Étude et implémentation d'une architecture temps réel pour l'optimisation de la compression H.264/AVC de vidéos SD/HD

Eloïse Vidal

Université de Valenciennes et du Hainaut-Cambrésis

Thèse de doctorat

Pour obtenir le grade de Docteur de l'Université de

VALENCIENNES ET DU HAINAUT-CAMBRESIS

Discipline : Électronique, optronique et systèmes

Spécialité : Électronique

Présentée et soutenue par Eloïse VIDAL le 15 avril 2014

Président de jury : Patrick Le Callet
Soutenance à huis clos

Ecole doctorale : Sciences Pour l'Ingénieur (SPI)

Laboratoire : Institut d'Electronique, de Micro-Electronique et de Nanotechnologie/Département d'Opto-Acousto-Electronique (IEMN/DOAE)

Etude et implémentation d'une architecture temps réel pour l'optimisation de la compression H.264/AVC de vidéos SD/HD

JURY

Rapporteurs

Marco Cagnazzo	Maître de Conférences HDR, TELECOM-ParisTech Département TSI, Paris
Serge Weber	Professeur des Universités, Institut Jean Lamour UMR CNRS 7198, Nancy

Examinateurs

Patrick Le Callet	Professeur des Universités, Institut de Recherche IRCCyN UMR CNRS 6597, Nantes
Joël Cambonie	Responsable des applications image et vidéo, Kalray, Montbonnot
Christine Guillemot	Directrice de recherche, INRIA, RENNES
Thierry Hauser	Responsable du département Hardware, Digigram, Montbonnot

Directeurs de thèse

Patrick Corlay	Maître de Conférences HDR, IEMN OAE UMR CNRS 8520, Valenciennes
François-Xavier Coudoux	Professeur des Universités, IEMN OAE UMR CNRS 8520, Valenciennes

Copie à l'attention de la bibliothèque universitaire de Valenciennes
Version 25/04/2014

Remerciements

Ces travaux de thèse CIFRE ont été réalisés en collaboration avec le département OAE de l'IEMN (UMR8520) de l'Université de Valenciennes et du Hainaut Cambrésis et l'entreprise Digigram. Je tiens à remercier mes encadrants académiques et industriels, qui par leur implication ont permis d'arriver à l'aboutissement des travaux que vous trouverez ici, en répondant aux exigences et contraintes des deux partis.

Je remercie donc chaleureusement mes directeurs de thèse François-Xavier Coudoux et Patrick Corlay pour leur encadrement sans faille malgré les 700 kilomètres séparant Grenoble de Valenciennes.

Je remercie également mes encadrants industriels, tout d'abord Patrice Manoutsis pour avoir initié cette thèse, ces travaux n'auraient pas eu lieu sans l'intérêt et la confiance qu'il m'a accordé. Puis Thierry Hauser pour l'attention et la rigueur qui ont permis de réaliser cette thèse dans les temps impartis malgré les aléas du rachat ; ainsi que Nicolas pour sa participation précieuse aux travaux en fin de thèse.

Je remercie également l'Association Nationale de Recherche et Technologie pour le financement de ces travaux.

Je remercie Serge Weber et Marco Cagnazzo pour avoir accepté de rapporter mes travaux, ainsi que Patrick Le Callet et Joël Cambonie pour leur participation au jury de soutenance. Je remercie tout spécialement Christine Guillemot pour la qualité de nos échanges et des travaux que nous avons menés ensemble.

Je remercie mes collègues de Digigram et spécialement ceux de la première heure Loïc, Laurent et Vincent pour avoir écouté et supporté avec le sourire les aléas d'humeur, doutes et remises en question en tout genre pendant les quatre dernières années.

Je remercie également mes collègues doctorants et docteurs de l'IEMN, particulièrement Imade et Christophe pour le soutien durant les six premiers mois de bibliographie, puis Othmane pour le partage des différentes étapes de la thèse.

Merci également aux trois stagiaires qui ont travaillées avec moi, Elodie, Noémie et Marine pour leur contribution à la qualité et la quantité des expériences menées et qui m'ont permis de m'initier à l'encadrement.

Je remercie Margaret Pinson pour les nombreux échanges sur la mesure de qualité vidéo qui ont beaucoup apporté à nos travaux.

Je remercie également le service de reprographie de l'université de Valenciennes pour leur efficacité lors du tirage du manuscrit final.

Enfin je remercie bien sûr mes proches, famille et amis, pour le soutien et les encouragements durant ces années intenses et les périodes difficiles ; ma mère pour la relecture du manuscrit, ma belle-mère et particulièrement mon père qui aurait été fier de cet aboutissement.

Résumé

La vidéo sur IP a connu un essor rapide ces dernières années allant de la diffusion télévisuelle en haute qualité via des réseaux dédiés à la diffusion sur internet de contenus vidéo grand public. L'optimisation de l'encodage vidéo H.264/AVC permet aux différents acteurs du marché de se différencier en proposant des solutions pour réduire le débit nécessaire à la représentation d'un flux vidéo ainsi que pour améliorer la qualité perçue par les utilisateurs. C'est dans ce contexte de vidéo professionnelle en haute qualité que s'inscrivent ces travaux de thèse CIFRE réalisés au sein de l'entreprise Digigram, proposant des encodeurs vidéo temps réel pour des diffusions professionnelles en direct. Nous proposons deux solutions de prétraitement pour répondre aux problématiques du secteur de la distribution vidéo. Les deux solutions considèrent les caractéristiques du système visuel humain en exploitant un modèle de JND (Just Noticeable Distortion) définissant des seuils de perception en fonction d'une analyse du contenu des séquences vidéo à encoder. La première solution utilise un préfiltre adaptatif indépendant de l'encodeur, contrôlé par un modèle JND afin d'éliminer le contenu perceptuellement non pertinent et ainsi réduire le débit sans altérer la qualité ressentie. Une analyse approfondie de plusieurs filtres de la littérature, dont le filtre AWA (Adaptive Weighted Averaging) et le filtre bilatéral, nous a également amené à définir deux nouveaux filtres à support étendu qui permettent d'exploiter au mieux les corrélations dans les images haute définition. A l'aide de tests subjectifs, nous montrons que les préfiltres perceptuels proposés permettent en moyenne de diminuer le débit en sortie du codeur d'environ 20% pour une qualité constante en encodage VBR (débit variable) Intra et Inter-image. Finalement, une deuxième solution s'attache à améliorer la qualité perçue dans un contexte d'encodage CBR (débit constant) en intégrant un modèle JND dans l'une des implémentations de la norme H.264/AVC la plus reconnue, le codec x264. Une quantification adaptative perceptuelle est ainsi proposée permettant d'améliorer les performances du codec x264 en améliorant le codage de l'information de contour à moyen et bas débits en encodage intra et inter-image.

Mots clés : Prétraitement, JND (Just Noticeable Distortion), AWA (Adaptive Weighted Averaging), Bilatéral, H.264/AVC, x264, quantification adaptative, traitement perceptuel.

Abstract

The use of digital video over IP has increased exponentially over the last years, due to the development of high-speed networks dedicated to high quality TV transmission as well as the wide development of the non-professional video webcast. Optimization of the H.264/AVC encoding process allows manufacturers to offer differentiating encoding solutions, by reducing the bandwidth necessary for transmitting a video sequence at a given quality level, or improving the quality perceived by final users at a fixed bit rate. This thesis was carried out at the company Digigram in a context of professional high quality video. We propose two solutions of preprocessing which consider the characteristics of the human visual system by exploiting a JND profile (Just Noticeable Distortion). A JND model defines perceptual thresholds, below which a distortion cannot be seen, according to the video content. The first solution proposes an adaptive pre-filter independent to the encoder, controlled by a JND profile to reduce the perceptually non-relevant content and so reduce the bitrate while maintaining the perceived quality. By analyzing the state-of-the-art literature, the AWA (Adaptive Weighted Averaging) and Bilateral filters have been selected. Then we define two new filters using a large convolution mask, which enable to better exploit correlations in high-definition video contents. Through subjective tests, we show that the proposed perceptual prefilters give an average bitrate reduction of 20% for the same visual quality in VBR (Variable Bitrate) H.264/AVC Intra and Inter encoding. Finally, the second solution enables to improve the perceived quality in CBR (Constant Bitrate) encoding, by integrating the JND profile into the x264 codec, one of the best implementation of the H.264/AVC standard. Thus, we propose a perceptual adaptive quantization which enhances the x264 performance by improving edge information coding in low and middle bitrate applications.

Key Words: Preprocessing, JND (Just Noticeable Distortion), AWA (Adaptive Weighted Averaging), Bilateral, H.264/AVC, x264, adaptive quantization, perceptual processing

Table des matières

Liste des Figures

Liste des Tableaux

Introduction générale

Le transport de la vidéo en haute résolution s'est démocratisé avec l'arrivée de la TNT (Télévision Numérique Terrestre) et prochainement la TNT2, ainsi qu'avec l'augmentation des débits disponibles sur les réseaux de communication IP. Les algorithmes d'encodage vidéo, utilisés aussi bien dans le domaine grand public que par les professionnels, cherchent à optimiser la qualité perçue par les utilisateurs pour un débit donné et à réduire le débit nécessaire à la représentation d'un flux pour une qualité cible. L'encodage en temps réel ajoute en plus une contrainte sur la complexité des algorithmes employés en imposant une faible latence et des ressources nécessairement limitées. La norme d'encodage H.264/AVC est aujourd'hui la plus largement répandue que ce soit pour le stockage, la transmission temps réel ou le streaming. Son successeur HEVC fait l'objet de développements actifs dans le monde industriel. Il a été développé dans le but de réduire par deux le débit nécessaire pour l'encodage HD ainsi que pour gérer les nouvelles haute résolutions 2K et 4K. La recherche du meilleur compromis entre la réduction de débit et la conservation de la meilleure qualité possible poussent les industriels à se différencier par le développement de solutions propriétaires, qui respectent la conformité du flux vidéo à la norme d'encodage. Une première solution est l'optimisation des parties non normatives de l'encodage vidéo, c'est à dire l'allocation binaire via le contrôle de débit et la sélection des outils de codage les plus performants sous contrainte de débit (Rate Distorsion Optimization). Une deuxième solution est l'utilisation de fonctions de prétraitement du contenu pour simplifier le processus d'encodage. Enfin, des solutions de post-traitement, placés au décodeur, permettent pour réduire les artefacts de codage.

Les travaux de cette thèse proposent deux solutions de prétraitement pour l'encodage H.264/AVC dans un contexte de vidéo HD. La première adresse la problématique de la réduction de débit à qualité constante en encodage VBR (Variable BitRate, sans contrôle de débit), à l'aide de préfiltres perceptuels indépendant du codeur. La deuxième propose une quantification adaptative perceptuelle intégrée au codec x264, pour améliorer, à même débit, la qualité perçue en encodage CBR (Constant BitRate). Dans un contexte de vidéo professionnelle, nous avons porté une attention particulière à proposer des prétraitements donnant la meilleure qualité perçue possible. Dans cette optique, la mesure de qualité a été une problématique abordée tout au long des travaux à travers l'utilisation de métriques objectives, mesurant la qualité globale des images et séquences ou simplement un artefact particulier. De plus, la mesure de qualité subjective a été également utilisée par la mise en place de tests perceptuels faisant intervenir des observateurs auxquels il a été demandé de noter la qualité des séquences qui leur ont été présentées.

Dans le premier chapitre nous introduisons les principes de l'encodage vidéo, puis la norme H.264/AVC et les innovations proposées par la norme HEVC. Nous présentons également les principes de fonctionnement des parties non-normatives d'un encodeur vidéo, pour enfin détailler le fonctionnement de l'une des implémentations les plus largement utilisées, le codec x264.

Des efforts constants ont été menés pour modéliser les caractéristiques du système visuel humain et ainsi guider le traitement et la compression de données vidéo. Parmi ces modèles, le JND (Just Noticeable Distortion) défini des seuils de perception de dégradation en fonction du contenu local de l'image. Nous proposons dans le chapitre 2 un préfiltre basé sur un filtre passe-bas de la littérature, le filtre AWA (Adaptive Weighted Averaging). Le prétraitement est guidé par un modèle JND pour réduire les détails peu perceptibles d'une séquence vidéo et ainsi de réduire le débit nécessaire à la représentation des séquences sans altérer la qualité perçue.

Ce prétraitement étant réalisé en amont de l'encodeur et n'exploitant pas les données de l'encodage, il peut être placé indifféremment devant n'importe quel encodeur. Dans le chapitre 3, nous proposons d'étudier spécifiquement le cas du traitement de contenu HD en réduction de bruit et en prétraitement perceptuel. Pour cela nous explorons l'intérêt d'étendre la taille du support de filtrage ainsi que l'utilisation d'un autre filtre reconnu de la littérature, le filtre Bilatéral. Nous serons amenés à définir deux nouveaux filtres, le BilAWA et le Bilatéral seuillé. Nous verrons que ces deux filtres apportent de meilleures performances que les filtres AWA et Bilatéral dans les deux applications étudiées.

Pour répondre au besoin d'optimisation de la qualité à débit constant, nous proposons au chapitre 4 de contrôler le processus d'allocation binaire par une quantification adaptative perceptuelle qui sélectionne le paramètre

de quantification par macrobloc en fonction d'un critère de JND. La solution proposée est intégrée dans le codec x264. En apportant une attention particulière à la quantification des contours d'une scène, la solution proposée s'attache à réduire l'un des artefacts typiques de l'encodage vidéo, l'effet de Ringing (apparition d'activité parasite (guige) aux contours d'une scène, Cf. paragraphe 4. 4. 2. 3.).

Contexte de l'étude

Digigram est une entreprise française située dans la région grenobloise fondée en 1985 et spécialisée dans les équipements audio professionnels. La réputation internationale de Digigram repose sur ses cartes de numérisation audio qui ont révolutionnées le monde de la radio à la fin des années 80. Digigram a proposé en 1989 la première carte son audio professionnelle utilisant un algorithme de compression, puis en 1993 le premier codec MPEG-1 layer 1 et 2 (ISO/IEC 11172) embarqué associé par la suite à un mixeur audio opérant sur le flux compressé. En 2001, Digigram dépose un brevet sur une technologie appelée Ethersound [1] de transport audio sur Ethernet aujourd'hui encore très présente sur scène, notamment sur des équipements de marques comme Yamaha. Depuis 2004, Digigram propose des solutions d'encodage audio avec transport sur IP temps réel, et s'ouvre en 2010 au monde de la vidéo en acquérant l'entreprise Ecrin Video&Broadcast, dans laquelle j'ai débuté mon doctorat, offrant des solutions d'encodage vidéo H.264/AVC avec transport sur IP.

Le secteur de la vidéo sur IP se découpe en deux marchés ayant des exigences différentes en termes de qualité/débit, latence et protocole de transmission : Le marché de la distribution et de la contribution. Ces deux marchés peuvent être séparés en plusieurs secteurs présentés de manière non exhaustive à la Figure 0.1.

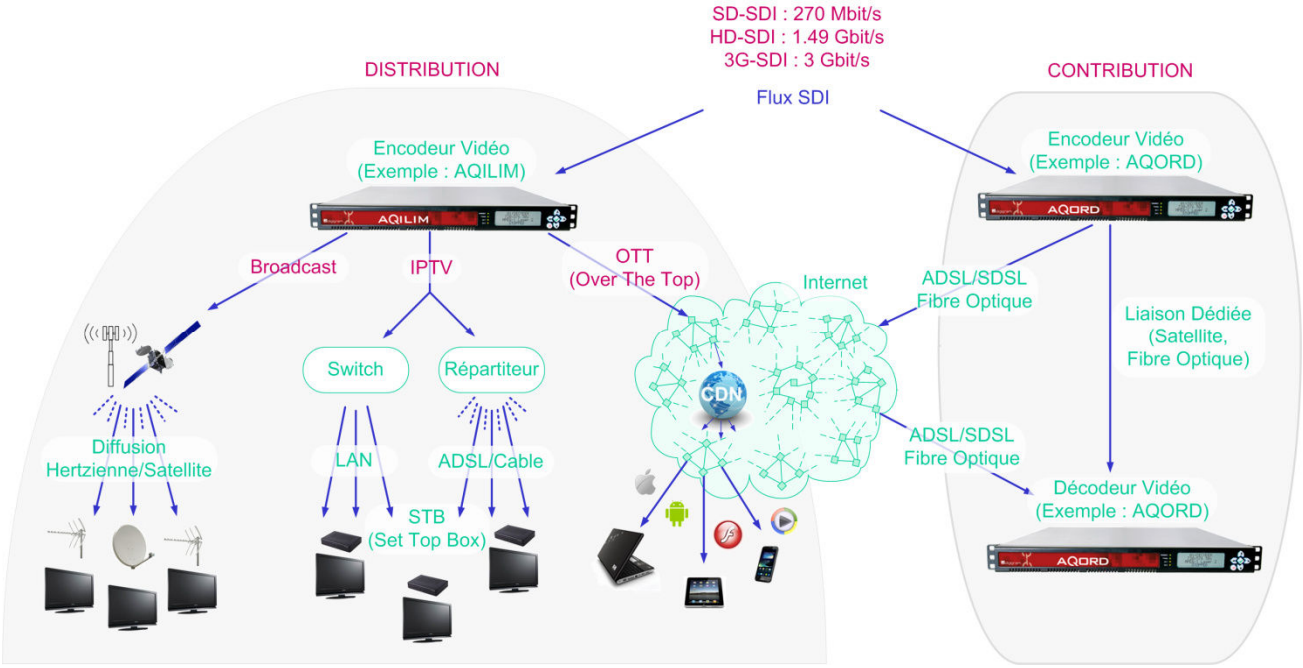


Figure 0.1. Illustrations des différents marchés de la vidéo sur IP

Le marché de la contribution concerne une liaison point à point temps réel entre un encodeur et un décodeur. Un exemple typique d'une liaison de contribution est la captation d'un événement encodé sur place et transmis en temps réel au studio pour être intégré à un programme télévisuel. Le marché de la contribution est caractérisé par le besoin d'une faible latence entre l'encodage et le décodage. Les exigences en termes de qualité sont variables et dépendent de la bande passante disponible. Les formats vidéo principaux utilisés dans ce marché sont : les formats HD 1920x1080, 1280x720 généralement à 50 images entrelacées par seconde en Europe et les formats SD PAL (720x576, 50i) et NTSC (720x480, 59.94i). Le conteneur audio et vidéo MPEG-TS est le plus largement utilisé.

Le marché de la distribution quant à lui concerne une liaison entre un encodeur et une multitude d'utilisateurs visionnant le flux sur différents types de terminaux intégrant des décodeurs (téléviseurs, ordinateurs, smartphone,

tablettes). La diffusion Broadcast utilise la transmission satellite, câblée ou hertzienne pour acheminer notamment la TNT vers les téléspectateurs en haute qualité.

Le secteur de l'IPTV repose dans de nombreux cas sur un réseau de transmission dédié ou du moins contrôlé, dont l'exemple le plus parlant est l'acheminement de la télévision sur IP par les FAI (Fournisseur d'Accès Internet), ou la diffusion de contenu via un LAN (Local Area Network) dans une chaîne d'hôtellerie.

Lorsque la transmission n'utilise pas de liaison dédiée et passe par internet, on parle du marché de l'OTT (Over The Top). Généralement, le fournisseur de contenu encode le flux audiovisuel est le transmet à un CDN (Content Delivery Network) qui dispose de serveurs de diffusion supportant la connexion d'une multitude d'utilisateurs qui accèdent au contenu en streaming. Les CDN les plus utilisés en France sont Akamai, Amazone et Level 3.

La latence du système d'encodage vidéo importe peu dans le marché de la Diffusion car les formats de transmission peuvent imposer à eux seuls plusieurs secondes de latence comme le streaming adaptatif (HLS, smooth streaming, Dash). En revanche la qualité du flux vidéo doit être optimale pour un débit donné, les débits utilisés sont très variables allant de la très haute qualité aux faibles débits. Les industriels se différencient sur ce secteur par la souplesse de leurs encodeurs en termes de formats supportés (résolution, conteneur) et par l'effort apporté à l'optimisation de la qualité via notamment des prétraitements.

Digigram propose deux gammes de produits destinées à répondre aux exigences des marchés de la Contribution et de la Distribution. L'encodeur AQORD (Figure 0.2 (a)) basé sur un ASSP (Application Specific Standard Product) réalisant l'encodage/transcodage audio et vidéo ainsi que l'encapsulation MPEG-TS. Cet encodage appelé Hardware a l'avantage de proposer un encodage à faible latence (500 ms pour l'AQORD) indispensables pour la transmission d'un direct. La deuxième gamme d'encodeur AQILIM (Figure 0.2 (b)) est basée sur le codec logiciel développé par Rovi (Mainconcept) [2], il permet d'offrir la souplesse d'un encodage logiciel pour s'adapter à l'évolution rapide des formats du marché de la diffusion OTT.

(a) (b)

Figure 0.2. Produits vidéo
(a) encodeur Broadcast AQORD – (b) encodeur Diffusion/IPTV AQILIM

Afin d'apporter un élément concurrentiel à ces encodeurs de la gamme AQILIM, Digigram a engagé des travaux sur le filtrage perceptuel pour proposer la meilleure qualité d'encodage possible à ses clients du marché de la distribution. C'est dans ce contexte que se place notre étude.

Le marché de la Diffusion ainsi que les encodeurs AQILIM nous imposent plusieurs contraintes :

- D'une part, le traitement ne peut demander une modification du décodeur, il doit par conséquent être placé à l'encodeur et ne pas modifier la conformité du flux encodé à la norme H.264/AVC.
- D'autre part, n'ayant pas accès au cœur d'encodage de l'AQILIM, nous nous sommes concentrés dans un premier temps sur un prétraitement indépendant de l'encodeur, pour ensuite placer nos traitements au sein du codec x264, implémentation reconnue de la norme H.264/AVC, représentant pour Digigram une solution ouverte et disponible à coût moindre.

Chapitre 1. La compression vidéo : Standard et Implémentations

1. 1. Introduction

La transmission de contenu vidéo sur des réseaux dédiés ou sur internet nécessite l'utilisation d'algorithmes de compression vidéo performant qui permettent le meilleur compromis entre la perte d'information nécessaire à une réduction significative du débit et la conservation de la qualité. Afin de réduire la quantité d'information à transmettre, les algorithmes d'encodage vidéo exploitent les redondances spatio-temporelles, statistiques et perceptuelles d'une séquence vidéo. Ces principes communs aux trois normes d'encodage vidéo principales de la famille MPEG (Moving Picture Expert Group) (MPEG2, H264/AVC et HEVC) coexistant dans le monde de la vidéo professionnelle, seront présentés dans ce chapitre. La norme MPEG2 a été définie en 1994 par le MPEG en collaboration avec l'ITU-T et normalisé sous le standard ISO/IEC 13818. L'encodage MPEG2 est principalement utilisé de nos jours pour l'encodage des chaînes SD de la TNT. De plus, cette norme ayant été largement déployée, certains équipements perdurent et bien que les performances de ce codec ne soient pas comparables aux codecs modernes H264/AVC et HEVC, la compatibilité avec cette norme doit toujours être assurée et le transcodage MPEG2 vers H264 fait partie de tous les produits vidéo proposé par Digigram. Le standard H.264/AVC, successeurs de MPEG2, normalisé en 2003, est le codec le plus largement utilisé et fait l'objet de nos travaux de prétraitement. Nous présenterons en détails son fonctionnement dans la suite de ce chapitre, puis nous décrirons les principales innovations proposées par le standard HEVC normalisé au début de l'année 2013.

Les différences de performances entre les implémentations d'une norme d'encodage vidéo en termes de qualité perçue pour un débit donné, dépendent des parties non normées d'un codeur vidéo. C'est-à-dire le contrôle de débit, qui contraint l'encodage pour atteindre un débit cible, et le RDO (Rate Distorsion Optimization) qui choisit les outils d'encodage permettant le meilleur compromis en qualité et débit. Nous présenterons les principes de ces deux modules qui participent fortement à la différence de qualité entre deux implémentations du même standard d'encodage.

Finalement nous détaillerons le fonctionnement du codec x264 et plus particulièrement son contrôle de débit. x264 est l'implémentation libre la plus connue de la norme H.264/AVC, elle est utilisée dans le logiciel FFmpeg et utilisée dans le milieu industriel comme point de comparaison pour le développement de codec vidéo. Nos travaux sur l'encodage en CBR, présentés au chapitre 4, sont basés sur ce codec.

1. 2. Les principes de l'encodage vidéo - Normes H.264/AVC et HEVC

Dans cette section, nous allons introduire les principes de l'encodage vidéo communs à tous les codeurs vidéo utilisés de nos jours, puis nous présenterons la norme d'encodage vidéo H.264/AVC qui a fait l'objet de nos travaux et nous exposerons les améliorations principales apportées par la nouvelle norme de codage HEVC.

L'expertise en encodage vidéo ainsi que le partage de connaissances faisant partie de mes missions au sein de Digigram, j'ai eu l'occasion de réaliser plusieurs formations sur lesquelles cette première partie s'appuie.

1. 2. 1. Les principes du codage vidéo

Afin de réduire la bande passante nécessaire au transport de flux audiovisuels, les algorithmes de compression vidéo cherchent à utiliser le minimum d'informations nécessaires pour représenter fidèlement une séquence vidéo. Pour cela ils cherchent à supprimer toutes les redondances dans le flux vidéo. Trois types principaux de redondances

sont exploités : les redondances temporelles, spatiales et statistiques [3]. La majeure partie des travaux sur l'encodage vidéo reposent sur le même principe de codeur hybride[1] illustré par la Figure 1.1, dont nous allons présenter les principes dans cette partie du chapitre.

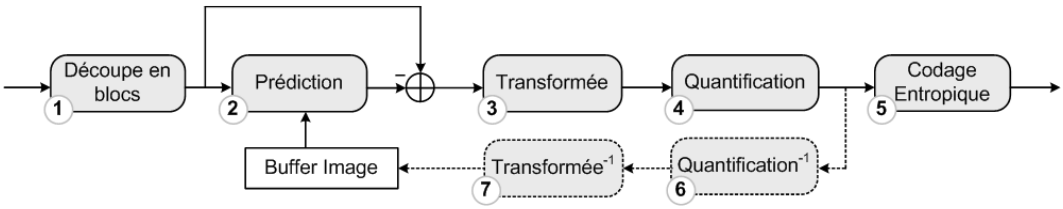


Figure 1.1. Architecture d'un codeur vidéo hybride

1. 2. 1. 1. Exploitation des redondances temporelles – Estimation/Compensation de Mouvement

Une séquence vidéo est constituée d'une succession d'images capturées par un réseau de capteurs photosensibles à une fréquence donnée. Une image est représentée par une matrice de données appelées pixel, représentées au format YUV (Cf. Figure 1.2.a). Cet espace colorimétrique permet de traiter différemment la composante de luminance Y des composantes de chrominance U et V et d'ainsi exploiter la faible sensibilité du système visuel humain aux informations de couleur en sous-échantillonnant les composantes de chrominance [4]. L'utilisation du format 4:2:2 (Cf. Figure 1.2.b) reste réservée aux professionnels de la télévision disposant d'une forte bande passante. Le format 4:2:0, dont les composantes de chrominance sont sous-échantillonnées par quatre (Cf. Figure 1.2.c), est le format le plus largement utilisé en encodage vidéo car la perte de qualité comparativement au format 4:2:2 est peu perceptible pour un gain important de bande passante. Nous avons utilisé ce format pour tous nous travaux sur l'encodeur H.264/AVC.

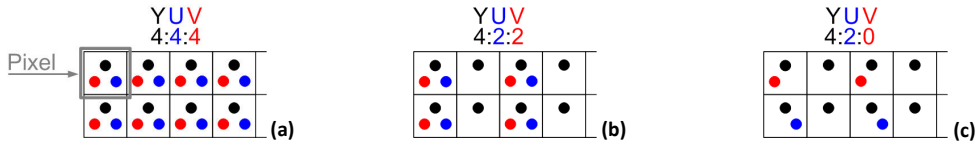


Figure 1.2. Illustration des formats de sous-échantillonnage couleur

La sensation de mouvement fluide est correctement restituée à partir d'une fréquence de 25 images capturées par seconde et d'autant plus avec des fréquences supérieures (50, 60 fps). Les images successives d'une séquence vidéo sont fortement corrélées, on parle de redondances temporelles. La différence de contenu entre deux images successives d'une séquence vidéo est généralement due au mouvement des objets de la scène et/ou de la caméra, intervenu entre la captation de ces deux images.

Afin d'exploiter les redondances temporelles, une idée simple est de coder uniquement l'information de différence entre deux images au lieu de coder deux images, on parle de codage FD (Frame Difference). La Figure 1.3 présente l'image de différence entre deux images consécutives d'une séquence. Lorsque la séquence contient peu de mouvement comme la séquence *Akiyo*, le procédé est très efficace car l'image de différence contient très peu d'information. Mais lorsque la séquence contient du mouvement, comme la séquence *Foreman*, la méthode devient sous-optimale.

Afin d'exploiter plus efficacement les redondances temporelles d'une séquence vidéo, la famille de codec MPEG estime à l'encodeur le mouvement intervenu entre deux images afin de transmettre au décodeur uniquement

[1] Le terme hybride fait référence à l'utilisation de deux techniques au sein d'un codeur, à savoir un codage dans le domaine DCT (Discrete Cosinus Transform) et un mécanisme de prédiction basé sur un algorithme d'estimation de mouvement.

l'information de mouvement, représentée par les vecteurs mouvements et l'erreur de prédiction. Ce principe est le cœur de fonctionnement des codecs MPEG-2, H.264/AVC et HEVC que nous allons détailler dans la suite de ce chapitre.

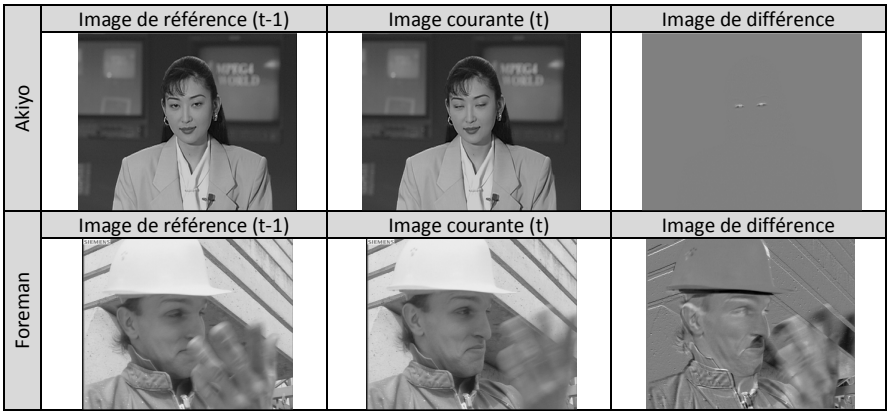


Figure 1.3. Illustration du principe de codage FD (Frame Difference)

Afin d'estimer le mouvement entre deux images d'une séquence vidéo, l'image courante est découpée en blocs de pixels appelés macroblocs, traités les uns après les autres (Figure 1.1. Etape 1). Le mouvement de chaque macrobloc est estimé entre l'image courante et précédente (Etape 2). L'algorithme d'estimation de mouvement employé consiste à trouver dans l'image précédente (dite de référence), le macrobloc le plus ressemblant au macrobloc courant selon un critère de distorsion. La SAE (Sum Of Absolute Errors) est couramment utilisée comme métrique de distorsion, elle mesure la somme absolue des différences entre le macrobloc courant et la prédiction:

$$SAE = \ i = 1N \sum_{j=1}^{N} |pix(i,j) - pred(i,j)|$$

Équation 1.1. *SAE (Sum Of Absolute Errors)*

Où *pix(i,j)* est un pixel appartenant au macrobloc/sous-partition original et *pred(i,j)* sa prédiction.

Ce principe est illustré par la Figure 1.4, la différence de position entre le macrobloc courant et sa prédiction est codée par l'intermédiaire d'un vecteur mouvement. Plusieurs stratégies d'estimation de mouvement ont été développées pour optimiser le temps de calcul, une revue des techniques proposées peut être trouvée dans [5].

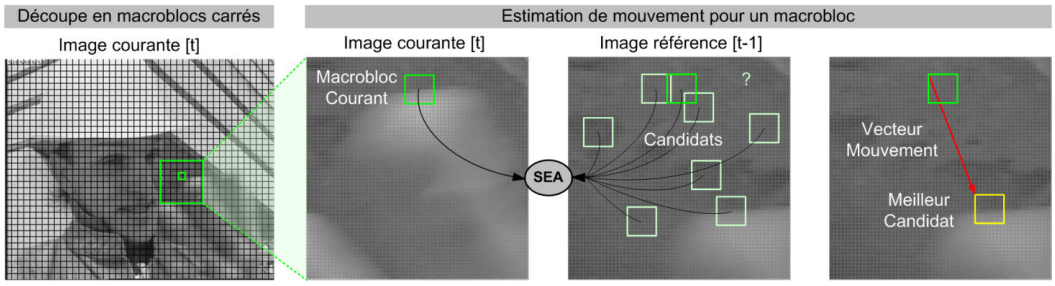


Figure 1.4. Illustration de l'estimation de mouvement

En réalisant le même travail pour tous les macroblocs de l'image, on obtient un champ de vecteur mouvement. Ces vecteurs appliqués à l'image de référence, donnent la prédiction de l'image courante à partir de l'image de référence. L'image prédite n'est pas parfaitement identique à l'image courante. Par conséquent, si uniquement les vecteurs mouvements sont transmis au décodeur, l'image décodée contiendra les erreurs de prédiction qui seront plus ou moins importantes en fonction de la complexité de la séquence. C'est pourquoi l'erreur de prédiction, appelée DFD (Displaced Frame Difference), est codée et transmise au décodeur en plus des vecteurs mouvements. La Dfd est obtenue par simple soustraction de l'image prédite à l'image originale. La Figure 1.5 présente le résumé du principe d'encodage prédictif inter-image, ce mécanisme est le cœur de fonctionnement d'un encodeur vidéo.

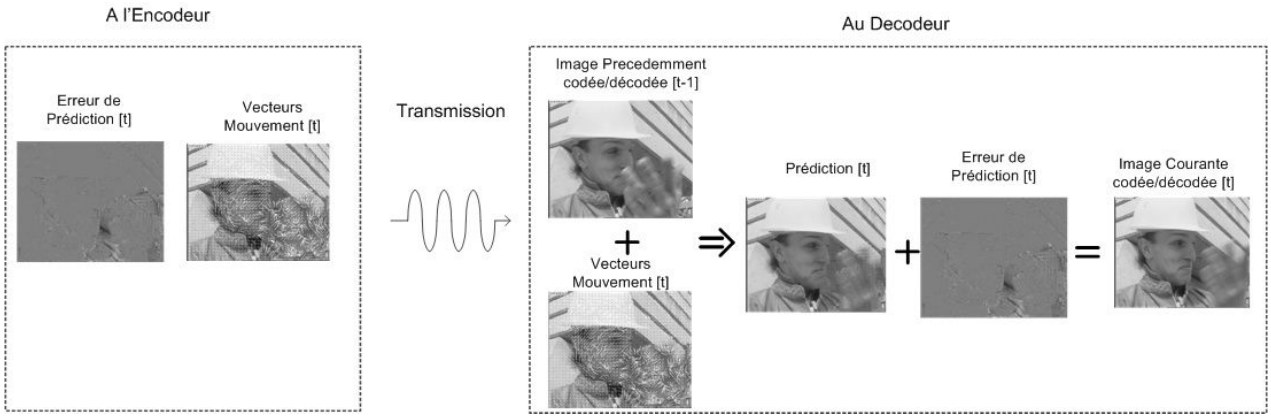


Figure 1.5. Principe du codage inter-image

Dans la pratique, pour accroître l'efficacité de codage, les macroblocs vont subir un codage par transformée. Chaque macrobloc est alors divisé en blocs de pixels de taille réduite (8x8 en MPEG2 et 4x4 en H.264/AVC) qui subissent une transformée dite en cosinus discrète (Cf. paragraphe 1. 2. 1. 3.).

Afin d'éviter un phénomène de glissement, les images de référence utilisées par l'estimation et la compensation de mouvement à l'encodeur et au décodeur doivent être identiques. Le décodeur n'ayant pas accès aux images originales, l'encodeur émule le décodeur en utilisant comme référence non pas l'originale mais l'image décodée (Figure 1.1. Etape 6 et 7).

1. 2. 1. 2. GOP : Groupe of Pictures

Les images encodées à l'aide du mécanisme de prédiction que nous avons présenté au paragraphe précédent sont appelées les images P (prédites), la prédiction est réalisée à partir d'une image précédemment encodée. Deux autres types d'images sont utilisés par la famille de codeur MPEG (Figure 1.6).

Les images bidirectionnelles (B), utilisent deux images de référence pour la prédiction, une référence passée et une référence future. Ce mécanisme est plus couteux en temps de calcul mais permet d'améliorer nettement la précision de la prédiction, notamment dans le cas de l'apparition d'un objet comme illustré dans la Figure 1.6.

Les images P et B sont appellées images Inter, le principal défaut du codage inter-image, est la propagation des erreurs. De par le mécanisme de prédiction, les artefacts de codage ainsi que les éventuelles erreurs de transmision, se répercutent d'image en image. Afin de limiter la propagation des erreurs, un troisième type d'images est utilisé : les images intra (I) appelées également image clés (keyframe). Les images I, ne sont pas codées en fonction des autres images de la séquence, elles sont décodables de manière indépendante et servent de point d'ancrage au sein de la séquence vidéo compressée.

Figure 1.6. Type d'image d'un GOP

L'ordonnancement des trois types d'image dans le flux vidéo est appelé GOP (Group Of Pictures), caractérisé par sa longueur (N) et la fréquence des images P (M). La Figure 1.7 présente quelques exemples de structures de GOP courantes. Le choix d'une structure et d'une longueur de GOP se fait en fonction de l'application visée (type de contenu, de la bande passante disponible, qualité et latence attendue). Par exemple la longueur moyenne de GOP est de 12 images pour les DVD et de 24 pour la diffusion TNT. Autre exemple, l'utilisation d'image B permet de réduire

sensiblement la bande passante nécessaire à la transmission d'une séquence vidéo, cependant le fait que les images B utilisent une prédiction bidirectionnelle induit une latence plus importante du système d'encodage/décodage, car les images ne sont pas encodées dans l'ordre de la séquence (Figure 1.7 (e)). Pour une application de contribution point à point nécessitant une faible latence et une haute qualité, on choisira un GOP IP (b) ou I (d), à condition d'avoir une forte bande passante à disposition. Une étude de l'influence du paramétrage du GOP sur l'encodage vidéo peut être trouvée dans l'article [6], publié durant la thèse.

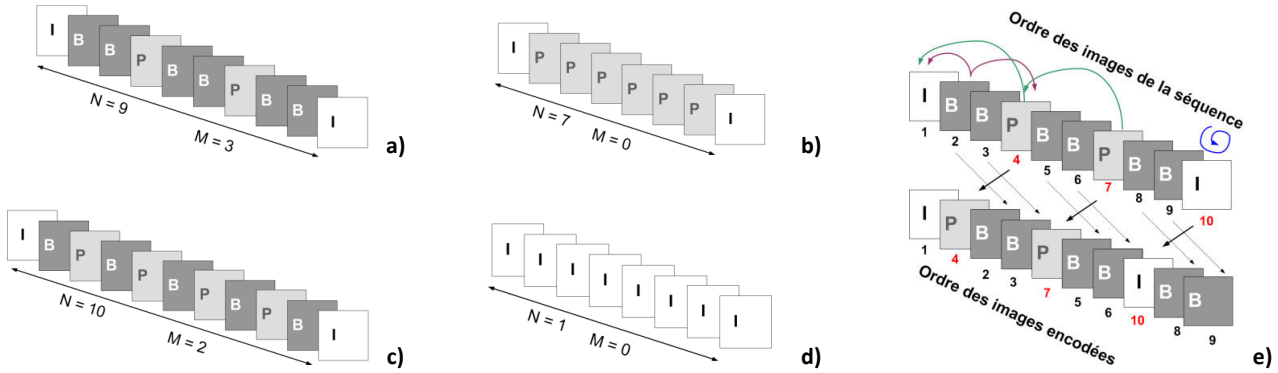


Figure 1.7. Exemple de structure de GOP
a : GOP IBBP(9) – b : IP(7) – c : IBP(10) – d : Intra – e : Ordre d'encodage et de présentation avec l'utilisation d'images B.

1. 2. 1. 3. Les redondances spatiales - DCT

Les pixels voisins dans une image sont fortement corrélés, on parle de redondances spatiales. Les encodeurs vidéo exploitent ces redondances en appliquant une transformation aux blocs de pixels pour les représenter de manière plus compacte dans le domaine des fréquences spatiales. Des différentes transformées proposées, la DCT (Discrete Cosine Transform) type 2 à deux dimensions est la plus populaire et utilisée dans la plupart des codeurs vidéo modernes [7].

Une DCT type 2 à deux dimensions décompose le contenu spatial d'un bloc de pixel suivant les composantes de base représentées par la Figure 1.8 (a). Un bloc de NxN pixels noté $x_{n,m}$ est alors transformé en une matrice de NxN coefficients DCT $X_{k,l}$, où k et l désignent les fréquences spatiales dans la direction horizontale et verticale respectivement. Dans la pratique, les blocs sont de taille 8x8 (MPEG2) ou 4x4 (H.264/AVC).

Deux exemples de décomposition fréquentielle sont données par les Figure 1.8 (b) et (c) respectivement pour un bloc homogène et un bloc contenant un contour. Le coefficient basse fréquence appelé coefficient DC représente la valeur moyenne du bloc, qui est à peu près identique pour les deux exemples. L'énergie des autres coefficients fréquentiels varie fortement en fonction du contenu, plus un bloc comporte de variations lumineuses, plus ses composantes hautes fréquences sont énergétiques.

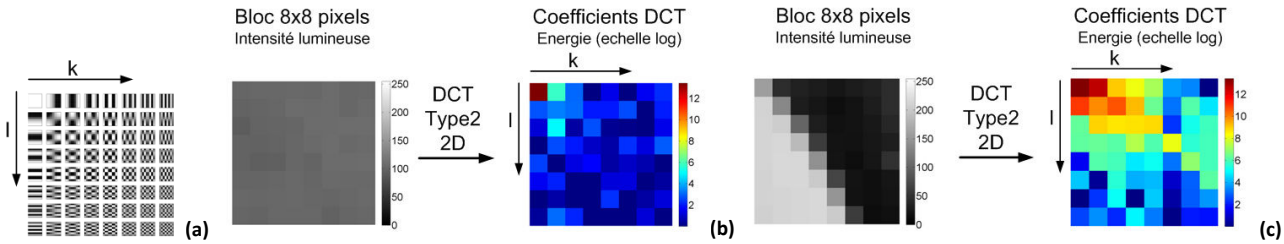


Figure 1.8. Illustration des fréquences spatiales
a : composantes de base de la DCT2D Type 2 – b : bloc homogène et ses composantes DCT– c : bloc contenant un contour et ses composantes DCT

L'étape de DCT est non destructive, un bloc est reconstruit à l'identique en appliquant une DCT inverse (IDCT) sur les coefficients transformés. La DCT est entièrement réelle ce qui la rend plus compacte que la TFD (Transformée de

Fourier Discrète) qui représente les données images par des données complexes, augmentant ainsi les ressources de calculs nécessaires. La DCT et l'IDCT à deux dimensions sont exprimées par les expressions suivantes :

$$X_{k,l} = \frac{2}{N} C_k C_l \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x_{n,m} \times cos \frac{\pi \left(n + \frac{1}{2}\right) k}{N} \times cos \frac{\pi \left(m + \frac{1}{2}\right) l}{N}, \qquad 0 \leq k, l \leq N-1$$

Équation 1.2. *DCT type 2, 2D*

$$x_{n,m} = \frac{2}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} C_k \times C_l \times X_{k,l} \times cos \frac{\pi \left(n + \frac{1}{2}\right) k}{N} cos \frac{\pi \left(m + \frac{1}{2}\right) l}{N}, \qquad 0 \leq k, l \leq N-1$$

Équation 1.3. *IDCT type 2, 2D*

Avec

N	:	Le nombre de pixels d'un bloc dans une dimension
K, l	:	Fréquences spatiales dans la dimension horizontale et verticale
$X_{k,l}$	:	Coefficient DCT de rang fréquentiel (k, l)
$x_{n,m}$	:	Pixel de coordonnées (n, m)
C_α, avec α = k ou l	:	constante de normalisation telle que $\begin{cases} \frac{1}{\sqrt{2}}, & \alpha = 0 \\ 1, & sinon \end{cases}$

Les coefficients fréquentiels sont ensuite quantifiés afin de réduire la précision avec laquelle un bloc est représenté, cette étape est destructive (Figure 1.1. Etape 4). En contrôlant la précision des coefficients fréquentiels, la quantification contrôle directement le nombre de bits nécessaire à la représentation d'un bloc de pixels.

1. 2. 1. 4. **Redondances Statistiques – Codage Entropique**

Le codage entropique consiste à attribuer un code binaire optimal aux éléments à transmettre au décodeur. Les coefficients fréquentiels quantifiés sont représentés à l'aide des valeurs entières qui apparaissent plus ou moins fréquemment, on parle de redondances statistiques. Pour réduire le nombre de bits nécessaire à la représentation des éléments à coder, le codage entropique exploite ces redondances statistiques en attribuant un code à chaque élément, dont la longueur est inversement proportionnelle à sa probabilité d'apparition. On parle de code à longueur variable (VLC) en opposition aux codes à longueur fixe qui attribue le même nombre de bits à chaque élément à coder. Le codage entropique fait partie de la théorie de l'information [8] et la complexité des algorithmes mis en œuvre dans les encodeurs vidéo est variable.

Pour illustrer le principe du codage entropique, le Tableau 1.1 prend l'exemple simple d'une source d'information à huit symboles de probabilités d'apparition connues, codée à l'aide d'un codage de type Huffman utilisé en compression d'image JPEG et vidéo MPEG2. Avec un code à longueur fixe, 3 bits sont nécessaires pour représenter les huit symboles, en considérant les probabilités d'apparition des différents symboles, la longueur moyenne de codage est réduite à 2.45 bit par symbole.

Symboles de la source d'information	Probabilités d'apparition	Code à longueur variable type Huffman	Longueur moyenne de code
Symbole S1	0.4	0	$\overline{L} = \sum_{i=1}^{8} p_i \times l_i$ $\overline{L} = 0.4 + 0.45 \times 3 + 0.05 \times 4 + 0.05 \times 5 + 0.05 \times 6$ $\overline{L} = 2.45\ bit/symbole$
Symbole S2	0.2	111	
Symbole S3	0.15	110	
Symbole S4	0.1	100	
Symbole S5	0.05	1010	
Symbole S6	0.05	10111	
Symbole S7	0.025	101101	
Symbole S8	0.025	101100	

Tableau 1.1. Exemple simple de codage entropique pour une source à 8 symboles

1. 2. 2. La norme H.264/AVC – Génération actuelle de codeur

Le standard d'encodage H.264/AVC a été normalisé en 2003 par le groupe de travail JVT (Joint Video Team) issu des groupes MPEG (Moving Picture Expert Group) et VCEG (Video Coding Expert Group, ITU-T). H.264/AVC, également appelé MPEG4-Part10, a été défini pour dépasser les limitations de ses prédécesseurs MPEG-2, H.263 et MPEG4-Visual et ainsi répondre aux besoins croissants de compression amenés notamment par l'avènement des contenus haute définition.

Le codeur H.264/AVC est un codeur hybride au sens qu'il est basé sur l'utilisation d'une estimation/compensation de mouvement et d'une DCT comme nous l'avons présenté au paragraphe précédent. La Figure 1.9 présente l'architecture du codeur H.264/AVC, que nous allons détailler dans ce paragraphe.

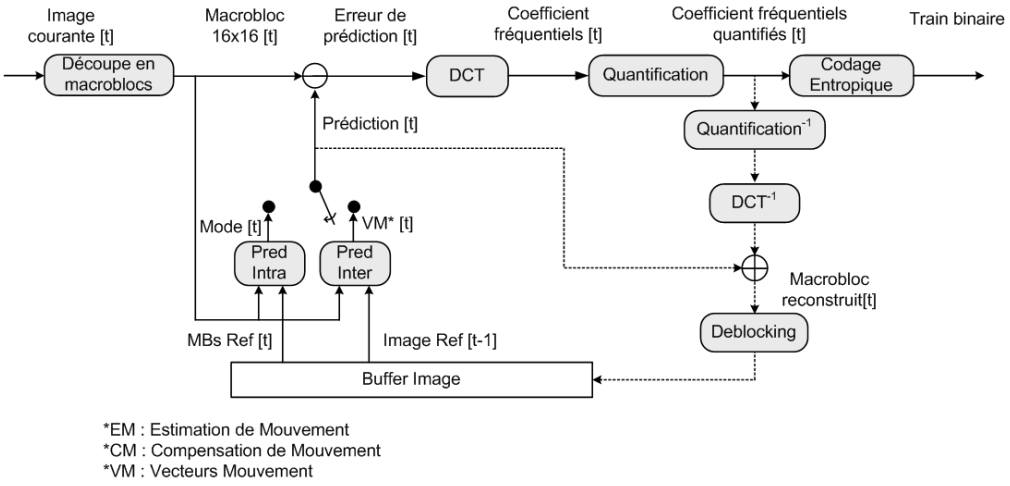


Figure 1.9. Schéma bloc de l'encodeur H.264/AVC

1. 2. 2. 1. La prédiction inter-image

Comme nous l'avons vu dans la section 1. 2. 1. 1. , le processus d'estimation et compensation de mouvement est le cœur de fonctionnement du codeur dont la qualité a un impact direct sur l'efficacité du codeur. L'étape d'estimation de mouvement a naturellement concentrée de nombreux efforts de recherche dont les plus notables sont : l'algorithme de *Tree Structured Motion Compensation* et le filtre d'interpolation permettant une recherche au ¼ de pixel, ainsi que l'utilisation de plusieurs images de référence [9].

- **Tree Structured Motion Compensation**

L'image est découpée en macroblocs de taille 16x16 pixels. L'estimation/compensation de mouvement est efficace pour des macroblocs contenant peu de détails, mais ne permet pas d'estimer finement les zones à forte activité spatiale et/ou temporelle. Pour cela, H.264/AVC autorise la redécoupe d'un macrobloc en sous partitions 16x8, 8x16 et 8x8, eux-mêmes pouvant être découpés en sous-partitions 8x4, 4x8 et 4x4. La Figure 1.10 (a) présente les 7 découpes possibles ainsi que l'ordre dans lequel les partitions au sein d'un macrobloc sont traitées et la Figure 1.10 (b) présente un exemple de partitionnement pour une partie d'une image issue de la séquence *Soccer*. On peut noter qu'un élément syntaxique (flag) doit être ajouté dans le flux encodé pour indiquer les découpes inférieures à 8x8.

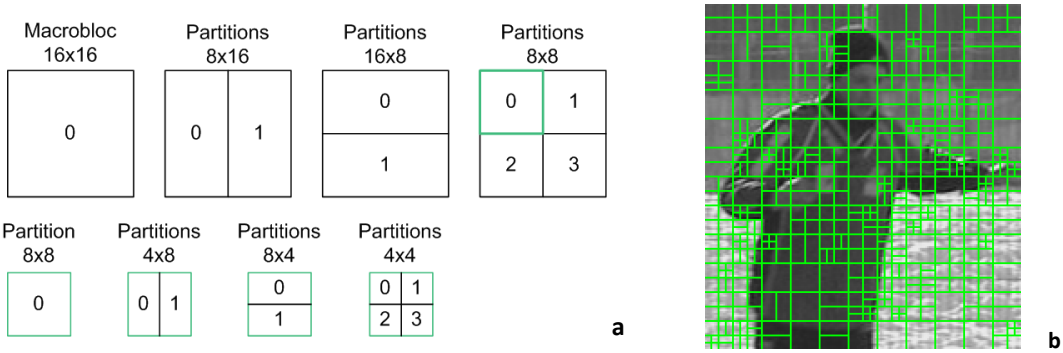


Figure 1.10. Tree Structured Motion Compensation
a : Partitions autorisées d'un macrobloc 16x16 – b : Exemple de partition H.264/AVC

- **Filtre d'interpolation et Précision au ¼ de pixel**

H.264/AVC propose un nouveau filtre d'interpolation permettant une précision de l'estimation de mouvement au quart de pixel. Les Figure 1.11 et Figure 1.12 présentent respectivement l'interpolation au ½ et ¼ de pixel pour une partie particulière de l'image.

Dans un premier temps, l'interpolation au demi-pixel utilise un filtre RIF de six coefficients appliqué tout d'abord verticalement et horizontalement (Figure 1.11 (1) et (2)). Les pixels interpolés **c** et **n** sont calculés respectivement à partir des échantillons A-F et G-K suivant :

$$c = Clip\{round((A - 5 * B + 20 * C + 20 * D - 5 * E + F)/32 + {}^{1}/_{2}), 0, 255\}$$

Équation 1.4.
Interpolation au ½ pixel

Avec Clip la fonction qui permet de borner le résultat entre 0 et 255.

Ensuite, l'échantillon interpolé **o** (Figure 1.11 (2) et (3)) est calculé à partir des échantillons interpolés [l,r] obtenus à l'étape précédente, de la même manière que l'Équation 1.4.

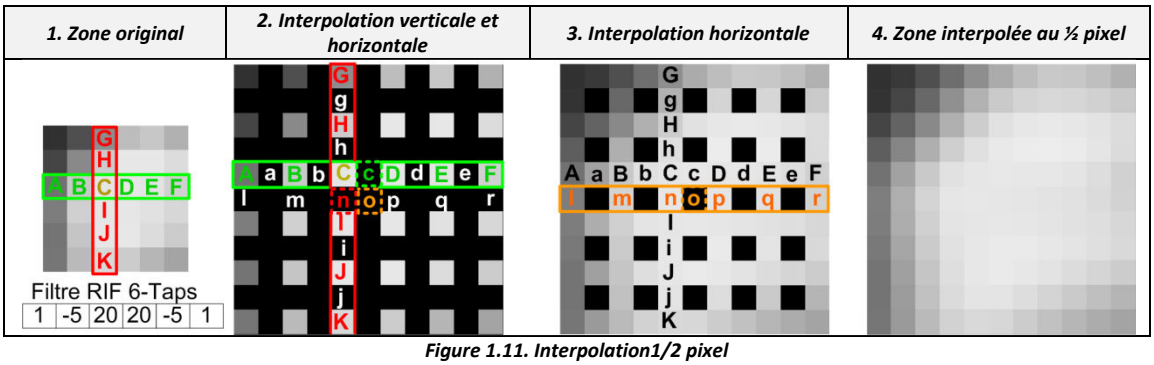


Figure 1.11. Interpolation1/2 pixel

A partir de l'image interpolée au ½ pixel (Figure 1.11 (4) et Figure 1.12 (1)), les échantillons interpolés au ¼ de pixels utilisent une moyenne simple de leurs deux voisins verticaux, horizontaux et diagonaux.

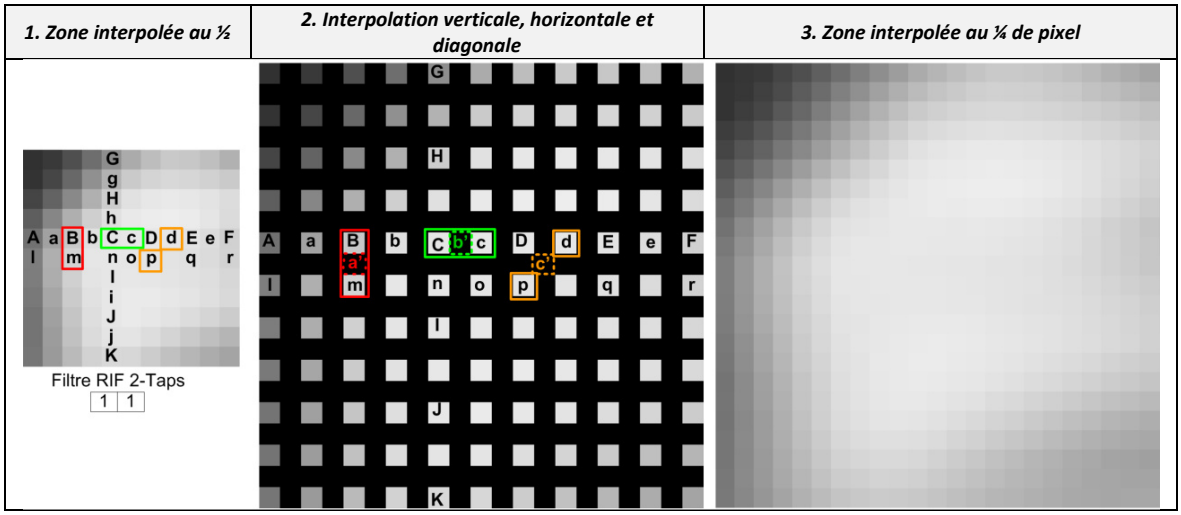


Figure 1.12. Interpolation1/4 pixel

- **Référence Multiples**

La norme H.264/AVC permet d'utiliser plusieurs images de référence potentielles afin de choisir le macrobloc candidat le mieux adapté au macrobloc en cours de traitement. Là où MPEG-2 ne permettait d'utiliser que l'image précédemment codée comme référence (ou les deux précédemment codées pour les images B), H.264/AVC permet d'utiliser jusqu'à 16 images de référence. Le nombre autorisé d'images de référence dépend du format d'image, de la taille du buffer de décodage et de l'échantillonnage couleur. Chaque partition 16x16, 16x8, 8x16 et 8x8 d'une image peut avoir sa propre image de référence repérée dans le flux par un index spécifique, cependant les partitions de taille inférieure à 8x8 partagent la même référence afin de limiter le nombre d'informations à transmettre au décodeur.

Généralement l'image la plus proche temporellement de l'image courante, est la plus ressemblante et donc choisie comme référence de l'estimation de mouvement. Dans ce cas l'augmentation du nombre de références n'a pas ou peu d'effet. Cependant, dans le cas de mouvements périodiques, une image temporellement éloignée de l'image courante peut être fortement ressemblante. En pratique, le nombre d'images de référence dépasse rarement 4 car cela entraîne une hausse importante du temps de calcul pour un gain assez faible en qualité. Une étude de l'impact de l'augmentation du nombre d'images de référence en termes de temps de calcul et qualité peut être trouvée dans [6].

1. 2. 2. 2. La Prédiction Intra Image

Une des innovations majeures amenées par la norme H.264/AVC est la définition d'une prédiction intra-image. Le codage intra-image H.264/AVC utilise un mécanisme de prédiction qui exploite les redondances spatiales au sein d'une image. Comme pour le codage inter-image, un macrobloc intra 16x16 peut-être découpé en partition de taille inférieure. Ainsi un macrobloc 16x16 intra peut être prédit de deux manières différentes : soit le macrobloc entier est prédit, soit il est segmenté en seize partitions 4x4 qui sont prédites indépendamment. Dans tous les cas, le macrobloc/partition est prédit(e) à partir des macroblocs voisins appartenant à la ligne et la colonne précédente comme le montrent les Figure 1.13 (1) et Figure 1.14 (1).

La norme définie neuf modes de prédiction intra 4x4 décrits dans la Figure 1.13 et quatre 16x16 décrits dans la Figure 1.14. Une métrique permet de sélectionner le mode de prédiction conduisant à la prédiction la plus fidèle au macrobloc/partition courant. La métrique SAE est la plus largement utilisée pour sélectionner le mode de prédiction (Cf. Équation 1.1). Ainsi, le mode de prédiction ayant la plus faible SAE est choisi comme meilleur candidat et son numéro est transmis au décodeur qui connaît également les modes de prédiction. Il faut noter que l'utilisation de la métrique permettant de choisir la meilleure prédiction n'est pas normée.

Dans les exemples des Figure 1.13 et Figure 1.14, le mode DC est choisi comme meilleure prédiction 16x16 et le mode vertical est choisi comme prédiction de la première partition 4x4 du macrobloc car ils présentent les plus faibles SAE avec le macrobloc/partition courant(e). Le nombre réel de modes intra testés à chaque macrobloc dépend des voisins disponibles dans le buffer de décodage.

Un dernier type de codage Intra est possible, le mode PCM (Pulse Code Modulation) où les valeurs de pixels du macrobloc sont directement transmises au décodeur.

Le choix entre le meilleur mode 16x16, 4x4 et le mode PCM nécessite d'évaluer l'efficacité de codage de ces trois possibilités en terme de qualité (Distorsion) et de nombre de bits. Cette décision est réalisée par le module RDO (Rate Distorsion Optimisation) dont nous reparlerons plus tard (paragraphe 1. 2. 4. 1.). Généralement, un macrobloc homogène pourra être efficacement prédit par un mode 16x16, alors qu'un macrobloc dont le contenu varie comme dans notre exemple, sera plus fidèlement prédit par les modes 4x4. Comme on peut le voir dans l'exemple ci-dessous, les seize partitions 4x4 (Figure 1.14 (c)) donnent une prédiction plus précise que le mode 16x16 DC (Figure 1.13).

A un macrobloc de 16x16 échantillons de luminance, correspond deux blocs 8x8 de chrominance U et V. Les blocs de chrominance sont prédits de façon similaire aux macroblocs de luminance 16x16. Quatre modes de prédiction intra 8x8 de chroma sont possibles : le mode vertical, horizontal, DC et plan.

H.264/AVC décrit un type particulier d'images Intra, les images IDR (instantaneous decoder refresh). Les images IDR sont codées suivant les modes intra mais elles ont la particularité de vider les buffers de décodage, par conséquent aucunes images situées après une IDR ne peut utiliser de références plus anciennes que la dernière IDR. Ceci permet d'insérer des points d'accès dans le flux encodé.

A la manière de la prédiction inter-image, une fois la prédiction intra réalisée, l'erreur de prédiction est codée puis transmise au décodeur avec le mode de prédiction (Cf. Figure 1.9).

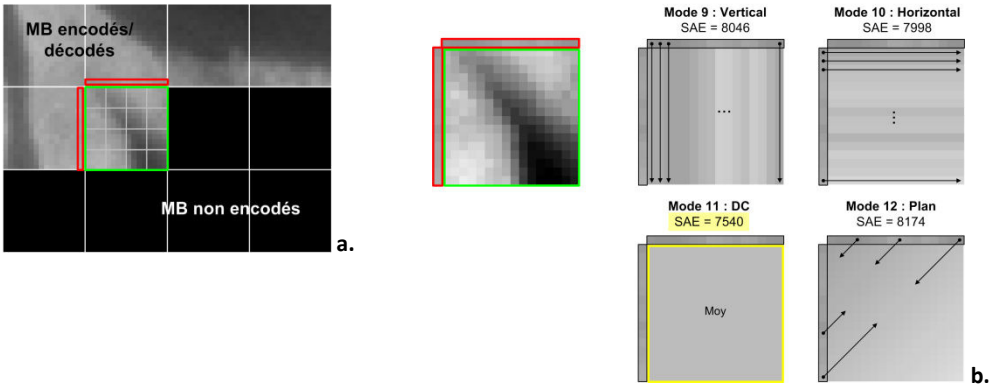


Figure 1.13. Mode Intra image 16x16

a : Macrobloc 16x16 et ses voisins précédemment encodés disponibles dans le buffer de décodage – b : les quatre modes de prédiction intra 16x16

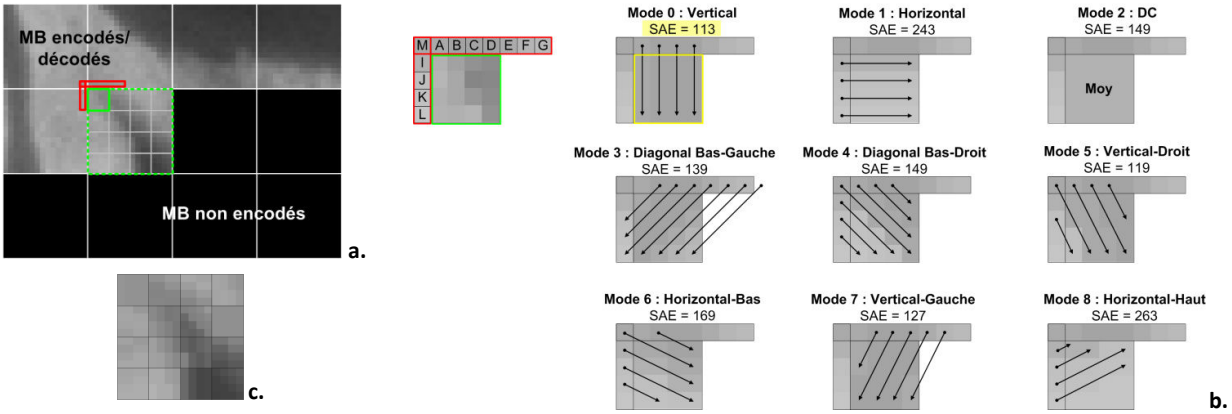


Figure 1.14. Mode Intra image 4x4

a : Partitions 4x4 et ses voisins précédemment encodés disponibles dans le buffer de décodage – b : les seize modes de prédiction intra 4x4 – c : Macrobloc prédit à partir des 16 prédictions 4x4.

1. 2. 2. 3. **Le GOP**

Une image peut contenir des macroblocs de différents types recensés dans le Tableau 1.2.

- Les Images Intra peuvent contenir des macroblocs encodés en mode I_PCM qui consiste à encoder directement les valeurs de pixels. Ce mode est principalement utilisé quand le coût débit-distorsion du meilleur mode Intra est trop élevé.
- Les images P et B peuvent contenir des macroblocs Intra (I ou I_PCM) dans le cas où l'estimation de mouvement est jugée trop mauvaise. En effet, l'estimation de mouvement trouve toujours un meilleur candidat, mais cela n'implique pas que le macrobloc courant ai été trouvé dans l'image de référence.
- Lorsqu'un macrobloc peut être prédit précisément, il est inutile de transmettre au décodeur l'erreur de prédiction, c'est le cas de macroblocs homogènes ou ayant un mouvement constant. Dans ces cas, le macrobloc peut être codé avec très peu de bits, aucune erreur de prédiction n'est transmise au décodeur et le vecteur mouvement est estimé au décodeur à partir des vecteurs mouvements des macroblocs voisins. Ce mode est appelé « Skip » pour les images P et « Direct » pour les macroblocs B.
- Les images de références passées et futures utilisées pour la prédiction bidirectionnelle sont conservées dans deux listes appelées respectivement L0 et L1. Les images B peuvent contenir des macroblocs utilisant plusieurs types de prédiction :
  - La prédiction peut utiliser uniquement l'image de référence passée (Liste L0), dans ce cas le macrobloc devient de type P.
  - La prédiction peut utiliser uniquement l'image de référence future (Liste L1).
  - La prédiction utilise une image de référence issue des deux listes.

	Image I	Image P	Image B
Type de Macrobloc	I	I	I
	I_PCM	I_PCM	PCM
		P_L0	B_L0 (équivalent P_L0)
		P_SKIP	B_L1
			B_L0_L1
			B_Direct

Tableau 1.2. Types de macrobloc autorisés par type d'image

1. 2. 2. 4. **DCT Entière et quantification**

H.264/AVC a introduit plusieurs nouveautés à l'étape de transformée comparativement à ses prédécesseurs. D'une part, H.264/AVC utilise une transformée de taille 4x4[2]. D'autre part, elle utilise uniquement des coefficients entiers afin d'optimiser l'implémentation. Les étapes de DCT et de quantification sont appliquées sur les 16 blocs 4x4 d'un macrobloc quel que soit le partitionnement utilisé à l'étape de prédiction.

L'implémentation de l'étape DCT est réalisée à l'aide d'un produit matriciel entre le bloc de pixel et le cœur DCT (*DCT core*), noté A dans les équations ci-dessous. Le cœur DCT est de même taille que le bloc de pixels à transformer. Dans le cas de la DCT 4x4, la transformation s'exprime par :

[2] On peut noter que H.264/AVC propose également la possibilité d'utiliser une DCT entière de taille 8x8 dans le profile High (Cf. 1. 2. 2. 1.) pour coder plus efficacement les macroblocs homogènes ou à faibles mouvement.

$$Y = A \times X \times A^T, \qquad A = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix},\ a = \frac{1}{2},\ b = \sqrt{\frac{1}{2}}\cos\frac{\pi}{8},\ c = \sqrt{\frac{1}{2}}\cos\frac{2\pi}{8}$$

Équation 1.5.
DCT 4x4

Avec

Y : Le bloc de coefficients fréquentiels
X : Le bloc de pixels résiduels
A, A^T : Le cœur DCT et sa transposée

Le cœur DCT peut être rendu entier (matrice T) en factorisant la matrice de poids flottants appelée «prescale factor » (matrice PF) de la manière suivante :

$$Y = (T \times X \times T^T) \otimes PF, \qquad T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix},\ PF = \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix},\ a = \frac{1}{2},\ b = \sqrt{\frac{2}{5}},\ c = a$$

Équation 1.6.
DCT entière 4x4

La matrice de coefficients prescale est intégrée à l'étape de quantification de manière à être transparente pour l'encodeur.

H.264/AVC utilise une quantification uniforme que nous détaillerons plus tard. Les coefficients prescale de la DCT sont intégrés à la quantification comme suit:

$$\overline{X_{k,l}} = round\left(\frac{X_{k,l}}{Q_{step}}\right), \qquad 0 \le k,l \le N-1$$
$$\overline{X_{k,l}} = round\left(X_{int,k,l} \times \frac{PF}{Q_{step}}\right)$$

Équation 1.7.
Quantification Uniforme

Avec $X_{k,l}$ et $\overline{X_{k,l}}$ respectivement les coefficients DCT de rang fréquentiel (k,l) et leur version quantifiées, et Q_{step} le pas de quantification uniforme. $X_{int,k,l}$ les coefficients issus de l'étape de DCT entière.

Afin de simplifier les calculs arithmétiques, la division des PF par le pas de quantification est ramenée à un simple décalage de registre :

$$\overline{X_{k,l}} = round\left(X_{int,k,l} \times \frac{MF}{2^{qbits}}\right) \qquad qbits = 15 + floor\left(\frac{QP}{6}\right)$$

Équation 1.8.
Quantification uniforme Implémentation simplifiée

Avec MF les « multiplication Factor » définis tels que : $\frac{MF}{2^{qbits}} = \frac{PF}{Q_{step}}$.

La norme H.264/AVC définit un paramètre de quantification QP, lié au pas de quantification par une relation logarithmique. Cinquante-deux valeurs de QP sont disponibles, de 0 à 51. A chaque fois que la valeur de QP augmente de 6 unités, le pas de quantification double. La relation entre le pas de quantification et le QP est donnée par :

$$Q_{step} = a\big(mod(QP,6)\big) \times 2^{round\left({}^{QP}/_{6}\right)}$$

Équation 1.9.
Relation Q_{step} et QP

Avec a une matrice de six coefficients définis dans la norme [0.625, 0.6875, 0.8125, 0.875, 1.0, 1.125], mod(QP,6) le modulo de QP par 6, round(.) la fonction arrondi.

Les facteurs MF dépendent uniquement du rang fréquentiel du coefficient à quantifier et du pas de quantification, par conséquent ils peuvent être définis dans une table à deux entrées. Les MF sont définis pour les six premiers QPs.

1. 2. 2. 5. Transformée d'Hadamard

La transformée d'Hadamard (Walsh-Hadamard) discrète, comme la DCT, représente des échantillons sous une forme plus compacte à l'aide de coefficients représentants l'énergie des différentes fréquences spatiales. La compaction d'énergie de la transformée d'Hadamard est moins efficace que la DCT, mais elle a l'avantage d'utiliser uniquement des poids entiers et de norme 1 qui facilitent son implémentation :

$$Y = H \times X \times H^T, \qquad A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Équation 1.10. *DWHT 4x4*

La transformée d'Hadamard est utilisée dans H.264/AVC pour représenter plus efficacement les coefficients DC issus de la DCT dans deux cas : pour le codage des informations de chrominance et pour les macroblocs prédits en Intra_16x16.

Lorsqu'une prédiction 16x16 est utilisée, la majorité de l'énergie est concentrée dans les seize coefficients DC des partitions 4x4 qui sont fortement corrélés. Afin de limiter le nombre de bits dépensés pour coder ces coefficients, une transformée d'Hadamard est appliquée sur le bloc DC regroupant les 16 coefficients DC d'un macrobloc. Ainsi les coefficients DC sont représentés sous une forme plus compacte (avec moins de coefficients). Par conséquent, les coefficients DC et AC sont quantifiés séparément comme le montre la Figure 1.15, les coefficients DC sont quantifiés en ordre Raster (ligne par ligne) et les coefficients AC de chaque partition 4x4 en ordre zigzag.

Pour le codage d'un bloc de chrominance 8x8 (on se place dans le cas du codage 4 :2 :0), une transformée DCT 4x4 entière est appliquée aux quatre blocs 4x4. Les quatre coefficients DC issus de la DCT sont ensuite regroupés et une transformée d'Hadamard 2x2 est appliquée.

