," <https://hevc.hhi.fraunhofer.de>.
- [57] ATEME, "<http://www.ateme.com>," .
- [58] The IEEE and The Open Group, "IEEE Std 1003.1," *The Open Group Technical Standard Base Specifications*, september 2016.
- [59] Ronan Parois, Wassim Hamidouche, Elie Mora, Mickaël Raulet, and Olivier Deforges, "Real-time UHD Scalable multi-layer HEVC encoder architecture," *24th European Signal Processing Conference*, september 2016.
- [60] ITU-T, *High Efficiency Video Coding*, 2014.
- [61] Gary Sullivan, "Color format downconversion for test sequence generation," *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Document JVT-I018*, september 2003.
- [62] "x265 : Preset options," <http://x265.readthedocs.io/en/default/presets.html>.
- [63] Ronan Parois, Wassim Hamidouche, Elie Mora, Mickaël Raulet, and Olivier Deforges, "Efficient Parallel Architecture of an Intra-only Scalable Multi-Layer HEVC encoder," *Conference on Design & Architectures for Signal & Image Processing (DASIP)*, october 2016.
- [64] Ronan Parois, Wassim Hamidouche, Jérôme Vieron, Mickaël Raulet, and Olivier Deforges, "Efficient Parallel Architecture for a Real-time UHD Scalable HEVC Encoder," *25th European Signal Processing Conference*, september 2017.

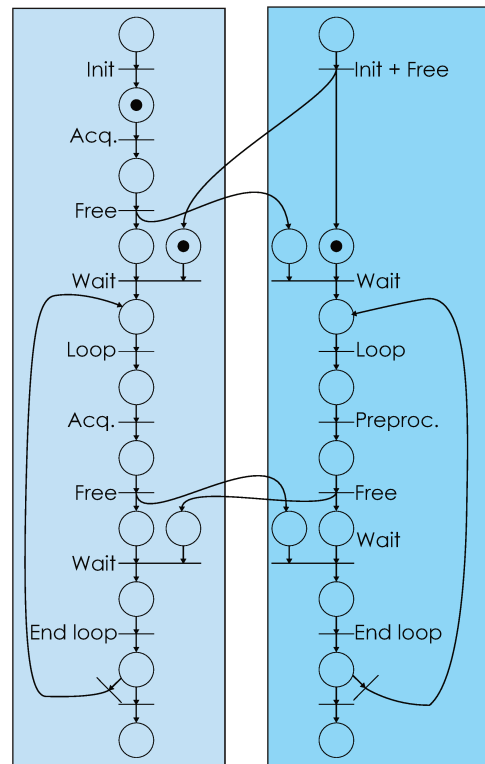
- [65] Yih Han Tan, “Non-TE5 : On the effectiveness of temporal and base layer co-located motion vector prediction candidates,” *ISO/IEC JTC 1/SC 29/WG 11 and Joint Collaborative Team on Video Coding, JCTVC-L0073, Geneva, CH*, January 2013.
- [66] Jun Xu and Ali Tabatabai, “Non-TE5 : Memory reduction for MV data of EL in SHVC,” *ISO/IEC JTC 1/SC 29/WG 11 and Joint Collaborative Team on Video Coding, JCTVC-L0221r2, Geneva, CH*, January 2013.
- [67] Jun Xu and Ali Tabatabai, “SCE5 5.2.1 on Effectiveness of temporal motion vector prediction (TMVP),” *ISO/IEC JTC 1/SC 29/WG 11 and Joint Collaborative Team on Video Coding, JCTVC-M0283, Incheon, KR*, April 2013.
- [68] Kalray, “<http://www.kalrayinc.com/kalray/products/>,” .
- [69] Intel, *Intel Xeon Phi Coprocessor x200 Product Family - Datasheet*, april 2017.
- [70] intel, “<https://software.intel.com/sites/landingpage/intrinsicguide/>,” .
- [71] Kiran Misra, Andrew Segall, Michael Horowitz, Shilin Xu, Arild Fuldseth, and Minhua Zhou, “An Overview of Tiles in HEVC,” *IEEE journal of selected topics in signal processing*, pp. 969–977, december 2013.
- [72] Tektronix Inc., “Mts4ea elementary stream analyser version 7.1.0.261,” .
- [73] G. Tech, Y. Chen, K. Müller, J. R. Ohm, A. Vetro, and Y. K. Wang, “Overview of the multiview and 3d extensions of high efficiency video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 35–49, Jan 2016.
- [74] B. Tajali and H. Roodaki, “Hevc-based view level rate-distortion model for multiview video,” in *2017 Iranian Conference on Electrical Engineering (ICEE)*, May 2017, pp. 2182–2187.
- [75] M. Cordina and C. J. Debono, “A cross-layer mv-hevc depth-texture rate allocation estimation technique in 3gpp lte systems,” in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, July 2017, pp. 19–24.
- [76] P. Bordes, P. Andrivon, X. Li, Y. Ye, and Y. He, “Overview of color gamut scalability,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 7, pp. 1580–1594, July 2017.
- [77] “OV 2082-0 :2016 - SMPTE Overview Document - 12G-SDI Bit-Serial Interfaces ; Overview for the SMPTE ST 2082 Document Suite,” *OV 2082-0 :2016*, pp. 1–10, Sept 2016.
- [78] “ST 259 - SMPTE Standard - For Television ; SDTV1 Digital Signal/Data ; Serial Digital Interface,” *SMPTE ST 259 :2008*, pp. 1–18, Jan 2008.
- [79] “ST 344 :2000 - SMPTE Standard - For Television ; 540 Mb/s Serial Digital Interface,” *SMPTE ST 344 :2000*, pp. 1–5, Sept 2000.
- [80] “ST 292-1 :2012 - SMPTE Standard - 1.5 Gb/s Signal/Data Serial Interface,” *ST 292-1 :2012*, pp. 1–20, Jan 2012.
- [81] “ST 424 :2012 - SMPTE Standard - 3 Gb/s Signal/Data Serial Interface, year=2012,” *SMPTE ST 424 :2012*, pp. 1–10, Oct.
- [82] “ST 425-5 :2014 Am1 :2015 - SMPTE Amendment - Image Format and Ancillary Data Mapping for the Quad Link 3 Gb/s Serial Interface ; Amendment 1,” *Amendment 1:2015 to SMPTE ST 425-5:2014*, pp. 1–5, June 2015.
- [83] “ST 2081-10 :2015 - SMPTE Standard - 2160-Line and 1080-Line Source Image and Ancillary Data Mapping for Single-Link 6GSDI,” *SMPTE ST 2081-10 :2015*, pp. 1–22, March 2015.
- [84] “ST 2081-11 :2016 - SMPTE Standard - 2160-line Source Image and Ancillary Data Mapping for Dual-link 6G-SDI,” *ST 2081-11 :2016*, pp. 1–20, Sept 2016.
- [85] “ST 2082-10 :2015 - SMPTE Standard - 2160-line Source Image and Ancillary Data Mapping for 12G-SDI, year=2015,” *SMPTE ST 2082-10 :2015*, pp. 1–21, March.
- [86] Dektel Digital Video BV, “DTA-2174 Leaflet,” June 2015.
- [87] Deborah McAdams, “Sinclair Demos HDR 4KTV over ATSC 3.0 in Vegas,” <http://www.tvtechnology.com/news/0002/sinclair-demos-hdr-4ktv-over-atsc-30-in-vegas/277546>, december 2015.

