

TABLE DES MATIÈRES

	Page
INTRODUCTION.....	1
CHAPITRE 1 LES PRINCIPES ET STANDARDS DE CODAGE VIDÉO	9
1.1 Introduction	9
1.2 Principes du codage vidéo.....	9
1.2.1 Schéma de principe d'un codeur vidéo	9
1.2.2 Exploitation de la redondance temporelle	11
1.2.3 Exploitation de la redondance spatiale	13
1.2.4 Estimation de mouvement	13
1.2.5 La précision des vecteurs de mouvement	16
1.3 Le décodage vidéo	17
1.4 Le standard MPEG-4	18
1.5 Le standard H.264/AVC	19
1.5.1 Le codeur et le décodeur H.264/AVC	20
1.5.2 Spécificités du standard H.264	21
1.6 Différences entre les standards MPEG-4 et H.264	26
1.7 Mesures de distorsion.....	26
1.8 Méthode d'optimisation débit-distorsion	30
1.8.1 Méthode de détermination des vecteurs de mouvement.....	32
1.8.2 Méthode d'optimisation du mode de codage	34
CHAPITRE 2 ÉTAT DE L'ART	38
2.1 Introduction	38
2.2 Approche dans le domaine transformé proposée par Shen	40
2.3 Approche basée sur les statistiques des modes de codage proposée par Lee et al.	42
2.4 Algorithme de Liang et al.....	46
2.5 Algorithme de Kim et al.	49
2.6 État de l'art sur d'autres types de transcodage	50
2.7 Conclusion	53
CHAPITRE 3 MÉTHODE ET ALGORITHME DE TRANSCODAGE MPEG-4 À H.264 BASÉS SUR L'INFORMATION RÉSIDUELLE	55
3.1 Introduction	55
3.2 Information résiduelle en tant que mesure de complexité des MBs et d'effica- cité des VMs et des MCs	57
3.2.1 Énergie résiduelle en tant que mesure d'efficacité des vecteurs de mou- vement	58
3.2.2 Lien entre l'énergie résiduelle et le partitionnement du MC.....	60

3.3	Détermination du mode de codage	63
3.3.1	Détermination du mode de codage pour des macroblocs codés Intra en MPEG-4	65
3.3.2	Détermination du mode de codage pour des macroblocs codés Skip en MPEG-4	65
3.3.3	Détermination du mode de codage pour des macroblocs codés Inter16×16 en MPEG-4	65
3.3.4	Détermination du mode de codage pour des macroblocs codés Inter8×8 en MPEG-4	68
3.3.5	Sommaire des modes de codage H.264 candidats selon le mode de codage MPEG-4 d'entrée	69
3.4	Détermination des vecteurs de mouvement.....	70
3.4.1	Détermination des vecteurs de mouvement pour un macrobloc codé Intra en MPEG-4	72
3.4.2	Détermination des vecteurs de mouvement pour un macrobloc codé Skip en MPEG-4	72
3.4.3	Détermination des vecteurs de mouvement pour un macrobloc codé Inter16×16 en MPEG-4	72
3.4.4	Détermination des vecteurs de mouvement pour un macrobloc codé Inter8×8 en MPEG-4.....	74
3.5	Simulations et résultats	77
3.6	Conclusion	87
CHAPITRE 4 MÉTHODE DE TRANSCODAGE MPEG-4 À H.264 BASÉE SUR LA SOMME ABSOLUE RELATIVE DU RÉSIDUEL		88
4.1	Introduction	88
4.2	Détermination du mode de codage.....	89
4.2.1	Influence du type de vidéo sur le mode de codage	90
4.2.2	Influence du débit binaire sur le mode de codage	94
4.2.3	Utilisation de la somme absolue relative du résiduel comme mesure quantitative de complexité du macrobloc	95
4.2.4	Utilisation du principe de corrélation de trames consécutives	101
4.3	Conclusion	103
CHAPITRE 5 ALGORITHME DE TRANSCODAGE MPEG-4 À H.264 BASÉ SUR LA SOMME ABSOLUE RELATIVE DU RÉSIDUEL		104
5.1	Détermination du mode de codage des macroblocs.....	104
5.1.1	Détermination du mode de codage H.264 pour des macroblocs d'entrée MPEG-4 de type Intra.....	106
5.1.2	Détermination du mode de codage H.264 pour des macroblocs d'entrée MPEG-4 de type Skip.....	107
5.1.3	Détermination du mode de codage H.264 pour des macroblocs d'entrée MPEG-4 de type Inter16×16.....	108

5.1.4	Détermination du mode de codage H.264 pour des macroblocs d'entrée MPEG-4 de type Inter8×8	116
5.2	Détermination des vecteurs de mouvement.....	120
5.2.1	Détermination des vecteurs de mouvement pour un mode MPEG-4 d'entrée Skip	122
5.2.2	Détermination des vecteurs de mouvement pour un mode MPEG-4 d'entrée Inter16×16	123
5.2.3	Détermination des vecteurs de mouvement pour un mode MPEG-4 d'entrée Inter8×8.....	124
5.3	Simulations et résultats obtenus.....	127
5.4	Conclusion	138
CHAPITRE 6 CONTRIBUTIONS		140
CONCLUSION.....		142
ANNEXE I	ÉTUDE COMPARATIVE DE LA MÉTHODE DE CALCUL DE LA SARR SUR LA TOTALITÉ ET SUR LA MOITIÉ DES PIXELS	144
ANNEXE II	LE TEST ANOVA	147
ANNEXE III	POURCENTAGE DE RAFFINEMENT DES VECTEURS DE MOUVEMENT	154
ANNEXE IV	VALEURS MOYENNES DE L'ASAR SELON LA COMPLEXITÉ DE LA VIDÉO	157
ANNEXE V	RÉSULTATS DE SIMULATIONS SUPPLÉMENTAIRES POUR VIDÉOS 1280×720	159
ANNEXE VI	INFLUENCE SUR LA QUALITÉ ET LA VITESSE DE L'ORDRE DES PROCESSUS DE RAFFINEMENT ET D'ÉVALUATION DU COÛT INTERVENANT DANS LA DÉTERMINATION DES VECTEURS DE MOUVEMENT	161
LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES		166

LISTE DES TABLEAUX

		Page
Tableau 1.1	Différences majeures entre le profil simple de MPEG-4 et les profils <i>baseline</i> ou <i>main</i> de H.264.....	27
Tableau 2.1	Table de correspondance des MCs H.264 et MPEG-4.....	44
Tableau 3.1	Moyenne de fréquence d'apparition des MCs H.264 en fonction des MCs MPEG-4 pour plusieurs types de vidéos QCIF transcodées à 32 kb/s en H.264, en utilisant le codeur H.264 d'Intel.....	64
Tableau 3.2	Moyenne de fréquence d'apparition des MCs H.264 en fonction des MCs MPEG-4 pour plusieurs types de vidéos QCIF transcodées à 160 kb/s en H.264, en utilisant le codeur H.264 d'Intel.....	64
Tableau 3.3	Conditions de simulation utilisées.....	79
Tableau 3.4	Résultats de vidéos QCIF en PSNR, en accélération et en fréquence de trames pour les méthodes Cascade, ResidualM, STAT, STAT+REF et MV+MS.....	81
Tableau 3.5	Résultats de vidéos CIF en PSNR, en accélération et en fréquence de trames pour les méthodes Cascade, ResidualM, STAT, STAT+REF et MV+MS.....	83
Tableau 3.6	Résultats de vidéos 4CIF, 720×576 en PSNR, en accélération et en fréquence de trames pour les méthodes Cascade, ResidualM, STAT, STAT+REF et MV+MS.....	84
Tableau 3.7	Résultats de vidéos HD, 1280×720 en PSNR, en accélération et en fréquence de trames pour les méthodes Cascade, ResidualM, STAT, STAT+REF et MV+MS.....	85
Tableau 3.8	Résultats de vidéos HD, 1920×1080 en PSNR, en accélération et en fréquence de trames pour les méthodes Cascade, ResidualM, STAT, STAT+REF et MV+MS.....	86
Tableau 4.1	Moyenne de fréquence d'apparition des MCs H.264 en fonction des MCs MPEG-4 pour plusieurs types de vidéos QCIF transcodées à 32 kb/s et 160 kb/s en H.264, en utilisant le codeur H.264 d'Intel.....	96

Tableau 4.2	Table ANOVA mettant en relief le lien entre la R_n/G et celui de H.264 pour plusieurs vidéos	101
Tableau 5.1	Moyenne de fréquence d'apparition des MCs H.264 en fonction des MCs MPEG-4 pour plusieurs types de vidéos QCIF transcodées à 32 kb/s et 160 kb/s en H.264, en utilisant le codeur H.264 d'Intel	105
Tableau 5.2	Moyenne de fréquence d'apparition des MCs H.264 selon la classification proposée des MBs Inter16×16 en trois classes Inter16×16 cas I, II et III	113
Tableau 5.3	Propriétés de quelques séquences vidéos tests QCIF et CIF utilisées pour les simulations	128
Tableau 5.4	Propriétés de séquences vidéos tests HD utilisées pour les simulations ..	129
Tableau 5.5	Conditions de simulations utilisées	130
Tableau 5.6	Résultats en PSNR, en accélération et en fréquence de trames pour différents types de vidéos QCIF	132
Tableau 5.7	Résultats en PSNR, en accélération et en fréquence de trames pour différents types de vidéos CIF	133
Tableau 5.8	Résultats en PSNR, en accélération et en fréquence de trames pour différents types de vidéos 4CIF, 704×576	135
Tableau 5.9	Résultats en PSNR, en accélération et en fréquence de trames pour différents types de vidéos HD 1920×1080	136

LISTE DES FIGURES

	Page
Figure 1.1	Schéma de principe d'un codeur vidéo..... 10
Figure 1.2	Principe de l'estimation de mouvement. 12
Figure 1.3	Répartition du temps de calcul des fonctions de codage du H.264. 14
Figure 1.4	Illustration de la recherche des vecteurs de mouvement. 15
Figure 1.5	Illustration de la notion de la précision des VMs. 17
Figure 1.6	Schéma de principe d'un décodeur vidéo. 18
Figure 1.7	Modes de codage intra et inter de la norme MPEG-4. 20
Figure 1.8	Performances du H.264 par rapport au MPEG-4 et au MPEG-2. 21
Figure 1.9	Codeur H.264..... 22
Figure 1.10	Décodeur H.264. 22
Figure 1.11	Modes de codage permis par H.264. 24
Figure 2.1	Schéma bloc d'un transcodeur en cascade MPEG-4 à H.264. 39
Figure 2.2	Schéma de principe d'un transcodage efficace. 40
Figure 3.1	Illustration de l'information résiduelle dans un schéma de codeur vidéo. .. 57
Figure 3.2	Illustration du lien entre l'énergie résiduelle et l'efficacité du VM par un codage MPEG-4 à différentes précisions de VMs. 59
Figure 3.3	Illustration du lien entre l'énergie résiduelle et l'efficacité du VM par un codage MPEG-4 à différent partitionnement..... 60
Figure 3.4	Illustration de la segmentation de codage d'une trame par rapport à l'énergie résiduelle pour la vidéo QCIF Akiyo. 62
Figure 3.5	Processus de détermination du mode de codage H.264 pour un MB codé Inter16×16 en MPEG-4. 68

Figure 3.6	Processus de détermination du mode de codage H.264 selon le mode de codage en MPEG-4.	71
Figure 3.7	Processus de détermination des VMs H.264 pour des MBs codés Inter16×16 en MPEG-4.	74
Figure 3.8	Algorithme de raffinement des VMs du demi au quart de pixel d'Intel (Bhaskaran et Konstantinides (1997)).	75
Figure 3.9	VMs MPEG-4 candidats selon le MC H.264 dans le cas d'un MB codé Inter8×8 en MPEG-4.	76
Figure 3.10	Processus de détermination des VMs H.264 pour des MBs codés Inter8×8 en MPEG-4.	77
Figure 3.11	Résultats en PSNR et en vitesse d'accélération de la méthode proposée, comparée aux autres méthodes.	82
Figure 3.12	Résultats en PSNR et en vitesse d'accélération de la méthode proposée, comparée aux autres méthodes.	82
Figure 4.1	Moyenne de la somme absolue du résiduel (ASAR) des MBs codés Inter16×16 en MPEG-4 et transcodés en Skip dans H.264 ($\mu_{16 \times 16, R}^{Skip}$) en fonction du débit pour différentes vidéos QCIF et CIF.	93
Figure 4.2	Moyenne de la somme absolue du résiduel (ASAR) des MBs codés Inter16×16 en MPEG-4 et transcodés en Inter16×16 dans H.264 ($\mu_{16 \times 16, R}^{16 \times 16}$) en fonction du débit pour différentes vidéos QCIF et CIF.	93
Figure 4.3	Graphique représentant l'énergie résiduelle des MBs codés en Inter16×16 dans MPEG-4 et transcodés en Skip dans H.264 ainsi que celle des MBs codés en Inter16×16 dans MPEG-4 et transcodés en Inter16×16 dans H.264.	94
Figure 4.4	Graphique représentant l'énergie résiduelle des MBs codés en Inter16×16 dans MPEG-4 et transcodés en Skip dans H.264 ainsi que celle des MBs codés en Inter16×16 dans MPEG-4 et transcodés en Inter16×16 dans H.264.	95
Figure 4.5	Pixels utilisés pour le calcul de la $R_{n/G}$	99
Figure 4.6	Lien entre $R_{n/G}$ et le MC H.264 pour la vidéo QCIF Akiyo.	100

Figure 5.1	Processus de détermination du MC H.264 en fonction du MC d'un MB d'entrée MPEG-4.	106
Figure 5.2	Processus de détermination des MCs candidats H.264 et du MC final pour un MB codé Intra en MPEG-4.	107
Figure 5.3	Processus de détermination des MCs candidats H.264 et du MC final pour un MB codé en Skip en MPEG-4.	108
Figure 5.4	Processus de détermination des MCs candidats H.264 et du MC final pour un MB codé Inter16×16 en MPEG-4.	109
Figure 5.5	Processus de détermination des MCs candidats H.264 et du MC final pour un MB appartenant à l'ensemble Inter16×16 cas II.	114
Figure 5.6	Processus de détermination des MCs candidats H.264 et du MC final pour un MB appartenant à l'ensemble Inter16×16 cas III.	115
Figure 5.7	Processus de détermination des MCs candidats H.264 et du MC final pour un MB MPEG-4 d'entrée Inter8×8.	117
Figure 5.8	Processus de détermination du VM H.264 dans le cas d'un MC MPEG-4 d'entrée Inter16×16.	124
Figure 5.9	Processus de détermination des VMs H.264 dans le cas d'un MC d'entrée MPEG-4 Inter8×8.	127
Figure 5.10	Résultats en PSNR et en accélération pour la vidéo Grandma (QCIF) à débits binaires différents.	131
Figure 5.11	Résultats en PSNR et en accélération pour la vidéo Highway (CIF) à débits binaires différents.	134

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

4CIF	<i>4× common intermediate format (704×574 pixels)</i>
AC	<i>Alternative current (non DC coefficients)</i> Coefficients d'une transformée associés à une fréquence non nulle
ANOVA	<i>Analysis of variance</i> Analyse de la variance
ASAR	Moyenne de la somme absolue du résiduel
ASARR	Moyenne de la somme absolue du résiduel relatif
ASP	<i>Advanced simple profile</i>
ATSC	<i>Advanced television systems committee</i>
AVC	<i>Advanced video coding</i>
CABAC	<i>Context adaptative binary arithmetic coding</i> Codage arithmétique binaire à contexte adaptatif
CAVLC	<i>Context adaptative variable length coding</i> Codage à longueur variable à contexte adaptatif
CE	Codage entropique
CIF	<i>Common intermediate format (352×288 pixels)</i>
CM	Compensation de mouvement
CPU	<i>Central processing unit</i> Unité centrale de traitement
DCT	<i>Discrete Cosinus Transform</i> Transformée en cosinus discrète
DD	Débit-distorsion
DDL	Degré de liberté
DVD	<i>Digital versatile disc</i>

EM	Estimation de mouvement
EPZS	<i>Enhanced predictive zonal search</i>
ER	Énergie résiduelle
FSA	<i>Full search algorithm</i>
H.263	Standard de codage vidéo de l'UIT
H.264	Standard de codage vidéo de l'UIT
HD	Haute définition
IDCT	DCT Inverse
IEC	<i>International electrotechnical commission</i>
IPP	<i>Integrated performance primitives</i>
IQ	<i>Inverse quantization</i> Quantification inverse
ISO	<i>International organization for standardization</i> Organisation internationale pour la standardisation
JM	<i>Joint model</i> (code de référence du H.264)
JVT	<i>Joint video team</i>
MAD	<i>Mean absolute difference</i> Moyenne des différences absolues
MANOVA	<i>Multivariate analysis of variance</i> Analyse de la variance multifactorielle
MB	Macrobloc(s)
MC	Mode de codage
Momusys	Code de référence MPEG-4
MPEG	<i>Motion picture experts group</i>
MPEG-4	Standard de codage multimédia MPEG d'ISO

MSE	<i>Mean squared error</i> Erreur quadratique moyenne
Pmvfast	<i>Predictive motion vector field adaptive search technique</i>
PSNR	<i>Peak signal noise ratio</i>
Q	Quantification
QCIF	<i>Quarter common intermediate format (176×144 pixels)</i>
QP	<i>Quantization parameter</i> Pas de quantification
Q/S	<i>Quality / speed up</i>
RD	<i>Rate distortion</i>
RDO	<i>Rate distortion optimization</i>
RGB	<i>Red green blue</i> Espace de couleur rouge, vert, bleu
SAD	<i>Sum of absolute differences</i> Somme des différences absolues
SAR	Somme absolue du résiduel
SARR	Somme absolue relative du résiduel
SATD	<i>Sum of absolute transformed differences</i> Somme des différences absolues transformées
SCE	Somme des carrés des écarts
SSB	<i>Sum of square between class</i> Somme des carrés entre classes
SSIM	<i>Structural Similarity</i>
SSD	<i>Sum of squared differences</i> Somme des différences au carré
SSW	<i>Sum of square within class</i> Somme des carrés dans une classe

Tr+Q	Opération combinée de transformée fréquentielle et de quantification
UIT	Union internationale des télécommunications
VCEG	<i>Video coding experts group</i>
VCR	<i>Video cassette recording</i>
VIF	<i>Visual information fidelity</i>
VLC	<i>Variable length coding</i> Codage à longueur variable
VM	Vecteur de mouvement
VSP	<i>Visual simple profile</i>
YCbCr	<i>Luminance, blue chrominance, red chrominance</i> Espace de couleur luminance, chrominance bleue, chrominance rouge

LISTE DES SYMBOLES ET UNITÉS DE MESURE

α_{mci}^{mco}	Coefficient d'ajustement du rapport qualité/vitesse relatif à la détermination des modes de codage
$\tilde{\alpha}_{mci}^{mco}$	Coefficient d'ajustement de contrôle du rapport qualité/vitesse relatif à la détermination des modes de codage
α_{vm}	Coefficient d'ajustement du rapport qualité/vitesse relatif à la détermination des VMs
E_n	Énergie résiduelle du macrobloc n
$E_{n,k}$	Énergie résiduelle du block k du macrobloc n
$\mathcal{F}(\cdot)$	Opération de raffinement d'un vecteur
$J_{MC}(m)$	Coût du lagragien lié au mode de codage m
$J_{motion}(\mathbf{v}_i)$	Coût du lagragien lié au vecteur de mouvement \mathbf{v}_i
$\mathbf{K}_n(i, j)$	MB original
$\hat{\mathbf{K}}_n(i, j)$	MB prédit
$MCC_{mci}^{H.264}$	Ensemble des modes de codage candidats d'H.264 pour un macrobloc codé mci en MPEG-4
$MC_{H.264}$	Mode de codage H.264
mci	MC d'entrée (<i>input</i>) MPEG-4
mco	MC de sortie (<i>output</i>) H.264
R_n	Somme absolue du résiduel du MB n
$\mathbf{R}_n(i, j)$	Valeur du résiduel du pixel à la position (i, j) du MB n
$R_{n/G}$	SARR d'un MB n d'une trame
$R_{n/G}^t$	SARR d'un MB n de la trame t
$R_{(n,k)/G}$	SARR du bloc k du MB n d'une trame
$R_{(n,k)/G}^t$	SARR du bloc k du MB n de la trame t

S_{mci}^{mco}	Ensemble des MBs d'une trame, transcodés du MC mci en MPEG-4 à mco en H.264
$ S_{mci}^{mco} $	Nombre d'éléments de S_{mci}^{mco} (cardinalité de S_{mci}^{mco})
$S_{mci}^{mco,t}$	Ensemble des MBs d'une trame t , transcodés du MC mci en MPEG-4 à mco en H.264
$(t \bmod T_r) = 0$	Condition respectée par une trame t qui est de type intra
$(t \bmod T_r) = 1$	Condition respectée par une trame t qui suit une trame intra
$(t \bmod T_r) > 1$	Condition respectée par une trame t qui suit ne pas une trame intra
T_c	Temps de transcodage de la méthode en cascade
T_M	Temps de transcodage des méthodes à tester
T_{mc}^{Skip}	Seuil utilisé pour la classification des MBs dans l'algorithme décrit au chapitre 3
$T_{mc}^{16 \times 16}$	Seuil utilisé pour la classification des MBs dans l'algorithme décrit au chapitre 3
T_r	Taux de rafraîchissement des trames intra
$T_{vm}^{8 \times 8}$	Seuil de raffinement des VMs d'entrée MPEG-4 Inter8×8 de l'algorithme décrit au chapitre 3
$T_{vm}^{16 \times 16}$	Seuil de raffinement du VM d'entrée MPEG-4 Inter16×16 de l'algorithme décrit au chapitre 3
$\mu_{8 \times 8, R/G}^{8 \times 8, t-1}$	Seuil de raffinement des VMs d'entrée MPEG-4 Inter8×8 de l'algorithme décrit au chapitre 5
$\mu_{16 \times 16, R/G}^{16 \times 16, t-1}$	Seuil de raffinement des VMs d'entrée MPEG-4 Inter16×16 de l'algorithme décrit au chapitre 5
μ_G	Moyenne de la somme absolue du résiduel d'une trame
$\mu_{mci, R}^{mco}$	Moyenne sur R_n de tous les MBs d'une trame transcodée de mci à mco
$\mu_{mci, R/G}^{mco, t}$	Moyenne sur $R_{n/G}^t$ pour tous les MBs $n \in S_{mci}^{mco, t}$ d'une trame t

$\mu_{mci,R/G}^{mco,t-1}$	Moyenne sur $R_{n/G}^t$ pour tous les MBs $n \in S_{mci}^{mco,t}$ de la trame à $t - 1$
\mathbf{v}_i	Vecteur de mouvement MPEG-4 de la partition i du mode de codage
$\mathbf{v}_{mco,k}$	Vecteur de mouvement de sortie H.264 associé à la partition k du MC H.264 <i>mco</i>
\mathbf{v}_{mp4}	Vecteur de mouvement MPEG-4
\mathbf{v}_p	Vecteur prédit en H.264 valeurs servent comme seuils de classification des MBs et comme seuil de raffinement des VMs MPEG-4

INTRODUCTION

Problématique

Le marché de la vidéo a connu une croissance phénoménale ces dernières années. Par exemple, un rapport de YouTube (2011) estimait qu'en 2010, plus de trois milliards de vidéos ont été visionnées par jour sur YouTube. Selon une analyse de Mahaney (2010) de Citigroup Investment Research, les revenus de YouTube ont atteint approximativement 412 millions de dollars en 2010 et devraient atteindre 848 millions de dollars en 2012. Le marché de la vidéo devrait continuer de croître exponentiellement pour encore plusieurs années. Selon ABI Research (2011), les revenus générés rien que par la vidéo mobile (la visiophonie, la messagerie, le partage vidéo ou la vidéo sur demande) devraient dépasser les 2 milliards de dollars d'ici 2013.

L'introduction des réseaux de troisième génération, offrant des débits binaires allant jusqu'à plusieurs centaines de kilobits par seconde (kb/s) pour du téléchargement, a contribué énormément à cette croissance. Ces réseaux ont permis le déploiement d'applications multimédias (notamment vidéos) sur des téléphones mobiles. Le caractère divertissant de la vidéo ainsi que la simplicité d'utilisation des médias sur lesquels la vidéo est supportée ont contribué également à accroître considérablement le marché de la vidéo. L'utilisateur peut voir des films, partager des souvenirs, visualiser des nouvelles, etc. La vidéo est par ailleurs supportée par des médias de masse (téléphones mobiles, téléphones intelligents, Internet, réseaux sociaux). Selon Pixmania (2011), 67% de la population mondiale est équipée de téléphones mobiles. Selon Stats (2011), 2,1 milliards de personnes utilisaient Internet en mars 2011. Les utilisateurs apprécient particulièrement la mobilité de ces supports qui leur permet d'être en contact permanent avec leurs proches. L'utilisateur peut, à tout moment et de n'importe quel endroit, partager de la vidéo avec ses proches ou s'informer visuellement en temps réels.

Plusieurs applications vidéos existantes sont supportées par le standard MPEG-4 Partie 2¹ (ISO/IEC (2004)). MPEG-4, développé par le groupe MPEG (Moving Picture Experts Group)

1. Dans cette thèse, nous utiliserons MPEG-4 pour désigner MPEG-4 partie 2.

d'ISO/IEC (International Standard Organisation/International Electrotechnical Commission), est une des normes les plus populaires actuellement. Il est adopté par les applications de téléchargement, de lecture en transit (*streaming*) sur Internet, de vidéo sur les terminaux mobiles, par les jeux vidéos et la télévision.

Toutefois, la toute dernière norme de codage vidéo, H.264 (UIT (2003)) offre des performances nettement supérieures, une qualité comparable pour des taux de compression 2 à 3 fois plus grands que MPEG-4, selon le profil choisi. H.264 a été développé par le groupe VCEG (Video Coding Experts Group) de l'UIT (Union internationale des télécommunications) en partenariat avec le groupe MPEG d'ISO/IEC. Cet effort de partenariat est connu sous le nom de JVT (Joint Video Team). H.264 a été complété en 2003 sous l'appellation H.264/AVC, H.264 au niveau de UIT et MPEG-4 AVC (Advanced Video Coding) au niveau d'ISO. Il est également désigné au niveau d'ISO comme *MPEG-4 Part 10*. Plusieurs applications l'ont adopté (films Blu-ray, applications vidéos sur téléphones intelligents, caméscopes AVCHD). En France, par exemple, H.264/AVC est utilisé dans les récepteurs de télévision HD depuis octobre 2008. Aux États-Unis, l'ATSC (Advanced Television Systems Committee) adoptait également en 2008, le format H.264/AVC. Les standards A/72 Part 1 et A/72 Part 2 ont alors été définis et approuvés. Le standard A/72, notamment A/72 Part 1, définit les caractéristiques H.264 de codage vidéo dans les systèmes de télévision numérique ATSC. La seconde partie, A/72 Part 2, quant à elle définit les caractéristiques H.264/AVC adoptées pour le transport de train binaire vidéo ATSC. La tendance est telle que H.264 remplace rapidement MPEG-4 dans toutes les nouvelles applications vidéos.

Le transcodage MPEG-4 à H.264 est devenu une nécessité pour deux raisons principales. La première est qu'il faut assurer la communication entre les terminaux supportant respectivement les formats MPEG-4 et H.264. En effet, les différences entre ces formats menant à l'incompatibilité de leurs spécifications au niveau du train binaire causent un sérieux problème d'interopérabilité entre les terminaux. Un terminal ne supportant qu'H.264 sera dans l'impossibilité de décoder un signal se conformant aux spécifications MPEG-4 envoyé par un autre terminal. C'est une situation réelle que l'on retrouve lorsqu'un terminal supportant MPEG-4

capture un clip vidéo et l'envoie à un terminal qui ne supporte qu'H.264. Pour que ce clip vidéo MPEG-4 soit décodé et affiché par le terminal H.264 qui doit le recevoir, il faudra d'abord le transcoder en un signal H.264 par un élément intermédiaire situé entre ces deux terminaux et agissant comme traducteur.

Ce problème d'incompatibilité est donc un problème majeur qui compromet sérieusement la communication entre des terminaux supportant les standards MPEG-4 (ISO/IEC (2004)) et H.264 (UIT (2003)). Ce problème est exacerbé par le fait que ces deux normes sont très présentes sur le marché et doivent donc coexister pour plusieurs années encore. MPEG-4, on le rappelle est une des normes les plus populaires actuellement et la présence d'H.264 croît très rapidement.

Le fait que de plus en plus d'applications tendent à adopter H.264 afin de bénéficier de ses taux de compression attrayants mène à la deuxième raison pour laquelle le transcodage MPEG-4 à H.264 est devenu primordial. Il est bénéfique pour des applications ayant accès à du contenu disponible en MPEG-4 de transcoder ce contenu en H.264 pour, d'une part, obtenir une meilleure qualité visuelle ou pour bénéficier d'un transport plus efficace, spécialement dans des contextes où la bande passante est très restreinte. En effet, un signal MPEG-4 transcodé en H.264 pourra bénéficier d'une taille de 2 à 3 fois plus faible tout en conservant la même qualité visuelle initiale. Ces taux de compression impressionnants permettent également d'assurer un transport plus efficace notamment dans des contextes de bande passante limitée.

L'approche simple de transcodage est de décoder complètement le signal vidéo reçu, c.-à-d. de le ramener dans le domaine des pixels avec un décodeur MPEG-4, puis de le réencoder selon les spécifications H.264 avec un encodeur H.264. On parle alors d'un transcodage cascadié dans le domaine spatial. Les meilleures performances en qualité sont normalement atteintes avec ce type de transcodage. Malheureusement cette approche est loin de convenir aux applications temps réel. En effet, la compression H.264 requiert une grande puissance de calculs.

Il est possible de réduire cette complexité de compression, mais, malheureusement, cela doit se faire au détriment de la qualité.

On pourrait toutefois exploiter l'information obtenue à l'étape de décodage MPEG-4 et la réutiliser judicieusement lors de l'étape d'encodage H.264 pour réduire les calculs. Mais comment bien exploiter cette information est un problème complexe. C'est pourquoi le transcodage de MPEG-4 à H.264 constitue un sujet d'actualité, d'autant plus que les enjeux économiques sont grands. L'importance de ce problème a été le sujet de nombreuses recherches (Shen (2004), Lee et al. (2006), Liang et al. (2007), Lee et Lee (2005)). Afin d'arriver à leurs fins, la plupart des algorithmes se servent d'informations extraites du train binaire MPEG-4 comme les vecteurs de mouvement (VMs), les modes de codage (MCs) etc. Ils les réutilisent au niveau du codeur H.264 afin d'éliminer ou de simplifier certaines étapes du processus de codage. Ils réduisent ainsi le temps de traitement. Malheureusement, ces algorithmes présentent certaines carences, qui sont brièvement présentées dans la section suivante et de façon plus détaillée dans le chapitre 2. Le véritable défi est donc de développer des algorithmes de transcodage qui minimisent la puissance de traitement tout en maintenant la meilleure qualité visuelle possible.

Les utilisateurs mobiles ne sont pas prêts à accepter des services de piètre qualité, au contraire, surtout que la barre a été placée assez haute avec la télédiffusion HD (haute définition). Pour être compétitif, il va falloir fournir des produits de bonne qualité à des utilisateurs de plus en plus exigeants. Seuls ceux qui sauront investir pour optimiser leurs réseaux et fournir des services de qualité sauront tirer leur épingle du jeu.

État de l'art et ses carences

Nous avons recensé dans la littérature les algorithmes qui traitent du transcodage MPEG-4 à H.264. Ils sont nombreux, mais la plupart effectuent le transcodage avec des opérations particulières telles que la réduction de résolution et de débit de trames. Peu d'entre eux traitent du transcodage d'un train binaire MPEG-4 à un train binaire H.264 (Shen (2004), Lee et Lee (2005), Lee et al. (2006), Liang et al. (2007)). Les algorithmes de transcodage MPEG-4 à

H.264 vidéo peuvent se diviser en deux catégories : ceux qui travaillent dans le domaine compressé et ceux qui se font dans le domaine spatial.

L'algorithme développé par Shen (2004) effectue le transcodage dans le domaine compressé. Le signal issu du décodeur MPEG-4 n'est pas ramené dans le domaine des pixels. Il existe cependant des différences majeures entre MPEG-4 et H.264, spécialement au niveau des transformées qui rendent ce type de transcodage difficile. L'auteur a donc utilisé une approche sous-optimale de matrices de transformées pour effectuer le transcodage. Par ailleurs le transcodage compressé souffre de sévères accumulations d'erreurs ; ce qui affecte grandement la qualité visuelle. Aussi, les approches fréquentielles sont moins flexibles et plus complexes à implémenter surtout pour l'estimation de mouvement.

Le transcodage dans le domaine spatial souffre moins d'accumulations d'erreurs. Les algorithmes détaillés en Lee et Lee (2005), Lee et al. (2006), Liang et al. (2007) travaillent dans ce domaine. En Lee et Lee (2005), Lee et al. (2006), les auteurs exploitent les VMs et les MCs MPEG-4. Les auteurs établissent des statistiques et dérivent une table de correspondance entre les MCs MPEG-4 et H.264. Ils déterminent ainsi les MCs H.264 les plus probables en fonction du MC MPEG-4 et ne testent que ces modes. Les VMs MPEG-4 sont réutilisés pour éviter de recommencer à zéro le coûteux processus d'estimation de mouvement (EM) et la qualité est améliorée en raffinant ces VMs. Bien que l'idée de tester uniquement les MCs les plus probables soit intéressante, la table a l'inconvénient de ne pas prendre en compte le débit binaire et le type de vidéo à transcoder. En effet la probabilité d'occurrence des MCs varie considérablement en fonction de ces deux facteurs. Par conséquent, les résultats obtenus sont assez inconsistants. À bas débits particulièrement, la perte en qualité est énorme. Par ailleurs, les auteurs de ces algorithmes donnent peu de détails sur le processus de réutilisation des VMs. Les résultats en accélération obtenus sont également peu fiables pour un contexte réel, les simulations ayant été validées sur des codecs de référence qui sont loin d'être optimaux en terme de vitesse. Ces accélérations seront nettement plus faibles une fois ces algorithmes implémentés sur des codecs commerciaux.

Liang et al. (2007) tout comme Lee et Lee (2005) et Lee et al. (2006), utilisent les VMs et les MCs issus de MPEG-4 pour sauter certaines étapes du codage H.264 et réduire ainsi le temps de traitement. Ils présentent une table de correspondance arbitraire entre les MCs MPEG-4 et H.264. Ils obtiennent de bonnes accélérations en vitesse de traitement. Toutefois l'algorithme ne tient aucunement compte du débit de codage et du type de la vidéo à transcoder. Comme conséquence, les résultats en qualité sont inconsistants et peuvent varier substantiellement selon le débit binaire et le type de vidéo. L'algorithme donne de mauvais résultats en qualité à bas et moyens débits (des pertes allant jusqu'à -6 dB), ce qui est inacceptable pour la plupart des applications. L'algorithme se comporte aussi moins bien pour certains types de vidéos. Par ailleurs, tout comme en Lee et Lee (2005) et Lee et al. (2006), l'algorithme a l'inconvénient d'avoir réalisé les simulations sur des codecs de référence.

Une analyse critique de l'état de l'art est présentée plus en détail dans le chapitre 2. Les méthodes de transcodage existantes, leurs avantages et leurs carences y sont détaillés. Ces carences nous poussent à développer des méthodes de transcodage MPEG-4 à H.264 plus performantes. Pour y arriver, tout comme les algorithmes existants, des informations seront extraites du train binaire MPEG-4 et exploitées afin d'alléger le processus de codage H.264. Nous nous servons notamment des MCs, des VMs et de l'information résiduelle. Cette dernière information sera exploitée judicieusement afin de réduire davantage la complexité tout en maintenant une bonne qualité. Nous exploiterons cette information tant au niveau du choix des modes que du raffinement des VMs. Elle nous servira aussi à adapter nos algorithmes aux conditions de débits et aux caractéristiques des vidéos à traiter.

Composition et contributions de la thèse

Cette thèse est structurée de la manière suivante :

Le chapitre 1 présente les principes du codage vidéo et les spécificités des standards MPEG-4 et H.264. Le chapitre 2 présente l'état de l'art sur le transcodage MPEG-4 à H.264. Des détails sont donnés sur les avantages et les carences des méthodes de transcodage existantes. Dans le chapitre 3, nous présentons une des méthodes de transcodage développées. L'algorithme in-

nove en proposant d'exploiter l'information résiduelle recueillie du décodeur MPEG-4 comme mesure de complexité du macrobloc (MB). En exploitant l'énergie résiduelle dans les processus de détermination des MCs et des VMs MPEG-4, l'algorithme arrive à accélérer davantage le codage H.264 tout en maintenant une bonne qualité. Dans un premier temps nous détaillons le processus de détermination des MCs et ensuite celui des VMs. Tout comme l'algorithme développé par Lee et al. (2006), une table est établie afin de déterminer les MCs H.264 les plus probables. En tirant avantage du fait que la taille du MC du MB augmente avec l'énergie résiduelle, nous réussissons à réduire de façon plus drastique l'ensemble des MCs à tester. Cette énergie est également exploitée de manière innovante dans le processus de détermination des VMs. Contrairement aux algorithmes de transcodage existants qui raffinent automatiquement tous les VMs MPEG-4, l'algorithme propose de ne raffiner que les VMs jugés inefficaces. L'énergie résiduelle sert de mesure d'efficacité. Cela évite des raffinements coûteux en calculs n'améliorant pas ou peu la qualité. Le chapitre est clos en présentant les résultats expérimentaux.

Dans les chapitres 4 et 5 nous présentons un nouvel algorithme de transcodage. Ce dernier est une variante de celui proposé dans le chapitre 3. De nouveaux outils et concepts sont exploités afin de pallier à l'inconsistance des algorithmes existants quant au débit de codage et aux caractéristiques des vidéos. Dans le chapitre 4, nous détaillons les concepts innovateurs utilisés. L'information résiduelle est exploitée différemment. Une mesure relative de l'information résiduelle est utilisée, la somme résiduelle relative, afin d'adapter l'algorithme au débit binaire et aux caractéristiques de chaque vidéo. Nous avons également tiré avantage du principe de corrélation entre les trames successives. Des statistiques rassemblées lors du transcodage d'une trame passée sont utilisées dans le transcodage de la trame suivante afin d'obtenir des seuils de classification nettement plus fiables et adaptés aux conditions de codage. Par ailleurs nous proposons d'utiliser conditionnellement de petites partitions de codage afin d'améliorer davantage la qualité tout en réduisant la complexité en calculs. De cette façon, nous avons pu développer un algorithme efficace pour toutes les conditions de codage. Le chapitre 5 est consacré à la description de l'algorithme. On y présente dans l'ordre suivant : le processus de

détermination des MCs, celui des VMs et pour finir les résultats en terme d'accélération et de qualité obtenues.

Dans le chapitre 6, nous présentons les contributions de notre recherche sur le transcodage MPEG-4 à H.264. Le dernier chapitre, la conclusion, fait l'objet de la conclusion de la thèse, en y résumant les résultats importants de la thèse, et en proposant de nouvelles bases de recherches éventuelles.

CHAPITRE 1

LES PRINCIPES ET STANDARDS DE CODAGE VIDÉO

1.1 Introduction

Avant de présenter l'état de l'art et les algorithmes développés pour le transcodage MPEG-4 à H.264, nous allons décrire les principes de base de la compression vidéo et comment ils sont utilisés dans les standards MPEG-4 et H.264. Nous comparerons ensuite les spécificités de ces deux standards et leurs différences. Le lecteur pourra ainsi comprendre plus aisément les éléments dont on parlera dans la suite de la thèse.

1.2 Principes du codage vidéo

Dans cette section, nous présentons les principes de base du codage vidéo. Nous y présentons le schéma de principe d'un codeur vidéo, les concepts de trames intra et inter, de redondances spatiale et temporelle, d'estimation du mouvement, de précision des vecteurs de mouvement, etc.

1.2.1 Schéma de principe d'un codeur vidéo

Le schéma de principe d'un codeur vidéo est présenté à la figure 1.1. Plus de détails sur les éléments de base du codage vidéo peuvent être trouvés dans Nian et al. (2004). Il faut préciser que ce schéma ne décrit que les éléments de base d'un codeur vidéo. Ce schéma peut varier à quelques différences près, d'un standard et d'un concepteur à un autre. Néanmoins, les éléments de base détaillés à la figure 1.1 s'y retrouvent toujours.

Les blocs DCT (Discrete Cosinus Transform), Q, IQ, IDCT (inverse DCT), VLC (Variable Length Coding) représentent respectivement les processus de transformée en cosinus discrète, de quantification, de quantification inverse, de transformée en cosinus discrète inverse et de codage à longueur variable (codage entropique). Normalement, les images à encoder sont segmentées en blocs de taille fixe appelés macroblocs (MB) qui peuvent ensuite être décomposés

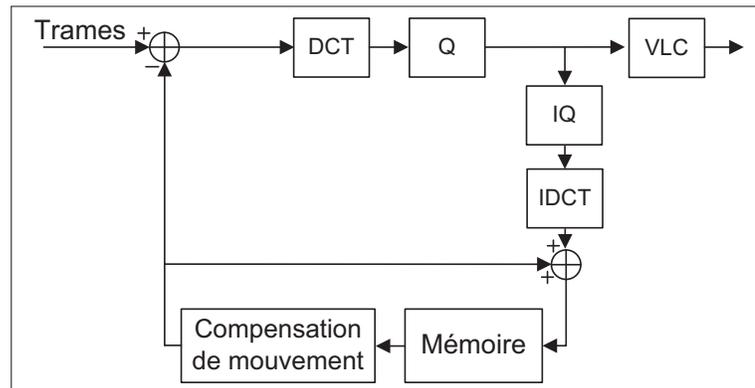


Figure 1.1 Schéma de principe d'un codeur vidéo.

en blocs de taille inférieure. Les blocs résultants passent par les diverses étapes de la figure 1.1. Tout d'abord, le bloc de compensation de mouvement génère un prédicteur efficace pour réduire la redondance temporelle entre les blocs de la séquence vidéo. Cette prédiction est soustraite du bloc à coder pour former un bloc résiduel qui passera par l'opération DCT. La DCT a pour objectif de réduire la redondance spatiale du bloc résiduel en transformant ce dernier dans le domaine fréquentiel. Les coefficients fréquentiels sont ensuite quantifiés pour réduire le nombre de bits nécessaires pour les représenter. La quantification contribue ainsi à réduire la quantité d'information à transmettre, mais entraîne des pertes. Plus la quantification sera agressive, plus le taux de compression sera élevé, mais au détriment de la qualité. Le codage entropique est finalement appliqué au bloc quantifié pour former le signal vidéo compressé final. Les étapes IQ et IDCT servent à reconstruire le bloc transmis qui servira de référence (prédiction) pour la prochaine trame à encoder. Notons que certaines images (ou trames) sont codées complètement, sans référence à d'autres images, c.-à-d. sans utiliser le bloc compensé par le mouvement comme prédicteur. Ce sont des trames intra. Elles sont notées généralement trame I. D'autres trames se servent de blocs appartenant à des trames précédentes ou futures comme prédicteur. Lorsqu'une trame se sert uniquement de trames précédentes, on parle de trame de type P. Dans le cas où elle utilise des trames précédentes et futures pour la prédiction, on parle de trame de type B. Ces trames faisant usage de la redondance temporelle sont appelées trames inter. Les prochaines sous-sections présentent plus en détail les concepts de redondance temporelle et spatiale.

L'encodage vidéo est précédé d'une conversion de l'image de l'espace de couleur RGB (red, green, blue) à l'espace de couleur YCbCr (luminance, chrominance bleue, chrominance rouge). Ceci n'est cependant pas illustré à la figure 1.1. On effectue cette conversion tout simplement à cause des propriétés du système visuel humain. Le système visuel humain est plus sensible à l'information de luminance (intensité lumineuse) qu'à celle de chrominance (couleur). Les informations de chrominance peuvent ainsi être sous-échantillonnées, l'oeil y étant moins sensible. Cela permet de réduire la quantité d'information du signal initial. Il faut noter toutefois que le sous-échantillonnage de la chrominance n'est pas obligatoire et dépend des applications ciblées. La luminance est toujours traitée intégralement. On utilise la notation $4 : 2 : 0$ pour désigner respectivement un signal dont les signaux de chrominance ont été sous-échantillonnés par un facteur 2, horizontalement et verticalement par rapport au signal de luminance. La notation $4 : 2 : 2$, quant à elle, désigne un signal où les signaux de chrominance ont été sous-échantillonnés de moitié, horizontalement seulement. Pour la notation $4 : 4 : 4$ aucun sous-échantillonnage n'a été effectué. Les trois composantes résultantes sont ensuite traitées séparément. Une fois cette étape terminée, on commence alors par effectuer le processus de compression proprement dit. Pour ce faire, on élimine les redondances spatiale et temporelle des séquences de la vidéo, pour une compression maximale.

1.2.2 Exploitation de la redondance temporelle

La redondance temporelle est réduite en tenant compte de l'observation suivante : d'une image à une autre, peu d'endroits changent et les changements sont souvent dus à des déplacements d'objets dans l'image. Les trames inter exploitent ce fait. Pour ces trames, on ne code que la différence entre l'image présente et l'image estimée à partir d'une trame passée (ou future). En fait, cette opération est effectuée bloc par bloc dans l'image. Le module de compensation de mouvement (CM) de la figure 1.1 est chargé de trouver, dans l'image passée (ou future), le bloc correspondant le mieux au bloc à encoder. Puisque les objets peuvent se déplacer spatialement d'une image à l'autre, le bloc représentant le mieux un bloc à encoder n'est pas nécessairement situé au même endroit spatial que ce dernier. Le module d'estimation de mouvement doit ainsi déduire les vecteurs de mouvement (le déplacement spatial) qui matérialisent le

déplacement en question. Le processus d'estimation est tel que montré à la figure 1.2. Le processus d'estimation de mouvement est illustré en prenant le macrobloc MB_c comme exemple. On recherche, dans la trame passée, le MB qui correspond le mieux au MB_c . Le MB_r de l'image passée, qui se trouve à la position spatiale de MB_c déplacé du vecteur $\mathbf{v}=(v_x, v_y)$, est le meilleur bloc correspondant. Le MB_c pourra alors être reconstitué, notamment au décodage, à partir de la trame passée, en appliquant un déplacement (v_x, v_y) au bloc MB_r . Si l'estimation

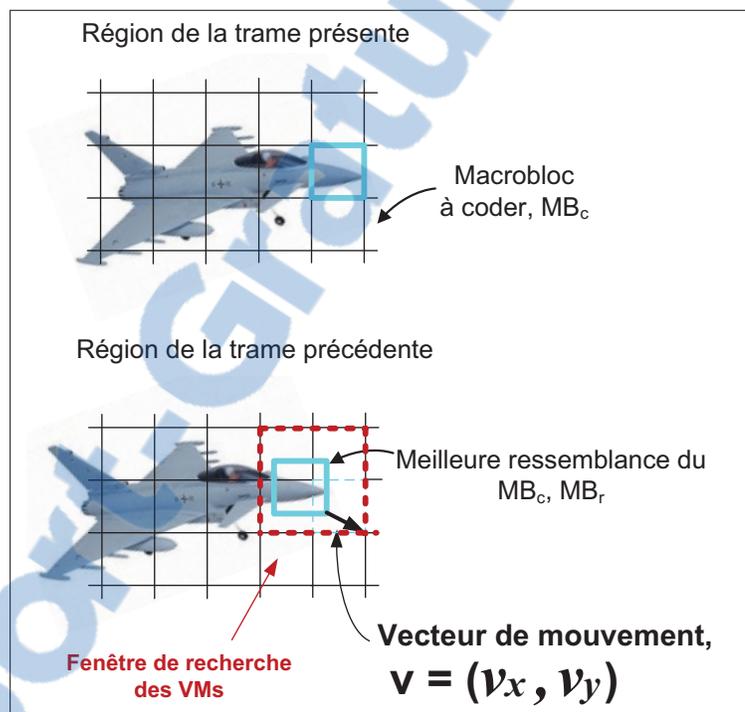


Figure 1.2 Principe de l'estimation de mouvement.

de mouvement est assez bonne, la différence entre le bloc de l'image présente et le bloc estimé (reconstitué par le bloc de compensation de mouvement, voir la figure 1.1) sera pratiquement nulle. Le bloc résiduel ainsi créé (disponible après le module de soustraction) contiendra des pixels de faible amplitude et nécessitera, par conséquent peu de bits pour être codé. Au lieu de transmettre le bloc au complet, on ne transmet que le bloc résiduel et le vecteur de mouvement (VM) qui servira à le reconstruire. Ce principe est appliqué à toute la trame. Une fois le bloc résiduel obtenu, on élimine sa redondance spatiale par le module DCT, comme expliqué en

détail à la sous-section suivante. Il faut noter aussi que tout comme pour les blocs inter, on peut créer une prédiction pour les blocs intra. Dans ce cas, c'est l'erreur de prédiction ou le résiduel qui passera par la DCT. Les modules de DCT inverse et de quantification inverse illustrés à la figure 1.1 servent à la reconstruction des blocs de l'image actuelle. Ils interviendront dans la prédiction et la compensation de mouvement dans le codage de l'image suivante.

1.2.3 Exploitation de la redondance spatiale

La redondance spatiale, quant à elle, est éliminée en procédant comme suit : les blocs de l'image passent par le bloc DCT illustré à la figure 1.1. La DCT se charge de représenter le bloc dans l'espace fréquentiel. La plus grande concentration de l'énergie des coefficients du bloc se retrouve dans les basses fréquences qui sont relatives à l'allure générale du bloc. Les hautes fréquences, quant à elles, sont relatives aux détails du bloc ; le système visuel humain y est moins sensible. L'image est ensuite compressée en lui appliquant une quantification (bloc Q à la figure 1.1). Évidemment la quantification dans les basses fréquences sera plus précise que celle effectuée dans les hautes. Cela permet une meilleure utilisation, d'un point de vue débit-distorsion (ou qualité versus débit), des bits disponibles. Une fois la redondance spatiale exploitée, on applique aux coefficients quantifiés un codage entropique. Ce module effectue une compression sans perte. Son rôle est de réduire la longueur du train binaire en supprimant les redondances statistiques toujours présentes dans l'image.

1.2.4 Estimation de mouvement

L'estimation de mouvement (EM) est une des opérations importantes d'un codeur vidéo. Elle jouera aussi un rôle très important dans le développement de nos algorithmes de transcoding. L'efficacité de compression du codeur dépend grandement de l'algorithme de recherche de mouvement utilisé. Il est aussi un des modules les plus complexes en calculs. A titre d'exemple, une analyse faite dans Shengfa et al. (2006) estime que le module d'estimation de mouvement occupait à lui seul 56% du temps de traitement d'un encodage H.264. Nous pouvons voir, à la figure 1.3, adaptée de Shengfa et al. (2006), la répartition du temps de codage qu'occupent les principales fonctions de codage. L'analyse a été faite pour le codeur

x264 de VideoLAN (2011) avec l'outil VTune Performance Analyzer d'Intel (2009). Il existe

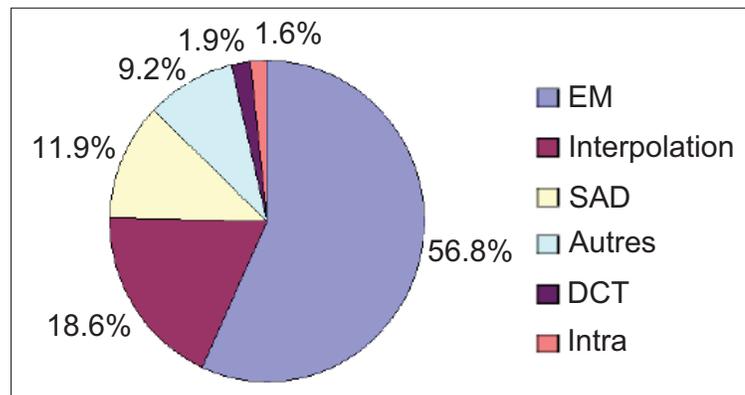


Figure 1.3 Répartition du temps de calcul des fonctions de codage du H.264. Adaptée de Shengfa et al. (2006).

plusieurs algorithmes de recherche de vecteurs de mouvement (VM). L'algorithme FSA (full search algorithm) donne d'excellents résultats en qualité, mais malheureusement ce dernier est loin de convenir aux applications temps réel. Il est très complexe en terme de calculs (beaucoup de calculs). Cette recherche de vecteurs de mouvement est effectuée de manière exhaustive, pour tous les déplacements entiers situés, dans une fenêtre centrée autour du bloc et généralement de longueur de 33 positions horizontales par 33 verticales (± 16 en x et y) pour plus de 1000 déplacements à tester. Le meilleur bloc est sélectionné en minimisant le rapport débit/distorsion.

Notons $D(x, y)$, la distorsion entre le MB, dont on recherche le VM, représenté par $I(k, l)$ et le MB choisi pour la prédiction, représenté par $J(k + x, l + y)$. Soit $coutbits(V(x, y))$, le coût en bits du vecteur $V(x, y)$. Le VM choisi sera celui qui minimise le coût débit distorsion tel que montré dans l'équation 1.1.

$$(x, y) = \underset{(x, y) \in F}{\operatorname{arg\,min}} [D(x, y) + coutbits(V(x, y))] \quad (1.1)$$

La mesure de distorsion est généralement la SAD (sum of absolute differences) ou la SATD (sum of absolute transformed differences) dans le cas de H.264. Si on choisit comme critère de distorsion la SAD, alors le VM choisi est celui qui minimise l'équation 1.2.

$$(x,y) = \arg \min_{(x,y) \in F} \left[\sum_{k=0}^{15} \sum_{l=0}^{15} |I(k,l) - J(k+x,l+y)| + \text{coutbits}(V(x,y)) \right] \quad (1.2)$$

Avec F , la fenêtre de recherche du VM. Le principe de recherche de VMs est illustré à la figure 1.4. De nombreux algorithmes plus rapides que le FSA ont été développés afin de pallier sa lourdeur.

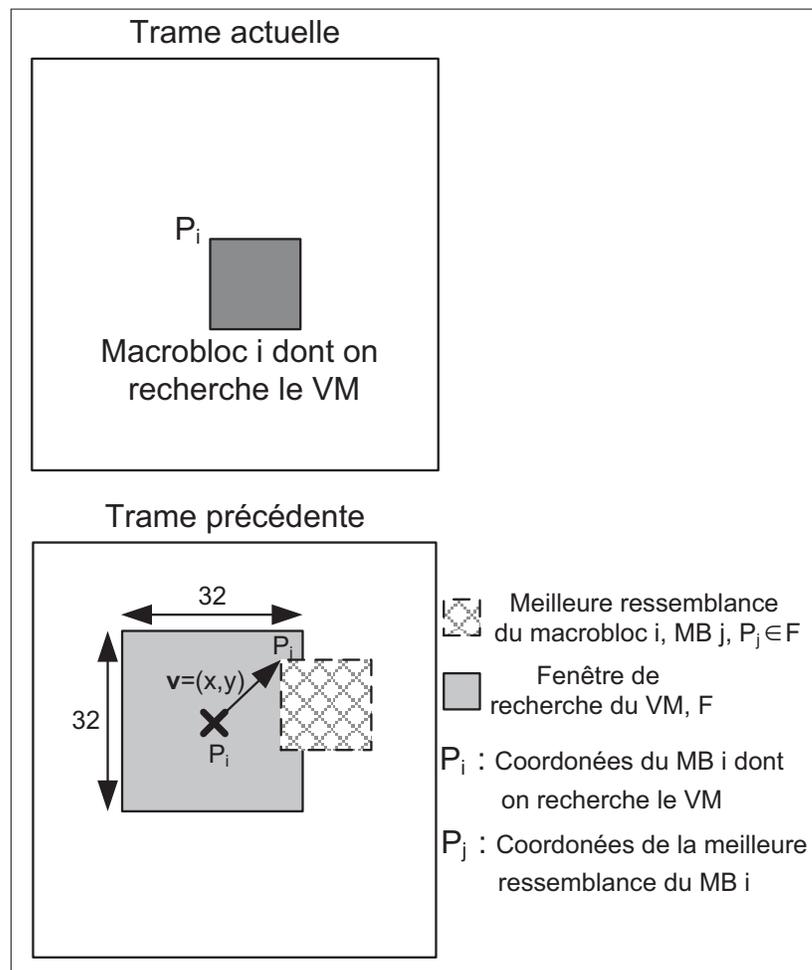


Figure 1.4 Illustration de la recherche des vecteurs de mouvement.

L'auteur C. Duanmu a effectué une étude comparative des principaux algorithmes de recherche de mouvement en se basant sur la distorsion qualité/débit. Les détails peuvent être trouvés dans la référence Duanmu (2005). Au terme de son analyse, il conclut que les algorithmes les plus efficaces de la littérature sont le PSAA (Predictive Search Area Algorithm, Chung et Chang (2003)), le PMVFAST (Predictive Motion Vector Field Adaptive Search Technique, Tourapis et al. (2002)). Les algorithmes PMVFAST et EPZS (Tourapis et Tourapis (2003)) sont parmi les plus efficaces de la littérature actuelle. Ces algorithmes donnent des résultats en qualité se rapprochant respectivement de 0,06 et 0,02 dB du FSA tout en obtenant des gains de vitesse impressionnants, des temps d'exécution respectivement 215 et 35 fois plus rapides que le FSA. Il faut aussi préciser que le choix de l'algorithme dépend de plusieurs facteurs, notamment, du type d'application et des plateformes sur lesquelles il sera déployé telles qu'une application temps réel, une application limitée au niveau de la puissance de traitement ou de la durée de la vie de la pile. Il faut préciser qu'en dehors de l'algorithme de recherche de mouvement utilisé, la précision des VMs influence aussi énormément la qualité de codage. La notion de précision des VMs est explicitée dans la section suivante.

1.2.5 La précision des vecteurs de mouvement

Tel qu'expliqué précédemment, les VMs matérialisent le déplacement entre la position spatiale du MB à coder et celle du MB de la trame précédente la plus similaire. Une meilleure correspondance pourrait être trouvée en interpolant des pixels à des positions fractionnaires dans la trame précédente. Lorsque l'interpolation est faite sur des demis, des quarts ou des huitièmes de positions, on dit que la précision des VMs est respectivement à la moitié, au quart ou au huitième de pixel. Il faut cependant préciser que, dans la recherche du VM, le codeur ne recherche pas le meilleur prédicteur uniquement sur les positions fractionnaires concernées, mais également sur les positions entières. La notion de précision des VMs est illustrée à la figure 1.5.

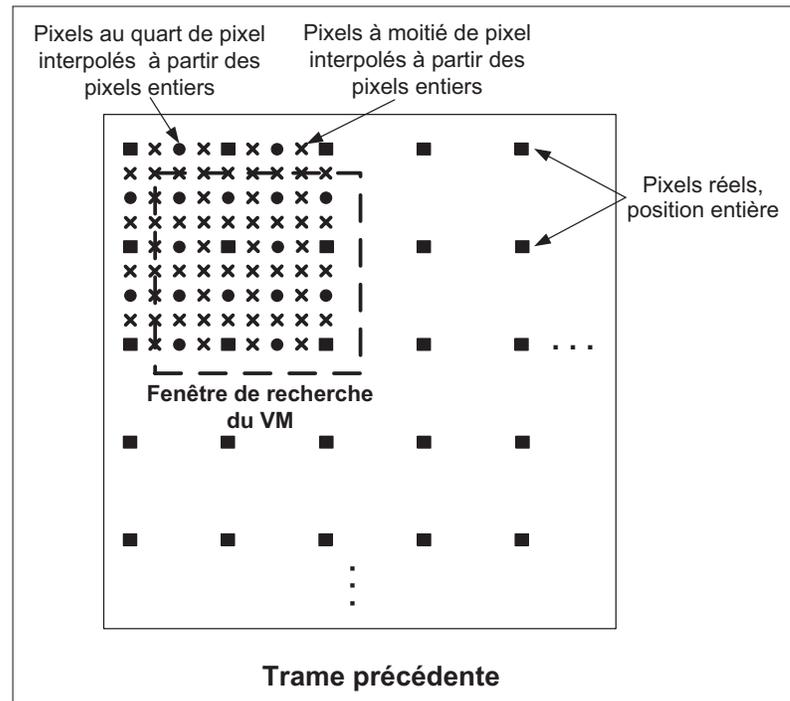


Figure 1.5 Illustration de la notion de la précision des VMs.