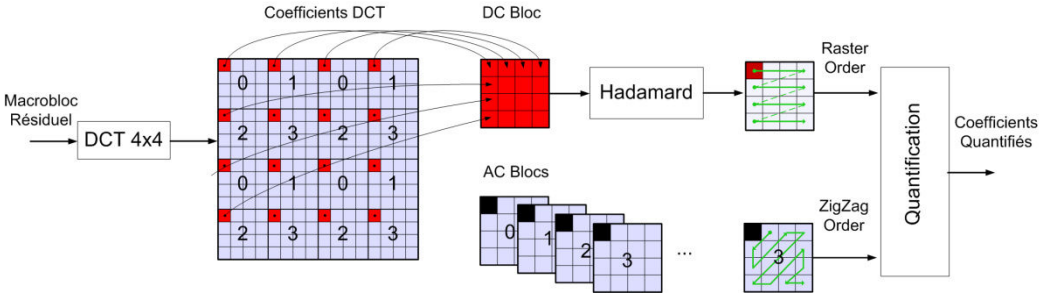


Figure 1.15. Codage du mode Intra_16x16

1. 2. 2. 6. Codage Entropique

La norme H.264/AVC propose deux modèles de codage entropique : CAVLC (Context Adaptive Variable Length Coding) et CABAC (Context Adaptive Binary Arithmetic Coding). Le codage CAVLC est similaire dans le principe au codage de Huffman déjà présent dans MPEG-2 en y apportant une adaptation au contexte c'est-à-dire aux éléments précédemment codés. H.264/AVC introduit le codage CABAC basé sur un codage arithmétique, qui est plus performant que le codage CAVLC au prix d'une complexité accrue. A la différence des codes VLC qui représentent un nombre fixe de symboles avec un code à longueur variable, les codes arithmétiques représentent un nombre variable de symboles avec un code à longueur variable. L'étude [10] montre que le codage CABAC apporte un gain moyen de l'ordre de 10% par rapport à CAVLC, dans une gamme de débit donnant une qualité acceptable (PSNR 30-38dB, Cf. Équation 1.12).

- **CAVLC (Context Adaptive Variable Length Coding)**

Le codage CAVLC exploite les observations suivantes : d'une part un bloc 4x4 de coefficients quantifiés est principalement constitué de zéros, par conséquent la représentation compacte RLC (Run Length Coding) est utilisée pour coder uniquement les coefficients non-nuls et le nombre de zéros qui les précédent *run_before*. D'autre part, les coefficients non nuls ont souvent une amplitude de +/- 1, ainsi CAVLC introduit le paramètre *trailing_ones* qui indique une succession de coefficients d'amplitude égale à +/- 1 en ordre zigzag, le signe d'un *trailing_ones* est codé dans le paramètre *trailing_ones_sign_flag*. Le nombre de coefficients non-nuls et de *trailing_ones* est regroupé dans le paramètre *coeff_token*. Le nombre de zéros existants après le premier coefficient non-nul est codé dans le paramètre *total_zero*. Deux adaptations au contexte sont utilisées à partir des observations suivantes :

- Il existe une corrélation entre le nombre de coefficients non-nuls de blocs voisins, cette caractéristique est exploitée pour coder le paramètre *coeff_token*. Quatre tables sont disponibles, le choix de la table de codage optimale dépend du nombre de coefficients non-nuls des blocs 4x4 adjacents (haut et gauche).
- Pour le codage de l'amplitude des coefficients non nuls, la table VLC choisie dépend des coefficients du bloc précédemment codés pour exploiter le fait que les coefficients énergétiques sont généralement de faible rang fréquentiel.

- **CABAC (Context Adaptive Binary Arithmetic Coding)**

L'algorithme CABAC est basé sur un codage arithmétique, il comprend quatre étapes.

- La binarisation : l'élément à coder (Coefficient fréquentiel, vecteur mouvement, mode de prédiction intra, partitionnement, etc) est en premier lieu binarisé, c'est-à-dire converti en code binaire. La table de binarisation dépend de la nature de l'élément à coder. Cette étape de binarisation n'est pas une simple représentation en base 2, elle peut être vu comme un codage VLC. Ensuite les bits du code généré sont traités un à un itérativement par les trois étapes suivantes.
- Plus de 400 modèles contextuels sont disponibles et initialisés en début de slice (Cf. 1. 2. 2. 9.), un modèle contextuel contient la probabilité d'apparition de l'élément 1 ou 0. Ces modèles sont mis à jour en cours d'encodage est choisi pour chaque élément binaire en fonction des données précédemment encodées.
- Une fois les probabilités d'apparition connues, un codage arithmétique à deux symboles (0 ou 1) est appliqué aux bits de l'élément à coder.
- Enfin, à chaque bit, le modèle de probabilité est mis à jour en fonction de la valeur du bit courant.

1. 2. 2. 7. **Deblocking**

La norme H.264/AVC introduit un nouvel outil à l'encodeur et au décodeur connu sous le nom de filtre de deblocking, pour minimiser l'effet de bloc qui reste l'artefact principal de codage en H.264/AVC [11].

Le filtre est appliqué à l'encodeur et au décodeur (Cf. Figure 1.9), on parle de « In-Loop Filter» en opposition au « Post filter » qui serait appliqué uniquement au décodeur avant la présentation des images. L'intérêt de l'implémentation « In-Loop » est double, d'une part l'application du filtre au décodeur avant la présentation des images, permet de lisser l'image pour la rendre plus agréable à l'œil. D'autre part, le fait d'appliquer le filtre sur les images de référence avant qu'elles soient sauvées dans le buffer image à l'encodeur et au décodeur, permet d'améliorer le procédé d'estimation/compensation de mouvement.

Le filtre de Deblocking est un filtre passe-bas adaptatif non-linéaire appliqué aux frontières des blocs 4x4 d'un macrobloc. Pour ne pas apporter une impression de flou gênante, le filtre n'est pas appliqué de façon systématique à chaque frontière de bloc. L'algorithme de filtrage s'attache à différencier les frontières artificielles causées par l'encodage des contours naturels de l'image.

Le filtre de Deblocking apporte une réelle amélioration de qualité à faible débit, lorsque l'effet de bloc est très présent (Figure 1.16). Cependant la complexité algorithmique du filtre est grande et demande jusqu'à un tiers des ressources du décodeur [11], ce qui peut être gênant dans le cas d'équipements embarqués. On peut noter que le filtre de Deblocking est peu utilisé à haut débit car il a tendance à lisser les images.

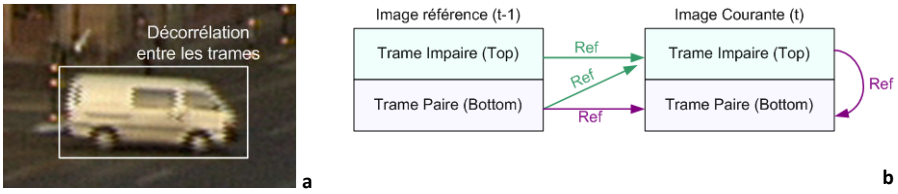

a. b.

Figure 1.16. Filtre de Deblocking
a : partie d'une Image 1280x720 encodée à 2M sans Filtre de Deblocking – b : Avec application du filtre de Deblocking

1. 2. 2. 8. Gestion des contenus entrelacés

La norme H.264/AVC prévoit différents outils pour optimiser l'encodage des contenus entrelacés, qu'on peut classer en deux catégories : les codages fixes et adaptatifs.

Le codage fixe dédié aux contenus entrelacés est appelé *Field coding* en opposition au *Frame coding* que nous avons décrit jusqu'ici. Le *Field Coding* (codage par trames), consiste à traiter séparément la trame paire et trame impaire d'une image. Afin de tirer parti de la décorrelation des trames d'une image, le mécanisme de prédiction inter-image est alors légèrement modifié, dans le cas d'une trame P les deux trames précédemment codées sont utilisées pour réaliser la prédiction comme le présente la Figure 1.17 (b).



a b

Figure 1.17. Codage par Trame
a : Effet de stries dans une image entrelacée – b : Références pour l'estimation d'une trame P en codage par trame

Deux types de codages adaptatifs sont disponibles dans H.264/AVC, le PAFF (Picture Adaptive Field/Frame Coding) et le MBAFF (MacroBloc Adaptive Field/Frame Coding). Comme leur nom l'indique, ces deux modes d'encodage permettent de choisir entre les deux modes de codage fixe à chaque image ou macrobloc en fonction de la décorrelation des trames présentes. En effet, dans une séquence entrelacée, la décorrelation de trame est visible uniquement dans les zones en mouvement, le codage MBAFF permet de choisir le codage optimal par macrobloc et d'ainsi gagner jusqu'à 10% de bande passante comparativement au codage PAFF [12].

1. 2. 2. 9. Organisation des données

- **Slice**

La norme H.264/AVC permettant de découper une image en groupes de macroblocs décodables de manière indépendante, appelés slice. Une slice peut uniquement contenir des macroblocs contigus en ordre raster (ligne par ligne). Une slice est au minimum de la taille d'un macrobloc et au maximum de la taille d'une image. L'intérêt principal est de limiter la propagation des erreurs au sein d'une image, car une erreur intervenant dans une slice ne pourra

atteindre la slice suivante. Certaines implémentations de la norme utilisent les slices pour paralléliser l'implémentation du codeur en consacrant un thread[3] à une slice.

- **NALU**

La norme H.264/AVC définit les mécanismes d'encodage permettant de représenter efficacement le contenu d'une séquence vidéo (VCL [Video Coding Layer]) que nous venons de voir, ainsi que l'organisation du flux de données codées permettant de faciliter le transport de la vidéo sur des réseaux de diffusion (NAL [Network Abstraction Layer]).

Les informations encodées sont regroupées en unité de transport de façon à faciliter leur transmission. Cette unité de transport est appelée une NAL unit (ou NALU). Une NALU contient une partie de données utiles appelée RBSP (Raw Byte Sequence Payload) d'une taille multiple d'octets, et un header d'un octet indiquant la nature des informations qu'elle contient. La Figure 1.18 présente la structure d'une NALU :

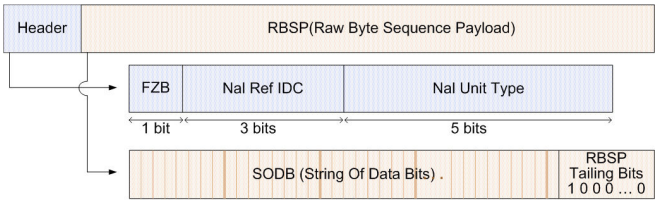


Figure 1.18. Composition d'une NAL unit

L'entête d'une NALU fait huit bits et contient trois types d'information :

- Le FZB (Forbidden_zero_bit), doit être égal à zéro d'après la norme. Si une erreur intervient lors de la transmission, ce bit peut être mis à 1 pour indiquer que la NAL contient une erreur et le décodeur pourra alors corriger ou jeter ce paquet.
- Le Nal_Ref_IDC, indique si la NALU contient des informations appartenant à une image utilisée comme référence pour une prédiction, ceci permet de placer l'image dans le buffer approprié au décodage.
- Nal_unit_Type : Indique le type de données contenu dans la partie utile de la NALU. Les NALUs contenant des informations de vidéo compressée sont appelée VCL NAL (Video Coding Layer), les autres types de NALUs sont appelées non-VCL NAL unit (ex : SEI).

Le RBSP (Raw Byte Sequence Payload) est la partie utile d'une NALU. Un RSBP contient un SODB (String of data bits), une suite de bits représentant les informations encodées en H.264/AVC. Le premier octet d'un RBSP contient les 8 premiers bits (de poids fort) d'un SODB et ainsi de suite. Si la taille d'un SODB n'est pas un multiple d'octets, le RSBP peut contenir des bits de remplissage (Tailing Bits). Le fait d'encapsuler les données encodées (SODB) dans un RBSP permet d'aligner les données sur un nombre d'octet, ce qui facilite le traitement.

Une NALU transportant des informations appartenant aux slices codées de l'image est appelée une slice VCL. Une NALU peut transporter des informations autres que les données codées d'une image, elle peut contenir des informations sur la séquence (SPS [Sequence Parameter Set]), sur l'image courante (PPS [Picture Parameter Set]), ou des messages SEI (Supplemental Enhancement Information) qui aident le processus de décodage (Cf. Figure 1.19).

Pour être transmis, le flux H.264/AVC est encapsulé dans un conteneur regroupant les données audio, vidéo et les données auxiliaires comme les sous-titres. Le conteneur utilisé en Contribution IP est le MPEG-TS normalisé par le groupe MPEG (ISO/IEC 13818-1). La Figure 1.20 présente l'organisation du flux H.264/AVC dans un conteneur MPEG-TS. L'unité d'accès est un élément commun aux normes d'encodage H.264/AVC et de transport MPEG-TS, elle fait le lien entre les deux standards.

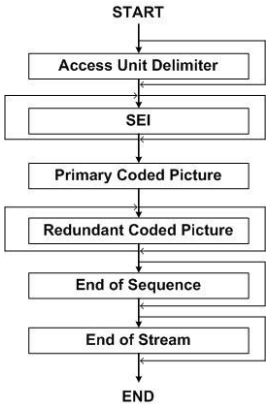


Figure 1.19.Composition d'une unité d'accès (AU)

[3] Chemin d'exécution, permet plusieurs traitements en parallèle sur un ou plusieurs CPU.

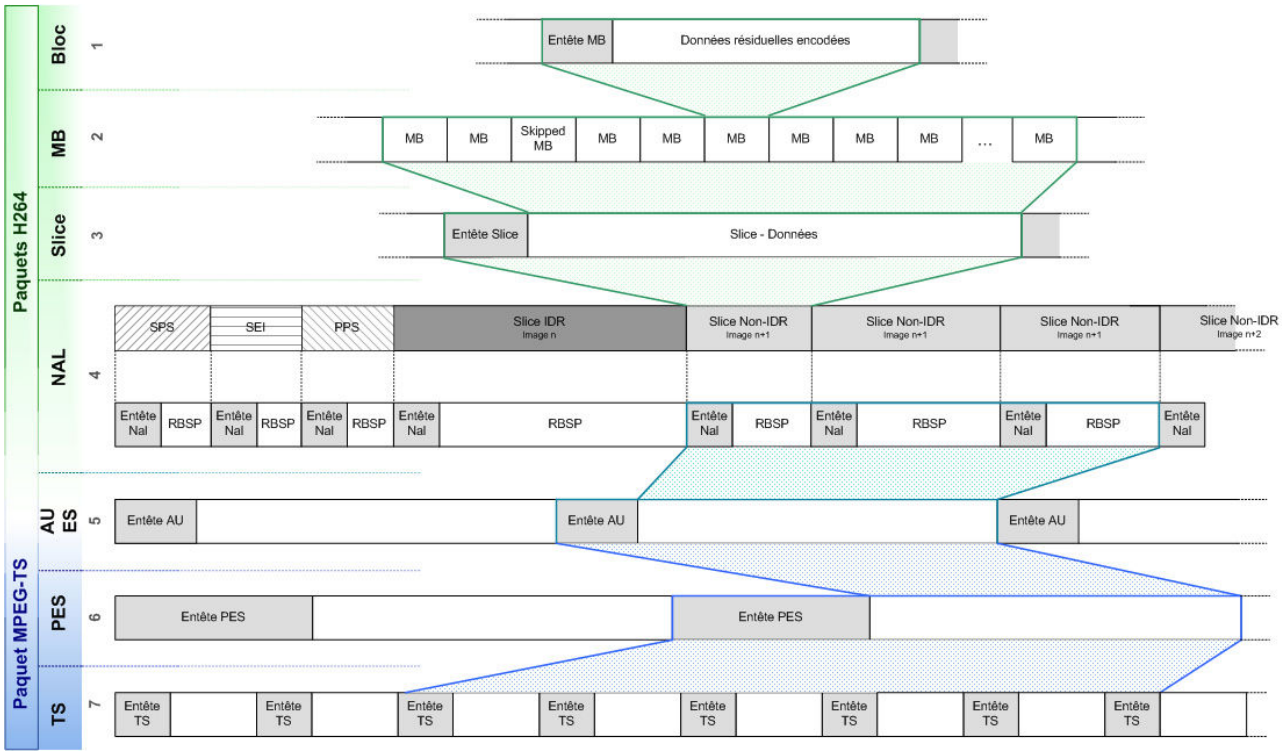


Figure 1.20 Organisation du flux H.264/AVC encapsulé dans un conteneur MPEG-TS

1. 2. 2. 1. Profils et niveaux

L'ensemble des outils d'encodage, dont ceux que nous venons de décrire, sont regroupés en profils dans la norme H.264/AVC. Les profils d'encodage différent par la complexité des outils d'encodage utilisés et par conséquent par la qualité de la séquence encodée. Les trois profils les plus répandus dans le monde industriel sont par ordre de complexité : le profile Baseline, Main et High. Le Tableau 1.3 indique la disponibilité des outils décrits dans ce chapitre en fonction de ces trois profils d'encodage. Le profil baseline ne supporte ni les images B ni le CABAC, il est utilisé principalement pour les séquences vidéo dédiées à être visionnées sur smartphones, qui ont des ressources de décodage limitées. Le profil High est couramment utilisé pour l'encodage de séquence HD en qualité professionnelle.

La norme définit également des niveaux d'encodage (levels) qui apporte des limitations sur la nature de la séquence vidéo à encoder (résolution et nombre d'image par seconde) ce qui impose les caractéristiques matérielles du décodeur visé (par exemple la mémoire nécessaire).

Outils	Baseline	Main	High
Slice I	x	x	x
Slice P	x	x	x
Slice B	ø	x	x
Format 4:2:0	x	x	x
Format 4:2:2 (Profil High 4:2:2)	ø	ø	x
Format 4:4:4 (Profil High 4:4:4)	ø	ø	x
Tree Structured Motion Compensation jusqu'à des partitions 4x4	x	x	x
Filtre d'interpolation ¼ de pixel	x	x	x
Multiples images de référence	x	x	x
Prédiction Intra 16x16 et 4x4	x	x	x
DCT 8x8	x	x	ø
Transformée d'Hadamard	x	x	x
Codage CAVLC	x	x	x
Codage CABAC	ø	x	x
Filtre de Deblocking	x	x	x
Codage entrelacé (PAFF, MBAFF)	ø	x	x

Tableau 1.3. Disponibilité des outils présentés dans ce chapitre en fonction du profile H.264/AVC

1. 2. 3. La norme HEVC - Nouvelle génération de codeur

La norme HEVC, normalisée par le JCT-VC au début de l'année 2013 [13], a été développée pour répondre aux besoins croissants de compression de données, dû à l'émergence des hautes résolutions 2K (HD 2048x1080), 4K (4*HD). Le but annoncé était également de réduire le débit de moitié à qualité constante pour permettre entre autre de transmettre les contenus HD vers un plus grand nombre d'abonnés aux services multimédia sur IP, ce but a été atteint puisque HEVC permet de réduire jusqu'à 70% le débit par rapport à H.264/AVC [14], [15]. De plus, HEVC s'est concentré à améliorer le parallélisme du processus d'encodage et de décodage afin de faciliter l'implémentation industrielle temps réel.

La Figure 1.21 présente l'architecture du codeur HEVC qui est très proche du codeur H.264/AVC. Dans la suite de ce paragraphe, nous allons présenter les principales améliorations proposées par HEVC par rapport à H.264/AVC.

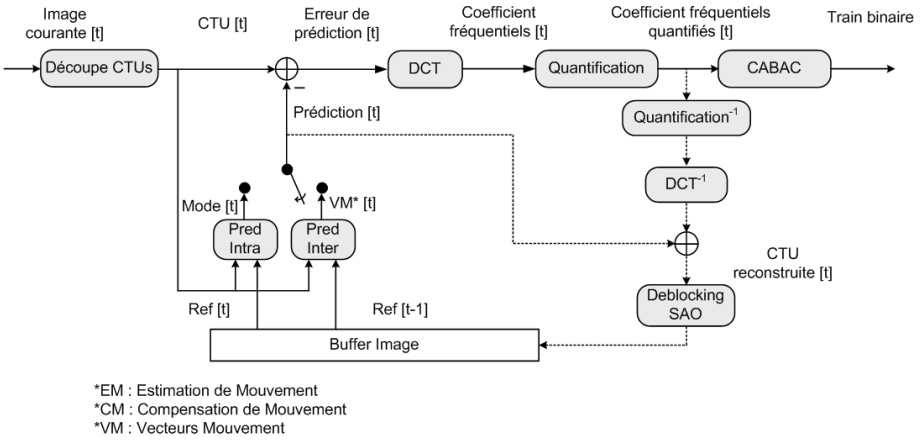


Figure 1.21. Architecture du codeur HEVC

1. 2. 3. 1. Structure de l'image

La norme HEVC introduit l'utilisation de blocs de codage appelés CTB (Coding Tree Block) pouvant atteindre 64x64 pixels, pour optimiser l'encodage des nouvelles hautes résolutions 2K et 4K. L'utilisation combinée de blocs de codage et de DCT de plus grande taille, permet de représenter efficacement les zones homogènes qui sont d'autant plus présente que la résolution augmente. Il faut noter que l'augmentation de la taille des blocs de codage est l'une des raisons principales de l'amélioration des performances d'encodage apportée par HEVC et qu'elle est d'autant plus intéressante que la résolution de la source est grande [16].

La norme HEVC propose un nouveau partitionnement de l'image présenté dans la Figure 1.22. Le macrobloc H.264/AVC est remplacé par une CTU (Coding Tree Unit) pouvant prendre une taille de 64x64, 32x32 ou 16x16 pixels. Une CTU regroupe un bloc de luminance et deux de chrominance appelés CTB (Coding Tree block) ainsi que les informations relatives au codage de cette partie de l'image.

Une CTU peut être découpée en partitions de plus petite taille appelées CU (Coding Unit) suivant l'algorithme quadtree qui décompose chaque bloc en quatre blocs de taille divisée par deux. Une CU peut au minimum avoir une taille de 8x8 pixels. La décision du type de codage (I, P ou B) est prise au niveau CU.

La taille d'une CU peut être trop importante pour réaliser une prédiction suffisamment précise, c'est pourquoi une CU est décomposée en PU (Prediction Unit) dont la décomposition diffère pour le codage inter ou intra (Figure 1.22(3)).

Une fois la/les prédictions inter ou intra réalisées pour la CU courante, HEVC utilise plusieurs tailles de DCT différentes : 32x32, 16x16, 8x8 et 4x4. La CU est découpée en TU (Transform Unit) suivant la décomposition quadtree. La taille des TU n'est pas liée à la taille des PU.

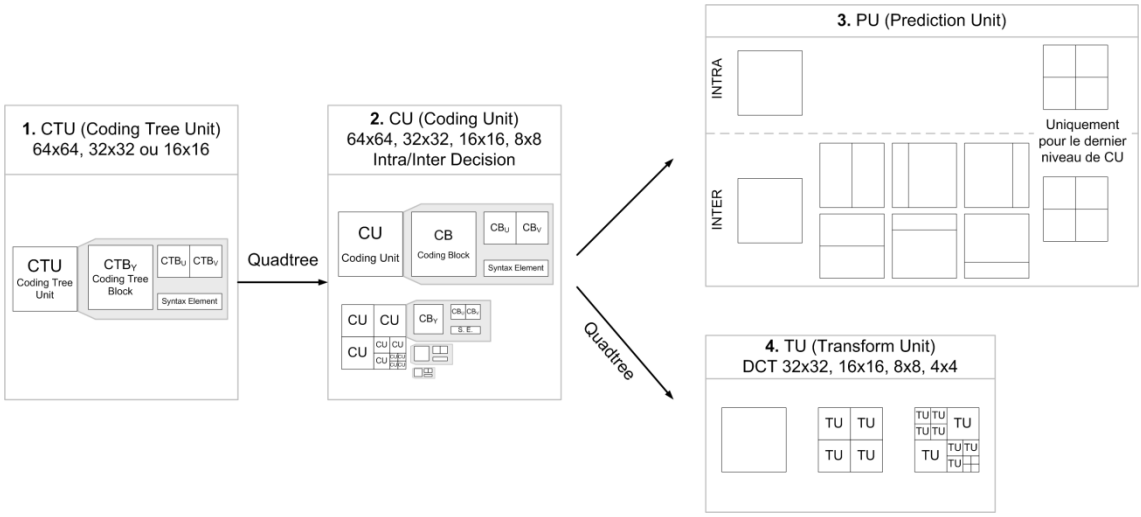


Figure 1.22. Partionnement HEVC

1. 2. 3. 2. Multiplication des modes intra

Le codage intra introduit par le codec H.264/AVC était l'une des innovations les plus importantes comparativement au codec MPEG-2, cependant la prédiction intra H.264/AVC reste peu précise et bien moins performante que la prédiction inter-image. Les expériences menées à Digigram montrent qu'il faut un débit environ six fois plus important pour obtenir la même qualité en codage intra qu'en codage inter-image. HEVC s'est naturellement concentré à améliorer cette étape d'encodage en multipliant le nombre de modes intra disponibles, qui sont comptés au nombre de 36 dans HEVC. La Figure 1.23 présente les 35 modes intra disponibles, cette illustration provient de l'article de Gary J. Sullivan et al. décrivant la norme HEVC [14]. L'alliance du nouveau partitionnement des blocs et des nouveaux modes intra apporte un gain entre 20 et 25% comparativement au profile H.264/AVC Intra [17].

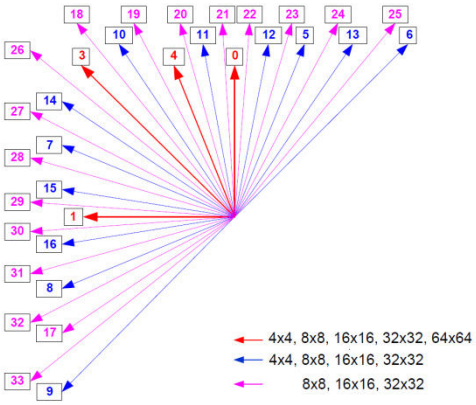


Figure 1.23. Illustration des modes Intra HEVC
Illustration issue de [15]

1. 2. 3. 3. Filtre d'interpolation et compensation de mouvement

Comme dans H.264/AVC, l'étape d'estimation et de compensation de mouvement peut-être réalisée au quart de pixel, cependant, un effort particulier a été apporté au filtre d'interpolation. Contrairement à H.264/AVC qui utilise un filtre à 6 entrées pour l'interpolation au demi-pixel, suivi d'une simple interpolation linéaire pour la génération des échantillons au quart de pixel (1. 2. 2. 1.), HEVC utilise un filtre à 7 ou 8 entrées.

1. 2. 3. 4. Filtres « In-Loop »

Le filtre de Deblocking est toujours présent dans HEVC, il est très proche du filtre défini par la norme H.264/AVC, cependant il a été simplifié afin de limiter son impact sur les performances de décodage.

Un nouveau filtre non-linéaire a été ajouté à la suite du filtre de Deblocking, le filtre SAO (Sample Adaptive Offset). Le filtre de Deblocking se concentre à réduire l'effet de bloc présent dans les images décodées sans connaissance des images originales. Le filtre SAO a pour but de corriger les erreurs de quantification et de prédiction qui génèrent notamment l'effet de Ringing [18] caractérisé par une activité parasite aux contours (Cf. paragraphe 4. 4. 2. 3.). Le SAO décrit deux filtres qui sont choisis à chaque CTB, le filtre EO (Edge Offset) et BO (Band Offset). Le filtre EO a pour but de restaurer les informations de contours en réduisant l'effet de Ringing, alors que le BO a pour but de restaurer les zones homogènes où l'effet de « banding » (escalier dans les dégradés) est présent, en minimisant l'erreur moyenne de codage entre le bloc original et reconstruit. Les deux filtres fonctionnent en ajoutant un offset à chaque échantillon de la CBT, les offsets sont calculés à l'encodeur et envoyés au décodeur. Les offsets dépendent de la classification des pixels, la classification est réalisée à l'encodeur et au décodeur pour limiter le nombre d'informations à envoyer au décodeur.

1. 2. 3. 5. Outils de Parallélisations : Tiles et Wavefront

- **Tiles et Slices**

Une des grandes améliorations apportées par HEVC est la proposition de nouveaux outils de parallélisme permettant de faciliter l'implémentation d'encodeur temps réel.

HEVC définit un nouveau partitionnement de l'image appelé Tiles (tuiles en français). Les tuiles sont des parties rectangulaires de l'image et sont totalement indépendantes les unes des autres (remise à zéro des contextes CABAC en début de tuile). Mais alors qu'elle est la différence entre les slices, toujours présentes dans HEVC et les tuiles ?

Une slice regroupe des CTUs en ordre raster (Figure 1.24) alors qu'une tuile est une zone rectangulaire de l'image ce qui a plus de signification du point de vue du contenu. Ensuite, chaque slice d'une image est précédée d'un entête indispensable à son décodage, ce qui ajoute un nombre de bits au flux qui devient significatif à bas débit, alors que les tuiles sont généralement définies une fois en début de séquence et ne demandent pas d'entête ensuite. Les slices ont pour but de limiter la propagation d'erreurs au sein d'une image ainsi que de contrôler la taille des paquets NAL. Les tuiles ont pour but de paralléliser l'encodage et le décodage du flux. Par exemple, les premières démonstrations d'encodage HEVC en 4K utilisent généralement quatre tuiles pour réaliser quatre encodages HD en parallèle (IBC 2013). On peut noter que le filtre de Deblocking trouve toute son importance à la frontière des tuiles.

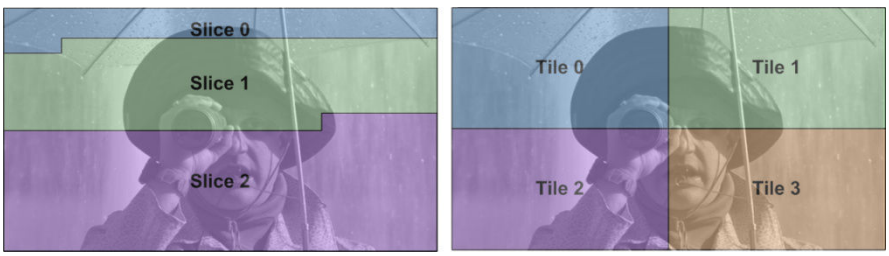


Figure 1.24. Exemple de découpage en Slices et Tiles

- **WPP (WaveFront Parallel Processing)**

Le WPP est un niveau de parallélisme plus fin que les tuiles, il permet de traiter les lignes de CTU en parallèles les unes des autres. Afin de ne pas rompre les références entre lignes de CTU nécessaires au codage Intra et à la synchronisation des contextes CABAC, chaque ligne de CTU est traitée avec deux CTU de retard par rapport à la ligne précédente (Cf. Figure 1.25).

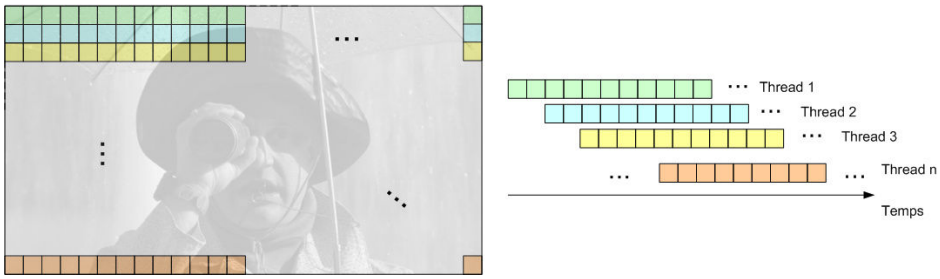


Figure 1.25. Illustration du parallélisme WPP

1. 2. 3. 6. Evolution des performances de la famille MPEG

Pour conclure la présentation des codecs H.264/AVC et HEVC, le Tableau 1.4 inspiré du papier technique [19], présente une comparaison des outils de codage entre les normes MPEG-2, H.264 et HEVC. La Figure 1.26 et le Tableau 1.5 présentent une comparaison des différents codecs de la famille MPEG, réalisée dans l'article [16]. Dans le monde industriel, seuls les codecs MPEG-2, H.264/AVC et très récemment HEVC sont présents.

La norme d'encodage vidéo MPEG-2 a été définie en 1994 par le MPEG (Moving Picture Expert Group) en collaboration avec l'ITU-T et normalisé sous le standard ISO/IEC 13818. L'encodage MPEG-2 est principalement utilisé de nos jours pour l'encodage des chaînes SD de la TNT et flux satellite. De plus, cette norme ayant été largement déployée, des équipements d'encodage MPEG-2 sont toujours présents, notamment dans le secteur Corporate (par exemple hôtelier). Bien que les performances du codec MPEG-2 ne soient pas comparables aux codecs modernes H.264/AVC et HEVC, la compatibilité avec cette norme doit toujours être assurée et le transcodage MPEG-2 vers H.264/AVC fait partie de tous les produits vidéo proposés par Digigram. Pour cette comparaison, tous les outils HEVC disponibles dans la norme ont été activés, le profil Main a été utilisé pour MPEG-2 et le profil High pour H.264/AVC.

La Figure 1.26 présente une comparaison des codecs de la famille MPEG, réalisée dans l'article [16]. Les courbes RD (Rate Distorsion) sont présentées pour deux séquences de résolution 1280x720 à 24 fps en 4:2:0, où la distorsion est mesurée par le PSNR (Peak Signal-to-Noise Ratio) moyen des trois composantes YUV [16] :

$$PSNR_{YUV} = \frac{(6\times PSNR_Y + PSNR_U + PSNR_V)}{8}$$

Équation 1.11. *PSNR Moyen YUV* ***[16]***

$$PSNR = 10\log_{10}\left(\frac{D^2}{EQM}\right)$$

Équation 1.12. *PSNR*

$$EQM = \frac{1}{W\times H}\sum_{i=1}^{W}\sum_{j=1}^{H}\left(pix_{ref}(i,j) - pix(i,j)\right)^2$$

Équation 1.13. *EQM (Erreur Quadratique Moyenne)*

Avec *D* : Dynamique des images, soit 255 pour un codage sur 8 bits
pix : Pixel de l'image dont on mesure la distorsion
pix_{ref} : Pixel de l'image de référence

On note que le codec HEVC apporte de meilleures performances, c'est-à-dire une réduction des distorsions à même débit ou une réduction du débit plus importante à même niveau de distorsion, que le codec H.264/AVC et d'autant plus par rapport au codec MPEG-2. La différence entre les codecs est plus forte à haut débit qu'à bas débit.

Le Tableau 1.5 donne la moyenne des réductions de débit sur cinq séquences HD 1920x1080 et quatre 832x480 entre 24 et 60 fps, 4:2:0. Pour chaque séquence le calcul de la différence en pourcentage entre deux courbes RD est réalisé à partir de plusieurs points de mesure. On note que l'amélioration de performances entre MPEG-2 et H.264/AVC est plus importante qu'entre H.264/AVC et HEVC. Les résultats indiquant que le codec HEVC apporte une amélioration de 35.4% par rapport à H.264/AVC présentés ici sont moyennés sur une large gamme de débits, il faut noter que les communications sur la norme HEVC annoncent généralement une amélioration de 50% des performances de H.264/AVC à haut débit.

	MPEG2	H.264/AVC	HEVC
Partition élémentaire	Macroblock 16x16	Macroblock 16x16	CU (Coding Unit) de 8x8 à 64x64
Partitionnement	Inter 16x8 (pour les macroblocs entrelacés), Intra 8x8	Jusqu'à 4x4	PU (Prediction Unit), décomposition en quadtree jusqu'à 4x4.
Transformée	DCT	DCT entière 4x4 et 8x8	TU (Transform Unit) DCT entière de 4x4 à 32x32
Prédiction Intra	Prédiction des coefficients DC	Jusqu'à 9 modes de prédiction	35 modes de prédiction
Prédiction de l'estimation de mouvement	Interpolation bilinéaire au ½ pixel	Filtre 6 entrées pour l'interpolation au ½ pixel et bilinéaire pour ¼ pixel	Filtre 6 entrées pour l'interpolation au ½ pixel et 7/8 entrées pour ¼ pixel
Codage Entropique	VCL	CABAC, CAVLC	CABAC
Filtre	ø	Deblocking	Deblocking et SAO

Tableau 1.4. Tableau comparatif des outils MPEG2, H.264/AVC et HEVC

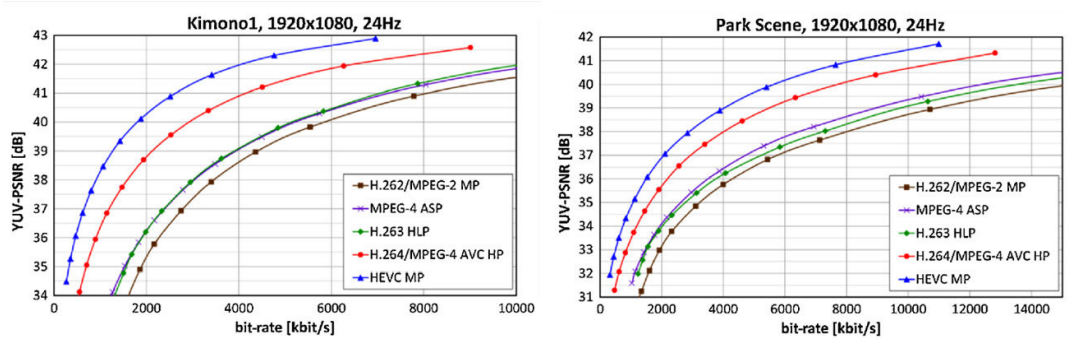


Figure 1.26. Comparaison des performances des codecs de la famille MPEG
Illustration issue de [16]

	Réduction de débit par rapport à			
	H.264/AVC AVC	MPEG-4	H.263	MPEG-2
HEVC	**35.4%**	63.7%	65.1%	**70.8%**
H.264/AVC AVC	-	44.5%	46.6%	**55.4%**
MPEG-4	-	-	3.9%	19.7%
H.263	-	-	-	16.2%

Tableau 1.5. Comparaison des performances des codecs de la famille MPEG
Moyenne sur 8 séquences et la gamme de qualité testée

1. 2. 4. La norme et les implémentations

Jusqu'ici nous avons détaillé les outils de codage proposés par les différentes normes qui permettent de générer un flux encodé décodable par n'importe quel décodeur respectant la norme. Cependant nous n'avons fait qu'aborder les stratégies de codage permettant de prendre les meilleures décisions (modes de prédiction, estimation de mouvement, pas de quantification, etc). Ces stratégies de codage ne sont pas normalisées et dépendent par conséquent de l'implémentation de la norme considérée. Nous aborderons dans la suite de ce chapitre les modules de RDO (Rate Distorsion Optimisation) permettant de choisir les meilleurs modes inter et intra ainsi que le contrôleur de débit permettant de sélectionner le pas de quantification pour atteindre un débit cible. La différence de qualité entre deux implémentations de la norme est principalement liée à ces deux modules.

1. 2. 4. 1. Notion de RDO

Le module de RDO (Rate Distortion Optimisation) décide du meilleur mode de prédiction à utiliser pour un macrobloc donné. Le meilleur mode est celui qui donne le meilleur compromis entre débit et distorsion. Le problème de minimisation sous contrainte se formule comme suit :

$$\min_{m \in Modes} D_m \; sous\; la\; contrainte\; R_m \leq R_c$$

Équation 1.14. *Problème RD sous contrainte*

Avec D_m la distorsion entre le macrobloc original et encodé avec le mode m, R_m le nombre bits réel ou estimé nécessaire à représenter le macrobloc (résidu et mode ou vecteur mouvement) et R_c le nombre de bit maximum autorisé par le contrôle de débit que nous verrons par la suite.

Ce problème complexe de minimisation sous contrainte peut être reformulé à l'aide du multiplicateur de Lagrange λ, permettant au module RDO de choisir le mode de prédiction ayant le plus faible coût-RD J_{mode} :

$$\min_{mode \in M} J_{mode}$$

$$J_{mode} = D_{mode} + \lambda_{mode} \times R_{mode}$$

Équation 1.15. *Coût RD - Mode de prédiction*

Si le macrobloc est codé en inter, le choix se porte sur la découpe qui donne le meilleur compromis débit-Distorsion. Cependant pour chaque découpe, la recherche du vecteur mouvement qui minimise le coût débit distorsion s'exprime également sous la forme d'une minimisation de Lagrange :

$$J_{motion} = D_{motion} + \lambda_{motion} \times R_{motion}$$

Équation 1.16. *Coût RD - Estimation de mouvement*

Avec D_{motion} la distorsion entre le macrobloc original et reconstruit, R_{motion} le nombre de bits nécessaire à la transmission du vecteur mouvement et λ_{motion} le multiplicateur de Lagrange associé à cette minimisation.

- **Mesure de la distorsion**

Généralement, la distorsion entre un macrobloc/bloc original et reconstruit (D_{mode}, D_{motion}) est définie par :

$$D = \sum_{i=0}^{N \times N} |p_i - p_{rec,i}|^{\beta}$$

Équation 1.17. *Métrique de distorsion*

Avec p_i le macrobloc/bloc de taille NxN, $p_{rec,i}$ le macrobloc reconstruit et β égal à 1 dans le cas de la SAD (Sum Of Absolute Differences) et 2 dans le cas de la SSD (Sum of Squared Differences). Toutefois on peut noter que d'autres métriques de distorsion sont utilisées comme la somme des coefficients d'Hadamard (SATD) utilisée dans l'implémentation x264 et que l'utilisation de la métrique perceptive SSIM (Structural Similarity Index Metric) a fait l'objet de l'étude [20].

- **Détermination du multiplicateur de Lagrange**

Pour calculer la fonction coût-RD de chaque mode de prédiction intra et inter, il est nécessaire de connaître la valeur du multiplicateur de Lagrange. Le Lagrangien représente la pente négative de la courbe débit-distorsion de l'erreur de prédiction [21]. Il a été montré que cette pente varie peu en fonction des contenus et pour le codec H.264/AVC sa valeur a été approximée par expérience à [22]:

$$\lambda_{mode} = 0.85 \times 2^{\frac{QP-12}{3}}$$

$$\lambda_{motion} = \begin{cases} \sqrt{\lambda_{mode}}, & Si\ SAD \\ \lambda_{mode}, & Si\ SSD \end{cases}$$

Équation 1.18. *Lagrangien en fonction du QP*

Dans le paragraphe ci-dessous nous présentons succinctement le protocole suivi pour obtenir ce résultat. Pour simplifier les équations on note J, D et R pour respectivement J_{mode}, D_{mode} et R_{mode}. En minimisant l'Équation 1.15, il vient :

$$J = D + \lambda \times R$$

$$\frac{\partial J}{\partial R} = \frac{\partial D(R)}{\partial R} + \lambda = 0 \quad \rightarrow \quad \frac{\partial D(R)}{\partial R} = -\lambda$$

$$\frac{\partial J}{\partial D} = 1 + \lambda \times \frac{\partial R(D)}{\partial D} = 0 \quad \rightarrow \quad \frac{\partial R(D)}{\partial D} = -\frac{1}{\lambda}$$

$$\frac{\partial J}{\partial \lambda} = R = 0 \quad \rightarrow \quad \text{R étant la contrainte initiale}$$

Équation 1.19. *Minimisation de la fonction coût RD*

Les relations liant la distorsion et le nombre de bits au pas de quantification sont typiquement approximées par les relations suivantes à haut et moyen débit [23] :

$$D(\Delta) = \frac{\Delta^2}{\beta}$$

Équation 1.20. *Relation Distorsion/pas de quantification*

$$R(\Delta) = \frac{1}{\alpha} log_e\left(\beta \times \frac{\sigma^2}{\Delta^2}\right)$$

Équation 1.21. *Relation Bit/pas de quantification*

Avec Δ le pas de quantification, β =3, α = 2/$\log_2$e, σ² la variance du signal.

On peut ainsi exprimer la relation entre la distorsion et le nombre de bits de codage (Équation 1.20 → Équation 1.21):

$$R(D) = \frac{1}{\alpha} log_e\left(\frac{\sigma^2}{D}\right)$$

Équation 1.22. *Relation Bit/Distorsion*

D'où :

$$\frac{\partial R(D)}{\partial D} = -\frac{1}{\lambda} = -\frac{1}{\alpha} \times \frac{1}{D}$$

(Équation 1.22)→(Équation 1.19)

$$\lambda = \alpha \times D = \alpha \times \frac{\Delta^2}{\beta}$$

$$\lambda = c \times \Delta^2$$

(Équation 1.20)→(Équation 1.22)

D'après l'Équation 1.9, la relation liant le pas de quantification au paramètre de quantification QP suit une loi logarithmique à base 2, la relation entre le paramètre de quantification QP et le Lagrangien a ainsi été approximée sur une banque de séquences de manière à approximer la pente des courbes débit/distorsion.

1. 2. 4. 2. Notion de Contrôle de débit

Comme nous venons de le voir, le module de RDO a besoin de connaître le QP utilisé pour encoder l'image, pour calculer les fonctions coût-RD de chaque mode de prédiction et ainsi choisir le mieux adapté aux conditions d'encodage. Le QP utilisé pour une image est choisi par l'utilisateur ou le module de contrôle de débit en fonction du type d'encodage utilisé.

L'encodage à QP constant applique à chaque image le QP choisi par l'utilisateur qui n'a alors aucun contrôle sur le débit de sortie, celui-ci est variable et dépend de la complexité du contenu à encoder. Pour cette raison, l'encodage à QP constant est parfois appelé encodage VBR (Variable Bitrate).

L'autre type d'encodage consiste à utiliser un contrôle de débit qui adapte le paramètre de quantification en fonction de la complexité du contenu à encoder pour atteindre le débit choisi par l'utilisateur. Dans le monde industriel, l'encodage avec contrôle de débit est majoritairement utilisé car les utilisateurs ont une bande passante limitée dans laquelle ils veulent transmettre un ou plusieurs flux audiovisuels. Deux types d'encodage avec contrôle de débit sont utilisés : l'encodage CBR (Constant Bitrate) et ABR (Average Bitrate), ce dernier est aussi appelé VBR même si il est différent du cas QP constant évoqué ci-dessus. Les deux utilisent un contrôle de débit qui choisit le QP par image (ou macrobloc) pour atteindre le débit cible, mais l'encodage ABR est moins contraint que le CBR. Il nécessite en plus du débit cible, un deuxième paramètre : le débit maximum autorisé. L'encodage ABR est par conséquent moins précis sur le débit de sortie que le CBR, mais il introduit moins d'artefacts en autorisant les parties complexes d'une séquence à atteindre le débit maximum fixé. L'encodage CBR peut être vu comme un cas particulier de l'encodage ABR où le débit maximum autorisé est égal au débit cible.

Le principe d'un contrôle de débit est de choisir le paramètre de quantification par image (ou macrobloc) en fonction de la complexité d'encodage de l'image, mesurée typiquement par la MAD (Mean of Absolute Differences) qui est l'erreur de codage entre l'image originale et l'image reconstruite. L'erreur de codage dépend des modes d'encodage choisis par le module RDO, qui lui, a besoin de connaître la valeur du paramètre de quantification pour faire son choix : on parle du dilemme de l'œuf et de la poule (qui des deux était là en premier…) illustré par la Figure 1.27.

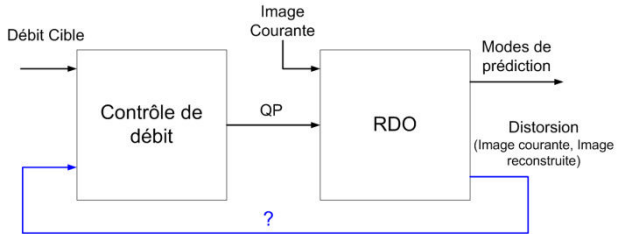


Figure 1.27. Illustration du dilemme RDO (Rate Distortion Optimization)-RC (Rate Control)

Pour résoudre ce problème, le contrôle de débit utilise une estimation de la complexité de l'image courante au lieu de la distorsion entre l'image courante et prédite. Un modèle simple de RDO et contrôle de débit est présenté par la Figure 1.28. Dans la suite de ce paragraphe, nous allons décrire ce modèle de contrôle de débit pour donner une vue globale qui nous permettra par la suite d'aborder un modèle plus complexe. Le module de contrôle de débit peut se décomposer en trois parties : l'allocation binaire, l'estimation de complexité et l'estimation du QP.

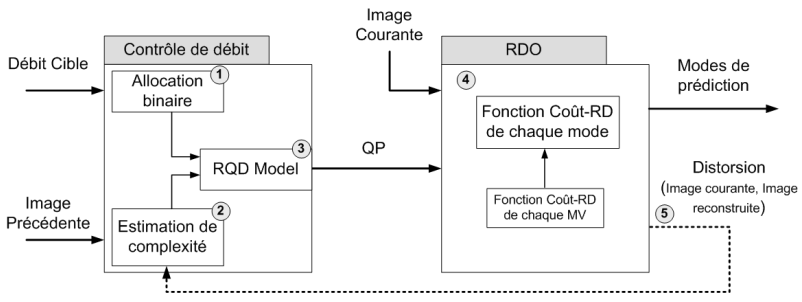


Figure 1.28. Modèle simple RC-RDO

- **Allocation Binaire**

L'Allocation binaire attribue à chaque unité élémentaire un budget en fonction du débit cible demandé par l'utilisateur, du frame-rate et de l'état de remplissage du buffer de décodage. L'allocation binaire peut s'effectuer à plusieurs niveaux : GOP, Frame, Slice et macrobloc. Un exemple simple d'allocation par image consiste à attribuer à chaque image un budget égal au débit cible divisé par le nombre d'images par seconde. Les images n'ont pas toutes la même complexité et le même type de codage (Intra ou Inter), elles ne consommeront par conséquent pas le même budget malgré l'adaptation du QP. Si le décodeur n'a pas suffisamment de données dans son buffer pour présenter les images en continu, on parle de *buffering underflow,* la séquence décodée sera figée. Au contraire, si le décodeur reçoit plus de données qu'il ne peut en traiter, on parle de *buffering overflow*, qui provoque des sauts d'image. Pour éviter ces comportements, le contrôleur de débit modélise l'état du buffer de décodage appelé HRD. En fonction du niveau de remplissage du HRD (Hypothetical Reference Decoder), le budget d'une image est ajusté.

- **Estimation de complexité**

La complexité d'encodage peut typiquement être estimée à l'aide des images précédemment encodées. L'erreur réalisée à l'image précédente mesurée en MAD (Mean of Absolute Differences), peut être utilisée pour prédire l'erreur d'encodage de l'image courante à l'aide d'un modèle linéaire [24] :

$$MAD_{pred}(t) = a_1 \times MAD(t-1) + a_2$$

Équation 1.23. *Modèle de prédiction de complexité*

Avec $MAD_{pred}(t)$ la MAD prédite de l'image courante, $MAD(t\text{-}1)$ la MAD réelle de l'image précédente, a1 et a2 les paramètres du modèle linéaire qui sont mis à jour à la fin de l'encodage de chaque image.

- **Estimation du QP (modèle RQD (Rate-Quantization-Distortion))**

Un modèle Débit-Distorsion-Quantification est utilisé pour estimer le QP de l'image courante à partir du budget binaire et de l'estimation de complexité. Le modèle quadratique suivant est couramment utilisé :

$$R = C_1 \times \frac{MAD_{pred}(t)}{QP} + C_2 \times \frac{MAD_{pred}(t)}{QP^2}$$

Équation 1.24. *Modèle Bit/Distorsion/QP*

Le QP ainsi prédit est utilisé par le module RDO pour choisir le mode optimal pour les macroblocs de l'image comme vu dans le paragraphe précédent 1. 2. 4. 1.

De la qualité du contrôle de débit va dépendre la qualité perçue de la séquence. Le modèle que nous venons de présenter est très simple, il faut bien comprendre que chaque implémentation du codec H.264/AVC porte un effort particulier à affiner son propre modèle de contrôle de débit.

1. 3. X264 – Une implémentation reconnue de la norme H.264/AVC

Dans le cadre de nos travaux sur le préfiltrage pour l'amélioration de la qualité d'encodage H.264/AVC, le besoin de coupler le prétraitement à l'encodeur a rapidement amené la nécessité d'avoir accès au cœur du codeur. Cependant l'encodeur software AQILIM de Digigram est basé sur l'implémentation de Rovi (Mainconcept) [2] fourni sous forme de librairie logicielle qui ne permet pas d'accéder au code source et d'insérer nos travaux au sein de l'encodeur. Le codec open source x264 [25] a été préféré à l'encodeur de référence JM [26] pour nos travaux, de par sa renommée et la fréquence de son utilisation dans le milieu industriel comme dans les logiciels grand public de transcodage de fichiers (ffmpeg entre autres). Le projet x264 est issu du groupe VideoLan (VLC) [27] initié par le développeur Laurent Aimar et aujourd'hui principalement dirigé par les développeurs Loren Merritt, Jason Garrett-Glaser. Dans cette section nous commencerons par présenter des études comparatives entre x264 et les autres principales implémentations professionnelles. Puis nous présenterons la structure de l'encodeur x264 et notamment l'implémentation de son contrôle de débit.

1. 3. 1. Etat de l'art des implémentations professionnelles

La qualité d'un encodeur H.264/AVC est variable et dépend entre autre de l'implémentation de son module de RDO, de contrôle de débit et d'estimation de mouvement (§1. 2. 4.). Cette section propose de comparer le codec x264 à d'autres implémentations professionnelles utilisées par le grand public et/ou dans le secteur industriel et d'ainsi valider le choix de l'implémentation x264 pour nos travaux sur le préfiltrage perceptuel.

1. 3. 1. 1. Comparatif de l'université de Moscou

Une étude comparative des différentes implémentations professionnelles des normes H.264/AVC et VP8[4] a été réalisée par le «Graphics & Media Lab Video group» du département « Computer Science » de l'université de Moscou en mai 2011 [28].

Les huit codecs présentés dans le Tableau 1.7 ont été comparés en termes de qualité/débit, temps d'encodage et qualité du contrôle de débit pour trois types d'applications (Vidéo-conférence, séquences SD et HDTV) et trois profils d'encodage appelés High Speed, Normal et High Quality. Les développeurs de chaque logiciel excepté XviD et DivX, ont fourni les paramètres des codecs pour chaque configuration testée. La Figure 1.29 présente les résultats de cette étude en moyenne et pour un cas particulier.

Caractéristiques de la plateforme de test	
Processeur	4-cores, Intel Core i7 920, 2.67GHz
Système d'exploitation	Microsoft Windows 7 Professional 64-bit
Mémoire Physique Totale	12 GB

Tableau 1.6. Etude du Graphics & Media Lab Video group - Caractéristiques de la Plateforme de test pour

Implémentation	Norme
DivX AVC/H.264 Video encoder (DivX)	H.264/AVC
Elecard AVC Video Encoder 8-bit edition (Elecard)	
Rovi (Mainconcept) AVC/H.246 Video Encoder Console Application	
Microsoft Expression Encoder 4 (MSE)	
X.264	
XviD raw mpeg4 bitstream encoder (Xvid)	
Discrete Photon	VP8
WebM VP8 codec	

Tableau 1.7. Etude du Graphics & Media Lab Video group - Codecs concernés

[4] Le codec VP8 a été développé en 2008 par la société On2 Technology, rachetée depuis par Google qui a intégré le codec au projet WebM dédié à créer un format multimédia libre et lisible nativement par les browsers internet. Le codec VP8 est très proche de la norme H.264/AVC et donne des performances similaires. Le nouveau codec VP9 open source concurrence aujourd'hui la norme HEVC.

Les graphiques a, b et c de la Figure 1.29 présentent trois des indicateurs principaux utilisés pour le classement global des codecs pour une application, une séquence et un profil particulier (HDTV, *BigBugBunny*, profile Normal). La comparaison entre le codec x264 (ligne verte) et le codec Rovi (Mainconcept) (ligne rose) nous intéresse particulièrement puisque les encodeurs vidéo software de Digigram sont basés sur ce codec.

- Le graphique débit-distorsion (a) présente la distorsion mesurée en SSIM[5] (Stuctural Similarity Index Metric) [29] en fonction du débit d'encodage. Le codec x264 offre la meilleure qualité sur la plage de débits testés. Le SSIM est une métrique de distorsion dite perceptuelle calculée ici uniquement sur la composante de luminance de la manière suivante :

$$MSSIM(X,Y) = \frac{1}{W \times H}\sum_{i=1}^{W}\sum_{j=1}^{H} SSIM(x(i,j), y(i,j)) \qquad \textit{Équation 1.25. MSSIM}$$

$$SSIM(x,y) = \frac{(2 \times \mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \qquad \textit{Équation 1.26. SSIM}$$

$$\mu_k = \frac{1}{N}\sum_{i=1}^{N} w_i * k_i$$

avec k = x ou y

$$\sigma_k^2 = \frac{1}{N-1}\sum_{i=1}^{N} w_i * \left(k_i - \overline{k}\right)^2$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N} w_i * (x_i - \mu_x)(y_i - \mu_y)$$

Avec	*X* et *Y*	Respectivement l'image originale (référence) et l'image traitée.
	x et *y*	pixel appartenant respectivement à l'image originale et à l'image traitée.
	μ_x et μ_y	Luminance moyenne autour d'un pixel appartenant respectivement à l'image originale et à l'image traitée.
	σ_x et σ_y	Variance autour d'un pixel appartenant respectivement à l'image originale et à l'image traitée.
	σ_{xy}	Covariance
	w_i	Fenêtre gaussienne 11x11 d'écart-type 1.5 utilisée pour le calcul de la moyenne et de la variance.

- Dans le graphique (b), la vitesse d'encodage est représentée en nombre d'images traitées par seconde. On note qu'avec la plateforme de tests dont les caractéristiques sont données dans le Tableau 1.6, le profil d'encodage «normal » de Rovi (Mainconcept) ne permet pas de réaliser l'encodage en temps réel d'une séquence HD, alors que x264 permet d'encoder 25 images par seconde en dessous de 4Mbit/s.

- Le graphique (c) représente le débit réel en fonction du débit cible, ainsi les codecs se rapprochant de 1 ont le meilleur contrôle de débit. On note que l'implémentation de Rovi (Mainconcept) utilise un contrôle de débit moins précis que x264.

La Figure 1.29 (d) présente les résultats de qualité/débit moyennés sur toutes les séquences, pour les trois applications (Vidéo-conférence, séquences SD et HDTV) et les trois profils (High Speed, Normal et High Quality). Les résultats des différents codecs sont représentés en termes de pourcentages de débit comparativement au codec XviD à même qualité. X264 apparaît comme le codec donnant le plus faible débit à même qualité comparativement à tous les autres codecs testés. Plus largement, cette étude place le codec x264 comme le meilleur codec en tenant compte de tous les indicateurs (qualité, latence et de précision du contrôle de débit).

[5] Par abus de langage lorsqu'on parle de SSIM, on fait référence au MSSIM (Mean SSIM) qui représente la moyenne des SSIM sur l'image.

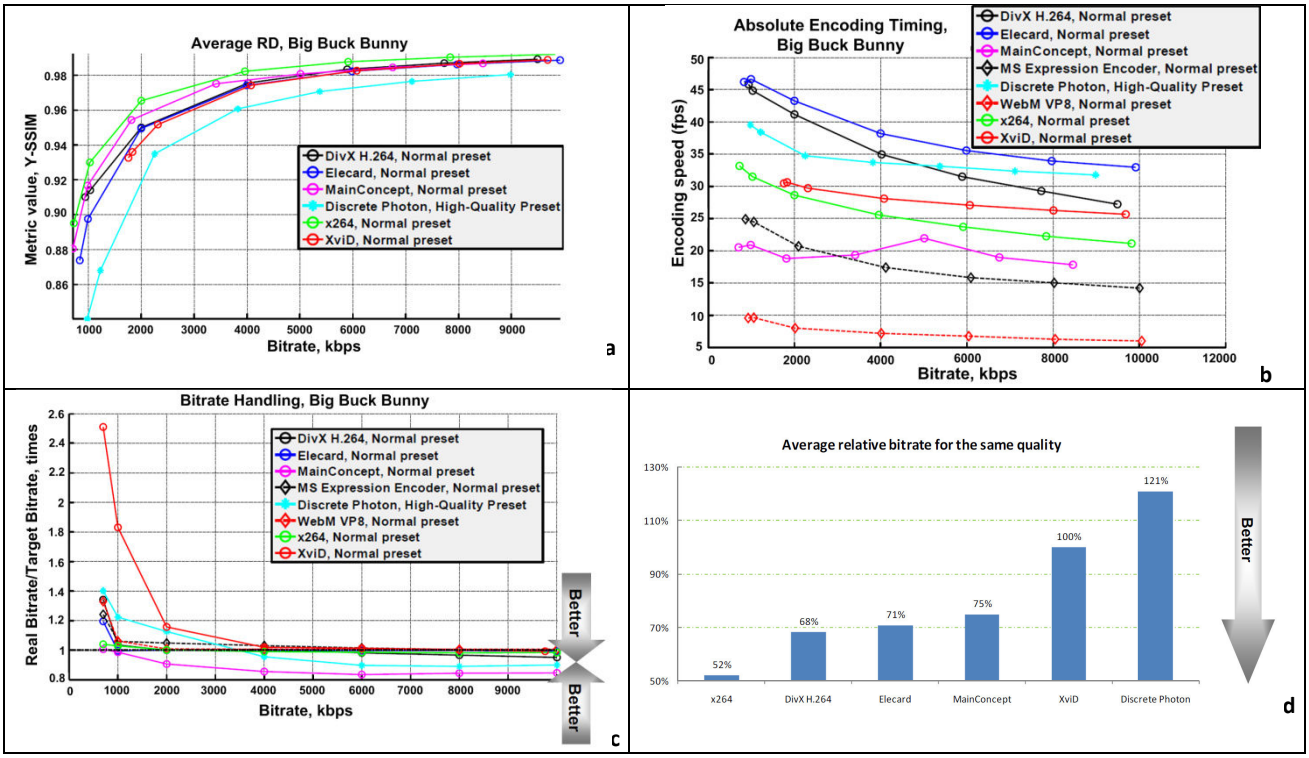


Figure 1.29. Résultats de l'étude comparative menée par l'université de Moscou
Illustration issue de [28]
(a) Courbe débit-distorsion (uniquement pour les encodeurs qui remplissent les exigences en termes de vitesse d'encodages), séquence BigBuckBunny, application HDTV, Profil Normal - (b) Vitesse d'encodage en fonction du débit cible, séquence BigBuckBunny, application HDTV, Profil Normal - (c) Débit réel en fonction du débit attendu, séquence BigBuckBunny, application HDTV, Profil Normal – (d) Débit moyen relatif à qualité constante entre tous les codec

1. 3. 1. 2. Comparatif x264 – AQILIM (Digigram)

Afin que nos travaux sur le codec x264 se rapprochent au plus près des conditions d'utilisation des clients de Digigram, nous avons mené une étude comparative entre les codecs x264 et Rovi (Mainconcept), pour s'assurer que le codec x264 est de qualité comparable à l'implémentation Rovi (Mainconcept).

Le codec Rovi (Mainconcept) propose un paramètre de réglage des performances qualité/complexité appelé « Video Performance » et pouvant varier de 1 à 15, 15 étant la meilleure qualité. Chaque valeur du paramètre « Video Performance » correspond à un jeu de paramètres décrit par la norme H.264/AVC. Ce paramètre peut être vu comme une couche d'abstraction qui permet aux utilisateurs de régler la qualité d'encodage sans avoir la connaissance de la norme H.264/AVC. Les clients de Digigram jugent la qualité des encodeurs AQILIM bonne lorsque le paramètre « Video Performance » est égal à 12 (VP12), c'est donc notre qualité de référence.

Le codec x264 propose quant à lui dix profils de vitesse nommés : *ultrafast, superfast, veryfast, faster, fast, medium, slow, slower, veryslow, placebo*. Les performances de vitesse et la qualité étant étroitement liées, le profil *ultrafast* donne une qualité médiocre, alors que le profil *veryslow* donne une bonne qualité.

La comparaison entre Rovi (Mainconcept) et x264 a été réalisée sur une plateforme à 8 cores XEON E5405 2.00GHz dédiée à la thèse, de performances inférieures aux produits AQILIM commercialisés.

La Figure 1.30 présente le PSNR (lignes pleines) et le SSIM (lignes pointillées) en fonction du nombre d'images encodées par seconde pour la séquence *ParkJoy* 1280x720 50p. Le profil d'encodage de Rovi (Mainconcept) VP12 (point jaune) que nous prenons pour référence, n'est pas encodé en temps réel sur cette plateforme, mais il faut noter que les produits AQILIM SERV/FIT sont eux basés sur une plateforme 8 cores XEON E5620 2.40GHz qui permet l'encodage d'une résolution 1280x720p 50fps en temps réel.

D'après le SSIM, tous les profils x264 sont de meilleure qualité que les profils Rovi (Mainconcept). Cependant, d'après le PSNR, seul les profils rapides (*faster* et au-delà) présentent une meilleure qualité que Rovi (Mainconcept).

Ce test démontre que l'implémentation open source x264 est de qualité comparable voire meilleur qu'une solution reconnue du monde professionnel, et valide ainsi le choix de ce codec pour la suite de nos travaux.

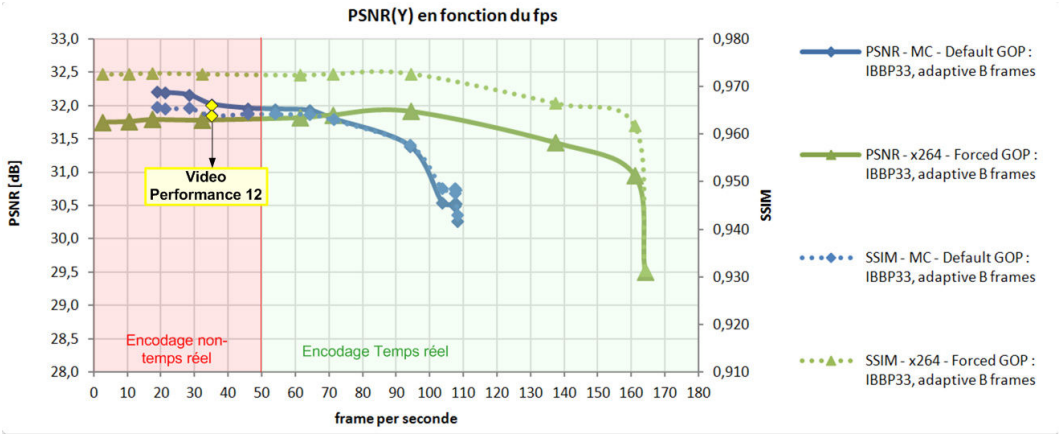


Figure 1.30. Comparaison des profils x264 et Rovi (Mainconcept) – Séquence ParkJoy

1. 3. 2. Présentation de l'architecture x264

Dans cette section nous allons présenter les étapes principales de l'encodage x264 avec contrôle de débit (Figure 1.32) dans le cas particulier du CBR (Constant Bitrate). Le contrôle de débit x264 est basé sur la librairie logicielle libavcodec développée dans le cadre du projet FFmpeg. Ce contrôle de débit est principalement empirique, la description qui suit est entièrement basée sur l'analyse du code x264 ainsi que sur l'article du développeur Lorent Merrit [30]. Il faut noter que nous donnons ici une vue d'ensemble du codec x264(CBR) et que certains outils spécifiques x264 jugés marginaux ou hors de notre contexte d'étude ne sont volontairement pas présentés.

Les étapes principales du codec x264 sont présentées par la Figure 1.32. L'architecture de l'encodeur x264 fonctionne en trois phases (Figure 1.31) : Une phase d'analyse permettant d'initialiser les paramètres d'encodage pour l'image courante, une phase d'encodage et une phase de régulation permettant d'ajuster les paramètres d'encodage pour le macrobloc/Image suivante.

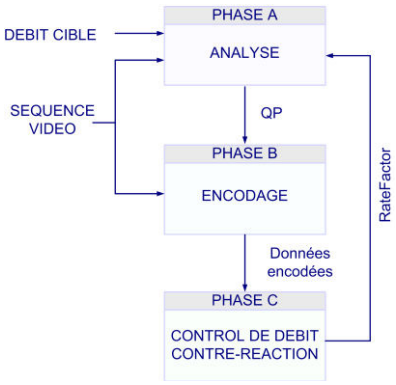


Figure 1.31. Schéma de principe x264

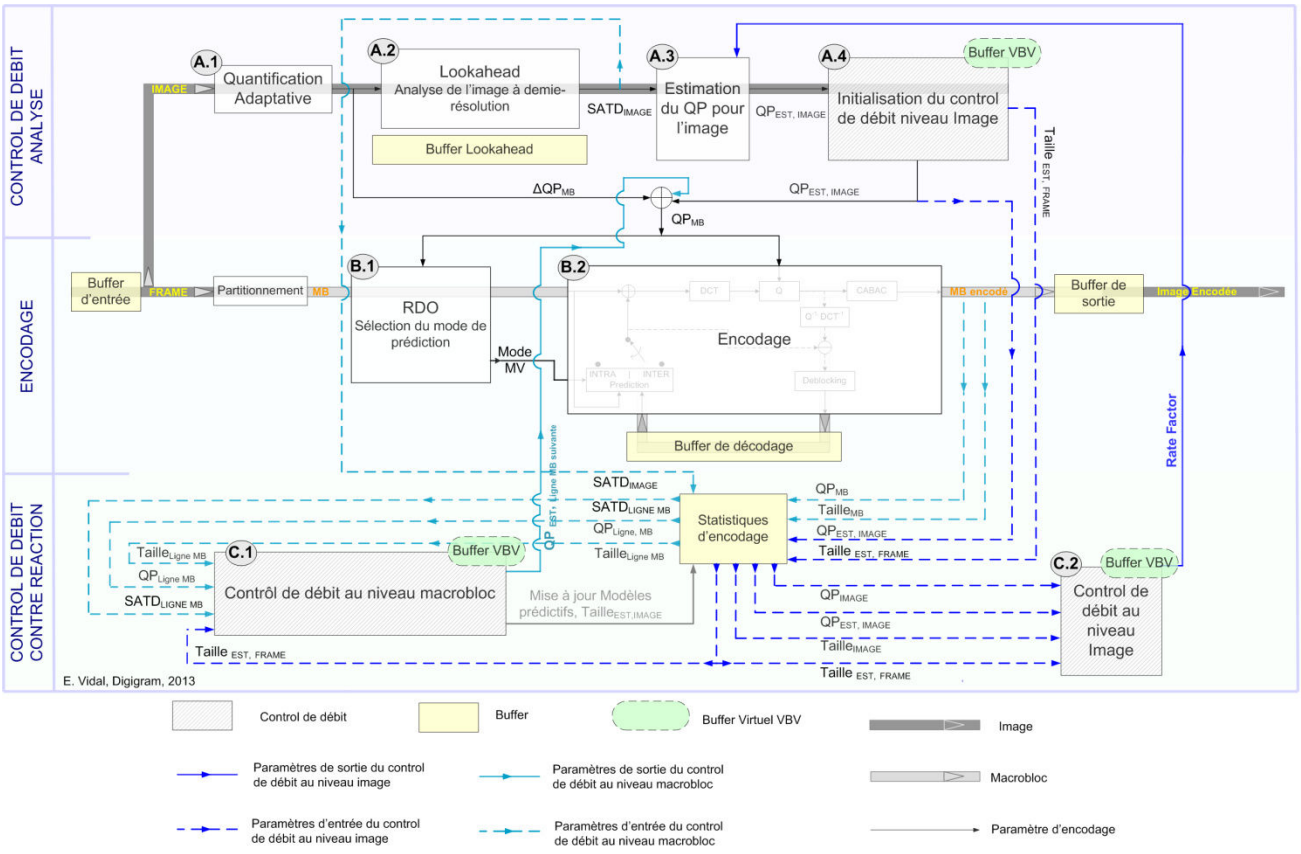


Figure 1.32. Architecture simplifiée de l'encodeur x264

1. 3. 2. 1. **HRD Model**

Comme nous l'avons évoqué au paragraphe 1. 2. 4. 2. , la norme H.264 décrit le comportement du décodeur de référence appelé HRD, présenté par la Figure 1.33. Le module HSS (Hypothetical Stream Scheduler) gère l'arrivée des NALUs. Lorsque toutes les NALUs d'une image sont reçues, l'AU (Access Unit) correspondante peut être décodée. Les images décodées sont stockées dans le buffer de décodage DPB pour servir de référence au décodage des prochaines images et être présentées aux utilisateurs après un éventuel réordonnement. Pour être conforme à la norme, un contrôle de débit doit produire un flux de NALU qui ne provoque pas de débordement (overflow) et alimente suffisamment le buffer CPB (underflow). Pour s'en assurer l'encodeur modélise l'état de ce buffer au fil de l'encodage.

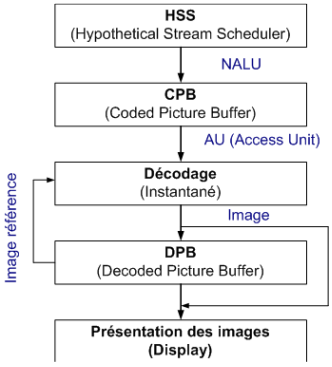


Figure 1.33. Model HRD H.264/AVC

Dans l'implémentation x264, l'état du buffer CPB du HRD est modélisé à l'encodeur par un buffer virtuel[6] appelé VBV (Video Buffer Verifier). Lorsqu'on réalise un encodage en CBR, deux paramètres x264 sont à fixer par l'utilisateur : la taille du buffer VBV (Video Buffer Verifier) et le débit cible (le débit maximum étant égal au débit cible en CBR). La taille du buffer VBV a une influence sur la latence et la qualité d'encodage. Plus sa taille est importante, plus le temps de mise en mémoire (buffering) est grand avant que le décodage ne commence. A l'encodeur, la taille du buffer d'analyse est fonction de la taille du buffer VBV, du GOP et du nombre d'images par seconde. Par conséquent

[6] Un buffer est un espace mémoire réservé pour stocker temporairement des données. On emploie ici le terme buffer virtuel car aucun espace mémoire n'est réservé à l'encodeur pour le buffer VBV, son état de remplissage est simplement estimé au fil de l'encodage.

plus le buffer VBV est grand, plus la latence du système augmente mais en contrepartie plus l'encodeur dispose d'images à analyser pour choisir les paramètres avant de commencer l'encodage. Pour réaliser une transmission CBR live, une taille d'une seconde (soit 50 images pour un format 50p) est un bon compromis entre latence et stabilité du débit.