- [88] TelecomEngine, “South Korea’s ETRI Recognized for New Core UHD and Next-Gen TV Tech Standards,” <http://www.telecomengine.com/south-koreas-etri-recognized-for-new-core-uhd-and-next-gen-tv-tech-standards/>, october 2016.
- [89] Ronan Parois, Wassim Hamidouche, Elie Mora, Mickaël Raulet, and Olivier Deforges, “Demo : UHD Live video streaming with a real-time scalable HEVC encoder,” *Conference on Design & Architectures for Signal & Image Processing (DASIP)*, october 2016.
- [90] GPAC, “<https://gpac.wp.imt.fr/>,” .

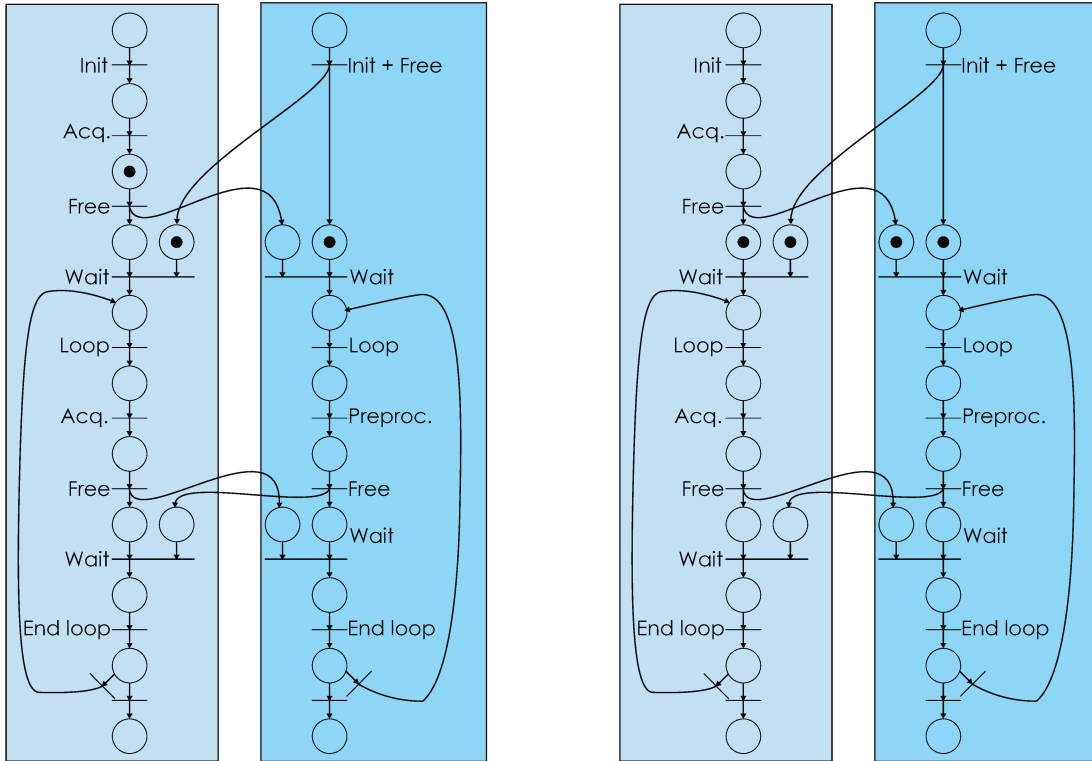
Annexe A : Cheminement du réseau de pétri de 2 étapes du pipeline du SLE.



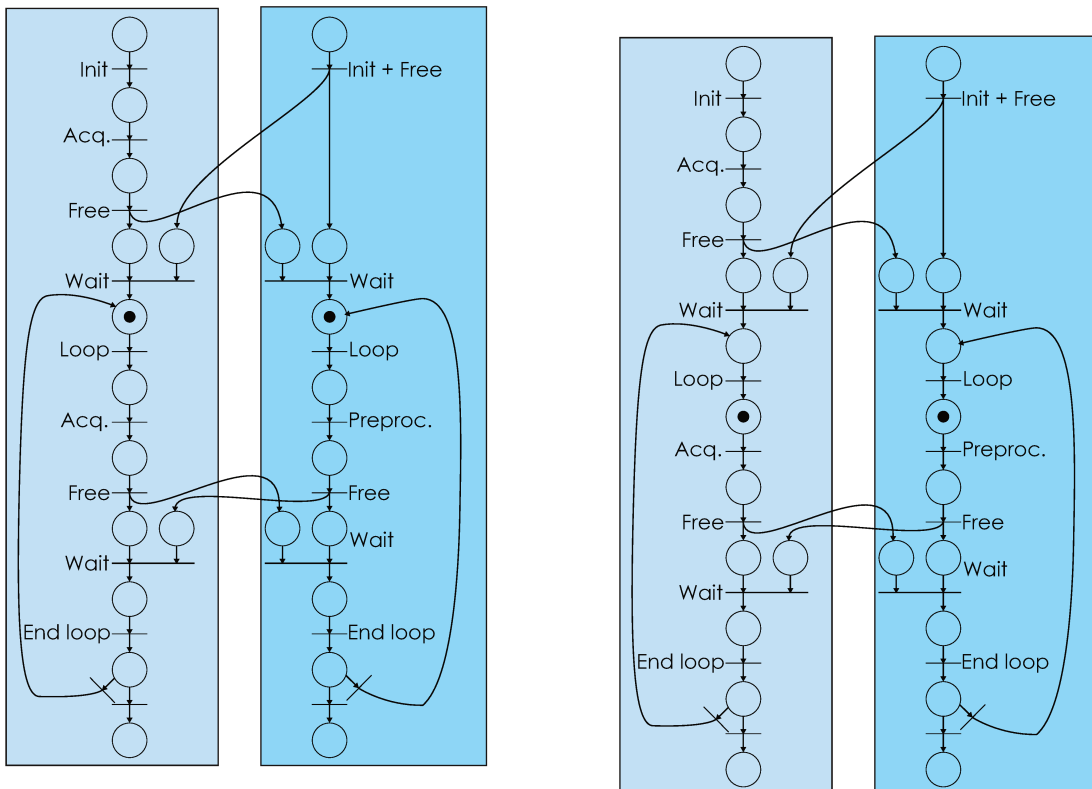
État initial.



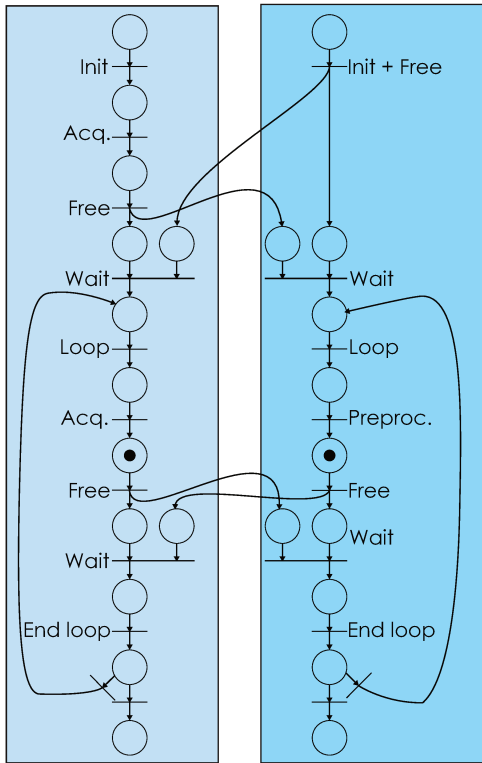
Les deux étapes sont initialisées, l'étape de pré-traitement autorise l'entrée dans la boucle synchronisée.