1.3 Le décodage vidéo

Le schéma de principe d'un décodeur vidéo est représenté à la figure 1.6. Il faut noter que, d'un standard à un autre, le schéma bloc diffère un peu, mais les principes de décodage restent les mêmes. Le décodeur vidéo réalise la fonction inverse du codage. Il fonctionne comme suit. Une fois le train binaire reçu, on effectue un décodage entropique inverse (noté VLC^{-1}) selon le codage entropique effectué au codage. Les codages arithmétiques et de Huffman sont souvent utilisés pour cette opération. Ensuite, le décodeur effectue une quantification inverse suivie d'une DCT inverse. Leur rôle est d'appliquer le processus inverse des blocs Q et DCT réalisé dans le codeur afin de reconstituer la trame résiduelle pour une trame inter, ou la trame originale, pour une trame intra. Le bloc mémoire, comme son nom l'indique, sert à mémoriser l'image reconstruite. Cela nous donnera accès à la trame passée reconstruite pour que nous puissions, ensuite, y appliquer la CM. Nous ajouterons par la suite le résultat au résiduel pour des blocs inter, permettant ainsi de reconstituer la trame vidéo. Pour chaque bloc inter, la CM utilise le VM transmis et la trame dans la mémoire pour déduire l'image prédite. VLC^{-1} , IQ,

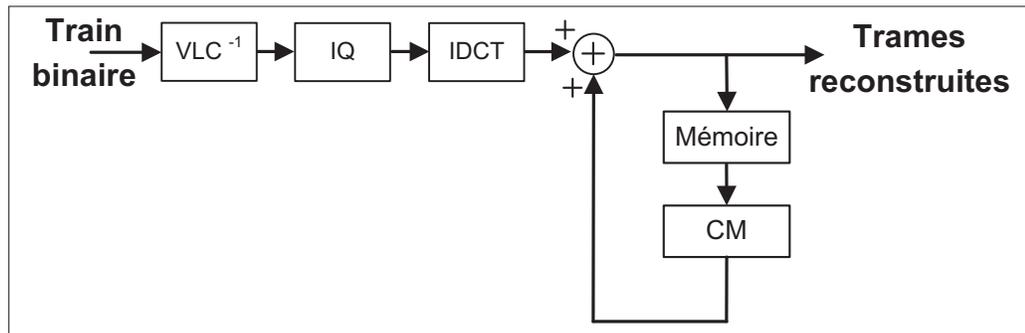


Figure 1.6 Schéma de principe d'un décodeur vidéo.

IDCT et CM sont, respectivement, le décodage entropique inverse, la quantification inverse, la transformée en cosinus discrète inverse et la compensation de mouvement.

1.4 Le standard MPEG-4

Plusieurs standards ont été développés dans le domaine de la vidéo, notamment MPEG-1, ISO/IEC (1993b), MPEG-2, ISO/IEC (2000), MPEG-4, ISO/IEC (2004), H.263, UIT (1996), et H.264, UIT (2003). Le standard MPEG-4, ISO/IEC (2004) a été finalisé par le groupe MPEG de ISO/IEC en 1998. C'est une norme orientée objet qui a été définie pour fonctionner sur des réseaux mobiles sans fil et satellitaires. Elle offre des fonctionnalités de codage plus avancées que la norme H.263 et une bonne robustesse aux erreurs. Elle a été définie pour supporter un grand nombre d'applications. Le codage d'objets rectangulaires, de formes arbitraires, de vidéos naturelles, synthétiques, hybrides, d'animation de visages sont tous supportés. Cet éventail de fonctionnalités est rendu possible par des algorithmes de compression avancés et un large ensemble d'outils pour le codage et la manipulation de ces médias. Il est invraisemblable qu'une seule application requiert tous les outils disponibles dans le standard MPEG-4. La notion de profil prend alors son importance. Le standard décrit donc une série de profils regroupant un ensemble d'outils adaptés à un certain type d'application. Au total, 19 profils sont définis dans le standard. Le plus usuel est le *simple profile*. Les concepteurs de codecs pourront choisir de conformer leur produit à un profil particulier. Ce dernier devra alors supporter tous les outils de codage qui lui sont relatifs, tels que définis dans le standard. Un

codec implémenté sur un processeur à faible puissance pourrait utiliser le profil simple. S'il s'agit d'une application de *streaming* vidéo, le codec pourrait plutôt choisir le profil *Advanced real time simple profile*. Des détails sur le MPEG-4 sont présentés dans l'ouvrage Richardson (2004).

Pour effectuer le codage, le codeur décompose chacune des trames en blocs de 16×16 pixels, appelé macrobloc (MB), et procède successivement au codage de chaque MB de la trame. MPEG-4 supporte des trames de type intra, de type inter P et de type inter B.

Il existe également plusieurs MCs (mode de codage) pour ses MBs de taille 16×16 pixels. Le MC représente le partitionnement du MB choisi par le codeur pour coder efficacement le MB. MPEG-4 supporte 4 modes de codage, tel qu'illustré dans la figure 1.7 : un mode intra 16×16 et trois modes inter (les modes Skip, Inter 16×16 et Inter 8×8). Une trame de type intra ne contient que des MCs intra tandis qu'une trame inter peut contenir tous les 4 MCs mentionnés. Le mode intra signifie qu'aucune prédiction temporelle n'est faite et les pixels du MB y sont codés intégralement. Par contre dans le mode inter, on code uniquement l'erreur de prédiction du MB. Le mode skip est un mode où aucun VM n'est transmis. En fait, il correspond au mode Inter 16×16 où le VM est le vecteur nul et le résiduel n'est pas transmis. Les pixels de la trame passée, à la position actuelle du MB à coder, sont alors utilisés pour la prédiction. La précision des VMs peut être au demi ou au quart de pixel dépendamment du profil. Le demi-pixel est utilisé pour tous les profils sauf le *advanced simple profile*, qui admet le quart de pixel.

1.5 Le standard H.264/AVC

Le standard H.264/AVC a été développé conjointement par l'UIT et ISO et complété en 2003, dans un effort de partenariat connu sous le nom de Joint Video Team (JVT). Il est appelé H.264 dans le contexte UIT et MPEG-4 AVC (advanced video coding) dans le contexte ISO. Il offre un mode de codage avancé et de bonnes performances de codage. La figure 1.8, adaptée de Sokol (2011), nous donne une idée des performances de H.264, en termes de bande passante et de taille par rapport au standard MPEG-2 et MPEG-4. Les résultats ont été obtenus considérant

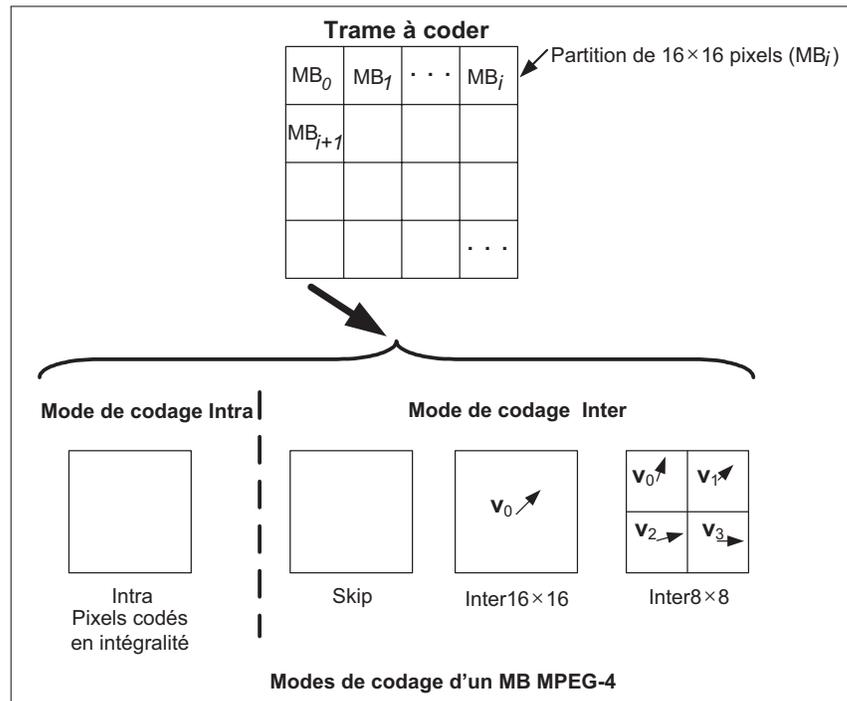


Figure 1.7 Modes de codage intra et inter de la norme MPEG-4.

une même qualité vidéo DVD de 90 min pour tous les standards. On peut constater que la bande passante et la taille du train binaire de H.264/AVC sont nettement inférieures à celles de MPEG-2 et MPEG-4.

Le standard H.264 (Richardson (2004)), a été premièrement développé pour supporter efficacement le codage et le transport de vidéos à trames rectangulaires. Le but était de fournir des fonctionnalités similaires aux standards H.263+ et MPEG-4 (*simple profile*), mais avec un taux de compression plus élevé et une robustesse plus grande. Les applications cibles incluent les communications à deux voies comme la vidéoconférence, la vidéophonie, le codage pour du *broadcast*, du *streaming* vidéo à travers les réseaux par paquets.

1.5.1 Le codeur et le décodeur H.264/AVC

Le standard H.264, comme les autres standards vidéos, ne définit pas explicitement un codec, mais plutôt la syntaxe du train binaire codé et la méthode de décodage de ce train binaire. Toutefois, la syntaxe et le processus de décodage imposent certaines contraintes à la conception

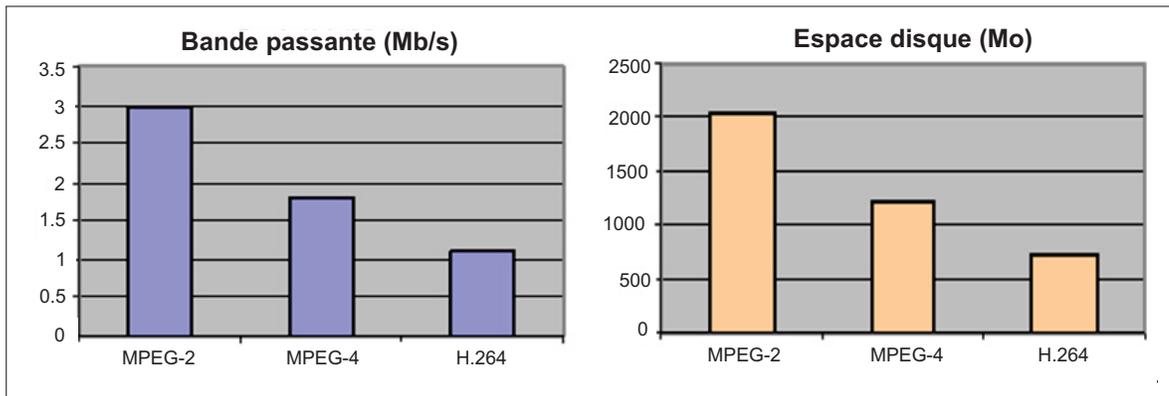


Figure 1.8 Performances du H.264 par rapport au MPEG-4 et au MPEG-2. Adaptée de Sokol (2011).

du codeur. Pratiquement, un codeur et un décodeur H.264 se présentent tels qu'illustrés dans les figures 1.9 et 1.10.

Le bloc CE représente le codage entropique. Le codeur H.264 inclut les mêmes fonctions de base (prédiction, transformée, quantification, codage entropique) que celles présentes dans les standards précédents, à l'exception du filtre anti-blocs (*deblocking filter*). Mais il faut noter qu'il existe des différences importantes quant aux outils de codage de ces fonctions. Dans H.264, les processus de transformée et de quantification sont combinés. Le bloc Tr+Q est donc chargé de réaliser simultanément la fonction de transformation fréquentielle et la fonction de quantification. Ces blocs jouent exactement le même rôle que ceux d'un codeur vidéo tel qu'expliqué précédemment, à l'exception que les deux blocs sont ici combinés en une seule opération. Le bloc I(Tr+Q) effectue le processus inverse de Tr+Q. À la différence des autres standards, H.264 utilise un filtre anti-blocs. Son rôle est de réduire les effets de blocs dus à la segmentation de l'image en MB et à la quantification.

1.5.2 Spécificités du standard H.264

Le standard H.264 apporte des fonctionnalités qui lui sont spécifiques, notamment :

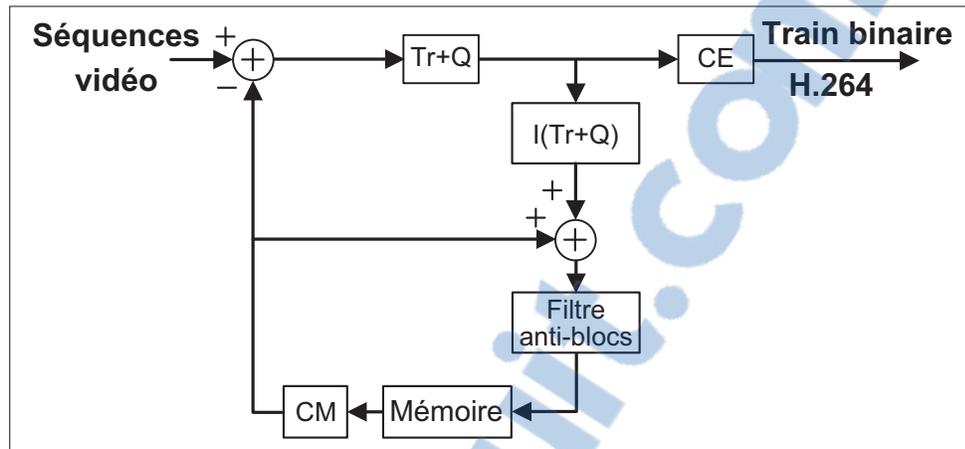


Figure 1.9 Codeur H.264.

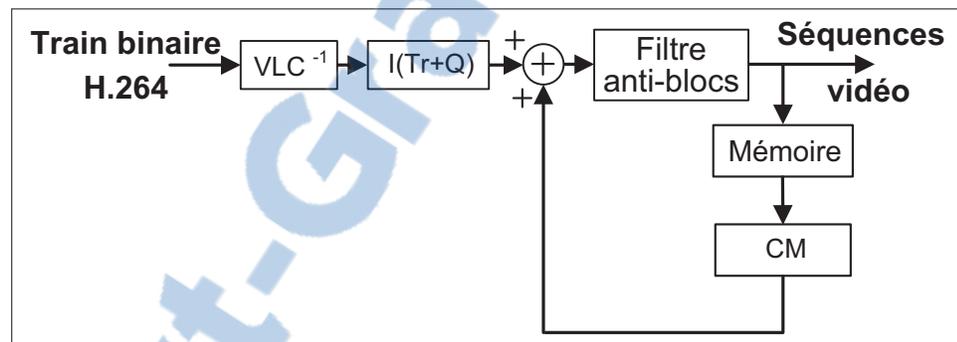


Figure 1.10 Décodeur H.264.

- **Les types de trames supportés.** En dehors des trames de type I (intra), P et B (inter), le standard H.264 supporte des trames de type SP et SI. Ces dernières ont été conçues pour permettre efficacement le changement de trains binaires et, également, pour faciliter l'accès aléatoire dans un décodeur vidéo.
- **Les images de référence.** Contrairement aux autres standards, H.264 utilise une ou plusieurs images de référence, jusqu'à 16 trames, pour la compensation de mouvement. Le codeur peut chercher la meilleure ressemblance du macrobloc courant parmi un vaste ensemble d'images, plutôt qu'uniquement dans l'image précédente. Cela est particulièrement utile dans les cas d'occlusion.

- **Le partitionnement des blocs et des vecteurs de mouvement.** L'une des importantes différences entre ce standard et les précédents est qu'il supporte un grand nombre de MCs des macroblocs. H.264 supporte 3 modes intra (Intra16×16, Intra4×4 et un mode intra PCM). Le mode intra PCM est le mode intra où les pixels sont codés en intégralité sans prédiction. C'est un mode sans perte. En fait, il correspond au mode intra de MPEG-4. Les 2 autres modes, Intra4×4 et Intra16×16, utilisent des prédictions intra. Les pixels sont prédits à partir des blocs de la trame présente qui ont déjà été codés. La prédiction peut être faite par blocs 4×4 ou 16×16, d'où les termes Intra4×4 et Intra16×16. H.264 supporte un total de 9 modes de prédiction Intra4×4 et 4 modes de prédiction Intra16×16. En ce qui concerne les modes inter, H.264 supporte 5 modes inter (Skip, Inter16×16, Inter16×8, Inter8×16 et Inter8×8). Si le mode Inter8×8 est choisi, les blocs 8×8 peuvent être encore subdivisés de 4 manières. Les 4 sous-modes sont Inter8×8, Inter8×4, Inter4×8 et Inter4×4. Les MCs et les sous-MCs permis par H.264 sont décrits dans la figure 1.11. La définition du mode skip n'est pas la même que celle de MPEG-4. H.264 affecte également un vecteur au mode skip, qui correspond au vecteur médian obtenu à partir de ceux des MB voisins. Tout comme dans le cas de MPEG-4, aucun résiduel n'est transmis. Ainsi, un MB est susceptible d'être codé dans ce mode s'il suit un mouvement identique à celui des MBs voisins.

À chaque partition est associé un VM, dont la précision est au quart de pixel. Choisir un MC à large partition nécessiterait moins de bits (peu de VMs à coder), mais le résiduel pourrait contenir plus d'énergie, surtout dans les régions avec peu d'uniformité de mouvement. Pour un MC à petites partitions, le résiduel contiendrait moins d'énergie, mais par contre, nécessiterait plus de bits pour les VMs. Le choix du MC a une influence significative sur l'efficacité de la compression. En général, les MCs à larges partitions conviendraient mieux aux régions homogènes ou à mouvements peu complexes.

- **La prédiction avancée des VMs.** Coder un VM pour chaque macrobloc peut coûter cher en bits, spécialement si de petites partitions sont utilisées. Les VMs des macroblocs voisins

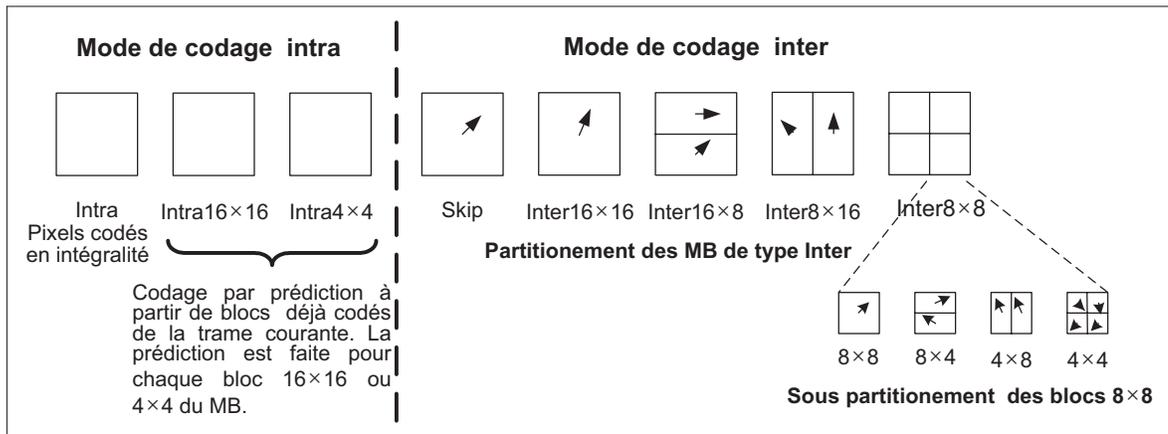


Figure 1.11 Modes de codage permis par H.264.

sont souvent corrélés et, donc un VM du bloc courant peut être prédit à partir des voisins et des précédents. Ce n'est que la différence entre le VM calculé et celui qui a été prédit qui est codée et envoyée. Évidemment, cette différence demandera moins de bits pour être codée, d'autant plus que les valeurs du VM réel et celles du VM prédit sont assez proches. La différence se rapprochera donc souvent de zéro.

- **Le filtre anti-blocs.** Ce filtre est utilisé pour atténuer les effets de blocs du codage vidéo et, également, pour réduire le bruit dans l'image. Il est appliqué avant le bloc mémoire (voir la figure 1.9). Il adoucit les bordures des blocs. Il est appliqué aux bords horizontaux et verticaux dans un macrobloc et son degré d'adoucissement est ajustable selon le débit.
- **La transformée entière.** H.264 utilise trois types de transformées selon le type de données à coder. Lorsque le macrobloc est codé en mode Intra 16×16, H.264 utilise une transformée de Hadamard pour les coefficients 4×4 DC de luminance, une transformée de Hadamard pour les coefficients 2×2 DC de chrominance et une transformée DCT entière pour tous les autres cas.
- **Le codage entropique.** En ce qui concerne le codage entropique, les données sont codées en utilisant un codage à longueur variable, généralement le codage de Huffman, un CABAC

(Context Adaptive Binary Arithmetic Coding), un CAVLC (Context Adaptive VLC) ou un codage Exp-Golomb. Le CAVLC est utilisé pour coder les coefficients résiduels. Il a été conçu pour tirer avantage de plusieurs caractéristiques des macroblocs quantifiés. Une méthode est utilisée pour compacter les nombreux zéros des coefficients après quantification. En effet, les coefficients non nuls après l'ordonnement zigzag sont souvent des séquences de ± 1 et le CAVLC compacte le nombre de coefficients ± 1 . Par ailleurs, le CAVLC utilise une table de codage dont le choix des paramètres tient compte de la corrélation des coefficients non nuls dans le voisinage et également, de la grande amplitude des coefficients près du DC. Le CABAC quant à lui permet d'atteindre de bonnes performances de codage par la sélection de modèles de probabilité pour chaque élément, en adaptant cette probabilité par des statistiques locales et en utilisant un codage arithmétique plutôt qu'un codage à longueur variable.

- **La précision des VMs.** La précision des VMs est au quart de pixel pour la compensation de mouvement de la luminance. Pour la chrominance, cette précision peut se faire jusqu'au huitième de pixel.

Toutefois les performances du H.264 sont obtenues au prix d'une grande complexité. En effet, les outils de codage tels que les MCs permis, la précision des VMs au quart de pixel, le nombre de trames de référence permises et la possibilité d'utiliser un codage arithmétique alourdissent considérablement le processus de codage.

Les spécificités citées plus haut sont supportées selon le profil. Le standard H.264 comporte 7 profils : le *Baseline*, le *Main*, le *Extended*, le *high*, le *high 10*, le *high 4 :2 :2* et le *high 4 :4 :4 profile*, supportant chacun un ensemble de fonctions de codage adaptées à un certain type d'applications. Présentons brièvement les profils les plus utilisés. Pour plus de détails sur les autres profils, on peut consulter Richardson (2004).

- **Le profil *baseline*** supporte le codage intra et inter (trames de type I et P) et le codage entropique avec le CAVLC seulement (pas de trames B ni de CABAC). Les applications poten-

tielles du *Baseline* sont la vidéophonie, la vidéoconférence et les communications sans fil.

- **Le profil *main*** supporte, quant à lui, la vidéo entrelacée, les trames de type B, le codage à prédiction pondérée et le codage entropique CABAC. Ses applications potentielles sont la télévision et le stockage vidéo.
- **Le profil *extended*** ne supporte pas la vidéo entrelacée et le codage arithmétique, mais ajoute des modes (les tranches de type SP et SI) permettant le passage efficace entre différents trains binaires et l'accès aléatoire pour le décodage vidéo. Cela peut être particulièrement intéressant pour des applications de type VCR (video cassette recording). La robustesse aux erreurs est également améliorée, avec ce profil, par l'introduction du partitionnement de données. Le profil *Extended* s'adapte davantage aux applications de *streaming*. Un outil utilisé fréquemment par le *streaming* vidéo est le passage entre un ou plusieurs trains binaires codés. Une vidéo peut être, par exemple, codée à des débits binaires différents pour une transmission par Internet. Le décodeur décode le train binaire dont le débit est le plus élevé, mais pourrait passer à un débit plus faible dans le cas où des données seraient perdues. Les tranches SI et SP facilitent cette façon de faire.

1.6 Différences entre les standards MPEG-4 et H.264

Dans notre thèse, nous mettrons l'accent sur le transcodage du profil simple de MPEG-4 au profil *baseline* ou *main* de H.264. Les différences majeures entre le profil simple de MPEG-4 et les profils *baseline* ou *main* de H.264 sont recensées dans la table 1.1.

1.7 Mesures de distorsion

Il existe plusieurs critères pour évaluer la distorsion résultant du traitement d'une image. Les plus utilisés sont notamment le PSNR (Peak Signal-to-Noise Ratio), la SAD (Sum of Absolute Differences), la SATD (Sum of Absolute Transform Differences), la MAD (Mean of Absolute Difference), la SSD (Sum of Square Differences) et l'erreur quadratique moyenne,

Tableau 1.1 Différences majeures entre le profil simple de MPEG-4 et les profils *baseline* ou *main* de H.264

Éléments de codage	MPEG-4 <i>simple profile</i>	<i>Baseline</i> ou <i>main</i> H.264 <i>profile</i>
Efficacité de compression	Moyenne	Élevée
Type de trames supportés	I, P, B	I, P, B (<i>main</i>)
Mode de codage	Intra, Skip, Inter16×16 et Inter8×8	Intra16×16, Intra4×4, intra PCM, Skip, Inter16×16, Inter16×8, Inter8×16 et Inter8×8)
Sous partitionnement des blocs8×8	Non	Oui, Inter8×8, Inter8×4, Inter4×8 et Inter4×4
Type de transformée	DCT 8×8	DCT 8×8, DCT 4×4, Hadamard
Filtre anti-blocs	Non	Oui
Codage entropique	Huffman	Huffman , arithmétique (<i>main</i>)
Précision des VMs au pixel	1/2	1/4
Nombre de trames de référence	1	Jusqu'à 16
Mesure de distorsion pour l'EM	SAD	SAD, SATD

MSE (Mean Squared Error). Le PSNR est généralement utilisé comme mesure de distorsion entre deux images ou trames d'une vidéo. La MAD, la SAD et la SATD sont utilisées généralement pour évaluer la distorsion lors d'une recherche de VMs. La SSD, quant à elle, est généralement utilisée pour évaluer la distorsion entre un macrobloc et le macrobloc reconstitué, lors des processus de RDO (rate distortion optimization) pour le choix du MC. D'autres critères tels que la moyenne quadratique sont aussi utilisés. Soit I et J deux partitions de dimensions $M \times N$ dont on veut mesurer la distorsion. La SSD, la MSE, la SAD, la MAD et la

SATD sont évaluées respectivement selon les formules 1.3, 1.5, 1.8, 1.6 et 1.10.

$$SSD(I, J) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} (I(k, l) - J(k, l))^2 \quad (1.3)$$

$$= \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} (D(k, l))^2 \quad (1.4)$$

$$MSE(I, J) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} (I(k, l) - J(k, l))^2 \quad (1.5)$$

$$= \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} (D(k, l))^2$$

$$= \frac{SSD(I, J)}{MN}$$

$$SAD(I, J) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} |I(k, l) - J(k, l)| \quad (1.6)$$

$$= \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} |D(k, l)| \quad (1.7)$$

Avec $I(k, l)$, $J(k, l)$ respectivement les valeurs de pixels à la position (k, l) des images I et J et $D = I - J$.

$$MAD(I, J) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} |I(k, l) - J(k, l)| \quad (1.8)$$

$$= \frac{1}{MN} \cdot SAD \quad (1.9)$$

$$SATD = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} |T(k, l)| \quad (1.10)$$

Avec $T(k, l)$, le (k, l) élément de $T_{M \times N}$, la transformée de Hadamard de $D_{M \times N}$ dont les éléments sont $D(k, l)$. $T_{M \times N}$ se calcule selon la formule 1.11.

$$T_{M \times N} = H_{M \times N} \cdot D_{M \times N} \cdot H_{M \times N}^T \quad (1.11)$$

Avec $H_{M \times N}$, la matrice $M \times N$ de Hadamard et $H_{M \times N}^T$, sa transposée. Pour tout entier $n > 1$, la matrice Hadamard de dimension $2n \times 2n$ suit la loi de récurrence suivante :

$$H_{2n \times 2n} = H_{2 \times 2} \otimes H_{n \times n} \quad (1.12)$$

avec \otimes étant le produit de Kronecker et $H_{2 \times 2}$ tel que décrit dans l'équation 1.13. On peut trouver de plus amples informations sur ce sujet dans la référence Horn et Johnson (1991).

$$H_{2 \times 2} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (1.13)$$

Le PSNR est la mesure de distorsion la plus utilisée pour mesurer la distorsion entre deux images. Il est, en fait, le ratio entre la puissance maximale d'un signal et celle de la distorsion (bruit). Il est fonction de l'erreur quadratique moyenne, MSE et s'exprime généralement en décibels parce que sa fourchette de valeurs est souvent grande. Lorsque les pixels sont représentés sur 8 bits, la formule du PSNR est donnée par :

$$PSNR = 10 \cdot \log_{10} \left(\frac{255^2}{MSE} \right) = 10 \cdot \log_{10} \left(MN \cdot \frac{255^2}{SSD} \right) \quad (1.14)$$

Les valeurs typiques du PSNR sont comprises entre 20 et 50 dB.

La complexité du PSNR du point de vue de la complexité des calculs est très faible. Cette simplicité de même qu'une représentation mathématique simple et efficace contribuent sans doute à en faire la mesure de qualité la plus utilisée pour des tests de performance. C'est la mesure que nous adopterons par la suite. Toutefois, cette dernière présente des carences : c'est une unité de mesure objective, mais qui ne traduit pas forcément la qualité telle que perçue par un humain. D'autres méthodes qui se rapprochent davantage de la perception humaine ont donc été développées. Les plus efficaces de la littérature sont notamment, la méthode décrite dans Rezazadeh et Coulombe (2009), la méthode SSIM (Structural Similarity) dans Wang et al. (2004) et le VIF (Visual Information Fidelity) dans Sheikh et Bovik (2006). Ces méthodes sont toutefois plus lourdes à calculer.

1.8 Méthode d'optimisation débit-distorsion

Le codage vidéo est soumis à de nombreuses contraintes notamment en terme de débit binaire. En effet, les canaux de transmission sur lesquels sont déployées les applications multimédias sont soumis à des limites en terme de débit binaire. Par ailleurs, le codage vidéo se doit également de répondre à certaines attentes en terme de qualité. Malheureusement, ces deux facteurs que sont la qualité et le débit binaire sont divergents, l'amélioration de la qualité se fait au détriment du débit binaire. D'où la nécessité d'optimiser le rapport débit/distorsion (DD), connu sous le nom de *rate distortion (RD)*. Le problème de l'optimisation revient à choisir, pour un MB donné, sa représentation la plus efficace (le VM, le pas de quantification (QP) et la segmentation du macrobloc (MC)) ou mode de codage du point de vue du DD. C'est-à-dire de choisir la meilleure représentation du MB (QP , VM, MC) sous une contrainte de débit.

Il faut cependant souligner que cette optimisation devrait se faire conjointement sur tous les MBs de la trame. Plusieurs processus sont impliqués : le contrôle de débit binaire, le choix du QP , le processus de détermination des VMs. Le VM d'un MB dépend fortement de ceux de ses MB voisins. En effet ce n'est pas le VM lui-même qui est transmis, mais la différence entre le VM et le vecteur prédit (v_p) calculé à partir de ceux des MBs voisins. Le contrôle de débit binaire s'effectue de façon plus efficace au niveau de la trame que du MB. L'allocation binaire est d'abord établie pour toute la trame et est ensuite répartie efficacement sur chaque MB selon sa complexité de codage relative à toute la trame. Une fois l'allocation binaire effectuée, le QP du MB peut ensuite être déterminé. La distorsion du MB dépend fortement du QP , qui lui-même est un paramètre qui dépend de toute la trame et non uniquement des caractéristiques du MB. Toutes ces dépendances entre les éléments de codage font que l'optimisation des éléments du codage d'un MB doit se faire conjointement sur toute la trame, voir même sur toute la séquence.

Le problème de l'optimisation de la représentation du MB peut se traduire sous la forme de l'équation du lagrangien J , qui représente le coût. Il est détaillé dans l'équation 1.15 suivante :

$$J = D + \lambda R \quad (1.15)$$

où J , D , λ et R représentent respectivement le coût du lagrangien, la distorsion entre le signal original s et reconstitué c , le multiplicateur du lagrangien et le nombre de bits pour transmettre les informations du MB. Des détails sur la technique d'optimisation par le lagrangien peuvent être trouvés dans la référence Lim et al. (2005). Il s'agit de déterminer les paramètres du MB que sont le QP , le VM et le MC qui optimisent l'équation 1.15. Réécrivons cependant l'équation 1.15 de façon plus complète afin de mettre en évidence ces paramètres de codage dont le coût lagrangien est fonction. L'équation 1.15 peut se réécrire sous la forme suivante :

$$J(s, c, m, \mathbf{v}_m, QP, \lambda_m) = D(s, c(m, \mathbf{v}_m, QP)) + \lambda_m R(s, c(m, \mathbf{v}_m, QP)), \quad (1.16)$$

$$m \in \{\text{modes disponibles}\}$$

où s et c représentent respectivement les signaux originaux et reconstitué et m , le MC du MB. La liste des MCs possibles sont déterminés par le standard de codage. \mathbf{v}_m représente l'ensemble des VMs du MC, avec $\mathbf{v}_m \in [-p, p]^2$, la région de recherche des VMs. QP quant à lui représente le pas de quantification du MB et λ_m , le multiplicateur du lagrangien associé au MC m . Optimiser une représentation du MB (VMs, MC, QP) revient donc à minimiser le coût $J(s, c, m, QP, \mathbf{v}_m, \lambda)$ pour une valeur particulière du multiplicateur du lagrangien, λ_m sous la contrainte du débit. Résoudre cette équation n'est toutefois pas aisé. En effet, il existe plusieurs interactions entre le MC, les VMs et le QP qui rendent cette tâche extrêmement compliquée. Ces interactions impliqueraient des modes de décision bouclés qui peuvent être bien complexes à résoudre et irréalisables en pratique, spécialement pour des applications temps réel. En pratique, certaines interactions doivent donc être ignorées lors des processus RDO afin d'éviter des modes de décisions bouclés. En pratique, l'équation 1.16 se résout en commençant tout d'abord par émettre une liste d'hypothèse et ensuite en procédant par étapes. La liste des hypothèses émises est la suivante :

- L'optimisation des MBs est indépendante. On optimise MB par MB.
- On suppose que le QP est fixe en ignorant qu'il doit être déterminé par le débit binaire. On utilise le QP de la trame passée pour le RDO du MB.

Ensuite la démarche de résolution de l'équation du lagragien est, dans l'ordre, la suivante :

- a. Pour chaque mode de codage m permis par le standard :
 - (a) On détermine les VMs de ses partitions s'il y a lieu, en minimisant $J_{motion}(\mathbf{v}_m)$, le coût lié au vecteur de mouvement. \mathbf{v}_m représente l'ensemble des VMs pour le mode de codage m .
 - (b) On évalue le coût du lagragien du mode de codage, $J_{MC}(m)$.
- b. Pour finir, on choisit le meilleur mode de codage, celui dont le coût du lagragien, $J_{MC}(m)$ est minimum.

Présentons de façon détaillée le processus de détermination des VMs et de l'évaluation du coût du lagragien du mode de codage, $J_{MC}(m)$.

1.8.1 Méthode de détermination des vecteurs de mouvement

Il faut donc trouver les vecteurs qui traduisent le mieux le mouvement des partitions du MC, ceux qui présentent donc la distorsion minimale entre le macrobloc original et reconstitué. Malheureusement, les limitations au niveau du débit binaire nous contraignent à choisir le VM de la partition optimale en terme de distorsion, mais également de bits à transmettre. En effet, il peut exister plusieurs VMs qui donnent une faible distorsion (spécialement pour les MB appartenant à des zones stationnaires), mais un seul minimise le nombre de bits pour transmettre ce dernier. Par contre, pour des zones plus complexes, un seul VM donnera généralement une faible distorsion. Par contre ce dernier pourrait demander plus de bits à transmettre, car, comme expliqué à la section 1.8, on pénalise les VMs qui s'éloignent du vecteur prédit. On pourrait donc choisir un VM qui donne une distorsion moins intéressante, mais qui est optimale du point de vue du débit binaire. L'équation générale du lagragien, $J = D + \lambda R$, traduit le problème de l'optimisation, il faut donc la résoudre pour déterminer les VMs.

Le coût lié au VM sera noté $J_{motion}(\mathbf{v}_m)$. Pour les VMs, le critère de distorsion D utilisé habituellement est la SAD (*sum of absolute differences*) pour H.263, MPEG-4, H.264 ou la SATD (*sum of absolute transformed differences*) pour H.264. Le multiplicateur lié au VM sera noté λ_{motion} . Le nombre de bits R nécessaire pour transmettre les VMs sera noté $R(\mathbf{v}_m - \mathbf{v}_p)$, pour mettre en évidence le fait que ce n'est pas le VM lui-même qui est transmis mais la différence avec le vecteur prédit, \mathbf{v}_p . L'équation à résoudre peut donc s'écrire sous la forme suivante :

$$J_{motion}(\mathbf{v}_m) = SA(T)D(s, c(\mathbf{v}_m)) + \lambda_{motion}R(\mathbf{v}_m - \mathbf{v}_p) \quad (1.17)$$

Le multiplicateur du lagrangien λ_{motion} permet de contrôler l'impact du débit binaire. \mathbf{v}_p , étant le vecteur prédit. Pour des modes de codage, à plusieurs partitions, il faudra résoudre cette équation pour chaque partition k du MC m pour trouver le VM qui lui est associé, $\mathbf{v}_{m,k}$. L'équation 1.18 peut se réécrire sous la forme suivante pour chaque partition :

$$J_{motion}(\mathbf{v}_{m,k}) = SA(T)D(s, c(\mathbf{v}_{m,k})) + \lambda_{motion}R(\mathbf{v}_{m,k} - \mathbf{v}_{p,k}) \quad (1.18)$$

$SAD(s, c(\mathbf{v}_{m,k}))$ ou $SA(T)D(s, c(\mathbf{v}_{m,k}))$ sont calculés suivant les formules suivantes :

$$SAD(s, c(\mathbf{v}_{m,k})) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |s(x, y) - c(x, y, \mathbf{v}_{m,k})| \quad (1.19)$$

$$= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |d(x, y)| \quad (1.20)$$

$$SA(T)D(s, c(\mathbf{v}_{m,k})) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |t(x, y)| \quad (1.21)$$

M, N étant les dimensions de la partition k sur laquelle l'estimation est effectuée. $d(x, y)$ est la différence entre le signal original et le signal reconstitué, donné par $d(x, y) = s(x, y) - c(x, y, \mathbf{v}_{m,k})$. Il est également le (x, y) ième élément de $D_{M \times N}$ détaillé dans la formule 1.22. $|t(x, y)|$ quant à lui, est le (x, y) élément de $T_{M \times N}$, la transformée de Hadamard de $D_{M \times N}$ dé-

taillée dans la formule 1.23.

$$D_{M \times N} = S_{M \times N} - C_{M \times N}(V_{m,k}) \quad (1.22)$$

$$T_{M \times N} = H_{M \times N} \cdot D_{M \times N} \cdot H_{M \times N}^T \quad (1.23)$$

$H_{M \times N}$ étant la matrice $M \times N$ de Hadamard et $H_{M \times N}^T$ sa transposée (Horn et Johnson (1991)). $\mathbf{v}_{m,k}$ et $\mathbf{v}_{p,k}$ sont respectivement le VM associé à la partition k et le vecteur prédit et sont donnés par l'équation 1.24. Le terme $R(\mathbf{v}_{m,k} - \mathbf{v}_{p,k})$ est calculé par une table *look up*.

$$\begin{aligned} \mathbf{v}_{m,k} &= (v_{x_{m,k}}, v_{y_{m,k}})^T \\ \mathbf{v}_{p,k} &= (v_{x_{p,k}}, v_{y_{p,k}})^T \end{aligned} \quad (1.24)$$

Le RDO utilise comme critère de distorsion la SSD (somme des différences au carré) au lieu du SAD ou de la SATD. Pour compenser le fait que le RDO utilise ici le SAD ou la SATD comparativement au SSD pour le mode de codage, λ_{motion} est ajusté pour le manque d'opération au carré (Lim et al. (2005)) par rapport au lagrangien utilisé pour le RDO du MC, $\lambda_m \cdot \lambda_{motion}$ est donc évalué comme suit :

$$\lambda_{motion} = \sqrt{\lambda_m} \quad (1.25)$$

1.8.2 Méthode d'optimisation du mode de codage

Tout comme le VM, le choix du MC se résume à un problème d'optimisation, car le MC choisi doit être celui dont le coût en terme de DD est minimal. Ce qui revient donc à résoudre l'équation générale, $J = D + \lambda R$, du lagrangien qui traduit cette optimisation. Le coût lié au MC sera noté J_{MC} et peut s'exprimer sous la forme de l'équation 1.26 :

$$J_{MC} = D_{MC} + \lambda_m R, \quad m \in \{\text{modes disponibles}\} \quad (1.26)$$

Cette équation peut se réécrire de façon plus complète sous la forme de l'équation 1.27.

$$J_{MC}(s, c, m, \mathbf{v}_m, QP, \lambda_m) = D_{MC}(s, c(m, \mathbf{v}_m, QP)) + \lambda_m R(s, c(m, \mathbf{v}_m, QP)), \quad (1.27)$$

$$m \in \{\text{modes disponibles}\}$$

Avec $R(s, c(m, \mathbf{v}_m, QP))$ étant le nombre de bits requis pour coder le MB dans le mode m , $D_{MC}(s, c(m, \mathbf{v}_m, QP))$ la distorsion entre le signal original s et le signal reconstitué c . Le critère de distorsion généralement utilisé pour calculer la distorsion dans le cas de l'optimisation du MC est la SSD entre le signal original s et reconstitué c , et sera notée $SSD(s, c)$. La distorsion D_{MC} s'écrit alors sous la forme de l'équation 1.28.

$$D_{MC}(s, c(m, \mathbf{v}_m, QP)) = SSD(s, c(m, \mathbf{v}_m, QP)) \quad (1.28)$$

La distorsion $SSD(s, c(m, \mathbf{v}_m, QP))$ se calcule selon la formule 1.29 suivante qui met en évidence le calcul sur l'ensemble des pixels de coordonnées (x, y) du MB des signaux s et c :

$$SSD(s, c(m, \mathbf{v}_m, QP)) = \sum_{(x, y) \in MB} |s(x, y) - c(x, y, m, \mathbf{v}_m, QP)|^2$$

Le coût lié au MC, $J_{MC}(s, c, m, \mathbf{v}_m, QP, \lambda_m)$ peut alors s'écrire de façon plus complète sous la forme de l'équation 1.29 et 1.30 en détaillant le calcul sur les partitions k du MC.

$$J_{MC}(s, c, m, \mathbf{v}_m, QP, \lambda_m) = \sum_{(x, y) \in MB} |s(x, y) - c(x, y, m, \mathbf{v}_m, QP)|^2 + \lambda_m R(s(x, y), c(x, y, m, \mathbf{v}_m, QP)) \quad (1.29)$$

$$J_{MC}(s, c, m, \mathbf{v}_m, QP, \lambda_m) = \sum_{k=1}^{nb_m} \sum_{(x, y) \in MB} |s(x + p_{kx}, y + p_{ky}) - c(x + p_{kx}, y + p_{ky}, m, V_{m, k}, QP)|^2 + \lambda_m R(s(x + p_{kx}, y + p_{ky}), c(x + p_{kx}, y + p_{ky}, m, V_{m, k}, QP)) \quad (1.30)$$

nb_m étant le nombre de partitions du MC m et (p_{kx}, p_{ky}) , les coordonnées de la partition k . Le MC H.264 choisi étant celui qui optimise le RDO, il est donc l'argument minimum du coût

$J_{MC}(s, c(m, \mathbf{v}_m, QP), \lambda_m)$) tel que le montre l'équation 1.31.

$$\begin{aligned} MC &= \arg \min_m J_{MC}(s, c(m, \mathbf{v}_m, QP), \lambda_m), \quad m \in \{\text{modes disponibles}\} \\ &= \arg \min_m \left[\left(\sum_{(x,y) \in MB} |s(x,y) - c(x,y, m, \mathbf{v}_m, QP)|^2 \right) + \lambda_m R(s, c(m, \mathbf{v}_m, QP)) \right] \end{aligned} \quad (1.31)$$

Bien que des études ont montré que le multiplicateur du lagrangien λ_m peut être déterminé théoriquement, des expérimentations ont montré qu'il dépend fortement du pas de quantification QP . Des simulations effectuées par Lim et al. (2005), Wiegand et al. (2003) ont montré que les meilleurs résultats sont obtenus pour l'équation 1.32 pour du codage H.263 et MPEG-4. Les équations 1.33 et 1.34 (Tang et Nasiopoulos (2010)) quand à elles sont utilisées en H.264 pour les trames P et B respectivement. λ est optimal pour $\lambda_m = c \cdot QP^2$, des simulations ont également montré qu'on obtient les meilleurs résultats pour $c = 0.85$ (Lim et al. (2005)).

$$\lambda_m = 0.85 \cdot QP^2 \quad (1.32)$$

$$\lambda_{m,P} = 0.85 \cdot 2^{\frac{QP-12}{3}} \quad (1.33)$$

$$\lambda_{m,B} = \max \left(2, \min \left(4, \frac{QP-12}{6} \right) \right) \quad (1.34)$$

Une fois le MC déterminé, on peut alors rajuster le QP en fonction du débit binaire avec les algorithmes de contrôle de débit binaire tel que TM5 ISO/IEC (1993a) pour MPEG-2, TMN8 UIT/SG16 (1997) pour H.263, VM8 Video Group (1997) pour MPEG-4 ou celui de Ma et al. (2002) pour H.264. C'est cette valeur ajustée qui sera utilisée dans le RDO du MB de la trame suivante.

Pour un MC m donné du standard, il faut tout d'abord trouver les VMs (processus assez lourd à calculer) pour chaque partition de ce MC en minimisant, pour chacune des partitions, le coût J_{motion} . Ensuite, il faut évaluer le coût lié à ce MC m , $J_{MC}(m)$. Pour cela, il faudra calculer la distorsion $SSD(s, c)$ et le nombre de bits requis, R . Calculer ces deux paramètres revient à simuler le codage du MC (soit les opérations de transformée, de codage entropique, de quanti-

fication, de quantification inverse, etc.), pour déterminer le signal reconstruit c et calculer ainsi $SSD(s, c)$ et pour aussi trouver le nombre de bits requis R . Ce processus doit être répété pour tous les MCs permis par le standard, afin de trouver les différents $J_{MC}(m)$ et choisir finalement le MC du MB qui présente le coût J_{MC} minimal. Le RDO, comme on peut le constater, peut être extrêmement lourd à évaluer, surtout quand on sait que certains standards comme H.264 permettent jusqu'à 8 modes et 4 sous modes. Ce lourd processus pose alors un problème sérieux pour les applications temps réels.

CHAPITRE 2

ÉTAT DE L'ART

2.1 Introduction

La façon la plus simple d'effectuer le transcodage de format de MPEG-4 à H.264 est de complètement décoder le train binaire MPEG-4 à l'aide d'un décodeur MPEG-4 et de le réencoder à l'aide d'un encodeur H.264. C'est-à-dire de ramener les trames dans le domaine des pixels et de les réencoder entièrement selon les spécifications H.264 (Ahmad et al. (2005)). On parle alors d'un transcodage en cascade. Le schéma d'un tel transcodeur est présenté à la figure 2.1. Les meilleures performances en qualité sont normalement atteintes avec ce type de transcodage. Cette méthode nous servira notamment de référence en qualité et en vitesse dans nos tests de simulations. Elle sera référée en tant que méthode **Cascade**.

Malheureusement, cette approche est extrêmement complexe en calculs, car elle nécessite un encodage H.264 complet des trames vidéo. Elle est ainsi peu adaptée aux applications en temps réel. L'encodage H.264 requiert une grande puissance en calculs à cause de ses outils de codage. En effet, H.264 utilise plusieurs MCs, 10 en tout (7 modes inter, 4 sous modes, un mode Skip et deux modes intra). Pour déterminer le meilleur MC, H.264 évalue le RDO pour chaque MC H.264 et choisit celui dont la distorsion est minimale. Ce processus est complexe, car il implique que le codeur doit effectuer des opérations de transformation fréquentielle (DCT, Hadamard) et de codage entropique (VLC, CABAC) pour simuler le nombre de bits nécessaires à la transmission des MBs de la trame. Il faut préciser qu'avant de pouvoir effectuer ces opérations, il faut trouver les VMs pour les modes et sous modes inter H.264. L'encodeur H.264 peut ainsi devoir réaliser jusqu'à 41 opérations d'estimation et de compensation de mouvement pour un seul MB (de 1 VM pour Inter16×16 à 16 VMs pour Inter4×4). Il faut aussi préciser que la précision des VMs H.264 au quart de pixel alourdit ce processus déjà complexe. Des recherches ont ainsi été effectuées afin de réduire la complexité en calculs de

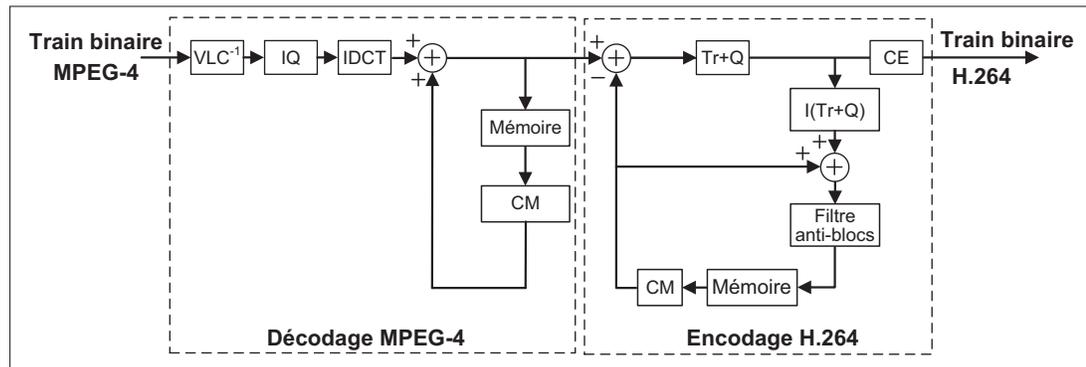


Figure 2.1 Schéma bloc d'un transcodeur en cascade MPEG-4 à H.264.

ce type de transcodage. Certains auteurs ont ainsi développé des algorithmes de transcodage MPEG-4 à H.264 efficaces qui réduisent le temps de traitement tout en maintenant une qualité acceptable. Pour ce faire, ces algorithmes recueillent des informations du train binaire MPEG-4 durant l'étape de décodage (notamment les MCs, les VMs, les coefficients DCT, etc.). Ils s'en servent ensuite pour exclure ou simplifier certaines étapes du processus de codage H.264. Le principe de transcodage efficace de ce type d'algorithme est illustré à la figure 2.2.

Nous avons recensé, dans la littérature, les algorithmes qui traitent du transcodage MPEG-4 à H.264. La plupart effectuent le transcodage avec des opérations supplémentaires telles que la réduction de résolution, de débit de trames, de réduction de débit binaire, etc. Certains auteurs se sont intéressés à d'autres types de transcodage tels que le transcodage de MPEG-2 à H.264, de H.263 à H.264 ou de H.264 à MPEG-4. Peu d'auteurs traitent du transcodage de format MPEG-4 à H.264 seulement (avec possible réduction de débit). Nous en avons recensé quelques-uns : Shen (2004), Liang et al. (2007), Lee et Lee (2005), Lee et al. (2006), Kim et al. (2005). Ces algorithmes peuvent se classer en deux catégories. Ceux qui travaillent dans le domaine fréquentiel et ceux qui travaillent dans le domaine spatial. Nous décrivons dans ce chapitre, de façon détaillée, les avantages et les inconvénients de chacune de ces méthodes. Ensuite, nous présentons brièvement l'état de l'art d'autres types de transcodage de formats. Bien que nous nous intéressions uniquement au transcodage MPEG-4 à H.264, les idées développées dans ces méthodes pourraient s'appliquer au transcodage qui nous intéresse. Finalement, la conclusion fait un résumé de l'état de l'art étudié dans ce chapitre.

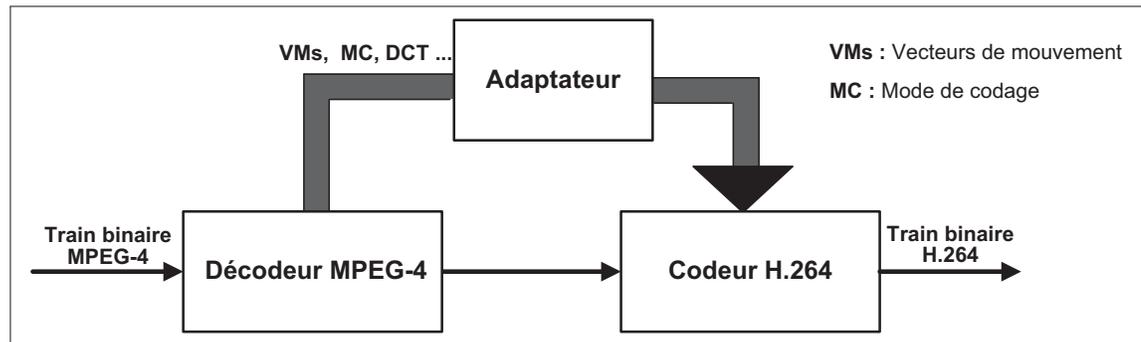


Figure 2.2 Schéma de principe d'un transcodage efficace.

2.2 Approche dans le domaine transformé proposée par Shen

La méthode développée par Shen (2004) effectue le transcodage dans le domaine transformé (aussi appelé domaine fréquentiel). L'auteur a mené certaines recherches dans le but d'effectuer le transcodage MPEG-4 à H.264 sans revenir au domaine de pixels, notamment en développant des matrices de transformation. Ces matrices permettent de transcoder une DCT 8×8 en une DCT 4×4 . L'auteur propose également de réutiliser les vecteurs de mouvement MPEG-4 pour alléger le coût en calculs. Le domaine transformé présente l'avantage de nécessiter moins de puissance de traitement que le transcodage spatial.

Cette méthode présente par contre de nombreux inconvénients. Nous avons d'une part les carences liées à la méthode elle-même et celles liées au fait de travailler dans un domaine transformé.

Un premier problème se pose pour le mode intra. MPEG-4 utilise une DCT 8×8 tandis que H.264 utilise trois types de transformées selon le type de données à coder. Lorsque le macrobloc est codé en mode Intra 16×16 , H.264 utilise une transformée de Hadamard pour les 4×4 coefficients DC de luminance, une transformée de Hadamard pour les 2×2 coefficients DC de chrominance et une transformée DCT pour tous les autres blocs 4×4 . D'autre part, les opérations de transformée et de quantification sont combinées, ce qui n'est pas le cas avec MPEG-4. En raison de cette différence de transformées, le transcodage optimal MPEG-4 à

H.264 nécessite une opération inverse complète. C'est-à-dire une quantification inverse, une DCT 8×8 inverse dans le décodeur MPEG-4 et ensuite une transformée et une quantification selon les spécifications H.264. La différence majeure des transformées fait que l'approche de matrices de transcodage de DCT 8×8 à DCT 4×4 devient difficile à réaliser et ne peut être utilisée pour le transcodage de blocs intra. Pour éviter de revenir dans le domaine des pixels, l'auteur propose une approche sous optimale et dérive une manière de convertir les transformées pour ce mode. Dans ce cas, seul le mode de prédiction DC est utilisé. Cette approche introduit toutefois de sérieuses dégradations de la qualité visuelle.

Un autre problème se pose également pour le transcodage de blocs inter qui utilisent une transformée DCT 8×8 en H.264. La DCT 8×8 en H.264 est différente de celle de MPEG-4. Il s'agit en fait d'une approximation entière. L'auteur propose alors, pour la transformation DCT 8×8 à DCT 8×8 , une approche similaire aux matrices de transformation DCT 8×8 à DCT 4×4 . L'auteur précise toutefois que des tests quantitatifs devront être menés pour justifier l'approximation des matrices et n'indique pas non plus précisément comment le faire.

Un autre inconvénient majeur est que les approches dans le domaine transformé (ou fréquentiel) sont beaucoup plus complexes à implémenter, sont moins flexibles et réutilisables. Une des raisons est que les boucles du décodeur et du codeur ne sont pas indépendantes (Ahmad et al. (2005)). Une autre raison est que le filtrage anti-blocs en H.264 doit se faire dans le domaine des pixels (Bialkowski et al. (2006)). Il n'existe actuellement aucune façon efficace de réaliser cette opération dans le domaine transformé (Bialkowski et al. (2006)).

L'auteur propose de réutiliser les vecteurs de mouvement de MPEG-4 mais ne donne aucune indication sur la façon de procéder. Il ne fait pas mention non plus d'un processus de raffinement. Comme nous le verrons dans les résultats de simulation aux chapitres 3 et 5, les VMs doivent être raffinés pour atteindre une bonne qualité. L'estimation de mouvement est toutefois assez complexe à réaliser dans le domaine fréquentiel. La méthode essaie tous les MCs permis par le standard H.264. Elle pourrait pu être plus efficace si les MCs MPEG-4 sont utilisés. Ces derniers auraient pu permettre de réduire significativement le nombre de MCs à

tester et accroître les gains en vitesse. L'approche fréquentielle manque toutefois de flexibilité quant à l'utilisation des MCs MPEG-4 (Bialkowski et al. (2006)).

Une autre carence majeure du domaine transformé est qu'il souffre de sévères accumulations d'erreurs. Cela est dû au fait que les boucles du décodeur et du codeur ne pas sont indépendantes. Les prédicteurs peuvent ainsi être différents de ceux du codeur original, ce qui conduit à une accumulation sévère d'erreurs. Aucune boucle de retour n'est faite pour compenser ces accumulations (Ahmad et al. (2005)).

Les résultats de simulation présentés par les auteurs sont peu détaillés. Aucun détail n'est donné sur le nombre et le type des séquences vidéos ainsi que sur les conditions de codage telles que le débit binaire. Les résultats ont été présentés pour des QPs fixes et les tests ont été effectués sur des résolutions CIF uniquement. Dans ce contexte, il est difficile de juger de l'efficacité de la méthode présentée. Il est également difficile de juger de la capacité d'adaptation de l'algorithme aux conditions de transcodage telles que le débit binaire, la résolution et les caractéristiques vidéos. Il aurait été intéressant de voir le comportement de l'algorithme dans un contexte de contrôle de débit binaire plutôt qu'uniquement dans des cas de QPs fixes.

En résumé, l'algorithme présenté en Shen (2004) présente l'avantage de nécessiter peu de puissance de calculs. Cependant, le manque de flexibilité et les accumulations d'erreurs des architectures du domaine transformé rendent cette approche peu efficace.

Le transcodage dans le domaine spatial est plus flexible et présente l'avantage de ne pas avoir d'accumulation d'erreurs (Ahmad et al. (2005)). Le problème du filtre anti-bloc ne se pose pas non plus dans ce cas. Les algorithmes détaillés en Kim et al. (2005), Lee et Lee (2005), Lee et al. (2006) et Liang et al. (2007) se réalisent dans ce domaine. Ces algorithmes sont détaillés dans les sections suivantes.

2.3 Approche basée sur les statistiques des modes de codage proposée par Lee et al.

Lee et Lee (2005) et Lee et al. (2006) exploitent des informations recueillies du décodeur MPEG-4, notamment les VMs et les MCs. Ils les utilisent ensuite pour sauter certaines étapes

du processus de détermination des MCs et de l'estimation de mouvement dans le codeur H.264. Cette méthode a ainsi l'avantage d'exploiter les VMs et les MCs de MPEG-4 pour réduire les deux modules les plus complexes que sont la détermination des modes de codage et l'estimation de mouvement. Cette méthode sera référée en tant méthode **STAT** et **STAT+REF** dans nos simulations. Nous utiliserons STAT pour le cas où les VMs ne sont pas raffinés et STAT+REF pour le cas où on raffine les VMs.

Cet algorithme se base essentiellement sur la probabilité de correspondance des modes de codage MPEG-4 à ceux de H.264. Nous écrivions précédemment que le codeur H.264 déterminait le taux de distorsion de 10 MCs candidats pour déterminer l'optimal. Lee et al. (2006) restreignent ce nombre de candidats en établissant des statistiques relatives au pourcentage de correspondance des modes de codage de MPEG-4 vers ceux de H.264. Plus concrètement, pour chaque MC MPEG-4 d'entrée donné, un ensemble de MCs comprenant uniquement les modes les plus probables est établi. Pour déterminer ces modes candidats, un transcodage en cascade est d'abord effectué pendant lequel les statistiques de correspondance sont rassemblées. Des tables de fréquence d'occurrence des MCs H.264 en fonction de ceux de MPEG-4 sont alors établies. Un exemple d'une telle table, adaptée de Lee et Lee (2005) est montrée dans le tableau 2.1. Seuls les modes surlignés en gris sont utilisés comme candidats et sont évalués par l'algorithme. Par exemple, lorsqu'un macrobloc a été codé en Inter16×16 dans MPEG-4, on n'évaluera que les modes candidats Inter16×16, Inter16×8, Inter8×16 et Inter8×8.

Les VMs MPEG-4 sont également réutilisés dans H.264 et ces derniers sont raffinés pour améliorer la qualité. Les résultats en accélération obtenus sont de l'ordre de 3× à 4,3× pour des pertes en qualité de 0,2 dB à 250 kb/s et de 0,9 dB autour de 130 kb/s pour des vidéos QCIF. Pour CIF, ils obtiennent des pertes de 0,4 à 0,5 dB. Ces résultats ont été comparés à la méthode en cascade.