1. 3. 2. 2. **Phase d'analyse**

La première phase permet d'analyser l'image courante ainsi qu'un nombre fixé d'images futures pour choisir les paramètres d'encodage les plus adaptés au débit cible et au contenu de la séquence. Le choix du type d'image peut être laissé à l'encodeur qui le décidera pendant cette phase d'analyse, on parle de GOP Adaptatif. Toutefois nous considèrerons pour la suite que le GOP est fixé par l'utilisateur et nous nous concentrerons sur le choix du paramètre de quantification au début de l'encodage de chaque image.

- **Quantification Adaptative – Etape A.1**

Le terme quantification adaptative se réfère à l'utilisation d'un paramètre de quantification par macrobloc et non par image. Le module de quantification adaptative de x264 calcule un offset appelé ΔQP_{MB} par macrobloc dans la Figure 1.32 qui sera par la suite ajouté au QP estimé pour l'image ($QP_{EST,IMAGE}$) afin d'obtenir une carte de QP par macrobloc. L'idée est de tenir compte du fait qu'une image n'est pas uniformément complexe et qu'on peut quantifier plus grossièrement certaines parties pour en préserver d'autres. La quantification adaptative x264 est basée sur la variance des macroblocs, plus la variance est grande, plus la quantification est grossière. Une discussion sur la signification de la quantification contôlée par la variance des macroblocs sera menée au Chapitre 4.

$$\Delta QP_{MB} = strength * log_2(max[variance_{MB}, 1]) - (14.427 + 2 * (BIT\_DEPTH - 8)) \quad \textit{Équation 1.27 Quantification Adaptative}$$

Avec

BIT_DEPTH	:	profondeur de codage, dans la majorité des cas égal à 8
Strength	:	Force de quantifictaion, réglable par l'utilisateur, par défaut 1.0397

- **Lookahead – Etape A.2**

Le module Lookahead (traduire regarder en avance) est la spécificité principale de l'implémentation x264. La philosophie de ce module est d'approcher la qualité d'un encodage double passe, en estimant finement la complexité des images à l'aide d'un premier encodage à bas coût [31], là où le modèle classique présenté précédemment (1. 2. 4. 2.), utilise un simple modèle linéaire fonction de la MAD (Mean Absolute Difference) de l'image précédente. Durant l'analyse, la résolution d'une image est divisée par deux, une prédiction inter et intra simplifiée est réalisée pour chaque macrobloc réduit (8x8). Trois modes intra sont testés (DC, Horizontal et vertical), et quatre estimations inter, dont une à partir d'une image de référence passée, une à partir de deux images de référence appartenant aux listes passées et futures, le mode « temporal direct » (l'estimation de mouvement n'est pas réalisée et les vecteurs mouvements sont estimés à partir des macroblocs voisins), et le mode « zero motion » (le vecteur mouvement est nul). Finalement, une transformée d'Hadamard est appliquée sur l'erreur de prédiction et la somme de l'amplitude des coefficients d'Hadamard est utilisée comme mesure de complexité appelée SATD (Sum of Absolute Transform Difference).

Plus le nombre d'images que le Lookahead peut analyser avant de commencer l'encodage est grand, plus le contrôle de débit dispose d'estimations précises sur le contenu de la séquence mais plus la latence augmente. La taille du buffer Lookahead peut être précisée par l'utilisateur ou calculée en fonction des caractéristiques de la séquence et de l'encodage.

- **Estimation du QP pour l'image – Etape A.3**

Le QP est estimé en début d'encodage pour toutes les images P, le QP des images B et I est basé sur le QP des images P [30]. L'estimation du pas de quantification ($q_{EST,\ IMAGE}$) à appliquer à l'image courante est fonction de la complexité d'encodage (*blurred_cplx*) et de l'erreur d'estimation réalisée à l'image précédente *RF* (Rate Factor) :

$$q_{EST,IMAGE,init} = complexity^{1-qcompress}$$

$$q_{EST\_IMAGE} = \frac{q_{EST,IMAGE,init}}{RF}$$

Équation 1.28 *Estimation du pas de quantification*

La relation entre le pas de quantification et le QP est approximé par :

$$QP = 12 + 6 \times log_2\left(\frac{q}{0.85}\right)$$

Équation 1.29 *Relation entre QP et pas de quantification*

Le pas de quantification estimé en début de chaque image dépend de la complexité des images encodées depuis le début de la séquence (*complexity*). La complexité d'encodage d'une image (*frame_cplx*) est estimée par la $SATD_{IMAGE}$ calculée à l'étape précédente et le temps disponible pour encoder une image qui est simplement l'inverse du débit image (fps) dans le cas d'une séquence progressive (Équation 1.32). La complexité de la séquence (*complexity)* est obtenue en accumulant la complexité de chaque image comme le montre l'Équation 1.30.

$$complexity = \frac{\text{short\_term\_cplxsum}}{\text{short\_term\_cplxcount}}$$

Équation 1.30. *Complexité estimée de la séquence*

- $\text{short\_term\_cplxsum}_t = 0.5 \times \text{short\_term\_cplxsum}_{t-1} + frame\_cplx_t$
- $\text{short\_term\_cplxcount}_t = 0.5 \times \text{short\_term\_cplxcunt}_{t-1} + 1$

$$frame\_cplx = \frac{SATD_{IMAGE}}{frame\_duration}$$

Équation 1.31. *Complexité estimée d'une image*

$frame\_duration = \frac{nbr\_field}{2 \times fps}$, pour une image progressive nbr_field = 2

Équation 1.32. *Temps disponible pour encoder une image*

RF (*Rate Factor*) représente l'erreur commise sur la taille de la séquence encodée, elle dépend de la taille de la séquence planifiée par le contrôle de débit ($Taille_{EST,SEQ}$) et de la taille réelle après encodage ($Taille_{SEQ}$). Le calcul détaillé du RF est présenté au paragraphe 1. 3. 2. 4. – Etape C2.

- **Initialisation du control de débit – Etape A.4**

Le choix du QP fait à l'étape précédente est affiné en considérant les aspects de contrôle de débit pour s'assurer qu'il permettra de respecter le débit cible. Pour cela, un modèle paramétrique permettant de prédire la taille d'une image en bits en fonction de sa complexité (SATD) et du QP choisi est utilisé :

$$predicted\_size_{Type} = \frac{coeff_{Type} \times SATD_{Type} + offset_{Type}}{QP_{EST} \times count_{Type}}$$

Équation 1.33. *Modèle de prédiction de la taille de données encodées*

Avec *coeff, offset* et *count,* les paramètres du modèle qui sont initialisés au début de l'encodage et mis à jour toutes les lignes de macroblocs en fonction de la SATD de la ligne et du nombre de bit dépensé pour la coder. Il existe trois jeux de paramètres *coeff, offset* et *count* pour les trois types d'image (I, P et B).

La taille des images futures disponibles pour l'analyse, est prédite à l'aide du modèle paramétrique en appliquant le QP choisi à l'étape précédente. En comparant la taille prédite de chaque image à la taille théorique d'une image, l'état du buffer VBV est analysé pour contraindre le choix du pas de quantification. A la fin de l'encodage d'une image, le nombre de bits réellement utilisé pour encoder l'image courante est utilisé pour mettre à jour l'état du buffer VBV.

La Figure 1.34 présente un exemple de prédiction de l'état du buffer VBV de taille 10 images avec un GOP de type IBBP. Δb représente la différence entre la taille prédite d'une image et la taille théorique. La taille théorique d'une image est constante et égale au débit cible divisé par le nombre d'image par seconde. En réalité la taille des images varie en fonction du contenu et surtout du type d'encodage, une image I étant plus consommatrice de bits qu'une P et qu'une B. Lorsqu'on parle de débit constant (CBR) on considère une seconde d'encodage durant laquelle les différences de tailles d'images se compensent. Après avoir virtuellement rempli le buffer VBV avec la taille prédite des images, le niveau de remplissage du buffer VBV est analysé. Si celui-ci est rempli à moins de 20% ou à plus de 50%, le pas de quantification prédit à l'étape précédente est augmenté ou diminué itérativement jusqu'à respecter ces critères.

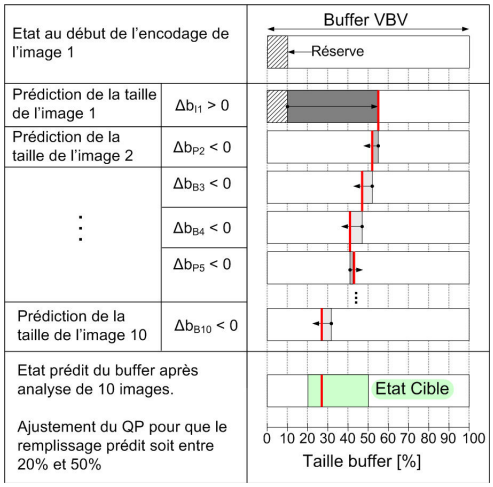


Figure 1.34. Exemple de prédiction de l'état du buffer VBV au début de l'encodage d'une image

A la fin de cette étape on dispose du pas de quantification (et donc le QP) pour l'image courante, l'encodage de l'image peut ainsi commencer avec les paramètres décidés durant cette première phase.

1. 3. 2. 3. **Phase d'encodage**

L'image est encodée par macroblocs 16x16 à partir du QP choisi à l'étape précédente auquel sont ajoutés les ΔQPs calculés par le module de quantification adaptative.

- **RDO – Etape B.1**

La phase de RDO (Rate-Distorsion Optimization) permet de choisir la meilleure découpe et le meilleur mode intra ou vecteur mouvement inter, pour encoder le macrobloc courant. On rappelle qu'un macrobloc Intra peut-être prédit à partir de treize modes (quatre modes 16x16 et neuf modes 4x4) et qu'un macrobloc 16x16 inter peut-être découpé en blocs 16x8, 8x16 et 8x8 pouvant lui-même être redécoupé en blocs 8x4, 4x8 et 4x4. Pour chacune de ces découpes, une estimation de mouvement doit-être réalisée avec plusieurs images de référence possibles. Afin d'accélérer le traitement, tous les modes inter ne sont pas testés. Par exemple, l'estimation de mouvement est d'abord réalisée sur les macroblocs 16x16 et sur la découpe 8x8 à partir de toutes les références disponibles. Si le coût débit-distorsion (RD) de la découpe 8x8 est plus fort que celui du macrobloc entier, alors les découpes 16x8 et 8x16 ne sont pas testées.

La complexité de cette phase peut être choisie par l'utilisateur. Dans une première approche le coût RD de chaque possibilité de prédiction est estimé par la SATD0 :

$$SATD0 = SATD(MB_{Orig}, MB_{pred}) + \lambda \times bit\_pred$$

Équation 1.34.
SATD coût RD prédit

Avec *SATD* la somme des différences des coefficients d'Hadamard entre le macrobloc original et prédit, λ le multiplicateur de Lagrange donné par une table dépendant du QP et *bit_pred* le nombre de bits nécessaires au codage du/des mode(s) intra ou vecteur(s) mouvement inter du macrobloc avec un code Exponentiel-Golomb [3].

Si la configuration de l'encodage choisi par l'utilisateur l'autorise, le choix précédent est affiné en calculant le coût-RD réel des possibilités de prédictions ayant un score SATD0 inférieur à un seuil prédéfini. Le calcul du coût RD réel nécessite de réaliser un premier encodage du macrobloc :

$$RD = SSD(MB_{Orig}, MB_{pred}) + \lambda \times bit$$

Équation 1.35.
SATD coût RD prédit

Avec SSD (Sum of Squared Differences) la Somme des différences quadratiques entre le macrobloc original et prédit, λ le multiplicateur de Lagrange et bit le nombre de bits en sortie du codeur entropique (CAVLC ou CABAC).

A la fin de cette étape, le mode de prédiction Intra ou Inter le plus performant pour le macrobloc courant est sélectionné et l'étape d'encodage (Etape B.2 - Figure 1.32) à proprement parler commence. Cette étape est conforme au schéma bloc H.264/AVC que nous avons présenté dans la première partie de ce chapitre, nous n'y reviendrons donc pas ici. A la sortie du module d'encodage, on dispose du train binaire correspondant au macrobloc courant.

1. 3. 2. 4. Phase de rétro-action

Les résultats de l'encodage sont analysés à chaque fin de ligne de macroblocs et à la fin d'une image pour contraindre le paramètre de quantification afin de respecter le débit cible.

- **Contrôle de débit au niveau ligne de macrobloc – Etape C.1**

Le module de contrôle de débit au niveau ligne de macroblocs vérifie au fil de l'encodage d'une image que le nombre de bits consommés respecte le critère de débit et modifie le QP appliqué à la ligne de macroblocs suivante si nécessaire. Ce module se décompose en trois phases présentées par la Figure 1.35.

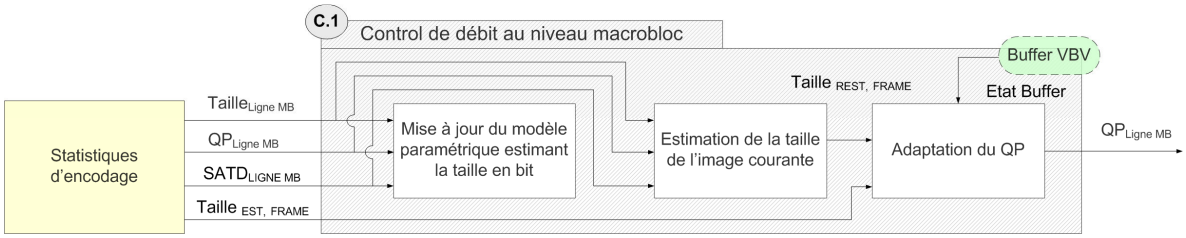


Figure 1.35. Module de contrôle de débit au niveau ligne de macrobloc

A la fin de chaque ligne de macroblocs, le modèle paramétrique permettant de prédire une taille de données encodées en bits à partir d'une complexité (SATD) et d'un QP (Équation 1.33), est mis à jour à partir du nombre réel de bits utilisés pour coder la ligne et de sa $SATD_{LIGNE, MB}$.

Puis la taille de l'image en cours est ré-estimée ($Taille_{REST, FRAME}$) et comparée à la taille planifiée au début de l'encodage de l'image ($Taille_{EST, FRAME}$). La taille ré-estimée à chaque fin de ligne de macroblocs contient à la fois le nombre de bits réellement utilisés pour coder les lignes précédentes, et une prédiction des lignes futures.

Si la taille ré-estimée est plus importante que la taille estimée initialement ou si le remplissage du buffer VBV n'est pas compris entre 20 et 50%, le QP est incrémenté/décrémenté de 0.5 de façon itérative jusqu'à remplir les contraintes imposées par le contrôle de débit.

- **Contrôle de débit au niveau image – Etape C.2**

A la fin de l'encodage d'une image, une dernière étape de contrôle de débit permet à la fois de régler la taille finale en ajoutant des paquets de remplissage remplis de zéros (padding) si l'image est de taille trop faible, ainsi que d'analyser la qualité du contrôle de débit pour calculer le paramètre RF (Rate Factor) qui sera appliqué à l'image suivante.

Le Rate Factor représente l'erreur commise sur la taille de la séquence encodée, elle dépend de la taille de la séquence planifiée par le contrôle de débit ($Taille_{EST,SEQ}$) et de la taille réelle après encodage ($Taille_{SEQ}$). En CBR, la taille

cible d'une image correspond simplement au débit cible multiplié par le temps disponible pour encoder une image (*frame_duration*). Ainsi la taille planifiée de la séquence au moment d'encoder l'image t ($Taille_{EST,SEQ}(t)$) est obtenue en accumulant la taille cible de toutes les images encodées depuis le début de la séquence (Équation 1.37). De la même manière, la taille réelle de la séquence après l'encodage de l'image t ($Taille_{SEQ}(t)$) est obtenue en accumulant la taille des images encodées ($Taille_{IMAGE}(t)$) pondérée par le rapport entre le pas de quantification estimé en début d'encodage ($q_{EST,IMAGE,init}$) et le pas de quantification moyen réellement appliqué à la séquence (q_{IMAGE}) (Équation 1.38).

Si la taille de l'image et/ou le QP ont été sous-estimés, le RF permettra d'augmenter le pas de quantification de l'image suivante (et inversement).

$$RF = \frac{\text{Taille}_{\text{EST,SEQ}}}{\text{Taille}_{\text{SEQ}}}$$

Équation 1.36. *RateFactor*

- $Taille_{EST,SEQ}(t) = \left(Taille_{EST,SEQ}(t-1) + frame\_duration * bitrate\right)$

Équation 1.37. *Taille planifiée de la séquence à encoder*

- Images P et I : $\text{Taille}_{\text{SEQ}}(\text{t}) = \left(\text{Taille}_{\text{SEQ}}(\text{t}-1) + \text{Taille}_{\text{IMAGE}}(\text{t}) * \frac{\text{q}_{\text{EST,IMAGE,init}}}{\text{q}_{\text{IMAGE}}}\right)$

  Image B : $\text{Taille}_{\text{SEQ}}(\text{t}) = \left(\text{Taille}_{\text{SEQ}}(\text{t}-1) + \text{Taille}_{\text{IMAGE}}(\text{t}) * \frac{\text{q}_{\text{EST,IMAGE,init}}}{\text{q}_{\text{IMAGE}} * \text{pb\_ratio}}\right)$

  pb_ratio : Rapport entre le QP appliqué aux images P et B. Permet de forcer les images B à être quantifiées plus fortement que les P, par défaut égal à 1.3.

Équation 1.38. *Taille réelle de la séquence*

1. 4. Conclusion

Dans ce premier chapitre, nous avons présenté les principes de l'encodage vidéo, puis la norme H.264/AVC largement déployée dans le secteur de la vidéo professionnelle, pour enfin présenter les principales innovations introduites par le standard HEVC qui fait l'objet d'importants efforts d'implémentation chez les fabricants d'encodeurs vidéo. Nous avons ensuite présenté un comparatif de différentes implémentations reconnues de la norme H.264/AVC, pour nous concentrer sur le codec x264 qui présente les meilleures performances.

Nous avons porté une attention particulière aux modules de RDO et de contrôle de débit qui ne font pas partie de la norme et qui font toute la différence de qualité entre les encodeurs du marché. Nous notons que la connaissance acquise sur l'implémentation du codec x264 et plus largement sur le fonctionnement d'un contrôle de débit réputé, est un gain de compétences important pour Digigram.

La suite de nos travaux que nous présenterons dans les prochains chapitres, vise à améliorer la qualité perçue de l'encodage H.264/AVC en se concentrant sur les implémentations Rovi (Mainconcept) et x264.

Chapitre 2. Proposition d'un préfiltre perceptuel et application au codage H.264/AVC

2. 1. Introduction

Les encodeurs software AQILIM proposés par Digigram sont dédiés au marché de la distribution caractérisé par une liaison entre un encodeur et un nombre indéterminé d'utilisateurs finaux, visionnant la vidéo sur des terminaux de différents types (ordinateur, tablette, smartphone). Dans une volonté de faire évoluer les encodeurs softwares AQILIM, Digigram a souhaité apporter des améliorations à ses solutions d'encodage en intégrant des solutions propriétaires de prétraitement/post-traitement. L'objectif final est d'apporter une plus-value par rapport aux autres solutions du marché. Ce prétraitement devra en effet permettre d'améliorer la qualité à débit constant et de réduire le débit à qualité équivalente, comme le montre la Figure 2.1 Cette figure représente le débit en fonction d'un critère de qualité pour une séquence encodée avec et sans prétraitement. L'intérêt des prétraitements à tendance à se réduire à bas débit car l'étape de quantification réalise une réduction grossière du contenu haute fréquence laissant peu de place à des traitements fins.

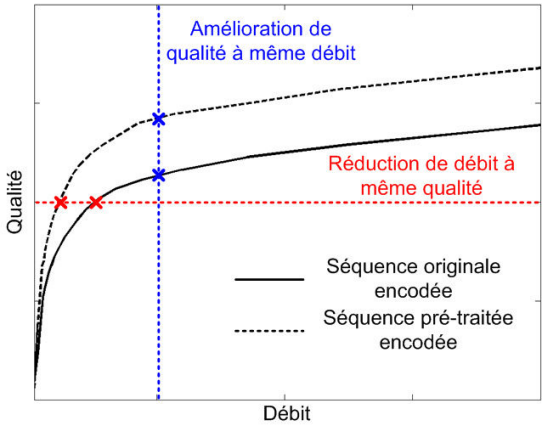


Figure 2.1. Illustration des objectifs de l'étude

Dans la littérature, les travaux visant à réduire le débit ou à améliorer la qualité des séquences encodées peuvent être regroupés en trois catégories : Les prétraitements, le codage perceptuel et les post-traitements. Les thèmes que nous aborderons dans ces trois catégories sont représentés par la Figure 2.2. Les prétraitements visent à réduire le contenu haute-fréquence des séquences vidéo pour améliorer les performances d'encodage. Ces traitements peuvent être indépendants de l'encodeur, agir en collaboration ou être embarqués dans l'encodeur. D'autres travaux cherchent à introduire un modèle perceptuel au sein du codeur pour optimiser le processus d'encodage, on parle alors de codage perceptuel. Enfin, des post-traitements cherchent à réduire les artefacts de codage. A la manière du filtre de réduction de l'effet de bloc (Deblocking) de la norme H.264/AVC, ces filtres se trouvent le plus souvent à l'encodeur (in-loop) et au décodeur.

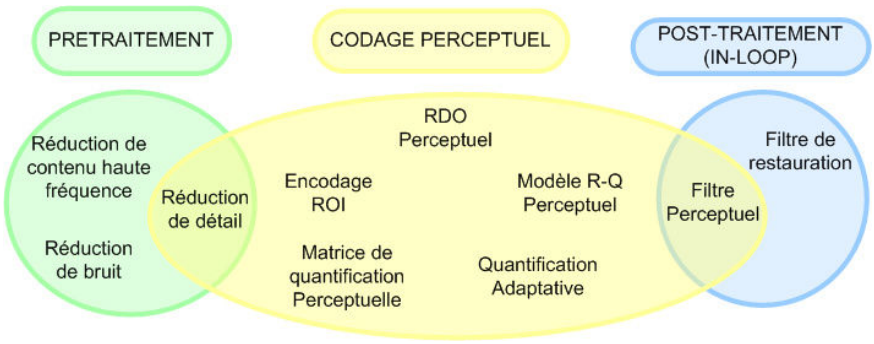


Figure 2.2. Thèmes de l'état de l'art

Contrairement à la gamme de produits dédiés au marché de la Contribution où l'encodeur et le décodeur sont vendus en couple pour une liaison point à point, pour le marché de la Distribution ce sont les terminaux des utilisateurs qui réalisent le décodage. Ainsi il est essentiel que le flux transmis par l'encodeur AQILIM soit conforme à la norme H.264/AVC et que la qualité de la séquence vidéo visualisée ne dépende pas du processus de décodage. Les travaux

nécessitant des modifications ou adaptations au décodeur n'ont par conséquent pas retenu notre attention. Les encodeurs vidéo AQILIM sont basés sur des cœurs de codage fournis par une tierce partie. N'ayant pas accès au cœur de codage, la solution du préfiltrage indépendant de l'encodeur a orienté nos travaux de recherche durant la première partie de la thèse.

De plus, les encodeurs de Digigram réalisent du streaming live, bien que la latence ne soit pas un critère déterminant du marché de la diffusion, la question de la latence apportée par l'ajout d'un préfiltre au système d'encodage reste importante. Ainsi, la question de l'implémentation temps-réel de nos solutions a guidé nos choix tout au long des travaux dans l'optique d'une prochaine intégration dans la gamme d'encodeurs live.

Dans ce chapitre nous commençons par présenter l'état de l'art des prétraitements ainsi que les raisons qui ont guidé nos choix vers le prétraitement perceptuel. Ensuite, nous présenterons un état de l'art des filtres de réduction de bruit ainsi que le filtre AWA (Adaptive Weighted Averaging) sur lequel porte nos travaux. Puis nous présenterons le modèle perceptuel JND (Just Noticeable Distortion) retenu pour notre étude pour enfin définir le préfiltre perceptuel que nous proposons. Celui-ci s'appuie sur le filtre AWA précédemment décrit et utilise un critère basé sur le JND pour adapter la force de filtrage selon les caractéristiques locales de l'image. Nous détaillons le mode de fonctionnement du filtre puis présentons les résultats expérimentaux qui ont nécessité le déploiement au sein de la société Digigram d'une salle de tests subjectifs normalisée. Les résultats montrent que l'application du préfiltre proposé permet à qualité constante, des réductions de débit en sortie de l'encodeur de l'ordre de 6% avec un maximum de 17% en moyenne pour des formats SD et de 5% en moyenne avec un maximum de 14% pour des formats HD.

2. 2. Etat de l'art des prétraitements pour l'encodage vidéo

Dans un contexte de bande passante limitée, l'optimisation de la qualité visuelle des vidéos encodées reste aujourd'hui une problématique essentielle. Dès les débuts du codec MPEG2, l'intérêt des prétraitements a largement été démontré dans le cas de sources dégradées par un bruit additionnel introduit entre autre par les capteurs photosensibles des caméras. Ces traitements visant à réduire le bruit présent dans les sources vidéo, se sont complexifiés avec le temps, passant de simples filtres passe-bas appliqués aux images d'une séquence, à des filtres à 3 dimensions couplés avec l'encodeur. De plus, les prétraitements ont pris en compte des critères perceptuels afin d'optimiser la qualité perçue par l'utilisateur final. Avec le standard de compression vidéo H.264, les outils d'encodage se sont affinés, laissant moins de possibilités d'amélioration aux prétraitements classiques, cependant l'attention portée aux prétraitements est toujours considérable.

Dans la suite de ce paragraphe, on différencie les différents prétraitements en trois groupes : Les préfiltres pour la réduction de bruit, de contenu haute-fréquence et de contenu perceptuellement non-significatif.

Chacun de ces trois groupes compte quatre types d'implémentations représentées par la Figure 2.3. Les prétraitements entièrement indépendants de l'encodeur (implémentation 1) peuvent être placés indifféremment devant n'importe quel codeur vidéo. Le fait de n'avoir besoin d'aucun accès à l'encodeur rend l'implémentation de ces filtres très simple, mais en contrepartie aucune optimisation algorithmique ne peut être apportée en exploitant les données de l'encodeur. Les filtres externes à l'encodeur et utilisant des données de l'encodeur (implémentation 2) nécessite généralement un fonctionnement en multi-passes. Une première passe d'encodage permet de contrôler le préfiltre, puis les images filtrées sont encodées une seconde fois. En intégrant le prétraitement dans l'encodeur, l'adaptation du prétraitement au contexte de codage peut être réalisée pour chaque image sans nécessiter d'encodage multi-passes. Les prétraitements intégrés à l'encodeur peuvent être appliqués à l'image résiduelle dans le domaine spatial (implémentation 3) ou fréquentiel, à partir des coefficients DCT (implémentation 4).

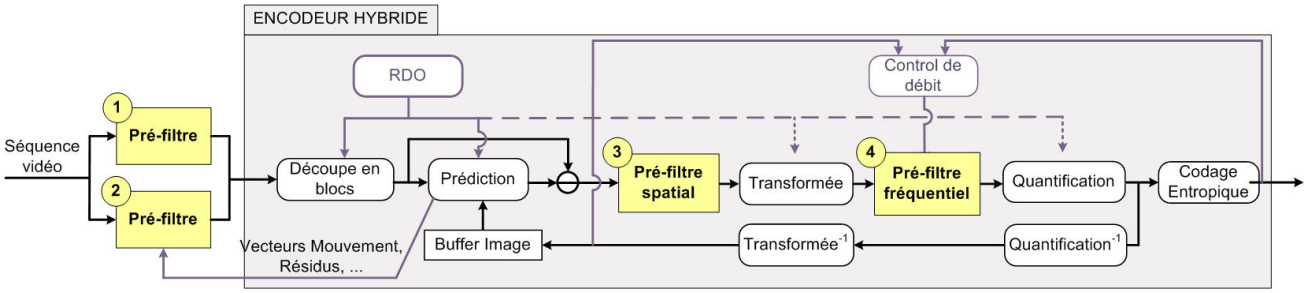


Figure 2.3. Illustration des quatre possibilités de prétraitement
1. Préfiltre indépendant de l'encodeur – 2. Préfiltre externe contrôlé par des données de l'encodeur – 3. Préfiltre appliqué à l'image résiduelle (Dfd) – 4. Préfiltre fréquentiel appliqué aux coefficients DCT

2. 2. 1. Prétraitements pour la réduction de bruit

Les prétraitements développés dès les débuts du codec MPEG2 ont pour origine les deux observations suivantes : d'une part, les capteurs photosensibles des caméras introduisent du bruit dans une séquence vidéo au moment de la captation. Cette dégradation généralement modélisée par un bruit blanc gaussien additif, ajoute un contenu haute-fréquence aux images originales d'une séquence [32].

Les encodeurs vidéo hybrides basés sur une prédiction par estimation de mouvement et une représentation dans le domaine DCT sont par nature sensibles à la présence de bruit. En effet, le bruit étant aléatoire, il ne peut être prédit par l'estimation de mouvement et se retrouve dans le résidu réduisant ainsi l'efficacité de codage.

De nombreuses études se sont attachées à introduire des filtres de réduction de bruit dans la chaîne d'encodage afin de limiter la dépense inutile de bits de codage engendrée par la présence de bruit. L'étape de filtrage a été dans un premier temps indépendante de l'étape d'encodage, chaque image d'une séquence étant alors traitée par un filtre passe-bas appartenant à la littérature des filtres de réduction de bruit classique ; puis avec l'introduction de supports de filtrage tenant compte de la dimension temporelle des séquences vidéos, les pré filtres en interaction avec l'encodeur ont permis d'optimiser le traitement et la complexité algorithmique. Quelle que soit l'implémentation du filtre, il est toujours contrôlé par une estimation du bruit dans les images de la séquence. De la même manière que le filtrage, cette estimation peut être entièrement dépendante du signal vidéo ou exploiter des données d'encodage.

Lorsque la réduction de bruit est réalisée en amont de l'encodage de manière indépendante (Figure 2.3 cas 1), la séquence vidéo filtrée est envoyée en entrée de l'encodeur sans que celui-ci n'ait connaissance de la version originalement bruitée. Des filtres initialement développés pour la restauration d'images bruitées sont alors employés comme prétraitement au codeur vidéo. Ainsi des filtres classiques de la littérature du débruitage, tels le filtre Médian, le filtre de Wiener [33], [34] et le filtre LLMSE [35], ont fait l'objet d'études pour les encodeurs JPEG [33] [35], MPEG2 [36] et H.264/AVC [37].

Lorsque le préfiltre externe à l'encodeur est contrôlé par des données issues de l'encodage (Figure 2.3 cas 2), un encodage double passe est souvent nécessaire. Les informations collectées lors de la première passe d'encodage sont alors utilisées pour régler le pré filtre, puis la séquence préfiltrée est encodée une seconde fois. Les vecteurs de mouvement calculés à l'étape de prédiction peuvent être exploités par un préfiltre spatio-temporel afin de réduire la complexité du préfiltre d'une part et pour réduire les différences entre les macroblocs et leur référence d'autre part, comme le propose [38] pour l'encodeur H.264. L'intensité du mouvement (norme des vecteurs) ainsi que la qualité de l'estimation de mouvement (résidu) peuvent être utilisées afin de régler la force du préfiltre, comme le propose [39] pour l'encodeur MPEG2 dont la solution est illustrée par la Figure 2.4.

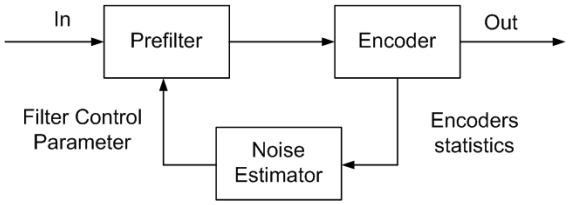


Figure 2.4. Schéma bloc d'un encodeur avec un préfiltre automatiquement contrôlé
Illustration issue de la publication [39]

La complexité d'un préfiltre spatio-temporel utilisant les mêmes vecteurs mouvement que l'encodeur peut-être drastiquement réduite en appliquant le filtre directement sur les informations résiduelles à encoder [40]. Différents travaux se sont attachés à réduire la présence de bruit dans les informations résiduelles (Figure 2.3 cas 3). Ainsi le filtre LLMSE (Linear Least Mean Squared Error) a été appliqué à l'image résiduelle (Dfd (Displaced frame difference)) dans l'encodeur H.264/AVC [41] [42] dont la solution est présentée par la Figure 2.5, tandis que le filtre de Kalman a été étudié pour l'encodeur MPEG2 [43] [44]. D'autres travaux ont pris le parti de réduire l'information haute fréquence due au bruit en traitant les coefficients DCT avant quantification (Figure 2.3 cas 4). Ainsi les auteurs de [40] appliquent un filtre de Wiener fréquentiel dans l'encodeur MPEG2, tandis que les auteurs de [34] et [45] utilisent une fonction de seuillage pour réduire respectivement l'amplitude des coefficients DCT liés à la présence de bruit dans l'encodeur MPEG2 et les coefficients d'ondelette dans l'encodeur MPEG4-Still.

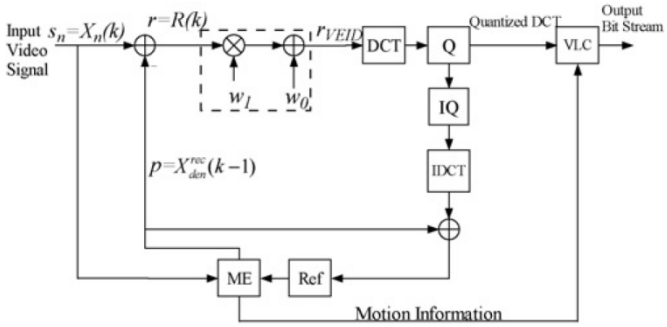


Figure 2.5. Solution de préfiltrage LLMSE des informations résiduelles
Illustration issue de la publication [41]
Avec VEID (Video Encoder with Integrated Denoising), r le résidu et r_{VEID} la version prétraitée tel que r_{VEID} = w0.r +w1, w0 et w1 sont les coefficients du filtre LMMSE qui minimise l'estimation de l'erreur quadratique moyenne du processus de débruitage

Bien que la réduction de bruit facilite grandement l'étape d'encodage, elle trouve rapidement ses limites dans le cas de séquences peu ou faiblement bruitées. Digigram développe des encodeurs dédiés à des applications pour la vidéo professionnelle dans lesquelles la présence de bruit est marginale.

2. 2. 2. Prétraitements pour la réduction de contenu Haute-fréquence

A faible débit, la présence de bruit devient marginale car la quantification élimine d'elle-même les composantes hautes fréquences dues au bruit [46]. Dans un contexte de plus haute qualité lorsque la présence de bruit est marginale et que l'encodage est réalisé à moyen et haut débit, le contenu haute-fréquence peut être réduit car l'œil humain y est peu sensible. La préoccupation principale est alors d'employer des filtres passe-bas à forte capacité de préservation de contours pour éviter d'introduire une impression de flou. Toutefois, l'utilisation de filtre passe-bas comme préfiltre de séquences non-bruitées trouve son intérêt uniquement dans le cas d'encodage à très bas débit, car l'introduction de flou est visible lorsqu'on travaille dans un contexte de haute qualité (haut débit).

Ainsi, différents travaux démontrent qu'un préfiltre passe-bas indépendant de l'encodeur (Figure 2.3 cas 1), lissant l'image permet de réduire l'effet de bloc à bas débit. Les auteurs de [47] [48] appliquent un filtre anisotropique sur les images avant encodage MPEG4, tandis que [49] utilise un filtre LLMSE. Dans [50] les auteurs utilisent une estimation de mouvement simple comme mesure de complexité de codage par macrobloc H.264/AVC et appliquent un filtre passe-bas uniquement en cas de forte complexité.

A partir des deux observations suivantes, à savoir qu'un préfiltre permet de réduire l'effet de bloc à bas débit mais qu'il apporte du flou à moyen et haut débit ; des travaux ont proposés d'introduire le filtre dans l'encodeur pour limiter son action uniquement aux cas d'usage où le préfiltrage apporte une amélioration significative [51]. Ainsi des travaux proposent de contrôler un préfiltre passe-bas en fonction du débit cible, dans [52] les auteurs proposent l'utilisation d'un filtre Bilatéral externe au codeur H.263 dont la force est contrôlée par le QP moyen de l'image précédente. Dans [51] et [32] les auteurs appliquent un filtre gaussien sur la Dfd dans le codeur MPEG2 et introduisent le choix de la variance du filtre dans le module RDO.

2. 2. 3. Prétraitements pour la réduction de contenu perceptivement non-significatif

On a vu dans les sections précédentes que la principale limitation des prétraitements réside dans l'introduction de flou qui dégrade la séquence vidéo filtrée. Dans l'idée de réduire le contenu haute fréquence sans introduire de perte de qualité, l'utilisation de modèles perceptifs pour guider les prétraitements a trouvé sa place afin de réduire le contenu perceptivement non-significatif. Pour ces travaux, l'exploitation des caractéristiques du système visuel humain ne se limite pas à la faible sensibilité de l'œil humain au contenu haute-fréquence, mais tient compte de propriétés plus complexes telles que l'attention visuelle ou les effets de masquage.

2. 2. 3. 1. Les modèles perceptuels utilisés en prétraitement vidéo

On retrouve principalement trois types de modèles perceptuels dans les préfiltres de la littérature : Les approches basées régions d'intérêt (ROI, Region-Of-Interest), les cartes de saillance et les modèles JND (Just Noticeable Distortion).

Les algorithmes de ROI détectent les objets d'une scène qui concentrent l'intérêt du spectateur. Des algorithmes de segmentation d'image basés sur l'analyse du mouvement intervenant entre les images d'une séquence permettent de séparer les objets en mouvement de l'arrière-plan fixe [53], [54]. La Figure 2.6 (a) illustre la détection de ROI proposée par [54]. La détection de ROI dans le cadre de l'encodage vidéo consiste souvent à détecter les personnages humains qui concentrent naturellement l'intérêt du spectateur [55], pour cela une analyse des composantes couleurs est nécessaire afin de détecter la couleur chair [56].

La mesure de saillance poursuit un but similaire aux algorithmes de ROI, à savoir détecter les zones d'une image qui attirent l'attention d'un observateur humain, appelé points d'attention. Cependant cette méthode n'est pas basée sur une détection d'objet sémantique, elle repose sur des critères perceptuels basés sur une analyse des caractéristiques de l'image en termes de contours, fréquences spatiales ou temporelles. La Figure 2.6 (b) illustre le modèle de saillance proposé dans [57].

Les modèles JND considèrent des effets de masquages pour définir un seuil de perception des différences ou des distorsions amenées dans l'image par un traitement. La Figure 2.6 (c) illustre les cartes de JND proposées dans [58], que nous avons choisi d'utiliser pour nos travaux de par son bon compromis entre efficacité et simplicité de calcul. Nous présenterons en détails ce modèle JND dans la suite de ce chapitre.

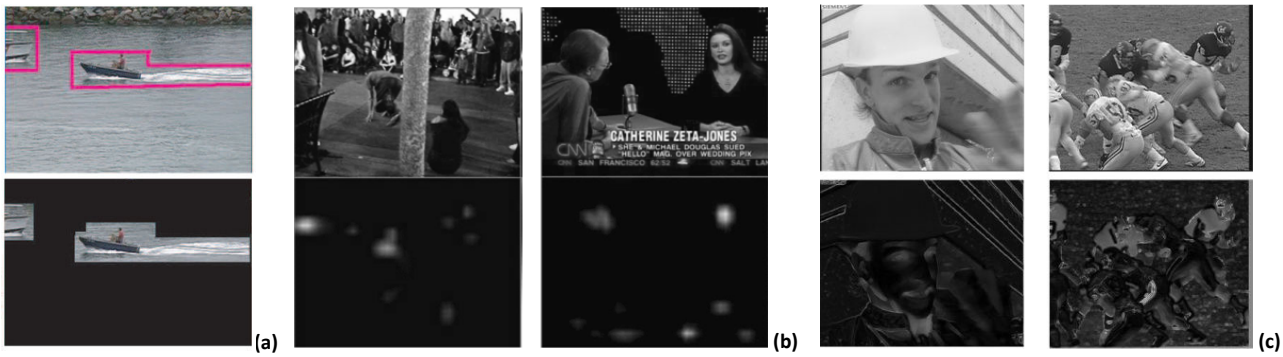


Figure 2.6. Exemples de cartes perceptuelles générées par une implémentation particulière des trois types de modèles rencontrés dans la littérature des prétraitements perceptuels.
(a) Détection de ROI proposée par [54] – (b) Carte de saillance suivant le modèle proposé par [57] – (c) Carte de JND suivant le modèle proposé par [58]

2. 2. 3. 2. Les prétraitements perceptuels

Pour réduire le contenu haute-fréquence non significatif, plusieurs travaux se sont concentrés à guider des filtres passe-bas classiques indépendants de l'encodeur à l'aide de critères perceptifs dans le cas d'encodage à faible débit. Les critères perceptifs varient du plus simple au plus complexe. L'hypothèse psycho-visuelle que les objets en mouvement concentrent notre intérêt, est exploitée pour contrôler un filtre Bilatéral par une mesure d'activité temporelle pour lisser les zones stationnaires avant encodage H.264/AVC [59], dont une partie des résultats est présentée par la Figure 2.7. Les séquences vidéos utilisées pour les tests sont au format CIF (352x288) et encodées à très bas débit. Dans un contexte de très faible débit et par conséquent de qualité médiocre, la technique proposée apporte une amélioration de la qualité subjective.

De manière générale, les travaux en prétraitements de la littérature considèrent le cas de séquences vidéo de faible résolution (CIF à SD) et à faible débit, ce qui nous écarte sensiblement des applications vidéo professionnelles des encodeurs de Digigram dans un contexte de séquences HD encodées en haute qualité.

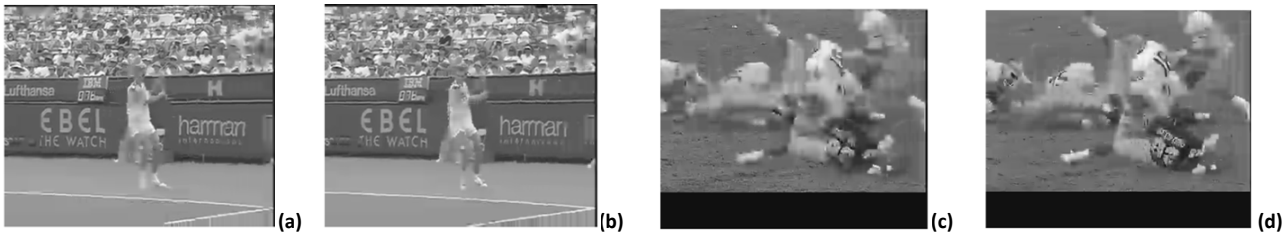


Figure 2.7. Illustration des résultats du préfiltre bilatéral
Illustration issue de la publication proposé par [59] pour des séquences CIF encodées par l'encodeur H264/AVC de référence JM12.2 à 128Kbps
(a) Image de la séquence Stefan encodée sans prétraitement – (b) Image de la séquence Stefan encodée avec prétraitement - (c) Image de la séquence Football encodée sans prétraitement – (d) Image de la séquence Football encodée avec prétraitement

Plusieurs travaux se concentrent à séparer les personnages humains de l'arrière-plan pour appliquer un filtre passe-bas uniquement sur l'arrière-plan perceptuellement moins attractif. Une détection ROI a été employée dans ce but pour l'encodeur MPEG1 [53] et MPEG2 [60] et x264 [61]. Des cartes de saillance permettent également de détecter les visages humains et sont alors exploitées avant encodage MPEG1, MPEG4 [57] et H.264/AVC [62].

Les travaux de prétraitement perceptuels intégrés à l'encodeur peuvent être classés parmi la littérature du codage perceptuel (Figure 2.2), ils utilisent majoritairement un modèle de JND (Just Noticeable Distortion) décrit dans le domaine pixel ou fréquentiel dont nous parlerons au paragraphe suivant. Xiaokang Yang définit un JND dans le

domaine pixel et l'intègre au codeur MPEG2 pour filtrer l'information de Dfd de manière à réduire l'amplitude des résidus en fonction du JND moyen du macrobloc [63] [64] [65] et [66]. Dans un principe similaire, [67]supprime les résidus inférieurs au seuil de perception JND dans l'encodeur H.264/AVC. Un JND fréquentiel est préféré au JND dans le domaine pixel pour sa capacité à prendre en compte l'effet de masquage fréquentiel et ainsi réduire les coefficients fréquentiels DCT dont l'amplitude est inférieure au JND dans le codeur image JPEG [68] et H.264/AVC [69]. On peut noter que des travaux récents intègrent un modèle JND fréquentiel à l'étape de quantification des codeurs H.264/AVC [70] et HEVC [71] pour définir une matrice de quantification non-uniforme perceptuelle, le décodeur doit alors avoir connaissance de ces matrices.

Les détections de ROI et carte de saillance s'attachent à détecter les zones d'attention de l'œil humain, ce sont des algorithmes complexes peu adaptés à un traitement temps réel. De plus, ces techniques sont principalement utilisées dans le cas de transmissions de type vidéoconférence à bas débit où les algorithmes de ROI et de saillance permettent de concentrer le budget binaire sur les personnages humains. De nombreux modèles JND plus ou moins complexes, ont été proposés dans la littérature depuis les années 90. Dans le paragraphe suivant nous présentons les principes de la mesure JND, puis nous détaillerons le modèle proposé par [66], défini dans le domaine pixel que nous proposons d'utiliser ensuite pour contrôler un préfiltre indépendant de l'encodeur.

2. 3. **Modèle perceptuel : JND**

Dans le domaine du prétraitement pour l'encodage vidéo, les modèles JND utilisés définissent des seuils de perception au-dessous desquels des distorsions introduites dans l'image ne peuvent être perçues (Just Noticeable Distortion). De manière plus large, on parle de seuil de perception des différences (Just Noticeable Difference), le concept est ainsi étendu aux traitements n'introduisant pas de dégradation comme par exemple le rehaussement de contours [72]. Il existe plusieurs modèles JND dans la littérature qui diffèrent par le type de masquage pris en compte et le domaine de calcul, nous en présentons une sélection dans la suite de ce paragraphe.

Une méthode de mesure du JND proposée dans [73], consiste à présenter à un groupe d'observateurs une image originale puis des versions de plus en plus dégradées jusqu'à ce que les distorsions soient perçues par les observateurs. Cette expérience est exprimée par l'équation suivante où x_t est l'image présentée aux observateurs, x_o l'image originale, x_d la version dégradée et h le paramètre permettant de contrôler la visibilité des distorsions qui varie entre 0 et 1:

$$x_t = x_o + h(x_d - x_o)$$

Équation 2.1. *Test pour la mesure du JND*

L'expérience commence en présentant l'image originale (paramètre h à 0) puis en augmentant le paramètre h. Le seuil correspondant à 1-JND est trouvé lorsque 75% des observateurs perçoivent une différence entre la version originale et la version test, notée alors $x_{t,JND}$. En substituant $x_{t,JND}$ à l'image originale et en continuant d'augmenter le paramètre h, de la même manière on peut trouver le seuil de différence correspondant à 2-JND.

On comprend bien que ce test est dépendant de l'image originale ainsi que du type de distorsion testée, ainsi la valeur de 1-JND n'est pas applicable à un autre contexte. Pour utiliser le seuil de perception JND au sein d'algorithmes de traitement d'image, il est nécessaire de pouvoir prédire les seuils de JND. Ainsi, des modèles de JND ont été développés depuis le début des années 90 et ont été appliqués aux domaines du traitement et de l'encodage vidéo.

2. 3. 1. Les principes du JND

Les modèles JND de la littérature sont définis dans le domaine pixel ou fréquentiel (généralement dans le domaine DCT ou ondelette), ils définissent respectivement un seuil de perception par pixel ou par coefficient fréquentiel. Quelque soit leur domaine de définition, les modèles JND considèrent les phénomènes de masquage pour déterminer les valeurs de seuils perceptifs.

Les phénomènes de masquage ne sont pas propres au domaine du traitement vidéo, on parle de masquage lorsque qu'un phénomène physique réduit ou cache la présence d'un autre. Pour appréhender simplement le principe des effets de masquage, nous nous plaçons dans le contexte du masquage auditif qui intervient lorsqu'un signal sonore (masquant) cache ou réduit la perception d'un autre (signal sonore masqué) [74]. Il existe deux types de masquage auditif, le masquage simultané (ou fréquentiel) (Figure 2.8) et non-simultané (Figure 2.9).

Lorsque que le masquage est simultané, c'est-à-dire que les sons masquant et masqué sont émis en même temps, la proximité de composantes fréquentielles produit un effet de masquage. La Figure 2.8 issue de [75], présente l'évolution du seuil perceptif pour des fréquences pures en présence d'un signal de bruit masquant de 90Hz de largeur centré sur 410Hz, les différentes courbes représentent l'expérience réalisée à différents niveaux de bruit. On remarque que les fréquences proches du signal masquant voient leur seuil de perception augmenter. Ainsi des fréquences dont l'amplitude se trouve inférieure au seuil de perception seront non-perceptibles en présence du signal masquant et pourront être supprimées sans altérer la perception globale du son.

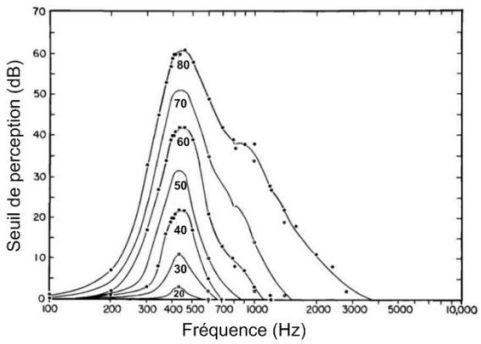


Figure 2.8. Illustration de l'effet de masquage simultané ou fréquentiel
Iillustration issue de la publication [75]

Les effets de masquages non-simultanés illustrés par la Figure 2.9, interviennent lorsque deux sons sont émis à des temps rapprochés. La présence d'un signal sonore masquant réduit alors la perception de signaux émis juste avant (masquage de Postériorité) [76] et juste après le signal masquant (masquage de Haas) [77].

De telles observations sont exploitées par l'encodage de signaux audio afin de réduire le volume de données en amenant le moins de distorsion perceptibles possibles.

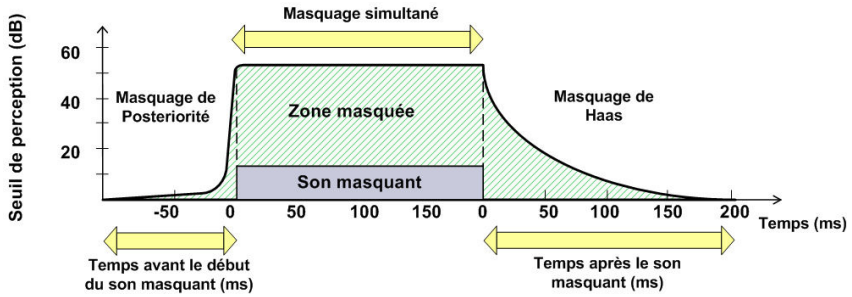


Figure 2.9. Illustration de l'effet de masquage non-simultané
Illustration inspirée de la thèse [74]

Dans le domaine du traitement vidéo, les principaux masquages rencontrés sont les masquages spatiaux, fréquentiels et temporels. Dans les masquages spatiaux, on considère principalement le masquage en luminance basé sur la loi de Weber-Fechner qui reflète la faible sensibilité de l'œil humain aux différences intervenant dans les zones sombres d'une image [72]. De plus, le seuil de perception des différences dépend de l'activité spatiale, il est

notamment largement reconnu par la communauté scientifique que le système visuel humain (SVH) est très sensible à l'information de contour des images [78], [79]. Le masquage temporel reflète l'influence du mouvement intervenant entre les images d'une séquence sur la perception des distorsions, ainsi le SVH est plus sensible aux distorsions intervenants dans les zones stationnaires [80].

Le masquage fréquentiel reflète la sensibilité du SVH aux fréquences spatiales, elle est définie par la fonction de sensibilité au contraste (CSF an anglais) [72] présentée par la Figure 2.10 où l'on peut constater que la sensibilité en luminance chute rapidement dans les hautes fréquences spatiales. Seuls les modèles JND définis dans le domaine fréquentiel tiennent compte de cet effet de masquage, ce qui les rend plus précis et plus complexes que les modèles JND définis dans le domaine pixel [67]. Les modèles JND dans le domaine fréquentiel ont été développés dans le début des années 90 avec les travaux de [81] et intégrés à l'encodeur image JPEG pour définir des matrices de quantification perceptuelles [82].

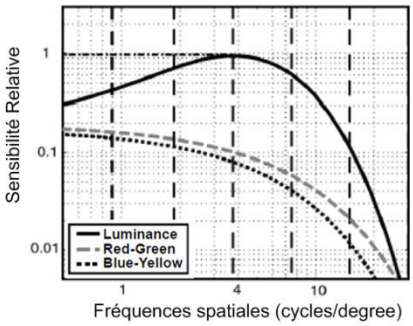


Figure 2.10. Fonction de sensibilité du système visuel humain aux fréquences spatiales Illustration issue de la publication [68]

Toutefois pour la suite de notre travail, nous nous concentrons sur un modèle JND décrit dans le domaine pixel. Dans le but de contrôler un préfiltre pour réduire le contenu non-perceptif dans un contexte d'encodage temps-réel, nous avons fait ce choix pour plusieurs raisons :

- D'une part, l'intégration d'un modèle JND DCT dans un préfiltre impose le choix d'un filtre décomposant les images en bloc DCT afin de réduire les coefficients en fonction des seuils JND. A forte force de filtrage, l'artefact de filtrage principal devient l'effet de bloc qui est déjà l'artefact principal de l'encodage vidéo. A contrario, les filtres appliqués par convolution ne traitent pas l'image par bloc, le principal artefact est alors l'introduction du flou.
- D'autre part, comme nous l'avons vu précédemment, les modèles JND décrit dans le domaine pixel sont moins complexes et se prêtent mieux au cas d'un traitement temps réel.

2. 3. 2. Présentation du JND de Yang

Parmi les modèles JND décrit dans le domaine pixel, celui défini par X. Yang et al. [65] est le plus largement utilisé car il est un bon compromis entre précision et simplicité algorithmique. Dans ce paragraphe nous allons détailler le calcul de ce modèle que nous utiliserons par la suite.

Ce modèle JND est dit spatio-temporel car il considère à la fois le masquage temporel dû au mouvement intervenant entre les images d'une séquence et le masquage spatial regroupant le masquage en texture et en luminance. On entend par effet de masquage la présence d'une caractéristique réduisant la sensibilité du SVH à l'information contenue par le pixel concerné. Le JND Spatio-temporel JND_{ST} est obtenu par multiplication des deux effets de masquage :

$$JND_{ST} = JND_S \,.\, JND_T$$ ***Équation 2.2.*** *JND Spatio-temporel*

2. 3. 2. 1. JND spatial

Le masquage spatial JND_S est obtenu en considérant les masquages en luminance JND_{Lum} et en texture JND_{Tex}. Lorsque les deux masquages sont présents, un phénomène de recouvrement diminue la somme des deux masquages.

$$JND_S = JND_{Lum} + JND_{Tex} - C.min\{JND_{Lum}, JND_{Tex}\}, C = 0.3$$ ***Équation 2.3.*** *JND Spatial*

- **Masquage en luminance**

Le masquage en luminance traduit la différence de sensibilité du système visuel humain aux changements intervenant dans une image en fonction du niveau de luminance de la zone concernée. La définition du masquage utilise l'approximation de la loi de Weber-Fechner décrite par l'Équation 2.4 :

$$JND_{lum} = \begin{cases} 17\left(1 - \sqrt{\frac{\overline{I(x,y)}}{127}}\right) + 3 & si\ \overline{I(x,y)} \leq 127 \\ \frac{3}{128}\left(\overline{I(x,y)} - 127\right) + 3, & sinon \end{cases}$$

Équation 2.4. *Masquage en Luminance*

Avec $\overline{I(x,y)}$ la luminance moyenne pondérée du voisinage 5x5 de chaque pixel. La loi de Weber-Fechner est illustrée par la Figure 2.11. La fenêtre de pondération utilisée pour le calcul de la luminance moyenne est présentée par la Figure 2.12.

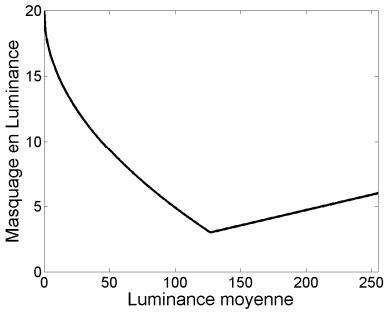


Figure 2.11. Approximation de la loi Weber-Fechner

1	1	1	1	1
1	2	2	2	1
1	2	0	2	1
1	2	2	2	1
1	1	1	1	1

Figure 2.12. Fenêtre de pondération pour le calcul de la luminance moyenne

La Figure 2.13 présente le masquage en luminance des images CIF *Football* et *Mobile & Calendar*. La luminance des cartes JND représente la valeur du seuil JND, c'est-à-dire que plus les cartes sont sombres, plus le seuil JND est bas et plus le système visuel humain est sensible à l'introduction de distorsions.

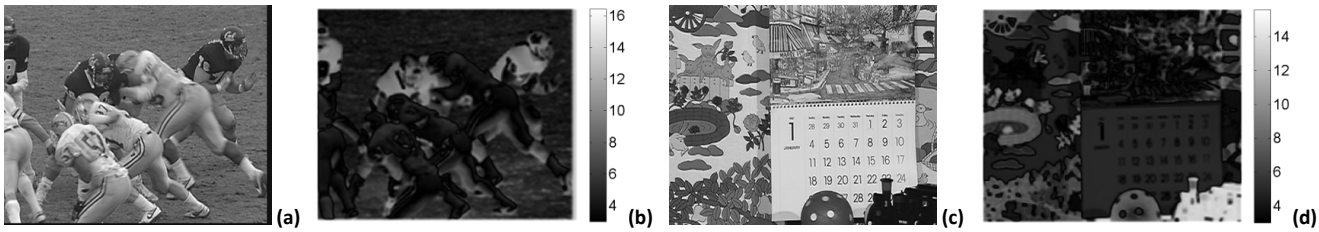


Figure 2.13. Représentation du masquage en luminance
(a) Image issue de la séquence CIF « Football » - (b) Carte de masquage en luminance pour l'image « Football » – (c) Image issue de la séquence CIF « Mobile & Calendar» - (d) Carte de masquage en luminance pour l'image « Mobile & Calendar»

- **Masquage en texture**

Le masquage en texture traduit le fait que la forte activité spatiale à l'intérieur d'une image aura tendance à réduire la sensibilité de l'œil à un stimulus visuel ; et par conséquent reflète la forte sensibilité du système visuel humain aux informations de contours ainsi qu'aux zones homogènes. En effet une perte de netteté des contours est ressentie comme une forte perte de qualité, de plus la perte d'information dans les zones à faible activité spatiale provoque un effet d'aplat gênant. Au contraire, les zones texturées de par leur forte activité spatiale peuvent abriter plus de distorsions sans que celles-ci ne soient perçues. Le masquage en texture est basé sur une analyse de gradient $G(x,y)$ ainsi qu'une détection de contours utilisant un filtre de Canny [83] $W_e(x,y)$:

$$JND_{text} = G(x,y).W_e(x,y)$$ ***Équation 2.5.*** *Masquage en texture*

Comme nous venons de le voir, le système visuel humain est fortement sensible aux contours et peu aux textures. Afin de séparer ces deux contenus haute fréquence, une carte gradient G(x,y) est d'abord calculée pour détecter à la fois les contours et les textures, puis une carte de contours $W_e(x,y)$ est calculée à l'aide de l'opérateur de Canny et d'un opérateur morphologique, permettant de supprimer les contours de la carte de gradient et d'ainsi conserver uniquement les textures qui présentent de forts seuils JND.

La carte de gradient est obtenue par l'analyse de quatre directions de gradient présentée par la Figure 2.14 pour l'image *Football*. La carte finale de gradient G(x,y) correspond au maximum des quatre directions de gradient en chaque pixel :

$$G(x,y) = \max_{k=1,2,3,4}\{|grad_k(x,y)|\}$$ ***Équation 2.6.*** *Carte de gradient*

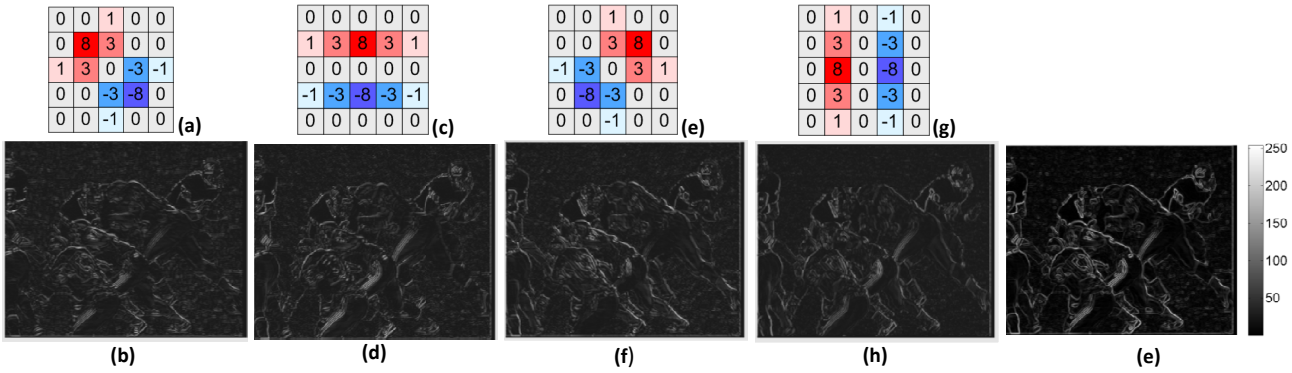


Figure 2.14. Carte de gradient selon quatre directions et gradient final
(a, c, e, g) Support de convolution pour le calcul de gradient dans quatre directions – (b, d, f, h) Carte de gradient calculée respectivement à partir des supports a, c, e, g – (e) Carte de gradient finale G(x,y)

Seuls les pixels de texture présentent un fort masquage, pour différencier les pixels de texture des pixels de contours, la carte de gradient G(x,y) est multipliée par une carte $W_e(x,y)$ basée sur une estimation de contours Canny, dont la Figure 2.15 (a) et (c) donne un exemple respectivement pour l'image *Football* et *Mobile & Calendar*, auxquelles un opérateur morphologique est appliqué. Finalement, un filtre gaussien est appliqué à la carte de contours pour adoucir la transition entre les pixels de contours et les autres. Les cartes (b) et (d) de la Figure 2.15 présentent l'information de contour W_e ainsi calculée pour les images *Football* et *Mobile & Calendar.* Finalement le JND en texture est obtenu en multipliant d'abord la carte de contour $W_e(x,y)$ à la carte de gradient G(x,y).

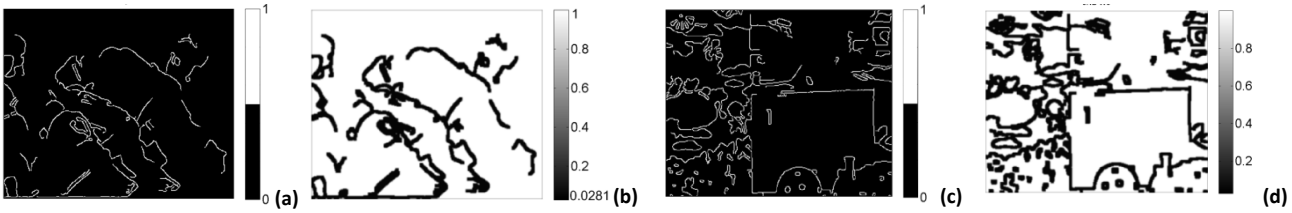


Figure 2.15. Cartes de l'information de contour
(a) ***Carte Binaire Détection Canny, image de la séquence Football – (b) Information de contour We(x,y), image de la séquence Football –*** **(c)** ***Carte Binaire Détection Canny, image de la séquence Mobile & Calendar – (d) Information de contour We(x,y), image de la séquence Mobile & Calendar***

La Figure 2.16 présente les cartes de masquage en luminance (b) et (f), en texture (c) et (g) et le JND spatial (d) et (h) pour les images *Football* (a) et *Mobile* (e) obtenues par l'Équation 2.3. Nous rappelons qu'un seuil JND faible

correspond à une information auquel le système visuel humain est sensible, en effet un faible niveau de distorsion peut être ajouté avant d'être perçue. Comme nous pouvons le voir sur les cartes de JND spatial (d) et (h), les informations pouvant abriter le plus de distorsions (ayant les seuils JND les plus élevés), sont les zones sombres (maillots des joueurs de football, petit train) ainsi que les zones fortement texturées (illustration du calendrier).

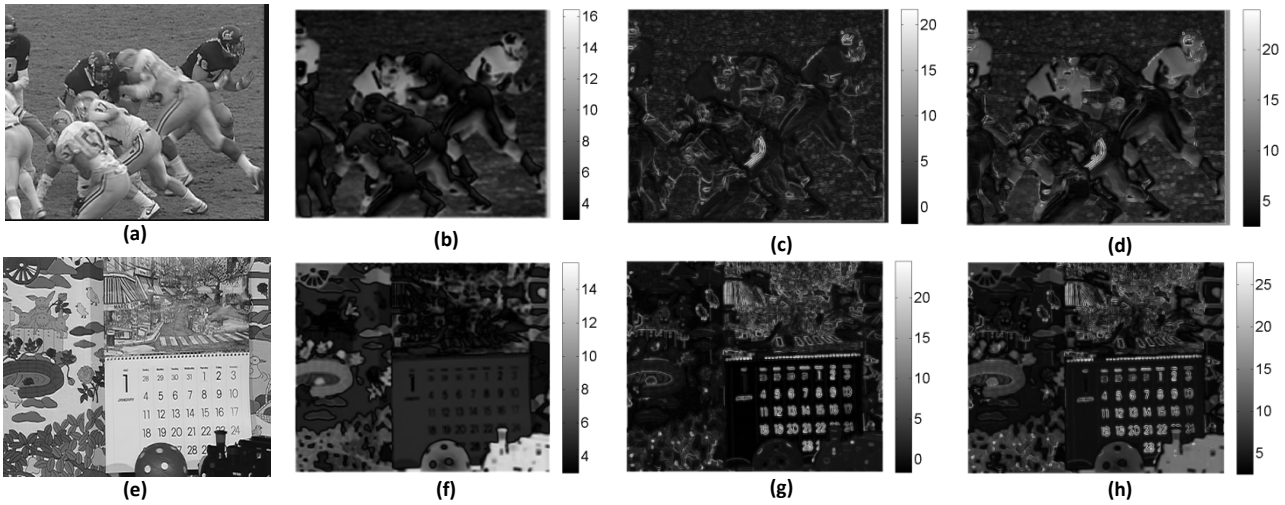


Figure 2.16. Représentation des étapes de calcul du JND spatial
(a) Image Football – (b) Carte de masquage en luminance pour l'image Football – (c) Carte de masquage en texture pour l'image Football – (d) Carte de JND spatial pour l'image Football - (e) Image Mobile – (f) Carte de masquage en luminance pour l'image Mobile – (g) Carte de masquage en texture pour l'image Mobile – (h) Carte de JND spatial pour l'image Mobile

2. 3. 2. 2. **JND temporel**

Le masquage temporel reflète la faible sensibilité du système visuel humain aux zones en mouvement. Le mouvement existant entre deux images d'une séquence est mesuré par la différence inter-image *idl(x,y,t)*. f3 est une fonction empirique définissant le masquage temporel en fonction de la différence inter-image. Par expérience, les auteurs de [84], définissent la fonction f3 par l'Équation 2.7 représentée par la Figure 2.17 où $\mathcal{H}$, $\mathcal{L}$ et τ sont des paramètres du modèle respectivement égaux à 8, 3.2, 0.8.

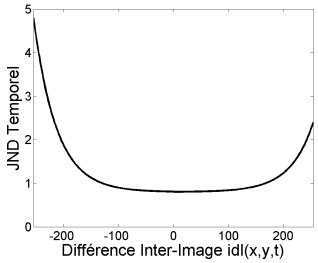


Figure 2.17. JND temporel en fonction de la différence inter-image

$$JND_T = f_3(ild(x,y,t))$$

$$JND_T = \begin{cases} max\left(\tau, \frac{\mathcal{H}}{2} exp\left(\frac{-0.15}{2\pi}(idl(x,y,t)+255)\right)+\tau\right), & idl(x,y,t) \le 0 \\ max\left(\tau, \frac{\mathcal{L}}{2} exp\left(\frac{-0.15}{2\pi}(255-idl(x,y,t))\right)+\tau\right), & idl(x,y,t) > 0 \end{cases}$$

Équation 2.7. *JND temporel*

$$idl(x,y,t) = \frac{1}{2}\left(I(x,y,t) - I(x,y,t-1) + \overline{I(x,y,t)} - \overline{I(x,y,t-1)}\right)$$

Équation 2.8. *Différence Inter-Image*

Avec *I(x,y,t)* le pixel de coordonnées (x,y) dans l'image t et $\overline{I(x,y,t)}$ la luminance moyenne autour du pixel dans un voisinage 5x5 (Figure 2.12)

On peut noter que plusieurs travaux ont proposé des extensions au modèle JND que nous venons de détailler. Les auteurs de ce modèle JND proposent de considérer le masquage des informations de chrominance dans [58]. Les auteurs de [84] considèrent l'attention visuelle en ajoutant un modèle fovéa dans le calcul du JND global.

Une méthode pour évaluer les modèles JND consiste à introduire du bruit dans une image de manière uniforme et d'introduire un même niveau de bruit mais cette fois guidé par le modèle JND à évaluer. La Figure 2.18 présente cette expérience réalisée avec le modèle JND que nous venons de présenter. Le même niveau de bruit ayant été ajouté dans les deux cas (b) et (c), les PSNR des deux images bruitées sont très proches, cependant le résultat visuel est très différent ! En guidant l'ajout de bruit par le modèle JND, le bruit semble beaucoup moins présent (b). Si on regarde avec attention la partie texturée de l'image (les plumes du chapeau) défini comme le contenu le moins sensible par le JND, vous verrez que le bruit est plus présent que dans la version avec le bruit uniforme.

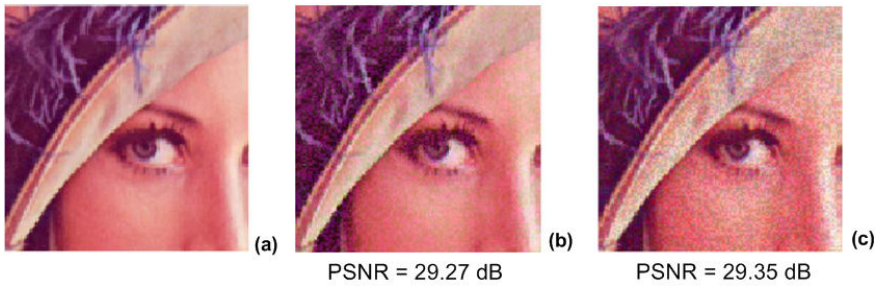


Figure 2.18. Expérience de validité du modèle JND
Illustration issue de la publication [65]
(a) Image originale – (b) Image avec un ajout de bruit contrôlé par le JND – (c) Image bruitée avec un bruit uniforme

Dans la suite de nos travaux, nous nous concentrerons sur le modèle de JND spatial décrit précédemment qui attribue une forte sensibilité aux contours et zones à faible activité spatiale. Nous proposons d'utiliser le modèle JND pour contrôler la force de filtrage d'un filtre passe-bas et ainsi réduire le contenu pas/peu perceptible.

2. 4. Filtre AWA perceptuel

Nous avons étudié les filtres passe-bas proposés pour la réduction de bruit, ce qui nous a amené à sélectionner le filtre AWA (Adaptive Weighted Averaging) pour son efficacité à préserver les contours, sa faible complexité algorithmique qui se prête à l'implémentation temps-réel, ainsi que pour la simplicité d'intégration du modèle JND que nous allons présenter dans la suite de ce chapitre.

2. 4. 1. Les filtres passe-bas de la littérature

Les filtres passe-bas de la littérature destinés à la réduction de bruit peuvent être appliqués dans le domaine pixel ou fréquentiel [85]. Pour les raisons citées précédemment, nous avons concentré notre attention sur les filtres appliqués dans le domaine pixel. Traditionnellement, ces filtres sont appliqués par convolution. Les redondances existantes entre les pixels d'une image ou d'une séquence sont exploitées afin d'évaluer la valeur non-bruitée de chaque pixel. Ce principe se traduit par l'application à chaque pixel d'une combinaison des autres pixels de l'image ou de la séquence. Les pixels sélectionnés constituent le support de filtrage qui est appliqué à l'image par convolution. En fonction de la dimension du support, on distingue les filtres spatiaux exploitant uniquement les redondances existant à l'intérieur d'une même image, des filtres spatio-temporels (ou temporels) considérant la corrélation temporelle entre les images d'une séquence.

2. 4. 1. 1. **Les Filtres Spatiaux**

Les filtres spatiaux développés pour la réduction de bruit d'image fixe depuis les années 80, utilisent majoritairement un support de filtrage de taille fixe. La valeur filtrée de chaque pixel est une fonction des pixels voisins contenus dans le support de filtrage. Les filtres de la littérature diffèrent par la nature de cette fonction.

