Pré-analyse de la vidéo pour un codage adapté. Application au codage de la TVHD en flux H.264

Olivier Brouard

▶ **To cite this version:**

Olivier Brouard. Pré-analyse de la vidéo pour un codage adapté. Application au codage de la TVHD en flux H.264. Sciences de l'ingénieur [physics]. Université de Nantes, 2010. Français. NNT : . tel-00522618v1

HAL Id: tel-00522618

https://theses.hal.science/tel-00522618v1

Submitted on 1 Oct 2010 (v1), last revised 1 Oct 2010 (v2)

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

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

UNIVERSITÉ DE NANTES

ÉCOLE DOCTORALE

« SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE MATHEMATIQUES »

Année : 2010

Thèse de Doctorat de l'Université de Nantes

Spécialité : AUTOMATIQUE, ROBOTIQUE, TRAITEMENT DU SIGNAL ET INFORMATIQUE APPLIQUEE

Présentée et soutenue publiquement par

Olivier BROUARD

Le 20 juillet 2010

à l'École polytechnique de l'université de Nantes

Pré-analyse de la vidéo pour un codage adapté Application au codage TVHD en flux H.264

Jury

Président	:	M. Joseph Ronsin	*Professeur, Laboratoire IETR, INSA Rennes*
Rapporteurs	:	M. Claude Labit	*Directeur de Recherche INRIA, IRISA/INRIA, Rennes*
		M. Denis Pellerin	*Professeur, Université Joseph Fourier, GIPSA-lab, Grenoble*
Examinateurs	:	Mme Sheila Hemami	*Professor, Cornell University, Etats-Unis*
		M. Dominique Barba	*Professeur émérite, IRCCyN, Polytech'Nantes*
		M. Vincent Ricordel	*Maître de Conférence, IRCCyN, Polytech'Nantes*

Directeur de Thèse : Dominique BARBA
Laboratoire : Institut de Recherche en Communications et en Cybernétique de Nantes
Co-encadrant : Vincent Ricordel
Laboratoire : Institut de Recherche en Communications et en Cybernétique de Nantes
Composante de rattachement du directeur de thèse : École polytechnique de l'université de Nantes

N° ED 503 - 091

À Malaïka, Célia et Cléa

Remerciements

Je voudrais tout d'abord remercier MM. Claude Labit et Denis Pellerin d'avoir accepté d'être rapporteurs de cette thèse. J'ai apprécié leurs remarques pertinentes et leurs commentaires constructifs.

Je remercie également M. Joseph Ronsin de m'avoir fait l'honneur de présider ce jury. Enfin, je tiens à remercier Sheila Hemami d'avoir accepté d'être examinatrice de ma thèse.

Je tiens à exprimer ma sincère gratitude envers mes encadrants : Vincent Ricordel pour son soutien de tous les jours qui m'a permis de mener à bien ces travaux de thèse et Dominique Barba pour son expérience et ses conseils avisés. Merci encore pour vos relectures qui ont contribué à l'amélioration de ce manuscrit.

Je remercie également Fabrice Delannay, engagé avec moi sur le projet ArchiPEG, pour sa collaboration au niveau du développement logiciel ; Romuald Pépion et Romain Cousseau pour leur contribution au niveau de l'organisation des tests subjectifs.

Je remercie tous mes compagnons de thèses Sylvain, Eddy, Guillaume, Stéphane, Fadi et Éloïse pour nos nombreuses discussions et tous les bons moments passés ensemble ; Marcus pour les discussions scientifiques et ses propositions. Je remercie tous les membres de l'équipe Image Video Communications du laboratoire IRCCyN, les nombreux pots et *social events* témoignent de cette ambiance chaleureuse qui règne dans notre laboratoire.

Enfin, je remercie mes amis, ma famille et particulièrement mes parents pour m'avoir soutenu durant toute ses années. Eux, qui après avoir acheté mon premier cartable et accompagné pour mon premier jour d'école, ont su organiser avec réussite mon pot de thèse qui fut apprécié par tous.

Et pour conclure, je tiens à remercier mon épouse qui m'a supporté et soutenu dans les moments difficiles : merci Malaïka pour tes encouragements incessants et pour tout ce que tu m'apportes dans la vie.

Table des matières

Introduction générale 1

I Pré-analyse de la vidéo 5

1 Optimisation classique d'un codeur vidéo prenant en compte des caractéristiques non-stationnaires intrinsèques 7

Introduction . . . 7
1.1 Structure de base d'un codeur vidéo fondé sur la transformation de blocs de pixels . . . 8
1.1.1 Introduction . . . 8
1.1.2 La décorrélation temporelle . . . 10
1.1.2.1 Prédiction depuis l'image précédente . . . 10
1.1.2.2 Changements dus au mouvement . . . 10
1.1.2.3 Estimation et compensation de mouvement par bloc . . . 11
1.1.2.4 Prédiction par compensation de mouvement d'un macrobloc . . . 12
1.1.2.5 Bi-prédiction . . . 13
1.1.2.6 Structure et taille des GOP . . . 13
1.1.2.6.1 Structure du GOP . . . 14
1.1.2.6.2 Taille du GOP . . . 14
1.1.3 La décorrélation spatiale . . . 14
1.1.3.1 La transformation . . . 14
1.1.3.2 Transformation en cosinus discrète . . . 15
1.1.4 La quantification . . . 16
1.1.4.1 Quantification scalaire . . . 17
1.1.5 Le codeur entropique . . . 18
1.1.5.1 Codage à longueur variable . . . 18
1.1.5.1.1 Codage de type *Huffman* . . . 18
1.1.5.1.2 Codage de type *Huffman* pré-calculé . . . 18
1.1.5.1.3 Autres types de codes à longueur variable . . . 19
1.1.5.2 Codage arithmétique . . . 19
1.1.5.2.1 Codage arithmétique basé sur le contexte . . . 19
1.1.6 Conclusion . . . 19
1.2 Méthodes d'optimisation au sens débit-distorsion . . . 20
1.2.1 Approches bas niveau . . . 21
1.2.1.1 Méthodes adaptatives . . . 21
1.2.1.1.1 Influence de la taille des blocs . . . 22
1.2.1.1.2 Estimation du mouvement sous-pixélique . . . 22
1.2.1.1.3 Estimation du mouvement multi-références . . . 24
1.2.1.2 Méthodes prédictives . . . 24
1.2.1.2.1 Prédiction intra . . . 25
1.2.1.2.2 Prédiction du vecteur de mouvement . . . 25
1.2.1.2.3 Prédiction du pas de quantification . . . 26
1.2.1.3 Conclusion . . . 26

1.2.2 Approches haut niveau . . . 27
1.2.2.1 Quantification perceptuelle . . . 27
1.2.2.1.1 Matrices de quantification . . . 27
1.2.2.1.2 Quantification pondérée en fonction de la fréquence spatiale . . . 28
1.2.2.2 Codage perceptuel . . . 28
1.2.2.3 Conclusion . . . 29
Conclusion . . . 30

2 Caractéristiques et modèles du système visuel humain et de l'attention visuelle 33
Introduction . . . 33
2.1 Propriétés et modélisation du système visuel humain . . . 34
2.1.1 Introduction . . . 34
2.1.2 La perception de la luminance . . . 34
2.1.3 La perception des couleurs . . . 36
2.1.4 La sensibilité aux contrastes . . . 36
2.1.4.1 Sensibilité aux fréquences spatiales . . . 37
2.1.4.2 Sensibilité aux fréquences temporelles . . . 38
2.1.4.3 Interactions spatio-temporelles . . . 38
2.1.5 L'organisation multi-canal . . . 39
2.1.5.1 Décomposition spatiale de l'information . . . 39
2.1.5.2 Décomposition temporelle de l'information . . . 39
2.1.6 Les effets de masquage . . . 40
2.1.6.1 Le masquage spatial . . . 40
2.1.6.2 Le masquage temporel . . . 41
2.1.7 Conclusion . . . 41
2.2 Mouvements oculaires et attention visuelle . . . 42
2.2.1 Mouvements oculaires . . . 42
2.2.1.1 Les saccades . . . 42
2.2.1.2 Les fixations . . . 42
2.2.1.3 Autres types de mouvement . . . 42
2.2.2 Attention visuelle sélective . . . 43
2.2.2.1 Définition . . . 43
2.2.2.2 Les mécanismes de sélection dits passifs . . . 43
2.2.2.3 Les mécanismes de sélection dits actifs . . . 43
2.2.2.4 Mécanisme inhibiteur de l'attention visuelle . . . 44
2.2.2.5 Les caractéristiques visuelles attirant l'attention . . . 46
2.2.3 Conclusion . . . 46
2.3 Modélisation de l'attention visuelle pré-attentive . . . 47
2.3.1 Introduction . . . 47
2.3.2 Modèles empiriques et statistiques . . . 47
2.3.2.1 Modèles empiriques . . . 47
2.3.2.2 Modèles statistiques . . . 48
2.3.3 Modèles psycho-visuels . . . 48
2.3.3.1 L'architecture de base . . . 48
2.3.3.2 Exemples de modèles psycho-visuels de l'attention visuelle . . . 49
2.3.4 La dimension temporelle dans la modélisation . . . 52
2.3.5 Conclusion . . . 53
2.4 Applications possibles d'un modèle de l'attention visuelle au sein d'un codeur vidéo . . . 55
Conclusion . . . 56