Les auteurs ont développé une idée innovatrice qui consiste à établir des statistiques de correspondance entre les MCs et à éliminer ensuite les modes les moins probables. Cette façon

Tableau 2.1 Table de correspondance des MCs H.264 et MPEG-4. Adaptée de Lee et al. (2006)

Fréquence d'apparition des MCs H.264 en fonction des MCs MPEG-4 dans un transcodage en cascade pour différentes vidéos QCIF transcodées à débits binaires entre 130 et 250 kb/s

Modes de codage H.264 / Modes Codage MPEG-4	Modes inter							Modes intra		Skip	Total
	16×16	16×8	8×16	P8×8				16×16	4×4		
				8×8	8×4	4×8	4×4				
Inter 16×16	42,2%	12,4%	13,6%	9,6%	1,7%	2,1%	1,2%	1,1%	0,8%	15,5%	100%
Inter 8×8	8%	11,8%	15,8%	28,6%	5,3%	6,7%	9,5%	0,7%	13,5%	0,1%	100%
Skip	5,8%	1,1%	1,1%	0,4%	0,1%	0,1%	0%	0,5%	0%	90,9%	100%
Intra	0%	0%	0%	2,6%	0%	0%	7,9%	21,1%	68,4%	0%	100%

Fréquence d'apparition des MCs H.264 en fonction des MCs MPEG-4 dans un transcodage en cascade pour différentes vidéos CIF transcodées à débits binaires entre 200 et 384 kb/s

Modes de codage H.264 / Modes Codage MPEG-4	Modes inter							Modes intra		Skip	Total
	16×16	16×8	8×16	P8×8				16×16	4×4		
				8×8	8×4	4×8	4×4				
Inter 16×16	43,6%	13,5%	12,8%	14,4%	4,3%	4,4%	2%	0,8%	0,5%	3,7%	100%
Inter 8×8	15,3%	13,5%	12,8%	27,5%	8,6%	8,7%	8,5%	0,4%	4,7%	0,1%	100%
Skip	7,2%	0,9%	0,8%	0,2%	0%	0%	0%	1,5%	0%	89,4%	100%
Intra	5,9%	3,5%	3,5%	8,2%	2,4%	1,2%	14,1%	8,2%	52,9%	0%	100%

de faire permet de garantir l'efficacité de l'ensemble des MCs à tester. Ce qui n'est pas le cas en Liang et al. (2007), où une table arbitraire a été utilisée.

L'algorithme a cependant l'inconvénient de ne pas tenir compte des conditions de transcodage, notamment du débit binaire et des caractéristiques vidéos. Nous verrons en effet dans la suite, notamment au chapitre 5, que les pourcentages d'occurrence des MCs H.264 varient énormément en fonction de ces facteurs. L'ensemble des MCs à considérer devrait pouvoir s'adapter à de telles conditions pour une meilleure efficacité. Ne pas tenir compte des conditions de codage pourrait créer également une inconsistance dans les résultats de simulations. C'est en effet ce qui se passe, comme on pourra le constater plus loin dans la section des résultats du chapitre 3. L'algorithme donne de moins bons résultats pour certains types de vidéos que d'autres. Les résultats sont également moins bons à bas qu'à hauts débits.

Par ailleurs, l'algorithme considère de petites partitions (Inter8×4, 4×8 et 4×4) dans le cas d'un MC MPEG-4 d'entrée Inter8×8. Pourtant, il a été montré que de telles partitions n'ont

un impact sur la qualité qu'à des débits élevés (Joch et Kossentini (2001)). En éliminant ces modes de l'ensemble des modes candidats à tester, la complexité aurait pu être réduite de façon plus notable. On remarque aussi, dans le tableau 2.1, que le nombre de MCs H.264 candidats considérés, notamment pour les MCs d'entrée Inter16×16 et Inter8×8, est élevé. Les auteurs auraient pu également restreindre ce nombre en prenant en considération la complexité du MB dans la classification des MBs. Les MBs complexes, tendent en effet à être codés en petites partitions et ceux peu complexes, en grandes partitions. Pour des MBs peu complexes, les petites partitions seraient éliminées de l'ensemble des candidats et vice versa.

Comme pour la plupart des algorithmes de la littérature, les auteurs valident leurs méthodes en utilisant des codecs de référence (ITU (2003),ISO/IEC (2001)). Ces codecs bien, qu'idéaux pour des tests en qualité, sont loin d'être idéaux en termes de vitesse de traitement. Ils sont très sous-optimaux en complexité des calculs. Ils ne peuvent ainsi pas servir de référence pour des tests de vitesse. Les résultats en vitesse ne seraient pas représentatifs de ceux à obtenir sur des produits commerciaux qui sont nettement optimisés et plus rapides. Par exemple, l'encodeur x264, une implémentation optimisée du standard H.264, est environ 50 fois plus rapide que l'implémentation de référence. Avec ce type d'implémentation, puisque les algorithmes de décision de modes et surtout de recherche de vecteurs de mouvement sont agressivement optimisés, les gains en vitesse obtenus en y appliquant les approches proposées ici seraient nettement réduits en comparaison de ceux obtenus avec le code de référence dans lequel ces opérations contribuent à la majeure partie des calculs (Merritt et Vanam (2007)).

Les résultats de simulations présentés par les auteurs de cette méthode sont par ailleurs peu exhaustifs. Seuls les résultats de deux séquences sont présentés (Foreman et Tempête). Par ailleurs, seulement des débits binaires élevés sont utilisés pour les simulations. À bas et moyens débits, les algorithmes de transcodage donnent généralement des résultats en qualité beaucoup plus faibles. Il aurait été intéressant d'avoir des simulations sur des séquences plus nombreuses et variées et à des débits plus faibles afin de juger de l'efficacité de la méthode. Nous pourrions constater dans les résultats de simulations, présentés au chapitre 5, qu'effectivement leurs méthodes se comportent nettement moins bien à bas qu'à haut débit. Par exemple

pour QCIF, on obtient à 192 kb/s, -0,18 dB tandis qu'à bas débits on obtient une perte de -1,19 dB par rapport à la méthode en cascade.

2.4 Algorithme de Liang et al.

Liang et al. (2007) tout comme Lee et Lee (2005) et Lee et al. (2006), utilisent les VMs et les MCs issus du décodage MPEG-4 pour réduire la complexité de certaines étapes du codage H.264 et réduire ainsi le temps de traitement. L'algorithme extrait la direction de prédiction intra des MBs en MPEG-4 et le nombre de coefficients AC non nuls de chaque bloc 8×8 . Ces informations sont ensuite combinées et exploitées afin de réduire le nombre de prédictions intra à considérer en H.264. Le nombre de zéros sert de mesure de complexité du MB afin de limiter ou augmenter le nombre de directions de prédiction intra à considérer.

Pour le transcodage inter, les auteurs se servent du MC MPEG-4 pour réduire et déterminer l'ensemble des MCs H.264 à tester. Ils établissent pour cela une table de correspondance entre les MCs MPEG-4 et H.264. Ils se servent également des VMs MPEG-4 comme point de départ du processus d'estimation de mouvement des VMs H.264. L'ensemble des MCs à tester et la détermination des VMs qui y sont associés se font comme suit :

- **Cas d'un MB codé Skip en MPEG-4.** Seul le mode Inter 16×16 est testé en H.264.
- **Cas d'un MB codé Inter 16×16 .** Les MCs considérés sont l'Inter 16×16 , l'Inter 16×8 , l'Inter 8×16 et l'Inter 8×8 . Les VMs sont déterminés de la façon suivante : le VM Inter 16×16 est le même que celui de MPEG-4. Pour les MCs Inter 16×8 et Inter 8×16 , un raffinement est effectué avec le VM MPEG-4 à la précision entière comme point de départ. Les positions au demi-pixel ne sont pas évaluées ; seules celles au quart de pixel le sont. Pour le MC Inter 8×8 , un raffinement est effectué en évaluant les positions au demi et au quart de pixel.
- **Cas d'un MB codé Inter 8×8 .** L'algorithme ne considère pas, dans ce cas, des partitions plus grandes que 8×8 . L'algorithme considère des partitions de taille allant de 4×4 à 8×8 . Les VMs 8×8 de MPEG-4 sont réutilisés tels quels. Pour les VMs Inter 8×4 et Inter 4×8 , un

raffinement est effectué avec comme point de départ, la position entière des VMs MPEG-4. Les positions au demi-pixel ne sont pas testées. Pour les VMs 4×4, les positions au demi et au quart de pixels sont évaluées.

L'idée de restreindre le nombre de directions de prédiction intra à tester, en se servant des informations MPEG-4, est intéressante. Elle pourra permettre de réduire davantage la complexité du transcodage intra MPEG-4 à H.264. L'impact de la réduction de la complexité du transcodage des trames intra sur la complexité totale de la séquence vidéo est cependant limité puisque les trames intra sont beaucoup moins nombreuses que les trames inter. Les efforts doivent donc être investis au niveau du transcodage des trames inter.

L'algorithme de transcodage inter proposé par Liang et al. (2007) présente cependant de nombreuses carences. En regardant les fréquences d'occurrence des MCs H.264 en fonction du MC MPEG-4 dans le tableau 2.1, on constate que l'ensemble des MCs H.264 considéré est peu efficace.

Par exemple, l'algorithme ne considère pas le MC Skip pour un MC MPEG-4 d'entrée Skip. Les statistiques du tableau 2.1 nous montrent pourtant que ce mode a une fréquence d'occurrence aussi élevée que 90%. Le même constat est fait pour un MC d'entrée Inter16×16. L'algorithme ne considère pas le MC Skip. Cela pourrait poser problème spécialement à bas débits. En regardant les tables de statistiques, on peut constater que le mode Skip apparaît avec un taux de 15,5% pour des débits de 130 à 250 kb/s. Dans les conditions réelles de communication, le débit considéré est souvent nettement plus bas. Par conséquent, le mode Skip apparaîtra à un taux plus élevé que 15,5%. Dans la suite de la thèse, on pourra constater qu'il apparaît à un taux aussi élevé que 42,2% à 32 kb/s. Pour obtenir de bons résultats en qualité, il est impératif de considérer le mode Skip pour un MC MPEG-4 d'entrée Inter16×16 ou Skip, et particulièrement à bas débits. L'ensemble des MCs considérés dans le cas d'un MC d'entrée Inter8×8 semble également complètement inapproprié. Liang ignore les grandes partitions, Inter16×16, Inter16×8 et Inter8×16. On constate, dans la table, que ces modes apparaissent pourtant à des taux de 10 à 15%. Le mauvais choix des MCs H.264 candidats expliquerait en

partie les mauvais résultats en qualité obtenus pour cet algorithme. Nous obtenons des pertes allant jusqu'à 4 dB dans nos résultats de simulations.

On peut aussi critiquer la décision des auteurs de ne pas avoir tenu compte des conditions de codage et principalement du débit binaire. Ce facteur joue en effet un rôle déterminant dans le choix du MC. Lorsque les contraintes binaires sont serrées, de grandes partitions sont privilégiées et deviennent le MC optimal. La non-consideration des conditions de codage et spécialement de la contrainte binaire expliquerait la faiblesse des résultats en qualité à bas débits. Les pertes en qualité sont souvent nettement plus basses à bas qu'à hauts débits. On obtient, par exemple, une perte de -3,19 dB à 128 kb/s tandis qu'à 768 kb/s, la perte est de -0,54 dB en moyenne pour des vidéos CIF (voir tableau 5.7).

Le processus de détermination des VMs et notamment de raffinement des VMs semble également peu efficace. Dans le cas d'une correspondance de MCs similaires (ex. Inter16×16 à Inter16×16), les VMs ne sont pas raffinés. Les conditions de codage telles que les caractéristiques vidéos et le débit binaire contribuent à ce que les VMs MPEG-4 ne soient pas optimaux. Raffiner ces derniers permettrait d'améliorer énormément la qualité vidéo ; jusqu'à 2 dB selon nos simulations. Par ailleurs, le processus de raffinement souffre d'une lacune importante. Dans certains cas, les auteurs omettent de tester certaines positions de pixels. L'inefficacité du processus de détermination des VMs expliquerait également les mauvais résultats en qualité obtenus pour cette méthode.

Les auteurs obtiennent, lors de simulations, des gains en vitesse assez impressionnants : de l'ordre de 8,5× à 10,4× pour des pertes en dB allant de 1 à 2 dB. Comme la plupart des algorithmes de la littérature, l'algorithme présente l'inconvénient d'avoir été simulé avec des codecs de référence. Les résultats en vitesse sont ainsi peu fiables. On pourra d'ailleurs le constater dans les chapitres suivants, notamment à la section des résultats. Cet algorithme nous donne des gains plus modestes, de l'ordre de 3× à 4,5×, quand il est implémenté sur des codecs optimisés en vitesse comme ceux d'Intel.

Cette méthode sera référée dans nos résultats de simulations en tant que méthode **MV+MS** ; MV pour *motion vector* et MS pour *mode selection*.

2.5 Algorithme de Kim et al.

L'algorithme de Kim et al. (2005), lui aussi, réduit la complexité du transcodage MPEG-4 en réutilisant les MCs MPEG-4 pour réduire le nombre de MCs H.264 à tester. Ils réutilisent aussi les VMs MPEG-4 dans le processus d'estimation de mouvement H.264. Les MCs H.264 retenus, selon le MC MPEG-4 en entrée, sont les suivants :

- **Cas d'un MC codé Skip en MPEG-4.** L'auteur ne précise pas les MCs retenus dans ce cas.
- **Cas d'un MC codé Inter16×16 en MPEG-4.** Les MCs testés sont l'Inter16×16, l'Inter16×8 et l'Inter8×16.
- **Cas d'un MC codé Inter8×8 en MPEG-4.** Les MCs testés sont l'Inter16×8, l'Inter8×16, l'Inter8×8 et les quatre sous modes 8×8, 8×4, 4×8 et 4×4.

Les auteurs ne précisent toutefois pas comment traiter le cas des MBs d'entrée intra.

En ce qui concerne la détermination des VMs, les auteurs utilisent 3 vecteurs prédits comme points de départ du processus d'estimation de mouvement. Ces vecteurs sont les VMs MPEG-4 suivants : le vecteur médian (des VMs des blocs à gauche, en haut, et de celui en haut, mais à droite) et les vecteurs des MCs dont on a déjà estimé le mouvement. Un processus d'arrêt est également utilisé pour terminer plus tôt le processus d'estimation de mouvement. Les auteurs commencent d'abord par estimer le mouvement à partir des MVs MPEG-4. Lorsque la distorsion (SAD) est inférieure à un seuil, les autres prédicteurs ne sont pas évalués. Autrement le processus d'estimation de mouvement continue avec les autres vecteurs prédits comme points de départ. Finalement le VM choisi est celui qui présente la distorsion minimale de toutes les estimations faites à partir des vecteurs prédits.

Les auteurs n'expliquent pas comment les MBs d'entrée Skip ou intra sont traités. Tout comme l'algorithme de Liang et al. (2007), le mode Skip n'est pas considéré quand on transcode les MBs Inter16×16 de MPEG-4. Tel qu'expliqué à la section 2.4, ne pas considérer ce mode pourrait nuire à la qualité, surtout à bas débits. Le processus d'estimation semble également plus complexe que celui des autres algorithmes. L'algorithme utilise trop de prédicteurs (les VMs MPEG-4, le vecteur médian et les vecteurs des MCs testés auparavant). Ces derniers peuvent être nombreux. Par exemple, dans le cas d'un MB d'entrée Inter8×8, il faut tester les MCs Inter16×8, Inter8×16, Inter8×8 et les 4 sous modes 8×8, 8×4, 4×8 et 4×4. Pour le processus d'estimation du mode Inter4×4, les VMs de tous ces autres MCs pourront être des prédicteurs. Il est vrai que l'algorithme se sert d'une condition d'arrêt, mais il faudrait faire une étude approfondie pour voir l'efficacité de cette condition d'arrêt et mener des statistiques sur le nombre de prédicteurs utilisés exactement. Autrement, le processus de détermination des VMs proposé risque d'être aussi complexe qu'un processus d'estimation de mouvement normal.

Les auteurs ne tiennent pas non plus des conditions de codage dans leur algorithme. Tout comme les algorithmes décrits précédemment, les simulations ont été faites sur les codecs de référence. Ils obtiennent des gains de vitesse autour de 3×. La validation des résultats est toutefois difficile à juger, car trois séquences vidéos seulement sont testées.

2.6 État de l'art sur d'autres types de transcodage

Très peu d'algorithmes de transcodage de format MPEG-4 à H.264 existent dans la littérature. D'autres types de transcodage ont par contre été traités dans la littérature. Bien qu'il ne s'agisse pas du même type de transcodage, leurs idées pourraient s'appliquer au transcodage qui nous intéresse. Nous présentons brièvement, dans ce qui suit, leurs avantages, leurs inconvénients et la mesure dans laquelle certaines idées émises pourraient être appliquées à notre cas.

- **Transcodage MPEG-4 à H.264 avec réduction de résolution.** Le contexte est un peu différent. Les algorithmes ne pourraient pas s'appliquer tels quels à notre cas de transcodage,

car il faut combiner plusieurs MCs et VMs et les ramener à un nombre plus réduit. Néanmoins, le principe de réutilisation et d'estimation des VMs pourrait s'appliquer à notre cas. En Nguyen et al. (2007) et Dinh et al. (2007), les algorithmes présentent l'avantage d'adapter la fenêtre de recherche des VMs en fonction de la complexité du MB. Pour des MBs complexes, la fenêtre est élargie, ce qui peut conduire à des VMs plus efficaces. Pour ceux qui sont moins complexes, cette fenêtre est réduite et du même coup la lourdeur du processus d'estimation, car moins de points sont testés. Les auteurs obtiennent des accélérations de vitesse de $3\times$ à $4\times$ pour une perte en qualité de 0,12 dB.

Les auteurs réalisent par contre le transcodage en considérant de petites partitions telles que 8×4 , 4×8 et 4×4 dans certains cas. Cela est un inconvénient, car la table 4.1 de correspondance des MCs a montré que ces modes interviennent rarement et n'ont pas d'un effet sur la qualité ou très peu, sauf à haut débit (Joch et Kossentini (2001)). Considérer de telles partitions ne ferait qu'augmenter la complexité sans pour autant améliorer la qualité.

Les deux travaux présentent l'inconvénient de ne valider les résultats que sur deux séquences vidéos à mouvements lents. Dans ce contexte, il est difficile de juger de l'efficacité des algorithmes proposés. Par ailleurs, tout comme la plupart des travaux de la littérature, ces tests ont été réalisés sur les codecs de référence de MPEG-4 et de H.264. Il est ainsi difficile de juger de l'efficacité de ces algorithmes sur le plan de la vitesse.

- **Transcodage MPEG-2 à H.264.** MPEG-2 a de nombreuses similarités avec MPEG-4. En Tang et Nasiopoulos (2010), le principe de transcodage de MPEG-2 à H.264 pourra ainsi aisément être adapté au transcodage MPEG-4 à H.264. L'algorithme présente l'avantage d'utiliser les VMs MPEG-4 du MB associé et ceux des MBs voisins comme points de départ du processus d'estimation de mouvement H.264. Le MC est déterminé en utilisant le processus de RDO. La distorsion et le coût en bits du RDO sont par contre évalués à l'aide de formules simplifiées. Elles sont prédites à partir du QP et de la distorsion obtenue lors du processus d'estimation de mouvement, soit de la SATD. Cela évite le complexe proces-

sus de transformation fréquentielle, de codage entropique et de reconstruction du MB pour évaluer la distorsion et le nombre de bits requis. Ce principe de prédiction de la distorsion et du nombre de bits pourrait s'appliquer dans notre cas. Il permettrait d'améliorer considérablement les gains de vitesse. Il faut cependant s'assurer de l'efficacité de ces prédictions dans toutes les conditions de transcodage.

Les auteurs obtiennent une perte en qualité moyenne de 0,2 dB et des temps plus rapides de 99% comparativement à la méthode en cascade. Ces résultats ont été faits sous des codecs de référence non optimaux sur le plan de la vitesse. Il est important de souligner que les résultats en qualité seraient nettement moins intéressants en appliquant cet algorithme au transcodage MPEG-4 à H.264, car MPEG-2 est moins performant en qualité que MPEG-4.

- **Transcodage H.263 à H.264.** Bialkowski et al. (2006) et Bialkowski et al. (2004) présentent un algorithme qui réduit la complexité de transcodage pour le processus de prédiction intra. Ils extraient l'information résiduelle et la direction de prédiction intra. L'énergie résiduelle est utilisée comme mesure d'efficacité du mode intra à l'aide d'un seuil déterminé empiriquement. La direction de prédiction d'H.263 est retenue en H.264 pour des MBs à énergie inférieure au seuil, autrement le processus normal de prédiction intra de H.264 est réalisé.

Leurs idées pourraient s'appliquer à notre transcodage. Cela éviterait, pour H.264, le complexe processus de RDO pour les 9 modes de prédiction intra permis par le standard H.264. Cependant, l'impact sur la vitesse ne devrait pas être très important. Les taux de rafraîchissement de trames intra est tel qu'il y a très peu de trames intra comparativement aux trames inter. Réduire la complexité de transcodage des trames intra devrait, par conséquent, avoir peu d'impact sur la vitesse pour toute la séquence vidéo, bien que des améliorations de vitesse intéressantes puissent être obtenues au niveau des trames intra. Il serait plus avantageux de s'intéresser à réduire la complexité des trames inter.

- **Algorithmes de réduction de la complexité de l’encodage H.264.** De nombreux auteurs ont contribué à réduire la complexité de l’encodeur H.264 sans s’intéresser à un transcodage de format proprement dit (Choi et al. (2006)). Les solutions présentées par ces auteurs pourraient aider à réduire davantage la complexité en calculs lors d’un tel transcodage. Il faut cependant souligner que ces solutions sont efficaces dans un contexte d’opération unique d’encodage H.264. Toutefois, dans un cadre de transcodage MPEG-4 à H.264, ces solutions ne sont pas suffisantes pour atteindre de bonnes performances en vitesse. Il faudrait les combiner à d’autres méthodes et informations supplémentaires telles que les MCs et les VMs MPEG-4 pour accélérer considérablement le processus de transcodage.

Choi et al. (2006) par exemple allègent la complexité du processus de RDO en détectant précocement les MBs susceptibles d’être codés en Skip. Ce mode est sélectionné à un pourcentage assez élevé. Après avoir subi le coûteux processus de RDO, de nombreux MBs finissent par être codés en Skip. Une détection précoce éviterait ainsi à de nombreux MBs de passer par le RDO. Les auteurs testent certaines conditions et lorsqu’elles sont remplies, le mode Skip est choisi. L’idée de détecter ce type de MCs est assez intéressante. Cependant, pour les MBs qui ne remplissent pas les conditions du mode Skip, il faudrait effectuer le processus normal de RDO. Même si le nombre de MCs qui passe par ce processus normal est faible, la complexité de ce processus est telle qu’une simple détection précoce des modes Skip serait insuffisante pour atteindre de bons gains de vitesse.

2.7 Conclusion

La littérature n’est pas très abondante en ce qui concerne le transcodage MPEG-4 à H.264. Les algorithmes développés ont l’avantage d’utiliser les informations MPEG-4 telles que la DCT, les VMs et les MCs pour réduire considérablement la complexité en calculs du transcodage MPEG-4 à H.264. De bonnes accélérations ont pu être obtenues. L’état de l’art présente néanmoins une carence générale, celle de ne pas adapter les méthodes aux conditions de codage. Ce qui fait que l’efficacité des méthodes est assez variable. Par ailleurs les tests et simula-

tions ont été réalisés sur des codecs de référence qui sont loin d'être optimaux en vitesse. Les résultats en vitesse présentés sont ainsi peu représentatifs de ceux à obtenir sur des produits commerciaux.

Dans cette thèse, nous opterons pour une architecture dans le domaine spatial à cause du manque de flexibilité et des sévères accumulations d'erreurs des architectures fréquentielles. L'idée émise par Lee et Lee (2005) et Lee et al. (2006), qui est de déterminer l'ensemble des MCs les plus probables en rassemblant les statistiques de correspondance des MCs H.264 et MPEG-4 est assez intéressante. Nous exploiterons cette idée, mais nous irons plus loin dans l'analyse statistique. Nous exploiterons d'autres informations, notamment l'information résiduelle pour réduire davantage les modes candidats H.264. Les VMs MPEG-4 seront également recueillis et adaptés pour réduire l'énorme complexité du module d'estimation de mouvement. Nous tenterons également de remédier à une carence générale de l'état de l'art, qui consiste à ne pas tenir compte des conditions de codage, en investiguant des algorithmes efficaces quels que soient les caractéristiques des séquences vidéos et le débit binaire. Deux algorithmes ont été développés dans le cadre de nos recherches. Ces derniers sont présentés dans les chapitres suivants.

CHAPITRE 3

MÉTHODE ET ALGORITHME DE TRANSCODAGE MPEG-4 À H.264 BASÉS SUR L'INFORMATION RÉSIDUELLE

3.1 Introduction

Nous opterons pour un algorithme de transcodage dans le domaine spatial plutôt que dans le domaine fréquentiel pour plusieurs raisons. Les gains en accélération sont souvent moins élevés pour le transcodage vers H.264, car plus de MCs sont considérés dans ce format de sortie que dans les autres standards. De plus, les opérations d'estimation du mouvement sont plus complexes à réaliser dans le domaine fréquentiel, surtout pour une précision au quart de pixel. Les approches fréquentielles sont également beaucoup plus complexes à implémenter. Les implémentations sont moins flexibles et réutilisables, contrairement au domaine spatial, dans lequel les boucles du décodeur et du codeur sont indépendantes. On évite aussi les erreurs de propagation souvent associées à un transcodage dans le domaine fréquentiel (les inconvénients et les avantages des domaines spatial et fréquentiel sont détaillés en Ahmad et al. (2005)).

Le signal MPEG-4 sera donc décodé puis réencodé selon les spécifications H.264. Par contre, afin de pallier à la complexité de ce type de transcodage, nous nous inscrirons dans la même optique que la plupart des algorithmes de transcodage décrits dans le chapitre précédent. Ces derniers atteignent leur efficacité en extrayant des informations du train binaire MPEG-4 et en les exploitant dans le codeur H.264 afin d'accélérer les modules les plus complexes en calcul. Ces modules sont notamment les processus de décision des MCs et des VMs.

Lee et al. (2006) exploitent la distribution fréquentielle des MCs de H.264 pour un MC MPEG-4 donné, en dérivant une table de conversion entre les MCs MPEG-4 et ceux d'H.264. Le temps de détermination des MCs est allégé en ne considérant que les MCs les plus probables. Au niveau des VMs, les auteurs utilisent les VMs MPEG-4 comme point de départ pour la recherche des VMs H.264 et allègent ainsi le temps de recherche des VMs.

Nous perfectionnerons l'idée de Lee et al. (2006), aussi bien pour les MCs que pour les VMs pour le transcodage des trames inter. Pour cela, nous exploiterons une autre information, l'information résiduelle, pour réduire davantage l'ensemble des MCs à considérer et, ainsi, réduire plus agressivement la complexité des calculs. L'information résiduelle sera également exploitée pour réduire le nombre de VMs à raffiner.

Les trames intra ont un impact important sur la qualité visuelle. Ainsi, le transcodage des trames intra se fera par la méthode en cascade, pour laquelle les meilleures performances en qualité sont normalement atteintes. Nous voulons atteindre la meilleure qualité possible pour ce type de trames. Les trames intras seront donc décodées à l'aide du décodeur MPEG-4 et re-encodées complètement selon les spécifications H.264. C'est-à-dire que le codeur H.264 choisira le meilleur MC et la meilleure prédiction intra en terme de qualité/distorsion, en effectuant le processus RDO normal pour tous les MCs H.264 permis par le standard. Par ailleurs, même si on s'attelle à réduire la complexité de calculs du transcodage des trames intra, l'impact sur la complexité totale de la séquence vidéo est faible. Les trames intra étant moins nombreuses que les trames inter. Les efforts seront plutôt investis au niveau du transcodage des trames inter.

Il faut mentionner que nous supposons que le type des trames H.264 est le même que celui des trames MPEG-4 entrant. C'est-à-dire que le taux de rafraîchissement des trames intras est le même.

Le chapitre est organisé comme suit : dans un premier temps, nous présentons les propriétés de l'information résiduelle, que nous exploiterons dans les processus de détermination des MCs et des VMs pour le transcodage inter. Nous décrirons ensuite la méthode de détermination de l'ensemble des MCs candidats et le processus de détermination des VMs H.264. Finalement, nous présenterons les résultats de simulation et terminerons par une conclusion.

3.2 Information résiduelle en tant que mesure de complexité des MBs et d'efficacité des VMs et des MCs

Les standards de compression vidéo, tels qu'expliqués dans le chapitre 1, tentent d'éliminer la redondance spatiale et temporelle entre les trames pour réaliser une compression vidéo efficace. Pour ce faire, un prédicteur de la trame est généré par un bloc compensateur de mouvement (bloc CM de la figure 3.1). Cette prédiction est soustraite de la trame à coder pour former une image différentielle, connue sous le nom d'image résiduelle. La trame résiduelle contiendra idéalement des valeurs de faible amplitude et aura donc, moins d'énergie que la trame originale. Elle nécessitera, par conséquent, moins de bits pour son encodage. La figure 3.1 présente un codeur vidéo et illustre l'emplacement de l'information résiduelle. I^t ,

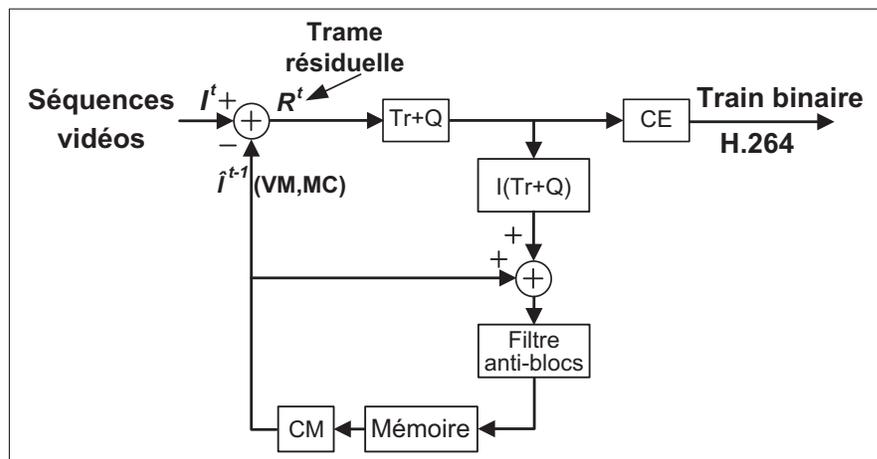


Figure 3.1 Illustration de l'information résiduelle dans un schéma de codeur vidéo.

$\hat{I}^{t-1}(\text{VM}, \text{MC})$ et R^t représentent, respectivement la trame à coder au temps t , le prédicteur de cette trame (c'est-à-dire la trame passée reconstituée, au temps $t-1$, qui constitue un estimé de cette trame, compensée par le vecteur de mouvement VM et utilisant le mode de codage MC) et la trame résiduelle au temps t .

$R^t = I^t - \hat{I}^{t-1}(\text{VM}, \text{MC})$. Si une bonne combinaison du MC et des VMs est choisie, $\hat{I}^{t-1}(\text{VM}, \text{MC})$ se rapproche de I^t et les pixels de R^t devraient tendre vers zéro. L'information résiduelle est donc un bon indicateur de l'efficacité du MC et des VMs choisis. L'information

résiduelle jouera un rôle essentiel dans nos algorithmes. Nous nous en servons dans les processus de détermination des MCs et des VMs, notamment pour réduire le nombre de MCs à tester et le nombre de VMs à raffiner. Concrètement, nous allons utiliser l'énergie résiduelle comme mesure quantitative de l'information résiduelle. Définissons formellement cette énergie résiduelle. Soit $\mathbf{K}_n(i, j)$ le MB n original et $\hat{\mathbf{K}}_n(i, j)$, son MB prédit. Avec la condition $0 \leq i, j \leq 15$. Le résiduel du MB n se définit comme suit :

$$\mathbf{R}_n(i, j) = \mathbf{K}_n(i, j) - \hat{\mathbf{K}}_n(i, j), \quad 0 \leq i, j \leq 15 \quad (3.1)$$

L'énergie résiduelle du MB n se définit alors comme :

$$E_n = \sum_{i=0}^{15} \sum_{j=0}^{15} [\mathbf{R}_n(i, j)]^2 = \sum_{i=0}^{15} \sum_{j=0}^{15} [\mathbf{K}_n(i, j) - \hat{\mathbf{K}}_n(i, j)]^2 \quad (3.2)$$

L'énergie résiduelle peut également être évaluée sur un bloc k , de dimension 8×8 du MB n . Nous la noterons $E_{n,k}$. Dans ce cas elle se calcule selon la formule 3.3.

$$E_{n,k} = \sum_{i=0}^7 \sum_{j=0}^7 [\mathbf{R}_n(i + p_{kx}, j + p_{ky})]^2 \quad (3.3)$$

$\mathbf{R}_n(i + p_{kx}, j + p_{ky})$, étant les valeurs de pixels du résiduel appartenant au bloc k du MB n . Avec $\mathbf{p}_k = (p_{kx}, p_{ky})$ pour $0 \leq k \leq 3$, où $\mathbf{p}_0 = (0, 0)$, $\mathbf{p}_1 = (8, 0)$, $\mathbf{p}_2 = (0, 8)$, et $\mathbf{p}_3 = (8, 8)$.

On peut constater clairement que l'énergie résiduelle d'un MB, E_n , peut s'exprimer comme étant la somme des énergies $E_{n,k}$ des quatre blocs 8×8 , $E_n = \sum_{k=0}^3 E_{n,k}$.

3.2.1 Énergie résiduelle en tant que mesure d'efficacité des vecteurs de mouvement

Lorsque le VM traduit efficacement le mouvement de la partition, l'énergie résiduelle tend vers zéro. Plus l'estimation de mouvement est précise, moins l'information résiduelle contiendra d'énergie. Ce qui n'est pas le cas lorsque l'estimation de mouvement est peu précise. L'énergie résiduelle est donc un bon indicateur de l'efficacité des VMs choisis et peut être

utilisée comme mesure d'efficacité du processus d'estimation de mouvement. Il existe donc un lien entre l'énergie résiduelle et l'efficacité du mouvement.

Les VMs MPEG-4 seront réutilisés dans H.264 afin d'éviter de recommencer à zéro le processus d'EM et de réduire ainsi la complexité en calculs. Les VMs seront, par contre, raffinés pour améliorer la qualité. Mais il est inutile de raffiner un VM si ce dernier est déjà efficace. Ainsi, en utilisant l'énergie résiduelle comme critère d'efficacité, on pourra éviter de raffiner inutilement les VMs qui n'en ont pas besoin et sauver ainsi des calculs. Avant de le faire, il serait important de vérifier notre hypothèse sur le lien entre l'énergie résiduelle et l'efficacité des VMs. L'efficacité d'un VM augmente avec sa précision, lorsque le mouvement est évalué sur des partitions plus petites (spécialement pour des vidéos à mouvement complexe). À la figure 3.2, nous avons mesuré l'énergie résiduelle pour différentes précisions (au pixel entier, au demi-pixel et au quart de pixel) et, à la figure 3.3, nous avons mesuré l'énergie résiduelle pour différents types de vidéos avec des types de partitionnement différents, afin d'illustrer le lien entre l'efficacité des VMs et l'énergie résiduelle. Comme on peut le constater à la

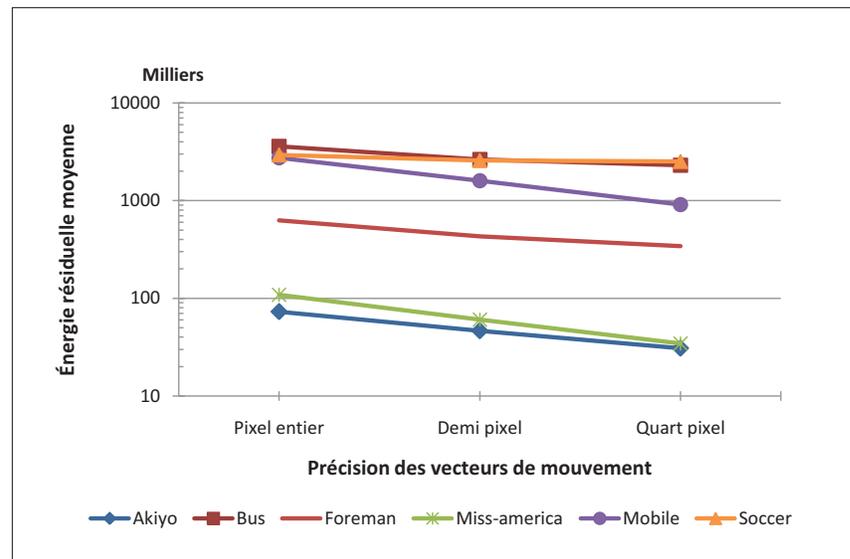


Figure 3.2 Illustration du lien entre l'énergie résiduelle et l'efficacité du VM par un codage MPEG-4 à différentes précisions de VMs. L'axe vertical est en échelle logarithmique.

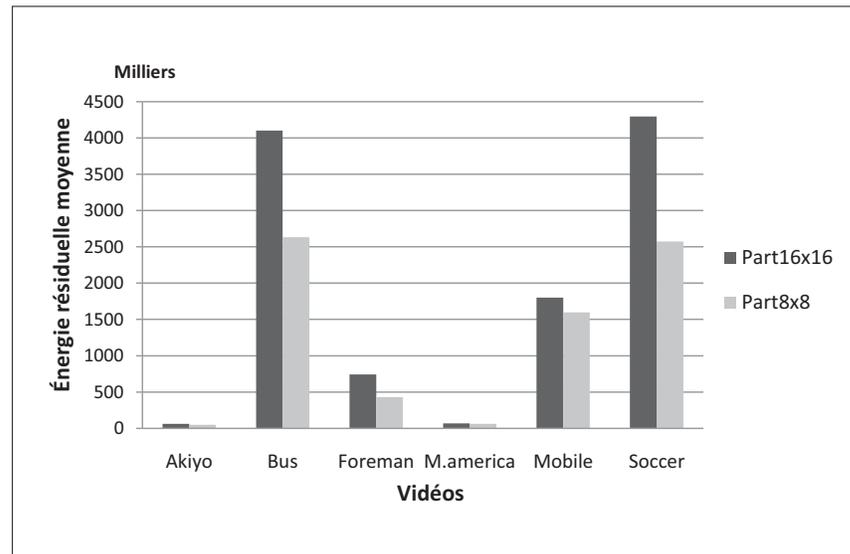


Figure 3.3 Illustration du lien entre l'énergie résiduelle et l'efficacité du VM par un codage MPEG-4 à différent partitionnement.

figure 3.2, lorsque la précision des VMs augmente, soit lorsque l'efficacité du VM augmente, l'énergie résiduelle baisse. C'est le cas également, à la figure 3.3, lorsque le partitionnement permis du MC passe de 16×16 à 8×8 . En permettant un partitionnement plus petit, l'énergie résiduelle est plus basse, car le VM est plus efficace puisqu'il peut être évalué sur une surface plus petite. Les résultats sont plus prononcés pour les vidéos à mouvements complexes. On constate également que l'énergie résiduelle augmente avec la complexité du mouvement des séquences vidéos (celle de Soccer et Mobile est nettement plus élevée que celle de Akiyo ou de Miss-america). Les séquences vidéos utilisées pour les simulations sont des séquences bien connues en traitement vidéo et sont celles habituellement utilisées pour les tests de validation (Arizona State University (2011), Xiph.org (2011)).

3.2.2 Lien entre l'énergie résiduelle et le partitionnement du MC

Lorsqu'un MB a un mouvement complexe ou est dans une zone avec beaucoup de détails, il faudrait évaluer ses VMs sur des partitions plus petites afin qu'ils soient plus efficaces et représentent mieux le mouvement du MB (voir la figure 3.3).

Toutefois, le standard MPEG-4 impose des limites au partitionnement du MC, la taille minimale de partitionnement est 8×8 . Le *visual simple profile* impose également des limites quant à la précision des VMs qui est limitée au demi-pixel. Par ailleurs, les contraintes de débit binaire imposent également des limites sur le choix du MC. En effet, le choix de petites partitions implique de transmettre plus de VMs, soit plus de bits pour les VMs. Par contre, le résiduel sera plus faible et ainsi, nécessitera moins de bits pour sa transmission. Le codeur peut être par conséquent, obligé de choisir des partitions plus grandes bien que n'étant pas parfaites. Toutes ces contraintes, de partitionnement, de précision de VMs et de débit binaire font que, bien souvent, le MC et les VMs choisis ne sont pas parfaits pour représenter le MB. Ce qui résulte en un résiduel plus élevé, spécialement pour des MBs plus complexes, qui nécessitent des partitions plus petites et des VMs plus précis. En effectuant le transcodage en H.264, un tel MB pourra être codé en partitions plus petites que ne le permettraient les outils de codage MPEG-4. Ce qui implique que, généralement, les MBs à résiduel élevé auront tendance à être codés en plus petites partitions en H.264.

En effet H.264 offre des outils de compression nettement plus efficaces que MPEG-4 (codage entropique binaire plus puissant qu'en MPEG-4, le CABAC, une précision des VMs plus grande, au $1/4$ de pixel, une prédiction intra) qui font que le codeur peut se permettre une allocation binaire plus grande pour les VMs, et ainsi, peut se permettre des partitions de MCs plus petites (par exemple des VMs plus précis entraînent un résiduel plus faible, soit moins de bits à coder pour le résiduel et un budget plus élevé pour permettre plus de VMs par MB). De plus, H.264 permet un partitionnement de MCs aussi petit que 4×4 .

Le partitionnement du MB augmente donc avec l'information résiduelle. Ce principe sera utilisé pour réduire le nombre de MCs à considérer, contrairement aux algorithmes de l'état de l'art qui testent un nombre élevé de MCs. Les MCs à petites partitions ne seront considérés que pour des MBs à énergie résiduelle élevée et vice versa. Il est important, avant de présenter l'algorithme, de mettre en évidence, ce lien car c'est un principe essentiel qui sera exploité dans le processus de détermination des MCs. Ce lien est illustré à la figure 3.4. On peut constater effectivement la relation entre l'énergie résiduelle et le MC H.264. Les régions à forte énergie

ont tendance à être codées en plus petites partitions que ne le sont celles à faible d'énergie. On constate également que, dans les régions à mouvement plus complexe, le partitionnement est plus petit et l'énergie résiduelle plus élevée. En résumé, nous proposons d'utiliser le résiduel

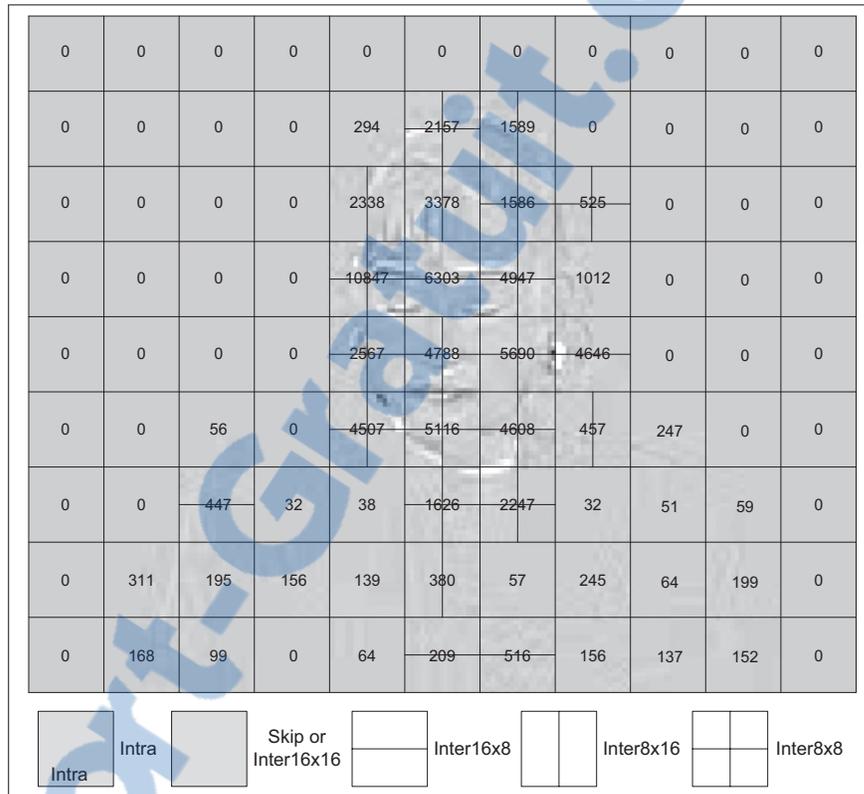


Figure 3.4 Illustration de la segmentation de codage d'une trame par rapport à l'énergie résiduelle pour la vidéo QCIF Akiyo. Les valeurs à l'intérieur du MB sont l'énergie résiduelle du MB.

comme indicateur d'efficacité des MCs et des VMs. L'usage de cette information permettra d'accélérer le processus de détermination des MCs ainsi que des VMs. Seuls les MCs les plus probables, compte tenu de l'énergie résiduelle, seront considérés. Les VMs jugés efficaces seront maintenus tels quels ; sinon, ils seront raffinés. On pourra ainsi éviter d'évaluer des MCs ou d'effectuer des raffinements de VMs inutiles et coûteux.

3.3 Détermination du mode de codage

Pour déterminer le MC du MB, un codeur H.264 devrait évaluer la distorsion pour tous les huit MCs et les quatre sous-modes 8×8 permis par le standard et finir par retenir celui dont la distorsion est minimale. Ce processus est assez lourd en terme de calculs surtout que cela implique de trouver avant tout les VMs pour les modes concernés. C'est pour cette raison que plusieurs encodeurs H.264 ne testent pas tous les modes. Ils commencent par tester certains modes et arrêtent de tester de nouveaux modes si ceux déjà testés mènent à des résultats satisfaisants (par exemple, une erreur de prédiction sous un seuil). Nous proposons aussi de réduire le nombre de MCs à tester, mais en fonction de l'information MPEG-4 reçue. Pour cela, nous effectuerons une classification efficace des MBs MPEG-4 d'entrée de sorte à ne retenir que les MCs les plus probables.

Premièrement, nous utiliserons l'idée tirée de Lee et al. (2006) qui consiste à ne tester que les modes les plus probables. Pour cela, les auteurs établissent la fréquence de codage des modes H.264 du MB en fonction du MC dans MPEG-4. Une table de correspondance des MCs H.264 et MPEG-4 est ainsi établie lors d'un transcodage en cascade. De telles tables sont montrées dans les tableaux 3.1 et 3.2, qui représentent les moyennes obtenues pour plusieurs vidéos QCIF pour des débits binaires H.264 de 32 et 160 kb/s. Les séquences étaient initialement encodées en MPEG-4 à 200 kb/s. Il faut noter que les valeurs dans les zones en gris sont les pourcentages des quatre sous-modes 8×8 par rapport au MC 8×8 . Les valeurs en pourcentage devant l'Inter 16×16 , l'Inter 8×8 et le Skip sont les répartitions en pourcentage de ces MCs pour la totalité des trames inter.

Ensuite, nous proposons de restreindre drastiquement le nombre de MCs à tester en tirant avantage du fait que le partitionnement du MB augmente avec l'énergie résiduelle.

Les pourcentages du mode PCM ne sont pas illustrés dans les tables, car leurs pourcentages d'occurrence sont toujours nuls parce que ce mode est utilisé pour la compression sans perte. Comme on peut le constater dans ces tables, certains modes et sous modes 8×8 apparaissent très peu, et ce, quel que soit le débit de codage H.264. C'est, par exemple, le cas des sous-

Tableau 3.1 Moyenne de fréquence d'apparition des MCs H.264 en fonction des MCs MPEG-4 pour plusieurs types de vidéos QCIF transcodées à 32 kb/s en H.264, en utilisant le codeur H.264 d'Intel. La séquence MPEG-4 est initialement encodée à 200 kb/s

Modes de codage H.264 / Modes de codage MPEG-4	Intra 4×4	Intra 16×16	Skip	Inter 16×16	Inter 16×8	Inter 8×16	Inter 8×8	sb8×8	sb8×4	sb4×8	sb4×4
Intra	64,2%	26,1%	7,8%	1,6%	0,1%	0,1%	0,1%	100,0%	0,0%	0,0%	0,0%
Inter16×16 (64%)	0,3%	1,9%	66,2%	27,7%	2,0%	1,8%	0,1%	99,8%	0,1%	0,1%	0,0%
Inter8×8 (21%)	2,9%	3,8%	21,7%	39,5%	14,9%	16,8%	0,4%	98,1%	0,5%	0,7%	0,7%
Skip (15%)	0,1%	0,6%	92,3%	6,8%	0,1%	0,1%	0,0%	0,0%	0,0%	0,0%	0,0%

Tableau 3.2 Moyenne de fréquence d'apparition des MCs H.264 en fonction des MCs MPEG-4 pour plusieurs types de vidéos QCIF transcodées à 160 kb/s en H.264, en utilisant le codeur H.264 d'Intel. La séquence MPEG-4 est initialement encodée à 200 kb/s

Modes de codage H.264 / Modes de codage MPEG-4	Intra 4×4	Intra 16×16	Skip	Inter 16×16	Inter 16×8	Inter 8×16	Inter 8×8	sb8×8	sb8×4	sb4×8	sb4×4
Intra	73,8%	21,6%	1,8%	1,9%	0,3%	0,2%	0,4%	100,0%	0,0%	0,0%	0,0%
Inter16×16 (64%)	1,0%	2,7%	28,5%	54,9%	6,0%	5,8%	1,1%	98,5%	0,4%	0,4%	0,7%
Inter8×8 (21%)	6,7%	1,0%	1,0%	22,8%	10,4%	16,5%	41,6%	78%	3,8%	6,0%	12,2%
Skip (15%)	0,1%	0,6%	85,7%	12,9%	0,3%	0,2%	0,2%	100,0%	0,0%	0,0%	0,0%

modes 8×4, 4×8 et 4×4. D'ores et déjà, ces MCs peuvent être exclus de la liste des candidats à considérer sans que cela n'ait un impact réel sur la qualité. On remarque aussi qu'un fort pourcentage des macroblocs codés en Inter16×16 (représentant plus de 60% des blocs reçus) est réencodé soit en Skip ou en Inter16×16. Nous allons tenter, en exploitant l'énergie résiduelle, de mieux classer ces macroblocs pour pouvoir déterminer plus rapidement ceux qui doivent être codés en Skip. Nous éviterons ainsi d'évaluer inutilement le mode Inter16×16 d'H.264. Nous cherchons ainsi à exploiter l'énergie résiduelle pour effectuer une classification efficace des macroblocs d'entrée nous permettant de réduire le nombre de candidats à évaluer, donc de réduire les calculs, sans affecter la qualité de manière notable.

Nous présentons maintenant l'algorithme de transcodage. Les sous-sections suivantes détaillent le processus de transcodage selon chacun des MCs MPEG-4, soit pour les MCs Intra, Skip, Inter16×16 et Inter8×8.

3.3.1 Détermination du mode de codage pour des macroblochs codés Intra en MPEG-4

Dans ce cas de transcodage, nous utiliserons uniquement la propriété de fréquence d'apparition des MCs H.264 en fonction de ceux de MPEG-4. Nous pouvons observer, en consultant les tables 3.1 et 3.2, que ces MBs ont tendance à être réencodés en Intra4×4 et en Intra16×16. Les autres MCs apparaissent rarement. Les deux MCs à tester seront donc Intra4×4 et Intra16×16. Soit mci , le mode de codage d'un MB en MPEG-4. Notons $MCC_{mci}^{H.264}$, l'ensemble des MCs candidats H.264 à tester pour ce MB. Pour le transcodage des MBs codés Intra en MPEG-4, $MCC_{Intra}^{H.264} = \{Intra16\times16, Intra4\times4\}$.

3.3.2 Détermination du mode de codage pour des macroblochs codés Skip en MPEG-4

Comme pour le cas Intra, nous n'utiliserons que la propriété de fréquence d'apparition des MCs H.264. Les MBs ont tendance à être réencodés en Skip (à 92,3% à 32 kb/s et à 85,7% à 160 kb/s). On ne considérera que le mode Skip dans ce cas, $MCC_{Skip}^{H.264} = \{Skip\}$.

3.3.3 Détermination du mode de codage pour des macroblochs codés Inter16×16 en MPEG-4

Dans ce cas de transcodage, la première étape consiste à éliminer certains candidats en tenant compte du principe d'occurrence des MCs les plus probables. Comme le montrent les tables 3.1 et 3.2, les MCs Skip et Inter16×16 apparaissent de façon plus importante que les autres MCs (à 66,2% pour du Skip et à 27,7% pour du Inter16×16 à 32 kb/s, à 28,5% pour du Skip et à 54,9% pour du Inter16×16 à 160 kb/s). Les autres modes apparaissent très peu, quel que soit le débit. On peut d'ores et déjà éliminer plusieurs modes pour ne tester que Skip et Inter16×16.

Ensuite, en classifiant les MBs selon leur énergie résiduelle, nous pourrions idéalement ramener le nombre de MCs à tester à un seul MC, Skip pour les MBs jugés à faible complexité ou

Inter16×16 pour ceux à haute complexité. Pour cela, deux seuils d'énergie, T_{mc}^{Skip} et $T_{mc}^{16 \times 16}$, seront concrètement utilisés pour la classification des MBs, qui se fera en trois catégories, Inter16×16 cas I, Inter16×16 cas II et Inter16×16 cas III comme suit :

- Inter16×16 cas I : Nous voulons regrouper dans cet ensemble les MBs susceptibles d'être codés en Skip. Ces MBs devraient être à faible énergie et suivre le même mouvement que celui de ses voisins. On rappelle que le mode Skip en MPEG-4 n'a pas la même définition qu'en H.264. Un MB est codé Skip en H.264 s'il suit le même mouvement que ses voisins, c'est-à-dire si son VM est égal au prédicteur. Notons $\mathbf{v}_{\text{mp4}} = (v_{\text{mp4}x}, v_{\text{mp4}y})$, le VM MPEG-4 du MB et $\mathbf{v}_p = (v_{px}, v_{py})$, le vecteur prédit en H.264. Pour qu'un MB ait une probabilité élevée d'être codé en Skip, il faudrait que ce MB ait un faible résiduel et que le VM MPEG-4 soit proche du vecteur prédit. Un MB appartient donc à la catégorie Inter16×16 cas I si la condition suivante est respectée :

$$E_n < T_{mc}^{\text{Skip}} \text{ et } |\mathbf{v}_{\text{mp4}} - \mathbf{v}_p| \leq 1 \quad (3.4)$$

Nous signifions par $|\mathbf{v}_{\text{mp4}} - \mathbf{v}_p| \leq 1$ que $|v_{\text{mp4}x} - v_{px}| \leq 1$ et $|v_{\text{mp4}y} - v_{py}| \leq 1$. Ainsi la distance doit être au plus d'un pixel entier dans chaque dimension.

- Inter16×16 cas II : Nous voulons regrouper, dans cet ensemble, les MBs susceptibles d'être codés Skip ou Inter16×16. Ces MBs devraient être, soit à complexité moyenne, soit à faible énergie, mais ne suivant pas le même mouvement que leurs voisins. Les MBs dont l'énergie est comprise entre les seuils T_{mc}^{Skip} et $T_{mc}^{16 \times 16}$ seront considérés à complexité moyenne, tel que décrit dans l'équation 3.6. Un MB qui respecte donc la condition 3.5 sera considéré comme ayant une faible énergie, mais ne respectant pas la condition de proximité entre le VM MPEG-4 et le vecteur prédit, c'est-à-dire que ce MB ne suit pas le même mouvement que ses voisins.

$$E_n < T_{mc}^{\text{Skip}} \text{ et } |\mathbf{v}_{\text{mp4}} - \mathbf{v}_p| > 1 \quad (3.5)$$

$$T_{mc}^{\text{Skip}} \leq E_n \leq T_{mc}^{16 \times 16} \quad (3.6)$$

Un MB appartient à l'Inter16×16 cas II si la condition 3.6 ou 3.5 est respectée. $|\mathbf{v}_{mp4} - \mathbf{v}_p| > 1$ signifie que $|v_{mp4x} - v_{px}| > 1$ ou $|v_{mp4y} - v_{py}| > 1$. Les candidats à considérer sont Skip et Inter16×16 pour ce cas, ce qui signifie que $MCC_{16 \times 16}^{H.264} = \{\text{Skip}, \text{Inter16} \times 16\}$.

- Inter16×16 cas III : Nous voulons regrouper dans cet ensemble, les MBs à complexité élevée. Ces MBs devraient avoir une énergie résiduelle élevée, c'est-à-dire que $E_n > T_{mc}^{16 \times 16}$. La condition 3.7 doit être respectée. Nous avons observé expérimentalement, et ce sera montré plus loin, que ce type de MB tend à être réencodé en tant que Inter16×16 la plupart du temps, $MCC_{16 \times 16}^{H.264} = \{\text{Inter16} \times 16\}$.

$$E > T_{mc}^{16 \times 16} \quad (3.7)$$

Le processus de détermination du MC H.264 pour un MB codé Inter16×16 en MPEG-4 est résumé à la figure 3.5. Les MBs Inter16×16 d'entrée MPEG-4 représentent le plus grand pourcentage des MCs (en moyenne 64% pour les vidéos). Et donc, le fait de classer les MBs afin de ramener le nombre de MCs candidats à un seul dans certaines classes, a un impact important sur la réduction de la complexité.

Il faut toutefois fixer T_{mc}^{Skip} adéquatement, de sorte que les MBs Inter16×16, avec une énergie résiduelle en dessous de T_{mc}^{Skip} et ayant un VM similaire au vecteur prédit, \mathbf{v}_p soient codés en Skip avec une très forte probabilité. Similairement, il faut fixer une valeur pour $T_{mc}^{16 \times 16}$, de sorte que les macroblocs avec une énergie résiduelle qui lui sont supérieure soient transcodés en Inter16×16 avec une très forte probabilité. Les seuils ont été fixés empiriquement à $\{T_{mc}^{\text{Skip}} = 125, T_{mc}^{16 \times 16} = 5000\}$, après des analyses et des centaines de simulations.

Il est important de noter que toutes les constantes utilisées dans notre système ont été déterminées par simulations sur différents types de séquences d'entraînement vidéos, complètement différentes de celles qui ont servi à la validation des résultats. Les constantes ont été choisies de sorte à obtenir un bon compromis entre la vitesse et la qualité. Les séquences d'entraînement utilisées sont les suivantes ; les séquences QCIF (Akiyo, Bridge-close, Coastguard) et CIF (Container, Mobile et Waterfall).

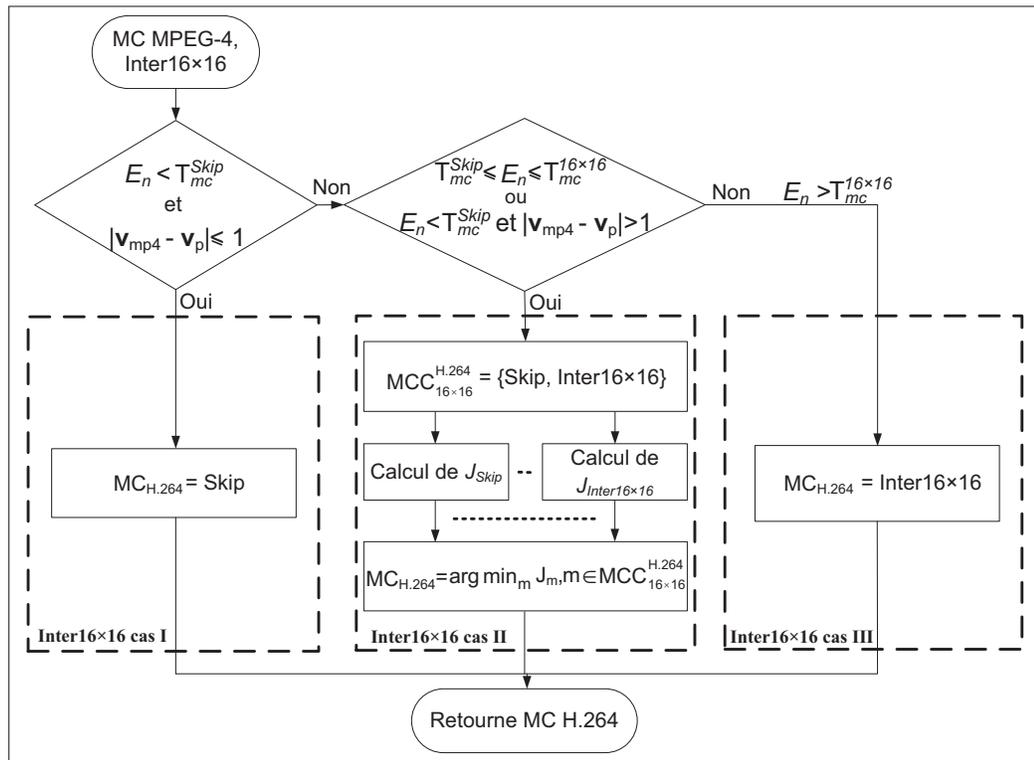


Figure 3.5 Processus de détermination du mode de codage H.264 pour un MB codé Inter16×16 en MPEG-4.

3.3.4 Détermination du mode de codage pour des macroblobs codés Inter8×8 en MPEG-4

Pour le cas du transcodage des MBs Inter8×8, les MCs H.264, Inter16×16, Inter16×8 et Inter8×16 apparaissent de façon importante, quel que soit le débit binaire, comme on peut le constater dans les tables 3.1 et 3.2. Ils feront donc partie de la liste des MCs candidats à considérer.

Tel qu'expliqué dans le paragraphe 1.8, Le choix du MC découle d'un processus d'optimisation du rapport débit distorsion. Ce qui fait que le pourcentage d'occurrence d'un MC est totalement différent selon le débit de codage. À bas débit, des partitions plus grandes seront privilégiées à cause du budget binaire plus serré qu'il ne l'est à des débits plus élevés. Cet état de fait se constate dans le pourcentage d'occurrence du MC Skip, de 21,7% à 32 kb/s à 1,0% à 160 kb/s. À un débit de codage plus grand, les contraintes binaires sont moindres

qu'elles ne le sont à bas débits, si bien que de petites partitions apparaissent en nombre plus important. C'est le cas du MC Inter8×8. On peut effectivement constater qu'à bas débit, le pourcentage est assez faible, de 0,4% à 32 kb/s pour augmenter considérablement à 41,6% à 160 kb/s. Bien qu'apparaissant rarement à certains débits, le pourcentage d'occurrence de Skip et de l'Inter8×8 peut devenir assez important pour d'autres débits. Pour cette raison, ces deux MCs devront être considérés. Pour un MB codé Inter8×8 en MPEG-4, l'ensemble des MCs candidats est le suivant : $MCC_{8 \times 8}^{H.264} = \{\text{Skip}, \text{Inter}16 \times 16, \text{Inter}16 \times 8, \text{Inter}8 \times 16, \text{Inter}8 \times 8\}$.