Les filtres linéaires ont un masque de filtrage fixe appliqué à tous les pixels d'une image sans distinction. Ces filtres ont une mauvaise capacité de préservation des contours amenant une impression de flou. Les filtres moyenneur et gaussien sont deux filtres bien connus de cette famille. Le filtre moyenneur attribue à chaque pixel une moyenne simple ou pondérée de ses voisins. Le filtre gaussien concentre les poids du support sur les pixels spatialement proches du pixel courant. L'opération de filtrage est exprimée par la formule suivante :

$$p_f(x,y) = \frac{1}{Z(x,y)} \sum_{(i,j)\in S_{x,y}} p(i,j) \times G(i,j)$$

$$G(i,j) = exp\left(\frac{-(x^2+y^2)}{2\times\sigma^2}\right)$$

Équation 2.9. Filtre Gaussien

Avec $p_f(x,y)$ le pixel filtré, $p(i,j)$ les pixels voisins du support de filtrage $S_{x,y}$ centré sur le pixel courant, G un noyau gaussien d'écart-type σ, et $Z(x,y)$ une constante de normalisation égale à la somme des poids du support. L'écart-type σ permet de régler la force de filtrage, un σ faible concentre les poids du masque de filtrage sur les voisins très proches du pixel courant (Figure 2.19 (a)). Lorsque σ est élevé, tous les pixels du support ont un poids similaires et le filtre gaussien se rapproche alors d'un filtre moyenneur (c). Le filtre gaussien permet de limiter l'effet de flou amené par le filtre moyenneur mais le support de filtrage reste indépendant du contenu de l'image.

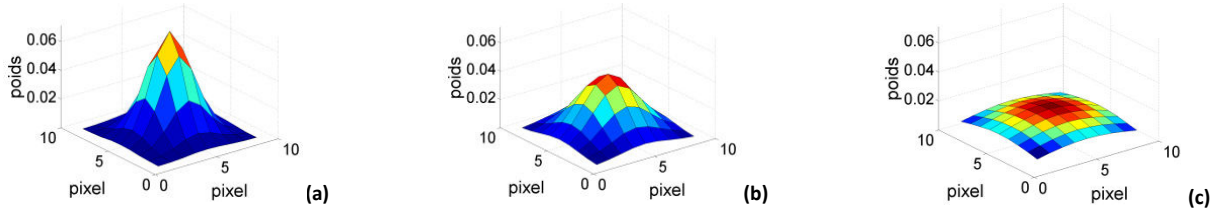


Figure 2.19. Noyaux gaussiens 9x9
(a) écart-type σ = 1.5 – (b) σ = 2 - (c) σ = 2.5

Les filtres adaptatifs utilisent un support de filtrage dépendant du contenu de l'image, parvenant de cette manière, par exemple, à améliorer la conservation des contours d'une scène où ils auraient été préalablement identifiés. Un état de l'art sur les filtres adaptatifs peut être trouvé dans l'article [86]. Parmi les filtres adaptatifs, les plus couramment rencontrés sont les filtres basés sur une moyenne pondérée des pixels du support. Les filtres à moyenne pondérée ou filtres de voisinage réalisent comme leur nom l'indique, une moyenne pondérée des voisins locaux du pixel courant. L'effort d'innovation se trouve dans la fonction de calcul des poids du support. Afin d'illustrer le fonctionnement des filtres adaptatifs, on présente ci-dessous le comportement d'un filtre de voisinage considérant la distance en luminance des pixels du support avec le pixel courant [87]:

$$p_f(x,y) = \frac{1}{Z(x,y)} \sum_{(i,j)\in S_{x,y}} p(i,j) \times e^{\frac{-|p(i,j)-p(x,y)|^2}{2\times\sigma^2}}$$

Équation 2.10. *Filtre de voisinage*

La Figure 2.20 présente les poids attribués aux pixels du support de filtrage, calculés par le filtre adaptatif défini ci-dessus avec un écart-type σ égal à 15, pour des tailles de filtrage 9x9. Le premier exemple concerne un pixel appartenant à un contour (a), et le deuxième un pixel courant appartenant à une zone homogène. On voit que le support de filtrage évolue en fonction du contenu, des poids élevés sont attribués aux pixels ressemblants au pixel courant. Lorsque le support de filtrage contient des pixels très ressemblants, le support de filtrage s'approche d'un filtre moyenneur simple (d).

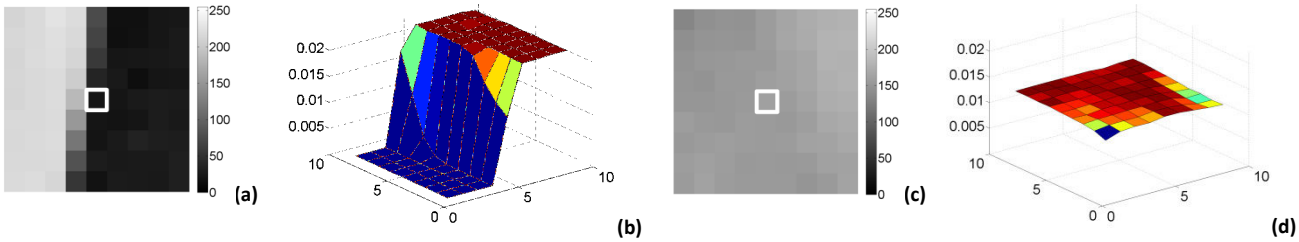


Figure 2.20. Illustration de supports de filtrage statiques et adaptatifs classiques
(a) Pixel courant appartenant à un contour et ses voisins 9x9 - (b) Support de filtrage adaptatif pour le contenu (a) – (c) Pixel courant appartenant à une zone homogène et ses voisins 9x9 - (d) Support de filtrage adaptatif pour le contenu (c)

2. 4. 1. 2. Les Filtre Spatio-temporels

A la différence des filtrages spatiaux, les filtrages spatio-temporels (ou simplement temporels) filtrent les images d'une séquence les unes par rapport aux autres. Ces filtres exploitent la dimension temporelle d'une séquence vidéo. Pour traiter un pixel, le filtre prend en compte ses voisins dans l'image courante ainsi que dans les images précédentes et suivantes.

Les filtres statiques ne tiennent pas compte du mouvement, ils sont obtenus par extension du modèle 2D en 3D. Les pixels voisins dans les images précédentes et suivantes, sont pris autour de la position du pixel en cours de traitement. On ne tient pas compte du fait que le pixel courant peut être en mouvement et donc avoir une position différente dans les images utilisées pour le filtrage. Les filtres statiques entraînent un effet de flou important, surtout dans les parties en fort mouvement d'une vidéo.

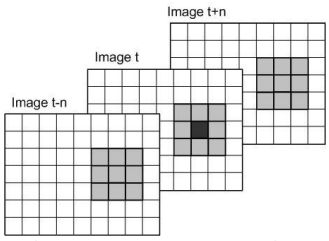


Figure 2.21. Support 3D Statique

Les filtres à compensation de mouvement réalisent une estimation de mouvement, afin de suivre le pixel en cours de traitement dans plusieurs images successives au lieu de filtrer entre eux des pixels placés à la même position dans plusieurs images, comme le font les filtres statiques. La compensation de mouvement permet de réduire nettement l'impression de traînée introduite par les filtres temporels statiques, toutefois l'algorithme d'estimation de mouvement doit être précis ce qui entraîne une importante augmentation du temps de calcul global du prétraitement.

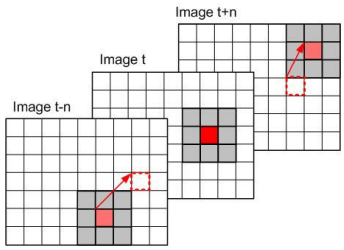


Figure 2.22. Support 3D avec Estimation de mouvement

Pour notre étude, nous avons concentré notre attention sur les filtres de voisinage spatiaux qui allient simplicité et efficacité en vue d'une implémentation temps réel hors de l'encodeur. Une étude plus approfondie des filtres passe-bas de la littérature pourra être trouvée au chapitre suivant. Le filtre AWA que nous avons sélectionné apporte de bonne capacité de préservation des contours et présente la particularité de proposer un seuil proportionnel à une différence de luminance ce qui se prête naturellement à l'intégration du JND.

Une étude réalisée durant la deuxième année de thèse a conforté notre choix du filtre AWA 3x3 en vue d'un traitement temps-réel. En effet, Digigram a accueilli un stage de fin d'étude sur l'implémentation temps réel du filtre AWA pour du débruitage de séquence HD. A l'aide d'optimisations vectorielles (instructions SSE[7] - Streaming SIMD Extensions) sur un cœur XEON E5405 2.00GHz, nous avons pu filtrer une séquence HD avec un support de filtrage AWA à plus de 60 images par seconde. Dans le paragraphe suivant nous allons présenter en détails le fonctionnement du filtre AWA que nous utiliserons par la suite comme préfiltre guidé par le modèle JND.

[7] Le jeu d'instructions SSE est proposé par Intel pour réaliser des calculs vectoriels dans des registres dédiés du microprocesseur. Les données sont chargées dans les registres SSE de 128 bits du microprocesseur (par exemple huit pixels représentés sur 16 bits) et des opérations élémentaires (addition, multiplication) sont réalisées sur les données du registre en même temps. La taille des registres diffère en fonction de la version du jeu d'instruction et de la technologie du microprocesseur associé : 64 bits pour MMX, 128 pour SSE.X et 256 pour AVX (technologie SANDY BRIDGE, 2011).

2. 4. 2. Fonctionnement du filtre AWA

Le filtre AWA proposé dans [88] attribue à chaque pixel *p(x,y)* une moyenne pondérée de ses voisins *p(i,j)* appartenant à un support carré $S_{x,y}$ centré autour du pixel courant. L'opération de filtrage s'exprime de la manière suivante :

$$p(x,y) = \sum_{(i,j)\in S_{x,y}} w(i,j) \times p(i,j)$$

Équation 2.11. *Fonction de filtrage AWA*

Les poids *w(i,j)* attribués aux pixels du support de filtrage sont fonction de leur ressemblance au pixel courant. Le calcul des poids AWA est exprimé par :

$$w(i,j) = Z(x,y) \times Y(i,j) = \frac{Z(x,y)}{1 + a \times max(\varepsilon^2, \Delta p(i,j)^2)}$$

Équation 2.12. *Poids AWA*

$$\boldsymbol{\Delta p(i,j) = p(x,y) - p(i,j)}$$

$$\boldsymbol{Z(x,y) = \left(\sum_{(i,j)\in S_{x,y}} Y(i,j)\right)^{-1}}$$

Avec *Δp(i,j)* la différence de luminance entre le pixel courant *p(x,y)* et les pixels *p(i,j)* du support de filtrage *Sx,y*. *Z(x,y)* est une constante de normalisation pour s'assurer que la somme des poids du support soit égale à 1 et ainsi conserver la luminance moyenne de l'image. Les paramètres ε et *a* contrôlent la force de filtrage.

La Figure 2.23 présente l'évolution de la fonction des poids AWA *Y(i,j)* en fonction de la différence de luminance *Δp(i,j)* intervenant entre un pixel et ses voisins compris dans le support de filtrage. Plus un pixel du support de filtrage est ressemblant au pixel courant (*Δp(i,j)* faible), plus il aura une influence importante dans le calcul de la valeur débruitée du pixel courant. Les pixels du support de filtrage dont la différence *Δp(i,j)* avec le pixel courant est inférieure au paramètre ε ont tous le même poids *Δp(i,j)*. Le seuil ε fait l'objet de toute notre attention dans la suite de notre étude.

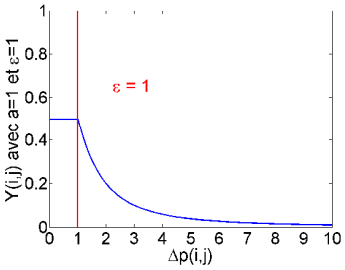


Figure 2.23. Allure de la fonction d'attribution des poids AWA avec les paramètres a = 1 et ε = 1

Dans le paragraphe suivant nous allons étudier l'influence des paramètres *a* et ε sur la fonction d'attribution des poids AWA *Y(i,j)*.

2. 4. 2. 1. Etude de la fonction d'attribution des poids AWA

La Figure 2.24 présente l'influence du paramètre *a* sur la fonction d'attribution des poids *Y(i,j),* le seuil ε est fixé à 1. Le paramètre *a* contrôle la vitesse de décroissance des poids attribués aux pixels du support ayant une différence *Δp(i,j)* au pixel courant supérieure au seuil ε. Plus le paramètre *a* est faible et plus rapidement les pixels différents du pixel courant seront rejetés du support de filtrage. Ainsi le paramètre *a* permet de régler la force de filtrage, plus *a* est faible et plus la force de filtrage est importante. Toutefois nos expériences ainsi que les auteurs du filtre AWA préconisent de fixer le paramètre *a* à 1 car son influence sur l'image filtrée est faible (la raison de cette faible influence sera expliquée au chapitre suivant). De plus la compréhension du réglage du paramètre *a* est moins aisée que celle du réglage seuil ε.

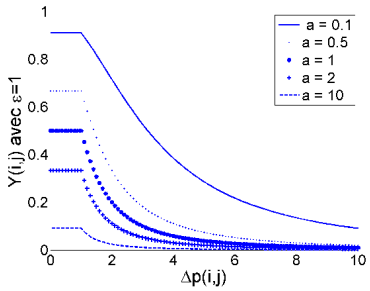


Figure 2.24. Influence du paramètre a sur l'évolution des poids AWA en fonction de la différence de luminance

La Figure 2.25 présente l'influence du paramètre ε sur la fonction d'attribution des poids *Y(i,j),* le paramètre *a* est fixé à 1. Le paramètre ε agit comme un seuil de différence de luminance par rapport au pixel courant, au-dessous duquel les pixels du support ont tous le même poids. Les pixels du support ayant une différence au pixel courant *Δp(i,j)* supérieure au seuil ε, voient leur poids *Y(i,j)* évoluer de façon inversement proportionnel à *Δp(i,j)*. Le paramètre ε a donc une influence directe sur la force de filtrage.

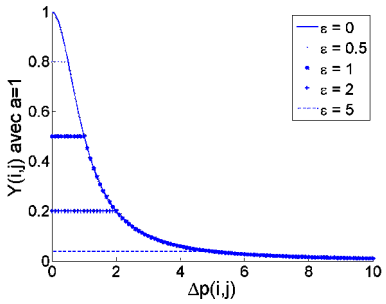


Figure 2.25. Influence du paramètre ε sur l'évolution des poids AWA en fonction de la différence de luminance

La Figure 2.26 présente un exemple de calcul des poids AWA pour un pixel appartenant à un contour avec un support de filtrage 9x9[8]. Lorsque le seuil ε est égal à 30 (b), seuls les pixels du support ressemblant au pixel courant ont un poids élevé, ainsi le pixel courant est uniquement moyenné avec les voisins qui lui ressemblent et la netteté du contour n'est pas compromise par le filtrage. Lorsque le seuil ε augmente (c), le poids attribué aux pixels différents du pixel courant augmente tout en restant toujours inférieur aux poids des pixels très similaires au pixel courant. Le seuil ε égal à 255 (d) représente la force de filtrage maximum pour une image codée sur 8 bits, le filtre AWA se comporte alors en simple moyenneur puisque tous les pixels du support reçoivent le même poids quel que soit leur différence au pixel courant.

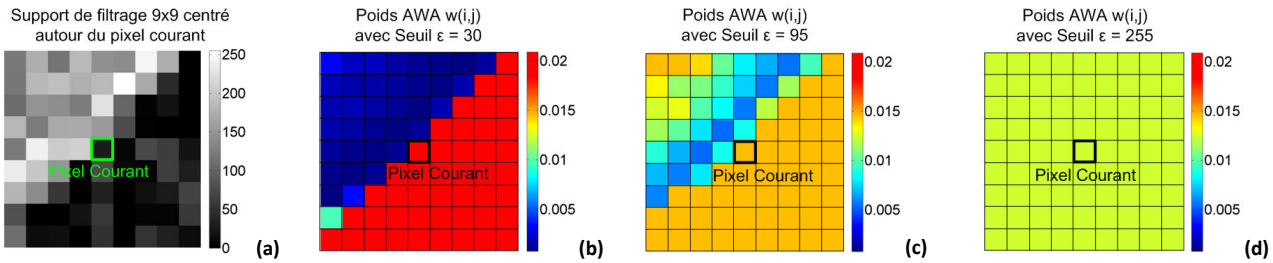


Figure 2.26. Illustration de l'influence du seuil ε sur le calcul des poids 9x9
(a) Support de filtrage AWA 9x9 centré sur le pixel courant appartenant à un contour - (b) Poids AWA attribués au pixel du support pour un seuil ε égal à 30 – (c) Poids AWA attribués au pixel du support pour un seuil ε égal à 95 – (d) Poids AWA attribués au pixel du support pour un seuil ε égal à 255

2. 4. 2. 2. Application du filtre AWA pour la réduction de bruit

Le filtre AWA a été proposé en 1993 pour la réduction de bruit dans les images d'une séquence vidéo [88]. Les auteurs du filtre AWA proposent de régler le paramètre ε^2 à deux fois la variance de bruit estimée dans l'image. Ainsi les différences quadratiques $\Delta p^2(i,j)$ inférieures à deux fois la variance de bruit sont considérées comme uniquement dues à la présence de bruit, alors que les différences quadratiques $\Delta p^2(i,j)$ supérieures sont considérées comme la structure de l'image qui auraient été présentes même sans présence de bruit. Le paramètre ε est donc un seuil exprimé en luminance en dessous duquel les différences entre des pixels voisins sont considérées comme du bruit à réduire.

[8] Il faut noter que nous prenons un support de filtrage 9x9 pour l'exemple mais les auteurs préconisent de ne pas dépasser une taille de 5x5 pour ne pas introduire trop de flou dans l'image filtrée.

La Figure 2.27 donne un exemple d'utilisation du filtre AWA avec un support 9x9 pour la réduction de bruit de l'image Lena (256x256) bruitée artificiellement avec un bruit blanc gaussien d'écart-type 30. L'ajout de bruit introduit des différences de luminance entre des pixels voisins initialement très ressemblants. La réduction du bruit par l'action du filtre AWA permet d'améliorer la note de PSNR et donc de diminuer les distorsions par rapport à l'image originale.

Nous prenons deux supports de filtrage particuliers dans l'image bruitée pour analyser l'action du filtre AWA. Les deux supports de filtrage sont présentés agrandis dans la Figure 2.28.

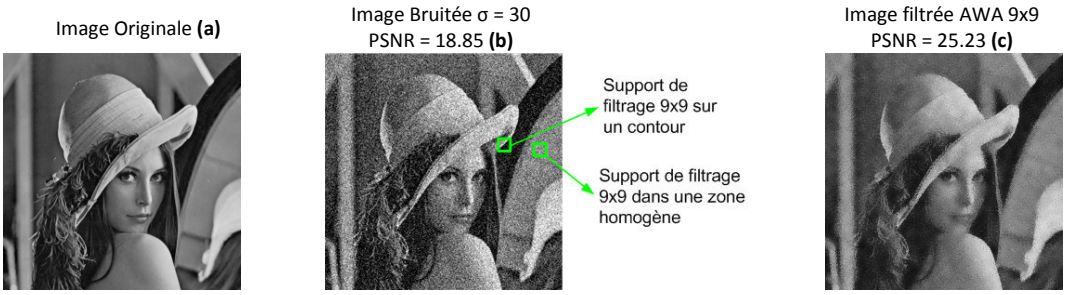


Figure 2.27. Exemple de réduction de bruit avec un filtre AWA 9x9
(a) Image Lena originale – (b) Image Lena bruitée avec un bruit blanc gaussien d'écart-type σ=30 – (c) Image Lena filtrée avec d'un filtre AWA 9x9, a=1, ε²=2* σ²

Pour l'exemple, nous avons choisi un support de filtrage 9x9 autour d'un pixel appartenant à un contour (a) et à une zone homogène (b) afin de mettre en avant l'adaptation du filtre AWA à ces différents types de contenus. Nous rappelons que le seuil ε^2 est fixé à 2 fois la variance de bruit et le paramètre *a* à 1. Pour chacun des supports, les poids AWA normalisés *W(i,j)* sont représentés sur la même échelle.

Le pixel de contour (a) est uniquement filtré avec ses voisins ressemblants permettant ainsi de garantir la netteté du contour du chapeau, tandis que le pixel appartenant à une zone homogène (b) est quasiment moyenné avec tous ses voisins permettant ainsi de réduire le bruit.

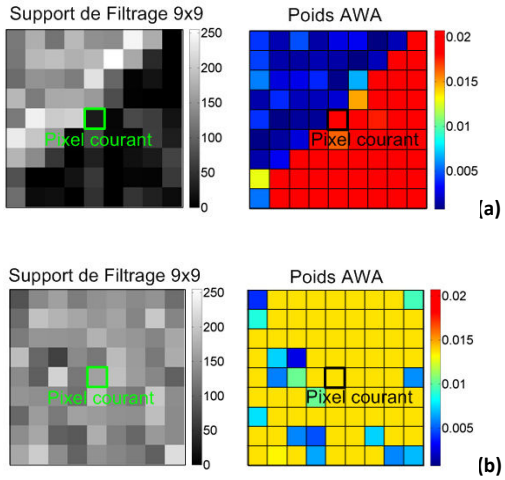


Figure 2.28. Illustration du calcul des poids AWA
(a) Pour un pixel appartenant à un contour – (b) Pour un pixel appartenant à une zone homogène

2. 4. 3. Intégration du modèle JND dans le filtre AWA

Notre objectif est de définir un filtre passe-bas guidé par un modèle psycho-visuel pour réduire les détails non perceptibles des images d'une séquence vidéo. L'intégration du modèle JND dans le filtre AWA est réalisée à partir des observations suivantes :

D'une part, le modèle JND de Yang que nous avons détaillé précédemment (2. 3. 2.), attribue à chaque pixel un seuil de visibilité exprimé comme la plus grande différence de luminance non perceptible.

D'autre part, le paramètre ε du filtre AWA représente un seuil de différence en luminance. L'intégration du JND comme contrôle du filtre AWA se fait donc naturellement en définissant ε = JND. Ainsi les poids du filtre AWA perceptuel s'expriment :

$$w_{JND}(i,j) = Z(x,y) \times Y(i,j) = \frac{Z(x,y)}{1 + a \times max(JND(x,y)^2, \Delta p(i,j)^2)}$$

Équation 2.13. *Poids AWA perceptuel*

Le JND utilisé ici est uniquement spatial, en effet, nous avons vérifié que le JND temporel a peu d'influence sur le JND spatio-temporel, ainsi afin de limiter la complexité de notre algorithme de prétraitement nous n'intégrons pas le masquage temporel.

Pour accélérer l'implémentation, nous proposons une simplification du filtre AWA en utilisant uniquement des poids binaires. Les pixels du support dont la différence par rapport au pixel courant est inférieure au seuil JND sont pris dans le calcul de la valeur filtrée, les autres pixels sont rejetés. Le procédé de filtrage simplifié, revenant à appliquer une moyenne conditionnelle, s'exprime par :

$$w_{JND}(i,j) = \begin{cases} 1, & \Delta p(i,j)^2 < JND(x,y)^2 \\ 0, & sinon \end{cases}$$

Équation 2.14. *Poids moyenneur conditionnel*

Le Tableau 2.1 présente la réduction du temps de calcul apporté par la simplification proposée sur la séquence *Binocular* (720p). Les filtres AWA perceptuel et sa version simplifiée ont été appliqués à la première image de la séquence puis aux 100 premières images. Les temps d'exécution absolus des deux filtres ne sont pas à retenir car le code utilisé n'est pas optimisé, seuls les réductions de temps de calcul sont réellement intéressantes. Dix essais ont été réalisés pour les deux tests, en moyenne le filtre simplifié apporte une réduction de l'ordre de 28 % du temps de calcul.

	Test 1 : 1 Image			**Test2 : 100 Image**		
	AWA(JND) [s]	AWA(JND) simplifié [s]	Réduction[%]	AWA(JND) [s]	AWA(JND) simplifié [s]	Réduction[%]
Image 1	0,2216	0,1587	-28,38	21,7474	15,4855	-28,79
Image 2	0,2202	0,1573	-28,58	21,7695	15,4735	-28,92
Image 3	0,2199	0,1573	-28,48	21,7812	15,5296	-28,70
Image 4	0,2199	0,1573	-28,49	21,7806	15,4782	-28,94
Image 5	0,2198	0,1573	-28,44	21,7678	15,4557	-29,00
Image 6	0,2216	0,1576	-28,92	21,7832	15,4845	-28,92
Image 7	0,2199	0,1571	-28,56	21,7703	15,4815	-28,89
Image 8	0,2199	0,1595	-27,49	21,7726	15,5170	-28,73
Image 9	0,2199	0,1576	-28,35	21,7694	15,4655	-28,96
Image 10	0,2198	0,1572	-28,48	21,7787	15,5016	-28,82
Moyenne	**0,2203**	**0,1577**	**-28,42**	**21,7721**	**15,4873**	**-28,87**

Tableau 2.1. Comparaison des temps d'exécution des filtres AWA perceptuel original et simplifié AWA(JND) : (Équation 2.13) et AWA(JND) simplifié : (Équation 2.14) Test réalisé sur les images de la séquence Binocular

La Figure 2.29 présente les poids calculés par le filtre AWA perceptuel et sa version simplifiée pour trois types de contenus : un pixel de contour (a), un pixel appartenant à une texture (b) et un pixel appartenant à une zone homogène (c). Comme nous l'avons vu précédemment, le modèle JND de Yang défini les contours et les zones homogènes d'une image comme perceptuellement importants alors que les zones texturées peuvent abriter des distorsions sans que celles-ci ne soient perçues. Ainsi, le seuil JND du pixel de texture à filtrer dans l'exemple (b) est plus élevé que celui des deux autres exemples. Contrairement au filtre AWA, le filtre AWA perceptuel simplifié attribue des poids binaires aux pixels du support.

Dans le cas du filtre AWA perceptuel simplifié (troisième colonne), le pixel de contour à préserver est uniquement moyenné avec ses voisins appartenant au contour. Le pixel de texture pouvant abriter plus de distorsions est moyenné avec un plus grand nombre de voisins conformément au seuil JND. Le pixel appartenant à une zone homogène a un seuil JND très faible puisque les deux masquages en luminance et en texture le définissent comme important. Toutefois ce pixel est celui qui est moyenné avec le plus de voisins, car ces voisins sont pratiquement identiques. Ainsi bien que le support de filtrage indique une force de filtrage élevée, le pixel courant sera très peu modifié par le filtrage.

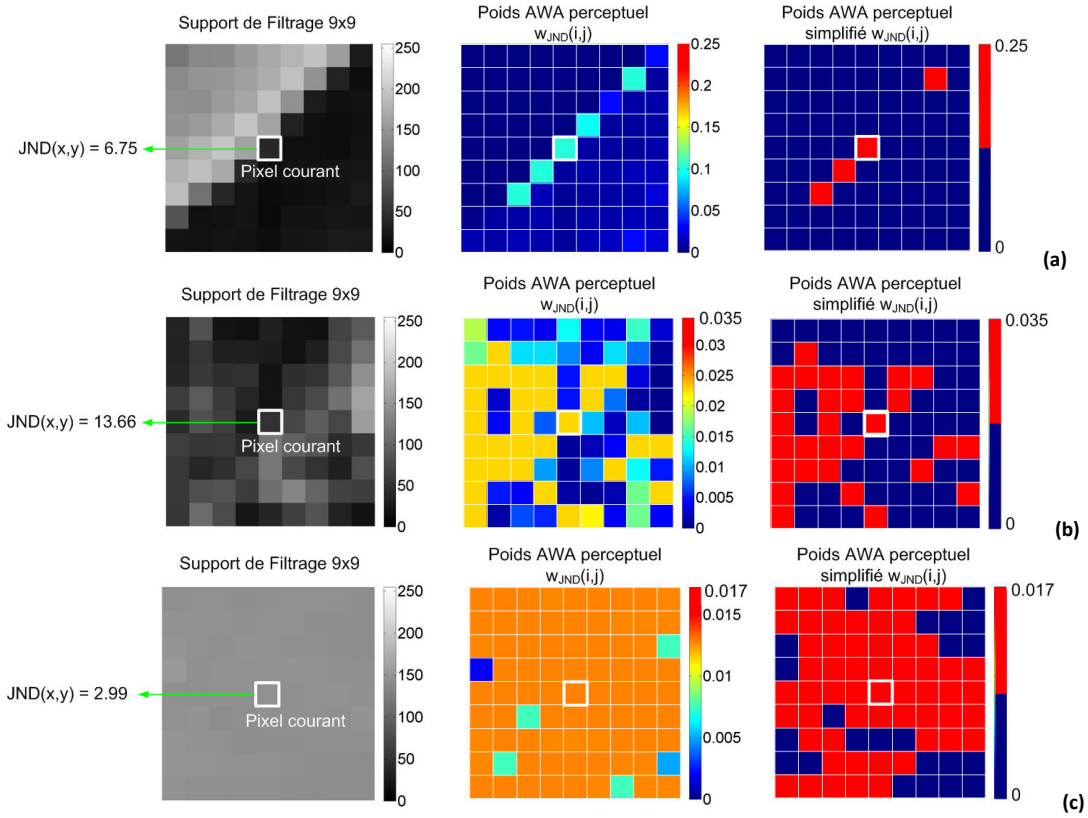


Figure 2.29. Comparaison des poids attribués par les filtres AWA perceptuel et sa version simplifiée
(a) support de filtrage centré sur un pixel de contour – (b) support de filtrage 9x9 centré sur un pixel de texture – (c) support de filtrage sur un pixel appartenant à une zone homogène

Finalement, la Figure 2.30 présente l'intérêt du filtre AWA simplifié contrôlé par le JND par rapport au filtre gaussien. Pour cela, l'image Lena (agrandissement a) est filtrée avec le filtre AWA simplifié 3x3 (b) en réglant le seuil perceptif à deux fois le JND. Ensuite l'image est filtrée par un filtre gaussien 3x3 dont la variance est choisie pour donner le même PSNR que l'image filtrée avec le filtre AWA. Ainsi nous comparons les images à même niveau moyen de distorsion. L'image traitée avec le filtre perceptuel que nous proposons paraît plus nette car les contours (yeux, nez, chapeau) sont préservés, en regardant attentivement on remarque que la peau est plus filtrée par le filtre AWA comparativement à la version filtrée avec le filtre gaussien.

La Figure 2.30 présente également des cartes de différences amenées dans l'image par l'application du filtre AWA (e) et le filtre Gaussien (f), les deux cartes sont représentées en échelle logarithmique. On note que la majorité des différences apportées par le filtre Gaussien sont concentrées aux contours et dans les zones texturées. En comparaison, le filtre AWA perceptuel simplifié suit la carte de JND (d) et concentre sa force de filtrage dans les zones texturées (plumes du chapeau) et sombres (cheveux, cadre de la fenêtre).

Ainsi, la réduction de contenu haute-fréquence réalisée par le filtre que nous proposons se concentre dans les zones perceptuellement peu sensibles et permet ainsi de limiter la sensation de flou apportée par des filtres passe-bas classiques. Dans la suite de ce chapitre, nous appliquons le préfiltre proposé en prétraitement de l'encodeur H.264/AVC dans le but de réduire le débit nécessaire à la représentation des séquences vidéo tout en conservant la qualité perçue.

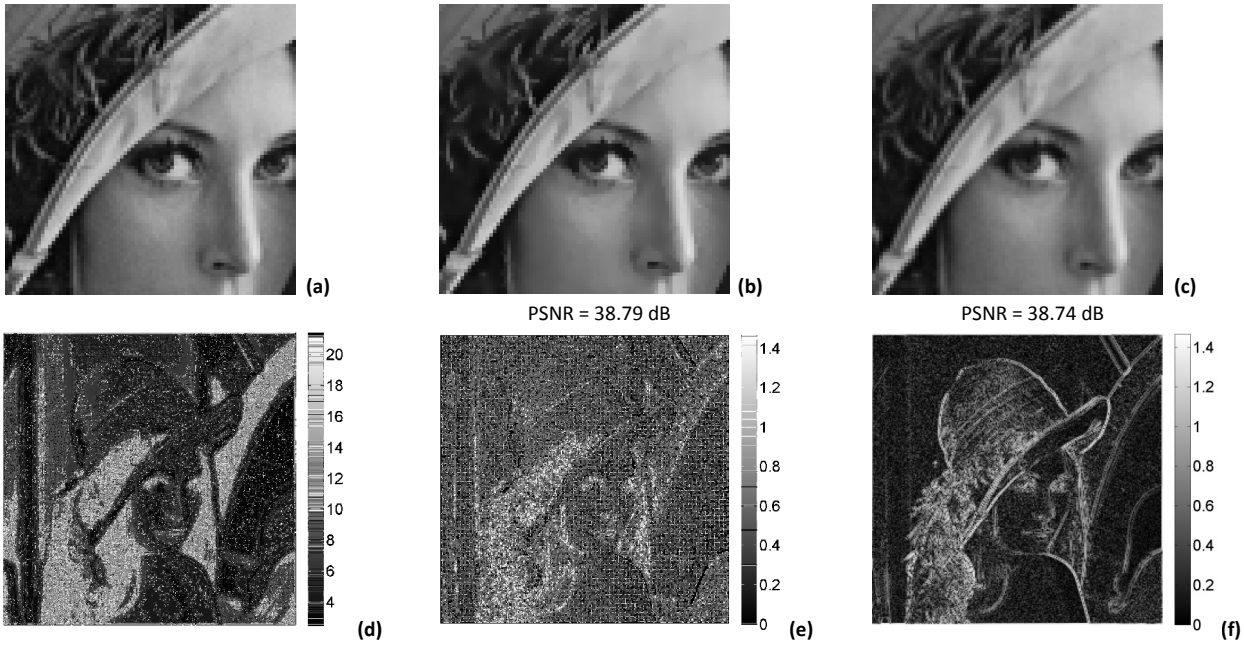


Figure 2.30. Comparaison du filtre AWA perceptuel simplifié et du filtre gaussien à même PSNR
(a) Agrandissement de l'image originale – (b) Agrandissement de l'image filtrée avec le filtre AWA perceptuel simplifié, seuil = JND – (c) Agrandissement de l'image filtrée par un filtre gaussien de taille 3x3 et d'écart-type σ = 0.47 – (d) Carte JND spatial – (e) Logarithme des différences amenées par le filtre AWA - (f) Logarithme des différences amenées par le filtre gaussien

2. 5. Le filtre perceptuel comme prétraitement de l'encodeur H.264/AVC

Le filtre AWA perceptuel que nous venons de présenter (Équation 2.14) est appliqué aux images de séquences vidéo comme préfiltre avant un encodage H.264/AVC afin de réduire les détails peu/pas perceptibles et ainsi diminuer le débit nécessaire à la représentation de la séquence sans altérer la qualité perçue. Deux séries de tests ont été réalisées pour des séquences SD et HD. La Figure 2.31 présente la chaîne de test utilisée, le calcul du JND par image ainsi que le filtrage sont réalisés à l'aide de Matlab, la version originale ainsi que la version filtrée sont encodées via l'encodeur AQILIM et x264 respectivement pour le test SD et HD. Le logiciel FFmpeg est utilisé pour le décodage et l'analyse des résultats est finalement réalisé avec Matlab et un banc de tests subjectifs mis en place à Digigram.

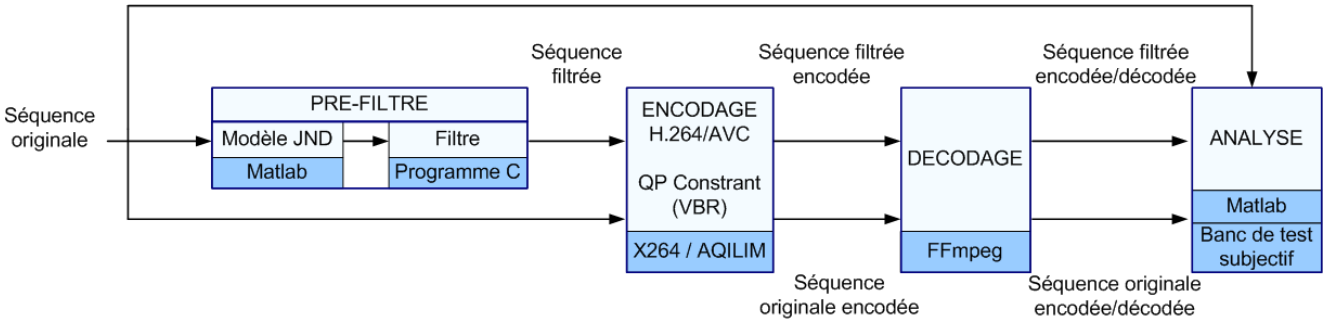


Figure 2.31. Chaîne de test pour la proposition d'un préfiltre avant encodage H.264/AVC

2. 5. 1. Analyse de résultats : Métriques objectives et tests subjectifs

La mesure de résultat a été une problématique récurrente tout au long des travaux de thèse. Dans le cas de nos travaux pour la réduction de débit à qualité constante en encodage H.264/AVC VBR, la réduction de débit apportée par l'application du préfiltre est valide uniquement si la qualité de la séquence filtrée et encodée n'est pas détériorée. En effet, l'application d'un simple filtre moyenneur sur une séquence avant encodage permet de réduire drastiquement le débit, mais la séquence filtrée et encodée présente un effet de flou non-acceptable. La mesure de qualité est donc un sujet important qui a concentré toute notre attention.

La mesure de qualité se sépare en deux catégories : les mesures objectives et subjectives. Les métriques objectives exploitent le signal vidéo pour générer une note de qualité par image d'une séquence ou par séquence. Ces métriques ont l'avantage de donner des résultats reproductibles et d'être automatisables. Cependant elles ne sont pas toujours bien corrélées avec l'impression ressentie par des observateurs humains. C'est pourquoi il est essentiel de coupler l'analyse objective à des tests subjectifs faisant intervenir des observateurs humains.

Les tests subjectifs, aujourd'hui incontournables en traitement vidéo sont décrits par le groupe de travail VQEG (Video Quality Expert Group) dans la recommandation [89]. Plusieurs techniques de tests subjectifs sont proposées, nous détaillerons par la suite la méthode PC (Paired Comparison) que nous avons utilisée pour valider nos résultats.

2. 5. 1. 1. Métriques objectives

Nous différencions les métriques objectives qui mesurent la qualité globale d'une image/séquence ayant subit une transformation par rapport à sa version originale, des métriques qui se concentrent à détecter la présence d'un artefact particulier (flou, effet de bloc, etc). On présente succinctement trois métriques globales que nous avons utilisées pour évaluer les performances du préfiltre AWA perceptuel.

- **Le PSNR (Peak Signal to Noise Ratio)**

Le PSNR que nous avons abordé précédemment est la métrique la plus répandue dans le domaine du traitement image/vidéo. Les raisons de cet usage sont sa simplicité algorithmique ainsi que la clarté de son sens physique. Cependant il est reconnu par l'ensemble de la communauté scientifique que le PSNR est mal corrélé avec la perception de qualité du système visuel humain [90], [72].

- **Le SSIM (Structural SIMilarity)**

Le SSIM, proposé en 2004 par Alan Bovik et al. [29] (Cf. Equation 1.25), a connu un réel succès dans le monde industriel comme scientifique et est aujourd'hui couramment utilisé en plus du PSNR. Le SSIM considère trois types de dégradations, la modification de la structure ($s(x,y)$) représentée par le coefficient de linéarité, la modification de luminance ($l(x,y)$) et de contraste ($c(x,y)$) représentée par la variance. Ces trois indices sont calculés sur le voisinage autour de chaque pixel, au lieu de considérer les pixels de l'image indépendamment les uns des autres, la Figure 2.32 présente cette différence de fonctionnement. Le critère Le SSIM varie entre 0.0 et 1.0, la note 1.0 correspondant à deux images identiques.

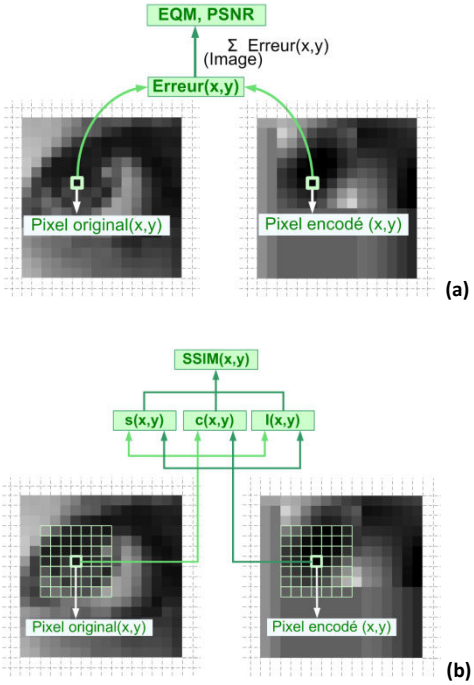


Figure 2.32. Principe de calcul du PSNR et du SSIM (a) Calcul du PSNR via l'EQM pixel à pixel – (b) Calcul du SSIM par comparaison de voisinage autour de chaque pixel

La Figure 2.33 présente les résultats obtenus avec la métrique SSIM pour une série de dégradations amenant la même note EQM (Erreur Quadratique Moyenne) et donc PSNR, mais présentant pourtant des qualités perçues très différentes. L'image floutée (e) a autant de différences pixel à pixel avec l'image originale que la version où le contraste est augmenté (b), menant les deux images à présenter la même note EQM. Le SSIM, en considérant la structure de l'image autour de chaque pixel, parvient à classer les images présentant une transformation linéaire (b) et (c) comme étant de bonne qualité, et donne une note plus basse aux versions dont la dégradation modifie la structure même de l'image. Cet exemple issu de l'article [29] montre l'intérêt de la métrique SSIM, cependant nous devons noter qu'au cours de nos expérimentations, les notes de SSIM calculées sur nos différentes expériences, ont rarement contredit les notes de PSNR. Ainsi nous employons de manière automatique le SSIM de par sa large acceptation dans le monde industriel en restant toutefois critique quant à sa réelle corrélation avec la perception de qualité.

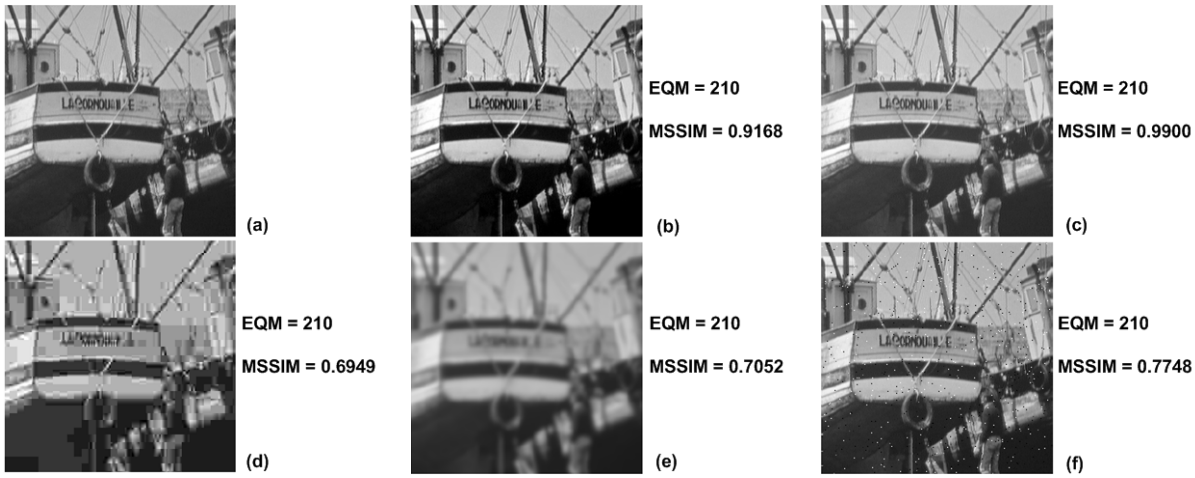


Figure 2.33. Illustration de la corrélation de la métrique SSIM avec l'impression visuelle
Illustration issue de la publication [29]
(a) Image originale – (b) Augmentation de contraste – (c) Réduction de luminance moyenne – (d) Image compressée en JPEG - (e) Image floutée – (f) Image contaminée par du bruit impulsionnel de type poivre et sel

- **La métrique VQM**

Le VQM (Video Quality Metric), proposé en 2004 [91] et standardisée par l'institut américain NTIA/ITS. Cette métrique détecte plusieurs types de changement intervenant entre une séquence traitée et sa version originale et combine ensuite les différentes mesures pour donner une note globale à la séquence qui varie entre 0.0 (la qualité de la séquence traitée est identique à l'originale) et 1.0 (la séquence traitée est de mauvaise qualité). La métrique VQM mesure notamment les différences de contraste et luminosité, ainsi que la présence de flou et de bruit.

La définition de métriques objectives permettant de prédire la perception (subjective) de la qualité est un vaste sujet d'étude encore ouvert aujourd'hui. Le VQEG (Video Quality Expert Group) a proposé une méthode d'évaluation des métriques objectives basée sur une mesure de corrélation avec la perception [92]. Pour cela, des tests subjectifs faisant intervenir des observateurs humains sont nécessaires. D'après les études réalisées dans [93], [94], le VQM présente la meilleure corrélation avec les tests subjectifs parmi les trois métriques présentées ici, suivie du SSIM et du PSNR. Bien que le VQM soit reconnu dans le milieu académique, il faut noter que cette métrique est très peu rencontrée dans le monde industriel[9].

[9] On peut noter que Tektronix a développé un outil de mesure appelé PQA [122], basé sur le modèle JND de Sarnoff [123], utilisé par des industriels et des chaînes de télévision pour régler la qualité en sortie d'encodage.

2. 5. 1. 2. Tests subjectifs

Une évaluation subjective consiste à demander à un certain nombre d'observateurs de noter la qualité d'une séquence vidéo afin d'extraire une note moyenne réellement représentative. Il existe plusieurs méthodes d'évaluation subjective décrites dans les recommandations de l'ITU (International Telecommunication Unit) BT.500 [95]. Elles diffèrent principalement par l'échelle de notation utilisée, et l'ordonnancement des séquences présentées. Les trois types d'échelles de notation principalement utilisées sont représentés par la Figure 2.34. Lorsqu'on utilise une échelle absolue, on demande aux observateurs de noter la qualité de chaque séquence indépendamment les unes des autres. Une échelle en qualité (a) ou en dégradation (b) peut être utilisée. Les exemples donnés par la Figure 2.34 sont des échelles à 5 niveaux mais des échelles à 3, 7 et 9 niveaux peuvent être utilisées. Ces échelles peuvent être également utilisées de manière continue, les paliers sont alors indicatifs et l'observateur règle un curseur sur l'échelle continue. Pour nos travaux, nous avons choisi un autre type d'évaluation en utilisant une échelle de notation comparative (c). Une séquence est alors notée comparativement à une référence. On différencie également les méthodes « simple stimulus » où les séquences sont présentées les unes après les autres, des méthodes « double stimulus » où les séquences sont présentées par paire. La moyenne des notes données par les observateurs est appelée MOS (Mean Opinion Score).

5 – Excellent
4 – Bon
3 – Assez bon
2 – Médiocre
1 – Mauvais **(a)**

5 – Imperceptible
4 – Perceptible mais non gênant
3 – Légèrement gênant
2 – Gênant
1 – Très gênant **(b)**

-3 – Beaucoup moins bon
-2 – Moins bon
-1 – Légèrement moins bon
0 – Identique
1 – Légèrement mieux
2 – Mieux
3 – Beaucoup mieux **(c)**

Figure 2.34. Echelle de notations couramment utilisées pour les tests subjectifs
(a) Echelle absolue de qualité à 5 niveaux – (b) Echelle absolue de dégradation à 5 niveaux – (c) Echelle comparative à 7 niveaux

Les méthodes les plus rencontrées dans l'état de l'art des travaux sur le prétraitement et le codage perceptuel sont les méthodes DSCQS (Double Stimulus Continuous Quality Scale) [84], [96], [66] et la méthode DSIS (Double Stimuli Impairement Scale) [67], [69]. Afin de choisir la méthode la plus adaptée à nos travaux, nous avons suivi les conseils d'experts du domaine (Margaret Pinson, ITS, VQEG et Patrick Le Callet, CNRS IRCCyN, VQEG), qui nous ont orientés vers la méthode PC (Paired Comparison) avec une échelle comparative à sept niveaux (Figure 2.34 (c)). Nous détaillons la méthode PC dans le paragraphe suivant.

2. 5. 1. 3. Test subjectif mis en place à Digigram

En plus de la description des méthodes subjectives, l'ITU définit un environnement de test : luminosité de la salle, couleur des murs, distance des observateurs à l'écran, calibration de l'écran ainsi que le choix des séquences. Le budget nécessaire à la réalisation d'une salle respectant les conditions de l'ITU n'était pas envisageable pour Digigram, toutefois nous nous sommes attachés à réaliser un environnement de test au plus proche des recommandations de l'ITU.

La Figure 2.35 présente la salle de test mise en place à Digigram. Nous utilisons un écran de 102 cm de diagonale. La distance d'observation est donnée par la recommandation de l'ITU [95] en fonction de la diagonale ou hauteur d'écran. Pour des séquences aux formats 1280x720, la distance préconisée est

de 2.25m. Des rideaux blancs permettent de créer un espace neutre pour le bon déroulement des tests.

Figure 2.35. Photographie de la salle de test mise en place à Digigram

La méthode PC est présentée par la Figure 2.36, une comparaison est composée d'une séquence de référence de 10 secondes, suivie d'un gris moyen de 3 secondes, ensuite la séquence à évaluer est jouée durant 10 secondes et pour finir l'observateur à 10 secondes pour voter (a). Ensuite la deuxième comparaison commence. La figure (b) présente le logiciel développé par Digigram pour une comparaison particulière de la séquence *SunFlower* 1920x1080. Durant la phase de vote, l'observateur dispose d'un clavier où l'échelle de notation a été reportée.

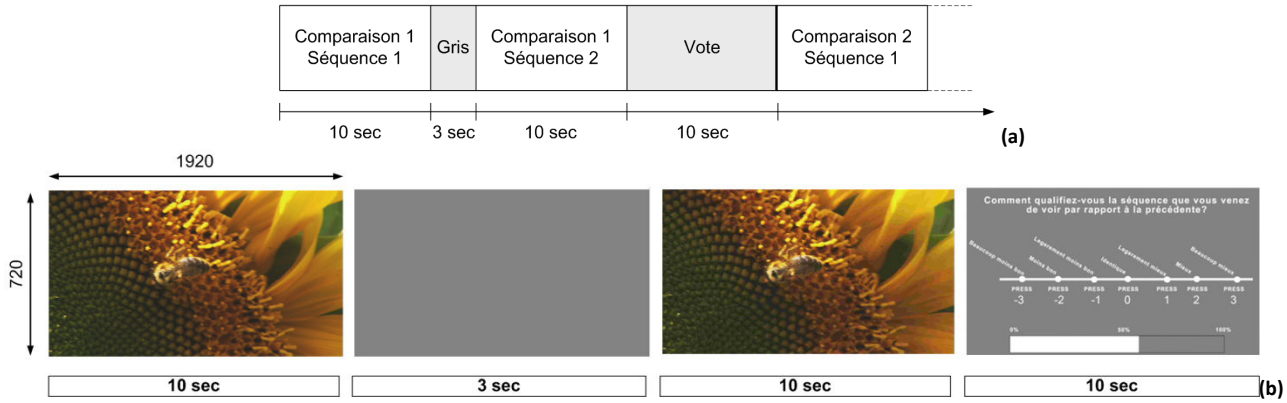


Figure 2.36. Séquence de test subjectif de type PC (Paired Comparison)
(a) Enchaînement des séquences décrit par la recommandation [95] – (b) Présentation du logiciel développé au sein de Digigram pour la méthode PC

2. 5. 2. Résultats

Dans cette dernière partie, nous présentons les résultats obtenus pour les deux séries de tests SD et HD. Les caractéristiques des deux tests sont récapitulées dans le Tableau 2.2. Les encodages sont réalisés en High Profile qui est le profil le plus largement utilisé par les clients de Digigram. Nous concentrons notre attention sur l'étude de l'effet du préfiltre proposé sur l'encodage Intra, car les images Intra demandent plus de débit dans un flux vidéo. Pour les tests HD, nous présenterons également des résultats en GOP inter classique IBBP12.

	Test SD	Test HD
Format source	704x576, 25p	1280x720, 50p
Echantillonnage couleur	4:0:0	4:2:0
Encodeur	AQILIM	x264
Profil	High	
Deblocking	Désactivé	
Codage Entropique	CABAC	
GOP	Intra	Intra et IBBP12
QP	22, 27, 32, 37	

Tableau 2.2. Caractéristiques des deux série de tests SD et HD

Bien que le PSNR et le SSIM aient été calculés pour tous nos tests, nous ne présenterons pas les résultats car la perte de détail apporté par le préfiltre, bien que peu/pas perceptible est systématiquement vue comme une dégradation qui réduit les notes de PSNR et SSIM. Un exemple de résultats de PSNR et SSIM est donné par la Figure 2.37 pour la séquence *Crew* 704x576 encodée avec et sans prétraitement par l'encodeur AQILIM. Une discussion avec Margaret Pinson (ITS, VQEG) nous a poussée à abandonner ces métriques pour se tourner vers la métrique VQM dans un premier temps et vers les tests subjectifs dans un deuxième temps.

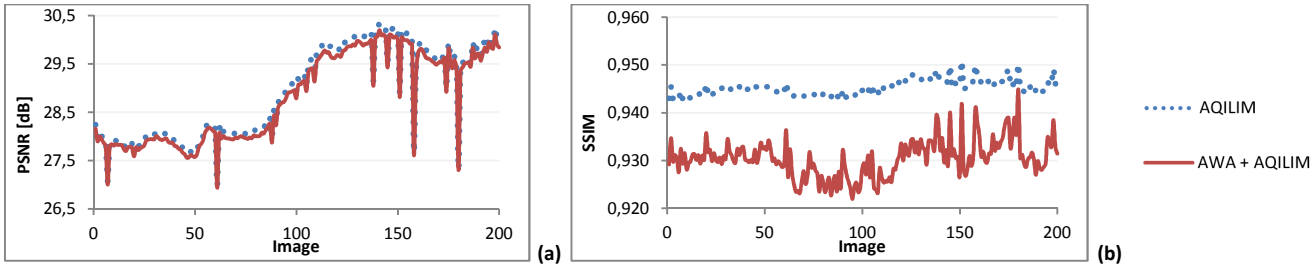


Figure 2.37. Courbes de PSNR et de SSIM par images de la séquence Crew 704x576 encodées avec et sans prétraitement (a) Résultats de PSNR – (b) Résultats de SSIM

2. 5. 2. 1. Première série de Test : Résultats SD

Dans la première série de tests publiés dans [97], le prétraitement perceptuel (AWA, Équation 2.13) est appliqué à quatre séquences au format 4CIF (704x576, format 4/3) et 768x432 (format 16/9) à 25 images progressives par seconde. Seule la luminance est filtrée et encodée. Pour cette première série de test, la version du préfiltre utilisée est le filtre AWA perceptuel défini par Équation 2.13, où le seuil perceptuel est dépendant du JND et utilise une fonction pour faire correspondre la valeur de JND aux différences de luminances existantes dans le support de filtrage [97]. Suite à ces premiers travaux, nous avons décidé d'intégrer directement le seuil JND au filtre AWA comme présenté précédemment. L'encodeur logiciel AQILIM de Digigram est utilisé pour encoder les vidéos originales et prétraitées, en High profile, en désactivant le filtre de réduction d'effet de blocs, à QP constant égal à 22, 27, 32 et 37.

Pour la mesure de qualité, nous avons utilisé la mesure de VQM à l'aide du logiciel pc 2.2 software [98] configuré avec le modèle « HRC Television ». Le Tableau 2.4 présente les débits et note VQM pour les versions encodées avec et sans prétraitement.

La première observation est que les réductions de débit amenées par le préfiltre sont logiquement dépendantes du contenu et diffèrent par conséquent en fonction des séquences tests. La Figure 2.38 montre les réductions de débit exprimées en pourcentage pour chacune des quatre séquences test et en moyenne sur les quatre séquences. Pour chaque série, seuls quatre points de mesure correspondant aux quatre QP testés sont représentés. Le préfiltre amène une réduction de débit spécialement importante pour la séquence *Crew*. Ceci peut s'expliquer par le fait que cette séquence est naturellement bruitée comparativement aux trois autres.

Ensuite, on note que les réductions de débit sont plus importantes à QP faible. Ce résultat était attendu car lorsque le QP augmente, la quantification joue elle-même le rôle de filtre passe-bas et minore l'action fine du préfiltre perceptuel.

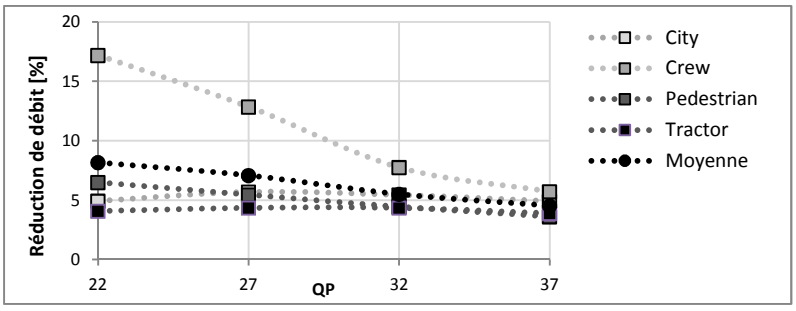


Figure 2.38. Réduction de débit par QP et par séquence SD

		AQILIM	Preproc. + AQILIM		AQILIM	Preproc. + AQILIM		AQILIM	Preproc. + AQILIM	
		Débit [Mbit/s]	Débit [Mbit/s]	ΔBItrate[%]	PSNR [dB]	PSNR [dB]	ΔPSNR	SSIM	SSIM	ΔSSIM
QP 22	City (704x576)	16,457	15,648	**-4,92**	29,19	29,04	**-0,53**	0,9538	0,9444	**-0,0094**
	Crew (704x576)	9,265	7,673	**-17,18**	28,92	28,80	**-0,41**	0,9452	0,9304	**-0,0148**
QP 27	City (704x576)	10,55	9,95	**-5,69**	28,84	28,69	**-0,53**	0,9293	0,9191	**-0,0103**
	Crew (704x576)	5,052	4,403	**-12,85**	28,72	28,63	**-0,33**	0,9178	0,9072	**-0,0106**
QP 32	City (704x576)	6,103	5,771	**-5,44**	28,11	27,98	**-0,48**	0,8805	0,8694	**-0,0111**
	Crew (704x576)	2,658	2,452	**-7,75**	28,39	28,31	**-0,26**	0,8802	0,8722	**-0,0080**
QP 37	City (704x576)	3,431	3,263	**-4,9**	27,07	26,97	**-0,39**	0,8058	0,7946	**-0,0112**
	Crew (704x576)	1,504	1,418	**-5,72**	27,94	27,88	**-0,24**	0,8341	0,8274	**-0,0067**

Tableau 2.3. Résultats de réduction de débit PSNR et SSIM pour la série de test SD (AQILIM) : séquences originales encodées avec l'encodeur AQILIM, (Preproc + AQILIM) : séquences préfiltrées encodées avec l'encodeur AQILIM

Afin de valider les réductions de débit amenées par notre préfiltre, nous avons voulu vérifier que la qualité visuelle des séquences encodées restait identique avec et sans prétraitement. Les notes de VQM (Tableau 2.4) montrent des différences très faibles de l'ordre de 3% entre les versions encodées avec et sans préfiltre. Une discussion avec l'auteur de la métrique (Margaret Pinson, [91]), nous a confirmé que des différences aux centièmes ne représentent aucune différence de qualité notable. Ainsi, pour une qualité perceptivement identique, le préfiltre que nous proposons amène en moyenne 6.3% de réduction de débit sur la gamme de débit 1.2-16.5 Mbit/s correspondant à une très bonne qualité pour des séquences SD. De plus, on note une réduction maximum de 17,1% pour la séquence *Crew* encodée à QP 22 correspondant environ à 8Mbit/s.

On peut noter que bien que les différences soient négligeables, les notes de VQM des séquences prétraitées sont toujours supérieures à celles des versions encodées sans préfiltre, c'est-à-dire que les versions filtrées sont considérées de moins bonne qualité. Ces observations nous ont poussées à nous tourner vers les tests subjectifs pour la suite de nos travaux.

La Figure 2.39 montre des parties et agrandissement d'une image tirée des deux versions encodées. Sur l'agrandissement prétraité (d), en observant très attentivement on perçoit une légère perte de détail comparativement à l'agrandissement encodé sans prétraitement (c), cependant les parties d'image (a) et (b) ne sont pas différenciables et d'autant moins lorsque les séquences sont jouées.

Figure 2.39. Image de la séquence Crew 4CIF encodée avec et sans prétraitement à QP 22 (correspondant à environ 8 Mbit/s) (a) Partie d'une image de la séquence Crew encodée sans préfiltre – (b) Partie d'une image de la séquence Crew encodée avec préfiltre à un débit 17,1% inférieur – (c) Agrandissement d'une image de la séquence Crew encodée sans préfiltre – (d) Agrandissement d'une image de la séquence Crew encodée avec préfiltre à un débit 17,1% inférieur

		AQILIM	Preproc. + AQILIM		AQILIM	Preproc. + AQILIM	
		Débit [Mbit/s]	Débit [Mbit/s]	ΔBItrate[%]	VQM	VQM	ΔVQM
QP 22	City (704x576)	16,457	15,648	-4,92	0,0086	0,0313	0,0227
	Crew (704x576)	9,265	7,673	-17,18	0,0176	0,0599	0,0423
	Pedestrain (768x432)	5,733	5,361	-6,49	0,0922	0,145	0,0528
	Tractor (768x432)	10,219	9,802	-4,08	0,0415	0,0754	0,0339
	Moyenne QP 22	-	-	**-8,17**	-	-	**0,0379**
QP 27	City (704x576)	10,55	9,95	-5,69	0,0372	0,0712	0,034
	Crew (704x576)	5,052	4,403	-12,85	0,0737	0,1162	0,0425
	Pedestrain (768x432)	3,56	3,366	-5,45	0,1545	0,1987	0,0442
	Tractor (768x432)	6,663	6,373	-4,35	0,0602	0,1044	0,0442
	Moyenne QP 27	-	-	**-7,08**	-	-	**0,0412**
QP 32	City (704x576)	6,103	5,771	-5,44	0,1299	0,1602	0,0303
	Crew (704x576)	2,658	2,452	-7,75	0,1811	0,2138	0,0327
	Pedestrain (768x432)	2,096	2,003	-4,44	0,2602	0,2944	0,0342
	Tractor (768x432)	3,95	3,778	-4,35	0,1332	0,163	0,0298
	Moyenne QP 32	-	-	**-5,5**	-	-	**0,0318**
QP 37	City (704x576)	3,431	3,263	-4,9	0,2826	0,3071	0,0245
	Crew (704x576)	1,504	1,418	-5,72	0,3225	0,3475	0,025
	Pedestrain (768x432)	1,242	1,197	-3,62	0,4055	0,4296	0,0241
	Tractor (768x432)	2,286	2,197	-3,89	0,2463	0,2711	0,0248
	Moyenne QP 37	-	-	**-4,53**	-	-	**0,0246**
Moyenne de la série de test		-	-	**-6,32**	-	-	**0,0339**

Tableau 2.4. Résultats de réduction de débit et de VQM pour la série de test SD (AQILIM) : séquences originales encodées avec l'encodeur AQILIM, (Preproc + AQILIM) : séquences préfiltrées encodées avec l'encodeur AQILIM

2. 5. 2. 2. **Deuxième série de Tests : Résultats HD**

Pour cette deuxième série de tests, nous utilisons l'encodeur x264 toujours en High profile en vue de la future intégration du préfiltre à l'encodeur, le filtre de réduction de l'effet de bloc est désactivé et le GOP Intra est utilisé. Les séquences tests sont cette fois HD (1280x720), seule la luminance est filtrée mais la séquence est encodée en 4:2:0. Pour cette série de test, nous avons réalisé des tests subjectifs avec 31 observateurs dont 5 experts en vidéo. Les réductions de débit ainsi que les notes subjectives (MOS) et objectives de ces tests sont regroupés dans le Tableau 2.7.

Afin de valider les réductions de débit apportées par le préfiltre, nous attendons du test subjectif qu'il confirme que la qualité perçue est identique après encodage. Le test subjectif que nous avons conçu comprend 21 comparaisons soit environ 12 minutes de test par observateurs. Le Tableau 2.5 présente l'ordre des comparaisons présentées aux 31 observateurs. Les notes MOS reportées dans la dernière colonne sont les notes moyennes que les 31 observateurs ont données à la deuxième séquence présentée par rapport à la première. L'échelle de notation MOS est reportée à côté du tableau. Prenons la comparaison 4 pour l'exemple, la séquence *IntoTree* encodée avec filtrage est présentée en première durant 10 secondes, suivie de 3 secondes de gris, puis la séquence encodée sans préfiltre est présentée également pendant 10 secondes. Cette dernière est alors notée par rapport à la première et en moyenne les observateurs lui ont attribué la note de -0,226 (sur une échelle à entre -3 et 3). Ils ont donc jugés en moyenne que la qualité de la séquence encodée sans préfiltre est très proche de celle encodée avec.

Lors de tests subjectifs plusieurs phénomènes peuvent influer sur les votes des observateurs, on note entre autres l'effet contextuel qui peut intervenir «*par exemple, si une image très dégradée est présentée après une séquence d'images peu dégradées, les observateurs peuvent, par inadvertance, évaluer cette image à un niveau inférieur que celui où ils l'auraient peut-être située normalement*» [95]. Ainsi afin d'améliorer la compréhension de nos résultats, nous avons inséré dans notre test des comparaisons qui ne nous intéressent pas directement pour valider le préfiltre, mais qui nous permettent d'affiner l'analyse ou de juger du biais des votes des observateurs.

Premièrement, la série de tests commence par la comparaison des trois séquences encodées à QP 37 (forte compression, mauvaise qualité) par rapport à la version brute (sans encodage et sans filtrage). Ces premières comparaisons nous ont permis de valider nos tests en vérifiant que les observateurs perçoivent bien les dégradations lorsqu'elles sont visibles. En effet dans notre cas, les versions des séquences encodées avec et sans filtrage sont de qualité très proche, le risque est d'obtenir uniquement des notes proches de zéro sans moyen de déterminer si les observateurs n'ont pas perçu de différences entre les séquences encodées avec et sans filtrage parce qu'elles ne sont pas visibles ou parce que les conditions de test ne permettent pas de les déceler. Les trois premières comparaisons nous permettent de confirmer que nos observateurs perçoivent bien des dégradations puisque les trois séquences encodées à QP37 sont jugées moins bonnes à beaucoup moins bonnes que les versions originales.

Deuxièmement, pour deux comparaisons (*ParkJoy* QP 27 (*) et *IntoTree* QP 32 (**)), nous avons réalisé le test deux fois en inversant l'ordre de comparaison. Ceci nous permet à la fois de nous assurer que le vote des observateurs est cohérent, ainsi que d'évaluer le bruit de mesure. En effet, ces tests étant par définition subjectifs, ils ne sont pas reproductibles.

Enfin, pour affiner nos résultats, nous avons inséré les comparaisons des séquences brutes et filtrées sans encodage. Ces comparaisons ne permettent pas de valider une réduction de débit puisqu'aucune compression n'est appliquée, mais elles permettent de valider le fait que le filtre AWA perceptuel réduit le contenu peu/pas perceptible des séquences vidéo. Les résultats de ces trois comparaisons (◊) sont très proches de zéro. Les séquences sont donc jugées de qualité identique.

Comparaison		Séquence	Référence	Séquence notée par rapport à la référence	Note MOS
1		ParkJoy	Brute	QP37	-2,677
2		Intotree	Brute	QP37	-2,839
3		Shield	Brute	QP37	-1,903
4		IntoTree	Prétraitée QP37	QP37	-0,226
5		ParkJoy	QP22	Prétraitée QP22	0,290
6		Shield	QP27	Prétraitée QP27	0,194
7	◊	ParkJoy	Prétraitée	Brute	-0,161
8		IntoTree	QP22	Prétraitée QP22	-0,581
9	*	ParkJoy	QP27	Prétraitée QP27	-0,065
10		Shield	Prétraitée QP27	QP27	0,098
11	◊	IntoTree	Prétraitée	Brute	0,065
12		IntoTree	QP27	Prétraitée QP27	-0,194
13		Shield	QP32	Prétraitée QP32	-0,226
14		ParkJoy	Prétraitée QP37	QP37	-0,548
15		Shield	QP22	Prétraitée QP22	0,065
16	◊	Shield	Brute	Prétraitée	0,065
17	**	IntoTree	QP32	Prétraitée QP32	-0,355
18		Shield	QP37	Prétraitée QP37	-0,452
19		ParkJoy	Prétraitée QP32	QP32	-0,129
20	**	IntoTree	QP32	Prétraitée QP32	-0,129
21	*	ParkJoy	Prétraitée QP27	QP27	0,226

-3 – **Beaucoup moins bon**
-2 – **Moins bon**
-1 – **Légèrement moins bon**
0 – **Identique**
1 – **Légèrement mieux**
2 – **Mieux**
3 – **Beaucoup mieux**

Echelle de notation MOS

Tableau 2.5. Résultats du test subjectif
(1,2,3) Vérification de la visibilité des dégradations – (◊) Comparaison des séquences brutes et filtrées sans compression – (*) Comparaison de la séquence ParkJoy à QP 27 réalisée deux fois en inversant le sens de comparaison – (**) Comparaison de la séquence IntoTree à QP 32 réalisée deux fois en inversant le sens de comparaison

2. 6. Analyse des résultats

Le Tableau 2.6 présente les statistiques des notes MOS présentées dans le Tableau 2.5, attribuées aux séquences prétraitées par rapport aux séquences originales[10]. En excluant les trois premières comparaisons, les versions préfiltrées des trois séquences, sans compression et aux quatre valeurs de QP, obtiennent des notes entre -0.581 et 0.548 avec une moyenne de -0.040 par rapport aux versions sans prétraitement. De plus 44,44% des notes des séquences prétraitées sont positives (jugées de meilleure qualité que les versions sans prétraitement). Les différences entre les versions originales et prétraitées avec ou sans compression ne sont donc pas perceptibles par notre panel d'observateurs.

Moyenne	-0,040
variance	0,079
Max	0,548
Min	-0,581
Notes positives [%]	44,44
Notes négatives [%]	55,56

Tableau 2.6. Statistiques des notes MOS attribuées aux séquences prétraitées par rapport aux séquences originales (avec et sans compression)

Le Tableau 2.7 présente les réductions de débits, ainsi que les notes MOS, PSNR et SSIM par séquence et par QP. En moyenne sur les trois séquences et les quatre QP, les séquences prétraitées encodées obtiennent une note MOS de -0.032 pour une réduction de débit de 5.10%. Comme nous l'avons dit précédemment, les métriques objectives PSNR et SSIM indiquent toujours une perte de qualité pour les séquences prétraitées. Ces métriques ne reflètent pas l'impression visuelle, cependant elles montrent que la quantité de différences introduites par le préfiltre diminue lorsque la quantification augmente. Cette observation est cohérente avec le fait que la réduction de débit amenée par le filtre diminue avec l'augmentation de la quantification.

La Figure 2.40 présente les réductions de débit pour les trois séquences HD à quatre valeurs de QP ainsi que les réductions moyennes sur les trois séquences. Les mêmes observations que pour les séquences SD peuvent être faites : la réduction de débit diminue avec l'augmentation du QP et dépend du contenu. La séquence *IntoTree* présente la plus forte réduction de débit du fait du fort pourcentage de zones détaillées contenues dans la scène, qui sont considérées par le JND comme peu sensibles aux dégradations et pouvant être fortement filtrées.

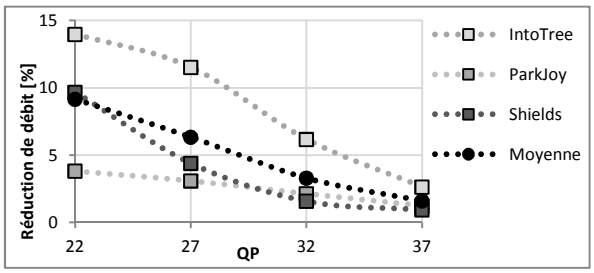


Figure 2.40. Réduction de débit par QP et par séquence HD

On détaille le cas de la séquence *IntoTree* encodée à QP 32 (comparaison 17 et 20). La Figure 2.41 présente la répartition des notes MOS données à la séquence *IntoTree* prétraitée par rapport à l'originale encodée à QP 32. La comparaison de ces séquences a été réalisée deux fois durant la session de test d'où les 62 valeurs. Les notes sont réparties entre zéro et ±1, avec une note moyenne est de -0.242. La qualité des deux séquences est donc jugée très proche pour une réduction de débit de 6,2%. Le débit moyen obtenu pour cette séquence à QP 32 est de l'ordre de 11Mbit/s ce qui correspond à la gamme de débit couramment utilisée pour encoder des séquences HD par les clients de Digigram.

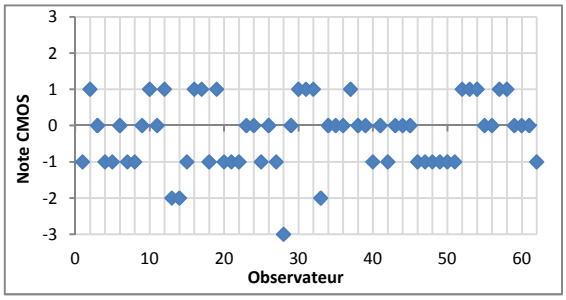


Figure 2.41. Notes MOS des 31 observateurs sur les 2 comparaisons de la séquence IntoTree prétraitée par rapport à la version originale encodée à QP32

[10] Pour établir des statistiques des notes CMOS attribuées aux séquences prétraitées par rapport aux séquences originales, on inverse le signe des notes présentées dans le Tableau 2.5 pour lesquelles la séquence non prétraitées est notée par rapport à la version prétraitée (comparaisons 4, 7, 10, 11, 14, 19, 21).