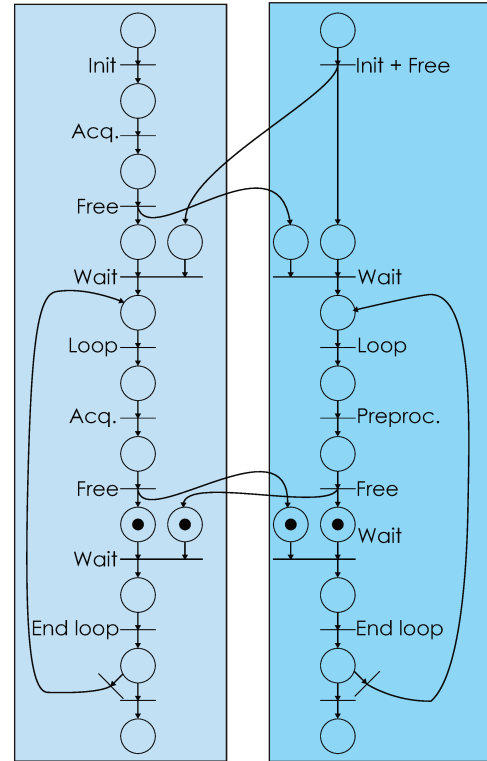
L'acquisition sur la première image est effectuée. Le pré- Après l'acquisition de la première image, les deux étapes sont traitement est en attente. autorisées à entrer dans la boucle synchronisée.



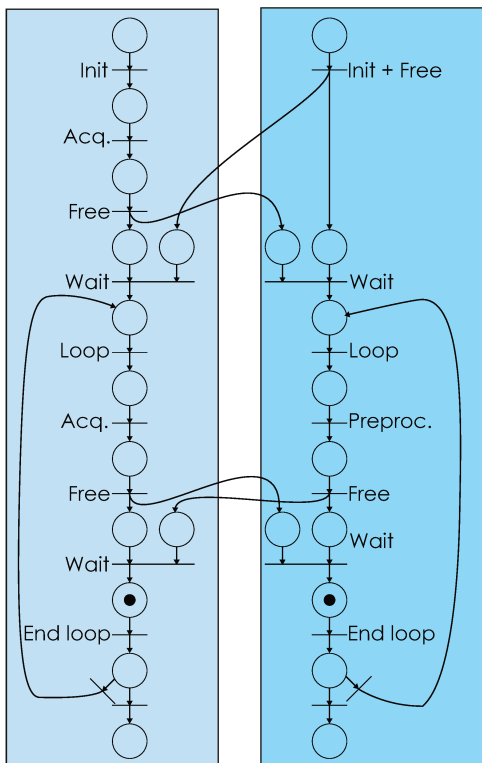
L'étape d'acquisition copie les données de la première image à destination de l'étape de pré-traitement. Les deux étapes entre dans la boucle synchronisée.



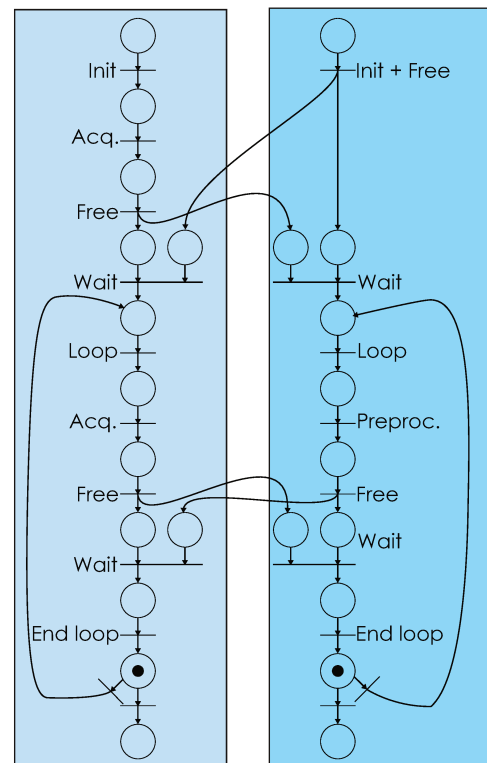
L'acquisition sur la deuxième image et, dans le même temps, le pré-traitement sur la première image sont effectués.



L'étape d'acquisition copie les données de la deuxième image vers l'étape de pré-traitement. L'étape de pré-traitement signale qu'elle est prête à recevoir ces informations.

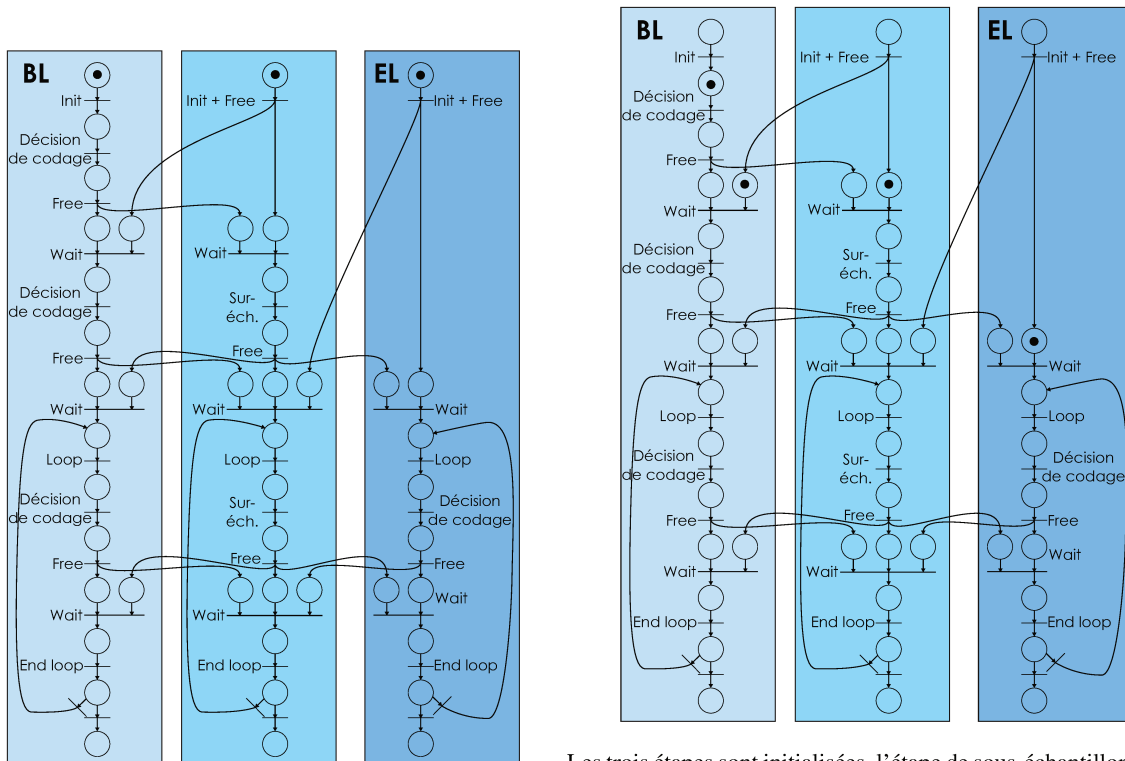


Les deux étapes sont synchrones.