Il faut noter que, tout comme dans le mode Inter16×16, une classification des MBs Inter8×8 pourrait être faite afin de réduire davantage le nombre de MCs à tester. Cependant, cela s'avère nettement plus complexe, notamment, pour l'isolation des MCs Inter16×16, Inter16×8 et Inter8×16. Il faudra notamment étudier la fusion des quatre VMs 8×8 pour obtenir un seul VM (dans le cas de l'Inter16×16) ou de deux VMs (pour l'Inter16×8 et Inter8×16). Cela nécessiterait une étude plus approfondie et pourrait faire l'objet d'un autre sujet de recherche. Les critères de fusion devraient inclure l'information résiduelle, la direction des VMs 8×8, etc. Nous avons, par ailleurs, concentré nos efforts sur la classification des MBs Inter16×16 parce que ces derniers représentent près de 60% des MBs et leur impact sur la complexité des calculs est plus élevé. Cet impact devrait être moindre pour les MBs Inter8×8, car ces derniers représentent environ 20% des MBs entrants.

3.3.5 Sommaire des modes de codage H.264 candidats selon le mode de codage MPEG-4 d'entrée

Résumons l'ensemble des MCs H.264 à considérer selon chacun des MCs MPEG-4 d'entrée :

- **Intra**, $MCC_{\text{Intra}}^{H.264} = \{\text{Intra}4 \times 4, \text{Intra}16 \times 16\}$.

- **Skip**, $MCC_{\text{Skip}}^{H.264} = \{\text{Skip}\}$.

- **Inter16×16**

- **Inter16×16 cas I** : Si $E_n < T_{mc}^{\text{Skip}}$ et $|\mathbf{v}_{mp4} - \mathbf{v}_p| \leq 1$, $MCC_{\text{Inter}16 \times 16}^{H.264} = \{\text{Skip}\}$.

- **Inter16×16 cas II** : $\left\{ \begin{array}{l} \text{Si } T_{mc}^{\text{Skip}} \leq E_n \leq T_{mc}^{16 \times 16} \text{ ou,} \\ E_n < T_{mc}^{\text{Skip}} \text{ et } |\mathbf{v}_{mp4} - \mathbf{v}_p| > 1 \end{array} \right\}$, $MCC_{16 \times 16}^{H.264} = \{\text{Skip}, \text{Inter}16 \times 16\}$.

– **Inter16×16 cas III** : Si $E_n > T_{mc}^{16 \times 16}$, $MCC_{16 \times 16}^{H.264} = \{\text{Inter16} \times 16\}$.

• **Inter8×8**, $MCC_{8 \times 8}^{H.264} = \{\text{Skip}, \text{Inter16} \times 16, \text{Inter16} \times 8, \text{Inter8} \times 16, \text{Inter8} \times 8\}$.

Le processus de détermination du mode de codage H.264 est résumé à la figure 3.6.

3.4 Détermination des vecteurs de mouvement

L'estimation de mouvement est une opération très complexe en calculs dans le transcodage. Afin de diminuer cette complexité, la plupart des algorithmes de transcodage existants récupèrent les VMs MPEG-4 disponibles au décodeur MPEG-4 et les réutilisent dans le codage H.264. En effet les VMs MPEG-4 pourraient être une bonne approximation des VMs H.264, d'autant plus que c'est la même vidéo que l'on recode sous différentes contraintes. Toutefois, les contraintes binaires, la précision inférieure des VMs comparée à ce qui est permis par H.264 font que les VMs MPEG-4 ne sont pas parfaits. On rappelle que les VMs H.264 ont une précision au quart de pixels, tandis que, dans MPEG-4, les VMs sont au demi-pixel pour le profil MPEG-4 qu'on considère, le *visual simple profile*. C'est le profil supporté par la plupart des applications mobiles. Selon plusieurs simulations que nous avons effectuées, la qualité vidéo peut s'améliorer de presque 2 dB, en passant d'une précision au demi-pixel à une précision au quart de pixel.

Pour améliorer l'efficacité des VMs recueillis du décodeur MPEG-4, il est impératif de les raffiner du demi au quart de pixel. Malheureusement, ce raffinement est assez coûteux en termes de calculs. Afin de réduire cette complexité, nous proposons de raffiner uniquement les VMs qui en ont besoin. En effet, il est inutile de raffiner des VMs qui sont déjà efficaces et cela ne ferait qu'augmenter la complexité des calculs. Tout comme dans le cas de la détermination des MCs, nous proposons de nous servir de l'énergie résiduelle pour déterminer si un VM a besoin de raffinement. En effet, l'énergie résiduelle peut servir de mesure d'efficacité de l'estimation du mouvement, tel qu'expliqué précédemment, à la section 3.2. Pour classifier les VMs H.264 d'une partition, l'énergie résiduelle de cette dernière sera d'abord comparée à un seuil. Dans le cas où cette énergie est inférieure au seuil, les VMs MPEG-4 concernés seront

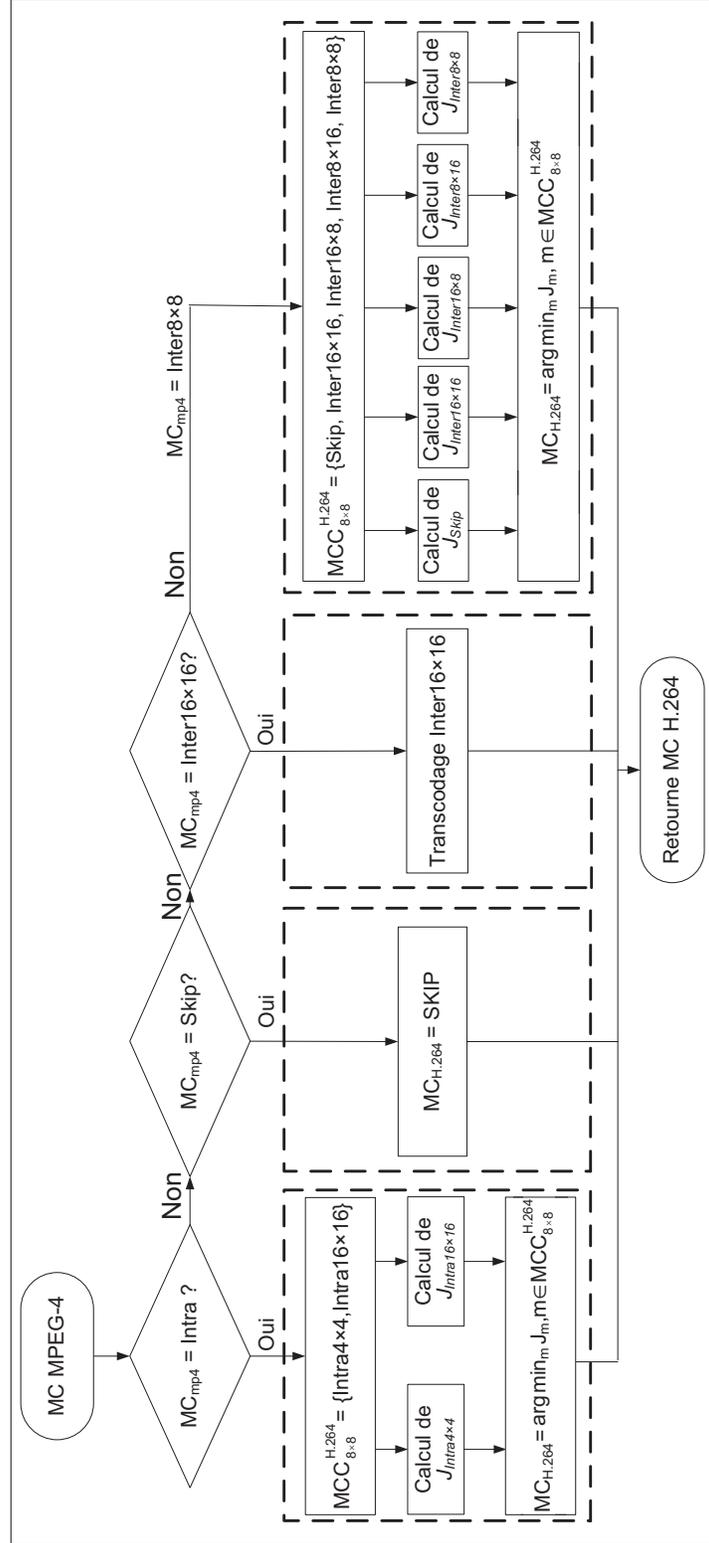


Figure 3.6 Processus de détermination du mode de codage H.264 selon le mode de codage en MPEG-4.

conservés tels quels, autrement, ils seront raffinés. Cette façon de faire nous permettra ainsi d'accélérer considérablement le processus d'estimation de mouvement de H.264. On évite ainsi des raffinements inutiles à la qualité et coûteux en vitesse, contrairement aux algorithmes de l'état de l'art où les VMs sont tous raffinés (Lee et al. (2006), Liang et al. (2007), Lee et Lee (2005)).

Le processus de détermination des VMs H.264, selon le MC MPEG-4 est détaillé dans les sections suivantes. Il est explicité pour chaque mode de codage de l'ensemble des MCs H.264 candidats.

3.4.1 Détermination des vecteurs de mouvement pour un macrobloc codé Intra en MPEG-4

Dans ce cas de transcodage, on considère deux MCs H.264, Intra4×4 et Intra16×16, comme candidats. $MCC_{\text{Intra}}^{\text{H.264}} = \{\text{Intra}4 \times 4, \text{Intra}16 \times 16\}$. Ces deux modes n'utilisent pas de VMs.

3.4.2 Détermination des vecteurs de mouvement pour un macrobloc codé Skip en MPEG-4

Ce mode est transcodé en mode Skip. $MCC_{\text{Skip}}^{\text{H.264}} = \{\text{Skip}\}$. H.264 affecte le vecteur prédit au mode Skip. Ce vecteur, v_p , est déterminé selon les spécifications H.264.

3.4.3 Détermination des vecteurs de mouvement pour un macrobloc codé Inter16×16 en MPEG-4

Ce mode est transcodé soit en Skip, soit en Inter16×16. Dans le cas du Skip, le vecteur affecté est le vecteur prédit et il est déterminé selon les spécifications H.264.

Dans le cas de la détermination du VM Inter16×16, le principe de raffinement est appliqué.

Un seuil $T_{vm}^{16 \times 16}$ est utilisé pour juger l'efficacité du VM MPEG-4. Lorsque l'énergie résiduelle du MB est inférieure au seuil, $E_n \leq T_{vm}^{16 \times 16}$, le VM MPEG-4 est gardé tel quel. Autrement ce dernier est raffiné du demi au quart de pixel. Il est toutefois important que le seuil établi soit fonction du débit afin de maintenir une bonne qualité. En effet, lorsque les contraintes de débit binaire sont plus serrées, le coût lié au MC et aux VMs devient plus important. Des partitions

plus grandes sont alors privilégiées. Cela a pour effet de diminuer la qualité. Par conséquent, le seuil de raffinement des VMs doit être abaissé, lorsque le débit diminue, afin d'augmenter le nombre de VMs à raffiner, et ainsi, ne pas diminuer significativement la qualité.

Un ensemble de séquences tests d'entraînement contenant des vidéos de types différents a été utilisé pour fixer le seuil de raffinement. Suite à de nombreuses simulations, nous avons fixé le seuil de raffinement aux valeurs de 500 et 2000, qui mènent à un bon compromis entre la qualité et la vitesse pour les différents types de vidéos utilisés. Nous avons fixé $T_{vm}^{16 \times 16} = \{2000\}$, pour des débits binaires supérieurs à 64 kb/s pour des vidéos QCIF et à 256 kb/s pour des vidéos CIF. Ce seuil a été ensuite abaissé à 500, $T_{vm}^{16 \times 16} = \{500\}$, pour des débits binaires inférieurs. On rappelle que les séquences d'entraînement choisies sont complètement différentes de celles qui ont servi à la validation des résultats. Ce sont les mêmes que celles qui ont servi à déterminer T_{mc}^{Skip} et $T_{mc}^{16 \times 16}$, les seuils de classifications des MBs, c'est-à-dire les séquences QCIF (Akiyo, Bridge-close, Coastguard) et CIF (Container, Mobile et Waterfall).

La détermination du seuil en tant que fonction du débit binaire pourrait toutefois faire l'objet de recherches plus approfondies. Les valeurs du seuil influencent également énormément les résultats en qualité et en vitesse. Des seuils plus petits ont pour effet d'améliorer la qualité, mais par contre, augmentent la complexité en calculs. L'effet est inversé pour des seuils plus grands. Le seuil devra donc être ajusté selon le critère à privilégier (la qualité ou la vitesse).

Soit $\mathcal{F}(\cdot)$, la fonction de raffinement d'un VM; \mathbf{v}_{mp4} , le vecteur MPEG-4; $\mathbf{v}_{16 \times 16}$, le VM du MC candidat Inter16×16; \mathbf{v}_{Skip} , le VM du MC candidat Skip et \mathbf{v}_p le vecteur prédit. Ce dernier est déterminé selon les spécifications du standard H.264 et est le VM affecté au MC Skip en H.264. La détermination du VM H.264 selon le MC candidat est résumée comme suit :

- $\mathbf{v}_{16 \times 16} = \begin{cases} \mathcal{F}(\mathbf{v}_{mp4}) & \text{si } E_n \leq T_{vm}^{16 \times 16} \\ \mathbf{v}_{mp4} & \text{si } E_n > T_{vm}^{16 \times 16} \end{cases}$
- $\mathbf{v}_{Skip} = \mathbf{v}_p$

La détermination des VMs pour un MC d'entrée MPEG-4 est illustrée à la figure 3.7. Les VMs sont raffinés en utilisant un algorithme rapide de raffinement utilisé dans le codeur MPEG-4 de Intel (Bhaskaran et Konstantinides (1997)). Selon cette méthode, cinq positions au quart de pixel sont évaluées au lieu des huit positions. Cet algorithme est décrit dans la figure 3.8. b_i représente la SAD des pixels à la position p_i . Les valeurs de j et de i résultantes représentent les déplacements horizontaux et verticaux à appliquer à la position p_0 pour obtenir la position avec une précision au quart de pixels.

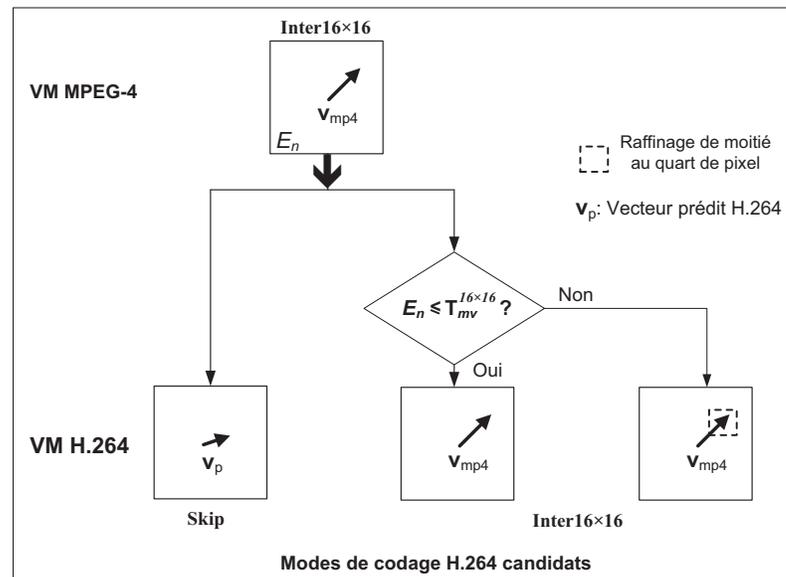


Figure 3.7 Processus de détermination des VMs H.264 pour des MBs codés Inter16x16 en MPEG-4.

3.4.4 Détermination des vecteurs de mouvement pour un macrobloc codé Inter8x8 en MPEG-4

Nous présentons, dans cette section, le processus de détermination des VMs H.264 pour chaque mode de codage candidat pour des MBs qui ont été codés Inter8x8 en MPEG-4. $MCC_{8 \times 8}^{H.264} = \{\text{Inter16} \times 16, \text{Inter16} \times 8, \text{Inter8} \times 16, \text{Inter8} \times 8\}$. Soit $v_{16 \times 16}$, le VM H.264 associé au MC Inter16x16. Soit également $v_{16 \times 8, k}$, $v_{8 \times 16, k}$ et $v_{8 \times 8, k}$, les VMs H.264 associés à la partition k respectivement des MCs Inter16x8, Inter8x16 et Inter8x8.

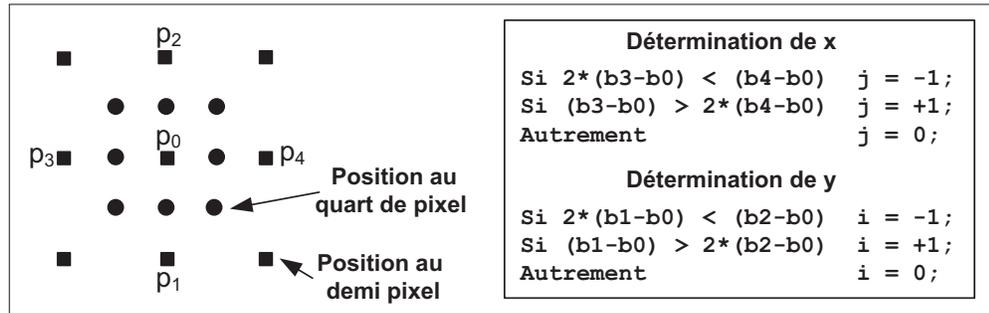


Figure 3.8 Algorithme de raffinement des VMs du demi au quart de pixel d'Intel (Bhaskaran et Konstantinides (1997)).

Puisque le MC a été codé en Inter8×8 en MPEG-4, nous avons quatre VMs candidats \mathbf{v}_0 , \mathbf{v}_1 , \mathbf{v}_2 et \mathbf{v}_3 , les VMs de chaque bloc k 8×8 du MB, k allant de 0 à 3. Tout comme pour le cas Inter16×16, on tirera encore une fois avantage des propriétés du résiduel. Les VMs, seront raffinés uniquement si c'est jugé nécessaire. Cela nous évitera des opérations de raffinement inutiles à la qualité et coûteuses en calculs.

Pour déterminer les VMs des MCs Inter16×16, Inter16×8 et Inter8×16, nous n'appliquerons, par contre, pas le principe de raffinement selon l'énergie résiduelle, puisque le partitionnement de ces MCs est plus grand que 8×8. Le fait que l'énergie résiduelle soit basse, $E_n < \text{Thr}$, signifie que le VM représentait assez bien chaque partition du MC MPEG-4. Par contre, cela ne signifie pas nécessairement que ces VMs seront aussi efficaces pour un mode de codage H.264 différent. Nous déterminons d'abord, pour chaque partition, les VMs candidats tel qu'indiqué, dans la figure 3.9. Une fois les VMs candidats identifiés pour chaque partition du MC, on évalue, au demi-pixel pour chaque VM, le coût qui lui est lié, $J_{\text{motion}}(\mathbf{v}_i)$. $J_{\text{motion}}(\mathbf{v}_i)$ représente le coût de la SAD et du nombre de bits nécessaires pour coder le VM (voir la section 1.8.1 pour plus de détails). Le VM choisi est celui qui présente le coût minimal. Finalement, ce VM choisi est raffiné du demi au quart de pixel.

Dans le cas du MC Inter8×8, on tirera avantage des propriétés du résiduel pour le raffinement des VMs. Les vecteurs de mouvement associés au bloc k sont déterminés comme suit. Pour chaque bloc k , le VM MPEG-4 de la partition k , \mathbf{v}_k , est raffiné quand l'énergie résiduelle est inférieure à un seuil $T_{vm}^{8 \times 8}$. Autrement, \mathbf{v}_k est conservé tel quel.

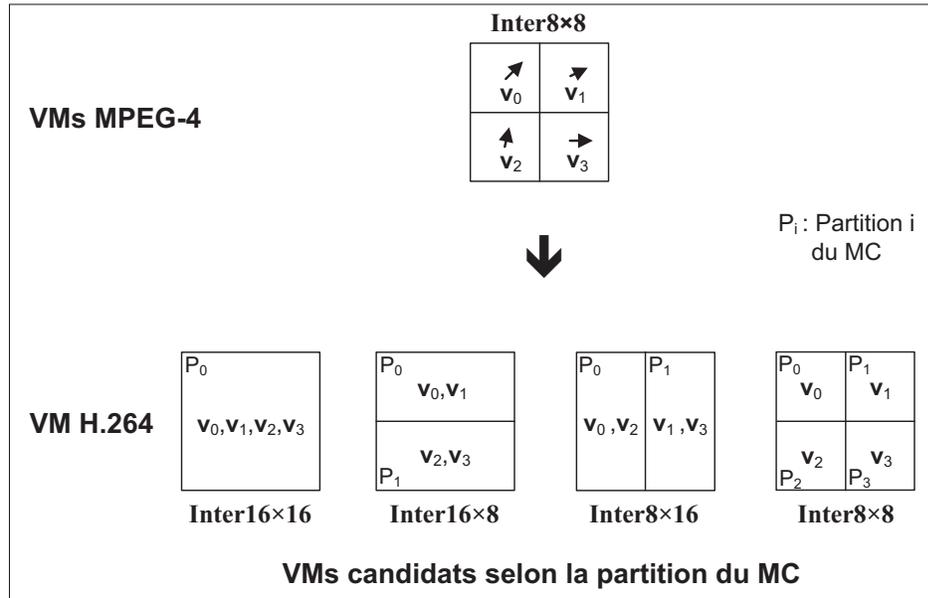


Figure 3.9 VMs MPEG-4 candidats selon le MC H.264 dans le cas d'un MB codé Inter8x8 en MPEG-4.

Le seuil $T_{vm}^{8 \times 8}$ a été déterminé de la même façon que dans le cas de l'Inter16x16, c'est-à-dire après des simulations sur différents types de séquences vidéos de sorte à obtenir un bon compromis vitesse/qualité. Les mêmes séquences d'entraînement que précédemment ont été utilisées. La valeur de $T_{vm}^{8 \times 8}$ a également été réduite pour des débits binaires faibles afin d'améliorer la qualité. $T_{vm}^{8 \times 8}$ a été fixé à 62.5, pour des débits inférieurs à 64 kb/s et 256 kb/s respectivement pour des vidéos QCIF et CIF. Pour des débits supérieurs, $T_{vm}^{8 \times 8}$ a été fixé à 125. Récapitulons le processus de détermination des VMs pour les MBs codés Inter8x8 en MPEG-4 selon le MC candidat :

- **Inter16x16**, $\mathbf{v}_{16 \times 16} = \mathcal{F}(\arg \min_{\mathbf{v}} J(\mathbf{v})), \mathbf{v} \in \{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$
- **Inter16x8**, $\mathbf{v}_{16 \times 8, k} = \mathcal{F}(\arg \min_{\mathbf{v}} J(\mathbf{v})), \mathbf{v} \in \{\mathbf{v}_{2k}, \mathbf{v}_{2k+1}\}, k \in [0, 1]$.
- **Inter8x16**, $\mathbf{v}_{8 \times 16, k} = \mathcal{F}(\arg \min_{\mathbf{v}} J(\mathbf{v})), \mathbf{v} \in \{\mathbf{v}_k, \mathbf{v}_{k+2}\}, k \in [0, 1]$.
- **Inter8x8**, $\mathbf{v}_{8 \times 8, k} = \begin{cases} \mathcal{F}(\mathbf{v}_k) & \text{si } E_{n,k} > T_{vm}^{8 \times 8} \\ \mathbf{v}_k & \text{si } E_{n,k} \leq T_{vm}^{8 \times 8} \end{cases}, \forall k \in [0, 3]$.

La détermination des VMs pour un MC d'entrée MPEG-4 est illustrée à la figure 3.10.

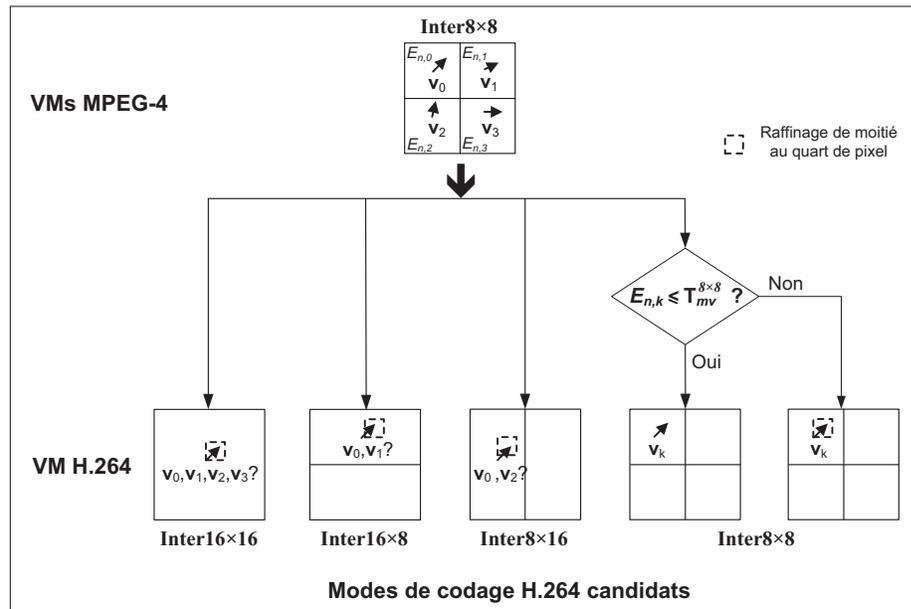


Figure 3.10 Processus de détermination des VMs H.264 pour des MBs codés Inter8x8 en MPEG-4.

Une fois les VMs déterminés pour chaque MC candidat, le coût associé au MC, J_{MC} , est évalué pour chaque MC candidat. Le mode de codage est finalement choisi en retenant celui dont le J_{MC} est minimal (voir la section 1.8 et la sous-section 1.8.2 pour plus de détails sur le calcul du coût lié au MC et à son choix).

3.5 Simulations et résultats

Des simulations ont été menées afin de tester notre algorithme en terme de qualité, de vitesse et de fréquence de trames transcodées. L'algorithme proposé a été comparé aux plus performants de l'état de l'art, notamment, ceux décrits dans les références Liang et al. (2007) et Lee et al. (2006). Les algorithmes ont été implémentés sous une plateforme HP *notebook* G62 avec un processeur Intel i5-430M à 2 coeurs/4 fils d'exécution (*thread*) à 2.53GHz avec une mémoire vive de 4Go. Les codecs utilisés pour l'implémentation sont les codecs IPP (Integrated Performances Primitives, Intel (2008)) de Intel, notamment avec sa version 5.3. Les codecs vidéos de Intel sont nettement optimisés en terme de vitesse comparativement aux codecs de référence de MPEG-4 et de H.264 (Momusys, ISO/IEC (2001) et JM, ITU (2003)). Bien que

le JM, le codec de référence d'H.264, soit une excellente référence pour valider les résultats en qualité, il n'est pas optimisé pour la vitesse. Il n'est, par conséquent, pas fiable pour mesurer les résultats en vitesse. Les résultats obtenus sur les codecs d'Intel sont plus représentatifs que ceux qui seraient obtenus sur des codecs commerciaux.

Les méthodes suivantes ont été comparées : la méthode proposée (ResidualM) (Metoevi et Coulombe (2009c)), la méthode en cascade (cascade), la méthode sélection de mode et raffinement (MV+MS, (*motion vector refinement with selection mode*), Liang et al. (2007)), la méthode statistique avec raffinement (STAT) (Lee et al. (2006)) ou sans raffinement (STAT+REF) (Lee et al. (2006)). La comparaison a porté sur la qualité (en PSNR), la vitesse d'accélération du temps de transcodage, $\frac{T_c}{T_M}$ (*Speed-up*), et la fréquence des trames codées, $\frac{N_{Trames}}{T_M}$ (F.T.). T_M , T_c et N_{Trames} sont respectivement les temps de transcodage de la méthode à tester, le temps de transcodage de la méthode en cascade et le nombre de trames à transcoder. La méthode en cascade nous servira de référence. On rappelle qu'il s'agit de la méthode de transcodage complète où le signal MPEG-4 est complètement décodé puis réencodé, selon les spécifications H.264, sans utilisation d'information autre que les valeurs de pixels décodés.

Différents types de vidéos, de résolutions ainsi que de débits binaires, ont été utilisés pour valider nos résultats. Notamment, des vidéos à mouvements complexes ou peu complexes, avec peu de détails ou beaucoup de détails. Plusieurs résolutions ont été également testées, QCIF, CIF, 4CIF, 1280×720, HD, 1920×1080. Les conditions sous lesquelles les simulations ont été effectuées sont recensées dans la table 3.3.

Les séquences vidéos ont été codées en MPEG-4 aux débits suivants : 200 kb/s, 720 kb/s, 2,88 Mb/s, 7,2 Mb/s et 16,4 Mb/s respectivement pour QCIF, CIF, 4-CIF, HD à 1280×720 et HD à 1920×1080. D'autres débits binaires d'encodage ont également été testés, avec très peu de différences sur le plan des résultats. Les résultats obtenus sont recensés dans les figures 3.11, 3.12 et les tables 3.4, 3.5, 3.6, 3.7 et 3.8.

Tableau 3.3 Conditions de simulation utilisées

Éléments de codage	MPEG-4	H.264
Codage entropique	CAVLC	CAVLC
Fréquence de trame (Hz)	30	30
Taux de rafraîchissement (s)	3,33	3,33
Algorithme de recherche des VMs	logarithme	EPZS
Précision des VMs (en pixel)	1/2	1/4
Algorithme de raffinement des VMs	logarithme	algorithme (Bhaskaran et Konstantinides (1997))
Trames de type B	Non	Non
Nombre de trames de référence	1	1
Profil	VSP	baseline
<i>Deblocking filter</i>	n'existe pas	non
Méthode de mesure de distorsion pour recherche de VMs	SAD	SATD

Plus de résultats sont montrés dans le chapitre 5. Dans ce chapitre, sont présentés les résultats d'un second algorithme développé. On pourra alors faire une comparaison entre ces deux algorithmes et ceux de l'état de l'art.

Les résultats obtenus sont assez impressionnants. L'algorithme proposé est en moyenne, de 3 à 5 fois plus rapide que la méthode en cascade, pour une perte moyenne en qualité de seulement 0,36 dB pour du QCIF et 0,64 dB pour du CIF. Comparativement aux autres méthodes, nous observons que c'est uniquement l'approche statistique sans raffinement qui s'avère être plus rapide que la méthode proposée. Par contre, la perte en qualité est beaucoup plus élevée (en moyenne 1,7 dB pour du QCIF et 2,47dB pour du CIF). De telles pertes en qualité sont inacceptables dans beaucoup d'applications. Comparativement aux méthodes STAT+REF et MV+MS, nous obtenons des gains en vitesse plus élevés pour des qualités supérieures. L'algorithme proposé est d'environ 30% plus rapide que le MV+MS pour une qualité meilleure de 0,5 dB en moyenne pour du QCIF et de 1 dB pour du CIF. Les différences en qualité sont particulièrement notables à bas débits.

On constate également que l'algorithme proposé se comporte moins bien à haut qu'à bas débit. Cela pourrait être attribuable au fait de ne pas considérer de modes de codage à petites partitions dans le cas du transcodage des MBs d'entrée Inter16×16. Les petites partitions s'adaptent davantage plus aux débits élevés et les MBs d'entrée Inter16×16 représentent une bonne majorité des MBs (64%). Les résultats en qualité pour ces MBs ont donc un impact plus important sur les résultats que les MBs d'entrée Inter8×8. Il faut aussi noter que le fait que l'algorithme se comporte moins bien à haut qu'à bas débit est accentué pour des vidéos à mouvements lents et peu complexes. Une autre raison serait l'utilisation de seuils fixes pour la classification des MBs et le raffinement des VMs. L'utilisation de seuils s'adaptant au type de vidéo et au débit binaire devrait donner de meilleurs résultats.

Il faut noter que les résultats présentés chez Liang et al. (2007) et Lee et al. (2006) ont été obtenus avec les codecs de référence Momusys et JM. On peut constater que, sous un codec plus optimisé comme celui d'Intel, les résultats étaient nettement moins impressionnants (2,6 en gain de vitesse au lieu de 10,36 par exemple pour l'algorithme présenté chez Liang et al. (2007)).

Tableau 3.4 Résultats de vidéos QCIF en PSNR, en accélération (*speed up*) et en fréquence de trames (F.T.) pour les méthodes Cascade, ResidualM, STAT, STAT+REF et MV+MS. Les résultats en PSNR montrent les différences en qualité avec la méthode en cascade. Les vidéos ont été codées en MPEG-4, au débit de 200 kb/s

QCIF		32 kb/s					64 kb/s					96 kb/s				
		Cas-cade	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	ResidualM	STAT+REF	STAT	MV+MS
Carphone	PSNR (dB)	30,44	-0,68	-2,08	-3,96	-5,90	33,60	-0,37	-0,84	-2,76	-2,11	35,39	-0,43	-0,51	-2,32	-1,19
	SPEED UP	1,00	3,23	2,00	3,20	2,94	1,00	3,49	2,17	3,70	2,85	1,00	3,85	2,29	4,06	2,90
	F.T.	31,57	101,87	63,25	101,06	92,94	24,24	84,51	52,54	89,67	69,08	20,58	79,25	47,04	83,59	59,78
Foreman	PSNR (dB)	29,27	-0,64	-1,95	-3,81	-6,93	32,86	-0,26	-0,78	-2,85	-2,69	34,76	-0,37	-0,52	-2,53	-1,37
	SPEED UP	1,00	3,23	2,09	3,33	2,83	1,00	3,48	2,22	3,89	2,87	1,00	3,76	2,26	4,13	2,81
	F.T.	29,79	96,15	62,24	99,34	84,27	22,56	78,53	50,00	87,72	64,66	19,44	73,17	43,86	80,21	54,55
Hall	PSNR (dB)	34,33	-0,22	-0,78	-1,84	-0,29	38,16	-0,15	-0,26	-1,06	-0,08	40,02	-0,21	-0,22	-0,81	-0,09
	SPEED UP	1,00	3,69	1,70	2,81	3,37	1,00	4,28	2,16	3,56	3,68	1,00	4,80	2,54	4,14	3,80
	F.T.	38,36	141,51	65,36	107,91	129,31	28,49	121,95	61,60	101,35	104,90	23,15	111,11	58,82	95,85	87,98
Miss-America	PSNR (dB)	40,24	-0,55	-1,02	-2,67	-0,89	43,15	-0,51	-0,45	-1,83	-0,39	44,58	-0,73	-0,32	-1,53	-0,34
	SPEED UP	1,00	3,51	1,78	2,85	2,56	1,00	3,91	2,06	3,40	2,60	1,00	4,51	2,29	3,84	2,70
	F.T.	37,13	130,43	66,08	105,63	94,94	28,85	112,78	59,29	98,04	75,00	24,43	110,29	55,97	93,75	66,08
News	PSNR (dB)	31,93	-0,41	-0,66	-2,39	-1,47	35,98	-0,30	-0,46	-2,13	-0,51	38,22	-0,42	-0,46	-1,74	-0,36
	SPEED UP	1,00	3,73	2,43	3,74	3,45	1,00	4,21	2,77	4,44	3,74	1,00	4,62	2,99	4,85	3,93
	F.T.	34,25	127,66	83,10	128,21	118,11	26,39	111,11	73,17	117,19	98,68	22,73	104,90	67,87	110,29	89,29
Salesman	PSNR (dB)	33,39	-0,06	-0,15	-1,65	-0,57	36,89	-0,11	-0,09	-1,25	-0,06	38,90	-0,19	-0,02	-1,02	0,13
	SPEED UP	1,00	4,02	2,39	3,85	3,60	1,00	4,46	2,64	4,43	3,80	1,00	4,93	2,91	4,82	4,02
	F.T.	31,31	125,77	74,83	120,70	112,81	25,04	111,69	66,13	110,86	95,13	21,68	106,90	63,15	104,42	87,18
Moyenne PSNR (dB)		33,27	-0,43	-1,11	-2,72	-2,68	36,77	-0,28	-0,48	-1,98	-0,97	38,65	-0,39	-0,34	-1,66	-0,54
Moyenne speed up		1,00	3,57	2,07	3,30	3,13	1,00	3,97	2,34	3,90	3,26	1,00	4,41	2,55	4,31	3,36
Moyenne F.T.		33,74	120,57	69,14	110,47	105,40	25,93	103,43	60,46	100,81	84,57	22,00	97,60	56,12	94,69	74,14

QCIF		128 kb/s					160 kb/s					190 kb/s				
		Cas-cade	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	ResidualM	STAT+REF	STAT	MV+MS
Carphone	PSNR (dB)	36,68	-0,40	-0,39	-2,00	-0,76	37,71	-0,40	-0,31	-1,84	-0,58	38,57	-0,39	-0,27	-1,66	-0,42
	SPEED UP	1,00	4,02	2,40	4,30	2,97	1,00	4,14	2,49	4,48	3,03	1,00	4,25	2,56	4,65	3,13
	F.T.	18,40	74,03	44,16	79,09	54,57	16,82	69,58	41,84	75,35	50,93	15,59	66,20	39,96	72,49	48,72
Foreman	PSNR (dB)	36,00	-0,32	-0,37	-2,13	-0,85	36,95	-0,31	-0,29	-1,86	-0,56	37,75	-0,30	-0,21	-1,69	-0,39
	SPEED UP	1,00	3,90	2,35	4,35	2,92	1,00	4,02	2,41	4,50	2,97	1,00	4,13	2,47	4,60	3,04
	F.T.	17,40	67,87	40,82	75,76	50,85	16,04	64,52	38,71	72,12	47,69	14,93	61,60	36,86	68,65	45,39
Hall	PSNR (dB)	41,23	-0,23	-0,14	-0,63	-0,08	42,22	-0,22	-0,13	-0,58	-0,04	43,15	-0,29	-0,13	-0,64	-0,07
	SPEED UP	1,00	5,16	2,85	4,59	3,85	1,00	5,46	3,12	4,94	3,98	1,00	5,59	3,23	5,11	3,94
	F.T.	19,70	101,69	56,18	90,36	75,76	17,40	94,94	54,25	85,96	69,28	16,26	90,91	52,45	83,10	64,10
Miss-America	PSNR (dB)	45,52	-0,75	-0,24	-1,35	-0,28	46,26	-0,76	-0,23	-1,21	-0,24	46,93	-0,81	-0,29	-1,25	-0,29
	SPEED UP	1,00	4,83	2,49	4,14	2,79	1,00	5,09	2,64	4,43	2,87	1,00	5,44	2,84	4,81	3,06
	F.T.	21,43	103,45	53,38	88,76	59,76	19,26	98,04	50,85	85,23	55,35	17,24	93,75	49,02	82,87	52,82
News	PSNR (dB)	39,68	-0,30	-0,30	-1,34	-0,15	41,01	-0,30	-0,25	-1,22	-0,08	42,05	-0,25	-0,23	-1,05	-0,10
	SPEED UP	1,00	4,83	3,15	5,08	4,08	1,00	4,99	3,31	5,27	4,21	1,00	5,16	3,48	5,47	4,36
	F.T.	20,51	99,01	64,66	104,17	83,57	18,74	93,46	61,98	98,68	78,95	17,29	89,29	60,12	94,64	75,38
Salesman	PSNR (dB)	40,44	-0,12	-0,02	-0,93	0,08	41,82	-0,21	-0,07	-0,89	0,07	42,82	-0,16	-0,09	-0,72	0,10
	SPEED UP	1,00	5,23	3,14	5,16	4,22	1,00	5,33	3,31	5,37	4,36	1,00	5,58	3,47	5,57	4,50
	F.T.	19,25	100,67	60,51	99,34	81,34	17,68	94,33	58,46	94,93	77,15	16,25	90,71	56,41	90,52	73,13
Moyenne PSNR (dB)		39,93	-0,35	-0,24	-1,40	-0,34	41,00	-0,37	-0,21	-1,27	-0,24	41,88	-0,37	-0,20	-1,17	-0,19
Moyenne speed up		1,00	4,66	2,73	4,60	3,47	1,00	4,84	2,88	4,83	3,57	1,00	5,02	3,01	5,03	3,67
Moyenne F.T.		19,45	91,12	53,28	89,58	67,64	17,66	85,81	51,02	85,38	63,23	16,26	82,08	49,13	82,05	59,92

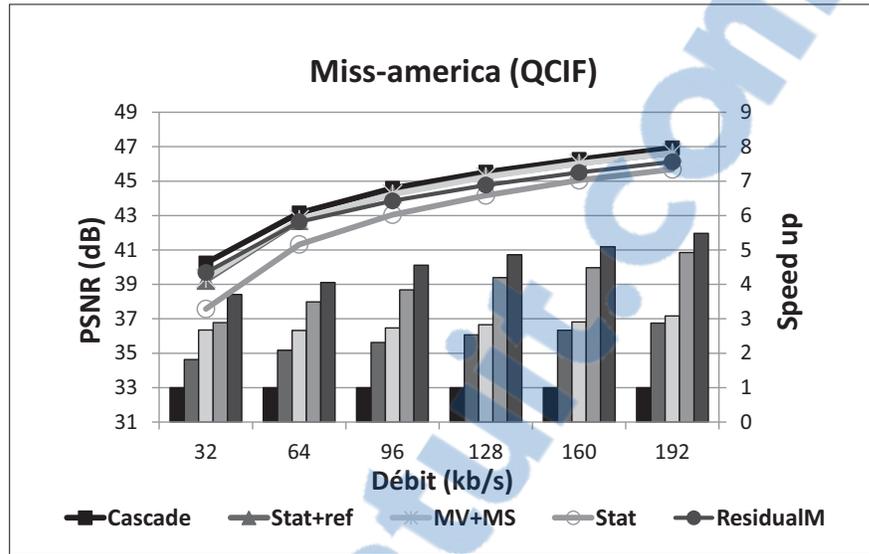


Figure 3.11 Résultats en PSNR (courbes) et en vitesse d'accélération (barres) de la méthode proposée, comparée aux autres méthodes. Cette accélération est définie comme T_C/T_M . Ces résultats sont ceux de la vidéo Miss America (QCIF). Le nom des méthodes est dans le même ordre que les barres, de gauche à droite.

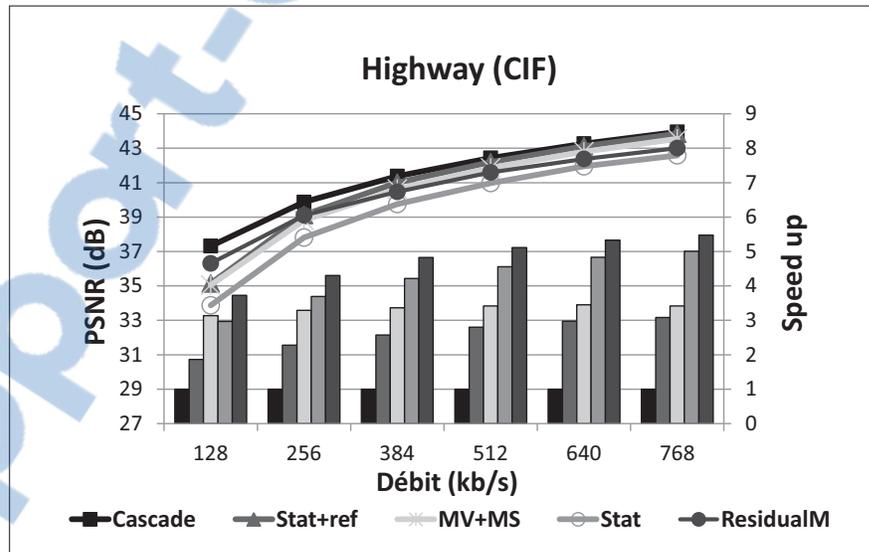


Figure 3.12 Résultats en PSNR (courbes) et en vitesse d'accélération (barres) de la méthode proposée, comparée aux autres méthodes. Cette accélération est définie comme T_C/T_M . Ces résultats sont ceux de la vidéo Highway (CIF). Le nom des méthodes est dans le même ordre que les barres, de gauche à droite.

Tableau 3.5 Résultats de vidéos CIF en PSNR, en accélération (*speed up*) et en fréquence de trames (F.T.) pour les méthodes Cascade, ResidualM, STAT, STAT+REF et MV+MS. Les résultats en PSNR montrent les différences en qualité avec la méthode en cascade. Les vidéos ont été codées en MPEG-4, au débit de 720 kb/s

CIF		128kb/s					256kb/s					384kb/s				
		Cas-cade	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	ResidualM	STAT+REF	STAT	MV+MS
Bus	PSNR (dB)	25,55	-0,33	-1,25	-3,90	-5,69	28,65	-0,13	-0,56	-3,01	-2,05	30,37	-0,13	-0,33	-2,51	-1,06
	SPEED UP	1,00	3,27	1,94	3,14	3,02	1,00	3,60	2,18	3,69	3,07	1,00	3,89	2,34	4,09	3,15
	F.T.	7,33	24,00	14,23	23,01	22,16	5,57	20,05	12,12	20,58	17,08	4,69	18,27	10,99	19,18	14,76
Football	PSNR (dB)	26,68	-2,50	-4,59	-5,73	-5,11	29,63	-1,63	-2,67	-4,04	-5,02	31,42	-1,32	-1,89	-3,17	-3,22
	SPEED UP	1,00	3,19	2,08	3,24	2,63	1,00	3,53	2,28	3,73	3,31	1,00	3,70	2,38	3,96	3,43
	F.T.	7,53	24,01	15,67	24,34	19,82	5,53	19,55	12,63	20,63	18,31	4,63	17,13	11,02	18,36	15,88
Foreman	PSNR (dB)	32,03	-0,85	-2,28	-4,22	-6,06	35,13	-0,48	-0,95	-2,74	-1,99	36,75	-0,60	-0,59	-2,21	-1,19
	SPEED UP	1,00	3,46	2,03	3,27	3,06	1,00	3,76	2,23	3,84	2,99	1,00	4,21	2,41	4,22	3,03
	F.T.	7,54	26,11	15,28	24,69	23,04	5,75	21,65	12,81	22,06	17,20	4,85	20,44	11,68	20,49	14,72
Highway	PSNR (dB)	37,32	-1,01	-2,18	-3,45	-2,27	39,87	-0,76	-0,75	-2,06	-1,05	41,38	-0,91	-0,38	-1,62	-0,75
	SPEED UP	1,00	3,60	1,82	2,87	3,02	1,00	4,16	2,23	3,58	3,18	1,00	4,72	2,53	4,11	3,28
	F.T.	9,53	34,34	17,31	27,39	28,74	7,08	29,48	15,78	25,37	22,54	5,83	27,51	14,78	24,00	19,14
Tempete	PSNR (dB)	27,00	-0,52	-1,23	-4,44	-5,01	29,66	-0,28	-0,47	-3,41	-1,68	31,09	-0,25	-0,28	-2,74	-0,95
	SPEED UP	1,00	3,52	2,17	3,48	2,98	1,00	3,80	2,39	4,08	2,84	1,00	4,03	2,52	4,42	2,87
	F.T.	7,22	25,44	15,70	25,15	21,52	5,54	21,07	13,25	22,61	15,72	4,76	19,20	11,98	21,05	13,68
M-daughter	PSNR (dB)	38,96	-0,33	-0,83	-2,07	-1,65	41,95	-0,34	-0,38	-1,79	-0,60	43,42	-0,77	-0,24	-1,62	-0,36
	SPEED UP	1,00	3,90	2,27	3,60	3,06	1,00	4,31	2,49	4,20	3,02	1,00	5,08	2,70	4,65	3,14
	F.T.	7,92	30,93	17,96	28,52	24,21	6,19	26,69	15,43	26,04	18,74	5,29	26,88	14,27	24,59	16,61
Moyenne PSNR (dB)		31,26	-0,92	-2,06	-3,97	-4,30	34,15	-0,60	-0,96	-2,84	-2,07	35,74	-0,66	-0,62	-2,31	-1,26
Moyenne speed up		1,00	3,49	2,05	3,27	2,96	1,00	3,86	2,30	3,85	3,07	1,00	4,27	2,48	4,24	3,15
Moyenne F.T.		7,85	27,47	16,03	25,52	23,25	5,94	23,08	13,67	22,88	18,27	5,01	21,57	12,45	21,28	15,80

CIF		512kb/s					640kb/s					768kb/s				
		Cas-cade	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	ResidualM	STAT+REF	STAT	MV+MS
Bus	PSNR (dB)	31,61	-0,10	-0,24	-2,16	-0,70	32,64	-0,15	-0,20	-1,89	-0,51	33,52	-0,15	-0,15	-1,73	-0,39
	SPEED UP	1,00	4,05	2,47	4,32	3,14	1,00	4,23	2,56	4,56	3,25	1,00	4,37	2,69	4,74	3,36
	F.T.	4,20	17,01	10,37	18,16	13,19	3,78	15,99	9,70	17,24	12,27	3,50	15,31	9,41	16,59	11,76
Football	PSNR (dB)	32,79	-1,11	-1,53	-2,70	-2,40	33,92	-1,00	-1,28	-2,42	-1,90	34,92	-0,93	-1,12	-2,29	-1,65
	SPEED UP	1,00	3,82	2,50	4,20	3,56	1,00	3,92	2,60	4,36	3,61	1,00	4,02	2,70	4,50	3,67
	F.T.	4,05	15,49	10,14	17,00	14,40	3,66	14,36	9,53	15,97	13,22	3,38	13,57	9,13	15,20	12,40
Foreman	PSNR (dB)	37,89	-0,60	-0,45	-1,86	-0,84	38,77	-0,58	-0,34	-1,64	-0,62	39,54	-0,57	-0,29	-1,56	-0,49
	SPEED UP	1,00	4,39	2,54	4,49	3,04	1,00	4,57	2,65	4,71	3,18	1,00	4,73	2,76	4,91	3,26
	F.T.	4,32	18,96	10,95	19,39	13,12	3,93	17,94	10,38	18,48	12,47	3,62	17,12	10,00	17,77	11,79
Highway	PSNR (dB)	42,44	-0,84	-0,25	-1,46	-0,58	43,27	-0,89	-0,15	-1,32	-0,47	43,95	-0,94	-0,11	-1,36	-0,40
	SPEED UP	1,00	5,02	2,77	4,48	3,34	1,00	5,24	2,94	4,74	3,39	1,00	5,40	3,06	4,93	3,38
	F.T.	5,10	25,59	14,13	22,85	17,05	4,60	24,12	13,54	21,84	15,61	4,24	22,89	12,99	20,88	14,35
Tempete	PSNR (dB)	32,15	-0,23	-0,20	-2,29	-0,68	33,02	-0,22	-0,17	-1,97	-0,52	33,79	-0,23	-0,13	-1,72	-0,44
	SPEED UP	1,00	4,22	2,62	4,69	2,93	1,00	4,38	2,70	4,88	2,99	1,00	4,55	2,80	5,09	3,08
	F.T.	4,25	17,93	11,13	19,97	12,45	3,90	17,08	10,53	19,03	11,69	3,60	16,38	10,07	18,34	11,08
M-daughter	PSNR (dB)	44,47	-0,77	-0,19	-1,49	-0,26	45,32	-0,74	-0,16	-1,35	-0,23	46,01	-0,69	-0,14	-1,24	-0,17
	SPEED UP	1,00	5,45	2,91	5,09	3,34	1,00	5,80	3,11	5,44	3,54	1,00	6,11	3,28	5,80	3,72
	F.T.	4,61	25,13	13,41	23,44	15,38	4,11	23,85	12,77	22,35	14,53	3,71	22,64	12,15	21,51	13,80
Moyenne PSNR (dB)		36,89	-0,61	-0,48	-1,99	-0,91	37,82	-0,60	-0,38	-1,77	-0,71	38,62	-0,59	-0,32	-1,65	-0,59
Moyenne speed up		1,00	4,49	2,63	4,54	3,22	1,00	4,69	2,76	4,78	3,33	1,00	4,86	2,88	5,00	3,41
Moyenne F.T.		4,42	20,02	11,69	20,14	14,27	4,00	18,89	11,08	19,15	13,30	3,67	17,99	10,63	18,38	12,53

Tableau 3.6 Résultats de vidéos 4CIF, 720×576 en PSNR, en accélération (*speed up*) et en fréquence de trames (F.T.) pour les méthodes Cascade, ResidualM, STAT, STAT+REF et MV+MS. Les résultats en PSNR montrent les différences en qualité avec la méthode en cascade. Les vidéos ont été codées en MPEG-4, au débit de 2,88 Mb/s

4CIF, 704×576		512kb/s					1,02Mb/s					1,53 Mb/s				
		Cas-cade	Resi-dualM	STAT +REF	STAT	MV+MS	Cas-cade	Resi-dualM	STAT +REF	STAT	MV+MS	Cas-cade	Resi-dualM	STAT +REF	STAT	MV+MS
City	PSNR (dB)	32,15	-0,28	-1,04	-3,66	-2,92	34,48	-0,13	-0,41	-2,62	-0,98	35,78	-0,32	-0,20	-2,14	-0,60
	SPEED UP	1,00	3,70	2,06	3,37	3,15	1,00	4,13	2,38	4,06	2,95	1,00	4,71	2,58	4,56	2,92
	F.T.	1,79	6,62	3,67	6,02	5,62	1,36	5,62	3,24	5,53	4,02	1,14	5,38	2,95	5,21	3,34
Crew	PSNR (dB)	32,50	-1,68	-3,94	-4,23	-6,13	35,75	-1,13	-1,73	-2,27	-3,18	37,60	-0,89	-1,15	-1,74	-2,07
	SPEED UP	1,00	2,88	2,09	3,02	2,95	1,00	3,27	2,38	3,45	3,10	1,00	3,56	2,51	3,72	3,11
	F.T.	1,86	5,36	3,90	5,61	5,49	1,33	4,35	3,17	4,60	4,13	1,11	3,95	2,78	4,13	3,45
Harbour	PSNR (dB)	28,22	-0,56	-1,59	-4,38	-7,15	30,85	-0,26	-0,68	-3,06	-3,20	32,54	-0,25	-0,44	-2,51	-1,98
	SPEED UP	1,00	3,45	2,28	3,73	3,02	1,00	3,89	2,58	4,52	3,31	1,00	4,09	2,67	4,86	3,24
	F.T.	1,60	5,54	3,66	5,99	4,85	1,18	4,58	3,04	5,31	3,89	1,01	4,11	2,68	4,89	3,26
Ice	PSNR (dB)	37,86	-1,19	-2,36	-3,92	-3,90	41,18	-0,93	-1,35	-2,53	-1,75	42,86	-0,83	-0,97	-1,90	-1,17
	SPEED UP	1,00	3,06	1,76	2,81	2,85	1,00	3,42	2,04	3,29	3,06	1,00	3,83	2,25	3,67	3,23
	F.T.	2,50	7,67	4,40	7,02	7,13	1,93	6,62	3,94	6,35	5,92	1,62	6,22	3,65	5,95	5,25
Soccer	PSNR (dB)	31,88	-1,85	-3,39	-4,33	-5,65	34,74	-1,44	-2,02	-2,84	-2,98	36,40	-1,22	-1,45	-2,27	-2,13
	SPEED UP	1,00	3,05	1,87	2,88	3,01	1,00	3,54	2,27	3,50	3,33	1,00	3,92	2,51	3,92	3,36
	F.T.	2,01	6,13	3,75	5,79	6,04	1,40	4,96	3,18	4,90	4,66	1,14	4,49	2,87	4,49	3,84
Moyenne PSNR (dB)		32,52	-1,11	-2,46	-4,10	-5,15	35,40	-0,78	-1,24	-2,66	-2,42	37,04	-0,70	-0,84	-2,11	-1,59
Moyenne speed up		1,00	3,23	2,01	3,16	2,99	1,00	3,65	2,33	3,76	3,15	1,00	4,02	2,50	4,15	3,17
Moyenne F.T.		1,95	6,26	3,88	6,09	5,83	1,44	5,22	3,31	5,34	4,52	1,21	4,83	2,99	4,93	3,83

4CIF, 704×576		2,04Mb/s					2,56Mb/s					3,07 Mb/s				
		Cas-cade	Resi-dualM	STAT +REF	STAT	MV+MS	Cas-cade	Resi-dualM	STAT +REF	STAT	MV+MS	Cas-cade	Resi-dualM	STAT +REF	STAT	MV+MS
City	PSNR (dB)	36,74	-0,33	-0,12	-1,74	-0,40	37,55	-0,30	-0,08	-1,47	-0,31	38,24	-0,34	-0,03	-1,26	-0,22
	SPEED UP	1,00	4,94	2,71	4,86	3,00	1,00	5,24	2,88	5,22	3,12	1,00	5,44	2,95	5,46	3,21
	F.T.	1,02	5,04	2,77	4,97	3,07	0,92	4,79	2,64	4,78	2,86	0,84	4,60	2,49	4,61	2,71
Crew	PSNR (dB)	38,97	-0,71	-0,87	-1,50	-1,56	40,04	-0,57	-0,66	-1,28	-1,17	40,97	-0,53	-0,56	-1,23	-0,96
	SPEED UP	1,00	3,64	2,58	3,88	3,14	1,00	3,71	2,64	3,99	3,15	1,00	3,78	2,72	4,08	3,19
	F.T.	0,98	3,57	2,53	3,80	3,07	0,89	3,31	2,36	3,56	2,81	0,83	3,13	2,25	3,38	2,64
Harbour	PSNR (dB)	33,78	-0,21	-0,33	-2,19	-1,36	34,82	-0,24	-0,26	-1,90	-1,03	35,71	-0,24	-0,22	-1,93	-0,83
	SPEED UP	1,00	4,18	2,68	5,06	3,23	1,00	4,32	2,75	5,30	3,33	1,00	4,41	2,77	5,42	3,38
	F.T.	0,91	3,80	2,43	4,59	2,93	0,82	3,56	2,27	4,37	2,75	0,77	3,40	2,14	4,18	2,61
Ice	PSNR (dB)	44,07	-0,78	-0,81	-1,67	-0,92	45,03	-0,75	-0,66	-1,45	-0,74	45,86	-0,77	-0,59	-1,40	-0,63
	SPEED UP	1,00	4,09	2,45	3,98	3,38	1,00	4,30	2,63	4,24	3,52	1,00	4,51	2,75	4,47	3,62
	F.T.	1,41	5,78	3,47	5,63	4,78	1,27	5,44	3,33	5,36	4,45	1,14	5,16	3,15	5,11	4,14
Soccer	PSNR (dB)	37,64	-1,04	-1,13	-1,94	-1,64	38,63	-0,89	-0,87	-1,69	-1,28	39,52	-0,80	-0,73	-1,54	-1,06
	SPEED UP	1,00	4,11	2,68	4,21	3,33	1,00	4,24	2,78	4,39	3,39	1,00	4,30	2,89	4,60	3,40
	F.T.	1,00	4,11	2,67	4,20	3,32	0,90	3,83	2,51	3,97	3,06	0,82	3,55	2,38	3,79	2,81
Moyenne PSNR (dB)		38,24	-0,61	-0,65	-1,81	-1,18	39,21	-0,55	-0,51	-1,56	-0,91	40,06	-0,54	-0,43	-1,47	-0,74
Moyenne speed up		1,00	4,19	2,62	4,40	3,21	1,00	4,36	2,74	4,63	3,30	1,00	4,49	2,82	4,81	3,36
Moyenne F.T.		1,06	4,46	2,77	4,64	3,43	0,96	4,19	2,62	4,41	3,19	0,88	3,97	2,48	4,21	2,98

Tableau 3.7 Résultats de vidéos HD, 1280×720 en PSNR, en accélération (*speed up*) et en fréquence de trames (F.T.) pour les méthodes Cascade, ResidualM, STAT, STAT+REF et MV+MS. Les résultats en PSNR montrent les différences en qualité avec la méthode en cascade. Les vidéos ont été codées en MPEG-4, au débit de 7,2 Mb/s

HD 1280×720		1,15Mb/s					2,3Mb/s					3,45 Mb/s				
		Cas-cade	Resi-dualM	STAT+REF	STAT	MV+MS	Cas-cade	Resi-dualM	STAT+REF	STAT	MV+MS	Cas-cade	Resi-dualM	STAT+REF	STAT	MV+MS
Mobcal-ter	PSNR (dB)	32,11	-0,83	-1,88	-4,73	-2,69	34,31	-0,47	-0,69	-3,41	-1,20	35,50	-0,49	-0,41	-2,59	-0,79
	SPEED UP	1,00	3,37	1,76	2,71	2,87	1,00	3,68	1,98	3,19	2,66	1,00	4,01	2,13	3,50	2,64
	F.T.	1,05	3,55	1,85	2,85	3,02	0,84	3,09	1,66	2,68	2,23	0,73	2,93	1,56	2,56	1,93
Parkrun-ter	PSNR (dB)	24,65	-0,52	-0,61	-2,47	-1,54	26,59	-0,24	-0,30	-2,21	-0,79	27,89	-0,21	-0,14	-2,09	-0,51
	SPEED UP	1,00	3,61	1,80	2,82	3,03	1,00	3,96	2,07	3,30	3,03	1,00	4,17	2,24	3,61	3,03
	F.T.	1,04	3,76	1,87	2,93	3,15	0,84	3,32	1,74	2,77	2,54	0,74	3,07	1,65	2,65	2,23
Shields-ter	PSNR (dB)	31,26	-1,20	-2,63	-4,50	-3,74	33,87	-0,60	-1,05	-3,33	-1,64	35,22	-0,62	-0,66	-2,62	-1,01
	SPEED UP	1,00	3,30	1,72	2,69	2,91	1,00	3,60	1,96	3,16	2,69	1,00	3,96	2,10	3,47	2,63
	F.T.	1,04	3,43	1,79	2,79	3,02	0,83	2,98	1,62	2,62	2,23	0,72	2,86	1,52	2,51	1,90
Moyenne PSNR (dB)		29,34	-0,85	-1,71	-3,90	-2,66	31,59	-0,44	-0,68	-2,98	-1,21	32,87	-0,44	-0,40	-2,43	-0,77
Moyenne speed up		1,00	3,43	1,76	2,74	2,94	1,00	3,74	2,00	3,22	2,79	1,00	4,05	2,16	3,53	2,77
Moyenne F.T.		1,04	3,58	1,84	2,86	3,06	0,84	3,13	1,67	2,69	2,33	0,73	2,95	1,57	2,57	2,02