La Figure 2.42 présente une image de la séquence *IntoTree* encodée avec (b) et sans préfiltre (a) à QP 32 ainsi que des agrandissements d'une partie fortement filtrée (c) et (d). Sur les agrandissements le filtrage est perceptible (d) mais la perte de détails est très peu visible sur l'image entière (b).

Figure 2.42. Image de la séquence IntoTree 1280x720 encodée avec et sans prétraitement à QP 32 (correspondant à environ 11 Mbit/s) (a) Partie d'une image de la séquence IntoTree encodée sans préfiltre – (b) Partie d'une image de la séquence IntoTree encodée avec préfiltre à un débit 6,2% inférieur – (c) Agrandissement d'une image de la séquence IntoTree encodée sans préfiltre – (d) Agrandissement d'une image de la séquence IntoTree encodée avec préfiltre à un débit 6,2% inférieur

			Débit [Mbit/s]		ΔBitrate [%]	MOS	ΔPSNR	ΔSSIM
			x264	Preproc. + x264				
QP 22	IntoTree	(comparaison 8)	59,50	51,18	-13,98	-0,581	-1,47	-0,024
	ParkJoy	(comparaison 5)	127,77	122,89	-3,82	0,290	-1,81	-0,015
	Shields	(comparaison 15)	92,17	83,25	-9,68	0,065	-1,82	-0,023
	Moyenne QP 22		-	-	**-9,16**	**-0,075**	**-1,70**	**-0,021**
QP 27	IntoTree	(comparaison 12)	25,31	22,39	-11,53	-0,194	-0,53	-0,017
	ParkJoy (*)	(comparaison 9, 21)	81,64	79,12	-3,09	-0,145	-0,68	-0,014
	Shields	(comparaison 6)	53,11	50,78	-4,40	0,194	-0,55	-0,011
	Moyenne QP 27		-	-	**-6,34**	**-0,048**	**-0,59**	**-0,014**
QP 32	IntoTre (*)	(comparaison 17, 20)	11,14	10,45	-6,17	-0,242	-0,18	-0,008
	ParkJoy	(comparaison 19)	48,48	47,44	-2,13	0,129	-0,23	-0,011
	Shields	(comparaison 13)	31,11	30,61	-1,58	-0,226	-0,18	-0,005
	Moyenne QP 32		-	-	**-3,30**	**-0,113**	**-0,20**	**-0,008**
QP 37	IntoTree	(comparaison 4)	5,28	5,14	-2,63	0,226	-0,05	-0,003
	ParkJoy	(comparaison 14)	26,29	25,97	-1,23	0,548	-0,07	-0,006
	Shields	(comparaison 18)	18,18	18,01	-0,94	-0,452	-0,08	-0,004
	Moyenne QP 37		-	-	**-1,60**	**0,108**	**-0,07**	**-0,004**
Moyenne de la série de test			-	-	**-5,10**	**-0,032**	**-0,64**	**-0,012**

-3 – Beaucoup moins bon
-2 – Moins bon
-1 – Légèrement moins bon
0 – Identique
1 – Légèrement mieux
2 – Mieux
3 – Beaucoup mieux

Echelle de notation MOS

Tableau 2.7. Résultats de réduction de débit, de MOS de PSNR et SSIM pour la série de test HD encodée en GOP Intra (x264) : séquences originales encodées avec le codec x264, (Preproc + x264) : séquences préfiltrées encodées avec le codeur x264 (*) Note MOS obtenue en moyennant la note des deux comparaisons réalisées

Afin d'approfondir la validation du préfiltre que nous proposons, nous avons réalisé les tests en GOP Inter classique IBBP12. Le Tableau 2.8 présente les réductions de débit associées à l'application du prétraitement ainsi que les variations de PSNR et SSIM. La Figure 2.43, Figure 2.44 et Figure 2.45 présentent une comparaison visuelle d'une image issue de chacune des trois séquences testées encodées à QP22.

Les réductions de débits ainsi que les réductions de PSNR et SSIM en GOP IBBP12 (Tableau 2.8) moyennées sur les trois séquences sont du même ordre que celles obtenues en GOP Intra (Tableau 2.7).

Nous n'avons pas réalisé de tests subjectifs pour cette expérience, toutefois l'analyse visuelle nous a permis de confirmer que les versions préfiltrées des séquences encodées sont de qualité très similaire aux versions encodées sans préfiltre. Les Figure 2.43, Figure 2.44 et Figure 2.45 présentent les images dans le cas d'un encodage à QP22 qui est le cas où le prétraitement est le plus susceptible d'être visible. En effet, lorsque la quantification augmente, elle joue elle-même le rôle de filtre passe-bas et l'écart entre les versions encodées avec et sans préfiltre se réduit. Les agrandissements à QP22 montrent que la réduction de détail des versions prétraitées est légèrement visible à l'analyse détaillée, cependant elle est à peine perceptible à résolution native et lorsqu'elle est jouée.

		Débit [Mbit/s]		ΔBitrate [%]	ΔPSNR	ΔSSIM
		x264	Preproc. + x264			
QP 22	IntoTree	19,12	14,80	-22,56	-1,38	-0,029
	ParkJoy	49,68	49,88	0,39	-1,47	-0,017
	Shield	33,16	25,95	-21,74	-1,49	-0,022
	Moyenne QP 22	-	-	**-14,64**	**-1,44**	**-0,023**
QP 27	IntoTree	3,27	2,79	-14,65	-0,52	-0,018
	ParkJoy	25,30	25,23	-0,26	-0,56	-0,016
	Shield	8,76	8,58	-2,10	-0,25	-0,005
	Moyenne QP 27	-	-	**-5,67**	**-0,44**	**-0,013**
QP 32	IntoTree	1,22	1,11	-8,74	-0,23	-0,010
	ParkJoy	11,79	11,65	-1,17	-0,21	-0,013
	Shield	3,55	3,54	-0,44	-0,11	-0,004
	Moyenne QP 32	-	-	**-3,45**	**-0,18**	**-0,009**
QP 37	IntoTree	0,59	0,57	-3,67	-0,07	-0,004
	ParkJoy	5,28	5,21	-1,21	-0,08	-0,008
	Shield	1,89	1,88	-0,56	-0,07	-0,003
	Moyenne QP 37	-	-	**-1,81**	-0,07	-0,005
Moyenne globale		-	-	**-6,39**	**-0,54**	**-0,012**

Tableau 2.8. Résultats de réduction de débit et de MOS pour la série de test HD encodée en GOP IBBP12
(x264) : séquences originales encodées avec le codec x264
(Preproc + x264) : séquences préfiltrées encodées avec le codeur x264


(a)

(b)

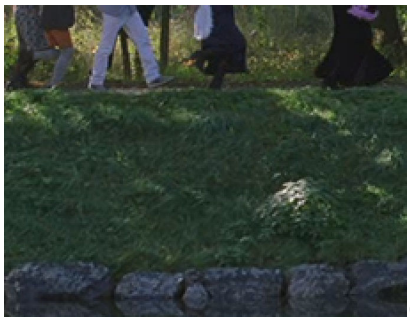

(c)


(d)

Figure 2.43. Comparaisons Visuelle d'une image de la séquence IntoTree 1280x720 50p encodée avec et sans préfiltre à QP22 en GOP IBBP12


(a)


(b)


(c)

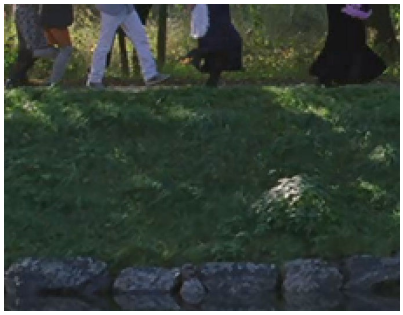

(d)

Figure 2.44. Comparaisons Visuelle d'une image de la séquence ParkJoy 1280x720 50p encodée avec et sans préfiltre à QP22 en GOP IBBP12


(a)


(b)


(c)


(d)

Figure 2.45. Comparaisons Visuelles d'une image de la séquence Shield 1280x720 50p encodée avec et sans préfiltre à QP22 en GOP IBBP12

(a) Image encodée sans préfiltre – (b) Image encodée avec préfiltre – (c) Agrandissement de l'image encodée sans préfiltre – (d) Agrandissement de l'image encodée avec préfiltre

2. 7. Conclusion

Dans ce chapitre nous avons présenté l'état de l'art des prétraitements pour l'encodage vidéo. Nos contraintes en termes d'implémentation et de qualité nous ont amenées à nous concentrer sur les prétraitements externes au codeur, guidés par un modèle perceptuel.

Nous avons ensuite présenté le principe du JND et décrit le modèle proposé par Yang et al., définissant un seuil de perception de distorsion par pixel. Nous avons intégré ce modèle dans le filtre passe-bas AWA, défini initialement pour la réduction de bruit, afin de réduire les détails peu ou pas perceptibles dans les séquences vidéo.

Ce prétraitement a ensuite été appliqué comme préfiltre avant l'encodage H.264/AVC afin de réduire le débit nécessaire à la représentation d'une séquence vidéo sans dégrader la qualité perçue. Nos tests ont été réalisés en encodage VBR (QP constant) sur des séquences SD et HD. Pour des besoins de mesure de qualité perçue, nous avons mis en place un protocole et une salle de test subjectif, qui seront dorénavant utilisés à Digigram pour évaluer les développements réalisés en interne ainsi que les différentes solutions du marché. Nos résultats mesurés en pourcentage de réduction de débit, VQM et tests subjectifs montrent une réduction moyenne de l'ordre de 5.5% sur l'ensemble de nos tests (QP 22 à 37, séquences SD et HD, GOP Intra et Inter) avec un maximum de 17% en SD et GOP Intra et 22% en HD et GOP Inter (IBBP12).

Dans le chapitre suivant nous proposons d'approfondir notre étude du préfiltrage perceptuel indépendant de l'encodeur, en étudiant d'autres filtres que le filtre AWA et en considérant un support de filtrage étendu pour traiter des contenus HD.

Chapitre 3. Etude des filtres perceptuels bilatéraux pour des applications de débruitage et de prétraitement pour l'encodage H.264/AVC en VBR et HD

3. 1. Introduction

En relaxant la contrainte de l'implémentation temps réel qui nous a poussé à limiter dans un premier temps le support du filtre AWA à une taille de 3x3 pixels, nous proposons d'étudier l'intérêt d'utiliser un support de filtrage plus étendu pour mieux exploiter les redondances spatiales des format HD. Pour cette étude nous serons amenés à considérer un autre filtre bien connu de la littérature des filtres adaptatifs, le filtre Bilatéral et à définir deux nouveaux filtres le BilAWA et le Bilatéral seuillé. Les filtres que nous proposons seront étudiés d'abord dans un contexte de réduction de bruit, puis en tant que prétraitement perceptuel pour l'encodage vidéo H.264.

Le Filtre AWA [88] sur lequel nous avons basé nos travaux de prétraitement perceptuel présentés au chapitre précédent, a été utilisé dans la littérature avec un support de taille 3x3 pour ne pas introduire d'effet de flou trop important sur des séquences majoritairement de taille CIF à SD [88], [44], [99], [43]. Dans le cadre de nos travaux, nous nous concentrons sur le traitement de résolutions HD. Or dans une image au format HD une même zone est représentée avec plus de pixels qu'au format SD, augmentant ainsi la corrélation entre les pixels voisins au sein d'une image. Ce principe est clairement visible sur les images (e) et (f) de la Figure 3.1 respectivement capturée avec une caméra SD et HD.

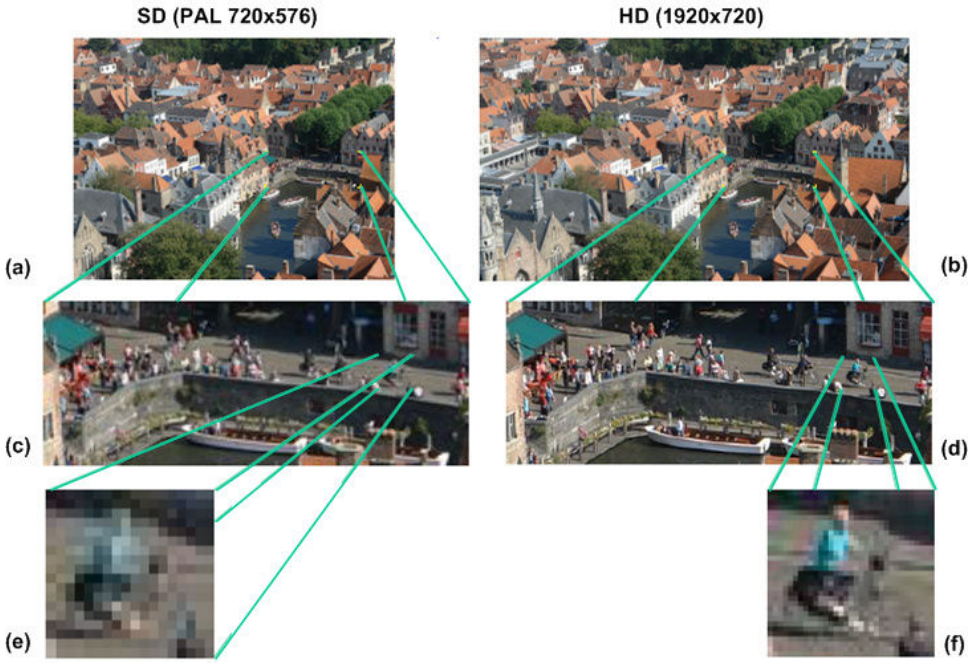


Figure 3.1. Illustration des différences entre une scène capturée en résolution SD et HD
Illustration issue de la présentation [100]

A partir de cette observation nous réalisons l'expérience simple décrite ci-après, dont les résultats sont donnés par la Figure 3.2. Une même image au format 854x480 (2ème ligne) et 1920x1080 (3ième ligne), sont filtrées à l'aide d'un filtre gaussien dont nous faisons augmenter le support de 3x3 à 11x11. Les images originales et filtrées sont ensuite encodées en JPEG au paramètre de qualité égal à 80. On note qu'au-delà du support 5x5, le filtre gaussien apporte une impression de flou très importante sur l'image SD. Pour l'image HD, les support 7x7 et 11x11 donnent une qualité acceptable et permettent de réduire plus fortement le nombre de bits nécessaire à la représentation de l'image.

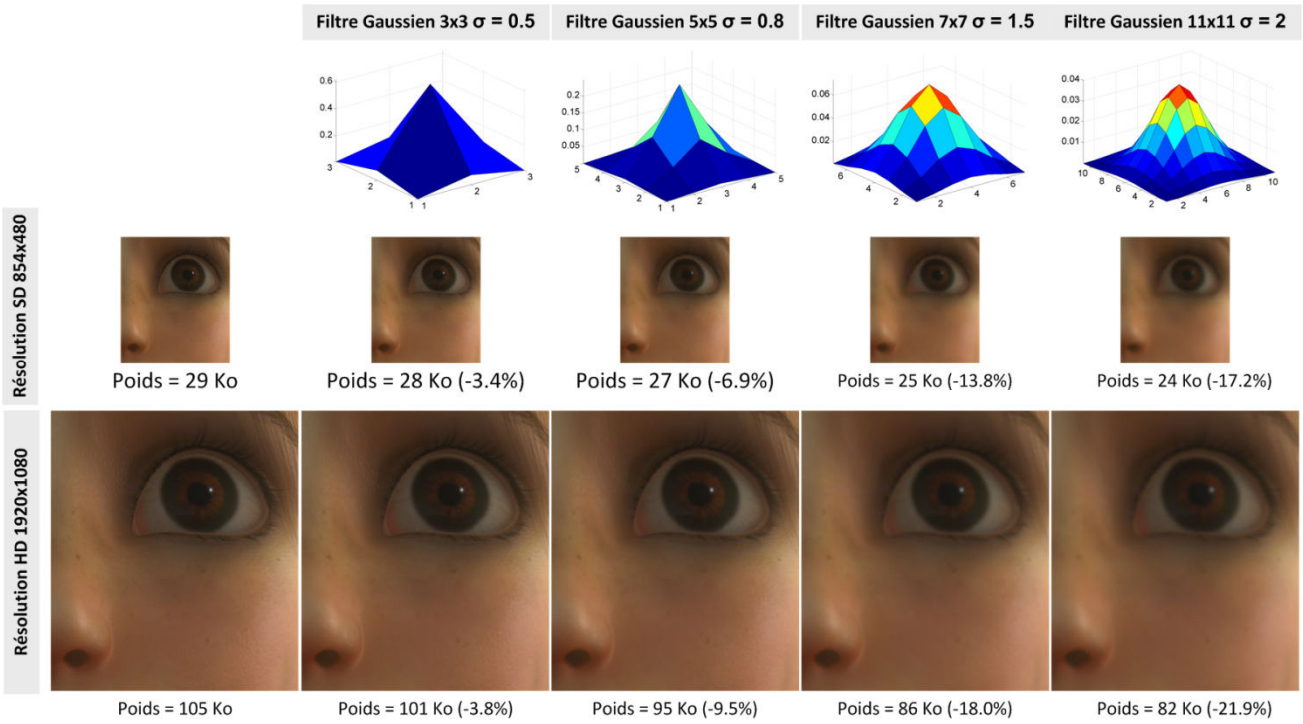


Figure 3.2. Illustration de la relation entre résolution, taille de support et compression
Cas filtre gaussien – Compression JPEG Qualité 80
(1ère ligne) support de filtrage utilisé – (2ème ligne) Partie d'une image SD 854x480 compressée sans filtrage (1ère colonne) et avec quatre taille de support – (3ème ligne) Partie d'une image HD 1920x1080 compressée sans filtrage (1ère colonne) et avec quatre taille de support

Ces observations nous poussent à considérer le cas de support de filtrage étendu pour améliorer les performances de notre préfiltre précédemment proposé pour les séquences HD. De plus nous considérons le filtre Bilatéral reconnu pour ses qualités de préservation des contours. Afin de trouver le meilleur compromis entre le filtre AWA et le filtre Bilatéral, nous serons amenés à proposer deux nouveaux filtres : le BilAWA et le Bilatéral seuillé. Les filtres AWA et Bilatéral appartenant à la littérature de la réduction de bruit, nous commencerons notre étude en comparant les performances de réduction de bruit des différents filtres étudiés, puis nous poursuivrons l'étude des filtres les plus performants dans une application de prétraitements perceptuels pour l'encodage H.264/AVC.

3. 2. Etat de l'art

Les filtres AWA et Bilatéral que nous étudions dans la suite de ce chapitre ont été développés pour la réduction de bruit. Il faut noter que les performances en termes de débruitage des filtres de voisinage ont été depuis surpassées par la famille de filtres non-locaux dont nous présentons ci-après le premier filtre proposé par [86]. Malgré les performances inégalées en débruitage de cette famille de filtres, nous ne les avons pas sélectionnés à cause de leur grande complexité algorithmique.

Le filtre Non-Local Means (NLM) proposé en 2005 conteste l'idée que les redondances spatiales d'une image existent uniquement entre voisins proches. Tous les pixels d'une image sont considérés comme des candidats potentiels pour appartenir au support de filtrage, le poids attribué à chaque pixel d'une image dépend de la ressemblance de son voisinage avec celui du pixel courant. De cette manière, on parvient à regrouper dans un support, tous les pixels d'une image ayant les mêmes caractéristiques, par exemple appartenant au même objet. La Figure 3.3, présente le principe de sélection des pixels du support pour un filtre de voisinage classique et pour le filtre NLM.

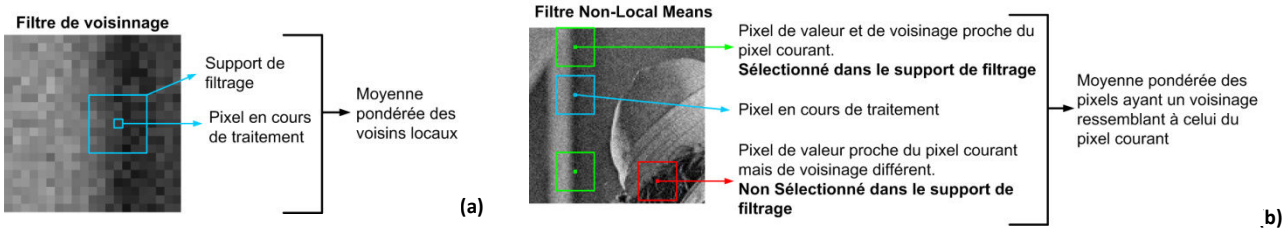


Figure 3.3. Principe des filtres locaux et non-locaux
(a) principe des filtres de voisinage – (b) principe du filtre NLM

Grâce aux performances remarquables de réduction de bruit du filtre NLM, les travaux basés sur une recherche non-locale de pixels similaires se sont étendus à la plupart des familles de filtres spatiaux. Toutefois, la recherche non-locale originalement proposée, possède une complexité algorithmique rendant très complexe l'implémentation en temps réel. De nombreuses propositions ont été faites pour réduire la complexité notamment à l'aide d'un pré-classement des voisinages à tester [101], [102]. On peut noter que le filtre BM3D [103] reposant sur une classification non-locale présente les meilleurs résultats de débruitage de l'état de l'art au prix d'une forte complexité qui ne permet pas d'envisager une intégration au sein d'une chaîne de codage temps-réel.

Comme nous l'avons présenté au chapitre précédent, le but final de l'étude est de proposer une solution embarquée au sein des encodeurs live proposés par Digigram, ainsi nous nous concentrons sur l'étude des filtres dits de voisinage qui appliquent à un pixel une moyenne pondérée de ses voisins locaux par convolution, comme défini par l'Équation 3-1 :

$$p_f(x,y) = \sum_{(i,j)\in S_{x,y}} w(i,j) \times p(i,j)$$

Équation 3-1. *Filtres de voisinage*

Avec $p_f(x, y)$ le pixel filtré de coordonnée x, y dans l'image et $w(i, j)$ le poids normalisé attribué au pixel $p(i, j)$ appartenant au support de filtrage $S_{x,y}$ centré sur le pixel courant. Les poids $w(i, j)$ sont définis par la fonction d'attribution des poids $Y_{Filtre}(i, j)$ divisés par la somme des poids du support de filtrage.

$$w(i,j) = \frac{Y_{Filtre}(i,j)}{\sum_{(i,j)\in S_{x,y}} Y_{Filtre}(i,j)}$$

Équation 3-2. *Poids normalisés des filtres de voisinage*

Le filtre Bilatéral est une technique de filtrage non linéaire proposée en 1998 pour la réduction de bruit par [87] qui a été largement utilisée pour ses performances en termes de conservation des contours. L'originalité de la fonction de filtrage du filtre Bilatéral $Y_{Bilateral}(i,j)$ décrite par l'Équation 3-3, réside dans l'utilisation de deux noyaux

gaussiens considérant la distance spatiale et photométrique entre les pixels d'une image. Dans la suite de ce chapitre nous appellerons filtre Géométrique le noyau gaussien $h_g(i,j)$ considérant la distance spatiale, et filtre de Similarité le noyau gaussien $h_s(i,j)$ considérant la distance photométrique.

$$Y_{Bilateral}(i,j) = h_g(i,j) \times h_s(i,j)$$ ***Équation 3-3.*** *Poids Filtre Bilatéral*

Le filtre Géométrique est un simple filtre gaussien qui concentre les poids sur les pixels spatialement proches du pixel central. La concentration des poids est contrôlée par la variance σ_g^2. Le filtre de Similarité est basé sur un noyau gaussien qui varie en fonction de la différence en luminance entre le pixel courant et les pixels du support de filtrage. De manière comparable au filtre AWA, le filtre de Similarité attribue des poids élevés aux pixels ressemblants au pixel courant. La décroissance des poids est contrôlée par la variance σ_s^2 du noyau gaussien

$$h_g(i,j) = exp\left(-\frac{|x-i|^2 \times |y-j|^2}{2\sigma_g^2}\right)$$ ***Équation 3-4.*** *Filtre Géométrique*

$$h_s(i,j) = exp\left(-\frac{\Delta p(i,j)^2}{2\sigma_s^2}\right)$$ ***Équation 3-5.*** *Filtre de Similarité*

$$\Delta p(i,j) = |p(x,y) - p(i,j)|$$

Parmi les travaux de réduction de bruit, on peut noter deux propositions d'amélioration du filtre Bilatéral original qui utilise un support de filtrage 11x11 sur des images de dimension au maximum de 512x512. La première [104] soutient l'hypothèse que la différence photométrique n'est pas suffisante en présence de bruit pour décider si des pixels sont ressemblants et doivent être filtrés ensemble. Ainsi les auteurs proposent d'ajouter un troisième noyau gaussien considérant des caractéristiques de structure de l'image comme l'énergie et la variance. Dans la deuxième solution [105], les auteurs remarquent que dans des images naturellement bruitées, le bruit possède des composantes basses et hautes fréquences. Ainsi ils proposent d'appliquer le filtre Bilatéral à la couche basse fréquence d'une décomposition en ondelettes, couplé à un seuillage des couches hautes fréquences. Cette solution permet d'approcher les performances des filtres les plus performants de l'état de l'art (NLM et BM3D) à moindre coût.

Le filtre Bilatéral a également été utilisé comme prétraitement perceptuel pour l'encodeur H.264/AVC (implémentation JM) dans le cas de séquences CIF encodées à très bas débit. Les auteurs de [106] proposent de sélectionner la variance des deux noyaux gaussiens en fonction d'une carte de saillance dans le but d'enlever des détails et représenter la séquence vidéo avec moins de bits. La méthode proposée amène des réductions de débits entre 20% et 50%, cependant la qualité des séquences prétraitées est clairement dégradée par un fort effet de flou comme le montre la Figure 3.4.

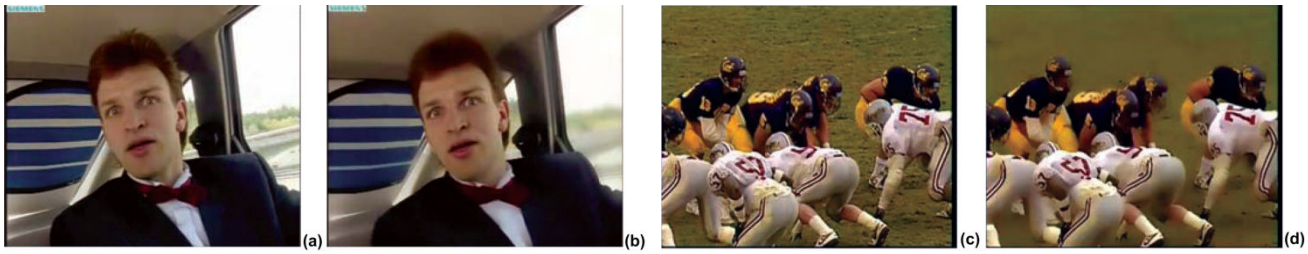


Figure 3.4. Résultats d'un préfiltre Bilatéral contrôlé par une carte de saillance
Illustration issue de la publication [106]
(a) Image de la séquence CarPhone CIF encodée sans prétraitement à QP 28 (292 kbps) – (b) avec le prétraitement proposé (243 kbps) – (c) Image de la séquence Football CIF encodée sans prétraitement à QP 28 (1843 kbps) – (d) avec le prétraitement proposé (1120 kbps)

Les auteurs de [59] proposent d'adapter uniquement la variance du noyau photométrique en fonction d'une mesure d'activité temporelle basée sur une différence inter-image afin de régler la force de filtrage et préserver les zones dynamiques. La solution proposée est comparée à un encodeur H.264 (implémentation JM) en encodage CBR à

bas débit (128 Kbps). L'amélioration de qualité apportée par cette solution n'est pas mesurée, des images issues des séquences encodées montrent une amélioration dans ce cas de codage en très basse qualité (Figure 3.5).

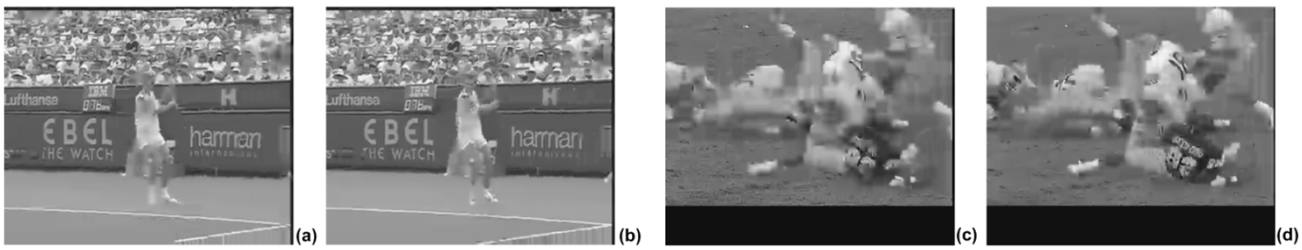


Figure 3.5. Résultats d'un préfiltre Bilatéral contrôlé par une mesure d'activité temporelle
Illustration issue de la publication [59]
Encodage CBR 128Kbps - (a) Image de la séquence Stephan CIF encodée sans prétraitement – (b) avec le prétraitement proposé – (c) Image de la séquence Football CIF encodée sans prétraitement – (d) avec le prétraitement proposé

Notre but est de proposer un prétraitement perceptuel réduisant le débit nécessaire à la représentation d'une vidéo sans altérer la qualité perçue, qui permette de surpasser les performances du préfiltre défini au chapitre précédent. Nous considérons pour cela un support de filtrage étendu et nous nous appuyons sur le filtre AWA précédemment étudié, ainsi que sur le filtre Bilatéral, autre filtre reconnu de la littérature. Les solutions que nous proposons dans la suite de ce chapitre en termes de réduction de bruit et de prétraitement perceptuel se différencient des travaux de la littérature par le niveau de qualité cible, l'utilisation dans un contexte HD, ainsi que l'attention portée à proposer des filtres de complexité contrôlée.

3. 3. Etude des filtres pour une application de Débruitage HD

Nous avons vu que le filtre AWA a été appliqué dans la littérature avec un support de filtrage 3x3, tandis que le filtre Bilatéral avec un support 11x11, majoritairement sur des séquences/images de résolution inférieure ou égale à la SD. En étendant le support de filtrage de 3x3 à 11x11 pixels nous entendons améliorer les performances de réduction de bruit du filtre AWA en exploitant les redondances d'images HD. Nous verrons que le filtre AWA parvient à réduire efficacement le bruit, cependant l'augmentation de la taille de support amène un effet de flou important qui détériore la qualité perçue, ceci étant dû à la faible décroissance des poids AWA. De plus nous verrons que le filtre Bilatéral parvient à conserver plus efficacement les contours et détails des images. Ainsi en cherchant le meilleur compromis entre réduction de bruit et préservation des détails, nous serons amenés à définir deux nouveaux filtres bilatéraux : le filtre BilAWA et le filtre Bilatéral seuillé.

3. 3. 1. Protocole expérimental

Dans le but de comparer le filtre AWA et le filtre Bilatéral, nous utilisons les paramètres suivants. Pour le filtre AWA nous utilisons les valeurs recommandées par les auteurs : le paramètre a est fixé à 1, tandis que le seuil ε est fixé à deux fois la variance de bruit. Pour le filtre Bilatéral, la variance σ_s^2 contrôlant la décroissance du filtre de Similarité est également fixée à deux fois la variance de bruit. La variance σ_g^2 contrôlant la concentration des poids au centre du support de filtrage est fixée à $\sqrt{1.8}$ dans la suite de notre étude, comme le propose une étude réalisée par les auteurs de [105].

3. 3. 1. 1. **Base d'images test**

Afin de maîtriser le niveau de bruit présent dans les images test et s'affranchir d'une estimation de variance de bruit, nous avons fait le choix de partir d'images de bonne qualité auxquelles nous avons ajouté artificiellement un bruit

blanc gaussien d'écart-type σ_n. Nous testons trois niveaux de bruit σ_n=10, 20 et 30. Nos tests sont réalisés sur une banque de huit images présentées par la Figure 3.6. Les images sont issues de séquences HD 1280x720, classiquement utilisées en traitement vidéo. Nous travaillons uniquement sur la composante de luminance de ces images.

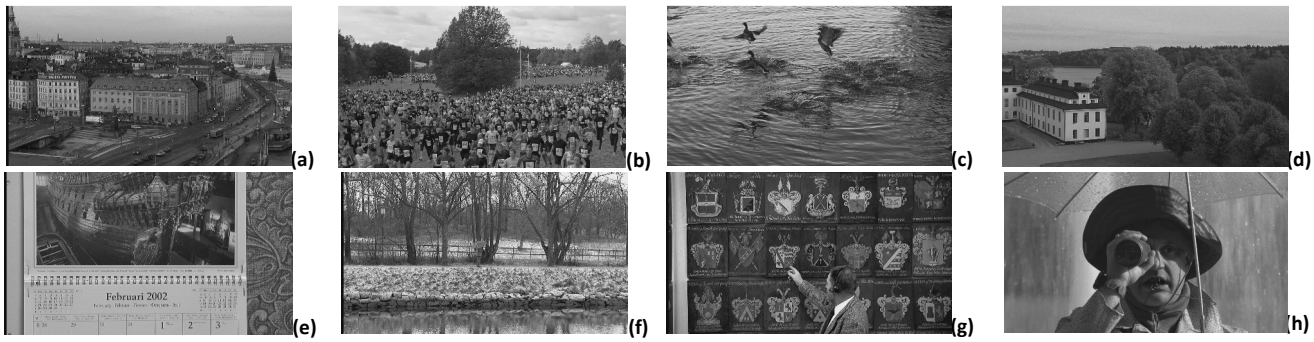


Figure 3.6. Images de la banque de test issues de séquences de 1280x720
(a) Stockholm ; (b) CrowdRun ; (c) Ducks ; (d) IntoTree ; (e) Mobile & Calendar; (f) ParkRun; (g) Shield; (h) Binocular

L'information spatiale des huit images a été mesurée à l'aide de la mesure SI (Spatial Perceptual Information) définie dans le standard [107]. Un filtre de Sobel est appliqué à chaque image F_n d'une séquence, l'écart-type std_{space} de la carte de gradient ainsi générée est calculée sur toute l'image. Finalement la valeur maximum sur toutes les images de la séquence est prise comme mesure de l'activité spatiale de la séquence.

$$SI = max_{time}\{std_{space}[Sobel(F_n)]\}$$

Équation 3-6. Mesure d'information spatiale (SI)

Dans notre cas nous travaillons uniquement sur une image de chaque séquence. Comme le montre la Figure 3.7, les huit images ont des caractéristiques d'activité spatiale variées ce qui nous permet d'obtenir des résultats moyennés représentatifs de l'effet des différents filtres sur des images naturelles.

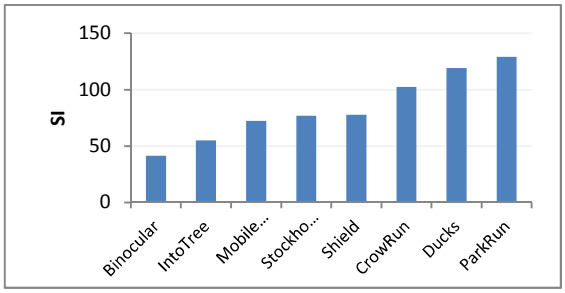


Figure 3.7. Information Spatiale SI de la banque d'image test

3. 3. 1. 2. **Mesure de résultats**

Pour la mesure de nos résultats, nous avons pris le parti de choisir quatre métriques regroupées en deux catégories : les métriques mesurant les performances au sens du problème inverse et les métriques mesurant l'effet de flou. Ainsi les premières métriques indiquent la qualité de réduction de bruit tandis que les deuxièmes indiquent la qualité de préservation des détails des différents filtres.

Dans la première catégorie, nous avons sélectionné les deux métriques les plus répandues que nous avons déjà présentées aux chapitres précédents : Le PSNR et le SSIM. En calculant ces métriques entre la version originale de l'image (sans ajout de bruit) et les versions filtrées, on peut sans ambigüités déterminer le filtre qui permet d'approcher le plus possible l'image originale en réduisant le bruit.

Afin de mesurer également l'artefact typique des filtres passe-bas, nous avons sélectionné deux métriques de flou de la littérature : Le LPC-SI [108] et la mesure de flou de Marziliano [109].

Le LPC-SI est basé sur une mesure de cohérence de phase dans l'image. Pour cela, une décomposition en ondelettes est appliquée à l'image à l'aide de filtres de Gabor. Une phase cohérente est déterminante pour décrire un

contour net (toutes les fréquences sont alignées). Afin de détecter cette cohérence de phase, les différentes échelles de la décomposition en ondelettes sont habilement multipliées entre elles : les endroits avec des contours francs vont donc donner des valeurs très élevées. Le LPC-SI est calculé à partir de ces valeurs, ainsi plus la note est élevée, meilleure est la qualité de l'image. Cette métrique mesure le flou dans une image sans référence à une image originale, c'est la seule métrique sans référence que nous utilisons.

La métrique de Marziliano mesure l'étalement moyen des contours d'une image, plus la note est élevée, plus l'image contient de flou. Cette métrique compare l'image originale et l'image à qualifier, on parle de métrique « full-référence ». Les contours verticaux dans l'image originale sont d'abord détectés à l'aide d'un filtre de Sobel 1D, puis l'étalement horizontal des contours est mesuré par la distance entre les deux extrema de luminance autour d'un pixel de contour. La Figure 3.8. présente un exemple donné dans l'article [110] de mesure de l'étalement d'un contour sur une ligne de luminance. Les pixels P1 et P3 sont deux pixels de contours verticaux détectés par l'analyse de Sobel. Lorsque la pente change de signe, l'extrema local est trouvé et considéré comme la fin du contour. Ainsi l'étalement du contour P1 est la distance entre les deux extrema locaux P2 et P2'. On peut noter que la métrique peut être utilisée sans référence en appliquant la détection de Sobel directement sur l'image à qualifier.

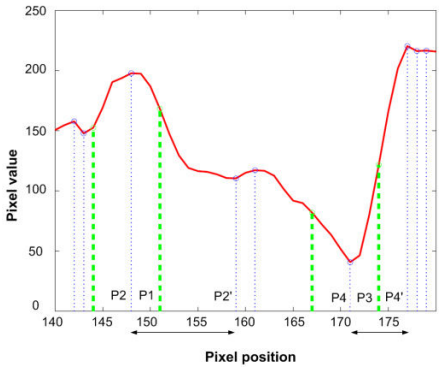


Figure 3.8. Métrique de mesure de flou Marziliano
Illustration issue de la publication [110]
Exemple de mesure de flou autour d'un pixel de contour sur une ligne de luminance

3. 3. 1. 3. Discussion sur l'interprétation des quatre métriques

Dans le cas de la mesure de performances de filtres passe-bas appliqués au cas de la réduction de bruit, le PSNR et le SSIM passent par un maximum correspondant au meilleur compromis entre réduction de bruit et destruction de la structure de l'image qui dépend de la force de filtrage. Pour illustrer ce principe, la Figure 3.9 présente les notes de PSNR (a) et de SSIM (b) pour trois images avec un écart-type de bruit de 20, auxquelles on applique un filtre gaussien de taille 11x11 dont on fait augmenter progressivement l'écart-type. Dans un premier temps, l'augmentation de la force de filtrage améliore les notes car le bruit est réduit. Dans un deuxième temps, les notes chutent car le filtre dégrade la structure de l'image en apportant trop de flou. Le point optimal dépend des caractéristiques de l'image, une image contenant une forte activité spatiale comme l'image *CrowdRun* verra rapidement ses notes de PSNR et SSIM chuter avec l'augmentation de la force de filtrage.

Pour les deux métriques de flou, la meilleure qualité est toujours pour la version bruitée sans filtrage (écart-type égal à 0 sur les courbes (c) et (d)). La présence de flou augmente logiquement avec la force de filtrage. La mesure LPC-SI varie peu en fonction du contenu, cependant pour Marziliano, l'image contenant les contours les plus francs (*Binocular*) est la plus sensible à l'introduction de flou. Ces tests sont réalisés avec un filtre simple, nous montrerons par la suite que nous parvenons à améliorer les notes de LPC-SI par nos filtres adaptatifs comparativement à la version bruitée non filtrée. Cependant la métrique Marziliano attribue toujours la meilleure note à la version bruitée non filtrée.

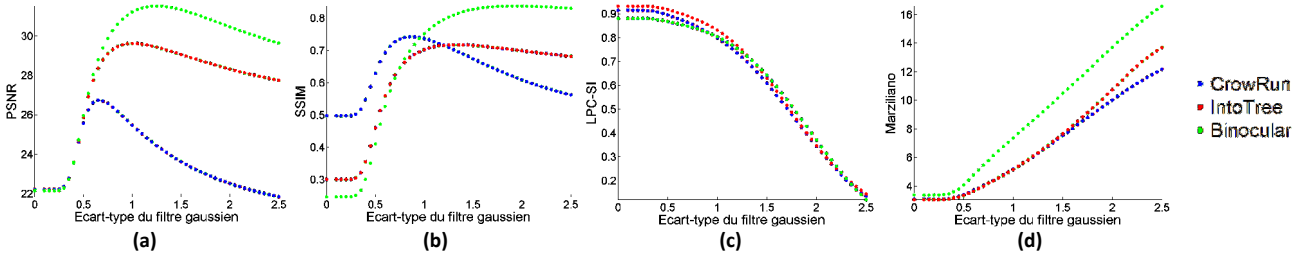


Figure 3.9. Evolution des métriques en fonction de l'écart-type d'un filtre gaussien 11x11 pour trois images
(a) PSNR - (b) SSIM - (c) LPC-SI – (d) Marziliano

3. 3. 2. Augmentation de la taille du support de filtrage AWA

En appliquant le filtre AWA 11x11 sur la composante de luminance d'images HD, nous nous apercevons que le filtre amène un fort effet de flou comme le montre les images présentées dans la Figure 3.10, la mesure de LPC-SI confirme l'impression visuelle que le filtre AWA 11x11 apporte plus de flou que tous les autres. En comparaison, le filtre Bilatéral conserve bien mieux les détails tout en enlevant plus de bruit que la version AWA avec un support 3x3. Par conséquent, nous souhaitons étudier les raisons qui permettent au filtre Bilatéral de retenir mieux les détails que le filtre AWA à même taille de support.

Pour cela nous commençons par comparer le filtre AWA et le filtre de Similarité (composante photométrique du filtre Bilatéral) qui considèrent tous deux la distance photométrique entre les pixels du support de filtrage. Dans un deuxième temps, nous étudierons l'intérêt de l'ajout du noyau Géométrique du filtre Bilatéral.

Figure 3.10. Comparaison des performances de réduction de bruit des filtres AWA 3x3, AWA 11x11 et Bilatéral 11x11 (Ligne1) CrowdRun 1280x720 ; (Ligne2) IntoTree 1280x720 ; (Ligne 3) Binocular 1280x720 (Colonne 1) Partie de l'image originale ; (Colonne 2) Image bruitée, bruit blanc gaussien de variance $\sigma_n^2 = 20$; (Colonne 3) Filtre AWA 3x3 $\varepsilon^2 = 2\sigma_n^2$; (Colonne 4) Filtre AWA 11x11 $\varepsilon^2 = 2\sigma_n^2$; (Colonne 5) Filtre Bilatéral 11x11 $\sigma_g = 1.8$ et $\sigma s^2 = 2\sigma_n$

3. 3. 3. Comparaison des filtres AWA et Similarité

Commençons par étudier la forme des fonctions de filtrage AWA et Similarité. Comme le montre la Figure 3.11 (a), à même paramètre ($\varepsilon = \sigma_s$), la décroissance du noyau gaussien du filtre de Similarité est plus rapide que celle du filtre AWA. Ainsi, les pixels trop différents du pixel courant sont plus rapidement exclus du support de filtrage avec le filtre de Similarité, ce qui lui permet d'avoir une meilleure préservation des contours, comme le montre les images de la Figure 3.12. Pour rappel, les poids du filtre AWA sont exprimés par :

$$w(i,j) = Z(x,y) \times Y_{AWA}(i,j)$$

$$Y_{AWA}(i,j) = \frac{1}{1+\mathrm{a} \times max(\varepsilon^2, \Delta p(i,j)^2)}$$

$$\Delta p(i,j) = |p(x,y) - p(i,j)|$$

Le paramètre a prévu pour contrôler la décroissance des poids, devient inefficace lorsque les différences au sein du support de filtrage augmentent. En effet, quand $(\mathrm{a} \times \Delta p(i,j)^2) \gg 1$, les poids du filtre AWA deviennent :

$$Y_{AWA}(i,j) = \frac{1}{a \times \Delta p(i,j)^2}$$

La somme des poids du support est alors égale à :

$$\boldsymbol{Z(x,y)} = \left(\sum_{(i,j) \in S_{x,y}} \boldsymbol{Y_{AWA}(i,j)} \right)^{-1} = \left(\frac{1}{\mathrm{a}} \sum_{(i,j) \in S_{x,y}} \frac{1}{\Delta p(i,j)^2} \right)^{-1}$$

Finalement, les poids normalisés du support de filtrage ne dépendent plus du paramètre a et sont décrits par :

$$w(i,j) = Z(x,y) \times Y_{AWA}(i,j) = \frac{a \times \left(\sum_{(i,j) \in S_{x,y}} \frac{1}{\Delta p(i,j)^2} \right)^{-1}}{a \times \Delta p(i,j)^2} = \frac{\left(\sum_{(i,j) \in S_{x,y}} \frac{1}{\Delta p(i,j)^2} \right)^{-1}}{\Delta p(i,j)^2}$$

Par conséquent le paramètre *a* n'a plus d'effet sur la décroissance des poids et le poids non négligeable donné aux pixels trop différents du pixel courant dans le support de filtrage ne peut être contrôlé, ce qui explique l'introduction de l'effet de flou gênant.

La Figure 3.11 (b) rappelle le fonctionnement du filtre AWA vu au chapitre précédent. Le seuil ε dépend de la variance de bruit, ainsi, les différences au sein d'un support de filtrage inférieures au seuil ε sont considérées comme dues au bruit et sont simplement moyennées pour être réduites. Le filtre de Similarité (c) ne possède pas de seuil, la décroissance gaussienne est réglée en fonction de la variance de bruit présent dans l'image.

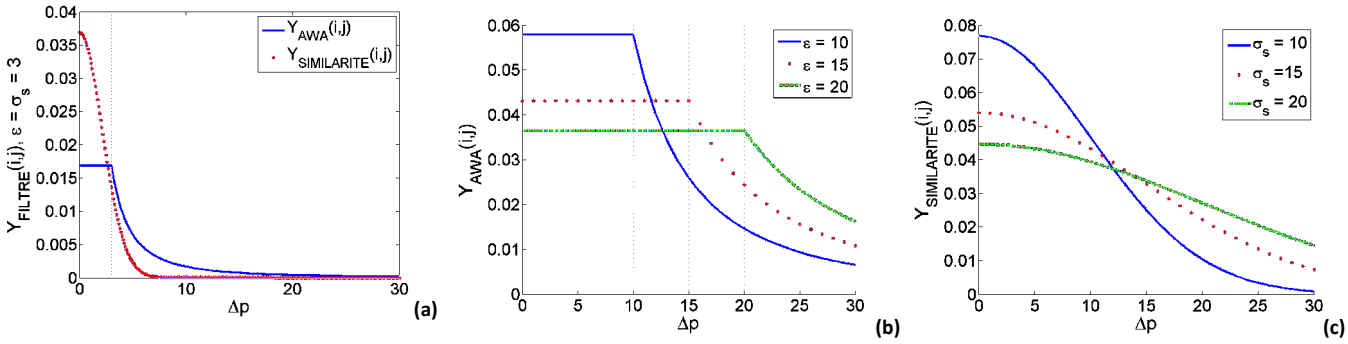


Figure 3.11. Comparaison des fonctions de filtrage AWA et Similarité
(a) Evolution des poids AWA et Similarité en fonction de la différence de luminance Δp, (b) Evolution des poids AWA en fonction du paramètre ε, (c) Evolution des poids Similarité en fonction de la variance σ_s^2

Le Tableau 3.1 présente les résultats pour les filtres AWA et Similarité en termes de métrique de distorsions (PSNR et SSIM) et de flou (LPC-SI et Marziliano).

- D'après la mesure LPC-SI qui mesure la présence de flou dans une image sans référence à une image originale. Le filtre AWA 11x11 apporte toujours le plus de flou, ce qui se confirme par l'analyse visuelle (Figure 3.12).
- A faible niveau de bruit (σ_n = 10), le filtre AWA 3x3 est celui qui donne le meilleur compromis entre réduction de bruit et perte de détails.
- Lorsque le niveau de bruit augmente, le filtre AWA (3x3 ou 11x11) réduit toujours le bruit de façon plus efficace que le filtre de Similarité (Cf. PSNR et SSIM), qui en contrepartie introduit moins de flou (Cf. LPC-SI et Marziliano).

		AWA 3x3 $\varepsilon^2 = 2*\sigma_n^2$	**AWA 11x11** $\varepsilon^2 = 2*\sigma_n^2$	**Similarité 11x11** $\sigma_s^2 = 2*\sigma_n^2$
σ_n =10	SSIM	**0,854**	0,836	0,838
	PSNR	**31,711**	30,350	31,078
	LPC-SI	0,925	0,918	**0,926**
	Marziliano	3,835	3,298	**3,221**
σ_n =20	SSIM	0,683	**0,684**	0,659
	PSNR	**27,211**	26,635	26,643
	LPC-SI	0,916	0,902	**0,921**
	Marziliano	3,880	3,211	**3,138**
σ_n =30	SSIM	0,546	**0,558**	0,523
	PSNR	24,425	**24,542**	24,182
	LPC-SI	0,907	0,884	**0,909**
	Marziliano	3,865	3,145	**3,089**

Tableau 3.1. Résultats moyens obtenus par les filtres AWA et Similarité en termes de PSNR, SSIM, LPC-SI et Marziliano pour trois niveaux de bruits

La Figure 3.12 propose une comparaison visuelle de l'effet des filtres AWA 11x11 et Similarité 11x11. Les quatre métriques de distance et de flou sont indiquées. On note que le filtre de Similarité permet de conserver plus de détails mais en même temps enlève moins de bruit dans les images. Le PSNR et le SSIM indiquent en moyenne le filtre AWA 11x11 comme le meilleur filtre au sens du problème inverse, alors que les métriques de flou préfèrent le filtre de Similarité. On conclut qu'il y a un compromis à trouver entre ces deux filtres afin de conserver la bonne réduction de bruit du filtre AWA et la bonne préservation des détails du filtre de Similarité.

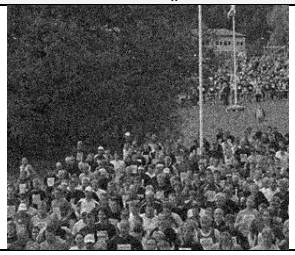
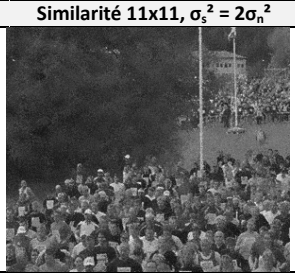
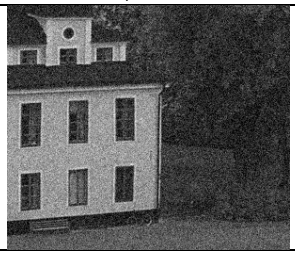
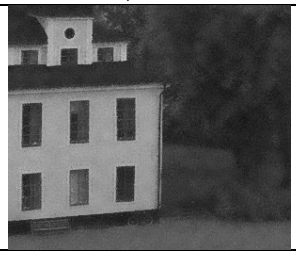
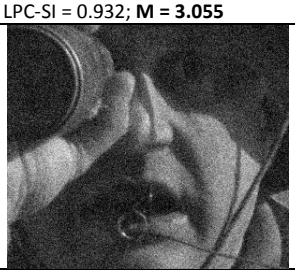
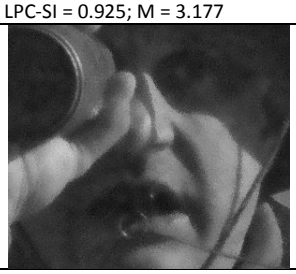
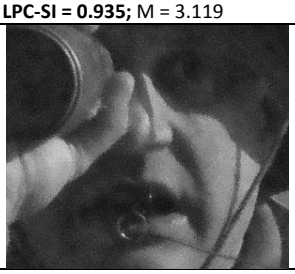
Original	**Bruitée σ_n = 20**	**AWA 11x11, $\varepsilon^2 = 2\sigma_n^2$**	**Similarité 11x11, $\sigma_s^2 = 2\sigma_n^2$**
			
	PSNR = 22.22 dB; SSIM = 0.498 LPC-SI = 0.915; **M = 3.087**	PSNR = 25.69 dB; **SSIM = 0.702** LPC-SI = 0.900; M = 3.209	**PSNR = 26.05** dB; SSIM = 0.688 **LPC-SI = 0.920**; M = 3.138
			
	PSNR = 22.15 dB; SSIM = 0.301 LPC-SI = 0.932; **M = 3.055**	**PSNR = 29.34 dB; SSIM = 0.649** LPC-SI = 0.925; M = 3.177	PSNR = 28.35 dB; SSIM = 0.594 **LPC-SI = 0.935;** M = 3.119
			
	PSNR = 22.16 dB; SSIM = 0.247 LPC-SI = 0.881; **M = 3.389**	**PSNR = 29.94 dB; SSIM = 0.685** LPC-SI = 0.880; M = 3.733	PSNR = 28.94 dB; SSIM = 0.600 **LPC-SI = 0.901;** M = 3.555

Figure 3.12. Comparaison des performances de réduction de bruit des filtres AWA 3x3, AWA 11x11 et Bilatéral 11x11
(Ligne1) CrowdRun 1280x720 ; (Ligne2) IntoTree 1280x720 ; (Ligne 3) Binocular 1280x720
(Colonne 1) Partie de l'image originale ; (Colonne 2) Image bruitée, bruit blanc gaussien de variance σ_n^2 = 20; (Colonne 3) Filtre AWA 11x11 $\varepsilon^2 = 2\sigma_n^2$; (Colonne 4) Filtre de Similarité 11x11 $\sigma s^2 = 2\sigma_n^2$

3. 3. 4. Intérêt du noyau Géométrique du filtre Bilatéral

Dans l'idée d'une part, de réduire le flou apporté par l'augmentation du support de filtrage du filtre AWA tout en conservant ses capacités de réduction de bruit, et d'autre part, d'analyser l'apport d'une composante géométrique, nous proposons un nouveau filtre nommé le BilAWA. Comme le filtre Bilatéral, le filtre BilAWA utilise deux noyaux de filtrage, un premier considérant la distance spatiale et un second considérant la différence de luminance. La composante géométrique $h_g(i,j)$ est identique à celle du filtre Bilatéral (Équation 3-4) et la composante en luminance $h_s(i,j)$ est remplacée par le filtre AWA.

$$h_s(i,j) = Y_{AWA}(i,j)$$

Équation 3.7. *Composante de comparaison en luminance du filtre BilAWA*

Pour rappel, les poids du filtre AWA sont définis par :

$$Y_{AWA}(i,j) = \frac{1}{1 + a \times max(\varepsilon^2, \Delta p(i,j)^2)}$$

$$\Delta p(i,j) = \|p(x,y) - p(i,j)\|$$

Équation 3-8. *Poids AWA*

Dans la suite de ce chapitre, les filtres considérant deux composantes sont nommés filtres bilatéraux, tandis que les filtres considérant une seule composante sont nommés filtres unilatéraux. Le Tableau 3.2 récapitule les différents filtres étudiés.

Filtres bilatéraux	**Filtres unilatéraux**	
	Composante photométrique	Composante géométrique
Filtre BilAWA	Filtre AWA	Filtre Géométrique
Filtre Bilatéral	Filtre de Similarité	Filtre Géométrique

Tableau 3.2. Filtres bilatéraux et unilatéraux

Nous proposons maintenant d'analyser l'intérêt de la composante géométrique en comparant d'une part le filtre de Similarité et le filtre Bilatéral et d'autre part le filtre AWA et le filtre BilAWA.

La Figure 3.13 illustre l'effet des deux composantes des filtres BilAWA et Bilatéral sur un support de filtrage particulier de taille 11x11 représenté par la figure (a). Les deux filtres utilisent la même composante géométrique dont les poids attribués au support de filtrage sont représentés par la figure (b). La composante considérant la distance photométrique du filtre Bilatéral (filtre de Similarité) est représentée par la figure (c) et le produit des deux composantes donnant les poids du filtre Bilatéral est présenté par la figure (d). Les figures (e) et (f) représentent respectivement la composante photométrique du filtre BilAWA (filtre AWA) le filtre BilAWA.

On observe dans les deux cas que la composante géométrique permet de concentrer les poids au centre du support de filtrage et ainsi de limiter le flou apporté dans l'image filtrée. Le pixel courant (au centre du support de filtrage (a)), appartient à une zone homogène contaminée par du bruit. Le support contient également un contour qui doit être préservé. Les poids attribués par le filtre de Similarité (c) et le filtre AWA (e) ont un comportement similaire, c'est-à-dire que seuls les pixels ressemblants au pixel courant sont pris dans le support de filtrage, les pixels appartenant à l'objet sombre ont un poids négligeable pour conserver la netteté du contour. Cependant le filtre AWA attribue des poids plus homogènes aux pixels voisins ressemblant au pixel courant ce qui permet de réduire plus fortement le bruit.

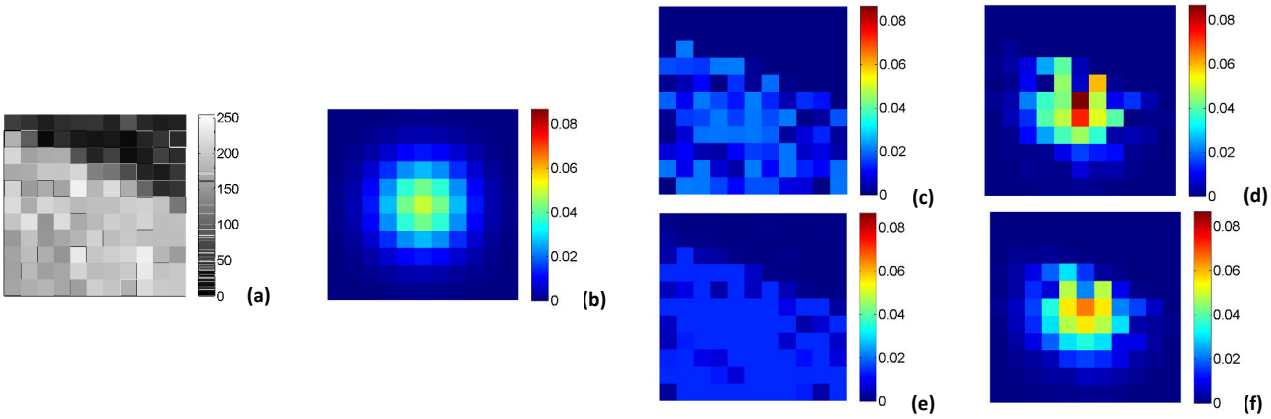


Figure 3.13. Action de la composante géométrique sur le filtre Bilatéral et le filtre BilAWA
(a) Support de filtrage bruit additif σn =10 – (b) Poids de la composante géométrique σ_g = 1.8 – (c) Poids du filtre de Similarité σs² = 2*σn² – (d) Poids du filtre Bilatéral – (e) Poids du filtre AWA ε² = 2*σn² – (f) Poids du filtre BilAWA

Le Tableau 3.3 présente les résultats moyens sur les huit images pour les filtres AWA 11x11 et Similarité 11X11 et leur version Bilatérale respectivement le filtre BilAWA et le filtre Bilatéral. L'ajout du noyau gaussien considérant la distance spatiale permet dans tous les cas de réduire le bruit (augmentation systématique du PSNR et du SSIM). Les deux métriques de flou ne permettent pas de conclure sur l'impact du noyau gaussien en termes de préservation des détails. En effet, la métrique de Marziliano indique une augmentation de flou systématique alors que la métrique LPC-SI indique une réduction.

On remarque que le filtre BilAWA donne la meilleure réduction de bruit dans pratiquement tous les cas (PSNR et SSIM). Les métriques de flou indiquent que le filtre de Similarité (Marziliano) et le filtre Bilatéral (LPC-SI) préservent le mieux la structure de l'image.

		Filtres unilatéraux		Filtres bilatéraux	
		AWA 11x11	**Similarité 11x11**	**BilAWA 11x11**	**Bilateral 11x11**
		ε² = 2*σn²	σs² = 2*σn²	σg² = 1.8² ε² = 2*σn²	σg² = 1.8² σs² = 2*σn²
σn = 10	SSIM	0,836	0,838	**0,860**	0,845
	PSNR	30,35	31,08	31,57	31,38
	LPC-SI	0,918	**0,926**	0,923	**0,926**
	M.	3,298	**3,221**	3,503	3,336
σn = 20	SSIM	0,684	0,659	**0,712**	0,668
	PSNR	26,64	26,64	**27,46**	26,82
	LPC-SI	0,902	0,921	0,915	**0,922**
	M.	3,211	**3,138**	3,479	3,294
σn = 30	SSIM	0,558	0,523	**0,585**	0,536
	PSNR	24,54	24,18	**25,02**	24,22
	LPC-SI	0,884	0,909	0,903	**0,914**
	M.	3,145	**3,089**	3,446	3,265

Tableau 3.3. Résultats moyens de PSNR, SSIM, LPC-SI et Marziliano pour trois niveaux de bruit

Il existe donc un compromis à trouver entre la réduction de bruit donnée par le filtre BilAWA et la préservation des détails et contours du filtre Bilatéral. Dans ce but nous proposons un deuxième filtre dans le paragraphe suivant, le filtre Bilatéral seuillé.

3. 3. 5. Proposition du filtre Bilatéral seuillé

Comme nous l'avons déjà abordé, la bonne capacité de réduction de bruit des filtres AWA et BilAWA vient de l'utilisation d'un seuil en-dessous duquel toute différence est réduite, et l'effet de flou est dû à la faible décroissance des poids qui donne une part non-négligeable dans le filtrage à des pixels trop différents du pixel courant. Pour apporter une solution, nous proposons le filtre Bilatéral seuillé (nommé TBilateral (Thresholded Bilateral) dans les tableaux et figures suivantes). Le filtre Tbilateral utilise la même composante géométrique $h_g(i,j)$ que le filtre Bilatéral et le filtre BilAWA (Équation 3-4), associée à une nouvelle composante photométrique $h_s(i,j)$ décrite par l'Équation 3-9. Le Tbilateral utilise ainsi le seuil de différence photométrique AWA T au-dessus duquel une décroissance gaussienne est appliquée aux pixels trop différents du pixel courant. La fonction de filtrage ainsi définie possède deux paramètres, le seuil T et la variance σ_s^2 réglant la vitesse de décroissance des poids gaussiens. Ces deux paramètres sont fixés égaux à deux fois la variance de bruit.

$$h_s(i,j) = min\left(exp\left(-\frac{T^2}{2{\sigma_s}^2}\right), exp\left(-\frac{\|\Delta p(i,j)\|^2}{2\sigma_s^2}\right)\right)$$

$$\Delta p(i,j) = \|p(x,y) - p(i,j)\|$$

Équation 3-9. *Composante de comparaison en luminance du filtre Tbilatéral*

Le Tableau 3.4 présente les résultats de tous les filtres de l'étude pour les quatre métriques et trois niveaux de bruit, moyennés sur les huit images. Les métriques objectives PSNR et SSIM indiquent que le filtre Tbilatéral réduit le plus fortement le bruit (mis à part le PSNR pour une faible variance de bruit qui le place deuxième). De plus, la mesure de flou LPC-SI indique qu'il est le filtre qui préserve le mieux la structure de l'image. La mesure Marziliano quant à elle indique toujours le filtre de Similarité comme celui qui introduit le moins de flou, cependant d'après cette métrique, le filtre Tbilateral représente bien un compromis entre le filtre BilAWA et Bilatéral.

		Filtres unilatéraux			**Filtres bilatéraux**		
		AWA 3x3	**AWA 11x11**	**Similarité 11x11**	**BilAWA 11x11**	**Bilateral 11x11**	**Tbilateral 11x11**
		$\varepsilon^2 = 2*\sigma n^2$	$\varepsilon^2 = 2*\sigma n^2$	$\sigma s^2 = 2*\sigma n^2$	$\varepsilon^2 = 2*\sigma n^2$	$\sigma g^2 = 1.8^2$; $\sigma s^2 = 2*\sigma n^2$	$\varepsilon^2 = 2*\sigma n^2$; $\sigma s^2 = 2*\sigma n^2$
σ_n = 10	SSIM	0,854	0,836	0,838	0,860	0,845	**0,862**
	PSNR	**31,711**	30,350	31,078	31,568	31,381	31,703
	LPC-SI	0,925	0,918	0,926	0,923	0,926	**0,927**
	M.	3,835	3,298	**3,221**	3,503	3,336	3,412
σ_n = 20	SSIM	0,683	0,684	0,659	0,712	0,668	**0,714**
	PSNR	27,211	26,635	26,643	27,458	26,824	**27,472**
	LPC-SI	0,916	0,902	0,921	0,915	0,922	**0,923**
	M.	3,880	3,211	**3,138**	3,479	3,294	3,424
σ_n = 30	SSIM	0,546	0,558	0,523	0,585	0,536	**0,598**
	PSNR	24,425	24,542	24,182	25,015	24,221	**25,098**
	LPC-SI	0,907	0,884	0,909	0,903	0,914	**0,914**
	M.	3,865	3,145	**3,089**	3,446	3,265	3,442

Tableau 3.4. Résultats moyens de PSNR, SSIM, LPC-SI et Marziliano pour trois niveaux de bruit

La Figure 3.14 présente des images filtrées avec les filtres Bilatéral, BilAWA et Tbilateral pour deux niveaux de bruit. Les observations précédentes sont confirmées, le filtre BilAWA réduit plus le bruit que le filtre Bilatéral mais introduit plus de flou, tandis que le filtre Tbilateral permet à la fois de réduire efficacement le bruit et de préserver la structure de l'image (spécialement visible dans l'herbe au premier plan).

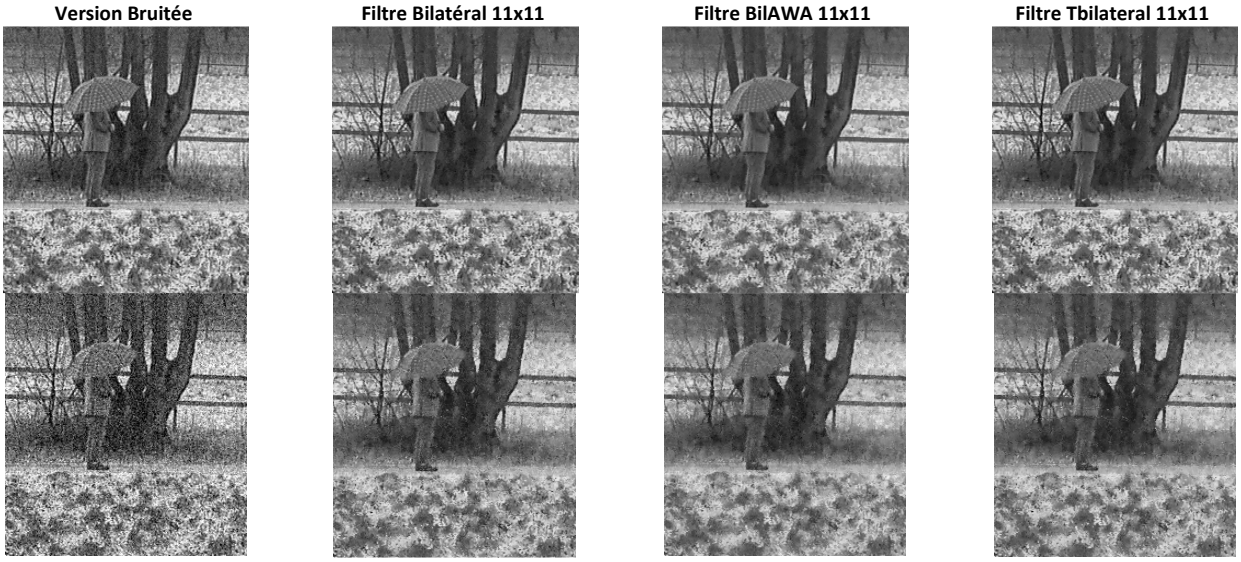


Figure 3.14. Comparaison visuelle des filtres Bilateral, BilAWA et Tbilateral sur une partie de l'image ParkRun (1ère ligne) bruit σ_n = 10 – (2ème ligne) bruit σ_n = 30

Jusqu'ici nous avons comparé les filtres en paramétrant tous les seuils et variances à deux fois la variance de bruit dans l'image, conformément à la recommandation des auteurs du filtre AWA. Cependant, ce choix étant empirique, on peut se demander si il est optimal pour tous les filtres. C'est pourquoi on propose d'étudier les différents filtres en fixant $\varepsilon^2 = \sigma_s^2 = \alpha * \sigma_n^2$ en faisant varier α de 1 à 6. La Figure 3.15 présente les résultats de cette étude pour trois images particulières et deux niveaux de bruit, pour le PSNR et le LPC-SI. Seules deux des quatre métriques sont représentées pour simplifier l'analyse. Pour la mesure de réduction de bruit, le PSNR et le SSIM donnent des résultats similaires, nous représentons le PSNR car c'est la métrique la plus largement utilisée. Pour la mesure de flou nous représentons la métrique LPC-SI qui est la seule métrique sans référence que nous utilisons dans cette étude.

On note que d'après les métriques de flou, le filtre BilAWA fait partie des filtres introduisant le plus de flou quel que soit l'image et la variance de bruit. On note également que le deuxième filtre proposé, le filtre Tbilateral est le seul à toujours faire partie des deux meilleurs filtres pour les quatre métriques. Il représente donc le meilleur compromis entre réduction de bruit et préservation de contenu dans toutes les conditions testées.

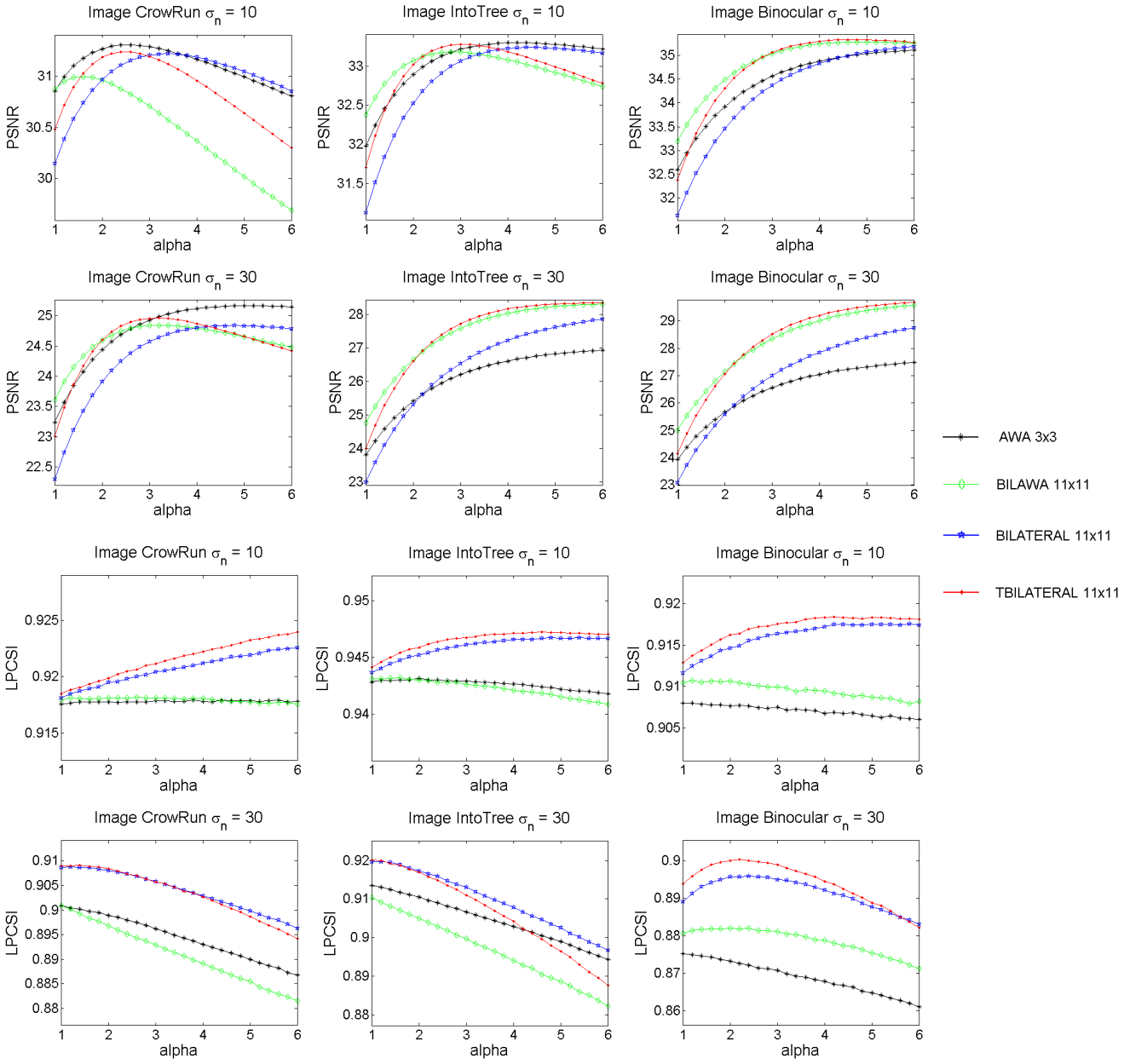


Figure 3.15. Evolution des métriques en fonction de la force de filtrage
Paramètres des différents filtres : $\varepsilon^2 = \sigma s^2 = \alpha * \sigma n^2$

Dans cette première étude, nous avons montré l'intérêt d'augmenter la taille du support de filtrage pour réduire le bruit d'images HD, ainsi que l'intérêt d'utiliser des filtres bilatéraux qui considèrent à la fois la distance photométrique et spatiale des pixels d'une image. Nous avons proposé deux nouveaux filtres, le filtre BiLAWA qui surpasse le filtre Bilatéral en réduction de bruit mais qui conserve moins efficacement la structure de l'image. Ainsi que le filtre Tbilateral qui donne le meilleur compromis à la fois en termes de réduction de bruit et de préservation d'image. Dans la suite de ce chapitre, nous étudions l'intérêt des nouveaux filtres proposés pour une application de prétraitement pour l'encodage H.264/AVC.