3 Pré-analyse spatio-temporelle de la vidéo en vue du codage **59**
Introduction . . . 59
3.1 Principe général de l'outil de pré-analyse proposé . . . 60
3.2 Segmentation spatio-temporelle et suivi d'objets . . . 61
3.2.1 Introduction . . . 61
3.2.2 État de l'art des techniques de segmentation spatio-temporelle et de suivi d'objets . . . 62
3.2.2.1 Suivi par mise en correspondance . . . 62
3.2.2.2 Suivi par projection et initialisation . . . 63
3.2.2.3 Approches plus long terme . . . 64
3.2.2.4 Conclusion . . . 66
3.2.3 Segmentation basée mouvement . . . 67
3.2.3.1 Mouvement apparent . . . 67
3.2.3.1.1 Méthodes de mise en correspondance . . . 67
3.2.3.1.2 Méthodes différentielles . . . 67
3.2.3.1.3 Méthodes par transformation . . . 68
3.2.3.2 Estimation optimisée du mouvement apparent . . . 69
3.2.3.2.1 Hypothèses pour le calcul des tubes spatio-temporels . . . 69
3.2.3.2.2 Estimation multi-résolution du mouvement . . . 70
3.2.3.2.3 Résultats de l'estimation de mouvement . . . 70
3.2.3.3 Estimation du mouvement global . . . 70
3.2.3.3.1 Les modèles de mouvement . . . 71
Modèle quadratique . . . 71
Modèle affine complet . . . 72
Modèle affine simplifié . . . 72
3.2.3.3.2 Les méthodes d'estimation du mouvement global . . . 73
Méthodes de mise en correspondance . . . 73
Méthodes différentielles . . . 73
Méthodes d'estimation du mouvement global à partir d'un champ de vecteurs . . . 73
3.2.3.3.3 Estimation du mouvement global par accumulation . . . 74
Indices de confiance pour une estimation robuste des paramètres . . . 75
Estimation robuste des paramètres du mouvement global . . . 75
Résultats expérimentaux de l'estimation des paramètres du mouvement global . . . 78
Résultats expérimentaux de l'estimation des paramètres du mouvement global sur des séquences réelles . . . 80
3.2.3.4 Segmentation spatio-temporelle . . . 83
3.2.3.4.1 Segmentation de l'espace d'accumulation . . . 84
3.2.3.4.2 Résultats expérimentaux de la segmentation basée mouvement . . . 85
3.2.4 Segmentation spatio-temporelle multi-critères . . . 86
3.2.4.1 Mise en forme du problème d'estimation . . . 87
3.2.4.2 Fonctions de potentiel . . . 88
3.2.4.2.1 Caractéristiques spatiales . . . 89
3.2.4.2.2 Caractéristiques de couleur . . . 89
3.2.4.2.3 Caractéristiques de texture . . . 90
3.2.4.2.4 Caractéristiques de mouvement . . . 91
3.2.4.2.5 Caractéristiques temporelles . . . 92
3.2.4.3 Suivi d'objets . . . 92
3.2.4.3.1 Compensation en mouvement de la carte de segmentation du segment *t-1* . . . 92
3.2.4.3.2 Étiquetage et suivi des objets . . . 93
3.2.4.4 Minimisation de l'énergie globale . . . 93

3.2.4.5 Facteur d'importance des critères ajoutés . . . 94
3.2.4.6 Résultats expérimentaux . . . 94
3.3 Détermination des cartes de saillance . . . 96
3.3.1 Saillance spatiale basée sur le contraste de couleur . . . 96
3.3.1.1 Caractéristiques de couleur influençant l'attention visuelle . . . 97
3.3.1.2 Calcul de la saillance spatiale . . . 97
3.3.2 Saillance temporelle . . . 100
3.3.2.1 Mouvement dominant . . . 101
3.3.2.2 Mouvement relatif et saillance temporelle . . . 101
3.3.3 Saillance spatio-temporelle . . . 102
3.3.4 Résultats qualitatifs . . . 102
3.4 Temps de calculs . . . 103
Conclusion . . . 106

II Applications au codage **109**

4 Le codeur H.264 et ses modes d'optimisations **111**
Introduction . . . 111
4.1 Description du codeur H.264 . . . 112
4.1.1 Introduction . . . 112
4.1.2 Structure du codeur H.264 . . . 112
4.1.2.1 Terminologie . . . 112
4.1.2.2 Le codec H.264 . . . 113
4.1.2.3 Les images de référence . . . 114
4.1.2.4 Les tranches . . . 115
4.1.3 Prédiction inter-image . . . 115
4.1.3.1 Les tranches P . . . 115
4.1.3.1.1 Compensation de mouvement à structure d'arbre . . . 116
4.1.3.1.2 Les vecteurs de mouvement . . . 116
4.1.3.1.3 Prédiction des vecteurs de mouvement . . . 117
4.1.3.2 Les tranches B . . . 117
4.1.3.2.1 Options de prédiction . . . 117
4.1.4 Prédiction intra-image . . . 118
4.1.4.1 Prédiction des blocs 4×4 de luminance . . . 118
4.1.4.2 Prédiction des blocs 16×16 de luminance . . . 119
4.1.4.3 Prédiction des blocs de chrominance . . . 120
4.1.5 Filtre anti effet de bloc . . . 120
4.1.6 Transformation et quantification . . . 121
4.1.6.1 Transformation . . . 121
4.1.6.2 Quantification . . . 121
4.1.7 Conclusion . . . 121
4.2 Optimisations du codeur H.264 . . . 122
4.2.1 Introduction . . . 122
4.2.2 Accélération de l'estimation de mouvement . . . 123
4.2.2.1 Estimation de mouvement réalisant des recherches partielles . . . 124
4.2.2.1.1 Prédiction du point de recherche initial . . . 124
4.2.2.1.2 Utilisation de « formes de recherche » . . . 126
4.2.2.2 Estimation de mouvement utilisant un sous-échantillonnage . . . 126
4.2.2.3 Optimisation de l'estimation de mouvement sous-pixélique . . . 127
4.2.2.4 Utilisation de techniques d'arrêt . . . 129
4.2.2.5 Discussion . . . 129
4.2.3 Optimisation de la prédiction intra-image . . . 130

4.2.3.1 Détection de l'orientation des contours . . . 130
4.2.3.2 Décision en fonction de la corrélation spatiale . . . 131
4.2.3.3 Décision en fonction de la corrélation temporelle . . . 131
4.2.3.4 Discussion . . . 131
4.2.4 Optimisation de la prédiction inter-image . . . 132
4.2.4.1 Détection des macroblocs codés en mode SKIP . . . 132
4.2.4.2 Décision en fonction de l'activité spatiale . . . 132
4.2.4.3 Décision en fonction de l'activité spatio-temporelle . . . 133
4.2.4.4 Décision en fonction de la corrélation spatio-temporelle . . . 134
4.2.4.4.1 Décision en fonction de la corrélation temporelle . . . 134
4.2.4.4.2 Décision en fonction de la corrélation spatiale . . . 134
4.2.4.5 Décision en fonction des coûts débit-distorsion . . . 135
4.2.4.6 Discussion . . . 135
4.2.5 Recherche réduite pour les multiples images référence . . . 135
4.2.5.1 Prépondérance de l'image précédente . . . 136
4.2.5.2 Corrélation temporelle du vecteur de mouvement . . . 136
4.2.5.3 Relation entre la fenêtre de recherche temporelle et la taille des partitions . 137
4.2.5.4 Sélection de l'image référence . . . 137
4.2.5.5 Discussion . . . 138
4.2.6 Optimisation de la qualité . . . 138
4.2.7 Conclusion . . . 138
Conclusion . . . 139

5 Adaptation des paramètres du codage H.264 de la vidéo : méthodes et performances comparées avec le codeur de référence 141
Introduction . . . 141
5.1 Modification adaptative de la structure des GOP . . . 142
5.1.1 Approches de modification adaptative de la structure du GOP . . . 143
5.1.2 Variation dynamique du nombre d'images B . . . 144
5.1.2.1 Objectif . . . 144
5.1.2.2 Analyse des séquences . . . 145
5.1.2.2.1 Tests pour différentes configurations du GOP . . . 145
5.1.2.2.2 Caractérisation spatio-temporelle des séquences et des segments temporels . . . 146
5.1.2.3 Classification des segments temporels . . . 148
5.1.2.4 Résultats . . . 150
5.1.2.4.1 Tests réalisés . . . 150
5.1.2.4.2 Évaluation de l'approche . . . 150
5.1.2.5 Limitations de l'approche . . . 151
5.1.3 Adaptation dynamique de la taille du GOP en fonction de l'évolution de l'activité temporelle . . . 153
5.1.3.1 Objectif . . . 153
5.1.3.2 Détection des changements au sein du plan . . . 153
5.1.3.3 Analyse de l'évolution de l'activité temporelle . . . 154
5.1.3.4 Résultats . . . 156
5.1.3.4.1 Tests réalisés . . . 156
5.1.3.4.2 Évaluation de l'approche . . . 156
5.1.3.5 Limitations de l'approche . . . 157
5.2 Codage adaptatif basé sur la saillance visuelle . . . 157
5.2.1 Principe général de la compression sélective . . . 157
5.2.2 Objectif de la compression sélective directe . . . 159
5.2.3 Carte de saillance dédiée pour le codage . . . 159
5.2.3.1 Modification de la carte de saillance . . . 159

5.2.4 Modification du coeur de codage . . . 160
5.2.5 Résultats . . . 161
5.2.5.1 Tests réalisés . . . 161
5.2.5.2 Évaluation de l'approche . . . 161
5.3 Évaluation subjective de la qualité . . . 163
5.3.1 Méthodologie . . . 163
5.3.2 Contenus évalués . . . 168
5.3.2.1 Séquences vidéo . . . 168
5.3.2.2 Génération des séquences dégradées . . . 169
5.3.3 Conditions d'observation . . . 169
5.3.3.1 Écran . . . 169
5.3.3.2 Salle de tests . . . 170
5.3.3.3 Adaptation des différents formats à l'écran . . . 170
5.3.4 Observateurs . . . 171
5.3.5 Résultats . . . 171
5.3.5.1 Résultats de la modification adaptative de la structure du GOP . . . 172
5.3.5.2 Résultats du codage adaptatif basé sur la saillance visuelle . . . 174
5.3.5.3 Résultats des deux approches utilisées conjointement . . . 176
Conclusion . . . 178