HD 1280×720		4,6Mb/s					5,76Mb/s					6,91Mb/s				
		Cas-cade	Resi-dualM	STAT+REF	STAT	MV+MS	Cas-cade	Resi-dualM	STAT+REF	STAT	MV+MS	Cas-cade	Resi-dualM	STAT+REF	STAT	MV+MS
Mobcal-ter	PSNR (dB)	36,37	-0,47	-0,30	-2,17	-0,63	37,08	-0,46	-0,21	-1,87	-0,52	37,72	-0,44	-0,18	-1,67	-0,47
	SPEED UP	1,00	4,19	2,24	3,74	2,67	1,00	4,34	2,32	3,93	2,71	1,00	4,51	2,40	4,11	2,78
	F.T.	0,66	2,77	1,48	2,48	1,77	0,61	2,65	1,42	2,40	1,66	0,57	2,57	1,37	2,34	1,58
Parkrun-ter	PSNR (dB)	29,00	-0,29	-0,15	-1,97	-0,52	29,88	-0,27	-0,09	-1,83	-0,45	30,65	-0,27	-0,08	-1,71	-0,39
	SPEED UP	1,00	4,33	2,35	3,84	3,04	1,00	4,49	2,47	4,07	3,07	1,00	4,64	2,56	4,26	3,10
	F.T.	0,67	2,89	1,57	2,56	2,03	0,61	2,74	1,51	2,49	1,87	0,57	2,64	1,46	2,42	1,76
Shields-ter	PSNR (dB)	36,18	-0,58	-0,46	-2,24	-0,75	36,94	-0,54	-0,35	-1,95	-0,59	37,61	-0,53	-0,29	-1,73	-0,51
	SPEED UP	1,00	4,11	2,20	3,70	2,65	1,00	4,26	2,28	3,88	2,68	1,00	4,41	2,38	4,06	2,74
	F.T.	0,65	2,69	1,44	2,42	1,74	0,61	2,58	1,38	2,35	1,62	0,56	2,48	1,34	2,28	1,54
Moyenne PSNR (dB)		33,85	-0,45	-0,30	-2,13	-0,63	34,63	-0,42	-0,22	-1,88	-0,52	35,33	-0,41	-0,18	-1,70	-0,46
Moyenne speed up		1,00	4,21	2,26	3,76	2,79	1,00	4,36	2,36	3,96	2,82	1,00	4,52	2,45	4,14	2,87
Moyenne F.T.		0,66	2,78	1,50	2,49	1,84	0,61	2,66	1,44	2,41	1,72	0,57	2,56	1,39	2,35	1,63

Tableau 3.8 Résultats de vidéos HD, 1920×1080 en PSNR, en accélération (*speed up*) et en fréquence de trames (F.T.) pour les méthodes Cascade, ResidualM, STAT, STAT+REF et MV+MS. Les résultats en PSNR montrent les différences en qualité avec la méthode en cascade. Les vidéos ont été codées en MPEG-4, au débit de 16,4 Mb/s

HD 1920×1080		2,62Mb/s					5,24Mb/s					7,87Mb/s				
		Cas-cade	Resi-dualM	STAT+REF	STAT	MV+MS	Cas-cade	Resi-dualM	STAT+REF	STAT	MV+MS	Cas-cade	Resi-dualM	STAT+REF	STAT	MV+MS
Riverbed	PSNR (dB)	29,65	-3,02	-5,46	-5,58	-5,16	32,50	-2,21	-3,25	-3,49	-7,72	34,33	-1,81	-2,44	-2,75	-4,73
	SPEED UP	1,00	2,87	2,62	3,74	2,55	1,00	3,33	2,94	4,13	3,71	1,00	3,47	2,87	4,18	3,62
	F.T.	0,29	0,84	0,76	1,09	0,74	0,20	0,65	0,57	0,80	0,72	0,16	0,56	0,46	0,67	0,58
Rush-hour	PSNR (dB)	41,23	-1,73	-3,47	-4,27	-5,79	43,76	-0,98	-1,33	-2,06	-2,21	45,00	-0,96	-0,84	-1,53	-1,43
	SPEED UP	1,00	3,16	2,03	3,12	3,12	1,00	3,57	2,31	3,65	3,21	1,00	4,08	2,56	4,06	3,34
	F.T.	0,36	1,13	0,73	1,12	1,12	0,25	0,91	0,59	0,93	0,82	0,21	0,84	0,53	0,83	0,69
Station2	PSNR (dB)	40,57	-0,81	-1,35	-3,26	-1,80	42,29	-0,51	-0,49	-2,02	-0,67	43,16	-1,26	-0,29	-1,57	-0,48
	SPEED UP	1,00	4,44	2,41	3,82	3,07	1,00	5,38	3,04	4,98	3,15	1,00	6,88	3,35	5,70	3,29
	F.T.	0,30	1,34	0,73	1,15	0,93	0,21	1,15	0,65	1,06	0,67	0,18	1,21	0,59	1,00	0,58
Sunflower	PSNR (dB)	41,06	-2,06	-1,87	-4,40	-3,44	43,85	-1,79	-0,83	-2,92	-1,68	45,20	-1,90	-0,52	-2,33	-1,06
	SPEED UP	1,00	3,83	1,93	3,10	2,81	1,00	4,41	2,33	3,87	2,71	1,00	4,99	2,56	4,36	2,70
	F.T.	0,37	1,42	0,72	1,15	1,04	0,27	1,20	0,63	1,05	0,74	0,23	1,13	0,58	0,99	0,61
Moyenne PSNR (dB)		38,13	-1,91	-3,04	-4,38	-4,05	40,60	-1,37	-1,48	-2,62	-3,07	41,92	-1,48	-1,02	-2,05	-1,93
Moyenne speed up		1,00	3,57	2,25	3,45	2,89	1,00	4,17	2,65	4,16	3,19	1,00	4,86	2,83	4,58	3,24
Moyenne F.T.		0,33	1,18	0,73	1,13	0,96	0,23	0,98	0,61	0,96	0,74	0,19	0,93	0,54	0,87	0,61

HD 1920×1080		10,49Mb/s					13,12Mb/s					15,74Mb/s				
		Cas-cade	Resi-dualM	STAT+REF	STAT	MV+MS	Cas-cade	Resi-dualM	STAT+REF	STAT	MV+MS	Cas-cade	Resi-dualM	STAT+REF	STAT	MV+MS
Riverbed	PSNR (dB)	35,77	-1,54	-2,06	-2,37	-3,43	36,93	-1,29	-1,73	-2,08	-2,69	38,00	-1,17	-1,58	-1,93	-2,27
	SPEED UP	1,00	3,47	2,78	4,15	3,54	1,00	3,48	2,77	4,24	3,64	1,00	3,45	2,70	4,19	3,68
	F.T.	0,14	0,49	0,40	0,59	0,50	0,13	0,45	0,36	0,55	0,47	0,12	0,41	0,32	0,50	0,44
Rush-hour	PSNR (dB)	45,80	-0,99	-0,65	-1,39	-1,14	46,42	-1,05	-0,65	-1,59	-1,04	46,98	-1,22	-0,65	-1,60	-0,94
	SPEED UP	1,00	4,28	2,72	4,29	3,30	1,00	4,56	2,92	4,67	3,46	1,00	4,56	2,94	4,64	3,34
	F.T.	0,18	0,78	0,49	0,78	0,60	0,16	0,72	0,46	0,73	0,54	0,15	0,67	0,44	0,69	0,49
Station2	PSNR (dB)	43,79	-1,12	-0,21	-1,41	-0,36	44,35	-1,13	-0,20	-1,35	-0,35	44,87	-1,05	-0,18	-1,29	-0,31
	SPEED UP	1,00	7,42	3,60	6,16	3,46	1,00	7,79	3,72	6,49	3,59	1,00	8,12	3,79	6,77	3,72
	F.T.	0,15	1,15	0,56	0,96	0,54	0,14	1,10	0,52	0,91	0,51	0,13	1,05	0,49	0,88	0,48
Sunflower	PSNR (dB)	46,06	-1,85	-0,43	-2,28	-0,81	46,76	-1,81	-0,39	-2,19	-0,68	47,35	-1,68	-0,34	-2,05	-0,57
	SPEED UP	1,00	5,55	2,74	4,91	2,82	1,00	5,90	2,92	5,25	2,90	1,00	6,26	3,11	5,59	3,06
	F.T.	0,19	1,06	0,52	0,94	0,54	0,17	1,00	0,50	0,89	0,49	0,15	0,96	0,48	0,86	0,47
Moyenne PSNR (dB)		42,86	-1,38	-0,84	-1,86	-1,44	43,62	-1,32	-0,74	-1,80	-1,19	44,30	-1,28	-0,69	-1,72	-1,02
Moyenne speed up		1,00	5,18	2,96	4,88	3,28	1,00	5,43	3,08	5,16	3,40	1,00	5,60	3,14	5,30	3,45
Moyenne F.T.		0,17	0,87	0,49	0,82	0,54	0,15	0,82	0,46	0,77	0,50	0,14	0,78	0,43	0,73	0,47

3.6 Conclusion

Ce chapitre présente une méthode de transcodage MPEG-4 à H.264. La méthode consiste à récupérer des informations disponibles lors de l'étape de décodage MPEG-4 et à les utiliser afin de sauter efficacement certaines étapes du codage H.264. L'algorithme innove en tirant particulièrement avantage des propriétés de l'information résiduelle, afin de réduire drastiquement le nombre de MCs à tester, contrairement aux algorithmes de l'état de l'art qui testent un nombre nettement plus élevé de MCs. Des détails supplémentaires peuvent être trouvés dans le brevet Metoevi et Coulombe (2009b).

Le processus de détermination des VMs s'effectue également de façon unique, totalement différente de l'état de l'art. L'information résiduelle est, encore une fois, exploitée de façon particulière, de sorte qu'à ne raffiner que certains VMs, ceux qui sont jugés inefficaces. On évite ainsi des raffinements inutiles à la qualité et coûteux en terme de calculs, contrairement aux algorithmes de l'état de l'art, qui raffinent automatiquement tous les VMs MPEG-4. Des détails supplémentaires peuvent être trouvés dans le brevet, Metoevi et Coulombe (2009a).

Nous avons pu ainsi obtenir de bons résultats, des moyennes d'accélération de $3\times$ à $5\times$ par rapport à la méthode cascade, tout en maintenant une bonne qualité. La méthode développée donne également des résultats supérieurs à ceux de l'état de l'art en plus de se prêter aisément aux applications en temps réels. Dans les chapitres suivants, nous tenterons d'améliorer la détermination des seuils utilisés dans le raffinement des VMs ainsi que dans la classification des MBs. On essaiera, notamment, d'adapter ces seuils au débit binaire ainsi qu'au type de séquences vidéos à transcoder.

CHAPITRE 4

MÉTHODE DE TRANSCODAGE MPEG-4 À H.264 BASÉE SUR LA SOMME ABSOLUE RELATIVE DU RÉSIDUEL

4.1 Introduction

Bien que menant à des résultats intéressants, l'algorithme développé au chapitre précédent et décrit dans (Metoevi et Coulombe (2009c)) comporte des imperfections. En effet, des seuils fixes ont été utilisés dans le processus de classification des macroblocs et de raffinement des VMs. Ces seuils ne tiennent aucunement compte du débit de codage et des caractéristiques de la vidéo à coder. Et pourtant, l'énergie résiduelle dont se sert l'algorithme dans le processus de classification des MBs et des VMs varie considérablement en fonction des caractéristiques de la vidéo. Par ailleurs, le MC et les VMs sont choisis pour minimiser le rapport entre la distorsion et le débit binaire. Le MC optimal devrait donc changer également en fonction du débit. Comme conséquence, l'algorithme proposé précédemment ne donne pas d'aussi bons résultats en qualité pour tous les débits et les types de vidéos. En particulier, l'algorithme se comporte moins bien à haut débit qu'à bas et moyens débits pour des vidéos à mouvements faibles ou peu complexes. Par ailleurs, l'algorithme semble donner des résultats moins bons en qualité pour des vidéos à mouvements complexes à bas débits. Par exemple, pour la vidéo CIF Football, on obtient des pertes en qualité de 2,5 dB à 128 kb/s comparativement à Mother-daughter dont la perte en qualité est de 0,33 dB. Toutes ces raisons justifient le développement d'une méthode plus efficace et adaptée aux caractéristiques vidéos et au débit de transcodage.

Dans ce chapitre, nous présentons un nouvel algorithme de transcodage MPEG-4 à H.264 basé sur plusieurs idées innovatrices. Premièrement, nous proposons d'utiliser une mesure relative de l'information résiduelle, la somme absolue relative du résiduel (SARR), au lieu de l'énergie résiduelle afin d'adapter notre algorithme au débit binaire et aux caractéristiques vidéos. Ensuite, nous proposons d'utiliser conditionnellement de petites partitions afin d'améliorer la qualité à hauts débits tout en maintenant de bonnes accélérations en vitesse. Et pour finir, nous exploitons le principe de corrélation entre les trames successives, en rassemblant des statis-

tiques sur la SARR pendant le transcodage d'une trame et en les utilisant comme seuils dans le transcodage de la trame suivante. Nous avons ainsi pu établir des seuils adaptés au débit binaire et aux caractéristiques de la vidéo à transcoder. Les outils innovateurs proposés nous ont permis de développer un algorithme adaptatif et d'arriver à de bons résultats en qualité quelles que soient les conditions de transcodage tout en maintenant de bonnes accélérations en vitesse.

Le chapitre est organisé comme suit. Dans un premier temps, nous analysons l'influence du débit binaire et des caractéristiques vidéos sur la qualité du transcodage, et la nécessité de tenir compte de ces deux facteurs afin de développer des algorithmes efficaces. Nous présentons ensuite les outils innovateurs proposés afin d'y arriver, tels que la mesure relative de l'information résiduelle et le principe de corrélation de trames successives. Pour finir, nous présentons le processus de détermination du MC, des VMs et les améliorations qui y sont apportées et terminons par une conclusion.

4.2 Détermination du mode de codage

Nous proposons, tout comme dans l'algorithme précédent de réduire la complexité du processus de détermination du MC en réduisant efficacement le nombre de MCs candidats en ne testant que les MCs les plus probables. Pour déterminer les MCs les plus probables, nous exploiterons les mêmes principes que ceux utilisés dans l'algorithme proposé précédemment. Une table de correspondance tout comme en Lee et al. (2006) est établie entre les MCs MPEG-4 et H.264, de sorte à éliminer les MCs qui apparaissent rarement. Ensuite nous exploitons les propriétés de l'information résiduelle afin de classifier plus précisément les MCs d'entrée et réduire ainsi plus drastiquement l'ensemble des MCs à considérer. Nous apportons toutefois d'importantes modifications et utilisons des principes innovateurs afin de pallier aux insuffisances de l'algorithme précédent, soulignées plus tôt.

Avant de présenter le nouvel algorithme proposé, analysons d'abord en profondeur l'influence du débit binaire et des caractéristiques vidéos sur la qualité du transcodage et la nécessité de tenir compte de ces deux facteurs afin de développer des algorithmes de transcodage efficaces.

Nous présenterons ensuite les outils innovateurs dont on s'est servis pour développer un algorithme adaptatif au débit binaire et aux caractéristiques vidéos, notamment la mesure relative de l'information résiduelle et le principe de corrélation de trames successives.

4.2.1 Influence du type de vidéo sur le mode de codage

L'énergie résiduelle (ER) est globalement plus importante pour des vidéos à mouvements complexes qu'elle ne l'est pour des vidéos à mouvements simples. On peut le constater dans les figures 3.2 et 3.3. On remarque effectivement que la moyenne d'ER est plus élevée pour les vidéos Bus et Soccer qu'elle ne l'est pour les vidéos Miss-america ou Akiyo. Le codage vidéo est généralement soumis à de nombreuses contraintes binaires. Généralement, les algorithmes de contrôle binaire établissent et fixent un budget binaire pour toute la trame et allouent ensuite les bits pour chaque MB. Pour maximiser la qualité, les algorithmes de contrôle de débit binaire attribuent moins de bits aux MBs appartenant à des zones stationnaires ou à mouvements simples qu'aux MBs appartenant à des zones présentant plus de texture ou à mouvements plus complexes. Conséquemment, les MBs complexes ont tendance à être codés en partitions plus petites que ne le sont les MBs à complexité faible.

La complexité d'un MB est une notion toutefois relative. En effet, les limitations de budget binaire font que le budget binaire des MBs d'une trame est plus serré lorsque cette dernière présente un nombre élevé de MBs complexes. Par conséquent des MBs complexes peuvent être codés en grande partition pour respecter les contraintes binaires. Par contre, pour une trame où le pourcentage de MBs complexes est plus faible, leur budget binaire est plus grand et donc des MBs à faible complexité pourraient être codés en petites partitions. L'algorithme de contrôle binaire proposé en UIT/SG16 (1997), par exemple, prend en compte cette relative en utilisant la variance de l'ER pour déterminer les bits à allouer à chaque MB et non l'ER elle-même. Pareillement, nos seuils de classification d'ER ou de toute autre mesure de l'information résiduelle des MCs doivent s'adapter au type de vidéo, ils devront être plus élevés pour des vidéos plus complexes. Inversement, pour des vidéos comportant peu de mouvement ou à

faible mouvement, l'ER est moins importante et ces seuils devraient être plus bas. Définissons maintenant quelques formules avant d'illustrer l'influence du type de vidéo sur le MC.

Soit R_n la somme absolue du résiduel (SAR) du MB n d'une trame définie comme suit :

$$R_n = \sum_{i=0}^{15} \sum_{j=0}^{15} |\mathbf{R}_n(i, j)| \quad (4.1)$$

On rappelle que $\mathbf{R}_n(i, j)$ est la valeur du résiduel du pixel à la position (i, j) du MB n . Définissons également les notations suivantes :

- mci : Le MC d'entrée, soit le MC en MPEG-4.
- mco : Le MC de sortie, soit le MC choisi par H.264.
- S_{mci}^{mco} : L'ensemble de tous les MBs dans une trame codée dans le mode mci en MPEG-4 et transcodés dans le mode mco en H.264.
- $|S_{mci}^{mco}| = \text{card}(S_{mci}^{mco})$: Le nombre d'éléments de l'ensemble S_{mci}^{mco} .

Soit également la moyenne de la somme absolue du résiduel (ASAR) de tous les MBs d'une trame transcodés du MC mci au MC mco en H.264. Elle se définit comme suit :

$$\mu_{mci,R}^{mco} = \frac{1}{|S_{mci}^{mco}|} \sum_{n \in S_{mci}^{mco}} R_n \quad (4.2)$$

Pour simplifier les notations, nous utiliserons la taille de la partition, c'est-à-dire 16×16 , 16×8 , 8×16 et 8×8 pour se référer respectivement aux MCs Inter 16×16 , Inter 16×8 , Inter 8×16 et Inter 8×8 dans les notations. Par exemple S_{mci}^{mco} pour $mci = \text{Inter}16 \times 16$ et $mco = \text{Skip}$ sera noté $S_{16 \times 16}^{\text{Skip}}$ au lieu de $S_{\text{Inter}16 \times 16}^{\text{Skip}}$.

Mettons maintenant en évidence l'influence du type de la vidéo sur le MC. Pour cela, nous avons rassemblé des statistiques sur la ASAR, pour plusieurs types de vidéos, à complexité faible, moyenne et haute pour 2 MCs H.264, Skip et Inter 16×16 . Les résultats sont recensés dans les figures 4.1 et 4.2. Les figures 4.1 et 4.2 montrent respectivement les valeurs de l'ASAR en fonction du débit pour les MCs de sortie H.264 Skip ($\mu_{16 \times 16,R}^{\text{Skip}}$) et Inter 16×16

$(\mu_{16 \times 16, R}^{16 \times 16})$ pour différents types de vidéos. On peut constater que les valeurs de l'ASAR varient énormément selon la vidéo à transcoder. Ces valeurs tendent à augmenter avec la complexité de la vidéo (les valeurs relatives à Bus, par exemple, sont plus élevées que celles relatives à la vidéo Akiyo). Cela signifie qu'en moyenne, la somme résiduelle des MBs qui deviennent des Inter16×16 (Figure 4.2) en H.264 croît avec la complexité de la vidéo. C'est le même constat qui est fait pour les MBs qui deviennent des Skips (Figure 4.1). Le même constat a été fait pour tous les MCs H.264.

Nous avons illustré l'influence des caractéristiques vidéos sur le MC en considérant la SAR. Le résultat est similaire pour l'énergie résiduelle et devrait l'être pour plusieurs autres mesures quantitatives de l'information résiduelle. Pour des résultats optimaux, les seuils de classification des MCs devraient donc prendre en compte les caractéristiques vidéos. Autrement, la classification des MBs peut être telle que tous les MBs se retrouvent dans une seule classe ; ce qui ne nous aide pas. En effet, une mauvaise classification va mener soit à une perte d'accélération (si on teste trop de modes) ou bien à une perte de qualité (si on n'en teste pas assez).

On peut constater également, dans les figures 4.3 et 4.4, l'influence du type de vidéo sur l'information résiduelle et le MC. On y présente des graphiques de l'ER des MBs $\in S_{16 \times 16}^{\text{Skip}}$ et des MBs $\in S_{16 \times 16}^{16 \times 16}$, c'est-à-dire codés en Inter16×16 dans MPEG-4 et transcodés en Skip ou en Inter16×16 dans H.264. On peut constater que l'ER des MBs tend à augmenter avec la complexité de la vidéo. On peut constater aussi que les seuils de classification des MCs, T_{mc}^{Skip} et $T_{mc}^{16 \times 16}$ peuvent être inappropriés pour certaines vidéos. Pour Akiyo (Figure 4.3), $T_{mc}^{16 \times 16}$ fixé à 5000 semble peu approprié, aucun MB ne se retrouve dans la classe Inter16×16 cas II, toutefois en baissant $T_{mc}^{16 \times 16}$ à 400, 57% des MBs de la classe Inter16×16 cas I se retrouveraient en classe Inter16×16 cas II. On gagnerait en accélération, le MC de plusieurs MBs serait fixé à Inter16×16, au lieu de tester 2 MCs, Skip et Inter16×16. Pour la vidéo Bus par contre (Figure 4.4), les seuils de classification T_{mc}^{Skip} et $T_{mc}^{16 \times 16}$ semblent assez bien adaptés. Il ne faut pas oublier que le but du seuil T_{mc}^{Skip} est de séparer les MBs qui seront codés en Skip (ceux sous le seuil T_{mc}^{Skip}) de ceux dont on considérera les modes Skip et Inter16×16. De même, le but du

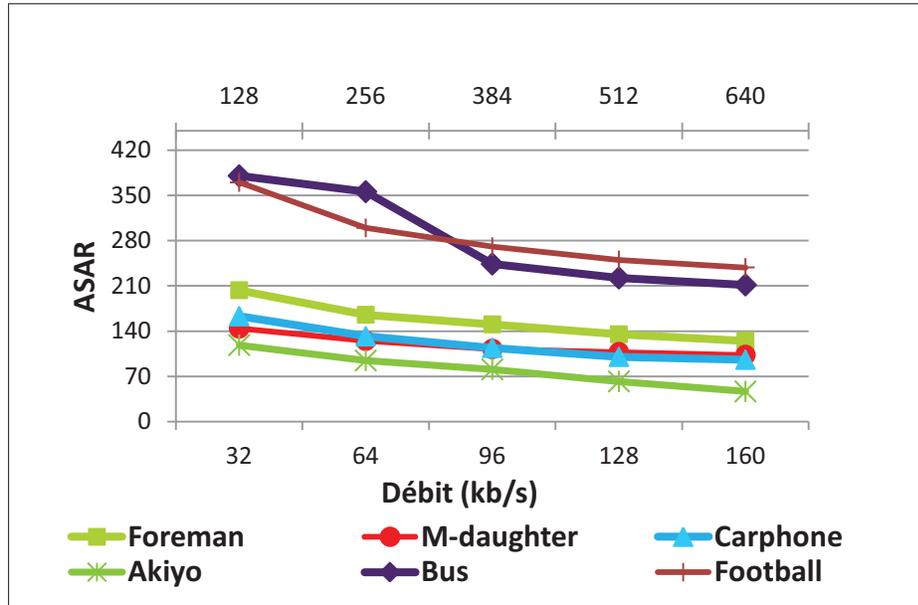


Figure 4.1 Moyenne de la somme absolue du résiduel (ASAR) des MBs codés Inter16×16 en MPEG-4 et transcodés en Skip dans H.264 ($\mu_{16 \times 16, R}^{Skip}$) en fonction du débit pour différentes vidéos QCIF et CIF. Les résultats sont relatifs à un transcodage en cascade de la trame 68.

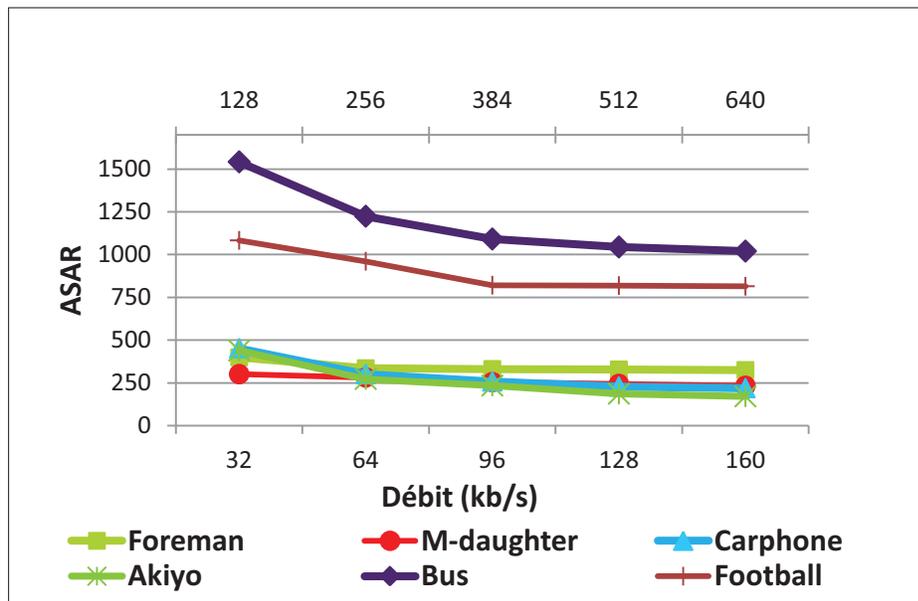


Figure 4.2 Moyenne de la somme absolue du résiduel (ASAR) des MBs codés Inter16×16 en MPEG-4 et transcodés en Inter16×16 dans H.264 ($\mu_{16 \times 16, R}^{16 \times 16}$) en fonction du débit pour différentes vidéos QCIF et CIF. Les résultats sont relatifs à un transcodage en cascade de la trame 68.

seuil $T_{mc}^{16 \times 16}$ est de séparer les MBs qui seront codés en Inter16×16 (ceux au-dessus du seuil $T_{mc}^{16 \times 16}$) de ceux dont on considérera les modes Skip et Inter16×16.

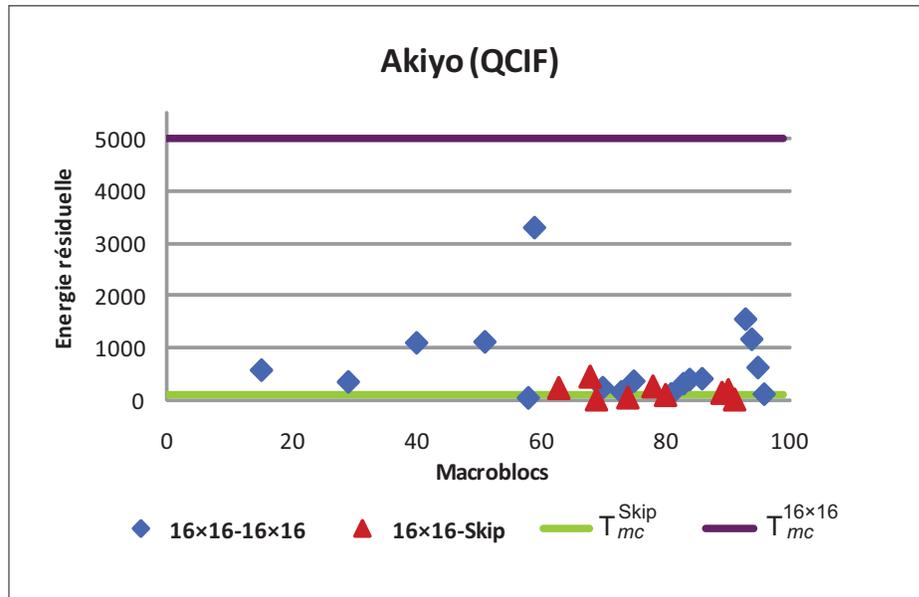


Figure 4.3 Graphique représentant l'énergie résiduelle des MBs codés en Inter16×16 dans MPEG-4 et transcodés en Skip dans H.264 ainsi que celle des MBs codés en Inter16×16 dans MPEG-4 et transcodés en Inter16×16 dans H.264. Les résultats sont obtenus pour un transcodage en cascade de la trame 68 de la vidéo QCIF Akiyo. L'axe horizontal représente le numéro du MB.

4.2.2 Influence du débit binaire sur le mode de codage

Le choix du MC résulte d'un processus RDO. Et donc, le MC optimal peut être complètement différent d'un débit à un autre. En effet, le coût de codage de petites partitions et des VMs croît avec la contrainte de débit. De plus grandes partitions sont alors privilégiées pour des débits moins élevés. Ce phénomène se constate dans les figures 4.1 et 4.2, puisque les valeurs de la ASAR baissent avec le débit de binaire. Ce qui signifie que le nombre de MBs à haute complexité qui sont transcodés avec de grandes partitions baisse quand le débit binaire augmente. À débits élevés, puisque les contraintes de débits sont moindres, le codeur peut se permettre de les coder avec des partitions plus petites. On peut constater également dans la table 4.1 de

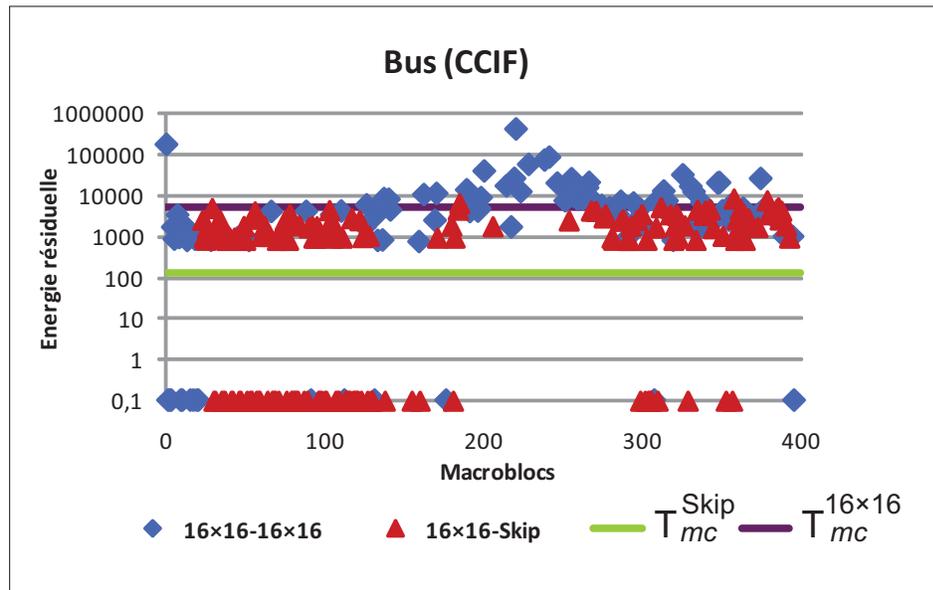


Figure 4.4 Graphique représentant l'énergie résiduelle des MBs codés en Inter16×16 dans MPEG-4 et transcodés en Skip dans H.264 ainsi que celle des MBs codés en Inter16×16 dans MPEG-4 et transcodés en Inter16×16 dans H.264. Les résultats sont obtenus pour un transcodage en cascade de la trame 68 de la vidéo CIF Bus. L'axe horizontal représente le numéro du MB. L'axe vertical est en échelle logarithmique.

fréquence de correspondance des MCs MPEG-4 et H.264, qu'à 160 kb/s que les petites partitions apparaissent de façon plus importante qu'à 32 kb/s. Pour de meilleures performances, les seuils de classification des MBs doivent donc tenir compte du débit binaire.

4.2.3 Utilisation de la somme absolue relative du résiduel comme mesure quantitative de complexité du macrobloc

Tel qu'explicité précédemment, la mesure de la complexité du MB ne peut se limiter uniquement à l'ER ou à la SAR, le processus de décision du MC étant contraint et fortement influencé par plusieurs facteurs (débit, caractéristiques de la vidéo, etc.). La complexité d'un MB est donc une notion relative. Il nous faut une mesure relative de l'information résiduelle pour une classification plus efficace des MBs. Nous proposons d'utiliser la somme absolue du résiduel du MB relative à la moyenne de la somme résiduelle de toute la trame. On l'appellera somme absolue relative du résiduel (SARR). Cette mesure représente la complexité relative du MB comparativement aux autres MBs de la trame. Seuls les MBs dont la SAR est supérieure

Tableau 4.1 Moyenne de fréquence d'apparition des MCs H.264 en fonction des MCs MPEG-4 pour plusieurs types de vidéos QCIF transcodées à 32 kb/s et 160 kb/s en H.264, en utilisant le codeur H.264 d'Intel. La séquence MPEG-4 est initialement encodée à 200 kb/s

Les statistics ont été rassemblées pour plusieurs types de vidéos QCIF et CIF à 32kb/s

Modes de codage H.264 / Modes de codage MPEG-4	Intra 4×4	Intra 16×16	Skip	Inter 16×16	Inter 16×8	Inter 8×16	Inter 8×8	sb8×8	sb8×4	sb4×8	sb4×4
Intra	64,2%	26,1%	7,8%	1,6%	0,1%	0,1%	0,1%	100,0%	0,0%	0,0%	0,0%
Inter16×16 (64%)	0,3%	1,9%	66,2%	27,7%	2,0%	1,8%	0,1%	99,8%	0,1%	0,1%	0,0%
Inter8×8 (21%)	2,9%	3,8%	21,7%	39,5%	14,9%	16,8%	0,4%	98,1%	0,5%	0,7%	0,7%
Skip (15%)	0,1%	0,6%	92,3%	6,8%	0,1%	0,1%	0,0%	0,0%	0,0%	0,0%	0,0%

Les statistics ont été rassemblées pour plusieurs types de vidéos QCIF et CIF à 160kb/s

Modes de codage H.264 / Modes de codage MPEG-4	Intra 4×4	Intra 16×16	Skip	Inter 16×16	Inter 16×8	Inter 8×16	Inter 8×8	sb8×8	sb8×4	sb4×8	sb4×4
Intra	73,8%	21,6%	1,8%	1,9%	0,3%	0,2%	0,4%	100,0%	0,0%	0,0%	0,0%
Inter16×16 (64%)	1,0%	2,7%	28,5%	54,9%	6,0%	5,8%	1,1%	98,5%	0,4%	0,4%	0,7%
Inter8×8 (21%)	6,7%	1,0%	1,0%	22,8%	10,4%	16,5%	41,6%	78%	3,8%	6,0%	12,2%
Skip (15%)	0,1%	0,6%	85,7%	12,9%	0,3%	0,2%	0,2%	100,0%	0,0%	0,0%	0,0%

à la moyenne de la SAR des MBs de la trame seront considérés comme complexes. Cette mesure prend mieux en compte les caractéristiques de la vidéo et traduit mieux la complexité du MB point de vue du RDO.

Par exemple, lorsqu'une trame possède de nombreux MBs complexes, ces derniers peuvent être codés en grandes partitions pour respecter les contraintes de débit binaires surtout à bas débit. En utilisant la somme absolue relative, seuls les MBs dont la complexité est supérieure à la moyenne seront considérés complexes. Les autres MBs ne le seront pas même si leur SAR est élevée et de grandes partitions pourront alors être considérées pour leur codage. Ce qui n'aurait pas été le cas en utilisant l'ER et des seuils fixes.

De même, pour une trame où il y a peu de MBs complexes, un MB à faible complexité et dont la SARR est faible pourra être considéré à haute complexité pourvu que sa SARR soit supérieure à celle de la moyenne. On pourra ainsi considérer de petites partitions de MCs pour ce MB. C'est l'objectif recherché, car dans une telle trame les contraintes binaires sont

moins serrées et le codeur peut se permettre de coder en petites partitions même des MBs à faible complexité et dont la SAR est faible. Cela n'aurait pas été le cas en utilisant l'ER, car ce MB, ayant un faible ER, aurait été considéré à faible complexité et on aurait considéré que de grandes partitions comme MCs candidats pour ce MB. Donc, en résumé, utiliser la SARR comme mesure de complexité pourra permettre une classification plus efficace des MBs comparativement à l'ER et à des seuils fixes utilisés dans l'algorithme décrit au chapitre précédent.

On pourrait utiliser l'ER d'un MB par rapport à la moyenne des ER de tous les MBs d'une trame. Nous avons cependant préféré la somme absolue à l'ER car cette dernière est plus gourmande en calculs que la somme absolue.

Définissons maintenant quelques formules. Soit μ_G , la moyenne de la SAR sur toute la trame. Elle est définie comme suit :

$$\mu_G = \frac{16 \times 16}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |\mathbf{R}(i, j)| \quad (4.3)$$

$\mathbf{R}(i, j)$ étant la valeur résiduelle à la position du pixel (i, j) dans la trame. μ_G peut également s'écrire en fonction de la SAR des MBs de la trame sous la forme suivante :

$$\begin{aligned} \mu_G &= \frac{1}{N_{mb}} \sum_{n=0}^{N_{mb}-1} \sum_{i=0}^{15} \sum_{j=0}^{15} |\mathbf{R}_n(i, j)| \\ &= \frac{1}{N_{mb}} \sum_{n=0}^{N_{mb}-1} R_n \end{aligned} \quad (4.4)$$

N_{mb} étant le nombre de MBs de la trame. Notons $R_{n/G}$ la SARR du MB n et définissons la formellement. Cette dernière est la SAR pour un MB n sur μ_G , la moyenne du SAR de tous les MBs de la trame. Elle peut s'écrire sous la forme de l'équation 4.5 suivante :

$$\begin{aligned} R_{n/G} &= \sum_{i=0}^{15} \sum_{j=0}^{15} \frac{|\mathbf{R}_n(i, j)|}{\mu_G + \varepsilon} \\ &= \frac{R_n}{\mu_G + \varepsilon} \end{aligned} \quad (4.5)$$

ε étant une petite valeur positive servant à éviter une division par 0. Elle a été fixée à 10^{-10} dans nos simulations. La SARR, $R_{n/G}$, peut également être évaluée sur un bloc k du MB n . La SARR du bloc k sera notée $R_{(n,k)/G}$ et se calcule selon la formule 4.6.

$$R_{(n,k)/G} = 4 \sum_{i=0}^7 \sum_{j=0}^7 \frac{|\mathbf{R}_n(i + p_{kx}, j + p_{ky})|}{\mu_G + \varepsilon} \quad (4.6)$$

$\mathbf{R}_n(i + p_{kx}, j + p_{ky})$ représentant la valeur du résiduel à la position (i, j) relatif au MB n et correspondant à un pixel à l'intérieur du bloc k . Avec $\mathbf{p}_k = (p_{kx}, p_{ky})$ pour $0 \leq k \leq 3$, où $\mathbf{p}_0 = (0, 0)$, $\mathbf{p}_1 = (8, 0)$, $\mathbf{p}_2 = (0, 8)$ et $\mathbf{p}_3 = (8, 8)$.

On peut voir clairement que $R_{n/G}$ peut s'exprimer comme la somme des quatre $R_{(n,k)/G}$ du MB n comme suit :

$$R_{n/G} = \frac{1}{4} \sum_{k=0}^3 R_{(n,k)/G} \quad (4.7)$$

L'objectif principal de l'algorithme étant de diminuer le temps de traitement, nous proposons d'effectuer le calcul de $R_{n/G}$ sur la moitié des pixels par un échantillonnage quinconce (donc avec un facteur de réduction de 2), afin de réduire davantage son temps de calcul. Le choix des pixels se fait selon la figure 4.5. $R_{n/G}$ se calcule alors tel qu'indiqué dans l'équation 4.8.

$$R_{n/G} = \frac{2}{\mu_G + \varepsilon} \sum_{j=0}^{15} \sum_{i=0}^7 |\mathbf{R}_n(2i + j \bmod 2, j)| \quad (4.8)$$

$R_{(n,k)/G}$ quant à elle peut se calculer en utilisant la formule 4.9 suivante :

$$R_{(n,k)/G} = \frac{8}{\mu_G + \varepsilon} \sum_{j=0}^7 \sum_{i=0}^3 |\mathbf{R}_n(p_{kx} + 2i + j \bmod 2, p_{ky} + j)| \quad (4.9)$$

mod étant l'opérateur modulo. En annexe I, dans les tableaux I.1 à I.4, nous pouvons voir les résultats expérimentaux en accélération de vitesse obtenus en effectuant le calcul de $R_{n/G}$ sur la moitié des pixels. On peut constater de bonnes améliorations de gains en vitesse, de 8

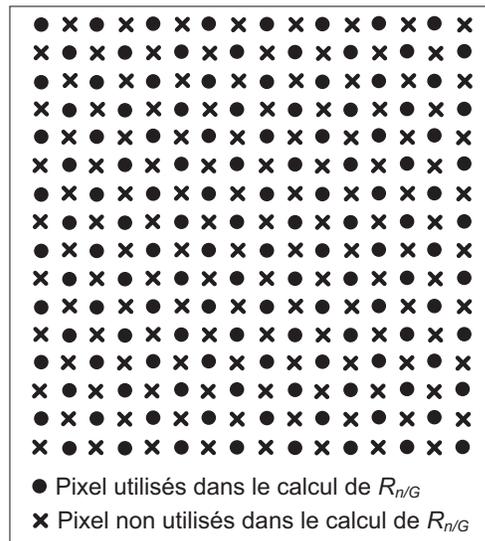


Figure 4.5 Pixels utilisés pour le calcul de la $R_{n/G}$.

à 15% pour une perte en qualité pratiquement nulle, en moyenne de 0,03 dB. Par exemple, en moyenne à hauts débits on obtient des gains en vitesse de 3,84× en évaluant la SARR sur tous les pixels et de 4,44× en l'évaluant sur la moitié des pixels pour du QCIF. Les gains et la différence en PSNR sont du même ordre pour toutes les résolutions.

Avant de présenter l'algorithme, il serait important de vérifier l'hypothèse émise qui dit qu'il existe un lien entre la SARR et le MC H.264. Pour cela, nous avons mis en évidence dans la figure 4.6, le lien entre $R_{n/G}$ et le MC H.264. Des résultats similaires ont été obtenus avec d'autres séquences et trames. On peut constater effectivement le lien entre $R_{n/G}$ et le MC

H.264. Les régions avec de forts contrastes (zones blanchâtres ou plus foncées) ont les valeurs de $R_{n/G}$ les plus grandes et ont tendance à être codées en partitions plus petites que ne le sont les autres. Notre hypothèse quant à la relation entre $R_{n/G}$ et le MC semble se valider. Toutefois, afin d'étayer avec certitude les différentes hypothèses émises jusque-là, nous utiliserons un outil plus fiable : l'ANOVA (ANalysis Of VAriance).

Des détails sur le fonctionnement du test ANOVA sont présentés à l'annexe II. Nous y avons également présenté des résultats supplémentaires de tests ANOVA effectués pour appuyer nos

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	104	393	389	0	0	0	0
0	0	0	0	468	686	486	129	0	0	0
0	0	0	0	1041	899	755	274	0	0	0
0	0	0	0	501	762	616	480	0	0	0
0	0	40	0	491	808	674	233	129	0	0
0	0	217	32	34	364	297	32	51	59	0
0	153	141	124	139	226	43	159	64	129	0
0	112	83	0	64	161	288	118	117	144	0

	Skip or Inter16×16		Inter16×8		Inter8×16		Inter8×8
--	-----------------------	--	-----------	--	-----------	--	----------

Figure 4.6 Lien entre $R_{n/G}$ et le MC H.264 pour la vidéo QCIF Akiyo. Les valeurs à l'intérieur du MB sont la SARR $R_{n/G}$.

hypotheses quant à la relation entre $R_{n/G}$, l'énergie résiduelle, le MC MPEG-4 et le MC H.264. Le tableau 4.2 montre les résultats d'un test ANOVA mettant en relief le lien entre $R_{n/G}$ et le MC H.264. Le test a été effectué pour les vidéos QCIF Akiyo, Foreman, Carphone, Soccer et Football sous 5 débits binaires de transcodage (32, 64, 96, 128 et 160 kb/s) pour les trames 2 et 68. Les séquences vidéos ont été encodées en MPEG-4 à 200 kb/s. La valeur nulle obtenue pour $Prob > F$ dans la table ANOVA démontre avec un haut niveau de certitude la véracité de notre hypothèse, c.à.d. qu'il existe une forte relation entre $R_{n/G}$ et le MC H.264. À l'annexe II, on démontre aussi qu'il y a une forte relation entre l'énergie résiduelle et le MC H.264 de même qu'entre le MC MPEG-4 et le MC H.264.

Tableau 4.2 Table ANOVA mettant en relief le lien entre la $R_{n/G}$ et celui de H.264 pour plusieurs vidéos

Analysis of Variance					
Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
SARR	31019	633	49.0031	9.56	0
Error	25198.3	4914	5.1279		
Total	56217.3	5547			

4.2.4 Utilisation du principe de corrélation de trames consécutives

Le taux de trame d'une vidéo, souvent de 25 ou 30Hz, est tel que les trames successives sont généralement très corrélées. Il est donc vraisemblable que les caractéristiques d'une trame (telles que la complexité, la direction du mouvement, les MCs, les paramètres de quantification) soient similaires d'une trame à la suivante. Ainsi donc, les statistiques et données rassemblées lors du transcodage d'une trame pourraient s'appliquer et être exploitées dans le transcodage de la trame suivante. On s'en servira notamment pour améliorer les seuils de classification des MBs et de raffinement des VMs.

Cette propriété de corrélation a déjà été exploitée par certains auteurs. En Pan et al. (2006), l'auteur s'en sert dans le processus RDO afin d'accélérer le processus de détermination du MC. Concrètement, il commence par établir la probabilité d'occurrence pour chaque MC H.264 permis et ordonne les MCs selon cette probabilité. Ensuite, il calcule, selon cet ordre d'occurrence, le coût lié à chaque MC. Ce processus RDO est cependant écourté par un critère d'arrêt. L'auteur termine le processus de calcul du coût des MCs lorsque le coût d'un MC est inférieur à un seuil qui est calculé à partir de la valeur moyenne et de la variance d'une valeur qui est prédite dynamiquement du coût du MB de la trame passée. L'auteur se sert ainsi des données de la trame passée en faisant l'hypothèse que le coût du MB est fortement corrélé avec celui des MBs voisins et de celui du MB de la trame passé.

Détaillons et définissons les statistiques que nous proposons d'exploiter. Soit :

- $S_{mci}^{mco,t}$: L'ensemble de tous les MBs, d'une trame t , codés avec le MC mci en MPEG-4 et transcodés dans le MC mco en H.264.
- $R_{n/G}^t$: La SARR d'un MB de la trame t .
- $R_{(n,k)/G}^t$: La SARR du bloc k du MB n de la trame t .
- $\mu_{mci,R/G}^{mco,t}$: La moyenne sur $R_{n/G}^t$ pour tous les MBs $n \in S_{mci}^{mco,t}$ d'une trame t ; cette moyenne sera aussi appelée ASARR pour moyenne de la somme absolue du résiduel relatif. $\mu_{mci,R/G}^{mco,t}$ est calculé comme suit :

$$\mu_{mci,R/G}^{mco,t} = \frac{1}{|S_{mci}^{mco,t}|} \sum_{n \in S_{mci}^{mco,t}} R_{n/G}^t \quad (4.10)$$

On rappelle que $|S_{mci}^{mco,t}|$ est la cardinalité de $S_{mci}^{mco,t}$, c'est-à-dire le nombre d'éléments de l'ensemble $S_{mci}^{mco,t}$.

Nous proposons de rassembler des statistiques, durant le transcodage de la trame t , sur $R_{n/G}^t$, pour tous les MBs $n \in S_{mci}^{mco,t}$. Nous calculons spécifiquement la moyenne de $R_{n/G}^t$ sur l'ensemble des MBs $n \in S_{mci}^{mco,t}$, pour chaque combinaison des MCs d'entrée MPEG-4 mci et de MCs de sortie H.264 mco . Nous supposons qu'un MB d'une trame t avec un mode de départ MPEG-4 mci et une $R_{n/G}^t$ proche de $\mu_{mci,R/G}^{mco,t}$ aura une forte probabilité d'être transcodé dans le MC H.264 mco .

Les moyennes $\mu_{mci,R/G}^{mco,t}$ sont calculées durant le transcodage effectif d'une trame t et sont utilisées comme seuils dans le transcodage de la trame suivante $t+1$, notamment comme seuils de classification des MBs et de raffinement des VMs. Se servir des $\mu_{mci,R/G}^{mco,t}$, qui représentent non seulement des moyennes résiduelles relatives, mais également des données rassemblées lors du transcodage effectif de la trame passée devrait avoir l'avantage d'adapter notre méthode au débit binaire et aux caractéristiques du média à encoder. Utiliser les $\mu_{mci,R/G}^{mco,t}$ comme seuils présente aussi l'avantage d'être continuellement adaptatif, puisque les $\mu_{mci,R/G}^{mco,t}$ sont mis à jour après le transcodage de chaque trame.

4.3 Conclusion

Dans ce chapitre, nous avons présenté des concepts innovateurs utilisés pour pallier aux limitations de la méthode de transcodage MPEG-4 à H.264 proposée au chapitre 3 ainsi que des méthodes proposées par d'autres chercheurs. En définissant une mesure résiduelle de complexité relative des MBs et en exploitant le principe de corrélation de trames successives, nous avons pu définir des outils qui nous permettront de développer une méthode adaptative aux débits binaires et aux caractéristiques vidéos. Ces outils, comme on le verra au chapitre suivant, nous permettront d'atteindre une bonne qualité de transcodage tout en maintenant des accélérations intéressantes. Nous avons également pu valider les hypothèses sur lesquelles reposent nos idées par des simulations et des outils statistiques tels que l'ANOVA. Nous pouvons donc présenter, dans le chapitre suivant, le nouvel algorithme de transcodage MPEG-4 à H.264 proposé. Dans ce chapitre, nous explicitons en détail l'exploitation et l'application de nos idées innovatrices aux processus de détermination des MCs et des VMs.

CHAPITRE 5

ALGORITHME DE TRANSCODAGE MPEG-4 À H.264 BASÉ SUR LA SOMME ABSOLUE RELATIVE DU RÉSIDUEL

Dans ce chapitre, nous présentons un algorithme de transcodage qui tire avantage des outils et propriétés présentés dans le chapitre précédent. L'algorithme sera notamment basé sur la somme absolue relative du résiduel et exploitera le principe de corrélation entre des trames successives, en se servant des statistiques $\mu_{mci,R/G}^{mco,t}$ rassemblées durant le transcodage de la trame passée. Pour finir, afin d'améliorer la qualité, spécialement à haut débit, de petites partitions seront considérées. Cependant, elles le seront de façon conditionnelle pour de maintenir de bonnes accélérations.

Tout comme l'algorithme dans l'algorithme présenté au chapitre 3, les trames intras seront transcodées par la méthode en cascade afin d'obtenir la meilleure qualité visuelle, car l'impact de ce type de trames sur la qualité visuelle est important. Par ailleurs, tel qu'expliqué dans ce chapitre, le choix d'un transcodage en cascade des trames intras devrait avoir un impact négligeable sur la complexité totale de la séquence vidéo. Nous supposons toujours que le type des trames H.264 est le même que celui des trames MPEG-4 entrant.

Le chapitre est organisé comme suit. Dans un premier temps, nous présentons le processus de détermination du MC H.264 des trames inter selon chaque MC d'entrée MPEG-4. Ensuite, nous présentons le processus de détermination des VMs de chaque MC candidat et terminons par une conclusion.

5.1 Détermination du mode de codage des macroblocs

Pour réduire la complexité du processus de détermination du MC H.264, tout comme expliqué dans le chapitre 4.2, nous proposons de réduire le nombre de MCs candidats en nous servant de la distribution fréquentielle des MCs MPEG-4, de la somme absolue relative du résiduel et du principe de corrélation des trames successives pour classer les MCs d'entrée. Une fois cette classification effectuée, nous pourrions déterminer les MCs les moins probables et les

éliminer de l'ensemble des MCs à tester. On rappelle que la distribution fréquentielle des MCs de sortie en fonction du MC d'entrée MPEG-4 est exploitée en établissant une table de correspondance (une telle table a déjà été présentée dans le chapitre précédent. Toutefois, étant donné que cette dernière est un outil essentiel dans l'algorithme de détermination du MC que nous allons présenter et pour éviter au lecteur un retour en arrière incessant, nous la présentons à nouveau dans le tableau 5.1). Nous pouvons constater que les sous MCs Inter8×4, Inter4×8 et Inter4×4 sont très peu probables quel que soit le débit de codage. On peut donc d'ores et déjà les éliminer des MCs à considérer.

Tableau 5.1 Moyenne de fréquence d'apparition des MCs H.264 en fonction des MCs MPEG-4 pour plusieurs types de vidéos QCIF transcodées à 32 kb/s et 160 kb/s en H.264, en utilisant le codeur H.264 d'Intel. La séquence MPEG-4 est initialement encodée à 200 kb/s

Les statistics ont été rassemblées pour plusieurs types de vidéos QCIF et CIF à 32kb/s

Modes de codage H.264 / Modes de codage MPEG-4	Intra 4×4	Intra 16×16	Skip	Inter 16×16	Inter 16×8	Inter 8×16	Inter 8×8	sb8×8	sb8×4	sb4×8	sb4×4
Intra	64,2%	26,1%	7,8%	1,6%	0,1%	0,1%	0,1%	100,0%	0,0%	0,0%	0,0%
Inter16×16 (64%)	0,3%	1,9%	66,2%	27,7%	2,0%	1,8%	0,1%	99,8%	0,1%	0,1%	0,0%
Inter8×8 (21%)	2,9%	3,8%	21,7%	39,5%	14,9%	16,8%	0,4%	98,1%	0,5%	0,7%	0,7%
Skip (15%)	0,1%	0,6%	92,3%	6,8%	0,1%	0,1%	0,0%	0,0%	0,0%	0,0%	0,0%

Les statistics ont été rassemblées pour plusieurs types de vidéos QCIF et CIF à 160kb/s

Modes de codage H.264 / Modes de codage MPEG-4	Intra 4×4	Intra 16×16	Skip	Inter 16×16	Inter 16×8	Inter 8×16	Inter 8×8	sb8×8	sb8×4	sb4×8	sb4×4
Intra	73,8%	21,6%	1,8%	1,9%	0,3%	0,2%	0,4%	100,0%	0,0%	0,0%	0,0%
Inter16×16 (64%)	1,0%	2,7%	28,5%	54,9%	6,0%	5,8%	1,1%	98,5%	0,4%	0,4%	0,7%
Inter8×8 (21%)	6,7%	1,0%	1,0%	22,8%	10,4%	16,5%	41,6%	78%	3,8%	6,0%	12,2%
Skip (15%)	0,1%	0,6%	85,7%	12,9%	0,3%	0,2%	0,2%	100,0%	0,0%	0,0%	0,0%

Nous présentons de façon détaillée, dans les sections suivantes, l'ensemble des MCs candidats en fonction de chaque MC MPEG-4 possible, c'est-à-dire pour un MB codé Intra, Skip, Inter16×16 et Inter8×8 en MPEG-4. Les MBs sont classifiés selon le MC MPEG-4 d'entrée afin de tirer avantage de la propriété de probabilité d'occurrence. Ce processus est illustré à

la figure 5.1. On y montre que pour chaque type de MC d'entrée, un transcodage différent est effectué pour identifier le MC H.264 à utiliser.

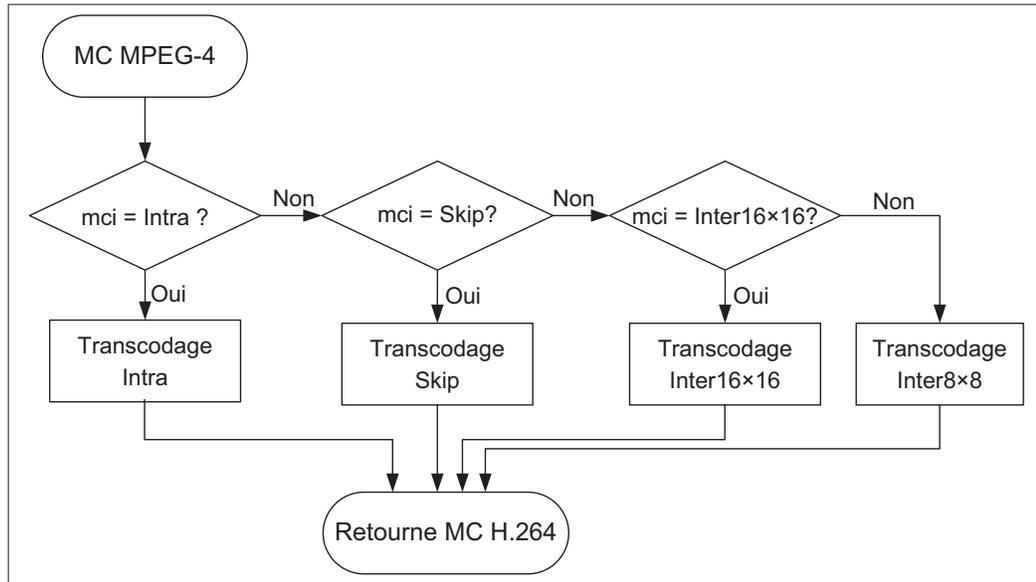


Figure 5.1 Processus de détermination du MC H.264 en fonction du MC d'un MB d'entrée MPEG-4.

5.1.1 Détermination du mode de codage H.264 pour des macroblochs d'entrée MPEG-4 de type Intra

Dans ce cas de transcodage, on utilisera uniquement la propriété de distribution fréquentielle des MCs H.264 en fonction du MC MPEG-4. Comme on peut le constater dans le tableau 5.1, les MBs ont tendance à être réencodés dans les modes Intra16×16 et Intra4×4 (à des taux respectifs de 64,2% et 26,1% à 32 kb/s et de 73,8% et 21,6% à 160 kb/s). Les autres MCs apparaissent rarement. Les modes candidats H.264 sont ainsi Intra4×4 et Intra16×16. L'ensemble des candidats H.264 correspondant à MB codé Intra en MPEG-4, $MCC_{\text{Intra}}^{\text{H.264}}$, est le suivant : $MCC_{\text{Intra}}^{\text{H.264}} = \{\text{Intra4}\times\text{4}, \text{Intra16}\times\text{16}\}$. Le processus de détermination du MC H.264 pour des MBs codés Intras est détaillé à la figure 5.2.

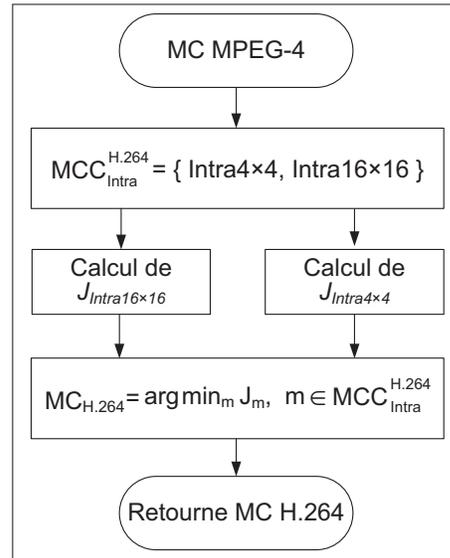


Figure 5.2 Processus de détermination des MCs candidats H.264 et du MC final pour un MB codé Intra en MPEG-4.