3. 4. Etude des filtres comme prétraitement pour l'encodeur H.264/AVC

Au chapitre précédent nous avons proposé de contrôler le filtre AWA à l'aide du modèle de JND décrit par X. Yang. Nous avons montré l'intérêt d'appliquer ce filtre avant un encodage H.264/AVC pour réduire le contenu pas/peu perceptible des séquences vidéo afin de réduire le débit nécessaire à la représentation de la séquence encodée. Nous comparons maintenant les performances des deux filtres proposés au paragraphe précédent à celles du filtre AWA 3x3 que nous prenons comme référence. Il faut noter qu'au chapitre précédent, nous avions utilisé une version simplifiée du filtre AWA utilisant uniquement des poids binaires dans le but de réduire la complexité du prétraitement. Ici nous utilisons la version originale du filtre ce qui explique par la suite les différences de réduction de débit avec le chapitre précédent.

3. 4. 1. Les Préfiltres perceptuels de l'étude

Le modèle JND est introduit dans le noyau photométrique des deux filtres bilatéraux, le noyau gaussien considérant la distance spatiale reste inchangé et son écart-type est toujours égal à 1.8. Les trois filtres étudiés ici sont dits perceptuels et sont définis de la manière suivante :

$$Y_{AWA,JND}(i,j) = \frac{1}{1 + a \times max(JND^2, \Delta p(i,j)^2)}$$

$$\Delta p(i,j) = \|p(x,y) - p(i,j)\|$$

Équation 3-10. *Poids AWA perceptuel*

Les filtres BilAWA et Tbilateral sont tous deux définis comme des filtres bilatéraux de la manière suivante :

$$Y_{Bilateral}(i,j) = h_g(i,j) \times h_s(i,j)$$

Équation 3-11. ***Poids Filtres bilatéraux***

Avec : $$h_g(i,j) = exp\left(-\frac{\|x-i\|^2 \times \|y-j\|^2}{2\sigma_g^2}\right)$$

Équation 3-12. ***Filtre Géométrique***

Ils diffèrent par leur noyau photométrique défini par :

$$h_{s,BilAWA,JND}(i,j) = Y_{AWA,JND}(i,j)$$

Équation 3-13. ***Composante de comparaison en luminance du filtre BilAWA***

$$h_{s,TBilateral}(i,j) = min\left(exp\left(-\frac{1}{2}\right), exp\left(-\frac{\|\Delta p(i,j)\|^2}{2 * JND^2}\right)\right)$$

Équation 3-14. ***Composante de comparaison en luminance du filtre Tbilateral***

$$\Delta p(i,j) = \|p(x,y) - p(i,j)\|$$

La Figure 3.16 présente des parties d'image appartenant à trois séquences 1280x720 50p prétraitées avant encodage. La première colonne présente les cartes de JND spatial, pour rappel plus le JND est élevé et plus le contenu est jugé peu sensible aux dégradations et par conséquent filtré plus fortement. Les seuils des filtres AWA, BilAWA et Tbilateral étant fixés à la valeur du JND local, les différences au sein du support de filtrage inférieures au JND sont considérées non perceptibles et reçoivent le même poids.

Les quatre métriques utilisées précédemment sont également reportées. Les métriques PSNR et SSIM indiquent ici la distance à l'image originale, comme nous l'avons vu au chapitre précédent ces métriques sont inefficaces pour mesurer la qualité subjective. Elles indiquent presque toujours le filtre AWA 3x3 comme celui de meilleure qualité car il filtre moins fortement les images. Or, mis à part pour l'arbre de la séquence *CrowdRun* (première ligne) où l'on perçoit un effet de flou, pour les autres images les filtres BilAWA et Tbilateral n'introduisent pas plus de flou que le filtre AWA 3x3. Les deux indicateurs de flou donnent des résultats différents, cependant les filtres AWA 3x3 et Tbilateral semblent introduire moins de flou que le filtre BilAWA. Comme au chapitre précédent, nous choisissons de nous baser sur des tests subjectifs pour juger la qualité des séquences encodées avec prétraitement.

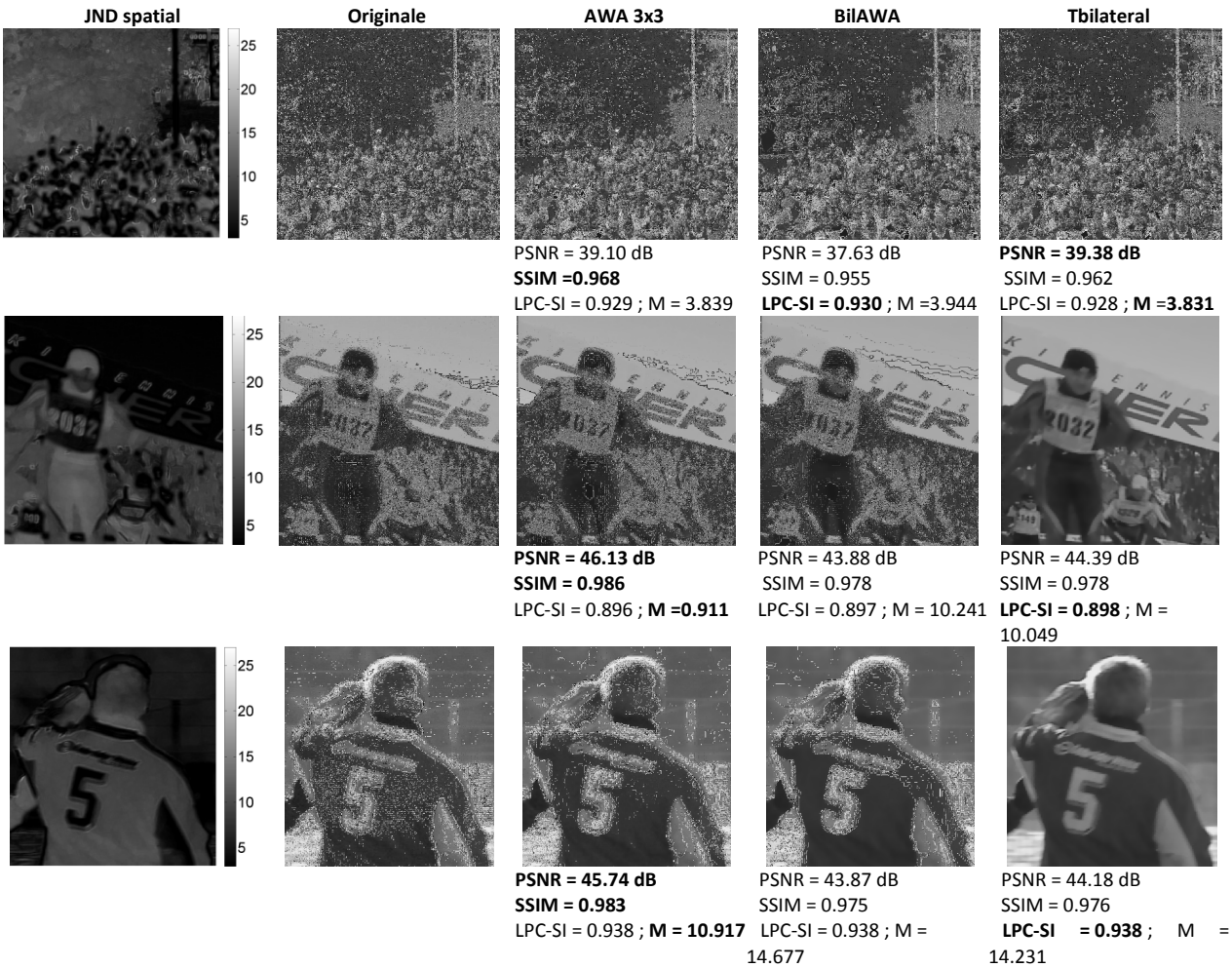


Figure 3.16. Comparaison visuelle des séquences prétraitées avant encodage

3. 4. 2. Protocole de mesure

Les différents préfiltres sont jugés sur la réduction de débit qu'ils amènent par rapport à un encodage classique, ainsi que sur la qualité visuelle des séquences. Pour mesurer l'impact visuel de ces filtres sur l'encodage de séquence HD, nous avons réalisé une session de tests subjectifs avec 16 observateurs, en utilisant la méthode PC (Paired Comparison) décrite au chapitre précédent. Nous souhaitons tester trois filtres, plusieurs séquences ainsi que plusieurs conditions d'encodage. La durée maximale d'un test subjectif étant limitée à 30 minutes, nous avons dû faire un choix parmi toutes les possibilités de test.

En ce qui concerne les conditions d'encodage, nous avons utilisé le codec x264 dont les caractéristiques sont données dans le Tableau 2.2. Nous avons choisi d'encoder les séquences en GOP Inter IBBP(12) à QP 22 et 27 pour s'approcher des conditions d'encodage des clients de Digigram, à savoir une haute qualité pour un débit raisonnable. De plus, nous avons testé le cas de l'encodage Intra à QP22 afin de comparer les résultats à l'encodage Inter à même QP. Par expérience de la première série de tests subjectifs présentés au chapitre précédent, nous n'avons pas souhaité tester le codage Intra à plus fort QP car à faible débit, l'effet de scintillement typique du codage Intra gène les observateurs plus que l'éventuelle perte de détails amenée par les préfiltres.

	Test HD
Format source	1280x720, 50p
Echantillonnage couleur	4 :2 :0
Profil	High
Deblocking	Désactivé
Codage Entropique	CABAC
GOP	Intra et IBBP12
QP	22, 27

Tableau 3.5. Caractéristiques d'encodage x264

Nous avons choisi trois séquences HD 720p parmi six possibilités. La Figure 3.17 présente les mesures d'activité spatiale et temporelle des six séquences que nous avons considérées. La mesure d'activité spatiale que nous avons déjà présentée est exprimée par l'Équation 3-6. La mesure d'activité temporelle est donnée par l'Équation 3-15, l'écart-type de l'image de différence entre deux images consécutives M_n est calculé pour toutes les images de la séquence et la valeur maximum est retenue comme mesure d'information temporelle.

$$TI = max_{time}\{std_{space}[M_n(i,j)]\}$$
$$\text{Avec :} \quad M_n(i,j) = F_n(i,j) - F_{n-1}(i,j)$$

Équation 3-15. Mesure d'information temporelle (ST)

La Figure 3.18 présente les réductions de débit moyennes sur les trois filtres et les trois conditions de test. En analysant les deux figures, on note que la réduction de débit la plus forte est obtenue pour la séquence ayant la plus faible activité spatiale et temporelle (*IntoTree*, 57%). Cette réduction de débit importante s'explique par le fait que le JND impose une force de filtre importante engendrant un effet de flou notable. Sur la séquence *Duck*, ayant une information spatiale plus importante que les autres séquences, les filtres apportent une réduction de débit légèrement inférieure (10%). Pour notre test nous avons sélectionné trois séquences présentant une forte activité temporelle et différents niveaux d'activité spatiale : *CrowdRun*, *Ski* et *Soccer*.

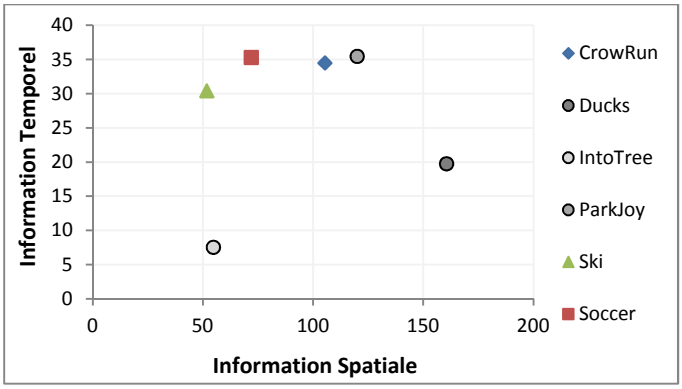


Figure 3.17. Activité spatiale et temporelle des séquences test possibles

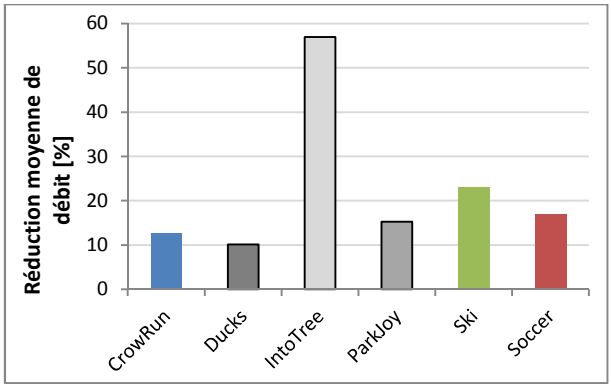


Figure 3.18. Réduction de débit moyen sur les séquences test possibles avec le filtre BilAWA perceptuel

Au total, trente comparaisons sont présentées par séance, listées dans le Tableau 3.6, soit une séance de test de 18 minutes par utilisateur. Pour chaque séquence test (*CrowdRun*, *Ski*, *Soccer*), nous comparons les versions préfiltrées (AWA, BilAWA et Tbilateral) à la version sans préfiltre, encodées dans les trois conditions d'encodage décrites précédemment. Nous utilisons une ancre en appliquant le filtre Gaussien 11x11 comme prétraitement, ce qui produit des séquences fortement floutées.

Sur les 16 observateurs, cinq scénarios différents de présentations des trente comparaisons ont été utilisés. Les cinq scénarios commencent tous par la présentation des ancres pour les trois séquences, ensuite l'ordre des comparaisons a été déterminé aléatoirement, ainsi que l'ordre de présentation entre la séquence encodée avec et sans prétraitement (Cf. Tableau 3.6). Ces précautions nous permettent de réduire le biais dû à l'ordre de visionnage.

Aléatoire

Ordre de présentation	Séquence Originale	Séquence Référence	Séquence A noter	Conditions d'encodage	
				QP	GOP
1	CrowdRun	Sans préfiltre	Gaussien 11x11	22	Inter
2	Ski	Sans préfiltre	Gaussien 11x11	22	Inter
3	Soccer	Sans préfiltre	Gaussien 11x11	22	Inter
Ordre Aléatoire parmi les séquences ci-contre	CrowdRun	Sans préfiltre	AWA 3x3	22	Intra
	CrowdRun	Sans préfiltre	AWA 3x3	22	Inter
	CrowdRun	Sans préfiltre	AWA 3x3	27	Inter
	CrowdRun	Sans préfiltre	BilAWA 11x11	22	Intra
	CrowdRun	Sans préfiltre	BilAWA 11x11	22	Inter
	CrowdRun	Sans préfiltre	BilAWA 11x11	27	Inter
	CrowdRun	Sans préfiltre	TBil 11x11	22	Intra
	CrowdRun	Sans préfiltre	TBil 11x11	22	Inter
	CrowdRun	Sans préfiltre	TBil 11x11	27	Inter
	Ski	Sans préfiltre	AWA 3x3	22	Intra
	Ski	Sans préfiltre	AWA 3x3	22	Inter
	Ski	Sans préfiltre	AWA 3x3	27	Inter
	Ski	Sans préfiltre	BilAWA 11x11	22	Intra
	Ski	Sans préfiltre	BilAWA 11x11	22	Inter
	Ski	Sans préfiltre	BilAWA 11x11	27	Inter
	Ski	Sans préfiltre	TBil 11x11	22	Intra
	Ski	Sans préfiltre	TBil 11x11	22	Inter
	Ski	Sans préfiltre	TBil 11x11	27	Inter
	Soccer	Sans préfiltre	AWA 3x3	22	Intra
	Soccer	Sans préfiltre	AWA 3x3	22	Inter
	Soccer	Sans préfiltre	AWA 3x3	27	Inter

Tableau 3.6. Design du test subjectif

Pour les cinq scénarios de comparaisons, la séance débute toujours par la comparaison entre la séquence encodée sans prétraitement et avec l'ancre, pour les trois séquences. Nous réalisons ceci avec deux objectifs :

- d'une part, en présentant des séquences fortement floutées on s'assure que nos observateurs perçoivent bien la dégradation lorsqu'elle est visible ce qui valide par la suite le fait qu'ils ne perçoivent pas les dégradations plus fines.

- d'autre part, cela permet de les informer dès le départ de l'artefact que l'on cherche à mesurer. Nous introduisons ici un biais qui permet d'éviter d'attirer l'attention de l'observateur sur des artefacts de codage qu'on ne cherche pas à mesurer.

3. 4. 3. Résultats

Nous proposons ci-après plusieurs niveaux d'analyse des résultats de réduction de débit et de qualité subjective. Pour la qualité subjective nous présentons à chaque fois la moyenne des notes (MOS) ainsi que l'intervalle de confiance à 95% $\delta_{x,95\%}$ calculé en fonction de l'écart-type std_x de la distribution x et du nombre d'observations N:

$$\delta_{x,95\%} = 1.96 \times \frac{std_x}{\sqrt{N}}$$

Équation 3-16. Intervalle de confiance à 95%

Ainsi si l'on augmentait le nombre des observateurs, dans 95% des cas la note donnée serait comprise dans l'intervalle $\left[\overline{\mu}_x - \delta_{x,95\%} \; ; \; \overline{\mu}_x + \delta_{x,95\%}\right]$, avec $\overline{\mu}_x$ la moyenne des notes subjectives considérées.

L'échelle de notation utilisée lors du test subjectif est donnée ci-contre. Les notes subjectives moyennes (MOS) données dans la suite de ce paragraphe sont toujours pour la comparaison de la version encodée avec prétraitement par rapport à la version encodée sans. Ainsi un MOS de -1 signifie que la version prétraitée est jugée légèrement moins bonne que la version encodée sans.

-3 — Beaucoup moins bon
-2 — Moins bon
-1 — Légèrement moins bon
0 — Identique
1 — Légèrement mieux
2 — Mieux
3 — Beaucoup mieux

3. 4. 3. 1. Résultats moyens pour toutes les conditions de test

Le Tableau 3.7 présente les résultats de réduction de débit et de MOS (Mean Opinion Score) pour les trois séquences, les trois préfiltres et les trois conditions de test. On note que les notes varient de 0.19 à -0.5, les différences sont donc globalement peu perçues entre les deux versions.

La plus forte réduction de débit est donnée par le filtre BilAWA (28,73%) pour la séquence Ski encodée en GOP Inter à QP22 pour un débit autour de 10Mbit/s. La dégradation amenée par le filtre BilAWA dans ces conditions est de -0.31 avec un intervalle de confiance de 0.47, ainsi dans 95% des cas les observateurs donneraient une note comprise entre -0.78 et 0.16, la différence de qualité est par conséquent perçue mais est jugée très faible.

La réduction de débit minimum est donnée par le filtre TBilateral pour la séquence *CrowdRun* encodée en Inter à QP22 (9.51%) avec une différence qualité de -0.5 et un intervalle de confiance de 0.40.

			x264	AWA 3x3				BILAWA 11x11				TBILATERAL 11x11			
			Débit [Mbit/s]	Débit [Mbit/s]	ΔDébit [%]	MOS	$\delta_{95\%}$	Débit [Mbit/s]	ΔDébit [%]	MOS	$\delta_{95\%}$	Débit [Mbit/s]	ΔDébit [%]	MOS	$\delta_{95\%}$
INTRA	QP22	CrowdRun	110,48	96,82	12,36	0,00	0,18	93,04	15,79	-0,13	0,43	97,40	11,84	-0,06	0,52
		Ski	27,47	22,73	17,24	-0,50	0,54	20,88	23,99	-0,31	0,46	21,63	21,27	-0,19	0,41
		Soccer	37,11	32,66	11,99	0,00	0,44	30,75	17,16	-0,06	0,46	32,23	13,15	-0,50	0,36
INTER	QP22	CrowdRun	53,41	47,41	11,24	-0,44	0,56	45,05	15,66	0,19	0,48	48,33	9,51	-0,50	0,40
		Ski	13,24	10,24	22,68	0,06	0,46	9,44	28,73	-0,13	0,47	9,96	24,76	-0,19	0,41
		Soccer	16,28	13,34	18,01	-0,13	0,30	12,32	24,33	-0,38	0,53	13,42	17,56	-0,50	0,31
	QP27	CrowdRun	28,33	25,59	9,68	-0,56	0,40	24,31	14,20	-0,38	0,43	26,25	7,35	-0,19	0,45
		Ski	6,28	5,48	12,72	-0,31	0,43	5,14	18,11	-0,25	0,33	5,39	14,16	-0,19	0,37
		Soccer	7,29	6,48	10,99	-0,25	0,42	6,12	15,95	0,00	0,47	6,61	9,22	-0,31	0,50

Tableau 3.7. Résultats de réduction de débit et de MOS pour les trois séquences, les trois préfiltres et les trois conditions d'encodage

Le Tableau 3.8 et le Tableau 3.9 présentent les résultats des métriques objectives PSNR et LPC-SI respectivement.

Le PSNR indique une dégradation systématique des séquences encodées avec les différents préfiltres. A même qualité subjective, l'analyse du PSNR permet de renseigner sur le filtre réduisant le plus fortement le contenu haute-fréquence. Le Filtre BilAWA 11x11 apporte les plus fortes réductions de PSNR et par conséquent les plus fortes réductions de débit (Tableau 3.7).

La métrique de LPC-SI indique une très faible variation du niveau de flou entre les séquences encodées avec et sans prétraitement. Ainsi les résultats de LPC-SI sont cohérents avec les notes subjectives indiquant que les préfiltres ne dégradent pas la qualité des séquences encodées tout en apportant une forte réduction de débit.

			x264	**AWA 3x3**		**BILAWA 11x11**		**TBILATERAL 11x11**	
			PSNR	PSNR	ΔPSNR [dB]	PSNR	ΔPSNR [dB]	PSNR	ΔPSNR [dB]
INTRA	QP22	CrowdRun	**40,39**	36,24	-4,16	35,07	-5,32	36,32	-4,07
		Ski	**43,97**	41,99	-1,98	40,82	-3,15	41,20	-2,77
		Soccer	**43,36**	41,28	-2,09	40,20	-3,17	40,95	-2,41
		Moyenne	**42,57**	39,83	-2,74	38,69	-3,88	39,49	-3,08
INTER	QP22	CrowdRun	**39,22**	35,95	-3,27	34,80	-4,42	36,02	-3,20
		Ski	**42,86**	41,43	-1,43	40,43	-2,43	40,78	-2,08
		Soccer	**42,09**	40,64	-1,45	39,72	-2,38	40,41	-1,68
		Moyenne	**41,39**	39,34	-2,05	38,31	-3,08	39,07	-2,32
	QP27	CrowdRun	**35,24**	33,55	-1,69	32,73	-2,51	33,63	-1,62
		Ski	**40,02**	39,34	-0,68	38,69	-1,33	38,97	-1,04
		Soccer	**39,05**	38,28	-0,77	37,66	-1,39	38,19	-0,86
		Moyenne	**38,10**	37,06	-1,05	36,36	-1,74	36,93	-1,17

Tableau 3.8. Résultats PSNR pour les trois séquences, les trois préfiltres et les trois conditions d'encodage

			x264	**AWA 3x3**		**BILAWA 11x11**		**TBILATERAL 11x11**	
			LPC-SI	LPC-SI	ΔLPC-SI	LPC-SI	ΔLPC-SI	LPC-SI	ΔLPC-SI
INTRA	QP22	CrowdRun	0,923	0,924	0,001	**0,925**	0,002	0,924	0,001
		Ski	0,893	0,895	0,002	0,895	0,002	**0,896**	0,003
		Soccer	0,924	0,925	0,001	**0,925**	0,001	0,925	0,001
		Moyenne	0,913	0,915	0,001	0,915	0,002	**0,915**	0,002
INTER	QP22	CrowdRun	0,923	0,924	0,001	**0,925**	0,002	0,924	0,001
		Ski	0,894	0,895	0,001	0,895	0,001	**0,897**	0,003
		Soccer	0,924	0,925	0,001	**0,925**	0,001	0,925	0,001
		Moyenne	0,914	0,915	0,001	0,915	0,001	**0,915**	0,002
	QP27	CrowdRun	0,922	0,924	0,001	**0,924**	0,002	0,924	0,001
		Ski	0,892	0,893	0,001	0,893	0,001	**0,895**	0,002
		Soccer	0,924	0,924	0,001	0,924	0,001	**0,924**	0,001
		Moyenne	0,913	0,914	0,001	0,914	0,001	**0,914**	0,001

Tableau 3.9. Résultats LPC-SI pour les trois séquences, les trois préfiltres et les trois conditions d'encodage

On propose de regarder plus en détail les résultats obtenus pour la meilleure note subjective du test, à savoir pour la séquence préfiltrée avec AWA 3x3 puis encodée en GOP Intra à QP 22, ainsi que pour une des plus mauvaises notes obtenues par la séquence Soccer préfiltrée avec le TBilateral 11x11 puis encodée en GOP IBBP12 à QP 22.

La Figure 3.19 présente la répartition des notes subjectives données par les seize observateurs pour deux cas étudiés. Comme le traduit l'intervalle de confiance, les observateurs ont jugé quasiment identique la qualité des séquences encodées en GOP Intra à QP 22 avec et sans prétraitement (Figure 3.19 (a)). Pour le cas de la séquence Soccer (Figure 3.19 (b)) les notes sont plus dispersées et trois observateurs ont jugé la séquence prétraitée moins bonne que la séquence encodée sans prétraitement (note -2).

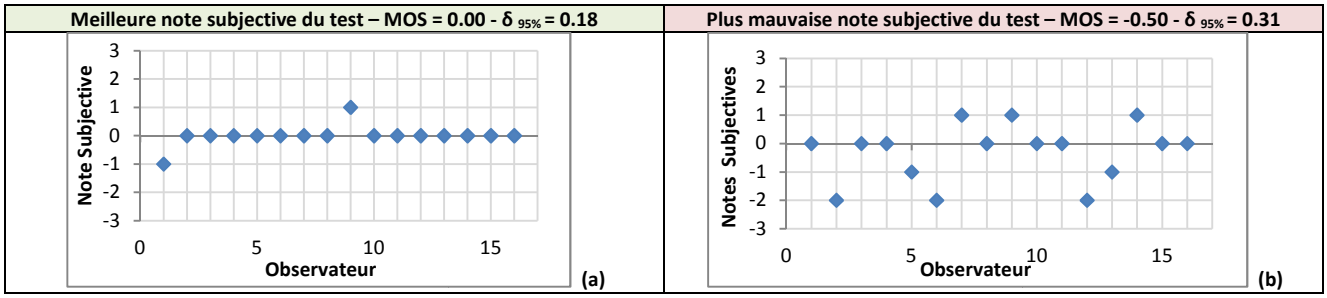


Figure 3.19. Note subjective pour la meilleure (a) et la plus mauvaise (e) note du test

La Figure 3.20 présente la comparaison visuelle des deux cas présentés ci-dessus en proposant deux agrandissements dans les séquences encodées avec et sans prétraitement. Pour la séquence *CrowdRun*, le flou introduit par le filtre AWA 3x3 est uniquement visible dans l'arbre. Pour la séquence Soccer, le filtre TBilateral amène quant à lui une impression générale de flou ce qui explique la mauvaise note CMOS pour cette comparaison.

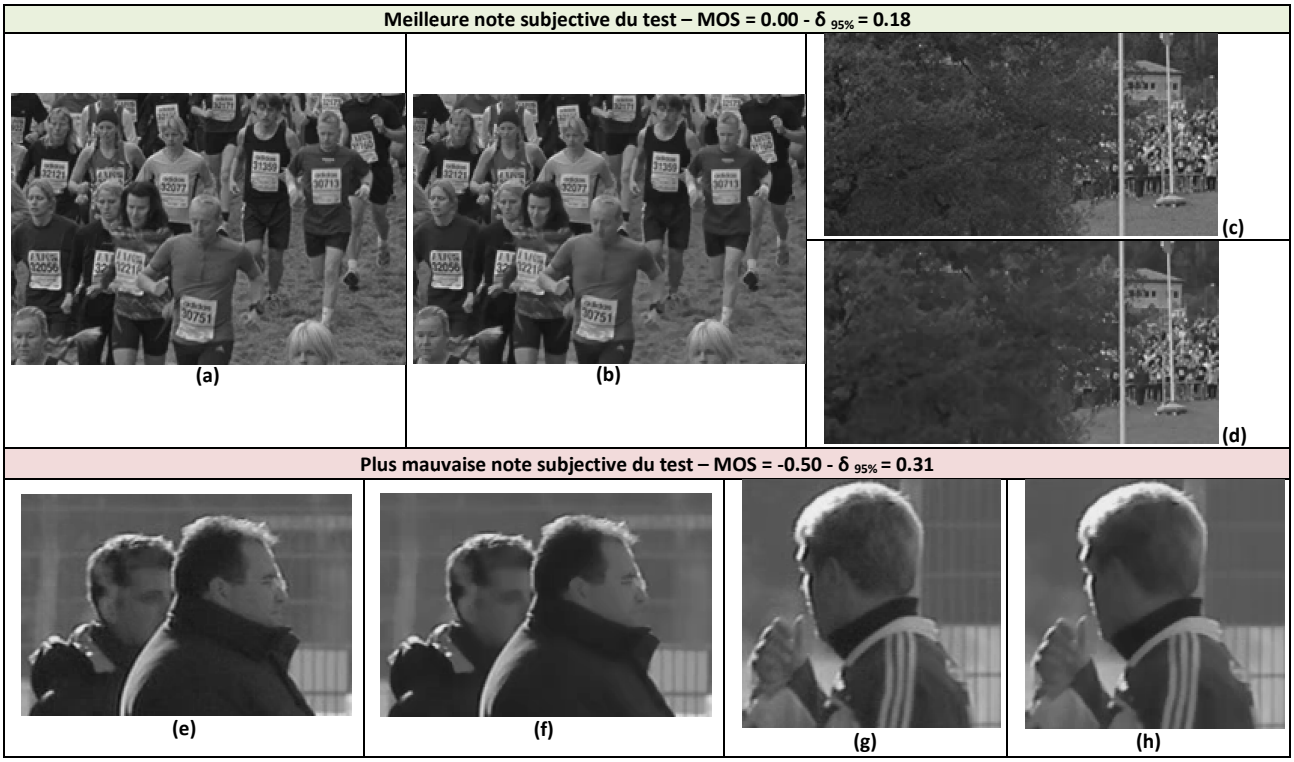


Figure 3.20. Comparaison visuelle pour la meilleure (a-d) et la plus mauvaise (e-h) note subjective
(a), (c) x264 GOP Intra QP 22 – (b), (d) AWA 3x3 + x264 GOP Intra QP 22
(e), (g) x264 GOP IBBP12 QP 27 – (b), (d) AWA 3x3 + x264 GOP IBBP12 QP 27

3. 4. 3. 2. Résultats par filtre et condition d'encodage

Pour simplifier la lecture des résultats à trois dimensions (séquences, encodage, préfiltre), la Figure 3.21 présente les résultats par filtre et par condition d'encodage moyennés sur les trois séquences. La note MOS ainsi que l'intervalle de confiance pour chaque couple filtre/conditions d'encodage y sont représentés.

Les différents filtres donnent les moins bonnes performances en GOP Inter à QP 27 en termes de réduction de débit et qualité subjective (points de mesure rouges). Mis à part pour le filtre TBilateral, les meilleures performances sont obtenues en GOP Inter à QP 22 (points de mesure verts). Cette observation a déjà été faite au chapitre précédent, lorsque le pas de quantification augmente, le filtrage apporte moins de réduction de débit car la quantification joue d'elle-même le rôle de filtre passe-bas.

A même paramètre de quantification (QP 22), les filtres apportent une réduction de débit moins importante en GOP Intra qu'en GOP Inter IBBP(12). Pour les filtres AWA 3x3 et BilAWA 11x11 les qualités perçues sont à peu près équivalentes dans les deux GOP, mais Le filtre TBilateral apporte une dégradation plus visible en GOP Inter.

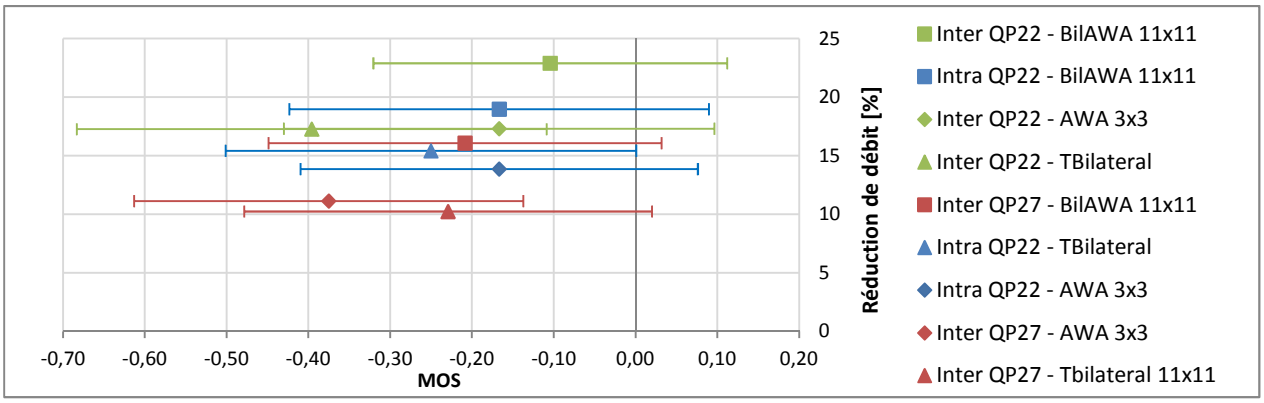


Figure 3.21. Résultats moyens par Filtre et configuration d'encodage

3. 4. 3. 3. Résultats par filtre et séquence

En moyennant les résultats sur les trois conditions d'encodage, on en ressort l'impact des filtres en fonction des contenus, présentés par la Figure 3.22. Les résultats sont analysés en considérant les activités spatiales et temporelles présentées dans la Figure 3.18. Les trois séquences ont une quantité d'information temporelle à peu près identique mais diffèrent par leur activité spatiale, la séquence *CrowdRun* contient le plus d'activité, puis la séquence *Soccer* et ensuite la séquence *Ski*. En globalité, les performances des filtres sur les séquences sont inverses à la quantité d'information spatiale, c'est-à-dire que les meilleures performances sont obtenues pour la séquence Ski (points de mesure verts) et ainsi de suite.

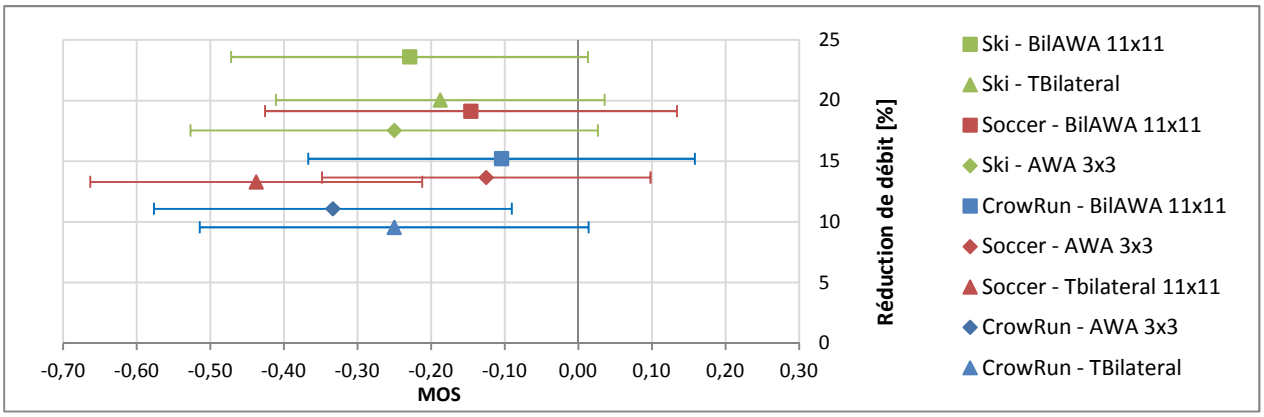


Figure 3.22. Résultats moyen par Filtre et séquence

3. 4. 3. 4. Résultats par filtre

En moyennant les résultats à la fois sur les trois séquences et les trois conditions d'encodage, il en ressort les performances globales des filtres (Figure 3.23). Le filtre BilAWA donne les meilleures performances à la fois en termes de réduction de débit et de qualité. En effet, pour une très faible réduction de qualité (-0.16 avec un intervalle confiance de 0.15), le filtre amène 19% de réduction de débit en moyenne sur toutes les conditions de test.

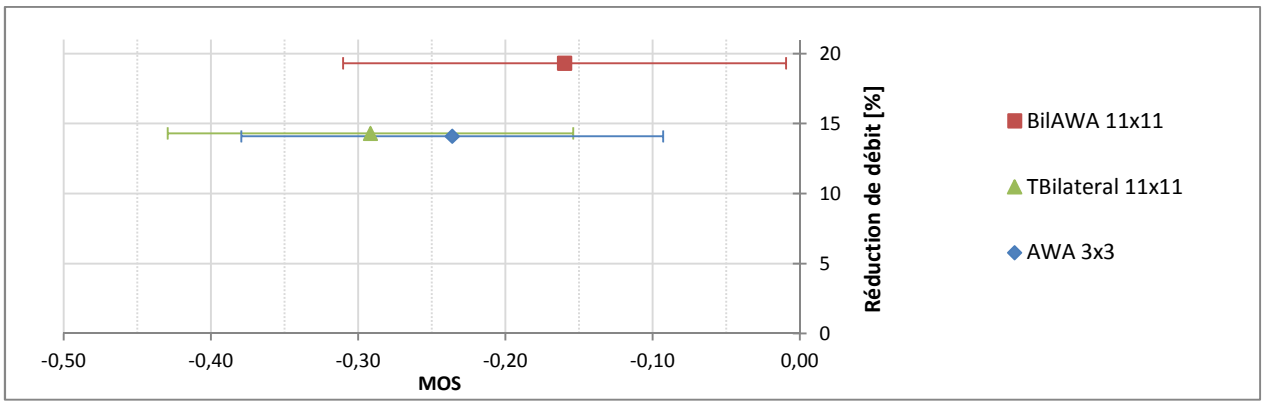


Figure 3.23. Résultats moyen par Filtre

3. 5. Conclusion

En poursuivant notre étude des prétraitements pour l'encodage vidéo HD basés sur des filtres passe-bas de la littérature, dans ce chapitre nous avons étudié l'intérêt d'augmenter la taille de support de filtrage du filtre AWA, ainsi qu'un autre filtre bien connu, le filtre Bilatéral. Notre étude a porté sur une application de réduction de bruit et de prétraitement perceptuel pour des images et séquences HD. Nos travaux nous ont amené à définir deux nouveaux filtres, le filtre BilAWA et le filtre TBilateral.

Dans une application de réduction de bruit, nous avons montré que le filtre TBilateral permet d'obtenir le meilleur compromis entre réduction de bruit et préservation de la structure de l'image devant les filtres AWA et Bilatéral bien connus de la littérature.

Pour une application de prétraitement pour l'encodage H.264/AVC, nous avons montré que le filtre BilAWA contrôlé par le modèle JND de X. Yang permet d'obtenir de fortes réductions de débit pour une différence de qualité peu perceptible comparativement aux séquences encodées sans prétraitement. Ce nouveau filtre améliore les performances du prétraitement proposé au chapitre précédent, offrant une réduction moyenne de débit de 20%.

Dans le chapitre qui suit, nous allons nous intéresser au cas de l'encodage à débit constant en étudiant la possibilité de contrôler l'allocation binaire par le modèle JND afin d'améliorer la qualité perçue.

Chapitre 4. Contribution à la quantification Adaptative guidée par le modèle JND pour l'encodeur H.264/AVC en CBR

4. 1. Introduction

Dans les deux chapitres précédents, nous avons proposé un prétraitement perceptuel pour réduire les informations peu/pas perceptibles d'une séquence vidéo dans le but de réduire le débit nécessaire à sa transmission sans altérer la qualité perçue. Pour cela nous avons proposé plusieurs filtres passe-bas basés sur deux filtres classiques de la littérature, le filtre AWA et le filtre bilatéral, contrôlés par le modèle JND proposé par Yang et al. [66]. Nous avons démontré l'intérêt de nos prétraitements pour la réduction de débit en encodage VBR (Variable Bitrate) sans contrôle de débit. Nous souhaitons maintenant nous intéresser au cas de l'optimisation de l'encodage CBR, notre but est d'améliorer la qualité perçue de la séquence encodée pour un débit cible.

Afin d'améliorer la qualité à débit constant, nous verrons dans ce chapitre qu'il nous a fallu agir sur le contrôle de débit en adaptant le paramètre de quantification par macrobloc en fonction du modèle JND. Contrairement à notre solution de prétraitement entièrement indépendante du codeur proposée pour l'amélioration de l'encodage VBR, pour l'encodage CBR nous avons apporté des modifications au sein du codeur. Comme nous l'avons vu précédemment l'encodeur AQILIM de Digigram est basé sur la librairie logicielle H.264/AVC développée par Rovi (Mainconcept) [2] qui ne nous permet pas d'avoir accès au cœur d'encodage. Pour nos développements en encodage CBR nous avons choisi de travailler avec le codec open source x264 présenté au chapitre 1.

Dans ce chapitre nous proposons une amélioration de la quantification adaptative utilisée par le codec x264 qui souffre de l'effet de Ringing, apparaissant aux contours d'une scène à bas débit. Pour cela nous nous basons sur les travaux de [84] qui adaptent par macrobloc, le paramètre de quantification décidé par le contrôle de débit au niveau image, en fonction d'un seuil JND. Nous proposons ainsi une nouvelle quantification adaptative contrôlée par le modèle JND en texture de X. Yang. Nos tests réalisés sur des séquences HD montrent une réduction de l'effet de Ringing amenant une amélioration de la qualité globalement perçue. Nous nous concentrons sur la réduction de l'effet de Ringing en encodage Intra qui est plus sensible à ce type d'artefact, toutefois nous vérifions également l'intérêt de notre solution en GOP classique Inter. Pour la mesure des résultats nous utilisons la métrique de Ringing proposée par [110] qui sera décrite dans ce chapitre. Les résultats montrent que notre solution permet de réduire ce type d'artefact et de préserver les zones faiblement texturées dans la majorité des cas testés amenant ainsi une amélioration de la qualité globale des séquences vidéo.

4. 2. Etude de l'impact du prétraitement sur l'allocation binaire en encodage VBR et CBR

Comme nous l'avons vu précédemment, le préfiltre perceptuel permet de réduire le débit nécessaire à l'encodage d'une séquence sans altérer la qualité perçue. Nous souhaitons étudier maintenant l'impact du préfiltre en encodage CBR, pour cela nous prenons l'exemple de la séquence *ParkJoy* 1280x720 50p encodée à 35 Mbit/s, les autres séquences testées présentent des résultats similaires. Le débit de 35 Mbits/s correspond à la qualité des images intra de cette séquence encodée en GOP Inter classique à 6Mbit/s[11], ce qui correspond à la gamme de débits utilisés par les

[11] Pour établir le débit Intra, nous avons encodé la séquence avec deux GOP Inter classiquement utilisés, le GOP IBBP de longueur 12 et 33 images. On calcule ensuite le débit moyen attribué aux images intra de ces deux encodages.

clients de Digigram. La Figure 4.1 présente les courbes de budget binaire (a) et de QP moyen (b) par image avec et sans prétraitement. On note que le débit cible est atteint avec ou sans prétraitement, l'application du préfiltre ne perturbe donc pas le contrôle de débit (a). A même débit, la séquence prétraitée est encodée avec un QP moyen inférieur en moyenne de 0.8 sur la séquence. Cela s'explique par le fait que le préfiltre réduit la complexité des images et permet au codeur de réduire le pas de quantification pour représenter le contenu avec un budget binaire donné. Nous pouvons ainsi nous attendre à ce que le préfiltre réduise les artefacts de codage en réduisant le pas de quantification, toutefois nous ne notons pas d'amélioration de qualité de la séquence prétraitée comparativement à la séquence encodée sans prétraitement (nous ne présentons donc pas les images).

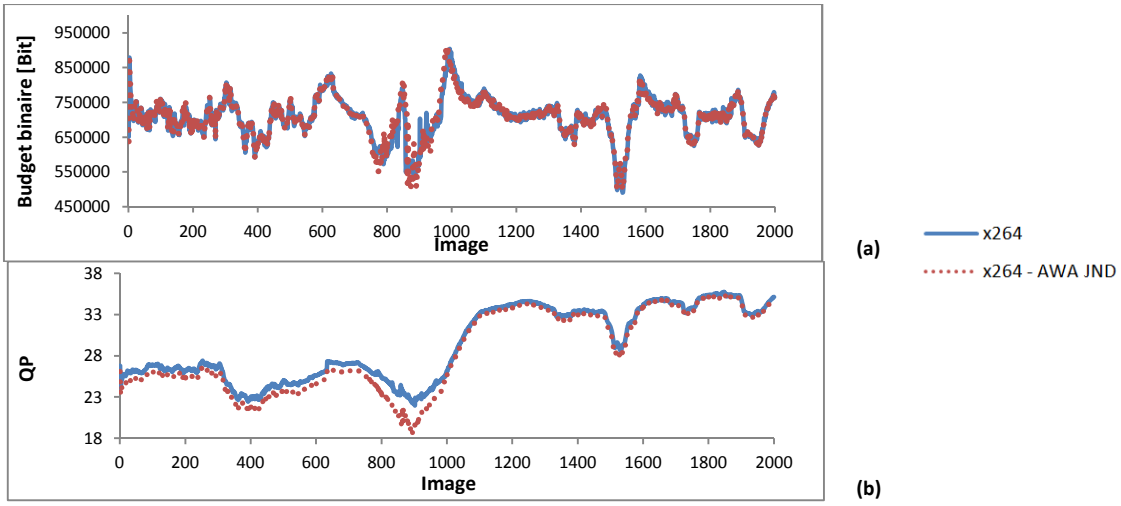


Figure 4.1. Comparaison de l'allocation binaire et du QP Moyen de l'encodage x264 avec et sans prétraitement
Résultats par image de la séquence ParkJoy encodées à 35 Mbit/s
(a) Budget binaire par images – (b) QP moyen par images

Afin de comprendre pourquoi le préfiltre ne permet pas d'améliorer la qualité perçue, nous analysons son impact sur l'allocation binaire en étudiant l'évolution du nombre de bits consommés par chaque macrobloc d'une image. Pour cela nous prenons l'exemple d'une image particulière de la séquence *ParkJoy* présentée par la Figure 4.3. Les figures (d) et (e) représentent le rapport de budget binaire par macrobloc (16x16) entre l'image encodée avec et sans préfiltre, en encodage VBR à QP 32 et CBR à 55 Mbit/s respectivement. Pour les cartes de rapport d'allocation binaire, le code couleur suivant est utilisé : les macroblocs verts voient leur budget binaire réduit par le préfiltre tandis que les macroblocs rouges voient leur budget augmenter. Les macroblocs noirs sont codés avec le même nombre de bits que le préfiltre soit appliqué ou non. Plus la couleur (rouge ou verte) est claire et plus le filtre modifie l'allocation binaire.

4. 2. 1. Encodage VBR

La carte (d) de la Figure 4.3. présente le rapport d'allocation binaire entre l'image encodée avec et sans prétraitement en GOP Intra à QP constant égal à 32. La carte de JND (b) contrôle le préfiltre AWA, cette carte représente donc la force de filtrage appliquée aux pixels de l'image. Pour analyser la modification d'allocation binaire apportée par le préfiltre par macrobloc, nous représentons la carte de JND moyen par macrobloc (c), plus la carte est claire, plus la force de filtrage appliquée en moyenne aux pixels d'un macrobloc est importante, et par conséquent plus on s'attend à réduire le budget binaire. L'analyse des résultats montre que :

- Comme attendu, en majorité (pour 60% des macroblocs) le préfiltre permet de réduire le budget en réduisant les détails de l'image (macroblocs rouge (d)).

- On comparant les cartes (c) et (d) on s'aperçoit que la modification d'allocation binaire apportée par le préfiltre en VBR (d), suit majoritairement la carte de JND moyen (c), par exemple les macroblocs appartenant au banc d'herbe voient leur budget réduit tandis que ceux appartenant à l'eau et au feuillage ensoleillé ne sont pas modifiés.

- Toutefois, sur les troncs d'arbres, l'application du préfiltre ne provoque pas de réduction de débit bien que cette zone soit fortement filtrée d'après le JND (macroblocs clairs carte (c)). Ceci s'explique par le fait que la zone contient peu de détails, ainsi l'application du filtre provoque très peu de perte d'informations bien que la force de filtrage soit forte.

- On note également que l'application du filtre provoque une augmentation du budget sur quelques macroblocs (macroblocs rouge (d)). En effet, dans certains cas, la modification du contenu apportée par le filtre rend l'image plus difficilement prédictible par les modes intra. La Figure 4.2 prend l'exemple d'un macrobloc particulier pour lequel le filtre augmente le budget binaire. Avec ou sans préfiltre, les partitions 4x4 sont choisies pour prédire le macrobloc 16x16, cependant certains modes 4x4 sont modifiés par le préfiltre (représentés en gris (c)). Bien que le contenu soit lissé par le préfiltre, la prédiction est moins précise et il en résulte une augmentation de l'amplitude des coefficients DCT représentée par les figures (b) et (d) respectivement pour la version encodée sans et avec préfiltre.

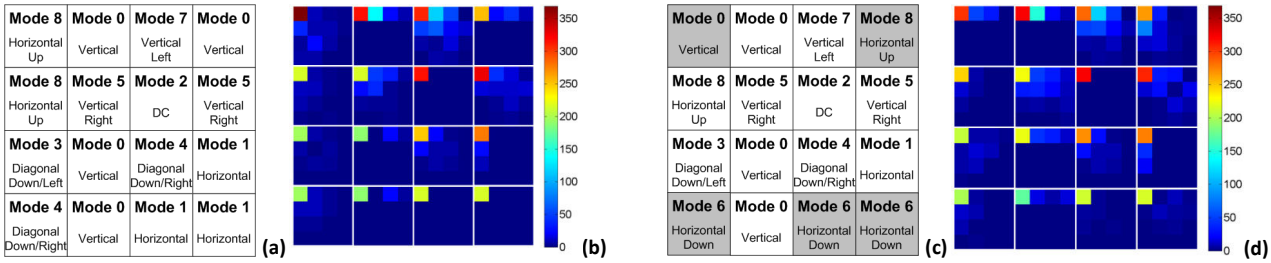


Figure 4.2. Modifications apportées par le filtre sur un macrobloc particulier en encodage VBR
Mode de prédiction 4x4 pour le macrobloc non filtrée (a) et filtré (c)
Valeur absolue des coefficients DCT pour le résidu du macrobloc non filtré (b) et filtré (d)

4. 2. 2. Encodage CBR

Pour étudier l'impact du préfiltre en encodage CBR, la séquence *ParkRun* est encodée avec et sans préfiltre au débit obtenu par l'encodage Intra à QP 32 soit 55Mbit/s. La Figure 4.3.e présente le rapport d'allocation binaire entre l'image encodée avec et sans prétraitement en GOP Intra. L'étude est réalisée sur l'image 400 de la séquence pour laisser le temps au contrôle de débit de se stabiliser.

- Comme en encodage VBR, le préfiltre permet toujours de réduire le budget alloué aux macroblocs fortement filtrés (feuillage et banc d'herbe).

- Le budget binaire sauvé par le préfiltre est réalloué par le contrôle de débit pour garantir le débit cible demandé par l'utilisateur. On voit clairement que la réallocation ne suit pas la carte de JND moyen (c), tout le budget est concentré sur les premières lignes de l'image. Ainsi la qualité globale ressentie de cette image et plus largement de la séquence ne peut être améliorée par le préfiltre.

Ces observations illustrent que pour améliorer la qualité perçue à débit constant, le budget binaire gagné par le préfiltre doit être réalloué aux zones perceptivement importantes de l'image indiquées par le modèle JND. Il faut ainsi contrôler l'allocation binaire à l'aide du modèle JND. Des travaux sur l'allocation binaire contrôlée par un modèle perceptuel ont été réalisés dans la littérature du codage perceptuel, dont nous proposons une présentation non exhaustive dans le paragraphe suivant.

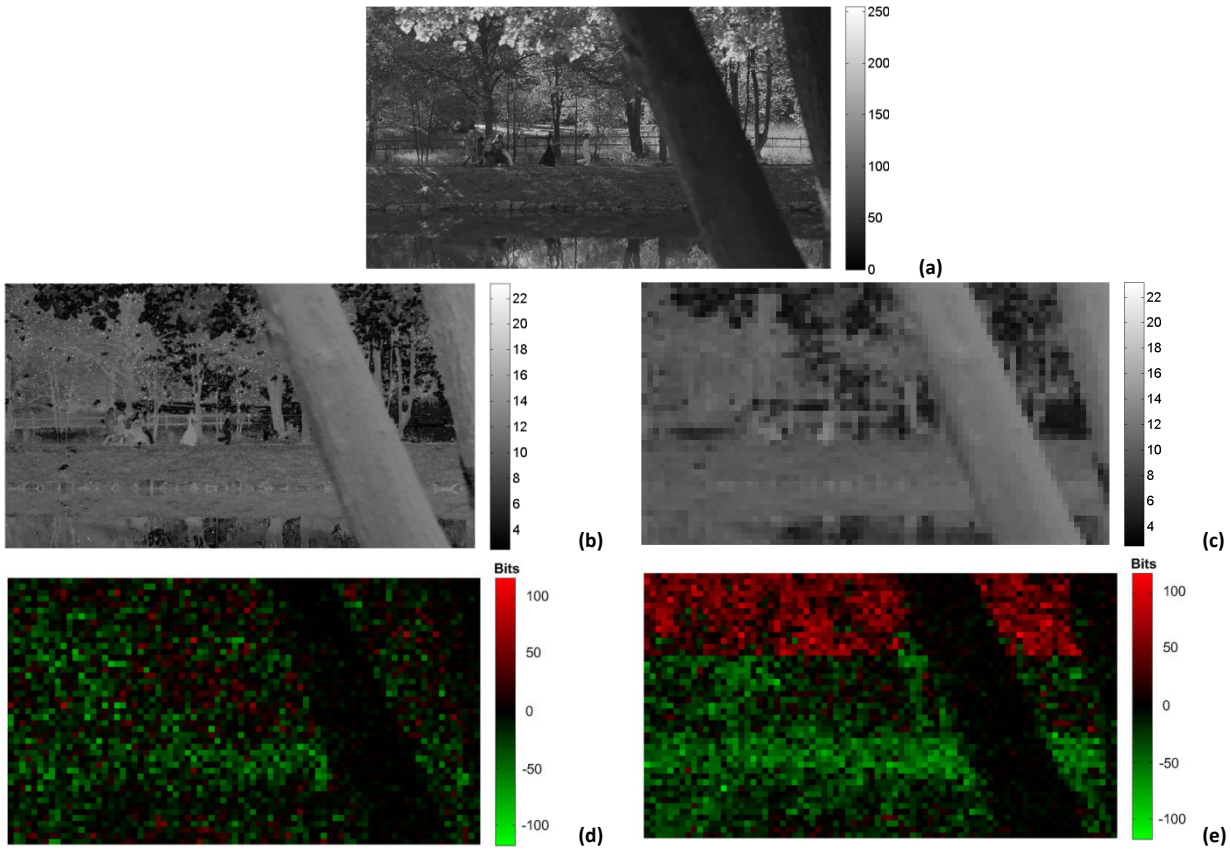


Figure 4.3. Effet du préfiltre sur l'allocation binaire par macrobloc en encodage VBR et CBR
(a) Image de la séquence IntoTree 1280x720 – (b) Carte de JND - (c) Carte de JND moyen par macrobloc – (d) Carte de différence d'allocation binaire entre la version encodée en Intra avec et sans préfiltre en VBR à QP32 - (e) Carte de différence d'allocation binaire entre la version encodée en Intra avec et sans préfiltre en CBR à 55 Mbit/sec

4. 3. Etat de l'art du codage perceptuel

Dans l'idée de compresser toujours plus efficacement les séquences vidéo, la réduction des redondances psycho-visuelles constitue la base du codage dit perceptuel. Pour ce faire, des modèles psycho-visuels sont intégrés aux différentes étapes d'un encodage vidéo. Dans ce paragraphe, on présente un état de l'art non exhaustif des techniques de codage perceptuel ainsi que les choix qui nous ont conduits à la solution proposée. Un état de l'art plus complet des techniques de codage perceptuel est proposé par les auteurs de [111].

La Figure 4.4 présente un schéma bloc simplifié d'un encodeur vidéo avec un contrôle de débit au niveau image. Pour rappel du chapitre 1, les images sont d'abord analysées afin de choisir le paramètre de quantification le mieux adapté au débit cible et la complexité de l'image, à partir d'un modèle R-Q (Rate-Quantization). Ce QP choisi pour l'image peut être modulé par macrobloc en définissant par exemple une carte de ΔQP par macrobloc. Le module RDO choisi pour chaque macrobloc la prédiction intra ou inter la plus adaptée en fonction du QP choisi à l'étape précédente. Ensuite l'information résiduelle est encodée par transformée DCT, quantification et codage entropique.

Finalement, le nombre de bits utilisés pour coder chaque macrobloc est accumulé et permet au contrôle de débit de réguler l'allocation en fonction du débit cible.

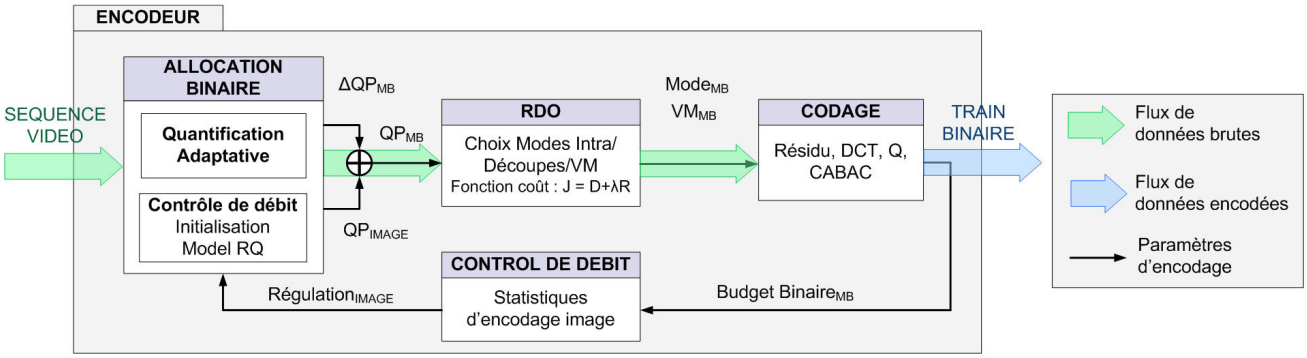


Figure 4.4. Schéma bloc d'un encodeur vidéo, cas d'un contrôle de débit au niveau image

Dans la suite de ce paragraphe on distingue trois techniques de codage perceptuel intervenant à différents stades de l'encodage. Premièrement, l'intégration de modèle perceptuel dans le module de RDO permet de guider les choix du codeur de manière à sélectionner les modes de prédiction intra et vecteurs de mouvement permettant de représenter fidèlement un macrobloc au sens perceptuel. Deuxièmement, un préfiltre perceptuel appliqué sur les informations résiduelles permet de réduire les informations significativement non-importantes. Troisièmement, l'allocation peut être contrôlée par des critères perceptifs conditionnant le choix du paramètre de quantification. Les trois types de solutions que nous présentons ci-dessous sont indépendantes et peuvent être couplés [112], [70], [84], [55].

Pour rappel, le module RDO choisit le mode intra ou vecteur mouvement donnant le meilleur compromis entre le nombre de bits R utilisés pour coder le macrobloc et la distorsion D entre le macrobloc courant et sa version reconstruite. La qualité du compromis est mesurée par la fonction coût R-D *J* à l'aide du multiplicateur de Lagrange λ. La meilleure prédiction est celle donnant la fonction coût R-D la plus faible :

$$J = D + \lambda \times R$$ ***Équation 4.1.*** *Fonction coût RD*

En intégrant un modèle perceptuel dans le module RDO (Rate Distortion Optimization), les choix des modes de prédiction intra, découpes de macroblocs inter et vecteurs mouvement sont guidées par la qualité subjective au lieu de la simple différence numérique. Ainsi, les auteurs de [113] proposent d'utiliser un modèle de masquage spatio-temporel au lieu des classiques SAD (Sum of Absolute Difference) et SSD (Sum of Squared Difference) dans le calcul de la fonction coût débit distorsion. Dans une approche similaire, les auteurs dans [114] et [115] proposent d'utiliser la métrique SSIM. La mesure de distorsion notée D_{SSIM} du macrobloc de coordonnées (i,j) devient :

$$D_{SSIM}(i,j) = 1 - SSIM(i,j)$$ ***Équation 4.2.*** *Distorsion - SSIM*

Ainsi les auteurs définissent une nouvelle fonction coût R-D perceptuelle J_{SSIM} où le multiplicateur de Lagrange λ_{SSIM} est estimé périodiquement dans la séquence en fonction de la courbes R-D_{SSIM}.

$$J_{SSIM}(i,j) = D_{SSIM}(i,j) + \lambda_{SSIM} \times R(i,j)$$ ***Équation 4.3.*** *Fonction coût R-D - SSIM*

Les auteurs ont implémenté cette solution dans l'encodeur de référence H.264 JM et comparée au RDO classique basé sur la mesure de distorsion EQM. Les tests ont été réalisés sans contrôle de débit, sur des séquences vidéo à faible résolution CIF et PAL. La Figure 4.5 présente les résultats obtenus par des tests subjectifs suivant la

méthode DSCQS (Double-Stimulus Continuous Quality-Scale) pour deux séquences, ainsi que des agrandissements d'une image encodée suivant les deux méthodes. Les tests mettent en avant une amélioration de la qualité perçue grâce à la méthode proposée.

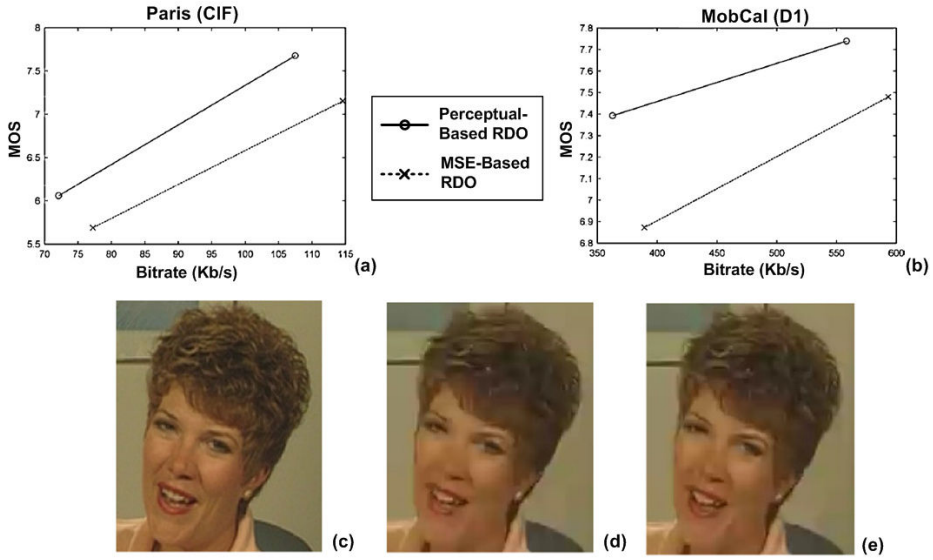


Figure 4.5. Résultats du RDO contrôlé par la métrique SSIM
Illustrations issues de la publication [115]
(a) MOS obtenu par la méthode de tests subjectifs DSCQS pour la séquence Paris et (b) Mobile & Calendar
(c) Image de la séquence originale Mother and Daughter (CIF) – (d) Image encodée à QP 36 par l'encodeur H.264 JM avec un RDO classique basé EQM - (e) Image encodée à QP 36 par l'encodeur H.264 JM avec le RDO basé SSIM proposé

Comme on a pu le voir, la complexité et la qualité d'un encodage dépend des outils de la norme utilisés ainsi que des parties non normatives (estimation de mouvement, RDO et contrôle de débit). Les détections de zones d'intérêt permettent de moduler les outils de codage en fonction des zones pour réduire la complexité/qualité des parties non-importantes et/ou améliorer l'encodage des zones prioritaires. Par exemple, pour des applications de type vidéoconférence, les macroblocs de l'arrière-plan peuvent être transmis en mode « skip » quand ils diffèrent d'une image à l'autre et ainsi concentrer indirectement le débit sur les zones prioritaires qui sont généralement les personnages [68], [54].

Contrairement au RDO, les prétraitements contrôlés par un modèle perceptuel agissent directement sur les données à encoder. Afin de réduire les données non perceptibles, ces prétraitements (dont nous avons déjà parlé au chapitre 2) appliquent des filtres spatiaux [67], [66] ou fréquentiels [68], [55] guidés par un modèle JND pour réduire les résidus ou coefficients fréquentiels dont l'amplitude est inférieure au seuil de perception JND. Ainsi, X. Yang a proposé dans [66] d'intégrer son modèle JND que nous utilisons, dans l'encodeur MPEG2 TM5 sur l'information résiduelle avant application de la DCT comme le montre la Figure 4.6. Le principe est de réduire les différences entre les résidus et la valeur moyenne du bloc résiduel en fonction de la note JND des pixels correspondant aux résidus. La force globale du filtrage est intégrée dans le module RDO pour s'adapter au débit cible et à la complexité des images. La solution est testée sur des séquences au format SD à moyen débit (5Mbit/s), des tests subjectifs suivant la méthode DSCQS montrent une amélioration moyenne de 7.9 MOS sur une échelle de 0 à 100 en comparaison de l'encodage MPEG2 TM5 classique.

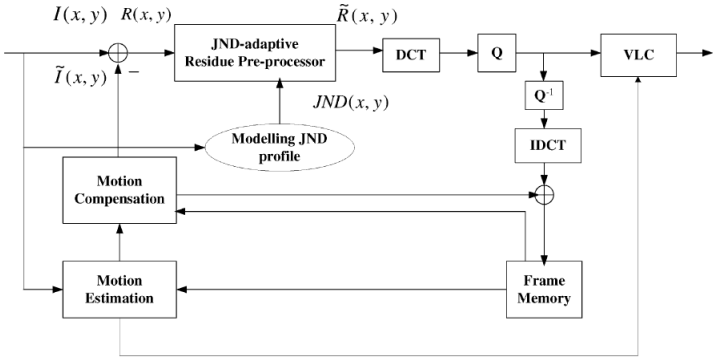


Figure 4.6. Intégration du Modèle JND de X. Yang dans l'encodeur MPEG2 TM5
Illustration issue de la publication [66]

Les prétraitements agissant sur les données résiduelles traitent l'image par macrobloc, par conséquent quand le débit est faible, d'après les tests que nous avons réalisés, ces traitements ont tendance à augmenter l'effet de bloc. En encodage à débit constant, les prétraitements perceptuels permettent de réduire les résidus pas ou peu perceptibles et ainsi gagner du budget binaire qui est réalloué par le contrôle de débit. De cette manière, l'effet des prétraitements sur l'amélioration de la qualité à débit constant est indirect et dépend de la bonne réallocation du budget gagné. En comparaison, les méthodes d'allocation binaire perceptuelle ont une action directe sur la qualité de la séquence encodée en concentrant le budget dans les zones perceptuellement importantes pour le système visuel humain. Pour cela, plusieurs techniques ont été proposées dans la littérature pour agir sur la quantification en contrôlant le pas ou le paramètre de quantification à différentes étapes de l'encodage au niveau macrobloc ou coefficients fréquentiels.

Au début de l'encodage d'une image, le QP appliqué à l'image est estimé à partir d'un modèle R-Q (Rate-Quantization) dont la version classique est basée sur la métrique de distorsion objective MAD (Mean of Absolute Difference) (Cf. Figure 4.4 et Équation 1.24). Afin que le choix initial du paramètre de quantification tienne compte des caractéristiques du système visuel humain, les auteurs dans [55] et [116] proposent de pondérer la mesure de distorsion objective par un modèle perceptif basé respectivement sur une détection de région d'intérêt et d'un JND. Comme pour les techniques de RDO perceptuel vues précédemment, ces travaux agissent sur un modèle d'estimation et non directement sur les données ou sur un paramètre d'encodage. Le modèle de R-Q perceptuel permet de choisir le meilleur QP pour encoder l'image au sens d'un critère perceptuel, tandis que le RDO perceptuel permet de choisir les meilleures prédictions intra et inter.

Pour avoir une action directe sur le pas de quantification, des matrices de quantification perceptuelles peuvent être utilisées en fonction d'un modèle JND défini dans le domaine DCT pour quantifier chaque coefficient fréquentiel en fonction de son seuil de perception [71], [70], [117]. En modifiant l'étape de quantification des encodeurs vidéo, ces traitements nécessitent une modification du décodeur pour que le flux généré soit décodable. Pour les raisons stratégiques que nous avons abordées précédemment, ces traitements n'ont pas retenu notre attention.

Afin de contrôler directement l'allocation binaire, une autre méthode est de modifier le pas de quantification choisi au début de l'encodage d'une image et de l'adapter par macrobloc en fonction de l'activité spatiale des macroblocs ou d'un critère perceptif. Ainsi le QP estimé pour l'image par le contrôle de débit peut être modulé par un facteur multiplicateur dépendant d'un critère perceptif. Par exemple, les auteurs de [112] modifient le paramètre de quantification estimé de l'image en fonction d'une carte de saillance par macrobloc. Sur le même principe, [84] utilisent le modèle JND défini par Yang et al. dans le domaine pixel.

L'adaptation du paramètre de quantification par macrobloc peut être réalisée via la définition de ΔQP ajoutés au QP de l'image, on parle alors de quantification adaptative. Ainsi le codec x264 propose une quantification adaptative

dépendante de l'activité spatiale des macroblocs de l'image mesurée par la variance. Les auteurs de [118] et [119] utilisent des masquages spatio-temporels pour adapter perceptivement le QP.

Nos observations sur l'application du filtre perceptuel que nous proposons en CBR nous poussent à choisir une technique d'adaptation directe du pas de quantification en fonction du modèle JND de Yang. Nous concentrons notre intérêt sur la quantification adaptative proposée par x264 qui est l'amélioration la plus importante en termes de qualité proposée par ce codec [120], ainsi que sur l'adaptation du pas de quantification proposée par les auteurs de [84] qui utilisent également le modèle JND de Yang et al. Nous proposons dans la suite de ce chapitre une adaptation de la méthode proposée par [84] au cas de la quantification adaptative x264 afin de réduire l'effet de Ringing à bas et moyen débit. Dans nos travaux nous nous comparons à la meilleure qualité proposée par l'encodeur x264 à savoir l'encodage en CBR avec quantification adaptative, de plus nous nous concentrons sur les formats HD.

4. 4. Quantification Adaptative perceptuelle

On désigne par quantification adaptative le mécanisme qui consiste à faire varier le QP par macrobloc au sein d'une image pour s'adapter au contenu local d'une image. La quantification adaptative proposée par x264 appelée VAQ (Variance Adaptive Quantization), est basée sur une mesure de variance des macroblocs, l'idée étant de concentrer le débit sur les macroblocs à faible activité spatiale. Nous proposons ici une solution pour améliorer la quantification adaptative VAQ à bas et moyen débit, pour cela nous nous appuyons sur l'adaptation du pas de quantification en fonction du modèle JND de Yang proposé par les auteurs de [84]. Dans cette partie nous commençons par présenter la solution des auteurs de [84] pour ensuite présenter la quantification adaptative du codec x264.

4. 4. 1. L'adaptation adaptative guidée par le JND de Yang

Le modèle JND proposé par Yang et al. considère le masquage en luminance, en texture et temporel. Les auteurs de [84] proposent d'améliorer la précision du modèle en tenant compte du point d'attention (ou point de foveation) des spectateurs. Les auteurs fixent le point d'attention au centre de l'image, le modèle proposé considère donc que le SVH est plus sensible aux dégradations intervenant sur les pixels proches au centre de l'image. Ainsi, le seuil JND des pixels proches du centre de l'image est diminué et inversement. Le JND modifié est appelé FJND (Foveated JND).

Les auteurs proposent de contrôler le paramètre de quantification par macrobloc en fonction du modèle FJND qu'ils définissent, dans l'idée de quantifier plus sévèrement les macroblocs pouvant tolérer plus de distorsions et d'ainsi sauver du budget binaire qui sera par la suite réalloué aux macroblocs ayant un seuil JND plus faible.

De plus les auteurs proposent de pondérer la mesure de distorsion utilisée dans la fonction de calcul du coût débit-distorsion par le modèle FJND, afin de minimiser la distorsion perçue dans les images de la séquence. Nous ne traiterons pas cette partie et nous concentrons sur l'adaptation du paramètre de quantification par macrobloc.