Conclusion générale et perspectives **181**

Annexes **187**

A Présentation des séquences vidéo utilisées lors des tests **187**
A.1 Les séquences 720p . . . 187
A.1.1 New Mobile and Calendar . . . 187
A.1.2 Parkrun . . . 187
A.1.3 Knightshields . . . 188
A.1.4 Crew . . . 188
A.1.5 Night . . . 188
A.2 Les séquences 1080p . . . 188
A.2.1 Blue Sky . . . 188
A.2.2 Station . . . 188
A.2.3 Tractor . . . 188
A.2.4 Parkjoy . . . 189
A.2.5 Umbrella . . . 189

B Informations complémentaires sur le codeur H.264 **191**
B.1 Profils et niveaux . . . 191
B.2 Gestion des images de référence . . . 191
B.3 Prédiction des vecteurs de mouvement . . . 193
B.4 Options de prédiction des tranches B . . . 194
B.4.1 Bi-prédiction . . . 194
B.4.2 Prédiction directe . . . 194
B.4.3 Prédiction pondérée . . . 195
B.5 Codage entropique . . . 196
B.5.1 Codage Exp-Golomb . . . 196
B.5.2 CAVLC . . . 196
B.5.3 CABAC . . . 196

C Résultats de la modification adaptative de la structure du GOP **199**
C.1 Variation dynamique du nombre d'images B 199
C.2 Évolution de l'activité temporelle 204
C.3 Modification adaptative de la structure du GOP 208

Bibliographie **217**

Liste des publications **233**

Introduction générale

Depuis sa naissance en 1926, la télévision n'a cessé d'évoluer. Le premier changement important fut le passage à la couleur en 1951. L'année 2005 marqua un tournant pour en France, puisque la télévision numérique terrestre (TNT) fit son apparition. Cinq ans plus tard, la télévision analogique disparaît peu à peu du paysage télévisuel français, et le 17 mai 2010, la télévision analogique vit ses dernières heures dans la région Pays de la Loire. Alors que la migration de la TNT vers la télévision numérique haute définition (TVHD) commence à s'opérer, on annonce déjà la diffusion en 3D (pour les spectateurs équipés d'écrans adéquats) d'événements sportifs tels des matchs de la prochaine coupe du monde de football. Il faudra néanmoins attendre encore quelques années avant de voir les premières chaînes entièrement dédiées à la diffusion de contenus 3D. La généralisation de la TVHD représente donc l'évolution majeure de la télévision grand public.

Pré-analyse de la vidéo

Le passage à la TVHD a été rendu possible grâce aux progrès réalisés en matière de compression numérique et de techniques de retransmission permettant la diffusion des chaînes de télévision avec une qualité d'image vidéo très supérieure. Afin d'assurer une qualité d'usage satisfaisante, la TVHD nécessite une allocation de débit conséquente ainsi que l'utilisation de techniques de compression performantes. À ces fins, les États-Unis et le Japon ont choisi d'utiliser la norme de compression MPEG-2. En Europe, la norme H.264, aussi appelée MPEG-4/AVC (*Advanced Video Coding*), a été retenue. Celle-ci vise à gagner jusqu'à 50% de la bande passante actuellement utilisée par MPEG-2 pour une qualité équivalente, ce gain est possible grâce à une combinatoire plus riche des techniques de prédiction, associée aux techniques de codage entropique avancées. Mais ce standard souffre des mêmes défauts que les codeurs vidéo classiques. Dans leur principe même de fonctionnement, ces derniers ne disposent d'aucune information quant à l'évolution temporelle de la séquence à coder. Le codeur est donc dans l'incapacité de prendre des décisions à moyen/long terme afin d'assurer une certaine cohérence des stratégies de codage. Par exemple, l'hétérogénéité temporelle des décisions prises pour coder un macrobloc à la même position spatiale entre les images successives peut occasionner l'apparition de dégradations visuellement perceptibles et gênantes pour le téléspectateur.

Le système visuel humain (SVH) développe des stratégies particulières pour traiter l'énorme quantité d'informations à laquelle il est confronté. Parmi celles-ci, l'attention visuelle permettant de concentrer les ressources sensorielles sur des zones particulières d'intérêt de notre environnement. La concentration de ces ressources est effectuée de façon involontaire (un traitement très rapide et inconscient) et/ou volontaire. Dans ce deuxième cas, cela correspond à la mise en œuvre d'un processus contrôlé nécessitant un effort cognitif important et la quasi totalité des ressources attentionnelles. Ce mécanisme est déployé lorsqu'une tâche particulière doit être effectuée. Du fait de la grande complexité des mécanismes et des interactions, aussi que des interdépendances existantes entre ces mécanismes, modéliser entièrement et finement l'attention

visuelle est une tâche encore aujourd'hui trop complexe. Une mise en œuvre possible est de modéliser seulement l'attention visuelle pré-attentive, qui est le mécanisme sélectionnant de façon involontaire les zones pertinentes de notre environnement visuel.

La modélisation de l'attention visuelle est un domaine en plein essor. L'intégration de propriétés de bas niveau (mécanisme *Bottom-Up*) permet d'obtenir des modèles de plus en plus performants. Bien que J. Wolfe [WCF89] ait clairement identifié le mouvement, plus précisément le contraste de mouvement, comme un attracteur visuel, encore peu de travaux intègrent la dimension temporelle dans la modélisation de l'attention visuelle. Dans ses travaux, R. Milanese [Mil93] considère que les régions perceptuellement importantes doivent correspondre aux objets les plus saillants physiquement et sémantiquement et que les formes des régions détectées doivent être compactes et approximer la forme d'un objet. Ces approches ont particulièrement retenues notre attention.

De nombreux travaux intégrant des propriétés du SVH sont proposés pour l'évaluation de la qualité, ou améliorer les techniques de codage. Cependant, ces méthodes sont peu retenues au niveau des standards de compression vidéo. Les efforts de recherche réalisés se portent davantage sur l'enrichissement des nouvelles normes de codage. Par exemple, le standard H.264 rassemble de nombreuses techniques de prédiction ainsi qu'une optimisation du codage entropique, lui permettant d'obtenir des performances supérieures en termes de compression par rapport aux standards précédents comme MPEG-2. Ces performances sont obtenues par une minimisation conjointe du débit et de la distorsion à partir de critères « bas niveau ». Cependant, les méthodes intégrant des propriétés du SVH confirment leur intérêt pour le codage. Les standards sont compétitifs et visent à offrir de nombreuses décisions qui pourraient aussi être prises à partir de telles informations.

Objectifs de la thèse

Les objectifs de la thèse se déclinent suivant deux grands axes.

- Le premier concerne la mise en oeuvre d'une méthode de pré-analyse de la vidéo. Le but est d'analyser le contenu de la vidéo en tenant compte d'informations « haut niveau » afin de transmettre au codeur le jeu de paramètres optimal permettant d'exploiter au mieux les différents outils de codage (prédiction et quantification).
- Le deuxième axe concerne le codage avancé de la TVHD. Par l'intermédiaire de deux applications (parmi les nombreuses possibles) de codage exploitant les informations obtenues après notre méthode de pré-analyse de la vidéo. L'objectif est de montrer l'intérêt en codage vidéo d'une telle méthode de pré-analyse. Les performances de ces deux méthodes seront évaluées à partir de tests d'évaluation subjective de la qualité.

Organisation du mémoire

Comme le reflète le titre de la thèse, ce mémoire est composé de deux parties. La première, composée des chapitres 1 à 3, traite de la pré-analyse de la vidéo en vue de son codage avancé. La seconde, composée des chapitres 4 et 5, est dédiée aux applications de cette pré-analyse de la vidéo pour le codage de la Télévision Haute Définition en flux H.264.

Le chapitre 1 analyse des différents blocs de traitement qui constituent un codeur vidéo classique : la décorrélation temporelle, la décorrélation spatiale, la quantification et le codeur entropique. Nous décrivons les deux catégories de techniques d'optimisation au sens débit-distorsion : les approches « bas niveau » et les approches « haut niveau ».

Le chapitre 2 porte sur la modélisation du SVH et de l'attention visuelle. D'abord, nous nous intéresserons à la modélisation des propriétés dites de bas niveau qui peuvent également être appelées passives. Ensuite, nous nous intéresserons à la description de l'attention visuelle ainsi qu'à ses modèles mathématiques. L'objectif étant de prédire, à partir d'attributs de bas niveau, les positions des zones visuellement importantes de la séquence vidéo. Les modèles présentés seront regroupés en deux catégories. La première est constituée d'une part de modèles dit empiriques et d'autre part aussi de modèles statistiques, qui sont relativement éloignés du SVH. La seconde catégorie, quant à elle, regroupe les modèles basés sur une architecture biologiquement plausible. Cette architecture proposée par C. Koch et S. Ullman [KU85] sera d'abord décrite, introduisant la notion de carte de saillance. Les grandes caractéristiques des modèles s'inspirant de cette architecture et leurs caractères innovants seront ensuite examinés. Enfin, nous présenterons les mises en œuvre possibles au sein d'un codeur vidéo, permettant d'exploiter les cartes de saillance issues des modélisations de l'attention visuelle pré-attentive.

Le chapitre 3 présente notre méthode de pré-analyse de la vidéo. Nous décrirons d'abord notre méthode de segmentation spatio-temporelle et de suivi des objets dans la séquence vidéo. Une segmentation fondée sur le mouvement est d'abord réalisée en estimant le mouvement local et en tenant compte du mouvement global. Afin d'améliorer les résultats de cette segmentation fondée seulement sur le mouvement et d'assurer le suivi des objets temporellement, de nouveaux critères (couleur, texture et connexité) sont intégrés. Finalement, nous présenterons notre modèle de l'attention visuelle pré-attentive.

Le chapitre 4 débute la seconde partie par une présentation de la norme H.264, plus particulièrement des techniques de prédiction intra et inter images à partir de multiples images référence. Dans la deuxième partie, nous nous intéresserons aux travaux déjà effectués pour optimiser les performances du codeur H.264.

Le chapitre 5 présente deux applications de codage à partir des résultats de notre méthode de pré-analyse de la vidéo. Après un bref état de l'art, nous présenterons notre méthode de modification adaptative de la structure d'un GOP (*Group of Pictures*). La seconde partie sera consacrée à notre méthode de compression sélective de la vidéo en fonction des cartes de saillance. La dernière partie de ce chapitre concerne l'analyse des performances de ces deux méthodes à partir de tests d'évaluation subjective de la qualité.