5.1.2 Détermination du mode de codage H.264 pour des macroblocs d'entrée MPEG-4 de type Skip

En MPEG-4, le mode Skip signifie que le VM $= (0, 0)$ et qu'aucun résiduel n'est transmis. Les pixels de la trame de référence à la même position spatiale sont tout simplement copiés dans la trame courante. Le fait que le MB ait été codé en Skip en MPEG-4 indique que son VM est près de $(0, 0)$ et que sa SAR est faible. Pour un tel MB, seulement de grandes partitions devraient être considérées en H.264. Le tableau 5.1 le confirme, les MBs codés Skip en MPEG-4 ont fortement tendance à être réencodés en Skip et en Inter16x16. En H.264, toutefois, la définition du mode Skip est différente. H.264 affecte un VM au mode Skip, le vecteur prédit, noté \mathbf{v}_p . Un MB sera donc codé en utilisant le mode Skip s'il a une SARR faible et un VM se rapprochant du vecteur prédit, \mathbf{v}_p . Puisque le VM Skip en MPEG-4 $= (0, 0)$, le MB sera codé Skip en H.264 si \mathbf{v}_p est proche de zéro. Autrement, ce dernier sera codé en Inter16x16. L'ensemble des MCs à considérer, $MCC_{\text{Skip}}^{\text{H.264}}$, est le suivant :

$$MCC_{\text{Skip}}^{\text{H.264}} = \left\{ \begin{array}{ll} \text{Skip} & \text{si } |\mathbf{v}_p| \leq 1 \\ \text{Inter16x16} & \text{si } |\mathbf{v}_p| > 1 \end{array} \right\}$$

Où $|\mathbf{v}_p| \leq 1$ signifie que $|v_{px}| \leq 1$ et $|v_{py}| \leq 1$, avec $\mathbf{v}_p = (v_{px}, v_{py})$. Le processus de détermination des MBs Skip est détaillé à la figure 5.3. Notez que $|\mathbf{v}_p| \leq 1$ signifie que l'on ne diverge pas du vecteur $(0,0)$ de plus de 1 pixel entier ni horizontalement, ni verticalement.

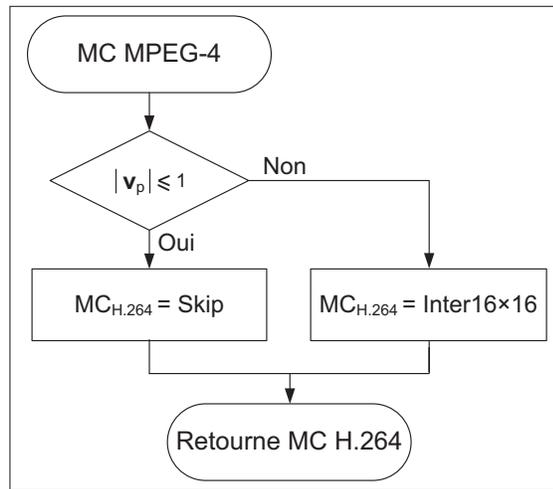


Figure 5.3 Processus de détermination des MCs candidats H.264 et du MC final pour un MB codé en Skip en MPEG-4.

5.1.3 Détermination du mode de codage H.264 pour des macroblocs d'entrée MPEG-4 de type Inter16x16

À la figure 5.4, nous présentons le processus de détermination des MCs candidats pour un MB d'entrée Inter16x16. Expliquons et détaillons maintenant ce processus.

Dans ce cas de transcodage des MBs Inter16x16, nous proposons de classer les MBs en trois classes selon leur complexité s'échelonnant de faible, moyenne à haute afin de réduire efficacement le nombre de MCs candidats. Pour classer les MBs, deux seuils générés à partir de statistiques rassemblées sur la SARR durant le transcodage de la trame passée, $\mu_{16 \times 16, R/G}^{\text{Skip}, t-1}$ and $\mu_{16 \times 16, R/G}^{16 \times 16, t-1}$ seront utilisés.

Étant donné que l'algorithme utilise des statistiques rassemblées lors du transcodage d'une trame passée, nous ferons une distinction entre les trames pour déterminer l'ensemble des MCs candidats. Dans le cas où la trame suit immédiatement une trame intra, nous n'avons

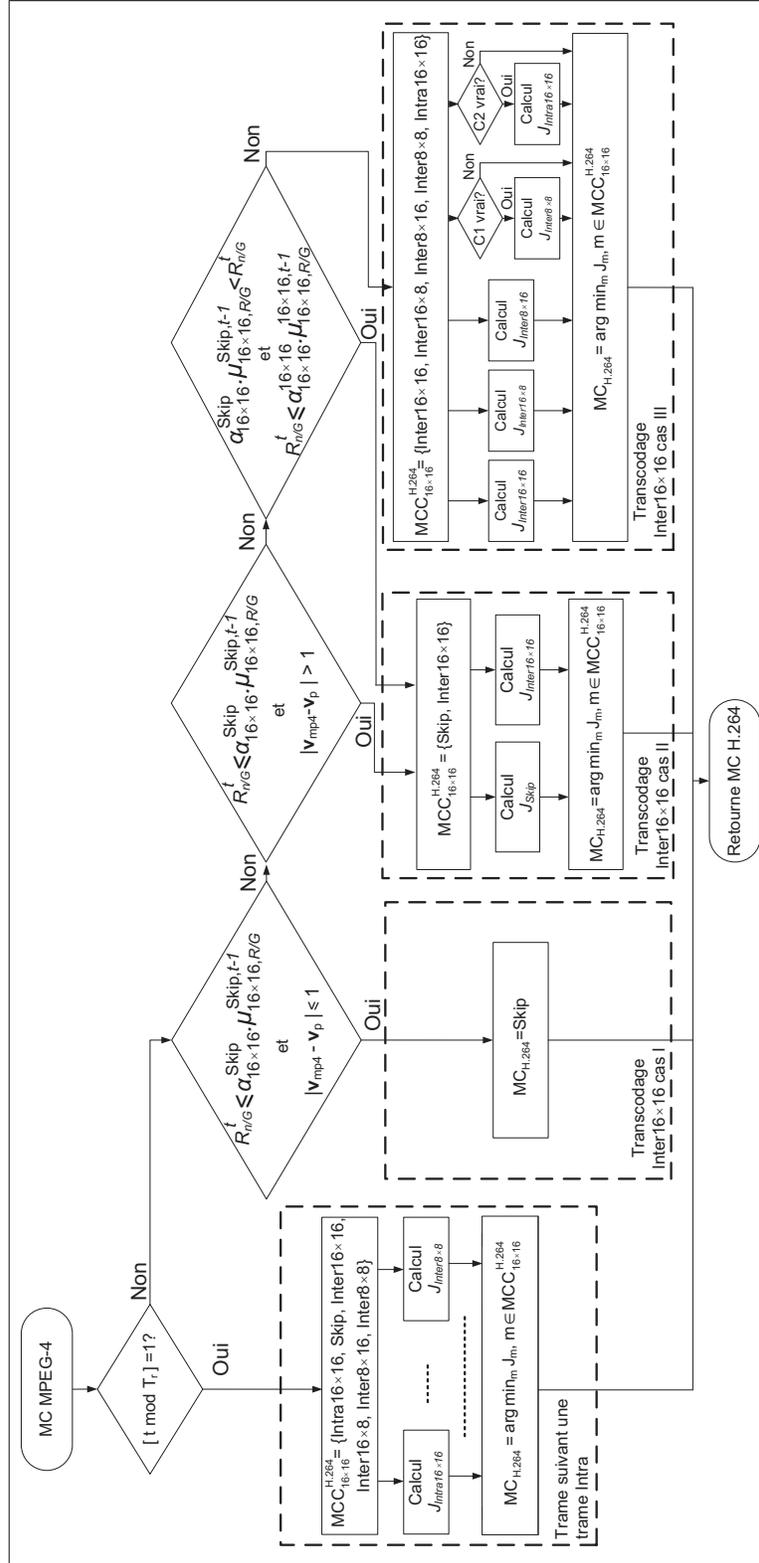


Figure 5.4 Processus de détermination des MCs candidats H.264 et du MC final pour un MB codé Inter16x16 en MPEG-4.

aucune statistique disponible. L'ensemble de MCs à considérer sera donc élargi pour obtenir une bonne qualité et pour également obtenir des données statistiques de départ fiables qui serviront dans le transcodage de la trame suivante. Pour cas des autres trames (c.-à-d. celles que ne suivent pas immédiatement une trame intra), les MBs seront classifiés en trois classes avec des seuils.

Soit t , la trame courante à transcoder, avec $t \in [0, T - 1]$, où T représente le nombre total de trames de la séquence vidéo. Soit T_r , le taux de rafraîchissement, c'est-à-dire le nombre de trames entre deux trames intra. Pour une trame intra, $[t \bmod T_r] = 0$. Pour les autres types de trames, $[t \bmod T_r] > 0$. Pour une trame dont $[t \bmod T_r] = 1$, c'est-à-dire pour une trame suivant une trame intra, nous disions tout juste que le nombre de MCs candidats est élargi. L'ensemble à considérer est le suivant : $MCC_{16 \times 16}^{H.264} = \{\text{Intra}16 \times 16, \text{Skip}, \text{Inter}16 \times 16, \text{Inter}16 \times 8, \text{Inter}8 \times 16, \text{Inter}8 \times 8\}$. Nous avons délibérément ignoré le cas Intra4×4, car les MBs sont codés en MPEG-4 en utilisant une large partition, Inter16×16. De plus, l'Intra4×4 apparaît rarement et nous considérons déjà l'Intra16×16 dans l'ensemble des candidats.

Pour une trame dont $[t \bmod T_r] > 1$, c'est-à-dire pour toutes les autres trames non intra et qui ne suivent pas immédiatement une trame intra, nous classifions les MBs en trois classes, Inter16×16 cas I à Inter16×16 cas III, comme suit :

- **Inter16×16 cas I.** Comme on peut le constater dans le tableau 5.1, bon nombre de MBs MPEG-4 Inter16×16 sont réencodés dans le mode Skip (66,2% à 32 kb/s et 28,5% à 160 kb/s). Cela est dû au fait que H.264 n'a pas la même définition que MPEG-4 du mode Skip, tel que nous l'expliquions précédemment. Un MB qui suit le même mouvement que ses voisins peut être codé en Skip si son énergie résiduelle est assez faible. Et donc un MB codé en Inter16×16 en MPEG-4 aurait très bien pu l'être en Skip si le standard avait été H.264.

Nous proposons de regrouper dans cet ensemble, les MBs susceptibles d'être reencodés en Skip. La logique est que ce MB devrait avoir une SARR faible (équation 5.1) et un VM d'entrée MPEG-4 (noté \mathbf{v}_{mp4}) qui est similaire au vecteur prédit H.264, \mathbf{v}_p . La condition de simi-

larité entre le VM MPEG-4 et le vecteur prédit est définie par l'équation 5.2.

$$R_{n/G}^t \leq \alpha_{16 \times 16}^{\text{Skip}} \cdot \mu_{16 \times 16, R/G}^{\text{Skip}, t-1} \quad (5.1)$$

$$|\mathbf{v}_{\text{mp4}} - \mathbf{v}_p| \leq 1 \quad (5.2)$$

$|\mathbf{v}_{\text{mp4}} - \mathbf{v}_p| \leq 1$ signifie que $|v_{\text{mp4}x} - v_{px}| \leq 1$ et $|v_{\text{mp4}y} - v_{py}| \leq 1$.

Un MB appartient à l'ensemble Inter16×16 cas I s'il remplit la condition décrite dans l'équation 5.3. Dans ce cas on considère un seul candidat, soit $\text{MCC}_{16 \times 16}^{\text{H.264}} = \{\text{Skip}\}$.

$$R_{n/G}^t \leq \alpha_{16 \times 16}^{\text{Skip}} \cdot \mu_{16 \times 16, R/G}^{\text{Skip}, t-1} \text{ et } |\mathbf{v}_{\text{mp4}} - \mathbf{v}_p| \leq 1 \quad (5.3)$$

$\alpha_{16 \times 16}^{\text{Skip}}$ est un coefficient d'ajustement qui nous permet de contrôler le rapport qualité/vitesse (Q/S). Par la suite, $\alpha_{mci}^{\text{mco}}$ désignera le paramètre de contrôle du rapport Q/S associé au seuil $\mu_{mci, R/G}^{\text{mco}, t-1}$.

• **Inter16×16 cas II.** Nous proposons de réunir dans cet ensemble les MBs de faible à moyenne complexité de mouvement ou de texture. Puisque de tels MBs ont tendance à être réencodés en de larges partitions, nous excluons de la liste des candidats, des MCs à petites partitions, sans que cela n'ait d'impact considérable sur la qualité. Nous regroupons dans l'ensemble Inter16×16 cas II deux sous-ensembles de MBs.

a. Le premier, rassemble les MBs dont $R_{n/G}^t$ est faible, mais qui suit un mouvement différent de ses voisins, $|\mathbf{v}_{\text{mp4}} - \mathbf{v}_p| > 1$, contrairement à l'ensemble Inter16×16 cas I dont le mouvement est semblable à celui de ses voisins. Puisque de tels MBs ont été codés Inter16×16 en MPEG-4, des partitions plus petites que Inter16×16 ne devraient pas être considérées. Ces MBs remplissent la condition 5.4 suivante :

$$R_{n/G}^t \leq \alpha_{16 \times 16}^{\text{Skip}} \cdot \mu_{16 \times 16, R/G}^{\text{Skip}, t-1} \text{ et } |\mathbf{v}_{\text{mp4}} - \mathbf{v}_p| > 1 \quad (5.4)$$

b. Le second sous-ensemble rassemble les MBs à complexité moyenne. L'idée est que si la complexité n'est pas trop élevée, il est avantageux de ne considérer que de grandes partitions, c'est-à-dire des partitionnements pas plus grands que 16×16 . Notre raisonnement est confirmé par les statistiques rassemblées dans le tableau de correspondance des MCs H.264 et MPEG-4, selon les trois ensembles Inter 16×16 (Inter 16×16 cas I, Inter 16×16 cas II et Inter 16×16 cas III, qui est explicité dans le paragraphe suivant). On peut constater dans le tableau 5.2 qu'effectivement les modes à petites partitions apparaissent assez faiblement pour ce cas, même pour des débits élevés (à 4,2%, 4,1% et 1,1% à 160 kb/s respectivement pour Inter 16×8 , Inter 8×16 et Inter 8×8). Un MB appartient au second sous-ensemble Inter 16×16 cas II s'il remplit la condition 5.5 suivante :

$$\alpha_{16 \times 16}^{\text{Skip}} \cdot \mu_{16 \times 16, R/G}^{\text{Skip}, t-1} < R_{n/G}^t \leq \alpha_{16 \times 16}^{16 \times 16} \cdot \mu_{16 \times 16, R/G}^{16 \times 16, t-1} \quad (5.5)$$

Pour les deux sous-ensembles constituant le cas Inter 16×16 cas II, les MCs candidats sont donc réduits à Skip et Inter 16×16 ; nous faisant sauver énormément de calculs. Nous avons ainsi $MCC_{16 \times 16}^{\text{H.264}} = \{\text{Skip}, \text{Inter}16 \times 16\}$. À la figure 5.5, nous résumons le processus de détermination des MCs H.264 à tester pour l'ensemble Inter 16×16 cas II.

- **Inter 16×16 cas III.** Nous proposons de regrouper dans cet ensemble les MBs à somme résiduelle élevée (zones à contraste élevé, zones blanchâtres ou noires dans la figure 4.6). Une somme résiduelle élevée signifie que la prédiction du MB en MPEG-4 n'était pas bonne, elle ne le sera probablement pas pour H.264 non plus. Des partitions plus petites pourraient alors nous donner une meilleure prédiction et conséquemment une meilleure qualité. Nous proposons les MCs suivants comme candidats : $MCC_{16 \times 16}^{\text{H.264}} = \{\text{Inter}16 \times 16, \text{Inter}16 \times 8, \text{Inter}8 \times 16, \text{Inter}8 \times 8, \text{Intra}16 \times 16\}$. Bien que Inter 16×16 représente une large partition, il est considéré, car les contraintes de faible débit peuvent faire de lui le meilleur MC. On peut d'ailleurs le constater dans le tableau 5.2 où ce mode apparaît en pourcentage assez considérable (62,5% à 32 kb/s et 64,6% à 160 kb/s) . Un MB appartient à l'ensemble Inter 16×16 cas III si la

Tableau 5.2 Moyenne de fréquence d'apparition des MCs H.264 selon la classification proposée des MBs Inter16×16 en trois classes Inter16×16 cas I, II et III. Les moyennes ont été faites sur plusieurs types de vidéos QCIF transcodées à 32 kb/s et 160 kb/s en H.264, en utilisant le codeur H.264 d'Intel. La séquence MPEG-4 est initialement encodée à 200 kb/s

Les statistics ont été rassemblées pour plusieurs types de vidéos QCIF et CIF à 32kb/s

Modes de codage MPEG-4 \ Modes de codage H.264	Intra 4×4	Intra 16×16	Skip	Inter 16×16	Inter 16×8	Inter 8×16	Inter 8×8	sb8×8	sb8×4	sb4×8	sb4×4
Inter16×16 (64%)	0,4%	2,9%	63,6%	29,0%	2,0%	1,9%	0,2%	100,0%	0,0%	0,0%	0,0%
Inter16×16 cas I (12%)	0,0%	1,1%	92,3%	6,3%	0,2%	0,1%	0,0%	100,0%	0,0%	0,0%	0,0%
Inter16×16 cas II (41%)	0,3%	2,5%	67,3%	26,4%	1,6%	1,7%	0,2%	100,0%	0,0%	0,0%	0,0%
Inter16×16 cas III (11%)	0,9%	5,0%	13,9%	62,5%	9,0%	8,4%	0,3%	100,0%	0,0%	0,0%	0,0%

Les statistics ont été rassemblées pour plusieurs types de vidéos QCIF et CIF à 160kb/s

Modes de codage MPEG-4 \ Modes de codage H.264	Intra 4×4	Intra 16×16	Skip	Inter 16×16	Inter 16×8	Inter 8×16	Inter 8×8	sb8×8	sb8×4	sb4×8	sb4×4
Inter16×16 (64%)	0,5%	2,0%	29,2%	54,0%	6,7%	6,5%	1,1%	98,2%	0,5%	0,5%	0,8%
Inter16×16 cas I (10%)	0,0%	0,9%	79,2%	17,9%	1,2%	0,7%	0,1%	100,0%	0,0%	0,0%	0,0%
Inter16×16 cas II (29%)	0,2%	2,6%	34,8%	53,0%	4,2%	4,1%	1,1%	99,3%	0,3%	0,4%	0,0%
Inter16×16 cas III (25%)	1,2%	1,3%	7,3%	64,6%	12,1%	12,3%	1,2%	95,4%	0,9%	1,2%	2,5%

condition 5.6 suivante est remplie.

$$R_{n/G}^t > \alpha_{16 \times 16}^{16 \times 16} \cdot \mu_{16 \times 16, R/G}^{16 \times 16, t-1} \quad (5.6)$$

Nous considérons les modes Inter8×8 et Intra16×16 bien que ces derniers apparaissent rarement (1,1% et 3,7% au total pour l'Intra16×16 à 160 kb/s). Bien que rares, ces modes jouent un rôle important sur la qualité spécialement pour des séquences à mouvements complexes. Toutefois, il faut se rappeler que MPEG-4 n'a pas choisi Inter8×8 pour le MB en question et qu'ainsi, ce mode ne devrait être considéré que pour des cas particuliers. Ainsi, ces deux modes seront considérés uniquement pour des MBs très complexes (c'est-à-dire ceux avec une SARR élevée), dans le but d'améliorer la qualité sans toutefois augmenter considérablement le temps de transcodage. Les statistiques $\mu_{16 \times 16, R/G}^{8 \times 8, t-1}$ serviront de seuils afin de déterminer si un MB a une SARR élevée. L'Inter8×8 et Intra16×16 seront testés si les conditions C1 et C2 sont respectivement respectées. Le mode Intra4×4 a été délibérément ignoré, car les modes intra

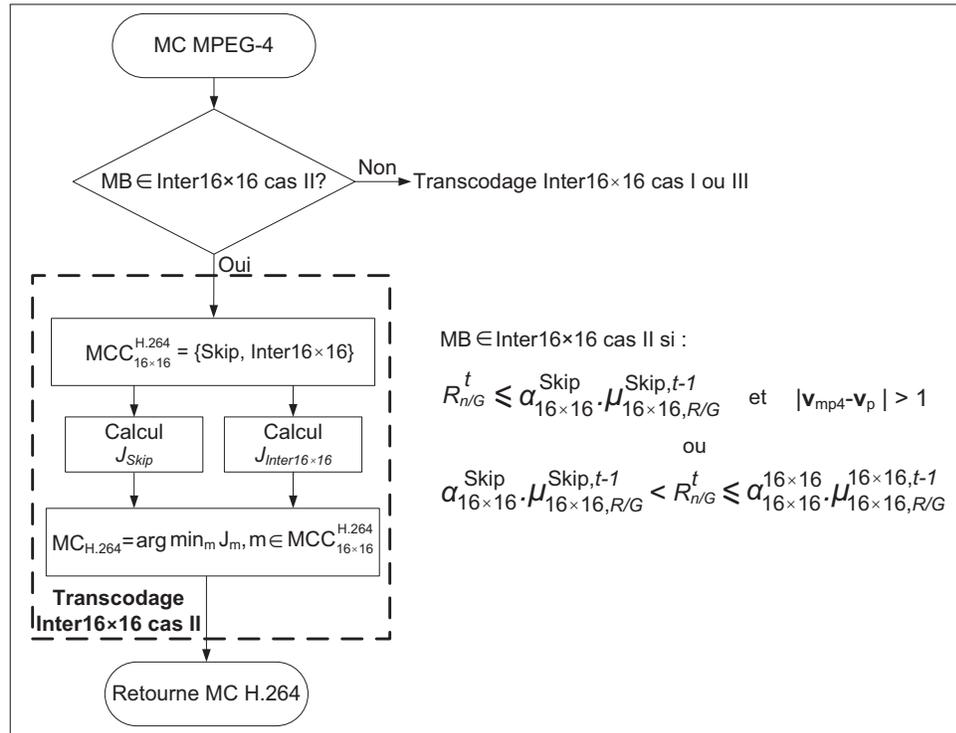


Figure 5.5 Processus de détermination des MCs candidats H.264 et du MC final pour un MB appartenant à l'ensemble Inter16×16 cas II.

apparaissent déjà de façon négligeable.

$$C1 : R_{(n,k)/G}^t > \alpha_{16 \times 16}^{8 \times 8} \cdot \mu_{16 \times 16, R/G}^{8 \times 8, t-1} \quad (5.7)$$

$$C2 : R_{(n,k)/G}^t > \tilde{\alpha}_{16 \times 16}^{8 \times 8} \cdot \mu_{16 \times 16, R/G}^{8 \times 8, t-1} \quad (5.8)$$

Les coefficients $\alpha_{16 \times 16}^{8 \times 8}$ et $\tilde{\alpha}_{16 \times 16}^{8 \times 8}$ sont des paramètres d'ajustement qui nous permettent une marge de manoeuvre sur le facteur Q/S .

À la figure 5.6, nous résumons le processus de détermination des MCs H.264 à tester pour l'ensemble Inter16×16cas III.

Pour résumer, le processus de détermination du MC pour un MB d'entrée MPEG-4 Inter16×16, d'une trame t , s'effectue comme suit :

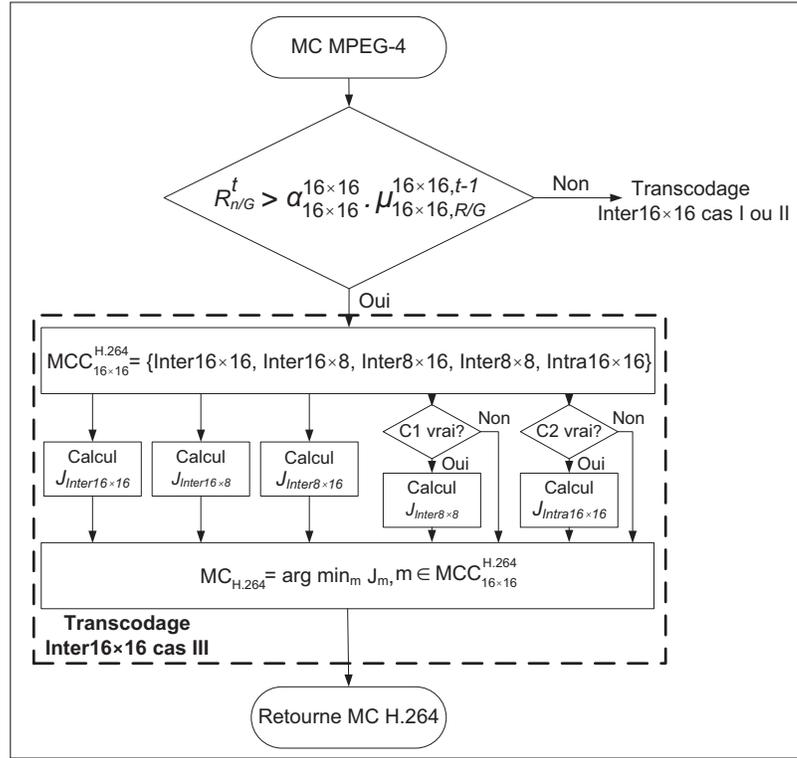


Figure 5.6 Processus de détermination des MCs candidats H.264 et du MC final pour un MB appartenant à l'ensemble Inter16×16 cas III.

- Si $[t \bmod T_r] = 1$ (pour une trame suivant immédiatement une trame intra), $MCC_{16 \times 16}^{H.264} = \{\text{Intra}16 \times 16, \text{Skip}, \text{Inter}16 \times 16, \text{Inter}16 \times 8, \text{Inter}8 \times 16, \text{Inter}8 \times 8\}$.
- Si $[t \bmod T_r] > 1$ (pour une trame qui ne suit pas immédiatement une trame intra), $MCC_{16 \times 16}^{H.264}$ est déterminé selon un des trois cas suivants :

1. Inter16×16 cas I :

$$\text{Si } R_{n/G}^t \leq \alpha_{16 \times 16}^{\text{Skip}} \cdot \mu_{16 \times 16, R/G}^{\text{Skip}, t-1} \text{ et } |\mathbf{v}_{\text{mp4}} - \mathbf{v}_p| \leq 1,$$

$$MCC_{16 \times 16}^{H.264} = \{\text{Skip}\}.$$

2. Inter16×16 cas II :

$$\text{Si } R_{n/G}^t \leq \alpha_{16 \times 16}^{\text{Skip}} \cdot \mu_{16 \times 16, R/G}^{\text{Skip}, t-1} \text{ et } |\mathbf{v}_{\text{mp4}} - \mathbf{v}_p| > 1 \text{ ou}$$

$$\alpha_{16 \times 16}^{\text{Skip}} \cdot \mu_{16 \times 16, R/G}^{\text{Skip}, t-1} < R_{n/G}^t \leq \alpha_{16 \times 16}^{16 \times 16} \cdot \mu_{16 \times 16, R/G}^{16 \times 16, t-1},$$

$$\text{MCC}_{16 \times 16}^{\text{H.264}} = \{\text{Skip}, \text{Inter}16 \times 16\}.$$

3. Inter16×16 cas III :

$$\text{Si } R_{n/G}^t > \alpha_{16 \times 16}^{16 \times 16} \cdot \mu_{16 \times 16, R/G}^{16 \times 16, t-1},$$

$$\text{MCC}_{16 \times 16}^{\text{H.264}} = \{\text{Inter}16 \times 16, \text{Inter}16 \times 8, \text{Inter}8 \times 16, (\text{Inter}8 \times 8 \text{ si C1}), (\text{Intra}16 \times 16 \text{ si C2})\}.$$

$$\text{C1 : } R_{n/G}^t > \alpha_{16 \times 16}^{8 \times 8} \cdot \mu_{16 \times 16, R/G}^{8 \times 8, t-1}$$

$$\text{C2 : } R_{n/G}^t > \tilde{\alpha}_{16 \times 16}^{8 \times 8} \cdot \mu_{16 \times 16, R/G}^{8 \times 8, t-1}$$

$\alpha_{16 \times 16}^{\text{Skip}}$, $\alpha_{16 \times 16}^{16 \times 16}$, $\alpha_{16 \times 16}^{8 \times 8}$ et $\tilde{\alpha}_{16 \times 16}^{8 \times 8}$ sont des coefficients d'ajustement qui nous permettent d'agir sur le rapport Q/S . Dans nos simulations, ces valeurs ont été fixées respectivement à 0,4, 1,5, 1,5 et 2,5. Ces valeurs peuvent toutefois être modifiées en fonction du critère à privilégier (la distorsion ou la vitesse).

Dans le tableau 5.2, nous montrons la distribution fréquentielle des MCs H.264 en fonction de la classification proposée des MBs MPEG-4 d'entrée Inter16×16. Nous pouvons observer que la classification en trois classes, Inter16×16 cas I, II et III nous permet de réduire significativement l'ensemble des MCs H.264 testés, tout en couvrant la plupart des MCs les plus probables.

5.1.4 Détermination du mode de codage H.264 pour des macroblocs d'entrée MPEG-4 de type Inter8×8

À la figure 5.7, nous présentons les MCs H.264 à considérer pour un MB d'entrée MPEG-4 Inter8×8. Nous allons maintenant expliquer et détailler le processus de détermination de ces MCs.

Le fait que le MC Inter8×8 ait été choisi comme meilleur mode en MPEG-4 et non Inter16×16 ou Skip indique que le MB est dans une région à mouvement complexe. Par conséquent, les

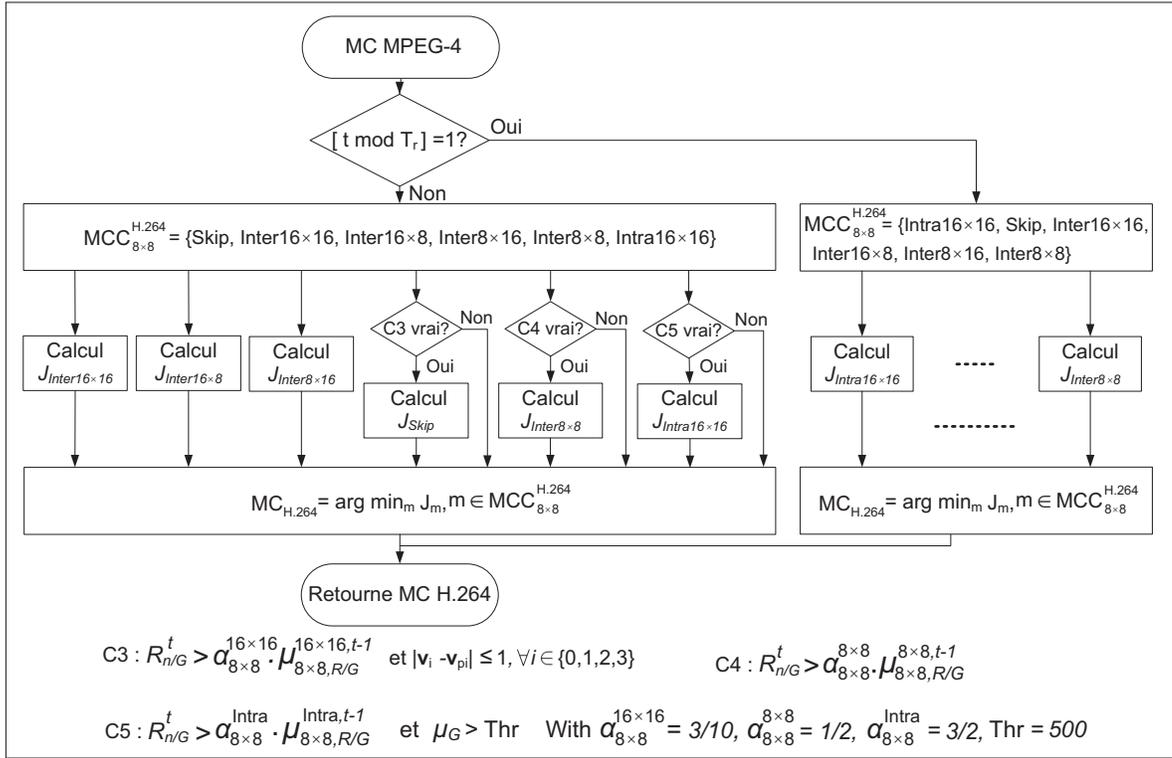


Figure 5.7 Processus de détermination des MCs candidats H.264 et du MC final pour un MB MPEG-4 d'entrée Inter8×8.

MCs Skip et Inter16×16 ne seront normalement choisis en H.264 que pour des raisons de contraintes de débits binaires. Nous pouvons observer cet état de fait dans le tableau 5.1. Les pourcentages d'occurrence de Skip et de Inter16×16 passent respectivement de 1,0% et 22,8% à 160 kb/s à 21,7% et 39,5% à 32 kb/s. Le mode Skip, en particulier, ne devrait pas être considéré à moins que $R_{n/G}^t$ soit faible et que les quatre VMs MPEG-4, \mathbf{v}_{mpi} soient proches du prédicteur H.264 \mathbf{v}_{pi} . Le mode Skip sera choisi si la condition C3 suivante est remplie :

$$C3 : R_{n/G}^t < \alpha_{8 \times 8}^{16 \times 16} \cdot \mu_{8 \times 8, R/G}^{16 \times 16, t-1} \text{ et } |\mathbf{v}_{mpi} - \mathbf{v}_{pi}| \leq 1, \forall i \in \{1, 2, 3, 4\} \quad (5.9)$$

\mathbf{v}_{mpi} étant le MV MPEG-4 associé à la partition i du bloc 8×8 et \mathbf{v}_{pi} le prédicteur H.264 associé à la partition i du bloc 8×8. Le coefficient $\alpha_{8 \times 8}^{16 \times 16}$ a été fixé à 0,3 dans nos simulations.

Pour le MC Inter8×8, les contraintes de débit binaire ont une influence significative sur sa sélection puisque ce mode requiert la transmission de quatre VMs (ce qui peut être ineffi-

cace à bas débits). On peut observer cet état de fait dans le tableau 5.1 où son pourcentage d'occurrence diminue grandement avec la réduction du débit (de 41,6% à 160 kb/s à 0,4% à 32 kb/s). Ainsi, tout comme pour le MC Skip, nous proposons de considérer le MC Inter8×8 uniquement si $R_{n/G}^t$ est élevée, c'est-à-dire si la condition C4 suivante est remplie :

$$C4 : R_{n/G}^t > \alpha_{8 \times 8}^{8 \times 8} \cdot \mu_{8 \times 8, R/G}^{8 \times 8, t-1} \quad (5.10)$$

$\alpha_{8 \times 8}^{8 \times 8}$ est un coefficient d'ajustement du rapport Q/S qui a été fixé à 0,5 dans nos simulations.

Nous proposons de considérer en permanence les MCs Inter16×8 et Inter16×8 comme candidats, car ces derniers n'existent pas dans le standard MPEG-4 et sont sélectionnés assez souvent comme le montre le tableau 5.1. Finalement le MC Inter16×16 est toujours testé car sa fréquence est également assez significative quel que soit le débit.

Pour le MC Intra, on peut constater qu'il apparaît très peu, autour de 6%. Toutefois, il est utilisé pour coder des MBs assez complexes et nous avons observé expérimentalement que l'ignorer aurait un impact négatif sur la qualité, spécialement dans les médias à mouvements complexes (Football par exemple). Nous proposons ainsi de tester le mode intra uniquement pour des vidéos à mouvements complexes. μ_G , la moyenne de la somme absolue du résiduel sur toute la trame (ASAR), décrite dans l'équation 4.4 servira à déterminer si la trame vidéo est complexe ou non.

Nous avons observé expérimentalement, pour plusieurs vidéos QCIF et CIF que les valeurs de μ_G tournent autour de 70, 450 et 1100 respectivement pour des vidéos à faible, à moyenne et à haute complexité. Pour plus de détails, le lecteur est invité à consulter le tableau IV.1 de l'annexe IV, où sont recensées les valeurs de μ_G pour plusieurs types de vidéos. Nous avons établi un seuil de décision pour la complexité des trames vidéos, Thr, à 500. Toutefois, comme nous ne voulons pas tester le mode intra pour tous les MBs mais uniquement pour ceux qui sont complexes, nous ajoutons une condition supplémentaire sur $R_{n/G}^t$. Nous proposons ainsi de

tester le mode MC intra si la condition C5 suivante est remplie.

$$C5 : R_{n/G}^t > \alpha_{8 \times 8}^{\text{Intra}} \cdot \mu_{8 \times 8, R/G}^{\text{Intra}, t-1} \text{ et } \mu_G > \text{Thr} \quad (5.11)$$

$\alpha_{8 \times 8}^{\text{Intra}}$ est un coefficient d'ajustement du rapport Q/S qui a été fixé à 1,5 dans nos simulations.

En ce qui concerne la détermination des MCs pour les trames t suivant une trame intra, c'est-à-dire pour $(t \bmod T_r) = 1$, les seuils $\mu_{8 \times 8, R/G}^{16 \times 16, t-1}$, $\mu_{8 \times 8, R/G}^{8 \times 8, t-1}$ et $\mu_{8 \times 8, R/G}^{\text{Intra}, t-1}$ ne sont pas connus et donc tous les MCs considérés pour les autres trames sont testés sans condition.

Récapitulons, pour un MB d'entrée MPEG-4 Inter 8×8 , l'ensemble des MCs H.264 candidats, $\text{MCC}_{8 \times 8}^{\text{H.264}}$, d'une trame t , est déterminé de la façon suivante :

$$\text{MCC}_{8 \times 8}^{\text{H.264}} = \begin{cases} \left\{ \begin{array}{l} \text{Skip, Inter}16 \times 16, \text{Inter}16 \times 8, \text{Inter}8 \times 16, \\ \text{Inter}8 \times 8, \text{Intra}16 \times 16 \end{array} \right\}, & \text{si } [t \bmod T_r] = 1 \\ \left\{ \begin{array}{l} (\text{Skip si C3}), \text{Inter}16 \times 16, \text{Inter}16 \times 8, \text{Inter}8 \times 16, \\ (\text{Inter}8 \times 8 \text{ si C4}), (\text{Intra}16 \times 16 \text{ si C5}) \end{array} \right\}, & \text{si } [t \bmod T_r] > 1 \end{cases}$$

Il est important de noter que toutes les constantes utilisées dans notre système ont été déterminées empiriquement par simulations sur des séquences d'entraînement vidéos. Les constantes ont été choisies de sorte à obtenir un bon compromis entre la vitesse et la qualité. Les séquences d'entraînement sont complètement différentes de celles utilisées pour valider les performances des algorithmes développés. Les séquences vidéos d'entraînement sont les suivantes : les séquences QCIF Akiyo, Bridge-close, Coastguard et CIF Container, Mobile et Waterfall.

Nous supposons également que le taux de rafraîchissement des trames intra est le même pour MPEG-4 et H.264.

5.2 Détermination des vecteurs de mouvement

Tel qu'expliqué dans le chapitre 3, les VMs influencent énormément la qualité du transcoding. Selon plusieurs simulations que nous avons effectuées, la qualité vidéo peut s'améliorer de presque 2 dB en passant d'une précision au demi-pixel à une précision au quart de pixel, dépendamment du type de vidéo. Malheureusement, la recherche de VMs est l'une des opérations les plus gourmandes en termes de calculs.

Notre objectif principal étant de diminuer le temps de traitement et de maintenir parallèlement une bonne qualité, nous allons procéder comme dans l'algorithme précédent (Metoevi et Coulombe (2009c)). Nous allons extraire les VMs MPEG-4 d'entrée disponibles au niveau du décodeur MPEG-4 et les réutiliser dans l'étape d'encodage H.264. L'estimation de mouvement effectuée par le codeur MPEG-4 serait une bonne approximation des VMs H.264, puisqu'il s'agit de la même vidéo à réencoder sous différentes contraintes. Ils pourraient ne pas être idéaux à cause de la précision des VMs et du partitionnement plus petit des MCs, mais pourraient néanmoins servir comme point de départ à l'estimation de mouvement H.264. On réduirait ainsi le temps de traitement en ne recommençant pas complètement le processus d'estimation de mouvement.

On rappelle que la précision des VMs H.264 est au quart de pixel tandis que celle de MPEG-4 est au demi-pixel pour le *visual simple profile*, le profil MPEG-4 auquel nous nous intéressons. C'est ce profil qui est supporté par la plupart des applications mobiles 3GPP (2011a); 3GPP2 (2006a); 3GPP (2011b); 3GPP2 (2006b).

Pour améliorer la précision des VMs, nous proposons de les raffiner, du demi au quart de pixel. Le processus de raffinement étant malheureusement assez coûteux en termes de calculs, nous exploiterons encore l'idée innovatrice de raffinement conditionnel au résiduel, explicitée et exploitée dans la section 3.4 du chapitre 3. Les VMs seront raffinés uniquement si nécessaire, c'est-à-dire s'ils sont jugés inefficaces. L'information résiduelle, nous servira de mesure d'efficacité des VMs d'entrée MPEG-4, étant donné le lien étroit entre l'information résiduelle et l'efficacité des VMs (voir section 3.2 du chapitre 3).

Tout comme le processus de détermination des MCs, nous proposons d'utiliser concrètement la SARR comme mesure d'efficacité des VMs. Un MB avec une faible $R_{n/G}^t$ est supposé avoir des VMs MPEG-4 qui représentent efficacement le mouvement de ce MB. Les VMs d'un tel MB ne seront pas raffinés. Par contre, un MB avec une $R_{n/G}^t$ élevée, est supposé ne pas représenter efficacement le mouvement du MB ; ses VMs MPEG-4 seront ainsi raffinés. Tout comme dans le processus de détermination des MCs, les statistiques rassemblées durant le transcodage de la trame passée, $\mu_{8 \times 8, R/G}^{8 \times 8, t-1}$ et $\mu_{16 \times 16, R/G}^{16 \times 16, t-1}$ nous serviront de seuils afin de déterminer si un raffinement de VMs est requis.

Ces seuils présentent l'avantage d'être adaptatifs aux conditions de codage, contrairement à l'algorithme présenté dans le chapitre 3 où deux seuils fixes, $T_{vm}^{8 \times 8}$ et $T_{vm}^{16 \times 16}$, ont été utilisés. Pour tenir compte du débit binaire, les valeurs de ces seuils avaient été abaissées à des valeurs plus petites pour certaines plages moins élevées de débit. $\mu_{8 \times 8, R/G}^{8 \times 8, t-1}$ et $\mu_{16 \times 16, R/G}^{16 \times 16, t-1}$ auront l'avantage d'être adaptatifs aux caractéristiques des séquences à transcoder et au débit binaire puisqu'ils sont déterminés et ajustés lors du transcodage effectif de chaque trame de la séquence.

Il faut préciser que le processus de raffinement conditionnel ne s'applique que dans les cas d'une transition de MCs MPEG-4 à H.264 semblables (Inter16×16 à Inter16×16 ou Inter8×8 à Inter8×8). Dans le cas d'une transition de MCs MPEG-4 à H.264 différents, les VMs sont tous raffinés sans exception, quelle que soit $R_{n/G}^t$.

Il faut noter également que lors du transcodage de trames t suivant immédiatement des trames intra, c'est-à-dire pour $[t \bmod T_r] = 1$, nous ne disposons pas des valeurs $\mu_{8 \times 8, R/G}^{8 \times 8, t-1}$ et $\mu_{16 \times 16, R/G}^{16 \times 16, t-1}$. Tous les VMs sont ainsi raffinés sans exception. Le processus de raffinement décrit dans les sous-sections suivantes ne s'appliquera que pour les trames inter dont $[t \bmod T_r] > 1$.

En proposant un raffinement conditionnel, on pourra éviter des raffinements inutiles à la qualité et coûteux en calculs. Dans le tableau III.1 de l'annexe III, nous avons recensé les pourcentages de VMs raffinés pour différents types de vidéos QCIF et CIF et débits binaires. Comme on peut le constater, le principe de raffinement conditionnel nous permet d'éviter de nom-

breux raffinement. Nous obtenons des pourcentages importants de raffinements évités (20% en moyenne pour des vidéos complexes, et jusqu'à 40% pour des séquences peu complexes). Quand le besoin de raffinement se fait le plus sentir, notamment pour des vidéos à mouvement plus complexe, comme Football ou Soccer, les VMs sont raffinés à un plus grand pourcentage. On réduit ainsi la complexité en calculs, mais en maintenant une excellente qualité, comme on pourra le constater plus loin, dans la section des résultats de simulations.

Un algorithme de recherche en petit diamant (*small diamond*) est utilisé pour le raffinement des VMs du demi au quart de pixel. Dans cet algorithme, la SATD du VM MPEG-4 courant est comparée aux SATD des positions de pixels en haut, en bas, à droite et à gauche. Si la position courante a la valeur SATD minimale, la position actuelle est choisie ; autrement la position actuelle se déplace à la position ayant la SATD minimale et le processus se répète. Plus de détails sur les algorithmes de VMs existants peuvent être trouvés dans le livre Kuhn (1999), dans lequel l'auteur présente une étude comparative complète des algorithmes de recherche de VMs les plus usuels.

La détermination des VMs H.264 à partir de ceux de MPEG-4 est détaillée dans les sous-sections suivantes. Mais rappelons d'abord quelques définitions pour faciliter la lecture :

- \mathbf{v}_i désigne le VM MPEG-4 d'entrée associé à la partition k du MB (nous abrégeons donc \mathbf{v}_{mpi} à \mathbf{v}_i).
- $\mathbf{v}_{mco,k}$ représente le VM de sortie H.264 associé à la partition k du MC H.264 *mco*. Par exemple, $\mathbf{v}_{16 \times 8,k}$ représentera le VM H.264 associé à la partition k du MC Inter16×8. Dans le cas du MC Inter16×16, nous allons omettre le k puisqu'il n'y a qu'une seule partition.
- $\mathcal{F}(\cdot)$ est adopté pour désigner l'opération de raffinement d'un vecteur (on utilise le \mathcal{F} pour *fine-tune* qui veut dire ajustement ou raffinement).

5.2.1 Détermination des vecteurs de mouvement pour un mode MPEG-4 d'entrée Skip

Pour ce type de MB d'entrée, nous avons vu dans la section 5.1.2, que l'ensemble des MC H.264 est le suivant : $\text{MCC}_{\text{Skip}}^{\text{H.264}} = \{\text{Skip}, \text{Inter16} \times 16\}$. Expliquons le processus de détermina-

tion des VMs pour chaque élément de cet ensemble, soit le processus de détermination de \mathbf{v}_p et de $\mathbf{v}_{16 \times 16}$.

- \mathbf{v}_p : correspond au vecteur prédit et est déterminé selon les spécifications du standard H.264.
- $\mathbf{v}_{16 \times 16}$: ce VM est déterminé en recommençant à zéro le processus d'estimation de mouvement, car on ne dispose d'aucun VM de départ MPEG-4. Il faut toutefois noter que la plupart des algorithmes de recherche de VMs commencent d'abord par tester le vecteur nul VM = (0,0) et d'autres VMs probables (par exemple les voisins) et terminent l'estimation si la distorsion est faible.

5.2.2 Détermination des vecteurs de mouvement pour un mode MPEG-4 d'entrée Inter16×16

Dans ce cas de transcodage, nous proposons de procéder comme suit. Dans le cas de trames t suivant une trame intra, pour $(t \bmod T_r) = 1$, les VMs sont tous raffinés. Dans le cas des autres trames inter, c'est-à-dire lorsque $(t \bmod T_r) > 1$, on procède selon le MC H.264. Pour le cas des MCs H.264 candidats différents de l'Inter16×16, le VM d'entrée MPEG-4 est toujours raffiné. Pour le cas du MC H.264 candidat Inter16×16, le VM MPEG-4 est raffiné uniquement si $R_{n/G}^t$ est au dessus d'un seuil, $\mu_{16 \times 16, R/G}^{16 \times 16, t-1}$. Le processus de détermination des VMs est résumé, pour chaque MC H.264 candidat, comme suit :

- **Inter16×16** : $\mathbf{v}_{16 \times 16} = \begin{cases} \mathcal{F}(\mathbf{v}_0) & \text{si } R_{n/G}^t > \alpha_{vm} \cdot \mu_{16 \times 16, R/G}^{16 \times 16, t-1} \\ \mathbf{v}_0 & \text{si } R_{n/G}^t \leq \alpha_{vm} \cdot \mu_{16 \times 16, R/G}^{16 \times 16, t-1} \end{cases}$
- **Inter16×8** : $\mathbf{v}_{16 \times 8, k} = \mathcal{F}(\mathbf{v}_0), k = 0, 1$.
- **Inter8×16** : $\mathbf{v}_{8 \times 16, k} = \mathcal{F}(\mathbf{v}_0), k = 0, 1$.
- **Inter8×8** : $\mathbf{v}_{8 \times 8, k} = \mathcal{F}(\mathbf{v}_0), k = 0, 1, 2, 3$.

α_{vm} est un coefficient d'ajustement qui contrôle le rapport Q/S . Ce coefficient a été fixé 0,2 dans nos simulations. Le processus de détermination des VMs dans ce cas est résumé à la figure 5.8.

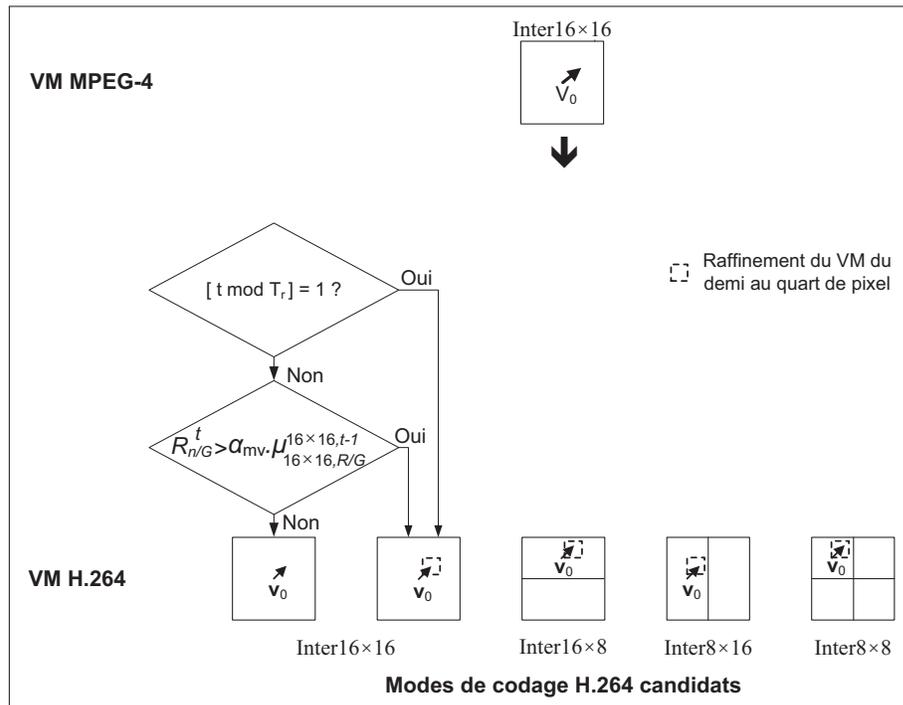


Figure 5.8 Processus de détermination du VM H.264 dans le cas d'un MC MPEG-4 d'entrée Inter16x16.

5.2.3 Détermination des vecteurs de mouvement pour un mode MPEG-4 d'entrée Inter8x8

Pour déterminer les VMs lorsque le mode MPEG-4 d'entrée est Inter8x8, nous allons distinguer deux cas : celui où les VMs sont déterminés pour les MCs candidats H.264 Inter16x16, Inter16x8 et Inter8x16 et celui où les VMs sont déterminés pour le MC H.264 candidat Inter8x8. Voici le traitement de chacun de ces deux cas :

- Inter16x16, Inter16x8 et Inter8x16** : Il existe quatre VMs d'entrée MPEG-4 candidats, v_k , $k = 0, 1, 2, 3$. Pour déterminer les VMs des MCs Inter16x16, Inter16x8 et Inter8x16, nous proposons de réutiliser et de raffiner les VMs d'entrée MPEG-4 candidats puisque ces partitions sont plus larges que la partition d'entrée 8x8. Les VMs à raffiner seront ceux dont le partitionnement est concerné par la partition dont on recherche le VM. Par exemple, pour déterminer le VM du MC H.264 Inter16x16, on considère les quatre VMs MPEG-4 candidats, v_0, v_1, v_2 et v_3 . Pour déterminer le VM de la partition du haut du MC H.264

Inter16×8, on considère les VMs \mathbf{v}_0 et \mathbf{v}_1 comme candidats. Une fois les VMs candidats déterminés, on évalue pour chaque VM candidat, \mathbf{v}_i , le coût du lagragien, $J_{motion}(\mathbf{v}_i)$. Le VM retenu est celui dont le coût du lagragien est minimal. Finalement, ce dernier est raffiné du demi au quart de pixel. Le coût du lagragien lié aux VMs se calcule selon la formule 5.12 et est explicité de façon détaillée à la section 1.8.1.

$$J_{motion}(\mathbf{v}_i) = SA(T)D(\mathbf{v}_i) + \lambda_{motion}R(\mathbf{v}_i - \mathbf{v}_p) \quad (5.12)$$

où λ_{motion} est le multiplicateur du lagragien lié aux VMs et $R(\mathbf{v}_i - \mathbf{v}_p)$ représente le nombre de bits nécessaires pour transmettre les VMs.

Il faut souligner que le processus inverse aurait pu se faire, c'est-à-dire que, les VMs candidats peuvent d'abord passer par le module de raffinement et ensuite d'évaluation du coût. Toutefois cette dernière méthode s'avère plus complexe en calculs que de raffiner seulement le VM candidat retenu. L'étude comparative en termes de complexité et de qualité effectuée sur plusieurs types de vidéos, de débits et de résolutions, nous pousse à retenir la première méthode ; c'est-à-dire celle qui consiste à raffiner uniquement le VM qui présente la distorsion minimale. En effet, on observe dans les résultats présentés en annexe VI dans les tableaux VI.1 à VI.4, que la deuxième méthode est légèrement plus complexe (gains en accélération plus faibles de 7% à 10%) pour une amélioration de la qualité plutôt faible (quasi nulle à haut débit ou haute résolution et environ 0,05 dB à faible débit et faible résolution).

- b. **MC candidat H.264 Inter8×8** : Dans le cas du MC H.264 candidat, Inter8×8, nous utilisons le principe de raffinement selon le résiduel. On ne raffinerait que les VMs dont la SARR est élevée. Si la SARR relative à la partition k , $R_{(n,k)/G}^t$ est supérieure au seuil $\alpha_{vm} \cdot \mu_{8 \times 8, R/G}^{8 \times 8, t-1}$, le VM MPEG-4 d'entrée relatif à la partition k , c'est-à-dire \mathbf{v}_k , est raffiné. Autrement, ce dernier est gardé tel quel. Il faut préciser qu'une telle façon de procéder ne sera faite que pour des trames t qui ne suivent pas une trame intra, c'est-à-dire pour $[(t \bmod T_r) > 1]$.

Lorsque la trame t suit une trame intra, pour $[(t \bmod T_r) = 1]$, on ne dispose pas encore du seuil $\mu_{8 \times 8, R/G}^{8 \times 8, t-1}$. Pour cette raison, les VMs d'entrée sont tous raffinés.

Résumons le processus de détermination des VMs selon le MC à évaluer, pour un MC d'entrée MPEG-4 Inter8×8, avec ses quatre VMs d'entrée \mathbf{v}_k , pour $(k = 0, 1, 2, 3)$:

- **Inter16×16** : $\mathbf{v}_{16 \times 16} = \mathcal{F}(\arg \min_{\mathbf{v}} J_{motion}(\mathbf{v}))$, $\mathbf{v} \in \{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$.
- **Inter16×8** : $\mathbf{v}_{16 \times 8, k} = \mathcal{F}(\arg \min_{\mathbf{v}} J_{motion}(\mathbf{v}))$, $\mathbf{v} \in \{\mathbf{v}_{2k}, \mathbf{v}_{2k+1}\}$, $k = 0, 1$.
- **Inter8×16** : $\mathbf{v}_{8 \times 16, k} = \mathcal{F}(\arg \min_{\mathbf{v}} J_{motion}(\mathbf{v}))$, $\mathbf{v} \in \{\mathbf{v}_k, \mathbf{v}_{k+2}\}$, $k = 0, 1$.
- **Inter8×8** : $\mathbf{v}_{8 \times 8, k} = \begin{cases} \mathcal{F}(\mathbf{v}_k) & \text{si } R_{(n,k)/G}^t > \alpha_{vm} \cdot \mu_{8 \times 8, R/G}^{8 \times 8, t-1} \\ \mathbf{v}_k & \text{si } R_{(n,k)/G}^t \leq \alpha_{vm} \cdot \mu_{8 \times 8, R/G}^{8 \times 8, t-1} \end{cases}, k = 0, \dots, 3.$

avec $\mathcal{F}(\cdot)$ étant la fonction de raffinement.

Le processus de détermination des VMs H.264 pour chaque MC H.264 candidat, pour un MC MPEG-4 d'entrée Inter8×8, est mis en évidence à la figure 5.9. Une fois l'ensemble des candidats et leur VMs respectifs déterminés, l'encodeur H.264 évalue le coût du lagragien, $J_{MC}(m)$, lié à chaque MC candidat m (Inter16×16, Inter16×8, etc.). Le MC sélectionné est celui qui présente le coût minimal. On rappelle que le coût lié au MC est défini comme suit :

$$J_{MC}(m) = D + \lambda_m R \quad (5.13)$$

où D est la distorsion, λ_m le multiplicateur du lagragien et R le nombre de bits nécessaires pour transmettre l'information du MC. On rappelle que des détails sur le coût du lagragien lié au MC peuvent être trouvés à la section 1.8.2.

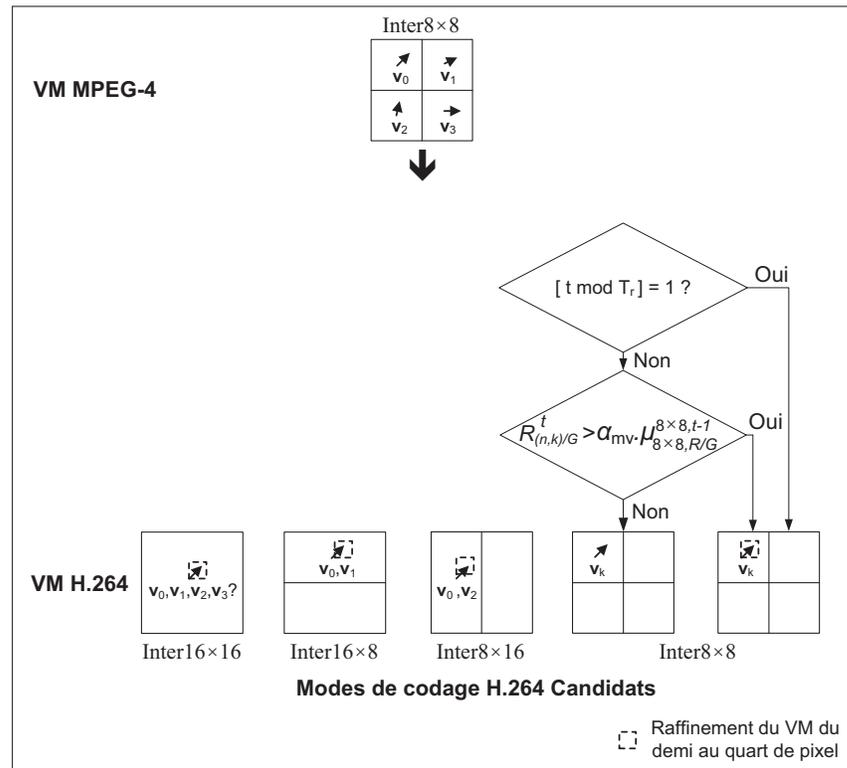


Figure 5.9 Processus de détermination des VMs H.264 dans le cas d'un MC d'entrée MPEG-4 Inter8x8.

5.3 Simulations et résultats obtenus

Des simulations ont été menées afin de tester notre algorithme en termes de qualité, de gains en accélération et de fréquences de trames traitées. L'algorithme proposé, ainsi que ceux les plus performants de l'état de l'art ont été implémentés en utilisant les codecs d'Intel, fournis en codes d'exemples pour la librairie IPP (Intel Performance Primitive), notamment la version 5.3 du code Intel (2008). Les codecs d'Intel sont nettement optimisés en terme de vitesse comparativement aux codecs de référence MPEG-4 (MoMuSys ISO/IEC (2001)) et H.264 (ITU (2003)). Ces derniers sont parfaits pour valider des résultats en termes de débit distorsion, mais ils ne sont pas fiables pour des tests d'accélération, car ils sont très peu optimisés pour la vitesse. Les résultats obtenus, en gain accélération et en fréquences de trames transcodées, sous les codecs d'Intel se rapprochent davantage des attendus avec des codecs commerciaux.

Les simulations ont été réalisées pour plusieurs séquences vidéos, résolutions et débits binaires pour valider nos résultats. Une plateforme HP G62 équipée d'un processeur i5-430M d'Intel à 2,53GHz a été utilisée. Il faut préciser que des résultats similaires ont été obtenus sous d'autres plateformes d'Intel. Au niveau des séquences vidéos, nous avons utilisé différents types de vidéos, notamment des vidéos à mouvements lents, modérément complexes et complexes. Les résolutions testées sont notamment QCIF, CIF, 4CIF (704×576), HD 1280×720 et HD 1920×1080. Certains auteurs ont essayé de catégoriser les vidéos usuelles utilisées pour des tests dans Xiph.org (2011). Les tableaux 5.3 et 5.4 présentent cette catégorisation. Ils pourront nous servir de référence afin d'analyser les résultats obtenus. Les séquences vidéos ont été co-

Tableau 5.3 Propriétés de quelques séquences vidéos tests QCIF et CIF (Xiph.org (2011)) utilisées pour les simulations

Séquences vidéos QCIF et CIF téléchargeables de Arizona State University (2011)	Classes
Akiyo	A : séquences vidéos ayant peu de détails spatiaux et peu de mouvement.
Coastguard	B : séquences vidéos ayant des détails spatiaux moyens et faibles ou du mouvement moyen.
Container	A : séquences vidéos ayant peu de détails spatiaux et peu de mouvement.
Foreman	B : séquences vidéos ayant des détails spatiaux moyens et faibles ou du mouvement moyen.
Hall	A : séquences vidéos ayant peu de détails spatiaux et peu de mouvement.
Mother-Daughter	A : séquences vidéos ayant peu de détails spatiaux et peu de mouvement.
News	B : séquences vidéos ayant des détails spatiaux moyens et faibles ou du mouvement moyen.
Silent	B : séquences vidéos ayant des détails spatiaux moyens et faibles ou du mouvement moyen.