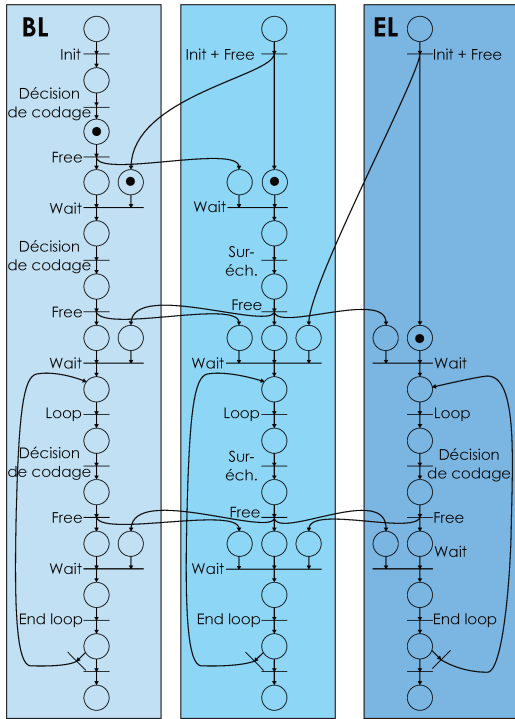
La boucle prend fin et peut plus haut avec les autres images. L'étape d'acquisition a toujours avec une image d'avance.

Annexe B : Cheminement du réseau de pétri de la synchronisation du SHVC-AE au niveau des étapes de décision de codage de la couche de base (d-BL), du sur-échantillonnage, la décision de codage de la couche de rehaussement (d-EL).

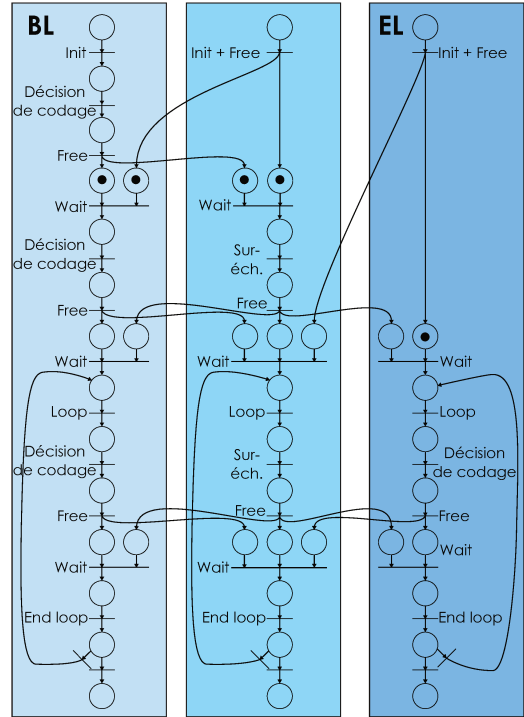


État initial.

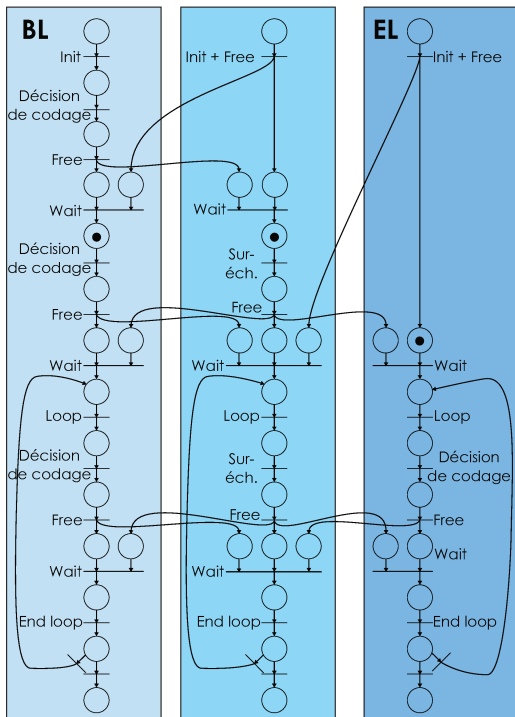
Les trois étapes sont initialisées, l'étape de sous-échantillonnage autorise l'entrée dans la boucle synchronisée de l'étape de d-BL et l'étape de d-EL autorise celle de l'étape de sur-échantillonnage.



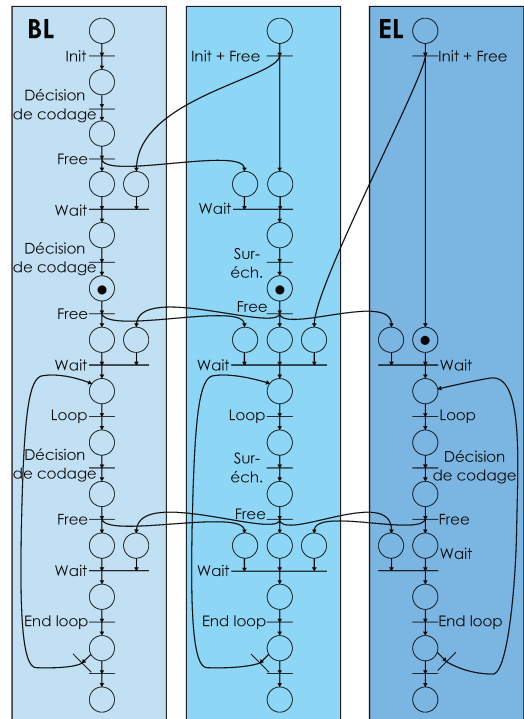
La d-BL est effectuée sur l'image i , le sur-échantillonnage et la d-EL sont en attente.



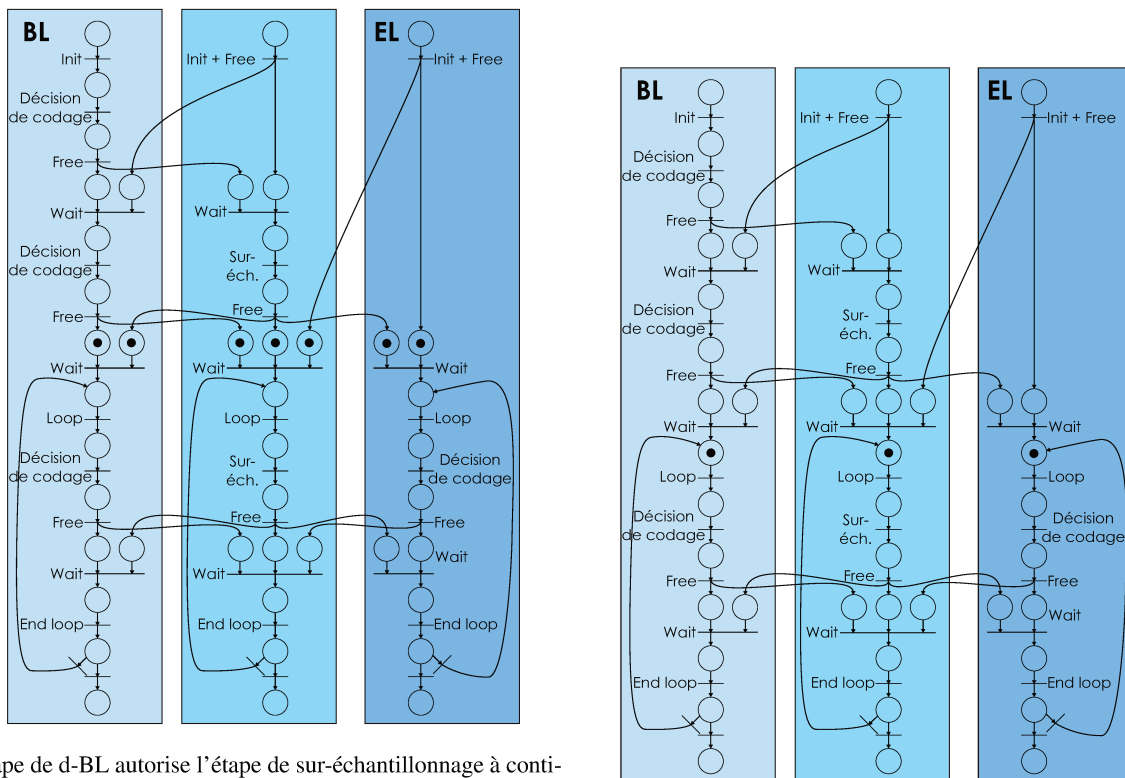
L'étape de d-BL autorise l'étape de sur-échantillonnage à commencer son traitement et lui transmet les informations de codages. L'étape de d-EL est en attente.