Première partie

Pré-analyse de la vidéo

Chapitre 1

Optimisation classique d'un codeur vidéo prenant en compte des caractéristiques non-stationnaires intrinsèques

Sommaire

Introduction **7**

1.1 Structure de base d'un codeur vidéo fondé sur la transformation de blocs de pixels . **8**

1.1.1 Introduction 8

1.1.2 La décorrélation temporelle 10

1.1.3 La décorrélation spatiale 14

1.1.4 La quantification 16

1.1.5 Le codeur entropique 18

1.1.6 Conclusion 19

1.2 Méthodes d'optimisation au sens débit-distorsion **20**

1.2.1 Approches bas niveau 21

1.2.2 Approches haut niveau 27

Conclusion **30**

Introduction

Dans cette première partie nous nous intéressons à la pré-analyse d'une séquence vidéo en vue de son codage par des méthodes modernes. Afin de comprendre le besoin d'une étape de pré-analyse de la vidéo en vue de son codage, il est important d'identifier les principes, mais également les carences de l'optimisation d'une technique classique de codage. C'est pourquoi la première partie de ce chapitre est consacrée à la description d'un codeur vidéo classique.

L'objectif étant de définir les différentes fonctions d'un tel codeur vidéo pour lesquelles des optimisations ont été proposées et intégrées dans les dernières normes de codage (telle que la norme H.264, comme nous le verrons dans le chapitre 4). Ces optimisations sont divisées en deux catégories. La première regroupe les techniques dites de « bas niveau ». Celles-ci visent à optimiser le codage de l'information à transmettre,

en suivant de façon basique les étapes fondamentales de traitement du signal : décorrélation, transformation, quantification puis codage (statistique). La deuxième catégorie de techniques concerne celles appelées « haut niveau » qui ajoutent l'exploitation de certaines propriétés du système visuel humain (SVH) afin de mieux réduire les informations au sein de la séquence d'images.

Dans un premier temps, nous présenterons donc les différents blocs de traitement qui constituent un codeur vidéo classique : la décorrélation temporelle, la décorrélation spatiale, la quantification et le codeur entropique. Pour finir, nous décrirons les deux catégories de techniques d'optimisation au sens débit-distorsion : les approches « bas niveau » et les approches « haut niveau ».

1.1 Structure de base d'un codeur vidéo fondé sur la transformation de blocs de pixels

1.1.1 Introduction

La compression ou codage vidéo est le procédé permettant de compacter le nombre de bits du signal vidéo initial en un nombre plus réduit de bits. Avec les capacités grandissantes des médias de stockage et l'augmentation des débits de transmission, la compression pourrait être considérée comme moins nécessaire. Mais une vidéo numérique non comprimée (on parle de données vidéo brutes ou « raw » en anglais) requiert d'énormes besoins : une seconde de vidéo brute en haute définition (1920×1080 pixels en mode progressif à 50 images par seconde) produit environ $2,5\,Gbits$! La compression implique deux systèmes complémentaires : un compresseur (codeur) et un décompresseur (décodeur). Le codeur convertit la source de données en une forme comprimée avant la transmission ou le stockage. Le décodeur convertit ensuite celle-ci en une représentation identique ou approchée de la vidéo originelle. Cette paire codeur/encodeur est souvent dénommée codec (COdeur et DECodeur). On parle de codage asymétrique lorsque l'essentiel de la complexité calculatoire est du côté du codeur. La figure 1.1 schématise ce mécanisme.

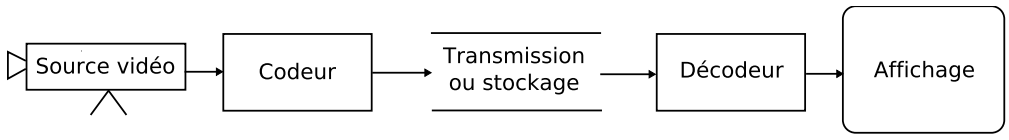


FIG. 1.1 – Schéma d'un codeur/décodeur.

La séquence décodée peut être identique à l'originale ; il s'agit alors de codage sans perte (*lossless coding* en anglais). Mais ce type de codage n'offre pas de taux de compression élevée. À titre d'exemple, le meilleur standard JPEG-LS ne permet une compression que d'un facteur trois à quatre pour les images fixes. Les différentes méthodes cherchent à comprimer l'information en se basant principalement sur la redondance statistique des données. À l'inverse, le codage avec pertes (*lossy coding*) reproduit approximativement la séquence vidéo mais permet d'atteindre des taux de compression bien plus élevés, malheureusement au dépend le plus souvent d'une perte de qualité visuelle. On exploite dans ce cas la « redondance subjective » : les éléments de la vidéo qui n'affectent pas significativement la perception visuelle de l'observateur peuvent donc être retirés. Cependant, une perte de la qualité visuelle peut être observée, si on va au delà de l'exploitation de la « redondance subjective » pour atteindre des taux de compression plus forts.

Le codage vidéo exploite (classiquement) la redondance temporelle (corrélation entre des images successives), spatiale (corrélation entre les pixels proches spatialement les uns des autres), ainsi que les informations non perçues (et donc inutiles). Dans ce paragraphe, nous présentons un codec vidéo et ses étapes

principales : la décorrélation temporelle, la décorrélation spatiale, la quantification et le codeur entropique. L'objectif est donc de présenter ces étapes de base, pour s'intéresser ensuite aux méthodes d'optimisation proposées dans la littérature et de fixer les limites d'un tel schéma de codage.

Un codec se décompose donc en quatre phases principales (figure 1.2).

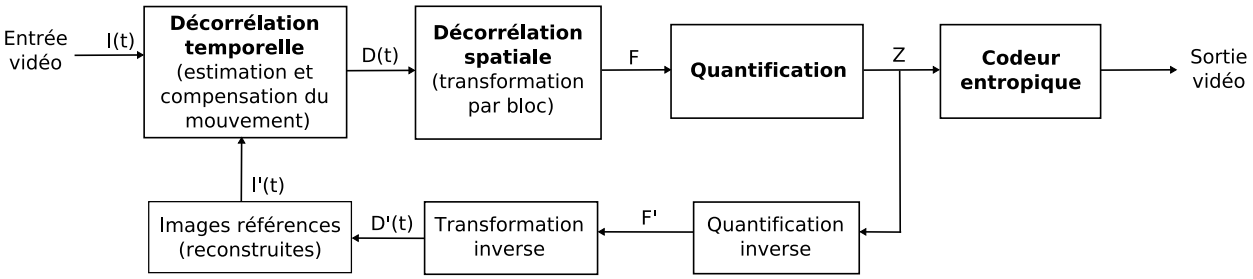


FIG. 1.2 – Codeur vidéo.

Le flux en entrée du codeur est une séquence vidéo couleur non comprimée. Celle-ci est représentée par un 3-uplet dont chaque élément indique une composante. La représentation RVB donne respectivement la densité de rouge (R), de vert (V) et de bleu (B) (on trouve également le sigle RGB pour *Red Green Blue* en anglais). Mais en codage d'images, on utilise généralement les trois composantes Y (luminance), Cb et Cr (différences de chrominances), le passage d'un espace à l'autre étant réalisé par une transformation linéaire [GP02]. La luminance représente les niveaux de gris de l'image, et les deux composantes Cb et Cr des différences de chrominances permettent la coloration. L'œil humain étant plus sensible à la luminance qu'aux couleurs, l'espace $YCbCr$ (ou YUV) se prête bien à la compression d'image. En utilisant les formats 4:2:2 ou 4:2:0, on sous-échantillonne horizontalement (4:2:2) et aussi verticalement (4:2:0) les signaux de chrominance U et V.

Le premier bloc (figure 1.2) a pour objectif de tirer parti de la redondance temporelle en exploitant les similarités entre les images ou parties d'images voisines dans le temps. Une image de prédiction I_p de l'image courante $I(t)$ est produite grâce à une ou des images de référence $I'(t)$. En sortie de l'étape de décorrélation temporelle, on obtient alors une « image résiduelle » $D(t)$ (soustraction de l'image de prédiction à l'image courante) et un jeu de paramètres lié à cette étape, tel par exemple qu'un ensemble de vecteurs de mouvement décrivant la compensation de mouvement. Nous détaillerons ce procédé dans la suite de cette partie. L' « image résiduelle » $D(t)$ est présentée à l'entrée du second bloc de décorrélation spatiale qui exploite les similarités entre les pixels voisins pour réduire la redondance spatiale. Ce traitement correspond souvent à une transformation produisant à partir de l' « image résiduelle » $D(t)$, un ensemble 2D d'éléments transformés appelés coefficients F. Ceux-ci sont ensuite par quantification, représentés de façon approchée (Z). Ceci se traduit également par l'annulation des coefficients d'amplitude non significative. Les coefficients quantifiés Z ainsi que les vecteurs de mouvement sont transmis au codeur entropique pour une dernière étape de codage. Celle-ci exploite (sans perte) les propriétés statistiques des données et produit un flux binaire comprimé structuré transmis ou stocké. Un flux vidéo comprimé est donc exactement hiérarchisé et composé de blocs, de paramètres des vecteurs de mouvement, de coefficients résiduels quantifiés et de leurs entêtes. Il est intéressant de noter, dans ce schéma de compression avec pertes, que les décodées locales $I'(t)$ sont prises en compte pour l'analyse temporelle afin de construire par compensation de mouvement les images de prédiction. En effet, afin d'éviter qu'un phénomène de dérive temporelle ne s'ajoute aux pertes « admises » dues à la quantification, on reconstitue à l'encodeur le niveau de qualité des images telles qu'elles seront décodées (si pas d'erreurs de transmission) et ces décodées formant les images de référence.

