Etude et optimisation d'un système de vidéotransmission conjoint Source-Canal basé " SoftCast "

Anthony Trioux

N° d'ordre : 1942

Thèse de doctorat

présentée en vue d'obtenir le grade de Docteur de

l'UNIVERSITÉ POLYTECHNIQUE HAUTS-DE-FRANCE

Discipline : Électronique, optronique et systèmes

Spécialité : Électronique, Télécommunication

Présentée et soutenue par

Anthony TRIOUX

Titre de la thèse :

Etude et optimisation d'un système de vidéotransmission conjoint Source-Canal basé "SoftCast"

Study and optimization of a Joint Source-Channel video-transmission system based on "SoftCast"

Soutenue le 10, Décembre 2019 à Valenciennes, devant le jury d'examen :

Président	M. Marco CAGNAZZO	Professeur, Télécom ParisTech
Rapporteur	Mme. Anissa MOKRAOUI	Professeur, Univ. Paris 13
Rapporteur	M. Yannis POUSSET	Professeur, Univ. Poitiers
Examinatrice	Mme. Anne Sophie DESCAMPS	MCF, Univ. Nantes
Examinateur	M. Giuseppe VALENZISE	Chargé de recherche CNRS, Centrale Supélec
Membre Invité	M. Michel KIEFFER	Professeur, Centrale Supélec
Directeur de thèse	M. François-Xavier COUDOUX	Professeur, UPHF
Co-directeur de thèse	M. Patrick CORLAY	Professeur, UPHF
Co-encadrant de thèse	M. Mohamed GHARBI	Maître de conférences, UPHF

Équipe de recherche, Laboratoire : Institut d'Électronique de Microélectronique et de Nanotechnologie,
Département Opto-Acousto-Électronique (IEMN DOAE-UMR 8520)
École doctorale : Sciences pour l'ingénieur (ED SPI 072)

"The mind is the limit. As long as the mind can envision the fact that you can do something, you can do it, as long as you really believe 100 percent."

Arnold Schwarzenegger

"移大山始于运小石。"

孔子

"I'm always looking for a new challenge. There are a lot of mountains to climb out there. When I run out of mountains, I'll build a new one."

Sylvester Stallone

"Early to bed, early to rise, work like hell, and advertise."

Ted Turner

Remerciements

Ces travaux de thèse ont été réalisé au sein de l'IEMN-DOAE UMR8520 (Institut d'Electronique, de Microélectronique et Nanotechnologie) de l'Université Polytechnique Hauts-de-France avec le soutien financier de la Région Hauts-de-France.

Je tiens en premier lieu à remercier mes directeurs de thèse M. François-Xavier COUDOUX et M. Patrick CORLAY de m'avoir fait confiance sur un nouveau sujet qui m'a permis d'évoluer sur différentes pistes enrichissantes. Je tiens également à vous remercier pour la confiance que vous m'avez accordée, me permettant de mener ma recherche avec une certaine autonomie tout en me donnant de précieux conseils lorsque j'en avais besoin ! Mes remerciements s'adressent également à mon encadrant M. Mohamed GHARBI, sans qui les parties théoriques n'auraient pas pu être aussi développées. Rares sont les personnes qui possèdent autant de connaissances et qui continuent de toujours s'intéresser à de nouvelles choses tout en en faisant profiter les autres. D'une manière générale, je tiens à vous remercier tous pour vos disponibilités respectives, votre aide et vos précieux conseils. Je sais que vous avez notamment passé des soirées et des week-ends sur mes travaux de thèse pour me permettre de soutenir dans les temps et pour ça je vous en suis très reconnaissant.

Ma profonde gratitude s'adresse également aux membres du jury de thèse. Tout d'abord je tiens à remercier Mme Anissa MOKRAOUI et M. Yannis POUSSET d'avoir accepté de rapporter ces travaux de thèse dans un temps assez restreint. Je vous remercie pour les précieux conseils et avis que vous avez formulé sur ce manuscrit. Je tiens également à remercier les examinateurs : Mme Anne-Sophie DESCAMPS, M. Marco CAGNAZZO, M. Giuseppe VALENZISE et M. Michel KIEFFER pour avoir accepté de faire partie de ce jury.

Ces travaux de thèse n'auraient pas pu être aussi diversifiés sans l'aide de M. CAGNAZZO, M. KIEFFER et M. VALENZISE avec qui j'ai eu le plaisir de travailler au cours d'un séjour de recherche lié à la qualité subjective SoftCast et financé par le GdR-ISIS (Mme Barbara NICOLAS et Mme Audrey GIREMUS). Je tiens à toutes et

tous vous remercier de m'avoir donné l'opportunité d'effectuer ce séjour. Aussi, je souhaite exprimer ma gratitude à l'ensemble des personnes qui ont accepté de participer aux tests subjectifs, ainsi qu'à M. Emin ZERMAN (pour l'aide sur l'interface de test et l'analyse des données), M. Frédéric RIVART, M. Sébastien GRAPTIN et M. François COMPS (pour la mise en place de la salle de test à Valenciennes).

Du point de vue de ma formation d'enseignant, je tiens à remercier M. David BOULINGUEZ et M. Christophe KRZEMINSKI, Mme Audrey DUBRULLE et M. GAZALET de m'avoir permis d'effectuer des enseignements respectivement à l'ISEN-YNCREA et à l'IUT de Valenciennes. Aussi, je tiens à remercier l'ensemble des enseignants chercheurs du département Electronique de Centrale Lille Institut et plus particulièrement M. Oliver BOU MATAR et M. Abdelkrim TALBI de m'avoir permis d'effectuer un ATER avec eux tout en aménageant mon emploi du temps pour la finalisation de ces travaux. Mes remerciements s'adressent également à l'ensemble des enseignants de l'UPHF (IUT, ISTV) qui m'ont permis d'arriver jusqu'ici et plus particulièrement mes encadrants de projet : M. Fabrice ROBERT, M. Michael BOCQUET et M. Yassin EL HILLALI. Enfin, je tiens également à remercier mon professeur de lycée M. Denis MASURE qui m'a donné goût à l'enseignement et surtout qui m'a tout appris en matière d'électronique et de CAO !

Je remercie également l'ensemble de mes collègues doctorants. J'adresse une pensée particulière à mes ami(e)s de bureaux : Salah, Hala, Bowei, Hassan et Hatem ainsi qu'à mes compagnons de galères : Aymen et Vivien ;).

Enfin, je ne peux terminer cette partie sans remercier les personnes qui m'entourent dans la vie de tous les jours. Tout d'abord mes amis : Steve, Alex, Steven, Benjamin, Cédric, Sylvain, 国玺 et 思琛 qui ont toujours été présents durant ces trois années de thèse et même avant ;). Je remercie également ma famille pour l'aide et le soutien durant mes nombreuses années d'études. Enfin, ces dernières lignes s'adressent à ma petite amie 袁媛 qui bien que spécialiste des mathématiques appliquées et de la logistique a beaucoup appris sur le schéma SoftCast et le traitement vidéo en général ;). Nous avons avancé ensemble dans la grande aventure que représente le Doctorat et je tiens à te remercier pour ton soutien, ton accompagnement sans faille et dernièrement surtout pour le fait d'avoir été toujours compréhensive même quand je devais me lever très tôt le matin pour finir mes travaux. Merci également pour ton aide précieuse lors des relectures d'articles et lors de la finalisation de ce manuscrit de thèse. 亲爱的，非常感谢你一直在我的身旁鼓励我，咱俩一起成为博士了，比心。

Table des matières

Table des figures

Liste des tableaux

Glossaire

ACC-JPEG2000 - Accordion-JPEG2000. Version alternative à JPEG2000 reposant sur une transformée en accordéon [75]
A-HDAVT - Adaptive Hybrid Digital–Analog Video Transmission
AGCC - Adaptive GoP-size based on Content and Cut detection
AGCUT - Adaptive GoP-size based on Cut detection
AIM - Additive Impairment Measure
H.264/AVC - Norme de compression vidéo également appelée MPEG-4 Part 10 ou Advanced Video Coding
AWGN - Additive White Gaussian Noise
BM3D - Block matching and 3D filtering
BPSK - Binary Phase-Shift Keying Modulation
CB - Constrained-Bandwidth
CCSC - Codage Conjoint Source-Canal voir JSCC
CDF - Cumulative Distribution Function
CIF - Format vidéo avec une largeur d'image de 352 pixels et une hauteur de 288 pixels
CNN - Réseau neuronal convolutif (Convolutional neural network)
CPL - Courant Porteur en Ligne
CR - Compression Ratio
CS - Compressive-Sensing
CSI - Channel State Information
CSF - Contrast Sensitivity Function
CSNR - Channel Signal to Noise Ratio
CVL - Codage Vidéo Linéaire
DC-2D (coefficient) Coefficient DCT X(0,0) de rang fréquentiel nul. Proportionnel à la valeur moyenne de l'image dans le domaine pixel

DC-3D (coefficient) Coefficient DCT X(0,0,0) de rang fréquentiel nul Proportionnel à la valeur moyenne des coefficients DC-2D dans le domaine DCT-2D
DCSFCN - Deep Compressed Sensing Fully Connected Network
DCSRN - Deep Compressed Sensing Residual Neural Network
DCT - Discrete Cosine Transform, Transformée en cosinus discrète
DLM - Detail Loss Metric
DPCM - Differential Pulse Code Modulation
DSC - Distributed Source Coding
DSIS - Double Stimulus Impairment Scale
DSL - Digital Subscriber Line
DVB-T - Digital Video Broadcasting – Terrestrial
DWT - Discrete Wavelet Transform
ETSI - European Telecommunications Standards Institute
FBMP - Fast Bayesian Matching Pursuit
FBW - Full BandWidth
FDFR - Full Decode Full Recode
FEC - Codes correcteurs d'erreurs (FEC : Forward Error Correction code)
GMRF - Gaussian Markov Random Field
GoP - Group of Pictures
GT - Gradient Transform, Transformée en Gradient
HD - Haute Définition
HDA - Hybrid Digital Analog
HEVC - High Efficiency Video Coding Standard de compression vidéo normalisé par le JCT-VC au début de l'année 2013. Appelé également H.265
HR - Haute Résolution
HVS - Système Visuel Humain, Human Vision System
INE - Impulse Noise Estimation
INM - Impulse Noise Mitigation
ITU - International Telecommunication Union
JCT-VC - Joint Collaborative Team on Video Coding
JND - Just Noticeable Difference
JSCC - Joint Source-Channel Coding
KLT - Karhunen-Loeve transform
KMV - Knowledge-Enhanced Mobile Video

LVC - Linear Video Coding (scheme)
LLSE - Linear Least Square Error (estimateur)
MC - Motion Compensation, compensation de mouvement.
MCTF - Filtrage temporel à compensation de mouvement (MCTF)
MDCT - Modified Discrete Cosine Transform
ME - Motion Estimation, estimation de mouvement
MIMO - Multiple Input Multiple Output
MOS - Mean Opinion Score
MPEG - Moving Picture Experts Group
MU-MIMO - Multi User (MIMO)
MVD - Multi-View-plus-Depth, applications multivues et multivues avec profondeur (MVD)
NIC - No Impulse noise Correction
OFDM - Orthogonal Frequency Division Multiplexing
OPA - Optimal Power Allocation(SoftCast+)/Optimized Power Allocation (OPA-SoftCast)
OSP - Optimal Subchannel Provisioning
PAISP - Power Allocation with Inferred Split Position
PALPA - Power Allocation with Local Power Adjustment
PCC - Pearson Correlation Coefficient
PDO - Power Distortion Optimization
PLT - Power Line Transmission
PSNR - Peak Signal to Noise Ratio
PWC - PairWise Comparison
QAM - Modulation d'amplitude en quadrature
QPSK - Quadrature Phase-Shift Keying Modulation
RGPD - Règlement Général sur la Protection des Données
RMSE - Root Mean Squared Error
ROI - Region of lnterest
SCS - Simple Chunk Scaling
SDR - Software Defined Radio
SI - Spatial Information
SISO - Single Input Single Output
SKM - Shannon-Kotel'Nikov Mapping

SROCC - Spearman Rank-Order Correlation Coefficient
SSIM - Structural SiMilarity, mesure objective de qualité avec référence basé sur la structure de l'image
SVC - Scalable Video Coding
SVH - Système Visuel Humain (acronyme français de HVS)
SVM - Support Vector Machine
TCP - Transmission Control Protocol
TI - Temporal Information
UEP - Unequal Error Protection
ULB - Ultra Large Bande
UIT - Union Internationale des Télécommunication
USRP - Universal Software Radio Peripheral
VIF - Visual Information Fidelity
VMAF - Video Multimethod Assessment Fusion
VQEG - Video Quality Experts Group
WCVC - Wireless Cooperative Video Coding
WSVC - Wireless Scalable Video Coding
WVGA - Wide Video Graphics Array
ZF - Zero-Forcing (estimateur)

Introduction

La transmission de vidéos vers et depuis les utilisateurs mobiles est un service en plein essor. Les opérateurs réseaux prévoient que la vidéo représentera 82% de tout le trafic de données dans les années à venir [50]. Par conséquent, un effort de recherche considérable est consacré à la conception de systèmes de transmission permettant à chaque récepteur d'obtenir la meilleure qualité vidéo. Cependant, ceci est particulièrement difficile lorsque les caractéristiques du canal de transmission changent au fil du temps. Ceci est d'autant plus vrai si l'on considère les transmissions broadcast (diffusion de l'information à plusieurs récepteurs) où chaque récepteur possède son propre canal de qualité différente, notamment selon la distance de l'émetteur.

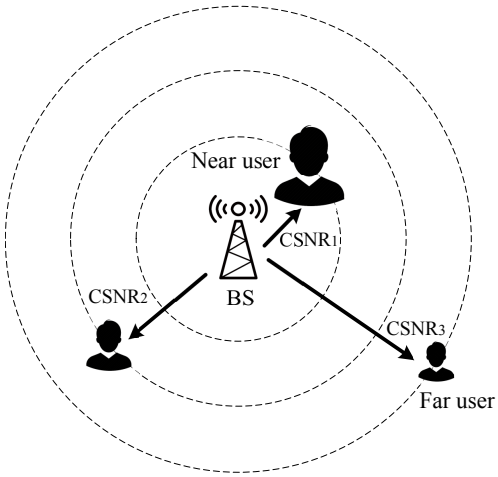


FIGURE 1 : Illustration d'une transmission vidéo dans un contexte sans-fil.

Cette transmission de contenu vidéos peut s'effectuer en utilisant des protocoles sans-fil tels que le protocole IEEE 802.11/Wifi, 4G/LTE, etc. ou via des réseaux filaires tels que les CPL (Courant Porteur en Ligne) [128, 129] comme illustré en Fig. 1 et Fig. 2. Dans le premier cas, un contenu vidéo est transmis via une connexion sans-fil à plusieurs utilisateurs plus ou moins éloignés de la station de base/d'émission (BS). Dans le second cas, un serveur multimédia transmet le contenu vidéo à plusieurs endroits/récepteurs reliés via CPL au sein d'une maison.

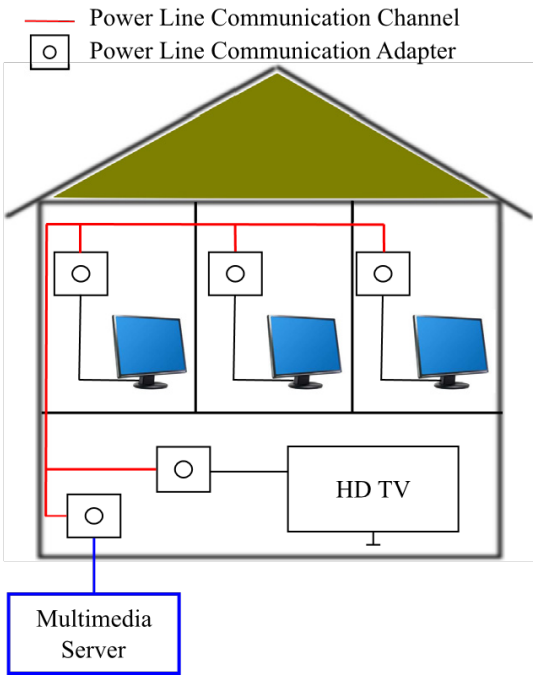


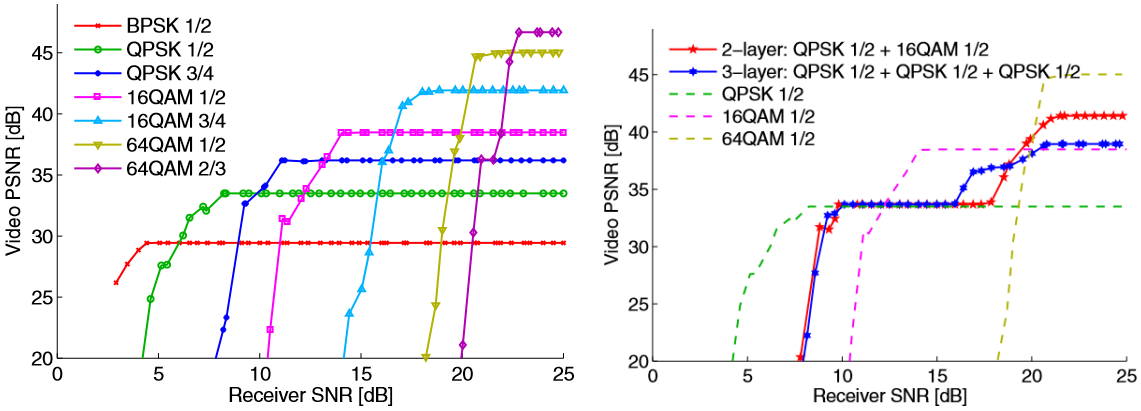
FIGURE 2 : Illustration d'une transmission vidéo dans un contexte CPL. Figure issue de [129].

Actuellement, les solutions de transmissions vidéo conventionnelles reposent sur une séparation source/canal où le codage source est effectué via les standards de compression vidéo MPEG-2, H.264/AVC [82] ou HEVC [93]. Les paramètres de codage canal et de transmission (type de modulation et niveau de protection utilisé) représentés par le MCS (Modulation and Coding Scheme) sont ajustés généralement en fonction d'une estimation du canal de transmission. Néanmoins, cette approche pose deux problèmes :

1. Tout d'abord, elle nécessite une adaptation permanente des paramètres de codage source/canal du flux envoyé par l'émetteur selon les fluctuations du canal ;
2. Ensuite, dans un contexte de diffusion, en raison de l'hétérogénéité des canaux des utilisateurs (e.g. le récepteur proche de la station d'émission dispose d'un bon canal alors que le récepteur éloigné dispose d'un mauvais canal), les transmissions sont dimensionnées pour un canal typique "intermédiaire". Les récepteurs dont les conditions de canal sont dégradées seront soumis à des perturbations visuelles importantes (e.g. gel d'image) alors que les récepteurs disposant d'un meilleur canal que celui initialement prévu ne pourront pas en tirer pleinement parti.

Il en résulte deux phénomènes bien connus qui sont le *cliff-effect* [57] et l'effet de saturation de la qualité (*levelling-off*) [64]. Ceux-ci sont illustrés dans la Fig. 3a. Le premier, fait référence à une perte soudaine et brutale de la qualité vidéo reçue. Par exemple, si le flux envoyé est prévu pour un CSNR (ou Receiver SNR) de 20dB, la modulation 64QAM avec un code de

protection 1/2 est employée (jaune) selon le standard Wifi 802.11a, ce qui permet d'obtenir une très bonne qualité vidéo exprimée ici en termes de PSNR (le flux vidéo est très peu compressé). Toutefois, si la qualité du canal du récepteur chute en dessous de 20dB, alors de graves erreurs de décodage surviennent empêchant le récepteur de visualiser correctement le flux vidéo. Ceci se traduit par une chute brutale du PSNR. Le second effet, fait référence au fait que la qualité reçue reste presque constante même si la qualité du canal de transmission (ici mesurée par le CSNR) augmente. Sur le même exemple, nous pouvons voir que même si la qualité du canal du récepteur augmente au-delà de 20dB, la qualité vidéo reçue (mesurée par le PSNR) restera constante du fait de la compression effectuée au niveau de l'émetteur.



(a) (b)

FIGURE 3 : Evolution de la qualité vidéo reçue (PSNR) en fonction de la qualité du canal de transmission (CSNR) et du MCS utilisé. a) Codec H.264/AVC. b) Codec H.264/SVC. Figure issue de [52].

Le codage vidéo scalable (H.264/SVC) [85] peut en partie réduire le problème de *cliff-effect*, sans pour autant l'éliminer complètement comme illustré dans la Fig. 3b. Dans ce cas, la qualité vidéo diminue par "paliers", le nombre de paliers étant directement fonction du nombre de couches d'amélioration du flux scalable. De plus, il est bien connu que l'efficacité de codage de cette solution est limité en raison du surdébit(overhead) engendré par le nombre de couches utilisées [104].

Parmi les applications récentes de la transmission vidéo sans-fil, nous pouvons citer les réseaux véhiculaires (VANET : Vehicular Adhoc NETworks) où la vidéo n'est pas seulement utilisée dans un contexte de divertissement mais aussi afin d'améliorer le confort de l'utilisateur (conduite supervisée/autonome), la sécurité (vidéosurveillance), etc. via les systèmes de transports intelligents (ITS). Nous illustrons en Fig. 4 un exemple proposé par Valeo, dénommé *XtraVue* (https://www.valeo.us/en/world-premiere-at-ces-2019-of-valeo-xtravue-trailer-the-invisible-trailer-system/). Ce dernier permet à l'utilisateur de rendre visibles des informations provenant de l'avant et/ou de l'arrière du véhicule qui ne le sont pas en temps normal (e.g. un camion devant la voiture ou encore une caravane

à l'arrière du véhicule occultant les informations de la route). Dans de tels cas et surtout dans le cas d'un véhicule autonome, il est crucial de supprimer le cliff-effect qui peut entraîner de graves accidents.

FIGURE 4 : Illustration du système *Xtravue* proposé par Valeo.

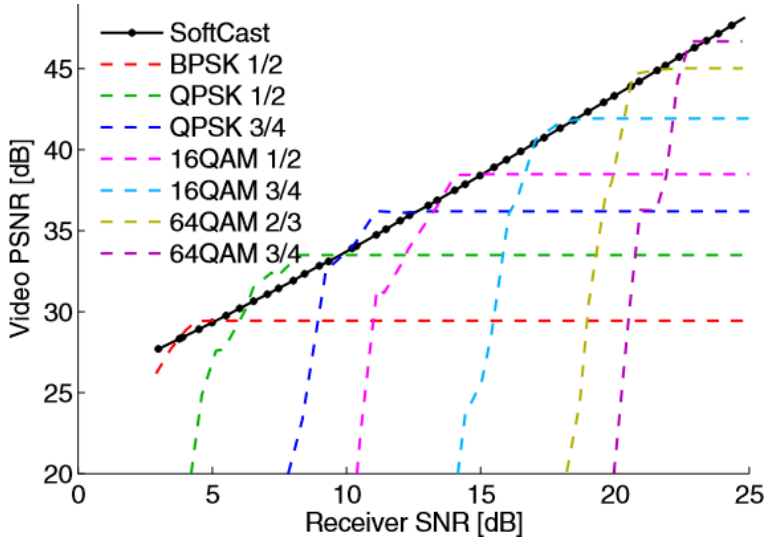
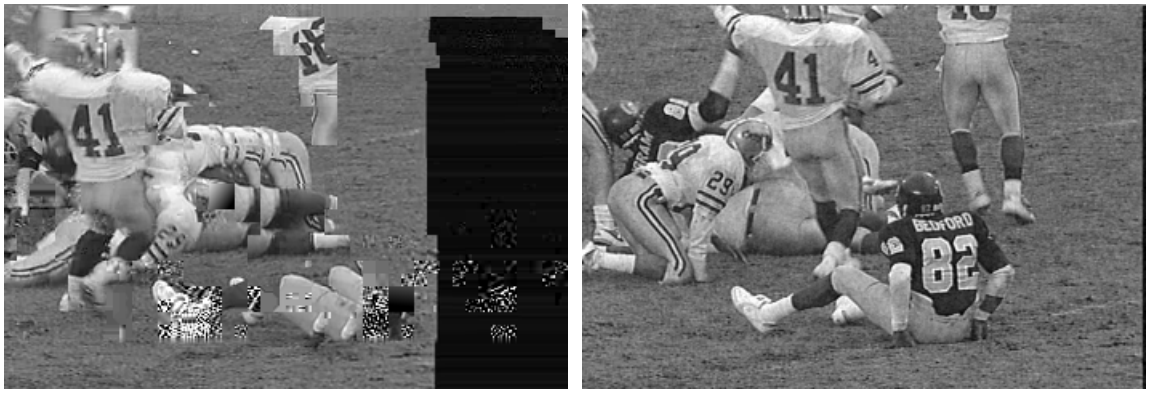


FIGURE 5 : Evolution de la qualité vidéo reçue (PSNR) en fonction de la qualité du canal de transmission (CSNR) pour SoftCast. Figure issue de [52].

C'est dans ce but que des systèmes de codage vidéo linéaire (CVL) ont été proposés récemment afin de pallier les limitations des systèmes de codage de source et de canal traditionnels. SoftCast [54] représente le pionnier des systèmes de CVL. Les pixels sont traités par des opérations linéaires successives (transformée DCT, allocation de puissance, modulation analogique) et directement transmis sans quantification ni codage (entropique ou de canal). Ceci permet d'obtenir une qualité vidéo reçue qui augmente proportionnellement avec la qualité du canal de transmission comme illustré dans la Fig. 5, sans aucune information

de retour sur l'état du canal et tout en évitant les mécanismes d'adaptation complexes des schémas classiques.

D'un point de vue perceptuel, le *cliff-effect* sur la vidéo obtenue avec MPEG-4 part 10 (H.264/AVC) et sa version équivalente obtenue avec SoftCast (sans *cliff-effect*) sont présentés en Fig. 6. Comme nous pouvons le voir, le standard conventionnel H.264/AVC souffre de graves erreurs de décodage. Au contraire, SoftCast continue à décoder et offre un niveau de qualité vidéo tout à fait acceptable.



(a) (b)

FIGURE 6 : Exemple de comparaison visuelle entre SoftCast et H.264/AVC. a) Codec H.264/AVC. b) SoftCast. Vidéo complète disponible dans le lien http://people.csail.mit.edu/szym/softcast/videos.html.

Compte tenu des propriétés intéressantes de SoftCast, et plus généralement des CVL, nous nous sommes naturellement tournés vers ceux-ci dans le cadre de cette thèse. Ainsi, nous avons proposé plusieurs améliorations du schéma de codage vidéo linéaire SoftCast [54], notamment d'un point de vue perceptuel. Le manuscrit est organisé comme suit, et comporte les contributions suivantes :

- Dans le premier chapitre, un état de l'art des codeurs vidéo linéaires est présenté. Tout d'abord le schéma général de SoftCast est introduit, puis plusieurs variantes de SoftCast sont présentées en considérant à la fois la partie codage source et la partie transmission.
- Dans le Chapitre 2, nous considérons la modélisation de l'évolution de la qualité vidéo reçue (de bout en bout) dans un contexte SoftCast. Plusieurs paramètres sont pris en compte avec notamment, la considération de la bande passante disponible au niveau de l'application, l'allocation de puissance effectuée à l'émetteur et le type d'estimateur utilisé au récepteur. A chaque fois, un modèle théorique est proposé. Les résultats obtenus montrent que les modèles proposés illustrent parfaitement les résultats obtenus

en simulation. Ils permettent dès lors de quantifier précisément l'apport des différents schémas proposés.

- Dans le Chapitre 3, nous proposons une étude complète et originale des différents artefacts visuels pouvant apparaître dans un contexte de transmission vidéo ayant recours à un CVL. En effet, la compression étant réalisée de manière très différente d'un schéma de codage classique par blocs, les artefacts visuels des CVL sont alors assez différents de ceux habituellement observés et méritent donc d'être étudiés. Nous évaluons ensuite à l'aide de tests subjectifs le ressenti global des utilisateurs quant à la qualité visuelle de contenus vidéos reçus via SoftCast puis celui lié au type d'estimateur utilisé à la réception. En outre, une évaluation des performances des métriques objectives telles que le PSNR, SSIM ou encore VMAF est proposée grâce aux scores subjectifs obtenus.
- Dans le Chapitre 4, nous introduisons et analysons des méthodes de prétraitement existantes améliorant significativement la qualité en réception. Une approche originale est ensuite proposée dans laquelle des performances similaires à l'état de l'art [40] sont obtenues (gain en PSNR de l'ordre de 3 dB) tout en réduisant de moitié le temps de calcul nécessaire au prétraitement ainsi que le volume des informations additionnelles à transmettre (réduction de 75%).
- Finalement, dans le Chapitre 5, un encodage adaptatif est proposé permettant de supprimer et/ou d'atténuer des artefacts gênants du codeur SoftCast aux niveaux des changements de scène, tout en proposant le meilleur compromis entre amélioration de la qualité reçue et complexité calculatoire. Ainsi, une amélioration du PSNR allant jusqu'à 16 dB et jusqu'à 0.55 pour l'index SSIM est observée avec la méthode proposée spécifiquement aux frontières des changements de scène (cuts vidéo). Ces méthodes permettent également de réduire les fluctuations temporelles de qualité visuelle en dessous de 1 dB en moyenne, démontrant ainsi leur efficacité.

Chapitre 1

Etat de l'art des schémas de codage vidéo linéaire

Sommaire

1.1 Introduction

Différent des standards de transmission vidéo actuels, SoftCast est un système de Codage Conjoint Source-Canal ou Joint Source-Channel Coding scheme (CCSC/JSCC) où les pixels sont traités par des opérations linéaires successives et directement transmis sans quantification ni codage. SoftCast représente le pionnier des architectures dites de codage vidéo linéaire ou Linear Video Coding (CVL/LVC), encore appelé transmission vidéo soft (Soft Video Delivery) ou transmission vidéo non codées (uncoded video transmission). Toutes ces dénominations font référence à des schémas basés SoftCast. Pour faciliter la lecture de ce travail, la dénomination codage vidéo linéaire est retenue parmi les autres. Dans la première partie de ce chapitre, le schéma SoftCast est introduit. Puis, différentes extensions de SoftCast sont présentées, qui modifient les parties codage source vidéo ou codage canal / transmission du schéma originel de codage vidéo linéaire.

1.2 Vue d'ensemble du schéma SoftCast

Le schéma d'ensemble de SoftCast [54] est donné en Fig. 1.1. La partie supérieure correspond à l'émetteur SoftCast, tandis que la partie inférieure représente le récepteur associé. Dans SoftCast, le canal est traditionnellement modélisé par un bruit additif blanc gaussien de moyenne nulle et de variance σ^2 (AWGN : Additive White Gaussian Noise). Les blocs qui composent SoftCast sont introduits dans la suite de cette partie.

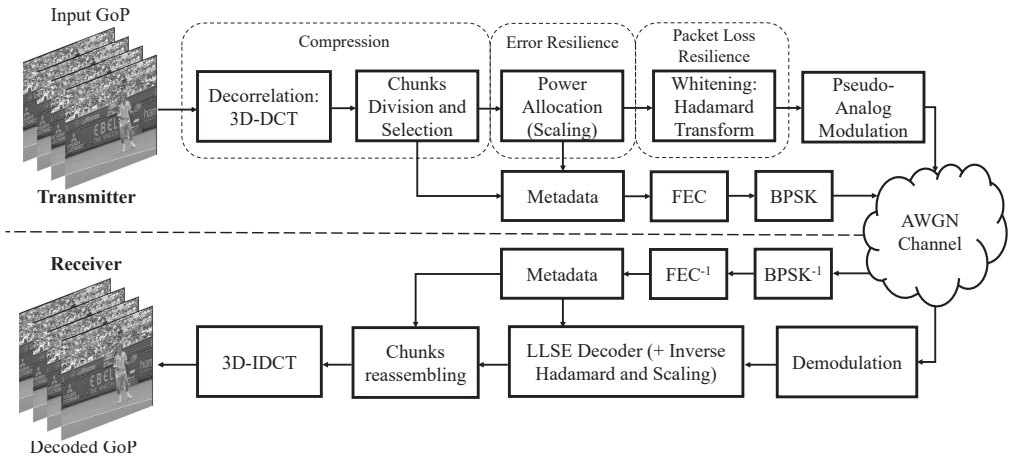


FIGURE 1.1 : Diagramme bloc du schéma de transmission vidéo SoftCast.

1.2.1 Compression

SoftCast transforme tout d'abord des groupes d'images (GoP) constitués à partir des images de la séquence vidéo via une DCT à trois dimensions. La DCT-3D est réalisée en exploitant le principe de séparabilité de la transformée DCT, i.e., le schéma transforme d'abord chaque image via une DCT-2D spatiale puis effectue une DCT-1D temporelle sur le GoP comme indiqué dans la Fig. 1.2.

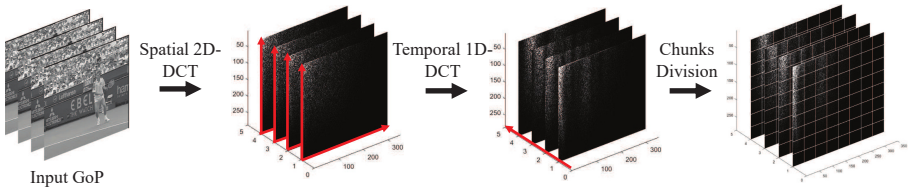


FIGURE 1.2 : Etape de compression dans SoftCast. De gauche à droite : GoP dans le domaine pixel, images transformées après DCT-2D, images transformées après DCT-3D, division en chunks.

La DCT-3D [77] d'un GoP $f(i,j,k)$ est dénotée par $\mathbf{F}(u,v,w)$ et définie comme :

$$\mathbf{F}(u,v,w) = \sum_{i=0}^{N_R-1} \sum_{j=0}^{N_C-1} \sum_{k=0}^{N_F-1} f(i,j,k) \cdot C_{i,u} \cdot C_{j,v} \cdot C_{k,w} \tag{1.1}$$

où les entiers N_R, N_C dénotent la taille de l'image et N_F la taille du GoP, et :

$$C_{p,q} = \begin{cases} \frac{1}{\sqrt{Z}}, q = 0, \\ \frac{2}{\sqrt{Z}} \cdot \cos(\frac{(2p+1)q\pi}{2Z}), \text{autrement} \end{cases} \tag{1.2}$$

avec Z égal à N_R, N_C ou N_F, selon le $C_{p,q}$ sélectionné.

A l'issue de la DCT-3D, l'énergie est compactée dans le coin supérieur gauche des premiers plans du GoP transformé. Les R plans DCT résultants sont ensuite divisés en blocs rectangulaires de coefficients transformés appelés *chunks* et réarrangés pour former une nouvelle matrice où chaque ligne contient un chunk. Ces chunks sont ordonnés par ordre décroissant d'énergie. L'étape de compression dans SoftCast peut être réalisée à ce niveau, où certains chunks sont supprimés afin de répondre aux limitations de la bande passante disponible. Ce nombre de chunks est fixé par la bande passante disponible. La procédure pour déterminer le nombre de chunks (dépendant de la bande passante disponible) est décrit dans la Section 1.2.4.

Dans les codeurs vidéo linéaires, le taux de compression noté ici CR est défini de la manière suivante [62] :

$$\text{CR} = \frac{K}{N} \tag{1.3}$$

où K représente le nombre de chunks transmis par GoP et N le nombre total de chunks au sein d'un GoP. Ce ratio est compris entre 0 (aucune information transmise) et 1 (pas de compression).

Le nombre total de chunks N est défini de la manière suivante :

$$N = \frac{N_R \cdot N_C \cdot N_F}{nb_r \cdot nb_c}, [chunks/GoP] \tag{1.4}$$

où nb_r, nb_c représentent la taille du chunk et N_F réfère au nombre d'images au sein d'un GoP.

1.2.2 Résistance aux erreurs

L'opération suivante consiste en une allocation de puissance (PDO : Power Distortion Optimization), ou mise à l'échelle des coefficients transformés (scaling) qui garantit une protection des données vidéo contre le bruit des canaux. Etant donné que la puissance à l'émission P est limitée et fixée, elle est distribuée aux chunks de telle manière à réduire l'erreur quadratique moyenne à la réception (MSE). Cette répartition est un problème typiquement lagrangien et la solution quasi-optimale est donnée par :

$$g_i = \lambda_i^{-1/4} \cdot \sqrt{\frac{P}{\sum_j \sqrt{\lambda_j}}} = \sqrt{\frac{P}{\sqrt{\lambda_i} \sum_j \sqrt{\lambda_j}}} \tag{1.5}$$

où $g_i, i = 1, 2, \ldots, K$ est le coefficient de scaling pour le $i^{ème}$ chunk, et $\lambda_i = E[\mathbf{X_i}^2]$ est l'énergie du $i^{ème}$ chunk $\mathbf{X_i}$ [111]. La démonstration est disponible en Annexe A.

Si l'émetteur possède une estimation du bruit du canal, une allocation de puissance optimale peut être définie où les $N - \ell$ chunks ($0 \leq \ell < N$) ayant une énergie proportionnellement inférieure à l'énergie du bruit du canal sont supprimés. Ce schéma est dénoté par SoftCast+ [13] et la solution [58, 107, 129] est donnée par :

$$g_m = \frac{\left(\sqrt{\frac{\lambda_m \sigma_n^2}{\gamma}} - \sigma_n^2\right)^{1/2}}{\sqrt{\lambda_m}}, \tag{1.6}$$

où $g_m, m = 1, \ldots, \ell$ est le coefficient de scaling optimal du $m^{ème}$ chunk, et avec

$$\sqrt{\gamma} = \frac{{\sigma_n}^2 \sum_{m=1}^{\ell} \sqrt{\lambda_m}}{P + \ell {\sigma_n}^2}. \tag{1.7}$$

Les chunks mis à l'échelle sont définis par $\mathbf{U}_i[j] = g_i \mathbf{X}_i[j]$, $j = 1, 2, \ldots, nb_r \cdot\ nb_c$ où $\mathbf{X}_i[j]$ représente le $j^{ème}$ coefficient DCT du chunk i avec $i = 1, 2, \ldots, K$.

Nous notons que seulement un facteur d'échelle par chunk est calculé dans SoftCast. Ceci résulte d'un compromis entre qualité reçue, quantité de métadonnées et coût de calcul [54]. Des alternatives issues de la littérature seront présentées dans la suite de ce chapitre.

1.2.3 Résistance aux paquets perdus

La transformée de Hadamard est par la suite appliquée aux coefficients mis à l'échelle afin de fournir une résistance à la perte de paquets [54]. La matrice Hadamard est composée des nombres $+1$, -1 et a pour rôle de rééquilibrer l'énergie entre les chunks. En effet, après la transformation DCT, la différence d'énergie entre les chunks est très hétérogène en raison de la décorrélation obtenue après DCT-3D. De fait, si un chunk est perdu, la distorsion impliquée dépend directement du niveau énergétique du chunk perdu. En s'assurant que chaque paquet contient approximativement la même quantité d'énergie, SoftCast permet d'offrir une dégradation limitée en cas de perte de paquets. A l'issue de ce processus les chunks sont appelés slices. Chaque slice est une combinaison linéaire de tous les chunks et est définie par $\mathbf{Y}_i[j] = W_i \cdot \mathbf{U}_i[j]$ où W_i dénote la $i^{ème}$ ligne de la matrice Hadamard.

1.2.4 Modulation

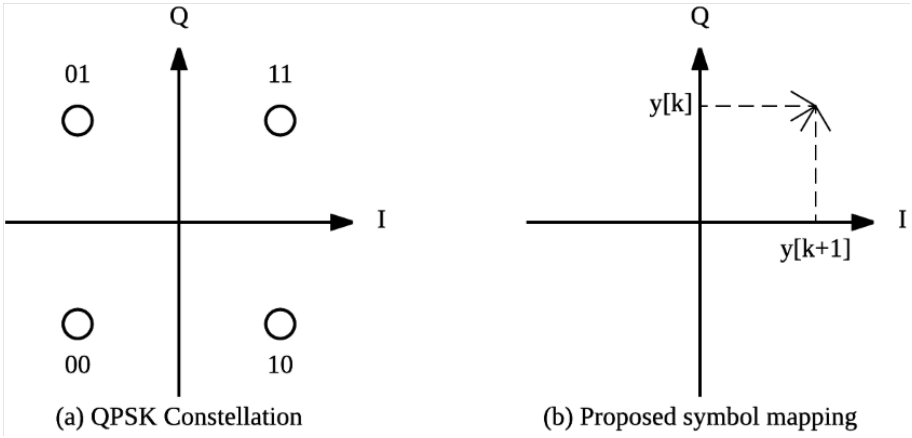


FIGURE 1.3 : Représentation de la modulation (pseudo)-analogique. (a) Exemple de modulation conventionnelle (QPSK). (b) Modulation (pseudo)-analogique. Figure issue de [99].

Après toutes les opérations listées ci-dessus, les valeurs obtenues après transformée de Hadamard issus des slices sont directement transmises à l'aide d'une modulation dense connue sous le nom : Raw Orthogonal Frequency Division Multiplexing (Raw-OFDM) [111] ou encore 64K-QAM. Cette étape se fait en ignorant les codes correcteurs d'erreurs (FEC) et les modulations classiques d'un système OFDM (e.g. QPSK). Ainsi, dans SoftCast, au lieu de débit binaire, le débit symbole est considéré et la dénomination pseudo-analogique ou quasi-analogique est retenue. Etant donné que les valeurs issues des slices sont envoyées par paires (plan I et Q en OFDM) comme indiqué en Fig. 1.3, la bande passante maximale du canal

dans SoftCast peut être définie de la manière suivante :

$$BW_{max} = \frac{(N_R \cdot N_C \cdot F_r)}{2}, [symbs/s] \tag{1.8}$$

où F_r représente le nombre d'image par seconde (fps).

Lorsque la bande passante de l'application ne permet pas la transmission de tous les coefficients transformés, SoftCast opère en jetant/supprimant des chunks en commençant par les moins énergétiques (les chunks sont préalablement triés).

Par exemple, un format vidéo CIF à 30 fps représente un volume de données de $352 \cdot 288 \cdot 30 = 3.04 \cdot 10^6$ valeurs réelles à transmettre par seconde [41]. Le nombre de symboles par seconde correspondant est de $3.04 \cdot 10^6/2 = 1.52$ Msymboles/s. Si l'on considère une bande passante disponible de 1MHz par utilisateur, environ 30% des coefficients doivent être jetés. Ceci est d'autant plus vrai si l'on considère le cas de la transmission de contenus Haute Définition (HD). Dans ce cas précis, la bande passante requise est trop importante et jeter/supprimer des chunks dans l'étape de compression devient inévitable.

Le taux de compression (1.3) peut également être exprimé en termes de ratio de bande passante comme suit :

$$\text{CR} = \frac{BW_{ava}}{BW_{max}} \tag{1.9}$$

où BW_{ava} dénote la bande passante disponible à l'émission.

Si BW_{ava} est inférieure à la bande passante du signal à transmettre issu de SoftCast le nombre de chunks transmis K à l'intérieur d'un GoP est ajusté en conséquence.

En utilisant (1.3), (1.4), (1.8) et (1.9) nous obtenons le nombre de chunks K pouvant être transmis :

$$K = \lfloor \frac{BW_{ava} \cdot N}{BW_{max}} \rfloor \tag{1.10}$$

$$= \lfloor \frac{2.BW_{ava} \cdot N_F}{(nb_r \cdot nb_c \cdot F_r)} \rfloor \tag{1.11}$$

où $\lfloor \bullet \rfloor$ désigne l'opération d'arrondi au nombre entier inférieur ;

1.2.5 Métadonnées

En parallèle du processus d'encodage, l'émetteur SoftCast envoie des métadonnées essentielles pour restaurer le signal vidéo. Elles représentent une faible proportion de données (0.014 bits/pixel dans [54]) composée de :

- la moyenne de chaque chunk, notée μ_i ; $i = 1, 2, \ldots, K$;
- la variance/énergie des chunks transmis, notée λ_i ;
- une carte binaire indiquant les positions des chunks ignorés au sein du GoP.

Tandis que les valeurs issues des slices sont transmises via une modulation (pseudo)-analogique comme indiquée dans la Section 1.2.4, les métadonnées sont fortement protégées et transmises de manière robuste (BPSK par exemple [27]) pour assurer une réception correcte de ces dernières.

1.2.6 Décodeur LLSE

Du côté du récepteur, un décodeur LLSE (Linear Least Square Error) est utilisé afin d'obtenir la meilleure estimation (au sens de la minimisation de l'erreur quadratique moyenne) des coefficients DCT reçus en sortie du canal AWGN. En utilisant les métadonnées et les opérations inverses du décodeur, ces coefficients sont ensuite réassemblés pour reformer les plans DCT. Ces derniers sont ensuite transformés via le processus DCT-3D inverse. Dans le cas où l'application est à bande passante réduite, les chunks supprimés à l'émission sont remplacés par des valeurs nulles à la réception avant transformation inverse.

Après application de toutes les opérations linéaires au niveau de l'encodeur, $\mathbf{Y}_i[j]$ est transmis.

Le signal émis peut être facilement réorganisé sous forme matricielle pour obtenir [54] :

$$\mathbf{Y} = WG\mathbf{X} = C\mathbf{X} \tag{1.12}$$

où $\mathbf{X}_i[j]$ représente l'élément de la $i^{ème}$ ligne et la $j^{ème}$ colonne de la matrice $\mathbf{X}$, de taille $N \times nb_c \cdot nb_r$. $\mathbf{Y}$, également de taille $N \times nb_c \cdot nb_r$, est constituée des slices correspondantes après mise à l'échelle et transformée de Hadamard. La matrice d'encodage C représente le produit WG où W et G représentent respectivement la matrice Hadamard et la matrice de mise à l'échelle. Si le CR<1, $\mathbf{X}$ et $\mathbf{Y}$ seront de dimension $K \times nb_c \cdot nb_r$.

La valeur reçue $\hat{\mathbf{Y}}_i[j]$, après traversée du canal AWGN, vaut $\hat{\mathbf{Y}}_i[j] = \mathbf{Y}_i[j] + n_i[j]$ où $n_i[j]$ représente le bruit additif gaussien centré de puissance $\sigma^2_{n,i}$.

Le récepteur SoftCast décode et obtient la meilleure estimation par [54] :

$$\hat{\mathbf{X}}_i[j] = \frac{g_i \lambda_i}{g_i^2 \lambda_i + \sigma^2} \cdot \hat{\mathbf{Y}}_i[j] \tag{1.13}$$

Par conséquent, l'équation LLSE peut également être écrite sous la forme matricielle comme suit :

$$\hat{\mathbf{X}}_{LLSE} = \Lambda C^T (C\Lambda C^T + \Sigma)^{-1} \hat{\mathbf{Y}} \tag{1.14}$$

où Λ est une matrice diagonale dont les éléments diagonaux représentent l'énergie λ_i du $i^{ème}$ chunk, et Σ est une matrice diagonale dans laquelle le $i^{ème}$ élément est la puissance du bruit additif centré σ_n^2 qui se superpose aux données du paquet contenant la $i^{ème}$ ligne de $\mathbf{Y}$.

Par simplicité et sans perte de généralité, il est admis qu'un paquet contient une slice

[54]. Dans ce cas, le décodeur LLSE peut être réécrit en présence de paquets perdus sous la forme :

$$\hat{X}_{LLSE} = \Lambda_{(*i,*i)} C_{*i}^T \Big(C_{*i} \Lambda_{(*i,*i)} C_{*i}^T + \Sigma_{(*i,*i)} \Big)^{-1} \hat{Y}_{*i} \tag{1.15}$$

où C_{*i} et $\mathbf{Y_{*i}}$ dénotent les matrices C et $\mathbf{Y}$ après suppression de la $i^{ème}$ ligne perdue et $\Lambda_{(*i,*i)}$ représente la matrice Λ après suppression de la $i^{ème}$ ligne et $i^{ème}$ colonne.

L'ensemble de ces blocs permet à SoftCast d'obtenir les caractéristiques suivantes faisant de lui un bon candidat pour la transmission de contenus vidéo dans un contexte broadcast :

- SoftCast permet d'obtenir une dégradation fluide de la qualité [112] ;
- Pour chaque utilisateur, la qualité reçue au récepteur est une fonction linéaire du CSNR (voir ci-après) ;
- Un seul flux est envoyé et peut-être décodé par n'importe quel récepteur même si ce dernier est soumis à de mauvaises conditions de réception [55] ;
- L'émetteur SoftCast fonctionne sans aucun retour (feedback) de la part des récepteurs [54].

1.2.7 Modélisation théorique du schéma SoftCast

Xiong *et al.* [111], ont récemment montré que les performances des systèmes basés SoftCast pouvaient être théoriquement modélisées.

Soit un système de transmission où un vecteur $\mathbf{x} = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^N$ est transmis sur un canal AWGN. Le vecteur reçu correspondant est noté $\hat{\mathbf{x}}$. Comme expliqué dans la Section 1.2, l'émetteur effectue tout d'abord une mise à l'échelle des coefficients avant la transmission :

$$y_i = g_i \cdot x_i. \tag{1.16}$$

Après transmission, le signal reçu est contaminé par un bruit blanc additif gaussien (AWGN) de variance σ_n^2 :

$$\begin{aligned} \hat{y}_i &= y_i + n_i, \\ &= g_i \cdot x_i + n_i. \end{aligned} \tag{1.17}$$

où n_i représente le bruit du canal.

Considérant l'estimateur Zéro-Forcing (ZF) à la réception, le vecteur estimé est donné par la relation :

$$\begin{aligned} \hat{x}_i &= \hat{y}_i \cdot \alpha_i \\ &= \frac{\hat{y}_i}{g_i} \end{aligned} \tag{1.18}$$

$$= \quad x_i + \frac{n_i}{g_i}.$$

où α_i représente le type d'estimateur utilisé. Ici, $\alpha_i = \frac{1}{g_i}$ pour l'estimateur ZF.

La distorsion attendue dans $\hat{x_i}$ est :

$$\begin{aligned} D_i &= E\left[(\hat{x}_i - x_i)^2\right], \\ &= \frac{E\left[n_i^2\right]}{g_i^2}, \\ &= \frac{\sigma_n^2}{g_i^2}. \end{aligned} \tag{1.19}$$

La puissance utilisée pour envoyer x_i est :

$$\begin{aligned} P_i &= E\left[{y_i}^2\right], \\ &= g_i^2 \cdot E\left[{x_i}^2\right]. \end{aligned} \tag{1.20}$$

Pour faciliter la notation, $E\left[{x_i}^2\right]$ est dénoté par λ_i par la suite.

En combinant les équations (1.19) et (1.20), nous obtenons la fonction puissance-distorsion des CVL :

$$D_i \cdot P_i = \sigma_n^2 \cdot \lambda_i,$$

ou encore

$$D_i(P_i) = \frac{\sigma_n^2}{P_i} \cdot \lambda_i. \tag{1.21}$$

Pour atteindre des performances optimales en considérant un estimateur ZF, la puissance totale de transmission disponible à l'émetteur P, est allouée et distribuée à tous les éléments x_i par :

$$(P1) : \min \sum_{i=1}^{N} D_i, s.t. \sum_{i=1}^{N} P_i \leq P \tag{1.22}$$

C'est un problème lagrangien :

$$\mathcal{L} = \sum_{i=1}^{N} D_i + \frac{1}{C_{\mathrm{ZF}}^2} \sum_{i=1}^{N} P_i, \tag{1.23}$$

où C_{ZF}^2 est le multiplicateur de Lagrange.

En annulant la dérivée de $\mathcal{L}$ (1.24) par rapport à P_i, on obtient :

$$C_{\mathrm{ZF}}^2 = \frac{P_i^2}{\lambda_i \sigma_n^2}. \tag{1.24}$$

Ceci détermine la puissance optimale pour envoyer x_i :

$$P_i = C_{\mathrm{ZF}}^2 \sigma_n \sqrt{\lambda_i}. \tag{1.25}$$

Etant donné qu'il existe une contrainte sur la puissance totale de transmission :

$$\sum P_i = P, \tag{1.26}$$

où P représente la puissance totale disponible à l'émission.

C_{ZF} peut être obtenu par :

$$C_{\mathrm{ZF}} = \frac{P}{\sigma_n \sum \sqrt{\lambda_i}}. \tag{1.27}$$

En rappelant les équations (1.21) et (1.25), on obtient facilement :

$$D_i = \frac{\sigma_n^2}{P_i}\lambda_i = \frac{\sigma_n}{C_{\mathrm{ZF}}}\sqrt{\lambda_i}. \tag{1.28}$$

Par conséquent, la distorsion totale attendue, dénotée par D, est donnée par :

$$\mathrm{D} = \sum_{i=1}^{N} D_i = \frac{\sigma_n^2}{P}\left(\sum_{i=1}^{N}\sqrt{\lambda_i}\right)^2, \tag{1.29}$$

En se basant sur les définitions suivantes du CSNR et du PSNR, exprimées en décibels :

$$\mathrm{CSNR} = 10\log_{10}(\bar{P}/\sigma_n^2), \quad \bar{P} = P/N, \tag{1.30}$$

$$\mathrm{PSNR} = 10\log_{10}(255^2/\bar{D}), \quad \bar{D} = D/N. \tag{1.31}$$

Xiong *et al.* [111] ont montré que la qualité vidéo reconstruite peut être modélisée par :

$$\mathrm{PSNR}_{[\mathrm{ZF/FB}]} = c + \mathrm{CSNR} - 20\log_{10}(H), \tag{1.32}$$

où $c = 20\log_{10}(255)$ et

$$H = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\lambda_i}, \tag{1.33}$$

représente *l'activité des données* (*data activity*). A CSNR fixe, une grande valeur de H entraîne une qualité de reconstruction faible en termes de PSNR. On observe bien une caractéristique linéaire du $\mathrm{PSNR}_{[\mathrm{ZF/FB}]}$ qui dépend des conditions de transmission (CSNR).

Nous notons ici que le modèle de Xiong *et al.* est proposé via deux hypothèses : la première est que la bande passante disponible de l'application permet la transmission des N éléments de $\mathbf{x}$ (i.e., pas de compression, tous les coefficients sont transmis). Ce n'est généralement pas le cas et c'est d'autant plus vrai si le format vidéo est en haute résolution (HD, 4K, etc.). La deuxième concerne le fait qu'à la réception, un décodeur ZF est utilisé. Ce n'est pas le cas du

schéma de base SoftCast proposé par [53] qui utilise un décodeur LLSE. Nous verrons dans le Chapitre 2 que ce modèle peut être étendu pour prendre en compte ces deux paramètres (bande passante disponible et type de décodeur utilisé).

1.3 Variantes de SoftCast...

Depuis les travaux pionniers de SoftCast, de nombreuses variantes ont été proposées dans la littérature. Parmi toutes ces variantes, nous présentons ici celles qui ont retenues le plus d'attention. Nous différencions ici deux approches, i.e., les schémas dits analogiques (SoftCast) et les schémas dits hybrides (HDA, i.e., les métadonnées transmises numériquement sont plus conséquentes que celles transmises par SoftCast). Ces derniers peuvent être décomposés en deux sous parties :

- Les signaux numériques et analogiques sont transmis de manière orthogonale.
- Les signaux numériques et analogiques sont superposés pour la transmission.

1.3.1 ...Orientées Codage Vidéo

1.3.1.1 DCast

DCast [18, 19, 21] est un schéma qui intègre un codage de source distribué (DSC : Distributed Source Coding [33]) dans le schéma classique SoftCast. Au lieu de travailler sur des groupement d'images tels que SoftCast, DCast reprend une structure 'IPPP' des encodeurs classiques tel H.264/AVC. Les frames I suivent directement le processus classique de SoftCast (DCT-2D, allocation de puissance, et transformée de Hadamard). Pour les frames prédites P, plutôt que de transmettre directement le signal transformé après DCT, DCast ajoute un codage coset [81] permettant de réduire significativement l'énergie du signal envoyé en supprimant une partie de l'information du signal d'origine. Le schéma bloc de l'émetteur pour les frames prédites P est représenté dans la Fig. 1.4. Tout d'abord, DCast effectue sur l'image actuelle une transformée DCT-2D spatiale tout en accomplissant les étapes d'estimation et de compensation de mouvement (ME/MC) permettant d'obtenir l'image prédite ainsi que les vecteurs mouvements (MV) associés. Une transformée DCT est appliquée à la fois sur l'image prédite et sur les vecteurs mouvements. Les coefficients transformés de l'image originale sont ensuite soumis à un codage coset. Enfin, un processus conjoint d'allocation de puissance est utilisé entre les vecteurs mouvements transformés et les valeurs résiduelles à l'issue du coset sous la contrainte $P = P_{\text{coset}} + P_{\text{MV}}$.

Le codage coset peut être synthétisé de la manière suivante :

$$C_i = X_i - \mathrm{Q_i}\left(X_i\right) \tag{1.34}$$

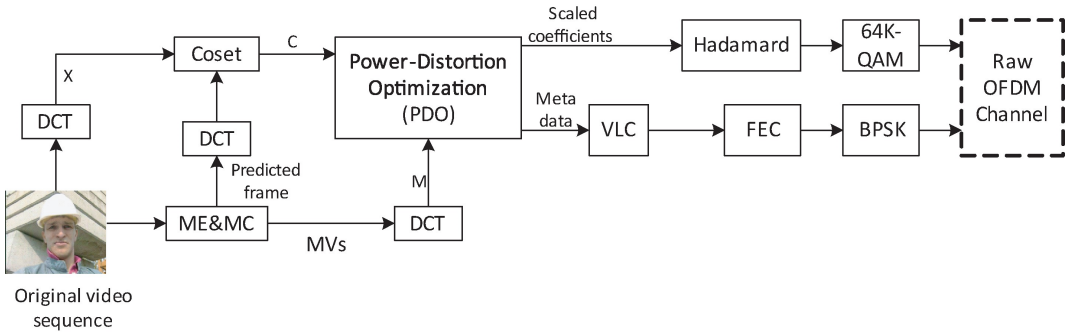


FIGURE 1.4 : Schéma bloc de l'émetteur DCast pour les images P. Figure issue de [21].

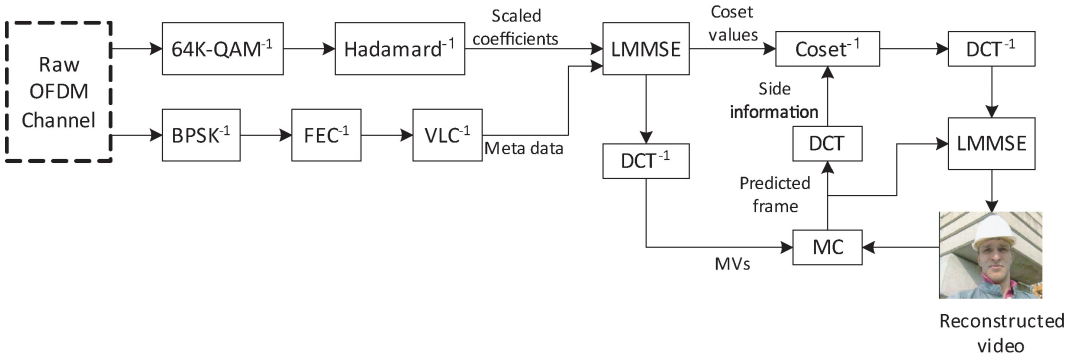


FIGURE 1.5 : Schéma bloc du récepteur DCast pour les images P. Figure issue de [21].

où X représente la matrice contenant les chunks après DCT-2D, X_i la sous-bande fréquentielle considérée, $Q_i(\cdot)$ la quantification pour la sous-bande considérée et C_i la valeur résiduelle obtenue après coset. Dans DCast, la quantification est calculée à l'encodeur de telle sorte que le décodage soit correct avec une haute probabilité. Ce dernier dépend entre autres de l'intensité du bruit considérée dans le canal (liée au CSNR) ainsi que de la différence entre les coefficients DCT prédits et originaux.

A la réception, le processus est similaire à SoftCast comme indiqué dans la Fig. 1.5, les métadonnées sont tout d'abord décodées. Puis, un estimateur LMMSE est utilisé (NB : dans le cas gaussien, les estimateurs LMMSE et LLSE sont équivalents [31]), suivi des processus de transformées inverses. A l'aide des vecteurs mouvements reconstruits et de l'image de référence, l'image prédite est obtenue au décodeur. Celle-ci est transformée via une DCT-2D spatiale et permet d'effectuer le décodage coset à partir des valeurs résiduelles transmises à l'issue du codage coset. Pour finir, une DCT-2D inverse est appliquée sur les valeurs reconstruites et un second estimateur LMMSE est utilisé pour obtenir l'image reconstruite en combinant l'image prédite ainsi que l'image décodée via le décodage coset.

Les auteurs ont montré que l'usage du codage de source distribué permettait à DCast d'obtenir des gains en termes de PSNR au plus égaux à 2dB par rapport à SoftCast. Ce gain

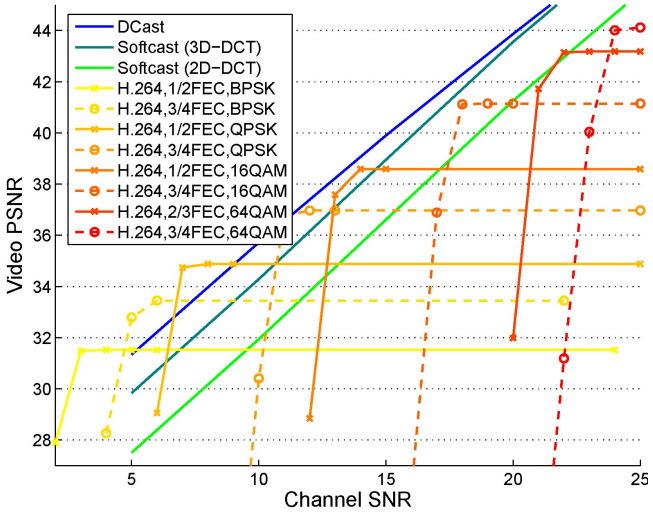


FIGURE 1.6 : Comparaison entre les schémas SoftCast, DCast et H.264. Le CSNR présumé pour DCast est de 5dB. Figure issue de [21].

dépend principalement de la bonne connaissance du canal comme indiqué dans la Fig. 1.6. En effet, les auteurs présument dans l'illustration un CSNR de 5dB pour les pas de quantification du codage coset. Or, si le CSNR est en réalité de 20dB, il en résulte que les performances de DCast diminuent très fortement et deviennent similaires à celles du schéma SoftCast.

Deux versions de DCast ont été proposées à savoir, une version analogique [21] et une version hybride analogique/numérique [18]. Celle-ci, se différencie par le fait que les vecteurs mouvements sont calculés uniquement au décodeur. Ceci permet de réduire grandement la complexité de l'encodeur, mais cela induit une perte de précision des informations collatérales (side information) et donc une perte de qualité à la réception (environ 1dB par rapport à DCast analogique). Par conséquent, nous nous focalisons dans ce chapitre sur la dernière version en date, i.e., la version analogique.

Des extensions de DCast ont ensuite été proposées [122, 47, 130, 22] permettant entre autres, de prendre en compte des tailles de blocs variables pour le processus d'estimation de mouvement (ME) ou encore d'utiliser un algorithme d'affinage des side information [73], ce schéma est dénommé par l'acronyme SIRCast. Les différentes améliorations permettent d'apporter des gains en termes de PSNR respectivement de 0.5dB, 1∼2dB et 1.5dB. Basé sur les travaux de DCast, LineCast [79] a été proposé pour la transmission d'images satellitaires via une exploitation de la corrélation existante entre les lignes de pixels. Enfin, le framework proposé par [22] dénommé LayerCast permet de répondre à l'hétérogénéité de bandes passantes pouvant exister entre différents utilisateurs via un encodage multicouche basé sur le codage coset.

1.3.1.2 WaveCast

Introduit en 2012, WaveCast [20] est un schéma hybride de catégorie 1, similaire à SoftCast qui propose de remplacer la transformée DCT-3D par une transformée en ondelettes à trois dimensions (DWT-3D) pour une meilleure exploitation de la redondance temporelle entre les images. Comme illustré dans la Fig. 1.7, la transformée DWT-3D proposée s'effectue en deux étapes. Un filtrage temporel à compensation de mouvement (MCTF) [86, 108] est tout d'abord appliqué sur la GoP. Celui-ci effectue une estimation de mouvement est tout d'abord effectuée entre deux images consécutives pour générer des images passe-bas et passe-haut sont générées. Cet algorithme est effectué de manière récursive n fois, ici $n = 4$. Une transformée en ondelettes spatiale (DWT-2D) est finalement appliquée sur les plans $\mathrm{LLL}_0, \mathrm{LLH}_1, \mathrm{LH}_1, \mathrm{LH}_3, \mathrm{H}_1, \mathrm{H}_3, \mathrm{H}_5$ et H_7 obtenus en sortie du MCTF.

A faible CSNR, WaveCast permet une amélioration de la qualité reçue d'environ 2dB.

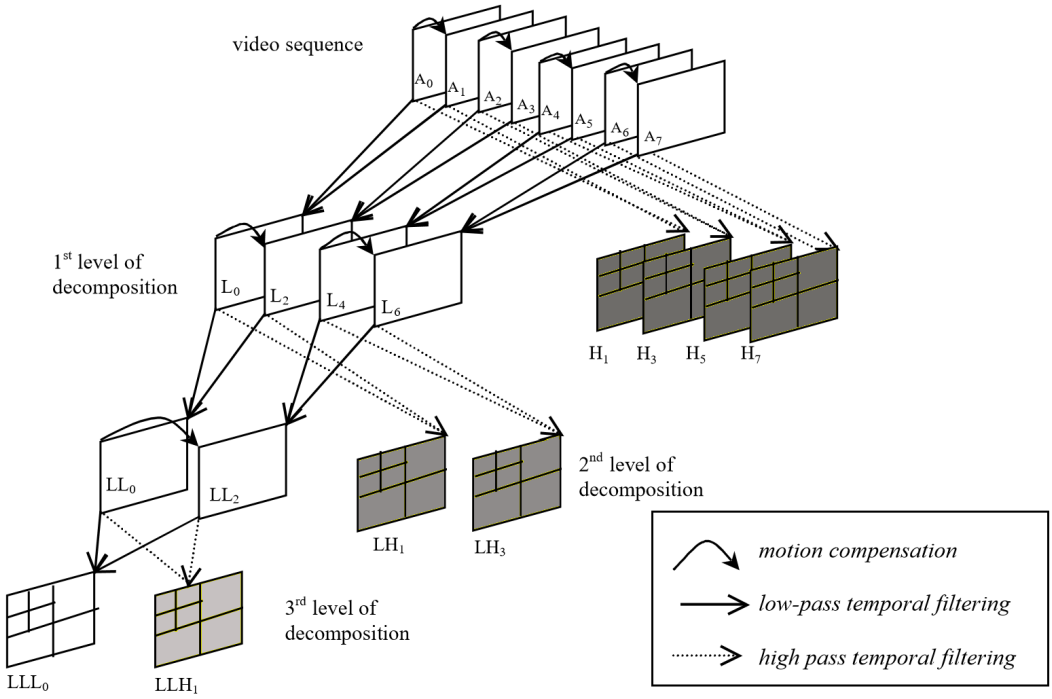


FIGURE 1.7 : Illustration de la transformée en ondelettes à trois dimensions. Figure issue de [20].

Le diagramme bloc de l'émetteur WaveCast est donné en Fig. 1.8. Les vecteurs mouvements (MV) à l'issue du MCTF sont transmis en métadonnées additionnelles, il s'agit donc d'un schéma hybride. Celles-ci sont fortement protégées (FEC) et transmises de manière robuste (BPSK) afin d'assurer une réception quasi-parfaite de ces données. Pour le reste, les coefficients à l'issue de la DWT-3D suivent un processus semblable à SoftCast : allocation de puissance (PDO), résistance à la perte de paquets (Hadamard), etc.

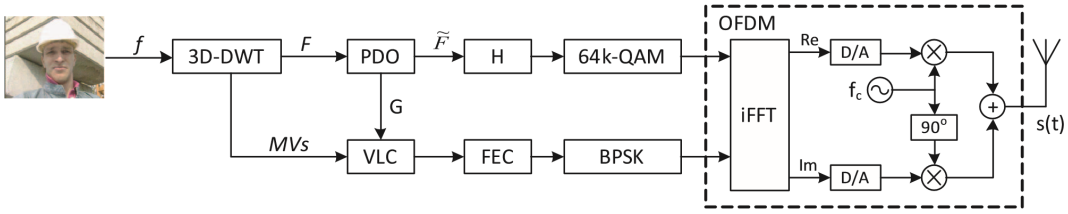


FIGURE 1.8 : Schéma bloc de l'émetteur WaveCast. Figure issue de [20].

Parmi les autres travaux ayant recourt à un filtrage MCTF, nous pouvons citer Cactus [12] qui utilise en plus de ce filtrage, la corrélation intra-image existante (au niveau pixel) pour effectuer des méthodes de débruitage (denoising) à la réception. En effet, aucune transformée de décorrélation n'est appliquée, les pixels sont directement transmis après allocation de puissance. Enfin, nous pouvons également noter que la Karhunen-Loeve transform (KLT) dans un contexte de transmission d'image satellitaire hyperspectrale a été adoptée par HyperCast [36].

1.3.1.3 WSVC

Le schéma intitulé : Wireless Scalable Video Coding (WSVC, [118]) fait partie des schémas dits hybrides de catégorie 2 où l'information numérique X_d est superposée au signal analogique X_a comme indiqué dans la Fig. 1.9.

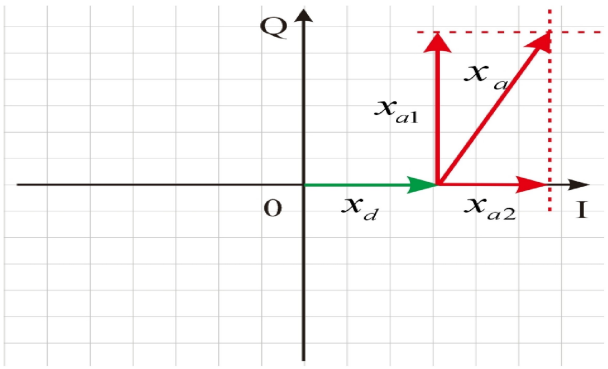


FIGURE 1.9 : Mapping des données analogiques et numériques. Figure issue de [118].

Le schéma bloc est disponible en Fig. 1.10. La vidéo haute résolution (HR) est tout d'abord décomposée spatialement via une transformée en ondelettes 2D. A l'issue de celle-ci, la bande LL qui contient l'image en basse résolution (LR) est encodée via un encodage vidéo standard comme H.264/AVC par exemple, et suit ensuite le processus classique de transmission numérique : protection FEC et modulation BPSK. Cette partie représente la couche de base. La couche d'amélioration contient quant à elle, le résidu issu de l'encodage

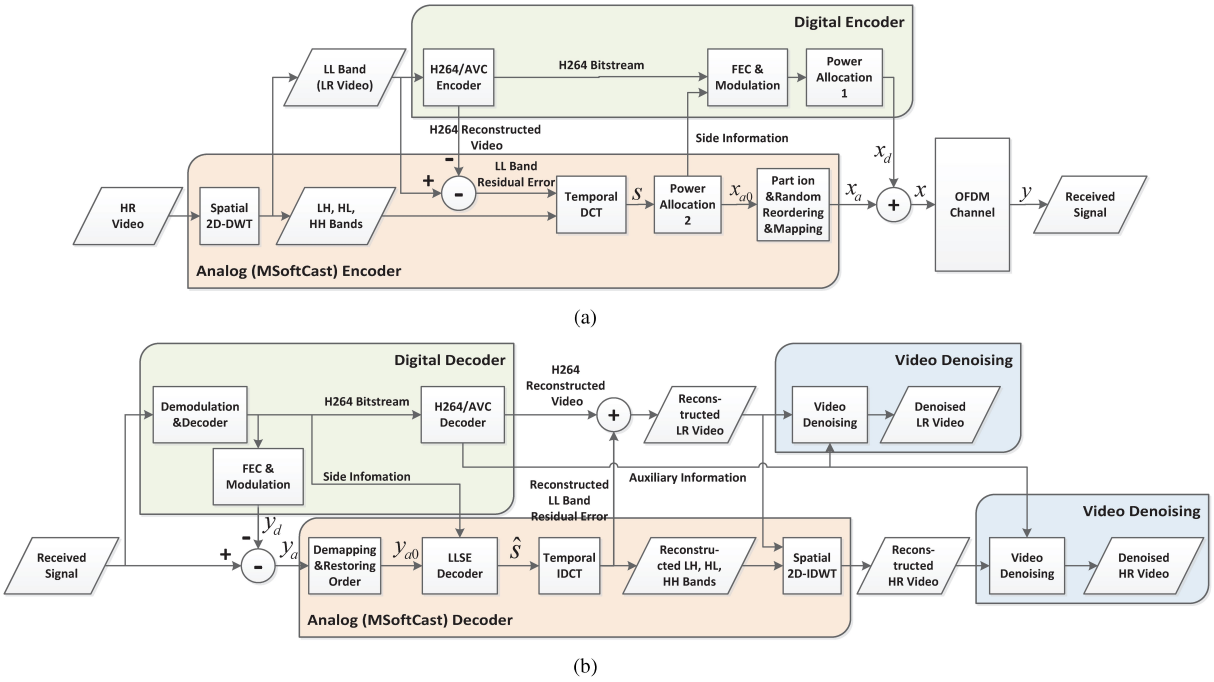


FIGURE 1.10 : Schéma bloc de WSVC. (a) Emetteur, (b) Récepteur. Figure issue de [118].

H.264 ainsi que les bandes LH, HL et HH qui sont transmis via une transmission pseudo-analogique semblable à celle de SoftCast, notée MSoftCast (DWT-2D spatiale puis DCT temporelle, allocation de puissance et réarrangement aléatoire des données au lieu de la transformée de Hadamard). Les deux signaux résultants sont ensuite superposés avant d'être transmis. Ceci introduit un problème d'allocation de puissance qui a été formalisé par [95, 97]. A la réception, des méthodes de débruitage sont utilisées pour les mauvais CSNR afin de réduire le bruit dû à la transmission. WSVC permet d'obtenir une amélioration globale de la qualité en termes de PSNR comprise entre 0.2 ∼3.3dB par rapport à DCast (version analogique).

Parmi les autres schémas hybrides dérivés de SoftCast, nous pouvons citer :

- SharpCast [42] où la vidéo est séparée en deux parties : le contenu et la structure. L'information structurelle ainsi que les chunks contenant des fréquences hautement énergétiques sont protégés et transmis numériquement afin d'offrir une meilleure qualité perçue par l'utilisateur. L'information résiduelle (ou contenu) est quant à elle transmise en utilisant la modulation (pseudo)-analogique ;
- Des schémas coopératifs parmi lesquels : Wireless Cooperative Video Coding (WCVC, [119]) qui représente l'extension de WSVC au cas coopératif où un récepteur disposant d'un bon canal relaie l'information à son voisin soumis à un plus faible CSNR afin d'améliorer la qualité à la réception. Ce schéma a par la suite été amélioré par Sun

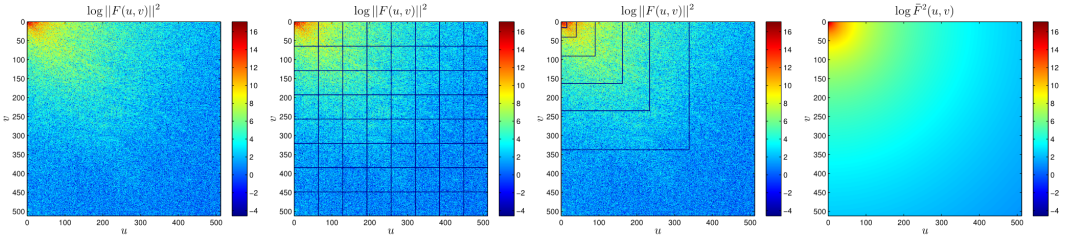
et al. [94] qui ont proposé de modifier la structure et d'ajouter un codage coset afin d'exploiter la corrélation entre les signaux reçus. Enfin, Shen *et al.* [89] ont proposé une analyse théorique permettant une sélection optimale du pas de quantification du codage coset ainsi que de l'allocation de puissance ;

- Adaptive Hybrid Digital–Analog Video Transmission (A-HDAVT) [126]) propose une extension des schémas hybrides aux canaux mobiles en prenant en considération l'évanouissement (fading) du canal. La séparation des contenus numériques / analogiques est effectuée via un filtrage MCTF ;
- Fujihashi *et al.* [30] propose un schéma hybride reposant sur le multihoming, i.e., le fait qu'un utilisateur puisse être connecté à plusieurs réseaux en même temps (par exemple LTE et Wi-Fi). Les informations numériques (i.e., issues d'un encodage H.264) sont transmises via le réseau cellulaire (LTE par exemple). Pour éviter une congestion de ce réseau, un processus de décharge sur les réseaux intermédiaires (Wi-Fi) est proposé. Les informations résiduelles sont quant à elles toujours envoyées via Wi-Fi aux travers d'un schéma pseudo-analogique basé SoftCast.