Tableau 5.4 Propriétés de séquences vidéos tests HD utilisées pour les simulations

Séquences vidéos HD 1920×1080 téléchargeables de Xiph.org (2011)	Description des propriétés vidéos
Blue sky	Contraste élevé, de nombreux détails, rotation de caméra.
Mobcal	Beaucoup de détails.
Parkrun	Beaucoup de détails.
Rush-hour	Mouvement lent.
Station	Beaucoup de détails, structures régulières.
Sunflower	Petites différences de couleur, peu de mouvement global.

dées en MPEG-4 sous le profil VSP à 30fps avec une trame intra pour chaque 100 trames inter (c'est-à-dire à chaque 3,3s). Les débits de codage MPEG-4 d'entrée sont 200 kb/s, 720 kb/s, 2,88 Mb/s, 7,2 Mb/s, 16,4 Mb/s respectivement pour QCIF, CIF, 4CIF, HD 1280×720 et HD 1920×1080. Il faut souligner que d'autres débits de codage MPEG-4 initiaux ont été testés avec très peu de différences dans les résultats finaux. Les VMs MPEG-4, aux demi-pixels, ont été trouvés en utilisant l'algorithme logarithmique (Jain et Jain (1981)).

Les conditions de codage H.264 utilisées sont les suivantes : profil baseline, une trame de référence, le critère de distorsion SATD au lieu de la SAD, une efficacité des VMs au quart de pixel trouvés en utilisant l'algorithme de recherche de VMs EPZS (Tourapis (2002)). Aucune trame de type B n'a été utilisée. Les tests ont été réalisés également pour des débits de sortie H.264 faibles, moyens et hauts. Les détails des conditions de transcodage utilisées sont recensés dans le tableau 5.5.

L'algorithme proposé (Proposed, Metoevi et Coulombe (2011b)) est comparé aux algorithmes de l'état de l'art suivant : STAT (la méthode statistique sans raffinement) décrite dans Lee et al. (2006), STAT+REF (la méthode statistique avec raffinement) décrite dans Lee et al. (2006), MV+MS (la méthode de sélection de mode avec raffinement des VMs) décrite dans Liang et al. (2007) et MResidual, la méthode développée précédemment et décrite dans Metoevi et

Tableau 5.5 Conditions de simulations utilisées

Éléments de codage	MPEG-4	H.264
Codage entropique	CAVLC	CABAC
Fréquence de trame (Hz)	30	30
Taux de rafraîchissement des trames intra (s)	3,33	3,33
Algorithme de recherche des VMs	logarithme (Jain et Jain (1981))	EPZS (Tourapis (2002))
Précision des VMs (en pixel)	1/2	1/4
Algorithme de raffinement des VMs	logarithme (Jain et Jain (1981))	<i>small diamond</i>
Trames de type B	Non	Non
Nombre de trames de référence	1	1
Profil	VSP	<i>baseline</i>
<i>Deblocking filter</i>	n'existe pas	non
Méthode de mesure de distorsion pour recherche des VMs	SAD	SATD

Coulombe (2009c). La méthode en cascade (Cascade) est aussi utilisée et sert comme référence pour la qualité et les gains en accélération. On rappelle au lecteur que la méthode en cascade est celle d'un transcodage complet où le signal vidéo est transcodé dans le domaine des pixels puis réencodé complètement selon les spécifications H.264.

Nous avons comparé toutes ces méthodes en termes de qualité (avec le PSNR comme mesure), de gains en accélération ($\frac{T_c}{T_M}$, *speed up*) et de fréquence de trames transcodées (N_{Trames}/T_M (F.T.)). T_c , T_M et N_{Trames} représentent respectivement le temps CPU de la méthode en cascade, celui de la méthode testée et le nombre de trames de la séquence vidéo.

Les résultats des simulations sont présentés dans les figures 5.10 et 5.11 et dans les tableaux 5.6 à 5.9 qui montrent les résultats en PSNR, en accélération (*speed up*) et en F.T. pour les résolutions QCIF, CIF, 4CIF et HD 1920×1080. D'autres résultats de simulations, relatifs à la résolution HD 1280×720, sont présentés à l'annexe V.

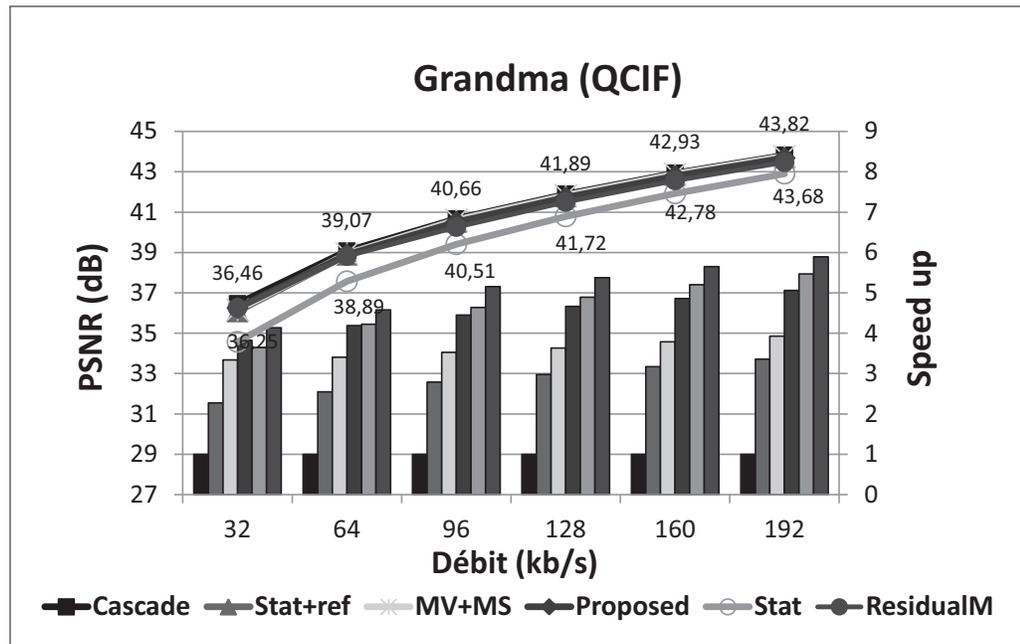


Figure 5.10 Résultats en PSNR et en accélération (*speed up*) pour la vidéo Grandma (QCIF) à débits binaires différents. Les accélérations sont définies comme $T_{cascade}/T_{method}$, avec T représentant les temps CPU. L'ordre des méthodes pour les résultats en accélération est le même que la légende. Les valeurs indiquées sur le graphique sont les PSNR des méthodes Cascade et Proposed respectivement pour les valeurs au dessus et celles en dessous.

Les résultats obtenus sont assez impressionnants. De bons résultats en qualité, spécialement à moyen et hauts débits, sont obtenus pour des gains de l'ordre de $3,42\times$ à $4,76\times$. L'algorithme donne, comparativement à la méthode en cascade, des pertes de qualité à bas débits en moyenne, de $-0,41$ dB, $-0,77$ dB, $-0,87$ dB, $-0,76$ dB et $-1,57$ dB respectivement pour QCIF, CIF, 4CIF, HD 1280×720 et HD 1920×1080 . À moyen et hauts débits les résultats en qualité sont meilleurs, les pertes sont nettement moins élevées. Nous obtenons $-0,16$ dB, $-0,3$ dB, $-0,70$ dB, $-0,6$ dB et $-0,93$ dB à moyens débits et $-0,16$ dB, $-0,3$ dB, $-0,4$ dB, $-0,3$ dB et $-0,75$ dB à haut débits.

Comparativement aux méthodes de l'état de l'art, l'algorithme proposé donne les meilleurs résultats en qualité pour tous les débits binaires de codage. De plus, l'algorithme donne des gains en accélération supérieurs aux méthodes STAT+REF et MV+MS. L'algorithme proposé est de 50 à 65% plus rapide que la méthode STAT+REF et 25% plus rapide que la méthode

Tableau 5.6 Résultats en PSNR, en accélération (*speed up*) et en fréquence de trames (F.T.) pour différents types de vidéos QCIF. Les simulations sont effectuées pour les méthodes Proposed, ResidualM, STAT, STAT+REF et MV+MS. Les résultats en PSNR montrent les différences en qualité avec la méthode cascade

QCIF		32kb/s						64kb/s						96kb/s					
		Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS
Carphone	PSNR (dB)	30,44	-0,69	-0,68	-2,08	-3,96	-5,90	33,60	-0,37	-0,37	-0,84	-2,76	-2,11	35,39	-0,24	-0,43	-0,51	-2,32	-1,19
	SPEED UP	1,00	3,16	3,34	2,07	3,29	3,02	1,00	3,35	3,59	2,23	3,80	2,91	1,00	3,47	3,92	2,35	4,13	2,95
	F.T.	32,24	101,87	107,61	66,78	106,11	97,20	24,98	83,59	89,67	55,60	95,02	72,62	21,33	74,03	83,59	50,07	88,02	62,83
Foreman	PSNR (dB)	29,27	-0,62	-0,64	-1,95	-3,81	-6,93	32,86	-0,24	-0,26	-0,78	-2,85	-2,69	34,76	-0,19	-0,37	-0,52	-2,53	-1,37
	SPEED UP	1,00	3,04	3,35	2,17	3,44	2,91	1,00	3,26	3,56	2,27	3,96	2,90	1,00	3,44	3,85	2,31	4,22	2,92
	F.T.	30,43	92,59	102,04	66,08	104,53	88,50	23,38	76,14	83,33	53,00	92,59	67,87	20,07	69,12	77,32	46,44	84,75	58,59
Grandma	PSNR (dB)	36,46	-0,21	-0,20	-0,38	-1,89	-0,31	39,07	-0,18	-0,25	-0,18	-1,50	-0,13	40,66	-0,15	-0,40	-0,14	-1,25	-0,05
	SPEED UP	1,00	3,82	4,13	2,27	3,65	3,34	1,00	4,19	4,58	2,55	4,23	3,40	1,00	4,45	5,15	2,79	4,64	3,53
	F.T.	33,89	129,46	140,10	77,06	123,58	113,13	27,27	114,17	124,82	69,49	115,23	92,85	23,54	104,82	121,34	65,71	109,16	83,02
Hall	PSNR (dB)	34,33	-0,07	-0,22	-0,78	-1,84	-0,29	38,16	-0,03	-0,15	-0,26	-1,06	-0,08	40,02	-0,11	-0,21	-0,22	-0,81	-0,09
	SPEED UP	1,00	3,50	3,74	1,75	2,86	3,45	1,00	4,03	4,37	2,21	3,60	3,71	1,00	4,33	4,86	2,57	4,17	3,84
	F.T.	39,37	137,61	147,06	68,97	112,78	135,75	29,44	118,58	128,76	64,94	106,01	109,09	24,14	104,53	117,19	62,11	100,67	92,59
Highway	PSNR (dB)	35,05	-0,59	-0,68	-1,57	-2,84	-2,16	37,73	-0,32	-0,49	-0,61	-1,89	-0,90	39,34	-0,26	-0,64	-0,36	-1,53	-0,57
	SPEED UP	1,00	3,47	3,61	1,97	3,11	3,08	1,00	3,93	4,15	2,35	3,85	3,23	1,00	4,24	4,68	2,62	4,35	3,35
	F.T.	35,63	123,46	128,62	70,08	110,86	109,65	26,45	104,06	109,65	62,03	101,73	85,32	22,04	93,55	103,20	57,82	95,92	73,86
Miss-america	PSNR (dB)	40,24	-0,35	-0,55	-1,02	-2,67	-0,89	43,15	-0,19	-0,51	-0,45	-1,83	-0,39	44,58	-0,16	-0,73	-0,32	-1,53	-0,34
	SPEED UP	1,00	3,36	3,61	1,84	2,92	2,63	1,00	3,78	4,02	2,11	3,49	2,64	1,00	3,99	4,60	2,34	3,91	2,75
	F.T.	37,78	127,12	136,36	69,44	110,29	99,34	29,64	111,94	119,05	62,50	103,45	78,13	25,25	100,67	116,28	59,06	98,68	69,44
M-daughter	PSNR (dB)	36,09	-0,33	-0,40	-0,56	-2,15	-1,62	39,14	-0,18	-0,23	-0,24	-1,99	-0,46	40,77	-0,20	-0,65	-0,20	-1,78	-0,27
	SPEED UP	1,00	3,61	3,88	2,43	3,89	3,23	1,00	3,94	4,25	2,59	4,45	3,20	1,00	4,14	4,88	2,72	4,78	3,27
	F.T.	31,58	114,07	122,45	76,73	122,95	102,04	25,04	98,68	106,38	64,94	111,52	80,21	21,96	90,91	107,14	59,76	104,90	71,77
Moyenne PSNR (dB)		34,55	-0,41	-0,48	-1,19	-2,74	-2,59	37,67	-0,22	-0,32	-0,48	-1,98	-0,97	39,36	-0,19	-0,49	-0,32	-1,68	-0,55
Moyenne speed up		1,00	3,42	3,67	2,07	3,31	3,09	1,00	3,78	4,07	2,33	3,91	3,14	1,00	4,01	4,56	2,53	4,31	3,23
Moyenne F.T.		34,42	118,03	126,32	70,73	113,02	106,51	26,60	101,02	108,81	61,79	103,65	83,73	22,62	91,09	103,72	57,28	97,44	73,16

QCIF		128kb/s						160kb/s						192kb/s					
		Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS
Carphone	PSNR (dB)	36,68	-0,22	-0,40	-0,39	-2,00	-0,76	37,71	-0,19	-0,40	-0,31	-1,84	-0,58	38,57	-0,17	-0,39	-0,27	-1,66	-0,42
	SPEED UP	1,00	3,60	4,07	2,44	4,35	3,02	1,00	3,71	4,19	2,52	4,54	3,09	1,00	3,85	4,36	2,63	4,73	3,19
	F.T.	19,14	68,83	77,96	46,70	83,22	57,70	17,52	64,97	73,46	44,21	79,58	54,11	16,18	62,21	70,61	42,54	76,55	51,62
Foreman	PSNR (dB)	36,00	-0,16	-0,32	-0,37	-2,13	-0,85	36,95	-0,11	-0,31	-0,29	-1,86	-0,56	37,75	-0,10	-0,30	-0,21	-1,69	-0,39
	SPEED UP	1,00	3,56	3,99	2,40	4,43	2,97	1,00	3,64	4,11	2,46	4,57	3,03	1,00	3,76	4,21	2,51	4,70	3,08
	F.T.	18,09	64,38	72,12	43,35	80,21	53,67	16,67	60,73	68,49	40,93	76,14	50,51	15,58	58,59	65,50	39,16	73,17	47,92
Grandma	PSNR (dB)	41,89	-0,17	-0,39	-0,11	-1,10	-0,06	42,93	-0,15	-0,36	-0,09	-1,00	-0,04	43,82	-0,14	-0,35	-0,10	-0,91	-0,05
	SPEED UP	1,00	4,66	5,37	2,98	4,90	3,63	1,00	4,86	5,65	3,17	5,20	3,79	1,00	5,06	5,89	3,36	5,47	3,93
	F.T.	21,11	98,42	113,43	62,86	103,33	76,72	19,10	92,85	107,94	60,63	99,43	72,38	17,52	88,59	103,20	58,78	95,81	68,77
Hall	PSNR (dB)	41,23	-0,08	-0,23	-0,14	-0,63	-0,08	42,22	-0,04	-0,22	-0,13	-0,58	-0,04	43,15	-0,06	-0,29	-0,13	-0,64	-0,07
	SPEED UP	1,00	4,59	5,25	2,91	4,63	3,89	1,00	4,85	5,55	3,15	4,99	4,01	1,00	4,94	5,62	3,27	5,11	3,98
	F.T.	20,55	94,34	107,91	59,88	95,24	80,00	18,20	88,24	101,01	57,25	90,91	72,99	17,01	84,03	95,54	55,66	86,96	67,72
Highway	PSNR (dB)	40,55	-0,24	-0,68	-0,23	-1,46	-0,40	41,57	-0,22	-0,70	-0,16	-1,36	-0,26	42,44	-0,24	-0,74	-0,12	-1,32	-0,19
	SPEED UP	1,00	4,49	5,00	2,83	4,74	3,45	1,00	4,72	5,28	3,03	5,04	3,57	1,00	4,92	5,53	3,17	5,29	3,67
	F.T.	19,25	86,43	96,25	54,50	91,16	66,45	17,27	81,60	91,16	52,26	87,03	61,71	15,79	77,70	87,34	50,14	83,51	58,00
Miss-america	PSNR (dB)	45,52	-0,20	-0,75	-0,24	-1,35	-0,28	46,26	-0,18	-0,76	-0,23	-1,21	-0,24	46,98	-0,22	-0,81	-0,29	-1,25	-0,29
	SPEED UP	1,00	4,25	4,90	2,54	4,23	2,84	1,00	4,53	5,15	2,71	4,50	2,94	1,00	4,82	5,55	2,89	4,87	3,09
	F.T.	22,19	94,34	108,70	56,39	93,75	63,03	19,95	90,36	102,74	53,96	89,82	58,59	17,90	86,21	99,34	51,72	87,21	55,35
M-daughter	PSNR (dB)	41,91	-0,20	-0,57	-0,11	-1,61	-0,11	42,84	-0,21	-0,56	-0,16	-1,41	-0,09	43,65	-0,19	-0,47	-0,11	-1,31	-0,06
	SPEED UP	1,00	4,33	5,05	2,86	5,02	3,40	1,00	4,59	5,35	3,03	5,35	3,60	1,00	4,74	5,49	3,17	5,51	3,70
	F.T.	19,85	85,96	100,33	56,71	99,67	67,57	17,87	81,97	95,54	54,15	95,54	64,24	16,66	78,95	91,46	52,82	91,74	61,60
Moyenne PSNR (dB)		40,54	-0,18	-0,48	-0,23	-1,47	-0,36	41,50	-0,16	-0,47	-0,20	-1,32	-0,26	42,33	-0,16	-0,48	-0,18	-1,25	-0,21
Moyenne speed up		1,00	4,21	4,81	2,71	4,61	3,31	1,00	4,41	5,04	2,87	4,89	3,43	1,00	4,58	5,24	3,00	5,10	3,52
Moyenne F.T.		20,03	84,67	96,67	54,34	92,37	66,45	18,08	80,10	91,48	51,91	88,35	62,08	16,66	76,61	87,57	50,12	84,99	58,71

Tableau 5.7 Résultats en PSNR, en accélération (*speed up*) et en fréquence de trames (F.T.) pour différents types de vidéos CIF. Les simulations sont effectuées pour les méthodes Proposed, ResidualM, STAT, STAT+REF et MV+MS. Les résultats en PSNR montrent les différences en qualité avec la méthode en cascade

CIF		128 kb/s						256 kb/s						384 kb/s					
		Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS
Bus	PSNR (dB)	25,55	-0,33	-0,33	-1,25	-3,90	-5,69	28,65	-0,15	-0,13	-0,56	-3,01	-2,05	30,37	-0,10	-0,13	-0,33	-2,51	-1,06
	SPEED UP	1,00	3,24	3,39	2,01	3,29	3,13	1,00	3,51	3,69	2,23	3,79	3,13	1,00	3,72	3,99	2,41	4,18	3,22
	F.T.	7,50	24,31	25,42	15,08	24,63	23,44	5,76	20,22	21,25	12,82	21,83	18,03	4,85	18,03	19,35	11,67	20,27	15,61
Bridge-far	PSNR (dB)	40,09	-0,31	-0,36	-0,76	-1,36	-0,46	41,45	-0,27	-0,39	-0,37	-0,98	-0,43	42,36	-0,26	-0,57	-0,22	-0,83	-0,37
	SPEED UP	1,00	4,34	4,49	2,04	3,21	3,51	1,00	5,04	5,28	2,61	4,15	3,64	1,00	5,34	6,07	2,96	4,64	3,67
	F.T.	8,66	37,57	38,91	17,66	27,81	30,44	6,27	31,60	33,08	16,34	26,01	22,84	5,26	28,09	31,91	15,55	24,41	19,29
Football	PSNR (dB)	26,68	-2,28	-2,50	-4,59	-5,73	-5,11	29,63	-1,60	-1,63	-2,67	-4,04	-5,02	31,42	-1,25	-1,32	-1,89	-3,17	-3,22
	SPEED UP	1,00	3,22	3,31	2,19	3,37	2,76	1,00	3,47	3,63	2,35	3,82	3,38	1,00	3,53	3,80	2,43	4,06	3,51
	F.T.	7,67	24,64	25,37	16,75	25,79	21,12	5,69	19,73	20,63	13,37	21,76	19,26	4,77	16,84	18,12	11,61	19,37	16,73
Foreman	PSNR (dB)	32,03	-0,86	-0,85	-2,28	-4,22	-6,06	35,13	-0,46	-0,48	-0,95	-2,74	-1,99	36,75	-0,32	-0,60	-0,59	-2,21	-1,19
	SPEED UP	1,00	3,39	3,58	2,10	3,36	3,14	1,00	3,65	3,86	2,30	3,90	3,06	1,00	3,86	4,29	2,45	4,30	3,08
	F.T.	7,72	26,16	27,60	16,17	25,93	24,27	5,94	21,69	22,92	13,66	23,15	18,14	5,04	19,47	21,60	12,37	21,65	15,54
Hall	PSNR (dB)	36,48	-0,43	-0,52	-1,57	-2,56	-1,08	39,08	-0,25	-0,44	-0,63	-1,39	-0,56	40,44	-0,21	-0,49	-0,39	-1,05	-0,44
	SPEED UP	1,00	3,44	3,64	1,71	2,83	3,27	1,00	3,83	4,14	2,18	3,61	3,29	1,00	4,12	4,73	2,52	4,17	3,36
	F.T.	9,59	33,04	34,92	16,45	27,15	31,38	6,99	26,79	28,96	15,26	25,21	23,04	5,72	23,57	27,03	14,42	23,83	19,18
Highway	PSNR (dB)	37,32	-0,91	-1,01	-2,18	-3,45	-2,27	39,87	-0,49	-0,76	-0,75	-2,06	-1,05	41,38	-0,39	-0,91	-0,38	-1,62	-0,75
	SPEED UP	1,00	3,63	3,77	1,91	3,03	3,17	1,00	4,08	4,29	2,29	3,69	3,28	1,00	4,45	4,84	2,59	4,22	3,36
	F.T.	8,95	32,50	33,76	17,10	27,13	28,35	6,80	27,71	29,17	15,57	25,12	22,29	5,63	25,07	27,28	14,59	23,78	18,95
Mother-daughter	PSNR (dB)	38,96	-0,28	-0,33	-0,83	-2,07	-1,65	41,95	-0,18	-0,34	-0,38	-1,79	-0,60	43,42	-0,17	-0,77	-0,24	-1,62	-0,36
	SPEED UP	1,00	3,67	3,98	2,33	3,68	3,12	1,00	4,00	4,38	2,55	4,29	3,10	1,00	4,28	5,16	2,74	4,72	3,19
	F.T.	8,13	29,79	32,33	18,93	29,91	25,34	6,40	25,58	28,01	16,29	27,47	19,82	5,49	23,51	28,36	15,05	25,91	17,53
Moyenne PSNR (dB)		33,87	-0,77	-0,84	-1,92	-3,33	-3,19	36,54	-0,49	-0,60	-0,90	-2,29	-1,67	38,02	-0,39	-0,68	-0,58	-1,86	-1,06
Moyenne speed up		1,00	3,56	3,74	2,04	3,25	3,16	1,00	3,94	4,18	2,36	3,89	3,27	1,00	4,19	4,70	2,59	4,33	3,34
Moyenne F.T.		8,32	29,72	31,19	16,88	26,91	26,33	6,26	24,76	26,29	14,76	24,36	20,49	5,25	22,08	24,81	13,61	22,75	17,55

CIF		512 kb/s						640 kb/s						768 kb/s					
		Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS
Bus	PSNR (dB)	31,61	-0,07	-0,10	-0,24	-2,16	-0,70	32,64	-0,08	-0,15	-0,20	-1,89	-0,51	33,52	-0,06	-0,15	-0,15	-1,73	-0,39
	SPEED UP	1,00	3,84	4,14	2,52	4,41	3,25	1,00	3,94	4,31	2,60	4,64	3,32	1,00	4,06	4,44	2,74	4,81	3,40
	F.T.	4,36	16,74	18,07	10,97	19,23	14,15	3,95	15,59	17,05	10,26	18,34	13,11	3,65	14,82	16,22	9,99	17,58	12,44
Bridge-far	PSNR (dB)	43,09	-0,22	-0,61	-0,20	-0,83	-0,37	43,75	-0,23	-0,74	-0,17	-0,89	-0,38	44,35	-0,21	-0,86	-0,19	-1,01	-0,39
	SPEED UP	1,00	5,69	6,40	3,20	5,07	3,69	1,00	5,73	6,40	3,28	5,14	3,59	1,00	5,85	6,49	3,35	5,27	3,53
	F.T.	4,64	26,38	29,66	14,83	23,49	17,10	4,39	25,13	28,07	14,40	22,56	15,74	4,13	24,14	26,78	13,83	21,75	14,57
Football	PSNR (dB)	32,79	-1,04	-1,11	-1,53	-2,70	-2,40	33,92	-0,88	-1,00	-1,28	-2,42	-1,90	34,92	-0,78	-0,93	-1,12	-2,29	-1,65
	SPEED UP	1,00	3,59	3,92	2,55	4,29	3,63	1,00	3,67	4,01	2,66	4,45	3,70	1,00	3,73	4,10	2,75	4,58	3,74
	F.T.	4,18	15,04	16,41	10,68	17,94	15,17	3,79	13,93	15,20	10,07	16,87	14,03	3,50	13,09	14,36	9,64	16,06	13,11
Foreman	PSNR (dB)	37,89	-0,29	-0,60	-0,45	-1,86	-0,84	38,77	-0,24	-0,58	-0,34	-1,64	-0,62	39,54	-0,23	-0,57	-0,29	-1,56	-0,49
	SPEED UP	1,00	4,04	4,49	2,58	4,56	3,16	1,00	4,19	4,63	2,69	4,78	3,22	1,00	4,34	4,83	2,81	4,99	3,30
	F.T.	4,49	18,14	20,13	11,57	20,46	14,16	4,09	17,13	18,93	11,00	19,56	13,18	3,77	16,38	18,21	10,58	18,82	12,45
Hall	PSNR (dB)	41,50	-0,18	-0,47	-0,30	-0,95	-0,35	42,39	-0,16	-0,48	-0,24	-0,88	-0,28	43,20	-0,14	-0,53	-0,18	-0,89	-0,27
	SPEED UP	1,00	4,42	5,11	2,84	4,62	3,51	1,00	4,69	5,44	3,07	5,00	3,59	1,00	4,88	5,68	3,24	5,23	3,71
	F.T.	4,89	21,65	25,02	13,89	22,62	17,19	4,34	20,35	23,58	13,33	21,68	15,59	3,98	19,44	22,62	12,91	20,82	14,79
Highway	PSNR (dB)	42,44	-0,35	-0,84	-0,25	-1,46	-0,58	43,27	-0,35	-0,89	-0,15	-1,32	-0,47	43,95	-0,33	-0,94	-0,11	-1,36	-0,40
	SPEED UP	1,00	4,70	5,14	2,83	4,59	3,42	1,00	4,88	5,35	3,01	4,85	3,45	1,00	5,06	5,56	3,15	5,09	3,49
	F.T.	4,93	23,16	25,36	13,94	22,64	16,88	4,46	21,75	23,84	13,40	21,62	15,38	4,07	20,60	22,62	12,81	20,74	14,21
Mother-daughter	PSNR (dB)	44,47	-0,18	-0,77	-0,19	-1,49	-0,26	45,32	-0,19	-0,74	-0,16	-1,35	-0,23	46,01	-0,19	-0,69	-0,14	-1,24	-0,17
	SPEED UP	1,00	4,61	5,54	2,95	5,16	3,38	1,00	4,91	5,88	3,15	5,54	3,58	1,00	5,17	6,16	3,31	5,86	3,76
	F.T.	4,79	22,09	26,55	14,14	24,71	16,21	4,28	20,99	25,17	13,49	23,70	15,30	3,88	20,04	23,87	12,85	22,73	14,57
Moyenne PSNR (dB)		39,11	-0,33	-0,64	-0,45	-1,64	-0,79	40,01	-0,30	-0,65	-0,36	-1,48	-0,63	40,78	-0,28	-0,67	-0,31	-1,44	-0,54
Moyenne speed up		1,00	4,41	4,96	2,78	4,67	3,43	1,00	4,57	5,15	2,92	4,91	3,49	1,00	4,73	5,32	3,05	5,12	3,56
Moyenne F.T.		4,61	20,46	23,03	12,86	21,59	15,84	4,18	19,27	21,69	12,28	20,62	14,62	3,86	18,36	20,67	11,80	19,79	13,73

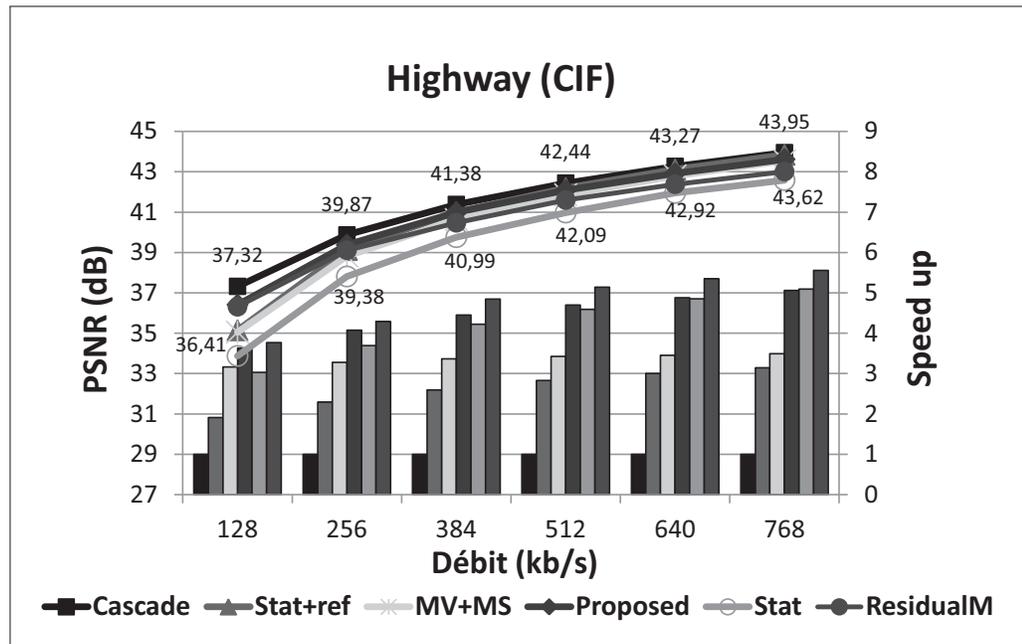


Figure 5.11 Résultats en PSNR et en accélération (*speed up*) pour la vidéo Highway (CIF) à débits binaires différents. Les accélérations sont définies comme $T_{cascade}/T_{method}$, avec T représentant les temps CPU. L'ordre des méthodes pour les résultats en accélération est le même que la légende. Les valeurs indiquées sur le graphique sont les PSNR des méthodes Cascade et Proposed respectivement pour les valeurs au dessus et celles en dessous.

MV+MS. La méthode STAT donne de meilleurs gains en accélération que la notre, mais au prix d'une perte en qualité énorme, allant jusqu'à -4 dB dans certains cas. Une telle perte en qualité pourrait être inacceptable dans la plupart des applications.

La méthode que nous avons décrite dans le chapitre 3 (ResidualM) donne également d'excellents gains en accélération, de $3,3\times$ à $5,68\times$. La dégradation de la qualité est cependant plus élevée que l'algorithme proposé, spécialement à moyen et haut débit. On observe une amélioration de la qualité en moyenne de 0,3 dB avec l'algorithme proposé. Toutefois, en analysant les résultats sur des conditions de test particulières, l'amélioration en qualité de l'algorithme proposé (Proposed), par rapport à la méthode ResidualM peut être nettement plus significative. Cet état de fait s'observe surtout pour des vidéos à faible complexité de mouvement et à hauts débits. C'est par exemple le cas des vidéos QCIF Miss-america, CIF Bridge-Far,

Tableau 5.8 Résultats en PSNR, en accélération (*speed up*) et en fréquence de trames (F.T.) pour différents types de vidéos 4CIF, 704×576. Les sont effectuées pour les méthodes Proposed, ResidualM, STAT, STAT+REF et MV+MS. Les résultats en PSNR montrent les différences en qualité avec la méthode en cascade

4CIF, 704×576		768kb/s						1,02Mb/s						1,53Mb/s					
		Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS
City	PSNR (dB)	33,55	-0,19	-0,16	-0,59	-3,10	-1,50	34,48	-0,13	-0,13	-0,41	-2,62	-0,98	35,78	-0,09	-0,32	-0,20	-2,14	-0,60
	SPEED UP	1,00	3,86	4,07	2,30	3,84	3,09	1,00	4,01	4,23	2,43	4,13	3,02	1,00	4,27	4,80	2,62	4,63	3,01
	F.T.	1,57	6,06	6,38	3,62	6,02	4,85	1,41	5,64	5,95	3,43	5,82	4,26	1,19	5,08	5,71	3,12	5,50	3,57
Crew	PSNR (dB)	34,40	-1,26	-1,30	-2,35	-2,81	-4,47	35,75	-1,09	-1,13	-1,73	-2,27	-3,18	37,60	-0,79	-0,89	-1,15	-1,74	-2,07
	SPEED UP	1,00	3,17	3,20	2,35	3,36	3,20	1,00	3,28	3,34	2,44	3,52	3,15	1,00	3,43	3,64	2,57	3,80	3,17
	F.T.	1,56	4,94	5,00	3,67	5,23	4,99	1,37	4,50	4,58	3,34	4,83	4,33	1,14	3,93	4,16	2,94	4,34	3,62
Harbour	PSNR (dB)	29,72	-0,36	-0,35	-0,95	-3,55	-4,63	30,85	-0,26	-0,26	-0,68	-3,06	-3,20	32,54	-0,19	-0,25	-0,44	-2,51	-1,98
	SPEED UP	1,00	3,67	3,83	2,53	4,21	3,36	1,00	3,81	3,98	2,66	4,60	3,36	1,00	3,91	4,17	2,71	4,93	3,29
	F.T.	1,39	5,08	5,31	3,51	5,82	4,65	1,22	4,65	4,85	3,24	5,60	4,09	1,04	4,09	4,36	2,83	5,15	3,43
Ice	PSNR (dB)	39,86	-0,96	-1,03	-1,68	-3,02	-2,38	41,18	-0,84	-0,93	-1,35	-2,53	-1,75	42,86	-0,67	-0,83	-0,97	-1,90	-1,17
	SPEED UP	1,00	3,15	3,30	1,97	3,14	3,02	1,00	3,30	3,48	2,09	3,36	3,12	1,00	3,56	3,90	2,31	3,73	3,30
	F.T.	2,21	6,96	7,29	4,36	6,95	6,67	1,98	6,53	6,90	4,14	6,65	6,18	1,67	5,95	6,52	3,86	6,23	5,51
Soccer	PSNR (dB)	33,57	-1,59	-1,63	-2,46	-3,33	-3,81	34,74	-1,40	-1,44	-2,02	-2,84	-2,98	36,40	-1,09	-1,22	-1,45	-2,27	-2,13
	SPEED UP	1,00	3,31	3,40	2,15	3,31	3,35	1,00	3,49	3,61	2,33	3,55	3,40	1,00	3,71	4,00	2,57	3,97	3,43
	F.T.	1,67	5,52	5,68	3,59	5,52	5,59	1,44	5,03	5,20	3,36	5,12	4,90	1,18	4,39	4,73	3,04	4,70	4,06
Moyenne PSNR (dB)		34,22	-0,87	-0,89	-1,61	-3,16	-3,36	35,40	-0,74	-0,78	-1,24	-2,66	-2,42	37,04	-0,57	-0,70	-0,84	-2,11	-1,59
Moyenne speed up		1,00	3,43	3,56	2,26	3,57	3,20	1,00	3,58	3,73	2,39	3,83	3,21	1,00	3,78	4,10	2,56	4,21	3,24
Moyenne F,T		1,68	5,71	5,93	3,75	5,91	5,35	1,48	5,27	5,50	3,50	5,60	4,75	1,25	4,69	5,10	3,16	5,18	4,04

4CIF, 704×576		2,04Mb/s						2,56Mb/s						3,07Mb/s					
		Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS
City	PSNR (dB)	36,74	-0,07	-0,33	-0,12	-1,74	-0,40	37,55	-0,04	-0,30	-0,08	-1,47	-0,31	38,24	-0,07	-0,34	-0,03	-1,26	-0,22
	SPEED UP	1,00	4,46	5,04	2,76	4,95	3,06	1,00	4,68	5,31	2,93	5,26	3,17	1,00	4,87	5,54	3,00	5,53	3,25
	F.T.	1,06	4,74	5,36	2,93	5,26	3,25	0,95	4,47	5,07	2,80	5,02	3,02	0,88	4,29	4,88	2,64	4,87	2,87
Crew	PSNR (dB)	38,97	-0,59	-0,71	-0,87	-1,50	-1,56	40,04	-0,44	-0,57	-0,66	-1,28	-1,17	40,97	-0,40	-0,53	-0,56	-1,23	-0,96
	SPEED UP	1,00	3,50	3,71	2,64	3,95	3,19	1,00	3,55	3,78	2,70	4,06	3,22	1,00	3,63	3,86	2,76	4,16	3,25
	F.T.	1,01	3,55	3,76	2,67	4,00	3,23	0,93	3,29	3,49	2,50	3,76	2,97	0,86	3,12	3,31	2,37	3,57	2,79
Harbour	PSNR (dB)	33,78	-0,16	-0,21	-0,33	-2,19	-1,36	34,82	-0,15	-0,24	-0,26	-1,90	-1,03	35,71	-0,16	-0,24	-0,22	-1,93	-0,83
	SPEED UP	1,00	3,98	4,27	2,73	5,17	3,29	1,00	4,05	4,37	2,78	5,35	3,36	1,00	4,17	4,47	2,81	5,41	3,41
	F.T.	0,94	3,74	4,02	2,57	4,86	3,09	0,86	3,49	3,77	2,40	4,62	2,90	0,80	3,35	3,60	2,26	4,35	2,75
Ice	PSNR (dB)	44,07	-0,60	-0,78	-0,81	-1,67	-0,92	45,03	-0,55	-0,75	-0,66	-1,45	-0,74	45,86	-0,51	-0,77	-0,59	-1,40	-0,63
	SPEED UP	1,00	3,74	4,15	2,51	4,03	3,45	1,00	3,91	4,35	2,67	4,29	3,57	1,00	4,07	4,57	2,82	4,53	3,65
	F.T.	1,46	5,47	6,07	3,66	5,89	5,04	1,31	5,12	5,70	3,50	5,62	4,68	1,19	4,84	5,44	3,36	5,39	4,35
Soccer	PSNR (dB)	37,64	-0,87	-1,04	-1,13	-1,94	-1,64	38,63	-0,70	-0,89	-0,87	-1,69	-1,28	39,52	-0,58	-0,80	-0,73	-1,54	-1,06
	SPEED UP	1,00	3,85	4,19	2,72	4,26	3,43	1,00	3,94	4,30	2,83	4,45	3,45	1,00	4,07	4,46	2,95	4,64	3,49
	F.T.	1,04	3,99	4,34	2,82	4,42	3,56	0,94	3,70	4,04	2,66	4,18	3,24	0,86	3,50	3,84	2,54	4,00	3,01
Moyenne PSNR (dB)		38,24	-0,46	-0,61	-0,65	-1,81	-1,18	39,21	-0,38	-0,55	-0,51	-1,56	-0,91	40,06	-0,34	-0,54	-0,43	-1,47	-0,74
Moyenne speed up		1,00	3,91	4,27	2,67	4,47	3,28	1,00	4,03	4,42	2,78	4,68	3,35	1,00	4,16	4,58	2,87	4,85	3,41
Moyenne F.T.		1,10	4,30	4,71	2,93	4,89	3,64	1,00	4,01	4,41	2,77	4,64	3,36	0,92	3,82	4,22	2,64	4,44	3,15

Tableau 5.9 Résultats en PSNR, en accélération (*speed up*) et en fréquence de trames (F.T.) pour différents types de vidéos HD 1920×1080. Les simulations sont effectuées pour les méthodes Proposed, ResidualM, STAT, STAT+REF et MV+MS. Les résultats en PSNR montrent les différences en qualité avec la méthode en cascade

HD, 1920×1080		2,62Mb/s						5,24Mb/s						7,87Mb/s					
		Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS
Blue-sky	PSNR (dB)	36,75	-0,61	-0,53	-0,88	-3,76	-2,54	39,81	-0,46	-0,35	-0,41	-2,42	-0,92	41,40	-0,52	-0,53	-0,32	-1,89	-0,64
	SPEED UP	1,00	3,20	3,31	1,99	3,10	2,69	1,00	3,48	3,62	2,15	3,56	2,62	1,00	3,63	3,96	2,25	3,80	2,65
	F.T.	0,39	1,26	1,30	0,78	1,22	1,06	0,31	1,09	1,14	0,68	1,12	0,82	0,28	1,01	1,10	0,63	1,06	0,74
Pedestrian	PSNR (dB)	39,25	-2,05	-2,06	-3,36	-3,87	-4,59	42,14	-1,32	-1,34	-1,81	-2,40	-2,35	43,57	-0,99	-1,26	-1,21	-1,84	-1,61
	SPEED UP	1,00	3,24	3,24	2,15	3,11	3,23	1,00	3,62	3,70	2,57	3,76	3,44	1,00	3,88	4,22	2,86	4,21	3,54
	F.T.	0,36	1,18	1,18	0,78	1,13	1,18	0,26	0,92	0,95	0,66	0,96	0,88	0,21	0,80	0,87	0,59	0,87	0,73
Riverbed	PSNR (dB)	29,65	-2,83	-3,02	-5,46	-5,58	-5,16	32,50	-2,15	-2,21	-3,25	-3,49	-7,72	34,33	-1,74	-1,81	-2,44	-2,75	-4,73
	SPEED UP	1,00	3,02	2,96	2,71	3,86	2,62	1,00	3,37	3,40	3,01	4,22	3,84	1,00	3,46	3,56	2,94	4,27	3,69
	F.T.	0,30	0,90	0,88	0,81	1,15	0,78	0,20	0,68	0,68	0,60	0,84	0,77	0,16	0,57	0,58	0,48	0,70	0,60
Rush-hour	PSNR (dB)	41,23	-1,77	-1,73	-3,47	-4,27	-5,79	43,76	-0,98	-0,98	-1,33	-2,06	-2,21	45,00	-0,78	-0,96	-0,84	-1,53	-1,43
	SPEED UP	1,00	3,09	3,25	2,06	3,20	3,22	1,00	3,39	3,65	2,35	3,69	3,27	1,00	3,61	4,17	2,61	4,13	3,40
	F.T.	0,37	1,14	1,20	0,76	1,18	1,19	0,26	0,90	0,96	0,62	0,98	0,86	0,21	0,77	0,89	0,56	0,88	0,73
Station	PSNR (dB)	40,57	-0,67	-0,81	-1,35	-3,25	-1,80	42,29	-0,33	-0,51	-0,49	-2,02	-0,68	43,16	-0,28	-1,26	-0,29	-1,56	-0,48
	SPEED UP	1,00	4,04	4,54	2,48	3,89	3,15	1,00	4,92	5,52	3,11	5,08	3,24	1,00	5,45	7,06	3,42	5,78	3,34
	F.T.	0,31	1,26	1,42	0,77	1,22	0,98	0,22	1,09	1,22	0,69	1,12	0,72	0,18	1,00	1,30	0,63	1,06	0,61
Sunflower	PSNR (dB)	41,06	-1,23	-2,06	-1,87	-4,40	-3,44	43,85	-0,73	-1,79	-0,83	-2,92	-1,68	45,20	-0,61	-1,90	-0,52	-2,33	-1,06
	SPEED UP	1,00	3,31	3,95	2,00	3,18	2,91	1,00	3,72	4,53	2,39	3,96	2,77	1,00	4,03	5,17	2,62	4,46	2,75
	F.T.	0,38	1,27	1,51	0,77	1,22	1,11	0,28	1,05	1,27	0,67	1,11	0,78	0,23	0,94	1,21	0,62	1,05	0,65
Tractor	PSNR (dB)	35,31	-3,96	-4,02	-5,60	-6,28	-10,63	37,99	-2,89	-2,94	-3,44	-4,25	-5,93	39,37	-2,16	-2,39	-2,46	-3,27	-3,84
	SPEED UP	1,00	3,23	3,32	2,49	3,71	3,36	1,00	3,65	3,75	2,79	4,20	3,54	1,00	3,76	4,00	2,80	4,33	3,40
	F.T.	0,32	1,04	1,07	0,80	1,19	1,08	0,23	0,82	0,84	0,63	0,95	0,80	0,19	0,73	0,77	0,54	0,84	0,66
Moyenne PSNR (dB)		37,69	-1,87	-2,03	-3,14	-4,49	-4,85	40,33	-1,27	-1,45	-1,65	-2,79	-3,07	41,72	-1,01	-1,44	-1,15	-2,17	-1,97
Moyenne speed up		1,00	3,31	3,51	2,27	3,44	3,03	1,00	3,73	4,02	2,62	4,07	3,24	1,00	3,97	4,59	2,79	4,43	3,25
Moyenne F.T.		0,35	1,15	1,22	0,78	1,19	1,05	0,25	0,93	1,01	0,65	1,01	0,80	0,21	0,83	0,96	0,58	0,92	0,67

HD, 1920×1080		10,49Mb/s						13,12Mb/s						15,74Mb/s					
		Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS
Blue-sky	PSNR (dB)	42,55	-0,55	-0,53	-0,31	-1,69	-0,54	43,45	-0,63	-0,54	-0,26	-1,46	-0,47	44,20	-0,70	-0,54	-0,24	-1,37	-0,43
	SPEED UP	1,00	3,74	4,08	2,33	3,95	2,70	1,00	3,86	4,20	2,42	4,11	2,76	1,00	3,97	4,32	2,49	4,25	2,82
	F.T.	0,25	0,95	1,04	0,59	1,01	0,69	0,23	0,90	0,98	0,57	0,96	0,65	0,22	0,86	0,94	0,54	0,92	0,61
Pedestrian	PSNR (dB)	44,55	-0,80	-1,11	-0,91	-1,56	-1,24	45,32	-0,71	-1,07	-0,75	-1,49	-1,04	45,96	-0,70	-1,10	-0,68	-1,46	-0,91
	SPEED UP	1,00	4,01	4,39	3,04	4,46	3,55	1,00	4,19	4,60	3,18	4,67	3,57	1,00	4,34	4,78	3,30	4,88	3,63
	F.T.	0,18	0,72	0,79	0,55	0,80	0,64	0,16	0,68	0,75	0,52	0,76	0,58	0,15	0,64	0,71	0,49	0,72	0,54
Riverbed	PSNR (dB)	35,77	-1,48	-1,54	-2,06	-2,37	-3,43	36,93	-1,24	-1,29	-1,73	-2,08	-2,69	38,00	-1,14	-1,17	-1,58	-1,93	-2,27
	SPEED UP	1,00	3,41	3,53	2,82	4,22	3,57	1,00	3,42	3,53	2,82	4,30	3,67	1,00	3,42	3,50	2,80	4,25	3,73
	F.T.	0,15	0,50	0,52	0,42	0,62	0,53	0,13	0,46	0,47	0,38	0,57	0,49	0,12	0,43	0,44	0,35	0,53	0,46
Rush-hour	PSNR (dB)	45,80	-0,71	-0,99	-0,65	-1,39	-1,14	46,42	-0,74	-1,05	-0,65	-1,59	-1,04	46,98	-0,78	-1,22	-0,65	-1,60	-0,94
	SPEED UP	1,00	3,79	4,37	2,77	4,36	3,36	1,00	4,03	4,64	2,97	4,75	3,52	1,00	4,07	4,64	2,99	4,71	3,38
	F.T.	0,19	0,71	0,82	0,52	0,82	0,63	0,16	0,66	0,76	0,49	0,78	0,58	0,15	0,63	0,72	0,46	0,73	0,52
Station	PSNR (dB)	43,79	-0,27	-1,11	-0,21	-1,41	-0,36	44,34	-0,28	-1,12	-0,19	-1,34	-0,34	44,87	-0,28	-1,05	-0,18	-1,29	-0,31
	SPEED UP	1,00	5,93	7,57	3,66	6,26	3,52	1,00	6,23	7,91	3,78	6,57	3,64	1,00	6,49	8,26	3,86	6,86	3,77
	F.T.	0,16	0,96	1,23	0,59	1,01	0,57	0,15	0,92	1,17	0,56	0,97	0,54	0,14	0,88	1,12	0,52	0,93	0,51
Sunflower	PSNR (dB)	46,06	-0,59	-1,85	-0,43	-2,28	-0,81	46,76	-0,59	-1,81	-0,39	-2,19	-1,68	47,35	-0,58	-1,68	-0,34	-2,05	-0,57
	SPEED UP	1,00	4,43	5,66	2,81	4,98	2,86	1,00	4,79	5,93	2,98	5,33	2,98	1,00	5,14	6,38	3,18	5,67	3,14
	F.T.	0,20	0,88	1,13	0,56	0,99	0,57	0,18	0,85	1,05	0,53	0,94	0,53	0,16	0,82	1,02	0,51	0,91	0,50
Tractor	PSNR (dB)	40,30	-1,58	-1,81	-1,81	-2,62	-2,77	41,07	-1,18	-1,41	-1,33	-2,15	-2,07	41,75	-0,92	-1,14	-1,04	-1,86	-1,57
	SPEED UP	1,00	3,68	3,90	2,74	4,30	3,37	1,00	3,69	3,91	2,75	4,36	3,34	1,00	3,74	3,94	2,78	4,39	3,37
	F.T.	0,17	0,64	0,68	0,47	0,75	0,58	0,16	0,59	0,63	0,44	0,70	0,54	0,15	0,56	0,59	0,42	0,66	0,51
Moyenne PSNR (dB)		42,69	-0,85	-1,28	-0,91	-1,90	-1,47	43,47	-0,77	-1,18	-0,76	-1,76	-1,19	44,16	-0,73	-1,13	-0,67	-1,65	-1,00
Moyenne speed up		1,00	4,14	4,79	2,88	4,65	3,27	1,00	4,32	4,96	2,99	4,87	3,36	1,00	4,45	5,12	3,06	5,00	3,41
Moyenne F.T.		0,19	0,77	0,89	0,53	0,86	0,60	0,17	0,72	0,83	0,50	0,81	0,56	0,16	0,69	0,79	0,47	0,77	0,52

HD 1920×1080 Sunflower, où l'amélioration de la qualité va respectivement jusqu'à 0,59 dB, 0,65 dB et 1,29 dB.

Cette amélioration notable de la qualité nous permet ainsi d'obtenir des pertes de qualité très intéressantes, presque nulles pour certaines vidéos, par rapport à la méthode en cascade. C'est le cas par exemple des vidéos QCIF Hall, 4CIF City où on atteint des pertes de 0,03 dB.

L'amélioration de la qualité à hauts débits, comparativement à la méthode Residual, se justifie par le fait que la nouvelle méthode considère des MCs à partitions plus petites telles que Inter16×8, Inter8×16 et Inter8×8. Contrairement à la méthode ResidualM qui ne les considérerait pas du tout. On sait que le codage en petites partitions influence particulièrement la qualité à débits élevés qu'à débits faibles (tableau 5.5). Bien qu'elles aient été sélectionnées à certaines conditions, les petites partitions justifient également les gains de vitesse plus faibles obtenus par rapport à la méthode ResidualM.

L'algorithme proposé donne de meilleurs résultats en qualité spécialement pour des vidéos à faible mouvement ou ayant peu de détails spatiaux. C'est le cas des vidéos Mother-Daughter (CIF) qui appartiennent à la classe A ou Sunflower (1920×1080), que le tableau 5.4 décrit comme une vidéo ayant peu de mouvement global.

Cela peut s'expliquer notamment par le fait que l'algorithme ResidualM utilise des seuils fixes. Pour que les VMs soient raffinés, il faut que l'énergie résiduelle soit plus élevée que ces seuils. Toutefois pour des vidéos peu complexes, l'énergie résiduelle est faible dans la trame, car il y a peu de mouvement. Le pourcentage de VMs raffinés peut être ainsi extrêmement faible. Ce qui pose problème, car de nombreuses zones étant statiques, la qualité de la trame dépend grandement de ces zones à faible mouvement. Ces zones devraient donc avoir des VMs précis pour garantir une bonne qualité. Des seuils peu adaptés feraient que le peu de VMs présents dans la trame ne seraient pas raffinés. Cela aurait pour conséquence des VMs peu efficaces, d'où les résultats en qualité moins bons pour les vidéos peu complexes. On peut d'ailleurs le constater à la figure III.2 de l'annexe III. Le pourcentage des VMs raffinés pour des vidéos à faible complexité comme Miss-America, Claire ou Highway est assez faible surtout à hauts

débites. Il tourne autour de 10%, tandis que celui de la méthode Proposed tourne autour de 55%.

Le problème de seuils fixes est résolu dans l'algorithme proposé. En effet, les seuils utilisés pour la classification des MCs et dans le processus de raffinement des VMs sont adaptés aux caractéristiques du média à coder. Les régions avec une somme absolue résiduelle faible pourront ainsi être raffinées si elles sont complexes relativement aux autres MBs de la trame. On peut constater en effet à la figure III.1 de l'annexe III, que le pourcentage de raffinement de ce type vidéos est plus élevé que dans la méthode ResidualM.

Les deux algorithmes proposés dans cette thèse, Proposed et ResidualM donnent d'excellentes accélérations en vitesse tout en donnant de bons résultats en qualité. Ces deux algorithmes se prêtent ainsi aisément au transcodage en temps réel. L'un pourrait être préféré à l'autre dépendamment des exigences de l'application en termes de vitesse et de qualité.

Il faut néanmoins préciser que les résultats expérimentaux présentés en Lee et al. (2006) et Liang et al. (2007) ont été obtenus pour les codecs MoMuSys (ISO/IEC (2001)) et JM (ITU (2003)). Sous un codec plus optimisé, tel que celui d'Intel, les gains en accélérations obtenus pour ces méthodes sont nettement moins impressionnants. Par exemple, en Liang et al. (2007), pour la méthode MV+MS, l'auteur obtenait une moyenne de 10,36× d'accélération en vitesse. Sous les codecs Intel, le gain moyen obtenu est de 3,5×. Les algorithmes que nous avons proposés devraient donner des gains en accélération nettement plus élevés sous les codecs MoMusys et JM. Ces gains ne seront toutefois pas représentatifs de ceux à obtenir sur des produits commerciaux.

5.4 Conclusion

Dans ce chapitre nous avons présenté un nouvel algorithme de transcodage vidéo MPEG-4 à H.264, Metoevi et Coulombe (2011b). L'algorithme exploite les nouveaux outils innovateurs décrits dans le chapitre précédent, notamment la mesure relative et le principe de corrélation de trames successives. L'utilisation de cette mesure nous a permis d'extraire efficacement les

MCs les plus probables et de réduire ainsi considérablement l'ensemble des MCs à tester et, du même coup, la complexité en calculs. L'idée de tester conditionnellement de petites partitions nous a permis d'atteindre d'excellentes qualités de transcodage (perte de l'ordre de 0,16 dB à 0,4 dB pour du QCIF comparativement à la méthode en cascade). L'exploitation du principe de corrélation de trames successives nous a permis également de solutionner une carence générale de l'état de l'art ; celle de ne pas adapter l'algorithme aux conditions de transcodage, notamment du débit binaire et des caractéristiques vidéos. Les statistiques rassemblées lors du transcodage de la trame passée, menant à des seuils dynamiques, nous ont permis d'adapter l'algorithme à ces deux critères (voir aussi brevet provisionnel (Metoevi et Coulombe (2011a))). Pour maintenir de bonnes accélérations de transcodage, nous avons également exploité une de nos idées innovantes qui consiste à raffiner uniquement les VMs jugés inefficaces (Metoevi et Coulombe (2009a)).

En exploitant tous ces outils, nous avons pu nettement améliorer la qualité de l'algorithme précédent et atteindre d'excellents résultats en qualité pour plusieurs débits et caractéristiques vidéos. Des accélérations impressionnantes, de l'ordre de $3\times$ à $4,7\times$, sont obtenues pour des pertes en qualité de minimales à raisonnables. Il faut souligner que ces gains de vitesse sont obtenus sur des codecs déjà optimisés en vitesse. Comparativement à l'état de l'art, nous obtenons, pour certaines méthodes, des résultats meilleurs en qualité et en accélération (MV+MS et STAT+REF). Une seule méthode donne des gains de vitesse légèrement meilleurs. Pour cette méthode par contre, la qualité est grandement affectée, jusqu'à 1,9 dB de perte alors que notre algorithme donne 0,39 dB pour ce même cas.

Les idées développées et l'algorithme proposé peuvent s'appliquer également à d'autres types de transcodages vidéos, tels que le transcodage H.263 à H.264. Le même algorithme de transcodage peut s'appliquer à d'autres encodeurs pour déterminer le meilleur ensemble de MCs candidats. Les résultats en qualité et en accélération pourraient être différents, mais devraient donner de bons résultats.

CHAPITRE 6

CONTRIBUTIONS

Les contributions apportées par ce travail de recherche sont les suivantes :

- 1) Innovation dans le domaine du transcodage vidéo MPEG-4 à H.264 en proposant une méthodologie efficace basée sur les modes de codage, les vecteurs de mouvement et **l'information résiduelle** de la source MPEG-4.
- 2) Création d'un processus de décision efficace des MCs H.264. Exploitation de l'information résiduelle (énergie résiduelle ou somme résiduelle absolue), de la disparité des VMs pour catégoriser et segmenter une table statistique de conversion des MCs MPEG-4 à H.264 à l'aide de seuils adaptatifs. On arrive ainsi à des ensembles plus restreints de MCs H.264 candidats.
- 3) Élaboration d'une mesure relative de l'information résiduelle (somme résiduelle absolue relative) utilisée dans le processus de décision des MCs et dans les seuils adaptatifs, dans le but de concevoir une méthode efficace de transcodage adaptative au débit binaire et aux caractéristiques vidéos.
- 4) Proposition d'une utilisation conditionnelle des petites partitions dans le processus de décision des MCs H.264 candidats afin d'obtenir une très bonne qualité à hauts débits tout en maintenant de bonnes accélérations de vitesse. On résout ainsi, parallèlement, les mauvais résultats en qualité, des méthodes existantes, pour les vidéos à faible mouvement à haut débit.
- 5) Innovation dans le processus de raffinement des VMs. Utilisation de l'information résiduelle pour développer une méthode efficace de raffinement conditionnelle, du demi au quart de pixels, des VMs. Réduction considérable de la complexité en calculs du processus de raffinement des VMs.
- 6) Élaboration de seuils d'énergie résiduelle variables en fonction du débit binaire dans le processus de raffinement des VMs afin d'atteindre une bonne qualité de transcodage.

- 7) Exploitation, dans le processus de transcodage, du principe de corrélation de trames successives en rassemblant des statistiques durant le transcodage de la trame passée et en les utilisant comme seuils. On arrive ainsi à une méthode adaptée au débit binaire et aux caractéristiques vidéos et de ce fait à élaborer un algorithme de transcodage, efficace, quelque soient les conditions de transcodage.
- 8) Création de méthodes qui résolvent la carence des méthodes existantes de l'état de l'art face aux débits et aux caractéristiques vidéos. Des méthodes de haute qualité adaptives aux conditions de transcodage ont pu être développées en utilisant des seuils adaptatifs dans les processus de décision des MCs et de raffinement des VMs. Ces seuils tirent avantage du principe de corrélation de trames successives.
- 9) Création d'une méthode capable de contrôler et de sélectionner le compromis désiré entre la vitesse et la qualité.
- 10) Création d'une méthode qui donne un bon compromis entre la vitesse et la qualité. La méthode proposée donne une meilleure qualité et des gains supérieurs pour tous les débits binaires et caractéristiques vidéos, spécialement à bas débits, comparativement aux autres méthodes connues de l'état de l'art.

Ces travaux ont fait l'objet d'un article de conférence (Metoevi et Coulombe (2009c)), d'un article de revue soumis (Metoevi et Coulombe (2011b)), de deux demandes de brevet (Metoevi et Coulombe (2009a)) (Metoevi et Coulombe (2009b)) et d'une demande de brevet provisionnel (Metoevi et Coulombe (2011a)).