1.1.2 La décorrélation temporelle

Le but de ce bloc de traitement est de réduire la redondance temporelle entre les images successives transmises, en formant une image prédite $I_p(t)$ à partir d'une ou plusieurs images passées ou futures codées/décodées :

$$I_p(t) = f(I(t), I'(t)) \tag{1.1}$$

et en la soustrayant à l'image courante, $I(t)$:

$$D(t) = I(t) - I_p(t) \tag{1.2}$$

En sortie de cette étape, on obtient une image résiduelle $D(t)$ (figure 1.2). Plus la prédiction est performante, moins l'image résiduelle résultante contiendra d'information. Ce type de prédiction est nommée prédiction inter-images.

1.1.2.1 Prédiction depuis l'image précédente

La méthode de prédiction temporelle la plus simple consiste à utiliser une des images précédentes décodées comme prédicteur de l'image courante (cf équation 1.1). On parle alors d'images P pour images prédites. Deux images successives de la séquence *Football* sont illustrées à la figure 1.3. L'image 20 est utilisée directement comme prédicteur de l'image 21 et le résidu (obtenu en soustrayant l'image 20 à l'image courante, l'image 21) est illustré à la figure 1.3(c). Dans cette image différentielle, le noir représente une différence nulle et indique une décorrélation temporelle optimale, alors que les zones plus claires correspondent aux autres valeurs absolues des différences. Le problème évident avec cette simple méthode de prédiction, réside dans le fait que l'image résiduelle contient encore beaucoup d'informations à comprimer. Cette information résiduelle est essentiellement due aux mouvements des objets entre les deux images. On veut alors effectuer une estimation de mouvement suivie d'une compensation de mouvement entre les deux images pour obtenir une meilleure prédiction et réaliser une meilleure décorrélation temporelle.

(a) image 20 de la séquence *Football.*

(b) image 21 de la séquence *Football.*

(c) valeurs absolues des différences entre les images 21 et 20 de la séquence *Football.*

FIG. 1.3 – Prédiction temporelle depuis l'image précédente.

1.1.2.2 Changements dus au mouvement

L'origine des changements entre des images peut être due au mouvement d'un objet rigide (par exemple une voiture), ou déformable (par exemple un bras), au mouvement de la caméra (translation, rotation,

zoom...), à une région découverte ou recouverte (par exemple, une zone du fond qui se découvre lors du mouvement d'un objet), ou à un changement de l'éclairage. Mis à part le découvrement ou le recouvrement de régions et la variation de luminosité, aux autres différences correspondent en première approximation à des mouvements de pixels d'une image à l'autre. Il est donc possible d'estimer les trajectoires de ces pixels entre deux images successives, et de produire un champ de trajectoires des pixels nommé flot optique [HS81] (*optical flow*). Si le flot optique est précisément connu et dense, une prédiction correcte de la majorité des pixels de l'image courante s'obtient en déplaçant les pixels de l'image de référence selon les vecteurs du flot optique. Mais en pratique, cette méthode de compensation de mouvement n'est pas retenue, en raison du grand nombre de calculs nécessaires, et de l'obligation de transmettre un vecteur par pixel (augmentation du débit) pour le décodage.

1.1.2.3 Estimation et compensation de mouvement par bloc

Cette méthode, que l'on retrouve majoritairement en compression vidéo, compense le mouvement de blocs rectangulaires de pixels d'une image. Pour chaque bloc B de $L \times H$ pixels, on applique classiquement l'algorithme suivant (dit de *block matching*) :

1. La première étape est l'estimation de mouvement. Soit I_t l'image à l'instant t et I'_{t-1} l'image référence qui pour illustrer notre exemple est donc choisie précédente de l'image courante. Il s'agit, pour tout bloc B_t de I_t, de rechercher dans I'_{t-1} le bloc B'_{t-1} le plus « proche ». Cette proximité est exprimée via le calcul d'une distance, la plus classique étant la SAD[1] (*Sum of Absolute Difference*) :

$$SAD_{(B_t(x,y),B'_{t-1}(x,y))}(u,v) = \sum_{j=0}^{H-1}\sum_{i=0}^{L-1} |B_t(x+i,y+j) - B'_{t-1}(x+u+i,y+v+j)| \tag{1.3}$$

   on cherche donc le déplacement au pixel près minimisant la distance :

$$(u',v') = \min_{(u,v)\in F} SAD(B_t(x,y),B'_{t-1}(x+u,y+v)) \tag{1.4}$$

   où F est une fenêtre de recherche choisie afin de limiter l'exploration à une région de taille inférieure à l'image et centrée sur la position (x,y).

2. L'étape suivante est la compensation de mouvement. Le bloc B'_{t-1} choisi devient le prédicteur du bloc courant de taille $L \times H$. On opère alors une soustraction entre ces deux blocs pour obtenir un bloc résiduel :

$$\hat{B}_t(x,y) = B_t(x,y) - B'_{t-1}(x+u',y+v') \tag{1.5}$$

3. Le bloc résiduel $\hat{B}_t(x,y)$ est ensuite codé $\tilde{B}_t(x,y)$ et transmis ainsi que le vecteur de mouvement correspondant (u',v').

De son côté, le décodeur recrée le bloc prédicteur d'après le vecteur de mouvement (u',v') qu'il a reçu, décode le bloc résiduel $\tilde{B}_t(x,y)$, l'ajoute au prédicteur et reconstruit une version du bloc originel :

$$B'_t(x,y) = B'_{t-1}(x+u',y+v') + \tilde{B}_t(x,y) \tag{1.6}$$

Deux images successives (20 et 21) de la séquence vidéo *Vectra* sont présentées figure 1.4. La figure 1.5 présente les résidus obtenus en réalisant une différence inter-image entre les images 20 et 21(figure 1.5(a)),

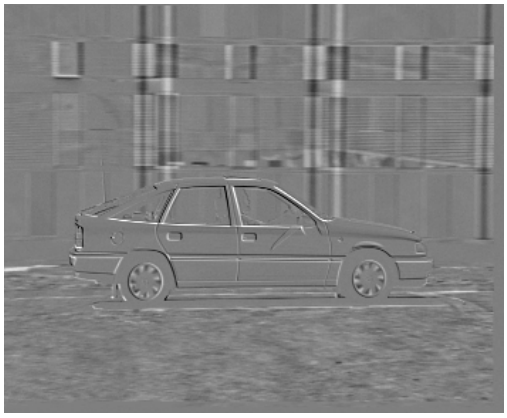
[1] La MSE (*Mean Square Error*) peut également être utilisée (voir la section 1.2).

et une différence inter-image compensée après une estimation de mouvement à l'aide de blocs de taille 16×16 (figure 1.5(b)). Nous pouvons alors constater combien l'énergie résiduelle est largement réduite par l'utilisation de la méthode d'estimation et de compensation de mouvement.

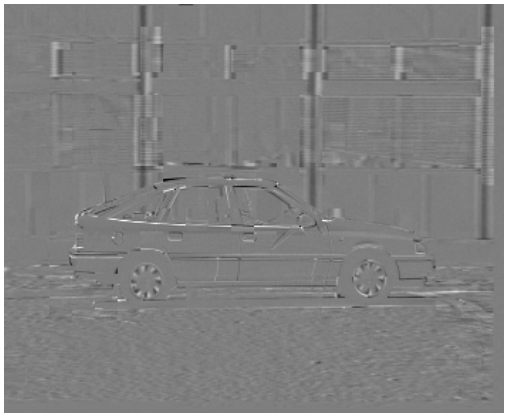
(a) image 20.

(b) image 21.

FIG. 1.4 – Images 20 et 21 de la séquence vidéo *Vectra*.



(a) résidu sans compensation de mouvement.



(b) résidu avec compensation de mouvement (blocs de taille 16×16).

FIG. 1.5 – Énergie résiduelle avec et sans compensation de mouvement.

Cette méthode doit sa popularité à sa relative simplicité, son adéquation aux formes rectangulaires des images vidéo et à la transformation de bloc de pixels qui suit (par exemple la Transformée en Cosinus Discrète ou TCD de l'anglais DCT pour *Discrete Cosine Transform*, cette notion sera abordée plus loin dans le chapitre). Mais elle présente des limites. Par exemple les objets « réels » ont rarement des bords réguliers s'adaptant aux bords rectangulaires, les objets ne se déplacent pas toujours d'un nombre entier de pixel(s) et de nombreux mouvements complexes d'objets sont difficiles à compenser ainsi. Malgré ces inconvénients, l'estimation et la compensation de mouvement par bloc sont à la base des techniques de décorrélation temporelle utilisées par tous les standards actuels de codage vidéo.

1.1.2.4 Prédiction par compensation de mouvement d'un macrobloc

Un macrobloc correspond typiquement a un carré de 16×16 pixels. C'est l'unité de base pour la prédiction par compensation de mouvement de nombreux standards incluant MPEG-1, MPEG-2, MPEG-4 et H.264. La figure 1.6 présente la structure d'un macrobloc dans une vidéo au format 4:2:0.

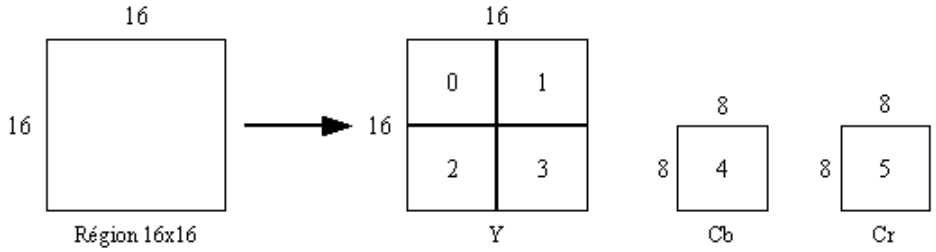


FIG. 1.6 – Structure d'un macrobloc (4:2:0).

Les mécanismes vus précédemment (équations 1.3, 1.4, 1.5 et 1.6) pour l'estimation et la compensation de mouvement s'appliquent donc directement au cas des macroblocs. En pratique, l'estimation de mouvement est réalisée sur les blocs de luminance (Y), le vecteur de mouvement (u,v) ainsi obtenu est adapté pour compenser les blocs 8×8 $C_b$et C_r.