1.3.1.4 Considération des métadonnées

L'optimisation des métadonnées envoyées en parallèle du flux (pseudo)-analogique constitue un point essentiel des schémas précédents, il a fait l'objet d'une attention particulière dans la littérature [112], [27], [123], [131]. En effet, il convient de rappeler que les métadonnées sont utilisées, d'une part pour indiquer les chunks ou coefficients non transmis, et d'autre part pour l'allocation de puissance inverse à la réception. Cette dernière information, utilise l'approximation selon laquelle à l'intérieur d'un chunk, les coefficients DCT suivent une même distribution et ont une même énergie. De ce fait, la complexité calculatoire ainsi que le surdébit (overhead) apporté par les métadonnées est réduit puisqu'un seul facteur de mise à l'échelle est calculé par chunk. En pratique, cette approximation n'est pas vraiment réaliste et induit une perte des performances du schéma SoftCast. Ceci est d'autant plus vrai que la taille du chunk est importante [114]. Afin de pallier cette approximation tout en réduisant le volume de métadonnées, des méthodes modélisant la distribution énergétique du signal ont été proposées. Ainsi, Xiong *et al.* [112] proposent d'utiliser une découpe adaptative en chunk basée sur des découpes en "L" comme illustré dans la Fig. 1.11. Une modélisation de l'énergie (piecewise log-linear fitting) est ensuite proposée et utilisée dans le processus d'allocation de puissance. Le principal inconvénient de cette méthode est son coût de calcul exigé par l'algorithme itératif qui cherche de manière exhaustive les meilleures tailles des découpes en "L" afin d'arriver à une solution quasi-optimale.

Tout comme [112], Fujihashi *et al.* [27] proposent une transmission pseudo-analogique de type SoftCast sans la découpe conventionnelle des chunks. Le nouveau schéma modélise



(a) (b) (c) (d)

FIGURE 1.11 : Découpe des plans DCT. (a) Energie des coefficients DCT (affichage logarithmique) dénotée F(u,v), (b) Découpe traditionnelle en chunks (SoftCast), (c) Découpe en L-chunk, (d) Modélisation de l'énergie (curve fitting based). Figure issue de [112].

le signal vidéo comme un champ de Markov aléatoire gaussien (GMRF, Gaussian Markov Random Field) et sur une fonction d'ajustement (fitting) Lorentzienne qui modélise l'énergie des coefficients DCT. Cette méthode permet de réduire le volume des métadonnées à 5 valeurs par GoP. Toutefois là encore, le processus permettant d'obtenir les valeurs est chronophage.

Les deux techniques proposées ci-dessus permettent d'améliorer la qualité à la réception tout en réduisant grandement le volume de métadonnées sans toutefois l'éliminer complètement. Récemment, deux nouvelles approches ont été proposées :

- Différent des méthodes précédentes où la distribution énergétique est modélisée, Zhang et Mao [123] proposent un algorithme de détection aveugle des données (blind data detection. En effet, dans cette approche, sans transmission des facteurs de mise à l'échelle (inversement proportionnels à la racine carrée des modules des coefficients à transmettre), le récepteur obtient une estimation biaisée du signal original juste à partir du signal reçu et d'une seule métadonnée : une constante satisfaisant la contrainte de puissance disponible. Dans ce schéma, un chunk contient un seul coefficient, ce qui permet d'obtenir des gains en termes de PSNR de l'ordre de 2dB par rapport à SoftCast.
- S'inspirant des travaux de Zhang et Mao, Zong *et al.* ont proposé un schéma basé SoftCast purement analogique sans aucune métadonnée (metadata-free) transmise [131]. Basé sur une estimation aveugle au niveau chunk, la méthode permet d'obtenir des gains par rapport à SoftCast d'environ 4dB en termes de PSNR. Chaque facteur de mise à l'échelle est inversement proportionnel à la racine carrée de la norme du chunk correspondant. L'émetteur obtient le signal mis à l'échelle directement à partir du signal original. Chaque récepteur décode ainsi le signal en estimant la norme de chaque vecteur à partir du vecteur bruité reçu.

Parmi les autres variantes de SoftCast, différents articles ont introduit des méthodes permettant la prise en compte du système visuel humain (HVS). Ceux-ci sont présentés brièvement ci-dessous.

1.3.1.5 GCast

GCast [109], [66], [67]) représente le premier schéma s'intéressant à la qualité perçue par l'utilisateur. L'œil étant sensible aux contours, les auteurs s'intéressent aux gradients de l'image plutôt qu'aux pixels. Une illustration de celui-ci est présentée en Fig. 1.12.

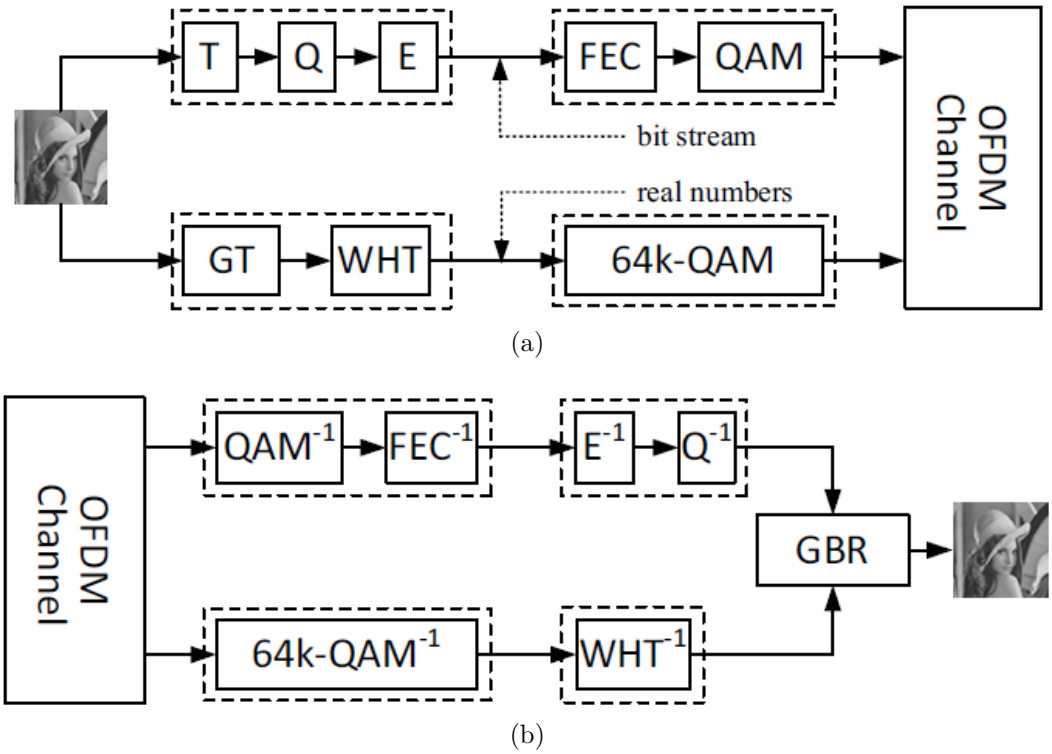
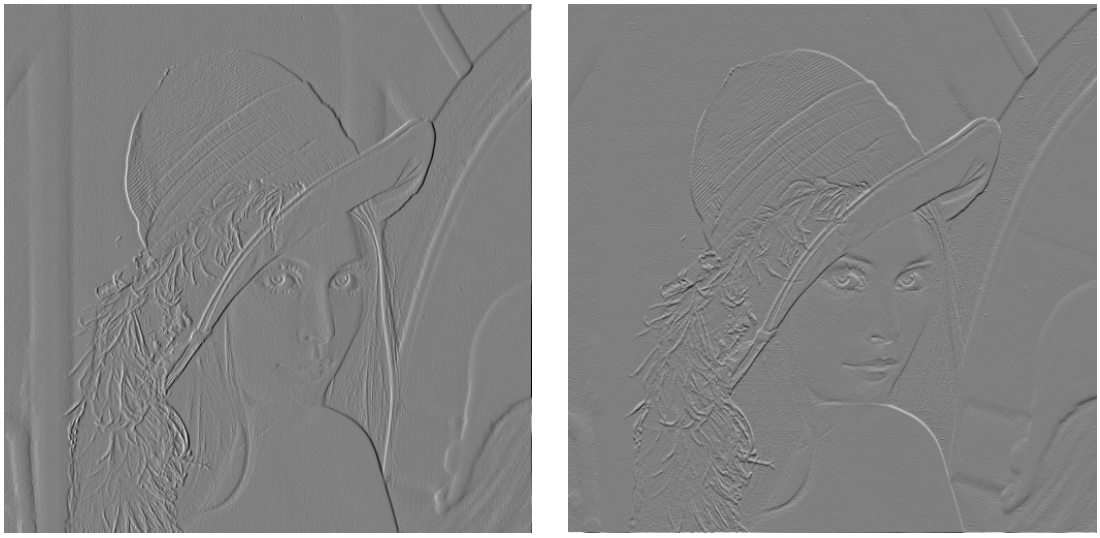


FIGURE 1.12 : Schéma bloc de GCast. (a) Emetteur, (b) Récepteur. Figure issue de [109].

Basé sur une transformée en gradient (GT), GCast transmet une couche de base et une couche d'amélioration. La couche de base permet la transmission de la composante continue ainsi que des basses fréquences permettant une reconstruction grossière de l'image. Les coefficients fréquentiels retenus sont quantifiés (Q) et encodés (E) de manière standards (codage entropique). Ils sont ensuite transmis de la même manière que les métadonnées de SoftCast pour assurer une haute probabilité de décodage même en cas de mauvais CSNR (codage FEC et modulation BPSK). L'image obtenue à partir de ces informations est affinée avec la couche d'amélioration qui contient les gradients de l'image. Ceux-ci sont transmis via le schéma pseudo-analogique SoftCast. A la réception, une reconstruction basée sur les gradients (GBR) est proposée. Comme illustré dans la Fig. 1.13, le principal problème de GCast est que l'information transmise comporte deux fois plus de données. Pour répondre à ce problème, des techniques basées sur le compressive sensing (CGCast) ont été proposées [68, 69, 70].



(a) (b)

FIGURE 1.13 : Images issues de la transformée gradient. Gauche : Gradient horizontal. Droite : Gradient vertical. Les images présentent un offset +128 à des fins d'affichage. Figure issue de [109].

(a) (b)

FIGURE 1.14 : Images reconstruites après transmission dans un canal à CSNR=0dB. Gauche : SoftCast. Droite : GCast. Figure issue de [109].

Récemment, Gao *et al.* [32] proposent de limiter l'effet de lissage (voir Fig. 1.14) de GCast en utilisant un algorithme de classification d'images permettant de séparer les blocs contenant des morceaux lisses, des contours et des détails. En fonction de leur importance,

une allocation de puissance sur les gradients des blocs est proposée.

Parmi les travaux s'intéressant au système visuel humain, nous pouvons également citer :

- SharpCast [42] présenté dans la Section 1.3.1.3 où l'information de structure (haute fréquence) est transmise de manière numérique.
- Des cartes de saillance visuelle ont été prises en compte [35] dans un processus d'allocation de puissance modifié. Une allocation de puissance plus importante est ainsi utilisée pour protéger les régions visuellement importantes de l'image. De la même manière, Zhao *et al.* [125] ont proposé une allocation de puissance modifiée prenant en compte un modèle perceptuel basé sur la métrique SSIM (Structure Similarity) [102].
- VCast [60] est un schéma hybride proposant de diviser une source vidéo X en deux parties et de transmettre en numérique après encodage H.264 les informations dites "visuelles" dénotées X_v (i.e., les moyennes et hautes fréquences). L'information dite "insensible" dénotée X_i suit un processus de décorrélation basé sur une DWT-2D. A l'issue de ce processus, la bande LL ainsi que X_v suivent un encodage H.264 distinct et sont transmis en numérique. Le reste des informations (i.e., les résidus de la bande LL issus de l'encodage H.264 ainsi que les bandes HL, LH et HH) est envoyé via une transmission pseudo-analogique.
- SCast [61] est un schéma proposant de segmenter une image en deux parties : le premier plan qui représente la région d'intérêt (ROI) et l'arrière-plan. Un processus d'allocation de puissance est alors proposé pour assurer une grande protection de la ROI aux erreurs du canal. Ce schéma est analogique, i.e., les deux parties sont envoyées via SoftCast.
- Une méthode de post-traitement basée sur les réseaux de neurones convolutionnels (CNN) ainsi que sur une représentation sparse (group-based sparse representation) permettant de réduire les artefacts (compression et transmission) de SoftCast-2D a récemment été proposée [117].
- Shen *et al.* [90] ont proposé un schéma dénommé FoveaCast qui prend en compte les caractéristiques de la fovéa du HVS pour la transmission d'images. Particulièrement, un facteur de sensibilité est ajouté au processus d'allocation de puissance avec une attention particulière portée au(x) visage(s) se trouvant dans l'image. Dans cet article, l'auteur propose aussi une adaptation de son schéma aux systèmes MIMO (Multiple Input Multiple Output). Ainsi, les coefficients les plus énergétiques sont assignés aux meilleurs sous-canaux du MIMO.

La plupart des articles mentionnés ci-dessus ont tenu compte des aspects liés au codage de la vidéo en considérant un canal AWGN de la même manière que l'article originel SoftCast. Dans ce qui suit, les travaux présentés s'intéressent plus spécifiquement à la partie transmission et notamment à des canaux de transmission relativement complexes.

1.3.2 ...Orientées Télécommunication

1.3.2.1 ParCast

D'abord introduit en 2012, ParCast (Parallel video uniCast) [71] prend en considération le fading (évanouissement) pouvant apparaitre dans le canal au cours d'une transmission vidéo. Ce schéma, considéré comme un schéma hybride de catégorie 1, a par la suite été amélioré dans une seconde version dénommée ParCast+ en 2014 [72] et les différences seront présentées ci-dessous.

ParCast a tout d'abord été proposé pour améliorer les transmissions vidéo via l'utilisation de la modulation MIMO-OFDM (Multiple Input Multiple Output-Orthogonal Frequency Division Multiplexing) où plusieurs antennes à l'émission et à la réception sont utilisées. L'idée principale de ParCast consiste à décomposer le canal MIMO-OFDM en un groupe de sous-canaux à bande étroite et d'utiliser les meilleurs sous-canaux afin de transmettre les chunks les plus énergétiques. En effet, les auteurs ont observé une similarité importante entre les distributions énergétiques des chunks après décorrélation et les gains des sous-canaux issus du MIMO OFDM. Toutefois, cette association de chunk et sous-canal nécessite que le transmetteur reçoive les informations sur l'état du canal, notamment le gain s_i^2 de chaque sous-canal. Nous pouvons donc noter, comme son nom l'indique, que ParCast est prévu pour un contexte Unicast (point à point) et non broadcast (diffusion) à l'inverse de SoftCast.

En assumant que le gain s_i^2 de chaque sous-canal est parfaitement connu, l'allocation de puissance initiale (dénommée dans le schéma en Fig. 1.15 UEP : Unequal Error Protection) de SoftCast devient [71] :

$$g_i = \sqrt{\frac{P}{\sqrt{\lambda_i s_i} \sum_j \sqrt{\lambda_j / s_j}}} \tag{1.35}$$

La version ParCast+, diffère principalement de la version initiale par le fait qu'elle remplace la DCT-3D par une transformée en ondelettes 3D précédemment introduite dans la Section 1.3.1.2 (via MCTF et DCT-2D spatiale) afin d'exploiter une meilleure corrélation temporelle due à l'alignement de mouvement effectuée par le MCTF. Cette version améliorée permet d'obtenir des gains supplémentaires d'environ 2dB en termes de PSNR par rapport à la version initiale.

Le schéma de ParCast+ est très similaire à celui de WaveCast comme indiqué dans la Fig. 1.15 où les informations des vecteurs mouvements couplées aux métadonnées classiques sont transmises de manière numérique (BPSK et FEC 1/2) alors que les coefficients issus de la DWT-3D sont transmis de manière analogique avec un facteur de mise à l'échelle spécifique tenant compte du canal MIMO avec fading.

Parmi les autres travaux s'intéressant aux canaux à évanouissement, nous pouvons citer Cui *et al.* [13, 14], qui s'intéressent respectivement aux canaux fast fading et Rayleigh fa-

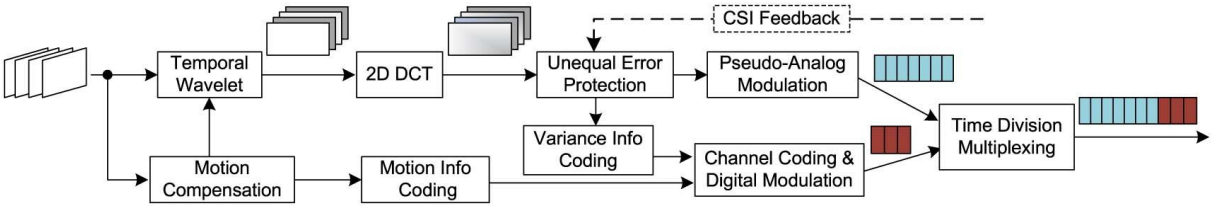


FIGURE 1.15 : Illustration du schéma bloc de transmission ParCast+. Figure issue de [72].

ding et pour lesquels, une optimisation est basée sur une connaissance statistique des CSI à l'émetteur et non pas une connaissance précise comme dans ParCast+. De plus le contexte multi-utilisateurs est pris en compte pour des canaux avec fading dans [124]. De la même manière d'autres travaux s'intéressant aux transmission MIMO peuvent être cités comme [16, 96] s'intéressant entre autres au contexte MU-MIMO (Multiple User MIMO).

1.3.2.2 Prise en compte de la bande passante disponible

Lorsque la bande passante de l'application est limitée, SoftCast supprime des chunks (en commençant par les moins énergétiques) à l'émission afin de répondre à cette contrainte. Toutefois, cette suppression entraine une perte irréversible de qualité au niveau du récepteur entraînant l'effet de saturation (*levelling-off* [64]) observé sur la qualité reçue à haut CSNR.

Pour répondre à ce problème, Cagnazzo *et al.* ont proposé de recourir à l'utilisation du *Shannon-Kotel'Nikov mapping* [11]. Initialement proposé dans [44], le *Shannon-Kotel'Nikov mapping* (SKM) est une technique permettant de transmettre plus d'informations qu'il n'est normalement possible de transmettre via un mapping "intelligent" des données. Plus précisément, des spirales d'Archimède sont utilisées pour "fusionner" M symboles issus de variables indépen-dantes et identiquement distribuées (dans le cas de SoftCast des chunks). Dans [11], le nombre de symboles M vaut 2, un mapping 2 :1 est ainsi utilisé.

Comme illustré dans la Fig. 1.16, dans le cas 2 :1, une paire de coefficients (x_1, x_2) est représentée dans le plan par le point $\mathbf{x}$. Ce point est ensuite ramené sur le point le plus proche d'une spirale et devient $\mathbf{x}' = (x_1', x_2')$ où $\mathbf{x}'$ désigne la version approximée de $\mathbf{x}$. Ce nouveau point est ensuite ramené sur un symbole à une dimension via une fonction de mapping prédéfinie [11]. Cette fonction de mapping dépend entre autres du paramètre Δ qui représente la distance entre les deux spirales d'Archimède. Ce paramètre est optimisé à partir des informations sur l'état du canal de transmission (CSI) et peut être vu comme un pas de quantification.

Bien que le SKM permet d'augmenter le nombre de symboles (ici, les chunks), l'inconvénient est que ceux-ci, transmis via SK mapping, comportent deux types de distorsion : la distorsion due à l'approximation effectuée sur le couple (x_1, x_2) et celle due au bruit du canal. Il y a donc un compromis qui doit être trouvé sur le nombre de chunks envoyés de manière

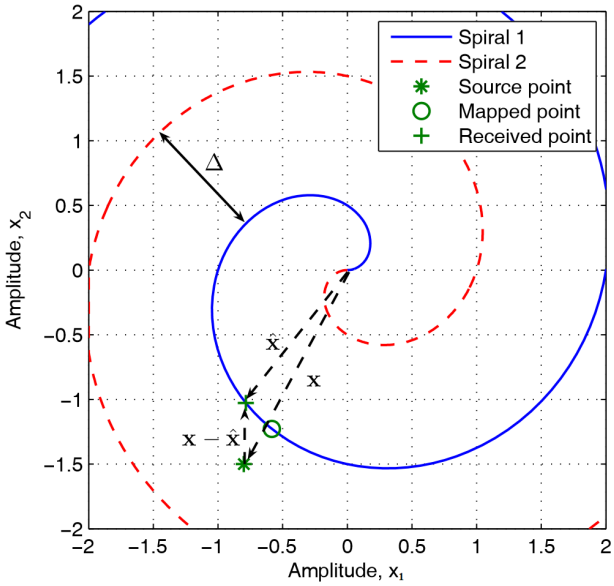


FIGURE 1.16 : Illustration du mapping de Shannon Kotel'Nikov 2 :1 sur des spirales d'Archimède. Figure issue de [44].

classique via SoftCast dénoté n_{SC} et ceux envoyés via SKM (dénotés $2\times n_{SK}$). Ce compromis est illustré en Fig. 1.17. Dans la version SoftCast classique, sur un total de n_T chunks, seuls n_C chunks peuvent être transmis. En revanche, grâce à l'utilisation du SKM dans SoftCast, ce nombre peut être augmenté à $n_{SC} + 2 \times n_{SK}$.

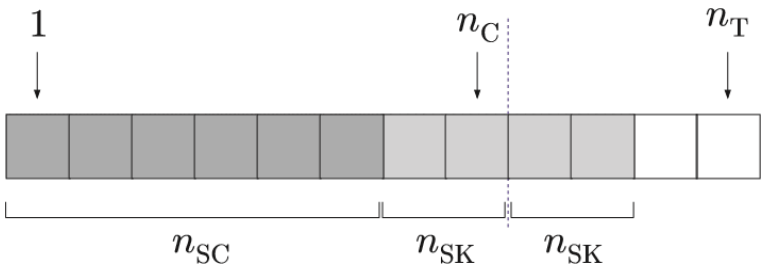
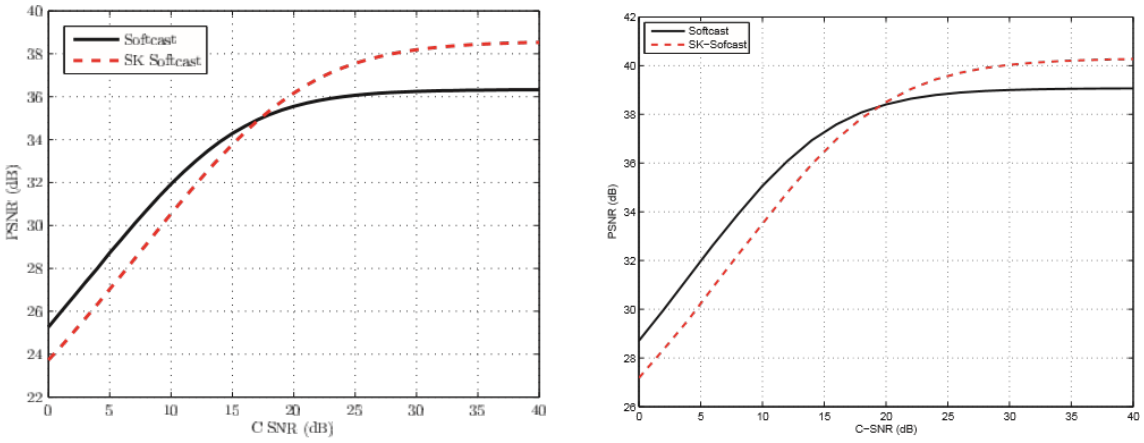


FIGURE 1.17 : Illustration de la répartition de chunks transmis via SoftCast et ceux transmis via SK-SoftCast. Figure issue de [11].

Ce schéma dénommé SK-SoftCast permet à partir d'un CSNR ciblé, d'améliorer la qualité reçue via SoftCast lorsque la bande passante est limitée et que l'effet de saturation apparait (i.e., CSNR≥20dB). Le prix à payer est une perte de qualité pour des valeurs de CSNR faibles comme nous pouvons le voir dans la Fig. 1.18. En plus, de proposer une solution optimale pour le problème d'allocation de puissance dans un tel contexte, Cagnazzo *et al.* ont aussi sur la base de simulations, proposé les paramètres suivants permettant d'obtenir les meilleures performances moyennes : le CSNR ciblé est fixé à 20dB, le nombre de chunks transmis via SK mapping est de $2 \times n_{SC} = 64$.



(a) (b)

FIGURE 1.18 : Illustration des performances de SK-SoftCast en fonction du CSNR par rapport à SoftCast classique. a) Séquence *Foreman* ($n_C = 128$). b) Séquence *Kimono* ($n_C = 512$). Configuration : Taille de GoP=8 images, CSNR ciblé = 20dB, $n_{SK} = 64$. Figure issue de [11].

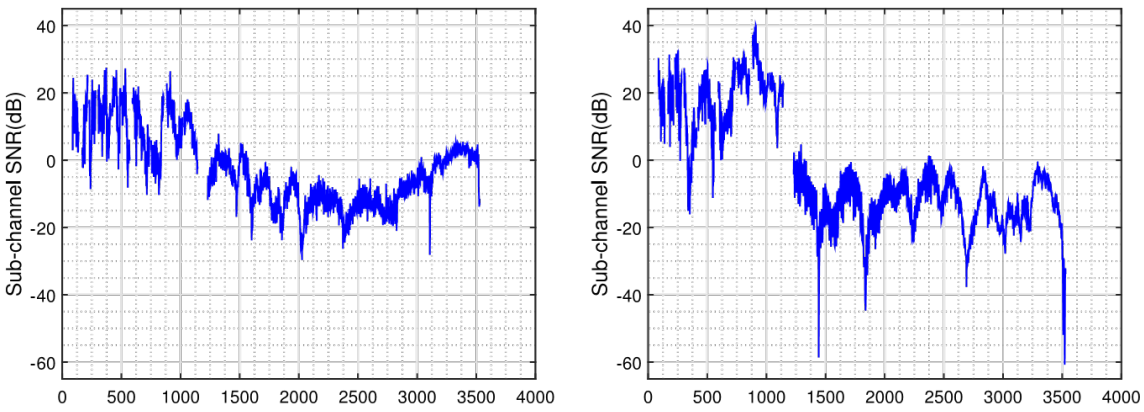
L'idée du SKM a été reprise récemment dans [63] où un schéma hybride (HEVC et transmission analogique avec SKM) est utilisé. La couche de base est ainsi transmise via HEVC et les résidus obtenus à l'issue de l'encodage HEVC sont transmis via SK-SoftCast. L'allocation des ressources (nombre de chunks transmis classiquement, nombre de chunks transmis via SKM et nombre de chunks jetés) est résolue dans cet article.

1.3.2.3 Contraintes de puissance par sous-canal

Zheng *et al.* [129] ont récemment résolu le problème de l'allocation de puissance par sous-canal (gabarit de puissance), i.e., chacun d'entre eux possède sa propre contrainte de puissance au lieu d'une contrainte de puissance unique comme dans la version originelle de SoftCast). Ce problème est rencontré entre autres dans des transmissions multi-antennes ou encore dans un contexte filaire DSL (Digital Suscriber Line) [74] ou CPL (voir Fig. 1.19).

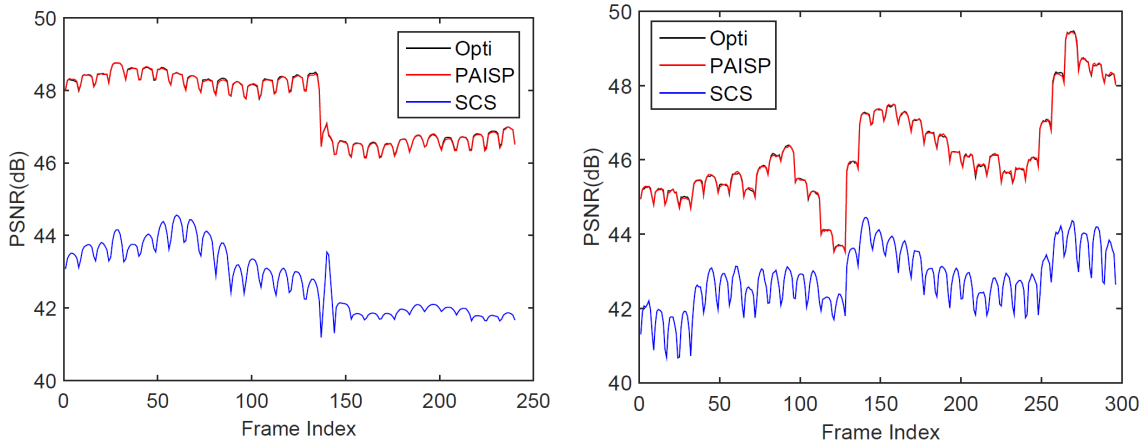
Ainsi, en minimisant l'erreur quadratique moyenne de reconstruction des coefficients transmis une solution optimale est d'abord proposée. Cette dernière repose sur l'utilisation de l'algorithme *multi-level water-filling* [76] au prix d'un coût de calcul important. Pour pallier cet inconvénient, trois alternatives suboptimales : *Simple Chunk Scaling (SCS)*, *Power Allocation with Inferred Split Position (PAISP)* et *Power Allocation with Local Power Adjustment (PALPA)* sont présentées.

Parmi ces trois solutions, les méthodes PAISP et PALPA permettent d'obtenir des résultats très proches de la solution optimale (gap inférieur à 0.03dB) comme observé en Fig. 1.20 (seule la méthode PAISP est illustrée, comme les deux méthodes ont des performances très similaires). La différence se situant sur le fait que la méthode PALPA ne nécessite pas d'ajustement de paramètre.



(a) (b)

FIGURE 1.19 : Exemple de canaux PLT provenant de la base de données ETSI STF 477. a) Canal 1. 2) Canal 250. Figure issue de [129].



(a) Kimono (class B) (b) Race Horses (class C)

FIGURE 1.20 : Evolution du PSNR image par image pour la méthode optimale, PAISP et SCS. a) Kimono, b) RaceHorses. Figure issue de [129].

1.3.2.4 Prise en compte du bruit impulsif

Zheng *et al.* [128] ont récemment proposé des méthodes de correction de bruits impulsifs pouvant apparaître sur des transmissions OFDM multiporteuses telles que les lignes CPL [92], DSL [74] ou encore sur des canaux sans-fil d'intérieur [7] ou des communications ULB (Ultra Large Bande) [87].

Spécifiquement, un algorithme Fast Bayesian Matching Pursuit (FBMP) [84] est utilisé pour obtenir une estimation du bruit impulsif $\hat{v}_{\mathrm{I}}$ représenté par le bloc INE (Impulse Noise Estimation) dans la Fig. 1.21.

Toutefois cette approche nécessite de réserver $q = n_{SC} - \ell$ sous-canaux qui sont mis à

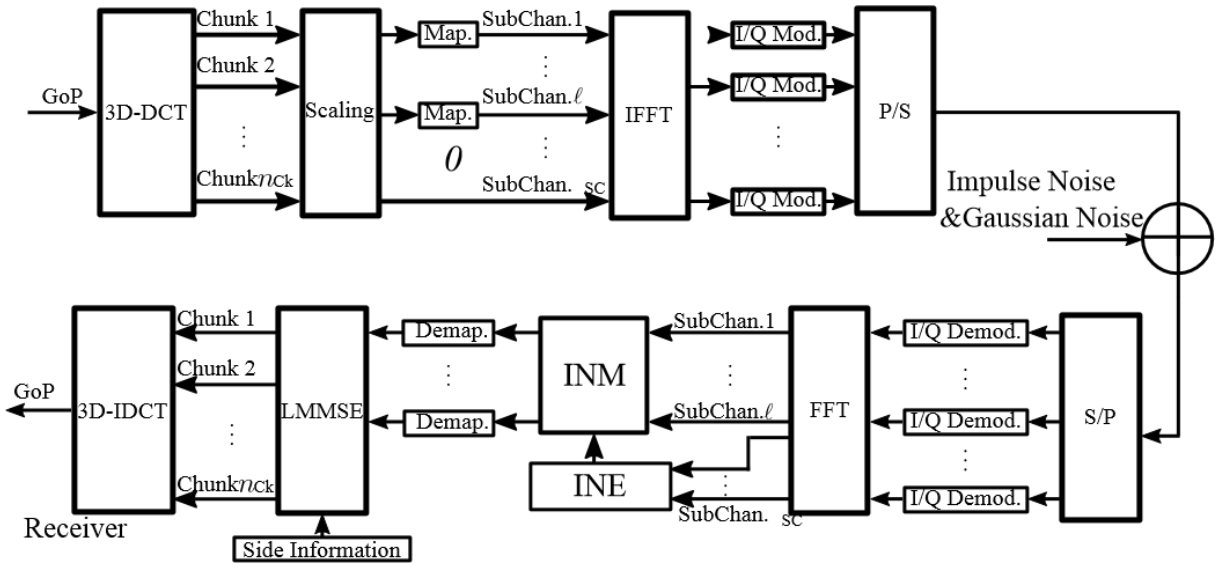


FIGURE 1.21 : Illustration du schéma modifié d'un CVL basé SoftCast avec sous-canaux mis à disposition pour la correction de bruit impulsif. Figure issue de [128].

disposition pour l'estimation du bruit impulsif. Ces sous-canaux non utilisés à l'émission (des valeurs nulles sont transmises) permettent de corriger le bruit impulsif v_{I} à la réception au niveau du bloc INM (Impulse Noise Mitigation).

La réservation de sous-canaux induit une diminution de la qualité reçue lorsque la vidéo n'est pas impactée par un bruit impulsif étant donné que certains chunks n'ont pas été transmis pour laisser ces q sous-canaux libres pour l'estimation du bruit impulsif. Afin d'optimiser q le nombre de sous-canaux réservés pour la correction du bruit impulsif, un modèle phénoménologique (MP) est introduit. Celui-ci, validé expérimentalement (écart maximal de 0.5dB) permet de modéliser la variance de l'erreur résiduelle $v_{\mathrm{r}} = v_{\mathrm{I}} - \hat{v}_I$ après correction du bruit impulsif. Cette erreur est utilisée pour estimer la valeur de q donnant le meilleur compromis entre correction du bruit impulsif et réduction du PSNR maximal pouvant être obtenu à la réception (i.e., sous l'hypothèse qu'aucun bruit impulsif n'est présent).

1.4 Conclusion

Nous avons tout d'abord explicité dans ce chapitre, tous les blocs constituant le schéma SoftCast et permettant d'obtenir une qualité vidéo évoluant de manière linéaire par rapport à la qualité du canal de transmission. Dans un second temps, nous avons présenté les variantes les plus significatives de SoftCast qui ont été proposées (DCast, WaveCast, etc.). Ces variantes apportent diverses améliorations au niveau de la partie traitement vidéo et/ou télécommunication avec prise en compte de canaux plus sévères (e.g., ParCast+) permettant d'améliorer la qualité reçue à la réception. De la même manière, nous avons également présenté quelques schémas ayant intégrés les propriétés du système visuel humain (HVS) comme GCast, CGCast, VCast, etc. Parmi les récents travaux, nous pouvons également citer :

- Des méthodes de débruitage comme : Cui *et al.* [39] qui utilisent un filtre BM3D (Block matching and 3D filtering, [17]) ou encore DAC-Mobi [107] et KMV-Cast [48] où l'image est reconstruite grâce au support du "cloud".
- FreeCast où les applications multivues et multivues avec profondeur (MVD) sont proposées dans [26, 28].
- Holocast [29] qui a été proposée pour répondre à la transmission de contenus holographiques 3D.
- Des schémas basés sur le deep learning et sur le Compressive-Sensing (CS) ont été proposés par Wu *et al.* [105] : DCSRN-Cast (Deep Compressed Sensing Residual Neural Network) et DCSFCN-Cast (Deep Compressed Sensing Fully Connected Network). La différence entre les deux versions repose sur le type de réseau de neurones utilisé. La dernière version permet d'obtenir un compromis entre qualité de reconstruction et complexité calculatoire.

Bien que les variantes de SoftCast améliorent la qualité reçue, ils recourent à des mécanismes plus complexes comme par exemple WaveCast introduisant un filtre MCTF ou encore les schémas hybrides réintroduisant les blocs d'estimation/compensation de mouvement des standards classiques. En fonction de l'application visée (exemple : embarqué, contraintes d'énergie, etc.), ces solutions peuvent être difficiles à mettre en place et une solution basée sur SoftCast peut être préférable.

Dans les chapitres suivants, nous introduisons nos contributions par rapport à cet état de l'art. Ainsi, des modèles théoriques d'évaluation de la qualité améliorant celui proposé par Xiong *et al.* [111] sont tout d'abord présentés dans le Chapitre 2.

Chapitre 2

Modèles théoriques d'évaluation de la qualité de bout en bout

Sommaire

2.1 Introduction

Comme nous venons de voir dans le chapitre précédent (Section 1.2.7), les performances théoriques de SoftCast ont été évaluées à l'aide du concept d'*activité des données* H. Cependant, ce modèle n'est pas général puisqu'il n'inclut pas :

- Les canaux à bande passante restreinte (i.e., nécessitant une compression du signal vidéo avant transmission) ;
- Les bénéfices de l'utilisation de l'estimateur LLSE par rapport au ZF ;
- Le recours à l'allocation optimale de puissance utilisée dans SoftCast+ [58, 107, 129], c'est-à-dire lorsque l'émetteur dispose d'une estimation de la qualité du canal.

Dans ce chapitre, nous proposons donc des modèles prenant en considération les différentes parties mentionnées ci-dessus.

Afin de faciliter la compréhension et la lecture de ce chapitre, nous rappelons tout d'abord brièvement les relations théoriques obtenues par Xiong *et al.* déjà présentées dans la Section 1.2.7.

De plus, nous introduisons la chaine de transmission simplifiée d'un schéma basé SoftCast dans la Fig. 2.1, où α_i représente le type d'estimateur utilisé à la réception (ZF ou LLSE), et les β_i concernent soit l'allocation quasi-optimale (classiquement utilisé dans SoftCast, (1.5)) soit l'allocation optimale (SoftCast+, (1.6)) :

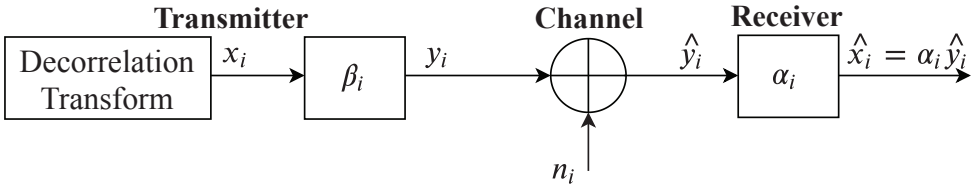


FIGURE 2.1 : Chaine de transmission généralisée d'un schéma basé SoftCast.

L'ensemble des modèles théoriques proposés ici repose sur l'hypothèse d'un canal AWGN. Bien que ce type de canal ne représente pas totalement ce qui se peut se produire dans des environnements sans-fil (évanouissement, erreurs en rafale, etc.), cette hypothèse est souvent utilisée dans un contexte SoftCast [54, 117, 63, 123, 131]. La principale raison est donnée ci-dessous [63]. Tout d'abord, nous rappelons que SoftCast transforme les chunks en slices à l'aide de la transformée de Hadamard. Cette dernière, joue le rôle de blanchiment des données en s'assurant que chaque slice possède environ la même énergie. Quand ces slices sont transmises dans le canal, elles peuvent être soumises à différents évanouissements. Au niveau du récepteur, elles sont alors divisées par leur coefficient d'évanouissement respectifs ce qui implique que la distribution de la puissance du bruit est non homogène sur les données.

Toutefois, grâce à la transformée de Hadamard inverse, la puissance du bruit est redistribuée et blanchie sur tous les chunks, ce qui peut être approximé par un canal AWGN.

Nous notons que la transformée de Hadamard W n'est pas considérée dans l'analyse théorique car elle ne modifie pas les caractéristiques de puissance de transmission et de bruit du canal AWGN [111]. En effet, si l'on considère un bruit AWGN quelconque :

- La transformée de Hadamard conserve la puissance car elle est orthogonale. En effet, notons $\mathbf{y} = W \cdot \mathbf{u}$, la puissance de $\mathbf{y} = \mathrm{E}\{\mathbf{y^T y}\} = \mathrm{E}\left\{(W\boldsymbol{u})^T(W\boldsymbol{u})\right\} = \mathrm{E}\left\{\boldsymbol{u}^T W^T W \boldsymbol{u}\right\} = \mathrm{E}\left\{\boldsymbol{u}^T\boldsymbol{u}\right\}$ = puissance de $\boldsymbol{u}$ puisque $W^T = W^{-1}$.
- En outre, comme la matrice de Hadamard est symétrique, si n représente un bruit additif blanc gaussien alors $n' = W^{-1} \cdot n = W \cdot n$ est également un bruit ayant les mêmes caractéristiques. Pour démontrer cela, il suffit juste de vérifier que Cov(n)=Cov(n') où Cov(n) dénote la matrice de covariance du bruit n. En effet, comme $\mathrm{E}\{n'\} = W \cdot \mathrm{E}\{n\}$, alors :

$$\begin{aligned} \mathrm{Cov}\,(n') &= \mathrm{E}\left\{(n' - \mathrm{E}\{n'\}) \cdot (n' - \mathrm{E}\{n'\})^T\right\} \\ &= \mathrm{E}\left\{(W \cdot n - \mathrm{E}\{W \cdot n\}) \cdot (W \cdot n - \mathrm{E}\{W \cdot n\})^T\right\} \\ &= W \cdot \mathrm{E}\left\{(n - \mathrm{E}\{n\}) \cdot (n - \mathrm{E}\{n\})^T\right\} \cdot W^T \\ &= W \cdot \mathrm{Cov}(n) \cdot W^T = \mathrm{Cov}(n), \text{ si } \mathrm{Cov}(n) = \sigma_n^2 I, \text{ où } I \text{ est la matrice identité.} \end{aligned} \tag{2.1}$$

On rappelle que les codeurs vidéo linéaires utilisent une étape de décorrélation telle que la DCT [53] ou encore la transformée en ondelettes (DWT) [20]. Dans ce travail, nous nous focalisons sur la transformée DCT, cependant tous les développements, y compris les modèles théoriques proposés, sont valables quelle que soit la transformation orthogonale considérée (par exemple, DCT-2D pour les images, DCT-3D ou encore DWT-3D pour les vidéos, etc.).

Dans la suite de ce chapitre, pour plus de clarté, nous dénotons par :

- SoftCast(ZF) : le schéma basé sur SoftCast utilisant l'estimateur ZF et l'allocation de puissance quasi-optimale ;
- SoftCast(LLSE) : le schéma basé sur SoftCast utilisant l'estimateur LLSE et l'allocation de puissance quasi-optimale ;
- SoftCast+ : le schéma basé sur SoftCast utilisant l'allocation de puissance optimale et l'estimateur LLSE (OPA-LLSE).

De plus, le cas où tous les coefficients peuvent être transmis (i.e., aucune restriction de bande passante, CR=1) est dénommé par l'acronyme FB (Full Bandwidth). De la même manière, les applications à bande passante limitée pour lesquelles seule une partie des coefficients peuvent être transmis sont représentées par l'acronyme CB (Constrained-Bandwidth).

Afin de faciliter la lecture de ce chapitre, nous rappelons tout d'abord brièvement dans la section suivante le modèle de Xiong introduit dans le Chapitre 1, puis nous introduisons les nouveaux modèles, incluant :

1. Les contraintes de bande passante ;
2. L'estimateur LLSE ;
3. L'allocation de puissance optimale.

2.2 Description des modèles théoriques pour l'estimateur ZF

2.2.1 Rappel de l'existant (Modèle de Xiong)

Xiong *et al.* ont montré que la distorsion totale attendue dans un schéma basé SoftCast, dénotée par $\mathrm{D}_{[\mathrm{ZF/FB}]}$ ici, est donnée par :

$$\mathrm{D}_{[\mathrm{ZF/FB}]} = \sum_{i=1}^{N} D_i = \frac{\sigma_n^2}{P} \left(\sum_{i=1}^{N} \sqrt{\lambda_i} \right)^2, \tag{2.2}$$

où P représente la puissance totale disponible à l'émission, σ_n^2 la puissance du bruit du canal et λ_i l'énergie des coefficients transmis x_i [53] (voir Fig. 2.1).

En se basant sur les définitions suivantes du CSNR et du PSNR, exprimées en décibels :

$$\mathrm{CSNR} = 10\log_{10}(\bar{P}/\sigma_n^2), \quad \bar{P} = P/N, \tag{2.3}$$

$$\mathrm{PSNR} = 10\log_{10}(255^2/\bar{D}), \quad \bar{D} = D/N. \tag{2.4}$$

Ils ont montré que la qualité vidéo reconstruite peut être modélisée par :

$$\mathrm{PSNR}_{[\mathrm{ZF/FB}]} = c + \mathrm{CSNR} - 20\log_{10}(H), \tag{2.5}$$

où $c = 20\log_{10}(255)$, et

$$H = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\lambda_i}, \tag{2.6}$$

représente *l'activité des données* (*data activity*). A CSNR fixé, une grande valeur de H entraîne une qualité de reconstruction faible en termes de PSNR. On observe une caractéristique linéaire du $\mathrm{PSNR}_{[\mathrm{ZF/FB}]}$ qui est fonction de la qualité du canal de transmission (CSNR).

Toutefois, comme nous l'avons noté précédemment, le modèle de Xiong *et al.* est proposé via deux hypothèses simplificatrices : la première est que la bande passante disponible pour l'application permet la transmission de l'ensemble des N coefficients (i.e., aucune compression

de la vidéo n'est effectuée, tous les coefficients sont transmis). En pratique, ce n'est généralement pas le cas, la bande passante est limitée, et c'est d'autant plus vrai lorsque le format vidéo est en haute résolution (HD, 4K, etc.). La seconde concerne le fait qu'à la réception, un estimateur ZF est utilisé. Ce n'est pas ce qui est utilisé par le schéma de base SoftCast proposé par Jakubczak *et al.* [53] qui utilise un décodeur LLSE. Nous proposons tout d'abord dans la section suivante d'étendre ce modèle ZF au cas où la bande passante de l'application est limitée, impliquant une compression des données vidéo avant transmission.

2.2.2 Description du modèle ZF proposé incluant les applications à bande passante limitée (CB)

Nous proposons d'étendre l'étude de Xiong *et al.* en considérant un cas plus général, c'est-à-dire le cas où seuls les K (avec $K \leq N$) éléments les plus énergétiques (après DCT-3D) peuvent être transmis en raison de la contrainte de bande passante de l'application. Ainsi, la distorsion totale $\mathrm{D}_{[\mathrm{ZF/CB}]}$ se compose désormais de deux termes :

- La distorsion D_i qui affecte les K coefficients transmis x_i, donnée par : $D_i = E\big[(\hat{x}_i - x_i)^2\big]$.
- La distorsion D_j due au $(N-K)$ coefficients jetés x_j, données par : $D_j = E\big[(0 - x_j)^2\big]$.

Par conséquent, au lieu de la distorsion (2.2) il faut considérer :

$$\begin{aligned} \mathrm{D}_{[\mathrm{ZF/CB}]} &= \sum_{i=1}^{K} D_i + \sum_{j=K+1}^{N} D_j, \\ &= \frac{\sigma_n^2}{P} \left(\sum_{i=1}^{K} \sqrt{\lambda_i} \right)^2 + \sum_{j=K+1}^{N} \lambda_j. \end{aligned} \tag{2.7}$$

Nous notons que la puissance moyenne de transmission totale dans (2.4) devient ici $\bar{P} = P/K$ car elle est maintenant répartie sur les seuls K coefficients transmis.

Soit $c = 20\log_{10}(255)$. En insérant (2.7) dans (2.4), nous obtenons :

$$\begin{aligned} \mathrm{PSNR}_{[\mathrm{ZF/CB}]} &= 10\log_{10}\left(\frac{255^2 \cdot N}{D_i + D_j}\right), \\ &= c - 10\log_{10}\left(1 + \frac{D_j}{D_i}\right) + 10\log_{10}\left(\frac{\bar{P}}{\sigma_n^2}\right) \\ &\quad - 10\log_{10}\left(\frac{1}{NK}\left(\sum_{i=1}^{K}\sqrt{\lambda_i}\right)^2\right). \end{aligned} \tag{2.8}$$

Par analogie avec (2.5), nous identifions la nouvelle *activité des données* transmises H_t comme suit :

$$H_t = \frac{1}{\sqrt{NK}} \sum_{i=1}^{K} \sqrt{\lambda_i}. \tag{2.9}$$

Pour faciliter la lecture, nous définissons également E_d, l'énergie globale de tous les coefficients jetés :

$$E_d = \frac{1}{N} \sum_{j=K+1}^{N} \lambda_j. \tag{2.10}$$

Selon ces nouvelles définitions, la qualité vidéo de bout en bout tenant compte des contraintes de bande passante (pour l'estimateur ZF) est finalement donnée par :

$$\begin{aligned} \mathrm{PSNR}_{[\mathrm{ZF/CB}]} = c + \mathrm{CSNR} - 20\log_{10}(H_t) \\ - 10\log_{10}\left(1 + \frac{\mathrm{CSNR}_{lin} \cdot E_d}{{H_t}^2}\right). \end{aligned} \tag{2.11}$$

où $\mathrm{CSNR}_{lin} = \frac{\bar{P}}{\sigma_n^2}$ représente le CSNR exprimé en linéaire.

Comparé à (2.5), l'équation ci-dessus comprend un nouveau terme qui reflète l'effet de la compression appliquée sur les données. Le PSNR dépend maintenant de trois paramètres : le CSNR, qui dépend des conditions de transmission, puis les deux autres termes E_d et H_t qui sont directement liés aux caractéristiques du contenu vidéo transmis.

Pour une bande passante donnée, plus le terme E_d (représentant les données jetées) est élevé, plus la dégradation est importante. Cependant, comme celui-ci est multiplié par le CSNR_{lin}, la dégradation devient moins perceptible dans les environnements à CSNR faibles (son effet sera masqué par le bruit du canal).

Lorsque $K = N$, c'est-à-dire qu'aucune compression des données n'est appliquée, (2.11) et (2.5) sont identiques. En d'autres termes, la qualité de la vidéo ne varie que linéairement avec le CSNR, comme indiqué dans [111].

2.2.3 Analyse des performances du modèle ZF proposé

L'efficacité du modèle proposé est comparée ci-dessous à celle du schéma d'origine Soft-Cast. Par conséquent, sans perte de généralité pour (2.11), les coefficients transformés sont regroupés en chunks et tous les blocs (DCT-3D, mise à l'échelle, etc.) sont implémentés conformément à [53]. Toutefois, pour cette section, l'estimateur ZF est utilisé à la place du LLSE. La modélisation pour cet estimateur est donnée dans la section suivante.

Le modèle ZF est évalué par le biais de nombreuses simulations en utilisant des séquences HD720p (classe E, 1280 × 720 pixels, 30 fps) et des séquences CIF (352 × 288 pixels, 30

fps) issues des collections Xiph [113] et/ou du JCT-VC (Joint Collaborative Team on Video Coding) lors de la standardisation du standard HEVC [103]. Seule la luminance de chaque vidéo est considérée. Le processus est exécuté GoP par GoP avec une taille de GoP de 16 images comme dans [53]. Chaque image est classiquement divisée en 64 chunks [27, 96, 53].

Des transmissions via des canaux AWGN dans la plage de [0∼30dB] sont considérées comme dans [53, 27, 42]. La puissance de transmission est normalisée, c'est-à-dire $P = 1$. Dans ce chapitre, quatre niveaux de compression sont considérés : CR=1, 0.75, 0.5 et 0.25.

Parmi les différentes configurations (contenu vidéo, taille de GoP, etc.), nous avons choisi d'afficher uniquement les résultats pour les séquences HD720p, les résultats pour les séquences CIF étant similaires. Nous créons tout d'abord une séquence composite (*Mixed HD720p*) en concaténant les 128 premières images des dix séquences suivantes [113] : *Ducks, Four People, Into tree, Johnny, Kristen and Sara, Old town, Parkjoy, Parkrun, Shields* et *Stockholm*. La taille du GoP et le nombre de chunks par image sont respectivement définis à 16 images et 64 chunks car ils représentent la configuration d'origine et la plus utilisée [53]. Nous avons vérifié des résultats similaires pour d'autres tailles de GoP (4, 8 et 32) et de chunks (découpage de chaque image en 256 chunks au lieu de 64).

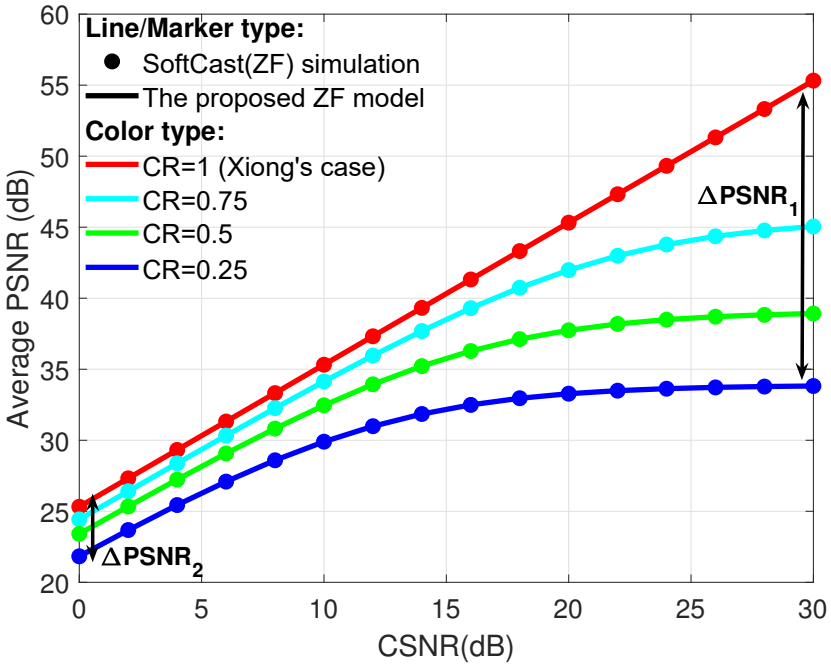


FIGURE 2.2 : Evolution du PSNR moyen obtenu en fonction du CSNR considéré pour le modèle théorique ZF proposé (trait continu) et les simulations SoftCast avec l'estimateur ZF : (points) pour la séquence *Mixed HD720p*. Configuration : taille de GoP = 16 images, 64 chunks/image. Les couleurs rouge, cyan, vert et bleu représentent respectivement les CR = 1, 0.75, 0.5 et 0.25.

La Fig. 4.3 présente la comparaison entre notre modèle et des simulations complètes de transmission pour le schéma SoftCast(ZF).

Nous observons que :

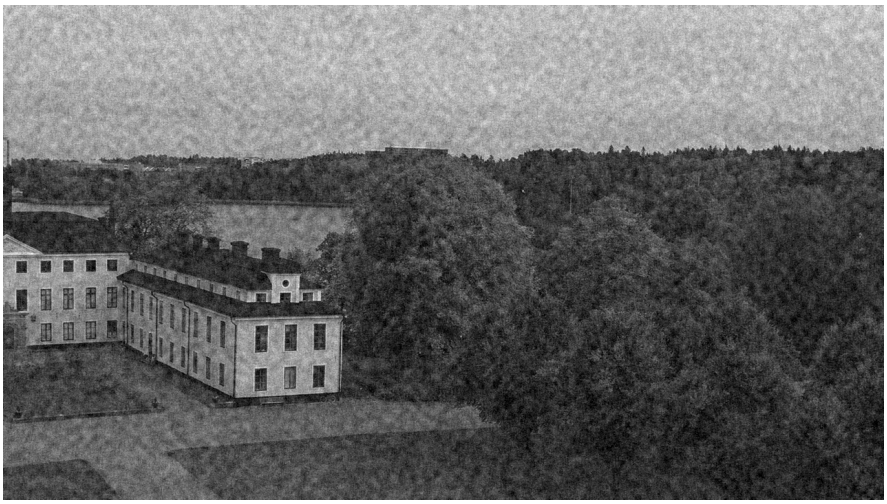
- Sans compression de données (FB, CR=1), c'est-à-dire $K = N$ (courbe rouge), et en supposant un estimateur ZF, les mêmes caractéristiques linéaires sont obtenues comme dans [111]. Notons toutefois que le modèle de Xiong *et al.* est limité au cas FB ;
- Lorsque la bande passante disponible du canal diminue (courbes cyan, verte et bleue), les courbes deviennent concaves pour des valeurs de CSNR élevées : l'effet bien connu de *saturation* de la qualité (*levelling-off*) apparaît [64] et est parfaitement décrit par notre modèle ;
- Dans tous les cas, notre modèle correspond parfaitement aux simulations sur toute la gamme de CSNR considérée, indépendamment du niveau de bande passante disponible considéré. Ceci est dû au fait qu'aucune approximation n'est faite dans le processus de dérivation de l'équation (2.11).

Nous notons que l'effet de *saturation* (*levelling-off*) mentionné ci-dessus devient beaucoup plus visible lorsque la bande passante disponible diminue (courbe bleue). Contrairement au modèle de Xiong *et al.*, le modèle proposé représente parfaitement les changements de qualité obtenus pour toutes les valeurs de CSNR, quelle que soit la quantité de données supprimée à l'émission. Comme mentionné précédemment, ceci est dû au fait que l'équation (2.11) intègre un terme supplémentaire qui prédit et modélise cet effet avec précision.

En termes de qualité vidéo, il est logique d'observer des scores de PSNR moins bons pour les cas où la bande passante est limitée. Au fur et à mesure que le CSNR diminue, la différence de PSNR entre deux cas de bande passante différente indiquée par ΔPSNR devient plus faible. Par exemple, le ΔPSNR entre le CR=1 et le CR=0.25 passe respectivement de $\Delta\text{PSNR}_1 = 21.5$dB à $\Delta\text{PSNR}_2 = 3.5$dB pour un CSNR de 30dB et de 0dB. Comme mentionné ci-dessus, cela s'explique parfaitement avec le modèle proposé. On peut également noter que le PSNR reconstruit passe en dessous de 35dB à des valeurs de CSNR faibles (< 10 dB). Dans de tels cas, les normes classiques telles que H.264/AVC ou HEVC offriraient une qualité inférieure et souffriraient de gel d'images en raison de graves erreurs de décodage (cf. Fig. 6 de l'Introduction). Au contraire, SoftCast fonctionne pour des qualités de canal plus faibles en fournissant un signal vidéo avec une qualité faible mais néanmoins acceptable [53]. Une illustration est donnée dans la Fig. 2.3.

L'efficacité du modèle ZF proposé est validée pour toutes les gammes de CSNR considérées et qu'importe le niveau de compression appliqué. Cependant, comme mentionné dans la Section 1.2.7, l'estimateur LLSE est plus couramment utilisé, comme dans les travaux originaux de Jakubczak *et al.* [53] ou comme rapporté dans [20, 125, 42, 127, 61, 96, 101, 27]. Par conséquent, nous proposons dans la section suivante un nouveau modèle intégrant les bénéfices de l'utilisation de l'estimateur LLSE au récepteur.

(a) Image originale



(b) PSNR=25.38dB

FIGURE 2.3 : Comparaison de la qualité visuelle obtenue à un CSNR = 0dB et CR=0.5 pour la séquence *Mixed HD720p* (image N°.257), taille de GoP=16. (a) image originale, (b) SoftCast(ZF).

2.3 Description des modèles proposés pour l'estimateur LLSE et l'allocation de puissance quasi-optimale

En rappelant la Fig. 2.1 et en considérant l'estimateur LLSE [53, 27] au lieu de l'estimateur ZF, (1.18) devient :

$$\begin{aligned} \hat{x}_i &= \alpha_i \cdot \hat{y}_i, \\ &= \frac{g_i \lambda_i}{g_i^2 \lambda_i + \sigma_n^2} \cdot \hat{y}_i. \end{aligned} \tag{2.12}$$

De même, (1.19) s'écrit :

$$\begin{aligned} D_{i[\text{LLSE/FB}]} &= E\left[(\hat{x}_i - x_i)^2\right], \\ &= \frac{\sigma_n^2 \alpha_i}{g_i}, \\ &= \frac{\sigma_n^2 \lambda_i}{g_i^2 \lambda_i + \sigma_n^2}, \\ &= \frac{\sigma_n^2 \lambda_i}{P_i + \sigma_n^2}. \end{aligned} \tag{2.13}$$

Les équations (1.17) et (1.20) restent identiques.

Comme dans la Section 1.2.7, nous voulons exprimer P_i en fonction de $D_{i[\text{LLSE/FB}]}$. Rappelons pour cela l'équation (1.5) du facteur de mise à l'échelle utilisé dans l'allocation de puissance quasi-optimale de [53] (voir la Section 1.2) : $g_i = \sqrt{\frac{P}{\sqrt{\lambda_i} \sum_j \sqrt{\lambda_j}}}$. En l'insérant dans (1.20), nous obtenons :

$$\begin{aligned} P_i &= g_i^2 \lambda_i, \\ &= \frac{P \sqrt{\lambda_i}}{\sum_j \sqrt{\lambda_j}}. \end{aligned} \tag{2.14}$$

Par conséquent, la distorsion par élément du signal peut être exprimée comme suit :

$$\begin{aligned} D_{i[\text{LLSE/FB}]} &= \frac{\sigma_n^2 \lambda_i}{\frac{P\sqrt{\lambda_i}}{\sum_j \sqrt{\lambda_j}} + \sigma_n^2}, \\ &= \frac{\lambda_i}{\frac{\sqrt{\lambda_i}}{\sum_j \sqrt{\lambda_j}} \cdot \frac{P}{\sigma_n^2} + 1}, \end{aligned} \tag{2.15}$$

$$= \frac{\lambda_i}{\frac{\sqrt{\lambda_i}}{\sum_j \sqrt{\lambda_j}} \cdot \frac{P}{\sigma_n^2} + 1},$$
$$= \frac{\lambda_i}{\frac{\sqrt{\lambda_i}}{\sum_j \sqrt{\lambda_j}} \cdot (\mathrm{CSNR}_{lin} \cdot N) + 1}.$$

En utilisant (2.14), la distorsion du LLSE peut également être définie selon celle du ZF comme suit :

$$D_{i[\mathrm{LLSE/FB}]} = \frac{\sigma_n^2 \lambda_i}{P_i + \sigma_n^2}. \tag{2.16}$$

En utilisant (1.28), nous obtenons facilement :

$$D_{i[\mathrm{LLSE/FB}]} = D_{i[\mathrm{ZF/FB}]} \cdot \frac{1}{1 + \frac{\sigma_n^2}{P_i}}. \tag{2.17}$$

En supposant que l'émetteur puisse transmettre tous les N éléments de $\boldsymbol{x}$ (c'est-à-dire dans un canal sans restriction de bande passante) et que l'estimateur LLSE soit utilisé du côté du récepteur, la distorsion totale attendue dans le cadre d'une allocation de puissance quasi-optimale pour le cas LLSE, notée $\mathrm{D}_{[\mathrm{LLSE/FB}]}$ est donnée par :

$$\begin{aligned} \mathrm{D}_{[\mathrm{LLSE/FB}]} &= \sum_{i=1}^{N} D_{i[\mathrm{LLSE/FB}]}, \\ &= \sum_{i=1}^{N} \frac{\lambda_i}{\frac{\sqrt{\lambda_i}}{\sum_j \sqrt{\lambda_j}} \cdot (\mathrm{CSNR}_{lin} \cdot N) + 1}, \\ &= \sum_{i=1}^{N} D_{i[\mathrm{ZF/FB}]} \cdot \frac{1}{1 + \frac{\sigma_n^2}{P_i}}. \end{aligned} \tag{2.18}$$

En rappelant les équations du CSNR (2.3) et du PSNR (2.4), nous pouvons obtenir l'expression théorique de la qualité vidéo attendue au récepteur en considérant le cas FB et l'estimateur LLSE :

$$\mathrm{PSNR}_{[\mathrm{LLSE/FB}]} = c - 10 \log_{10} \left(\frac{1}{N} \sum_{i=1}^{N} \frac{\lambda_i}{\frac{\sqrt{\lambda_i}}{\sum_j \sqrt{\lambda_j}} \cdot (\mathrm{CSNR}_{lin} \cdot N) + 1} \right), \tag{2.19}$$

où $c = 20 \log_{10}(255)$.

Cependant, cette équation ne peut pas être simplifiée car le second terme $\frac{1}{1+\frac{\sigma_n^2}{P_i}}$ dans (2.18) n'est pas une constante et dépend de P_i. Néanmoins, en considérant l'approximation

$P_i \simeq P/N$, nous pouvons obtenir un modèle simplifié pour les schémas basés SoftCast utilisant une allocation de puissance quasi-optimale et l'estimateur LLSE à la réception. Dans ce cas, $\mathrm{D}_{[\mathrm{LLSE/FB}]}$ devient :

$$\mathrm{D}_{[\mathrm{LLSE/FB}]^*} = \sum_{i=1}^{N} D_{i[\mathrm{ZF/FB}]} \cdot \frac{1}{1+\frac{N\sigma_n^2}{P}}. \tag{2.20}$$

Pour faciliter la compréhension, ce cas est dénoté par $\mathrm{D}_{[\mathrm{LLSE/FB}]^*}$. En rappelant les équations (2.3) et (2.4), nous montrons que la qualité vidéo reconstruite attendue en considérant l'estimateur LLSE sans compression de données est donnée par :

$$\begin{aligned}
\mathrm{PSNR}_{[\mathrm{LLSE/FB}]^*} &= c - 10\log_{10}\left(\frac{\mathrm{D}_{[\mathrm{LLSE/FB}]^*}}{N}\right), \\
&= c - 10\log_{10}\left(\mathrm{D}_{[\mathrm{ZF/FB}]}\right) \\
&\quad + 10\log_{10}\left(1+\frac{1}{\mathrm{CSNR}_{lin}}\right), \\
&= c + \mathrm{CSNR} - 20\log_{10}\left(H\right) \\
&\quad + 10\log_{10}\left(1+\frac{1}{\mathrm{CSNR}_{lin}}\right).
\end{aligned} \tag{2.21}$$

Lorsque $P_i \simeq P/N$, et seulement dans ce cas, le modèle théorique d'un schéma basé SoftCast avec une allocation de puissance quasi-optimale et l'estimateur LLSE est donné par :

$$\mathrm{PSNR}_{[\mathrm{LLSE/FB}]^*} = \mathrm{PSNR}_{[\mathrm{ZF/FB}]} + G_{\mathrm{LLSE}}, \tag{2.22}$$

où $G_{\mathrm{LLSE}} = 10\log_{10}\left(1+\frac{1}{\mathrm{CSNR}_{lin}}\right)$.

Comme indiqué dans la Section 2.2.2, ce modèle peut être étendu aux cas (CB) en ajoutant un terme supplémentaire représentant les coefficients supprimés :

$$\begin{aligned}
\mathrm{PSNR}_{[\mathrm{LLSE/CB}]^*} &= c + \mathrm{CSNR} - 20\log_{10}\left(H_t\right) \\
&\quad + G_{\mathrm{LLSE}} \\
&\quad - 10\log_{10}\left(1+\frac{(\mathrm{CSNR}_{lin}+1)\cdot E_d}{{H_t}^2}\right), \\
&\simeq \mathrm{PSNR}_{[\mathrm{ZF/CB}]} + G_{\mathrm{LLSE}}.
\end{aligned} \tag{2.23}$$

La démonstration est disponible en Annexe B.

De manière similaire, la qualité vidéo reconstruite attendue en considérant le cas général ($P_i \not\approx P/N$), l'estimateur LLSE et des applications à bande passante limitée peut être obtenue à partir de :

$$\mathrm{PSNR}_{[\mathrm{LLSE/CB}]} = c - 10\log_{10}\left({D_i}/{N} + {D_j}/{N}\right), \tag{2.24}$$

où D_i et D_j sont obtenus de manière similaire à l'équation (2.7).

Notons que (2.23) ressemble à (2.11), excepté que le cinquième et dernier terme inclut ($\text{CSNR}_{lin} + 1$) au lieu de (CSNR_{lin}) pour (2.11), en raison de l'utilisation de l'estimateur LLSE. A partir des deux équations (2.22) et (2.23), on peut constater que la différence entre le modèle ZF et le modèle LLSE$*$ est définie par le terme G_{LLSE}. Nous donnons dans le Tableau 2.1 les valeurs numériques correspondantes du gain G_{LLSE} pour plusieurs valeurs de CSNR :

Tableau 2.1 : Evolution du gain apporté par le LLSE par rapport au ZF pour différentes valeurs de CSNR

CSNR	0	5	10	15	20	25
$\mathbf{CSNR}_{lin}$	1	3.16	10	31.6	100	316
$\mathbf{G_{LLSE}}$	3.01	1.193	0.413	0.135	0.043	0.013

Nous pouvons conclure à partir du Tableau 2.1 qu'au-dessus de 10dB, les améliorations apportées par l'estimateur LLSE en termes de scores PSNR sont insignifiantes. Ceci est cohérent avec [53, 111] et est confirmé ci-dessous. Dans ce qui suit, nous vérifions d'abord la validité de ce modèle en considérant un signal aléatoire gaussien centré comme signal d'entrée du schéma (Fig. 2.1). En effet, ceci nous permet de vérifier l'approximation $P_i \simeq P/N$ ou $P_i \simeq P/K$ en cas de contrainte de bande passante (CB). Nous évaluons ensuite l'écart entre ce modèle approximé considérant l'estimateur LLSE et l'allocation de puissance quasi-optimale (c'est-à-dire, les travaux originels de Jakubczak *et al.* [53]) et les simulations considérant certaines distributions de puissance du signal d'entrée générées aléatoirement. Enfin, des simulations sont effectuées en considérant des distributions énergétiques issues d'images ou de contenu vidéo réels.

2.3.1 Analyse des performances du modèle LLSE proposé

L'efficacité du modèle proposé est comparée à celle du schéma originel SoftCast (i.e., considérant l'allocation de puissance quasi-optimale et l'estimateur LLSE).

Le modèle est tout d'abord évalué via des simulations en utilisant des distributions énergétiques générées aléatoirement utilisées comme signal d'entrée du schéma. Précisément, une matrice de 512 chunks composés chacun de 36×44 coefficients aléatoires est créée et représente ainsi ce que l'on peut classiquement obtenir dans un schéma basé SoftCast en supposant la transmission de séquences CIF avec une taille GoP de 8 images et une découpe de 64 chunks/image. Chaque chunk est ensuite ajusté par multiplication avec une constante définie ci-après. Nous supposons sans perte de généralité qu'un CR=1 est utilisé pour la transmission (pas de compression des données vidéo).

Des transmissions via des canaux AWGN dans la plage [0∼30dB] sont considérées comme dans la Section 2.2.2.

Pour vérifier la validité du modèle approximatif, nous générons des nombres aléatoires normalement distribués (jouant le rôle de chunks) sans différence majeure de puissance, comme illustré dans la Fig. 2.4a. Les résultats des simulations sont donnés dans la Fig. 2.4b. Comme on peut le constater, l'approximation $P_i \simeq P/N$ étant respectée, la limite représentée par l'équation (2.21) en pointillé avec des marqueurs en croix peut être atteinte par l'estimateur LLSE. Ceci est vérifié à la fois par des simulations (lignes tiretées) et théoriquement par l'équation (2.19) représentée par les grands marqueurs en forme de cercle. Nous créons maintenant deux autres distributions de puissance aléatoires différentes, les deux ayant une valeur élevée (qui représente le chunk contenant les basses fréquences, contenant donc une énergie importante) et 511 autres. Le premier contient des valeurs relativement élevées pour ces 511 valeurs comme représenté dans la Fig. 2.4b, alors que pour l'autre représenté dans la Fig. 2.4c, 200 valeurs sont laissées presque nulles. Les résultats pour ces deux distributions de puissance sont disponibles dans les Figs. 2.4e et 2.4f, respectivement. Nous observons que :

- Les résultats de simulations pour le schéma SoftCast(LLSE) et le modèle associé sont délimités par le modèle LLSE* et par le modèle/les simulations SoftCast(ZF).
- À mesure que la distribution de puissance devient hétérogène, le gain apporté par l'estimateur LLSE diminue et l'écart entre les deux modèles proposés (i.e., LLSE et LLSE*) devient plus grand.

Nous examinons à présent l'impact d'un contenu vidéo réel sur l'écart existant entre l'estimateur ZF et l'estimateur LLSE. Pour caractériser le contenu vidéo, nous utilisons les quantités d'information spatiale et temporelle d'une séquence vidéo, définie par les index (SI) et (TI) proposés par proposés par l'Union Internationale des Télécommunications [51]. Ces index définis comme suit :

$$\text{SI} = max_{time}\{std_{space}[Sobel(I(i,j,k))]\}, \tag{2.25}$$

$$\text{TI} = max_{time}\{std_{space}[I(i,j,k) - I(i,j,k-1)]\}, \tag{2.26}$$

où $I(i,j,k)$ représente la $k^{ème}$ image, (i,j) les coordonnées spatiales correspondantes et $Sobel()$ l'opération de filtrage de Sobel, respectivement.

Cependant, comme mentionné dans [10], en raison de la définition actuelle qui sélectionne la valeur la plus élevée sur l'axe temporel, le calcul du TI pour une vidéo avec des mouvements relativement lents mais présentant un changement de plan (ou scène) résulte en une valeur élevée. Par conséquent, en raison des fortes disparités pouvant survenir au cours d'une vidéo, nous avons choisi de moyenner les résultats sur la séquence. Les nouvelles définitions des index sont donc :

$$\text{SI} = mean_{time}\{std_{space}[Sobel(I(i,j,k))]\}, \tag{2.27}$$

$$\text{TI} = mean_{time}\{std_{space}[I(i,j,k) - I(i,j,k-)]\}. \tag{2.28}$$

Ces définitions sont considérées dans le reste de cette thèse au lieu de (2.25) et (2.26).