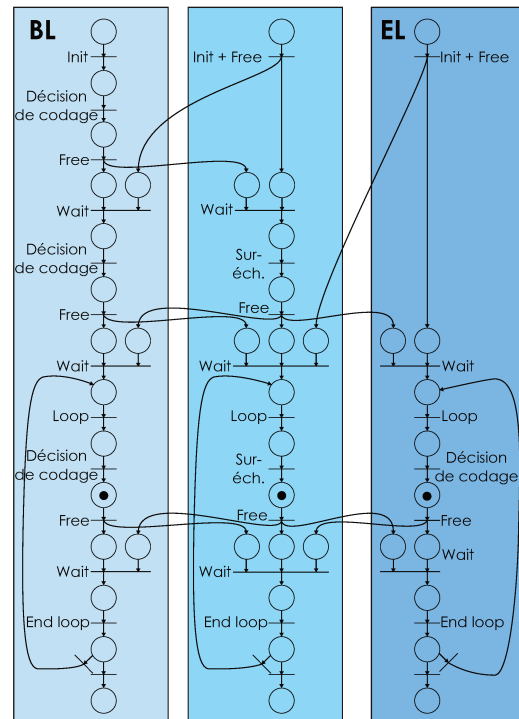
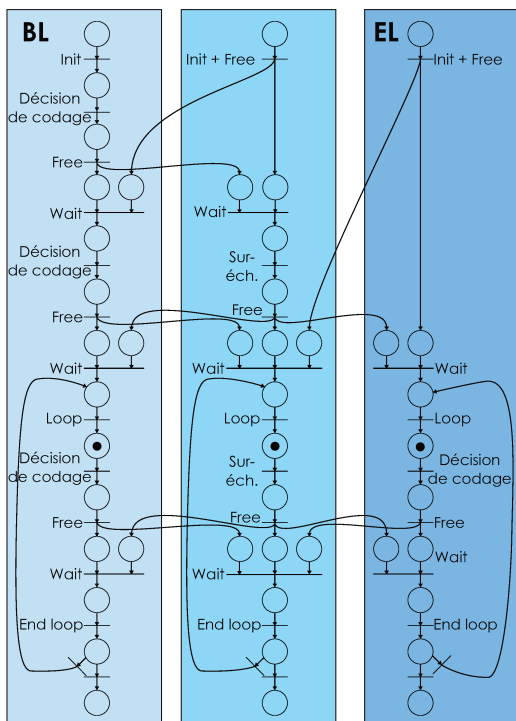
Les étapes de d-BL et de sur-échantillonnage sont synchrones. L'étape de d-EL est en attente.



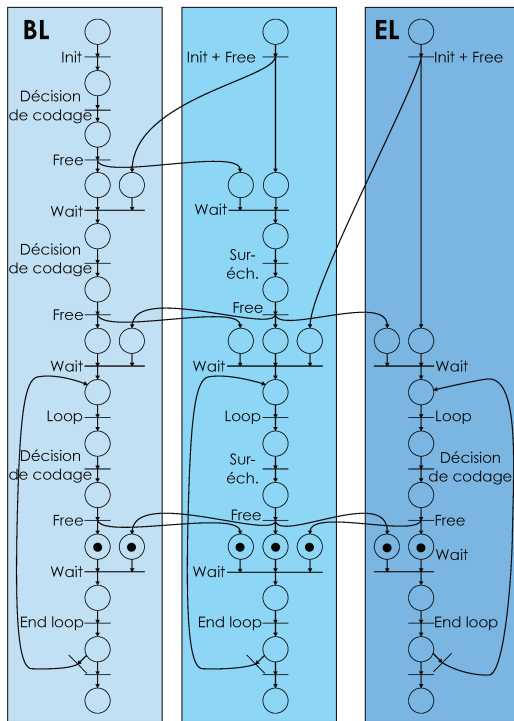
Les opérations de sur-échantillonnage et de décision de codage sont effectuées respectivement sur les images i et $i+1$. L'étape de d-EL est en attente.



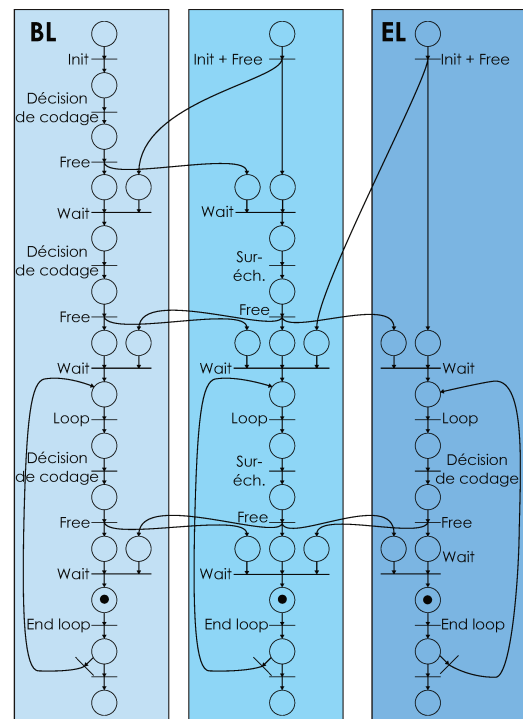
L'étape de d-BL autorise l'étape de sur-échantillonnage à continuer son traitement et lui transmet les informations de codages de l'image $i+1$. L'étape de sur-échantillonnage indique à l'étape Toutes les étapes sont synchrones et entre dans la boucle de traitement de d-BL qu'elle peut recevoir les informations de codage de l'image $i+1$ et autorise l'étape de d-EL à démarrer son traitement et lui transmet les informations de codages de l'image i .



Les opérations de d-EL, de sur-échantillonnage et de décision de codage sont effectuées respectivement sur les images i , $i+1$ et $i+2$.



L'étape de d-BL autorise l'étape de sur-échantillonnage à continuer son traitement et lui transmet les informations de codages de l'image $i+2$. L'étape de sur-échantillonnage indique à l'étape de d-BL qu'elle peut recevoir les informations de codage de l'image $i+2$ et autorise l'étape de d-EL à démarrer son traitement et lui transmet les informations de codages de l'image $i+1$.



Toutes les étapes sont synchrones.

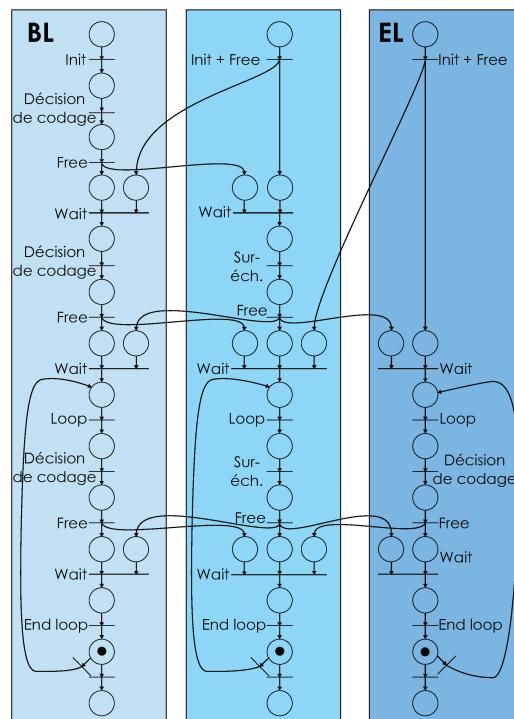


FIGURE 7.10 – La boucle prend fin et peut plus haut avec les autres images. L'étape de d-BL est en avance d'une image sur l'étape de sur-échantillonnage elle-même en avance d'une image sur l'étape de d-EL.