CONCLUSION

Dans cette thèse, nous nous sommes intéressés au transcodage MPEG-4 à H.264. Le transcodage MPEG-4 à H.264 est devenu une nécessité, non seulement afin d'assurer la communication entre les terminaux supportant ces deux standards, mais également pour les plateformes qui veulent convertir du contenu MPEG-4 existant à H.264. Ce sujet est donc d'actualité, surtout que l'enjeu économique est énorme, le marché de la vidéo génère actuellement des revenus énormes et devrait croître exponentiellement dans les années futures.

Malheureusement, ce type de transcodage rend la conversion MPEG-4 à H.264 coûteuse pour les applications temps réels à grands volumes car il est réalisé au prix d'une grande complexité de traitement. Nous avons par conséquent développé des algorithmes de transcodage efficaces, dont le but est de réduire la complexité de traitement tout en maintenant une bonne qualité. Deux méthodes de transcodage efficaces ont été développées. Ces méthodes extraient les MCs, les VMs et l'information résiduelle du train binaire MPEG-4 entrant et les réutilisent judicieusement dans le processus d'encodage H.264. Elles réduisent, de ce fait, significativement la complexité en calculs, tout en préservant une excellente qualité (le chapitre 6 présente les contributions détaillées de notre recherche).

Nos méthodes ont été testées et comparées aux méthodes les plus efficaces de l'état de l'art. Nous les avons comparées, en utilisant une cinquantaine de tests vidéos pour plusieurs débits binaire et résolutions allant de QCIF (176×144 pixels) au HD (1920×1080). Nous obtenons des gains en accélération impressionnants, de l'ordre 3× à 5× comparé à la méthode en cascade conventionnelle pour des pertes moyennes de 0,15 dB à 0,5 dB pour du QCIF. Ces résultats rendent les algorithmes proposés très attrayants pour le transcodage MPEG-4 à H.264 en temps réels. Comparativement aux méthodes de transcodage rapides de l'état de l'art, les algorithmes proposés donnent des résultats significativement meilleurs tant en accélération qu'en qualité visuelle. Les simulations ont été faites sous les codecs IPP de Intel, qui sont déjà nettement optimisés en vitesse. Les résultats présentés sont ainsi représentatifs de ceux qui pourraient être attendus avec des produits commerciaux.

Il faut mentionner que les méthodes proposées peuvent s'étendre à d'autres transcodages de formats vidéos. Finalement, ce travail de recherche fournit également de nouvelles bases pour des recherches futures. Voici une liste non exhaustive de recherches intéressantes :

- Étude approfondie du transcodage des MBs d'entrée Inter8×8. Étude de l'usage des directions des VMs MPEG-4 8×8 dans le processus de décision des MCs H.264.
- Étude, application et extension des méthodes dans le cadre d'un transcodage de format différent, c'est-à-dire autre que MPEG-4 à H.264.
- Application des outils utilisés dans les méthodes développées, au transcodage de trames intra.
- Étude portant sur une réduction plus drastique de la complexité du transcodage en combinant des travaux effectués sur la réduction de la complexité du codeur H.264 avec les méthodes de réduction par la réutilisation d'informations issues du décodeur MPEG-4.
- Développement et application des algorithmes proposés dans le cadre d'un transcodage de format combiné à des opérations spécifiques telles que la réduction de débit binaire, la réduction du taux de trames ou de la résolution.

ANNEXE I

ÉTUDE COMPARATIVE DE LA MÉTHODE DE CALCUL DE LA SARR SUR LA TOTALITÉ ET SUR LA MOITIÉ DES PIXELS

Les tableaux I.1 à I.4 montrent les résultats comparatifs en termes de vitesse d'accélération, de PSNR et de fréquence de trames (F.T.) en évaluant la SARR, $R_{n/G}$, sur la totalité des pixels et sur la moitié des pixels. Le PSNR pour la méthode travaillant sur la moitié des pixels représente le gain en qualité (qui est souvent une perte dans les faits) par rapport à la méthode qui travaille sur tous les pixels. Dans les deux cas, les accélérations en vitesse obtenues ont été calculées en référence à la méthode de transcodage en cascade. On peut constater des gains d'accélération assez notables, de 8 à 15% en effectuant le calcul de la SARR sur la moitié des pixels plutôt que sur la totalité des pixels, et ce, pour des pertes en qualité pratiquement nulles (en moyenne de 0,03 dB). Le constat est le même quelle que soit la résolution. La perte en qualité est toutefois légèrement plus élevée pour les vidéos à mouvements complexes où la perte en qualité est autour de 0,07 dB. Elle tend aussi à augmenter avec le débit.

Tableau I.1 Tableau comparatif des accélérations en vitesse, de la qualité et de la fréquence de trames (F.T.) de l'algorithme proposé en calculant la SARR sur tous les pixels et sur la moitié des pixels pour des vidéos QCIF

QCIF		32 kb/s		96 kb/s		128 kb/s		192 kb/s	
		Calcul $R_{n/G}$ tous-pixel	Calcul $R_{n/G}$ moitié-pixel						
Akiyo	PSNR (dB)	38,96	0,01	43,91	-0,02	45,22	-0,02	47,39	-0,04
	SPEED UP	3,60	3,58	4,17	4,21	4,40	4,49	5,00	5,18
	F.T.	133,33	132,74	108,30	109,49	98,68	100,67	85,96	89,02
Carphone	PSNR (dB)	29,75	0,00	35,21	-0,06	36,49	-0,03	38,45	-0,05
	SPEED UP	2,88	3,08	3,07	3,43	3,16	3,57	3,18	3,79
	F.T.	92,72	99,22	65,19	72,90	59,97	67,85	51,41	61,22
Mobile	PSNR (dB)	23,52	-0,01	27,62	-0,06	28,48	-0,06	29,86	-0,07
	SPEED UP	2,56	3,10	2,98	3,88	3,08	4,05	3,33	4,36
	F.T.	78,74	95,24	58,14	75,76	54,74	71,94	50,85	66,67
Moyenne PSNR (dB)		30,74	0,00	35,58	-0,05	36,73	-0,04	38,57	-0,05
Moyenne SPEED UP		3,01	3,25	3,41	3,84	3,55	4,04	3,84	4,44
Moyenne F.T.		101,60	109,07	77,21	86,05	71,13	80,15	62,74	72,30

Tableau I.2 Tableau comparatif des accélérations en vitesse, de la qualité et de la fréquence de trames (F.T.) de l'algorithme proposé en calculant la SARR sur tous les pixels et sur la moitié des pixels pour des vidéos CIF

CIF		128kb/s		384kb/s		512kb/s		768kb/s	
		Calcul $R_{n/G}$ tous-pixel	Calcul $R_{n/G}$ moitié-pixel						
M-Daughter	PSNR (dB)	38,75	-0,07	43,30	-0,05	44,33	-0,04	45,86	-0,04
	SPEED UP	3,57	3,67	3,95	4,28	4,16	4,62	4,57	5,17
	F.T.	28,96	29,79	21,65	23,46	19,88	22,06	17,65	19,96
Coastguard	PSNR (dB)	28,21	0,02	32,23	-0,02	33,39	-0,01	35,27	-0,03
	SPEED UP	3,17	3,56	3,79	4,33	3,91	4,48	4,01	4,72
	F.T.	22,61	25,45	16,88	19,24	15,53	17,81	13,62	16,04
Bus	PSNR (dB)	25,23	-0,01	30,35	-0,08	31,61	-0,07	33,53	-0,07
	SPEED UP	3,04	3,27	3,24	3,71	3,26	3,85	3,32	4,08
	F.T.	22,66	24,35	15,66	17,94	14,16	16,69	12,06	14,82
Moyenne PSNR (dB)		30,73	-0,02	35,29	-0,05	36,44	-0,04	38,22	-0,05
Moyenne SPEED UP		3,26	3,50	3,66	4,11	3,78	4,31	3,97	4,66
Moyenne F.T.		24,74	26,53	18,06	20,21	16,52	18,85	14,44	16,94

Tableau I.3 Tableau comparatif des accélérations en vitesse, de la qualité et de la fréquence de trames (F.T.) de l'algorithme proposé en calculant la SARR sur tous les pixels et sur la moitié des pixels pour des vidéos 4CIF, 704×576

4CIF, 704×576		512kb/s		1,53Mb/s		2,04Mb/s		3,07Mb/s	
		Calcul $R_{n/G}$ tous-pixel	Calcul $R_{n/G}$ moitié-pixel						
City	PSNR (dB)	31,82	0,03	35,74	-0,05	36,71	-0,04	38,23	-0,06
	SPEED UP	3,11	3,76	3,43	4,30	3,60	4,48	3,90	4,90
	F.T.	5,54	6,70	4,05	5,09	3,81	4,74	3,42	4,29
Ice	PSNR (dB)	36,70	-0,01	42,23	-0,04	43,50	-0,03	45,38	-0,03
	SPEED UP	3,01	2,99	3,41	3,58	3,51	3,77	3,74	4,10
	F.T.	7,63	7,59	5,68	5,96	5,12	5,49	4,41	4,84
Soccer	PSNR (dB)	30,09	-0,01	35,38	-0,07	36,85	-0,08	39,02	-0,08
	SPEED UP	2,90	3,12	3,30	3,74	3,40	3,87	3,51	4,06
	F.T.	5,88	6,33	3,88	4,38	3,50	3,99	3,03	3,50
Moyenne PSNR (dB)		32,87	0,00	37,78	-0,05	39,02	-0,05	40,88	-0,06
Moyenne SPEED UP		3,00	3,29	3,38	3,87	3,50	4,04	3,72	4,35
Moyenne F.T.		6,35	6,87	4,54	5,15	4,14	4,74	3,62	4,21

Tableau I.4 Tableau comparatif des accélérations en vitesse, de la qualité et de la fréquence de trames (F.T.) de l’algorithme proposé en calculant la SARR sur tous les pixels et sur la moitié des pixels pour des vidéos HD, 1920×1080

HD, 1920×1080		2,62 Mb/s		5,24 Mb/s		7,87 Mb/s		13,12 Mb/s	
		Calcul $R_{n/G}$ tous-pixel	Calcul $R_{n/G}$ moitié-pixel						
Blue-sky	PSNR (dB)	36,19	-0,05	39,42	-0,07	40,92	-0,04	42,88	-0,06
	SPEED UP	3,03	3,20	3,20	3,48	3,24	3,63	3,31	3,87
	F.T.	1,19	1,26	1,01	1,09	0,90	1,01	0,77	0,90
Pedestrian-area	PSNR (dB)	37,19	0,01	40,82	0,00	42,60	-0,02	44,64	-0,03
	SPEED UP	3,16	3,24	3,42	3,63	3,56	3,89	3,79	4,20
	F.T.	1,15	1,18	0,87	0,92	0,73	0,80	0,61	0,68
Tractor	PSNR (dB)	31,35	0,00	35,14	-0,04	37,26	-0,05	39,94	-0,05
	SPEED UP	2,98	3,25	3,38	3,67	3,48	3,79	3,39	3,71
	F.T.	0,96	1,04	0,76	0,82	0,66	0,72	0,54	0,59
Moyenne PSNR (dB)		34,91	-0,01	38,46	-0,04	40,26	-0,04	42,49	-0,05
Moyenne SPEED UP		3,06	3,23	3,33	3,60	3,43	3,77	3,50	3,92
Moyenne F.T.		1,10	1,16	0,88	0,95	0,76	0,84	0,64	0,72

ANNEXE II

LE TEST ANOVA

L'analyse de la variance (*ANOVA : ANalysis Of Variance*) est un outil statistique qui permet d'étudier le comportement d'une variable en fonction d'une ou de plusieurs variables explicatives. Ces variables sont appelées facteurs de variabilité. Lorsqu'on souhaite étudier le comportement de deux ou plusieurs variables explicatives en même temps, on utilise plutôt le MANOVA (Multivariate ANalysis Of Variance (MANOVA)), la forme généralisée de l'ANOVA. On parle alors d'analyse multifactorielle. Dans le cas de l'ANOVA, on parle d'analyse à un facteur. Concrètement, l'analyse consiste à mesurer l'influence de chacune des variables explicatives sur la variable à expliquer de même que les différentes interactions entre ces variables et leur effet sur la variable à expliquer. Pour finir l'ANOVA, permet d'infirmier ou non les hypothèses émises quant au lien entre la variable et les facteurs explicatifs. Des détails sur l'ANOVA peuvent être lus dans les références Scherrer (2008) et Lindman (1974).

La première étape d'une analyse consiste à modéliser théoriquement la problématique en mettant en évidence les variables explicatives à étudier. On peut écrire le modèle général sous la forme de l'équation II.1.

$$y(x, z, w) = \mu + f(x, z, w, \dots) + \varepsilon \quad (\text{II.1})$$

Avec $y(x, z, w)$, la variable à expliquer, μ une constante, $f(x, z, w, \dots)$ la relation entre les variables explicatives et ε l'erreur de mesure. Dans notre cas, la variable à expliquer est le MC H.264. Plusieurs variables peuvent déterminer le MC H.264, notamment, le MC MPEG-4, l'information résiduelle, le débit binaire, le type de vidéo, le codage entropique utilisé, la mesure de distorsion, la SAD ou la SATD, le débit de codage MPEG-4, la résolution vidéo, l'algorithme d'estimation de mouvement, de contrôle de débit binaire, etc. Notre but est toutefois de déterminer l'effet de certaines variables sur le MC H.264, notamment le MC MPEG-4 MC_{mp4} , la SARR ($R_{n/G}$), le débit binaire d_b et les caractéristiques vidéo (C_v). Nous nous limiterons donc à l'influence que peuvent avoir ces dernières variables. L'influence des autres facteurs constitueront sans doute l'erreur ε . Dans ce cas le MC H.264 peut être modélisé sous

Tableau II.1 Table ANOVA à un facteur. Adaptée de Scherrer (2008)

Variabes	SCE	DDL(df)	Variance	F	p
Interclasses	SCE_f	DDL_f	$S_f^2 = \frac{SCE_f}{DDL_f}$	$F = \frac{S_f^2}{S_{résidu}^2}$	$P_{H_0}(F > F_{obs})$
Intraclases	$SCE_{résidu}$	$DDL_{résidu}$	$S_{résidu}^2 = \frac{SCE_f}{DDL_f}$		
Total	SCE_{total}	DDL_{total}			

la forme de l'équation II.2.

$$MC_{H.264}(MC_{mp4}, R_{n/G}, d_b, C_v) = \mu + f(MC_{mp4}, R_{n/G}, d_b, C_v \dots) + \varepsilon \quad (II.2)$$

Une fois, la problématique modélisée, il reste donc à vérifier les hypothèses qui ont été émises. Il faut savoir comment l'ANOVA s'y prend concrètement pour infirmer ou affirmer les hypothèses émises, pour mesurer l'influence des facteurs de variabilité sur la variable à expliquer et les interactions entre ces différents facteurs. Pour mesurer les hypothèses émises, on établit un seuil de rejet F et on calcule p , la probabilité que notre hypothèse soit vraie. La valeur du seuil de rejet dépend du champ d'études. Pour la plupart des applications, elle se situe autour de 0,05. Par contre, dans des cas d'applications biologiques, la barre est plus haute ; souvent autour de 0,1. Le terme employé pour désigner la véracité des hypothèses émises est que l'hypothèse nulle est rejetée. Si $p < F$, on peut affirmer avec une grande certitude que l'hypothèse nulle est rejetée, qu'il existe un lien fort entre les facteurs de variabilité et la variable à expliquer. Cette valeur p est indiquée dans la dernière colonne d'une table établie par l'ANOVA : la table ANOVA telle que décrite dans le tableau II.1. Cette table ANOVA calcule également certains paramètres, notamment l'erreur due au modèle. Expliquons plus en détail chaque paramètre de la table ANOVA et leur signification. Comme le nom le mentionne (Analysis Of Variance), l'ANOVA effectue une analyse de la variance. La première étape de l'ANOVA consiste à décomposer la variance. Elle tente ainsi d'expliquer la variance totale en fonction

de la variance due à chaque facteur, soit la variance expliquée par le modèle de l'équation II.1, en fonction de la variance due à l'interaction entre les facteurs et de la variance due à l'erreur résiduelle aléatoire (non expliquée par le modèle). La variance totale peut être représentée par son estimateur biaisé S_n , la somme des carrés des écarts (SCE). Elle se calcule selon la formule II.3. Détaillons et décomposons davantage l'estimateur de la variance pour de mettre en évidence certains paramètres explicatifs qui apparaissent dans la table ANOVA.

$$SCE = \sum_{x,y,w} (y(x,z,w...) - \bar{y})^2 \quad (\text{II.3})$$

L'estimateur non biaisé de la variance S_{n-1}^2 se calcule selon la formule II.4

$$S_{n-1}^2 = \frac{SCE}{n-1} \quad (\text{II.4})$$

L'estimateur de la variance totale peut être écrit comme une combinaison linéaire de la variance de chaque variable explicative, SCE_{f_i} et de la variance relative aux interactions entre les variables explicatives, $SCE_{interaction_{ij}}$. La variance totale se décompose donc de manière suivante :

$$SCE = \sum_i SCE_{f_i} + \sum_{i,j} SCE_{interaction_{ij}} \quad (\text{II.5})$$

Pour revenir à notre variable à expliquer, le MC H.264, la variance totale peut se décomposer sous la forme suivante :

$$SCE = SCE_{f_{MC_{mp4}}} + SCE_{f_{R_{n/G}}} + SCE_{f_{db}} + SCE_{f_{C_v}} + \sum_{i,j} SCE_{interaction_{ij}} \quad (\text{II.6})$$

$$SCE = \sum_i SCE_{f_i} + \sum_{i,j} SCE_{interaction_{ij}}, \quad i \in \{MC_{mp4}, R_{n/G}, db, C_v\} \quad (\text{II.7})$$

La table ANOVA décompose ainsi la variance afin de mettre en évidence la variance due aux facteurs de celle due aux interactions. Elle les recense dans la première colonne de la table ANOVA. Dans le cas d'une analyse à un facteur, on peut simplifier la modélisation du problème. L'estimateur de la variance et la SCE, peuvent se réduire respectivement aux

équations II.8 et II.9.

$$Y_{ij} = \alpha_i + \varepsilon_{ij} \quad (\text{II.8})$$

$$SCE_{total} = SCE_f + SCE_{résidu} \quad (\text{II.9})$$

SCE_f représente la variance expliquée par le modèle, elle aussi appelée variabilité inter-classe ou SSB (sum of square between class). $SCE_{résidu}$ quant à elle représente la variance non expliquée par le modèle, elle aussi appelée variabilité aléatoire ou SSW (sum of square within class). Elles se calculent respectivement comme suit dans les équations II.10 et II.11.

$$SCE_f = \sum_{i=1}^p n_i (\bar{y}_i - \bar{y})^2 \sim (DDL_f) \quad (\text{II.10})$$

$$SCE_{résidu} = \sum_{i=1}^p \sum_{j=1}^{n_i} (y_i^j - \bar{y}_i)^2 \sim (DDL_{résidu}) \quad (\text{II.11})$$

Avec $DDL_f = p - 1$ et $DDL_{résidu} = n - p$. Avec p , le nombre de niveaux que peut prendre le facteur de variabilité. La table ANOVA évalue ensuite les estimateurs pour chaque facteur S_f^2 et pour l'erreur (colonne), $S_{résidu}^2$. Ils sont calculés selon les formules respectives II.12 et II.13.

$$S_f^2 = \frac{SCE_f}{p-1} = \frac{1}{p-1} \sum_{i=1}^p n_i (\bar{y}_i - \bar{y})^2 \quad (\text{II.12})$$

$$S_{résidu}^2 = \frac{SCE_{résidu}}{n-p} = \frac{1}{n-p} \sum_{i=1}^p \sum_{j=1}^{n_i} (y_i^j - \bar{y}_i)^2 \quad (\text{II.13})$$

Pou finir, la table ANOVA évalue F , le rapport entre la probabilité du facteur de variabilité par rapport à celle de l'erreur. Le calcul se fait comme l'indique l'équation II.14.

$$F = \frac{S_f^2}{S_{résidu}^2} = \frac{\frac{SCE_f}{p-1}}{\frac{SCE_{résidu}}{n-p}} \sim F(p-1, n-p) \quad (\text{II.14})$$

Tableau II.2 Table ANOVA mettant en relief le lien entre $R_{n/G}$ et le MC H.264 utilisé pour la vidéo QCIF Akiyo

Analysis of Variance					
Source	Sum Sq.	d. f.	Mean Sq.	F	Prob>F
SARR	22965.8	187	122.812	38.68	0
Error	14490.7	4564	3.175		
Total	37456.5	4751			

Dans le cas d'une analyse à plusieurs facteurs, l'estimateur de la variance totale se décompose selon la variance due aux facteurs, SCE_i , celle due aux interactions entre ces facteurs $SCE_{interaction}$, et celle due à des facteurs non déterminés $SCE_{résidu}$. L'équation II.15 suivante détaille cette décomposition.

$$SCE_{total} = \sum_i SCE_{facteur_i} + SCE_{interaction} + SCE_{résidu} \quad (II.15)$$

Plusieurs tests ANOVA ont été effectués pour différents types de vidéos, à des débits binaires variables, afin de démontrer le lien entre $R_{n/G}$ et le MC H.264. Le tableau II.2 recense les résultats obtenus pour la vidéo QCIF Akiyo à 5 débits binaires (32, 64, 96, 128 et 160 kb/s) pour 4 trames (les trames numéro 2, 68, 119 et 240). Similairement à la SARR, nous avons également vérifié la relation entre l'énergie résiduelle et le MC H.264. La table ANOVA a été obtenue pour la vidéo QCIF Akiyo à un débit de transcodage binaire de 96 kb/s pour 4 trames (les trames numéro 2, 68, 119 et 240). La séquence MPEG-4 a été encodée à 200 kb/s. Le test ANOVA effectué est présenté dans le tableau II.3.

Les résultats nuls obtenus pour $P > F$ dans les tables ANOVA confirment nos hypothèses avec très peu de risque. Ce qui montre qu'il existe une forte relation entre $R_{n/G}$ et le MC H.264 de même qu'entre l'énergie résiduelle et le MC H.264.

Tableau II.3 Table ANOVA mettant en relief le lien entre l'énergie résiduelle et le MC H.264 utilisé pour la vidéo QCIF Akiyo

Analysis of Variance					
Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Énergie résiduelle	23088.1	215	107.387	33.9	0
Error	14368.4	4536	3.168		
Total	37456.5	4751			

Tableau II.4 Table ANOVA mettant en relief le lien entre le MC MPEG-4 et celui de H.264 pour la vidéo QCIF Foreman

Analysis of Variance					
Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
MC MPEG-4	7144.05	3	2381.35	307.12	0
Error	36814.95	4748	7.75		
Total	43959	4751			

Similairement aux tests ANOVA qui viennent d'être présentés, il était important de vérifier également la relation entre le MC MPEG-4 et H.264. Nous exploitons, en effet, cette relation en établissant des tables de statistiques de correspondance entre les MCs MPEG-4 et H.264 et en testant ensuite les MCs les plus probables. Le test ANOVA a été effectué entre le MC MPEG-4 et H.264 et les résultats sont recensés dans le tableau II.4. La séquence MPEG-4 a été encodée à 200 kb/s. Le test ANOVA a été réalisé pour la vidéo QCIF Foreman à 5 débits binaires (32, 64, 96, 128 et 160 kb/s) pour les trames 2 et 68. Les résultats nuls obtenus pour $P > F$ dans ces tables confirment l'hypothèse qui dit qu'il existe un lien entre le MC MPEG-4 et le MC H.264.

Tableau II.5 Table ANOVA mettant en relief le lien entre le MC MPEG-4, le débit binaire et le MC H.264 pour la vidéo CIF Bus

Analysis of Variance					
Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
MC MPEG-4	704.9	2	352.434	53.19	0
Débit	3.2	5	0.639	0.1	0.9928
MC MPEG4*Débit	308.9	10	30.887	4.66	0
Error	15622.9	2358	6.626		
Total	18216.6	2375			

Le tableau II.5 présente les tests ANOVA mettant en relief le MC MPEG-4, le débit binaire et le MC H.264. Le test ANOVA a été effectué pour la vidéo Bus CIF avec des trames P à 5 débits binaires de transcodage (128, 256, 384, 512 et 640 kb/s).

Ce test ANOVA confirme également qu'il existe une forte relation entre le MC MPEG-4 et celui d'H.264. On peut remarquer aussi qu'il n'existe pas de relation directe entre le débit binaire et le MC H.264.

L'idéal aurait été de mettre en évidence l'influence d'autres facteurs comme la résolution et de réaliser des tables où les interactions entre tous les facteurs intervenant dans le processus de décision du MC H.264 auraient été mises en évidence. Toutefois, l'élaboration de ces tables demande de nombreux calculs spécialement quand le DDL est élevé, la mémoire de nos ordinateurs de tests devient vite saturée. Nous avons toutefois pu vérifier les éléments les plus importants que sont la relation entre la $R_{n/G}$, le MC MPEG-4 et le MC H.264. Les résultats obtenus sont concluants, on obtient des valeurs nulles qui prouvent qu'il existe effectivement une forte relation entre $R_{n/G}$, le MC en MPEG-4 et le MC H.264.

ANNEXE III

POURCENTAGE DE RAFFINEMENT DES VECTEURS DE MOUVEMENT

Un des avantages clés de l'algorithme développé comparativement à ceux qui existent dans la littérature est de raffiner les VMs uniquement si ces derniers sont jugés inefficaces. Cette méthode nous permet d'augmenter la vitesse de traitement tout en minimisant l'impact sur la qualité. Nous avons, dans les tableaux III.1 et III.2, recensé le pourcentage de VMs raffinés pour différents types de vidéos. Comme on peut le constater, pour la méthode proposée (Proposed), pour certaines vidéos comme Miss-america et Highway, à peine 55% des VMs sont raffinés. Cela contribue à réduire significativement la complexité sans pour autant que la qualité en soit affectée. En effet comme en font état les résultats présentés au chapitre 5, pour Miss-america, par exemple, nous avons, par rapport à la méthode en cascade, une différence en PSNR variant de 0,16 dB à 0,35 dB. Notons que la méthode STAT+REF obtient de 0,24 dB à 1,02 dB en perte de qualité. La méthode STAT+REF donne les meilleurs résultats de la littérature en qualité en raffinant tous les VMs. Quand le besoin se fait le plus sentir, notamment pour des vidéos à mouvement plus complexe, comme Football ou Soccer, les VMs sont raffinés à un pourcentage plus élevé.

Tableau III.1 Pourcentage de VMs raffinés pour différents types de vidéos pour la méthode proposée (Proposed, Metoevi et Coulombe (2011b))

Séquences vidéos QCIF	32 kb/s	64 kb/s	96 kb/s	128 kb/s	16 kb/s	192 kb/s
Bus	84,54	85,96	86,86	87,45	87,85	88,17
Carphone	75,33	73,84	73,42	73,23	72,65	72,31
Claire	67,92	64,59	60,89	59,55	58,53	58,17
Foreman	79,35	79,11	79,15	79,15	79,06	78,69
Miss-america	63,59	60,04	58,97	57,32	56,53	55,55
Soccer	89,61	90,05	88,89	89,04	88,1	88
Séquences vidéos CIF	128 kb/s	256 kb/s	384 kb/s	512 kb/s	64 kb/s	768 kb/s
Crew	75,41	75,93	75,93	75,37	74,8	74,36
Football	80,78	82,34	82,8	82,64	82,61	82,22
Highway	64,4	60,15	57,6	55,54	54,11	52,77
Mother-daughter	75,76	74,01	73,02	72,35	71,82	71,56
Tempete	90,55	92,07	92,27	92,28	92,24	92,12

Tableau III.2 Pourcentage de VMs raffinés pour différents types de vidéos pour la méthode ResidualM (Metoevi et Coulombe (2009c))

Séquences vidéos QCIF	32 kb/s	64 kb/s	96 kb/s	128 kb/s	16 kb/s	192 kb/s
Bus	88,38	88,62	77,74	76,93	76,73	76,15
Carphone	74,39	72,24	43,54	41,78	40,09	38,72
Claire	53,2	41,08	10,8	9,09	8,3	7,78
Foreman	81,38	81,17	51,44	49,61	48,39	47,09
Miss-america	43,88	34,9	10,1	8,97	8,35	7,96
Soccer	72,6	72,59	55,89	53,73	51,64	49,7
Séquences vidéos CIF	128 kb/s	256 kb/s	384 kb/s	512 kb/s	64 kb/s	768 kb/s
Crew	80,08	80,58	49,71	47,68	46,13	44,92
Football	81,87	81,47	65,75	63,86	61,86	60
Highway	56,53	46,87	24,45	21,27	19,36	18,2
Mother-daughter	60,22	54,03	8,6	7,84	7,51	7,2
Tempete	90,27	91,28	80,35	79,49	78,48	77,43

ANNEXE IV

VALEURS MOYENNES DE L'ASAR SELON LA COMPLEXITÉ DE LA VIDÉO

Dans l'algorithme présenté au chapitre 5, pour le transcodage des MBs codés Inter8×8 en MPEG-4, le MC Intra est testé à certaines conditions. Son pourcentage d'occurrence tourne autour de 6% (voir le tableau 5.1). Bien que ce mode apparaisse rarement, l'ignorer aurait un impact sur la qualité, surtout que le mode intra est utilisé pour coder des MBs assez complexes. Pour éviter de nuire à la qualité sans augmenter considérablement le temps de transcodage, nous choisissons de tester ce mode uniquement pour les trames complexes. μ_G , la moyenne de la somme absolue du résiduel de la trame, nous servira de mesure de complexité de la trame vidéo.

Nous avons ainsi effectué des simulations sur plusieurs types de vidéos pour mesurer les valeurs typiques de μ_G pour différentes catégories de complexité vidéos. Dans le tableau IV.1, nous avons recensé les résultats de cette expérimentation. Nous pouvons observer pour plusieurs vidéos QCIF et CIF que la valeur de μ_G tourne autour de 130, 450 et 1100 respectivement pour des vidéos à faible, à moyenne et à haute complexité.

Tableau IV.1 Moyenne de la somme absolue du résiduel, μ_G , pour différents types de vidéos

Séquences vidéos	Résolution	Complexité des vidéos	μ_G
Akiyo	QCIF	faible (peu de détails spatiaux et peu de mouvement (Xiph.org (2011)))	98,47
Bus	QCIF	élevée* (beaucoup de détails spatiaux, beaucoup de mouvement translatif rapide)	1276,91
Claire	QCIF	faible* (peu de détails spatiaux, peu de mouvement (Xiph.org (2011)))	116,38
Coastguard	QCIF	moyenne (détails spatiaux moyens et faibles, mouvement moyen (Xiph.org (2011)))	609,66
Mobile	QCIF	élevée* (de nombreux détails spatiaux, mouvement complexe)	1153,72
Soccer	QCIF	élevée* (beaucoup de mouvement rotatif rapide et complexe)	907,45
City	CIF	moyenne* (beaucoup de détails spatiaux, mouvement moyen)	535,78
Crew	CIF	moyenne* (peu de détails spatiaux, mouvement moyen)	501,95
Flower	CIF	élevée* (beaucoup de détails spatiaux, beaucoup de mouvement rapide)	1024,43
News	CIF	faible (détails spatiaux moyens et faibles, mouvement moyen (Xiph.org (2011)))	195,38
Tempete	CIF	élevée* (beaucoup de détails spatiaux, beaucoup de mouvement rapide)	868,28
Waterfall	CIF	moyenne* (beaucoup de détails spatiaux, mouvement moyen rapide)	429,02

* selon la complexité subjective de l'auteur

ANNEXE V

RÉSULTATS DE SIMULATIONS SUPPLÉMENTAIRES POUR VIDÉOS 1280×720

Nous présentons dans le tableau V.1 et à la figure V.1, d'autres résultats de simulations en termes de qualité et de gains de vitesse. Ces résultats sont relatifs aux méthodes Proposed et ResidualM, développées et présentées dans les chapitres 3 ainsi qu'à celles de l'état de l'art STAT, STAT+REF et MV+MS présentées dans les références respectives Lee et al. (2006), Lee et al. (2006) et Liang et al. (2007).

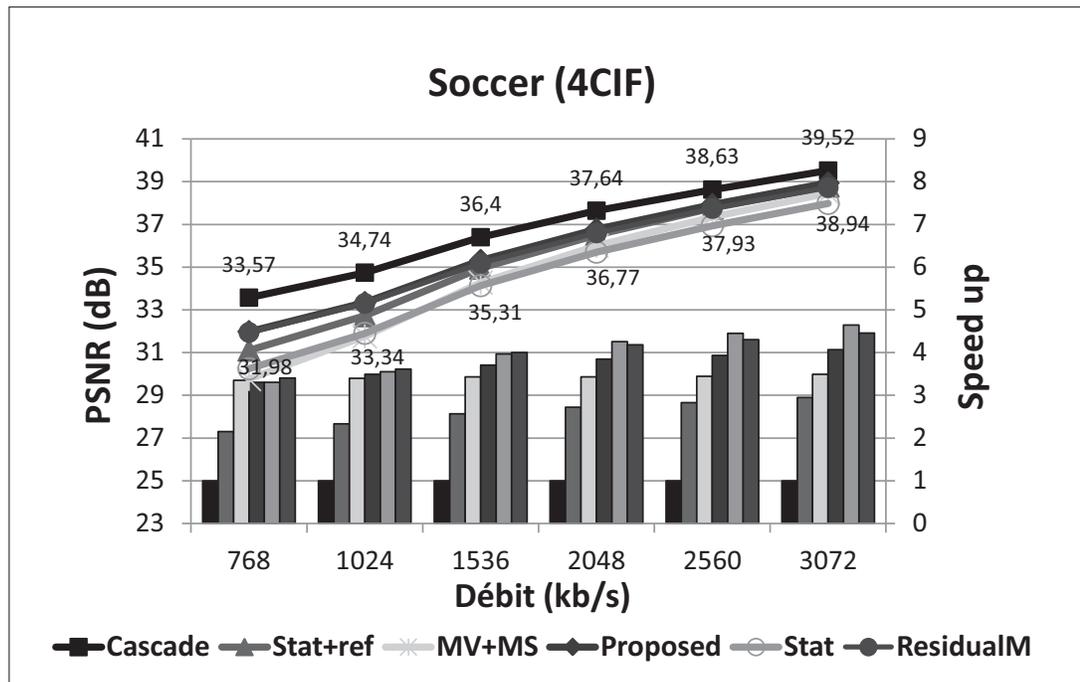


Figure V.1 Résultats en PSNR et en accélération (*speed up*) pour la vidéo Soccer (4CIF) à débits binaires différents. Les accélérations sont définies comme $T_{cascade}/T_{method}$, avec T représentant les temps CPU. L'ordre des méthodes pour les résultats en accélération est le même que la légende. Les valeurs indiquées sur le graphique sont les PSNR des méthodes Cascade et Proposed respectivement pour les valeurs au dessus et celles en dessous.

Tableau V.1 Résultats en PSNR, en accélération (*speed up*) et en fréquence de trames (F.T.) pour différents types de vidéos HD 1280×720. Les simulations sont effectuées pour les méthodes Proposed, ResidualM, STAT, STAT+REF et MV+MS. Les résultats en PSNR montrent les différences en qualité avec la méthode en cascade

HD, 1280×720		1,72Mb/s						2,30Mb/s						3,45Mb/s					
		Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS
Parkrun	PSNR (dB)	25,81	-0,67	-0,69	-0,69	-2,30	-1,12	26,65	-0,60	-0,55	-0,54	-2,28	-0,98	27,89	-0,50	-0,43	-0,27	-2,16	-0,68
	SPEED UP	1,00	3,81	3,91	1,96	3,03	3,10	1,00	3,97	4,07	2,09	3,26	3,11	1,00	4,13	4,27	2,24	3,51	3,10
	F.T.	1,00	3,82	3,92	1,97	3,04	3,11	0,91	3,62	3,72	1,91	2,97	2,84	0,81	3,35	3,47	1,82	2,85	2,52
Shields	PSNR (dB)	34,19	-0,99	-1,00	-2,08	-3,91	-2,66	35,22	-0,72	-0,72	-1,31	-3,34	-1,82	36,59	-0,50	-0,77	-0,78	-2,58	-1,12
	SPEED UP	1,00	3,45	3,59	1,88	2,93	2,92	1,00	3,58	3,70	1,98	3,12	2,82	1,00	3,76	4,11	2,13	3,44	2,74
	F.T.	0,99	3,41	3,54	1,86	2,89	2,88	0,90	3,22	3,33	1,78	2,82	2,54	0,79	2,96	3,24	1,68	2,71	2,16
Stockholm	PSNR (dB)	34,12	-0,61	-0,65	-1,38	-3,43	-1,67	35,06	-0,50	-0,53	-1,00	-3,00	-1,21	36,27	-0,35	-0,57	-0,60	-2,31	-0,77
	SPEED UP	1,00	3,62	3,74	1,96	3,04	2,96	1,00	3,73	3,84	2,03	3,23	2,84	1,00	3,99	4,31	2,22	3,61	2,82
	F.T.	0,95	3,45	3,57	1,87	2,90	2,83	0,88	3,29	3,39	1,79	2,85	2,51	0,76	3,03	3,28	1,69	2,75	2,15
Moyenne PSNR (dB)		31,37	-0,76	-0,78	-1,38	-3,21	-1,82	32,31	-0,61	-0,60	-0,95	-2,87	-1,34	33,58	-0,45	-0,59	-0,55	-2,35	-0,86
Moyenne speed up		1,00	3,63	3,75	1,93	3,00	2,99	1,00	3,76	3,87	2,03	3,20	2,92	1,00	3,96	4,23	2,20	3,52	2,89
Moyenne F.T.		0,98	3,56	3,68	1,90	2,94	2,94	0,90	3,38	3,48	1,83	2,88	2,63	0,79	3,12	3,33	1,73	2,77	2,27

HD, 1280×720		4,60Mb/s						5,76Mb/s						6,91Mb/s					
		Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS	Cas-cade	Proposed	ResidualM	STAT+REF	STAT	MV+MS
Parkrun	PSNR (dB)	28,88	-0,45	-0,41	-0,26	-1,93	-0,61	29,65	-0,42	-0,36	-0,22	-1,79	-0,57	30,37	-0,38	-0,39	-0,16	-1,71	-0,51
	SPEED UP	1,00	4,30	4,44	2,37	3,75	3,11	1,00	4,40	4,54	2,45	3,91	3,10	1,00	4,45	4,63	2,53	4,06	3,09
	F.T.	0,74	3,17	3,27	1,75	2,76	2,29	0,69	3,03	3,12	1,69	2,69	2,13	0,65	2,89	3,00	1,64	2,63	2,00
Shields	PSNR (dB)	37,55	-0,40	-0,72	-0,50	-2,21	-0,81	38,27	-0,34	-0,64	-0,36	-1,90	-0,61	38,91	-0,32	-0,62	-0,28	-1,64	-0,52
	SPEED UP	1,00	3,91	4,28	2,21	3,66	2,71	1,00	4,00	4,40	2,28	3,82	2,72	1,00	4,10	4,52	2,37	3,99	2,75
	F.T.	0,72	2,80	3,07	1,59	2,63	1,94	0,67	2,67	2,94	1,53	2,55	1,82	0,63	2,57	2,83	1,48	2,49	1,72
Stockholm	PSNR (dB)	37,13	-0,30	-0,53	-0,42	-2,03	-0,58	37,79	-0,27	-0,51	-0,30	-1,68	-0,46	38,42	-0,25	-0,49	-0,26	-1,52	-0,41
	SPEED UP	1,00	4,12	4,47	2,31	3,78	2,81	1,00	4,23	4,58	2,39	3,99	2,82	1,00	4,40	4,75	2,52	4,22	2,88
	F.T.	0,70	2,87	3,11	1,61	2,63	1,96	0,65	2,74	2,97	1,55	2,59	1,83	0,60	2,64	2,85	1,51	2,53	1,72
Moyenne PSNR (dB)		34,52	-0,38	-0,55	-0,39	-2,06	-0,67	35,24	-0,34	-0,50	-0,29	-1,79	-0,55	35,90	-0,32	-0,50	-0,23	-1,62	-0,48
Moyenne speed up		1,00	4,11	4,40	2,30	3,73	2,88	1,00	4,21	4,51	2,37	3,91	2,88	1,00	4,32	4,64	2,47	4,09	2,91
Moyenne F.T.		0,72	2,95	3,15	1,65	2,67	2,06	0,67	2,82	3,01	1,59	2,61	1,93	0,62	2,70	2,89	1,54	2,55	1,81

ANNEXE VI

INFLUENCE SUR LA QUALITÉ ET LA VITESSE DE L'ORDRE DES PROCESSUS DE RAFFINEMENT ET D'ÉVALUATION DU COÛT INTERVENANT DANS LA DÉTERMINATION DES VECTEURS DE MOUVEMENT

Pour déterminer les VMs, les méthodes développées commencent par évaluer le coût lié aux VMs candidats au demi-pixel et choisissent celui dont le coût est minimal. Ensuite, ce VM est raffiné du demi au quart de pixel. Cette méthode sera nommée méthode RafCoût par la suite. Il faut souligner que le processus inverse aurait pu se faire. C'est-à-dire qu'on aurait pu d'abord raffiner tous les VMs candidats du demi au quart de pixel, évaluer ensuite le coût et retenir le VM qui présente le plus faible coût. Cette méthode sera nommée méthode CoûtRaf. Dans cette section, nous avons mené des simulations afin de déterminer la méthode la plus efficace sur le plan de vitesse et de la qualité. Les tableaux VI.1 à VI.4, montrent les résultats sur le plan de la qualité (PSNR), du *speed ups*, et de la fréquence de trames obtenues pour les deux méthodes RafCoût et CoûtRaf.

On peut constater que la méthode Rafcoût est plus complexe que la méthode CoûtRaf. On observe une perte en vitesse d'environ 10% à 20% pour une amélioration presque nulle de la qualité. Cette tendance est la même quels que soient le débit binaire, le type de vidéo ou la résolution. Pour des vidéos à mouvements lents et moyens, la méthode CoûtRaf donne une qualité pratiquement pareille à la méthode RafCoût, une amélioration d'environ 0,02 dB. Pour des vidéos à mouvement plus complexes (Bus, Soccer), la différence en qualité est plus élevée à bas débits, mais elle est toujours relativement faible, d'environ 0,11 dB pour Bus et 0,06 dB pour Soccer. À haut débit, on retrouve une amélioration pratiquement nulle de 0,01 dB de la qualité. La première méthode, celle adoptée dans nos algorithmes, s'avère donc plus efficace. On pourrait toutefois changer de stratégie à très bas débit et à faible résolution pour obtenir une meilleure qualité au prix d'une complexité plus élevée.

Tableau VI.1 Etude comparative en gain de vitesse (*speed ups*), en fréquence de trames transcodées (F.T.) et en qualité (PSNR en dB) des méthodes CoûtRaf et Rafcoût pour des vidéos de résolution QCIF

Vidéos QCIF		32Kb/s		96Kb/s		160Kb/s	
		Méthode-RafCoût	Méthode-CoûtRaf	Méthode-RafCoût	Méthode-CoûtRaf	Méthode-RafCoût	Méthode-CoûtRaf
Bus	PSNR (dB)	22,45	-0,11	28,13	-0,05	30,64	0,00
	SPEED UP	2,78	3,16	2,94	3,35	3,10	3,52
	F.T.	75,00	85,23	51,72	59,06	44,91	51,02
Carphone	PSNR (dB)	29,78	-0,03	35,16	-0,01	37,52	0,00
	SPEED UP	2,77	3,13	3,09	3,47	3,30	3,69
	F.T.	90,74	102,69	66,32	74,61	58,23	65,19
Soccer	PSNR (dB)	24,47	-0,06	30,94	-0,04	33,98	0,00
	SPEED UP	2,52	2,89	2,85	3,30	2,96	3,37
	F.T.	83,33	95,54	56,82	65,79	46,88	53,38
Suzie	PSNR (dB)	33,76	-0,01	38,99	-0,02	41,19	-0,02
	SPEED UP	2,95	3,28	3,39	3,78	3,70	4,14
	F.T.	90,91	101,35	69,77	77,72	61,98	69,44
Moyenne PSNR (dB)		27,62	-0,05	33,31	-0,03	35,83	0,00
Moyenne speed up		2,75	3,12	3,07	3,48	3,26	3,68
Moyenne F.T.		84,99	96,20	61,16	69,29	53,00	59,76

Tableau VI.2 Etude comparative en gain de vitesse (*speed ups*), en fréquence de trames transcodées (F.T.) et en qualité (PSNR en dB) des méthodes CoûtRaf et et Rafcoût pour des vidéos de résolution CIF

Vidéos CIF		128Kb/s		384Kb/s		640Kb/s	
		Méthode-RafCoût	Méthode-CoûtRaf	Méthode-RafCoût	Méthode-CoûtRaf	Méthode-RafCoût	Méthode-CoûtRaf
Bus	PSNR (dB)	25,26	-0,04	30,29	-0,02	32,56	0,00
	SPEED UP	2,87	3,24	3,36	3,78	3,54	3,94
	F.T.	21,77	24,59	16,09	18,12	14,07	15,69
Foreman	PSNR (dB)	31,18	-0,01	36,43	0,00	38,53	0,00
	SPEED UP	3,02	3,38	3,44	3,86	3,74	4,16
	F.T.	23,64	26,41	17,48	19,63	15,46	17,22
Hall	PSNR (dB)	36,12	-0,07	40,24	-0,01	42,25	-0,02
	SPEED UP	3,30	3,54	3,90	4,13	4,51	4,74
	F.T.	31,38	33,67	22,44	23,79	19,51	20,52
Harbour	PSNR (dB)	26,32	-0,05	30,29	-0,01	32,35	-0,01
	SPEED UP	3,23	3,61	3,71	4,11	3,91	4,31
	F.T.	22,08	24,67	16,48	18,26	14,65	16,17
Soccer	PSNR (dB)	28,63	-0,02	34,14	-0,02	36,75	0,00
	SPEED UP	2,83	3,23	3,35	3,81	3,60	4,08
	F.T.	23,77	27,17	16,80	19,12	14,27	16,16
Moyenne PSNR (dB)		29,50	-0,04	34,28	-0,01	36,49	-0,01
Moyenne speed up		3,05	3,40	3,55	3,94	3,86	4,25
Moyenne F.T.		24,53	27,30	17,86	19,78	15,59	17,15

Tableau VI.3 Etude comparative en gain de vitesse (*speed ups*), en fréquence de trames transcodées (F.T.) et en qualité (PSNR en dB) des méthodes CoûtRaf et et Rafcoût pour des vidéos de résolution 4CIF

Vidéos 4CIF		768Kb/s		1,53Mb/s		2,56Mb/s	
		Méthode-RafCoût	Méthode-CoûtRaf	Méthode-RafCoût	Méthode-CoûtRaf	Méthode-RafCoût	Méthode-CoûtRaf
Crew	PSNR (dB)	33,16	-0,02	36,83	-0,02	39,61	-0,01
	SPEED UP	2,84	3,15	3,06	3,41	3,20	3,55
	F.T.	4,52	5,02	3,56	3,97	3,00	3,32
Harbour	PSNR (dB)	29,39	-0,03	32,37	-0,02	34,68	-0,01
	SPEED UP	3,14	3,67	3,35	3,92	3,48	4,06
	F.T.	4,40	5,13	3,53	4,12	3,02	3,53
Ice	PSNR (dB)	38,92	-0,02	42,21	-0,02	44,49	-0,01
	SPEED UP	2,83	3,17	3,19	3,57	3,57	3,92
	F.T.	6,33	7,09	5,39	6,03	4,72	5,19
Soccer	PSNR (dB)	31,98	0,00	35,31	0,00	37,94	-0,01
	SPEED UP	2,99	3,31	3,35	3,70	3,58	3,94
	F.T.	5,07	5,61	4,02	4,43	3,39	3,73
Moyenne PSNR (dB)		33,36	-0,02	36,68	-0,01	39,18	-0,01
Moyenne speed up		2,95	3,32	3,24	3,65	3,46	3,87
Moyenne F.T.		5,08	5,71	4,12	4,64	3,53	3,94

Tableau VI.4 Etude comparative en gain de vitesse (*speed ups*), en fréquence de trames transcodées (F.T.) et en qualité (PSNR en dB) des méthodes CoûtRaf et et Rafcoût pour des vidéos de résolution HD, 1920×1080

Vidéos HD 1920×1080		2,62 Mb/s		5,24 Mb/s		13,12 Mb/s	
		Méthode-RafCoût	Méthode-CoûtRaf	Méthode-RafCoût	Méthode-CoûtRaf	Méthode-RafCoût	Méthode-CoûtRaf
Blue-sky	PSNR (dB)	36,16	-0,02	39,36	-0,01	42,81	0,01
	SPEED UP	2,98	3,20	3,24	3,48	3,62	3,85
	F.T.	1,19	1,27	1,03	1,11	0,85	0,91
Pedestrian	PSNR (dB)	37,21	-0,01	40,83	-0,01	44,61	0,00
	SPEED UP	2,99	3,25	3,34	3,61	3,90	4,18
	F.T.	1,10	1,20	0,86	0,93	0,64	0,69
Riverbed	PSNR (dB)	26,83	-0,01	30,35	0,00	35,68	0,01
	SPEED UP	2,48	3,00	2,83	3,35	2,91	3,41
	F.T.	0,75	0,91	0,58	0,68	0,39	0,46
Station	PSNR (dB)	39,90	0,00	41,97	-0,01	44,08	-0,02
	SPEED UP	3,90	4,04	4,74	4,89	6,03	6,22
	F.T.	1,23	1,27	1,06	1,10	0,90	0,92
Moyenne PSNR (dB)		35,03	-0,01	38,13	-0,01	41,80	0,00
Moyenne speed up		3,09	3,37	3,54	3,83	4,12	4,42
Moyenne F.T.		1,07	1,16	0,88	0,95	0,70	0,75

LISTE DE RÉFÉRENCES

- 3GPP, Mars 2011a. 3GPP TS 26.140 v10.0.0, Multimedia Messaging Service (MMS), Media formats and codecs (Release 10).
- 3GPP, Juin 2011b. 3GPP TS 26.234 v10.1.0, Packet-switched Streaming Service (PSS); Protocols and codecs (Release 10).
- 3GPP2, Mars 2006a. 3GPP2 C.S0045-A, Multimedia Messaging Service (MMS) Media Format and Codecs for CDMA2000 Spread Spectrum Systems, version 1.0.
- 3GPP2, Février 2006b. 3GPP2 C.S0046-0, 3G Multimedia Streaming Services, version 1.0.
- ABI Research, 2011. Site web generation nouvelles technologies. En ligne. <<http://www.generation-nt.com/abi-research-services-video-mobile-estimation-actualite-1052801.html>>. Consulté le 15 septembre 2011.
- Ahmad, I., X. Wei, Y. Sun, et Y.Q. Zhang, Octobre 2005. Video transcoding : An overview of various techniques and research issues. *IEEE Transactions on multimedia*, vol. 7, no. 5, p. 793–804.
- Arizona State University, 2011. Video trace library. En ligne. <<http://trace.eas.asu.edu/yuv/>>. Consulté le 2 octobre 2011.
- Bhaskaran, V. et K. Konstantinides, 1997. *Image and video compression standards : algorithms and architectures*. Kluwer Academic Publishers, Norwell, Massachusetts, 2e édition.
- Bialkowski, J., A. Kaup, et K. Illgner, Octobre 2004. Fast transcoding of intra frames between H.263 and H.264. *ICIP, International Conference on Image Processing*, vol. 4, p. 2785–2788.
- Bialkowski, J., M. Barkowsky, et A. Kaup, Juillet 2006. Overview of low-complexity video transcoding from H.263 to H.264. *IEEE International Conference on Multimedia and Expo*, p. 49–52.
- Choi, I., J. Lee, et B. Jeon, Décembre 2006. Fast coding mode selection with rate-distortion optimization for MPEG-4 Part-10 AVC/H.264. *IEEE transactions on circuits and systems for video technology*, vol. 16, no. 12, p. 1557–1561.

- Chung, K. et L.C. Chang, Juin 2003. A new predictive search area approach for fast block motion estimation. *IEEE transactions on Image Processing*, vol. 12, p. 648–652.
- Dinh, T., J.Yoo, S.Park, G. Lee, T.Y. Chang, et H.J. Cho, Octobre 2007. Reducing spatial resolution for MPEG-4/H.264 transcoding with efficient motion reusing. *IEEE international conference on computer and information technology*, p. 577–580.
- Duanmu, C., Août 2005. *New motion estimation techniques and their SIMD implementations for vidéo coding*. Thèse de doctorat, Département de génie électrique, Université Concordia.
- Horn, R. et C. Johnson, 1991. *Topics in Matrix Analysis*. Cambridge University Press.
- Hur, J. et Y.L. Lee, Novembre 2005a. H.264 to MPEG-4 resolution reduction transcoding. *TENCON*, p. 1–6.
- Hur, J. et Y.L. Lee, Novembre 2005b. H.264 to MPEG-4 transcoding using block type information. *TENCON*, p. 1–6.
- Intel, 2008. Intel integrated performance primitives 5.3-code samples. En ligne. <<http://software.intel.com/en-us/articles/intel-integrated-performance-primitives-code-samples/>>. Consulté le 2 octobre 2011.
- Intel, 2009. Intel vtune performance analyzer. En ligne. <software.intel.com/en-us/intel-vtune>. Consulté le 2 octobre 2011.
- ISO/IEC, Avril 1993a. ISO/IEC-JTC1/SC29/WG11 : Test Model 5, MPEG93/N0400.
- ISO/IEC, 1993b. MPEG-1, ISO/IEC 11172-2 :1993, Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 2 : Video.
- ISO/IEC, 2000. MPEG-2, ISO/IEC 13818-2 :2000, Information technology - Generic coding of moving pictures and associated audio information : Video.
- ISO/IEC, Décembre 2001. MPEG-4 Reference Software 2e édition, ISO/IEC 14496-5 :2001, Information technology - Coding of audio-visual objects - Part 5.
- ISO/IEC, 2004. MPEG-4, ISO/IEC 14496-2 :2004, Information technology - Coding of audio-visual objects - Part 2 : Visual.
- ITU, 2003. H.264/AVC reference software JM 15.1. En ligne. <<http://iphome.hhi.de/suehring/tml/>>. Consulté le 2 octobre 2011.

- Jain, J. et A. Jain, Décembre 1981. Displacement measurement and its application in interframe image coding. *IEEE transactions on communications*, vol. 29, p. 1799–1808.
- Joch, A. et F. Kossentini, Décembre 2001. Performance analysis of H.26L coding features purpose. *Document VCEG-O42 Video coding experts group (VCEG), 15th meeting, Pattaya, Thaïlande*, p. 2209–2212.
- Kim, S., J.K. Han, et J.G. Kim, Septembre 2005. Efficient motion estimation algorithm for MPEG-4 to H.264 transcoder. *IEEE international conference on image processing*, vol. 3, p. 659–702.
- Kuhn, P., 1999. *Algorithms, complexity analysis and VLSI architectures for MPEG-4 motion estimation*. Kluwer Academic Publishers, Boston.
- Lee, Y. et Y.L. Lee, Novembre 2005. MPEG-4 to H.264 transcoding. *TENCON*, vol. 3, p. 1–6.
- Lee, Y., S.S. Lee, et Y.L. Lee, Juillet 2006. MPEG-4 to H.264 transcoding using macroblock statistics. *IEEE international conference on multimedia and expo*, p. 57–60.
- Liang, Y., X. Wei, I. Ahmad, et V. Swaminahan, Août 2007. MPEG-4 to H.264/AVC transcoding. *The International Wireless Communications and Mobile Computing Conference*, p. 689–693.
- Lim, K., G. Sullivan, et T. Wiegand, Avril 2005. Text description of joint model reference encoding methods and decoding concealment methods. *Joint Video Team Document JVT-0079*.
- Lindman, H., 1974. *Analysis of variance in complex experimental designs*. W. H. Freeman & Co. Hillsdale, NJ USA.
- Ma, S., W. Gao, Y. Lu, et JVT, Octobre 2002. *Improved rate control algorithm, JVT-D069*. Genève, Suisse.
- Mahaney, M., Novembre 2010. Updating the long thesis - five key updates. Technical report, Citi Group.
- Merritt, L. et R. Vanam, Octobre 2007. Improved rate control and motion estimation for h.264 encoder. *IEEE International Conference on Image Processing, ICIP 2007*, vol. 5, p. 309–312.

- Metoevi, I. et S. Coulombe, Décembre 2009a. Methods and systems for efficient video transcoding. *demande de brevet PCT/CA2009/001811*.
- Metoevi, I. et S. Coulombe, Décembre 2009b. Method and system for generating block mode conversion table for efficient video transcoding. *demande de brevet PCT/CA2009/001807*.
- Metoevi, I. et S. Coulombe, Septembre 2009c. Efficient MPEG-4 to H.264 transcoding exploiting MPEG-4 block modes, motion vectors, and residuals. *International Symposium on Communications and Information Technologies*, p. 224–229.
- Metoevi, I. et S. Coulombe, Juillet 2011a. Method and system for efficient video transcoding using coding modes, motion vectors and residual information. *demande de brevet provisionnel US 61/510,845*.
- Metoevi, I. et S. Coulombe, Août 2011b. Efficient MPEG-4 to H.264 transcoding exploiting MPEG-4 coding modes, motion vectors and relative sum of absolute residuals. *soumis à IEEE transactions on circuits and systems for video technology*.
- Nguyen, T., G.S. Lee, J.Y. Chang, et H.J. Cho, Décembre 2007. Efficient MPEG-4 to H.264/AVC transcoding with spatial downscaling. *ETRI*, vol. 29, p. 826–828.
- Nian, Z., L. Mark, et S. Drew, 2004. *Fundamentals of Multimedia*. Prentice Hall.
- Pan, P. F., Y. Hongtao, et L. Zhiping, Mai 2006. Scalable fast rate-distortion optimization for H.264/AVC. *EURASIP Journal on Applied Signal Processing*, p. 1–10.
- Pixmania, 2011. Site web de pixmania. En ligne. <<http://blog.pixmania.com/high-tech/2863-telephone-portable.html>>. Consulté le 15 septembre 2011.
- Pratt, W., 2007. *Digital image processing*. John Wiley & sons, 4e édition.
- Rezazadeh, S. et S. Coulombe, 2009. A novel approach for computing and pooling structural similarity index in the discrete wavelet domain. *IEEE International Conference on Image Processing (ICIP)*, p. 2209–2212.
- Richardson, I., Mai 2004. *H.264 and MPEG-4 video compression video coding for next generation multimedia*. Wiley.
- Scherrer, B., Octobre 2008. *Biostatistique*. Gaëtan Morin, 2e édition.
- Sheikh, H. et A.C. Bovik, Février 2006. Image information and visual quality. *IEEE Transactions on Image Processing*, vol. 15, no. 2, p. 430–444.

- Shen, B., Octobre 2004. From 8-tap DCT to 4-tap integer-transform for MPEG-4 to H.264/AVC transcoding. *IEEE international conference on image processing*, vol. 1, p. 115–118.
- Shengfa, Y., C. Zhenping, et Z. Zhaowen, Juin 2006. Instruction-level optimization of H.264 encoder using SIMD instructions. *IEEE International Conference on Communications Circuits and Systems Proceedings*, vol. 1, p. 126–129.
- Sokol, J., 2011. ITU H.263 video compression. En ligne. <<http://www.h2631.com>>. Consulté le 28 septembre 2011.
- Stats, I. W., 2011. Site web de Internet World stats. En ligne. <<http://www.internetworldstats.com/stats.htm>>. Consulté le 15 septembre 2011.
- Tang, Q. et P. Nasiopoulos, Février 2010. Efficient motion re-estimation with rate-distortion optimization for MPEG-2 to H.264/AVC transcoding. *IEEE transactions on circuits and systems for video technology*, vol. 20, no. 2.
- Tourapis, A., Janvier 2002. Enhanced predictive zonal search for single and multiple frame motion estimation. *Visual Communications and Image Processing*, p. 1069–1079.
- Tourapis, A., O.C. Au, et M.L. Liou, Octobre 2002. Highly efficient predictive zonal algorithms for fast block-matching motion estimation. *IEEE transactions on circuits and systems for video technology*, vol. 12, no. 10, p. 934–947.
- Tourapis, Y. et A.M. Tourapis, Juillet 2003. Fast motion estimation within the H.264 codec. *Proceedings International Conference on Multimedia and Expo IC-ME'03*, vol. 3, p. 517–520.
- UIT, Mai 1996. rec. H.263, Video codec for low bit rate communication.
- UIT, Mars 2003. rec. H.264, Advanced video coding for generic audiovisual services.
- UIT/SG16, Juin 1997. Video codec test model, TMN8, Portland.
- Video Group, 1997. Text of ISO/IEC 14496-2 MPEG4 video VM-version 8.0, ISO/IEC JTC1/SC29/WG11 coding of moving pictures and associated audio MPEG 97/W1796.
- VideoLAN, 2011. x.264. En ligne. <<http://www.videolan.org/developers/x264.html>>. Consulté le 2 octobre 2011.

- Wang, Z., A.C. Bovik, H.R. Sheikh, et E.P. Simoncelli, Avril 2004. Image quality assessment : From error visibility to structural similarity. *IEEE Transactions on Image Processing*, vol. 13, no. 4, p. 600–612.
- Wiegand, T., H. Schwarz, A. Joch, F. Kossentini, et G.J. Sullivan, Juillet 2003. Rate-constrained coder control and comparison of video coding standards. *IEEE transactions on circuits and systems for video technology*.
- Xiph.org, 2011. Xiph.org test media. En ligne. <<http://media.xiph.org/video/derf/>>. Consulté le 2 octobre 2011.
- YouTube, 2011. Site web de YouTube press statistics. En ligne. <http://www.youtube.com/t/press_statistics>. Consulté le 15 septembre 2011.