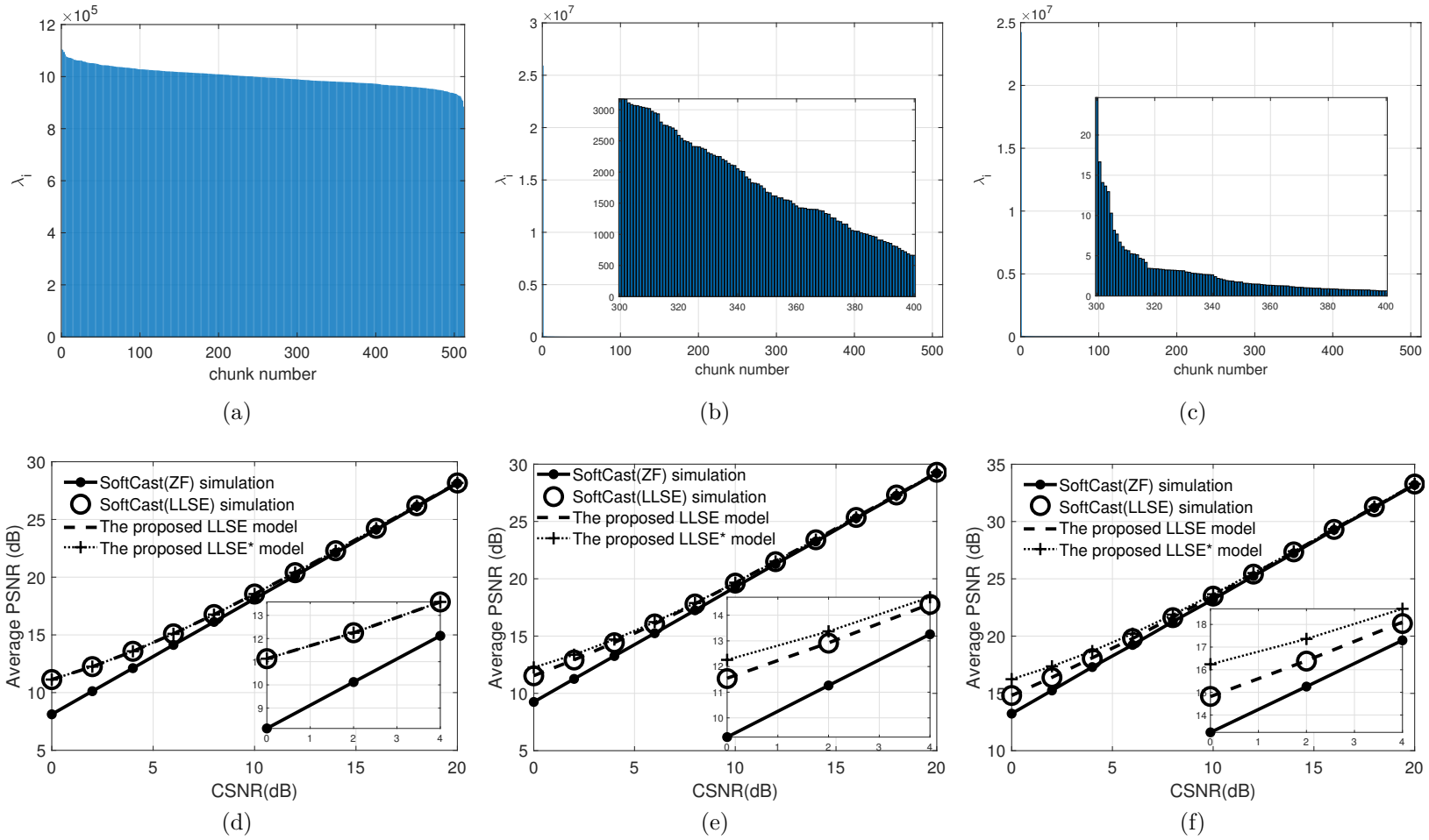


FIGURE 2.4 : Evolution du PSNR moyen obtenu pour le modèle théorique LLSE proposé (ligne tiretée), le modèle théorique approximé (ligne en pointillé avec des marqueurs en croix), les simulations SoftCast avec estimateur LLSE : (marqueurs : grands cercles) et les simulations SoftCast avec estimateur ZF : (ligne continue avec des points) pour une distribution de puissance générée aléatoirement. Configuration : taille de GoP = 8 images, 64 chunks/image, CR=1. Première ligne : Illustration de la distribution de puissance générée aléatoirement. Deuxième ligne : Résultats de PSNR moyens correspondants. (a), (d) Amplitude=[1000*ones(1,512)], (b), (e) Amplitude=[5000 100*randn(1,511)], (c), (f) Amplitude=[5000 100*randn(1,311) randn(1,200)]. Veuillez agrandir la figure pour observer les détails.

Les écarts de PSNR entre l'estimateur LLSE et l'estimateur ZF sont illustrés dans la Fig. 2.5 pour cinq séquences vidéo sélectionnées : *Akiyo, Husky, ParkJoy, Into tree* et *Shields*. Nous avons vérifié des comportements similaires pour les autres contenus vidéo mentionnés dans la Fig. 2.6. Nous observons que :

- Comme indiqué ci-dessus, l'écart entre les estimateurs ZF et LLSE varie en fonction du contenu vidéo ou plus précisément de la répartition de la puissance des chunks / coefficients transmis ;
- Ce gap se situe en dessous du modèle LLSE* qui semble définir une limite supérieure pouvant être atteinte lorsque le signal n'est pas corrélé et est uniformément réparti sur les chunks, c'est-à-dire lorsque $P_i \simeq P/N$;
- À mesure que les indices spatio-temporels augmentent, les performances de l'estimateur LLSE augmentent, comme observé avec les séquences *Husky* ou *ParkJoy*, où le signal vidéo est difficile à décorréler en raison de mouvements élevés et de contours marqués. Dans le cas contraire, les performances de l'estimateur LLSE diminuent lorsque les indices spatio-temporels sont faibles. En effet, dans de tels cas, la corrélation est élevée et ainsi, après la transformation de décorrélation, la majeure partie de l'énergie est localisée sur les chunks basses fréquences ne laissant qu'une faible part d'énergie pour les autres chunks, comme observé de la même manière dans les Fig. 2.4*c* et 2.4*f*.

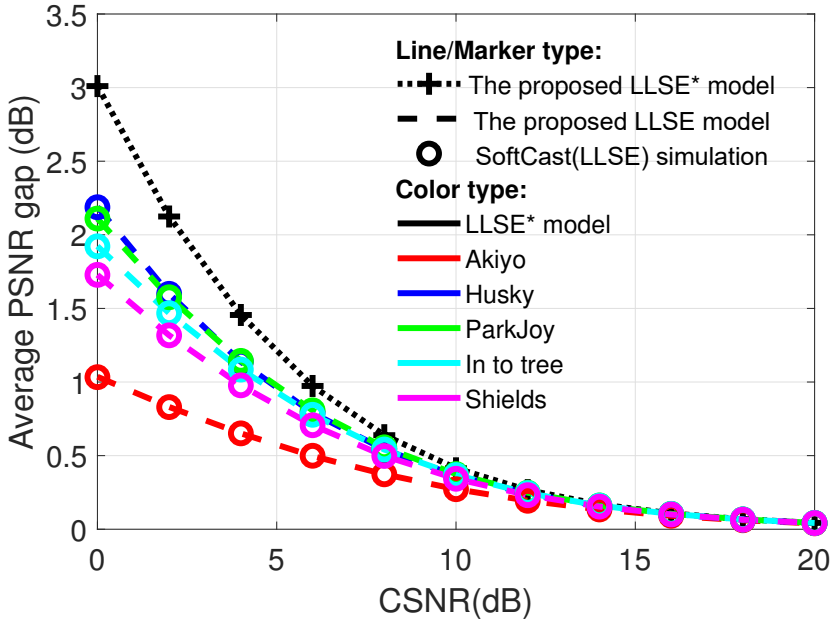


FIGURE 2.5 : Evolution de l'écart existant entre l'estimateur LLSE et l'estimateur ZF en fonction du CSNR. Configuration : taille de GoP = 16 images, CR=1. Couleurs : noir = modèle LLSE*, rouge = *Akiyo*, bleu = *Husky*, vert = *ParkJoy*, cyan = *Into tree* et magenta = *Shields*.

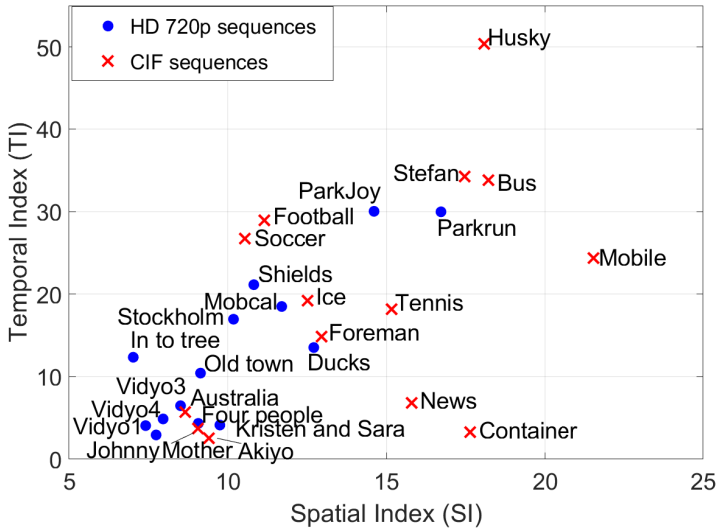


FIGURE 2.6 : Illustration des index spatio-temporels moyens pour les séquences vidéo HD720p et CIF sélectionnées.

Sur la base de ces observations et pour une évaluation rapide des performances LLSE, nous suggérons d'utiliser les équations (2.22) et (2.23) pour des contenus spatio-temporels élevés ou (2.11) pour des contenus spatio-temporels bas car le biais est limité. Pour une évaluation précise, et en particulier pour des valeurs de CSNR faibles, nous recommandons d'utiliser le modèle théorique non simplifié (2.19) pour le cas FB et le modèle (2.24) pour les applications CB, car aucune approximation n'est faite dans le processus de dérivation, ils n'introduisent donc pas de biais entre eux et les simulations de bout en bout.

Comme précédemment, nous donnons également la comparaison entre nos modèles et les simulations complètes de transmission de bout en bout avec SoftCast(LLSE) pour la séquence *Mixed HD720p* dans la Fig. 2.7. Comme nous pouvons l'observer, les résultats de simulation se situent entre la version ZF et LLSE* étant donné que la séquence composite *Mixed HD720p* contient 10 types de séquences différentes.

Nous avons vu dans cette section le modèle théorique d'un schéma basé SoftCast utilisant l'estimateur LLSE du côté du récepteur et une allocation de puissance quasi-optimale au niveau de l'émetteur, c'est-à-dire une allocation de puissance effectuée sans aucun retour de canal au niveau de l'émetteur. Ceci constitue le schéma principalement utilisé dans la littérature, car il convient au contexte de diffusion (broadcast), où un seul flux de données envoyé peut être décodé par n'importe quel récepteur. Cependant, dans certains articles, les auteurs ont supposé que les informations de l'état du canal étaient disponibles à l'émetteur [107, 58]. Dans de tels cas, une allocation de puissance optimale peut être utilisée, où certains coefficients dont l'énergie est proportionnellement inférieure au niveau de bruit sont jetés et où la puissance totale P est redistribuée entre les coefficients transmis. Nous montrons dans la section suivante que ce schéma, dénommé SoftCast+, peut être aussi modélisé théoriquement.

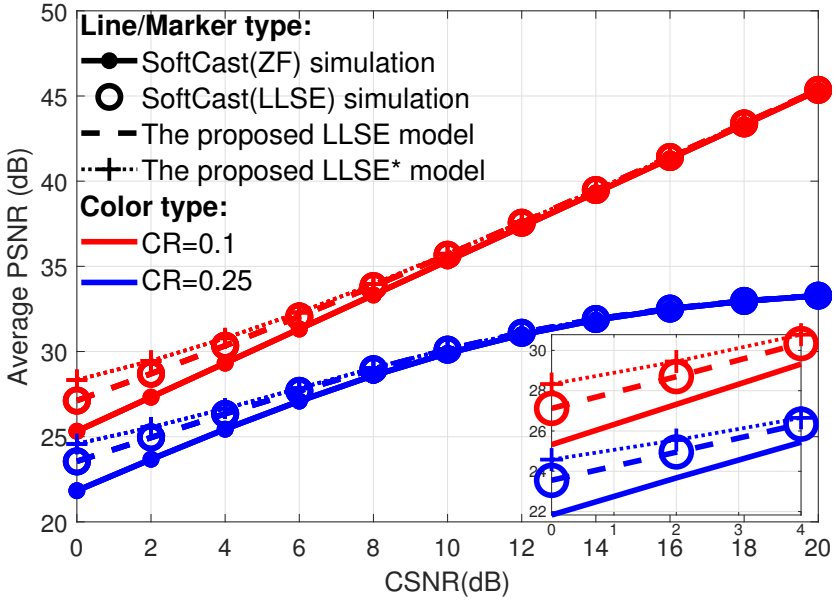


FIGURE 2.7 : Evolution du PSNR moyen obtenu pour les modèles LLSE théoriques proposés : (modèle LLSE : ligne tiretée, modèle LLSE* : ligne en pointillé et marqueurs en croix) et les simulations SoftCast(LLSE) : (grand cercle) et SoftCast(ZF) : (trait plein et points) pour la séquence *Mixed HD720p*. Configuration : taille de GoP=16 images, 64 chunks/image. Les couleurs rouge et bleu représentent respectivement les CR=1 et 0.25.

2.4 Description du modèle proposé pour l'estimateur LLSE et l'allocation de puissance optimale

Dans cette section, nous évaluons les performances de SoftCast+ [107] c'est-à-dire considérant que les informations sur la qualité du canal sont disponibles au niveau de l'émetteur et que l'estimateur LLSE est utilisé du côté du récepteur. L'allocation de puissance optimale est définie dans [58] où certains coefficients proportionnellement inférieurs au niveau de bruit sont ignorés et la puissance totale P est réaffectée aux coefficients transmis.

Par conséquent, comme dans le cas ZF/CB, la distorsion totale est composée de deux termes : $\mathrm{D}_{[\text{OPA-LLSE}]} = D_i + D_j$, où D_i et D_j correspondent respectivement à la distorsion due aux ℓ coefficients transmis et à celle due à l'allocation de puissance optimale qui jette les $N - \ell$ coefficients ayant une énergie insuffisante par rapport au niveau de bruit. La manière de calculer ℓ, le nombre optimal de coefficients transmis en tenant compte de la qualité du canal, sera détaillée ultérieurement.

La distorsion attendue pour chaque coefficient transmis $\hat{x}_i$ est la même puisque l'estimateur LLSE est utilisé du côté récepteur (voir la Section 2.3) :

$$D_i = E\big[(\hat{x}_i - x_i)^2\big] = \frac{\sigma_n^2 \lambda_i}{P_i + \sigma_n^2}. \tag{2.29}$$

Le nouveau problème d'optimisation est défini comme suit :

$$(P2) : \min \sum_{i=1}^{\ell} D_i + \sum_{i=\ell+1}^{N} D_j, s.t. \sum_{i=1}^{\ell} P_i \leq P \tag{2.30}$$

C'est un problème lagrangien :

$$\mathcal{L} = \sum_{i=1}^{N} D_i + \frac{1}{C^2_{\text{OPA-LLSE}}} \sum_{i=1}^{N} P_i, \tag{2.31}$$

où $C^2_{\text{OPA-LLSE}}$ est le multiplicateur de Lagrange.

En annulant la dérivée du lagrangien $\mathcal{L}$ (2.31) par rapport à P_i, nous obtenons :

$$\sigma_n^2 \frac{\lambda_i}{(P_i + \sigma_n^2)^2} = \frac{1}{C^2_{\text{OPA-LLSE}}}. \tag{2.32}$$

Ceci détermine la puissance optimale pour envoyer x_i :

$$P_i = C_{\text{OPA-LLSE}} \sigma_n \sqrt{\lambda_i} - \sigma_n^2. \tag{2.33}$$

Notons que seuls les coefficients de puissance $P_i > 0$ sont transmis. A partir de (2.33) on obtient la condition :

$$\lambda_i > \frac{\sigma_n^2}{C_{\text{OPA-LLSE}}}. \tag{2.34}$$

Comme la puissance totale de transmission est fixée, $C^2_{\text{OPA-LLSE}}$ peut être calculé. En effet, sans nuire à la généralité, si l'on suppose que les puissance P_i sont ordonnées par ordre décroissant et que seulement ℓ coefficients vérifient l'inégalité précédente alors :

$$\begin{aligned}
\sum_{i=1}^{\ell} P_i &= P, \qquad (2.35)\\
\sum_{i=1}^{\ell} C_{\text{OPA-LLSE}} \sigma_n \sqrt{\lambda_i} - \sigma_n^2 &= P,\\
C_{\text{OPA-LLSE}} \sigma_n \sum_{i=1}^{\ell} \sqrt{\lambda_i} - \ell \sigma_n^2 &= P,\\
C_{\text{OPA-LLSE}} \sigma_n \sum_{i=1}^{\ell} \sqrt{\lambda_i} &= P + \ell \sigma_n^2.
\end{aligned}$$

Finalement,

$$C_{\text{OPA-LLSE}} = \frac{P + \ell \sigma_n^2}{\sigma_n \sum\limits_{i=1}^{\ell} \sqrt{\lambda_i}}. \tag{2.36}$$

A partir de (2.29) et (2.33), nous obtenons facilement :

$$\begin{aligned} D_i &= \frac{\sigma_n^2}{P_i + \sigma_n^2}\lambda_i, \\ &= \frac{\sigma_n^2}{C_{\text{OPA-LLSE}}\sigma_n\sqrt{\lambda_i}}\lambda_i, \\ &= \frac{\sigma_n}{C_{\text{OPA-LLSE}}}\sqrt{\lambda_i}. \end{aligned} \tag{2.37}$$

Finalement, la distorsion totale pour le cas LLSE avec allocation de puissance optimale est donnée par [58, 107] :

$$\begin{aligned} \mathrm{D}_{[\text{OPA-LLSE}]} &= \sum_{i=1}^{N} D_i, \\ &= \sum_{i=1}^{\ell} D_i + \sum_{j=\ell+1}^{N} D_j, \\ &= \frac{\sigma_n^2\left(\sum_{i=1}^{\ell}\sqrt{\lambda_i}\right)^2}{P + \ell\sigma_n^2} + \sum_{j=\ell+1}^{N}\lambda_j. \end{aligned} \tag{2.38}$$

où P représente pour rappel la puissance totale disponible à l'émetteur, σ_n^2 est la puissance du bruit et λ_i l'énergie des coefficients transmis.

Notons que l'équation $\mathrm{D}_{[\text{OPA-LLSE}]}$ (2.38), i.e., la distorsion pour l'allocation optimale de puissance avec l'estimateur LLSE, est similaire à $\mathrm{D}_{[\text{ZF/CB}]}$ (2.7) sauf que :

1. le nombre de chunks ignorés ℓ varie en fonction de la qualité du canal (CSNR),
2. il existe un terme supplémentaire ($\ell\sigma_n^2$) au dénominateur dû à l'estimateur LLSE.

Par conséquent, le développement est similaire à la Section 2.2.2, et en utilisant les équations du CSNR (2.3) et du PSNR (2.4), nous obtenons :

$$\begin{aligned} \mathrm{PSNR}_{[\text{OPA-LLSE}]} &= 10\log_{10}\left(\frac{255^2}{D_i/_N + D_j/_N}\right), \\ &= c - 10\log_{10}\left(1 + \frac{D_j}{D_i}\right) \\ &\quad + 10\log_{10}\left(\frac{\bar{P} + \sigma_n^2}{\sigma_n^2}\right) \\ &\quad - 10\log_{10}\left(\frac{1}{N\ell}\left(\sum_{i=1}^{\ell}\sqrt{\lambda_i}\right)^2\right). \end{aligned} \tag{2.39}$$

Notons que la puissance moyenne de transmission totale dans l'équation (2.4) devient $\bar{P} = P/\ell$ car la puissance de transmission est maintenant répartie sur les seuls ℓ coefficients transmis.

Par analogie avec (2.8), nous identifions la nouvelle activité des données concernant les ℓ coefficients restants transmis comme suit :

$$H_{t2} = \frac{1}{\sqrt{N\ell}} \sum_{i=1}^{\ell} \sqrt{\lambda_i}. \tag{2.40}$$

Pour faciliter la lecture, nous définissons également E_{d2}, l'énergie globale de tous les coefficients jetés :

$$E_{d2} = \frac{1}{N} \sum_{j=\ell+1}^{N} \lambda_j. \tag{2.41}$$

Avec ces nouvelles définitions, la qualité vidéo de bout en bout pour l'estimateur LLSE avec une allocation de puissance optimale peut être définie par l'équation (2.42). La démonstration détaillée se trouve en Annexe B.

$$\begin{aligned} \mathrm{PSNR}_{[\text{OPA-LLSE}]} \quad = \quad & c + \mathrm{CSNR} + G_{\mathrm{LLSE}} \\ & -20\log_{10}\left(H_{t2}\right) - 10\log_{10}\left(1 + \frac{(\mathrm{CSNR}_{lin} + 1)\cdot E_{d2}}{{H_{t2}}^2}\right). \end{aligned} \tag{2.42}$$

L'équation (2.42) a une forme semblable à celle de (2.11), excepté le fait que :

- Comparée à cette dernière, tout comme (2.23), elle inclut le terme G_{LLSE} qui reflète les avantages de l'estimateur LLSE. Cependant, contrairement à (2.23), ce nouveau modèle est valable quelle que soit la distribution de la puissance ;
- le cinquième et dernier terme inclut $(\mathrm{CSNR} + 1)$ comme (2.23), à la place de (CSNR) pour (2.11) ;
- la définition de H_{t2} et E_{d2} dépend de ℓ au lieu de K.

Alors que dans (2.23), les coefficients/chunks ne sont jetés qu'en raison de contraintes de bande passante, dans (2.42), afin d'optimiser la qualité reçue, SoftCast+ peut supprimer certains coefficients/chunks même si la bande passante disponible à l'émetteur permet de tous les transmettre. En conséquence, le modèle ci-dessus inclut déjà les cas FB et CB. Pour les applications à bande passante limitée, ℓ représente en fait la valeur minimale entre le nombre de coefficients jetés Nb_1 en raison d'une allocation de puissance optimale et le nombre de coefficients jetés Nb_2 correspondant à la bande passante disponible ($\ell = min(Nb_1, Nb_2)$). Nous notons que le nombre de coefficients jetés pour une allocation optimale de puissance n'est pas fixe et dépend des caractéristiques du canal. Il est mis à jour pour chaque CSNR.

2.4.1 Analyse des performances du modèle SoftCast+ proposé

L'efficacité du modèle proposé est comparée à celle du schéma SoftCast d'origine en considérant l'allocation de puissance optimale et l'estimateur LLSE (i.e., SoftCast+).

Le modèle est évalué via les mêmes configurations de simulation que celles données dans la Section 2.2.3. Les résultats obtenus sont affichés dans la Fig. 2.8.

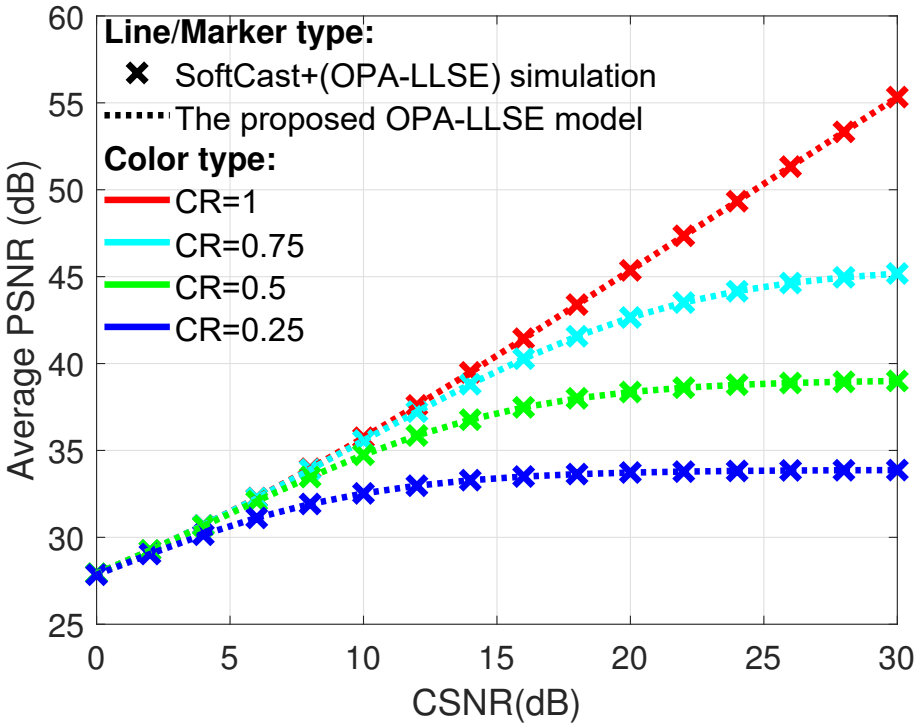


FIGURE 2.8 : Evolution du PSNR moyen obtenu pour le modèle théorique SoftCast+ proposé (ligne en pointillé) et les simulations SoftCast+ (allocation de puissance optimale et estimateur LLSE) : (marqueurs : croix) pour la séquence *Mixed HD720p*. Configuration : taille de GoP = 16 images, 64 chunks/image. Les couleurs rouge, cyan, vert et bleu représentent respectivement les CR = 1, 0.75, 0.5 et 0.25.

Nous observons que :

- Contrairement aux modèles précédents, où les contraintes de bande passante impliquaient directement une perte de qualité, même à des valeurs de CSNR faibles, SoftCast+ donne presque la même qualité reçue pour l'ensemble des bandes passantes disponibles à faible CSNR (c'est-à-dire en dessous d'un CSNR $\leq$ 5dB pour cette séquence vidéo). En effet, avec de telles valeurs de CSNR, le nombre de chunks transmis avec SoftCast+ est généralement très petit (par exemple, seuls $\frac{206}{1024}$ et $\frac{276}{1024}$ chunks par GoP en moyenne, respectivement pour un CR=0.25 et CR=1 pour la séquence *Mixed HD720p* avec un CSNR = 0dB, ce qui signifie que seulement $\sim 20\%$ de la bande passante totale est utilisée).

- Dans tous les cas, notre modèle correspond parfaitement aux simulations sur toute la gamme de CSNR, indépendamment de la bande passante disponible considérée. Ceci est prévisible, étant donné qu'aucune approximation n'est faite dans le processus de dérivation de (2.42).

2.5 Evaluation des performances globales des schémas

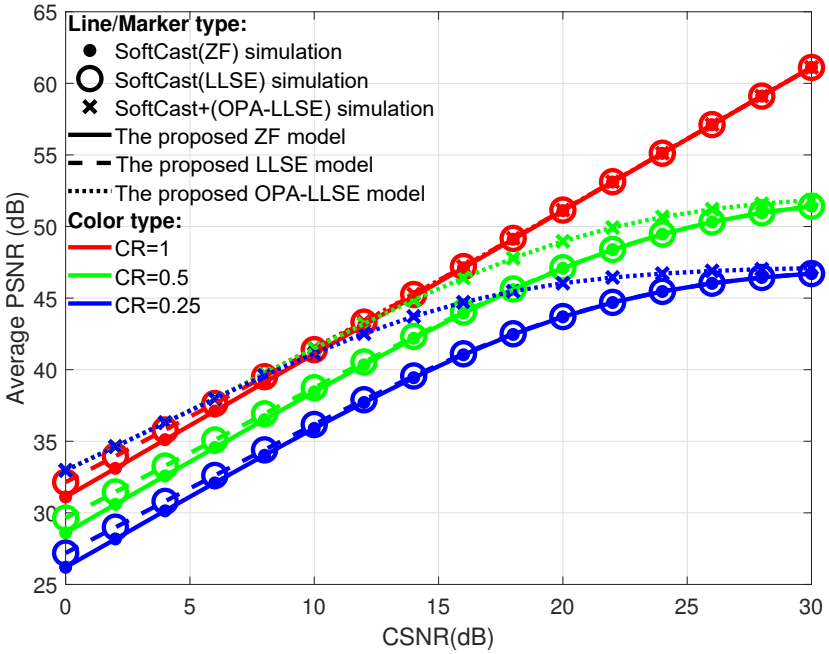
Dans cette section, nous comparons les performances des trois schémas les uns par rapport aux autres grâce à nos modèles et donnons un exemple possible d'utilisation de ces derniers. Particulièrement, nous montrons qu'une taille GoP optimale peut être définie pour le processus de codage en fonction du contenu vidéo.

Nous comparons tout d'abord les résultats obtenus par les modèles proposés ou de manière équivalente, nous évaluons l'apport des différentes variantes de SoftCast les unes par rapport aux autres : SoftCast(ZF), SoftCast(LLSE) et SoftCast+.

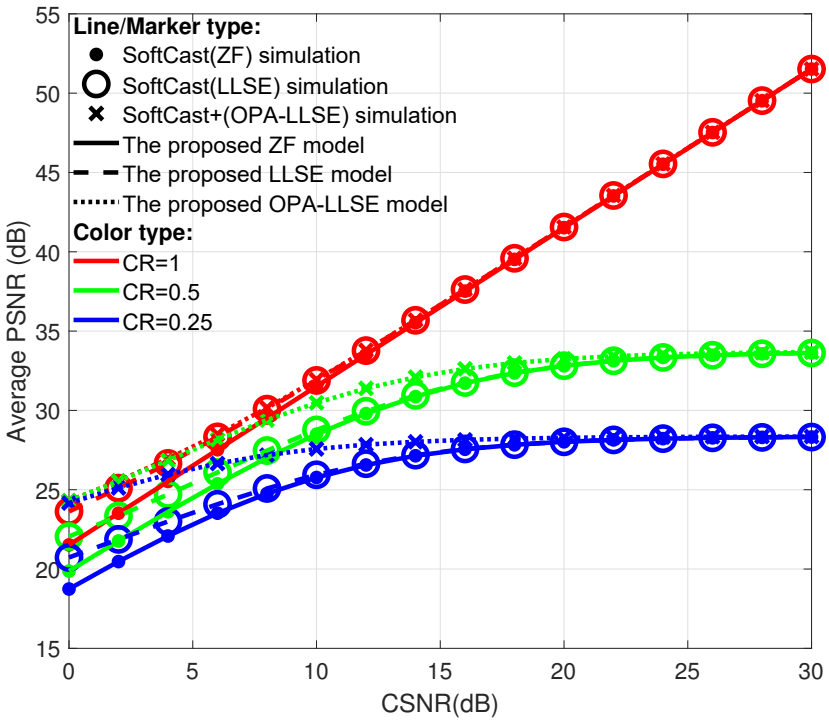
Les paramètres utilisés dans les simulations sont les mêmes que ceux décrits dans la Section 2.2.3. Parmi toutes les séquences vidéo, nous avons choisi d'afficher les résultats des séquences vidéo *Johnny* et *ParkJoy* en raison de leurs disparités spatio-temporelles observées dans la Fig. 2.6. Par souci de clarté, nous affichons uniquement les résultats pour trois largeurs de bande disponibles, à savoir, les cas CR=1, 0.5 et 0.25 respectivement représentées par les couleurs rouge, vert et bleu. Les résultats pour les autres cas de bande passante sont similaires. La taille de GoP retenue est de 16 images, puisqu'il s'agit de la taille de GoP couramment utilisée dans les articles de référence [53, 64].

A partir des résultats obtenus dans la Fig. 2.9, nous observons que :

- Comme indiqué dans la Section 2.2.3, 2.3.1 et 2.4.1, quel que soit le contenu vidéo transmis et la bande passante disponible, les trois modèles coïncident parfaitement sur l'ensemble des simulations ;
- Au-dessous de 10 dB, quel que soit le contenu vidéo transmis, il est bien connu que l'estimateur LLSE (lignes tiretées, marqueurs : grands cercle) donne de meilleurs résultats que l'estimateur ZF (ligne en trait plein, marqueurs : points) [111]. Les résultats sont conformes à cette affirmation. Cependant, l'amélioration en termes de PSNR est limitée et faible comme indiquée avec l'approximation du modèle LLSE dans le Tableau 2.1 ;
- Quelle que soit la configuration (taille de GoP, bande passante disponible, contenu vidéo transmis), pour un même CSNR, SoftCast (ZF) offre le plus faible PSNR, suivi de SoftCast(LLSE) dont les performances sont inférieures à celles de SoftCast+. En effet, SoftCast+ utilise à la fois l'estimateur LLSE et l'allocation de puissance optimale ;



(a) Johnny



(b) ParkJoy

FIGURE 2.9 : Evolution du PSNR moyen obtenu pour les modèles théoriques proposés : ZF(ligne continue), LLSE(ligne tiretée) et OPA-LLSE(ligne pointillée) ; et les simulations SoftCast : SoftCast ZF (points), SoftCast LLSE (cercle) et SoftCast+ (croix) pour les séquences *Johnny* et *ParkJoy*. Configuration : taille de GoP = 16 images, 64 chunks/image. Les couleurs rouge, cyan, vert et bleu représentent respectivement les CR = 1, 0.75, 0.5 et 0.25.

- En comparant les performances de SoftCast (LLSE) (lignes tiretées et cercles) à celles de SoftCast+ (lignes pointillées et croix), nous constatons que SoftCast+ présente une amélioration marginale des performances par rapport à SoftCast (avec estimateur LLSE), comme indiqué dans [13]. Toutefois, cela n'est vrai que si l'on considère que la transmission de tous les coefficients/chunks est possible (aucune restriction de bande passante). Par exemple, si nous considérons qu'un CR=0.25 est utilisé pour la transmission et que nous nous focalisons sur un CSNR=0dB, nous pouvons voir que l'écart entre ces deux versions est d'environ 5.76dB et 3.43dB, respectivement pour les séquences *Johnny* et *ParkJoy*. Cet écart diminue à mesure que le CSNR augmente et devient pratiquement nul après un CSNR≥30dB ou un CSNR≥15dB, respectivement pour les séquences vidéo *Johnny* et *ParkJoy*. Ceci est parfaitement expliqué par les modèles proposés, dans lesquels, pour la séquence *ParkJoy*, la plupart des chunks devraient être transmis, mais ne peuvent pas en raison des contraintes de bande passante. En revanche, pour la séquence vidéo *Johnny*, l'amélioration par rapport au schéma SoftCast classique (LLSE) reste importante même après un CSNR = 15dB, car la plupart des coefficients/chunks sont énergétiquement faibles et peuvent être jetés pour réallouer intelligemment toute la puissance disponible au niveau de l'émetteur.

Une fois de plus, nous vérifions que les trois modèles proposés représentent avec précision l'ensemble des résultats de simulations obtenus pour les schémas basés SoftCast. Sur la base de la distribution de puissance (λ_i) disponible au niveau de l'émetteur après la transformation de décorrélation, il est possible d'évaluer rapidement les performances du schéma sans avoir à effectuer des simulations complètes de bout en bout. Nous montrons également que les performances et les comportements des schémas basés sur SoftCast dépendent fortement du contenu transmis. Dans la suite, nous proposons un exemple d'utilisation des modèles proposés en analysant si une taille GoP optimale peut être sélectionnée en fonction du contenu transmis. Par optimale, nous voulons dire une taille de GoP permettant d'augmenter la qualité reçue dans les mêmes conditions de transmission.

Nous évaluons d'abord les trois schémas : SoftCast (ZF), SoftCast (LLSE) et SoftCast+ en considérant une taille de GoP de 4, 8, 16 et 32 images. Dans un souci de lisibilité, nous choisissons d'afficher uniquement les résultats obtenus pour deux CR sélectionnés : CR=1 (première ligne), et CR=0.25 (deuxième ligne). Lorsque la compression est nécessaire en raison de bande passante limitée, nous nous assurons de conserver le même débit symboles pour toutes les méthodes. Par exemple, avec un CR=0.25, nous conservons l'équivalent de 2 et 4 images pour les tailles de GoP égales à 8 et 16, respectivement. Nous avons vérifié que les résultats pour d'autres bandes passantes de canal avaient des comportements similaires. Dans cette thèse, la taille maximale de GoP sélectionnée est définie sur 32 images. En effet, la complexité augmente en fonction de $O(Klog(K))$ avec K le nombre d'images dans un GoP [53, 23]. Choisir une taille GoP supérieure à 32 images implique des capacités matérielles

élevées ainsi qu'une augmentation de la latence et du temps nécessaire au décodage car le récepteur doit attendre toutes les images avant de traiter la DCT temporelle inverse. Ces deux contraintes peuvent ne pas être compatibles avec toutes les applications.

Dans cette section, les simulations de bout en bout sont toujours effectuées afin de prouver l'efficacité des modèles proposés.

La première séquence vidéo sélectionnée est *Johnny* car elle contient des informations spatio-temporelles relativement basses. Indépendamment du schéma considéré et de la bande passante disponible, les résultats présentés dans la Fig. 2.10 montrent que l'augmentation de la taille de GoP entraîne une meilleure qualité de réception pour toutes les gammes de CSNR considérées. Le gain entre une taille GoP de 4 et 32 images est d'environ 6 dB, quel que soit le schéma SoftCast considéré. Cet énorme gain est dû à la meilleure utilisation de la DCT temporelle. En effet, en raison de la forte corrélation temporelle entre les images (mouvements lents), l'utilisation d'un GoP plus grand permet de mieux compacter les informations et donc de réduire *l'activité des données* H_t. Cependant, comme expliqué précédemment, lors de la prise en compte des contraintes de bande passante, l'effet de *levelling-off* apparaît. Par conséquent, le gain diminue et l'augmentation de la taille du GoP pour des valeurs de CSNR élevées ($\geq$25dB) n'apporte pas une grande amélioration, car des coefficients/chunks ont été jetés à l'émetteur et cette partie d'information perdue ne peut pas être reconstruite à la réception.

Nous réalisons les mêmes simulations en considérant la séquence vidéo *ParkJoy* à contenu spatio-temporel élevé. Les résultats sont donnés dans la Fig. 2.11. Contrairement à la séquence *Johnny*, l'amélioration est limitée et l'augmentation de la taille GoP de 4 à 32 images n'apporte qu'un gain de 1.2 dB. Quel que soit le schéma étudié, l'amélioration n'est que d'environ 0.2 dB pour le passage d'une taille de GoP = 16 à 32 images. Une telle amélioration est insignifiante, car le comité MPEG estime qu'une différence n'est visuellement perceptible qu'au-delà de 0.5 dB [83]. Par conséquent, nous recommandons d'utiliser une taille GoP intermédiaire (8$\sim$16 images) pour ce type de contenu. Cela est d'autant plus vrai que le gain entre ces deux tailles de GoP diminue rapidement et devient nul ou légèrement négatif lorsque l'on considère les applications à bande passante réduite dans des valeurs de CSNR $\geq$15dB.

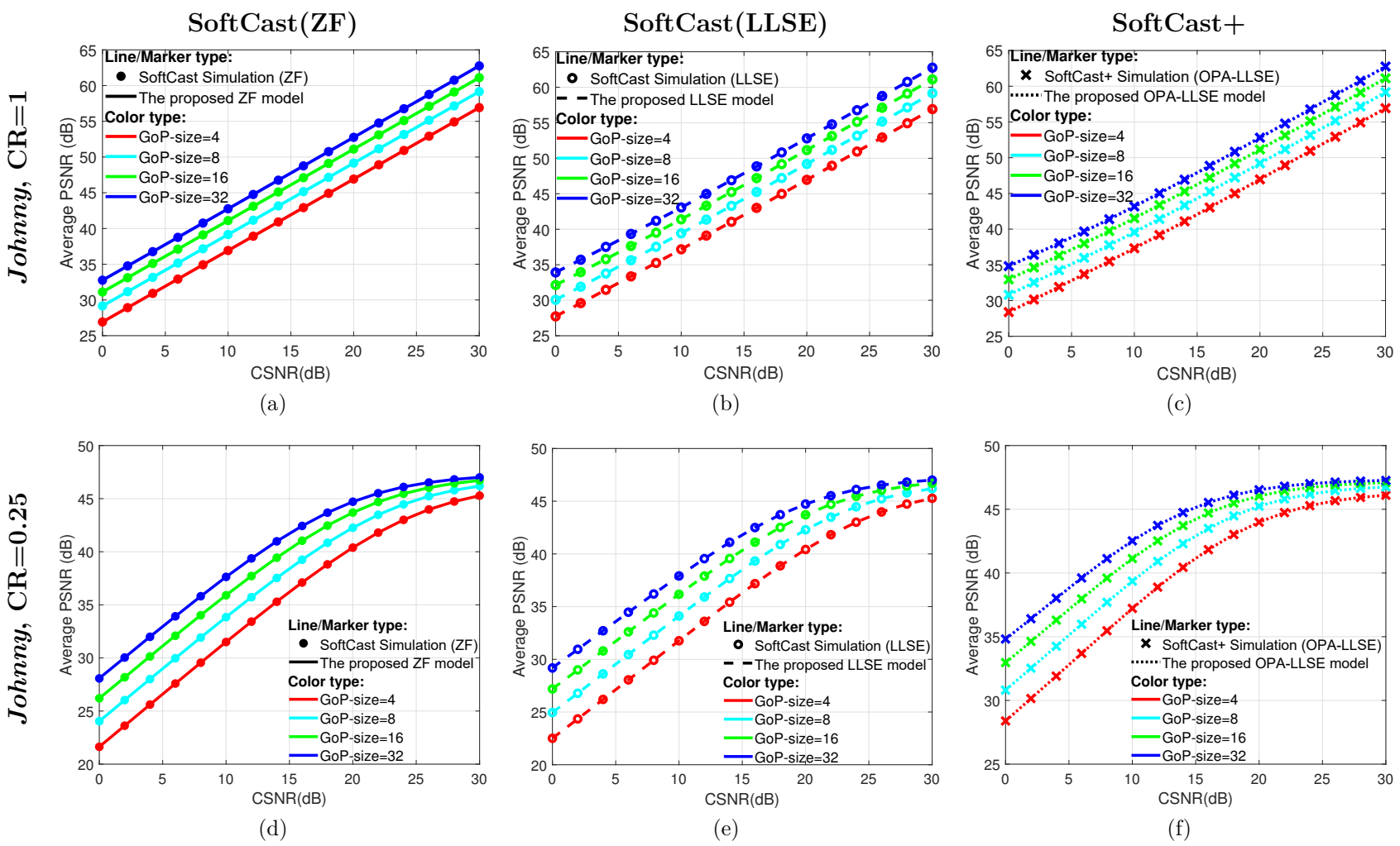


FIGURE 2.10 : Evolution du PSNR moyen obtenu pour la séquence *Johnny*, avec les modèles théoriques et les schémas basé SoftCast. (a),(b),(c) : CR = 1. (d),(e),(f) : CR=0.25. (a),(d) : SoftCast ZF et modèle théorique associé. (b),(e) : SoftCast LLSE et modèle théorique associé. (c),(f) : SoftCast+ et modèle théorique associé.

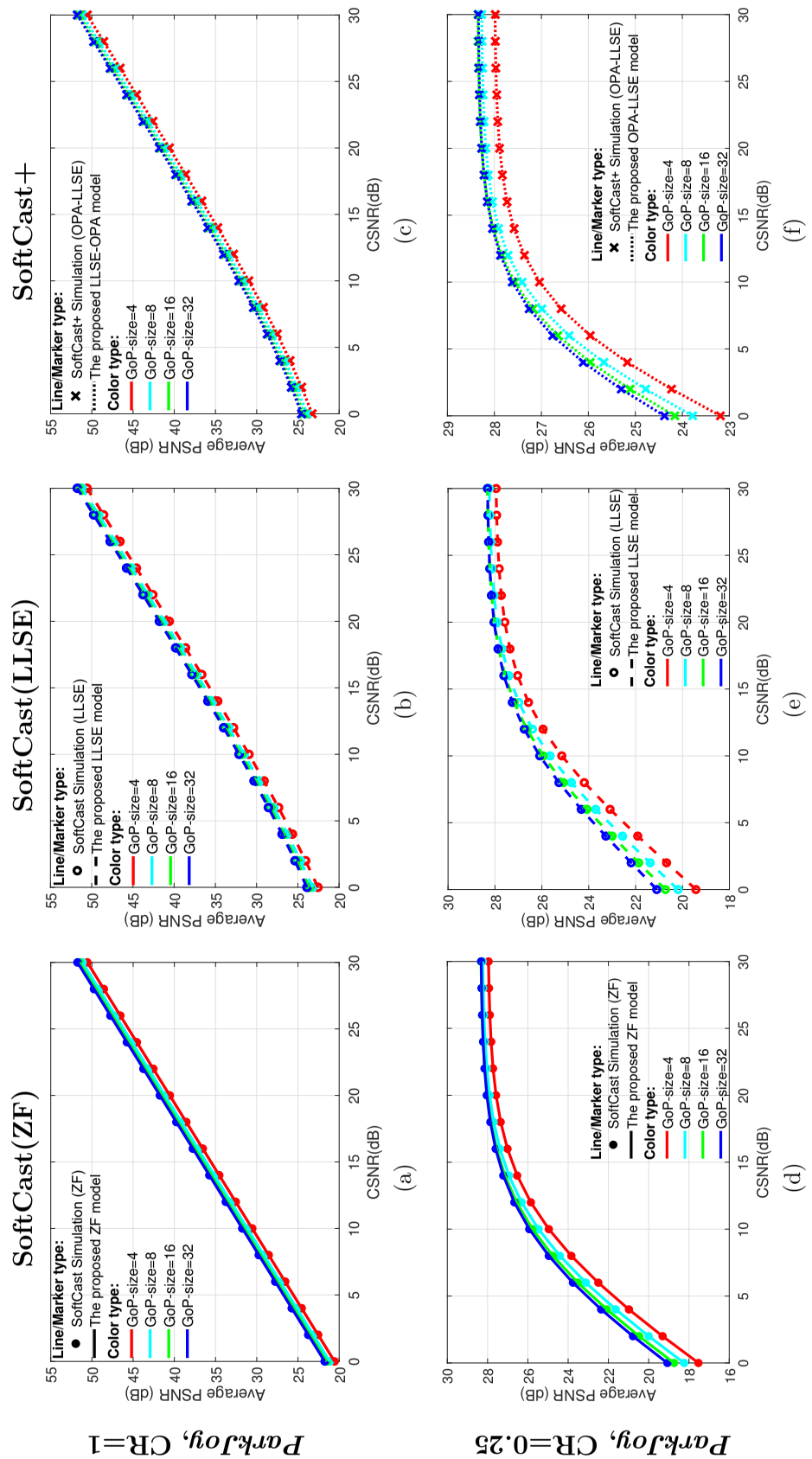


FIGURE 2.11 : Evolution du PSNR moyen obtenu pour la séquence *ParkJoy*, avec les modèles théoriques et les schémas basé SoftCast. (a),(b),(c) : CR = 1. (d),(e),(f) : CR=0.25. (a),(d) : SoftCast ZF et modèle théorique associé. (b),(e) : SoftCast LLSE et modèle théorique associé. (c),(f) : SoftCast+ et modèle théorique associé.

Nous avons vérifié que les deux déclarations ci-dessus sont en moyenne valides pour tous les contenus vidéos de la Fig. 2.6. Cependant, il n'est pas facile de donner une taille de GoP optimale en ne tenant compte que des caractéristiques du contenu vidéo lui-même, car plusieurs autres paramètres, tels que la qualité du canal ou la bande passante disponible, ont une incidence sur la qualité reçue. Ci-après, nous donnons des tendances générales mais recommandons pour une application spécifique l'utilisation des modèles proposés pour trouver rapidement la taille GoP optimale en fonction de la largeur de bande du canal disponible et, si elle est connue à l'émetteur, de la qualité du canal.

Si le retard induit par le décodage n'est pas un critère important dans l'application visée, nous suggérons d'utiliser une taille de GoP plus grande (par exemple 32 images) pour les séquences à faibles contenus spatiotemporels tels que *Johnny* ou encore *Akiyo*, étant donné que pour un même CSNR, cette taille permet d'obtenir de meilleurs gains en termes de qualité reçue.

D'un autre côté, utiliser des tailles de GoP de taille moyenne à petite (par exemple, 8∼16 images) pour un contenu spatio-temporel élevé (comme par exemple pour des événements sportifs, etc.) tel que celui des séquences vidéos *ParkJoy* ou *Stefan* est suffisant car le gain entre ces petites tailles et une taille de GoP plus grande est insignifiant, voire négatif. Ceci est particulièrement vrai pour la séquence vidéo *Husky* (SI élevé, TI très élevé), où l'augmentation de la taille du GoP de 8 à 32 images n'apporte qu'une amélioration moyenne inférieure à 0.4dB, ce qui est imperceptible [83].

Indépendamment du contenu vidéo, nous notons que la taille GoP de 4 images n'est jamais privilégiée car elle ne tire pas suffisamment parti de la corrélation temporelle entre les images.

Nous avons présenté dans ce chapitre trois modèles pouvant être utilisés pour prédire les performances de bout en bout des schémas basés SoftCast, prenant en compte notamment :

- Les applications à bande passante limitée ;
- L'utilisation de l'estimateur LLSE du côté récepteur ;
- L'utilisation de l'allocation de puissance optimale.

En fonction des applications ciblées, nous recommandons d'utiliser (2.8) dans des contextes de diffusion (broadcast), où un flux de données codées est envoyé pour tous les destinataires, ou (2.42) dans un contexte de monodiffusion, où un retour sur la qualité de canal peut être transmis à l'émetteur permettant de profiter des avantages de l'allocation optimale de puissance. En effet, (2.19) n'apporte qu'une légère amélioration par rapport à (2.8) et ceci uniquement pour des valeurs de CSNR faibles ($\leq$ 10dB). Cette amélioration est quantifiée par le modèle approximé (2.21) dans le Tableau 2.1.

2.6 Conclusion

Dans ce chapitre, nous avons fourni une évaluation théorique complète des performances de bout en bout des schémas basés SoftCast. Cette évaluation théorique comprend :

1. Les applications à largeur de bande limitée ;
2. L'estimateur LLSE ;
3. L'allocation de puissance optimale.

Trois modèles théoriques basés sur la métrique PSNR ont été ainsi formulés et les résultats des simulations ont montré qu'ils représentent avec précision les performances complètes de bout en bout.

Contrairement au modèle de Xiong *el al.*, les modèles proposés peuvent aider à optimiser les paramètres d'un schéma basé SoftCast incluant les applications à bande passante limitée. De plus, ils aident également à caractériser clairement l'origine de l'effet de saturation de la qualité (*levelling-off* [64] qui apparaît lorsque des coefficients transformés sont jetés), ainsi qu'à quantifier l'amélioration apportée par l'estimateur LLSE.

Étant donné que ces modèles théoriques ne dépendent que de la valeur du CSNR ainsi que de la distribution énergétique des coefficients/chunks (obtenue après 3D-DCT), ils peuvent être utilisés pour évaluer rapidement des schémas de codage vidéo linéaire sans nécessiter de simulations approfondies de bout en bout. En outre, puisque ces derniers prédisent la qualité pouvant être obtenue à la réception sans avoir besoin d'effectuer une 3D-DCT inverse, il en résulte une complexité deux fois moins importante par rapport au schéma SoftCast de bout en bout. À l'aide d'un exemple simple, nous avons montré qu'en utilisant uniquement la distribution énergétique des coefficients, des tendances générales concernant une taille de GoP optimale peuvent être obtenues en fonction du contenu vidéo transmis. Cette étude a été vérifiée au moyen de simulations complètes de bout en bout, montrant l'efficacité des modèles proposés.

Chapitre 3

Etude des artefacts de codage et de transmission des schémas de codage vidéo linéaire

Sommaire

3.1 Introduction

Bien que les systèmes de codage vidéo linéaires aient été largement étudiés suite à la parution de la version originelle de SoftCast [54], beaucoup de pistes n'ont pas encore été explorées notamment d'un point de vue perceptuel. La compression étant réalisée de manière différente d'un schéma de codage classique, les artefacts visuels sont eux aussi différents et méritent d'être étudiés afin d'optimiser les schémas CVL d'un point de vue perceptuel. Par exemple, de par l'utilisation d'une transformée DCT pleine image et non bloc par bloc comme classiquement effectuée (e.g. bloc de 8×8 pixels), le schéma SoftCast n'introduit pas d'effet de bloc lors de la reconstruction. Un test subjectif lié à FoveaCast (voir Section 1.3.1.5) a récemment été publié par Shen *et al.* [91], cependant ces travaux ne concernent pas spécifiquement l'étude des défauts propres aux schémas basés SoftCast. A notre connaissance, il n'existe donc pas aujourd'hui d'articles présentant les artefacts des CVL basés SoftCast ainsi qu'une évaluation subjective approfondie de la qualité vidéo reçue via de tels schémas.

Nous proposons donc dans ce chapitre :

1. Une introduction de différents artefacts produits par le schéma SoftCast parmi lesquels : l'effet de neige, l'effet fantôme, l'effet de flou, les fluctuations temporelles de qualité ;
2. Une étude subjective concernant la qualité vidéo reçue via SoftCast ;
3. Une étude des performances des métriques actuelles existantes (PSNR, SSIM, VMAF, etc.) dans un contexte de réception de contenus vidéo transmis via SoftCast.
4. En outre, nous mettons à disposition de la communauté scientifique, la première base de données subjectives et annotées concernant les distorsions liées au codeur SoftCast.

3.2 Présentation des artefacts de codage et de transmission

A la manière des articles proposées par Yuen *et al.* [120] et Lin *et al.* [65], introduisant respectivement les distorsions introduites par les codeurs hybrides DCT/DPCM à compensation de mouvement (utilisés par exemple dans H.261, H.263, MPEG-1 et MPEG-2) et les artefacts de compression du standard HEVC, nous proposons ici de présenter et d'analyser les artefacts apparaissant dans le cadre des codeurs vidéo linéaires basés SoftCast. A notre connaissance, il s'agit de la première étude de ce type concernant ce type de codeurs. Dans cette partie, les distorsions dues à la transmission et/ou à la compression sont présentées.

3.2.1 L'effet de neige

Parmi les différents artefacts des codeurs vidéo linéaires, l'effet de neige est sans doute le plus caractéristique. A la différence des standards de transmission classiques où le canal de transmission peut entraîner des gels d'images, des parties manquantes ou blocs très altérés dans les images dues aux erreurs de décodage des paquets reçus, dans les codeurs vidéo linéaires, le canal agit directement sur les coefficients transmis.

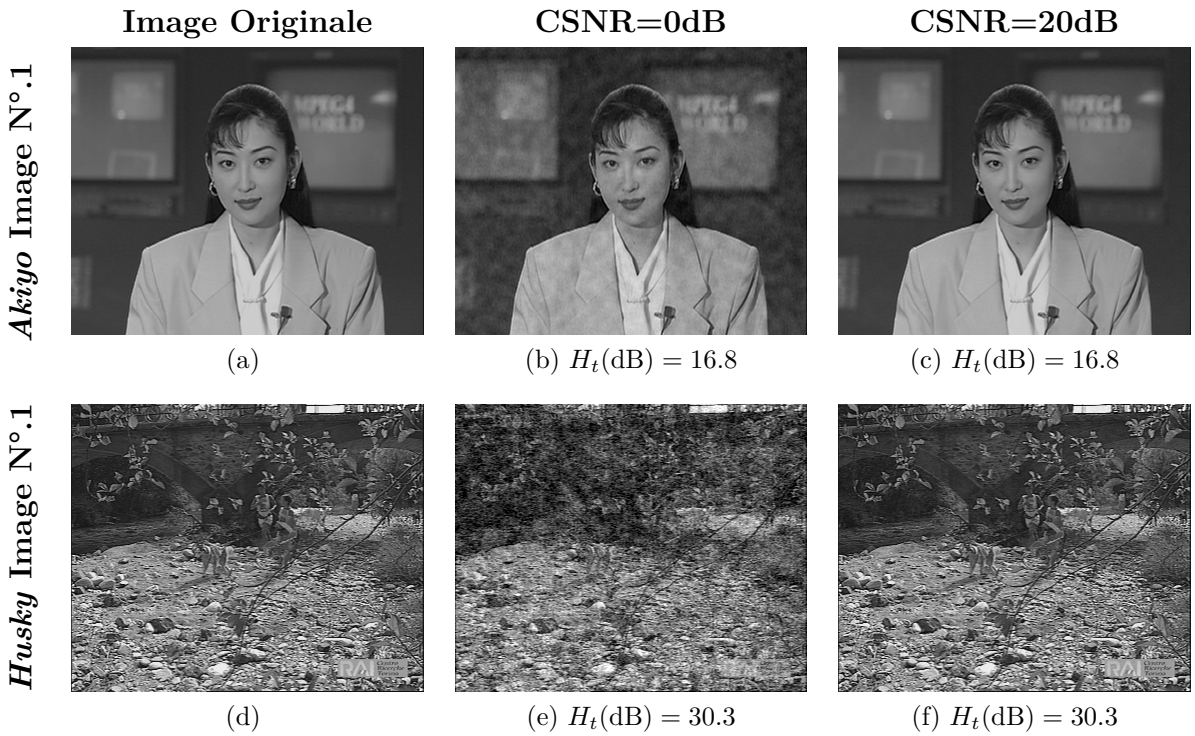


FIGURE 3.1 : Illustration de l'effet de neige, aucune compression appliquée (CR = 1). (a),(b),(c), : première image de *Akiyo*. (d),(e),(f) : première image de *Husky*. (a),(d) : Image d'origine. (b),(e) : CSNR=0dB. (d),(f) : CSNR=20dB.

Dans le domaine pixel, après décodage, cet effet de neige se caractérise par une "trame" venant se superposer à l'image décodée. Plus le canal de transmission est mauvais (bas CSNR), plus cette trame est visible. Pour de très bons canaux (CSNR>20dB), elle est par contre presque invisible. Nous notons que cet effet est caractéristique des codeurs vidéo linéaires et se retrouve dans beaucoup de variantes actuellement proposées et introduites dans le Chapitre 1 [21, 117, 128, 129].

Au travers de notre étude théorique (Chapitre 2) basée sur le PSNR, nous montrons que l'effet de neige est lié à la fois à la qualité du canal mais également à l'activité spatiotemporelle H_t du contenu transmis. Pour illustrer cela, nous montrons tout d'abord un exemple de la qualité reconstruite avec SoftCast pour deux contenus vidéo CIF *Akiyo* et *Husky* et

considérant deux qualités de canal : CSNR = 0dB et 20dB. Aucune compression n'est appliquée (CR=1), les tailles de chunks et des GoPs sont choisies classiquement comme étant respectivement égales à 36×44 coefficients et 16 images [54]. Les résultats sont présentés en Fig. 3.1. Comme nous pouvons le voir, l'effet de neige est beaucoup plus marqué à même CSNR pour la séquence *Husky* que pour la séquence *Akiyo* étant donné la valeur élevée de son activité H_t.

Xiong *et al.* [110] ont montré que cette trame est représentée par de fortes composantes basses fréquences. Ceci est dû à l'allocation de puissance actuelle, basée sur la minimisation de la MSE (critère non visuel), qui vient relativement (étant donné la contrainte de puissance totale) diminuer la puissance allouée aux coefficients DCT basses fréquences par rapport à celle des coefficients DCT hautes fréquences.

3.2.2 Les fluctuations temporelles de qualité (effet de cloche)

Nous avons relevé dans plusieurs travaux de la littérature [54, 129] y compris les nôtres, un effet de fluctuation temporelle de la qualité qui se caractérise par une chute du PSNR aux frontières des GoPs encodés. Comme nous l'observons sur la Fig. 3.2 (courbes vertes, bleues et noires), cet effet est visible à la fois sur le PSNR et sur le SSIM et donne des courbes en "cloches" pour chaque GoP encodé. Etant donné qu'il est visuellement difficile à décrire, nous invitons les lecteurs à visualiser les fichiers vidéo complets présents dans le lien suivant : https://drive.google.com/open?id=1iCiqrVMClSvwLRSpwaRXx_PCO6_e4UGL. De plus, comme ces fluctuations ont lieu sur la durée d'un GoP, nous donnons des exemples de vidéo reconstruites pour trois tailles de GoP communément utilisées à savoir $8, 16$ et 32 images. Nous donnons en outre, deux CSNR différents (CSNR=0 et 20dB). Tout comme précédemment, nous laissons le CR=1 (aucune compression de la vidéo) et avons vérifié que cet artefact est également présent pour d'autres niveaux de compression.

Nous invitons également les lecteurs à consulter les fichiers vidéo proposés par Zheng *et al.* [128] (dans un contexte de transmission avec bruit impulsif) inti BTtulées *Y_BQSquareLVC_NIC_PSNR30dB_416×240* et *Y_BQSquareLVC_OSP_IC_PSNR37dB_416×240* (https://drive.google.com/drive/folders/13LB5nR3nY79bF3CEMUl41HY4Bc_ekhBF). La configuration choisie par les auteurs est la suivante : taille de GoP = 8 images, CSNR=15dB, probabilité de bruit impulsif = 1%, variance du bruit impulsif = 100. Les acronymes NIC et OSP_IC réfèrent respectivement à No Impulse noise Correction et Optimal Subchannel Provisioning with Impulse noise Correction.

Comme nous pouvons l'observer, cet effet est particulièrement visible sur des surfaces lisses/aplat (e.g. ciel de la séquence *Snow Mountain*). En revanche, celui-ci n'est pas/peu visible sur des séquences présentant des activités spatiotemporelles élevées (e.g. *Husky*), car l'effet de neige est beaucoup plus marqué. La variation temporelle de qualité est ainsi noyée

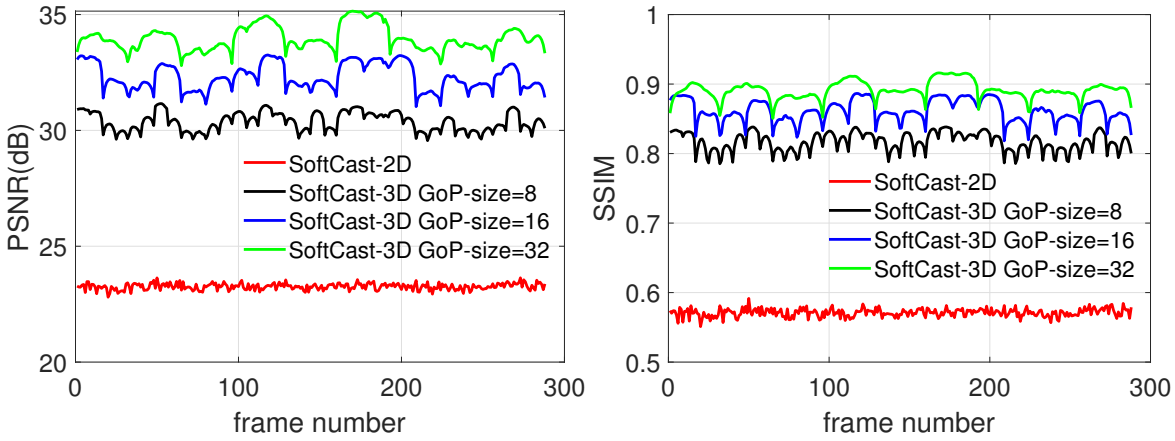


FIGURE 3.2 : Illustration des variations temporelles de qualité (effet de cloche). Couleurs : Rouge = SoftCast-2D, Bleu, Noir et Vert = SoftCast-3D respectivement avec taille de GoP=8,16 et 32 images. Configuration : Séquence *Akiyo*, CR=1, CSNR=0dB.

dans le bruit et n'est donc pas/peu perçue. Tout comme l'effet de neige, nous notons que l'effet de fluctuation de la qualité devient de moins en moins visible à mesure que la qualité du canal de transmission s'améliore.

Cet effet de cloche provient en réalité d'une combinaison de l'utilisation d'une DCT temporelle couplée au bruit présent dans le canal de transmission. Nous vérifions cela en encodant le contenu vidéo via l'algorithme SoftCast en désactivant la DCT-1D temporelle (i.e., seule la DCT-2D est utilisée, les paramètres des autres blocs de SoftCast ainsi que du canal de transmission restent inchangés). Comme nous pouvons le voir dans la vidéo *Akiyo_cif_LLSE_GoP_size 1_SNR0* présente dans le premier lien et sur la Fig. 3.2 (courbes rouges), le fait de ne pas recourir à la version 3D de la DCT permet d'éviter ce phénomène d'effet de cloche. Toutefois, dans ce cas, comme la corrélation temporelle n'a pas été exploitée, il en résulte une grande perte de qualité (objective ou subjective).

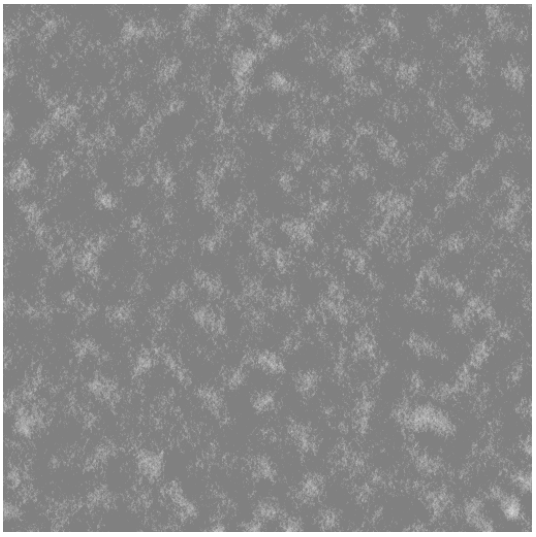
Nous avons vérifié que cet effet ne provenait pas spécifiquement de l'utilisation de l'estimateur LLSE puisque des courbes en cloches similaires sont également obtenues lors d'un décodage effectué avec un estimateur ZF.

3.2.3 L'effet de flou

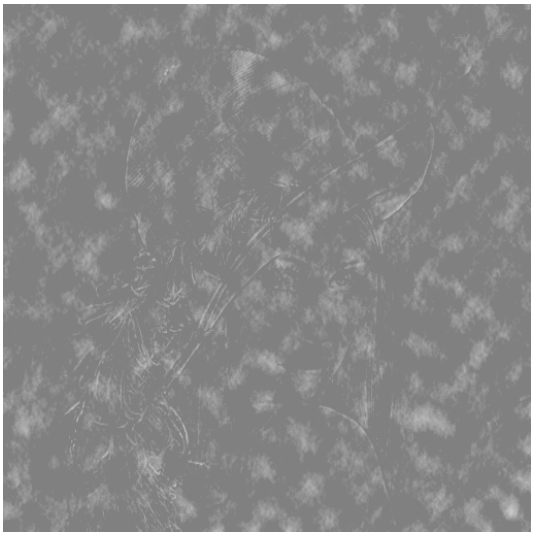
Après réception d'un contenu vidéo transmis via SoftCast, le décodeur utilise un estimateur LLSE afin d'obtenir la meilleure estimation au sens de la maximisation du PSNR reconstruit. Nous montrons ici, que bien que le PSNR obtenu est meilleur, le recours à un estimateur LLSE vient modifier la netteté des contours de la vidéo ou d'une image transmise. Cette modification des contours de l'image introduit un effet de flou pouvant être gênant pour l'utilisateur comme illustré dans la Fig. 3.3. Ce dernier fait l'objet d'un test subjectif présenté en Section 3.3.3.

(a) SoftCast(ZF), PSNR=23.24dB

(b) SoftCast(LLSE), PSNR=24.63dB



(c) SoftCast(ZF), PSNR=23.24dB



(d) SoftCast(LLSE), PSNR=24.63dB

FIGURE 3.3 : Illustration du flou induit par l'estimateur LLSE. Configuration : image transmise *Lena* avec CR=1 et CSNR=0dB. (a) SoftCast avec estimateur ZF, (b) SoftCast avec estimateur LLSE. (c) Image d'erreur ZF. (d) Image d'erreur LLSE.

Pour illustrer mathématiquement le phénomène de flou engendré, nous renvoyons tout d'abord les lecteurs aux équations (1.16), (1.17) et (1.18) vues dans le Chapitre 1. Pour faciliter la compréhension, nous effectuons l'analyse sur une transmission où la DCT temporelle est désactivée, comme nous l'avions vu, l'émetteur effectue tout d'abord une mise à l'échelle des coefficients avant la transmission :

$$y_i = g_i \cdot x_i.$$

Après transmission, le signal reçu est contaminé par un bruit blanc additif gaussien (AWGN) :

$$\begin{aligned} \hat{y}_i &= y_i + n_i, \\ &= g_i \cdot x_i + n_i. \end{aligned}$$

Dans le cas d'un décodage avec un estimateur ZF, le signal estimé est donc :

$$\begin{aligned} \hat{x}_i(\mathrm{ZF}) &= \frac{\hat{y}_i}{g_i}, \\ &= x_i + \frac{n_i}{g_i}. \end{aligned}$$

Après DCT inverse, l'image reconstruite est donc :

$$\begin{aligned} \mathrm{I}_{rec} &= \mathrm{DCT}^{-1}\{\hat{x}_i(\mathrm{ZF})\}, \\ &= \mathrm{DCT}^{-1}\{x_i\} + \mathrm{DCT}^{-1}\{b_i\}, \\ &= \mathrm{I}_{ori} + B_i. \end{aligned} \tag{3.1}$$

où I_{rec} représente l'image reconstruite, b_i, le bruit équivalent après l'opération inverse de mise à l'échelle et B_i le bruit équivalent après DCT inverse. On remarque que l'estimateur (3.1) est statistiquement non biaisé.

Dans le cas d'un décodage avec un estimateur LLSE, le signal estimé est (voir Section 2.3) :

$$\begin{aligned} \hat{x}_i(\mathrm{LLSE}) &= \alpha_i \cdot \hat{y}_i, \\ &= \frac{g_i \lambda_i}{g_i^2 \lambda_i + \sigma_n^2} \cdot \hat{y}_i, \\ &= \frac{1}{1 + \frac{\sigma_n^2}{g_i^2 \lambda_i}} \cdot \frac{\hat{y}_i}{g_i}, \\ &= \frac{1}{1 + \frac{\sigma_n^2}{g_i^2 \lambda_i}} \cdot \hat{x}_i(\mathrm{ZF}). \end{aligned} \tag{3.2}$$

On remarque dans l'équation (3.2) que les coefficients DCT estimés par l'estimateur LLSE sont toujours des versions atténuées de ceux estimés par l'estimateur non biaisé ZF.

En rappelant que $g_i^2 \lambda_i = P_i$, nous pouvons voir que plus cette puissance P_i des coefficients DCT est faible plus cette atténuation est importante engendrant ainsi un biais LLSE d'autant plus grand. Il est bien connu que les hautes fréquences représentent les contours d'une image, et que pour des images réelles, ces hautes fréquences sont généralement peu énergétiques. Par conséquent, bien que l'estimateur LLSE permet de réduire la MSE (et donc d'augmenter le PSNR), il agit comme un "filtre" venant atténuer plus fortement les contours des images.

3.2.4 L'effet fantôme

Les effets précédemment mentionnés, sont dus aux effets du canal ou à l'utilisation de l'estimateur LLSE, et représentent donc des artefacts de transmission. Dans cette dernière partie, nous présentons un effet dû à la compression du flux SoftCast effectuée avant transmission. En effet, dans SoftCast, une DCT temporelle est utilisée pour prendre en considération la corrélation existante entre des images successives. Cependant, en présence de changements de plans et lorsque la bande passante est limitée, nous avons constaté que les performances de SoftCast déclinent, conduisant à un artefact gênant : un effet fantôme entre les deux plans. Cela est encore plus vrai pour la transmission de publicité, de contenu de film / bande-annonce et d'événements sportifs où de nombreux cuts apparaissent.

Comme illustré dans la Fig. 3.4, l'effet fantôme est caractérisé par l'apparition d'une superposition des contours (hautes fréquences) entre les images précédant et suivant le changement de scène.

Un phénomène similaire appelé effet *cross-fade* ou *transparency effect* a été observé dans [25, 24] dans un contexte de compression via une 3D-DCT par blocs. Bien que certaines solutions aient été proposées pour réduire le *cross-fade*, elles sont toujours liées à un compromis entre la réduction de cet effet et l'augmentation du débit binaire. Les auteurs de [75] ont également remarqué l'effet de transparence dans un schéma de codage en accordéon basé sur JPEG2000 (ACC-JPEG2000). Un module de détection de changement de scène basé sur une comparaison locale a été proposé pour éviter cet effet de transparence. Cependant, aucun détail n'a été donné concernant ce module. De plus, dans tous ces travaux, aucune clarification théorique n'a été introduite. Dans cette section, nous analysons l'origine de l'effet fantôme pouvant apparaître dans un schéma SoftCast et montrons comment l'annuler.

Pour illustrer l'origine de l'effet fantôme, les propriétés de linéarité et de séparabilité de la DCT sont utilisées, comme indiqué dans la Fig. 3.5. Sans perte de généralité et pour faciliter la compréhension, nous prenons comme exemple un cas avec 4 composantes représentant une DCT temporelle sur 4 images. Tout le développement ci-après concerne les valeurs sur une même coordonnée spatiale. Les indices i et j seront ci-après omis pour alléger la notation. Notons d'abord $x = [y_1, y_2, z_3, z_4]$ un vecteur représentant aux mêmes coordonnées spatiales (i, j) et le long de l'axe temporel, soit des pixels soit des coefficients issus de la DCT-2D

(a) Image originale N°.89

(b) Image originale N°.90

(c) Image reconstruite N°.89

(d) Image reconstruite N°.90

FIGURE 3.4 : Exemple d'illustration de l'effet fantôme, CR = 0.25, taille du GoP = 8. Première ligne : image originale N°.89-90 de la séquence *Tennis*. Deuxième ligne : image reconstruite après compression (pas de transmission).

spatiale. Les composantes $[y_1, y_2]$ et $[z_3, z_4]$ représentent respectivement deux séquences vidéo différentes adjacentes. Posons :

$$x \triangleq y + z \triangleq [y_1, y_2, 0, 0] + [0, 0, z_3, z_4] \tag{3.3}$$

Désignons alors par X, Y, Z, les vecteurs résultants après DCT-1D le long de l'axe temporel. La linéarité conduit alors à :

$$X \triangleq \mathrm{DCT}(x) = \mathrm{DCT}(y + z) = \mathrm{DCT}(y) + \mathrm{DCT}(z) \triangleq Y + Z \tag{3.4}$$

Après le processus de compression illustré en rouge (par exemple, CR = 0.5), les composantes supprimées sont remplacées par des valeurs nulles. Nous obtenons alors les nouveaux vecteurs correspondant $\tilde{X}, \tilde{Y}$ et $\tilde{Z}$. Par l'application de la DCT-1D inverse, nous obtenons $\tilde{x} = \text{DCT}^{-1}(\tilde{X}) = \text{DCT}^{-1}\left\{\tilde{Y} + \tilde{Z}\right\}$. De la même manière, en considérant chaque séquence individuellement, nous avons $\tilde{y} = \text{DCT}^{-1}\left\{\tilde{Y}\right\}$ et $\tilde{z} = \text{DCT}^{-1}\left\{\tilde{Z}\right\}$. On a bien entendu $\tilde{x} = \tilde{y} + \tilde{z}$ étant donné que $\tilde{X} = \tilde{Y} + \tilde{Z}$. Comme démontré, chaque pixel ou coefficient DCT-2D reconstruit est en réalité une addition des composantes perturbées de $\tilde{y}$ et $\tilde{z}$. L'information restante contenues dans chaque séquence après compression a été respectivement répartie sur les quatre composantes après DCT-1D inverse. En généralisant cette explication à l'image complète nous obtenons l'effet fantôme.

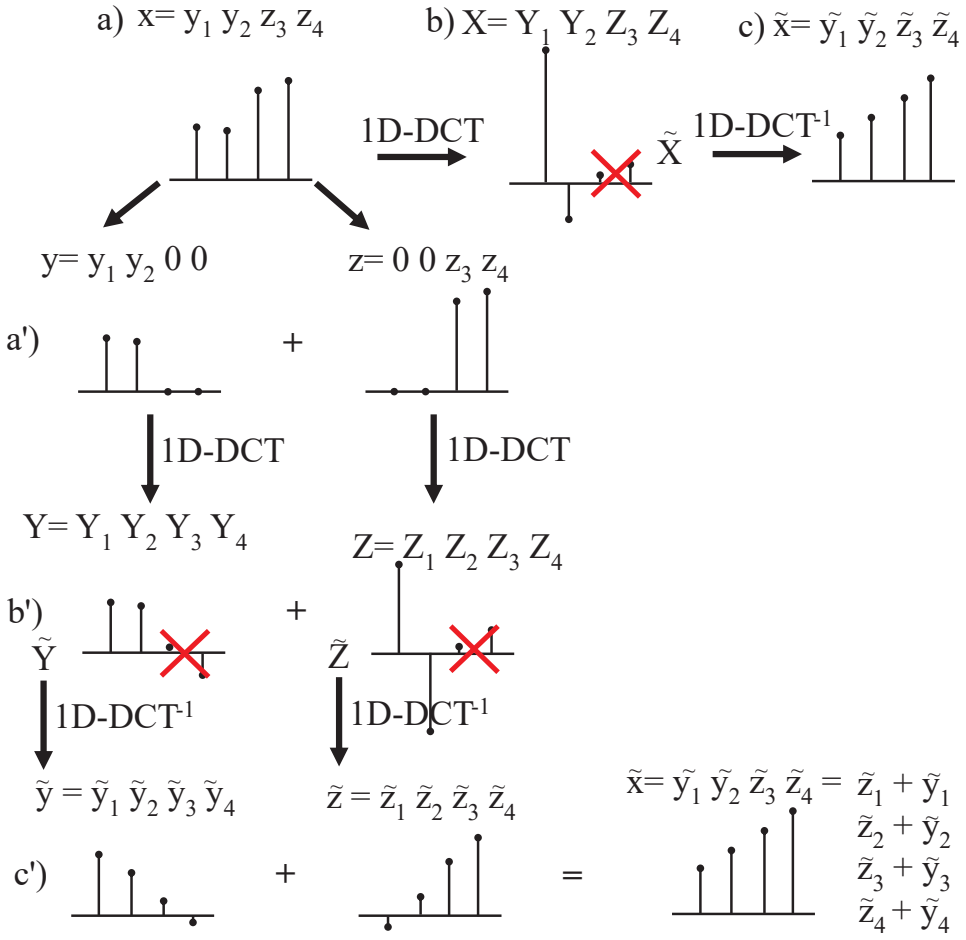


FIGURE 3.5 : Illustration de l'effet fantôme dans le cas unidimensionnel en considérant 4 images. a) et a') Vecteur de pixels ou de coefficients DCT-2D. b) et b') Coefficients résultants après DCT-1D temporelle ; processus de compression en rouge. c) et c') Coefficients reconstitués après DCT-1D inverse. Les étapes désignées par ') utilisent la propriété de linéarité de la DCT.