Annexe D : Optimisation bas niveau sur les filtres de sous-échantillonnage et de sur-échantillonnage.

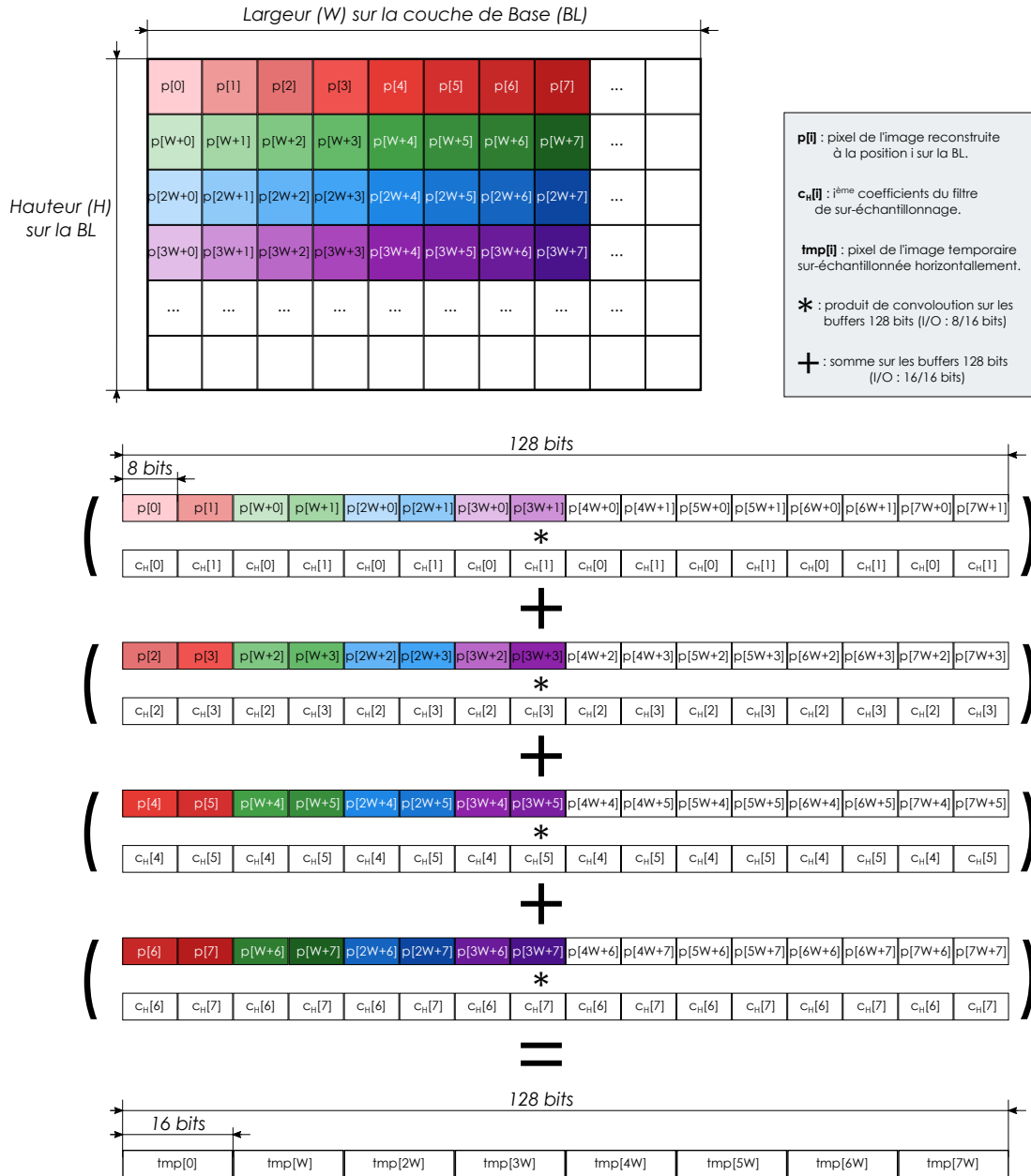


FIGURE 7.11 – Schéma du produit de convolution réalisé sur les filtres horizontaux avec les instructions intrinsèques sur des buffers 128 bits pour la composante Y.