Ainsi la décorrélation temporelle en prédisant le mouvement permet de réduire la redondance temporelle. Les vecteurs de mouvement sont transmis au codeur entropique et les résidus sont traités lors de l'étape suivante de décorrélation spatiale, que nous allons décrire.

1.1.2.5 Bi-prédiction

Afin d'exploiter au mieux la prédiction temporelle, le concept d'images prédites bi-directionnellement fut introduit pour la norme MPEG-1 (figure 1.7). On parle alors d'images B qui sont également appelées images prédites en arrière (*backwards-predicted frames* en anglais) ou images bi-prédites. Les images B sont assez similaires aux images P, à la différence qu'elles peuvent être prédites à partir de deux images de référence : une antérieure et l'autre postérieure à l'image courante. Le décodeur doit aussi décoder la prochaine image I (décrite dans la section suivante 1.1.2.6) ou P, utilisée comme référence future par l'image B, avant de décoder celle-ci et de pouvoir l'afficher. Le codage/décodage des images B est donc plus complexe et requiert de grandes tailles de mémoires tampon pouvant provoquer des retards côté codeur et décodeur.

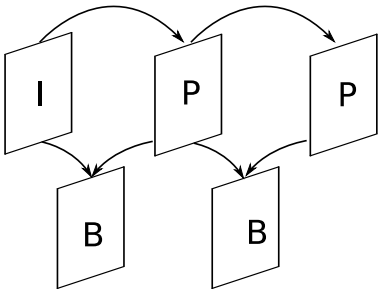


FIG. 1.7 – Images prédites et bi-prédites.

1.1.2.6 Structure et taille des GOP

Les codeurs vidéo utilisent plusieurs types d'images codées qui ont différents buts. Le type d'image codée le plus important et le plus simple à obtenir sont celles de type I. Une image I est une image codée en mode intra-image, appelée ainsi car elle peut être décodée indépendamment des autres images. Elles sont parfois également appelées images clés (*keyframes* en anglais). L'encodage d'une image I est très rapide, mais généralement avec moins d'efficacité en termes de taux de compression : une vidéo encodée avec

seulement des images I aura en moyenne une taille trois fois plus grande comparée à la taille d'une vidéo encodée normalement au format MPEG-1 [LG91].

MPEG-1 définit des « *Groups Of Pictures* » ou GOP qui décrivent la succession des images I, P et B. En effet, les codeurs vidéo exploitent la redondance temporelle entre les images successives d'une séquence vidéo et pour cela, ils utilisent des modes de codage P et B (prédit et bi-prédit). Le mode P assure une compression efficace en utilisant des vecteurs de mouvement relatifs à l'image décodée précédente (I ou P). Le mode B est le plus efficace en termes de compression mais nécessite les images décodées précédentes et futures pour être prédites. Le décodage des images B nécessite donc l'usage d'une mémoire plus importante dans le codeur et le décodeur.

1.1.2.6.1 Structure du GOP

Un GOP débute toujours par une image I. Ensuite, plusieurs images P suivent à des intervalles réguliers. Dans les « espaces » entre deux images P ou entre une image P et une image I, une ou plusieurs images B sont intercalées. Certains codeurs vidéo permettent d'utiliser des GOP contenant plus d'une image I.

Plus le flux généré par un codeur de type MPEG contient des images codées en mode intra (I), plus il est éditable. Cependant, la taille des images codées en intra (en termes de bits) est plus importante que celles des images P ou B, et augmenter le nombre d'images I au sein du GOP aura donc pour conséquence d'augmenter la taille de la vidéo encodée. Afin de limiter la bande passante ou l'espace de stockage nécessaire, les vidéos encodées pour la diffusion sur internet n'ont généralement qu'une seule image I par GOP.

1.1.2.6.2 Taille du GOP

La distance entre deux images I successives est appelée la taille du GOP. Le standard MPEG-1 utilise généralement des GOP de taille entre 15 et 18, ce qui signifie qu'il y a une image I toutes les 14 ou 17 images (combinaison d'images P et B).

La structure du GOP est souvent indiquée par deux nombres, par exemple $M = 3$ et $N = 12$. Le premier indique la distance entre deux images d'ancrage références (I ou P), le second indique la distance entre deux images codées en intra (I) : c'est la longueur du GOP. La structure du GOP de l'exemple où $M = 3$ et $N = 12$ est alors : IBBPBBPBBPBB.

1.1.3 La décorrélation spatiale

Les images naturelles sont souvent difficiles à compresser dans leur état original en raison de la forte corrélation entre les échantillons voisins. L'estimation et la compensation de mouvement permettent de réduire la corrélation temporelle locale dans l'image résiduelle et ainsi de faciliter la compression de l'image originale. L'objectif du bloc de traitement suivant est d'exploiter davantage la corrélation résiduelle (on trouve dans la littérature le terme « décorréler ») spatiale d'une image ou des données résiduelles pour obtenir une forme efficacement compressible par le codeur entropique. Classiquement ce bloc est constitué d'une transformation qui sera suivie d'une quantification (si codage avec perte d'information).

1.1.3.1 La transformation

La transformation doit permettre de convertir une image ou un résidu dans un autre domaine, le domaine de la transformation. Plusieurs critères sont nécessaires pour le choix de la transformation :

1. L'information dans le domaine de transformation doit être décorrélée et compacte.

2. La transformation doit être réversible.
3. La transformation doit être envisageable en termes de coûts de calculs.

De nombreuses transformations ont été proposées pour la compression d'images et de vidéos. Les plus usuelles se classent en deux catégories : les transformations de blocs de pixels et celles sur l'image entière. Par exemple, la TCD [RY90] opère généralement sur des blocs $N \times N$ d'une image ou du résidu et procède par unités de bloc. On obtient en sortie un bloc $N \times N$ de coefficients. Les transformations de bloc de pixels nécessitent une faible quantité de mémoire et conviennent convenablement à la compression des résidus de l'estimation et compensation de mouvement par bloc (section 1.1.2.3) mais peuvent favoriser l'apparition d'artéfacts a la frontière des blocs. *A contrario*, la Transformée en Ondelettes Discrète [Mal89, Mal90] (TOD, DWT pour *Discrete Wavelet Transform*) opère sur l'image entière ou sur une large portion de l'image appelée tuile (*tile* en anglais). Classiquement, l'opération consiste à appliquer au signal une paire de filtres passe-haut et passe-bas pour obtenir une décomposition de ce dernier en deux signaux de bande réduite : l'un correspond aux basses fréquences (L) et l'autre correspond aux hautes fréquences (H) (bandes de fréquences de même largeur). Puis l'opération est ensuite réitérée sur ces deux signaux selon l'autre direction. Pour un sous-échantillonnage critique, chaque bande est sous-échantillonnée par un facteur deux et contient ainsi $\frac{N}{2} \times \frac{N}{2}$ échantillons. Ces coefficients représentent généralement le poids d'une composante fréquentielle. Avec des filtres appropriés, cette opération est réversible. Il a été observé que les transformations s'appliquant sur l'image entière telle que la TOD obtenaient de meilleurs résultats que les transformations de bloc de pixels pour la compression des images fixes, mais leurs exigences en termes de mémoire sont plus importantes et ne s'accordent pas correctement aux méthodes d'estimation et de compensation du mouvement par bloc. La TCD quant à elle, est plus adaptée aux schémas de codage vidéo réalisant une estimation et compensation de mouvement par bloc et est souvent retenue par les normes de codage vidéo, telles que le standard H.264 (cf chapitre 4). De plus, l'espace de représentation des coefficients issus de la TCD peut être exploité d'un point de vue perceptuel, via des méthodes adaptées pour leur quantification (section 1.2.2.1.1).

1.1.3.2 Transformation en cosinus discrète

Nous décrivons ici la TCD car c'est la plus répandue en codage vidéo. La transformation en cosinus discrète s'applique à un bloc X de $N \times N$ pixels (typiquement des échantillons d'images, ou les valeurs des résidus après la prédiction) et crée un bloc F de $N \times N$ coefficients. L'action de la TCD et de son inverse (IDCT pour *Inverse Discrete Cosine Transform* en anglais) peut être décrite à l'aide d'une matrice de transformation A (ou noyau). La TCD d'un bloc de $N \times N$ échantillons est donnée par :

$$F = AXA^T \tag{1.7}$$

et la TCD inverse par :

$$X = A^T FA \tag{1.8}$$

où X est la matrice des échantillons, F est la matrice des coefficients et A est la matrice de transformation de taille $N \times N$. Les éléments de A sont :

$$A_{ij} = C_i \cos \frac{(2j+1)i\pi}{2N} \text{ où } C_i = \sqrt{\frac{1}{N}} \text{ pour } i = 0 \text{ et } C_i = \sqrt{\frac{2}{N}} \text{ pour } i > 0 \tag{1.9}$$

Les équations 1.7 et 1.8 peuvent être écrites sous la forme :

$$F_{xy} = C_x C_y \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{ij} \cos \frac{(2j+1)y\pi}{2N} \cos \frac{(2i+1)x\pi}{2N} \tag{1.10}$$

$$X_{ij} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C_x C_y F_{xy} \cos \frac{(2j+1)y\pi}{2N} \cos \frac{(2i+1)x\pi}{2N} \tag{1.11}$$

La sortie d'une transformation en cosinus discrète est un ensemble de $N \times N$ coefficients représentant les données du bloc de l'image dans le domaine de la transformation et ces coefficients peuvent être considérés comme les poids de fonctions de base standard. Les fonctions de base pour une transformation en cosinus discrète de taille 8×8 sont illustrées visuellement figure 1.8.