Comme expliqué ci-dessus, en raison de la compression appliquée, le processus de reconstruction est perturbé et les composantes d'origine de x, y, z ne peuvent pas être complètement récupérées. Ceci est encore plus vrai lorsque la différence entre les composantes est grande et que le taux de compression est petit. En effet, dans un tel cas, le processus de décorrélation ne peut pas entièrement concentrer l'énergie sur une composante, ce qui entraîne une perte d'énergie considérable après la compression. Cependant, si aucune compression n'est appliquée, les composantes initiales peuvent être entièrement récupérées après transformation inverse, puisque la DCT est réversible. En outre, si les composants proviennent d'un même plan (d'une même séquence), la corrélation est élevée sur l'axe temporel. Même avec un taux de compression appliqué élevé, par exemple CR = 0.25 (où 75 % des coefficients sont jetés), la composante continue qui transporte la plus grande partie de l'énergie est conservée, entraînant juste une légère perte d'énergie.

Pour illustrer et décrire plus en détail le processus de décorrélation temporelle du schéma SoftCast, nous introduisons le domaine pseudo-pixel. Comme nous pouvons le voir sur la Fig. 3.6, le GoP est d'abord transformé par une DCT-2D spatiale suivie d'une DCT-1D temporelle. Suivant les étapes classiques de SoftCast, le processus de compression est appliqué et les coefficients jetés sont remplacés par des valeurs nulles. En raison des propriétés de séparabilité, de linéarité et d'invariance par GoP de la DCT-3D, nous utilisons d'abord un processus de DCT-2D inverse suivi d'une DCT-1D inverse temporelle sur les plans contenant les coefficients DCT-3D. Le domaine pseudo-pixel est situé juste après le processus de DCT-2D inverse et avant le processus inverse de DCT-1D temporelle.

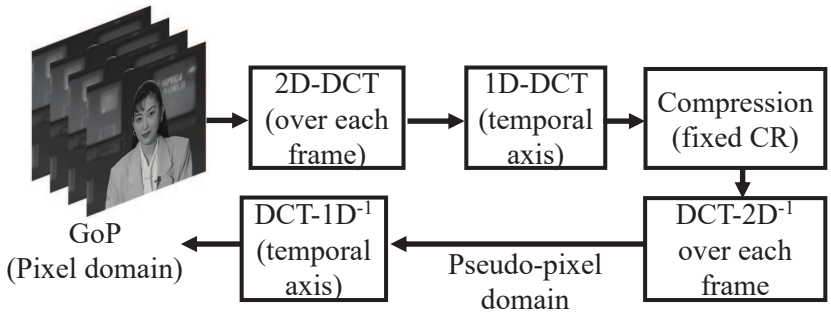


FIGURE 3.6 : Illustration de la localisation du domaine pseudo-pixel dans le schéma SoftCast.

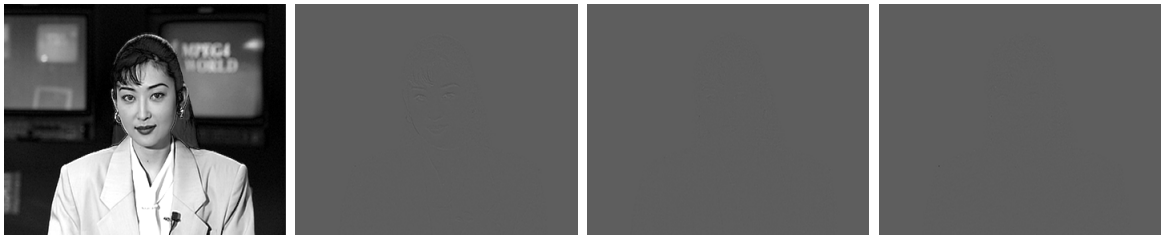
Lorsqu'aucune compression n'est appliquée (CR=1), cela représente en réalité l'effet de la DCT temporelle, comme indiqué sur la figure Fig. 3.7. Comme la DCT-1D inverse temporelle n'a pas encore été calculée, la dénomination pixel ne peut pas être utilisée. Cependant, en prenant la valeur absolue des coefficients résultants et en les écrêtant pour être compris dans l'intervalle [0-255], une comparaison visuelle peut être faite. Nous utilisons les quatre premières images des séquences vidéo *Akiyo* et *Husky*, ainsi qu'une séquence composite qui contient les deux premières images d'*Akiyo* et de *Husky* désignée par *Akiyo / Husky* séquence.

Les résultats en considérant qu'aucune compression n'est appliquée (CR = 1) sont donnés dans la Fig. 3.7. Comme on peut le constater, pour les séquences *Akiyo* et *Husky*, le premier plan (ou la première pseudo-image du GoP) avant le processus de DCT-1D inverse temporelle contient la plupart des détails (ou l'énergie), car elle représente en réalité les valeurs DC de chaque DCT temporelle appliquée aux positions spatiales (i,j). En ce qui concerne les pseudo-images temporelles qui suivent, nous pouvons observer une énorme différence entre les séquences *Akiyo* et *Husky*. Ceci est dû aux fortes disparités existantes entre ces deux séquences, qui contiennent respectivement des mouvements lents (TI bas) et de nombreuses variations temporelles (TI élevé). Malgré ces disparités, on peut globalement observer que la décorrélation temporelle fonctionne bien pour les séquences où aucun changement de scène (cut) n'apparaît à l'intérieur du GoP. En revanche, si un cut apparaît à l'intérieur d'un GoP (Fig. 3.7c), la DCT temporelle ne peut pas correctement décorréler le signal, ce qui résulte en une valeur énergétique relativement élevée pour chaque pseudo-image, représentée par un mélange des deux séquences dans le domaine pseudo-pixel. Comme expliqué ci-dessus, si aucune compression n'est appliquée, les images peuvent être entièrement récupérées sans apparition de l'effet fantôme. En revanche, si un processus de compression est nécessaire pour la transmission, il en résulte une perte d'informations considérable, les pixels ne peuvent pas être totalement récupérés et l'effet fantôme apparaît, comme indiqué dans les Fig. 3.4 et les Fig. 3.8, où 75 % des coefficients ont été jetés. Cette analyse met en évidence l'importance de la détection des changements de scène dans un contexte de transmission vidéo SoftCast.

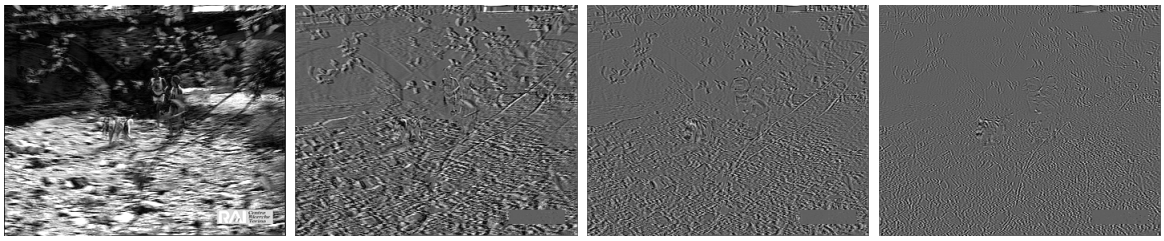
Nous venons de présenter les différents artefacts que nous avons observés aux cours de nos travaux. Ce recensement des différents artefacts vidéo introduits par les CVL est à notre connaissance le premier. Nous allons à présent étudier le ressenti global des utilisateurs par rapport à ces artefacts.

3.3 Evaluation subjective de la qualité vidéo dans un contexte SoftCast

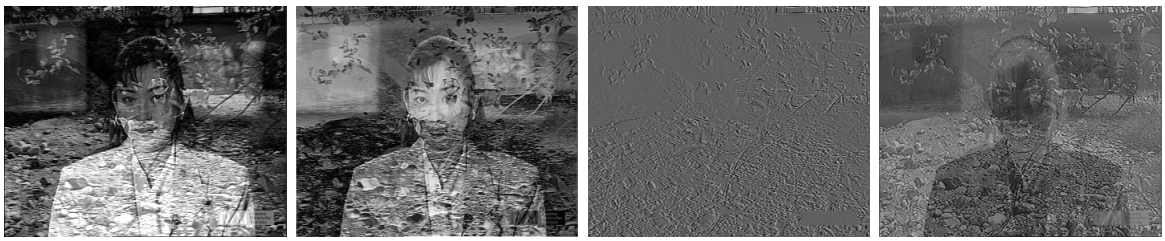
Dans cette partie, nous présentons les résultats de deux tests subjectifs évaluant la perception des utilisateurs quant à la qualité offerte par SoftCast à la réception. Pour évaluer ces artefacts, nous générons des vidéos reconstruites selon différents niveaux de CSNR, différents niveaux de compression (CR), différentes tailles de GoP et ceci pour les deux estimateurs ZF et LLSE. Un premier test permettant d'obtenir des scores MOS (Mean Opinion Score) est tout d'abord mené afin d'évaluer le ressenti global des utilisateurs soumis à une réception de contenu vidéo transmis via SoftCast. Les principaux artefacts notés sont l'effet de variation temporelle de la qualité ainsi que l'effet de neige classiquement observés dans les codeurs basés SoftCast. D'autre part, l'effet de flou engendré par l'utilisation d'un estimateur LLSE à la réception est également évalué au travers d'un autre test présenté en fin du chapitre.



(a) Pseudo-image issue de la DCT temporelle de la séquence *Akiyo*



(b) Pseudo-image issue de la DCT temporelle de la séquence *Husky*



(c) Pseudo-image issue de la DCT temporelle de la séquence composite *Akiyo/Husky*

FIGURE 3.7 : Comparaison visuelle de la DCT temporelle sur 4 images dans le domaine pseudo-pixel. Première ligne : séquence vidéo *Akiyo*. Deuxième ligne : séquence vidéo *Husky*. Troisième ligne : Séquence vidéo composite *Akiyo/Husky*. De gauche à droite : Première à quatrième pseudo-image du GoP.

(a) Résultat de la reconstruction de la séquence composite *Akiyo/Husky* après CR=0.25

FIGURE 3.8 : Comparaison visuelle des images reconstruites pour la séquence vidéo composite *Akiyo/Husky* après CR = 0.25. De gauche à droite : Première à quatrième pseudo-image du GoP.

3.3.1 Ressenti global de la qualité reçue

3.3.1.1 Choix du test

Les travaux de recherche présentés dans cette section ont été effectués au cours d'un séjour de recherche au sein du laboratoire LTCI (Laboratoire de Traitement et Communication de l'Information) de Télécom ParisTech et du laboratoire L2S (Laboratoire des Signaux et des Systèmes) de Centrale Supélec. Au cours de cette mobilité, nous avons tout d'abord cherché à évaluer le ressenti global de l'utilisateur sur la qualité obtenue à la réception avec un schéma de transmission SoftCast. Une illustration de la salle de test disponible dans les locaux de Télécom ParisTech est donnée en Fig. 3.9. Les paramètres précis de cette dernière seront présentés dans la Section 3.3.1.4.

FIGURE 3.9 : Illustration de la salle de test du LTCI (Télécom ParisTech).

Ainsi, pour ce premier test, le niveau de dégradation global perçu par l'utilisateur à l'issue d'une transmission SoftCast est évalué. Dans cette optique, nous avons fait le choix de recourir à la méthode DSIS séquentielle (Double Stimulus Impairment Scale) [106] présentée ci-dessous.

La méthode DSIS séquentielle consiste à présenter à l'utilisateur un stimulus A suivi d'un stimulus B représentant le même contenu vidéo, la différence se situe sur le fait qu'un des deux stimulus représente la vidéo originale, i.e., sans aucune distorsion tandis que le second représente la vidéo présentant des distorsions. Dans notre cas, cette dernière version représente la vidéo reçue et décodée au niveau d'un récepteur via le schéma de transmission SoftCast. L'observateur doit évaluer le niveau de dégradation de la vidéo reçue par rapport à la version originale. Pour ce test, la vidéo originale était toujours présentée au début (stimulus A) et explicitement mentionnée aux observateurs.

La méthode DSIS présente deux variantes I et II, spécifiant le nombre de fois où l'utilisateur visualise les stimuli A et B avant de voter. Des tests préliminaires effectués sur un nombre d'observateurs limité ont montré qu'il n'y avait pas vraiment d'intérêt à remontrer une seconde fois les séquences vidéo. De plus, étant donné le nombre important de stimuli à évaluer (différents niveaux de compression, différentes valeurs de CSNR et différentes tailles de GoP) et afin de ne pas dépasser la durée maximale fixée par l'UIT, nous avons donc choisi la méthode de type I où les stimuli A et B sont présentés de manière séquentielle une seule fois avant l'étape du vote. Dans notre cas, un stimulus DSIS auquel s'ajoute un temps de vote (choisi arbitrairement de 5 secondes) dure : $5+1+5+1+5 = 17$secondes (les $+1$ représentent la durée de l'écran gris séparant chaque étape).

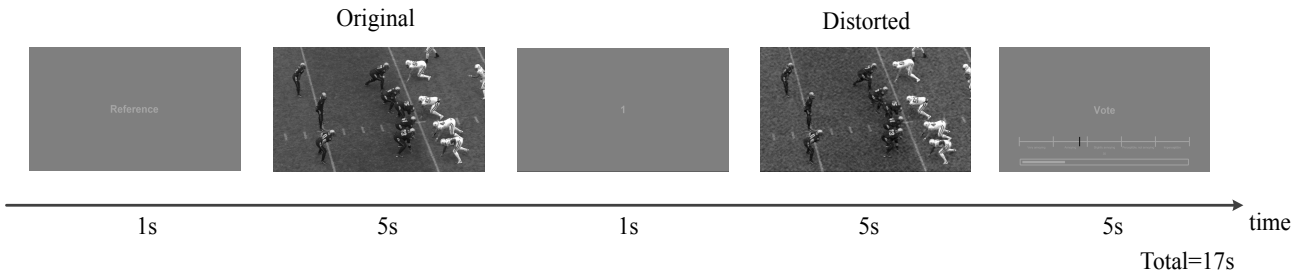


FIGURE 3.10 : Illustration de la méthode DSIS Type I.

L'échelle utilisée correspond à l'échelle de dégradation à 5 niveaux préconisée par l'UIT [2]. Celle-ci est graduée de 0 à 100 avec les correspondances suivantes :

- Imperceptible (80 à 100) : le niveau de dégradation est imperceptible ;
- Perceptible but not annoying (60 à 80) : Le niveau de dégradation est perceptible mais n'est pas gênant ;
- Slightly Annoying (40 à 60) : le niveau de dégradation est légèrement gênant ;
- Annoying (20 à 40) : le niveau de dégradation est gênant pour l'observateur ;
- Very annoying (0 à 20) : le niveau de dégradation est gênant pour l'observateur.

Nous avons fait le choix d'utiliser une échelle continue (illustrée dans la Fig. 3.11) afin de laisse le choix à l'utilisateur de se positionner comme il le souhaite entre les différents niveaux.

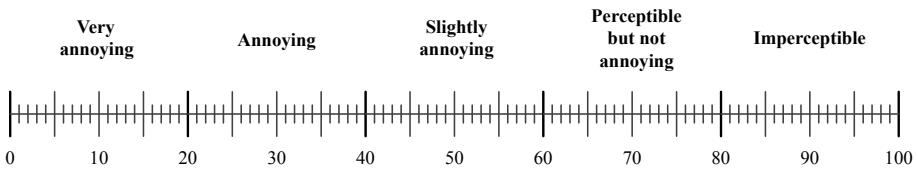


FIGURE 3.11 : Illustration de l'échelle de notation continue utilisée (5 niveaux de dégradation).

A la fin de la campagne de test, une moyenne des scores obtenues pour chaque stimulus est effectuée et le MOS est alors obtenu (Mean Opinion Score).

L'interface utilisée pour récolter les votes a été développée à partir des travaux de M. Emin Zerman (ancien Doctorant du LTCI). Celle-ci repose sur Matlab et la PsychToolBox [56]. Nous avons intégré dans l'interface la possibilité de faire des pauses pour l'observateur (à chaque fin de vote, s'il le désirait pour quelques raisons que ce soit). Des modifications ont été effectuées pour y intégrer les méthodes utilisées et présentées dans ce chapitre.

3.3.1.2 Choix des séquences

Le choix des séquences pour un test subjectif représente une étape importante dans la conception puisqu'il est nécessaire, à partir d'un échantillon restreint, de représenter le plus possible tous les cas pouvant être obtenus. En outre, fournir à l'observateur des contenus hétérogènes permet de ne pas le lasser durant le test et ainsi éviter qu'il ne donne des résultats aléatoires. Nous avons fait le choix de recourir aux contenus HD1080p dans ce test, puisqu'ils représentent un format classiquement utilisé de nos jours. Nous avons également fait le choix d'encoder seulement la luminance pour effectuer les tests subjectifs.

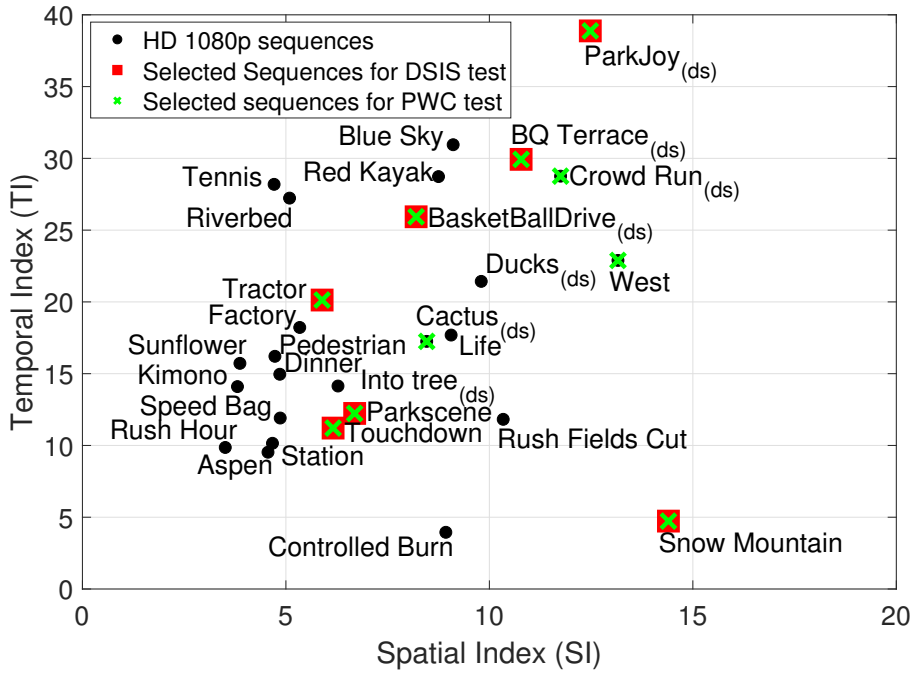


FIGURE 3.12 : Illustration des index spatio-temporels moyens calculés sur 5 secondes (durée d'un stimulus) pour les séquences vidéo HD1080p sélectionnées.

Un set de 28 contenus vidéo différents était à notre disposition. Les séquences ont tout d'abord été sous échantillonnées temporellement (si besoin) afin d'obtenir un frame rate compris entre [24∼30Hz], ceci afin d'éviter tout post traitement effectué par l'écran utilisé. Pour chaque séquence, nous avons calculé les index spatiotemporels définis dans le Chapitre 2 sur une durée de 5 secondes (durée d'un stimulus). Les index spatiotemporels sont disponibles en Fig. 3.12. Les séquences ayant bénéficiées du sous échantillonnage sont indiquées dans la figure à l'aide de l'acronyme (ds). A l'issue de cette cartographie, nous avons sélectionné 5 séquences tests ayant des caractéristiques hétérogènes parmi lesquelles : *BasketBallDrive*, *ParkJoy*, *ParkScene*, *Snow Mountain* et *Tractor* ainsi que 2 séquences ayant des caractéristiques proches des contenus évalués dans le test (*BQ Terrace* et *Touchdown*) pour l'étape d'entraînement. Ces différentes séquences sélectionnées sont indiquées dans la Fig. 3.12 via des carrés rouges. Une illustration de ces séquences est disponible en Fig. 3.13.

Un ensemble de vidéo a ensuite été généré à partir de ces contenus avec comme paramètres :

- Un CSNR variant de 0 à 30dB par pas de 3dB ;
- Trois tailles de GoP différentes : 8, 16 et 32 images ;
- Deux types d'estimateurs différents : LLSE et ZF ;
- Quatre niveaux de compression (i.e., quatre niveaux de bande passante disponibles) : CR=1, 0.75, 0.5 et 0.25.

Parmi l'ensemble des vidéos générées, nous en avons sélectionné un sous-ensemble afin de respecter les conditions suivantes :

- Tous les niveaux de l'échelle MOS doivent être équitablement représentés ;
- Les contenus et les conditions de transmission doivent être équitablement répartis ;
- La durée du test ne doit pas excéder 30 minutes (comme suggéré par les recommandations de l'UIT [2]).

Un total de 85 stimuli auxquels s'ajoutent 10 stimuli d'entraînement et 4 "dummies" (expliqués ci-dessous) a ainsi été retenu. Pour chacune des cinq séquences, 17 stimuli sont ainsi représentés. Nous avons fait le choix de restreindre le test DSIS à l'estimateur LLSE uniquement, en considérant les tailles de GoP 8 et 32 ainsi que les niveaux de compression CR=1 et CR=0.25. La comparaison des deux estimateurs a fait l'objet d'un autre test qui sera présenté dans la suite de ce chapitre. La durée du test incluant l'entraînement est ainsi d'approximativement $\frac{(85+4+10)\cdot 17}{60} \sim 28$min.

Pour chaque observateur, une liste aléatoire basée sur les 85 stimuli (liste complète en Annexe C) a été générée en portant une attention particulière sur le fait de ne pas montrer deux fois de suite le même contenu vidéo. De plus, 4 stimuli factices (dummies) ont été ajoutés au début de chaque test afin de laisser le temps à l'observateur de se familiariser avec

(a) *BasketBallDrive*

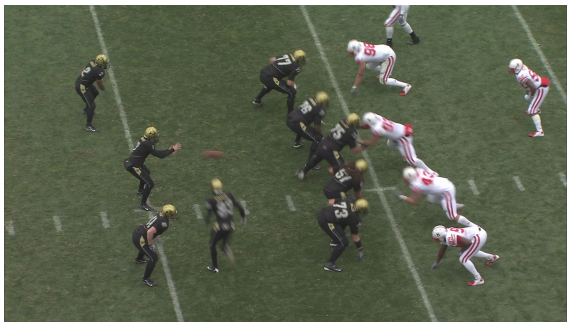
(b) *ParkJoy*

(c) *ParkScene*

(d) *Snow Mountain*

(e) *Tractor*

(f) *BQ Terrace*



(g) *TouchDown*

FIGURE 3.13 : Illustration des séquences sélectionnées pour le test DSIS. a) *BasketBallDrive*, b) *ParkJoy*, c) *ParkScene*, d) *Snow Mountain*, e) *Tractor*, f) *BQ Terrace* (Training), g) *TouchDown* (Training).

l'interface et les conditions de vote. Ces 4 stimuli sont rejoués à la fin du test et seuls les derniers votes sur ceux-ci sont comptabilisés dans les résultats finaux.

3.3.1.3 Observateurs

Afin de répondre aux recommandations de l'UIT, un nombre d'observateurs supérieur à 15 personnes a été choisi. Ainsi, 21 personnes ont pris part au test DSIS. Parmi ces personnes, un tiers étaient des femmes. La moyenne d'âge du groupe était de 29 ans avec un âge variant entre 25 et 47 ans. Le panel d'observateurs était constitué d'experts ($\leq 15\%$) ou non dans le domaine. Par expert, nous entendons les personnes familières des défauts de transmission vidéo. De même, par non expert, nous entendons les personnes n'étant pas familière du schéma SoftCast et n'ayant pas de connaissances spécifiques dans le domaine du traitement vidéo.

3.3.1.4 Procédure de test

Au début de chaque test, les observateurs sont accompagnés dans la salle de test dont les paramètres recommandés par l'UIT [3] sont synthétisés dans le Tableau 3.1.

Tableau 3.1 : Paramètre de la salle de test de ParisTech.

Parameter	**ITU Recommendation**	**Measured value**
Screen dimension	Specify L×H	55cm × 26cm
Viewing distance	2H to 8H	3H
Screen luminance	100 cd/m^2	<400cd/m^2
Ambient luminance	< 20 lux	∼2 lux
Background chromaticity	D65	D65
Color Temperature	6500K	6500K

Avant toute chose, un formulaire leur est donné (disponible en Annexe C) spécifiant clairement l'objectif du test, la durée, le protocole et les données qui seront conservées pour chaque utilisateur. Bien que les données recueillies soient anonymisées, afin de respecter les règles en vigueur du RGPD (Règlement général sur la protection des données), les observateurs sont informés des éventuels recours leur permettant de supprimer ces données. Celles-ci comprennent : le sexe, l'âge ou la tranche d'âge et le port ou non de dispositifs de correction de vision. La méthode d'Ishihara classiquement effectuée (pour détecter le daltonisme) avant le début d'un test n'a pas été prise en compte ici étant donné que les tests se font via la composante de luminance seule.

Ensuite, avant de démarrer le test, une discussion a lieu afin de vérifier que les observateurs ont compris le but du test et permet de répondre à leurs éventuelles questions.

Une phase d'entraînement a ensuite lieu où nous montrons à l'observateur l'interface du test, comment voter ainsi que les différents niveaux de dégradations qui vont devoir être

évalués. Une fois l'entraînement effectué, nous quittons la salle et laissons l'observateur effectuer le test. Enfin, à l'issue du test, les avis/commentaires des observateurs sont recueillis de manière informelle.

3.3.1.5 Analyse des résultats obtenus

La première étape avant toute analyse de résultat consiste en une détection d'*outliers* potentiels. Un *outlier* est défini par le fait que ses votes effectués durant le test dévient d'une manière importante du reste du panel d'observateurs. Nous calculons les scores MOS pour le test DSIS après avoir effectué cette étape. Nous utilisons la procédure fournie par l'UIT-T [2] (Section 2.3.1 de l'Annexe 2). Aucun outlier n'a été détecté durant cette procédure.

Les scores MOS en ensuite été calculé à l'aide de la formule suivante [2] :

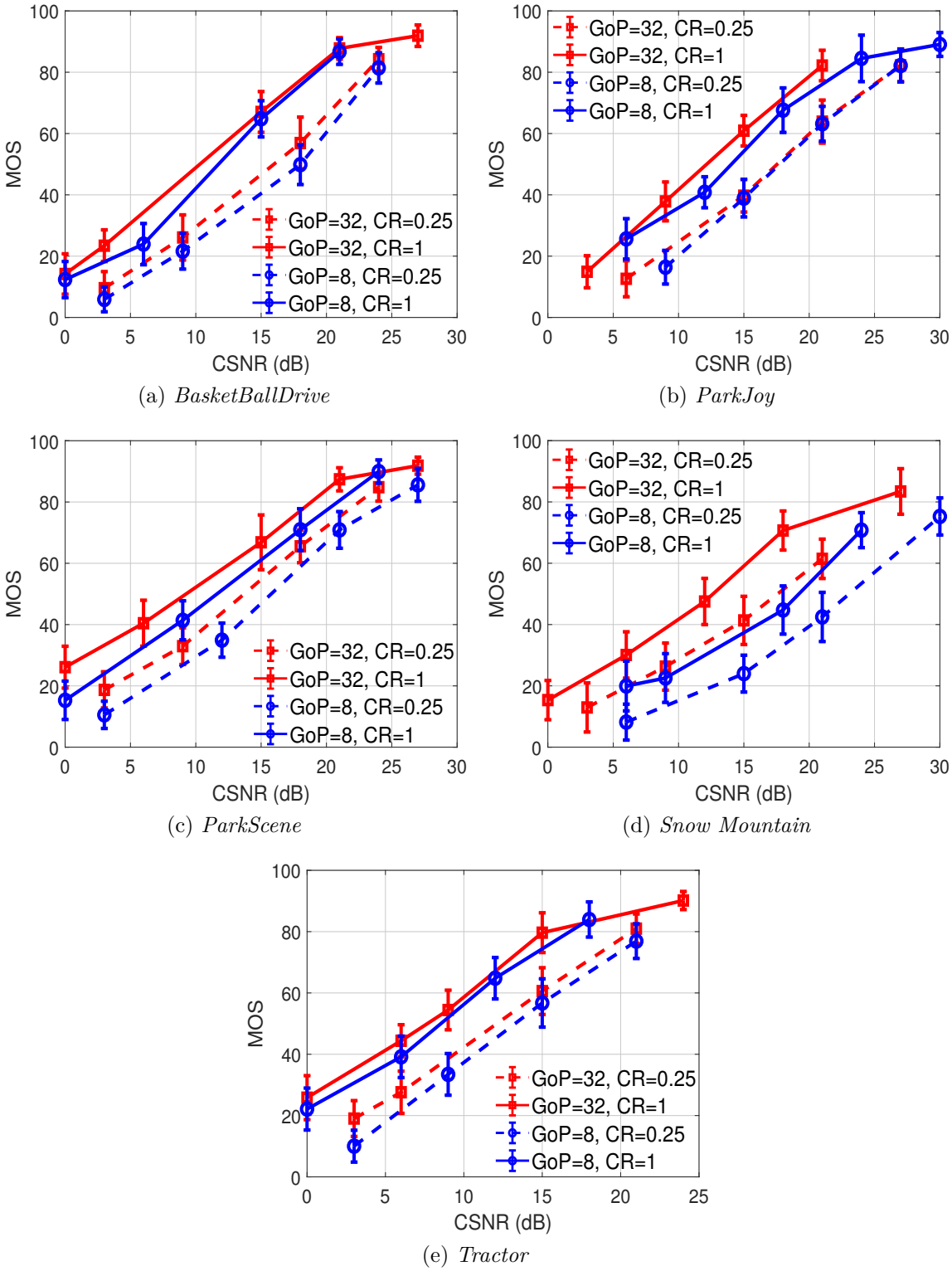
$$\mathrm{MOS}_k = \frac{1}{N}\sum_{i=1}^{N}\mathrm{OS}_{ik} \tag{3.5}$$

où N représente le nombre total d'observateurs après détection et suppression d'éventuels outliers, et OS_{ik} représente la note de l'observateur i pour le stimuli k.

Etant donné l'incertitude du score MOS (puisque seul un nombre limité d'observateurs est utilisé : ceux ayant pris part au test), l'UIT-T [2] recommande d'utiliser des intervalles de confiance (CI : Confidence Interval) associé aux scores MOS obtenus. Plus précisément, étant donné que le nombre d'observateurs est inférieur à 30 dans ce test, des intervalles de confiance à 95% sont calculés en assumant que les scores suivent une distribution *T de Student* [4]. Les résultats obtenus pour les cinq séquences sont illustrés dans la Fig. 3.14.

Nous observons que :

- Tout d'abord, le test DSIS est dans l'ensemble plutôt réussi. En effet, les intervalles de confiance pour les scores MOS ne sont pas larges et deviennent plus petits pour les deux extrema de dégradation (Very Annoying et Imperceptible). De plus, l'ensemble des niveaux de dégradation sur l'échelle sont bien représentés dans les résultats ;
- Il est intéressant de noter que l'atout clé des schémas de codage vidéo linéaire se retrouve au niveau des scores MOS. En effet, on observe bien que le score MOS évolue de manière plutôt linéaire en fonction de la qualité du canal de transmission. Toutefois, à haut CSNR ($\geq 20 \sim 25$dB), un effet de saturation apparait puisque les scores atteignent déjà le palier imperceptible ;
- Les scores MOS obtenus pour les cas CR=0.25 (75% des coefficients DCT-3D supprimés) sont logiquement inférieurs à ceux du cas CR=1 (pas de compression). Ceci n'est toutefois pas tout le temps vérifié comme on peut le voir sur la séquence *Snow Mountain* où le cas "Taille de GoP=32, CR=0.25" obtient des scores similaires au cas GoP=8, CR=1. Ceci illustre bien l'intérêt de l'augmentation de la taille du GoP pour



(a) *BasketBallDrive* (b) *ParkJoy*

(c) *ParkScene* (d) *Snow Mountain*

(e) *Tractor*

FIGURE 3.14 : Evolution des scores MOS obtenus en fonction de la qualité du canal. a) Séquence *BasketBallDrive*, b) *ParkJoy*, c) *ParkScene*, d) *Snow Mountain*, e) *Tractor*. Les points indiquent les scores MOS obtenus, les barres verticales associées indiquent les intervalles de confiance. Couleur rouge : taille de GoP=32. Couleur bleue : taille de GoP=8. CR=1 : Lignes en trait plein. CR=0.25 : Lignes tiretées.

ce type de séquence, où à score équivalent, 75% des coefficients peuvent être supprimés grâce à l'augmentation de la taille du GoP ;

- Les hypothèses émises dans le Chapitre 2 (grâce au modèle théorique du PSNR) concernant les tailles de GoP selon le type de contenu vidéo transmis sont globalement vérifiées. En effet, on observe bien l'intérêt d'augmenter la taille du GoP pour les séquences *ParkScene* et *Snow Mountain* où les informations spatiotemporelles sont faibles alors que ce n'est pas le cas pour les séquences *BasketBallDrive*, *ParkJoy* et *Tractor* présentant des informations spatiotemporelles élevées. Sur la base de ces conclusions, nous proposerons dans le Chapitre 5 un algorithme adaptatif permettant d'optimiser la taille du GoP en fonction des variations temporelles du contenu vidéo transmis.

A l'issue des tests subjectifs, les observateurs ont majoritairement mentionné que :

- Les artefacts engendrés par les codeurs vidéo linéaires sont clairement différents de ceux à quoi ils sont habitués (effet de blocs, gel d'image, etc.) ;
- Les artefacts liés aux codeurs SoftCast n'étaient pas pour eux vraiment gênants. En effet, même si le bruit est accentué à très bas CSNR, la vidéo reste clairement visualisable sans pertes d'informations. Par pertes d'informations, nous entendons le fait que l'ensemble des parties constituant la scène sont reconstituables (il n'y a pas dans SoftCast de gel d'image, d'écran noir).

Ceci illustre bien l'intérêt des codeurs vidéo linéaires. A titre illustratif, nous avons sélectionné l'un des commentaires recueillis à la fin du test pour illustrer les propos ci-dessus :

During subjective evaluation session I have observed following phenomena : Generally speaking this kind of artifacts are not that annoying. I personally, without training before viewing session, will never select very annoying. One of the possible reasons, there is no deformation on objects boundaries, so we can clearly see each object. The edges are sharp. If the noise is visible on the homogeneous areas like sky in the sequence with mountains, the main object in the scene - the mountains remain well recognizable and visible. I guess that the noise in homogeneous areas can be easily removed with even basic post-processing methods. It would be interesting to observe how the perception of quality will change in this case. Another possible reason could be that we are trained to see other kind of artifacts i.e. blocking. Even the people who are not involved in video/image research are familiar with blocking artifacts during watching streamed video or even images compressed with JPEG. That's why training session was useful to understand how the errors are scaled.

3.3.2 Performances des métriques objectives

Dans la suite de ce chapitre, nous nous intéressons aux performances des métriques objectives par rapport au scores MOS obtenus ci-dessus dans un contexte de transmission basée

sur des codeurs vidéo linéaires. Nous rappelons tout d'abord quelques métriques objectives couramment utilisées dans la littérature. Puis, des indicateurs de performances tels que le SROCC (Spearman Rank-Order Correlation Coefficient) ou encore le PCC (Pearson Correlation Coefficient) sont utilisés afin d'évaluer la corrélation existante entre ces métriques objectives et les scores subjectifs obtenus précédemment. Ces indicateurs sont présentés dans la suite de ce chapitre.

3.3.2.1 Métriques objectives considérées

Plusieurs métriques couramment utilisées sont considérées ici : le PSNR basé sur l'erreur quadratique moyenne (MSE), le SSIM, le DLM, l'index VIF et la nouvelle métrique développée par Netflix : VMAF. Ces différentes métriques sont brièvement introduites ci-dessous.

PSNR : Il est utilisé en tant que métrique purement objective alors que le SSIM fournit un indice de qualité, qui est davantage corrélé avec le système visuel humain (HVS) [41]. Le PSNR est donné par :

$$\mathrm{PSNR}_{\mathrm{dB}} = 10\log_{10}\left(\frac{R^2}{\mathrm{MSE}}\right) \tag{3.6}$$

où R = 255 et où la MSE est définie par :

$$\mathrm{MSE} = \frac{1}{N_R N_C N_F}\sum_{i=1}^{N_R}\sum_{j=1}^{N_C}\sum_{k=1}^{N_F}\left[(I(i,j,k) - \tilde{I}(i,j,k))^2\right] \tag{3.7}$$

avec N_R, N_C représentent la taille de chaque image, N_F représente le nombre d'images et $I(i,j,k)$, $\tilde{I}(i,j,k)$ dénotent respectivement les pixels de l'image originale et ceux reconstruits.

SSIM : Le SSIM est couramment utilisé en traitement d'image/vidéo couplé au PSNR. Celui-ci est toutefois jugé plus pertinent avec le HVS de par le fait qu'il évalue 3 types de dégradations : la modification de la structure, de la luminance et du contraste entre deux images successives. A l'issue du calcul, le SSIM fournit des valeurs comprises entre 0 (plus bas niveau de qualité) et 1 (plus haut niveau de qualité) [102].

DLM : La DLM (Detail Loss Metric) mesure deux types de défauts à savoir, les pertes de détails ainsi que les artefacts additifs (AIM : Additive Impairment Measure) tels que l'effet de bloc, le bruit, etc. Ces deux mesures sont effectuées après la prise en compte des propriétés du système visuel humain (HVS) via des courbes de sensibilité au contraste (CSF : Contrast Sensitivity Function) et de masquage du contraste [59]. Celles-ci sont ensuite fusionnées afin d'obtenir le score DLM.

VIF : La métrique VIF (Visual Information Fidelity) repose également sur des modèles HVS également et a pour vocation de quantifier la fidélité de l'information visuelle de l'image reconstruite [88]. Les scores obtenus vont généralement de 0 à 1. Des valeurs supérieures peuvent être obtenues pour des images présentant des contrastes accrus.

VMAF : Récemment introduit par Netflix, la métrique VMAF (Video Multimethod Assessment Fusion) a été pensée pour mesurer des artefacts de compression et de mise à l'échelle (quand la résolution d'encodage est inférieure à celle de l'écran d'affichage). Le schéma bloc est donné en Fig. 3.15. VMAF est une métrique qui n'est pas figée car elle utilise un ensemble de métriques objectives et repose sur des algorithmes de machine learning. A l'heure actuelle elle est constituée de :

- La métrique VIF calculée sur 4 échelles spatiales différentes (sous échantillonnage de l'image par 2^k avec $k = 0, 1, 2, 3$) ;
- La métrique DLM ;
- L'index TI défini dans le Chapitre 2 ;

Les résultats obtenus sont ensuite fusionnés via un algorithme basé sur le machine learning (dans le cas de VMAF un SVM est utilisé : Support Vector Machine) qui assigne un poids à chaque métrique. Le SVM a été préalablement entraîné les résultats de tests subjectifs réalisés sur les bases de données Netflix [8]. Un score est obtenu pour chaque image variant entre 0 (plus mauvais niveau de qualité) et 100 (image identique) et le score global est à l'heure actuelle un moyennage des scores sur la durée de la séquence.

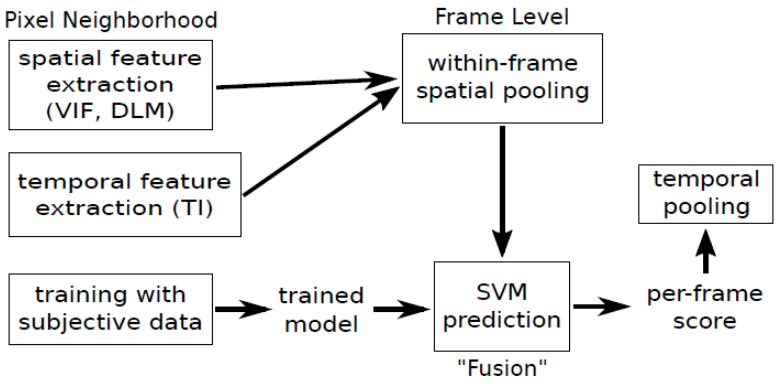


FIGURE 3.15 : Schéma bloc de la métrique VMAF. Figure issue de [5].

3.3.2.2 Indicateurs de performances utilisés

Pour évaluer les performances des métriques objectives et leurs capacités à prédire les scores subjectifs MOS, nous suivons les recommandations fournies par l'UIT [2, 1]. Ainsi, les résultats des métriques objectives obtenues ont tout d'abord été transformés via une régression non linéaire donnée par la fonction $Q(x)$ [34]. Ceci permet d'obtenir une relation de linéarité avec les scores MOS, et permet de convertir la métrique objective à la même échelle que les scores MOS i.e., entre 0 et 100, facilitant ainsi la comparaison des scores subjectifs

et objectifs et l'application des métriques citées ci-dessus. Une illustration du processus est donnée en Fig. 3.16.

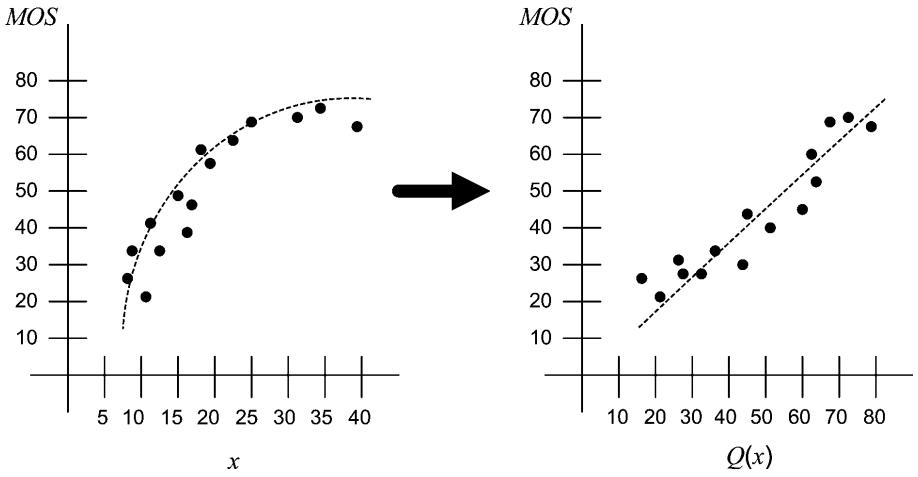


FIGURE 3.16 : Illustration de l'effet de la régression non-linéaire. Figure issue de [59].

La fonction $Q(x)$ est définie dans le rapport final de VQEG FR Phase I [34] comme suit :

$$Q(x_i) = \beta_2 + \frac{\beta_1 - \beta_2}{1 + e^{-\left(\frac{x_i - \beta_3}{|\beta_4|}\right)}} \tag{3.8}$$

où x_i représente le score obtenu avec la métrique objective considérée sur le $i^{ème}$ stimuli, et $Q(x_i)$ le score objectif transformé (ou de manière équivalente, le score MOS prédit). Afin de faciliter la compréhension de la suite de ce chapitre, les termes $Q(x_i)$ et x_i sont respectivement remplacés par les termes $M_{pred,i}$ et M_i où $M_{pred,i}$ et M_i définissent respectivement la métrique après prédiction et la valeur de la métrique considérée pour le $i^{ème}$ stimuli. Cette fonction cherche à minimiser, au sens de l'erreur quadratique moyenne, l'erreur entre les scores MOS et objectifs de la métrique considérée. Les différents paramètres optimaux β ont été obtenus par régression non linéaire pour chaque métrique considérée. Après avoir effectuée cette étape, nous pouvons évaluer les trois attributs suivants permettant de juger des performances entre la métrique objective considérée et les scores MOS obtenus :

- La précision de la prédiction (mesure de l'erreur moyenne entre le score MOS et la métrique objective) ;
- La monotonicité de la prédiction, c'est-à-dire la capacité de la métrique objective à suivre les variations des scores objectifs (une augmentation/diminution du score MOS se traduit par une augmentation/diminution de la métrique objective) ;
- La cohérence de la prédiction, c'est-à-dire la capacité de la métrique objective à donner des résultats cohérents pour l'ensemble du dataset.

Ces trois attributs sont évalués à l'aide de quatre métriques :

- Le Coefficient de Corrélation de Pearson (PCC) qui évalue la précision de la prédiction défini comme suit :

$$PCC = \frac{1}{n-1} \frac{\sum_{i=1}^{n} (MOS_i - <MOS>) \cdot \left(M_{pred,i} - <M_{pred}>\right)}{\sigma_{MOS} \cdot \sigma_{M_{pred}}}, \tag{3.9}$$

où n représente le nombre de stimuli évalué dans l'étude, MOS_i et $M_{pred,i}$ représentent respectivement le score MOS obtenu et le score objectif (après transformation), tous deux vis à vis du $i^{ème}$ stimuli, $< MOS >$ et $< M_{pred} >$ représentent respectivement la moyenne des scores MOS et la moyenne des scores pour la métrique considérée (après transformation), enfin σ_{MOS} et $\sigma_{M_{pred}}$ représentent respectivement les écarts-types des scores subjectifs et des scores de la métrique considérée (après transformation) ;

- La racine de l'erreur quadratique moyenne (RMSE : Root Mean Squared Error) qui évalue également la précision de la prédiction définie comme suit :

$$RMSE = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (MOS_i - M_{pred,i})^2}; \tag{3.10}$$

- Le Coefficient de Corrélation de Spearman (SROCC : Spearman Rank-Order Correlation Coefficient) qui évalue la monotonicité de la prédiction défini comme suit :

$$SROCC = \frac{\sum_{i=1}^{n} (R(MOS_i) - <R(MOS)>) \cdot \left(R(M_{pred,i}) - <R(M_{pred})>\right)}{\sqrt{\sum_{i=1}^{n} (R(MOS_i) - <R(MOS)>)^2 \cdot \left(R(M_{pred,i}) - <R(M_{pred})>\right)^2}}, \tag{3.11}$$

où $R(MOS_i)$ et $R(M_{pred,i})$ représentent respectivement les rangs de MOS_i et de $M_{pred,i}$ pour le $i^{ème}$ stimuli, et $< R(MOS) >$ et $< R(M_{pred}) >$ représentent respectivement les rangs moyens des scores objectifs et de la métrique considérée (après transformation) ;

- L'Outlier Ratio (OR) qui évalue la cohérence de la prédiction défini comme suit :

$$OR = \frac{\text{Nombre total d'outliers}}{n} \tag{3.12}$$

où $M_{pred,i}$, la métrique considérée (après transformation) est définie comme outlier dès lors où la valeur absolue de la différence entre le score MOS et la prédiction est supérieure à deux fois l'écart-type du score MOS :

$$|MOS_i - M_{pred,i}| > 2\sigma_{MOS}. \tag{3.13}$$

Bien que l'index PCC et l'index RMSE évaluent tous les deux la précision de la prédiction, ils sont complémentaires, puisque la RMSE est une mesure absolue de la différence entre les

valeurs obtenues et prédites, alors que l'index PCC évalue la relation de linéarité entre les scores obtenus et prédits.

3.3.2.3 Analyse des résultats obtenus

Un exemple d'illustration (scatter plot) présentant le processus de transformation des scores PSNR est donné en dans la Fig. 3.17. Les scatter plots pour les autres métriques sont présentés en Annexe C. Les cercles rouges vides représentent les différents stimuli évalués. Les cercles rouges pleins représentent les scores faisant partie des outliers (cf. (3.12)), la figure de gauche représente les scores avant transformation/prédiction alors que la figure de droite représente les scores transformés/prédits.

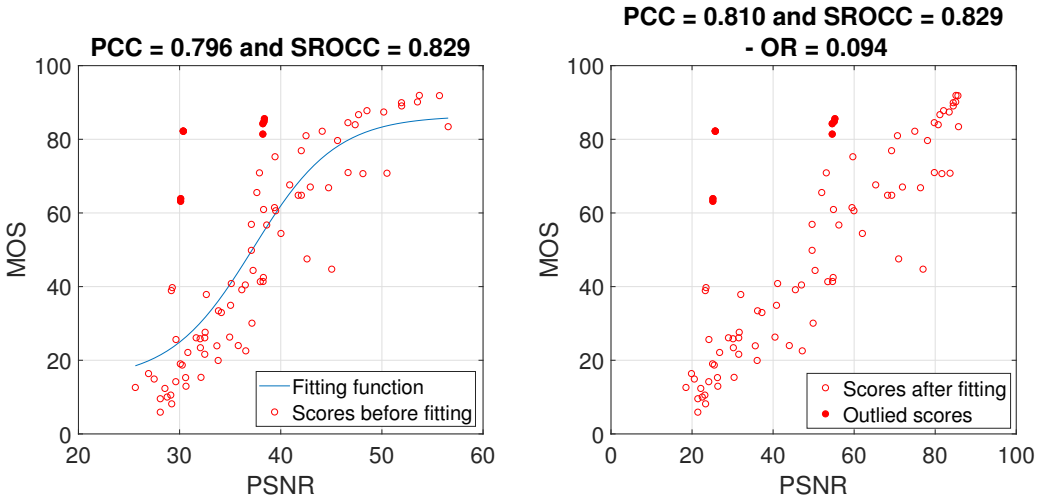


FIGURE 3.17 : Illustration du scatter plot MOS/PSNR. Gauche : Avant régression non linéaire. Droite : Après régression non linéaire.

Une synthèse des résultats obtenus pour les quatre métriques (PCC, RMSE, SROCC et OR) et les différents scores objectifs (PSNR, SSIM, DLM, VMAF et VIF) est disponible dans le Tableau 3.2. Comme nous pouvons le constater :

- Quelle que soit la métrique considérée, les scores obtenus sont globalement bons (très peu d'outliers, les corrélations sont élevées et ≥ 0.75) ;
- Les meilleurs résultats sont obtenus pour la métrique SSIM, suivie de la métrique VMAF. Il est intéressant de noter que bien que la métrique VMAF n'ait pas été entraînée sur des artefacts SoftCast, elle est à même de donner des scores très corrélés avec les scores MOS ;

- Hormis la métrique VIF_0, le PSNR obtient les moins bons scores bien qu'également satisfaisants (SROCC et PCC $\geq$0.81). Ceci est dû au fait que le PSNR et le CSNR sont liés par une relation linéaire et donc contrairement aux scores MOS obtenus, le PSNR ne sature pas à haut CSNR (quand CR=1), comme nous l'avons montré dans le Chapitre 2 à l'aide de (2.11). Au contraire, les autres métriques saturent (tous les comme les scores MOS) pour des valeurs importantes de CSNR ($\geq 20dB$), ce qui peut expliquer le score plus faible obtenu pour le PSNR.

Tableau 3.2 : Résultats des métriques $SROCC, PCC, OR$ et $RMSE$ pour le dataset complet (85 stimuli).

	PSNR	SSIM	DLM	VMAF	VIF_0	VIF_1	VIF_2	VIF_3
PCC	0.810	**0.963**	0.828	0.940	0.757	0.931	0.932	0.934
SROCC	0.829	**0.970**	0.840	0.943	0.772	0.939	0.938	0.942
OR	0.094	**0.000**	0.082	0.024	0.141	0.000	0.000	0.000
RMSE	15.897	**7.289**	15.191	9.245	17.703	9.914	9.851	9.649

Afin de vérifier l'hypothèse énoncée ci-dessus, nous avons refait les calculs en réduisant les stimuli dans le dataset. Plus particulièrement, les stimuli présentant des valeurs de CSNR$\geq$21dB ont tout d'abord été enlevés. Puis le dataset a encore été réduit en enlevant les stimuli ayant des CSNR$\geq$18dB. Les résultats sont respectivement disponibles dans les Tableaux 3.3 et 3.4.

Tableau 3.3 : Résultats des métriques $SROCC, PCC, OR$ et $RMSE$ pour le dataset contenant les stimuli pour les valeurs de CSNR$\leq$21dB (70 stimuli).

	PSNR	SSIM	DLM	VMAF	VIF_0	VIF_1	VIF_2	VIF_3
PCC	0.864	**0.952**	0.827	0.938	0.806	0.900	0.902	0.907
SROCC	0.868	**0.969**	0.846	0.947	0.818	0.915	0.914	0.922
OR	0.029	**0.000**	0.029	0.000	0.057	0.000	0.000	0.000
RMSE	11.977	**7.292**	13.398	8.234	14.088	10.392	10.277	10.031

Tableau 3.4 : Résultats des métriques $SROCC, PCC, OR$ et $RMSE$ pour le dataset contenant les stimuli pour les valeurs de CSNR$\leq$18dB (59 stimuli).

	PSNR	SSIM	DLM	VMAF	VIF_0	VIF_1	VIF_2	VIF_3
PCC	0.900	**0.949**	0.822	0.934	0.819	0.876	0.882	0.889
SROCC	0.900	**0.965**	0.838	0.933	0.822	0.893	0.894	0.905
OR	0.000	**0.000**	0.000	0.000	0.017	0.000	0.000	0.000
RMSE	8.999	**6.506**	11.763	7.394	11.831	9.937	9.729	9.429

En outre, étant donné la taille relativement petite de la base de données construite (total de 85 stimuli), nous utilisons également des techniques de bootstrap (création d'échantillons

virtuels supplémentaires à partir des résultats obtenus sur le dataset initial) afin de calculer des intervalles de confiance à 95% sur les valeurs obtenues pour les index SROCC et PCC. Dans notre cas, la fonction Matlab bootci est utilisée avec comme paramètres 2000 échantillons bootstrap [121]. Les résultats synthétisés des métriques de corrélation (SROCC et PCC) sont affichés sous la forme d'histogrammes dans les Figs. 3.18 et 3.19.

Comme nous pouvons le constater, lorsque les valeurs de CSNR entraînant des saturations de qualité sont supprimées du dataset, les index de corrélation obtenus pour le PSNR augmentent et passent de ∼0.81 à ∼0.9. Ceci est cohérent avec les conclusions obtenues dans [49] où il est stipulé que le PSNR fonctionne bien pour des distorsions de type additif.

Nous venons de voir que les métriques évaluées offrent globalement des performances intéressantes dans un contexte de transmission vidéo SoftCast. Nous allons à présent nous intéresser à l'effet de flou introduit par l'estimateur LLSE à bas CSNR et au ressenti de l'utilisateur par rapport à celui-ci.

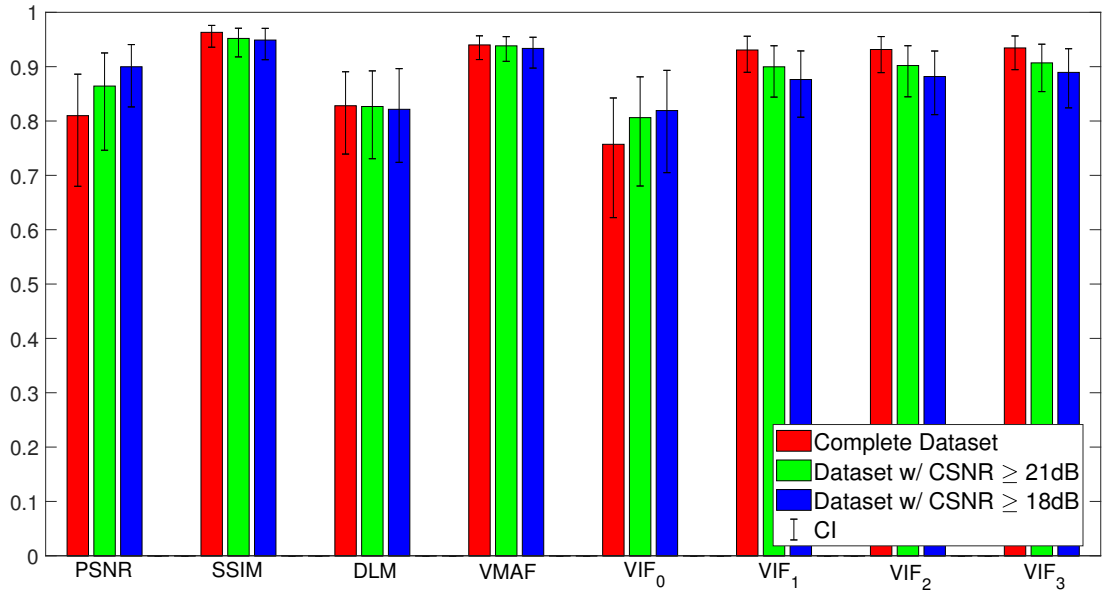


FIGURE 3.18 : Illustration des scores PCC. Les petites barres noires représentent les intervalles de confiance. Couleurs : rouge = dataset complet, vert = dataset avec les stimuli ≤21dB et bleu = dataset avec les stimuli ≤18dB.

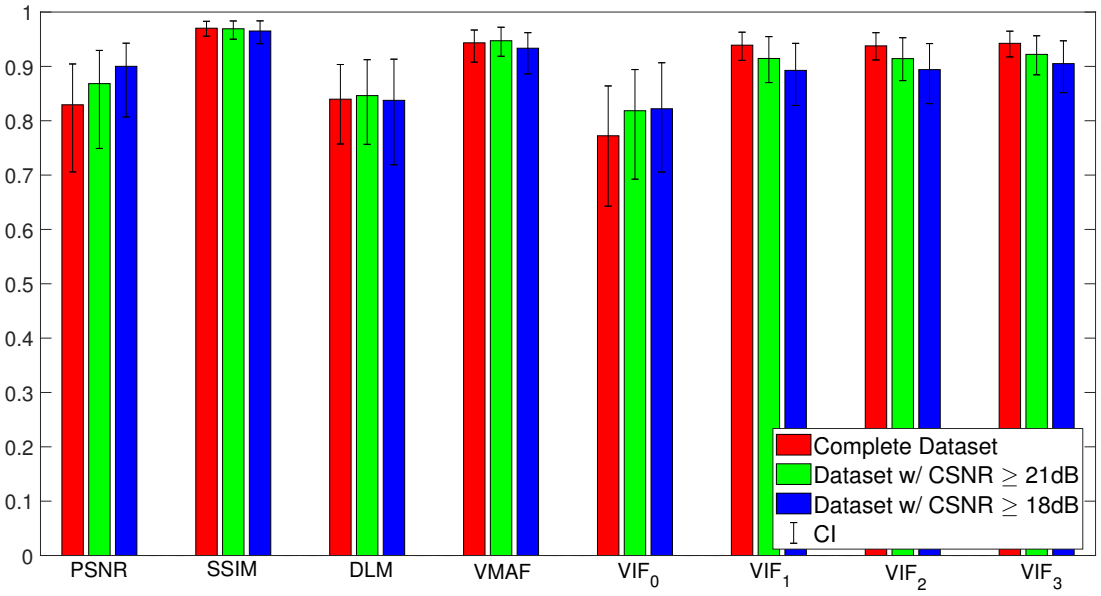


FIGURE 3.19 : Illustration des scores SROCC. Les petites barres noires représentent les intervalles de confiance. Couleurs : rouge = dataset complet, vert = dataset avec les stimuli ≤21dB et bleu = dataset avec les stimuli ≤18dB.

3.3.3 Préférences liées à l'estimateur utilisé

Comme nous venons de le voir dans la Section 3.2.3, un artefact gênant de SoftCast concerne le flou induit par le décodeur LLSE à la réception lorsque la qualité du canal est mauvaise (CSNR≤15dB). Bien que l'estimateur LLSE permet d'obtenir un gain en termes de PSNR par rapport à l'estimateur ZF pouvant aller jusqu'à 3dB (pour un CSNR de 0dB, cf. Chapitre 2), l'effet de flou engendré par un tel estimateur nous laisse penser qu'un estimateur ZF (ayant de moins bonnes performances en termes de PSNR) puisse être préféré à l'estimateur LLSE d'un point de vue visuel. Afin de confirmer ou d'infirmer cette hypothèse, nous avons décidé d'effectuer une deuxième campagne de tests subjectifs.

Celle-ci a été réalisée au sein du laboratoire IEMN-DOAE ou une salle de test subjectif a été installée en se rapprochant au plus possible des recommandations de l'Union Internationale des Télécommunications (ITU-R BT.500-13) [2] : luminosité de l'environnement, couleurs des murs, distance des observateurs par rapport à l'écran, etc. (cf. Fig. 3.20). Une attention particulière a été portée sur l'ambiance neutre, le calme de la salle ainsi que sur la lumière (isolation des sources de lumières externes, diffusion de la lumière sur les côtés et aucune réflexion de la lumière sur l'écran). La luminosité ambiante ainsi que la température de couleur ont été contrôlées pour être au plus proche des valeurs mesurées lors des tests à ParisTech (∼2 lux pour la luminosité et 6500K pour la température de couleur). La distance d'observation ainsi que le positionnement des observateurs ont également été contrôlés

FIGURE 3.20 : Illustration de la salle de test à l'IEMN-DOAE.

à chaque passage pour respecter la valeur de $3H$ où H représente la hauteur de l'écran (40cm) soit une distance par rapport à l'écran de 120cm et le fait que les yeux de l'utilisateur soient positionnés au milieu de l'écran.

3.3.3.1 Choix du test

Pour ce test, nous avons choisi de recourir à la comparaison par paire à choix forcée (Forced PWC : Forced PairWise Comparison) [106]. En effet, il semble plus naturel de recourir à cette méthode étant donné que le but est de discriminer un estimateur par rapport à l'autre. Le choix forcé permet de s'affranchir du choix de "facilité" consistant à dire que les deux estimateurs offrent des qualités similaires. Afin de faciliter le test et d'éviter de devoir mémoriser une version pour la comparer à l'autre, deux contenus vidéo identiques mais décodées de manière différentes (ZF ou LLSE) étaient présentés côte à côte à l'utilisateur. Pour pouvoir réaliser cette opération, les contenus vidéo HD1080p ont tout d'abord été croppés à la taille 952×1080 pixels avec une barre verticale de 16 pixels centrée afin de séparer les deux contenus. La position du crop (rectangle rouge) a été choisie manuellement afin d'isoler les informations les plus importantes de la séquence comme illustré dans la Fig. 3.22.

La méthode PWC est présentée dans la Fig. 3.21. Comme dit ci-dessus, deux contenus vidéo croppés sont tout d'abord présentés à l'utilisateur durant 5 secondes suivit d'un écran

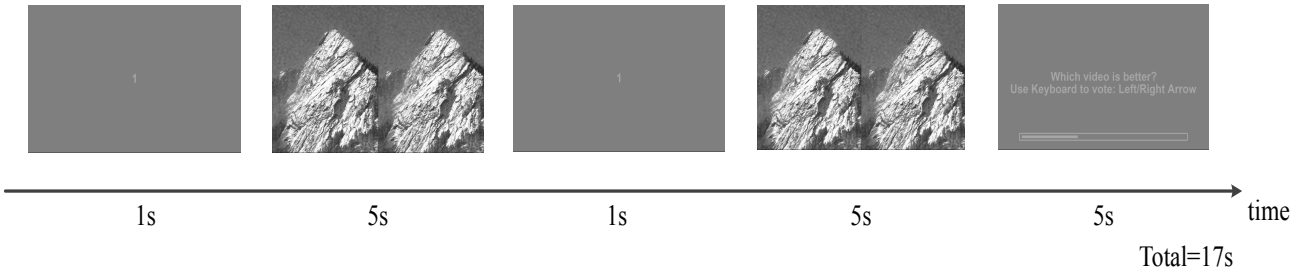


FIGURE 3.21 : Illustration de la méthode comparaison par paires (PWC) à choix forcé.

gris d'une seconde. Les mêmes stimuli sont ensuite rejoués une seconde fois (après un écran gris d'une seconde) afin de confirmer le choix de l'observateur. En effet, contrairement au test précédent, où une durée de 5 secondes était suffisante pour évaluer le niveau de dégradation, ici le défaut existant entre les deux décodeurs n'est pas toujours évident à voir (CSNR$\geq$15dB) et une seconde visualisation apparaissait nécessaire pour fixer le choix de l'observateur. Au total, un stimulus représente donc $5 + 1 + 5 + 1 + 5$=17 secondes.

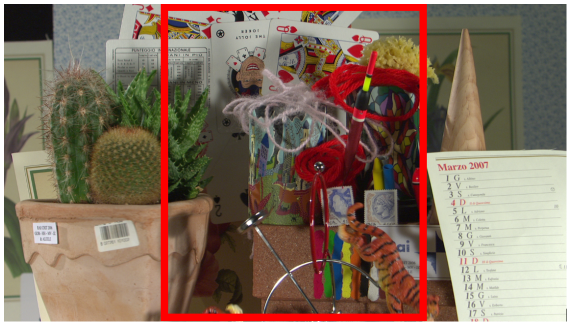
3.3.3.2 Choix des séquences

Comme précédemment, le test a été préalablement réfléchi afin de répondre aux critères de durée fixée par l'UIT (i.e., un maximum de 30 minutes par participant). De plus, les contenus vidéo ont été minutieusement choisis afin d'illustrer différents contenus spatio-temporels et différents contenus pouvant générer ou non du flou (présence de texte, contours élevés dans la scène, aucune différence visible, etc.). Parmi l'ensemble des contenus vidéo observés, nous avons tout d'abord gardé les cinq contenus évalués grâce au test DSIS et avons rajoutés trois séquences présentant du flou à bas CSNR et ayant des caractéristiques différentes (présence de texte, de nombres, etc.). Au final, les huit séquences suivantes ont été retenues : *Basket-Ball Drive, ParkJoy, ParkScene, Snow Mountain, Tractor* et *Cactus, CrowdRun, West*. Une illustration de ces contenus est disponible en Fig. 3.22 où le crop choisi est illustré en rouge.

La taille du GoP retenue pour ces tests a été choisie égale à 32 images comme il s'agit de la taille donnant généralement les meilleurs scores (objectif ou subjectif). De plus, nous notons que l'effet de flou est également observable pour des tailles de GoP de 8 ou 16 images.

En outre, contrairement au test précédent, où la gamme de CSNR devait être entièrement représentée (du fait de l'étude du niveau de dégradation perçu et afin de représenter l'ensemble de l'échelle de vote), dans ce test, il n'y a aucun intérêt à aller au-delà de 15$\sim$20dB étant donné que l'effet du LLSE n'intervient qu'à bas CSNR comme montré dans le Chapitre 2. Ainsi, nous avons choisi de n'inclure que quelques stimuli à des valeurs de CSNR supérieures à 18dB pour vérifier cela. Ceux-ci représentent environ 20% des stimuli.

(a) *BasketBallDrive*



(b) *Cactus*

(c) *CrowdRun*

(d) *ParkJoy*

(e) *ParkScene*

(f) *Snow Mountain*

(g) *Tractor*

(h) *West*

FIGURE 3.22 : Illustration des séquences sélectionnées et du découpage effectué (rectangle rouge) pour l'affichage côte à côte. a) *BasketBallDrive*, b) *Cactus*, c) *CrowdRun*, d) *ParkJoy*, e) *ParkScene*, f) *Snow Mountain*, g) *Tractor*, h) *West*.

3.3.3.3 Observateurs

Comme dans le test précédent et afin de répondre aux recommandations de l'UIT, un nombre d'observateurs supérieur à 15 personnes a été choisi. Compte tenu de l'aspect plus que subjectif du test proposé (image floue mais présentant moins de bruit ou image avec contours marqués mais présentant du bruit), nous avons tenu à recueillir un nombre plus important de votes en comparaison du test DSIS. Ainsi, 30 personnes ont pris part au test de comparaison par paire à choix forcé. Parmi ces personnes, 21 d'entre elles étaient des hommes. La moyenne d'âge du groupe était de 33 ans avec un âge variant entre 25 et 62 ans. Le panel d'observateurs était constitué d'experts ($\leq 20\%$) ou non dans le domaine.

La procédure de test reste similaire à la procédure du test DSIS. Toutefois, le nombre de stimuli relatif à l'entraînement est plus réduit puisqu'il n'y a pas les cinq niveaux de distorsion de l'échelle DSIS. Au final, trois stimuli sont utilisés pour illustrer les cas où l'effet de flou apparait de façon marquée, de façon moins marquée et quand il n'apparait pas du tout. Les séquences utilisées pour l'entraînement sont : *BQ Terrace* et *TouchDown* comme dans le test DSIS. De plus, comme précédemment une liste (avec 4 dummies) générée aléatoirement est générée pour chaque utilisateur. Toutefois, ici la liste aléatoire est générée aussi bien en changeant l'ordre des stimuli qu'en changeant le côté d'apparition des estimateurs ZF/LLSE.

3.3.3.4 Analyse des résultats obtenus

Comme précédemment, une détection d'*outliers* potentiels a été effectuée. Toutefois, étant donné qu'aucune recommandation de l'UIT-T n'est proposée quant à la détection d'outlier dans un contexte de ce type, nous avons eu recours au mécanisme de détection proposé par Mantiuk *et al.* [80] qui repose sur le calcul du maximum de vraisemblance par rapport à l'ensemble du panel d'observateurs. A l'issue de ce mécanisme de détection, un score est attribué à chaque observateur et ceux obtenant un score supérieur à 1.5 représentent les observateurs qui requièrent une analyse plus précise. Dans de tel cas, une comparaison visuelle est effectuée entre les réponses de cet observateur et le reste du panel mais le choix final de définir celui-ci comme outlier ou non est laissé libre au concepteur du test. Afin d'améliorer la probabilité de détection d'un outlier potentiel, nous avons ajouté aléatoirement certaines comparaisons par paire "évidentes" dans le test. Par exemple, un stimulus représente l'estimateur ZF dans un mauvais canal ($\leq$6dB) comparé à l'estimateur LLSE dans un très bon canal ($\geq$25dB). Parmi les 30 observateurs, le mécanisme de Mantiuk *et al.* a souligné un utilisateur demandant plus d'attention. Grâce aux stimuli "évidents" répartis dans le test et aux résultats associés à cet observateur nous avons conclu qu'il n'y avait pas d'outlier dans le panel.

Nous appliquons ensuite l'analyse statistique proposé par Hanhart *et al.* [38] pour déterminer si une différence entre les deux estimateurs est statistiquement observée.

Ainsi, pour chaque stimulus ZF/LLSE, nous calculons la probabilité de préférence de

choisir l'estimateur ZF plutôt que le LLSE notée :

$$P_{ZF} = \frac{w_{ZFi}}{N}, \tag{3.14}$$

où w_{ZFi} représente le nombre de votes en faveur de l'estimateur ZF pour le $i^{ème}$ stimuli et N représente le nombre total d'observateurs (après détection d'outlier).

Nous devons ensuite déterminer à partir de quel seuil nous pouvons considérer que la différence est statistiquement visible. Pour cela, nous prenons l'hypothèse de départ qui consiste à dire que les estimateurs ZF et LLSE ont à priori autant de chance d'être préféré l'un à l'autre. La probabilité de préférence est donc issue d'un processus de Bernoulli $B(N, p)$ où N représente le nombre total d'observateurs et $p = 0.5$, représente la probabilité de succès dans un essai de Bernoulli, suivant une loi binomiale. La fonction de distribution cumulative (CDF : Cumulative Distribution Function) $B(x, N, p)$, où x représente le nombre d'observateurs préférant le ZF au LLSE issue de cette loi binomiale est affichée en Fig. 3.23. Elle sert à déterminer les régions critiques qui serviront de seuils pour déterminer si ZF>LLSE, ZF=LLSE ou ZF>LLSE.

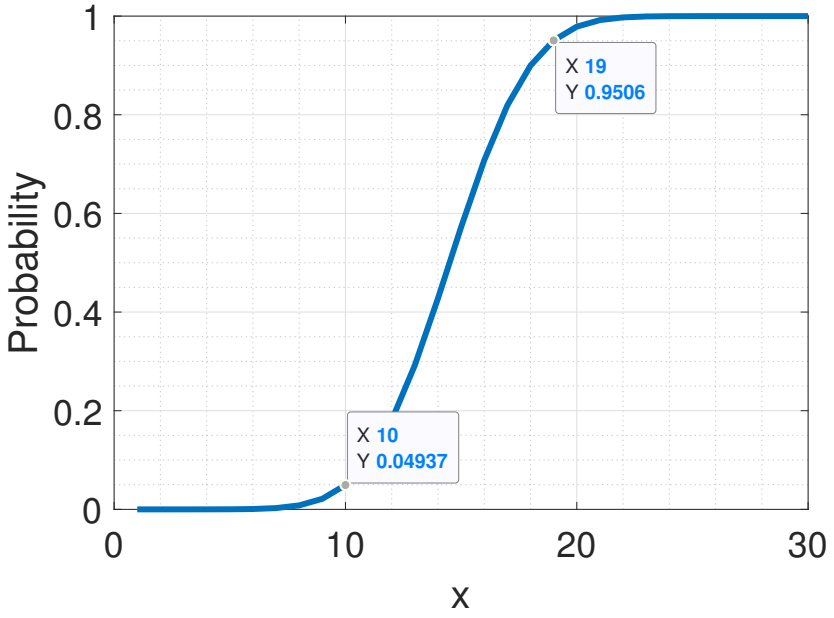


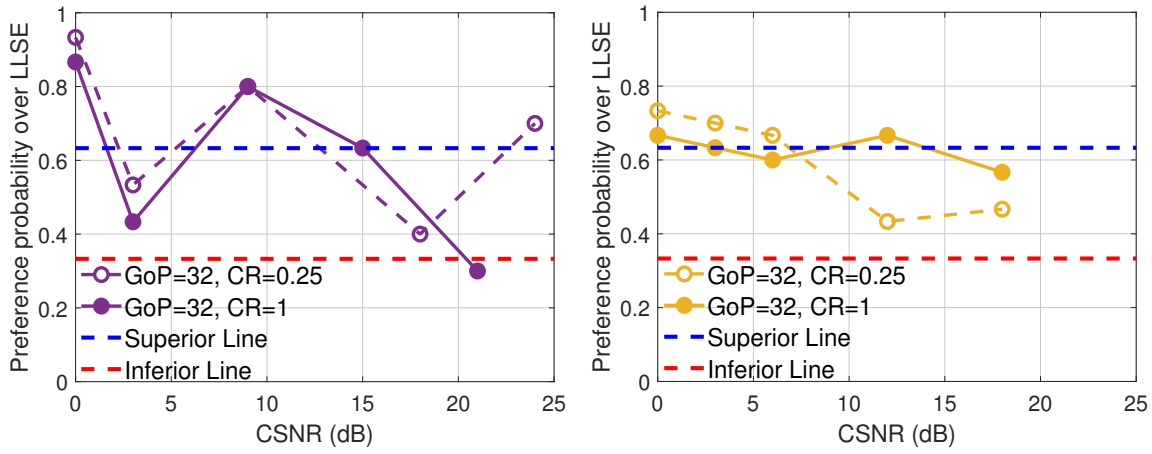
FIGURE 3.23 : Illustration de la fonction de distribution cumulative pour la loi Binomiale (30, 0.5).