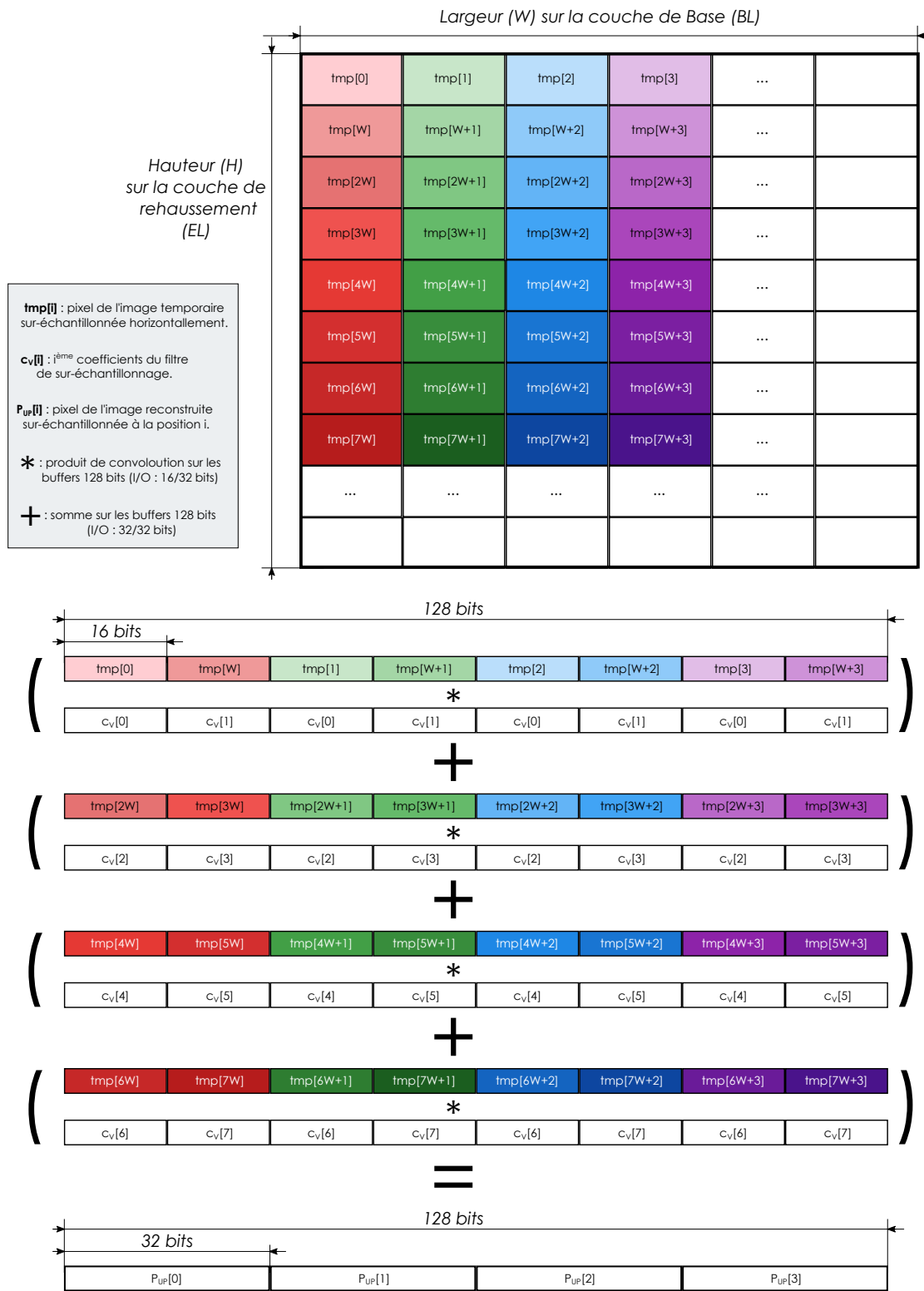


FIGURE 7.12 – Schéma du produit de convolution réalisé sur les filtres verticaux avec les instructions intrinsèques sur des buffers 128 bits pour la composante Y.

Annexe E : Différentes vues de la réponse impulsionnelle des filtres de sous-échantillonnage considéré.

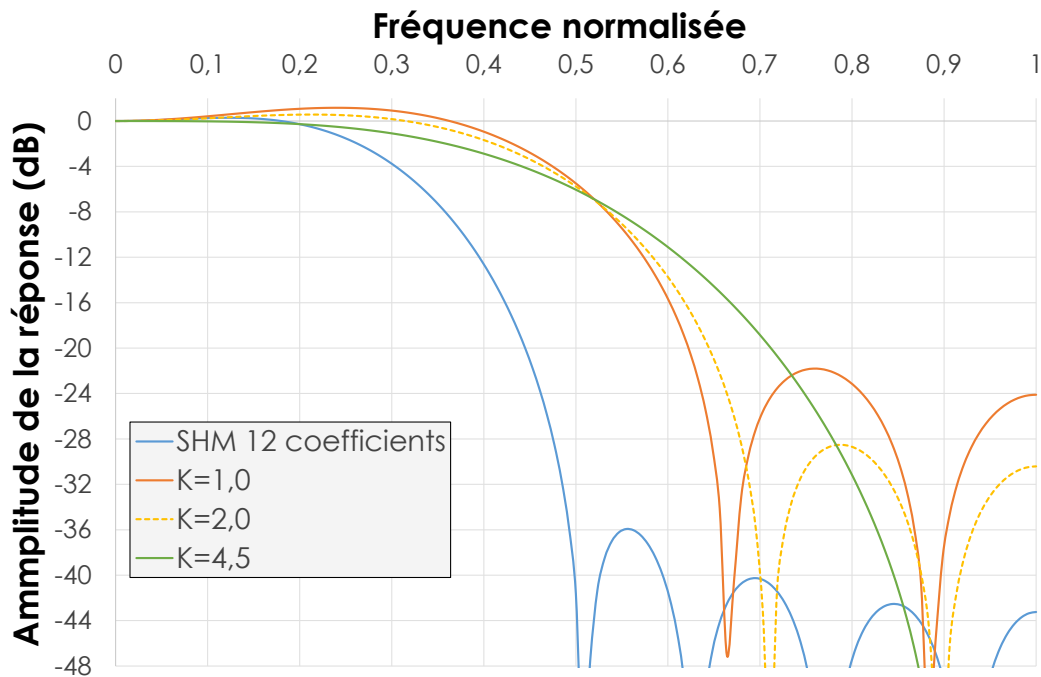


FIGURE 7.13 – Vue globale des réponses impulsionnelles des filtres de sous-échantillonnage considérés. SHM est le filtre proposé dans le codeur de référence sur 12 coefficients, $K=n$ sont les filtres proposés sur 8 coefficients avec le paramètre K égale à n . La fréquence de coupure est atteinte plus tard pour les filtres proposés mais cela permet de conserver davantage d'information issues des hautes fréquences.

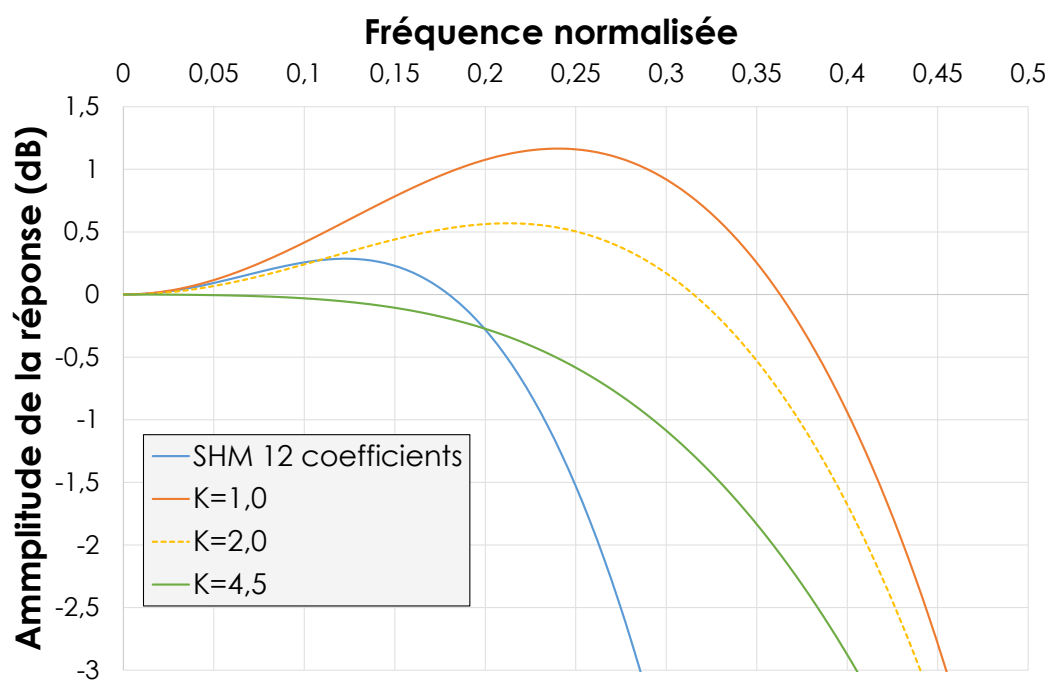


FIGURE 7.14 – Vue de la courbe sur les basses fréquences. La distorsion est plus élevée pour une valeur de K faible.