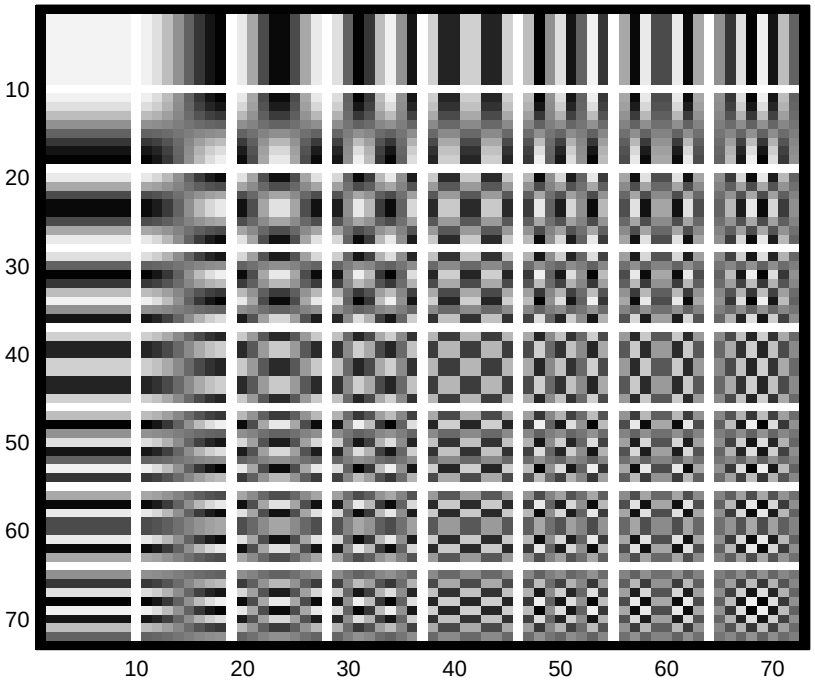


FIG. 1.8 – Fonctions de base d'une transformation en cosinus discrète de taille 8×8.

1.1.4 La quantification

La quantification est utilisée pour réduire la précision des données d'une image après l'étape de transformation en supprimant les valeurs insignifiantes (exemple : les coefficients de la TCD proches de zéro). Au sein d'un codeur vidéo ou d'images fixes, la quantification a donc pour but de mettre à zéro les coefficients insignifiants tout en conservant un nombre réduit de coefficients significatifs non nuls. La sortie d'un quantificateur est typiquement un tableau clairsemé de coefficients quantifiés, contenant principalement des zéros (on parle de matrice « creuse »).

Formellement, un quantificateur transforme chaque échantillon d'un signal Y en un échantillon discret quantifié Z ayant une gamme plus réduite de valeurs. Il est ainsi possible d'encoder le signal quantifié avec moins de bits. En compression vidéo, les coefficients issus de la transformation sont quantifiés. Un quantificateur scalaire fait correspondre à chaque échantillon du signal d'entrée une valeur quantifiée, et un

quantificateur vectoriel[2] fait correspondre directement à un groupe d'échantillons (un « vecteur ») un groupe de valeurs quantifiées. On obtient bien compression mais cela implique dans les deux cas une dégradation irréversible du signal. On parle ici de distorsion du signal (ou d'erreur de quantification). Bien que la quantification vectorielle soit efficace pour le codage audio, elle est cependant moins adaptée à la compression de séquences vidéo. En effet, la diversité importante des séquences vidéo naturelles (leur non « stationnarité » intrinsèque) ne permet pas d'obtenir un dictionnaire de vecteurs optimal pour la compression vidéo. C'est pourquoi dans la suite du document nous détaillerons seulement la quantification scalaire retenue par les différentes normes de codage vidéo (MPEG-1, MPEG-2, MPEG-4, H.264…).

1.1.4.1 Quantification scalaire

Un exemple simple de quantification scalaire est le procédé d'arrondi d'un nombre décimal à l'entier le plus proche, la fonction se fait du domaine $\mathbb{R}$ vers le domaine $\mathbb{Z}$. C'est une méthode de compression avec pertes (non réversible) puisqu'il n'est plus possible de déterminer la valeur exacte du nombre réel original à partir de l'entier arrondi.

Un exemple plus général d'une quantification uniforme est donné par :

$$\begin{aligned} Z &= arrondi(\tfrac{F}{QP}) \\ F' &= Z.QP \end{aligned} \tag{1.12}$$

où QP est le pas de quantification, F le nombre à quantifié, Z est la valeur quantifiée et F' la valeur reconstruite. Les intervalles des valeurs quantifiées en sortie sont espacés de façon uniforme (et de taille QP).

La figure 1.9 illustre deux exemples de quantificateurs scalaires, un quantificateur uniforme linéaire (avec une fonction linéaire entre les valeurs d'entrée et de sortie) et un quantificateur non linéaire qui possède une « zone morte » (de l'anglais *dead zone*) autour de la valeur zéro (les valeurs proches de zéro en entrée prendront la valeur zéro en sortie).

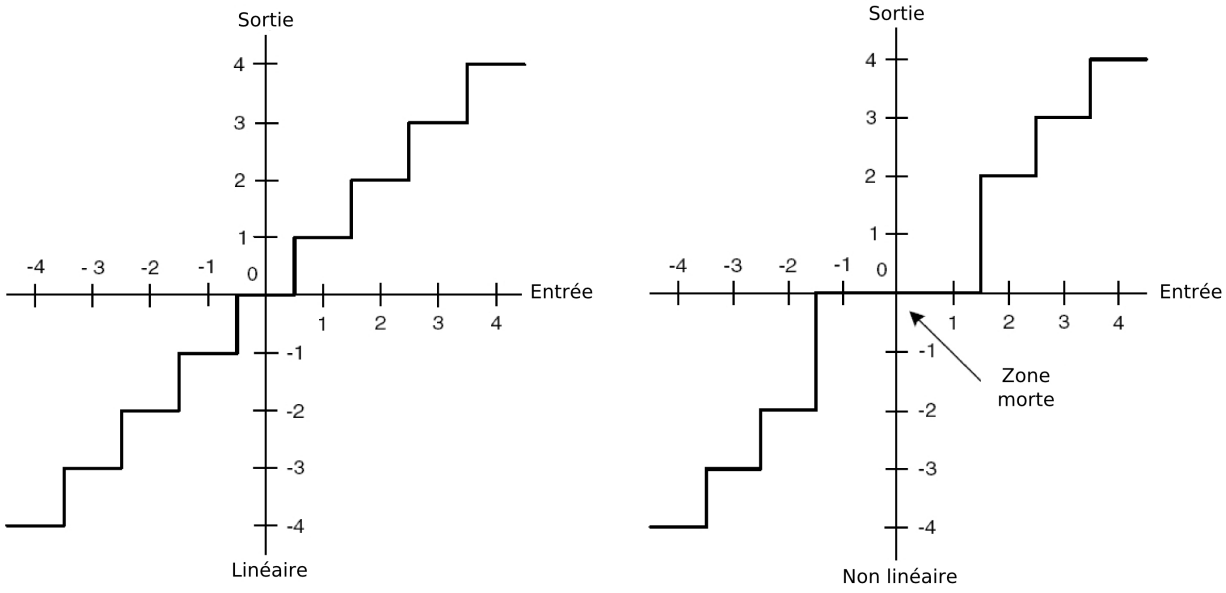


FIG. 1.9 – Quantificateurs scalaires : uniforme linéaire et non linéaire avec une zone morte.

[2]La quantification vectorielle ne sera pas abordée ici, cependant le lecteur intéressé pourra consulter l'ouvrage de référence de Gersho and Gray [GG92].

Dans les systèmes de compression de vidéos et d'images, l'opération de quantification est généralement constituée de deux parties : le quantificateur (*forward quantizer*) au sein du codeur et un quantificateur inverse au sein du décodeur (la quantification n'étant pas réversible, l'expression « mise à l'échelle » serait plus adaptée). Un paramètre critique est la taille du pas de quantification, *QP*, entre les valeurs « déquantifiées » successives. Si la taille du pas de quantification est importante, le nombre des valeurs quantifiées est faible et de ce fait, elles peuvent être efficacement représentées (forte compression), au détriment des valeurs reconstruites qui deviennent des approximations grossières du signal original pouvant entraîner une perte de qualité. Et si la taille du pas de quantification est faible, les valeurs reconstruites correspondent plus étroitement au signal original, mais le nombre important des valeurs quantifiées réduit les performances en termes de compression.

1.1.5 Le codeur entropique

Pour finir, nous détaillons le codeur entropique qui est la dernière étape du codeur. Le codeur entropique convertit une série de symboles représentant les éléments de la séquence vidéo en un flux binaire compressé approprié pour la transmission ou le stockage. Les symboles présents en entrée du codeur entropique peuvent représentés les coefficients transformés, quantifiés et réordonnés (encodage *run-level*), les vecteurs de mouvements (un vecteur de déplacement pour chaque bloc compensé en mouvement), des données de marquages (pour indiquer une resynchronisation), des en-têtes (en-têtes de macroblocs, en-têtes d'images, en-têtes de séquences, etc.) et d'autres informations supplémentaires. L'objectif de la thèse sera de proposer un outil de pré-analyse afin d'exploiter au mieux les capacités des différents outils au sein d'un codeur vidéo. L'étape du codage entropique peut aussi être optimisée comme nous le verrons avec la norme H.264 (cf chapitre 4). Nous allons présenter dans cette section deux méthodes qui illustrent le principe d'un tel codage entropique.

1.1.5.1 Codage à longueur variable

Un codeur à longueur variable transforme les symboles d'entrée en une série de mots de code (codes de longueur variable ou VLC[3] en anglais), ceux-ci sont donc de longueur variable mais contiennent un nombre entier de bits. Les symboles ayant une fréquence d'occurrence importante sont représentés avec des mots de code de faible taille, tandis que les symboles moins fréquents sont représentés avec des mots de code de taille plus importante. Pour un nombre suffisamment grand de symboles, cela conduit à la compression moyenne des données.

1.1.5.1.1 Codage de type *Huffman*

Le codage de type *Huffman* assigne un code de longueur variable à chaque symbole. D'après la méthode originale proposée par Huffman en 1952 [Huf52], il est d'abord nécessaire de calculer la probabilité d'occurrence de chaque symbole, puis de construire un ensemble de mots de code à longueur variable.

Si les distributions des probabilités sont l'inverse de puissances de deux, le codage de type *Huffman* permet d'obtenir une représentation optimale des données originales.