A partir de cette fonction de distribution cumulative, nous déterminons les seuils de signification statistique à 5% et 95%. Comme $B(19, 30, 0.5) = 0.950$, nous considérons que lorsqu'au moins 20 observateurs sur les 30 ont choisi l'estimateur ZF, il y a statistiquement une meilleure qualité offerte par l'estimateur ZF que par l'estimateur LLSE. De même, lorsqu'il y

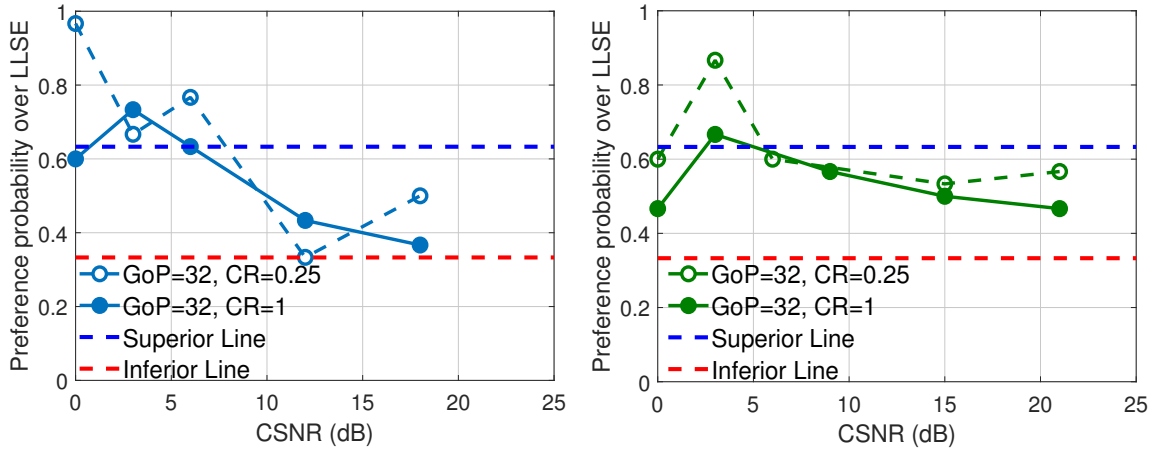
a moins de 10 observateurs ($B(10, 30, 0.5) = 0.049$) qui ont choisi le ZF, nous considérons que l'estimateur LLSE offre une meilleure qualité que le ZF. Entre, les deux il n'y a statistiquement pas de préférence. Ainsi, les régions critiques dans la probabilité de préférence du ZF par rapport au LLSE sont donnés par les intervalles :

- ZF<LLSE : $[0, \frac{10}{30}] = [0, 0.333]$;
- ZF = LLSE : $[\frac{10}{30}, \frac{19}{30}] = [0.333, 0.633]$;
- ZF>LLSE : $[\frac{19}{30}, 1] = [0.633, 1]$.

Ces régions critiques ont été indiquées en pointillés bleu (ZF>LLSE) et rouge (ZF<LLSE) dans chaque figure. Les résultats pour chaque vidéo sont disponibles en Fig. 3.24 et 3.25.

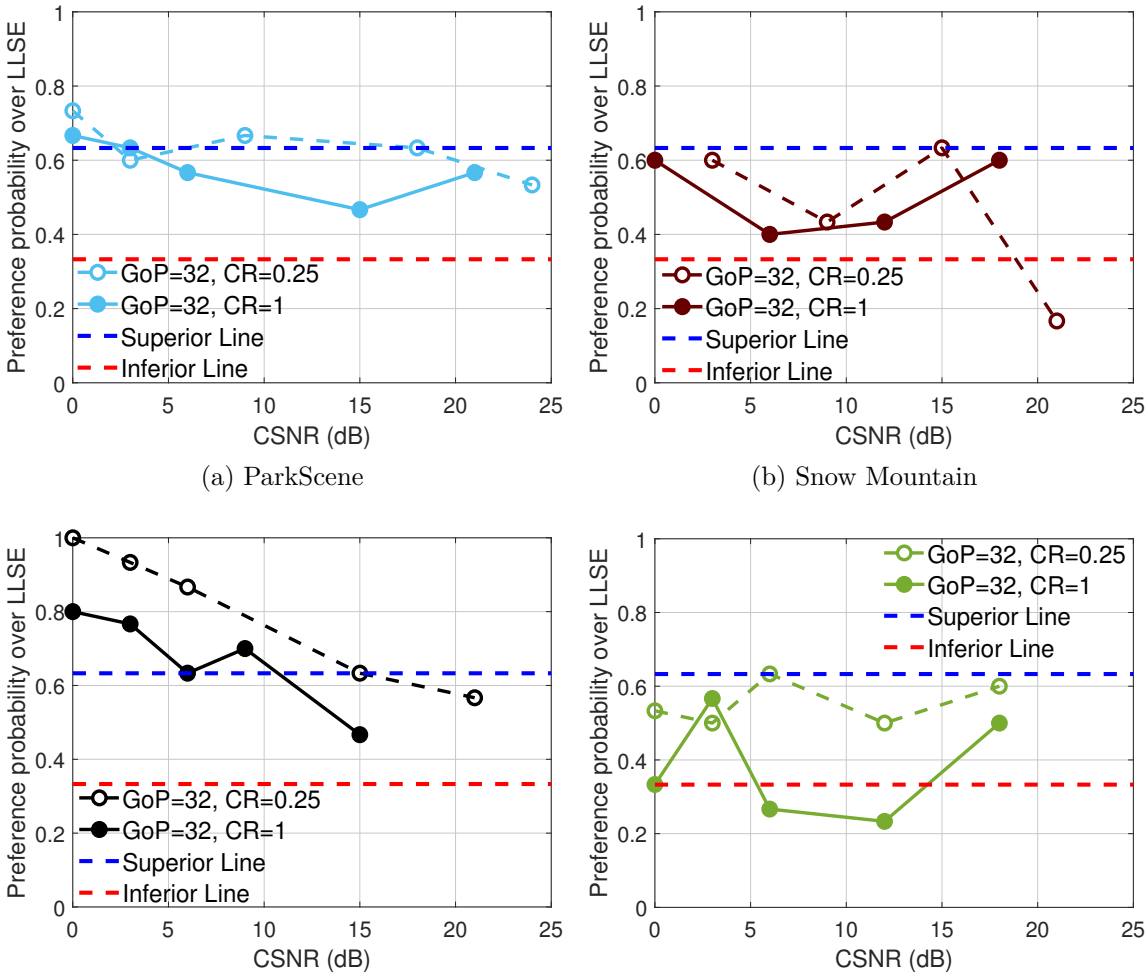


(a) BasketBall (b) Cactus



(c) CrowdRun (d) ParkJoy

FIGURE 3.24 : Evolution de la probabilité de préférence de l'estimateur ZF par rapport à l'estimateur LLSE en fonction du CSNR. Séquences : a) *BasketBallDrive*, b) *Cactus*, c) *CrowdRun*, d) *ParkJoy*.



(a) ParkScene (b) Snow Mountain

(c) Tractor (d) West

FIGURE 3.25 : Evolution de la probabilité de préférence de l'estimateur ZF par rapport à l'estimateur LLSE en fonction du CSNR. Séquences : e) *ParkScene*, f) *Snow Mountain* g) *Tractor*, h) *West*.

Les résultats montrent que :

- Au-delà de 15dB pour l'ensemble des séquences évaluées et comme prévu par le modèle théorique du Chapitre 2, il n'y a statistiquement pas de différence entre les estimateurs LLSE et ZF ;
- La préférence de l'estimateur ZF par rapport au LLSE est clairement marquée jusqu'à des CSNR de 15dB pour la séquence *Tractor* (g). En effet, les observateurs ont mentionné l'effet de flou clairement visible sur le logo du tracteur ;
- Bien que la séquence *West* (h) contienne beaucoup de contours floutés dus au texte qui défile, les utilisateurs n'ont globalement pas de préférence voire une petite préférence pour le LLSE. Ils ont en effet, pour beaucoup d'entre eux mentionnés à l'issue du test, que le fond noir où défile le texte est particulièrement bruité avec l'estimateur ZF là

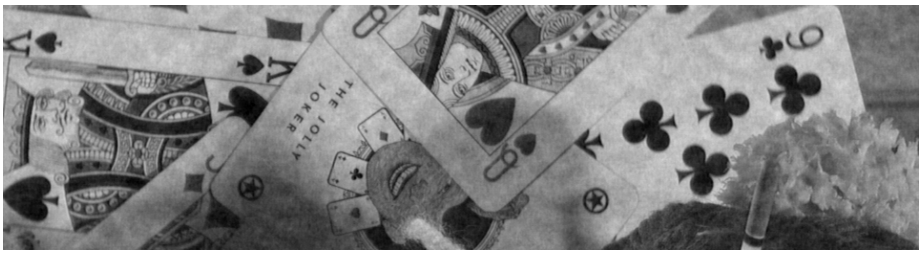
où l'estimateur atténue grandement celui-ci, ce qui influe donc sur le jugement global.

- La préférence de l'estimateur ZF par rapport au LLSE pour la séquence *Cactus* (b), n'est pas aussi marquée que ce que nous avions imaginé, bien que l'effet de flou soit clairement visible sur les cartes dans le fond de la scène (illustration de cette partie en Fig. 3.26). La raison est la même que pour la séquence *West*, les observateurs ont en effet mentionné que les objets bougent dans tous les sens et qu'il n'est pas facile de se focaliser sur un seul endroit pour essayer d'observer le flou ;
- A très bas CSNR, l'estimateur ZF est également statistiquement meilleur pour les séquences *BasketBallDrive* et *CrowdRun*. Nous notons toutefois une baisse de préférence observée pour la séquence Basket à CSNR = 3dB qui n'a logiquement aucune raison d'être. Nous pensons que cela est dû au fait que l'activité temporelle est trop importante, et que certains des observateurs aient voté aléatoirement sur cette séquence ;
- Aucune préférence sur l'ensemble de la séquence *Snow Mountain* est observée. Ceci est plutôt normal puisque nous n'avions remarqué aucune différence visible même à très bas CSNR lors de la sélection des matériels de test ;

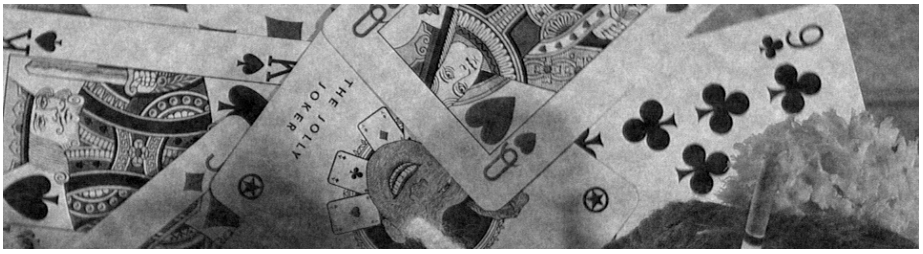
Pour vérifier les résultats obtenus et à titre illustratif, nous donnons dans l'Annexe C, les images d'erreur obtenues pour quelques séquences afin d'observer le type d'erreur engendré par les estimateurs ZF et LLSE ainsi que leur degré de visibilité. La configuration retenue est : taille de GoP = 32, CSNR = 0dB, CR=1.

Sur la base de ces résultats, nous pouvons conclure que la préférence de l'estimateur ZF par rapport au LLSE ne dépend pas uniquement de l'effet de flou engendré par l'estimateur LLSE mais dépend aussi beaucoup du type de contenu transmis et de l'activité spatiotemporelle de celui-ci. En effet, beaucoup d'observateurs ont mentionné le fait que les séquences étaient pour la plupart très rapides et qu'il n'était pas évident de se concentrer et détecter les différences entre les séquences, un effet de masquage temporel semble donc être observé pour ce type de séquences. Enfin, il convient de noter que l'estimateur ZF n'applique aucun traitement au niveau du bruit si ce n'est la remise à l'échelle des coefficients après transmission. Il serait intéressant de voir comment les préférences des observateurs sont modifiées avec l'utilisation d'un filtre en post traitement qui n'agirait que sur les parties homogènes/aplats de l'image afin de préserver les contours (défaut du LLSE).

(a) Image originale

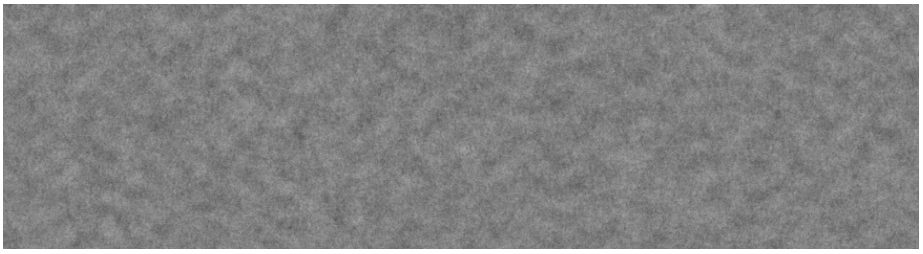


(b) Image reconstruite avec le LLSE



(c) Image reconstruite avec le ZF

(d) Image d'erreur résultante du LLSE



(e) Image d'erreur résultante du ZF

FIGURE 3.26 : Illustration du flou engendré par le LLSE. Zoom sur la séquence *Cactus*. Configuration : Taille de GoP = 32, CR = 1, CSNR = 0dB, image n°.125. a) Image originale, b) Image reconstruite avec le LLSE, c) Image reconstruite avec le ZF, d) Image d'erreur résultante du LLSE, e) Image d'erreur résultante du ZF.

3.4 Conclusion

Nous venons de voir dans ce chapitre différents artefacts pouvant être observés dans un contexte SoftCast tels que l'effet fantôme, l'effet de neige, les fluctuations temporelles de qualité ou encore l'effet de flou introduit par l'utilisation du décodeur LLSE à la réception. Ces artefacts ont été introduits, analysés et l'origine de ces différents artefacts a été dans la mesure du possible identifiée.

Parmi ces artefacts, l'effet de neige, les fluctuations temporelles de qualité ainsi que l'effet de flou ont fait l'objet de tests subjectifs pour étudier l'impact de ceux-ci sur la qualité d'expérience vécue par l'utilisateur. Ces tests subjectifs ont montré que :

- Les scores MOS obtenus suivent globalement une relation linéaire par rapport à la qualité du canal (CSNR), tout comme nous l'avons montré au travers des modèles théoriques du Chapitre 2, basés sur la métrique objective PSNR ;
- Les métriques les plus corrélées dans un contexte SoftCast sont les métriques SSIM et VMAF (bien que cette dernière n'ait pas été entraînée par rapport aux artefacts de SoftCast) du fait de leurs saturations de qualité à haut CSNR également identifiée dans les scores MOS. De plus, bien que le PSNR soit critiqué par la communauté scientifique, il obtient de bons scores de corrélation montrant qu'il peut être utilisé pour évaluer les performances des schémas de codage vidéo linéaires ;
- Une préférence pour une taille de GoP élevée (e.g. 32 images) est obtenue pour des contenus vidéo à index spatiotemporels bas alors qu'à l'inverse la préférence était moins claire (les scores MOS obtenus étant très proches), montrant l'efficacité des modèles théoriques proposés dans le Chapitre 2 ;
- A l'issue de la comparaison par paire, une légère préférence pour le décodeur ZF est statistiquement observé par rapport au décodeur LLSE introduisant un effet de flou. Cet effet étant particulièrement visible à bas CSNR (0∼15dB), la préférence est plus marquée dans de tels niveaux.

Dans les deux derniers chapitres, nous allons proposer des solutions originales permettant de réduire ou supprimer certains des artefacts introduits dans de chapitre.

Chapitre 4

Méthodes de prétraitement pour les systèmes de codage vidéo linéaire

Sommaire

4.1 Introduction

Nous venons de voir dans le chapitre précédent que l'artefact caractéristique de SoftCast est l'effet de neige (voir Section 3.2.1) qui vient se superposer sur le contenu vidéo. Particulièrement visible à bas CSNR et gênant pour l'utilisateur, nous proposons dans ce chapitre des méthodes de prétraitement permettant d'atténuer ce phénomène.

Pour ce faire, des méthodes existantes de la littérature sont tout d'abord introduites et analysées. Celles-ci agissent soit dans le domaine pixel soit dans le domaine fréquentiel et permettent de réduire l'effet de neige au prix d'une augmentation du temps de calcul ainsi que des métadonnées à transmettre.

Afin de minimiser ces inconvénients, une méthode de prétraitement alternative et originale est introduite. En fonction de l'application et des contraintes (bande passante disponible, délai maximal toléré, etc.) qui lui sont associées, une méthode de prétraitement peut être préférée à l'autre.

Comme nous le verrons dans le chapitre, les méthodes de prétraitement consistent en une réduction de l'énergie des coefficients transmis en (pseudo)-analogique. Ceci permet une meilleure répartition de la puissance P disponible lors de l'étape de scaling, offrant ainsi une plus grande protection face aux perturbations du canal.

4.2 Etat de l'art des méthodes de prétraitement

Plusieurs méthodes de prétraitement des données dans un contexte SoftCast ont été proposées dans la littérature. Afin d'introduire et d'analyser celles-ci, nous distinguons dans la Fig. 4.1 l'emplacement des prétraitements agissant dans le domaine pixel (bloc bleu) de ceux agissant dans le domaine fréquentiel (i.e., agissant après la 3D-DCT, bloc vert). Nous notons que les résultats et les conclusions présentés dans ce chapitre sont valables quel que soit le type d'estimateur utilisé (ZF, LLSE). Toutefois, nous choisissons de représenter uniquement les résultats pour l'estimateur LLSE étant donné qu'il représente la version originelle de SoftCast.

4.2.1 Prétraitement dans le domaine pixel

Hagag *et al.* [37] ont récemment utilisé un prétraitement dénoté ici (P1). Ce pré-traitement consiste à enlever la moyenne d'une image à chaque pixel la constituant avant d'effectuer les opérations de transformation de SoftCast. Pour pouvoir reconstruire l'image à la réception, il est nécessaire de transmettre chaque moyenne d'image en métadonnées additionnelles (8 bits par image).

Une solution alternative existante auparavant et dénotée (P2) dans la suite de cette thèse

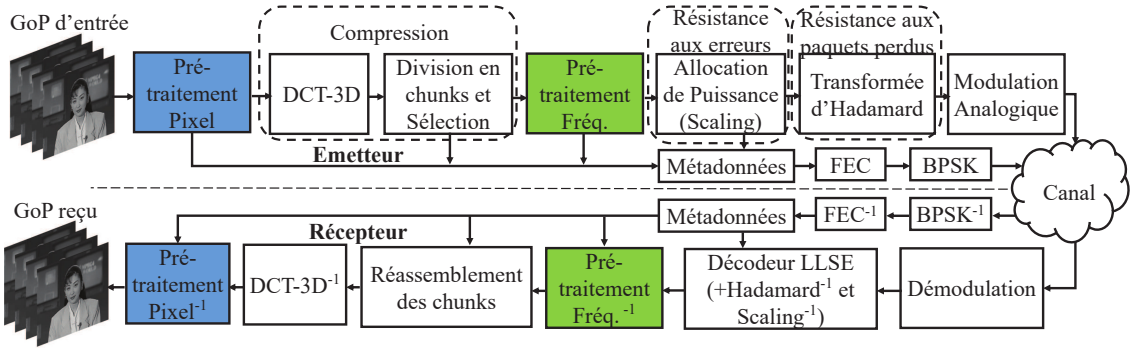


FIGURE 4.1 : Diagramme bloc du schéma de transmission vidéo SoftCast incluant des blocs de prétraitement.

a été proposée par Cui *et al.* [15]. Contrairement à la méthode précédente (P1), cette dernière consiste à soustraire 128 à tous les pixels, (représente un niveau de gris moyen sur 8 bits). Ceci permet d'éviter une augmentation de la bande passante allouée pour la transmission des métadonnées additionnelles. En effet, l'ajout d'un décalage de +128 après le processus de décodage ne nécessite l'envoi d'aucune information supplémentaire au récepteur.

Les deux méthodes de prétraitement ci-dessus agissant dans le domaine pixel (et indiquées en vert dans la Fig. 4.1) peuvent être synthétisées comme suit :

$$I_{proc}(i,j,k) = I(i,j,k) - \bar{p}(k) \tag{4.1}$$

où I_{proc} et I désignent l'image après le prétraitement et l'image d'origine respectivement, (i,j) les coordonnées pixels, k l'index temporel et $\bar{p}$ le terme soustrait. Dans la première méthode $\bar{p}(k) = \lfloor mean(f(i,j,k)) \rceil$ et $p(\bar{k}) = 128$ dans la seconde. $\lfloor \bullet \rceil$ dénote l'opération d'arrondi au nombre entier le plus proche.

Nous notons que la méthode (P1) peut être de manière équivalente effectuée dans le domaine fréquentiel. En effet, nous rappelons pour cela les équations (1.1) et (1.2) de la DCT vues dans le Chapitre 1 et nous nous focalisons sur le cas de la DCT spatiale i.e., la DCT en 2 dimensions. Le calcul de la composante fréquentielle DC donne :

$$\begin{aligned} \mathbf{F}(0,0) &= \sum_{i=0}^{N_R-1} \sum_{j=0}^{N_C-1} f(i,j) \cdot C_{i,0} \cdot C_{j,0} \\ &= C_{i,0} \cdot C_{j,0} \sum_{i=0}^{N_R-1} \sum_{j=0}^{N_C-1} f(i,j) \\ &= \frac{1}{\sqrt{N_R}} \cdot \frac{1}{\sqrt{N_C}} \cdot N_R \cdot N_C \cdot \bar{f} \end{aligned} \tag{4.2}$$

où $C_{i,0} = \frac{1}{\sqrt{N_R}}$ et $C_{j,0} = \frac{1}{\sqrt{N_C}}$, avec N_R, N_C représentant la taille de l'image et $\bar{f}$ dénotant la valeur moyenne de l'image dans le domaine pixel.

Cependant, il n'y a pas vraiment d'intérêt à effectuer ce prétraitement dans le domaine fréquentiel puisqu'un simple calcul permet de montrer que la transmission de chaque composante DC (issues de la DCT-2D spatiale) en métadonnées représente plus de bits que la transmission de l'image moyenne obtenue dans le domaine pixel. Prenons le cas d'une séquence vidéo CIF (352×288 pixels). Le calcul montre que la composante DC sera représentée sur au moins 18 bits (8 bits pour coder la valeur moyenne de l'image $\bar{f}$, 1 bit de signe et 9 bits supplémentaires nécessaires pour coder l'opération mathématique $\sqrt{352} \cdot \sqrt{288}$) alors que juste 8 bits sont nécessaires quand la valeur moyenne de chaque image est soustraite directement dans le domaine pixel.

En revanche, une méthode de prétraitement plus intéressante en termes de gains obtenus agissant elle aussi dans le domaine fréquentiel a été proposée et est introduite dans la suite de ce chapitre.

4.2.2 Prétraitement dans le domaine fréquentiel

Proposée par He *et al.* [40], le prétraitement OPA, **Optimized** Power Allocation, est à ne pas confondre avec l'allocation de puissance optimale (**Optimal** Power Allocation, dénotée SoftCast+) vue dans le Chapitre 2. Dénoté ici P3, il fonctionne dans le domaine fréquentiel et s'insère après la 3D-DCT et la division en chunks comme indiqué en bleu dans la Fig. 4.1.

OPA-SoftCast est un algorithme itératif qui vise à réduire l'énergie des chunks transmis en (pseudo)-analogique. Ceci est rendu possible en supprimant et en transmettant à part, en métadonnées additionnelles, N_d coefficients DCT hautement énergétiques, appelés coefficients fréquentiels spéciaux (SFC, Special Frequency Components).

Les données d'entrées de l'algorithme sont représentées par la matrice $\mathbf{X}$ dont chacune des lignes représente un chunk de moyenne nulle. Tout d'abord, l'énergie de chaque chunk est calculée et la valeur la plus élevée parmi tous les chunks est sélectionnée. Le coefficient fréquentiel le plus énergétique (SFC) dans ce bloc est mis à 0 et envoyé en métadonnée additionnelle. Enfin, la moyenne du bloc sélectionné est ensuite ajustée pour conserver une valeur moyenne nulle. À la fin de la boucle, la nouvelle énergie du bloc sélectionné est calculée et mise à jour. La boucle est exécutée N_d fois où N_d représente le nombre de SFC sélectionnés dans le GoP. En raison de :

- La position inconnue des chunks,
- La position inconnue des SFC les plus énergétiques,
- La valeur des SFC retenues,
- Des moyennes ajustées à chaque itération,

OPA-SoftCast doit transmettre ces 4 informations pour chaque SFC supprimé afin de pouvoir reconstruire les plans DCT du côté du récepteur.

Les auteurs ont supposé que chaque information est quantifiée avec 20 bits en moyenne. Avec une modulation BPSK et un code FEC de redondance 1/2, le nombre total de bits à transmettre pour chaque GOP est de [40] :

$$4 \cdot 20 \cdot 2 \cdot N_d = 160 \cdot N_d \tag{4.3}$$

Par conséquent, la largeur de bande nécessaire pour la transmission de ces métadonnées additionnelles est donnée par :

$$\frac{160 \cdot N_d \cdot F_r}{N_F} \tag{4.4}$$

où F_r et N_F désignent respectivement le nombre d'image par seconde et la taille du GoP. Afin de limiter l'augmentation de la bande passante liée aux métadonnées, les auteurs ont proposé un compromis entre amélioration de la qualité et quantité supplémentaire de métadonnées. Ils ont sélectionné un seuil équivalent à 2 SFC par image soit 16 SFC pour un GoP de 8 images. Toutefois, Fujihashi *et al.* ont montré qu'il est primordial de limiter la quantité de métadonnées, car celles-ci entraînent une dégradation de la qualité vidéo en raison de la perte de puissance et de débit pour la transmission des chunks eux-mêmes [27]. Dans ce but, nous proposerons dans la Section 4.4 une alternative à OPA-SoftCast permettant de réduire le volume de métadonnées additionnelles ainsi que le temps de calcul nécessaire pour effectuer l'opération de prétraitement.

Une évaluation des méthodes de prétraitement existantes est tout d'abord présentée afin d'évaluer les avantages et inconvénients de chacune d'entre elles.

4.3 Performances des méthodes existantes

4.3.1 Prétraitement pixel

4.3.1.1 Environnement de simulation

Vidéos considérées : Les méthodes de prétraitement pixels sont évaluées et comparées par le biais de simulations approfondies. La luminance de séquences CIF ou HD720p de la collection Xiph [113] est utilisée. Le processus d'encodage est exécuté GoP par GoP avec une taille de GoP de 16 images et chaque image est divisée en 64 chunks (e.g. 44×36 pixels pour les séquences CIF), comme dans [27, 112]. Comme dans les chapitres précédents, quatre bandes passantes disponibles sont ici considérées représentées par les CR=1, 0.75, 0.5 et 0.25.

Caractéristiques du canal : Des transmissions via des canaux AWGN dans la plage [0∼30dB] sont considérées. La puissance totale de transmission est normalisée à $P = 1$. Pour assurer une comparaison équitable, le même bruit est généré et appliqué à toutes les méthodes.

Métriques d'évaluation : Deux métriques sont ici considérées : le PSNR et l'index SSIM.

En effet, l'index SSIM présente la meilleure corrélation avec les scores MOS obtenus dans le Chapitre 3. Le PSNR quant à lui est utilisé pour sa simplicité d'application et compte tenu de ses scores corrects en termes de corrélation avec les valeurs MOS obtenues précédemment.

4.3.1.2 Résultats de simulation

Dans les tests suivants, nous choisissons les séquences *Australia*, *News* et *Stefan* en raison de leurs caractéristiques spatio-temporelles hétérogènes. *Stefan* contient des activités temporelles et spatiales élevées. En revanche, *Australia* et *News* présentent respectivement des mouvements lents et des contenus spatiaux bas à moyens. Les résultats obtenus avec d'autres séquences CIF et HD720p sont similaires.

Pour évaluer ces disparités, les index spatio-temporels [51] déjà abordés dans les chapitres précédents sont utilisés. Nous rappelons cependant que nous avons choisi de moyenner les résultats SI, TI sur toute la séquence au lieu de prendre la valeur maximale dans la définition originelle. La Fig. 4.2 rappelle les valeurs moyennes résultantes du couple (SI, TI) obtenues pour chaque séquence.

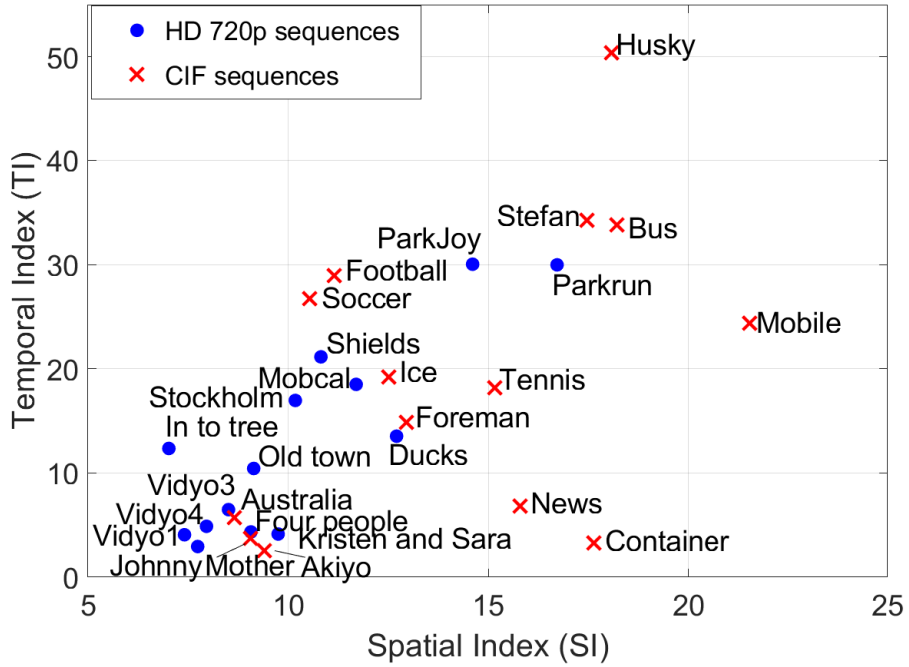


FIGURE 4.2 : Illustration des index spatio-temporels moyens pour les séquences vidéo HD et CIF sélectionnées.

La Fig. 4.3 présente les résultats moyens de qualité vidéo reçue en termes de PSNR et SSIM pour le schéma originel SoftCast où aucun prétraitement n'est appliqué et les deux autres où la soustraction dans le domaine de pixel est effectuée avant le processus d'encodage.

Un zoom sur la partie [5∼25dB] a été effectué afin de mieux observer les différences. Nous avons choisi d'illustrer le cas CR=1 (pas de compression) dans cette section avec une taille de GoP=16 images, et avons vérifié des comportements similaires pour les autres tailles de GoP et pour les cas où la bande passante est restreinte.

Comme nous l'avons vu dans le Chapitre 2, lorsqu'aucune compression de la vidéo n'est nécessaire (CR=1), nous pouvons observer la linéarité existante entre les valeurs de PSNR et les valeurs de CSNR (voir l'équation théorique (2.11)).

Ensuite, quelle que soit la séquence vidéo d'entrée, nous pouvons noter que la méthode (P1) (soustraction de l'image moyenne de chaque image avant le processus de codage) donne les meilleurs résultats. La plus grande différence dans la qualité de reconstruction obtenue entre le schéma originel SoftCast et les méthodes de prétraitement apparaît pour la séquence *Australia*. En effet, *Australia* contient peu d'activité spatio-temporelle facilitant le processus de décorrélation. La plus grande partie de l'énergie après 3D-DCT est ainsi concentrée dans les basses fréquences et la protection de presque toute l'information contenue dans les coefficients DC-2D entraîne une amélioration supérieure à 2dB. En revanche, *News* et *Stefan* ont un contenu spatiotemporel plus élevé, c'est-à-dire que l'énergie est répartie sur un plus grand nombre de coefficients fréquentiels, ce qui réduit l'écart avec la version originelle.

Nous pouvons également noter que l'écart entre les deux méthodes est plus grand pour la séquence *News* et presque nul pour la séquence *Stefan*. Cela est dû au fait que la valeur moyenne de chaque image est globalement égale à 132 (valeur proche de 128) pour la séquence *Stefan* alors qu'elle n'est que de 78 pour *News*. Il est normal d'observer de meilleurs résultats avec le bloc de prétraitement, car la soustraction de la valeur moyenne de l'image ou d'un niveau de gris moyen permet de grandement réduire la valeur de la composante continue de chaque plans DCT-2D, comme expliqué ci-dessus. La réduction énergétique des coefficients DC-2D entraîne une réduction de l'*activité des données* H_t (présentée dans le Chapitre 2), ce qui permet une meilleure allocation de puissance qui se traduit par une meilleure qualité vidéo reconstruite selon (2.11).

En fonction des caractéristiques de la vidéo, l'amélioration du PSNR et du SSIM entre les deux méthodes peut être visible ou presque nulle (c'est-à-dire lorsque la valeur moyenne pixel de chaque image est d'environ 128). Comme exemple illustratif, nous donnons une synthèse obtenues pour quelques séquences CIF et HD720p dans le Tableau 4.1. Pour rappel, (P1) et (P2) représentent les méthodes de prétraitement dans le domaine pixel, c'est-à-dire respectivement, la soustraction de la valeur moyenne de l'image et la soustraction de la valeur 128.

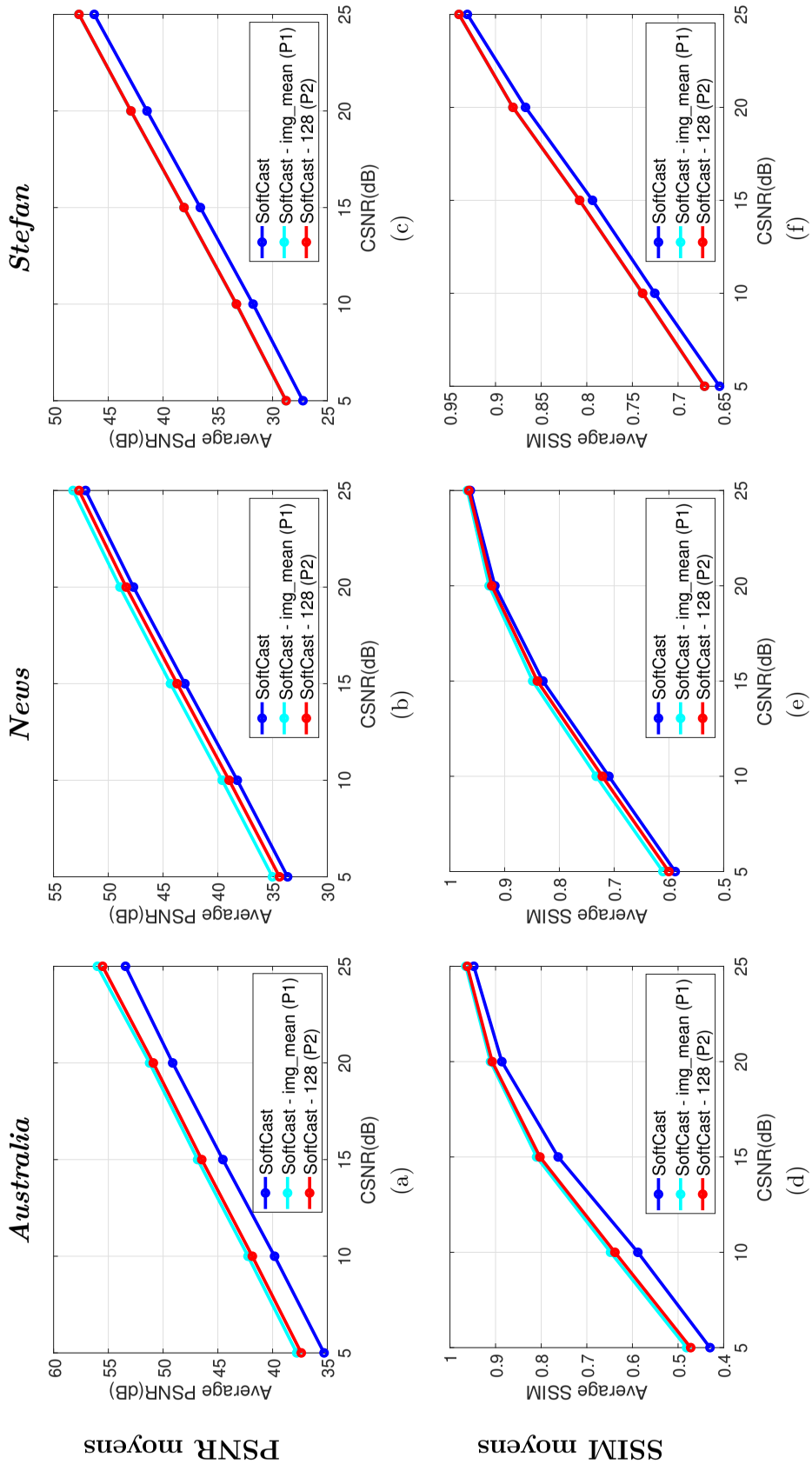


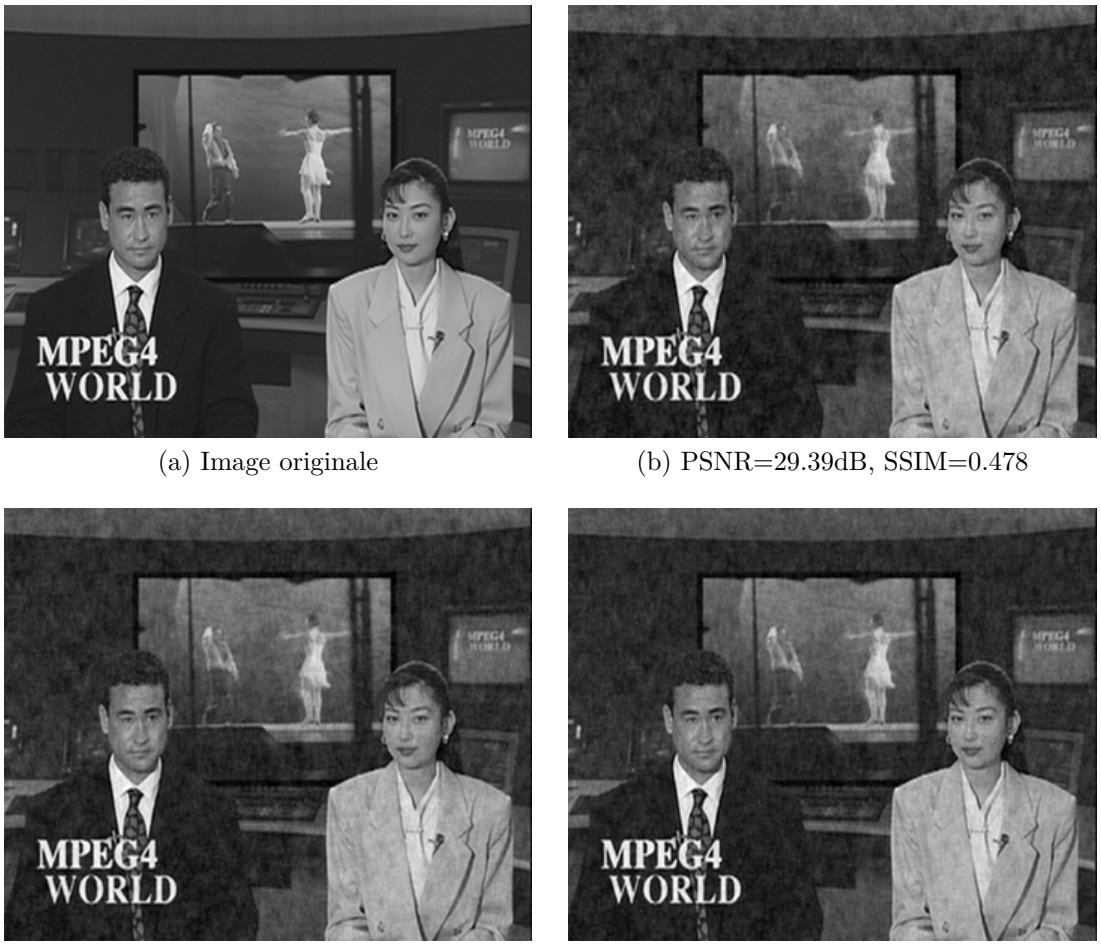
FIGURE 4.3 : Evolution des scores de qualité moyens en fonction du CSNR pour le schéma SoftCast originel et deux méthodes de prétraitement additionnelles : Soustraction du niveau de gris moyen sur 8 bits (128) et soustraction de la valeur moyenne de tous les pixels pour chaque image. Taille de GoP = 16 images, CR = 1. (a), (d) Séquence *Australia*, (b), (e) Séquence *News* et (c), (f) Séquence *Stefan*. (a),(b),(c) : Résultats de PSNR moyens. (d),(e),(f) : Résultats de SSIM moyens.

Tableau 4.1 : Evaluation de l'amélioration maximale de qualité obtenue pour les séquences CIF et HD720p

Amélioration maximale de qualité				
Séquence vidéo	SoftCast vs (P1)	SoftCast vs (P2)	(P1) vs (P2)	Valeur moyenne pixel
Australia(CIF)	PSNR = 2.51dB SSIM = 0.052	PSNR = 2.10dB SSIM = 0.043	PSNR = 0.41dB SSIM = 0.009	Min = 97 Max = 103
City(CIF)	PSNR = 2.40dB SSIM = 0.056	PSNR = 2.12dB SSIM = 0.050	PSNR = 0.28dB SSIM = 0.006	Min = 103 Max = 115
Coastguard(CIF)	PSNR = 1.91dB SSIM = 0.044	PSNR = 1.87dB SSIM = 0.043	PSNR = 0.04dB SSIM = 0.001	Min = 106 Max = 132
Container(CIF)	PSNR = 3.35dB SSIM = 0.062	PSNR = 3.26dB SSIM = 0.060	PSNR = 0.09dB SSIM = 0.002	Min = 136 Max = 142
Crew(CIF)	PSNR = 2.47dB SSIM = 0.062	PSNR = 1.55dB SSIM = 0.040	PSNR = 0.91dB SSIM = 0.022	Min = 81 Max = 122
Foreman(CIF)	PSNR = 2.82dB SSIM = 0.062	PSNR = 2.54dB SSIM = 0.056	PSNR = 0.28dB SSIM = 0.005	Min = 124 Max = 194
News(CIF)	PSNR = 1.4dB SSIM = 0.023	PSNR = 0.73dB SSIM = 0.012	PSNR = 0.67dB SSIM = 0.011	Min = 76 Max = 85
Stefan(CIF)	PSNR = 1.56dB SSIM = 0.017	PSNR = 1.53dB SSIM = 0.016	PSNR = 0.03dB SSIM = 0.001	Min = 132 Max = 144
Soccer(CIF)	PSNR = 2.65dB SSIM = 0.077	PSNR = 2.60dB SSIM = 0.076	PSNR = 0.05dB SSIM = 0.001	Min = 116 Max = 146
Ducks(HD720p)	PSNR = 1.97dB SSIM = 0.035	PSNR = 1.90dB SSIM = 0.034	PSNR = 0.070dB SSIM = 0.001	Min = 105 Max = 120
Into tree(HD720p)	PSNR = 1.58dB SSIM = 0.045	PSNR = 0.78dB SSIM = 0.023	PSNR = 0.80dB SSIM = 0.022	Min = 58 Max = 100
Parkjoy(HD720p)	PSNR = 0.88dB SSIM = 0.020	PSNR = 0.45dB SSIM = 0.011	PSNR = 0.43dB SSIM = 0.009	Min = 53 Max = 90

En ce qui concerne les courbes SSIM, nous observons que l'écart entre les trois schémas évalués diminue lorsque le CSNR devient plus élevé. Cela est dû au fait qu'avec un CSNR élevé (>25dB), la perturbation du bruit AWGN devient négligeable et que, par conséquent, la reconstruction de la composante fréquentielle DC dans le schéma originel SoftCast se rapproche de la valeur DC réelle avant la transmission.

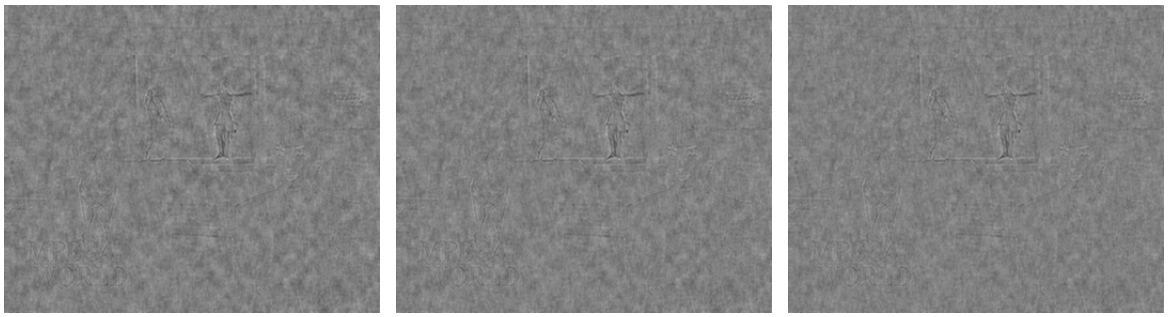
Finalement, un exemple de comparaison visuelle est donné dans les Figs. 4.4 et 4.5 où les images reconstruites et les images d'erreurs sont affichées (les images d'erreurs ont été décalées de +128 à des fins d'affichage). Nous avons délibérément réglé le CSNR à 0 dB afin d'accentuer le bruit pendant la transmission. Nous pouvons clairement observer que la version originelle de SoftCast donne la moins bonne qualité reçue avec un effet de neige particulièrement agressif. En revanche, les méthodes de prétraitement étudiées permettent d'obtenir une amélioration de la qualité sous les mêmes caractéristiques de canal, diminuant ainsi l'effet de neige observé classiquement.



(a) Image originale (b) PSNR=29.39dB, SSIM=0.478

(c) PSNR=30.07dB, SSIM=0.488 (d) PSNR=30.79dB, SSIM=0.499

FIGURE 4.4 : Comparaison visuelle de la qualité reçue à un CSNR égal à 0 dB pour la séquence *News* (première image). (a) Image originale, (b) SoftCast originel, (c) SoftCast avec prétraitement (P2) (soustraction du niveau de gris moyen 128), (d) SoftCast avec prétraitement (P1) (soustraction de la valeur moyenne de chaque image).



(a) PSNR=29.39dB, SSIM=0.478 (b) PSNR=30.07dB, SSIM=0.488 (c) PSNR=30.79dB, SSIM=0.499

FIGURE 4.5 : Comparaison visuelle de la qualité reçue à un CSNR égal à 0 dB pour la séquence *News* (première image). Images d'erreur résultantes de la Fig. 4.4 : (a) SoftCast originel, (b) SoftCast avec prétraitement (P2), (c) SoftCast avec prétraitement (P1).

Nous venons de voir deux méthodes de prétraitement issues de la littérature agissant dans le domaine pixel. Une synthèse de ces deux méthodes est disponible dans le Tableau 4.2.

Tableau 4.2 : Comparaison des différentes méthodes de prétraitement étudiées (séquences issues du Tableau 4.1 utilisées pour les calculs des gains moyens)

Méthode	Description	Avantages	Inconvénients	Gain Moyen
Valeur moyenne de l'image (P1)	Soustraction de la valeur moyenne pixel sur chaque image	Bon compromis entre performance et consommation additionnelle de bande passante	Envoi de 8 bits supplémentaires en métadonnées pour chaque image	PSNR = 2.55dB SSIM = 0.055
Niveau de gris moyen 8-bit = 128 (P2)	Soustraction d'un niveau de gris moyen = 128 sur chaque image	Pas de consommation additionnelle de bande passante	Les performances dépendent des caractéristiques de la vidéo	PSNR = 2.14dB SSIM = 0.046

Comme nous pouvons le voir, les méthodes de prétraitement dans le domaine pixel permettent d'obtenir un gain en termes de qualité reçue par rapport à la version originelle de SoftCast d'environ 2.55dB et 2.14dB respectivement pour les méthodes (P1) et (P2). Nous allons voir par la suite que ce gain peut être encore amélioré avec l'utilisation d'un prétraitement fréquentiel (au lieu d'un prétraitement pixel) au prix d'une augmentation des métadonnées additionnelles.

4.3.2 Prétraitement fréquentiel

4.3.2.1 Analyse de l'algorithme OPA-SoftCast

Comme illustré dans la Fig. 4.6, l'algorithme OPA-SoftCast permet d'obtenir des gains moyens supérieurs (2.8dB en moyenne) par rapport aux méthodes de prétraitement agissant dans le domaine pixel. Cependant, comme nous l'avons vu dans la Section 4.2.2 précédente, cette amélioration en termes de gain se fait au prix d'une augmentation des métadonnées additionnelles. Nous proposons donc ici d'étudier OPA-SoftCast afin de proposer une méthode alternative et originale permettant d'offrir une augmentation de qualité similaire tout en réduisant d'une part les métadonnées additionnelles à transmettre et d'autre part, le temps de calcul nécessaire à l'opération de prétraitement. Nous notons que seule la version 3D d'OPA-SoftCast est ici considérée. En effet, les versions 2D mentionnées dans la Fig. 4.6 correspondent aux mêmes schémas de transmission mais considérant une transmission d'image au lieu d'une transmission vidéo (la dimension temporelle est ainsi supprimée).

Nous analysons tout d'abord la position des chunks sélectionnés par l'algorithme OPA ainsi que la position des SFC au sein de ceux-ci en se basant sur l'environnement de simulation [40] : Le processus est similaire à la section précédente cependant, cette fois-ci une taille de GoP de 8 images est utilisée (comme utilisé dans [40]). Le nombre de SFC N_d est fixé à

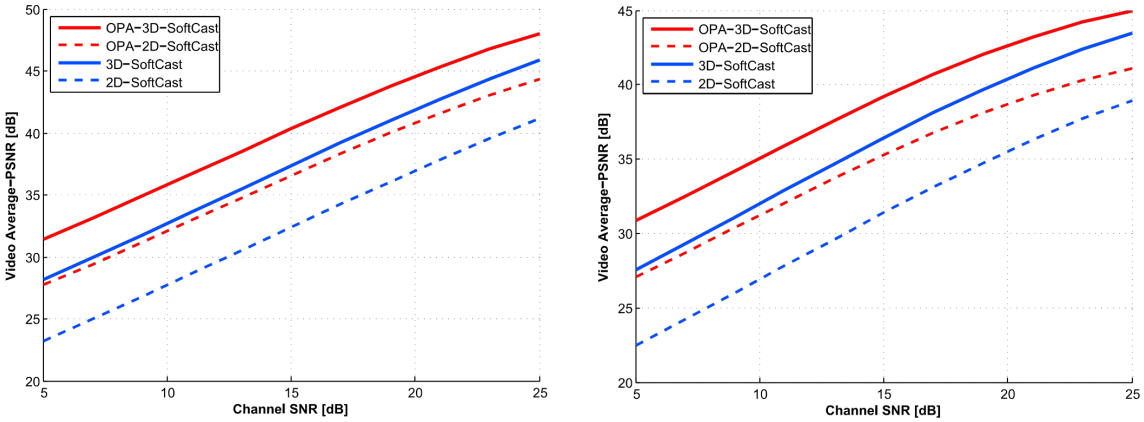


FIGURE 4.6 : Evolution du PSNR moyen en fonction du CSNR pour le schéma SoftCast et OPA-SoftCast. Gauche : La bande passante est fixée à 1.14 MHz (CR=0.75). Droite : La bande passante est fixée à 0.91 MHz (CR=0.6). Moyenne des PSNR obtenus pour les séquences : *Foreman, Akiyo, Coastguard, Flower, Paris* et *Bus*. Figure issue de [40].

16. Nous notons que des résultats similaires ont été obtenus pour la taille de GoP = 16 images (avec $N_d = 32$). La séquence vidéo utilisée par He *et al.* consiste à concaténer les 32 premières images des séquences *Foreman, Akiyo, Coastguard , Flower, Paris* et *Bus* dans une séquence composite dénotée par $Mixed_{CIF\ (He\ and\ al.)}$ ci-après. Pour une comparaison et une analyse plus détaillée, nous ajoutons également la séquence composite $Mixed_{HD}$ définie dans le Chapitre 2 et regroupant pour rappel les 128 premières images de *Ducks, Four People, Into Tree, Johnny, Kristen and Sara, Old Town, Parkjoy, Shields, Parkrun* et *Stockholm*. Ces vidéos ont été choisies afin d'étudier différents contenus présentant des disparités spatio-temporelles importantes (voir Fig. 4.2).

Comme indiqué dans les Figs. 4.7 et 4.8, les SFC sélectionnés représentés par des petits carrés blancs sont situés de manière logique dans le premier chunk (grand rectangle noir) contenant les basses fréquences. Pour la plupart des séquences vidéo, ces SFC sont situés dans le coin supérieur gauche du premier plan fréquentiel du GoP. Sur la base de simulations approfondies sur chaque séquence vidéo de la Fig. 4.2, nous voyons que, pour presque toutes les séquences, la position du chunk sélectionné ne change pas si moins de 80 SFC par GoP sont sélectionnés. En effet, après 3D-DCT, l'énergie est principalement localisée sur les coefficients basse fréquence (c'est-à-dire le coin supérieur gauche du premier plan fréquentiel). De plus, He *et al.* [40] ont montré que le gain en termes de PSNR ralentissait après avoir sélectionné un SFC par image (illustré dans la Fig. 4.9).

En outre, la transmission de plus de 80 SFC est préjudiciable en raison de l'augmentation importante du nombre de métadonnées. Finalement, nous pouvons donc noter qu'il n'est pas nécessaire de transmettre la position du chunk sélectionné, car c'est généralement le premier. Quelques exceptions ont toutefois été relevées pour les séquences ayant des index TI très

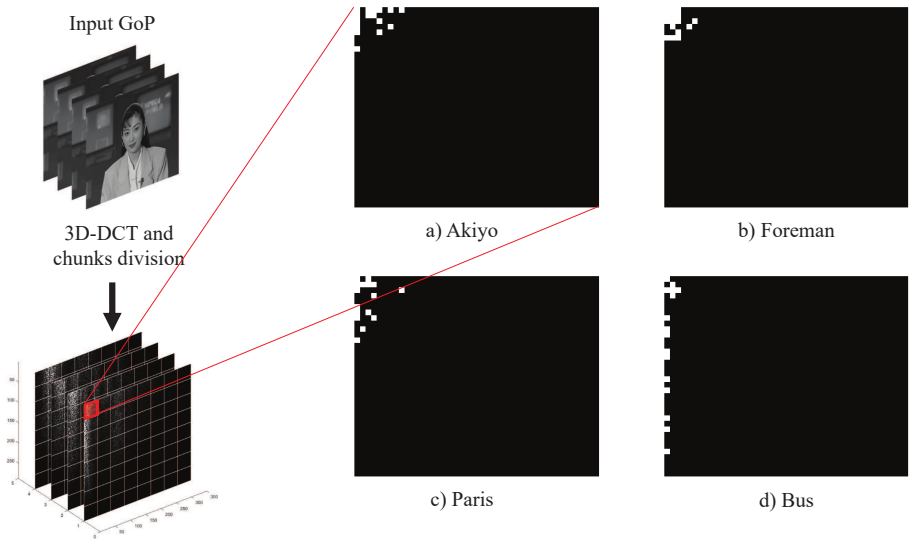


FIGURE 4.7 : Représentation visuelle de la localisation des 16 SFC (petits carrés blancs) sélectionnés par OPA-SoftCast dans le chunk supérieur gauche (coefficients 44×36) du premier plan DCT-3D pour les séquences vidéo CIF sélectionnées.

(a) Ducks (b) Four People (c) Into Tree (d) Johnny (e) Kristen And Sara

(f) Old Town (g) ParkJoy (h) Shields (i) Parkrun (j) Stockholm

FIGURE 4.8 : Représentation visuelle de la localisation des 16 SFC (petits carrés blancs) sélectionnés par OPA-SoftCast dans le chunk supérieur gauche (taille 160×90) du premier plan DCT-3D pour les séquences vidéo HD720p sélectionnées. Seule la partie supérieure gauche (40×22) de chaque chunk est affichée pour faciliter la visualisation.

élevés, comme par exemple, *Soccer* ou *ParkJoy*, où quelques SFC ont été sélectionnés dans le chunk en haut à gauche du deuxième plan fréquentiel à l'issue de la DCT-3D. Cependant, nous montrerons par la suite que les performances restent similaires avec la méthode proposée.

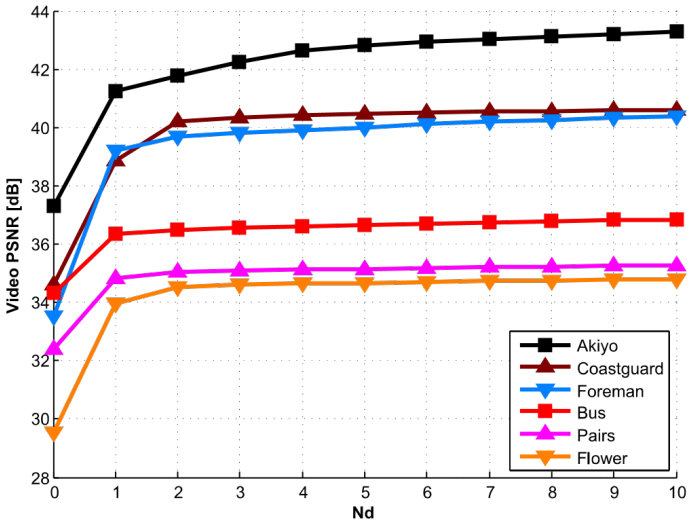


FIGURE 4.9 : Comparaison de l'amélioration de qualité reçue en termes de PSNR pour différents N_d avec un CSNR de 15dB. Figure issue de [40].

De plus, nous pouvons voir que la localisation de la plupart des SFC sélectionnés par OPA-SoftCast dans les Figs. 4.7 et 4.8 peut être approximée par un balayage en zigzag (à partir de la valeur DC dans le coin supérieur gauche) comme indiqué dans la Fig. 4.10.

Par conséquent, nous proposons d'utiliser une solution alternative à faible coût de calcul qui ne nécessite pas de processus itératif pour choisir les coefficients fréquentiels envoyés en métadonnées. Cette solution, dénommée OPA2-SoftCast (P4), sélectionne en effet toujours le premier chunk représentant les basses fréquences issues de la DCT-3D et utilise un balayage en zigzag pour sélectionner N_d coefficients fréquentiels (le même nombre qu'OPA-SoftCast). Comme la méthode proposée n'est pas un processus itératif, elle permet de s'affranchir de la transmission de la moyenne ajustée. En effet, celle-ci est calculée une fois comme dans le schéma SoftCast originel et envoyée sous forme de métadonnées classiques. De plus, étant donné que les positions du chunk et des coefficients fréquentiels sélectionnés sont toujours les mêmes dans notre approche, la méthode proposée réduit les métadonnées additionnelles de 4 valeurs par coefficient fréquentiel à une seule, soit une économie de 75 % de la bande passante additionnelle requise par rapport à OPA-SoftCast (réduction de 9600Hz à 2400Hz obtenue avec les équations (4.3) et (4.4) lorsque $N_d = 16$, $F_r = 30$ et $N_F = 8$). Même si le placement des SFC sélectionnés ne suit pas toujours un balayage en zigzag (par exemple, *Shields*) et / ou le fait que ces SFC ne soient pas totalement situés dans le premier chunk (par exemple, *ParkJoy*), nous montrons dans la section suivante que les performances de la méthode proposée restent compétitives.

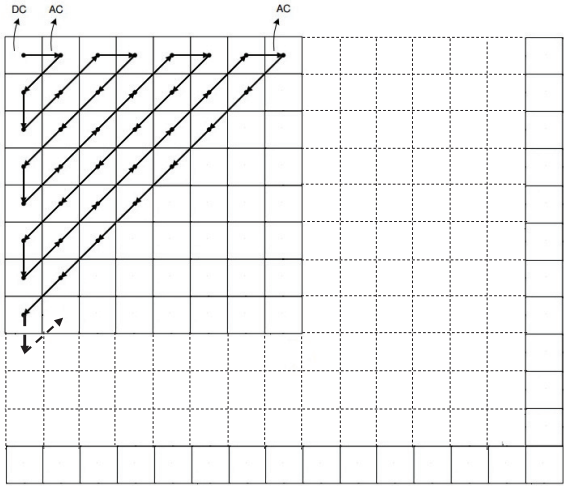


FIGURE 4.10 : Représentation visuelle du balayage en zigzag au sein d'un chunk (taille variable du chunk selon le format vidéo indiquée en pointillés).

4.4 Analyse des performances d'OPA2-SoftCast

La méthode proposée (OPA2) est évaluée par rapport à l'algorithme OPA-SoftCast (OPA) et le schéma d'origine SoftCast (SC) au moyen de simulations approfondies décrites ci-dessous.

Caractéristiques du canal : Les caractéristiques des canaux simulés restent les mêmes que dans la Section 4.3.1.1. Nous avons choisi d'afficher les résultats pour les niveaux de compression CR=1 et 0.25, ce qui correspond respectivement à aucune compression appliquée, et 75% des coefficients jetés. Les résultats pour les autres cas sont similaires puisque le premier chunk est toujours envoyé quelle que soit la bande passante disponible du côté de l'émetteur.

Les simulations sont effectuées avec les séquences HD720p sélectionnées (1280×720 pixels, 30 fps) et CIF (352 ×288 pixels, 30 fps) indiquées dans la Fig. 4.2. Les séquences vidéo $Mixed_{CIF\ (He\ and\ al.)}$ et $Mixed_{HD}$ décrites dans la Section 4.3.2 sont également utilisées dans cette section pour évaluer la méthode proposée.

Tableau 4.3 : *Activité des données* H_t exprimée en dB pour les schémas : SoftCast originel, OPA-SoftCast et la méthode proposée (OPA2). CR=1.

Séquences	H_t **(SC)**	H_t **(OPA)**	H_t **(OPA2)**	**Perte PSNR**
Soccer	23.88dB	20.11dB	20.27dB	0.16dB
Into Tree	21.52dB	17.94dB	18.10dB	0.14dB
ParkJoy	26.53dB	25.42dB	25.47dB	0.05dB
Shields	23.58dB	21.99dB	22.18dB	0.19dB

En tant que premier indicateur de performance, nous donnons les valeurs d'activité H_t (exprimées en dB) obtenues pour le schéma originel SoftCast et les deux méthodes de pré-

traitement OPA. Pour rappel, l'activité est obtenue à partir de l'équation (2.11). Le CR est ici fixé à 1. Des résultats similaires pour des valeurs de CR plus faibles ont été observés. Nous choisissons de montrer les valeurs d'activités pour les séquences non conformes aux hypothèses clés de notre méthode (le fait que les coefficients fréquentiels sélectionnés se trouvent dans le premier chunk et/ou le fait que ceux-ci suivent un balayage en zigzag). Les résultats donnés dans le Tableau 4.3 montrent que, même si ces vidéos ne suivent pas les règles énoncées ci-dessus, la réduction de l'activité reste proche des performances optimales données par l'algorithme OPA-SoftCast (rappelant qu'une activité inférieure implique une meilleure qualité reçue). Cela est dû au fait que même si les coefficients fréquentiels sélectionnées avec la méthode proposée ne sont pas les coefficients fréquentiels spéciaux (SFC), i.e., présentant l'énergie la plus élevée, il s'agit toujours de coefficients DCT hautement énergétiques puisqu'ils se situent dans les basses fréquences issues de la DCT-3D. En conséquence, la qualité reçue reste presque identique comme illustré dans la Fig. 4.11. Comme expliqué dans le Chapitre 2, nous notons une différence entre le PSNR théorique donné par l'équation (2.11) et celui affiché à la Fig. 4.11 en raison du fait que l'équation (2) considère un décodeur ZF au lieu d'un décodeur LLSE tel qu'utilisé dans le Schéma SoftCast. Même s'il existe un petit biais d'environ 1-3 dB (selon le type de séquence évalué) en termes de PSNR à très faible CSNR (ici 0 dB), l'activité des données H_t représente un moyen rapide d'évaluer l'apport des méthodes de prétraitement par rapport au schéma originel.

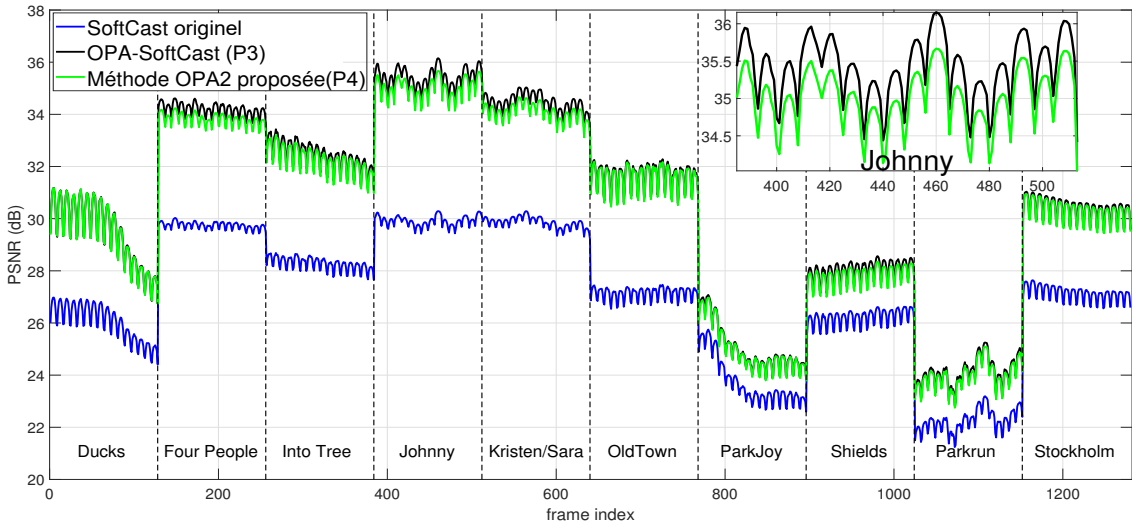


FIGURE 4.11 : Evolution du PSNR image par image pour la séquence $Mixed_{HD}$, CSNR=0dB, CR=1 (pas de compression appliquée).

La perte de PSNR moyenne entre la méthode proposée et la méthode originelle OPA-SoftCast pour ces séquences est d'environ 0.2 dB, ce qui est insignifiant. Pour vérifier cela, nous donnons deux exemples de comparaison visuelle dans les Figs. 4.12 et 4.14, le CSNR est réglé à 0 dB pour accentuer le bruit pendant la transmission.

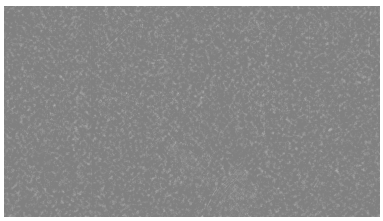
(a) Image originale

(b) PSNR=30.44dB, SSIM=0.686

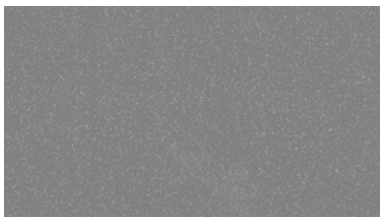
(c) PSNR=31.54dB, SSIM=0.728

(d) PSNR=31.44dB, SSIM=0.724

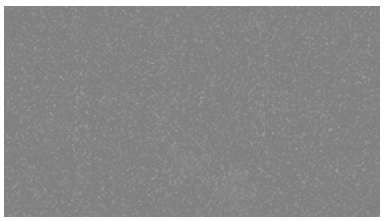
FIGURE 4.12 : Comparaison visuelle de la qualité obtenue pour la séquence *Shields* (première image) avec un CSNR=0dB et CR=1. De gauche à droite, de haut en bas : (a) Image originale, (b) SoftCast originel, (c) OPA-SoftCast, (d) La méthode proposée (OPA2-SoftCast).



(a) PSNR=30.44dB, SSIM=0.686



(b) PSNR=31.54dB, SSIM=0.728



(c) PSNR=31.44dB, SSIM=0.724

FIGURE 4.13 : Comparaison visuelle de la qualité reçue avec un CSNR=0dB et CR=1 pour la séquence *Shields* (première image). Images d'erreur résultantes de la Fig. 4.12 : (a) SoftCast originel, (b) OPA-SoftCast, (c) La méthode proposée (OPA2-SoftCast).

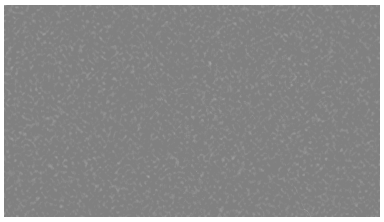
(a) Image originale

(b) PSNR=30.01dB, SSIM=0.334

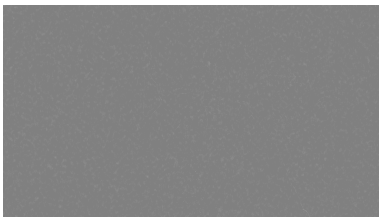
(c) PSNR=35.43dB, SSIM=0.445

(d) PSNR=35.01dB, SSIM=0.436

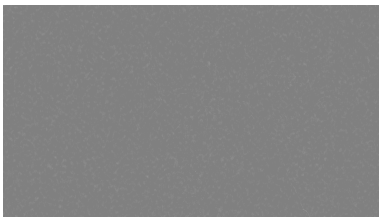
FIGURE 4.14 : Comparaison visuelle de la qualité obtenue pour la séquence *Johnny* (première image) avec un CSNR=0dB et CR=1. De gauche à droite, de haut en bas : (a) Image originale, (b) SoftCast originel, (c) OPA-SoftCast, (d) La méthode proposée (OPA2-SoftCast).



(a) PSNR=30.01dB, SSIM=0.334



(b) PSNR=35.43dB, SSIM=0.0.445



(c) PSNR=35.01dB, SSIM=0.436

FIGURE 4.15 : Comparaison visuelle de la qualité reçue avec un CSNR=0dB et CR=1. pour la séquence *Johnny* (première image). Images d'erreur résultantes de la Fig. 4.14 : (a) SoftCast originel, (b) OPA-SoftCast, (c) La méthode proposée (OPA2-SoftCast).

Le premier cas correspond à un cas où les SFC sélectionnés par OPA-SoftCast ne suivent pas un balayage en zigzag, c'est-à-dire la séquence *Shields*, comme indiqué dans la Fig. 4.8. Nous pouvons clairement observer que le schéma originel SoftCast donne la plus basse qualité vidéo reçue. En revanche, les méthodes de prétraitement permettent d'obtenir une meilleure qualité reconstruite dans le même canal. La perte entre les deux méthodes est imperceptible comme observé dans les images d'erreurs de la Fig. 4.13. Celles-ci sont inférieures à $\leq$ 0.1dB même si les coefficients sélectionnés par OPA-SoftCast ne suivent pas un balayage en zigzag.

Le deuxième cas correspond à la première image de la séquence *Johnny*, là où les pertes sont les plus importantes. Comme illustré dans les Fig. 4.14 et 4.15, la perte maximale observée est inférieure à 0.5dB. Celle-ci est considérée comme visuellement insignifiante [83].

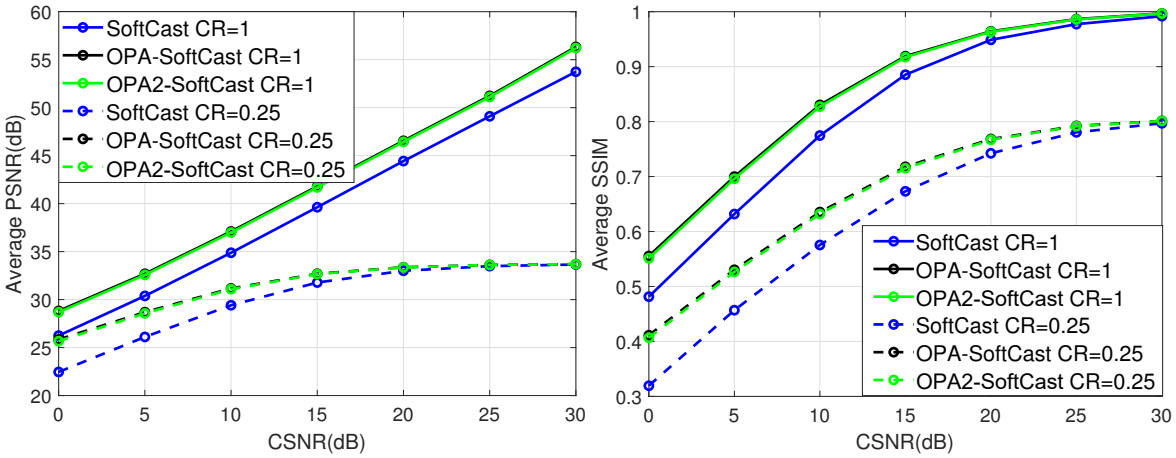
La méthode proposée augmente la qualité reçue entre 1.3 dB (séquence *ParkJoy*) et jusqu'à 5 dB (séquence *Johnny*), comme indiqué dans la Fig. 4.11. La différence entre les gains obtenus est due aux caractéristiques des vidéos. En effet, la séquence *Johnny* qui a des valeurs de SI, TI faibles est facile à décorréler et la majeure partie de l'énergie est donc concentrée sur les coefficients basses fréquences après la DCT-3D. La protection des coefficients basse fréquence les plus importants permet d'améliorer considérablement la qualité reçue. Au contraire, en raison de forts mouvements et textures dans la séquence *ParkJoy*, le signal est difficile à décorréler et l'énergie est plus répartie dans l'ensemble du GoP. En conséquence, la qualité vidéo reçue n'est pas aussi bonne que celle de la séquence *Johnny*. Nous avons vérifié que des conclusions similaires sont obtenues avec les séquences CIF.

Tableau 4.4 : Comparaison entre la méthode OPA-SoftCast et celle proposée (séquence $Mixed_{HD}$)

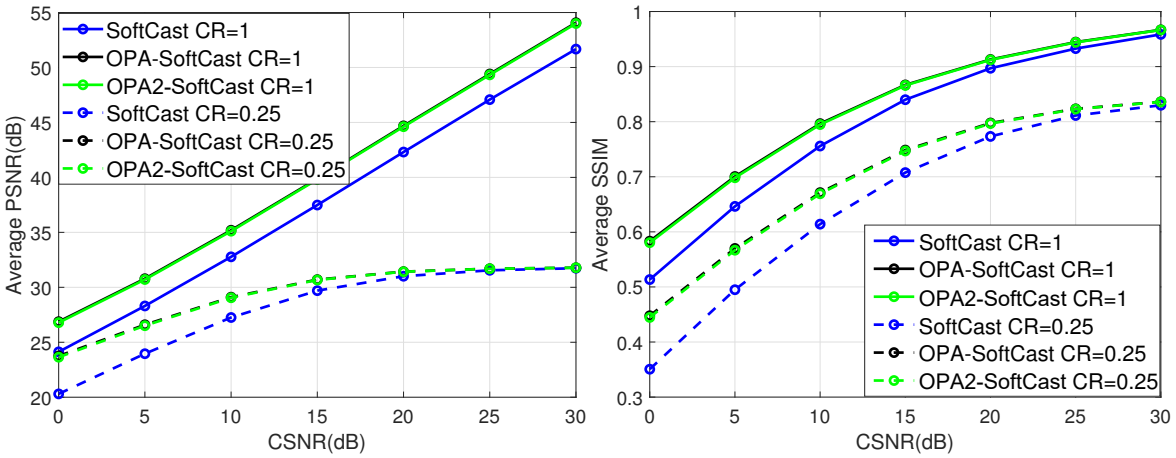
Méthode	**Gain moyen**	**Tps calcul**	**BP additionnelle**
OPA-SoftCast[40] (P3)	**PSNR = 3.12dB SSIM = 0.049**	12.43s	9600Hz
Méthode OPA2 proposée (P4)	PSNR = 2.93dB SSIM = 0.047	**5.05s**	**2400Hz**

Le temps de calcul et la bande passante supplémentaire requise pour exécuter les méthodes de prétraitement OPA sont également indiqués dans le Tableau 4.4. Le temps de calcul est défini comme étant le temps total nécessaire pour effectuer le prétraitement au niveau de l'émetteur et l'opération inverse au récepteur sur la séquence complète $Mixed_{HD}$ composée de 1280 images. Les différents temps ont été obtenus avec Matlab R2018b sur un ordinateur équipé d'un processeur Intel Core (TM) i7-4510U, 2GHz, 12G de RAM. Comme on peut le constater, la méthode proposée divise le temps de calcul par 2.5 par rapport à la méthode OPA-SoftCast. Cela est dû au fait que la méthode proposée n'est pas un processus itératif. De plus, étant donné que seule la valeur de chaque coefficient fréquentiel sélectionné est envoyée en tant que métadonnées supplémentaires, la largeur de bande additionnelle requise pour la

méthode proposée est réduite de 75% par rapport à celle utilisée pour OPA-SoftCast. Nous notons que des résultats similaires ont été obtenus pour la séquence $Mixed_{CIF\ (He\ and\ al.)}$.