Le paramètre de quantification perceptuel Q_i pour le i^{ème} macrobloc est obtenu en adaptant le paramètre de quantification Q_r choisi par le contrôle de débit pour l'image, en fonction d'un poids perceptuel w_i.

$$Q_i = \sqrt{w_i} \times Q_r$$ ***Équation 4.4.*** *QP perceptuel* ***[84]***

Le poids perceptuel w_i du i^{ème} macrobloc dépend de son seuil FJND. Les auteurs définissent les poids perceptuels à l'aide d'une fonction sigmoïde dépendant du seuil FJND s_i du i^{ème} macrobloc et du seuil moyen dans l'image $\bar{s}$ comme suit :

$$w_i = a + b\,\frac{1 + m\,exp\left(-c\,\frac{s_i - \bar{s}}{\bar{s}}\right)}{1 + n\,exp\left(-c\,\frac{s_i - \bar{s}}{\bar{s}}\right)}$$

Équation 4.5. *poids perceptuel* ***[84]***

Avec a, b, c, n et m des paramètres de la fonction sigmoïde fixés de manière empirique à a=0.7, b=0.6, c=4, n=1, m=0.

La Figure 4.7 représente l'évolution des poids $\sqrt{w_i}$ en fonction de la différence entre le seuil FJND moyen s_i du i^{ème} macrobloc et du FJND moyen calculé sur l'ensemble de l'image $\bar{s}$.

- Lorsque le seuil FJND d'un macrobloc est égal à la valeur moyenne dans l'image, alors le poids perceptuel est égal à 1, et le paramètre de quantification Q_i du macrobloc est égal au QP choisi par le contrôle de débit pour l'image.
- Lorsque le seuil FJND d'un macrobloc est supérieur à la valeur moyenne dans l'image, le macrobloc peut abriter plus de distorsions et son QP est de fait augmenté.
- Inversement, lorsque le seuil FJND d'un macrobloc est inférieur à la valeur moyenne dans l'image, le macrobloc est sensible aux dégradations et son QP est diminué pour préserver son contenu.

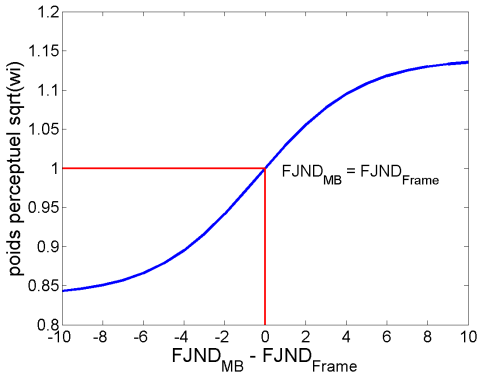


Figure 4.7. Evolution des poids perceptuels en fonction du FJND

Afin de valider la solution proposée, les auteurs ont réalisés des tests subjectifs suivant la méthode SDSCE (simultaneous double stimulus for continuous evaluation) [95] avec onze observateurs. Dans cette méthode, deux séquences sont jouées en même temps. La vidéo de gauche est la référence et la vidéo de droite est celle à noter. Les auteurs ont choisi d'utiliser l'échelle comparative à 7 niveaux que nous avons également utilisés pour nos tests. Le PSNR et la variation de débit entre les versions encodées avec et sans l'adaptation du QP proposé, sont également calculés.

-3 – Beaucoup moins bon
-2 – Moins bon
-1 – Légèrement moins bon
0 – Identique
1 – Légèrement mieux
2 – Mieux
3 – Beaucoup mieux

Figure 4.8. Echelle de notation

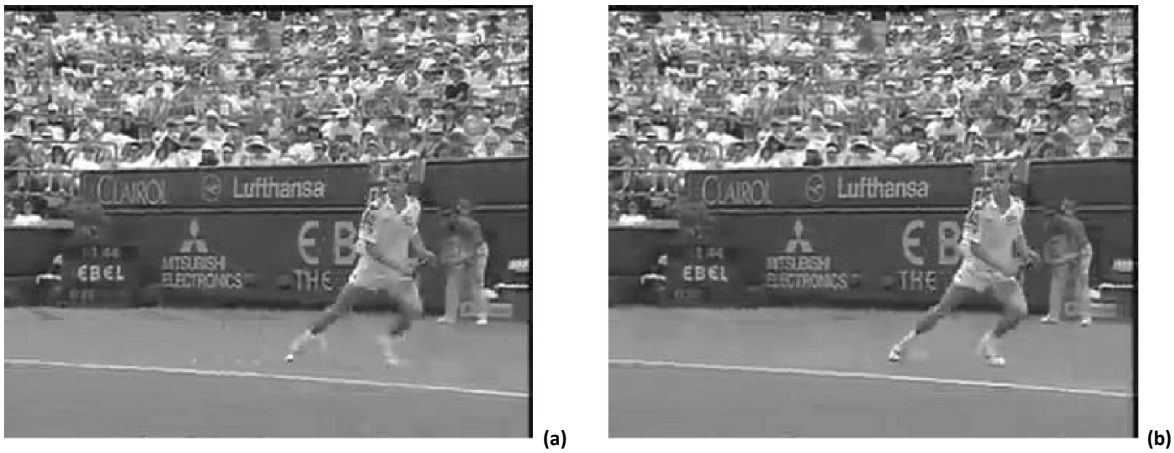
Les tests sont réalisés sur cinq séquences vidéo au format CIF à 30 images par seconde. Les séquences *Akiyo*, *Stefan*, *Football*, *Bus*, *Flower* sont respectivement encodées à 50 kb/s, 300 kb/s, 500 kb/s, 300 kb/s. La solution proposée est implémentée dans le codec de référence H.264/AVC KTA [121], configuré en GOP IPPP (seule la première image de la séquence est une image Intra), avec un codage entropique CABAC, une estimation de mouvement UMH avec zone de recherche de ±32 pixels.

Les résultats sont regroupés dans le Tableau 4.1. Comme nous l'avons dit précédemment, il faut noter que ces résultats intègrent à la fois l'adaptation du QP présentée au paragraphe précédent et la modification du module RDO. Les variations de débit par rapport au débit cible sont très faibles et ne sont pas influencées par la solution proposée. Pour quatre séquences sur cinq, la solution proposée réduit le PSNR comparativement à l'encodage KTA. Cependant, les notes subjectives sont toujours positives ce qui indique que la séquence encodée avec la solution proposée est préférée. Dans le cas le plus favorable (séquence *Flower*), la séquence encodée avec la solution proposée obtient une

note CMOS de 1, elle est donc jugée légèrement meilleure que la séquence encodée avec le codec KTA. La Figure 4.9 présente l'amélioration apportée par la solution proposée pour une image de la séquence Stephan encodée à 300 Kbit/s.

Test Sequence	Joint Model		FJND Method		ΔPSNR (dB)	Mean Comparison Scale
	ΔR	PSNR (dB)	ΔR	PSNR (dB)		
Akiyo	0.2%	37.85	0.2%	37.81	−0.04	0.41
Stefan	0.1%	28.48	0.1%	28.23	−0.25	0.72
Football	0.1%	27.88	0.1%	27.63	−0.25	0.75
Bus	0.1%	27.88	0.1%	27.06	−0.82	0.28
Flower	0.2%	25.24	0.2%	25.28	0.04	1.00

Tableau 4.1. Résultats de PSNR, variation de débit et tests subjectifs pour l'adaptation du QP en fonction du FJND
ΔR : Variation de débit par rapport au débit cible



(a) **(b)**

Figure 4.9. Comparaison visuelle de la solution d' l'adaptation du QP en fonction du FJND
Illustration issue de la publication [84]
Images décodées de la séquence Stefan encodée à 300Kbit/s
(a) Encodeur JM - (b) Solution proposée

Cette solution a été testée sur des séquences de faible résolution. Le paramètre de quantification est adapté au sein du cœur de codage de manière assez fine (variation de 0.85 à 1.15). Nous proposons de nous baser sur cette solution pour proposer une quantification adaptative permettant des variations plus importantes au sein d'une image afin de réduire l'effet de Ringing (4. 4. 2. 3.) sur des séquences HD.

4. 4. 2. La quantification adaptative x264

Le codec x264 est reconnu comme l'une des meilleures implémentations de la norme H.264/AVC [28]. Bien qu'il soit plus connu pour son utilisation grand public de transcodage fichier à fichier, le codec x264 est également utilisé dans le monde industriel comme point de comparaison pour le développement d'encodeurs professionnels.

La quantification adaptative définie par x264, connue sous le nom de VAQ fait figure de référence en matière d'allocation binaire pour sa simplicité et son efficacité. Le module de quantification adaptative situé à l'entrée du codeur, analyse l'image courante et définit une carte de ΔQPs par macrobloc qui sont ensuite ajoutés au QP global de l'image choisi par le contrôle de débit. L'intelligence de l'algorithme se trouve dans la fonction d'attribution des ΔQP qui dépend de la variance de chaque macrobloc 16x16. La fonction définissant les ΔQP en fonction de la variance de chaque macrobloc est basée sur l'observation suivante : La perte d'information amenée par l'application d'un pas de quantification dépend du contenu fréquentiel d'un bloc de pixels.

Cette idée est illustrée par la Figure 4.10 : pour deux types de contenu, un contour et une zone homogène, on observe l'effet de la quantification H264 à QP30 sur le contenu fréquentiel et spatial d'un bloc. Les cartes b et c présentent respectivement l'énergie des coefficients fréquentiels DCT et quantifiés. Pour aider la visualisation du contenu haute fréquence, le coefficient DC est mis à zéro. Pour le bloc de contour, la précision des coefficients fréquentiels est réduite et les coefficients de plus faible énergie sont perdus. La netteté du contour dans le bloc décodé est ainsi détériorée, mais le contenu est toujours reconnaissable. Pour le bloc contenant peu d'information, l'intégralité du contenu haute fréquence est perdu et le bloc décodé est par conséquent totalement homogène. Ainsi pour un même niveau de quantification, la sensation de perte de contenu est plus importante pour un bloc comportant peu d'activité spatiale.

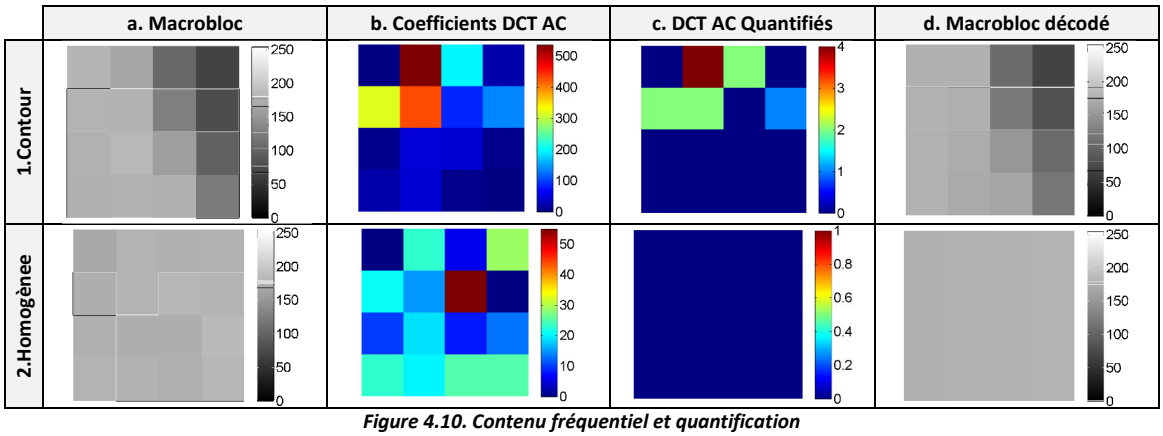


Figure 4.10. Contenu fréquentiel et quantification

A partir de cette observation, la quantification adaptative x264 cherche à préserver les informations dans les zones contenant peu de contenu haute fréquence, et inversement se permet de réduire le contenu haute fréquence des macroblocs de forte énergie. La fonction définissant les ΔQPs est donnée comme suit :

$$\Delta QP_{QA} = strength * \ log_2(max[variance, 1]) - (14.427 + 2 * (BIT\_DEPTH - 8))$$ ***Équation 4.6.*** *Quantification Adaptative x264*

Avec

- BIT_DEPTH = 8, profondeur de codage
- Strength = Force de quantifictaion, réglable, par défaut 1.0397

La Figure 4.11 présente l'évolution des ΔQPs en fonction de la variance d'un macrobloc. La quantification adaptative autorise un ΔQP maximum de -15 pour un macrobloc totalement homogène. Les figures ci-dessous considèrent une variance maximum correspondant à un macrobloc composé uniquement de pixels à 0 et 255 à nombre égal. Pour ce macrobloc particulièrement texturé, la quantification adaptative augmente le QP de l'image de +7.87

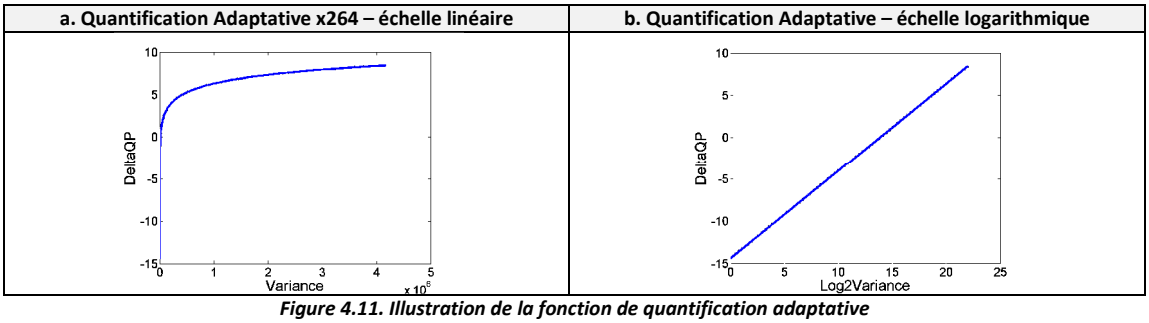


Figure 4.11. Illustration de la fonction de quantification adaptative

De par sa définition, la carte de quantification calculée par le module de quantification adaptative va préserver les zones les plus homogènes et gagner du budget binaire dans les zones contenant le plus d'activité spatiale.

4. 4. 2. 1. Analyse de l'effet de la quantification adaptative sur l'attribution des QPs

Le module de quantification adaptative calcule des ΔQPs en précision flottante en tout début d'encodage, deux exemples de cartes de ΔQPs sont données par la Figure 4.12 (b) pour la séquence *ParkJoy* et (e) pour la séquence Soccer au format 1280x720 50p. On observe une augmentation du paramètre de quantification pour les macroblocs fortement texturés (feuillage dans *ParkJoy* et contours dans Soccer) et une réduction dans les zones à faibles activité spatiale (banc d'herbe dans *ParkJoy* et arrière-plan et maillot des joueurs dans Soccer).

La précision de ces cartes de quantification adaptative est diminuée durant l'encodage pour deux raisons : D'une part le paramètre de quantification QP de chaque macrobloc, obtenu par ajout du ΔQP du macrobloc au QP estimé pour l'image, est une valeur entière pouvant variée de 0 à 51. D'autre part, le contrôle de débit agit sur le paramètre de quantification à chaque ligne de macrobloc pour contraindre le débit de sortie. Les cartes (c) et (f) présentent les différences de QP réellement appliqués avec l'action du contrôle de débit par rapport à un encodage x264 sans quantification adaptative.

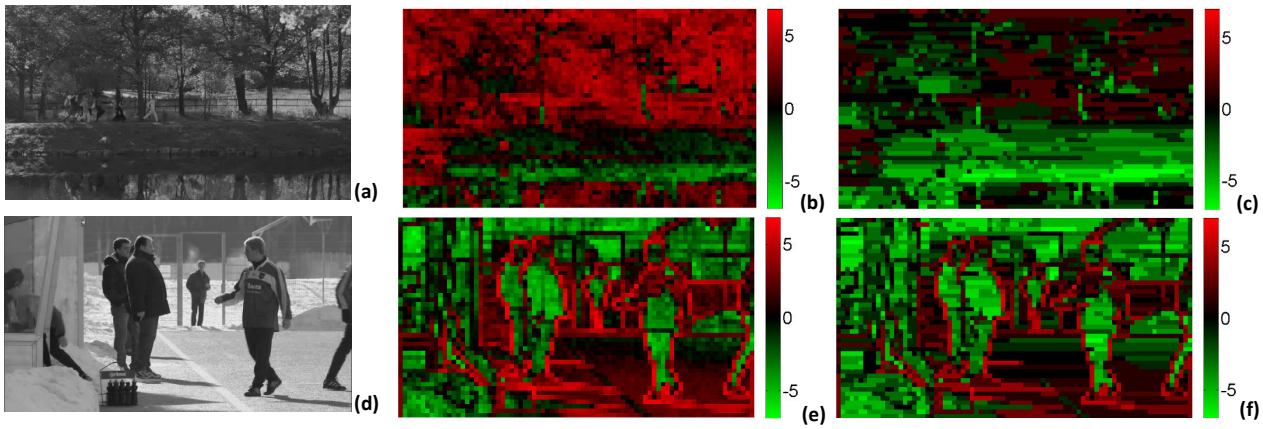


Figure 4.12. Effet de la quantification adaptative sur l'attribution des QP par macrobloc
(a) Image originale de la séquence ParkJoy – (b) ΔQP calculés par la quantification adaptative pour l'image ParkJoy – (c) ΔQP réellement appliqués après l'action du contrôle de débit pour la séquence ParkJoy - (d) Image originale de la séquence Soccer – (e) ΔQP calculés par la quantification adaptative pour l'image Soccer – (f) ΔQP réellement appliqués après l'action du contrôle de débit pour la séquence Soccer

4. 4. 2. 2. Analyse de l'effet de la quantification adaptative sur la qualité

La Figure 4.13, Figure 4.14 et Figure 4.15 présentent l'action de la quantification adaptative sur la qualité des séquences par rapport à x264 sans quantification adaptative. Ces figures comportent l'image originale (a), la carte de différences de QP réellement appliquées par rapport à un encodage x264 sans quantification adaptative telles que présentées au paragraphe précédent, et des agrandissements de zones caractéristiques. Les encodages sont réalisés en GOP Intra, avec des macroblocs 16x16 sans découpe 4x4, à débit constant correspondant à un débit Inter de 3M. Pour connaître le débit Intra correspondant à 3M en GOP Inter, le poids binaire des images I d'un GOP classique IBBP(12) est moyenné.

En concentrant le budget binaire sur les macroblocs à faible activité spatiale, la quantification adaptative apporte un gain de qualité important pour des séquences vidéo du type *ParkJoy,* composée de zones à forte activité spatiale (feuillage) et de zones à faible activité spatiale (le banc d'herbe et l'eau). Les agrandissements de la Figure 4.13 montent une réduction visible de l'effet de bloc.

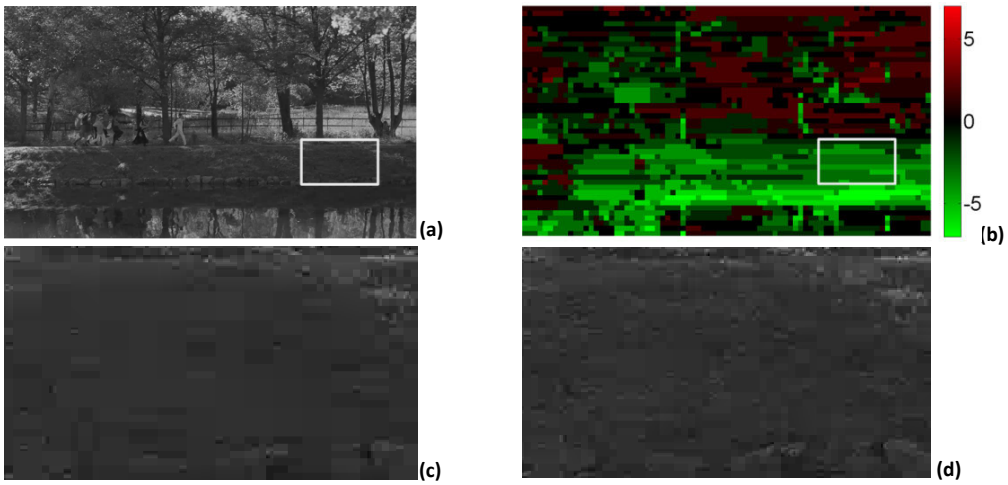


Figure 4.13. Effet de la quantification adaptative x264 pour une image de la séquence ParkJoy 1280x720 Encodage Intra (modes 16x16 uniquement) à 22Mbit/s
(a) Image originale - (b) Carte de Delta QP utilisés à l'encodeur par la quantification adaptative x264– (c) Zoom dans l'image encodée x264 – (d) Zoom dans l'image encodée x264 + AQ

Les macroblocs contenant des contours ont une variance spécialement élevée et sont par conséquent considérés comme très peu sensibles aux dégradations amenées par la quantification, par conséquent, leur QP est spécialement élevé. A faible débit, la réduction de budget binaire alloué aux contours se traduit rapidement par l'apparition de « Ringing » qui détériore sévèrement la qualité globale de l'image. Deux exemples de l'effet de « Ringing » amené par la quantification adaptative sont donnés par la Figure 4.14 et la Figure 4.15.

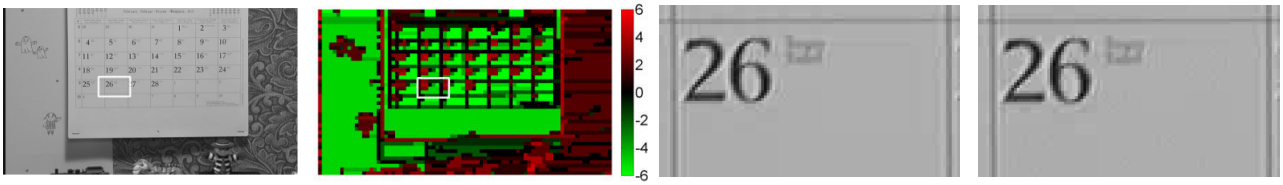


Figure 4.14. Illustration de l'effet de Ringing apporté par la quantification adaptative x264 sur une image de la séquence Mobile & Calendar
(a) Image originale - (b) ΔQP entre x264 sans et avec quantification adaptative - x264 sans (c) et avec quantification adaptative (d), à 32Mbit/s en GOP Intra (Qualité Intra équivalente à 3Mbit/s en GOP Inter) avec uniquement les partitions 16x16

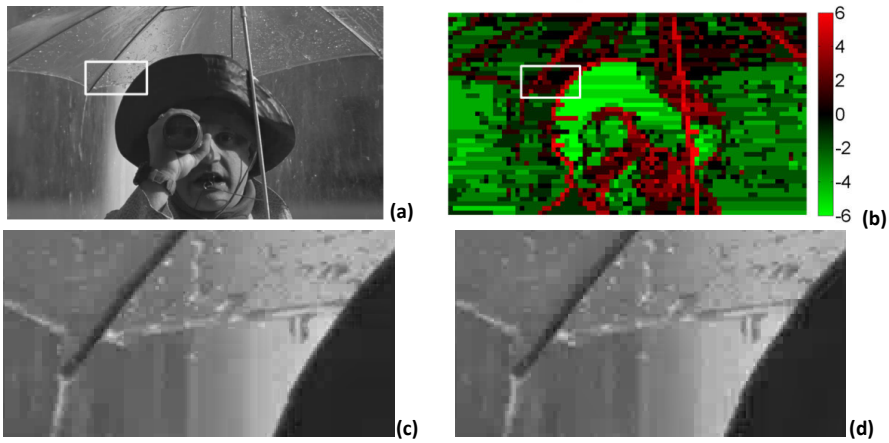


Figure 4.15. Illustration de l'effet de Ringing apporté par la quantification adaptative x264 sur une image de la séquence Binocular
(a) Image originale - (b) ΔQP entre x264 et x264 avec quantification adaptative - (c) x264 à 32Mbit/s en GOP Intra (Qualité Intra équivalente à 3Mbit/s en GOP Inter) avec uniquement les partitions 16x16 - (d) x264 avec quantification adaptative

4. 4. 2. 3. **L'effet de ringing**

L'effet de « Ringing » est un artefact caractéristique de la compression vidéo apparaissant sur les contours des images lorsque la réduction de débit devient plus importante. La Figure 4.16 présente un exemple de l'effet de Ringing sur une image Intra encodée à QP40 avec uniquement des modes de prédictions 16x16 (b), ainsi qu'avec tous les modes disponibles (d). La compression provoque l'apparition d'activité spatiale initialement inexistante autour d'un contour. Comme le montre le partitionnement (e), lorsque tous les modes Intra sont autorisés, les modes 4x4 sont souvent choisis pour coder plus finement l'information de contour, réduisant ainsi l'effet de Ringing. Pour expliquer le mécanisme d'apparition de l'effet de « Ringing », nous prenons l'exemple du macrobloc 16x16 particulier contenant un contour représenté en rouge dans la Figure 4.16.

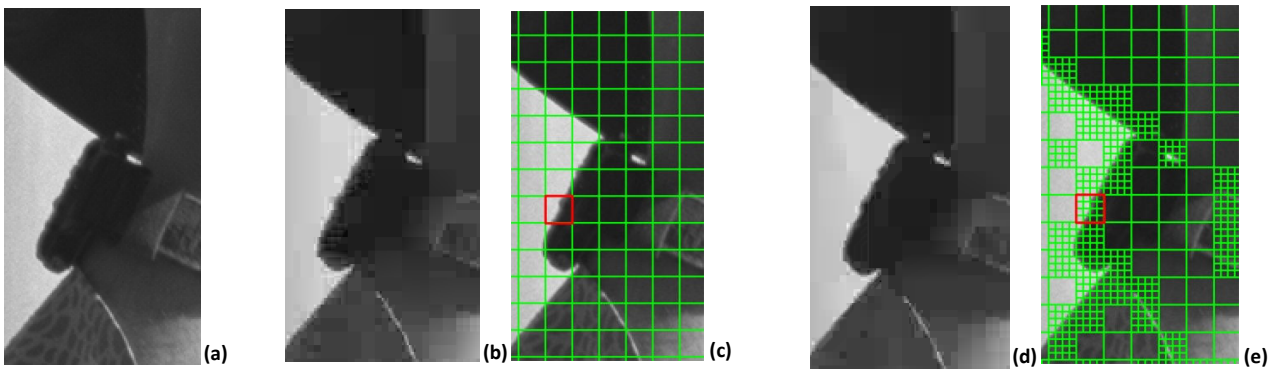


Figure 4.16. Illustration de l'artefact de Ringing en fonction des partitions Intra autorisées
(a) Agrandissement de d'une image de la séquence Binocular – (b) Agrandissement de l'image encodée à 9Mbit/s en GOP Intra avec uniquement des modes 16x16 – (c) Partitions Intra 16x16 – (d) Agrandissement de l'image encodée à 9Mbit/s en GOP Intra avec les modes intra 4x4 et 16x16 – (c) Partitions Intra 4x4 et 16x16 utilisées

La Figure 4.17 présente les étapes d'encodage pour ce macrobloc particulier avec une prédiction 16x16 (ligne 1) et une prédiction 4x4 (ligne 2). En mode 16x16, la meilleure prédiction Intra choisie est le mode vertical (1.b), la prédiction étant peu précise, le contenu du macrobloc se retrouve dans l'erreur de prédiction à encoder (1.c). H264/AVC utilise une DCT et une quantification 4x4 quel que soit le partitionnement utilisé pour l'étape de prédiction. Les cartes (1.d) et (1.e) présentent respectivement l'énergie des coefficients AC DCT et leur version quantifiée pour les 16 blocs 4x4 du macrobloc. Pour plus de visibilité, les coefficients DC ne sont pas représentés. Après quantification, certains blocs ne conservent aucun coefficient AC et sont reconstruits uniquement à partir de la prédiction, les autres conservent entre un et trois coefficients AC. Le manque de précision dans la reconstruction des blocs 4x4 situés en bordure de contour provoque une augmentation de l'activité spatiale dans ces zones et provoque l'impression de « débordement » du contour.

En mode 4x4, la prédiction est plus précise et seuls les pixels de contour se retrouvent dans le résidu à encoder (2.b). Les coefficients AC DCT sont par conséquent moins énergétiques (2.c) et seuls trois blocs conservent un à deux coefficients AC après quantification (2.d). A même niveau de quantification, le macrobloc décodé est plus fidèle au macrobloc original avec des modes de prédiction 4x4, mais l'effet de Ringing persiste.

En résumé, la prédiction Intra 4x4 est logiquement moins sensible à l'artefact de Ringing, que la prédiction Intra 16x16, cependant les modes Intra 4x4 peuvent également créer de l'activité spatiale aux contours à faible débit et les modes Intra 16x16 peuvent être choisis par le module RDO pour représenter un macrobloc représentant un contour. Il faut toutefois noter ici qu'en encodage Inter, un macrobloc peut être découpé suivant un plus grand nombre de possibilités (16x16, 8x16, 16x8, 8x8, 8x4, 4x8 et 4x4), ceci sera abordé au paragraphe 4. 5. 5.

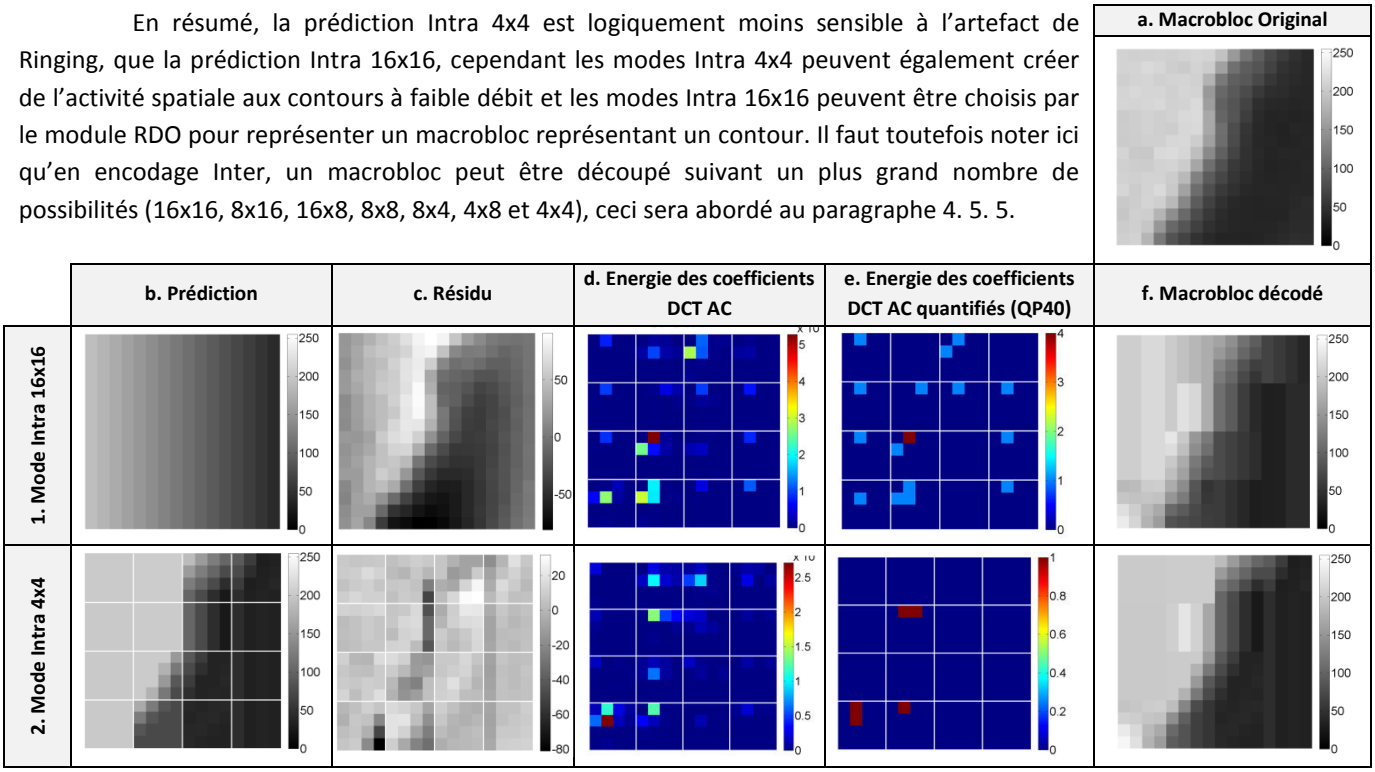


Figure 4.17. Causes de l'apparition de l'effet de Ringing

En liant le niveau de quantification à l'activité spatiale des macroblocs, la quantification adaptative définie par x264 permet une amélioration globale de la qualité ressentie, la principale faiblesse de l'algorithme réside dans la gestion des contours dont la dégradation est spécialement gênante à bas débit.

4. 4. 3. La quantification adaptative contrôlée par le modèle JND

Afin de réduire l'effet de Ringing amené par la quantification adaptative à bas débit, nous proposons de définir une quantification adaptative contrôlée par le modèle JND défini par Yang et al. Dans une première approche, nous avons considéré uniquement le masquage en texture pour définir une carte de ΔQP. En effet, le masquage en texture proposé par Yang et al. est proche de l'analyse de variance utilisée par la quantification adaptative x264, la différence principale est qu'il indique les contours comme du contenu très sensible aux dégradations.

4. 4. 3. 1. Le masquage en texture de Yang

On propose de considérer un modèle de masquage en texture pour améliorer la quantification adaptative x264. Le masquage en texture défini par Yang traduit la sensibilité du SVH (Système Visuel Humain) à chaque pixel d'une image en fonction de l'activité spatiale de son voisinage. Il définit un seuil de visibilité des distorsions pour chaque pixel.

Les zones homogènes et contours sont considérés comme les parties les plus sensibles d'une image, dans lesquelles une dégradation sera perçue comme hautement nuisible à la qualité globale d'une image. Les zones texturées peuvent abriter de plus fortes distorsions sans que celles-ci ne soient perceptibles.

Le masquage en texture défini par Yang est basé sur un calcul de gradient, le principe est par conséquent proche du modèle de variance utilisé par la quantification adaptative x264. Cependant il existe deux différences fondamentales entre le modèle de variance et de masquage en texture :

- Le modèle de variance calcule la variance de chaque macrobloc d'une image tandis que le modèle de masquage en texture attribue une valeur par pixel. Afin de comparer les deux modèles par macrobloc d'une image, on utilise dans un premier temps le JND moyen d'un macrobloc, nous verrons par la suite comment mieux exploiter la précision du modèle JND.

- Le modèle de variance ne différencie pas les différents types de zones à forte activité spatiale. Ainsi un macrobloc contenant un contour sera considéré comme faiblement sensible aux dégradations. Au contraire, le modèle de masquage en texture utilise une détection de contour pour différencier les pixels de contours des pixels de texture, et considère ces pixels comme fortement sensibles aux dégradations. La Figure 4.18 illustre cette différence de comportement.

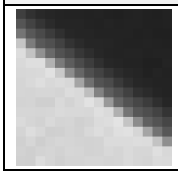
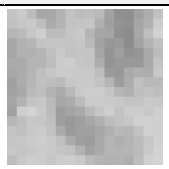
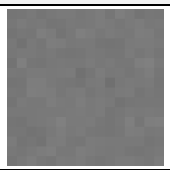
Macrobloc Contour		Macrobloc Texture		Macrobloc Homogène	
	Variance = 29,45 JNDmoyen = 3,86		Variance =1,35 JNDmoyen = 20,01		Variance = 0,04 JNDmoyen = 3,52

Figure 4.18. Comparaison de la variance et du JND moyen

En exploitant cette deuxième observation, on compte définir une quantification adaptative contrôlée par la carte de JND en texture moyen par macrobloc et ainsi allier l'efficacité de la quantification adaptative à la variance dans les zones texturées, à la préservation des contours.

4. 4. 3. 2. La quantification adaptative contrôlée par le masque en texture

En définissant une quantification adaptative contrôlée par le modèle de masquage en texture, on compte concentrer le débit sur les zones à faible activité spatiale ainsi que sur les contours et réduire le budget binaire des zones à forte activité spatiale. A la manière des auteurs de [84], nous utilisons une fonction tangente hyperbolique dépendant de la différence entre le masquage en texture moyen d'un macrobloc et de l'image. Ainsi, le ΔQP d'un macrobloc augmente lorsque son JND texture moyen est supérieur au JND texture moyen de l'image et inversement (Cf. Figure 4.19). La relation donnant la valeur de ΔQP à appliquer, calculée en fonction du JND, est définie de façon empirique par :

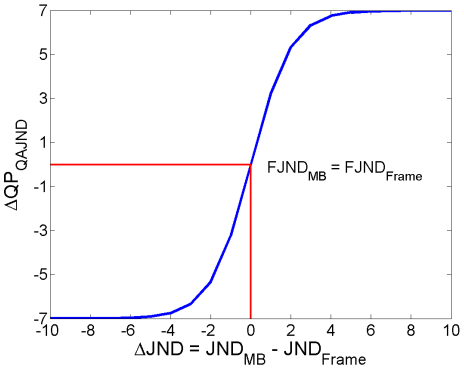


Figure 4.19. Quantification adaptative contrôlée par le masquage en texture

$$\Delta QP_{QAJND} = \Delta QP_{max} * tanh(c * \Delta JND)$$

$$\Delta JND = (JND_{MB} - JND_{Frame})$$

Équation 4.7. Quantification Adaptative x264

Avec

- ΔQPmax : ΔQP maximum autorisé fixé expérimentalement à 7.
- tanh(.) : Tangent hyperbolique : $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$
- C : constante fixée à 0.5 expérimentalement, permettant de régler la décroissance de la tangente hyperbolique

Le paramètre ΔQPmax définit la plage de variation maximale autorisée. Par expérimentation ce paramètre est fixé pour autoriser des ΔQPs de +/- 7. Le paramètre c contrôle la pente de la tangente hyperbolique, il est fixé à 0.5 pour décroître suffisamment rapidement pour préserver les macroblocs à faible activité spatiale. La Figure 4.20 présente une comparaison de la quantification adaptative contrôlée par la variance des macroblocs et de notre solution contrôlée par le masquage en texture moyen par macrobloc, pour une image de la séquence *Mobile & Calendar*. L'échelle des ΔQPs attribués à l'image par les deux modèles est équivalente (entre +/- 6). La différence principale entre les deux modèles réside dans les contours détectés par le modèle de texture (ici les contours du calendrier et les chiffres).

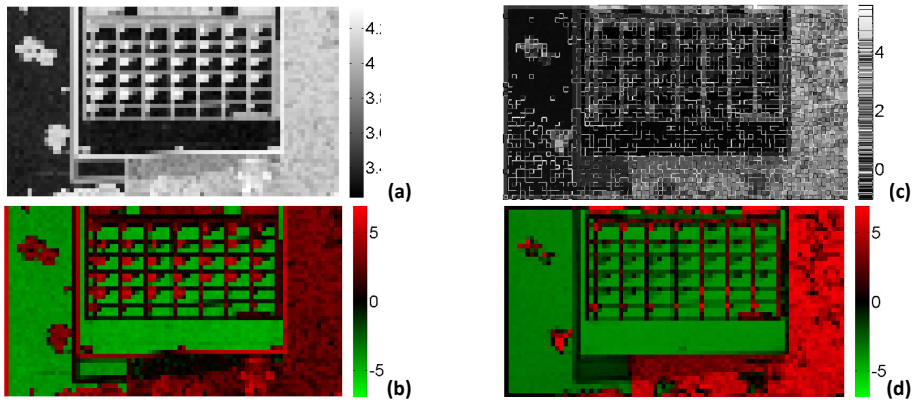


Figure 4.20. Comparaison du modèle de variance et de JND texture
(a) Modèle de contrôle de la quantification adaptative : Logarithme de la variance de chaque macrobloc – (b) Carte de ΔQP définis par la quantification adaptative – (c) Modèle de contrôle de la quantification adaptative proposée : Masquage en texture moyen par macrobloc – (d) Carte de ΔQP définis par la quantification adaptative proposée

4. 5. Résultats

Dans ce dernier paragraphe, nous présentons les conditions de test ainsi que les résultats obtenus avec notre solution de quantification adaptative contrôlée par le masquage en texture sur des séquences HD. Nous proposons une analyse détaillée de l'effet de notre solution sur l'encodage Intra image, puis nous vérifions son l'intérêt sur l'encodage Inter.

4. 5. 1. Choix du profil d'encodage x264 CBR

Les résultats de nos travaux diffèrent logiquement en fonction des outils d'encodage utilisés. Pour choisir le paramétrage du codec x264 utilisé pour nos tests, nous avons cherché à approcher les performances des encodeurs AQILIM de Digigram. Nous rappelons que le codec de Rovi (Mainconcept) utilisé dans les encodeurs AQILIM propose un paramètre pour régler la qualité de l'encodage appelé « Video Performance » variant de 0 à 15, 0 correspondant à une qualité médiocre et 15 à une très bonne qualité. Les clients de Digigram jugent la qualité des encodeurs AQILIM bonne à partir du paramètre « Video Performance » égal à 12, c'est donc notre qualité de référence. Nous avons défini un profil x264 spécifique donnant des performances comparables au profil Video Performance 12 de Rovi (Mainconcept) en termes de nombre d'images traitées par seconde (fps) et PSNR (Tableau 4.2). Le test est réalisé sur trois séquences différentes de résolution couramment utilisée. On peut noter que le PSNR de la séquence CrowdRun est faible car cette séquence est particulièrement complexe à encoder.

		FPS	Bitrate [Kb/s]	ΔBitrate [Kb/s]	|Δbitrate| [%]	PSNR
Pedestrian 1920x1080 25p 8Mbit/s	MC - VP 12	16,60	8356,49	356,49	4,46	42,362
	x264 – Slow custom	13,91	7042,88	-957,12	-11,96	42,242
CrowdRun 1280x720 50p 6Mbit/s	MC - VP 12	36,20	6665,59	665,59	11,09	28,010
	x264 – Slow custom	28,44	5959,37	-40,63	-0,68	28,727
Stockholm 768x432 25p 2Mbit/s	MC - VP 12	77,44	2183,98	183,98	9,20	39,519
	x264 – Slow custom	99,55	1702,80	-297,20	-14,86	39,343

Tableau 4.2. Comparaison des encodages Rovi (Mainconcept) (MC) (Video performance 12) et x264 (slow custom)
ΔBitrate : variation de débit par rapport au débit cible

Le codec x264 propose dix profils nommés en fonction de la latence d'encodage : *ultrafast, superfast, veryfast, faster, fast, medium, slow, slower, veryslow, placebo*. Les performances de vitesse et la qualité étant étroitement liées, le profil *ultrafast* donne une qualité médiocre, alors que le profil *veryslow* donne une bonne qualité. Le profil x264 que nous avons sélectionné pour nos tests est basé sur le profil *slow* dans lequel nous avons fait le choix de désactiver le filtre de Deblocking ainsi que les trois paramètres perceptifs proposés par x264 présentés comme expérimentaux et décrits succinctement dans le Tableau 4.4. Les paramètres principaux du profile d'encodage x264 utilisé dans la suite de ce chapitre sont présentés dans le Tableau 4.3.

Paramètres H.264/AVC		Paramètres spécifiques x264	
Nombre d'images de référence	5	RDO appliqué aux étapes de	Modes intra, Estimation de mouvement, Découpes inter
Deblocking	Désactivé	Quantification par treillis	Désactivé
Codage entropique	CABAC	Psy RDO	Désactivé
Estimation de mouvement	UHM	Psy Treillis	Désactivé
Zone de recherche	16	Nombre d'images dans le lookahead	50
Interpolation	¼ de pixel		

Tableau 4.3. Paramètres du profil d'encodage utilisé pour les tests

Psy-rdo	Autorise le choix d'une prédiction donnant une image décodée plus éloignée de l'originale mais plus agréable à l'œil.
Psy -Trellis	Utilisation une quantification non-uniforme. Pour chaque coefficient fréquentiel d'un bloc 4x4, plusieurs pas de quantification autour du pas de quantification choisi pour le bloc. Choix du pas de quantification donnant le meilleur compromis qualité/débit en considérant des paramètres perceptuels.
MBTree	Préservation des macroblocs de référence en attribuant le QP d'un MB en fonction d'une estimation du nombre de fois où il sera utilisé comme référence pour une estimation de mouvement [31].

Tableau 4.4. Outils Perceptifs proposés par x264

4. 5. 2. Conditions de test

Nos tests sont réalisés sur huit séquences HD dont cinq au format 1280x720 50p et trois 1920x1080 à 25p. Les encodages sont réalisés avec les paramètres définis au paragraphe précédent.

Dans un premier temps pour simplifier notre analyse, nous nous concentrons uniquement sur le cas des images Intra, nous validerons par la suite nos résultats sur un GOP inter classique. Nous testons deux configurations d'encodage Intra, le cas où uniquement les modes de prédiction 16x16 sont utilisés et le cas utilisant tous les modes disponibles (4x4 et 16x16).

Pour se placer à des niveaux de qualité correspondant aux cas d'usage réels, nous avons calculé pour chaque séquence le débit moyen des images Intra dans deux GOP Inter (IBBP(12) et IBBP(33)), encodés à des débits classiquement utilisés par les clients de Digigram. Nos tests sont réalisés à faible et moyen débit respectivement 3 et 6 Mbits/sec pour le format 1280x720 50p et 4 et 8 Mbits/sec pour le format 1920x1080 25p. Les débits utilisés pour encoder les séquences en GOP Intra sont donnés dans le Tableau 4.5.

	Débit des images Intra dans un GOP Inter encodé à :	
1280x720 50p	**3 Mbit/s**	**6 Mbit/s**
ParkJoy	22 Mbit/s	40 Mbit/s
Soccer	16 Mbit/s	27 Mbit/s
Binocular	9 Mbit/s	17 Mbit/s
MobCal	32 Mbit/s	53 Mbit/s
LostPeople	17 Mbit/s	32 Mbit/s
1920x1080 25p	**4 Mbit/s**	**8 Mbit/s**
Pedestrian	17 Mbit/s	31 Mbit/s
SunFlower	29 Mbit/s	48 Mbit/s
Tractor	17 Mbit/s	31 Mbit/s

Tableau 4.5. Moyenne des débits des images Intra au sein de GOP Inter IBBP12 et IBBP33 encodés à bas et moyen débit.

4. 5. 3. Analyse des résultats moyens par séquence

Pour la mesure de résultats, nous utilisons deux métriques mesurant les distorsions globales des images, le PSNR et le SSIM, ainsi que la métrique proposée par les auteurs de [110] qui mesure l'effet de Ringing aux contours des images d'une séquence. Nous calculons les métriques sur la composante de luminance, pour toutes les images d'une séquence et donnons la valeur moyenne sur la séquence. Les résultats des trois métriques pour les quatre conditions de tests (bas/moyens débits et modes Intra 16x16/modes Intra 4x4 et 16x16), sont présentés par le Tableau 4.7, le Tableau 4.8, le Tableau 4.9 et le Tableau 4.10.

4. 5. 3. 1. **Métrique de Ringing**

La mesure de Ringing proposée par [110] est basée sur la métrique de flou des mêmes auteurs présentée au chapitre 3. La mesure de Ringing est réalisée par l'analyse horizontale des contours verticaux. La mesure de Ringing est effectuée de chaque côté d'un pixel de contour sur un support d'analyse (Fixed Ring width) dont la taille est définie égale à l'étalement de contour moyen de l'image calculé par la métrique de flou.

Prenons l'exemple de la mesure de Ringing à gauche du pixel de contour (Left Ring Measure, |P3'-P3|), elle est définie comme la différence entre la mesure d'étalement de contour (Left Edge width, |P3-P1|) et la taille du support d'analyse de Ringing (Fixed Ring width, |P3'-P1|). La même mesure est effectuée du côté droit du contour. La métrique de Ringing d'une image est obtenue en moyennant la note obtenue à chaque pixel de contour.

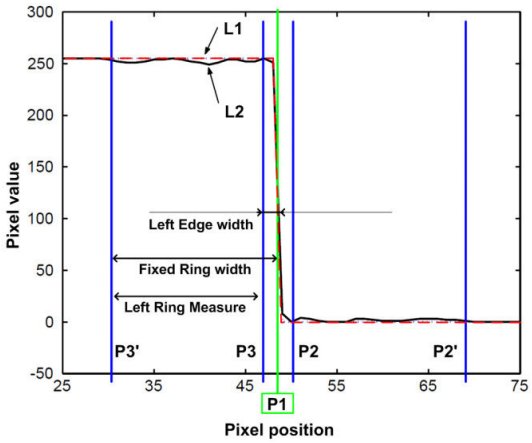


Figure 4.21. Métrique de Ringing Marziliano
Illustration issue de la publication [110]
L1 est une ligne de luminance dans l'image originale et L2 la même ligne dans l'image encodée en JPEG2000

La métrique de Ringing que nous venons de présenter et que nous utilisons pour mesurer nos résultats dans la suite de ce chapitre, trouve ses limites lorsque l'image contient peu de contours. Nous venons de voir que l'effet de Ringing est mesuré autour de chaque pixel de contour détecté par une analyse de Sobel. Or, le seuil au-dessus duquel un gradient est considéré comme un contour est calculé en fonction de la carte de gradient de l'image. Ainsi, comme le montre la Figure 4.22, lorsque l'image ne contient pas de contours francs (par exemple la séquence *ParkRun*), des pixels de texture sont retenus comme pixels de contours. La mesure d'étalement réalisée sur ces pixels ne mesure donc pas l'effet de Ringing. Pour ces raisons, les résultats de Ringing ne sont pas présentés pour les séquences *ParkJoy* et *LostPeople*. Pour la séquence *Mobile & Calendar* ((b) et (e)), le même problème intervient sur la partie droite de l'image correspondant à la tapisserie. Pour ne pas corrompre les résultats de Ringing sur cette séquence, la mesure est réalisée en excluant cette partie des images de la séquence.

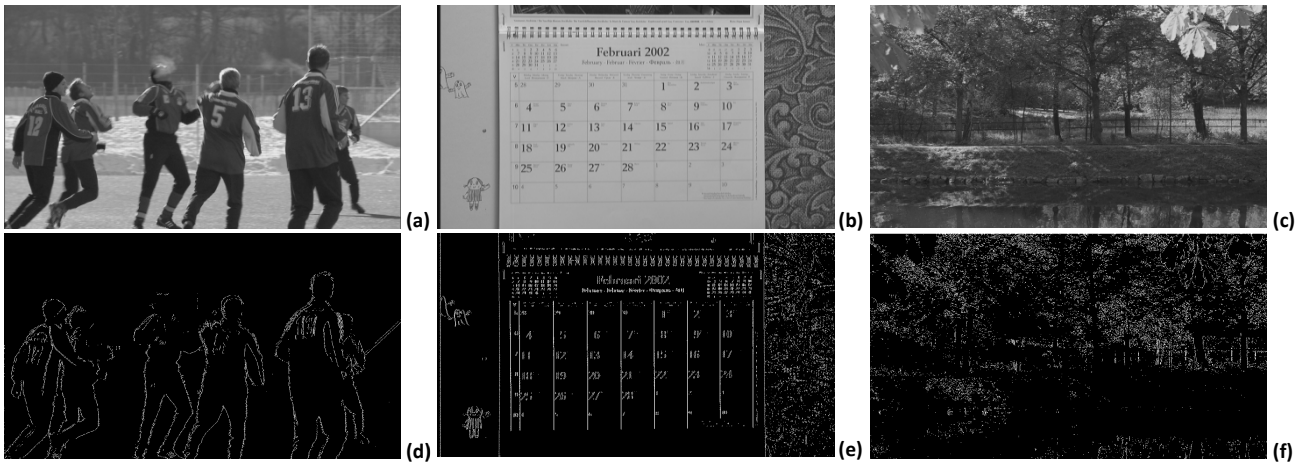


Figure 4.22. Détection de contours utilisés pour la mesure de Ringing
(a), (b), (c) : Images issues respectivement des séquences Soccer, Mobile & Calendar et ParkRun 1280x720p
(d), (e), (f) Carte de détection de contour par analyse de Sobel

4. 5. 3. 2. **Résultats moyens sur les quatre configurations d'encodage**

Avant tout, nous vérifions que la quantification adaptative que nous proposons ne perturbe pas le contrôle de débit. Le Tableau 4.6 présente les variations de débit amenées par la quantification adaptative x264 et notre solution par rapport à un encodage sans quantification adaptative, pour les huit séquences test au deux débits testés. Au maximum notre solution amène une variation de 1.12% sur l'ensemble de la séquence en encodage tout Intra.

		Prédiction Intra 16x16					**Toutes prédictions Intra**				
		x264 ss AQ	x264 + AQ		x264 + AQJND		x264 ss AQ	x264 + AQ		x264 + AQJND	
		[Mb/s]	[Mb/s]	Δ [%]	[Mb/s]	Δ [%]	[Mb/s]	[Mb/s]	Δ [%]	[Mb/s]	Δ [%]
Soccer	16M	16,20	16,19	**-0,02**	16,21	**0,10**	16,22	16,16	**-0,36**	16,18	**-0,19**
	27M	27,30	27,32	**0,09**	27,33	**0,10**	27,25	27,28	**0,13**	27,28	**0,11**
Binocular	9M	8,80	8,78	**-0,25**	8,86	**0,75**	8,91	8,83	**-0,88**	8,89	**-0,17**
	17M	15,68	15,60	**-0,46**	15,76	**0,54**	15,61	15,58	**-0,20**	15,78	**1,12**
Mobile Calendar	32M	31,57	31,55	**-0,06**	31,56	**-0,03**	30,81	30,80	**-0,01**	30,80	**-0,04**
	53M	52,27	52,22	**-0,10**	52,17	**-0,19**	51,01	50,93	**-0,15**	50,85	**-0,31**
Pedestrian	17M	33,43	33,42	**-0,03**	33,42	**-0,01**	25,06	25,06	**0,01**	25,07	**0,04**
	31M	60,93	60,94	**0,01**	60,95	**0,02**	45,69	45,69	**-0,01**	45,70	**0,01**
SunFlower	29M	59,60	59,62	**0,03**	59,57	**-0,04**	59,76	59,63	**-0,21**	59,61	**-0,25**
	48M	95,61	95,40	**-0,22**	95,35	**-0,27**	95,11	94,20	**-0,96**	95,36	**0,25**
Tractor	17M	33,68	33,53	**-0,45**	33,59	**-0,27**	33,74	33,64	**-0,31**	33,76	**0,04**
	37M	61,57	61,31	**-0,43**	61,48	**-0,15**	61,45	61,29	**-0,26**	61,53	**0,12**

Tableau 4.6. Variations de débit amenées par les quantifications adaptatives
(x264 ss AQ) Encodage x264 sans quantification adaptative – (x264 + AQ) Encodage x264 avec quantification adaptative – (x264 + AQJND) Encodage x264 avec quantification adaptative contrôlée par le JND

De manière générale, quelles que soient les séquences, les trois métriques évoluent rarement dans le même sens, ce qui complique l'analyse des résultats. La Figure 4.23 présente un exemple pour la séquence *Binocular* encodée à 9Mbit/s en GOP Intra. On observe que le PSNR et la mesure de Ringing classent les trois versions encodées dans le même ordre, la version x264 sans quantification adaptative est préférée, puis notre solution (QAJND) et enfin la quantification adaptative x264 (QA). Pour le SSIM, la quantification adaptative proposée par x264 donne une meilleure qualité que notre solution.

Afin d'apporter une autre analyse que celle des trois métriques citées ci-dessus, nous avons réalisé une analyse visuelle des séquences dont nous proposons des agrandissements au paragraphe 4. 5. 4.

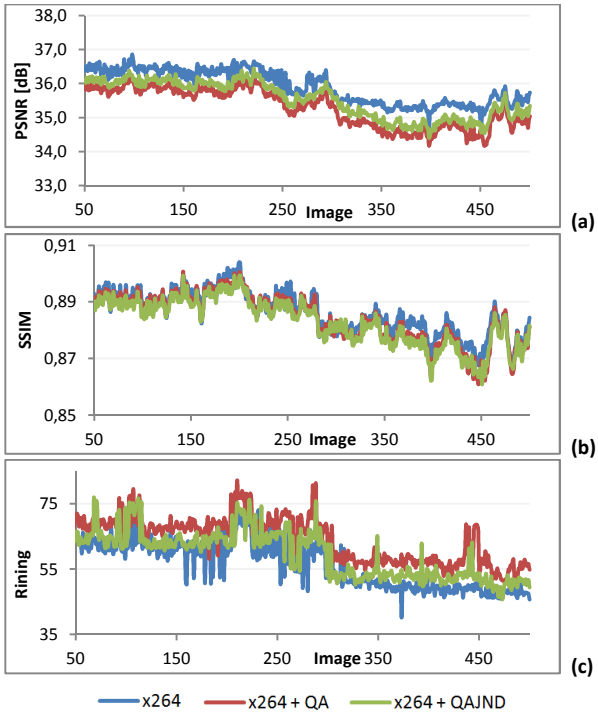


Figure 4.23. Courbes des métriques par images PSNR (a), SSIM (b) et Ringing (c) pour la séquence Binocular 1280x720 50p encodée en GOP Intra à 9 Mbit/s en autorisant tous les modes Intra

Le Tableau 4.7 au Tableau 4.10 présentent les résultats de SSIM, PSNR et de Ringing pour les trois séquences 720 50p et les trois séquences 1080 25p, encodées avec uniquement les modes intra 16x16 à bas débit (Tableau 4.7), à moyens débits (Tableau 4.8) et avec tous les modes intra disponibles à bas débit (Tableau 4.9) et moyens débits (Tableau 4.10).

- **PSNR**

Les versions encodées avec la quantification adaptative x264 ou notre solution ont systématiquement une note de PSNR plus basse que la version encodée sans quantification adaptative (avec un QP par image).

Dans le cas des séquences HD, notre proposition apporte moins de dégradations que la quantification adaptative x264 (colonne (f1), dans les quatre tableaux).

- **SSIM**

La quantification adaptative utilisée dans x264 est de meilleure qualité que l'encodage sans quantification adaptative, tandis que notre solution obtient toujours des notes plus basses. (colonne (f2), dans les quatre tableaux).

- **Ringing**

Logiquement, la présence de Ringing est moins importante à moyen débit qu'à bas débit (comparaison du Tableau 4.7 avec le Tableau 4.8 et du Tableau 4.9 avec le Tableau 4.10).

Etonnamment, à même débit, l'utilisation de tous les modes Intra disponibles augmente le Ringing d'après la métrique (comparaison du Tableau 4.7 avec le Tableau 4.9 et du Tableau 4.8 avec le Tableau 4.10).

En moyenne sur les six séquences HD, la quantification adaptative que nous proposons réduit l'effet de Ringing comparativement à la quantification x264 quels que soit les débits et les modes intra utilisés (colonne (f3) dans les quatre tableaux).

Quelles que soient les séquences et les conditions d'encodage, la métrique indique que la version x264 sans quantification adaptative donne toujours la meilleure qualité en terme de Ringing. Malgré tout, nous verrons par la suite que notre solution permet de réduire l'effet de Ringing ou du moins de préserver autant les contours que l'encodage x264 sans quantification adaptative, tout en préservant les zones homogènes comme le propose la quantification adaptative x264.

Modes Intra 16x16 **Bas Débits**			PSNR						SSIM						Ringing					
			x264	x264 avec QA		x264 avec QAJND			x264	x264 avec QA		x264 avec QAJND			x264	x264 avec QA		x264 avec QAJND		
			PSNR (a1)	PSNR (b1)	ΔP (c1) (b1)-(a1)	PSNR (d1)	ΔP (e1) (d1)-(a1)	ΔP (f1) (d1)-(b1)	SSIM (a2)	SSIM (b2)	ΔS (c2) (b2)-(a2)	SSIM (d2)	ΔS (e2) (d2)-(a2)	ΔS (f2) (d2)-(b2)	Ringing (a3)	Ringing (b3)	ΔR (c3) (b3)-(a3)	Ringing (d3)	ΔR (e3) (d3)-(a3)	ΔR (f3) (d3)-(b3)
720p	Soccer	16 Mb/s	36,62	36,03	-0,60	36,09	-0,53	0,06	0,924	0,926	0,002	0,923	-0,001	-0,003	19,202	22,454	3,252	22,312	3,110	-0,142
	Binocular	9 Mb/s	35,05	34,57	-0,48	34,73	-0,31	0,17	0,872	0,872	0,000	0,870	-0,003	-0,003	58,483	58,513	0,030	59,197	0,714	0,684
	MobCal	32 Mb/s	34,65	34,10	-0,55	33,42	-1,23	-0,68	0,911	0,914	0,003	0,903	-0,008	-0,011	8,908	9,440	0,532	8,436	-0,471	-1,003
	Moyenne		**35,44**	**34,90**	**-0,54**	**34,75**	**-0,69**	**-0,15**	**0,902**	**0,904**	**0,002**	**0,899**	**-0,004**	**-0,006**	**28,864**	**30,136**	**1,272**	**29,982**	**1,118**	**-0,154**
1080p	Pedestrian	17 Mb/s	40,66	40,18	-0,48	40,40	-0,26	0,21	0,949	0,949	-0,000	0,949	-0,000	0,000	17,641	18,613	0,972	18,310	0,668	-0,304
	SunFlower	29 Mb/s	41,22	40,88	-0,34	40,97	-0,26	0,09	0,961	0,962	0,001	0,959	-0,001	-0,003	21,748	24,825	3,077	22,493	0,745	-2,332
	Tractor	17 Mb/s	37,17	36,83	-0,34	36,87	-0,30	0,04	0,921	0,920	-0,001	0,918	-0,003	-0,002	22,285	23,516	1,231	23,446	1,161	-0,070
	Moyenne		**39,68**	**39,30**	**-0,39**	**39,41**	**-0,28**	**0,11**	**0,944**	**0,943**	**0,000**	**0,942**	**-0,001**	**-0,002**	**20,558**	**22,318**	**1,760**	**21,416**	**0,858**	**-0,902**

Tableau 4.7. Résultats de PSNR, SSIM et Ringing pour les encodages x264, x264 avec quantification adaptative (QA) et avec quantification adaptative contrôlée par le JND en texture (QAJND) Utilisation uniquement des modes intra 16x16 – Débits Intra correspondant à des GOP Inter (IBBP12 et IBBP33) à 3Mbit/s pour les formats 720p et 4Mbit/s pour les formats 1080p

Modes Intra 16x16 **Moyens Débits**			PSNR						SSIM						Ringing					
			x264	x264 avec QA		x264 avec QAJND			x264	x264 avec QA		x264 avec QAJND			x264	x264 avec QA		x264 avec QAJND		
			PSNR (a1)	PSNR (b1)	ΔP (c1) (b1)-(a1)	PSNR (d1)	ΔP (e1) (d1)-(a1)	ΔP (f1) (d1)-(b1)	SSIM (a2)	SSIM (b2)	ΔS (c2) (b2)-(a2)	SSIM (d2)	ΔS (e2) (d2)-(a2)	ΔS (f2) (d2)-(b2)	Ringing (a3)	Ringing (b3)	ΔR (c3) (b3)-(a3)	Ringing (d3)	ΔR (e3) (d3)-(a3)	ΔR (f3) (d3)-(b3)
720p	Soccer	27 Mb/s	39,72	39,02	-0,70	39,20	-0,51	0,18	0,955	0,957	0,002	0,956	0,001	-0,001	8,826	11,210	2,383	10,002	1,175	-1,208
	Binocular	17 Mb/s	37,24	36,81	-0,44	36,97	-0,28	0,16	0,908	0,909	0,002	0,907	-0,000	-0,002	28,355	31,695	3,340	30,294	1,939	-1,401
	MobCal	53 Mb/s	37,70	36,99	-0,71	36,25	-1,45	-0,74	0,945	0,949	0,005	0,943	-0,002	-0,006	4,216	5,075	0,859	4,492	0,277	-0,582
	Moyenne		**38,22**	**37,60**	**-0,61**	**37,47**	**-0,75**	**-0,13**	**0,936**	**0,939**	**0,003**	**0,935**	**-0,001**	**-0,003**	**13,799**	**15,993**	**2,194**	**14,929**	**1,130**	**-1,064**
1080p	Pedestrian	31 Mb/s	42,85	42,36	-0,49	42,60	-0,25	0,25	0,966	0,966	0,000	0,966	0,000	0,000	8,985	10,604	1,619	9,754	0,769	-0,850
	SunFlower	48 Mb/s	43,89	43,52	-0,37	43,65	-0,23	0,14	0,976	0,977	0,001	0,976	-0,001	-0,001	14,477	17,042	2,565	15,098	0,620	-1,944
	Tractor	31 Mb/s	40,21	39,90	-0,31	39,97	-0,24	0,07	0,955	0,956	0,00	0,955	-0,001	-0,001	12,279	13,023	0,744	12,771	0,491	-0,253
	Moyenne		**42,31**	**41,92**	**-0,39**	**42,07**	**-0,24**	**0,15**	**0,966**	**0,966**	**0,000**	**0,966**	**-0,000**	**-0,001**	**11,914**	**13,556**	**1,643**	**12,541**	**0,627**	**-1,016**

Tableau 4.8. Résultats de PSNR, SSIM et Ringing pour les encodages x264, x264 avec quantification adaptative (QA) et avec quantification adaptative contrôlée par le JND en texture (QAJND) Utilisation uniquement des modes intra 16x16 – Débits Intra correspondant à des GOP Inter (IBBP12 et IBBP33) à 6Mbit/s pour les formats 720p et 8Mbit/s pour les formats 1080p

Modes Intra 4x4 et 16x16 **Bas Débits**			PSNR						SSIM						Ringing					
			x264	x264 avec QA		x264 avec QAJND			x264	x264 avec QA		x264 avec QAJND			x264	x264 avec QA		x264 avec QAJND		
			PSNR (a1)	PSNR (b1)	ΔP (c1) (b1)-(a1)	PSNR (d1)	ΔP (e1) (d1)-(a1)	ΔP (f1) (d1)-(b1)	SSIM (a2)	SSIM (b2)	ΔS (c2) (b2)-(a2)	SSIM (d2)	ΔS (e2) (d2)-(a2)	ΔS (f2) (d2)-(b2)	Ringing (a3)	Ringing (b3)	ΔR (c3) (b3)-(a3)	Ringing (d3)	ΔR (e3) (d3)-(a3)	ΔR (f3) (d3)-(b3)
720p	Soccer	16 Mb/s	37,68	36,96	-0,71	37,13	-0,55	0,17	0,936	0,937	0,001	0,936	-0,000	-0,001	19,668	24,275	4,607	22,186	2,518	-2,089
	Binocular	9 Mb/s	35,94	35,33	-0,61	35,56	-0,38	0,23	0,887	0,8845	-0,002	0,884	-0,003	-0,001	57,434	64,355	6,921	60,135	2,701	-4,219
	MobCal	32 Mb/s	35,38	34,81	-0,57	34,12	-1,25	-0,69	0,920	0,924	0,004	0,913	-0,007	-0,010	8,257	9,066	0,810	7,747	-0,510	-1,320
	Moyenne		**36,33**	**35,70**	**-0,63**	**35,60**	**-0,73**	**-0,10**	**0,914**	**0,915**	**0,001**	**0,911**	**-0,003**	**-0,004**	**28,453**	**32,565**	**4,112**	**30,023**	**1,570**	**-2,543**
1080p	Pedestrian	17 Mb/s	41,40	40,84	-0,55	41,12	-0,28	0,27	0,956	0,955	-0,001	0,955	-0,000	0,001	18,881	21,800	2,919	20,527	1,647	-1,273
	SunFlower	29 Mb/s	42,58	42,23	-0,35	42,30	-0,28	0,07	0,970	0,971	0,001	0,970	-0,001	-0,002	22,803	26,328	3,525	23,477	0,675	-2,850
	Tractor	17 Mb/s	38,02	37,63	-0,39	37,72	-0,30	0,09	0,932	0,931	-0,001	0,930	-0,002	-0,001	23,286	25,144	1,858	24,341	1,056	-0,803
	Moyenne		**40,67**	**40,23**	**-0,43**	**40,38**	**-0,29**	**0,14**	**0,952**	**0,952**	**-0,000**	**0,951**	**-0,001**	**-0,001**	**21,656**	**24,424**	**2,768**	**22,782**	**1,126**	**-1,642**

Tableau 4.9. Résultats de PSNR, SSIM et Ringing pour les encodages x264, x264 avec quantification adaptative (QA) et avec quantification adaptative contrôlée par le JND en texture (QAJND) Utilisation de tous les modes intra 4x4 et 16x16 – Débits Intra correspondant à des GOP Inter (IBBP12 et IBBP33) à 3Mbit/s pour les formats 720p et 4Mbit/s pour les formats 1080p

Modes Intra 4x4 et 16x16 **Moyens Débits**			PSNR						SSIM						Ringing					
			x264	x264 avec QA		x264 avec QAJND			x264	x264 avec QA		x264 avec QAJND			x264	x264 avec QA		x264 avec QAJND		
			PSNR (a1)	PSNR (b1)	ΔP (c1) (b1)-(a1)	PSNR (d1)	ΔP (e1) (d1)-(a1)	ΔP (f1) (d1)-(b1)	SSIM (a2)	SSIM (b2)	ΔS (c2) (b2)-(a2)	SSIM (d2)	ΔS (e2) (d2)-(a2)	ΔS (f2) (d2)-(b2)	Ringing (a3)	Ringing (b3)	ΔR (c3) (b3)-(a3)	Ringing (d3)	ΔR (e3) (d3)-(a3)	ΔR (f3) (d3)-(b3)
720p	Soccer	27 Mb/s	40,79	40,05	-0,74	40,29	-0,50	0,24	0,963	0,965	0,002	0,964	0,001	-0,001	8,441	11,499	3,058	9,659	1,218	-1,840
	Binocular	17 Mb/s	38,05	37,54	-0,52	37,76	-0,30	0,22	0,918	0,919	0,001	0,918	-0,000	-0,001	28,783	34,843	6,060	31,432	2,649	-3,411
	MobCal	53 Mb/s	38,43	37,73	-0,70	36,97	-1,46	-0,76	0,950	0,955	0,005	0,949	-0,001	-0,006	3,865	4,757	0,892	4,112	0,247	-0,645
	Moyenne		**39,09**	**38,44**	**-0,65**	**38,34**	**-0,75**	**-0,10**	**0,944**	**0,946**	**0,003**	**0,944**	**-0,000**	**-0,003**	**13,696**	**17,033**	**3,336**	**15,068**	**1,371**	**-1,965**
1080p	Pedestrian	31 Mb/s	43,55	43,02	-0,52	43,29	-0,26	0,26	0,970	0,970	0,000	0,971	0,000	0,000	9,292	11,801	2,509	10,571	1,279	-1,230
	SunFlower	48 Mb/s	45,18	44,76	-0,42	44,97	-0,20	0,22	0,982	0,982	0,001	0,981	-0,000	-0,001	14,463	17,497	3,034	14,972	0,509	-2,526
	Tractor	31 Mb/s	41,35	41,05	-0,31	41,13	-0,23	0,08	0,965	0,965	0,000	0,964	-0,000	-0,001	12,983	13,612	0,629	13,050	0,067	-0,562
	Moyenne		**43,36**	**42,94**	**-0,42**	**43,13**	**-0,23**	**0,19**	**0,972**	**0,972**	**0,000**	**0,972**	**-0,000**	**-0,000**	**12,246**	**14,303**	**2,057**	**12,864**	**0,618**	**-1,439**

Tableau 4.10. Résultats de PSNR, SSIM et Ringing pour les encodages x264, x264 avec quantification adaptative (QA) et avec quantification adaptative contrôlée par le JND en texture (QAJND) Utilisation de tous les modes intra 4x4 et 16x16 – Débits Intra correspondant à des GOP Inter (IBBP12 et IBBP33) à 6Mbit/s pour les formats 720p et 8Mbit/s pour les formats 1080p

4. 5. 3. 1. **Résultats de Ringing pour l'encodage utilisant tous les modes Intra autorisés**

Afin d'analyser plus finement les résultats précédents, le Tableau 4.11 et Tableau 4.12 reprennent les résultats de la métrique de Ringing pour le cas de l'encodage Intra utilisant tous les modes de prédiction à bas et moyen débit respectivement.

- Comme nous l'avons vu précédemment, dans tous les cas, la séquence encodée sans quantification adaptative obtient la meilleure note (colonne a), cependant notre solution réduit toujours l'effet de Ringing comparativement à la quantification x264 (colonne f).

- Entre les trois séquences 1280x720p, il existe des différences importantes de niveau de Ringing, la séquence *Binocular* est spécialement sensible à cet effet de par ces nombreux contours francs.

- La réduction de Ringing amenée par notre solution comparativement à la quantification adaptative x264 est d'autant plus forte que l'effet de Ringing est présent. Ainsi, la plus forte réduction de l'artefact est obtenue pour la séquence *Binocular*, de plus, la réduction du Ringing est plus importante à bas débit qu'à moyen débit.

Pour approfondir l'analyse de nos résultats, nous présentons des comparaisons visuelles des séquences 1280x720 dans la suite du chapitre.

			x264	**x264 avec QA**		**x264 avec QAJND**		
	Séquence	**Débit [Mb/s]**	**Ringing** (a)	**Ringing** (b)	ΔRinging (c) =(b)-(a)	**Ringing** (d)	ΔRinging (e) = (d)-(a)	ΔRinging (f) = (d)-(b)
720 50p	Soccer	16	19,668	24,275	4,607	22,186	2,518	-2,089
	Binocular	9	57,434	64,355	6,921	60,135	2,701	-4,219
	MobCal	32	8,257	9,066	0,810	7,747	-0,510	-1,320
	Moyenne		**28,453**	**32,565**	**4,112**	**30,023**	**1,570**	**-2,543**
1080 25p	Pedestrian	17	18,881	21,800	2,919	20,527	1,647	-1,273
	SunFlower	29	22,802	26,328	3,525	23,477	0,675	-2,850
	Tractor	17	23,286	25,144	1,858	24,341	1,056	-0,803
	Moyenne		**21,656**	**24,424**	**2,768**	**22,782**	**1,126**	**-1,642**

Tableau 4.11. Résultats de la métrique « Ringing » pour les huit séquences HD encodées en x264, x264 avec quantification adaptative et avec quantification adaptative contrôlée par le JND en texture – Tous modes Intra autorisés – <u>Bas Débits</u>

			x264	**x264 avec QA**		**x264 avec QAJND**		
	Séquence	**Débit [Mb/s]**	**Ringing** (a)	**Ringing** (b)	ΔRinging (c) =(b)-(a)	**Ringing** (d)	ΔRinging (e) = (d)-(a)	ΔRinging (f) = (d)-(b)
720 50p	Soccer	27	8,441	11,499	3,058	9,659	1,218	-1,840
	Binocular	17	28,783	34,843	6,060	31,432	2,649	-3,411
	MobCal	53	3,865	4,757	0,892	4,112	0,247	-0,645
	Moyenne		**13,696**	**17,033**	**3,336**	**15,068**	**1,371**	**-1,965**
1080 25p	Pedestrian	31	9,292	11,801	2,509	10,571	1,279	-1,230
	SunFlower	48	14,463	17,497	3,034	14,972	0,509	-2,526
	Tractor	31	12,983	13,612	0,629	13,050	0,067	-0,562
	Moyenne		**12,246**	**14,303**	**2,057**	**12,864**	**0,618**	**-1,439**

Tableau 4.12. Résultats de la métrique « Ringing » pour les huit séquences HD encodées en x264, x264 avec quantification adaptative et avec quantification adaptative contrôlée par le JND en texture – Tous modes Intra autorisés – <u>Moyens Débits</u>

4. 5. 4. Analyse visuelle des images décodées

Nous présentons dans ce paragraphe différentes images issues des séquences *Soccer*, *Mobile & Calendar*, *Binocular* et *ParkJoy* au format 1280x720 50p pour des encodages à bas débit utilisant uniquement les modes Intra 16x16 (Figure 4.24, Figure 4.25 et Figure 4.26), et autorisant tous les modes de prédiction (Figure 4.27, Figure 4.28 et Figure 4.29).