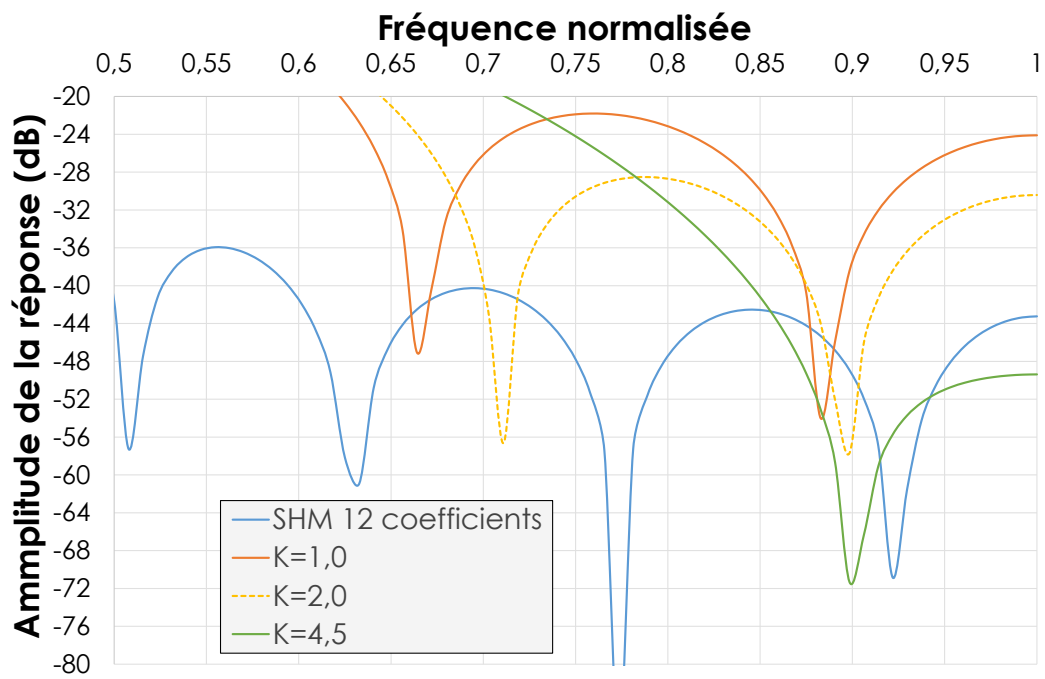


FIGURE 7.15 – Vue de la courbe sur les hautes fréquences. Les lobes secondaires sont plus grands pour les filtres à 8 coefficients. Ils sont également plus importants pour des valeurs de K faible.

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse:

Codeur vidéo scalable haute-fidélité SHVC modulable et parallèle

Nom Prénom de l'auteur : PAROIS RONAN

Membres du jury :

- Monsieur MASMOUDI Nouri
- Monsieur DEFORGES Olivier
- Monsieur HAMIDOUCHE Wassim
- Monsieur AKIL Mohamed
- Monsieur MATTAVELLI Marco
- Madame MOKRAOUI Anissa
- Monsieur RAULET Mickaël

Président du jury : *Mme MOKRAOUI*

Date de la soutenance : 27 Février 2018

Reproduction de la these soutenue

- Thèse pouvant être reproduite en l'état
 Thèse pouvant être reproduite après corrections suggérées

Fait à Rennes, le 27 Février 2018

Signature du président de jury

Le Directeur,

M'hamed DRISSI



Après l'entrée dans l'ère du numérique, la consommation vidéo a évolué définissant de nouvelles tendances. Les contenus vidéo sont désormais accessibles sur de nombreuses plateformes (télévision, ordinateur, tablette, smartphone...) et par de nombreux moyens, comme les réseaux mobiles, les réseaux satellites, les réseaux terrestres, Internet ou le stockage Blu-ray par exemple. Parallèlement, l'expérience utilisateur s'améliore grâce à la définition de nouveaux formats comme l'Ultra Haute Définition (UHD), le « High Dynamic Range » (HDR) ou le « High Frame Rate » (HFR). Ces formats considèrent une augmentation respectivement de la résolution, de la dynamique des couleurs et de la fréquence d'image. Les nouvelles tendances de consommation et les améliorations des formats imposent de nouvelles contraintes auxquelles doivent répondre les codeurs vidéo actuels et futurs.

Dans ce contexte, nous proposons une solution de codage vidéo permettant de répondre à des contraintes de codage multi-formats, multi-destinations, rapide et efficace en termes de compression. Cette solution s'appuie sur l'extension Scalable du standard de compression vidéo « High Efficiency Video Coding » (HEVC) définie en fin d'année 2014, aussi appelée SHVC. Elle permet de réaliser des codages scalables en produisant un unique bitstream à partir d'un codage sur plusieurs couches construites à partir d'une même vidéo à différentes échelles de résolutions, fréquences, niveaux de distorsion, profondeurs des pixels ou espaces de couleur. Le codage SHVC améliore l'efficacité du codage HEVC grâce à une prédiction inter-couches qui consistent à employer les informations de codage issues des couches les plus basses. La solution proposée dans cette thèse s'appuie sur un codeur HEVC professionnel développé par la société Ateame qui intègre plusieurs niveaux de parallélisme (inter-images, intra-images, inter-blocs et inter-opérations) grâce à une architecture en pipeline. Deux instances parallèles de ce codeur sont synchronisées via un décalage inter-pipeline afin de réaliser une prédiction inter-couches. Des compromis entre complexité et efficacité de codage sont effectués au sein de cette prédiction au niveau des types d'image et des outils de prédiction. Dans un cadre de diffusion, par exemple, la prédiction inter-couches est effectuée sur les textures pour une image sur deux. A qualité constante, ceci permet d'économiser 18.5% du débit pour une perte de seulement 2% de la vitesse par rapport à un codage HEVC. L'architecture employée permet alors de réaliser tous les types de scalabilité supportés par l'extension SHVC.

De plus, pour une scalabilité en résolution, nous proposons un filtre de sous-échantillonnage, effectué sur la couche de base, qui optimise le coût en débit global. Nous proposons des modes de qualité intégrant plusieurs niveaux de parallélisme et optimisations à bas niveau qui permettent de réaliser des codages en temps-réel sur des formats UHD. La solution proposée a été intégrée dans une chaîne de diffusion vidéo temps-réel et montrée lors de plusieurs salons, conférences et meetings ATSC 3.0.

After entering the digital era, video consumption evolved and defined new trends. Video contents can now be accessed with many platforms (television, computer, tablet, smartphones...) and from many medias such as mobile network or satellite network or terrestrial network or Internet or local storage on Blu-ray disc for instance. In the meantime, users experience improves thanks to new video format such as Ultra High Definition (UHD) or High Dynamic Range (HDR) or High Frame Rate (HFR). These formats respectively enhance quality through resolution, dynamic range and frequency. New consumption trends and new video formats define new constraints that have to be resolved by current and future video encoders.

In this context, we propose a video coding solution able to answer constraints such as multi-formats coding, multidestinatons coding, coding speed and coding efficiency in terms of video compression. This solution relies on the scalable extension of the standard « High Efficiency Video Coding » (HEVC) defined in 2014 also called SHVC. This extension enables scalable video coding by producing a single bitstream on several layers built from a common video at different scales of resolution, frequency, quality, bit depth per pixel or even colour gamut. SHVC coding enhance HEVC coding thanks to an inter-layer prediction that use coding information from lower layers. In this PhD thesis, the proposed solution is based on a professional video encoder, developed by Ateame company, able to perform parallelism on several levels (inter-frames, intraframes, inter-blocks, inter-operations) thanks to a pipelined architecture. Two instances of this encoder run in parallel and are synchronised at pipeline level to enable inter-layer predictions. Some trade-off between complexity and coding efficiency are proposed on inter-layer prediction at slice and prediction tools levels. For instance, in a broadcast configuration, inter-layer prediction is processed on reconstructed pictures only for half the frames of the bitstream. In a constant quality configuration, it enables to save 18.5% of the coding bitrate for only 2% loss in terms of coding speed compared to equivalent HEVC coding. The proposed architecture is also able to perform all kinds of scalability supported in the SHVC extension. Moreover, in spatial scalability, we propose a down-sampling filter processed on the base layer that optimized global coding bitrate. We propose several quality modes with parallelism on several levels and low-level optimization that enable real-time video coding on UHD sequences. The proposed solution was integrated in a video broadcast chain and showed in several professional shows, conferences and at ATSC 3.0 meetings.