(a) Séquence $Mixed_{HD}$



(b) Séquence $Mixed_{CIF\ (He\ and\ al.)}$

FIGURE 4.16 : Evolution des scores de qualité moyens obtenus en fonction du CSNR. Première colonne : PSNR. Deuxième colonne : SSIM. CR=1 (trait plein), CR=0.25 (trait tireté).

Enfin, nous montrons dans la Fig. 4.16, les scores moyens de qualité pour les séquences $Mixed_{HD}$ et $Mixed_{CIF\ (He\ and\ al.)}$. Quelle que soit la bande passante totale disponible pour la transmission, les méthodes de prétraitement apportent des améliorations significatives à la qualité reçue par rapport au schéma originel SoftCast. Le gain moyen est d'environ 3.12 dB pour le PSNR et de 0.049 pour l'index SSIM, respectivement. La perte de qualité entre la méthode proposée et OPA-SoftCast n'est que d'environ 0.19 dB pour le PSNR et de 0.002 pour l'index SSIM en moyenne, alors que la bande passante supplémentaire nécessaire pour les métadonnées est réduite de 75 % et que le temps de calcul est divisé par 2.5. La perte de qualité est due au fait que la méthode proposée ne garantit pas de toujours sélectionner

les coefficients fréquentiels les plus énergétiques par rapport à OPA-SoftCast. Cependant, comme mentionné ci-dessus, la majeure partie de l'énergie est localisée sur des coefficients basse fréquence et comme vérifié ci-dessus, même lorsque les fréquences sélectionnées ne sont pas bien décrites par un balayage en zigzag (voir Fig. 4.7 et 4.8), la perte de PSNR reste inférieure à 0.5 dB (voir la Fig. 4.11) montrant l'efficacité de la version proposée.

Nous notons que dans leur forme la plus simple (i.e., $N_d = 1$), les versions OPA et OPA2 sont équivalentes en termes de qualité reconstruite puisque seule la valeur de la composante DC-3D est retirée. En effet, nous avons vérifié que la première SFC correspond toujours à la valeur de la composante DC. Ceci est logique puisqu'il est bien connu que la valeur de la composante DC, après une DCT, contient la plupart de l'énergie. Cette solution est dénotée par la suite DC-3D SoftCast ou méthode (P5). Elle représente le cas le plus défavorable de l'algorithme OPA et OPA2, où un seul SFC est choisi dans le GoP, le coefficient DC (après DCT-3D). L'intérêt de cette version simplifiée réside dans le fait que grâce à la position connue de ce coefficient DC (i.e., la position (0,0,0) dans le plan 3D), les métadonnées additionnelles sont réduites de 160 à 40 bits (20 bits avant insertion du code FEC). Ceci entraîne une réduction de la bande passante additionnelle pour ces métadonnées de 9600Hz à 150Hz. Pour rappel, pour la méthode (P1), la moyenne de chaque image est codée sur 8 bits, par conséquent la consommation additionnelle de bande passante est de 480Hz ($\frac{8 \cdot 2 \cdot N_F \cdot F_r}{N_F}$, avec $F_r = 30$ et $N_F = 8$ dans cette configuration).

Nous montrons dans la section suivante que la méthode (P5), bien que sous optimale à la version OPA (P3), représente un bon compromis entre qualité de reconstruction, métadonnées additionnelles et temps d'exécution dans la section suivante.

4.5 Evaluation globale des méthodes et récapitulatif

Un récapitulatif des gains moyens, avantages et inconvénients de chaque méthode par rapport au schéma d'origine SoftCast est tout d'abord donné dans le Tableau 4.5. Ce dernier illustre bien l'intérêt du prétraitement dans un contexte de transmission basé sur un CVL.

En effet, un gain moyen de 2.3dB est observé pour la séquence $Mixed_{CIF\ (He\ and\ al.)}$ pour les méthodes (P1) et (P5). Des gains similaires sont obtenus pour ces deux méthodes, puisque le fait de soustraire la valeur moyenne de chaque image revient en effet à réduire la valeur des coefficients DC-2D de chaque plan DCT-2D [98]. Le coefficient DC-3D dans le domaine DCT-3D n'est autre qu'une moyenne des coefficients DC-2D issues du domaine DCT-2D spatial [98].

Ainsi, si les valeurs de ces coefficients au sein d'un même GoP sont très différentes (un changement de plan par exemple), les solutions proposée OPA2 et SoftCast DC-3D deviennent

Tableau 4.5 : Comparaison des différentes méthodes de prétraitement étudiées (séquence de référence : $Mixed_{CIF\ (He\ and\ al.)}$, plage de CSNR=[0∼25dB].)

Méthode	Description	Avantages	Inconvénients	Gain Moyen	Temps d'exécution	BP requise
Valeur moyenne de l'image (P1)	Soustraction et envoi de la valeur moyenne (pixel) sur chaque image	Bon compromis entre performance et consommation additionnelle de bande passante	Envoi de 8 bits supplémentaires en métadonnées pour chaque image	PSNR = 2.33dB SSIM = 0.032	379.9ms	480Hz
Niveau de gris moyen 8-bit = 128 (P2)	Soustraction d'un niveau de gris moyen = 128 sur chaque image	Pas de consommation additionnelle de bande passante	Les performances dépendent des caractéristiques de la vidéo	PSNR = 1.93dB SSIM = 0.026	308.7ms	**0Hz**
OPA-SoftCast [40] (P3)	Sélection de certaines fréquences (SFC) envoyées en métadonnée et mises à zéro ensuite	Meilleure Amélioration de la qualité	Grande consommation de métadonnées additionnelles (80 bits par SFC avant FEC) et complexité plus importante	**PSNR = 3.11dB SSIM = 0.041**	293.6ms	9600Hz
Méthode OPA2 proposée (P4)	Envoi de N_d coefficients en méta-donnée (après DCT-3D) mis à zéro ensuite	Meilleur compromis entre performance et consommation additionnelle de bande passante	Les performances dépendent de l'homogénéité de la valeur moyenne pixel dans le GoP	PSNR = 2.97dB SSIM = 0.040	**121.2ms**	2400Hz
SoftCast DC-3D (P5)	Envoi de la DC en méta-donnée (après DCT-3D) mise à zéro ensuite	Temps d'exécution le plus court et consommation additionnelle de bande passante réduite (20 bits par GoP)	Les performances dépendent de l'homogénéité de la valeur moyenne pixel dans le GoP	PSNR = 2.33dB SSIM = 0.032	**0.955ms**	**150Hz**

sous-optimales puisque l'énergie n'est pas totalement compactée sur la coefficient fréquentiel DC-3D. Toutefois, l'insertion d'une méthode de détection des changements de scènes/plans liée à un encodage adaptatif permet de s'affranchir de ce problème. Cette méthode de détection des changements de scène sera présentée dans le chapitre suivant.

Les solutions proposées peuvent donc être utilisées permettant ainsi une réduction des métadonnées additionnelles de $160N_d$ bits par image (P3), soit un total de $160 \cdot N_d \cdot N_F = 2560$ bits pour un GoP de 8 images et 2 SFC par image, à $40 \cdot N_d \cdot N_F = 640$ bits en considérant OPA2 (P4) et à $20 \cdot 2 = 40$ bits par GoP avec la méthode (P5) (avec application du code FEC). De plus, comme indiqué dans le Tableau 4.5, le temps d'exécution total de la méthode (P4) sur la séquence considérée est insignifiant par rapport à (P1). En effet, notre méthode n'est pas itérative et ne nécessite qu'une seule opération par GoP contrairement aux autres qui nécessitent de nombreuses opérations (une par pixel pour les méthodes (P1), (P2)) ou sont itératives (P3).

Finalement, une comparaison visuelle est donnée dans les Figs. 4.17 et 4.18. Le canal le plus défavorable est choisi (CSNR=0dB) afin d'accentuer le bruit de transmission. L'image où la différence de gain entre la méthode OPA-SoftCast (P3) et la méthode proposée (P4) est la plus importante est choisie (première image de *Foreman*). Nous pouvons observer une amélioration maximale du PSNR et du SSIM respectivement de 4.85dB et 0.11 entre la version SoftCast d'origine et la méthode P3.

Au prix d'un écart maximal et moyen de PSNR entre SoftCast DC-3D (P5) et la méthode (P3) de 1.1dB et 0.47dB respectivement, la bande passante additionnelle de métadonnées

passe de 9600 à 150 Hz et le temps d'exécution de 380 à 1 milliseconde(s). De même, au prix d'un écart maximal et moyen de PSNR entre OPA2 SoftCast (P4) et la méthode (P3) de 0.5dB et 0.14dB respectivement, la bande passante additionnelle de métadonnées passe de 9600 à 2400 Hz et le temps d'exécution de 380 à 122 milliseconde(s). Le choix du prétraitement se fera donc en fonction de la bande disponible ainsi que des contraintes de temps liées à l'application visée. Comme dans [98], nous avons également vérifié l'évolution du PSNR en fonction du CSNR qui reste quasi-linéaire [54, 27] qu'importe le prétraitement choisi.

Qu'importe la méthode utilisée, les Figs. 4.17 et 4.18 illustrent bien l'importance des méthodes de prétraitement dans un contexte SoftCast.

(a) Image originale (b) PSNR=25.57dB, SSIM=0.410

(c) PSNR=28.96dB, SSIM=0.489 (d) PSNR=29.4dB, SSIM=0.498

FIGURE 4.17 : Comparaison visuelle de la qualité reçue dans un CSNR à 0dB pour la séquence $Mixed_{CIF\ (He\ and\ al.)}$ (première image), CR=1. (a) Image originale, (b) SoftCast originel, (c) SoftCast (P2), (d) SoftCast (P1).

(a) Image originale

(b) PSNR=30.42dB, SSIM=0.524

(c) PSNR=30.1dB, SSIM=0.516

(d) PSNR=29.4dB, SSIM=0.498

FIGURE 4.18 : Comparaison visuelle de la qualité reçue dans un CSNR à 0dB pour la séquence $Mixed_{CIF\ (He\ and\ al.)}$ (première image), CR=1. (a) Image originale, (b) OPA-SoftCast (P3), (c) La méthode proposée : OPA2 (P4), (d) SoftCast DC-3D (P5).

4.6 Conclusion

Dans ce chapitre, nous avons tout d'abord étudié et présenté des méthodes de prétraitement pouvant être utilisées dans un contexte de transmission vidéo via SoftCast. Ces méthodes agissent soit dans le domaine pixel, soit dans le domaine fréquentiel. Elles ont pour but de réduire l'énergie des coefficients transmis en (pseudo)-analogique entraînant une meilleure allocation de puissance et par conséquent, une meilleure protection des coefficients transmis face aux erreurs du canal.

Une méthode alternative et originale a ensuite été introduite. Celle-ci est basée sur une lecture directe des coefficients fréquentiels hautement énergétiques via un balayage en zigzag à l'issue de la DCT-3D de SoftCast.

Par rapport au schéma d'origine, la méthode proposée contribue à améliorer la qualité reçue en termes de PSNR jusqu'à 5 dB et 2.97 dB en moyenne et jusqu'à 0.105 et 0.040 en moyenne en termes de SSIM. Par rapport à la méthode OPA-SoftCast, cette solution proposée réduit de 75% la bande passante nécessaire aux métadonnées additionnelles et divise par 2.5 le temps nécessaire à l'application du prétraitement tout en offrant des performances similaires en termes de qualité reçue. La bande passante économisée peut être utilisée pour transmettre plus de coefficients fréquentiels de manière (pseudo)-analogique. Elle est utile pour les applications à bande passante limitée ou lorsque le matériel (hardware) est limité en puissance de calcul.

Cette méthode alternative dans sa version la plus basique correspond à la transmission unique du coefficient DC-3D issue de la DCT-3D. Elle permet d'obtenir un compromis entre amélioration de la qualité, temps d'exécution et métadonnées additionnelles. En effet, une seule composante fréquentielle est envoyée en métadonnée supplémentaire.

Le choix de la méthode de prétraitement est effectué en fonction de la bande passante disponible. En effet, les différentes méthodes (P1), (P3) et (P4) nécessitent une allocation plus importante de la bande passante aux métadonnées. En revanche, la méthode (P2) bien que sous optimale, ne requière aucune métadonnée additionnelle, ce qui en fait une solution possible dans les environnements à bande passante très limitée. Si l'application permet l'envoi d'un nombre limité de métadonnées, la méthode proposée (P4) donne le meilleur compromis bande passante/amélioration de la qualité/temps d'exécution avec un gain moyen de 2.97dB sur les séquences testées. Enfin, notons que le volume de métadonnées additionnelles est variable et qu'il peut être réduit au maximum, dans sa version la plus basique (i.e., DC-3D-SoftCast), à 20 bits par GoP.

Chapitre 5

Encodage adaptatif basé sur l'information temporelle

Sommaire

5.1 Introduction

Dans les deux chapitres précédents, nous avons montré que certains blocs de la chaîne de traitement introduisaient des artefacts pouvant être gênants pour l'utilisateur. Pour limiter l'un d'entre eux (effet de neige), nous avons tout d'abord présenté dans le Chapitre 4 des méthodes de prétraitement.

Dans ce dernier chapitre, nous proposons un algorithme adaptatif d'encodage de la vidéo permettant de :

1. Réduire les fluctuations temporelles de la qualité vidéo ;
2. Supprimer l'effet fantôme introduit par la compression et analysé dans la Section 3.2.4 du Chapitre 3.
3. Proposer le meilleur compromis entre amélioration de la qualité et complexité calculatoire en fonction du contenu vidéo transmis.

Pour ce faire, l'information temporelle du contenu vidéo est analysée et prise en compte au niveau de l'encodage. Ainsi, deux extensions du schéma SoftCast sont proposées :

- AGCC-SoftCast (Adaptive GoP-size based on Content and Cut detection for SoftCast) : cette solution ajuste localement la taille du GoP en fonction des variations de l'index temporel (TI) du contenu vidéo transmis. Ceci permet de réduire les fluctuations de qualité et d'éviter le phénomène d'effet fantôme lorsque la bande passante est limitée grâce à la détection des changements de scène. L'adaptation locale de la GoP (au sein d'une même scène) permet en outre de proposer le meilleur compromis amélioration de la qualité/complexité.
- AGCut-SoftCast (Adaptive GoP-size based on Cut detection for SoftCast) : cette méthode alternative basée uniquement sur la détection des changements de scènes permet de répondre aux contraintes physiques (buffer, mémoire, etc.) et/ou de latence d'un système, tout en assurant la suppression de l'effet fantôme ainsi que la réduction des fluctuations de qualité.

Une analyse préliminaire basée sur l'étude de la qualité reçue en fonction de la taille de GoP choisie est présentée dans ce qui suit. Nous notons tout d'abord ici que la méthode de prétraitement P1 (soustraction de la valeur moyenne pour chaque image) vue dans le chapitre précédent a été appliquée ici au schéma SoftCast. Cette nouvelle version est implicitement utilisée et remplace le schéma SoftCast originel lors des comparaisons. De même que dans le chapitre précédent, nous notons que les résultats et les conclusions présentés dans ce chapitre sont valables quel que soit le type d'estimateur utilisé (ZF, LLSE). Toutefois, nous choisissons de représenter uniquement les résultats pour l'estimateur LLSE étant donné qu'il représente la version originelle de SoftCast.

5.2 L'algorithme AGCC proposé pour SoftCast

5.2.1 Analyse préliminaire

Dans cette section, nous examinons l'impact du contenu vidéo sur la qualité reçue pour différentes tailles de GoP. Comme vu dans les chapitres précédents, nous utilisons les index Spatial Information (SI) et Temporal Information (TI) proposés par l'Union Internationale des Télécommunications [51] pour évaluer les quantités d'informations spatiales et temporelles d'une séquence vidéo. Nous rappelons toutefois que nous avons choisi de moyenner les résultats SI, TI sur toute la séquence au lieu de prendre la valeur maximale de la définition originelle.

5.2.1.1 Environnement de simulation

Source vidéo : Comme précédemment, deux formats vidéo couramment utilisés (provenant de la collection Xiph [113]) sont ici considérés : HD720p (1280×720 pixels, 60fps) et CIF (352×288 pixels, 30fps). De plus, des séquences utilisées par le comité MPEG pour la standardisation d'HEVC [103] sont également utilisées. Ces vidéos comprennent des séquences issues des classes B (séquences HD1080p, 1920×1080 pixels) et classe C (séquences WVGA, 832×480 pixels). Seule la luminance des vidéos est considérée. Le processus est exécuté GoP par GoP en considérant trois tailles différentes : $8, 16$ et 32 images. Pour chacune de ces trois tailles, la taille est maintenue fixe tout au long de la séquence. Chaque image CIF ou WVGA est divisée en 64 chunks, [27, 96] tandis que chaque image HD est divisée en 256 chunks comme dans [42, 43].

Caractéristiques du canal : Des transmissions en présence de bruit blanc additif gaussien (AWGN) sont considérées dans la plage de CSNR de [0∼25dB]. Dans ces travaux, quatre niveaux de bande passante canal disponible sont considérées et représentées via les CR suivants : CR=1, 0.75, 0.5 et 0.25. Lorsqu'un niveau de compression (CR) est appliqué, nous nous assurons de conserver le même débit symbole pour toutes les méthodes. Par exemple, avec un CR = 0.5, nous conservons l'équivalent de 4 et 8 images pour les tailles de GoP égales à 8 et 16, respectivement.

Métriques d'évaluation : Comme nous l'avons vu précédemment dans le Chapitre 3, l'index SSIM obtient les meilleures corrélations avec les scores subjectifs MOS, il est donc naturellement choisi en plus du PSNR pour illustrer les performances des algorithmes proposés.

Dans cette section, nous avons choisi de présenter les résultats pour les séquences CIF *Akiyo*, *Husky* et *Container* ainsi que les résultats pour les séquences HD720p *Johnny*, *Parkrun* et *Parkjoy* en raison de leurs caractéristiques spatio-temporelles hétérogènes. Comme illustré sur la Fig. 5.1, *Husky*, *Parkjoy* et *Parkrun* contiennent des activités spatiotemporelles globalement élevées. En revanche, *Akiyo*, *Johnny* et *Container* présentent globalement des mouvements lents ainsi qu'une information spatiale respectivement basse à élevée. Nous

avons vérifié que des résultats similaires étaient obtenus pour d'autres séquences. Pour évaluer les fluctuations des index SI et TI au sein de la vidéo, nous montrons également dans la Fig. 5.1 les valeurs minimales et maximales obtenues représentées respectivement par une barre horizontale et verticale.

5.2.1.2 Résultats de simulation

Le Tableau 5.1 présente la qualité reçue pour différents contenus vidéo en considérant trois tailles de GoP différentes, à savoir 8, 16 et 32 images. Les résultats montrent que :

- Une grande taille de GoP permet d'obtenir une meilleure qualité à la réception pour les séquences vidéo à faible activité spatio-temporelle (e.g., *Akiyo*, *Johnny*) ;
- Dans le cas contraire, i.e., pour des séquences vidéo à forte activité spatio-temporelle comme *Husky*, il n'y a aucun intérêt à augmenter la taille du GoP en termes de qualité reconstruite. C'est d'autant plus vrai que la complexité augmente en fonction de $O(Klog(K))$ avec K le nombre d'images dans un GoP [53, 23]. Le maintien d'une petite taille de GoP permet dans ce cas une réduction de la complexité jusqu'à 40%.
- Une taille de GoP optimale permettant de maximiser la qualité reçue ou minimisant le coût de complexité peut être définie pour chaque séquence. Dans ces travaux, nous fixons un seuil informel de PSNR de 0.4dB pour décider de la taille optimale du GoP, le comité MPEG estimant en effet qu'une différence de PSNR de 0.5dB est visuellement perceptible [83]. Les tailles de GoP finalement retenues sont indiquées en gras. D'après ces résultats, on constate que le choix est davantage lié à l'index temporel (TI) qu'à l'index spatial (SI). Effectivement, pour les séquences vidéo *Akiyo* et *Container*, la taille optimale est de 32 images alors que les deux séquences présentent de fortes différences en termes de caractéristiques spatiales.
- De même nous pouvons noter que les valeurs du CSNR et/ou du CR appliqué n'influencent que très peu la sélection de la taille du GoP. Par exemple, le GoP de 32 images représente la meilleure qualité reçue pour la séquence *Johnny*, quelles que soient les valeurs de CSNR et de CR prises en compte.

Une comparaison visuelle donnée en Fig. 5.2 illustre les images reconstruites pour les différentes tailles de GoP sélectionnées dans le canal le plus perturbé, c'est-à-dire correspondant à un CSNR = 0dB. Les faibles valeurs résultantes de PSNR et de SSIM peuvent attirer l'attention, mais nous rappelons ici que dans ce type de canal (CSNR = 0 dB), les standards classiques tels que H.264/AVC ou HEVC offriraient une qualité vraisemblablement inférieure aux séquences reconstruites et présenteraient des gels d'images/écran noir en raison de graves erreurs de décodage. En revanche, SoftCast peut faire face à n'importe quelle qualité de canal, même celles étant très mauvaises (CSNR = 0 dB), en délivrant un signal vidéo de qualité faible mais acceptable [53].

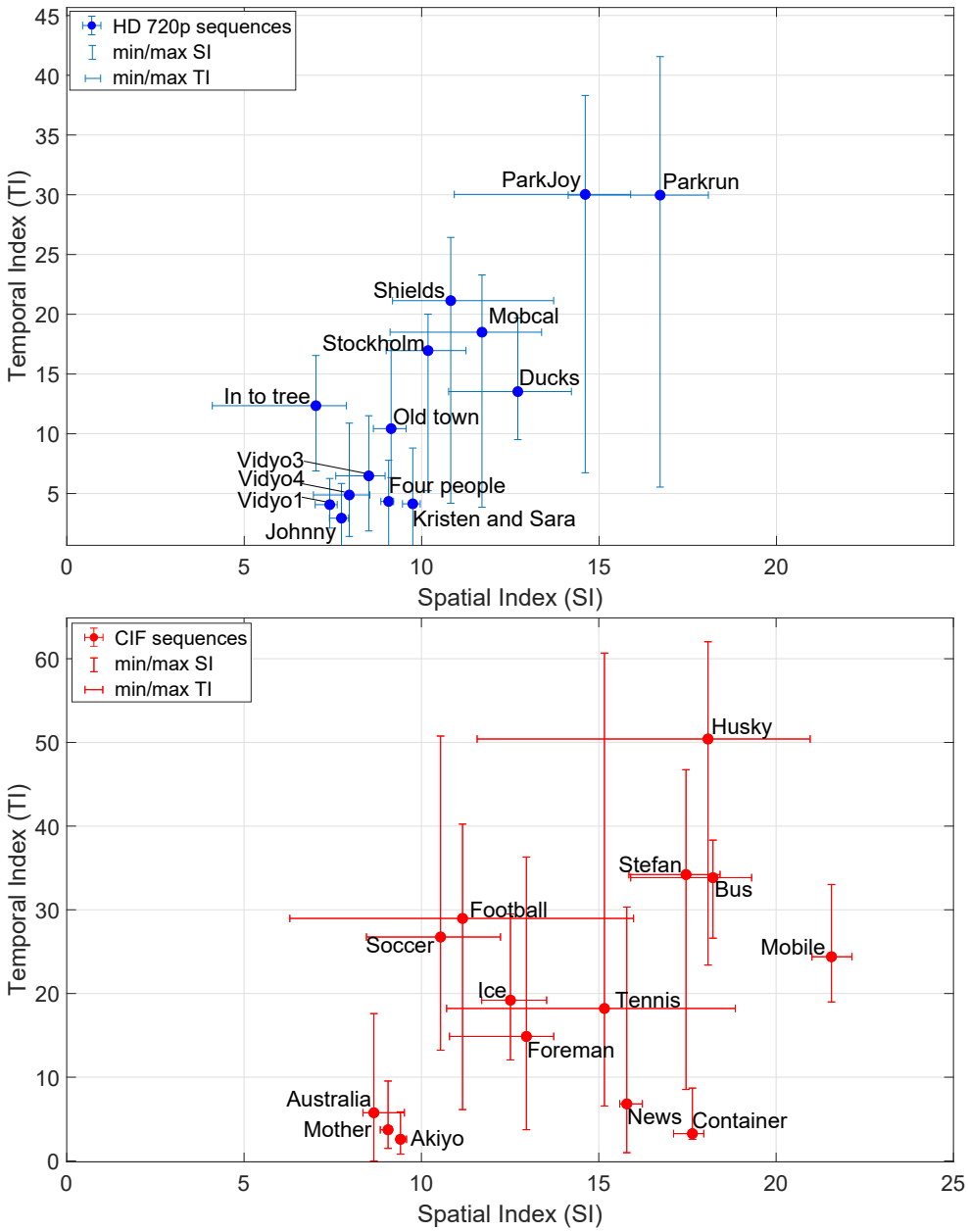


FIGURE 5.1 : Illustration des index spatio-temporels (SI,TI) pour les séquences vidéo HD720p (classe E) et CIF sélectionnées. Les points correspondent aux valeurs moyennes sur toute la séquence vidéo. Les barres verticales et horizontales représentent respectivement la valeur min / max de l'index temporel et de l'index spatial. De haut en bas : séquences HD720p, séquences CIF.

Tableau 5.1 : Tableau des scores PSNR et SSIM obtenus pour différentes tailles de GoP et différents CSNR avec CR = 1 (pas de compression) et CR = 0.25 (75 % des coefficients jetés). Les tailles de GoP retenues sont indiquées en gras.

Environnement de simulation			CSNR(dB) 0		10		20	
			PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
Taille de GoP = 8	CR=1	*Akiyo*	36.27	0.875	45.06	0.977	53.77	0.996
		Container	30.11	0.781	39.17	0.956	48.55	0.994
		Husky	**20.56**	**0.671**	**28.62**	**0.912**	**38.26**	**0.987**
		Johnny	35.87	0.916	44.66	0.984	53.35	0.997
		Parkjoy	**24.71**	**0.672**	**32.83**	**0.903**	**42.41**	**0.986**
		Parkrun	24.53	0.738	32.83	0.935	42.42	0.992
	CR=0.25	*Akiyo*	29.12	0.764	37.46	0.938	44.36	0.988
		Container	25.66	0.628	33.93	0.883	41.42	0.977
		Husky	**18.02**	**0.476**	**21.44**	**0.726**	**22.50**	**0.804**
		Johnny	31.61	0.844	39.65	0.955	45.19	0.984
		Parkjoy	**22.11**	**0.545**	**26.54**	**0.781**	**28.16**	**0.876**
		Parkrun	21.58	0.603	26.90	0.840	29.18	0.921
Taille de GoP = 16	CR=1	*Akiyo*	35.13	0.901	44.05	0.982	52.84	0.997
		Container	31.57	0.821	40.61	0.967	49.84	0.996
		Husky	20.60	0.671	28.66	0.912	38.29	0.987
		Johnny	37.17	0.930	45.82	0.987	54.52	0.998
		Parkjoy	24.84	0.676	32.95	0.905	42.52	0.987
		Parkrun	**24.89**	**0.750**	**33.17**	**0.939**	**42.75**	**0.992**
	CR=0.25	*Akiyo*	30.72	0.808	38.89	0.952	45.29	0.990
		Container	27.09	0.677	35.41	0.908	42.71	0.982
		Husky	18.09	0.477	21.43	0.725	22.45	0.801
		Johnny	33.07	0.869	40.83	0.963	45.77	0.985
		Parkjoy	22.26	0.551	26.65	0.784	28.25	0.877
		Parkrun	**21.99**	**0.622**	**27.23**	**0.849**	**29.44**	**0.925**
Taille de GoP = 32	CR=1	*Akiyo*	**36.27**	**0.917**	**45.06**	**0.985**	**53.77**	**0.998**
		Container	**32.81**	**0.851**	**41.79**	**0.974**	**50.88**	**0.996**
		Husky	20.61	0.671	28.65	0.912	38.29	0.987
		Johnny	**38.13**	**0.939**	**46.67**	**0.988**	**55.53**	**0.998**
		Parkjoy	24.88	0.676	32.97	0.905	42.55	0.987
		Parkrun	25.01	0.753	33.26	0.939	42.84	0.992
	CR=0.25	*Akiyo*	**31.99**	**0.839**	**39.97**	**0.961**	**45.85**	**0.991**
		Container	**28.34**	**0.717**	**36.65**	**0.927**	**43.64**	**0.985**
		Husky	18.11	0.476	21.36	0.721	22.35	0.796
		Johnny	**34.21**	**0.886**	**41.70**	**0.967**	**46.11**	**0.986**
		Parkjoy	22.31	0.552	26.66	0.783	28.24	0.875
		Parkrun	22.17	0.629	27.27	0.850	29.37	0.924

(a) (b) PSNR=29.60dB, SSIM=0.756 (c) PSNR=30.68dB, SSIM=0.783 (d) PSNR=31.52dB, SSIM=0.805

(e) (f) PSNR=20.40dB, SSIM=0.658 (g) PSNR=20.29dB, SSIM=0.651 (h) PSNR=20.14dB, SSIM=0.645

FIGURE 5.2 : Comparaison de la qualité visuelle pour un CSNR = 0 dB, et sans compression appliquée (CR = 1). Séquence *Container*, première image : (a) originale ; (b) SoftCast taille de GoP fixe = 8 ; (c) SoftCast taille de GoP fixe = 16 ; (d) SoftCast taille de GoP fixe = 32. Séquence *Husky*, première image : (e) originale ; (f) SoftCast taille de GoP fixe = 8 ; (g) SoftCast taille de GoP fixe = 16 ; (h) SoftCast taille de GoP fixe = 32.

Comme on peut le constater, l'augmentation de la taille de GoP pour la séquence *Container* (bas TI, haut SI) contribue à améliorer la qualité reçue jusqu'à 2 dB en moyenne en termes de PSNR, dans les mêmes conditions de simulation, i.e., en utilisant le même CSNR et la même bande passante disponible pour la transmission. En revanche, pour la séquence *Husky*, utiliser une taille de GoP de 8 images peut entraîner une légère amélioration de certaines images (jusqu'à 0.3 dB), mais induit une perte moyenne sur la séquence d'environ 0.1 dB, comme indiqué dans le Tableau 5.1. Cependant, choisir une taille de GoP égale à 8 images au lieu de 32 permet de réduire le coût de complexité de 40%.

Enfin, une synthèse globale de toutes les vidéos est présentée en Fig. 5.3 où chaque étiquette associée à un point indique le nom de la vidéo, la taille optimale du GoP ainsi que l'activité des données résultante H_t comme suit <Nom de la vidéo, Taille optimale du GoP, Activité H_t>.

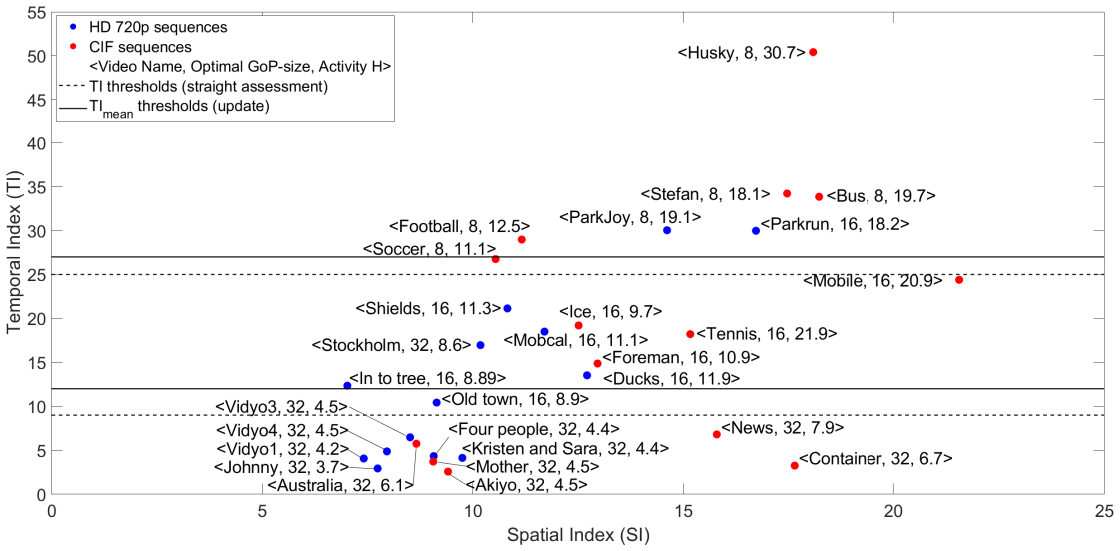


FIGURE 5.3 : Illustration de la taille de GoP optimale et de l'activité résultante H_t par rapport aux index spatio-temporels pour les séquences vidéo CIF et HD720p sélectionnées (classe E). Les points rouges et bleus correspondent respectivement aux valeurs moyennes des index SI, TI pour les séquences CIF et HD720p. Le label associé à chaque point fait référence au triplet de données suivantes : <Nom de la vidéo, Taille optimale du GoP, Activité résultante H_t>.

De prime abord, nous pouvons observer qu'il est possible de choisir des seuils basés sur l'index TI (par exemple 9 et 25) pour définir rapidement une relation entre la taille de GoP optimale et les index SI,TI pour une séquence donnée. Ces seuils sont indiqués en pointillés dans la Fig.5.3. Cependant, nous pouvons observer des incompatibilités pour certaines séquences. Par exemple, la séquence vidéo *Parkrun* possède une taille de GoP optimale de 16 images selon le Tableau 5.1 mais cette séquence est située au-dessus du seuil qui indique une taille GoP optimale de 8 images dans la Fig. 5.3. Ceci est dû au fait que choisir une seule

taille de GoP pour la séquence entière n'est pas suffisant étant donné les très fortes variations de l'index temporel instantané comme indiqué en Fig. 5.4. L'index temporel instantané est dénommé $\sigma_{\mathrm{FD}}(k)$, où $\sigma_{\mathrm{FD}}(k) = std_{space}[F(i,j,k) - F(i,j,k-1)]$. Une taille de GoP optimale de 8 images peut en effet être choisie pour les 336 premières images, tandis que pour les images restantes, une taille GoP de 32 images est sélectionnée pour obtenir la meilleure qualité de reconstruction.

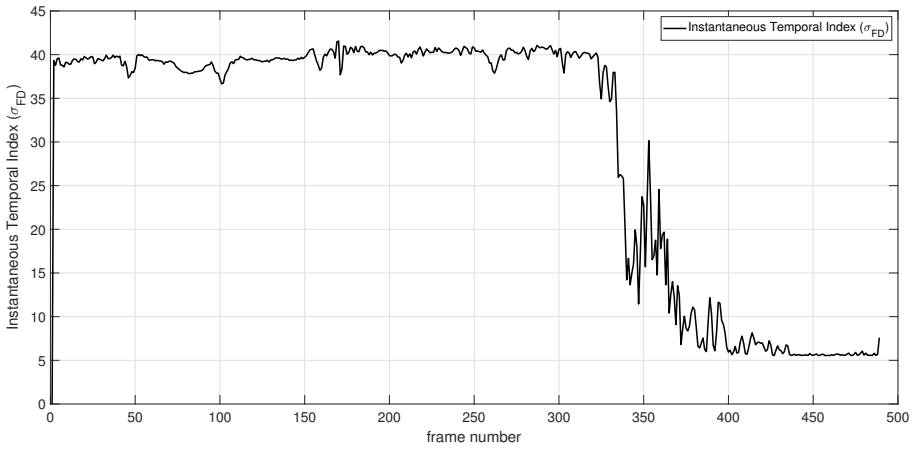


FIGURE 5.4 : Illustration des variations instantanées de l'index temporel notées (σ_{FD}) pour la séquence vidéo *Parkrun*.

Pour pallier ce problème, nous proposons, au lieu de recourir à des seuillages sur l'index TI (seuils en pointillés) obtenus sur la séquence complète, de développer un algorithme basé sur une moyenne arithmétique locale (TI_{mean}) à partir des valeurs instantanées de l'index TI. Cette moyenne locale est évaluée toutes les 8 frames.

Etant donné que les seuils initiaux sont basés sur des résultats moyens obtenus sur la séquence entière, ils doivent être mis à jour et affinés. Dans ces travaux, nous considérons une analyse empirique image par image réalisée sur toutes les séquences de la Fig. 5.3 pour fixer les nouvelles valeurs des seuils. Ces nouveaux seuils (12 et 27) sont introduits dans l'algorithme adaptatif (Tableau 5.2) et sont visibles dans la Fig. 5.3 en traits pleins horizontaux.

Tableau 5.2 : Look-up table pour l'adaptation de la taille de la GoP basée sur un seuillage de l'index TI_{mean}.

Valeur TI_{mean}	Taille de GoP optimale
$\mathrm{TI}_{mean} \leq 12$	32
$12 < \mathrm{TI}_{mean} < 27$	16
$\mathrm{TI}_{mean} \geq 27$	8

Les seuils établis à l'aide des séquences présentes dans la Fig. 5.3 ont été testés avec succès sur d'autres séquences vidéo ne faisant pas partie du dataset de départ parmi lesquelles les classes B et C du JCT-VC (The Joint Collaborative Team on Video Coding) utilisées par le comité MPEG pour la standardisation d'HEVC. Les résultats en Annexe D montrent la validité des seuils proposés.

D'après ce tableau et la valeur de TI_{mean} résultante, la mémoire tampon dont la taille varie de 8 à 32 images est soit vidée soit remplie. Nous choisissons 8 images pour chaque calcul de TI_{mean} afin d'éviter un retard constant de 32 images (environ une seconde pour une séquence CIF) avant d'encoder les données. Nous réduisons ainsi périodiquement le retard à seulement 8 images. Par exemple, si la valeur de TI_{mean} sur 8 images est déjà supérieure ou égale au seuil de 27, nous considérons que ces images ne feront pas partie d'un GoP de 16 ou 32 images même si les 8 images suivantes ont une valeur de TI_{mean} faible ($\leq$12). En revanche, lorsque la valeur TI_{mean} est inférieure à 27, les images restent dans la mémoire tampon. Dans ce cas, la valeur du TI_{mean} est mise à jour en considérant 8 images supplémentaires pour le calcul. Le nombre maximal d'images pouvant être stockées dans la mémoire tampon est de 32 images, car il s'agit du plus grand groupe d'images (GoP) considéré dans ce travail.

De prime abord, on aurait pu penser à la proposition d'une adaptation reposant sur un seuillage de l'activité des données H_t toutefois nous vérifions que celle-ci n'est pas adéquate. En effet, prenons l'exemple, des séquences *Mobile* et Bus pour lesquelles l'activité H_t moyenne est similaire. Cependant leur taille de GoP optimales sont respectivement de 16 et 8 images selon nos critères.

L'activité H_t est en réalité une mesure de la diversité de la distribution énergétique du signal après 3D-DCT dans un GoP [111], c'est-à-dire après une décorrélation spatiale et temporelle, alors que la valeur instantanée du TI donne une mesure de la différence entre deux images consécutives. Même s'il est vrai qu'une GoP optimale peut être définie avec précision en utilisant les équations du Chapitre 2, nos résultats préliminaires montrent qu'il n'est pas nécessaire de recourir au calcul de l'activité H_t puisque le CR et la valeur du CSNR n'influent que très peu sur le choix de la taille de la GoP. De plus, recourir au calcul de l'activité demanderait d'effectuer la transformation DCT-3D de multiples fois (pour chaque taille de GoP) avant de trouver la taille permettant de donner le meilleur compromis qualité de réception / coût de complexité. Au contraire, notre méthode, basée sur une mesure de différence entre images adjacentes, donne en fait une indication sur la possibilité de décorrélation du signal sur l'axe temporel. En effet, une faible valeur de σ_{FD} signifie que la différence entre deux images adjacentes est faible et que, par conséquent, le signal peut être bien décorrélé après la DCT temporelle. En utilisant les valeurs instantanées de l'index TI, nous pouvons prédire une possible réduction de l'activité des données H_t, sans avoir à la calculer. Comme expliqué dans le Chapitre 2, cette réduction d'activité conduit à une amélioration du PSNR reconstruit selon les équations (2.11) et (2.22).

Dans cette analyse préliminaire, nous venons de montrer que selon les caractéristiques du contenu vidéo, le changement de taille de GoP est un moyen efficace permettant d'améliorer la qualité du côté récepteur ou de réduire la complexité tout en offrant des performances similaires. Cependant, ceci n'est valable uniquement si aucun changement de scène n'est présent au sein d'un même GoP. Comme nous l'avons vu dans le Chapitre 3, lorsqu'un GoP contient un changement de plan, les performances de SoftCast sont médiocres puisqu'un effet fantôme apparaît. Le changement de taille de GoP dans ce cas-là n'est pas suffisant et peut conduire à des résultats sous-optimaux. Cela est encore plus vrai pour la transmission de contenus publicitaires, de film / bande-annonce et d'événements sportifs où de nombreux cuts apparaissent. Nous proposons donc dans la suite de ce chapitre de prendre en considération à la fois ces changements de scènes et aussi les variations temporelles des contenus transmis.

Ainsi, sur la base des résultats obtenus dans les sections 3.2.4 et 5.2.1, nous présentons dans la section suivante une extension originale du schéma SoftCast. Tout d'abord, un mécanisme adaptatif de la taille de GoP basé sur la détection de contenu et de cut pour SoftCast (AGCC-SoftCast : Adaptive GoP-size based on Content and Cut detection for SoftCast) est proposé. Cette solution ajuste la taille du GoP en fonction des variations instantanées de l'index temporel (TI) du contenu vidéo transmis tout en évitant l'effet fantôme grâce à la détection de changement de scène / plan. Néanmoins, augmenter la taille du GoP peut entraîner une complexité excessive et des besoins matériels importants (mémoire, processeur). Cela peut ne pas être compatible avec toutes les applications, même s'il est bien connu que les codeurs vidéo classiques sont beaucoup plus complexes que SoftCast en raison de l'estimation / compensation de mouvement [100]. Par conséquent, une méthode alternative basée uniquement sur la détection de cut (AGCut-SoftCast : Adaptive GoP-size based on Cut detection for SoftCast) est également proposée pour les applications dont les ressources sont limitées.

5.2.2 Description des mécanismes proposés

Comme observé dans la Section 5.2.1, l'index TI prévaut sur l'index SI pour le choix de la taille optimale de GoP. De plus, le calcul de l'index SI passe par l'application de l'opérateur Sobel sur chaque image, ce qui demande beaucoup de calculs et prend du temps. Par conséquent, les méthodes proposées ici ne prennent en compte que la valeur instantanée de l'index TI, qui repose sur une mesure de la différence entre deux images consécutives.

5.2.2.1 Détection des changements de scène

La première étape de l'algorithme AGCC consiste à détecter les changements de scène. Le processus de détection est basé sur la valeur instantanée de l'index TI $\sigma_{\mathrm{FD}}(k)$ et introduit dans la Section 5.2.1.2. Lors d'un changement de scène, la valeur instantanée de l'index TI est élevée puisque la différence entre les deux images est importante. Cependant, des valeurs élevées

peuvent également apparaître en raison de changements rapides dans une même scène (par exemple, un contenu sportif). Afin d'éviter une fausse détection, nous effectuons une moyenne mobile ($\mathrm{TI}_{mov}(k)$) sur les valeurs instantanées de l'index TI avec une fenêtre glissante de 7 images. Ensuite, pour chaque image, la valeur $\mathrm{TI}_{mov}(k)$ correspondante est soustraite à la valeur instantanée du TI ($\sigma_{\mathrm{FD}}(k)$). Le signal résultant est comparé à un seuil fixe afin de détecter la position des changements de scène. Sur la base de nombreuses simulations, nous avons fixé le seuil à 10 pour assurer une détection correcte. Un exemple du processus de détection des cuts est donné en Fig. 5.5. Ici, les changements de scène (cuts) apparaissent au niveau des images N°.90 et N°.149 et sont parfaitement détectés par la solution proposée.

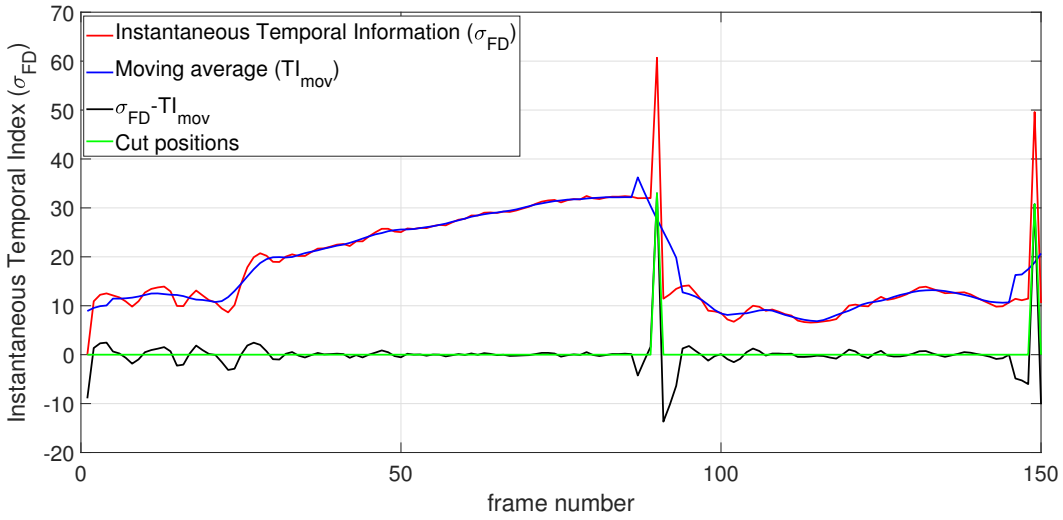


FIGURE 5.5 : Exemple du processus de détection des cuts sur l'index TI pour la séquence vidéo *Tennis* (cut au niveau des images N°.90, 149).

Nous avons montré dans le Chapitre 3 (Section 3.2.4) que le processus de détection des cuts est d'une importance capitale dans un contexte SoftCast afin d'éviter le phénomène d'effet fantôme. De tels changements de scène peuvent apparaître n'importe où dans le contenu vidéo. Par conséquent, il est nécessaire d'adapter la taille du GoP autour de la position des cuts. Dans le cas de la solution AGCut, la taille du GoP doit être ajustée uniquement aux frontières du cut ; ailleurs, cette taille est fixe et choisi égale à 8, 16 ou 32 images, en fonction des ressources matérielles disponibles et des besoins de l'application. Dans le cas du schéma AGCC-SoftCast, la taille du GoP est localement mise à jour à l'intérieur d'une scène, tels que décrit dans la Section 5.2.1. Cette adaptation supplémentaire basée sur des variations temporelles intra-scène permet de tirer pleinement parti des propriétés de décorrélation de la 3D-DCT.

5.2.2.2 Adaptation des tailles de GoP

Dans ce qui suit, nous supposons raisonnablement qu'au moins 8 images séparent deux cuts consécutifs. Considérons tout d'abord la méthode AGCut, basée sur une taille de GoP de 8 images. La taille du GoP résultante après détection des changements de scène est donc comprise entre $8 + 1 = 9$ et $8 + 7 = 15$ images. La méthode AGCut nécessite un buffer de taille raisonnable (15 images) induisant une latence maximale de 15 images (c'est-à-dire, une demi-seconde pour une séquence vidéo avec 30 images par seconde). Bien entendu, en fonction des capacités matérielles et de la latence visée, la taille du GoP peut être étendue à 16 ou 32 images. Enfin, la méthode AGCC permet d'adapter de manière dynamique la taille du GoP à l'intérieur d'une scène, offrant ainsi le meilleur compromis entre amélioration de la qualité visuelle et coût en terme de complexité. Dans ce cas, la taille de GoP peut varier entre 8 et $32 + 7 = 39$ images.

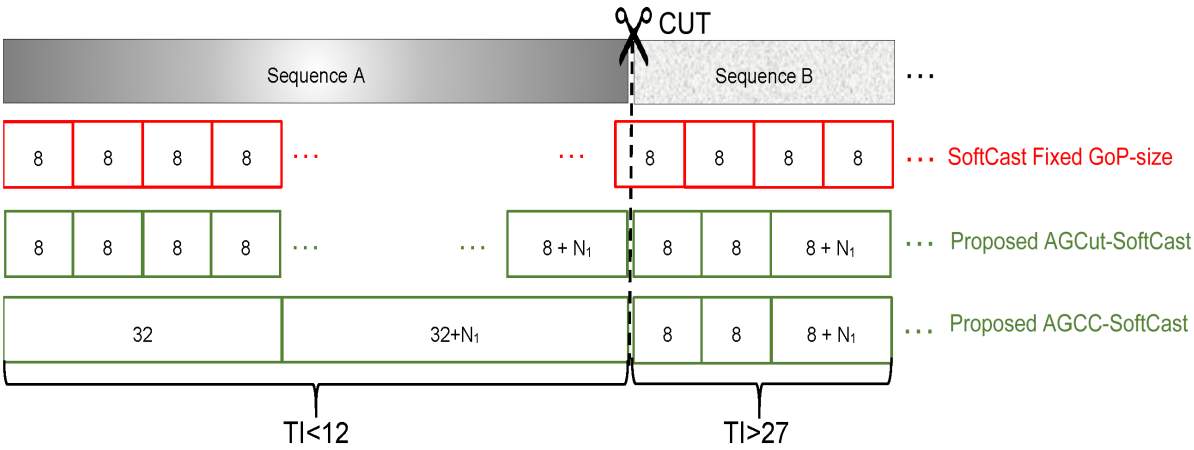


FIGURE 5.6 : Exemple des méthodes proposées pour l'adaptation de la taille du GoP.

La Fig. 5.6 résume les différentes approches décrites ci-dessus. Dans cette figure, un changement de plan se produit entre deux séquences. Par exemple, et sans perte de généralité, supposons que la séquence A contienne des images de *Akiyo* et que la séquence B contienne des images de la séquence *Husky*. Avec le schéma SoftCast originel, les changements de scène sont ignorés et les caractéristiques temporelles de la vidéo ne sont pas prises en compte. Cependant, avec la méthode de détection de cut proposée (AGCut-SoftCast), les changements sont détectés, ce qui résulte en un dernier GoP de taille $8+N_1$ images pour les deux séquences vidéo (ou scènes). Le terme N_1 indique la position du cut compris entre 1 et 7, (puisque nous avons fait l'hypothèse qu'au moins 8 images séparent deux cuts consécutifs). De plus, une adaptation de la taille du GoP est localement effectuée pour la deuxième méthode proposée, à savoir l'extension AGCC-SoftCast, ce qui résulte respectivement en un dernier GoP de taille $32 + N_1$ images pour la séquence *Akiyo* et de $8 + N_1$ images pour la séquence *Husky*.

5.3 Résultats de simulation

La configuration de l'environnement de simulation reste essentiellement la même que celle utilisée dans la Section 5.2.1. Les modifications concernent uniquement les séquences vidéo décrites ci-dessous :

Sources vidéo : Deux grandes séquences vidéo composites en niveaux de gris sont générées dans cette section à partir de petits morceaux de séquences choisis aléatoirement pour évaluer l'extension proposée. Elles sont dénotées respectivement par la séquence $Mixed_{CIF\_cut}$ et la séquence $Mixed_{HD\_cut}$. Les séquences vidéo utilisées ainsi que le nombre d'images de chaque séquence sont donnés dans le Tableau 5.3. Elles ont été choisies pour couvrir une grande partie de la carte SI, TI (voir la Fig. D.1 dans la Section 5.2.1). Nous notons que deux morceaux de la séquence vidéo *Parkrun* ont été utilisés dans la séquence $Mixed_{HD\_cut}$ pour illustrer les variations temporelles illustrées dans la Fig. 5.4. Plus précisément, les 71 premières images de *Parkrun* ont été utilisées, ainsi que 69 images à la fin de la vidéo (images n°400∼469 dénotées par *Parkrun** dans le Tableau 5.3).

Les méthodes proposées (AGCC et AGCut) sont comparées au schéma SoftCast originel en supposant trois tailles de GoP fixes et standards de 8, 16 et 32 images [54, 111, 27].

5.3.1 Analyse des performances image par image

La décomposition selon les différentes tailles de GoP résultant de l'algorithme AGCC est indiquée entre parenthèses dans le Tableau 5.3. Comme observé, les cuts ainsi que les variations temporelles du contenu sont parfaitement détectés : la taille de GoP optimale pour chaque morceau de séquence est obtenue et la dernière taille de GoP est ajustée en conséquence pour éviter l'effet fantôme. Une taille de GoP supérieure est sélectionnée pour les séquences dont les mouvements sont lents, tandis qu'une taille GoP plus petite est sélectionnée pour les vidéos avec des variations temporelles élevées.

5.3.1.1 Transmission sans contrainte de bande passante

Nous montrons d'abord dans les Figs. 5.7 et 5.8 l'évolution de la qualité vidéo reçue mesurée en termes de PSNR et de SSIM pour une transmission sans contrainte de bande passante, c'est-à-dire CR = 1. Les lignes verticales en pointillées représentent la position des cuts. Nous choisissons ici un CSNR intermédiaire (CSNR = 15dB), les résultats pour les autres CSNR sont similaires. Indépendamment de la métrique utilisée, les résultats montrent que :

- Comme rappelé dans la Section 1.2.7 du Chapitre 1, dans les mêmes conditions de canal, la qualité reçue dépend directement de l'activité des données H (voir l'équation (1.32) [111], ou de manière plus générale, l'équation (2.11) de H_t proposée dans le Chapitre 2).

Tableau 5.3 : Caractéristiques des séquences vidéo composites

Séquence vidéo	Séquence vidéo $Mixed_{CIF\_cut}$ Nombre d'images (Décomposition résultante d'AGCC)
News	30 (30)
Husky	22 (8+14)
Mother	64 (2x32)
Stefan	40 (2x8+24)
Australia	64 (2x32)
Akiyo	70 (32+38)
Container	66 (32+34)
Mobile	31 (16+15)
Foreman	54 (32+22)
Football	39 (3x8+15)
Séquence vidéo	**$Mixed_{HD\_cut}$ Nombre d'images (Décomposition résultante d'AGCC)**
Parkjoy	75 (8x8+11)
Johnny	94 (2x32+30)
Into tree	97 (2x32+33)
Parkrun	71 (7x8+15)
Kristen and Sara	125 (3x32+29)
Shields	105 (6x16+9)
Vidyo3	88 (2x32+24)
*Parkrun**	69 (32+37)
Stockholm	76 (4x16+12)

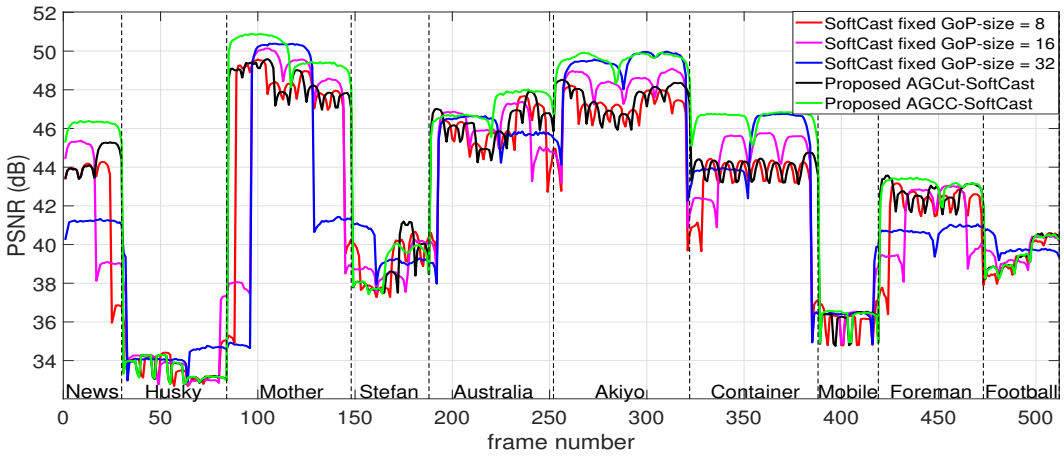


FIGURE 5.7 : Evolution du PSNR en fonction du numéro d'image pour la séquence composite $Mixed_{CIF\_cut}$, CSNR=15dB, CR=1 (pas de compression appliquée).

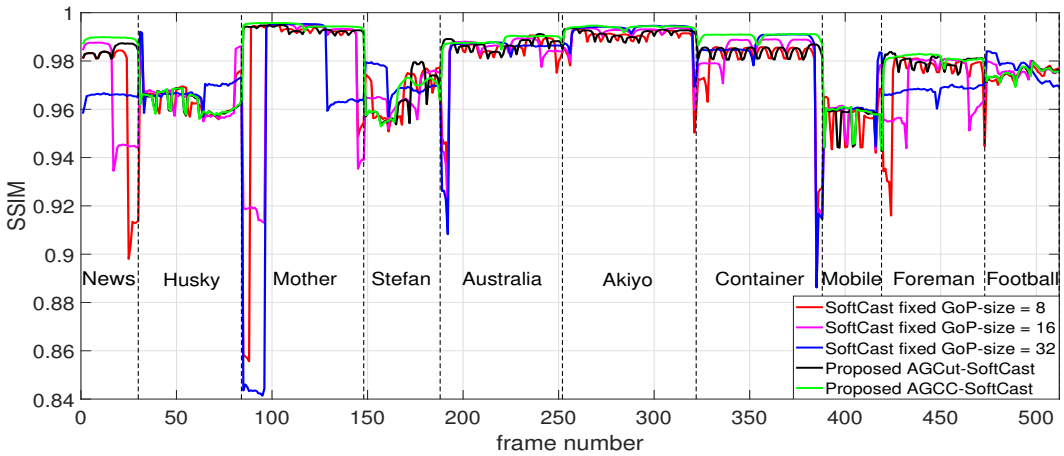


FIGURE 5.8 : Evolution du SSIM en fonction du numéro d'image pour la séquence composite $Mixed_{CIF\_cut}$, CSNR=15dB, CR=1 (pas de compression appliquée).

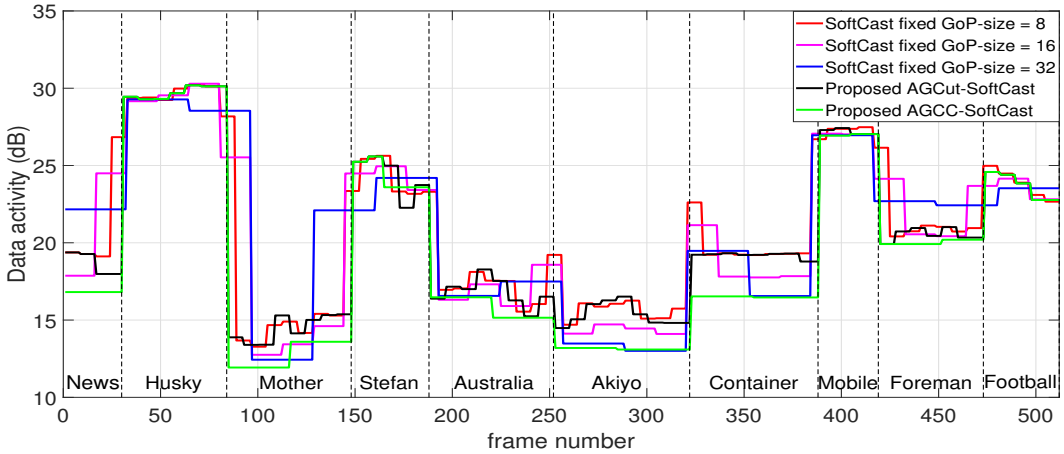


FIGURE 5.9 : Représentation de l'activité des données H_t par GoP exprimée en dB pour la séquence composite $Mixed_{CIF\_cut}$.

Par conséquent, une vidéo contenant des variations temporelles élevées (par exemple, *Husky*) est plus difficile à transmettre qu'une vidéo présentant de faibles variations temporelles (par exemple, *Akiyo*), ce qui entraîne une qualité vidéo reconstruite plus faible ;

- Comme expliqué dans la Section 5.2.1, pour une vidéo à faible activité spatio-temporelle (par exemple, *Container*), une taille de GoP importante permet d'obtenir une meilleure qualité de reconstruction du côté du récepteur. Dans le cas contraire, c'est-à-dire pour des séquences vidéo à activité spatio-temporelle élevée telle que *Husky*, l'utilisation d'une taille de GoP réduite permet de diminuer la complexité avec une qualité reçue équivalente ;
- Même en l'absence de l'effet fantôme (aucune compression n'étant appliquée), nous observons que le PSNR peut devenir sous-optimal pour des solutions basées sur des tailles de GoP fixes. Ceci est particulièrement visible pour la séquence *News*, où la taille optimale du GoP est normalement de 32 images (courbe bleue). De plus, pour la séquence *Mother*, le fait d'utiliser une taille de GoP fixe entraîne une perte de qualité drastique et des fluctuations très importantes de la qualité. En revanche, les méthodes proposées AGCut-SoftCast et AGCC-SoftCast, fournissent une qualité relativement constante pour chaque morceau de séquence ;
- La méthode proposée, AGCC-SoftCast, fournit la plupart du temps la meilleure qualité reçue, car elle bénéficie des processus de détection de cut et de contenu. Cependant, on peut remarquer des images pour lesquelles la qualité donnée par les solutions à taille de GoP fixe est meilleure.

Nous apportons des éclaircissements sur les deux derniers points mentionnés ci-dessus. Pour ce faire, nous utilisons l'équation (2.11) que nous avons proposée dans le Chapitre 2. La Fig. 5.9 représente l'activité des données H_t exprimée en décibels telle qu'elle est utilisée dans (2.11).Nous rappelons d'abord qu'une faible valeur de H_t signifie que la qualité reçue (en termes de PSNR) sera plus élevée d'après (2.11) car elle est soustraite aux termes c+CSNR où c=20log$_{10}$(255). Nous notons également que, comme le montre la figure, l'activité des données est constante sur un GoP puisqu'il s'agit d'un indicateur de la qualité reçue au niveau du GoP (en raison de la 3D-DCT). Comme nous pouvons le constater, lorsqu'une taille de GoP fixe est utilisée lors d'un cut, l'activité résultante est en réalité un mélange des deux activités des vidéos présentes dans le GoP contenant le cut.

Par conséquent, pour la taille de GoP fixe = 32 images (courbe bleue), l'activité de la séquence *News* (30 images) est mélangée à une partie de l'activité de la séquence *Husky* (2 images) aboutissant à une activité intermédiaire. Ce mélange est bénéfique pour les 2 premières images de *Husky* dans la mesure où il augmente la qualité reçue, comme indiqué dans les Figs. 5.7, 5.8 et 5.9. Cependant, ce mélange réduit considérablement la qualité

visuelle des 30 images de la séquence *News*. Nous soulignons le fait que même si la solution basée sur une taille de GoP fixe peut parfois donner des résultats optimaux, elle entraîne une perte de qualité drastique pour les contenus vidéo à faible TI et ne fonctionne que lorsque le GoP contient un cut. De plus, cela n'est valable que dans le cas où un contenu à bas TI est suivi d'un contenu à TI élevé et inversement (voir les changements entre les séquences *Australia*, *Akiyo* et *Container*). Enfin, nous notons que l'activité ne peut pas être utilisée pour le processus de détection des cuts, car elle ne permet pas de différencier les GoP contenant un contenu vidéo à variation temporelle élevée des GoP contenant des cuts.

5.3.1.2 Transmission avec contrainte de bande passante

Nous considérons à présent les environnements à bande passante réduite. Nous choisissons ici le cas CR=0.25. Des résultats similaires sont obtenus pour les autres niveaux de compression. Les Figs. 5.10 et 5.11 représentent l'évolution du PSNR et du SSIM image par image.

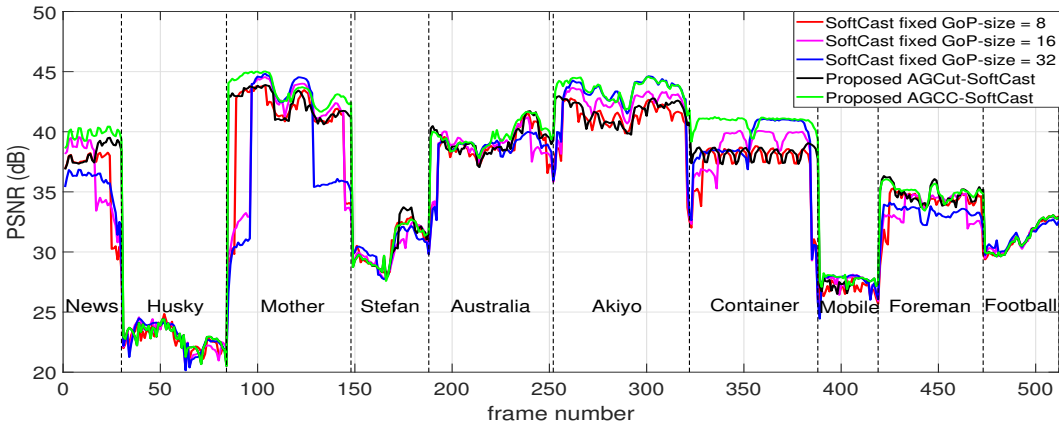


FIGURE 5.10 : Evolution du PSNR en fonction du numéro d'image pour la séquence composite $Mixed_{CIF\_cut}$, CSNR=15dB, CR=0.25 (75% des coefficients sont jetés).

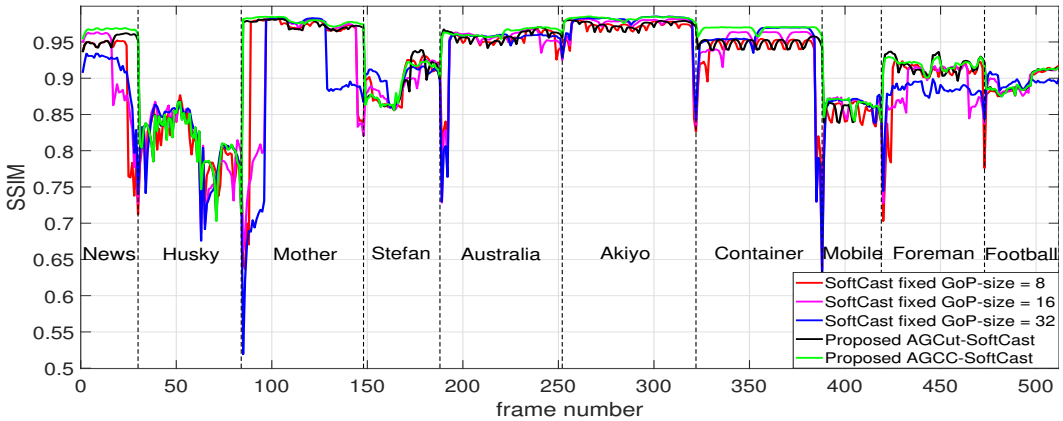


FIGURE 5.11 : Evolution du SSIM en fonction du numéro d'image pour la séquence composite $Mixed_{CIF\_cut}$, CSNR=15dB, CR=0.25 (75% des coefficients sont jetés).

Indépendamment de la métrique utilisée, les résultats montrent que :

- Les améliorations occasionnelles précédemment obtenues aux niveaux des cuts (à CR=1), pour les tailles de GoP fixes et pour les contenus vidéo à haut TI, sont grandement atténuées étant donné que l'effet fantôme apparaît (dû à la compression effectuée sur un changement de plan comme illustré dans la Section 3.2.4 du Chapitre 3) ;
- L'effet fantôme s'étale sur toute la durée du GoP. Par conséquent, il est primordial de détecter les changements de scène étant donné qu'une taille de GoP importante augmente le nombre d'images perturbées ;
- L'index SSIM semble être plus sensible que le PSNR à l'effet fantôme puisque la diminution en terme de qualité est plus grande aux niveaux des cuts en comparaison du PSNR pour lequel la diminution reste semblable au cas où aucune compression n'est effectuée (CR=1). Nous supposons que ceci est dû au fait que la structure de l'image reconstruite a été détruite par les faux contours introduits par l'image fantôme.

Une synthèse des scores PSNR et SSIM pour chaque morceau de séquence est disponible dans le Tableau 5.4. Les deux meilleures qualités par morceau de séquence ont été mises en gras tout comme les gains obtenus entre la méthode proposée AGCC et les méthodes utilisant une taille de GoP fixe égale à 16 ou 32 images. Nous ajoutons également le σ_{PSNR} qui sert à évaluer les fluctuations de PSNR existantes au sein de chaque morceau de séquence. En effet, de nombreuses études [46, 115, 78] ont montré que les fluctuations de qualité sont gênantes pour l'utilisateur et doivent être également considérées pour évaluer les performances d'un système.

La métrique σ_{PSNR} est définie comme :

$$\sigma_{\text{PSNR}} = \sqrt{\frac{1}{F}\sum_{k=1}^{F}(\text{PSNR}_k - \text{PSNR}_{avg})^2}, \tag{5.1}$$

où F représente le nombre d'images dans le morceau de séquence considéré, PSNR_k et PSNR_{avg} sont respectivement les scores de PSNR de la $k^{ème}$ image et la valeur du PSNR moyen sur le morceau de séquence considéré. Les valeurs de PSNR sont bien entendu exprimées en décibels.

Comme illustré dans ce tableau :

- Les solutions proposées garantissent une meilleure qualité dans l'ensemble avec des fluctuations de PSNR grandement limitées en comparaison des solutions actuelles reposant sur un encodage à taille de GoP fixe. Indépendamment du niveau de compression appliqué, les scores de σ_{PSNR} pour les extensions proposées : AGCC-SoftCast et AGCut-SoftCast demeurent toujours en dessous de 1dB en moyenne ;
- En revanche, les solutions à taille de GoP fixe sont sujettes à des fluctuations de PSNR de l'ordre de 2.5-3.5dB en moyenne. Comme expliqué ci-dessus et montré dans les

Tableau 5.4 : Tableau des différents scores PSNR, σ_{PSNR} et SSIM pour la séquence $Mixed_{CIF\_cut}$, pour un CSNR=15dB et considérant différentes configurations de taille de GoP et différents niveaux de compression : CR = 1 ; CR = 0.25.

Environnement de simulation			Morceau de séquence vidéo									
			News	*Husky*	*Mother*	*Stefan*	*Australia*	*Akiyo*	*Container*	*Mobile*	*Foreman*	*Football*
CSNR=15dB CR=1	PSNR(dB)	GoP=8	41.17	33.66	44.16	**38.89**	45.09	46.52	42.55	35.99	41.43	39.34
		GoP=16	41.17	**33.86**	43.18	38.84	45.18	47.59	43.05	**36.41**	41.03	**39.62**
		GoP=32	41.16	**34.33**	40.43	**39.53**	44.98	**48.62**	43.45	**36.57**	40.71	**39.80**
		AGCut	**44.35**	33.51	**48.31**	38.70	**46.14**	47.49	**43.95**	36.05	**42.62**	39.45
		AGCC	**46.19**	33.51	**49.87**	38.81	**47.09**	**49.58**	**46.53**	36.24	**43.12**	39.43
		Gain AGCC/16	**5.01**	**-0.35**	**6.69**	**-0.03**	**1.92**	**1.99**	**3.48**	**-0.17**	**2.08**	**-0.19**
		Gain AGCC/32	**5.03**	**-0.82**	**9.44**	**-0.73**	**2.12**	**0.96**	**3.09**	**-0.33**	**2.40**	**-0.37**
	σ_{PSNR}(dB)	GoP=8	2.93	0.77	3.75	1.17	1.76	1.51	1.93	0.69	1.64	0.81
		GoP=16	3.11	1.32	4.83	**0.77**	1.85	1.61	2.44	0.81	1.66	**0.63**
		GoP=32	**0.20**	1.06	6.11	**0.87**	1.85	1.59	2.57	1.02	**0.28**	**0.38**
		AGCut	0.67	**0.52**	**0.79**	1.28	**0.88**	**0.73**	**0.46**	**0.61**	0.52	0.79
		AGCC	**0.29**	**0.52**	**0.78**	0.96	**0.69**	**0.35**	**0.44**	**0.52**	**0.35**	0.77
	SSIM	GoP=8	0.968	0.964	0.982	**0.966**	0.983	0.990	0.979	0.957	0.973	0.975
		GoP=16	0.967	**0.965**	0.976	**0.966**	0.983	0.991	0.981	**0.960**	0.970	**0.976**
		GoP=32	0.966	**0.968**	0.957	**0.971**	0.982	**0.993**	0.983	**0.962**	0.968	**0.977**
		AGCut	0.985	0.963	**0.993**	0.965	**0.987**	0.991	**0.984**	0.956	**0.980**	0.975
		AGCC	**0.989**	0.963	**0.995**	0.965	**0.989**	**0.994**	**0.991**	0.958	**0.981**	0.975
		Gain AGCC/16	**0.023**	**-0.003**	**0.019**	**-0.001**	**0.005**	**0.003**	**0.009**	**-0.003**	**0.011**	**-0.001**
		Gain AGCC/32	**0.024**	**-0.006**	**0.038**	**-0.006**	**0.006**	**0.001**	**0.008**	**-0.004**	**0.013**	**-0.001**
CSNR=15dB CR=0.25	PSNR(dB)	GoP=8	34.63	22.63	38.37	30.23	38.22	40.53	36.47	**26.89**	33.73	31.01
		GoP=16	34.97	22.63	37.27	29.90	38.21	41.47	36.66	27.36	33.49	31.12
		GoP=32	35.09	22.72	35.32	30.10	38.12	**42.26**	37.00	27.49	33.18	**31.18**
		AGCut	**38.17**	**22.79**	**42.17**	**30.18**	**39.16**	41.57	**38.17**	27.39	**34.76**	31.06
		AGCC	**39.78**	**22.79**	**43.35**	**30.18**	**39.61**	**43.50**	**40.81**	**27.72**	**34.93**	**31.06**
		Gain AGCC/16	**4.81**	**0.15**	**6.07**	**0.27**	**1.40**	**2.03**	**4.16**	**0.36**	**1.44**	**-0.06**
		Gain AGCC/32	**4.69**	**0.07**	**8.02**	**0.07**	**1.48**	**1.24**	**3.82**	**0.23**	**1.75**	**-0.11**
	σ_{PSNR}(dB)	GoP=8	3.18	0.94	3.65	1.63	1.72	1.72	2.19	0.58	1.36	1.23
		GoP=16	3.12	1.11	4.66	**1.43**	1.70	1.88	2.83	0.73	1.16	**1.16**
		GoP=32	1.74	1.13	5.53	**1.31**	1.58	1.98	2.95	0.78	**0.65**	**0.91**
		AGCut	**0.80**	**0.88**	**1.10**	1.83	**1.08**	**0.91**	**0.48**	**0.38**	0.73	1.21
		AGCC	**0.46**	**0.89**	**1.07**	1.60	**0.99**	**0.80**	**0.48**	**0.32**	**0.59**	1.21
	SSIM	GoP=8	0.910	0.808	0.946	**0.896**	0.947	0.968	0.935	0.852	0.898	0.894
		GoP=16	0.908	0.808	0.928	0.892	0.947	0.974	0.940	0.862	0.888	**0.896**
		GoP=32	0.906	**0.812**	0.895	**0.900**	0.946	**0.978**	0.946	**0.864**	0.881	**0.895**
		AGCut	**0.953**	**0.812**	**0.974**	0.893	**0.958**	0.974	**0.948**	0.858	**0.918**	**0.895**
		AGCC	**0.967**	**0.812**	**0.979**	0.893	**0.963**	**0.983**	**0.969**	**0.864**	**0.919**	**0.895**
		Gain AGCC/16	**0.059**	**0.003**	**0.051**	**0.001**	**0.017**	**0.009**	**0.028**	**0.002**	**0.031**	**-0.001**
		Gain AGCC/32	**0.060**	**-0.001**	**0.084**	**-0.007**	**0.017**	**0.005**	**0.023**	**0.000**	**0.039**	**0.000**

Figs. 5.7, 5.8 et 5.9, ceci est dû au fait que les solutions originelles mélangent les activités de deux séquences différentes au niveau d'un cut ce qui induit d'importantes fluctuations temporelles de la qualité ;

- Il est intéressant de noter que la solution proposée AGCut-SoftCast (basée sur une taille de GoP = 8 images) fonctionne globalement mieux que la version fixe avec une taille de GoP = 32 images pour les mêmes raisons que mentionnées ci-dessus ;
- Les gains obtenus pour le SSIM ont des valeurs numériques relativement faibles. Ceci est dû à la dynamique de la métrique à haut niveau de qualité (CSNR≥15dB). Néanmoins, nous observons des gains en terme de PSNR pouvant aller jusqu'à 8dB selon le contenu vidéo transmis et la taille de GoP fixe considérée, ce qui illustre bien l'amélioration de la qualité reçue.