Dans chacune de ces figures, l'image originale analysée est présentée (a), ainsi que les cartes de ΔQP définies par la quantification adaptative x264 (b) et celle que nous proposons (c). Pour la représentation des cartes de ΔQP nous avons adopté le code couleur suivant : les macroblocs rouges représentent ceux pour lesquels la quantification adaptative augmente le QP et par conséquent réduit le budget binaire. Les macroblocs verts représentent ceux pour lesquels la quantification adaptative diminue le QP pour préserver l'information. Les macroblocs noirs ne voient pas leur QP modifié. Des agrandissements pour l'encodage x264 sans quantification adaptative, avec quantification adaptative x264 et avec notre solution sont respectivement présentés en (d), (e) et(f).

Nous comparons la carte de ΔQP définie par la quantification adaptative x264 (b) et celle contrôlée par le masquage en texture (c) des Figure 4.24, Figure 4.25 et Figure 4.26, correspondant respectivement aux images des séquences *Soccer*, *Mobile & Calendar* et *Binocular*. Les deux quantifications adaptatives concentrent le budget binaire dans les zones à faible activité spatiale en réduisant le paramètre de quantification dans ces régions (arrière-plan des trois images). La différence principale entre les deux quantifications adaptatives concerne la gestion des contours. En effet, notre solution préserve les contours présents dans les images alors qu'ils sont sévèrement quantifiés par la quantification adaptative x264. La préservation des contours se traduit par une nette réduction de l'effet de Ringing (cartes (f)) quels que soient les modes Intra utilisés à l'encodage.

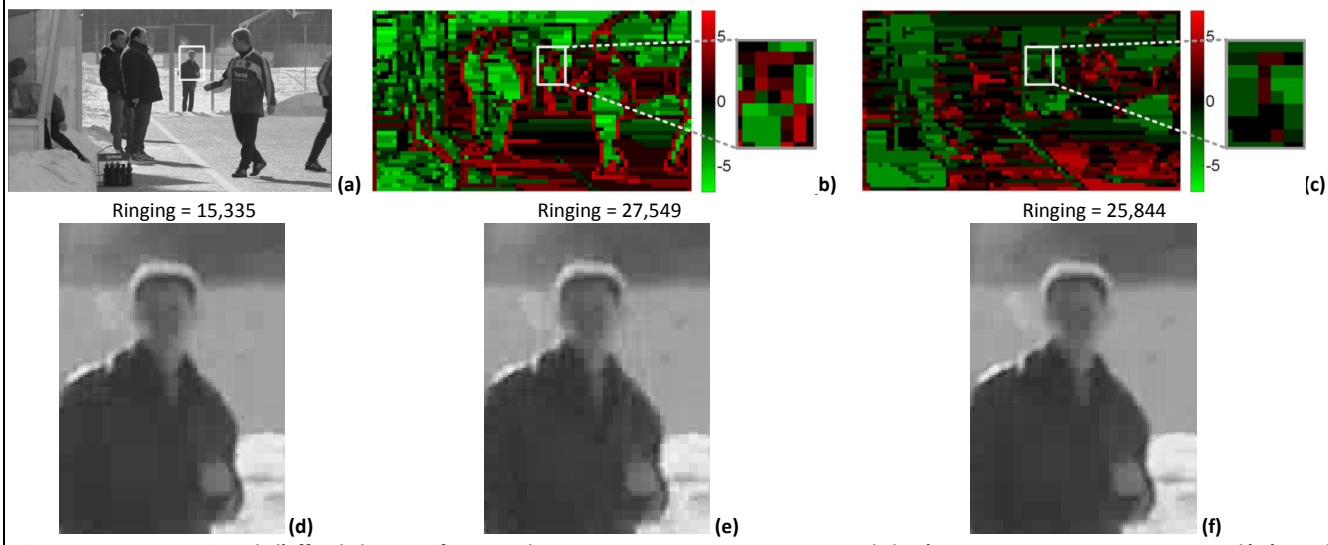


Figure 4.24. Comparaison de l'effet de la quantification adaptative QA et QA_{JND} pour une image de la séquence Soccer 1280x720 50p encodée à 16Mbit/s en GOP Intra (Qualité Intra équivalente à 3Mbit/s en GOP Inter) avec uniquement les partitions 16x16

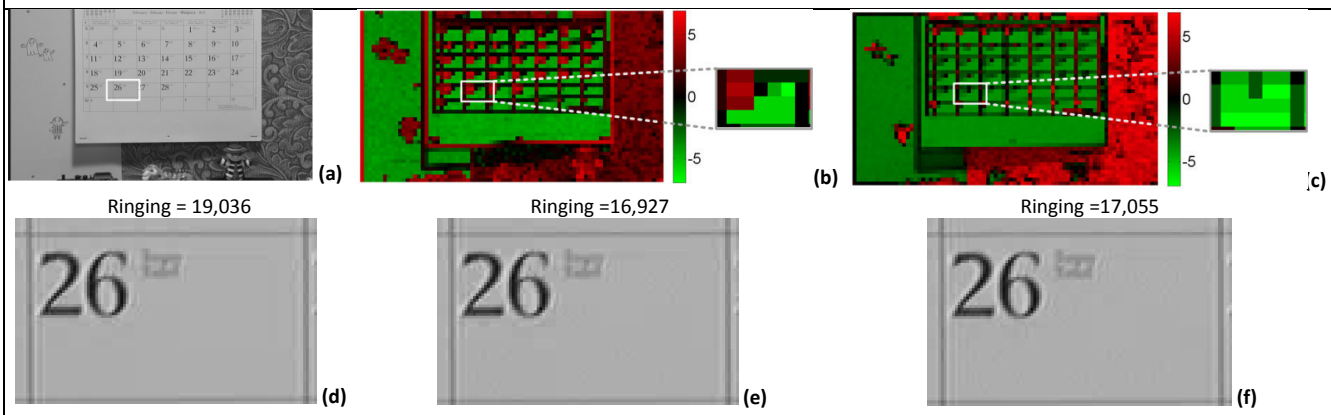


Figure 4.25. Comparaison de l'effet de la quantification adaptative QA et QA_{JND} pour une image de la séquence Mobile & Calendar 1280x720 50p encodée à 32Mbit/s en GOP Intra (Qualité Intra équivalente à 3Mbit/s en GOP Inter) avec uniquement les partitions 16x16

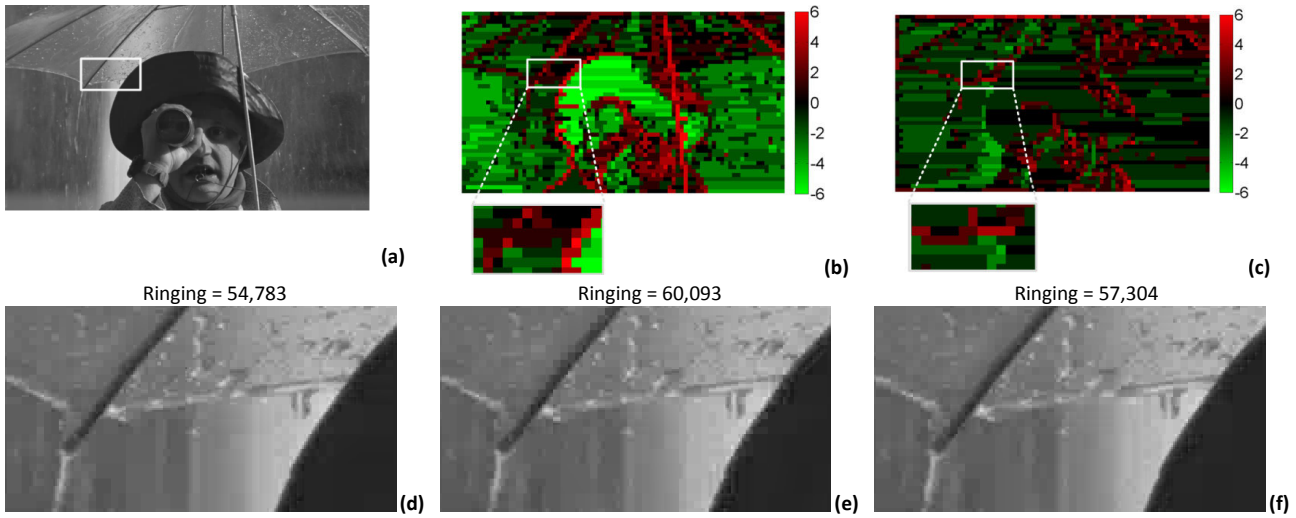


Figure 4.26. Comparaison de l'effet de la quantification adaptative QA et QA_{JND} pour une image de la séquence Binocular 1280x720 50p encodée à 32Mbit/s en GOP Intra (Qualité Intra équivalente à 3Mbit/s en GOP Inter) avec uniquement les partitions 16x16

(a) Image originale – (b) Carte de ΔQP utilisés à l'encodeur par la quantification adaptative (QA) – (c) Carte de ΔQP utilisés à l'encodeur par la quantification adaptative contrôlée par le masquage en texture (QA_{JND}) – (d) Agrandissement dans l'image encodée/décodée x264 – (e) Agrandissement dans l'image encodée/décodée x264 avec QA – (f) Agrandissement dans l'image encodée/décodée x264 avec QA_{JND}

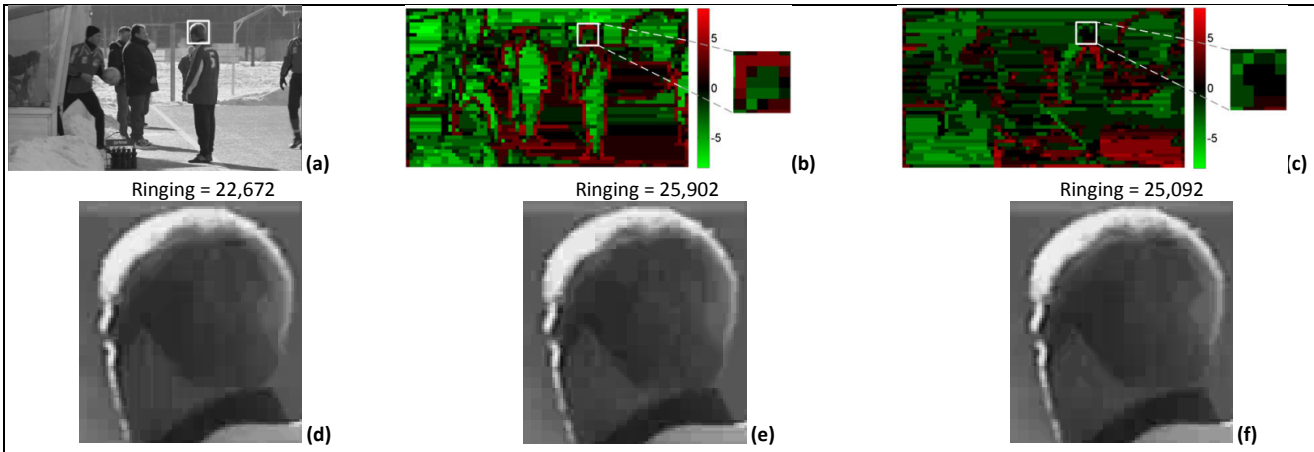


Figure 4.27. Comparaison de l'effet de la quantification adaptative QA et QA_{JND} pour une image de la séquence Soccer 1280x720 50p encodée à 16Mbit/s en GOP Intra (Qualité Intra équivalente à 3Mbit/s en GOP Inter) avec les partitions 4x4 et 16x16

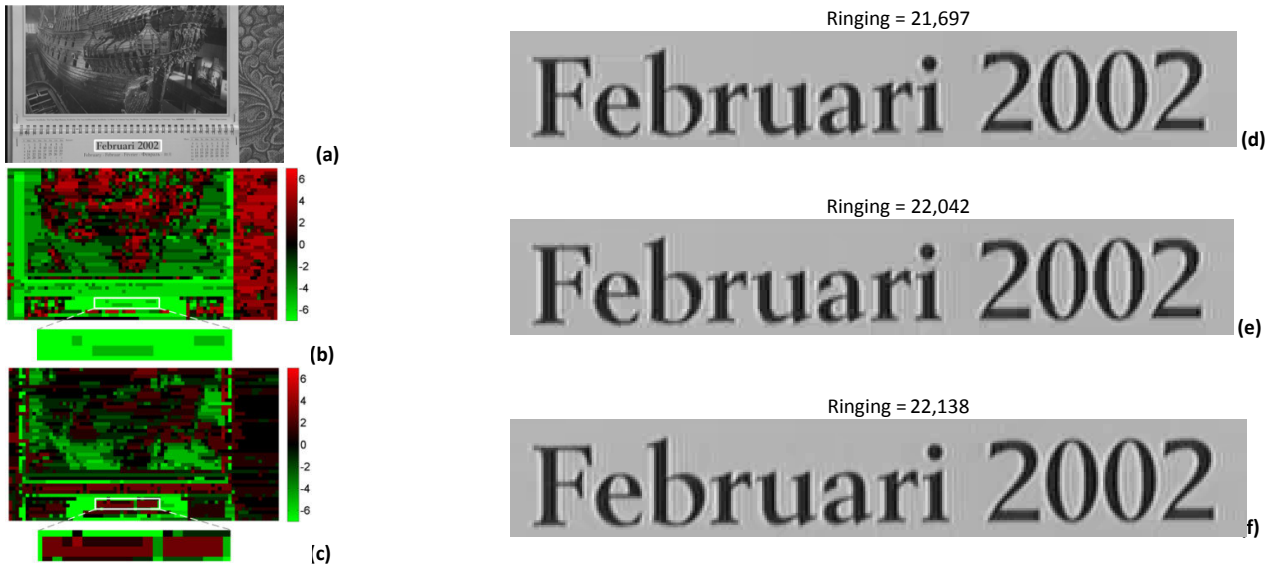


Figure 4.28. Comparaison de l'effet de la quantification adaptative QA et QA_{JND} pour une image de la séquence Mobile & Calendar 1280x720 50p encodée à 32Mbit/s en GOP Intra (Qualité Intra équivalente à 3Mbit/s en GOP Inter) avec les partitions 4x4 et 16x16

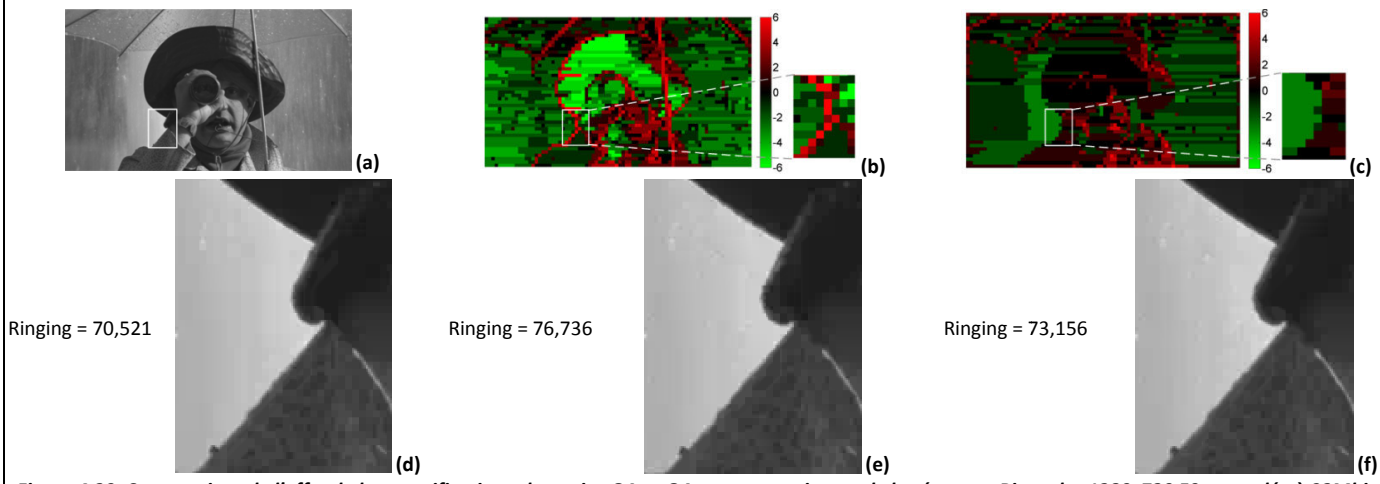


Figure 4.29. Comparaison de l'effet de la quantification adaptative QA et QA_{JND} pour une image de la séquence Binocular 1280x720 50p encodée à 32Mbit/s en GOP Intra (Qualité Intra équivalente à 3Mbit/s en GOP Inter) avec les partitions 4x4 et 16x16

(a) Image originale – (b) Carte de ΔQP utilisés à l'encodeur par la quantification adaptative (QA) – (c) Carte de ΔQP utilisés à l'encodeur par la quantification adaptative contrôlée par le masquage en texture (QA_{JND}) – (d) Agrandissement dans l'image encodée/décodée x264 – (e) Agrandissement dans l'image encodée/décodée x264 avec QA – (f) Agrandissement dans l'image encodée/décodée x264 avec QA_{JND}

Comme on vient de le voir, la quantification adaptative que nous proposons parvient à augmenter le budget binaire aux contours et ainsi à réduire l'effet de Ringing. Etant en encodage CBR, le budget supplémentaire ajouté aux contours est retiré autre part dans l'image. La réduction de précision dans les zones peu sensibles au sens du JND est peu visible, tandis que la préservation des contours permet d'améliorer la qualité globale ressentie de l'image. Pour illustrer ce propos, nous étudions la modification de répartition binaire apportée par notre solution par rapport à la quantification adaptative x264, pour une image particulière de la séquence *Binocular*. La Figure 4.30 présente la carte de rapport d'allocation binaire entre l'image encodée avec la quantification adaptative que nous proposons et la quantification adaptative x264. Pour comparer la répartition binaire au sein des deux images encodées avec les deux quantifications adaptatives, on vérifie qu'elles ont le même budget global, la différence entre les deux images est de 23 bits, soit 0,01% de variation de débit. Ainsi les macroblocs rouges représentent ceux pour lesquels notre proposition augmente le budget binaire comparativement à la quantification x264 et les macroblocs verts ceux pour lesquels le budget diminue.

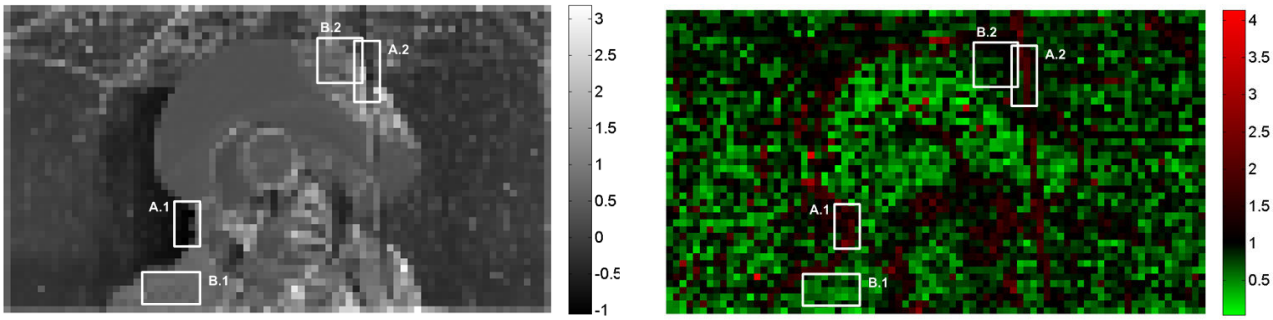


Figure 4.30. Etude de la modification d'allocation binaire amenée par notre solution par rapport à la quantification adaptative x264 pour une image encodée en Intra (mode 16x16) à 9Mbit/s
(a) Carte de JND texture moyen par macrobloc – (b) Carte de Rapport d'allocation binaire entre notre solution et la quantification adaptative x264

Pour analyser l'effet de notre solution, nous prenons deux zones pour lesquelles notre solution augmente le budget par rapport à la quantification adaptative x264 (zones A.1 et A.2) et deux zones pour lesquelles le budget est diminué (zones B.1 et B.2). Sur la Figure 4.31 présentant les zones A.1 et A.2 encodées avec les deux quantifications adaptatives, on confirme que notre solution permet de réduire l'effet de Ringing. En contrepartie, sur la Figure 4.32 présentant les zones B.1 et B.2, on voit que notre solution apporte une perte de précision dans le codage de ces zones texturées. La perte de précision dans les zones indiquées comme peu importantes par le JND sont peu visibles alors que la réduction de Ringing est notable. De cette manière, la répartition du budget obtenue avec notre solution permet une amélioration globale de la qualité ressentie.

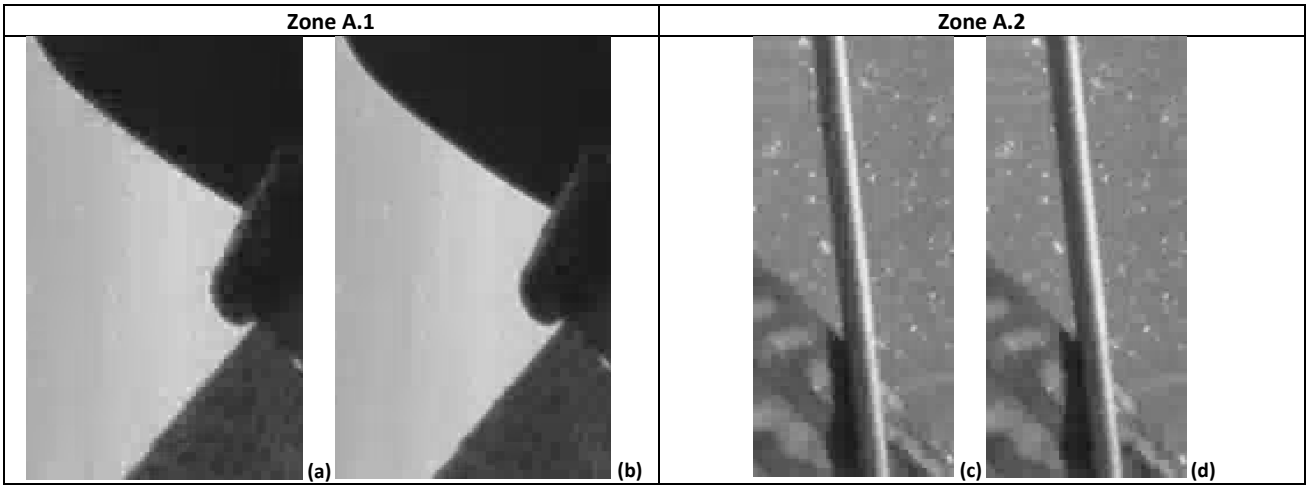


Figure 4.31. Agrandissements des zones A.1 et A.2 de la Figure 4.30 pour lesquelles notre proposition augmente le budget binaire
(a) et (c) image encodée avec quantification adaptative x264 - (b) et (d) image encodée avec notre proposition

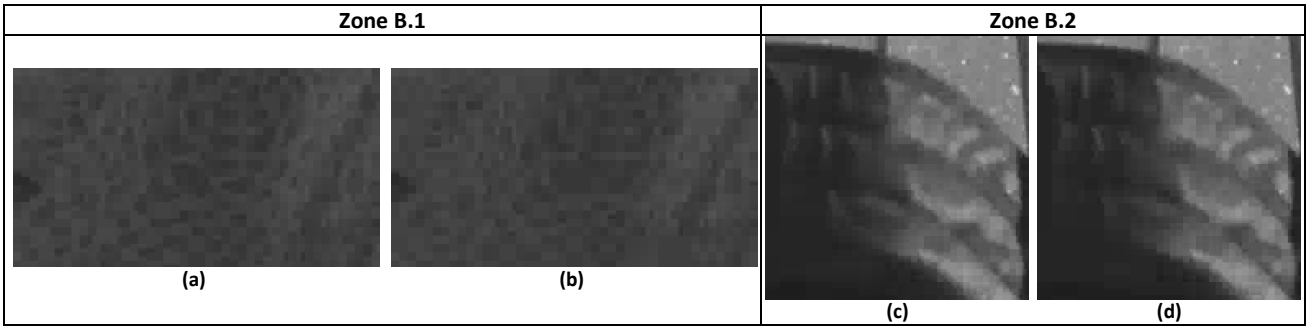


Figure 4.32. Agrandissements des zones B.1 et B.2 de la Figure 4.30 pour lesquelles notre proposition diminue le budget binaire (a) et (c) image encodée avec quantification adaptative x264 - (b) et (d) image encodée avec notre proposition

Pour la séquence *ParkJoy*, la métrique indique une augmentation du *Ringing* par la quantification contrôlée par le JND, cependant le *Ringing* n'est pas visible car l'image contient peu de contours. En revanche, la quantification adaptative contrôlée par le masquage en texture réduit la qualité perçue des zones faiblement texturées comparativement à la quantification x264. Comme le montre la Figure 4.33, dans les zones à faible activité spatiale comme le banc d'herbe, les deux quantifications adaptatives préservent le contenu (ΔQP négatif), mais la quantification contrôlée par le JND attribue un ΔQP tout de même plus fort que la quantification x264, ce qui se traduit par une impression d'aplat dans les agrandissements proposés par la Figure 4.33.

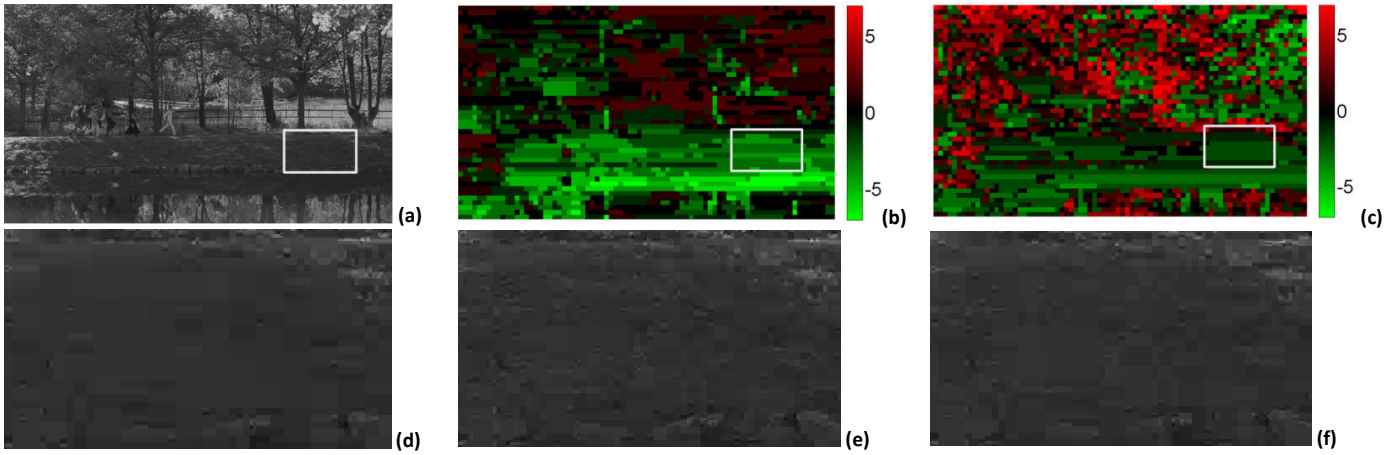


Figure 4.33. Comparaison de l'effet de la quantification adaptative x264 et JND pour une image de la séquence ParkJoy 1280x720 22Mbit/s, mode Intra 16x16
(a) Image originale - (b) et (c) Cartes de Delta QP utilisés à l'encodeur par la quantification adaptative x264 et JND par rapport à x264 – (d) Zoom dans l'image encodée x264 – (e) Zoom dans l'image encodée x264 + AQ - (f) Zoom dans l'image encodée x264+AQJND

En conclusion, l'introduction du masquage en texture dans la quantification adaptative permet de conserver les caractéristiques globales de la quantification adaptative x264, à savoir la concentration du budget binaire sur les macroblocs à faible activité spatiale, au détriment des macroblocs à forte activité spatiale qui peuvent abriter plus de distorsions. De plus, la gestion des contours par le modèle de texture permet une réduction de l'effet de *Ringing* à bas et moyen débit.

4. 5. 5. Effet de la quantification Adaptative proposée sur l'encodage Inter

L'étude présentée ci-dessus a été conduite en encodage Intra pour simplifier l'analyse des résultats en s'affranchissant des éventuelles différences de comportement par type d'image. Cependant nous vérifions dans ce paragraphe que notre solution permet également de réduire l'effet de Ringing en GOP Inter. Pour cela nous encodons deux séquences 1280x720 50p en GOP IBBP12 à 3Mbit/s et 6Mbit/s en profile x264 slow, avec CABAC et sans filtre de réduction de l'effet de bloc, en autorisant toutes les partitions intra et inter.

Comme pour le codage Intra, on vérifie que notre solution ne perturbe pas le contrôle de débit. Le Tableau 4.13 présente les débits des deux séquences encodées sans quantification adaptative, avec quantification adaptative x264 et avec notre solution, à 3 et 6Mbit/s. On note que notre solution amène une modification de débit au maximum de 2.81% par rapport à l'encodage x264 sans quantification adaptative.

		x264	QA + x264		QAJND + x264	
		[Mb/s]	[Mb/s]	Δ [%]	[Mb/s]	Δ [%]
Bas Débit (3Mb/s)	**Binocular**	3,11	3,10	-0,06	3,10	-0,04
	MobCal	3,01	2,92	-3,06	2,93	-2,81
Moyen Débit (6Mb/s)	**Binocular**	6,22	6,21	-0,12	6,21	-0,15
	MobCal	6,19	6,18	-0,16	6,04	-2,56

Tableau 4.13. Débits atteints par l'encodage x264 sans quantification adaptative, avec quantification adaptative et avec notre solution

Le Tableau 4.14 et le Tableau 4.15 présentent une comparaison des résultats des métriques de PSNR, SSIM et Ringing pour l'encodage Intra et Inter. On note que notre solution a un effet équivalent en GOP Intra et Inter d'après le PSNR et le SSIM. Cependant la métrique de Ringing indique que notre solution réduit moins voire augmente le Ringing lorsqu'elle est appliquée en encodage Inter Image.

Pourtant en analysant visuellement les séquences encodées en GOP Inter, nous remarquons que le Ringing aux contours des images est toujours réduit par notre solution comparativement à la quantification adaptative x264. Des parties d'image sont présentées par la Figure 4.34 et Figure 4.35 respectivement pour une image B de la séquence *Mobile & Calendar* et *Binocular*.

Ringing = 10,039 **(a)**

Ringing = 10,481 **(b)**

Ringing = 11,286 **(c)**

Figure 4.34. Comparaison de la quantification adaptative x264 et JND pour une image de la séquence Mobile & Calendar, 3Mbit/s, GOP IBBP12 (a) Zoom dans l'image encodée x264 – (b) Zoom dans l'image encodée x264 + AQ - (c) Zoom dans l'image encodée x264+AQJND

(a) Ringing = 27,729

(b) Ringing = 28,121

(c) Ringing = 28,217

Figure 4.35. Comparaison de la quantification adaptative x264 et JND pour une image de la séquence Mobile & Binocular, 3Mbit/s, GOP IBBP12 (a) Zoom dans l'image encodée x264 – (b) Zoom dans l'image encodée x264 + AQ - (c) Zoom dans l'image encodée x264+AQJND

			x264	QA + x264		QAJND + x264			x264	QA + x264		QAJND + x264		
			PSNR	**PSNR**	ΔPSNR	**PSNR**	ΔPSNR	ΔPSNR	**SSIM**	**SSIM**	ΔSSIM	**SSIM**	ΔSSIM	ΔSSIM
			(a)	(b)	(b)-(a)	(c)	(c)-(a)	(c)-(b)	(a)	(b)	(b)-(a)	(c)	(c)-(a)	(c)-(b)
Bas Débit	**Binocular**	**Intra** 9Mb/s	35,942	35,329	-0,614	35,561	-0,382	0,232	0,8867	0,8848	-0,0019	0,8835	-0,0032	-0,0013
		Inter 3Mb/s	35,192	34,573	-0,620	34,775	-0,418	0,202	0,8759	0,8715	-0,0044	0,8711	-0,0047	-0,0003
	MobCal	**Intra** 32 Mb/s	35,375	34,809	-0,567	34,122	-1,253	-0,687	0,9199	0,9235	0,0036	0,9134	-0,0065	-0,0101
		Inter 3Mb/s	33,283	32,801	-0,482	32,068	-1,215	-0,733	0,8980	0,8970	-0,0010	0,8811	-0,0169	-0,0160
Moyen Débit	**Binocular**	**Intra** 17Mb/s	38,054	37,535	-0,519	37,756	-0,298	-0,298	0,9181	0,9186	0,0005	0,9177	-0,0003	-0,0009
		Inter 6Mb/s	37,109	36,514	-0,595	36,695	-0,414	-0,414	0,9051	0,9026	-0,0025	0,9021	-0,0030	-0,0005
	MobCal	**Intra** 53 Mb/s	38,431	37,731	-0,700	36,968	-1,463	-1,463	0,9503	0,9552	0,0049	0,9494	-0,0008	-0,0058
		Inter 6Mb/s	35,939	35,371	-0,568	34,681	-1,259	-1,259	0,9321	0,9318	-0,0003	0,9234	-0,0087	-0,0085

Tableau 4.14. Résultats de PSNR et SSIM en encodage Intra (tous modes Intra) et Inter (IBBP12)

			x264	QA + x264		QAJND + x264		
			Ringing	**Ringing**	ΔRinging	**Ringing**	ΔRinging	ΔRinging
			(a)	(b)	(b)-(a)	(c)	(c)-(a)	(c)-(b)
Bas Débit	**Binocular**	**Intra** 9Mb/s	57,4343	64,355	6,921	60,135	2,701	-4,219
		Inter 3Mb/s	37,1948	37,742	0,548	38,779	1,584	1,036
	MobCal	**Intra** 32 Mb/s	16,2592	15,954	-0,305	16,591	0,332	0,636
		Inter 3Mb/s	14,2846	13,845	-0,439	16,413	2,129	2,568
Moyen Débit	**Binocular**	**Intra** 17Mb/s	28,7829	34,843	6,060	31,432	2,649	-3,411
		Inter 6Mb/s	22,0733	23,528	1,455	23,396	1,322	-0,132
	MobCal	**Intra** 53 Mb/s	6,8686	7,903	1,034	7,378	0,510	-0,525
		Inter 6Mb/s	8,3843	8,706	0,322	9,114	0,730	0,409

Tableau 4.15. Résultats de Ringing en encodage Intra (tous modes Intra) et Inter (IBBP12)

4. 6. Voies d'amélioration

Dans certains cas, la préservation des zones faiblement texturées par la quantification au JND n'est pas suffisante introduisant une impression gênante d'aplat coloré. Deux voies d'améliorations de notre solution sont envisagées et actuellement en cours d'étude : la préservation des zones faiblement texturées et l'exploitation de la carte de JND au niveau pixel plutôt qu'au niveau macrobloc.

En particulier, pour étudier la première voie d'amélioration, des tests visuels informels ont été réalisés au sein de Digigram consistant à présenter à des observateurs une image encodée à un QP fixe de plus en plus élevé et à demander aux observateurs quelles sont les dégradations les plus gênantes. Il ressort de ce test que les dégradations intervenant dans les zones contenant des textures faiblement contrastées sont les plus rapidement gênantes avant les zones homogènes et les contours. Il est donc envisagé d'apporter une amélioration au modèle JND utilisé jusqu'alors pour préserver les faibles textures.

4. 7. Conclusion

Dans ce chapitre nous avons analysé l'effet du préfiltre proposé au chapitre 2 dans un contexte d'encodage à débit constant. Nous avons montré que l'allocation du budget binaire au sein d'une image que réalise le contrôle de débit, ne suit pas nécessairement le modèle perceptuel de JND lorsque le filtre est appliqué de façon indépendante de l'encodeur. Nous avons ainsi conclu que l'amélioration de la qualité perçue à même débit nécessite d'intervenir au cœur de l'encodage pour contrôler l'allocation binaire.

Pour cela, différents travaux de la littérature du codage perceptuel ont été étudiés, et nous avons choisi de proposer une amélioration à la quantification adaptative utilisée par le codec x264, qui est reconnue comme la plus grande amélioration de qualité apportée à ce codec depuis le début de son développement [120]. La solution que nous proposons adapte localement le pas de quantification en fonction du seuil JND en texture moyen des macroblocs d'une image. Cette nouvelle quantification adaptative a été testée sur des séquences HD à faible et moyen débit et apporte une réduction de l'effet de Ringing dans la majorité des cas testés en GOP Intra et Inter.

Dans l'idée d'améliorer la qualité globale ressentie de notre solution, nous avons conduit le test subjectif suivant : des versions de plus en plus quantifiées d'une image HD sont présentées à un observateur et il lui est demandé d'indiquer quelles dégradations il perçoit. Dans les différentes images testées, il en ressort que les zones faiblement texturées sont les plus rapidement perçues comme dégradées par la quantification. Aussi, une voie de travail pour améliorer les performances de notre solution est d'apporter une attention particulière aux zones faiblement texturées en plus des contours.

Conclusion Générale

Les travaux de thèse présentés dans ce manuscrit ont porté sur le prétraitement pour optimiser l'encodage H.264/AVC. Nous avons proposé deux solutions distinctes, la première pour réduire le débit à qualité constante et la deuxième pour améliorer la qualité perçue à débit constant.

La première proposition est un préfiltre indépendant du codeur. Nous avons tout d'abord proposé dans le chapitre 2 un préfiltre basé sur le filtre passe-bas AWA, contrôlé par un modèle JND permettant de réduire le contenu perceptuellement peu important. Dans un souci de réduction de complexité en vue d'une application temps réel, une version simplifiée du filtre AWA utilisant uniquement des poids binaires a été proposée, et le support de filtrage a été limité à une taille de 3x3 pixels. Ce préfiltre a été testé sur des séquences SD et HD 720p pour les encodeurs H.264/AVC AQILIM (développé par Digigram) et x264. L'encodage réalisé en VBR utilise un paramètre de quantification constant, le débit en sortie de l'encodeur est par conséquent dépendant du contenu de la séquence vidéo. En appliquant le prétraitement proposé, nous amenons un gain moyen en débit de 5.5% sur l'ensemble de nos tests avec un maximum de 17% en SD et 22% en HD pour une différence de qualité non perçue par un panel d'observateurs (-0.036 MOS sur une échelle PC (Paired Comparison) à sept niveaux).

Dans le chapitre 3 nous avons étudié les filtres AWA et Bilatéral avec des supports de filtrage étendus de taille 11x11 afin d'exploiter au mieux les redondances des séquences HD pour une application de réduction de bruit et de prétraitement pour l'encodage H.264/AVC. Afin de trouver le meilleur compromis entre réduction de bruit et préservation de la structure de l'image, nous avons été amenés à définir deux nouveaux filtres passe-bas, le BilAWA et le Bilatéral seuillé. Nous avons ensuite intégré le modèle JND à ces filtres pour définir deux nouveaux prétraitements perceptuels qui ont montré une réduction moyenne de 20% pour une différence de qualité très peu perçue (-0.16 avec un intervalle confiance de 0.15).

La deuxième proposition intègre le modèle JND au sein de l'encodeur pour contrôler l'allocation binaire et ainsi améliorer la qualité perçue à débit constant. Notre travail a porté sur le contrôle de débit du codec x264 et la réduction de l'effet de Ringing à moyens et bas débits pour des séquences HD au format 1080 25p et 720 50p. Nous avons proposé une amélioration de la quantification adaptative x264 en contrôlant le paramètre de quantification attribué par macrobloc avec le masquage en texture défini par le modèle JND de Yang et al. La métrique de Ringing définie par [110] ainsi que l'analyse visuelle montrent une réduction notable de l'effet de Ringing dans la majorité des cas testés.

A l'issue de ces travaux de thèse, plusieurs perspectives sont ouvertes et Digigram compte particulièrement investiguer les voies suivantes :

- De par les résultats obtenus par nos deux solutions, nous souhaitons coupler les deux prétraitements contrôlés par le même modèle perceptuel afin d'améliorer la qualité perçue en encodage CBR. Le préfiltre appliqué en début d'encodage permettra alors de sauver du budget binaire qui sera réalloué par la quantification adaptative.
- En vue d'une intégration dans les encodeurs H.264/AVC de Digigram, nous comptons étudier la faisabilité d'une implémentation temps réel des préfiltres proposés au chapitre 2 et 3. Pour cela, nous considérerons plusieurs solutions : l'implémentation GPU, la vectorisation et l'implémentation multi-cœur.
- Nous souhaitons étudier l'intérêt d'ajouter le masquage temporel au profile JND spatial pour le contrôle des préfiltres indépendants de l'encodeur, ainsi que considérer le filtrage des couches de chrominance.
- Ces travaux ayant révélé l'importance critique de la mesure de la qualité vidéo, nous souhaitons faire évoluer le protocole de tests subjectifs et spécialement l'exploitation des résultats afin d'affiner nos analyses sur la qualité perçue des solutions que nous proposons.

Ces travaux se situeront naturellement autour des futurs encodeurs HEVC autant que sur les encodeurs H264 existant.

Glossaire

ASSP	Application Specific Standard Product Circuit intégré dédié à un marché spécifique.
AVC	Advanced Video Coding
AC (coefficient)	Coefficient DCT X(u,v) de rang fréquentiel (u,v) non nul.
Bloc	Partition carrée d'un macrobloc sur laquelle est appliquée la DCT. Bloc 8x8 en MPEG2, 4x4 ou 8x8 en H.264/AVC, et 32x32, 16x16, 8x8 ou 4x4 en HEVC.
Buffer	Espace mémoire contenant des informations temporaires en attente de traitement. Permet d'absorber les variations de temps de traitement et d'arrivée des données.
CABAC	Context-based Adaptive Binary Arithmetic Coding. Codage entropique présent dans les normes H.264/AVC et HEVC.
CAVLC	Context-based Adaptive Variable Length Coding Codage entropique présent dans la norme H.264/AVC.
DC (coefficient)	Coefficient DCT X(0,0) de rang fréquentiel nul. Proportionnel à la valeur moyenne du bloc de pixel.
DCT	Discrete Cosine Tranform (Transformée en cosinus discrete)
Deblocking	Voir Filtre de réduction de l'effet de bloc
EQM	Erreur Quadratique Moyenne
Filtre de réduction de l'effet de bloc	Filtre appliqué aux bordures des blocs décodés pour réduire l'artefact appelé effet de blocs. Présent dans les normes H.264/AVC et HEVC.
GOP	Group Of Pictures Ordonnancement des types d'image B et P entre deux image I.
H.262	Voir MPEG-2
H.263 HLP	H.263 High Latency Profile Norme de compression vidéo standardisée par l'ITU-T Video Coding Experts Group (VCEG) en 1996.
H.264/AVC	H.264 Advanced Video Coding Norme de compression vidéo standardisée par le JVT (Joint Video Team) issu des groupes MPEG (Moving Picture Expert Group) et VCEG (Video Coding Expert Group, ITU-T) en 2003.
HEVC	High Efficiency Video Coding Nouveau standard de compression vidéo normalisé par le JCT-VC au début de l'année 2013. Appelé également H.265.
HM	Logiciel de référence HEVC
Image B	Image codée en mode Inter Bidirectionnel. Utilisation d'une image de référence passée et future pour l'estimation de mouvement.
Image I	Image codée en mode Intra, de manière indépendante des autres images de la séquence.

Image P	Image codée en mode Inter Prédite. Utilisation d'une image de référence passée pour l'estimation de mouvement.
In-loop	Implémentation d'un filtre à l'encodeur et au décodeur.
ITU-T	International Telecommunication Union - Telecommunication standardization sector
JCT-VC	Joint Collaborative Team on Video Coding Groupe d'experts vidéo issu des groupes VCEG de l'ITU-T et MPEG (ISO/IEC JTC 1/SC 29/WG 11), crée en 2010 pour le développement du standard HEVC.
JM	Logiciel H.264/AVC de référence
JVT	Joint Video Team Groupe d'experts vidéo issu des groupes VCEG de l'ITU-T et MPEG (ISO/IEC JTC 1/SC 29/WG 11), crée pour le développement du standard H.264/AVC.
LPC-SI	Local Phase Coherence - Sharpness Index Métrique objective sans référence mesurant l'artefact de flou
Macrobloc	Unité élémentaire pour les codeurs MPEG2 et H.264/AVC (16x16 pixels). Plus petit élément pouvant avoir son propre paramètre de quantification et mode de codage (Intra- ou Inter-image).
MPEG-2 MP	MPEG-2 Main Profile Norme de compression vidéo standardisée par l'ITU-T Video Coding Experts Group (VCEG) et ISO/IEC Moving Picture Experts Group (MPEG) en 1996. Aussi appelé H.262.
MPEG-4 (Part 2) ASP	MPEG-4 Advanced Simple Profile Norme de compression vidéo standardisée par le groupe ISO/IEC Moving Picture Experts Group (MPEG) en 1999. Aussi appelé MPEG-4 Visual.
MSE	Mean Squared error. Voir EQM.
NAL	Network Abstraction Layer Description de l'organisation du flux de données codées permettant de faciliter le transport de la vidéo sur des réseaux de diffusion.
NALU	NAL Unit Sous-ensemble de données représentant des données vidéo et/ou des informations nécessaires au bon fonctionnement du décodage.
Partition (Sous-partition)	Découpe d'un macrobloc pouvant avoir sa propre prédiction. En codage Inter-image les découpes autorisées sont 8x16, 16x8, 8x8, 4x8, 8x4 et 4x4. Uniquement 4x4 en codage Intra-image
PSNR	Peak Signal to Noise Ratio Mesure objective avec référence basée sur l'EQM (MSE en anglais) et exprimée en dB.
QP	Quantization Parameter – Paramètre de quantification Paramètre permettant de régler la quantification.
Raster order	Balayage ligne par ligne de pixels
Ringing	Artefact de compression se traduisant par l'apparition d'activité parasite (gigue) aux contours des images d'une séquence vidéo.
SAD	Sum Of Absolute Differences – Somme des différences absolues
SAO (filtre)	Sample Adaptive Offset

	Filtre in-loop appliqué à la suite du filtre de réduction de l'effet de bloc dans la norme HEVC, pour réduire les erreurs de quantification.
Slice	Ensemble de macroblocs (H.264/AVC) ou CTU (HEVC) d'une image en ordre raster. Une slice est décodable indépendamment des autres slices de l'image.
SSD	Sum of Squared Differences – Somme des différences quadratiques
SSE	Streaming SIMD extensions Instructions de calculs vectoriels pour processeur x86.
SSIM	Structural SiMilarity Mesure objective de qualité avec référence.
Tile (Tuile)	Ensemble de CTU regroupés en zones rectangulaire. Les tuiles sont décodables indépendamment les unes des autres et permettent de paralléliser les processus d'encodage et de décodage.
VQM	Video Quality Metric Métrique objective de qualité avec référence.
WPP	WaveFront Parallel Processing Traitement des lignes de CTU HEVC en parallèle.
Zigzag Order	Balayage des coefficients fréquentiels de la plus basse à la plus haute fréquence.

Références

[1] M. Marinescu, Y. Ansade et J. Weber, «Audio data transmission system between a master module and slave modules by means of a digital communication network». Brevet Brevet FR 2 829 655, WO 03/023759, US 2003/0050989, Sep. 2001.

[2] Rovi et Mainconcept, «Video SDK,» [En ligne]. Available: http://www.mainconcept.com/eu/products/sdks/video/h264avc.html. [Accès le 2013].

[3] I. E. Richardson, H.264 and Mpeg-4 Video Compression: Video Coding for Next-Generation Multimedia, Wiley-Blackwell, 2003.

[4] Y. Wang, J.Ostermann et Y. Q. Zhang, Video Processing and Communications, Prentice-Hall, 2002.

[5] M. S. Zhu et Kai-Kuang, «A New Diamond Search Algorithm for Fast Block-Matching,» *IEEE Transactions on Image Processing,* vol. 9 (Issue: 2), pp. 287 - 290, Feb 2000.

[6] E. Vidal, «Auswirkungen der GOP-Parametrierung auf H.264,» *FKT,* pp. 456-462, 08-09 2012.

[7] P. Yip et K. R. Rao, Discrete Cosine Transform: Algorithms, Advantages, Applications, Boston: Academic Press, 1990.

[8] C. E. Shannon, «A Mathematical Theory of Communication,» *The Bell System Technical Journal,* vol. 27, pp. 379–423, 623–656, Jul., Oct. 1948.

[9] T. Wiegand, G. Sullivan, G. Bjontegaard et A. Luthra, «Overview of the H.264/AVC video coding standard,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 13 (Issue: 7), pp. 560 - 576, Jul. 2003.

[10] D. Marpe, H. Schwarz et T.Wiegand, «Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 13 (Issue: 7), p. 620–636, Jul. 2003.

[11] P. List, A. Joch, J. Lainema, G. Bjøntegaard et M. Karczewicz, «Adaptive deblocking filter,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 13 (Issue: 7), p. 614–619, Jul. 2003.

[12] L. Wang, R. Gandhi, K. Panusopone, Y. Yu et A. Luthra, «Adaptive Frame/Field Coding for JVT,» *Joint Video Team of ISO/IEC MPEG & ITU-T VECG, 4th Meeting: Klagenfurt, Austria,* Jul. 2002.

[13] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm et T. Wiegand, «High efficiency video coding (HEVC) text specification, draft 8, ITU-T/ISO/IEC Joint Collaborative Team on Video,» Jul. 2012.

[14] G. J. Sullivan, J.-R. Ohm, W.-J. Han et T. Wiegand, «Overview of the High Efficiency Video Coding (HEVC) Standard,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 22 (Issue: 12), Dec. 2012.

[15] P. Bordes, G. Clare, F. Henry, M. Raulet et J. Viéron, «An overview of the emerging HEVC standard,» *ISIVC,* Jul. 2012.

[16] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan et T. Wiegand, «Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC),» *IEEE Transactions on circuits and systems for video technology,* vol. 22 (Issue: 12), pp. 1669 - 1684, Dec. 2012.

[17] B. Li, G. J. Sullivan et J. Xu, «Comparison of Compression Performance of HEVC Draft 7 with AVC High Profile,» *JCTVC-J0236,* 2012.

[18] C.-M. Fu, E. Alshina, A. Alshin, Y.-W. Huang, C.-Y. Chen, C.-Y. Tsai, C.-W. Hsu, S.-M. Lei, J.-H. Park et W.-J. Han, «Sample Adaptive Offset in the HEVC Standard,» *IEEE Transactions on circuits and systems for video technology,* vol. 22 (Issue: 12), pp. 1755 - 1764, 2012.

[19] Gregory Cox, Senior Application Engineer, ATEME, «An Introduction to UHDTV and HEVC,» Nov. 2013. [En ligne]. Available: http://ateme.com/an-introduction-to-uhdtv-and-hevc.

[20] S. Wang, R. A., Z. Wang et M. Siwei, «SSIM-Motivated Rate-Distortion Optimization for Video Coding,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 22 (Issue: 4), pp. 516 - 529, Apr. 2011.

[21] T. Wiegand et B. Girod, «Lagrange multiplier selection in hybrid video coder control,» *International Conference on Image Processing,* vol. 3, pp. 542 - 545, 2001.

[22] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini et G. J. Sullivan, «Rate-Constrained Coder Control and Comparison of Video Coding Standards,» *IEEE Transactions on circuits and systems for video technology,* vol. 13 (Issue: 7), pp. 688 - 703, Jul. 2003.

[23] T. Wiegand et B. Girod, «Parameter selection in Lagrangian hybrid video coder control,» *International Conference on Image Processing,* vol. 3, pp. 542 - 545, 2001.

[24] L. Z.G., G. W., P. F., M. S.W., L. K.P., F. G.N., L. X., R. S., H. Lu et L. Y., «Adaptive rate control for H.264,» *Journal of Visual Communication and Image Representation,* vol. 7 (Issue: 2), pp. 376-406, April 2006.

[25] x264. [En ligne]. Available: http://www.videolan.org/developers/x264.html. [Accès le 2013].

[26] Fraunhofer, «H.264/AVC Software Coordination,» Jan. 2001. [En ligne]. Available: http://iphome.hhi.de/suehring/tml/.

[27] VideoLAN. [En ligne]. Available: http://www.videolan.org/.

[28] V. G. MSU Graphics & Media Lab, «MSU Video Codecs Comparison,» May 2003. [En ligne]. Available: http://compression.ru/video/codec_comparison/codec_comparison_en.html.

[29] Z. Wang, A. C. Bovik, H. R. Sheikh et E. P. Simoncelli, «Image quality assessment: from error visibility to structural similarity,» *IEEE Transactions on image processing,* vol. 13 (Issue: 4), pp. 600 - 612, Apr. 2004.

[30] L. Merritt et R. Vanam, «Improved Rate Control and Motion Estimation for H.264 Encoder,» *IEEE International Conference on Image Processing,* vol. 5, pp. 309 - 312, Oct. 2007.

[31] J. Garrett-Glaser , Department of Computer Science, Harvey Mudd College, «A novel macroblock-tree algorithm for high-performance optimization of dependent video coding in H.264/AVC,» Date Unknown, Available since 04/29/2011 at. [En ligne]. Available: http://x264.nl/developers/Dark_Shikari/MBtree%20paper.pdf.

[32] P. Karunaratne, C. Segall et A. Katsaggelos, «A rate-distortion optimal video pre-processing algorithm,» *IEEE International Conference on Image Processing,* vol. 1, pp. 481 - 484, Oct. 2001.

[33] H. Kacem, F. Kammoun et M. Bouhlel, «Improvement of the compression JPEG quality by a pre-processing algorithme based on denoising,» *IEEE International Conference on Industrial Technology,* vol. 3, pp. 1319 - 1324, Dec. 2004.

[34] P. van Roosmalen, R. Lagendijk et J. Biemond, «Embedded coring in MPEG video compression,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 12 (Issue: 3), pp. 205 - 211, Mar. 2002.

[35] O. Al-Shaykh et R. Mersereau, «Lossy compression of noisy images,» *IEEE Transactions on Image Processing,* vol. 7 (Issue: 12), pp. 1641 - 1652, Dec. 1998.

[36] P. Van Roosmalen, A. C. Kokaram et J. Biemond, «Noise reduction of image sequences as preprocessing for MPEG2 encoding,» *European Signal Processing Conference,* n° %19, pp. 2253-2256, Sept. 1998.

[37] J. Deng, A. Giladi et F. G. Pancorbo, «Noise Reduction Prefiltering for Video Compression,» *Stanford University,* 2006.

[38] C. Jain et S. Sethuraman, «A low-complexity, motion-robust, spatio-temporally adaptive video de-noiser with in-loop noise estimation,» *IEEE International Conference on Image Processing,* pp. 557 - 560, Oct. 2008.

[39] J. Lee, «Automatic prefilter control by video encoder statistics,» *IET Electronics Letters,* vol. 38 (Issue: 11), pp. 503 - 505, May 2002.

[40] B. Song et K. Chun, «Motion-compensated temporal filtering for denoising in video encoder,» *Electronics Letters,* vol. 40 (Issue: 13), pp. 802 - 804, Jun. 2004.

[41] L. Guo, O. Au, M. Ma et P. Wong, «Integration of Recursive Temporal LMMSE Denoising Filter Into Video Codec,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 20 (Issue: 2), pp. 236 - 249, Feb. 2010.

[42] L. Guo, O. C. Au, M. Ma et Z. Liang, «An Encoder-Embedded Video Denoising Filter Based on the Temporal LMMSE Estimator,» *IEEE International Conference on Multimedia & Expo,* pp. 841 - 844, Jul. 2006.

[43] M. Biloslavo, G. Ramponi, S. Olivieri et L. Albani, «Joint Kalman-based noise filtering and motion compensated video coding for low bit rate videoconferencing,» *International Conference on Image Processing,* vol. 1, pp. 992 - 995, Sep. 2000.

[44] S. Olivieri et L. Albani, «Rate-distortion motion compensated noise filtering for low bit rate video coding,» *International Conference on Image Processing,* vol. 2, pp. 46 - 50, Oct. 1999.

[45] T. Chan, T.-C. Hsung et D.-K. Lun, «Improved MPEG-4 still texture image coding under noisy environment,» *IEEE Transactions on Image Processing,* vol. 12 (Issue: 5), pp. 500 - 508, May 2003.

[46] D. Florencio, «Motion sensitive pre-processing for video,» *IEEE International Conference on Image Processing,* vol. 2, pp. 399 - 402, Oct. 2001.

[47] H. Tsuji, S. Tokumasu, H. Takahashi et M. Nakajima, «Spatial prefiltering scheme based on anisotropic diffusion in low-bitrate video coding,» *Systems and Computers in Japan,* vol. 38 (Issue: 10), pp. 34-45, 2007.

[48] H. Y. S. T. Tsuji, Y. Yashima et N. Kobayashi, «A nonlinear spatio-temporal diffusion and its application to prefiltering in MPEG-4 video coding,» *IEEE International Conference on Image Processing,* vol. 1, pp. 85 - 88, 2002.

[49] A. Uchida et K. Tanaka, «Adaptive prefilter for bit-rate improvement in video compression,» *SPIE Image and Video Communications and Processing,* vol. 5022, pp. 984-993, May 2003.

[50] R. Kawada, A. Koike et Y. Nakajima, «Prefilter Control Scheme for Low bitrate TV Distribution,» *IEEE International Conference on Multimedia and Expo,* pp. 769 - 772, Jul. 2006.

[51] C. A. Segall, P. Karunaratne et A. K. Katsaggelos, «Pre-Processing of compressed digital video,» *SPIE Image and Video Communication And Processing,* 2001.

[52] L. Mao-quan et Zheng-quan, «An Adaptive Preprocessing Algorithm for low Bitrate Video Coding,» *Journal of Zhejiang University,* vol. 7 (Issue: 12), pp. 2057-2062, Dec. 2006.

[53] A. Cavallaro, O. Steiger et T. Ebrahimi, «Perceptual prefiltering for video coding,» *International Symposium on Intelligent Multimedia, Video and Speech Processing,* pp. 510 - 513, Oct. 2004.

[54] M. Bosch, F. Zhu et E. Delp, «Video coding using motion classification,» *IEEE International Conference on Image Processing,* pp. 1588 - 1591, 12-15 Oct. 2008.

[55] Y. Liu, Z. G. Li et Y. C. Soh, «Region-of-Interest Based Resource Allocation for Conversational Video Communication of H.264/AVC,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 18 (Issue: 1), pp. 134 - 139, Jan. 2008.

[56] M. Wang, T. Zhang et S. Goto, «Pre-processor of the region-of-interest based H.264 encoder for low power application,» *IEEE International Conference on ASIC,* pp. 171 - 174, Oct. 2009.

[57] L. Itti, «Automatic foveation for video compression using a neurobiological model of visual attention,» *IEEE Transactions on Image Processing,* pp. 1304 - 1318, Oct. 2004.

[58] X. Yang, W. Lin, Z. Lu et E. Ong, «Just-noticeable-distortion profile with nonlinear additivity model for perceptual masking in color images,» *IEEE International Conference on Acoustics, Speech, and Signal Processing,* vol. 3, pp. 609-12, Apr. 2003.

[59] M. de-Frutos-Lopez, H. Medina-Chanca, S. Sanz-Rodriguez, C. Pelaez-Moreno et F. Diaz-de-Maria, «Perceptually-aware bilateral filtering for quality improvement in low bit rate video coding,» *Picture Coding Symposium,* pp. 477 - 480, May 2012.

[60] N. Young et A. Evans, «Psychovisually tuned attribute operators for pre-processing digital video,» *IEE Proceedings Vision, Image and Signal Processing,* vol. 150, pp. 277-286, Oct. 2003.

[61] H. Kwon, H. Han, S. Lee, W. Choi et B. Kang, «New video enhancement preprocessor using the region-of-interest for the videoconferencing,» *IEEE Transactions on Consumer Electronics,* vol. 56 (Issue: 4), pp. 2644 - 2651, Nov. 2010.

[62] S.-P. Lu et S.-H. Zhang, «Saliency-Based Fidelity Adaptation Preprocessing for Video Coding,» *Journal of Computer Science and Technology,* vol. 26 (Issue: 1), pp. 195-202, Jan. 2011.

[63] X. Yang, W. Lin, Z. Lu, E. Ong et S. Yao, «On incorporating just-noticeable-distortion profile into motion-compensated prediction for video compression,» *IEEE International Conference on Image Processing,* vol. 3, pp. 833-6, Sept. 2003.

[64] X. Yang, W. Lin, Z. Lu, E. Ong et S. Yao, «Perceptually-adaptive pre-processing for motion-compensated residue in video coding,» *IEEE International Conference on Image Processing,* vol. 1, pp. 489 - 492 , Oct. 2004.

[65] X. K. Yang, W. S. Lin, Z. K. Lu, E. P. Ong et S. S. Yao, «Just noticeable distortion model and its applications in vide coding,» *Signal processing. Image communication,* vol. 20 (Issue: 7), p. 662–680, Aug. 2005.

[66] X. Yang, W. Lin, Z. Lu, E. Ong et S. Yao, «Motion-compensated residue preprocessing in video coding based on just-noticeable-distortion profile,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 5 (Issue: 6), pp. 742 - 752, Jun. 2005.

[67] H. Chen, R. Hu et Z. Wang, «Temporal color Just Noticeable Distortion model and its application for video coding,» *IEEE International Conference on Multimedia and Expo,* pp. 713 - 718, 19-23 July 2010.

[68] K.-C. Liu, «DCT-Based Just Noticeable Distorsion for Color Images and its Application to JPEG,» *European Signal Processing Conference,* Oct. 2013.

[69] C.-M. Mak et K. N. Ngan, «Enhancing compression rate by just-noticeable distortion model for H.264/AVC,» *IEEE International Symposium on Circuits and Systems,* pp. 609 - 612, May 2009.

[70] M. Naccari et F. Pereira, «Advanced H,264/AVC-Based Perceptual Video Coding: Architecture, Tools, and Assessment,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 21 (Issue: 6), pp. 766 - 782, Jun. 2011.

[71] M. Naccari et F. Pereira, «Integrating a spatial just noticeable dictorsion model in the under developpement HEVC CODEC,» *IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 817 - 820, 22-27 May 2011.

[72] H. Wu et K. Rao, Digital Video Image Quality and Perceptual Coding, CRC Press, 2006.

[73] A. B. Watson., «Proposal: Measurement of a JND Scale for Video Quality,» *Submission to the IEEE G-2.1.6 Subcommittee on Video Compression Measurements Meeting on August 7th, ,* 2000.

[74] T. Necciari, «Masquage auditif temps-fréquence : Mesures psychoacoustiques et application à l'analyse-synthèse des sons,» *Université de Provence - Aix-Marseille I,* Oct. 2010.

[75] J. P. Egan et H. W. Hake, «On the masking pattern of a simple auditory stimulus,» *The Journal of the Acoustical Society of America,* vol. 22(5), p. 622–630, 1950.

[76] H. Duifhuis, « Consequences of peripheral frequency selectivity for nonsimultaneous masking,» *The Journal of the Acoustical Society of America,* vol. 54(6), p. 1471–1488, 1973.

[77] L. L. Elliott, « Backward and forward masking of probe tones of different frequencies,» *The Journal of the Acoustical Society of America,* vol. 34(8), p. 1116–1117, 1962.

[78] M. Eckert et A. Bradley, «Perceptual quality metrics applied to still image compression,» *Signal Processing 70,* vol. 70 (Issue: 3), p. 177–200, Nov. 1998.

[79] B. Girod, What's wrong with mean-squared error?, A.B. Watson (Ed.), Digital Images and Human Vision, MIT Press, 1993.

[80] C.-H. Chou et C.-W. Chen, «A Perceptually Optimized 3-D Subband Image Codec for Video Communication over Wireless Channels,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 6 (Issue: 2), p. 143–156, Apr. 1996.

[81] A. J. Peterson et A. H. A., «Luminance-model-based dct quantization for color image compression,» *Proc. SPIE Int. Conf. Human Vision, Visual Processing and Digital Display—III,* p. 365–374, 1992.

[82] A. Watson, «DCTune: A technique for visual optimization of DCT Quantization Matrices for Individual Images,» *Society for Information Display(SID) Digest 24,* p. 946–949, 1993.

[83] J. Canny, «A computational approach to edge detection,» *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, Vols. %1 sur %2PAMI-8 (Issue: 6), p. 679–698, Nov. 1986.

[84] Z. Chen et C. Guillemot, «Perceptually-Friendly H.264/AVC Video Coding Based on Foveated Just-Noticeable-Distortion Model,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 20 (Issue: 6), pp. 806 - 819, Jun. 2010.

[85] R. C. Gonzalez et R. E. Woods, Digital Image Processing, 3rd Edition, Prentice Hall, 2008.

[86] A. Buades, B. Coll et J.-M. Morel, «A non-local algorithm for image denoising,» *IEEE Conference on Computer Vision and Pattern Recognition,* vol. 2, pp. 60 - 65, Jun. 2005.

[87] C. Tomasi et R. Manduchi, «Bilateral filtering for gray and color images,» *IEEE Conference on Computer Vision,* pp. 839 - 846, Jan. 1998.

[88] M. Ozkan, M. Sezan et A. Tekalp, «Adaptive Motion-Compensated Filtering of Noisy Image Sequences,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 3 (Issue: 4), pp. 277 - 290, Aug. 1993.

[89] S. ITU-R Rec. BT. 500-9, «ITU-R, Methodology for the Subjective Assessment of the Quality of Television Pictures,» 1999.

[90] Z. Wang et A. Bovik, «Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures,» *IEEE Signal Processing Magazine,* vol. 26 (Issue: 1), pp. 98 - 117, Jan. 2009.

[91] M. Pinson et S. Wolf, «A new standardized method for objectively measuring video quality,» *IEEE Transactions on Broadcasting ,* vol. 50 (Issue: 3), pp. 312 - 322, Sept. 2004.

[92] VQEG, «Final report from the video quality experts group on the validation of objective models of video quality assessment,» March 2000. [En ligne]. Available: http://www.vqeg.org/.

[93] K. Seshadrinathan, R. Soundararajan, A. C. Bovik et L. K. Cormack, «A subjective study to evaluate video quality assessment algorithms,» *Proc. SPIE: Human Vision and Electronic Imaging,* vol. 7527, p. 10, Jan. 2010.

[94] K. Seshadrinathan, R. Soundararajan, A. C. Bovik et L. K. Cormack, «Study of subjective and objective quality assessment of video,» *IEEE Transactions on Image Processing,* vol. 19 (Issue: 6), p. 1427–1441, Jun. 2010.

[95] I.-R. BT.500-11, «Methodology for the SubjectiveAssessment of the Quality of Television Pictures,» 2002.

[96] Z. Wei et K. Ngan, «Spatio-Temporal Just Noticeable Distortion Profile for Grey Scale Image/Video in DCT Domain,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 19 (Issue: 3), pp. 337 - 346, Mar. 2009.

[97] E. Vidal, T. Hauser, P. Corlay et F. Coudoux, «An adaptive video pre-processor based on just-noticeable distortion,» *6th International Symposium on Signal, Image, Video and Communications,* 2012.

[98] NTIA/ITS, «Video Quality Metric (VQM),» [En ligne]. Available: http://www.its.bldrdoc.gov/resources/video-quality-research/request-software.aspx.

[99] N. Benmoussat, M. F. Belbachir et B. Benamar, «Motion estimation and compensation from noisy image sequences: A new filtering scheme,» *Image and Vision Computing,* vol. 25 (Issue: 5), p. 686–694, May 2007.

[100] Sony, «Sony Training,» [En ligne]. Available: https://training.sony-europe.com/?CMP=TRAINING. [Accès le 2014].

[101] M. Mahmoudi et G. Sapiro, «Fast image and video denoising via nonlocal means of similar neighborhoods,» *IEEE Signal Processing Letters,* 2005.

[102] J. Orchard et M. Ebrahimi, «Efficient NonLocal-Means Denoising Using the SVD,» *IEEE Conferences on Image Processing,* 2008.

[103] K. Dabov, A. Foi, V. Katkovnik et K. Egiazarian, «Image denoising by sparse 3D transform-domain collaborative filtering,» *IEEE Transactions on Image Processing,* vol. 16 (Issue: 8) , pp. 2080-2095, Aug. 2007.

[104] I. Butt et N. Rajpoot, «Multilateral filtering: A novel framework for generic similarity-based image denoising,» *IEEE International Conference on Image Processing,* pp. 2981 - 2984, Nov. 2009.

[105] M. Zhang et B. Gunturk, «Multiresolution Bilateral Filtering for Image Denoising,» *IEEE Transactions on Image Processing,* vol. 17 (Issue: 12), pp. 2324 - 2333, Dec. 2008.

[106] L. Shao-Ping et Z. Song-Hai, «Saliency-Based Fidelity Adaptation Preprocessing for Video Coding,» *Journal of Computer Science and Technology,* vol. 26 (Issue: 1), pp. 195-202, 2011.

[107] R. P. ITU-T, «Subjective video quality assessment methods for multimedia applications,» *Recommandation of the ITU Telecommunication standardization sector,* Apr. 2008.

[108] R. Hassen, Z. Wang et M. Salama, «Image Sharpness Assessment Based on Local Phase Coherence,» *IEEE Transactions on Image Processing,* vol. 22 (Issue: 7), pp. 2798 - 2810, Jul.2013.

[109] P. Marziliano, F. Dufaux, S. Winkler et T. Ebrahimi, «A no-reference perceptual blur metric,» *International Conference on Image Processing,* vol. 3, pp. 57-60, 2002.

[110] P. Marziliano, F. Dufaux, S. Winkler et T. Ebrahimi, «Perceptual blur and ringing metrics: Application to JPEG2000,» *Signal Processing: Image Communication,* vol. 19 (Issue: 2), p. 163–172, Feb. 2004.

[111] H. R. Wu, A. Reibman, W. Lin et F. Pereira, «Perceptual Visual Signal Compression and Transmission,» *Proceedings of the IEEE,* vol. 101 (Issue: 9), pp. 2025 - 2043, Sept. 2013.

[112] Z. Lia, S. Qina et L. Ittib, «Visual attention guided bit allocation in video compression,» *Image and Vision Computing,* vol. 29 (Issue: 1), p. 1–14, Jan. 2011.

[113] Y. Bandoh, K. Hayase, S. Takamura et K. Kamikura, «Encoder design for H.264/AVC based on contrast sensitivity considering spatio-temporal direction dependency,» *IEEE International Conference on Image Processing,* pp. 2124 - 2127, 12-15 Oct. 2008.

[114] R. A. Z. W. M. S. Shiqi Wang, «SSIM-Motivated Rate-Distortion Optimization for Video Coding,» *IEEE Transactions on Circuits and Systems for Video Technology ,* Vols. %1 sur %222, Issue 4, pp. 516 - 529, 2011.

[115] Y.-H. Huang, T.-S. Ou, P.-Y. Su et H. Chen, «Perceptual Rate-Distortion Optimization Using Structural Similarity Index as Quality Metric,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 20 (Issue: 11), pp. 1614 - 1624, Nov. 2010.

[116] X. Yang, W. Lin, Z. Lu et X. Lin, «Rate control for videophone using local perceptual cues,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 15 (Issue: 4), pp. 496 - 507, Apr. 2005.

[117] J. Albert J. Ahumada et H. A. Peterson, «Luminance-model-based DCT quantization for color image compression,» *Human Vision, Visual Processing, and Digital Display III,* vol. 1666, Feb. 1992.

[118] C.-W. Tang, «Spatiotemporal Visual Considerations for Video Coding,» *IEEE Transactions on Multimedia,* vol. 9 (Issue: 2), pp. 231 - 238, Feb. 2007.

[119] C.-W. Tang, C.-H. Chen, Y.-H. Yu et C.-J. Tsai, «Visual sensitivity guided bit allocation for video coding,» *IEEE Transactions on Multimedia,* vol. 8 (Issue: 1), pp. 11 - 18, Feb. 2006.

[120] G.-G. J., «Diary Of An x264 Developer,» [En ligne]. Available: http://x264dev.multimedia.cx/archives/377. [Accès le 2013].

[121] K. Software. [En ligne]. Available: http://iphome.hhi.de/suehring/.

[122] Tektronix, «Picture Quality Analyzers,» 2014. [En ligne]. Available: http://www.tek.com/picture-quality-analyzer.

[123] J. Lubin, «A human vision system model for objective picture quality measurements,» *International Broadcasting Convention ,* pp. 498 - 503, Sep. 1997.