En tant qu'indicateur de performance supplémentaire, nous évaluons également le pourcentage du temps durant lequel la qualité vidéo obtenue (via le PSNR) avec la solution proposée AGCC-SoftCast est meilleure ou égale que la solution originelle (avec une taille de GoP fixe). Nous utilisons un seuil informel de ±0.4dB [83] pour déterminer si la vidéo reçue est supérieure, égale ou inférieure à la solution originelle en terme de qualité reconstruite. Comme nous pouvons le voir dans le Tableau 5.5 : la version proposée AGCC-SoftCast donne une qualité reçue supérieure ou égale à la version originelle (taille de GoP = 32) pour plus de 82% des images pour la séquence $Mixed_{CIF\_cut}$ et plus de 72% du temps pour la séquence $Mixed_{HD\_cut}$. Plus le CR diminue, plus le pourcentage augmente jusqu'à des valeurs pouvant atteindre 89% étant donné que l'effet fantôme apparait et qu'il impacte la qualité de reconstruction. Ces pourcentages augmentent entre 90% et 94% dès lors qu'une taille de GoP fixe égale à 16 images est considérée, ce qui illustre bien l'intérêt de la solution proposée.

Tableau 5.5 : Représentation du pourcentage du temps où la qualité obtenue par la méthode proposée est supérieure ou égale à la version originelle de SoftCast utilisant des tailles de GoP fixes.

Paramètres		Pourcentage du temps AGCC ≥ GoP 16	Pourcentage du temps AGCC ≥ GoP 32
$Mixed_{CIF\_cut}$	CR=1	90.1	82.7
	CR=0.75	89.3	82.3
	CR=0.5	91.6	87.3
	CR=0.25	92.6	88.6
$Mixed_{HD\_cut}$	CR=1	90.1	72.6
	CR=0.75	92.5	78.5
	CR=0.5	94.3	85.6
	CR=0.25	92.3	86.6

5.3.2 Analyse des performances globales

Pour mieux voir la contribution des méthodes proposées, nous regardons également l'évolution des scores moyens (PSNR et SSIM) sur une large plage de CSNR [0 $\sim$ 25dB]. Les résultats sont présentés dans la Fig. 5.12. Quel que soit le taux de compression appliqué, ceux-ci montrent que :

- Les méthodes proposées fournissent toujours de meilleurs résultats moyens que le schéma SoftCast originel basé sur une taille de GoP fixe. Le gain moyen est d'environ 1 dB en termes de PSNR et de 0.05 en termes de SSIM ;
- Nous notons que l'amélioration du SSIM est supérieure pour de plus faibles valeurs de CSNR. En effet, pour des valeurs de CSNR élevées, le bruit devient insignifiant et la qualité vidéo reçue se rapproche donc de la qualité maximale disponible après compression ;
- Lorsqu'un taux de compression est appliqué, comme indiqué dans la figure en bas à gauche de la Fig. 5.12, la linéarité du schéma SoftCast est brisée en raison de la perte d'informations à l'émission, impossible à récupérer au récepteur. Pour rappel, ce phénomène est connu sous le nom de *levelling-off effect* [64] et a été modélisé théoriquement dans le Chapitre 2 ;
- Étant donné que les changements de scène (cuts) n'ont pas été pris en compte par le schéma SoftCast originel (taille de GoP fixe), les améliorations de qualité initialement observées dans la Section 5.2.1 entre les tailles 8, 16 et 32 images deviennent ici erronées et insignifiantes (<0.5 dB), ce qui démontre bien l'efficacité des méthodes proposées et l'importance de la prise en compte des changements de scène ;

5.3.3 Comparaison visuelle

Enfin, une comparaison visuelle est donnée dans les Figs. 5.13 et 5.14 pour évaluer les images reconstruites par les différentes méthodes. Dans le but de synthétiser les résultats, nous choisissons de montrer uniquement les séquences CIF en choisissant comme référence la taille de GoP fixe de 32 images, car elle donne généralement les meilleurs résultats [111]. Nous montrons également les deux méthodes proposées, c'est-à-dire AGCut-SoftCast qui ne prend en compte que le processus de détection de cut et est basée sur la plus petite taille de GoP dans notre étude (8 images) ; ainsi que le schéma AGCC-SoftCast, où les processus de détection de cut et de contenu sont utilisés. Nous choisissons un CSNR de 15 dB (qualité de canal intermédiaire). Des conclusions similaires sont obtenues pour la séquence $Mixed_{HD\_cut}$ et les autres conditions de transmission.

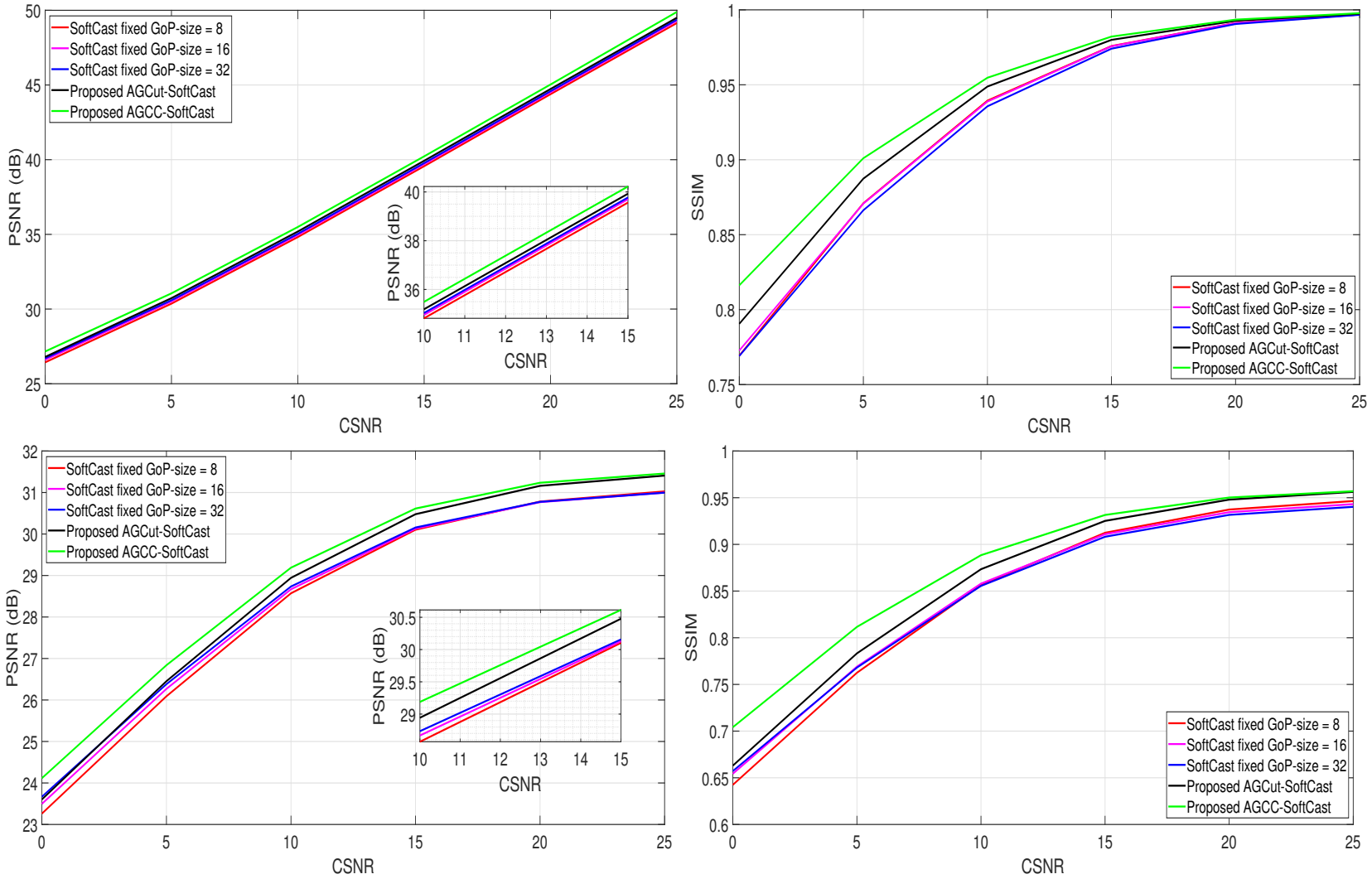


FIGURE 5.12 : Evolution des scores moyens (PSNR et SSIM) en fonction du CSNR. Première colonne : métrique PSNR. Deuxième colonne : métrique SSIM. Première ligne : CR=1. Deuxième ligne : CR=0.25.

(a)

(b) PSNR=46.67dB, SSIM=0.990

(c) PSNR=43.94dB, SSIM=0.983

(d) PSNR=46.75dB, SSIM=0.990

(e)

(f) PSNR=36.41dB, SSIM=0.914

(g) PSNR=43.57dB, SSIM=0.981

(h) PSNR=45.23dB, SSIM=0.986

(i)

(j) PSNR=36.29dB, SSIM=0.959

(k) PSNR=34.86dB, SSIM=0.944

(l) PSNR=34.91dB, SSIM=0.944

(m) (n) PSNR=39.15dB, SSIM=0.980 (o) PSNR=36.38dB, SSIM=0.957 (p) PSNR=36.36dB, SSIM=0.957

FIGURE 5.13 : Comparaison visuelle de la qualité reçue à CSNR = 15dB, CR = 1 pour la séquence $Mixed_{CIF}$ (images n° 361, 388, 389, 417). Première ligne : Image N°.361 = image issue de *Container* (position : 28 images avant le cut). Deuxième ligne : Image N°.388 = image issue de *Container* (position : dernière image avant le cut). Troisième ligne : Image N°.389 = image issue de *Mobile* (position : première image après le cut). Quatrième ligne : Image N°.417 = image issue de *Mobile* (28 images après le cut). Première colonne : Image d'origine. Deuxième colonne : SoftCast originel (taille de GoP = 32 images). Troisième colonne : AGCut-SoftCast (base de taille de GoP = 8 images). Quatrième colonne : AGCC-SoftCast.

(a) (b) PSNR=41.05dB, SSIM=0.969 (c) PSNR=38.19dB, SSIM=0.947 (d) PSNR=41.11dB, SSIM=0.970

(e) (f) PSNR=25.82dB, SSIM=0.627 (g) PSNR=37.74dB, SSIM=0.934 (h) PSNR=39.34dB, SSIM=0.957

(i) (j) PSNR=24.44dB, SSIM=0.790 (k) PSNR=27.18dB, SSIM=0.838 (l) PSNR=27.57dB, SSIM=0.846

(m) (n) PSNR=27.40dB, SSIM=0.875 (o) PSNR=27.80dB, SSIM=0.861 (p) PSNR=27.81dB, SSIM=0.862

FIGURE 5.14 : Comparaison visuelle de la qualité reçue à CSNR = 15dB, CR = 0.25 pour la séquence $Mixed_{CIF}$ (images n° 361, 388, 389, 417). Première ligne : Image N°.361 = image issue de *Container* (position : 28 images avant le cut). Deuxième ligne : Image N°.388 = image issue de *Container* (position : dernière image avant le cut). Troisième ligne : Image N°.389 = image issue de *Mobile* (position : première image après le cut). Quatrième ligne : Image N°.417 = image issue de *Mobile* (28 images après le cut). Première colonne : Image d'origine. Deuxième colonne : SoftCast originel (taille de GoP = 32 images). Troisième colonne : AGCut-SoftCast (base de taille de GoP = 8 images). Quatrième colonne : AGCC-SoftCast.

5.3.3.1 Transmission sans contrainte de bande passante

Nous comparons tout d'abord les résultats en ne tenant compte d'aucune restriction de bande passante (Fig. 5.13). Comme on peut le voir dans les lignes du milieu représentant les frontières d'un changement de scène de la séquence $Mixed_{CIF\_cut}$ (images n° 388 et 389), les méthodes proposées donnent de meilleurs résultats pour les contenus à faible TI (c'est-à-dire *Container*). Cependant, et comme expliqué précédemment, la méthode utilisant une taille de GoP fixe bénéficie de la faible activité de la séquence *Container* pour améliorer quelques images de la séquence *Mobile*. Pour montrer que cet effet ne dépend que de la durée du GoP contenant le changement de scène, nous affichons également dans la première et quatrième ligne de la figure, les résultats obtenus pour les GoP adjacentes (c'est-à-dire la $28^{ème}$ image présente avant et après le cut). Comme observé, la séquence *Container* souffre de sévères dégradations et de fluctuations temporelles de qualité (>10 dB en termes de PSNR) alors que les méthodes proposées restent relativement constantes. Comme le nombre total d'images pour le morceau de séquence *Mobile* a été choisi de façon arbitraire égal à 31 images, il n'y a pas assez d'images pour encoder un GoP de 32 images sans inclure un cut. Par conséquent, l'image n°417 profite également de la faible activité du contenu vidéo suivant, à savoir *Foreman*, ce qui explique le fait que les deux images n° 389 et 417 aient une qualité reçue supérieure aux méthodes proposées.

5.3.3.2 Transmission avec contrainte de bande passante

D'un autre côté, lorsque la bande passante disponible pour la transmission est limitée (Fig. 5.14), nous pouvons clairement constater que la version originelle de SoftCast (i.e., avec taille de GoP fixe) offre une qualité vidéo reçue plus mauvaise, quelle que soit la taille de GoP considérée. En revanche, les extensions proposées permettent d'obtenir une meilleure qualité de réception sous les mêmes caractéristiques de canal avec la même quantité de bande passante disponible. La différence de PSNR entre les solutions originelles et les solutions proposées peut atteindre 16 dB en termes de PSNR au niveau des changements de plan. Si aucun cut n'est détecté, les méthodes AGCC-SoftCast et SoftCast originel (taille de GoP fixe = 32) offrent des résultats similaires, car elles reposent toutes deux sur une taille GoP de 32 images. Cependant, AGCC-SoftCast réduit périodiquement la taille du GoP afin de réduire la complexité, laissant le matériel hardware (processeur, RAM, etc.) disponible pour d'autres tâches.

Le Tableau 5.6 donne la répartition, en nombre de GoP et selon les tailles de GoP disponibles, pour les méthodes proposées et la méthode originelle SoftCast (taille de GoP fixe) afin d'évaluer la complexité globale des méthodes. La méthode AGCut proposée représente un bon compromis entre qualité reçue et coût de complexité, car elle repose sur une taille de GoP de base de 8 images. Toutefois, si le matériel disponible permet d'utiliser une taille

Tableau 5.6 : Répartition, en nombre de GoP et selon les tailles de GoP disponibles, pour les méthodes proposées et la méthode classique SoftCast (taille de GoP fixe).

Paramètres		Taille de GoP considérée			
		8	16	32	autres
$Mixed_{CIF\_cut}$	GoP=8	64	-	-	-
	GoP=16	-	32	-	-
	GoP=32	-	-	16	-
	AGCut	52	-	-	7
	AGCC	10	1	7	9
$Mixed_{HD\_cut}$	GoP=8	100	-	-	-
	GoP=16	-	50	-	-
	GoP=32	-	-	25	-
	AGCut	88	-	-	8
	AGCC	15	10	10	9

de GoP plus importante, l'extension AGCC-SoftCast proposée offre alors la meilleure qualité reconstruite avec un bon compromis en matière de coût de complexité.

Les résultats présentés dans cette section soulignent clairement les avantages des méthodes proposées. Quelles que soient les conditions du canal, la qualité reçue est améliorée tandis que les fluctuations de qualité sont considérablement réduites. En outre, par rapport à une taille de GoP fixe de 32 images, la solution AGCut-SoftCast proposée réduit la complexité jusqu'à 40%. Dans le cas de la version AGCC-SoftCast, il est plus difficile de calculer la réduction de la complexité puisque cette dernière dépend essentiellement du type de contenu vidéo transmis. Ainsi, pour un contenu vidéo transmis ayant une forte activité spatiotemporelle, l'algorithme AGCC-SoftCast reposera sur une taille de GoP de 8 images offrant une réduction similaire à AGCut-SoftCast. Dans le cas contraire, (i.e., un contenu vidéo présentant une faible activité), la réduction de l'activité sera quasi-nulle puisque AGCC-SoftCast choisira principalement une taille de GoP de 32 images pour obtenir une meilleure qualité à la réception.

5.4 Conclusion

Dans ce chapitre, nous avons évalué et optimisé les performances du schéma SoftCast en analysant les variations temporelles du contenu vidéo transmis. Une analyse préliminaire a montré qu'il est primordial de prendre en compte les caractéristiques spatio-temporelles des vidéos dans un contexte SoftCast. En fonction du contenu vidéo transmis, le changement de taille de GoP est un moyen efficace permettant soit d'améliorer la qualité du côté récepteur soit de réduire la complexité tout en offrant des performances similaires.

Nous avons également montré que l'utilisation d'une taille de GoP fixe entraîne de graves fluctuations de la qualité visuelle (environ 2.5-3.5 dB en termes de PSNR) ainsi que des artefacts fantômes gênants lorsque la bande passante disponible est limitée. L'algorithme proposé permet soit de supprimer ces altérations (effet fantôme) ou de les réduire considérablement (fluctuations de la qualité) en veillant à ce qu'aucun GoP ne contienne deux scènes différentes.

Sur la base de ces résultats, une extension du schéma SoftCast a été proposée consistant en un mécanisme adaptatif de la taille GoP basé sur la détection de contenu et de cut pour SoftCast (AGCC-SoftCast). Une méthode alternative basée uniquement sur la détection de cut (AGCut-SoftCast) a également été proposée pour les applications limitées en ressources hardware et/ou ayant des contraintes de faible latence.

Une amélioration en termes de score PSNR allant jusqu'à 16 dB et jusqu'à 0.55 pour l'index SSIM est observée avec les méthodes proposées aux frontières des changements de scène. Ces méthodes permettent également de réduire les fluctuations temporelles de qualité visuelle en dessous de 1 dB en moyenne, démontrant ainsi leur efficacité.

La méthode AGCut-SoftCast proposée reposant sur une taille de GoP de base de 8 images permet de maintenir de bonnes performances tout en réduisant les coûts de complexité et les exigences matérielles requises. Enfin, la méthode AGCC permet d'adapter de manière dynamique la taille du GoP à l'intérieur d'une même scène, offrant ainsi le meilleur compromis entre amélioration de la qualité visuelle et coût de complexité avec des améliorations en termes de PSNR pouvant aller jusqu'à 2.6 dB par rapport à la version AGCut-SoftCast.

Conclusions générales et Perspectives

Conclusions générales

Dans cette thèse, nous avons étudié et optimisé un nouveau schéma de transmission vidéo conjoint Source-Canal basé SoftCast [54]. SoftCast représente le pionnier des architectures appelées “Codeurs Vidéo Linéaires” où la qualité vidéo reçue est une fonction linéaire de la qualité du canal. Nous avons proposé diverses contributions par rapport à ce schéma :

- Dans le Chapitre 2, nous avons proposé des modèles théoriques d'évaluation de la qualité reçue (de bout en bout). Ceux-ci représentent une extension des travaux de Xiong *et al.* [111] prenant en considération plusieurs paramètres additionnels : 1) les contraintes de bande passante disponible existantes au niveau de l'émetteur, 2) l'allocation de puissance effectuée à l'émetteur et 3) le type d'estimateur (ZF ou LLSE) utilisé au récepteur. Les résultats montrent que les modèles proposés concordent parfaitement avec les résultats obtenus en simulation. Ils permettent dès lors de quantifier précisément l'apport des différents schémas proposés. Ainsi, le gain apporté par l'estimateur LLSE par rapport au ZF a été quantifié, de même que les gains pouvant être obtenus en passant d'une allocation quasi-optimale (SoftCast originel, i.e., sans connaissance des caractéristiques du canal) à une allocation optimale (SoftCast+, où les caractéristiques du canal sont envoyées à l'émetteur).
- Dans le Chapitre 3, nous avons proposé un recensement des différents artefacts visuels (effet de neige, effet de cloche, effet de flou et effet fantôme) pouvant apparaître dans un contexte de transmission vidéo ayant recours à un CVL. Nous avons ensuite évalué à l'aide de tests subjectifs le ressenti global des utilisateurs quant à la qualité visuelle de contenus vidéos reçus via SoftCast puis celui lié au type d'estimateur (ZF/LLSE) utilisé à la réception. En outre, nous avons également évalué les performances de métriques objectives (e.g. PSNR, SSIM ou encore VMAF) face aux scores objectifs obtenus. Il s'agit de la première étude complète de ce type dans un contexte SoftCast. Les résultats

montrent que la linéarité observée dans de nombreux travaux entre le PSNR reçue et la qualité du canal est aussi obtenue dans le cadre des scores subjectifs (MOS). Selon nos résultats, nous avons montré que la meilleure corrélation avec les scores objectifs est obtenue pour l'index SSIM. En outre, nous avons montré que le PSNR fournit également une bonne correspondance avec les scores subjectifs validant les modèles proposés dans le Chapitre 2 pour l'évaluation des performances des CVL.

- Dans le Chapitre 4, nous avons introduit et analysé des méthodes de prétraitement existantes permettant de réduire l'artefact caractéristique de certains CVL (effet de neige) et améliorant significativement la qualité en réception. Une version originale a ensuite été proposée sur la base des travaux de He *et al.* [40]. Les résultats ont montré que la méthode proposée permet d'obtenir des performances similaires à la méthode originelle (gain en PSNR de l'ordre de 3 dB) tout en réduisant de moitié le temps de calcul nécessaire au prétraitement par deux ainsi que le volume des informations additionnelles à transmettre (réduction de 75%).
- Enfin, dans le dernier chapitre, nous avons tout d'abord montré que l'utilisation d'une taille de GoP fixe (classiquement observée dans de nombreux travaux liés au CVL) dans un schéma SoftCast entraîne de graves fluctuations de la qualité visuelle (environ 2.5-3.5 dB en termes de PSNR) ainsi que des artefacts fantômes gênants lorsque la bande passante disponible est limitée. Pour répondre à ces problèmes, nous avons proposé une extension du schéma SoftCast consistant en un mécanisme adaptatif de la taille GoP basé sur l'analyse des propriétés spatiotemporelles du contenu vidéo et sur la détection de cuts pour SoftCast (AGCC-SoftCast). Une méthode alternative basée uniquement sur la détection de cuts (AGCut-SoftCast) a également été proposée pour les applications limitées en ressources hardware et/ou ayant des contraintes de faible latence. Ainsi, une amélioration du PSNR allant jusqu'à 16 dB et jusqu'à 0.55 pour l'index SSIM est observée avec les méthodes proposées spécifiquement aux frontières des changements de scène. Ces méthodes permettent également de réduire les fluctuations temporelles de qualité visuelle en dessous de 1 dB en moyenne, démontrant ainsi leur efficacité.

Perspectives

A l'issue de ces travaux de thèse, plusieurs perspectives à moyen et long termes sont envisagées et présentées ci-dessous.

Tout d'abord, nous comptons implémenter les solutions proposées à l'aide d'outils SDR (SoftWare Defined Radio) comme des cartes USRP [52, 6, 99] (Universal Software Radio Peripheral, https://www.ni.com/fr-fr/shop/select/usrp-software-defined-radio-device). Ceci nous permettra d'évaluer les performances de SoftCast lors d'expérimentations en environnement réel. En outre, les résultats récemment obtenus par Tung et Gunduz [99] mettent en avant des différences entre résultats de simulations et expérimentations réelles via USRP (Fig. 5.15). L'objectif serait alors d'identifier les causes de ces différences pour raffiner les modèles théoriques que nous avons développé dans ce travail de thèse, et ainsi mieux représenter les transmissions réelles.

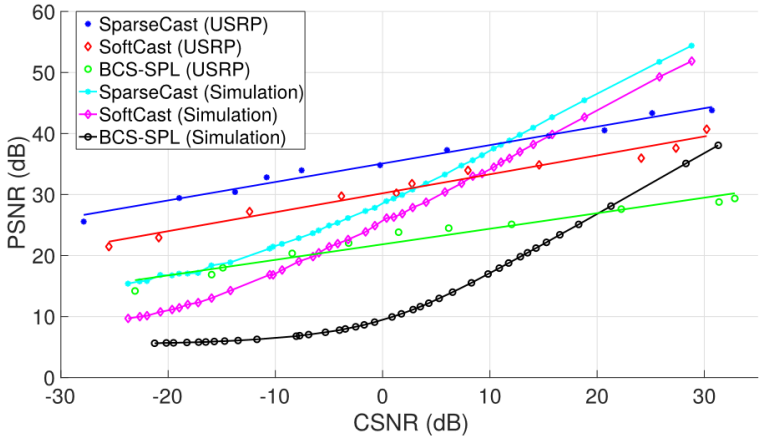


FIGURE 5.15 : Evolution de la qualité d'image reçue (PSNR) en fonction de la qualité du canal de transmission (CSNR) via simulation et implémentation (USRP) pour un CR=1. Figure issue de [99]. Les courbes BCS-SPL réfèrent à [116].

D'un autre côté, nous avons fait l'hypothèse tout au long de nos travaux que les métadonnées étaient toujours reçues et décodables sans erreurs au récepteur. Des récents travaux [6, 131] ont montré que l'effet de *cliff* pouvait encore apparaître dans un contexte CVL de par le fait qu'en cas d'erreurs sur les métadonnées envoyées numériquement, SoftCast ne peut pas reconstruire l'image/vidéo (car le récepteur ne dispose d'aucune information sur l'allocation de puissance). Il serait intéressant, dans nos travaux futurs, de recourir à l'allocation de puissance que Zong *el al.* [131] ont récemment proposé. Celle-ci supprime en effet totalement le *cliff-effect* car le schéma proposé fonctionne sans métadonnées et est donc 100% analogique. Une illustration des résultats qu'ils ont obtenus par rapport à SoftCast est présentée en Fig. 5.16.

Concernant les travaux subjectifs, il serait judicieux d'intégrer tout d'abord des solutions de post traitement (préservant les contours de l'image) pour réduire l'effet de neige permettant l'utilisation de l'estimateur ZF tout en évitant le flou engendré par le LLSE à bas CSNR. Du point de vue théorique, il serait judicieux d'envisager une solution de compromis entre l'estimateur ZF non biaisé mais à forte variance pour des valeurs de CSNR faibles et le LLSE qui fournit une estimation biaisée mais de variance relativement plus faible. Finalement, il serait également intéressant d'évaluer de manière continue la qualité perçue par l'utilisateur dans un contexte SoftCast où le canal varie dans le temps via l'utilisation par exemple d'un test SSCQE (Single Stimulus Continuous Quality Evaluation) [9] illustré en Fig. 5.17.

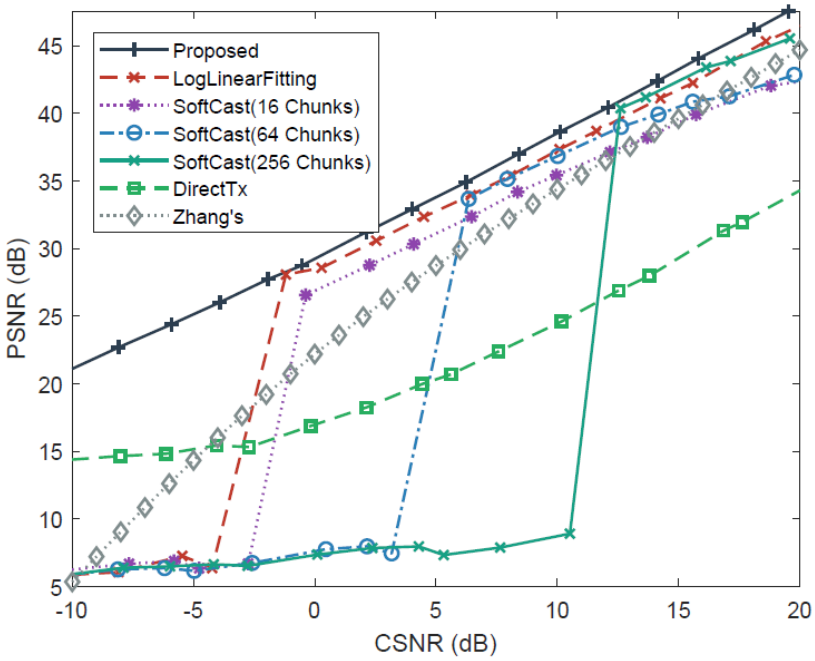


FIGURE 5.16 : Evolution de la qualité vidéo reçue (PSNR) en fonction de la qualité du canal de transmission (CSNR) pour la séquence CIF *Hall*. Figure issue de [131]. Les courbes LogLinearFitting et Zhang's réfèrent respectivement à [112] et à [123].

Compte tenu des résultats subjectifs obtenus et de la corrélation importante existante entre les MOS et la métrique SSIM, il serait également judicieux de trouver un modèle théorique d'évaluation de la qualité de bout en bout pour cette métrique. Des pistes de travail publiées par Horé *et al.* [45] pourraient représenter un début de réflexion.

De plus, afin d'améliorer l'efficacité de codage de SoftCast, il serait intéressant de recourir à des techniques de prétraitement permettant d'éliminer des informations non perceptibles à l'intérieur de la vidéo. Le pré-filtrage perceptuel indépendant du schéma de codage vidéo, permet de préserver les contours et textures tout en éliminant les informations visuelles non

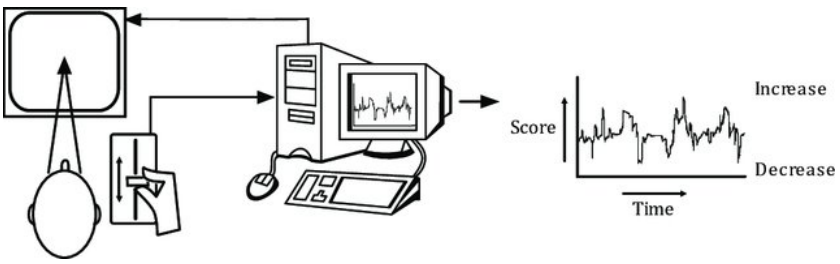


FIGURE 5.17 : Illustration du processus d'évaluation en continue de la qualité vidéo (SSCQE). Figure issue de [9].

pertinentes. Il est bien connu que plusieurs propriétés du système visuel humain (SVH) telles que la fonction de sensibilité au contraste peuvent être décrites dans le domaine fréquentiel. Puisque SoftCast fait appel à une transformée DCT pleine image, il devient envisageable d'intégrer des critères psychovisuels dans le schéma SoftCast. A notre connaissance, ceci n'a jamais été proposé et constitue donc une originalité. Il serait par conséquent intéressant d'introduire par exemple une pondération psychovisuelle des chunks (3D-CSF, JND : Just Noticeable Difference) avant d'effectuer les étapes successives de sélection, d'ordonnancement et d'allocation de puissance. Ceci permettrait de prendre en compte l'importance perceptuelle des chunks dans l'algorithme SoftCast pour optimiser la qualité de reconstruction. L'impact du pré-filtrage sur les courbes débit-distorsion (nombre de chunks supprimés/qualité) pourrait ainsi être étudié.

Enfin, pour assurer la diffusion à grande échelle du schéma SoftCast en assurant la continuité entre le "monde" LVC (sans-fil) et les réseaux traditionnels, il serait judicieux de considérer des techniques de transcodage SoftCast⇆H.264/AVC ou HEVC. En effet, aujourd'hui aucun équipement (téléphone, télévision, ordinateur, etc.) n'est capable de lire un flux issu de SoftCast. Des solutions de transcodage sont donc indispensables. Une solution directe consiste à effectuer un décodage complet suivi d'un ré-encodage complet (solution dite Full Decode-Full Recode, FDFR). Cette solution, quoiqu'optimale, est généralement complexe et des solutions alternatives de complexité réduite sont généralement souhaitables. Pour cela, il serait judicieux de mutualiser les blocs de traitement communs aux deux systèmes de compression. Dans le cas présent, SoftCast et les standards de compression vidéo actuels utilisent tous les deux une transformée DCT (ou une version dérivée), sur l'image entière pour SoftCast et par blocs de pixels pour les codecs standards. Il est donc à priori possible de proposer des solutions de transcodage directement dans le domaine transformé en passant d'une DCT à une autre.

Production scientifique

Revues internationales avec comité de lecture :

1. A. Trioux, M. Gharbi, F.-X. Coudoux, P. Corlay, “A Generalized Approach for the Evaluation of the End-to-End Performances of SoftCast-based Linear Video Delivery Schemes”, Signal Processing : Image Communication, to be submitted (Feb. 2020).
2. A. Trioux, F.-X. Coudoux, P. Corlay, M. Gharbi, “ Temporal Information based GoP Adaptation for Linear Delivery Schemes ”, Signal Processing : Image Communication, Volume 82, 2020, 115734, ISSN 0923-5965.

   Lien : http://www.sciencedirect.com/science/article/pii/S0923596519302577.

Conférences internationales avec comité de lecture :

1. A. Trioux, F.-X. Coudoux, P. Corlay, M. Gharbi, “A Comparative Preprocessing Study for SoftCast Video Transmission,” in IEEE 9th International Symposium on Signal, Image, Video and Communications (ISIVC), Rabat, Morocco, Nov. 2018, pp. 54-59. Lien : https://ieeexplore.ieee.org/document/8709171.
2. A. Trioux, F.-X. Coudoux, P. Corlay, M. Gharbi, “A Reduced Complexity/Side Information Preprocessing Method For High Quality SoftCast-Based Video Delivery,” in IEEE 8th European Workshop on Visual Information Processing (EUVIP), Roma, Italy, Oct. 2019, pp. 205-210. Lien : https://ieeexplore.ieee.org/document/8946224.

Conférences nationales avec comité de lecture :

1. A. Trioux, F.-X. Coudoux, P. Corlay, M. Gharbi, “Etude de l’influence du contenu vidéo dans une transmission Pseudo-Analogique de type SoftCast,” in 20 ème édition du colloque COmpression et REprésentation des Signaux Audiovisuels (CORESA), Poitiers, Nov. 2018, 3rd best paper. Lien : https://hal-uphf.archives-ouvertes.fr/hal-02423790.
2. A. Trioux, F.-X. Coudoux, P. Corlay, M. Gharbi, “Méthode de prétraitement pour les systèmes de codage vidéo linéaire basé SoftCast,” in 27 ème édition du colloque GRETSI, Lille, Aug. 2019. Lien : https://hal-uphf.archives-ouvertes.fr/hal-02423792.

3. A. Trioux, F.-X. Coudoux, P. Corlay, M. Gharbi, “Nouvelles méthodes de codage vidéo linéaire pour les transmissions vidéo sans-fil,” in 21 ème édition des Journées Nationales du Réseau Doctoral en Micro-nanoélectronique (JNRDM’2019), Montpellier, Jun. 2019, ISSN 2496-0160. Lien : https://jnrdm2019.sciencesconf.org/data/pages/book_jnrdm2019_fr_1.pdf.

Communications sans actes :

1. A. Trioux, F.-X. Coudoux, P. Corlay, M. Gharbi, “Adaptive GoP for Broadcast Linear Video Coding Under Bandwidth Constraints”, Journée du club EEA, Valenciennes, Nov. 2017.
2. A. Trioux, F.-X. Coudoux, P. Corlay, M. Gharbi, “Modeling of the End-to-End Performances for Soft Video Delivery Under Bandwidth Constraints”, Wallers-Arenberg, Mardi des chercheurs, Mar. 2018.

Annexe A

Démonstration allocation de puissance quasi-optimale

Rappelons ici que : $\mathbf{X}_i[j]$ représente l'élément de la $i^{ème}$ ligne et la $j^{ème}$ colonne de la matrice $\mathbf{X}$, de taille $N \times nb_c \cdot nb_r$. $\mathbf{Y}$, également de taille $N \times nb_c \cdot nb_r$, est constituée des slices correspondantes après mise à l'échelle et transformée de Hadamard. Si le CR<1, $\mathbf{X}$ et $\mathbf{Y}$ seront de dimension $K \times nb_c \cdot nb_r$.

La valeur reçue $\hat{\mathbf{Y}}_i[j]$, après traversée du canal AWGN, vaut $\hat{\mathbf{Y}}_i[j] = \mathbf{Y}_i[j] + n_i[j]$ où $n_i[j]$ représente le bruit additif gaussien centré de puissance $\sigma^2_{n,i}$. Le bruit est ici supposé stationnaire donc $\sigma^2_{n,i} = \sigma$.

Le récepteur SoftCast décode (dans le processus d'allocation de puissance quasi-optimale, le récepteur ZF est utilisé) :

$$\hat{\mathbf{X}}_i[j] = \frac{\hat{\mathbf{Y}}_i[j]}{g_i} = \mathbf{X}_i[j] + \frac{n}{g_i}.$$

L'erreur quadratique moyenne attendue est dans ce cas :

$$err = \mathrm{E}\left[\sum_i \left(\hat{\mathbf{X}}_i - \mathbf{X}_i\right)^2\right] = \sum_i \frac{E\left[n^2\right]}{g_i^2} = N \sum_i \frac{\sigma^2}{g_i^2}$$

Le but est de minimiser cette erreur err. Posons $\lambda_i = E\left[\mathbf{X}_i^2\right]$, l'énergie du $i^{ème}$ chunk, $\mu_i = E\left[\mathbf{Y}_i^2\right]$ sa version analogue après mise à l'échelle (scaling) et P la puissance disponible à l'émission.

Le problème consiste à minimiser l'erreur err comme suit :

$$\min err = \sigma^2 \sum_i \frac{\lambda_i}{\mu_i}, \text{ s.t. : } \sum_i \mu_i \leq P \text{ avec } \mu_i \geq 0$$

A. DÉMONSTRATION ALLOCATION DE PUISSANCE QUASI-OPTIMALE

C'est un problème lagrangien qui peut être réécrit sous la forme :

$$\mathcal{L} = \sigma^2 \sum_{i=1}^{N} \frac{\lambda_i}{\mu_i} + C\left(\sum_{i=1}^{N} \mu_i - P\right),$$

où C est le multiplicateur de Lagrange.

En annulant respectivement la dérivée du lagrangien $\mathcal{L}$ par rapport à μ_i, et C nous obtenons :

$$\begin{aligned}
\sqrt{C} &= \sum_i \frac{\sqrt{\lambda_i \sigma^2}}{P} \\
\mu_i &= \sqrt{\frac{\lambda_i \sigma^2}{C}} = P \frac{\sqrt{\lambda_i}}{\sum_i \sqrt{\lambda_i}} \\
g_i &= \sqrt{\frac{\mu_i}{\lambda_i}} = \sqrt{\frac{P}{\sqrt{\lambda_i} \sum_i \sqrt{\lambda_i}}}
\end{aligned}$$

Annexe B

Démonstrations modèles théoriques

Démonstration. Dans les cas où la bande passante est limitée, la distorsion totale pour la version approximée du modèle SoftCast(LLSE) est définie par (voir aussi (2.18) et (2.20)) :

$$\begin{aligned} \mathrm{D}_{\mathrm{T[LLSE/CB]^*}} &= \sum_{i=1}^{K} D_{i[\mathrm{ZF/CB}]} \cdot \frac{1}{1+\frac{K\sigma_n^2}{P}} \\ &\quad + \sum_{j=K+1}^{N} \lambda_j. \end{aligned} \tag{B.1}$$

En utilisant les équations du CSNR (2.3) et du PSNR (2.4), nous obtenons :

$$\begin{aligned} \mathrm{D}_{\mathrm{[LLSE/CB]^*}} &= \sum_{i=1}^{K} D_{i[\mathrm{ZF/CB}]} \cdot \frac{1}{1+\frac{1}{\mathrm{CSNR}_{lin}}} \\ &\quad + \sum_{j=K+1}^{N} \lambda_j. \end{aligned} \tag{B.2}$$

En utilisant la propriété : $\log_{10}(a+b) = \log_{10}(a)+\log_{10}(1+\frac{b}{a})$, où $a = \sum_{i=1}^{K} D_{i[\mathrm{ZF/CB}]} \cdot \frac{1}{1+\frac{1}{\mathrm{CSNR}_{lin}}}$ et $b = \sum_{j=K+1}^{N} \lambda_j$, et en rappelant que $D_{i[\mathrm{ZF/CB}]} = \frac{\sigma_n^2}{P}\left(\sum_{i=1}^{K}\sqrt{\lambda_i}\right)^2$ nous obtenons :

$$\begin{aligned} \mathrm{PSNR} &= 10\log_{10}\left(\frac{255^2 \cdot N}{\mathrm{D}_{\mathrm{[LLSE/CB]}}}\right), \\ &= c - 10\log_{10}\left(\left(\frac{1}{\mathrm{CSNR}_{lin}+1}\right)\cdot\frac{1}{NK}\left(\sum_{i=1}^{K}\sqrt{\lambda_i}\right)^2\right) \\ &\quad - 10\log_{10}\left(1+\frac{(\mathrm{CSNR}_{lin}+1)\cdot E_d}{{H_t}^2}\right), \end{aligned} \tag{B.3}$$

Par conséquent, l'équation peut être réécrite sous la forme :

$$\begin{aligned}\mathrm{PSNR} &= c + 10\log_{10}\left(\mathrm{CSNR}_{lin}+1\right) - 20\log_{10}(H_t) \\ &\quad - 10\log_{10}\left(1+\frac{(\mathrm{CSNR}_{lin}+1)\cdot E_d}{{H_t}^2}\right), \\ &= c + \mathrm{CSNR} + 10\log_{10}\left(1+\frac{1}{\mathrm{CSNR}_{lin}}\right) \\ &\quad - 20\log_{10}\left(H_t\right) \\ &\quad - 10\log_{10}\left(1+\frac{(\mathrm{CSNR}_{lin}+1)\cdot E_d}{{H_t}^2}\right), \\ &= \mathrm{PSNR}_{\mathrm{dB[LLSE/CB]^*}},\end{aligned} \tag{B.4}$$

où $10\log_{10}\left(1+\frac{1}{\mathrm{CSNR}_{lin}}\right)$ peut être dénoté par le terme G_{LLSE} pour faciliter la lecture.

□

Démonstration. La démonstration de (2.42) est directe si l'on considère que $\frac{\bar{P}+\sigma_n^2}{\sigma_n^2}$ peut être vu comme $\tilde{\mathrm{CSNR}}_{lin} = \mathrm{CSNR}_{lin} + 1$. En effet, en insérant $\tilde{\mathrm{CSNR}}_{lin}$ dans (2.8), en remplaçant K par ℓ et en utilisant la propriété : $\log_{10}(a+b) = \log_{10}(a) + \log_{10}(1+\frac{b}{a})$, nous obtenons facilement :

$$\begin{aligned}\mathrm{PSNR} &= 10\log_{10}\left(\frac{255^2\cdot N}{D_i+D_j}\right), \\ &= c - 10\log_{10}\left(1+\frac{D_j}{D_i}\right) + 10\log_{10}\left(\tilde{\mathrm{CSNR}}_{lin}\right) \\ &\quad - 10\log_{10}\left(\frac{1}{N\ell}\left(\sum_{i=1}^{\ell}\sqrt{\lambda_i}\right)^2\right), \\ &= c + \mathrm{CSNR} + 10\log_{10}\left(1+\frac{1}{\mathrm{CSNR}_{lin}}\right) \\ &\quad - 20\log_{10}\left(H_{t2}\right) - 10\log_{10}\left(1+\frac{\tilde{\mathrm{CSNR}}_{lin}\cdot E_{d2}}{{H_{t2}}^2}\right), \\ &= c + \mathrm{CSNR} + G_{\mathrm{LLSE}} \\ &\quad - 20\log_{10}\left(H_{t2}\right) \\ &\quad - 10\log_{10}\left(1+\frac{(\mathrm{CSNR}_{lin}+1)\cdot E_{d2}}{{H_{t2}}^2}\right), \\ &= \mathrm{PSNR}_{\mathrm{[OPA\text{-}LLSE]}}.\end{aligned} \tag{B.5}$$

□

Annexe C

Matériels additionnels tests subjectifs

Feuille d'information relative au RGPD

Afin d'effectuer l'analyse statistique des résultats des tests de qualité subjective, Telecom Paris demande aux participants d'indiquer leur sexe et âge.

Les données enregistrées sont conservées pendant trois mois et sont accessibles au personnel en charge du projet scientifique (équipe MM, département IDS).

Pour exercer vos droits Informatique et Libertés et pour toute information sur ce dispositif, contactez notre délégué à la protection des données (DPO) en écrivant à Thierry Oddon <thierry.oddon@telecom-paris.fr> ou à l'adresse postale suivante : 46 rue Barrault, Paris 13.

Le règlement général sur la protection des données (RGPD), qui entre en application le 25 mai 2018, impose une information concise, transparente, compréhensible et aisément accessible des personnes concernées. Cette obligation de transparence est définie aux articles 12, 13 et 14 du RGPD. (https://www.cnil.fr/fr/conformite-rgpd-information-des-personnes-et-transparence)

- Identité et coordonnées de l'organisme (responsable du traitement de données) = Télécom Paris
- Finalités (à quoi vont servir les données collectées) ; Moyenne statistique
- Caractère obligatoire ou facultatif du recueil des données (ce qui suppose une réflexion en amont sur l'utilité de collecter ces données au vu de l'objectif poursuivi – principe de « minimisation » des données) et conséquences pour la personne en cas de non-fourniture des données : Obligatoire
- Destinataires ou catégories de destinataires des données (qui a besoin d'y accéder ou de les recevoir au vu des finalités définies) : Equipe Multimédia
- Durée de conservation des données (ou critères permettant de la déterminer) : 3 mois

- Droits d'opposition, d'accès, rectification, effacement , limitation, en s'adressant à RGPD@telecom-paris.fr
- Coordonnées du délégué à la protection des données de l'organisme, s'il a été désigné, ou d'un point de contact sur les questions de protection des données personnelles = Robert Malek, dpo@imt.fr
- Base juridique du traitement de données : consentement des personnes concernées
- Droit d'introduire une réclamation (plainte) auprès de la CNIL ;
- Intérêts légitimes poursuivis par le responsable du traitement ou par un tiers (exemple : prévention de la fraude) : recherche
- Le droit au retrait du consentement : à tout moment

Briefing form (DSIS)

Thank you for taking part in the experiment. The experiment will take approximately 28min. Please wear your glasses/lens if you have a corrected vision. Please read the following instruction carefully before starting the experiment.

The purpose of the experiment is to evaluate the performance of a new robust video transmission scheme over wireless networks namely, SoftCast. This test includes several test sequences. The results will help to analyze the artifacts caused by this new scheme and find solutions to reduce/cancel them.

In this experiment, you will be asked to evaluate short video sequences at different compression and channel quality levels (i.e., different impairment levels). For each stimulus, you will see two versions of the same video content : the reference video which is the video sequence with the highest quality and no distortion artifacts, followed by the test sequence which is the distorted version of the same video content.

Your task will be to evaluate the test sequence having in mind (relative to) the reference. The evaluation will be performed on the following scale (0 : worst score, 100 : best score).

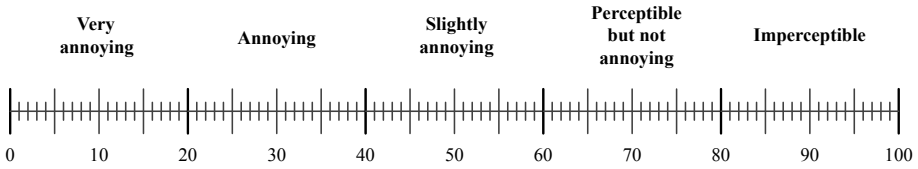


FIGURE C.1 : Illustration de l'échelle de notation continue utilisée (5 niveaux de dégradation).

- Very annoying : Very strong artifacts that are clearly visible everywhere and are very annoying.
- Annoying : Strong artifacts that are visible in the sequence at an annoying level.
- Slightly annoying : Noticeable artifacts at some regions of the videos.
- Perceptible but not annoying : Visible but acceptable artifacts at some regions of the videos, they are not overall annoying.
- Imperceptible : the artifacts are not visible to you.

Possible impairments you may see (but not limited to) :

- Blurring
- Loss of details
- Low-frequency-noise (snow effect)
- Bell artifact (temporal periodic distortion over the video)

In a single session you will need to evaluate 94 trials. The time pattern for the stimulus presentation is illustrated in the figure below.

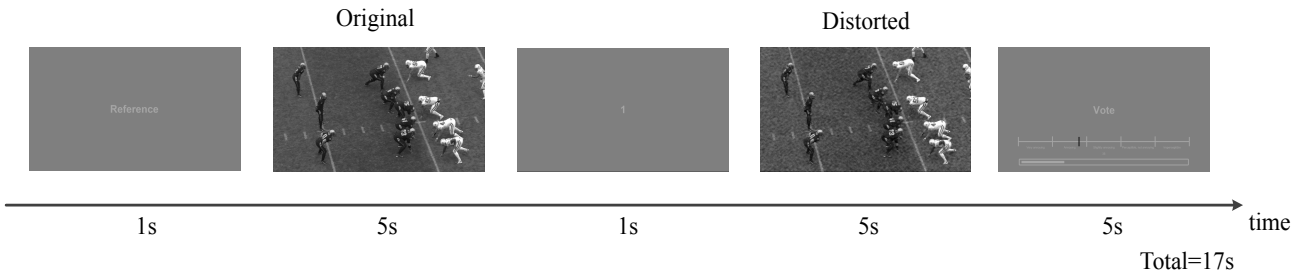


FIGURE C.2 : Illustration de la méthode DSIS Type I.

Each video sequence is 5 seconds long and played once. You should evaluate the second sequence only once you finished watching it, during the 5 seconds "Vote" screen. You will vote by clicking on the scale with the mouse at the desired score level.

The assessment begins with 10 examples (training video sequences) to understand and get familiar with the test procedure, the type of contents, as well as the impairments involved. Each quality level of the scale will be shown. The training sequences will be followed by the test, where the assessment is composed of a session of about 25min on average. You can take a break after each vote if necessary or stop the evaluation at any time for whatever reason.

Please note that none of the tasks is a test of your personal intelligence or ability. Remember that there is no right or wrong answer. The objective is to study human visual perception in general and not individual's abilities.

Thank you again for your participation !

Briefing form Forced PWC

Thank you for taking part in the experiment. The experiment will take approximately 28min. Please wear your glasses/lens if you have a corrected vision. Please read the following instruction carefully before starting the experiment.

The purpose of the experiment is to evaluate the performance of a new robust video transmission scheme over wireless networks namely, SoftCast. This test includes several test sequences. The results will help to analyze the artifacts caused by this new scheme and find solutions to reduce/cancel them.

In this experiment, you will be asked to evaluate short video sequences at different compression and channel quality levels (i.e., different impairment levels). For each stimulus, you will see two versions of the received video in a side-by-side fashion. These two versions represent the same video content, they only differ by the way they have been reconstructed at the received side (two different decoders are used).

Your task will be to choose the test sequence that you prefer by clicking on the Left (←) or Right (→) arrow of the Keyboard.

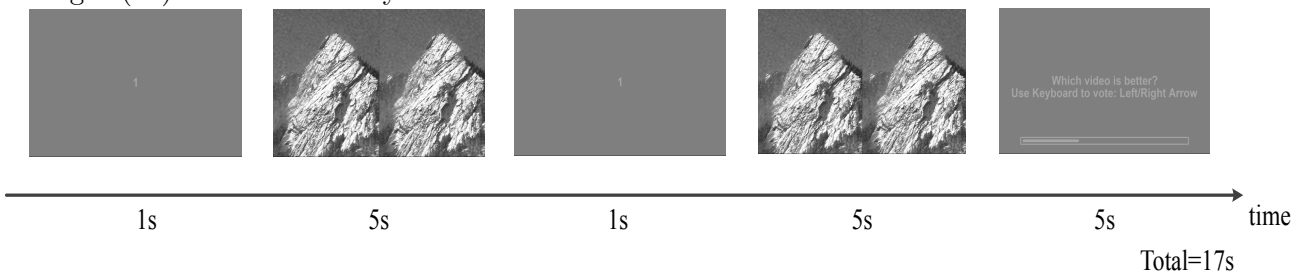


FIGURE C.3 : Illustration de la méthode de comparaison par paires à choix forcé.

Possible impairments you may see (but not limited to) :

- Blurring
- Loss of details
- Low-frequency-noise (snow effect)
- Pumping artifacts (periodic distortion over the video)

In a single session you will need to evaluate 98 trials. The time pattern for the stimulus presentation is illustrated in the figure above.

Each stimulus is 5 seconds long and played twice. You should select the best one according to your feelings during the (5 seconds) “Vote” screen.

The assessment begins with 3 examples (training video sequences) to understand and get familiar with the test procedure, the type of contents, as well as the impairments involved. The training sequences will be followed by the test, where the assessment is composed of a session of about 25min on average. You can take a break after each vote if necessary or stop the evaluation at any time for whatever reason.

Please note that none of the tasks is a test of your personal intelligence or ability. Remember that there is no right or wrong answer. The objective is to study human visual perception in general and not individual's abilities.

Thank you again for your participation !

Tableau C.1 : Liste des stimuli pour le test DSIS

Stimulus original	**Stimulus reconstruit après décodage**
BasketDrive2__ori2.avi	BasketDrive2__GoP__32__CR__0.25__LLSE__18dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__32__CR__0.25__LLSE__24dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__32__CR__0.25__LLSE__3dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__32__CR__0.25__LLSE__9dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__32__CR__1.00__LLSE__0dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__32__CR__1.00__LLSE__15dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__32__CR__1.00__LLSE__21dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__32__CR__1.00__LLSE__27dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__32__CR__1.00__LLSE__3dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__8__CR__0.25__LLSE__18dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__8__CR__0.25__LLSE__24dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__8__CR__0.25__LLSE__3dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__8__CR__0.25__LLSE__9dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__8__CR__1.00__LLSE__0dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__8__CR__1.00__LLSE__15dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__8__CR__1.00__LLSE__21dB.avi
BasketDrive2__ori2.avi	BasketDrive2__GoP__8__CR__1.00__LLSE__6dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__32__CR__0.25__LLSE__15dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__32__CR__0.25__LLSE__21dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__32__CR__0.25__LLSE__27dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__32__CR__0.25__LLSE__6dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__32__CR__1.00__LLSE__15dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__32__CR__1.00__LLSE__21dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__32__CR__1.00__LLSE__3dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__32__CR__1.00__LLSE__9dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__8__CR__0.25__LLSE__15dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__8__CR__0.25__LLSE__21dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__8__CR__0.25__LLSE__27dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__8__CR__0.25__LLSE__9dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__8__CR__1.00__LLSE__12dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__8__CR__1.00__LLSE__18dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__8__CR__1.00__LLSE__24dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__8__CR__1.00__LLSE__30dB.avi
Park__joy2__ori2.avi	Park__joy2__GoP__8__CR__1.00__LLSE__6dB.avi
ParkScene__ori2.avi	ParkScene__GoP__32__CR__0.25__LLSE__18dB.avi
ParkScene__ori2.avi	ParkScene__GoP__32__CR__0.25__LLSE__24dB.avi
ParkScene__ori2.avi	ParkScene__GoP__32__CR__0.25__LLSE__3dB.avi
ParkScene__ori2.avi	ParkScene__GoP__32__CR__0.25__LLSE__9dB.avi
ParkScene__ori2.avi	ParkScene__GoP__32__CR__1.00__LLSE__0dB.avi
ParkScene__ori2.avi	ParkScene__GoP__32__CR__1.00__LLSE__15dB.avi
ParkScene__ori2.avi	ParkScene__GoP__32__CR__1.00__LLSE__21dB.avi
ParkScene__ori2.avi	ParkScene__GoP__32__CR__1.00__LLSE__27dB.avi
ParkScene__ori2.avi	ParkScene__GoP__32__CR__1.00__LLSE__6dB.avi
ParkScene__ori2.avi	ParkScene__GoP__8__CR__0.25__LLSE__12dB.avi
ParkScene__ori2.avi	ParkScene__GoP__8__CR__0.25__LLSE__21dB.avi
ParkScene__ori2.avi	ParkScene__GoP__8__CR__0.25__LLSE__27dB.avi
ParkScene__ori2.avi	ParkScene__GoP__8__CR__0.25__LLSE__3dB.avi
ParkScene__ori2.avi	ParkScene__GoP__8__CR__1.00__LLSE__0dB.avi
ParkScene__ori2.avi	ParkScene__GoP__8__CR__1.00__LLSE__18dB.avi
ParkScene__ori2.avi	ParkScene__GoP__8__CR__1.00__LLSE__24dB.avi
ParkScene__ori2.avi	ParkScene__GoP__8__CR__1.00__LLSE__9dB.avi

Snow_mnt_ori.avi	Snow_mnt_GoP_32_CR_0.25_LLSE_15dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_32_CR_0.25_LLSE_21dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_32_CR_0.25_LLSE_3dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_32_CR_0.25_LLSE_9dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_32_CR_1.00_LLSE_0dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_32_CR_1.00_LLSE_12dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_32_CR_1.00_LLSE_18dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_32_CR_1.00_LLSE_27dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_32_CR_1.00_LLSE_6dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_8_CR_0.25_LLSE_15dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_8_CR_0.25_LLSE_21dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_8_CR_0.25_LLSE_30dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_8_CR_0.25_LLSE_6dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_8_CR_1.00_LLSE_18dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_8_CR_1.00_LLSE_24dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_8_CR_1.00_LLSE_6dB.avi
Snow_mnt_ori.avi	Snow_mnt_GoP_8_CR_1.00_LLSE_9dB.avi
Tractor_ori2.avi	Tractor_GoP_32_CR_0.25_LLSE_15dB.avi
Tractor_ori2.avi	Tractor_GoP_32_CR_0.25_LLSE_21dB.avi
Tractor_ori2.avi	Tractor_GoP_32_CR_0.25_LLSE_3dB.avi
Tractor_ori2.avi	Tractor_GoP_32_CR_0.25_LLSE_6dB.avi
Tractor_ori2.avi	Tractor_GoP_32_CR_1.00_LLSE_0dB.avi
Tractor_ori2.avi	Tractor_GoP_32_CR_1.00_LLSE_15dB.avi
Tractor_ori2.avi	Tractor_GoP_32_CR_1.00_LLSE_24dB.avi
Tractor_ori2.avi	Tractor_GoP_32_CR_1.00_LLSE_6dB.avi
Tractor_ori2.avi	Tractor_GoP_32_CR_1.00_LLSE_9dB.avi
Tractor_ori2.avi	Tractor_GoP_8_CR_0.25_LLSE_15dB.avi
Tractor_ori2.avi	Tractor_GoP_8_CR_0.25_LLSE_21dB.avi
Tractor_ori2.avi	Tractor_GoP_8_CR_0.25_LLSE_3dB.avi
Tractor_ori2.avi	Tractor_GoP_8_CR_0.25_LLSE_9dB.avi
Tractor_ori2.avi	Tractor_GoP_8_CR_1.00_LLSE_0dB.avi
Tractor_ori2.avi	Tractor_GoP_8_CR_1.00_LLSE_12dB.avi
Tractor_ori2.avi	Tractor_GoP_8_CR_1.00_LLSE_18dB.avi
Tractor_ori2.avi	Tractor_GoP_8_CR_1.00_LLSE_6dB.avi

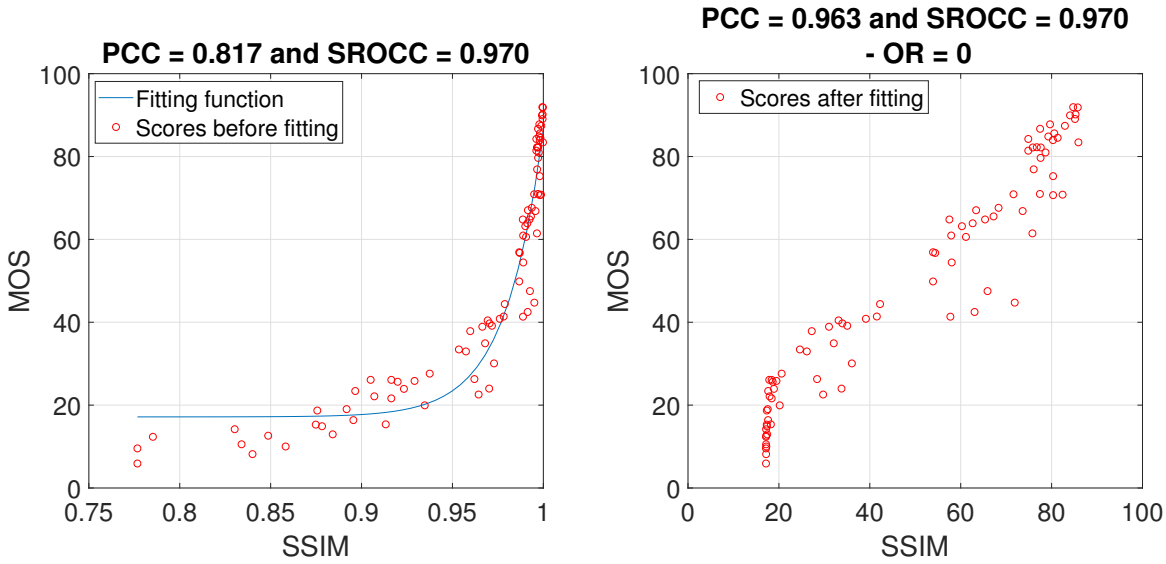


FIGURE C.4 : Illustration du scatter plot MOS/SSIM. Gauche : Avant régression non linéaire. Droite : Après régression non linéaire.

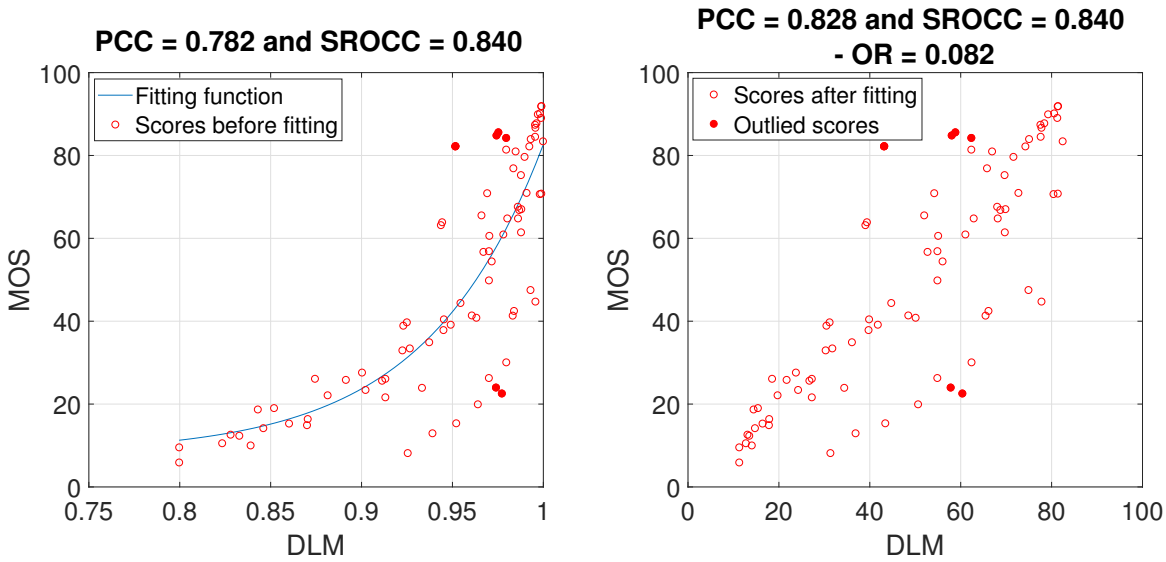


FIGURE C.5 : Illustration du scatter plot MOS/DLM. Gauche : Avant régression non linéaire. Droite : Après régression non linéaire.

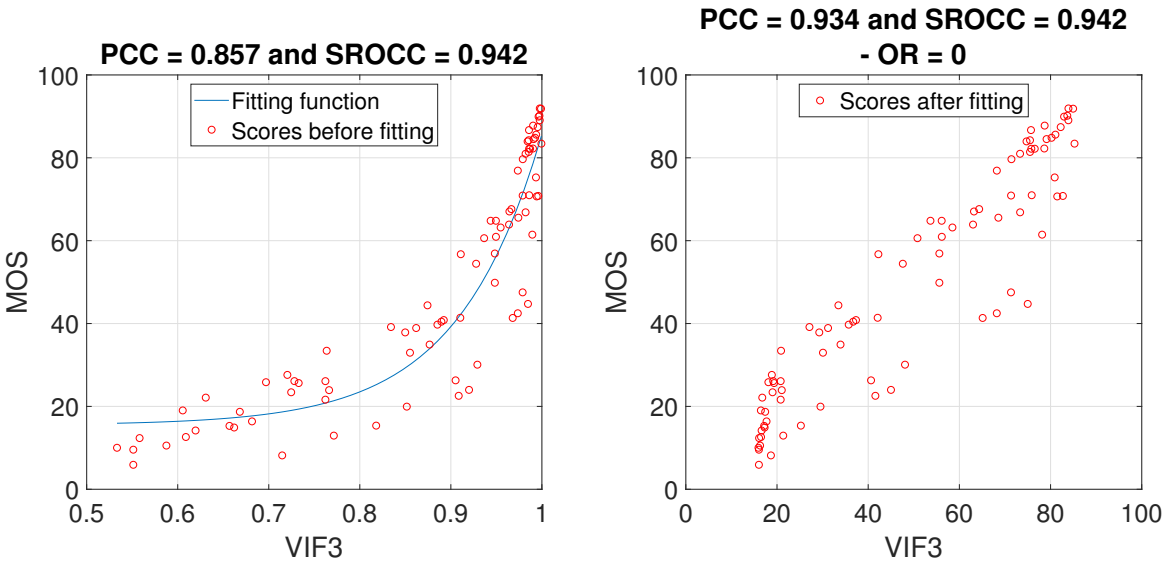


FIGURE C.6 : Illustration du scatter plot MOS/VIF. Gauche : Avant régression non linéaire. Droite : Après régression non linéaire.

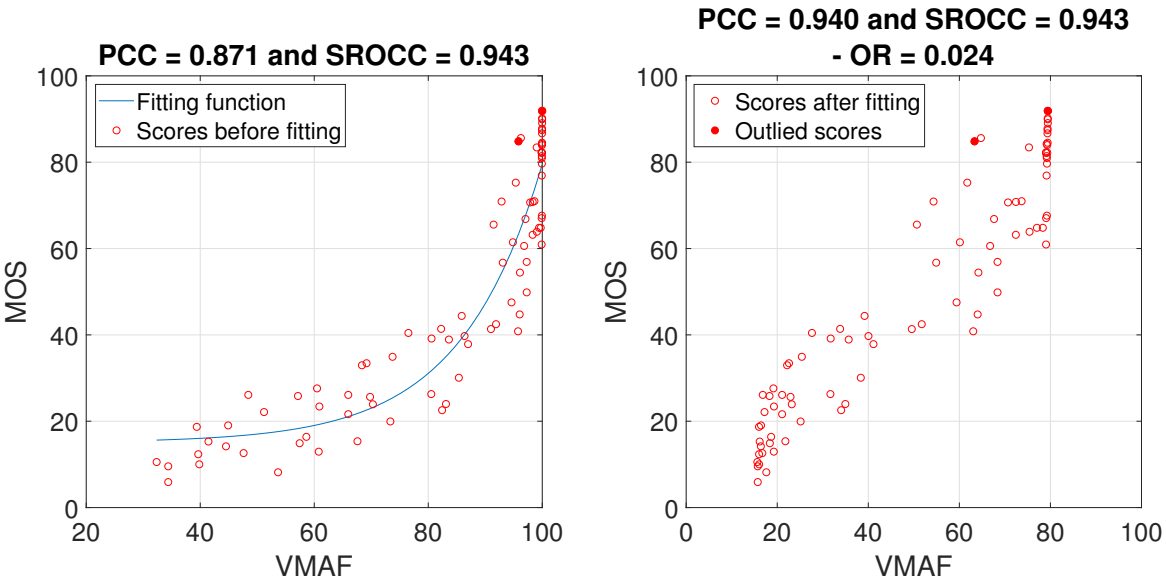


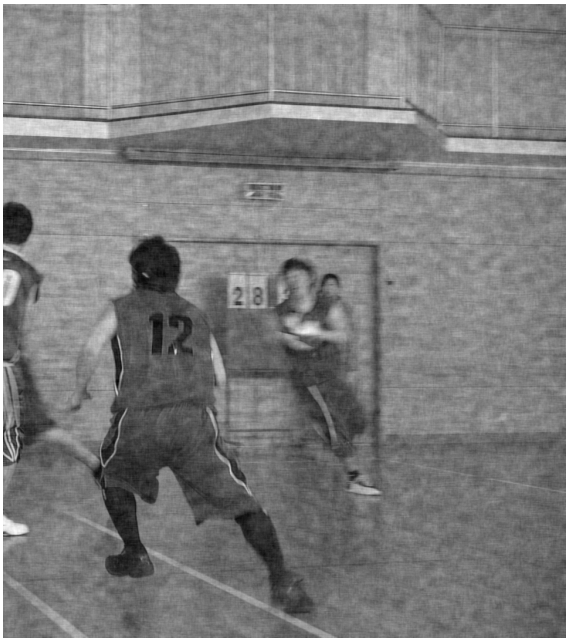
FIGURE C.7 : Illustration du scatter plot MOS/VMAF. Gauche : Avant régression non linéaire. Droite : Après régression non linéaire.

Tableau C.2 : Liste des stimuli pour les comparaisons par paires à choix forcé.

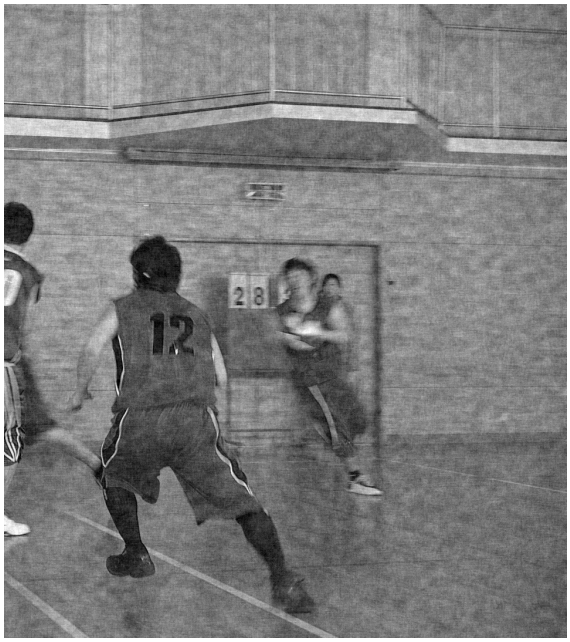
Stimulus gauche	**Stimulus droit**
BasketDrive2_GoP_32_CR_0.25_LLSE_0dB.avi	BasketDrive2_GoP_32_CR_0.25_ZF_0dB.avi
BasketDrive2_GoP_32_CR_0.25_LLSE_24dB.avi	BasketDrive2_GoP_32_CR_0.25_ZF_24dB.avi
BasketDrive2_GoP_32_CR_0.25_LLSE_3dB.avi	BasketDrive2_GoP_32_CR_0.25_ZF_0dB.avi
BasketDrive2_GoP_32_CR_0.25_LLSE_9dB.avi	BasketDrive2_GoP_32_CR_0.25_ZF_9dB.avi
BasketDrive2_GoP_32_CR_0.25_ZF_18dB.avi	BasketDrive2_GoP_32_CR_0.25_LLSE_18dB.avi
BasketDrive2_GoP_32_CR_0.25_ZF_3dB.avi	BasketDrive2_GoP_32_CR_0.25_LLSE_9dB.avi
BasketDrive2_GoP_32_CR_0.25_ZF_3dB.avi	BasketDrive2_GoP_32_CR_0.25_LLSE_3dB.avi
BasketDrive2_GoP_32_CR_1.00_LLSE_0dB.avi	BasketDrive2_GoP_32_CR_1.00_ZF_0dB.avi
BasketDrive2_GoP_32_CR_1.00_LLSE_15dB.avi	BasketDrive2_GoP_32_CR_1.00_ZF_15dB.avi
BasketDrive2_GoP_32_CR_1.00_LLSE_15dB.avi	BasketDrive2_GoP_32_CR_1.00_ZF_3dB.avi
BasketDrive2_GoP_32_CR_1.00_LLSE_9dB.avi	BasketDrive2_GoP_32_CR_1.00_ZF_9dB.avi
BasketDrive2_GoP_32_CR_1.00_ZF_0dB.avi	BasketDrive2_GoP_32_CR_1.00_LLSE_3dB.avi
BasketDrive2_GoP_32_CR_1.00_ZF_21dB.avi	BasketDrive2_GoP_32_CR_1.00_LLSE_21dB.avi
BasketDrive2_GoP_32_CR_1.00_ZF_3dB.avi	BasketDrive2_GoP_32_CR_1.00_LLSE_3dB.avi
Cactus2_GoP_32_CR_0.25_LLSE_0dB.avi	Cactus2_GoP_32_CR_0.25_ZF_0dB.avi
Cactus2_GoP_32_CR_0.25_LLSE_18dB.avi	Cactus2_GoP_32_CR_0.25_ZF_18dB.avi
Cactus2_GoP_32_CR_0.25_LLSE_6dB.avi	Cactus2_GoP_32_CR_0.25_ZF_6dB.avi
Cactus2_GoP_32_CR_0.25_ZF_12dB.avi	Cactus2_GoP_32_CR_0.25_LLSE_12dB.avi
Cactus2_GoP_32_CR_0.25_ZF_3dB.avi	Cactus2_GoP_32_CR_0.25_LLSE_3dB.avi
Cactus2_GoP_32_CR_1.00_LLSE_12dB.avi	Cactus2_GoP_32_CR_1.00_ZF_12dB.avi
Cactus2_GoP_32_CR_1.00_LLSE_3dB.avi	Cactus2_GoP_32_CR_1.00_ZF_3dB.avi
Cactus2_GoP_32_CR_1.00_ZF_0dB.avi	Cactus2_GoP_32_CR_1.00_LLSE_0dB.avi
Cactus2_GoP_32_CR_1.00_ZF_18dB.avi	Cactus2_GoP_32_CR_1.00_LLSE_18dB.avi
Cactus2_GoP_32_CR_1.00_ZF_6dB.avi	Cactus2_GoP_32_CR_1.00_LLSE_6dB.avi
CrowdRun2_GoP_32_CR_1.00_ZF_0dB.avi	CrowdRun2_GoP_32_CR_1.00_LLSE_0dB.avi
CrowdRun2_GoP_32_CR_0.25_LLSE_0dB.avi	CrowdRun2_GoP_32_CR_0.25_ZF_0dB.avi
CrowdRun2_GoP_32_CR_0.25_LLSE_18dB.avi	CrowdRun2_GoP_32_CR_0.25_ZF_18dB.avi
CrowdRun2_GoP_32_CR_0.25_LLSE_6dB.avi	CrowdRun2_GoP_32_CR_0.25_ZF_6dB.avi
CrowdRun2_GoP_32_CR_0.25_ZF_12dB.avi	CrowdRun2_GoP_32_CR_0.25_LLSE_12dB.avi
CrowdRun2_GoP_32_CR_0.25_ZF_3dB.avi	CrowdRun2_GoP_32_CR_0.25_LLSE_3dB.avi
CrowdRun2_GoP_32_CR_1.00_LLSE_3dB.avi	CrowdRun2_GoP_32_CR_1.00_ZF_3dB.avi
CrowdRun2_GoP_32_CR_1.00_LLSE_6dB.avi	CrowdRun2_GoP_32_CR_1.00_ZF_6dB.avi
CrowdRun2_GoP_32_CR_1.00_ZF_12dB.avi	CrowdRun2_GoP_32_CR_1.00_LLSE_12dB.avi
CrowdRun2_GoP_32_CR_1.00_ZF_18dB.avi	CrowdRun2_GoP_32_CR_1.00_LLSE_18dB.avi
Park_joy2_GoP_32_CR_0.25_LLSE_0dB.avi	Park_joy2_GoP_32_CR_0.25_ZF_0dB.avi
Park_joy2_GoP_32_CR_0.25_LLSE_15dB.avi	Park_joy2_GoP_32_CR_0.25_ZF_6dB.avi
Park_joy2_GoP_32_CR_0.25_LLSE_6dB.avi	Park_joy2_GoP_32_CR_0.25_ZF_6dB.avi
Park_joy2_GoP_32_CR_0.25_ZF_15dB.avi	Park_joy2_GoP_32_CR_0.25_LLSE_15dB.avi
Park_joy2_GoP_32_CR_0.25_ZF_21dB.avi	Park_joy2_GoP_32_CR_0.25_LLSE_21dB.avi
Park_joy2_GoP_32_CR_0.25_ZF_3dB.avi	Park_joy2_GoP_32_CR_0.25_LLSE_3dB.avi
Park_joy2_GoP_32_CR_1.00_LLSE_0dB.avi	Park_joy2_GoP_32_CR_1.00_ZF_0dB.avi
Park_joy2_GoP_32_CR_1.00_LLSE_21dB.avi	Park_joy2_GoP_32_CR_1.00_ZF_21dB.avi
Park_joy2_GoP_32_CR_1.00_LLSE_3dB.avi	Park_joy2_GoP_32_CR_1.00_ZF_0dB.avi
Park_joy2_GoP_32_CR_1.00_LLSE_9dB.avi	Park_joy2_GoP_32_CR_1.00_ZF_9dB.avi
Park_joy2_GoP_32_CR_1.00_ZF_15dB.avi	Park_joy2_GoP_32_CR_1.00_LLSE_15dB.avi
Park_joy2_GoP_32_CR_1.00_ZF_3dB.avi	Park_joy2_GoP_32_CR_1.00_LLSE_3dB.avi
Park_joy2_GoP_32_CR_1.00_ZF_3dB.avi	Park_joy2_GoP_32_CR_1.00_LLSE_9dB.avi
ParkScene_GoP_32_CR_0.25_LLSE_24dB.avi	ParkScene_GoP_32_CR_0.25_ZF_24dB.avi
ParkScene_GoP_32_CR_0.25_LLSE_9dB.avi	ParkScene_GoP_32_CR_0.25_ZF_3dB.avi
ParkScene_GoP_32_CR_0.25_LLSE_9dB.avi	ParkScene_GoP_32_CR_0.25_ZF_9dB.avi
ParkScene_GoP_32_CR_0.25_ZF_0dB.avi	ParkScene_GoP_32_CR_0.25_LLSE_3dB.avi

C. MATÉRIELS ADDITIONNELS TESTS SUBJECTIFS

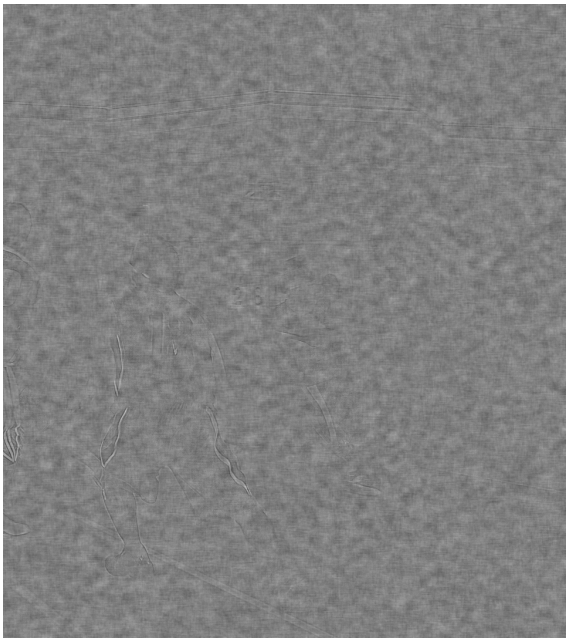
ParkScene_GoP_32_CR_0.25_ZF_0dB.avi	ParkScene_GoP_32_CR_0.25_LLSE_0dB.avi
ParkScene_GoP_32_CR_0.25_ZF_18dB.avi	ParkScene_GoP_32_CR_0.25_LLSE_18dB.avi
ParkScene_GoP_32_CR_0.25_ZF_3dB.avi	ParkScene_GoP_32_CR_0.25_LLSE_3dB.avi
ParkScene_GoP_32_CR_1.00_LLSE_15dB.avi	ParkScene_GoP_32_CR_1.00_ZF_15dB.avi
ParkScene_GoP_32_CR_1.00_LLSE_3dB.avi	ParkScene_GoP_32_CR_1.00_ZF_3dB.avi
ParkScene_GoP_32_CR_1.00_ZF_0dB.avi	ParkScene_GoP_32_CR_1.00_LLSE_0dB.avi
ParkScene_GoP_32_CR_1.00_ZF_0dB.avi	ParkScene_GoP_32_CR_1.00_LLSE_6dB.avi
ParkScene_GoP_32_CR_1.00_ZF_0dB.avi	ParkScene_GoP_32_CR_1.00_LLSE_3dB.avi
ParkScene_GoP_32_CR_1.00_ZF_21dB.avi	ParkScene_GoP_32_CR_1.00_LLSE_21dB.avi
ParkScene_GoP_32_CR_1.00_ZF_6dB.avi	ParkScene_GoP_32_CR_1.00_LLSE_6dB.avi
Snow_mnt_GoP_32_CR_0.25_LLSE_15dB.avi	Snow_mnt_GoP_32_CR_0.25_ZF_15dB.avi
Snow_mnt_GoP_32_CR_0.25_LLSE_3dB.avi	Snow_mnt_GoP_32_CR_0.25_ZF_3dB.avi
Snow_mnt_GoP_32_CR_0.25_ZF_21dB.avi	Snow_mnt_GoP_32_CR_0.25_LLSE_21dB.avi
Snow_mnt_GoP_32_CR_0.25_ZF_9dB.avi	Snow_mnt_GoP_32_CR_0.25_LLSE_9dB.avi
Snow_mnt_GoP_32_CR_1.00_LLSE_0dB.avi	Snow_mnt_GoP_32_CR_1.00_ZF_0dB.avi
Snow_mnt_GoP_32_CR_1.00_LLSE_18dB.avi	Snow_mnt_GoP_32_CR_1.00_ZF_18dB.avi
Snow_mnt_GoP_32_CR_1.00_ZF_12dB.avi	Snow_mnt_GoP_32_CR_1.00_LLSE_12dB.avi
Snow_mnt_GoP_32_CR_1.00_ZF_6dB.avi	Snow_mnt_GoP_32_CR_1.00_LLSE_6dB.avi
Tractor_GoP_32_CR_0.25_LLSE_15dB.avi	Tractor_GoP_32_CR_0.25_ZF_15dB.avi
Tractor_GoP_32_CR_0.25_LLSE_3dB.avi	Tractor_GoP_32_CR_0.25_ZF_3dB.avi
Tractor_GoP_32_CR_0.25_LLSE_6dB.avi	Tractor_GoP_32_CR_0.25_ZF_3dB.avi
Tractor_GoP_32_CR_0.25_LLSE_6dB.avi	Tractor_GoP_32_CR_0.25_ZF_6dB.avi
Tractor_GoP_32_CR_0.25_ZF_0dB.avi	Tractor_GoP_32_CR_0.25_LLSE_0dB.avi
Tractor_GoP_32_CR_0.25_ZF_21dB.avi	Tractor_GoP_32_CR_0.25_LLSE_21dB.avi
Tractor_GoP_32_CR_1.00_LLSE_15dB.avi	Tractor_GoP_32_CR_1.00_ZF_15dB.avi
Tractor_GoP_32_CR_1.00_LLSE_6dB.avi	Tractor_GoP_32_CR_1.00_ZF_0dB.avi
Tractor_GoP_32_CR_1.00_ZF_0dB.avi	Tractor_GoP_32_CR_1.00_LLSE_0dB.avi
Tractor_GoP_32_CR_1.00_ZF_0dB.avi	Tractor_GoP_32_CR_1.00_LLSE_3dB.avi
Tractor_GoP_32_CR_1.00_ZF_3dB.avi	Tractor_GoP_32_CR_1.00_LLSE_3dB.avi
Tractor_GoP_32_CR_1.00_ZF_6dB.avi	Tractor_GoP_32_CR_1.00_LLSE_6dB.avi
Tractor_GoP_32_CR_1.00_ZF_9dB.avi	Tractor_GoP_32_CR_1.00_LLSE_9dB.avi
West_GoP_32_CR_0.25_LLSE_0dB.avi	West_GoP_32_CR_0.25_ZF_0dB.avi
West_GoP_32_CR_0.25_LLSE_18dB.avi	West_GoP_32_CR_0.25_ZF_18dB.avi
West_GoP_32_CR_0.25_LLSE_6dB.avi	West_GoP_32_CR_0.25_ZF_6dB.avi
West_GoP_32_CR_0.25_ZF_12dB.avi	West_GoP_32_CR_0.25_LLSE_12dB.avi
West_GoP_32_CR_0.25_ZF_3dB.avi	West_GoP_32_CR_0.25_LLSE_3dB.avi
West_GoP_32_CR_1.00_LLSE_18dB.avi	West_GoP_32_CR_1.00_ZF_18dB.avi
West_GoP_32_CR_1.00_LLSE_3dB.avi	West_GoP_32_CR_1.00_ZF_3dB.avi
West_GoP_32_CR_1.00_ZF_0dB.avi	West_GoP_32_CR_1.00_LLSE_0dB.avi
West_GoP_32_CR_1.00_ZF_12dB.avi	West_GoP_32_CR_1.00_LLSE_12dB.avi
West_GoP_32_CR_1.00_ZF_6dB.avi	West_GoP_32_CR_1.00_LLSE_6dB.avi



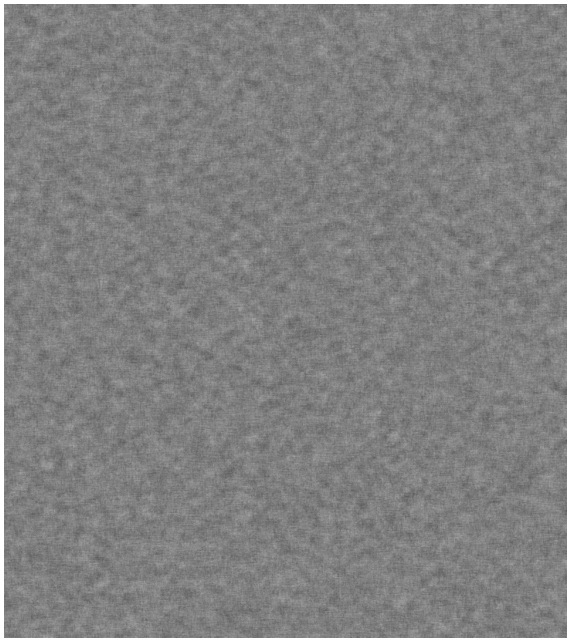
(a) Image reconstruite avec le LLSE



(b) Image reconstruite avec le ZF



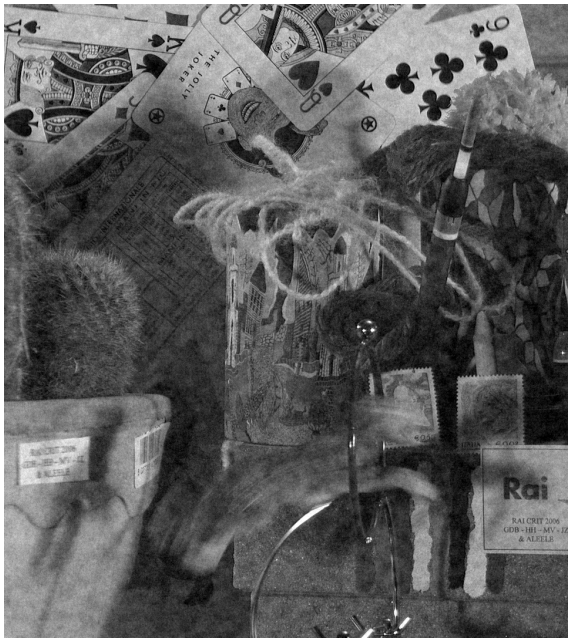
(c) Image d'erreur résultante du LLSE



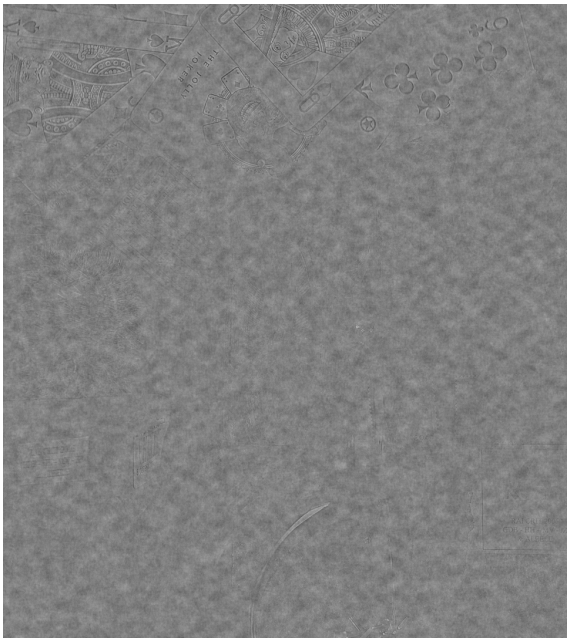
(d) Image d'erreur résultante du ZF

FIGURE C.8 : Illustration des images reconstruites pour la séquence *BasketBallDrive*. Configuration : Taille de GoP = 32, CR = 1, CSNR = 0dB, image n°125. a) Image reconstruite avec le LLSE , b) Image reconstruite avec le ZF, c) Image d'erreur résultante du LLSE, d) Image d'erreur résultante du ZF.

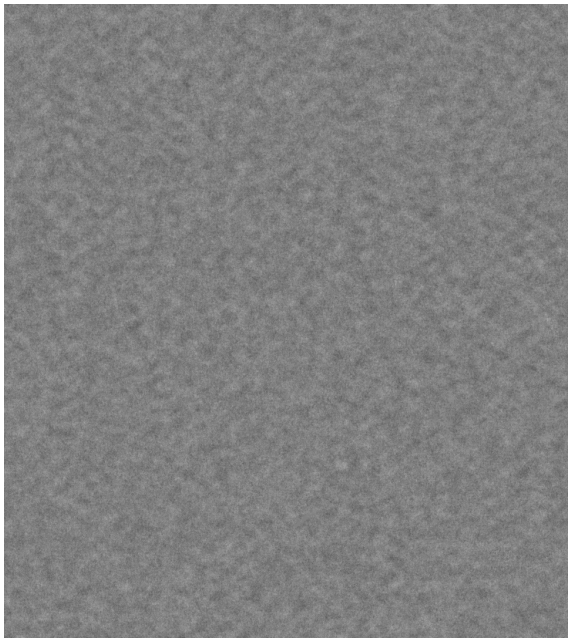
(a) Image reconstruite avec le LLSE



(b) Image reconstruite avec le ZF

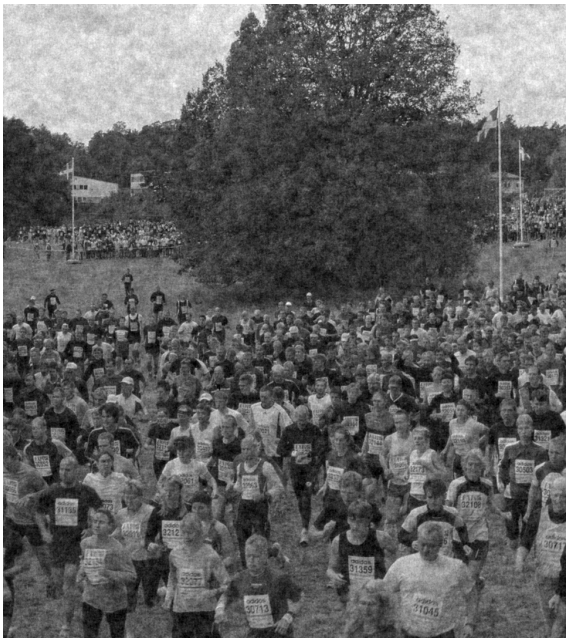


(c) Image d'erreur résultante du LLSE

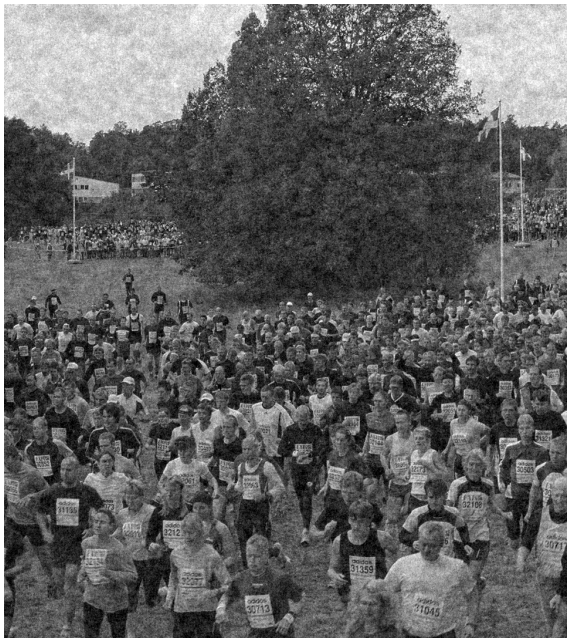


(d) Image d'erreur résultante du ZF

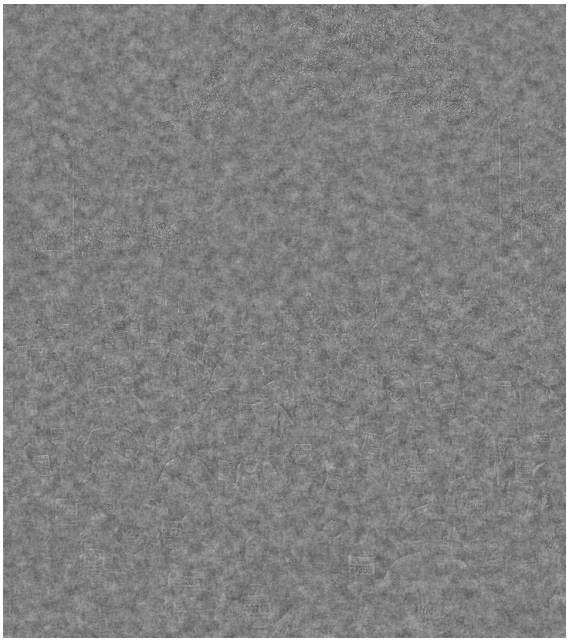
FIGURE C.9 : Illustration des images reconstruites pour la séquence *Cactus*. Configuration : Taille de GoP = 32, CR = 1, CSNR = 0dB, image n°125. a) Image reconstruite avec le LLSE , b) Image reconstruite avec le ZF, c) Image d'erreur résultante du LLSE, d) Image d'erreur résultante du ZF.



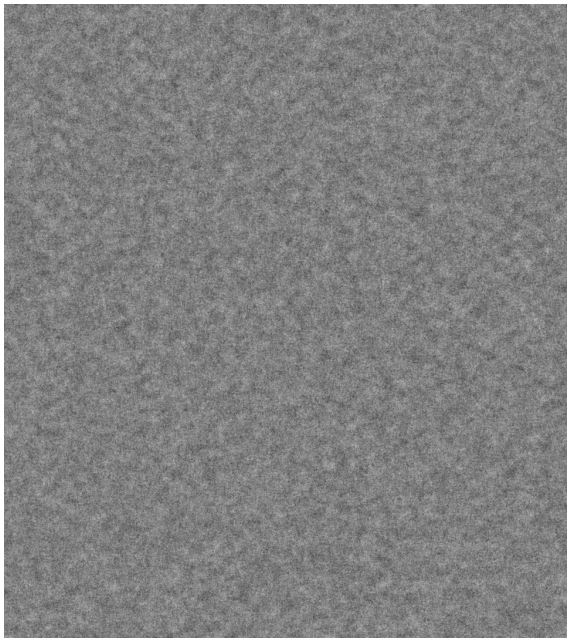
(a) Image reconstruite avec le LLSE



(b) Image reconstruite avec le ZF

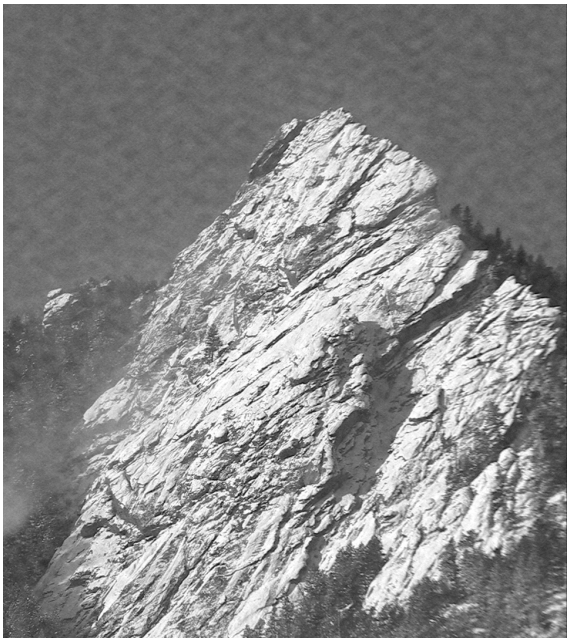


(c) Image d'erreur résultante du LLSE

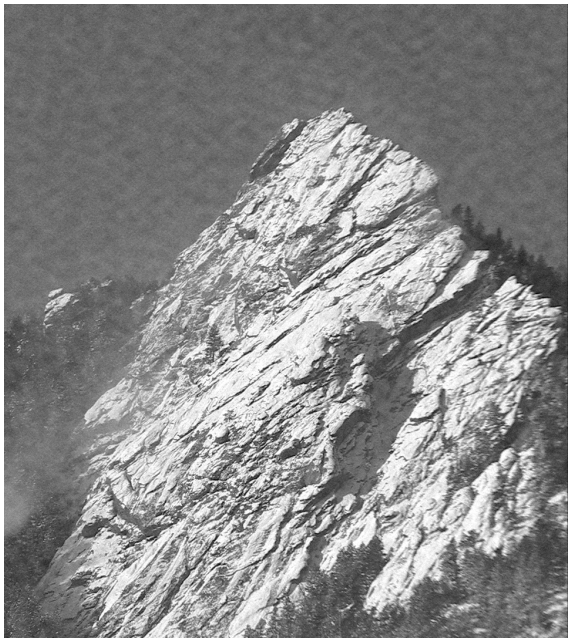


(d) Image d'erreur résultante du ZF

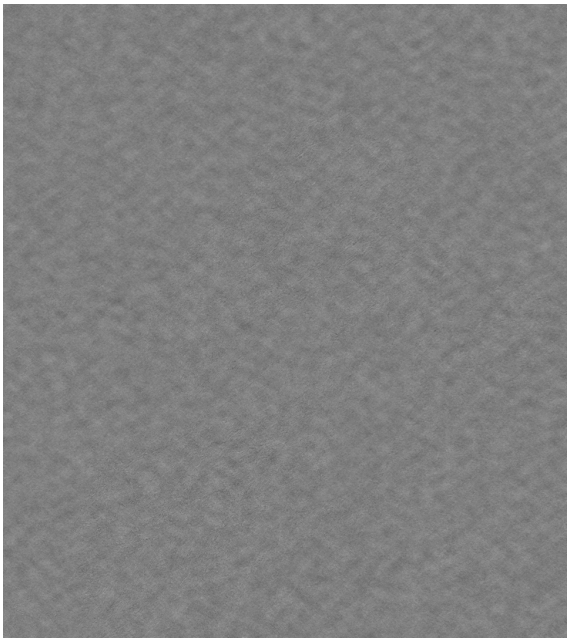
FIGURE C.10 : Illustration des images reconstruites pour la séquence *CrowdRun*. Configuration : Taille de GoP = 32, CR = 1, CSNR = 0dB, image n°125. a) Image reconstruite avec le LLSE , b) Image reconstruite avec le ZF, c) Image d'erreur résultante du LLSE, d) Image d'erreur résultante du ZF.



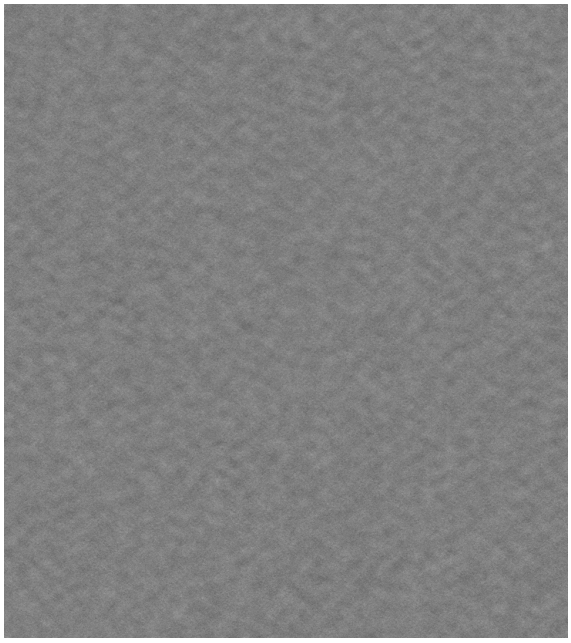
(a) Image reconstruite avec le LLSE



(b) Image reconstruite avec le ZF

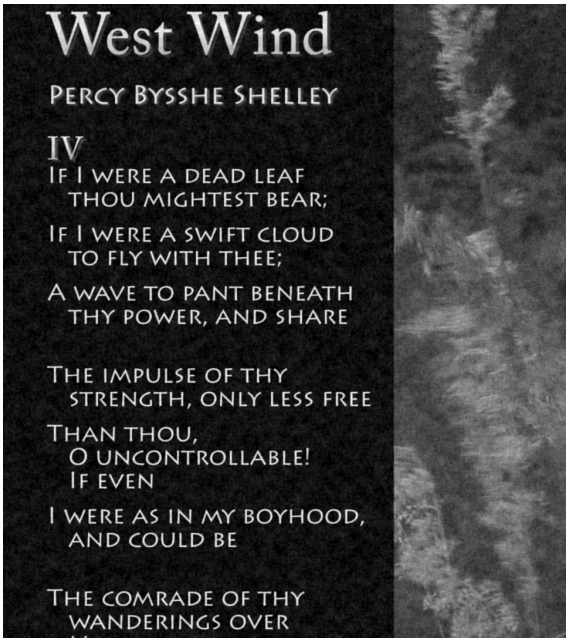


(c) Image d'erreur résultante du LLSE

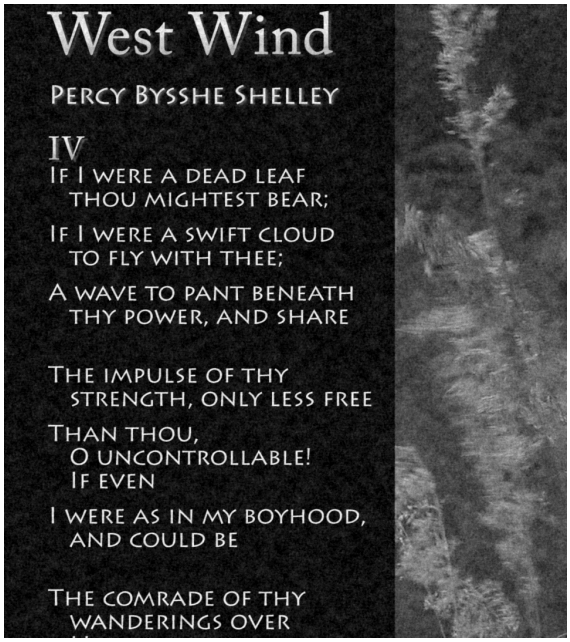


(d) Image d'erreur résultante du ZF

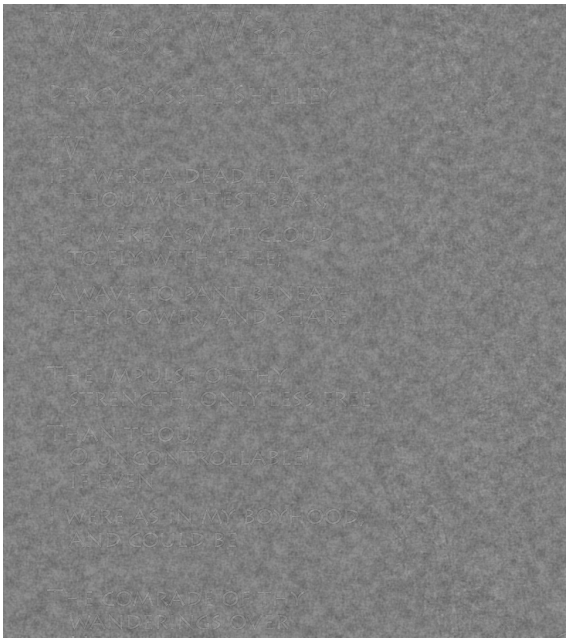
FIGURE C.11 : Illustration des images reconstruites pour la séquence *Snow Mountain*. Configuration : Taille de GoP = 32, CR = 1, CSNR = 0dB, image n°150. a) Image reconstruite avec le LLSE , b) Image reconstruite avec le ZF, c) Image d'erreur résultante du LLSE, d) Image d'erreur résultante du ZF.



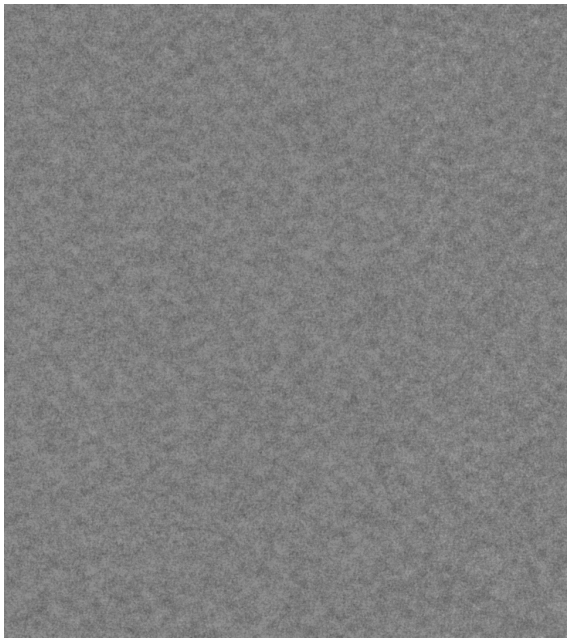
(a) Image reconstruite avec le LLSE



(b) Image reconstruite avec le ZF



(c) Image d'erreur résultante du LLSE



(d) Image d'erreur résultante du ZF

FIGURE C.12 : Illustration des images reconstruites pour la séquence *West*. Configuration : Taille de GoP = 32, CR = 1, CSNR = 0dB, image n°150. a) Image reconstruite avec le LLSE , b) Image reconstruite avec le ZF, c) Image d'erreur résultante du LLSE, d) Image d'erreur résultante du ZF.

Annexe D

Matériels additionnels algorithme adaptatif

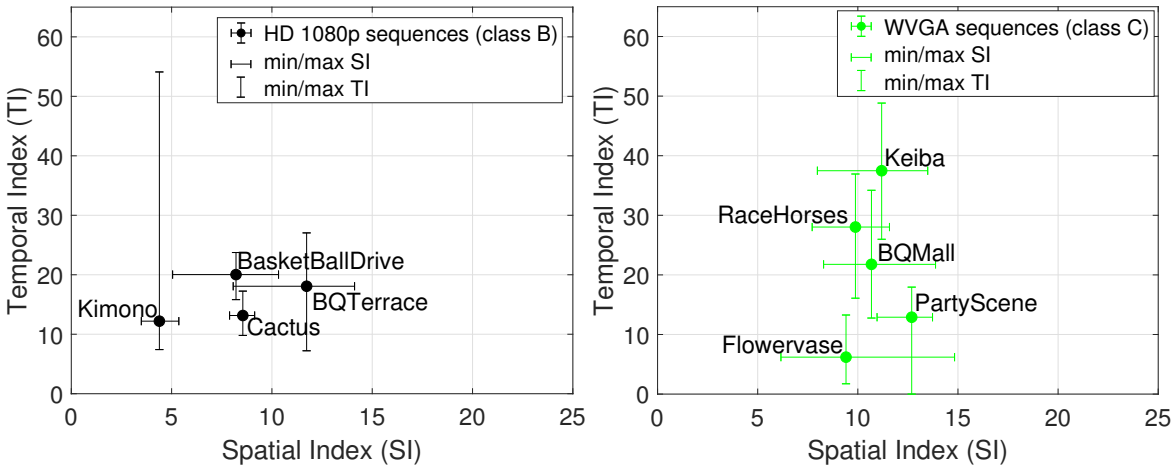


FIGURE D.1 : Illustration des index spatio-temporels (SI,TI) pour les séquences vidéos HD1080p (classe B) et WVGA (classe C) sélectionnées. Les points correspondent aux valeurs moyennes sur toute la séquence vidéo. Les barres verticales et horizontales représentent respectivement les valeurs min / max de l'index temporel et de l'index spatial. De haut en bas : séquences HD1080p (classe B), séquences WVGA (classe C).

Tableau D.1 : Tableau des scores PSNR et SSIM obtenus pour différentes tailles de GoP et différents CSNR avec CR = 1 (pas de compression) et CR = 0.25 (75 % des coefficients jetés). Séquences vidéos issues du JCT-VC : classe C (WVGA sequences, 832 × 480 pixels) [103].

Simulation Setup			CSNR(dB) 0		10		20	
			PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
GoP = 8	CR=1	*BQ Mall*	**28.38**	**0.756**	**36.94**	**0.938**	**46.40**	**0.992**
		Flower Vase	30.86	0.764	39.45	0.941	48.76	0.992
		Keiba	**27.49**	**0.710**	**36.32**	**0.925**	**45.84**	**0.990**
		PartyScene	26.63	0.741	34.89	0.934	44.42	0.991
		Racehorse	**27.68**	**0.713**	**36.00**	**0.919**	**45.50**	**0.989**
	CR=0.25	*BQ Mall*	**24.86**	**0.636**	**31.63**	**0.854**	**35.35**	**0.938**
		Flower Vase	27.27	0.654	34.10	0.870	37.99	0.962
		Keiba	**23.46**	**0.574**	**31.11**	**0.825**	**36.55**	**0.945**
		PartyScene	23.76	0.618	28.98	0.841	31.13	0.918
		Racehorse	**24.67**	**0.608**	**29.97**	**0.817**	**32.09**	**0.902**
GoP = 16	CR=1	*BQ Mall*	28.69	0.765	37.22	0.941	46.67	0.992
		Flower Vase	31.57	0.783	40.12	0.948	49.38	0.993
		Keiba	27.57	0.711	36.40	0.925	45.92	0.990
		PartyScene	**27.04**	**0.754**	**35.30**	**0.939**	**44.81**	**0.992**
		Racehorse	27.74	0.714	36.05	0.920	45.55	0.989
	CR=0.25	*BQ Mall*	25.180	0.648	31.912	0.860	35.579	0.939
		Flower Vase	28.05	0.679	34.77	0.883	38.56	0.965
		Keiba	23.58	0.577	31.19	0.827	36.57	0.945
		PartyScene	**24.175**	**0.635**	**29.418**	**0.851**	**31.621**	**0.924**
		Racehorse	24.77	0.611	29.99	0.817	32.04	0.900
GoP = 32	CR=1	*BQ Mall*	28.83	0.769	37.35	0.942	46.79	0.992
		Flower Vase	**32.02**	**0.794**	**40.55**	**0.951**	**49.77**	**0.993**
		Keiba	27.63	0.710	36.43	0.924	45.95	0.990
		PartyScene	27.29	0.760	35.53	0.941	45.04	0.992
		Racehorse	27.77	0.714	36.06	0.919	45.56	0.989
	CR=0.25	*BQ Mall*	25.35	0.654	32.02	0.863	35.59	0.939
		Flower Vase	**28.56**	**0.694**	**35.19**	**0.890**	**38.86**	**0.967**
		Keiba	23.64	0.576	31.23	0.826	36.58	0.945
		PartyScene	24.43	0.644	29.67	0.856	31.88	0.926
		Racehorse	24.81	0.611	29.99	0.817	32.00	0.899

Tableau D.2 : Tableau des scores PSNR et SSIM obtenus pour différentes tailles de GoP et différents CSNR avec CR = 1 (pas de compression) et CR = 0.25 (75 % des coefficients jetés). Séquences vidéos issues du JCT-VC : classe B (HD 1080p sequences, 1920 × 1080 pixels) [103].

Simulation Setup			CSNR(dB) 0		10		20	
			PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
GoP = 8	CR=1	*BasketBallDrive*	**30.40**	**0.781**	**38.91**	**0.949**	**48.24**	**0.993**
		BQ Terrace	27.88	0.723	36.40	0.931	45.88	0.991
		Cactus	29.56	0.776	38.09	0.947	47.49	0.993
		Kimono	33.98	0.894	42.68	0.979	51.61	0.997
	CR=0.25	*BasketBallDrive*	**26.93**	**0.672**	**33.77**	**0.861**	**37.64**	**0.930**
		BQ Terrace	24.40	0.590	31.21	0.823	34.92	0.913
		Cactus	25.98	0.654	32.87	0.857	36.59	0.927
		Kimono	29.95	0.807	37.69	0.937	42.52	0.969
GoP = 16	CR=1	*BasketBallDrive*	30.53	0.783	39.01	0.949	48.34	0.993
		BQ Terrace	**28.48**	**0.741**	**36.95**	**0.937**	**46.40**	**0.992**
		Cactus	30.25	0.794	38.73	0.953	48.07	0.994
		Kimono	**34.41**	**0.900**	**43.06**	**0.980**	**51.94**	**0.997**
	CR=0.25	*BasketBallDrive*	27.09	0.676	33.86	0.862	37.62	0.930
		BQ Terrace	**25.08**	**0.614**	**31.73**	**0.834**	**35.22**	**0.916**
		Cactus	26.81	0.683	33.44	0.868	36.86	0.930
		Kimono	**30.47**	**0.819**	**38.06**	**0.940**	**42.66**	**0.969**
GoP = 32	CR=1	*BasketBallDrive*	30.53	0.781	39.00	0.949	48.33	0.993
		BQ Terrace	28.80	0.749	37.23	0.939	46.67	0.992
		Cactus	**30.78**	**0.808**	**39.21**	**0.956**	**48.51**	**0.994**
		Kimono	34.60	0.902	43.22	0.980	52.07	0.997
	CR=0.25	*BasketBallDrive*	27.14	0.675	33.83	0.860	37.52	0.929
		BQ Terrace	25.47	0.628	31.96	0.840	35.26	0.917
		Cactus	**27.45**	**0.707**	**33.87**	**0.876**	**37.03**	**0.932**
		Kimono	30.72	0.825	38.22	0.941	42.70	0.969

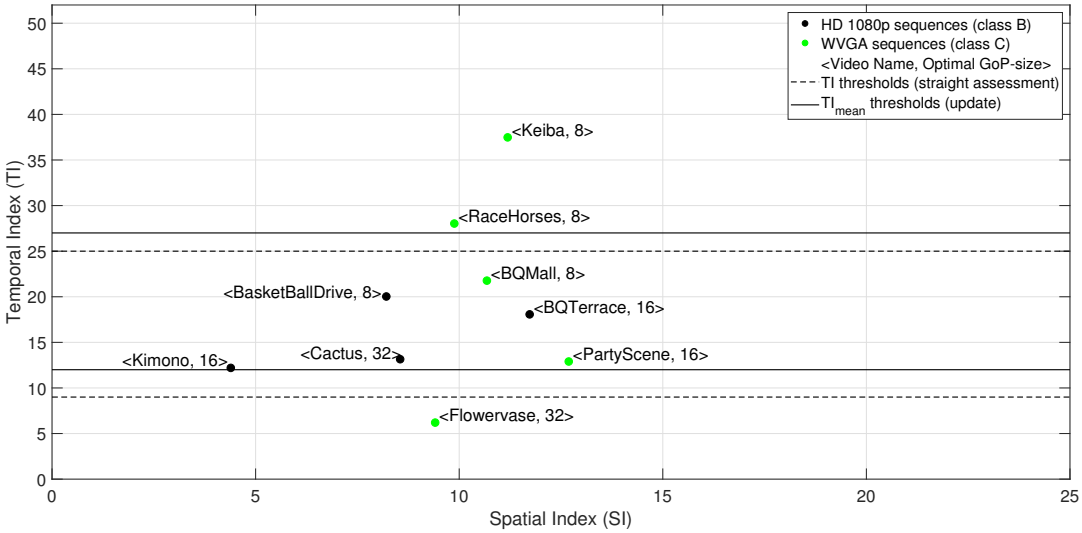


FIGURE D.2 : Illustration de la taille de GoP optimale par rapport aux index spatio-temporels pour les séquences vidéo WVGA et HD1080p sélectionnées. Les points verts et noirs correspondent respectivement aux valeurs moyennes des indices SI, TI pour les séquences WVGA (classe C) et HD1080p (classe B). L'étiquette associée à chaque point fait référence au couple de données suivantes : <Nom de la vidéo, Taille optimale du GoP>.

Bibliographie

[1] Recommendation itu-r bt.1683- objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference. June 2004. 88

[2] RECOMMENDATION ITU-R BT.500-13 - Methodology for the subjective assessment of the quality of television pictures. 2012. 79, 81, 84, 88, 94

[3] Recommendation itu-r bt.2022 - general viewing conditions for subjective assessment of quality of sdtv and hdtv television pictures on flat panel displays. Aug. 2012. 83

[4] Recommendation itu-r p.1401 - methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models. July 2012. 84

[5] C. G. Bampis, Z. Li, and A. C. Bovik. Spatiotemporal Feature Integration and Model Fusion for Full Reference Video Quality Assessment. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(8) :2256–2270, Aug. 2019. doi : 10.1109/TCSVT.2018.2868262. x, 88

[6] G. Baruffa and F. Frescura. Performance of SoftCast and H.265 in software radio video multicasting systems. In *2017 International Symposium on Wireless Communication Systems (ISWCS)*, pages 25–30, Aug. 2017. doi : 10.1109/ISWCS.2017.8108119. 163

[7] K. Blackard, T. Rappaport, and C. Bostian. Measurements and models of radio frequency impulsive noise for indoor wireless communications. *IEEE Journal on Selected Areas in Communications*, 11(7) :991–1001, Sept. 1993. ISSN 0733-8716, 1558-0008. doi : 10.1109/49.233212. 32

[8] N. T. Blog. Toward A Practical Perceptual Video Quality Metric, Apr. 2017. URL https://medium.com/netflix-techblog/toward-a-practical-perceptual-video-quality-metric-653f208b9652. 88

[9] A. Borowiak, U. Reiter, and U. P. Svensson. Quality evaluation of long duration audio-visual content. In *2012 IEEE Consumer Communications and Networking Conference (CCNC)*, pages 337–341. IEEE, 2012. xv, 164, 165

[10] T. Brandao, L. Roque, and M. P. Queluz. Quality assessment of H.264/AVC encoded video. In *Proc. of Conference on Telecommunications - ConfTele*, page 5, Sta. Maria da Feira, Portugal, 2009. 48

[11] M. Cagnazzo and M. Kieffer. Shannon-Kotelnikov mappings for softcast-based joint source-channel video coding. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 1085–1089, Sept. 2015. doi : 10.1109/ICIP.2015.7350967. vi, 29, 30, 31

[12] H. Cui, Z. Song, Z. Yang, C. Luo, R. Xiong, and F. Wu. Cactus : A hybrid digital-analog wireless video communication system. In *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems*, pages 273–278. ACM, 2013. 21

[13] H. Cui, C. Luo, C. W. Chen, and F. Wu. Robust uncoded video transmission over wireless fast fading channel. In *INFOCOM, 2014 Proceedings IEEE*, pages 73–81. IEEE, 2014. 10, 28, 59

[14] H. Cui, C. Luo, C. W. Chen, and F. Wu. Robust uncoded video transmission over wireless fast fading channel. In *INFOCOM, 2014 Proceedings IEEE*, pages 73–81. IEEE, 2014. 28

[15] H. Cui, D. Liu, Y. Han, and J. Wu. Robust Uncoded Video Transmission under Practical Channel Estimation. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec. 2016. ISBN 1-5090-1328-8. 107

[16] H. Cui, C. Luo, C. W. Chen, and F. Wu. Scalable Video Multicast for MU-MIMO Systems With Antenna Heterogeneity. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(5) :992–1003, May 2016. ISSN 1051-8215, 1558-2205. doi : 10.1109/TCSVT.2015.2430651. 29

[17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing*, 16(8) :2080–2095, Aug. 2007. ISSN 1057-7149. doi : 10.1109/TIP.2007.901238. 34

[18] X. Fan, F. Wu, and D. Zhao. D-Cast : DSC based soft mobile video broadcast. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*, pages 226–235. ACM, 2011. 17, 19

[19] X. Fan, F. Wu, D. Zhao, O. C. Au, and W. Gao. Distributed soft video broadcast (DCAST) with explicit motion. In *Data Compression Conference (DCC), 2012*, pages 199–208. IEEE, 2012. 17

[20] X. Fan, R. Xiong, F. Wu, and D. Zhao. Wavecast : Wavelet based wireless video broadcast using lossy transmission. In *Proc. IEEE Visual Communications and Image Processing (VCIP)*, pages 1–6, Nov. 2012. v, vi, 20, 21, 37, 42

[21] X. Fan, F. Wu, D. Zhao, and O. C. Au. Distributed wireless visual communication with power distortion optimization. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(6) :1040–1053, 2013. v, 17, 18, 19, 67

[22] X. Fan, R. Xiong, D. Zhao, and F. Wu. Layered soft video broadcast for heterogeneous receivers. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(11) : 1801–1814, 2015. 19

[23] M. Frigo and S. Johnson. The Design and Implementation of FFTW3. *Proceedings of the IEEE*, 93(2) :216–231, Feb. 2005. ISSN 0018-9219. doi : 10.1109/JPROC.2004.840301. 59, 134

[24] T. Fryza. Improving Quality of Video Signals Encoded by 3d DCT Transform. In *Proceedings ELMAR 2006*, pages 89–93, June 2006. doi : 10.1109/ELMAR.2006.329522. 72

[25] T. Fryza and S. Hanus. Video signals transparency in consequence of 3d-dct transform. In *Radioelektronika 2003 Conference Proceedings*, pages 127–130, 2003. 72

[26] T. Fujibashi, T. Koike-Akino, T. Watanabe, and P. V. Orlik. Soft video delivery for free viewpoint video. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2017. doi : 10.1109/ICC.2017.7996602. 34

[27] T. Fujihashi, T. Koike-Akino, T. Watanabe, and P. V. Orlik. High-Quality Soft Video Delivery With GMRF-Based Overhead Reduction. *IEEE Transactions on Multimedia*, 20(2) :473–483, Feb. 2018. ISSN 1520-9210. doi : 10.1109/TMM.2017.2743984. 13, 23, 41, 42, 44, 109, 127, 133, 144

[28] T. Fujihashi, T. Koike-Akino, T. Watanabe, and P. V. Orlik. FreeCast : Graceful Free-Viewpoint Video Delivery. *IEEE Transactions on Multimedia*, 21(4) :1000–1010, Apr. 2019. ISSN 1520-9210. doi : 10.1109/TMM.2018.2870074. 34

[29] T. Fujihashi, T. Koike-Akino, T. Watanabe, and P. V. Orlik. HoloCast : Graph Signal Processing for Graceful Point Cloud Delivery. In *ICC 2019 - 2019 IEEE International*

Conference on Communications (ICC), pages 1–7, May 2019. doi : 10.1109/ICC.2019.8761819. 34

[30] T. Fujihashi, I. Otomo, K. Endo, Y. Hirota, S. Kobayashi, and T. Watanabe. Wi-Fi Offloading for Multi-Homed Hybrid Digital-Analog Video Streaming. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2019. doi : 10.1109/ICC.2019.8761622. 23

[31] R. G. Gallager. *Stochastic processes : theory for applications*. Cambridge University Press, 2013. 18

[32] S. Gao, M. Jiao, and S. Zong. Content-based power allocation for perception-friendly SoftCast. In *Eleventh International Conference on Digital Image Processing (ICDIP 2019)*, volume 11179, page 1117909. International Society for Optics and Photonics, Aug. 2019. doi : 10.1117/12.2539764. 26

[33] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed Video Coding. *Proceedings of the IEEE*, 93 :71–83, 2005. doi : 10.1109/jproc.2004.839619. 17

[34] V. Q. E. Group et al. Final report from the video quality experts group on the validation of objective models of video quality assessment, phase i. *2000 VQEG*, 2003. 88, 89

[35] H. Hadizadeh. Saliency-guided wireless transmission of still images using SoftCast. In *2016 8th International Symposium on Telecommunications (IST)*, pages 506–509. IEEE, 2016. 27

[36] A. Hagag, X. Fan, and F. E. A. El-Samie. HyperCast : Hyperspectral satellite image broadcasting with band ordering optimization. *Journal of Visual Communication and Image Representation*, 42 :14–27, 2017. 21

[37] A. Hagag, X. Fan, and F. E. A. El-Samie. Satellite Images Broadcast Based on Wireless SoftCast Scheme. *IAENG International Journal of Computer Science*, 44(1), Feb. 2017. 106

[38] P. Hanhart, M. Rerabek, and T. Ebrahimi. Towards high dynamic range extensions of HEVC : subjective evaluation of potential coding technologies, 2015. URL `https://infoscience.epfl.ch/record/210206`. 98

[39] Hao Cui, Ruiqin Xiong, Chong Luo, Zhihai Song, and Feng Wu. Denoising and Resource Allocation in Uncoded Video Transmission. *IEEE Journal of Selected Topics in Signal Processing*, 9(1) :102–112, Feb. 2015. ISSN 1932-4553, 1941-0484. doi : 10.1109/JSTSP.2014.2338279. 34

[40] C. He, H. Qin, Z. He, and K. Niu. Adaptive GoP dividing video coding for wireless broadcast based on power allocation optimization. In *Int. Conf. on Wireless Comm. & Signal Process. (WCSP)*, pages 1–5, 2016. xi, xii, 6, 108, 109, 115, 116, 118, 123, 126, 162

[41] D. He, C. Luo, C. Lan, F. Wu, and W. Zeng. Structure-preserving hybrid digital-analog video delivery in wireless networks. *IEEE Trans. Multimedia*, 17(9) :1658–1670, Sep. 2015. 12, 87

[42] D. He, C. Luo, C. Lan, F. Wu, and W. Zeng. Structure-preserving hybrid digital-analog video delivery in wireless networks. *IEEE Transactions on Multimedia*, 17(9) : 1658–1670, Sept. 2015. 22, 27, 41, 42, 133

[43] D. He, C. Lan, C. Luo, E. Chen, F. Wu, and W. Zeng. Progressive Pseudo-analog Transmission for Mobile Video Streaming. *IEEE Transactions on Multimedia*, 19(8) : 1894–1907, Aug. 2017. ISSN 1520-9210. doi : 10.1109/TMM.2017.2686703. 133

[44] F. Hekland, P. A. Floor, and T. A. Ramstad. Shannon-kotel-nikov mappings in joint source-channel coding. *IEEE Transactions on Communications*, 57(1) :94–105, Jan. 2009. ISSN 0090-6778, 1558-0857. doi : 10.1109/TCOMM.2009.0901.070075. vi, 29, 30

[45] A. Horé and D. Ziou. Image Quality Metrics : PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, Aug. 2010. doi : 10.1109/ICPR.2010.579. ISSN : 1051-4651, 1051-4651. 164

[46] S.-C. Hsia and S.-H. Wang. High-performance adaptive group-of-picture rate control for H.264/AVC. *Signal, Image and Video Processing*, 5(2) :155–163, June 2011. ISSN 1863-1711. doi : 10.1007/s11760-009-0150-3. 149

[47] W. Huang, X. Fan, and D. Zhao. Soft mobile video broadcast based on side information refining. In *Visual Communications and Image Processing (VCIP), 2013*, pages 1–6. IEEE, 2013. 19

[48] X.-L. Huang, J. Wu, and F. Hu. Knowledge-Enhanced Mobile Video Broadcasting (KMV-Cast) Framework with Cloud Support. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017. 34

[49] Q. Huynh-Thu and M. Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 44(13) :800–801, June 2008. ISSN 0013-5194. doi : 10.1049/el:20080522. 93

[50] C. V. N. Index. Forecast and methodology, 2016–2021. *White paper, Cisco public*, 6, 2017. 1

[51] P. ITU-T RECOMMENDATION. Subjective video quality assessment methods for multimedia applications. Sept. 1999. 48, 110, 133

[52] S. Jakubczak and D. Katabi. SoftCast : Clean-slate scalable wireless video. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 530–533. IEEE, 2010. v, 3, 4, 163

[53] S. Jakubczak and D. Katabi. A cross-layer design for scalable mobile video. In *Proc. of the 17th annual international conference on Mobile computing and networking (MobiCom)*, pages 289–300, Sept. 2011. 17, 37, 38, 39, 40, 41, 42, 44, 47, 57, 59, 134

[54] S. Jakubczak and D. Katabi. SoftCast : Clean-slate scalable wireless video. *MIT Technical report*, Feb. 2011. 4, 5, 8, 10, 11, 12, 13, 14, 36, 66, 68, 127, 144, 161

[55] S. Jakubczak, H. Rahul, and D. Katabi. One-Size-Fits-All Wireless Video. In *HotNets*, 2009. 14

[56] M. Kleiner, D. Brainard, D. Pelli, A. Ingling, R. Murray, and C. Broussard. What's new in psychtoolbox-3. *Perception*, 36(14) :1–16, 2007. ISSN 0301-0066. URL `http://psychtoolbox.org/`. 80

[57] S. Kokalj-Filipović and E. Soljanin. Suppressing the cliff effect in video reproduction quality. *Bell Labs Technical Journal*, 16(4) :171–185, Mar. 2012. ISSN 1089-7089. doi : 10.1002/bltj.20540. 2

[58] Kyong-Hwa Lee and D. Petersen. Optimal Linear Coding for Vector Channels. *IEEE Transactions on Communications*, 24(12) :1283–1290, Dec. 1976. ISSN 0096-2244. doi : 10.1109/TCOM.1976.1093255. 10, 36, 51, 52, 54

[59] S. Li, F. Zhang, L. Ma, and K. N. Ngan. Image Quality Assessment by Separately Evaluating Detail Losses and Additive Impairments. *IEEE Transactions on Multimedia*, 13(5) :935–949, Oct. 2011. doi : 10.1109/TMM.2011.2152382. x, 87, 89

[60] Y. Li, Y. Liu, Y. Wang, and Z. Li. Visual information exploited hybrid digital-analog scheme for wireless video multicast. In *Visual Communications and Image Processing (VCIP), 2016*, pages 1–4. IEEE, 2016. 27

[61] Y. Li, Z. Li, Y. Liu, and Y. Wang. SCAST : Wireless Video Multicast Scheme Based on Segmentation and Softcast. In *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*, pages 1–6. IEEE, 2017. 27, 42

[62] Z. Li, H. Lu, and Y. Wu. Compressed uncoded screen content video transmission in bandwidth-constrained wireless networks. In *IEEE Int. Conf. Wireless Commun. & Signal Process. (WCSP)*, pages 1–5, Nov. 2016. 9

[63] F. Liang, C. Luo, R. Xiong, W. Zeng, and F. Wu. Hybrid Digital–Analog Video Delivery With Shannon–Kotel'nikov Mapping. *IEEE Transactions on Multimedia*, 20(8) :2138–2152, Aug. 2018. ISSN 1520-9210. doi : 10.1109/TMM.2017.2785264. 31, 36

[64] F. Liang, C. Luo, R. Xiong, W. Zeng, and F. Wu. Superimposed Modulation for Soft Video Delivery with Hidden Resources. *IEEE Trans. Circuits Systems Video Technol.*, 28(9) :2345–2358, Sept. 2018. ISSN 1051-8215. doi : 10.1109/TCSVT.2017.2703605. 2, 29, 42, 57, 64, 152

[65] L. Lin, S. Yu, T. Zhao, Member, IEEE, Z. Wang, Fellow, and IEEE. PEA265 : Perceptual Assessment of Video Compression Artifacts. *arXiv :1903.00473 [cs, eess]*, Mar. 2019. URL http://arxiv.org/abs/1903.00473. arXiv : 1903.00473. 66

[66] H. Liu, R. Xiong, S. Ma, X. Fan, and W. Gao. Gradient based image/video softcast with grouped-patch collaborative reconstruction. In *Visual Communications and Image Processing Conference, 2014 IEEE*, pages 141–144. IEEE, Dec. 2014. ISBN 978-1-4799-6139-9. doi : 10.1109/VCIP.2014.7051524. 25

[67] H. Liu, R. Xiong, S. Ma, X. Fan, and W. Gao. Gradient based image transmission and reconstruction using non-local gradient sparsity regularization. In *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, pages 1–6. IEEE, July 2014. ISBN 978-1-4799-4761-4. doi : 10.1109/ICME.2014.6890272. 25

[68] H. Liu, R. Xiong, X. Fan, C. Luo, and W. Gao. Compressive gradient based scalable image SoftCast. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, Dec. 2017. doi : 10.1109/VCIP.2017.8305069. 25

[69] H. Liu, R. Xiong, X. Fan, S. Ma, and W. Gao. Wireless Image SoftCast Using Compressive Gradient. In *2017 Data Compression Conference (DCC)*, pages 451–451, Apr. 2017. doi : 10.1109/DCC.2017.64. 25

[70] H. Liu, R. Xiong, X. Fan, D. Zhao, Y. Zhang, and W. Gao. CG-Cast : Scalable Wireless Image SoftCast Using Compressive Gradient. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(6) :1832–1843, June 2019. ISSN 1051-8215. doi : 10.1109/TCSVT.2018.2842818. 25

[71] X. L. Liu, W. Hu, Q. Pu, F. Wu, and Y. Zhang. ParCast : Soft video delivery in MIMO-OFDM WLANs. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 233–244. ACM, 2012. 28

[72] X. L. Liu, W. Hu, C. Luo, Q. Pu, F. Wu, and Y. Zhang. ParCast+ : Parallel video unicast in MIMO-OFDM WLANs. *IEEE Transactions on Multimedia*, 16(7) :2038–2051, 2014. vi, 28, 29

[73] R. Martins, C. Brites, J. Ascenso, and F. Pereira. Refining Side Information for Improved Transform Domain Wyner-Ziv Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(9) :1327–1341, Sept. 2009. ISSN 1051-8215. doi : 10.1109/TCSVT.2009.2022783. 19

[74] N. Nedev, S. McLaughlin, D. Laurenson, and R. Daley. Data errors in ADSL and SHDSL systems due to impulse noise. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages IV–4048–IV–4051, May 2002. doi : 10.1109/ICASSP.2002.5745546. ISSN : 1520-6149. 31, 32

[75] T. Ouni, W. Ayedi, and M. Abid. New Non Predictive Wavelet Based Video Coder : Performances Analysis. In *Image Analysis and Recognition*, Lecture Notes in Computer Science, pages 344–353. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-13772-3. xix, 72

[76] D. P. Palomar, Y. Jiang, et al. Mimo transceiver design via majorization theory. *Foundations and Trends® in Communications and Information Theory*, 3(4-5) :331–551, 2007. 31

[77] J. S. Park and T. Ogunfunmi. A 3D-DCT video encoder using advanced coding techniques for low power mobile device. *Journal of Visual Commun. Image Representation*, 48 :122–135, Oct. 2017. ISSN 10473203. doi : 10.1016/j.jvcir.2017.06.004. 9

[78] M. Paul, Weisi Lin, Chiew-Tong Lau, and Bu-Sung Lee. Explore and Model Better I-Frames for Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(9) :1242–1254, Sept. 2011. ISSN 1051-8215, 1558-2205. doi : 10.1109/TCSVT.2011.2138750. 149

[79] X. Peng, J. Xu, and F. Wu. Line-cast : Line-based semi-analog broadcasting of satellite images. In *Proc. Int. Conf. Image Process.(ICIP)*, pages 2929–2932. Citeseer, 2012. 19

[80] M. Perez-Ortiz and R. K. Mantiuk. A practical guide and software for analysing pairwise comparison experiments. *arXiv :1712.03686 [cs, stat]*, Dec. 2017. URL `https://github.com/mantiuk/pwcmp`. 98

[81] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS) : design and construction. *IEEE Transactions on Information Theory*, 49(3) : 626–643, Mar. 2003. ISSN 0018-9448. doi : 10.1109/TIT.2002.808103. 17

[82] I. E. G. Richardson. *The H.264 advanced video compression standard*. Wiley, Chichester, 2. ed edition, 2010. ISBN 978-0-470-51692-8. 2

[83] D. Salomon and G. Motta. *Handbook of Data Compression*. Springer-Verlag, London, 5 edition, 2010. ISBN 978-1-84882-902-2. 60, 63, 123, 134, 151

[84] P. Schniter, L. C. Potter, and J. Ziniel. Fast bayesian matching pursuit. In *2008 Information Theory and Applications Workshop*, pages 326–333, San Diego, CA, USA, Jan. 2008. IEEE. ISBN 978-1-4244-2670-6. doi : 10.1109/ITA.2008.4601068. 32

[85] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9) :1103–1120, Sept. 2007. ISSN 1051-8215. doi : 10.1109/TCSVT.2007.905532. 3

[86] A. Secker and D. Taubman. Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression. *IEEE Transactions on Image Processing*, 12(12) :1530–1542, Dec. 2003. ISSN 1057-7149. doi : 10.1109/TIP.2003.819433. 20

[87] S. Sharma, V. Bhatia, and A. Gupta. Sparsity based UWB receiver design in additive impulse noise channels. In *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, July 2016. doi : 10.1109/SPAWC.2016.7536788. 32

[88] H. R. Sheikh and A. C. Bovik. Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2) :430–444, Feb. 2006. doi : 10.1109/TIP.2005.859378. 87

[89] J. Shen, F. Liang, C. Luo, H. Li, and W. Zeng. Cooperative Hybrid Digital-Analog Video Transmission in D2d Networks. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3274–3278, Oct. 2018. doi : 10.1109/ICIP.2018.8451265. 23

[90] J. Shen, L. Yu, L. Li, and H. Li. Foveation-Based Wireless Soft Image Delivery. *IEEE Transactions on Multimedia*, 20(10) :2788–2800, Oct. 2018. ISSN 1520-9210. doi : 10.1109/TMM.2018.2811622. 27

[91] J. Shen, L. Yu, L. Li, and H. Li. Foveation Based Wireless Soft Image Delivery. *IEEE Trans. Multimedia*, 20(10) :2788–2800, Oct. 2018. ISSN 1520-9210. doi : 10.1109/TMM.2018.2811622. 66

[92] T. Shongwey, A. J. H. Vinck, and H. C. Ferreira. On impulse noise and its models. In *18th IEEE International Symposium on Power Line Communications and Its Applications*, pages 12–17, Mar. 2014. doi : 10.1109/ISPLC.2014.6812360. 32

[93] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. Circuits Syst. Video Technol.*, 22(12) : 1649–1668, Dec. 2012. ISSN 1051-8215. doi : 10.1109/TCSVT.2012.2221191. 2

[94] M. Sun, Y. Wang, H. Yu, and Y. Liu. Distributed cooperative video coding for wireless video broadcast system. In *Multimedia and Expo (ICME), 2015 IEEE International Conference on*, pages 1–6. IEEE, 2015. 23

[95] B. Tan, H. Cui, J. Wu, and C. W. Chen. An Optimal Resource Allocation for Superposition Coding-Based Hybrid Digital–Analog System. *IEEE Internet of Things Journal*, 4(4) :945–956, Aug. 2017. ISSN 2327-4662. doi : 10.1109/JIOT.2017.2680407. 22

[96] B. Tan, J. Wu, H. Cui, R. Wang, J. Wu, and D. Liu. A Hybrid Digital Analog Scheme for MIMO Multimedia Broadcasting. *IEEE Wireless Communications Letters*, 6(3) : 322–325, June 2017. ISSN 2162-2337. doi : 10.1109/LWC.2017.2683490. 29, 41, 42, 133

[97] B. Tan, J. Wu, R. Wang, W. Luo, and J. Liu. An Optimal Resource Allocation for Hybrid Digital-Analog with Combined Multiplexing. *IEEE Internet of Things Journal*, pages 1125–1135, 2018. ISSN 2327-4662, 2372-2541. doi : 10.1109/JIOT.2018.2867524. 22

[98] A. Trioux, F.-X. Coudoux, P. Corlay, and M. Gharbi. A comparative preprocessing study for softcast video transmission. In *Proc. IEEE International Symposium on Signal Image and Video Communications (ISIVC)*, Nov. 2018. 125, 127

[99] T. Tung and D. Gündüz. SparseCast : Hybrid Digital-Analog Wireless Image Transmission Exploiting Frequency-Domain Sparsity. *IEEE Communications Letters*, 22(12) : 2451–2454, Dec. 2018. doi : 10.1109/LCOMM.2018.2877316. v, xv, 11, 163

[100] F. Urban, R. Poullaouec, J. Nezan, and O. Deforges. A Flexible Heterogeneous Hardware/Software Solution for Real-Time HD H.264 Motion Estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(12) :1781–1785, Dec. 2008. ISSN 1051-8215. doi : 10.1109/TCSVT.2008.2004927. 141

[101] Y. Wang, H. Lu, Z. Li, and J. Li. Robust satellite image transmission over bandwidth-constrained wireless channels. In *Proc. IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017. doi : 10.1109/ICC.2017.7997077. 42

[102] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment : from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4) :600–612, Apr. 2004. ISSN 1057-7149. doi : 10.1109/TIP.2003.819861. 27, 87

[103] M. Wien. *High Efficiency Video Coding : Coding Tools and Specification*. Signals and Communication Technology. Springer-Verlag, Berlin Heidelberg, 2015. ISBN 978-3-662-44275-3. xviii, 41, 133, 192, 193

[104] M. Wien, H. Schwarz, and T. Oelbaum. Performance Analysis of SVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9) :1194–1203, Sept. 2007. ISSN 1051-8215, 1558-2205. doi : 10.1109/TCSVT.2007.905530. 3

[105] H. Wu, A. Wang, J. Liang, S. Li, and P. Li. DCSN-Cast : Deep compressed sensing network for wireless video multicast. *Signal Processing : Image Communication*, 76 : 56–67, Aug. 2019. ISSN 0923-5965. doi : 10.1016/j.image.2019.04.017. 34

[106] H. R. Wu and K. R. Rao. *Digital Video Image Quality and Perceptual Coding (Signal Processing and Communications)*. CRC Press, Inc., Boca Raton, FL, USA, 2006. ISBN 978-0-8247-2777-2. 78, 95

[107] J. Wu, J. Wu, H. Cui, C. Luo, X. Sun, and F. Wu. DAC-Mobi : Data-Assisted Communications of Mobile Images with Cloud Computing Support. *IEEE Transactions on Multimedia*, 18 :893–904, May 2016. ISSN 1520-9210, 1941-0077. 10, 34, 36, 51, 52, 54

[108] R. Xiong, J. Xu, F. Wu, and S. Li. Barbell-Lifting Based 3-D Wavelet Coding Scheme. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9) :1256–1269, Sept. 2007. ISSN 1051-8215. doi : 10.1109/TCSVT.2007.905507. 20

[109] R. Xiong, H. Liu, S. Ma, X. Fan, F. Wu, and W. Gao. G-CAST : Gradient Based Image SoftCast for Perception-Friendly Wireless Visual Communication. In *2014 Data Compression Conference*, pages 133–142, Mar. 2014. doi : 10.1109/DCC.2014.55. vi, 25, 26

[110] R. Xiong, J. Zhang, F. Wu, and W. Gao. High quality image reconstruction via non-local collaborative estimation for wireless image/video softcast. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 2542–2546. IEEE, 2014. 68

[111] R. Xiong, F. Wu, J. Xu, X. Fan, and al. Analysis of decorrelation transform gain for uncoded wireless image and video communication. *IEEE Trans. Image Process.*, 25(4) : 1820–1833, Apr. 2016. 10, 11, 14, 16, 34, 37, 40, 42, 47, 57, 140, 144, 152, 161

[112] R. Xiong, J. Zhang, F. Wu, J. Xu, and W. Gao. Power Distortion Optimization for Uncoded Linear Transformed Transmission of Images and Videos. *IEEE Transactions on Image Processing*, 26(1) :222–236, Jan. 2017. vi, xv, 14, 23, 24, 109, 164

[113] Xiph. Xiph.org media. URL https://media.xiph.org/video/derf/. 41, 109, 133

[114] D. Yang, Y. Bi, Z. Si, Z. He, and K. Niu. Performance evaluation and parameter optimization of SoftCast wireless video broadcast. In *Proceedings of the 8th International Conference on Mobile Multimedia Communications*, pages 79–84. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2015. 23

[115] C. Yim and A. C. Bovik. Evaluation of temporal variation of video quality in packet loss networks. *Signal Processing : Image Communication*, 26(1) :24–38, Jan. 2011. ISSN 09235965. doi : 10.1016/j.image.2010.11.002. 149

[116] W. Yin, X. Fan, Y. Shi, R. Xiong, and D. Zhao. Compressive sensing based soft video broadcast using spatial and temporal sparsity. *Mobile Networks and Applications*, 21 (6) :1002–1012, 2016. xv, 163

[117] W. Yin, X. Fan, and Y. Shi. Convolutional Neural Networks Based Soft Video Broadcast. In R. Hong, W.-H. Cheng, T. Yamasaki, M. Wang, and C.-W. Ngo, editors, *Advances in Multimedia Information Processing – PCM 2018*, Lecture Notes in Computer Science, pages 641–650. Springer International Publishing, 2018. ISBN 978-3-030-00764-5. 27, 36, 67

[118] L. Yu, H. Li, and W. Li. Wireless scalable video coding using a hybrid digital-analog scheme. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(2) : 331–345, 2014. vi, 21, 22

[119] L. Yu, H. Li, and W. Li. Wireless Cooperative Video Coding Using a Hybrid Digital–Analog Scheme. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(3) :436–450, Mar. 2015. ISSN 1051-8215, 1558-2205. doi : 10.1109/TCSVT.2014.2347532. 22

[120] M. Yuen and H. Wu. A survey of hybrid MC/DPCM/DCT video coding distortions. *Signal Processing*, 70(3) :247–278, Nov. 1998. ISSN 01651684. doi : 10.1016/S0165-1684(98)00128-5. URL https://linkinghub.elsevier.com/retrieve/pii/S0165168498001285. 66

[121] E. Zerman, G. Valenzise, F. De Simone, F. Banterle, and F. Dufaux. Effects of display rendering on HDR image quality assessment. In *SPIE Optical Engineering+ Applications, Applications of Digital Image Processing XXXVIII*, San Diego, CA, United States, Aug. 2015. 93

[122] A. Zhang, X. Fan, R. Xiong, and D. Zhao. Distributed soft video broadcast with variable block size motion estimation. In *2013 Visual Communications and Image Processing (VCIP)*, pages 1–5, Nov. 2013. doi : 10.1109/VCIP.2013.6706380. 19

[123] T. Zhang and S. Mao. Metadata Reduction for Soft Video Delivery. *IEEE Networking Letters*, pages 84–88, Apr. 2019. ISSN 2576-3156. doi : 10.1109/LNET.2019.2912831. xv, 23, 24, 36, 164

[124] Z. Zhang, D. Liu, X. Ma, and X. Wang. ECast : An Enhanced Video Transmission Design for Wireless Multicast Systems Over Fading Channels. *IEEE Systems Journal*, 11(4) :2566–2577, Dec. 2017. ISSN 1932-8184. doi : 10.1109/JSYST.2015.2438071. 29

[125] J. Zhao, R. Xiong, C. Luo, F. Wu, and W. Gao. Wireless image and video soft transmission via perception-inspired power distortion optimization. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, Dec. 2017. doi : 10.1109/VCIP.2017.8305082. 27, 42

[126] X. Zhao, H. Lu, C. W. Chen, and J. Wu. Adaptive Hybrid Digital–Analog Video Transmission in Wireless Fading Channel. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(6) :1117–1130, 2016. 23

[127] S. Zheng, M. Antonini, M. Cagnazzo, L. Guerrieri, M. Kieffer, I. Nemoianu, R. Samy, and B. Zhang. Softcast with per-carrier power-constrained channels. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 2122–2126, Aug. 2016. 42

[128] S. Zheng, M. Cagnazzo, and M. Kieffer. Channel Impulsive Noise Mitigation for Linear Video Coding Schemes. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2019. ISSN 1051-8215. doi : 10.1109/TCSVT.2019.2937451. vi, 1, 32, 33, 67, 68

[129] S. Zheng, M. Cagnazzo, and M. Kieffer. Optimal and suboptimal channel precoding and decoding matrices for linear video coding. *Signal Processing : Image Communication*, 78 :135–151, Oct. 2019. ISSN 0923-5965. doi : 10.1016/j.image.2019.06.011. v, vi, 1, 2, 10, 31, 32, 36, 67, 68

[130] X. Zhu, N. Zhang, X. Fan, R. Xiong, and D. Zhao. Correlation estimation for distributed wireless video communication. In *Visual Communications and Image Processing (VCIP), 2013*, pages 1–5. IEEE, 2013. 19

[131] S. Zong, S. Gao, G. Tu, C. Zhang, and D. Chen. A metadata-free pure soft broadcast scheme for image and video transmission. *Signal Processing : Image Communication*, July 2019. ISSN 0923-5965. doi : 10.1016/j.image.2019.07.010. xv, 23, 24, 36, 163, 164

Study and optimization of a Joint Source-Channel video-transmission system based on "SoftCast"

Abstract

Linear video coding (LVC) schemes have recently demonstrated a high potential for delivering video content over challenging wireless channels. SoftCast represents the pioneer of the LVC schemes. Different from current video transmission standards and particularly useful in broadcast situation, SoftCast is a joint source-channel coding system where pixels are processed by successive linear operations (DCT transform, power allocation, quasi-analog modulation) and directly transmitted without quantization or coding (entropic or channel). This allows to provide a received video quality directly proportional to the transmission channel quality, without any feedback information, while avoiding the complex adaptation mechanisms of conventional schemes. A first contribution of this thesis is the study of the end-to-end performances of SoftCast. Theoretical models are thus proposed taking into account the bandwidth constraints of the application, the power allocation, as well as the type of decoder used at the reception (LLSE, ZF). Based on a subjective test campaign, a second part concern an original study of the video quality and specific artifacts related to SoftCast. In a third part, preprocessing methods are proposed to increase the received quality in terms of PSNR scores with an average gain of 3 dB. Finally, an adaptive algorithm modifying the size of the group of pictures (GoP) according to the characteristics of the transmitted video content is proposed. This solution allows to obtain about 1 dB additional gains in terms of PSNR scores.

Keywords : Joint Source-Channel Coding (JSCC), Video transmission, Linear Video Coding (LVC), SoftCast, Modeling, Adaptive video processing, Coding artefacts, Subjective quality assessment.

Etude et optimisation d'un système de vidéotransmission conjoint Source-Canal basé « SoftCast »

Résumé

Des nouveaux schémas de Codage Vidéo Linéaire (CVL) ont démontré ces dernières années un potentiel élevé pour la diffusion de contenus vidéo sur des canaux de transmission sans-fil sévères. SoftCast représente le pionnier des schémas CVL. Différent des standards de transmission vidéo actuels et particulièrement utile en situation de broadcast, SoftCast est un système de codage conjoint source-canal où les pixels sont traités par des opérations linéaires successives (transformée DCT, allocation de puissance, modulation quasi-analogique) et directement transmis sans quantification ni codage (entropique ou de canal). SoftCast permet ainsi d'offrir une qualité vidéo reçue directement proportionnelle à la qualité du canal de transmission, sans aucune information de retour et tout en évitant les mécanismes d'adaptation complexes des schémas classiques. Un premier objectif de ces travaux de thèse concerne l'étude des performances de bout en bout de SoftCast. Des modèles théoriques sont ainsi proposés prenant en compte les contraintes de bande passante de l'application, l'allocation de puissance, ainsi que le type de décodeur utilisé à la réception (LLSE, ZF). Une deuxième partie basée sur une campagne de tests subjectifs concerne une étude originale de la qualité vidéo et des artefacts spécifiques associés à SoftCast. Dans une troisième partie, des méthodes de prétraitement permettant d'accroître la qualité reçue sont proposées avec un gain moyen en PSNR de l'ordre de 3 dB. Finalement, un algorithme adaptatif modifiant la taille du groupe d'images (GoP) en fonction des caractéristiques du contenu vidéo transmis est proposé. Cette solution permet d'obtenir des gains supplémentaires en PSNR de l'ordre de 1 dB.

Mots clés : Codage Conjoint Source-Canal (CCSC), Transmission vidéo, Codage Vidéo Linéaire (CVL), SoftCast, Modélisation, Traitement vidéo adaptatif, Artefacts de codage, évaluation subjective de la qualité.