1.1.5.1.2 Codage de type *Huffman* pré-calculé

Le procédé de codage de type *Huffman* a deux inconvénients pour un codec vidéo. D'abord, le décodeur

[3] *Variable Length Codes*

doit utiliser le même ensemble de mots de code que le codeur, ce qui implique, que la table des probabilités calculée du côté du codeur, soit transmise. Ceci ajoute un surplus de données à transmettre et réduit l'efficacité de la compression, particulièrement pour de courtes séquences vidéos. Ensuite, la table des probabilités d'une longue séquence vidéo ne peut pas être calculée avant que toutes les données de la vidéo ne soient encodées, ce qui introduit un délai inacceptable dans le procédé de codage. Pour ces raisons, les standards récents pré-définissent un ensemble de mots de code basé sur des distributions de probabilité génériques [MPE01] (mais souvent sous-optimales).

1.1.5.1.3 Autres types de codes à longueur variable

Un autre inconvénient majeur des codes de type Huffman, réside dans le fait qu'ils sont sensibles aux erreurs de transmission. Une erreur dans une séquence de codes à longueur variable, peut entraîner la perte de la synchronisation et empêcher le décodage des mots de code ultérieurs, entraînant la propagation d'erreurs dans la séquence décodée. Les codes à longueur variable réversibles [MPE01] peuvent décoder efficacement dans les deux directions. Si en cours de décodage une partie de la séquence s'avère indécodable en raison d'une corruption trop élevée du signal, le décodage pourra reprendre en sens inverse en partant de la fin de la séquence (ou du prochain point de synchronisation) pour sauver l'information qui peut l'être. Un inconvénient de ces tables de mots de code prédéfinies, est qu'elles doivent être stockées du côté du codeur et du décodeur. Une approche alternative est d'utiliser des codes qui peuvent être générés automatiquement (à la volée) si le symbole d'entrée est connu. Les codes de Golomb exponentiels [Gol66] utilisés au sein de la norme H.264 font partie de cette catégorie et sont décrits dans la chapitre 4.

1.1.5.2 Codage arithmétique

Les schémas de codage à longueur variable décrits dans la section précédente présentent l'inconvénient fondamental d'assigner un mot de code contenant un nombre entier de bits pour chaque symbole alors que le nombre optimal de bits pour un symbole est généralement un nombre décimal. L'efficacité de compression avec ces codes à longueur variable est donc faible pour les symboles dont la probabilité est supérieure à $0,5$, car la représentation alors offerte sera d'utiliser un mot de code à un seul bit.

Le codage arithmétique amène une alternative pratique au codage de type *Huffman* et permet de se rapprocher des taux théoriques maximum de compression [WNC87]. Un codeur arithmétique convertit une séquence de symboles de données en un simple nombre décimal et s'approche du nombre décimal optimal de bits requis pour représenter chaque symbole.

1.1.5.2.1 Codage arithmétique basé sur le contexte

L'efficacité du codage entropique dépend de l'exactitude des modèles de la probabilité d'occurrence des symboles. Le codage arithmétique basé sur le contexte (CAE[4] en anglais) utilise des caractéristiques spatiales et/ou temporelles pour estimer la probabilité d'occurrence du symbole à coder. Le codage arithmétique basé sur le contexte est utilisé dans le standard JBIG [JBI], il a été aussi adopté pour le codage des masques binaires des formes dans MPEG-4 Visual [MPE01], et dans le profil principal de H.264 (voir le chapitre 4).

1.1.6 Conclusion

Dans cette première partie du chapitre, nous avons abordé les différentes étapes d'un schéma de compression vidéo. Nous avons d'abord principalement détaillé l'étape de décorrélation temporelle. Ce bloc de

[4] *Context-based Arithmetic Encoding*

traitements coûteux en calculs, permet de supprimer la redondance temporelle entre les images successives d'une séquence vidéo. Comme nous le verrons dans la suite du document, il est l'objet de nombreuses études afin d'optimiser les temps de calcul et la compression au sens débit-distorsion. Mais les décisions prises par le codeur lors de la décorrélation temporelle, à savoir le choix du bloc correspondant dans l'image référence, ne sont prises qu'à très court terme. En effet, hormis les images *B* où le codeur dispose d'informations sur la prochaine image *P* (future), celui-ci prend des décisions locales afin de minimiser l'erreur du bloc courant. Dans certains cas, ces décisions locales peuvent s'avérer temporellement incohérentes et provoquer l'apparition de dégradations visuellement perceptibles, telles que des effets de papillotement (*flickering effect* en anglais). Les codeurs actuels ne bénéficient d'aucun outil leur permettant de réaliser une analyse à long terme avant d'effectuer leur choix de codage. La solution envisageable est donc de réaliser une pré-analyse de la vidéo. Disposant ainsi d'informations sur la séquence vidéo et les évolutions des objets la constituant, le codeur pourra prendre des décisions à moyen/long terme assurant une meilleure cohérence de codage.

Ensuite, nous avons abordé l'étape de décorrélation spatiale souvent réalisée par une transformation orthogonale. Cette dernière est suivie d'une étape de quantification et enfin d'un codage entropique. Cette dernière partie a été abordée brièvement, l'objectif étant de réaliser une description suffisante pour la compréhension de la thèse.

Comme nous le verrons dans le chapitre 4 faisant une description du codeur H.264, de nombreuses optimisations portant sur les différents blocs de traitement du codeur, ont été intégrées au nouveau standard de codage. Nous nous situons dans une problématique différente, puisque l'objectif de la thèse n'est pas de concevoir un ou plusieurs nouveaux blocs de traitement du codeur H.264, mais d'utiliser au mieux les capacités des différents outils disponibles après pré-analyse de la vidéo. Les étapes de codage vidéo les plus coûteuses en termes de calcul sont celles de prédictions (surtout celle temporelle), et la plus décisive est la quantification. C'est pourquoi, avec *a priori*, dans la suite de ce premier chapitre, nous allons présenter et commenter les différentes méthodes d'optimisation au sens débit-distorsion d'un codeur et plus particulièrement celles portant sur la décorrélation temporelle.

1.2 Méthodes d'optimisation au sens débit-distorsion

L'un des principes fondamentaux du codage vidéo à compression élevée avec pertes est le choix adapté des paramètres de quantification et des modes de codage vidéo (en fait, on le verra par la suite, tout un jeu de paramètres conditionne le débit et la distorsion). La sélection de ces paramètres peut être optimisée à l'aide des techniques de minimisation de Lagrange qui fondent la théorie de débit-distorsion [Ber71].

L'optimisation consiste à choisir pour chaque bloc de l'image la représentation codée la plus efficace au sens débit-distorsion. Ce procédé devient complexe lorsque les diverses options de codage varient en efficacité selon le débit binaire et le contenu de la scène. Ainsi le codage inter-image est efficace pour représenter un contenu évolutif de séquences vidéos, cependant le codage intra peut être plus efficace lorsque le modèle de mouvement translationnel utilisé par l'estimation de mouvement ne peut représenter avec suffisamment d'exactitude les changements dans la séquence vidéo. Le but du système de compression vidéo est d'obtenir la meilleure qualité compte tenu de la contrainte de débit. L'optimisation peut être conduite via la méthode des multiplicateurs de Lagrange s'exprimant de la manière suivante :

$$J = D + \lambda \times R \tag{1.13}$$

où J est le coût débit-distorsion estimé, D est la distorsion calculée, λ est le multiplicateur de Lagrange

et R indique le nombre de bits nécessaires pour coder les données (en tenant compte du paramètre de quantification, des vecteurs de mouvement et de tous les résidus des coefficients TCD quantifiés). Les critères les plus usuels pour l'évaluation de la distorsion D sont basés sur des mesures mathématiques simples telles que des distances ou des mesures issues du traitement du signal comme l'erreur quadratique moyenne (MSE[5]) ou le rapport signal à bruit (SNR[6]). Les normes L_p (ou leur logarithme) entre deux images I et I' (typiquement l'image originale et l'image codée/décodée) sont souvent utilisées :

$$L_p(I,I') = \left(\frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} |I(i,j) - I'(i,j)|^p \right)^{\frac{1}{p}} \tag{1.14}$$

où N et M représentent les dimensions des images. Pour $p = 2$, nous avons la RMSE (*Root MSE*), la MSE est alors donnée par $(L_2)^2$, ou encore :

$$MSE(I,I') = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} |I(i,j) - I'(i,j)|^2 \tag{1.15}$$

Le rapport signal à bruit est une mesure souvent utilisée en traitement du signal, donné par :

$$SNR = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} I(i,j)^2}{\sum_{i=1}^{N} \sum_{j=1}^{M} |I(i,j) - I'(i,j)|^2} \tag{1.16}$$

Les concepteurs de schémas de codage utilisent plutôt le PSNR (*Peak Signal to Noise Ratio*) pour apprécier les performances de leur codeur :

$$PSNR_{dB} = 10 log_{10} \left(\frac{I_{max}^2}{\frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} |I(i,j) - I'(i,j)|^2} \right) \tag{1.17}$$

où I_{max} est la valeur maximale qu'un pixel peut avoir ($I_{max} = 255$ pour un codage sur 8 bits).

La minimisation de J, c'est-à-dire celle conjointe de D et de R peut être résolue par des méthodes différentes (méthodes analytiques, programmation dynamique...). Cependant, dans un contexte de codage vidéo, les solutions analytiques sont généralement inappropriées (le nombre de paramètres à considérer est trop important). Ce sont donc les techniques de programmation dynamique qui sont retenues. Typiquement, on fait varier un paramètre (par exemple le pas de quantification) et on mesure l'influence de ce changement.

Afin d'optimiser le coût débit-distorsion, des méthodes ont été proposées dans la littérature, on distingue deux types d'approches. La première est une approche signal, également qualifiée de « bas niveau » et elle peut être prédictive ou adaptative. La deuxième utilise des propriétés du SVH (qui seront décrites dans le chapitre suivant) et sera qualifiée de « haut niveau ».

1.2.1 Approches bas niveau

1.2.1.1 Méthodes adaptatives

L'estimation de mouvement réalisée entre l'image courante et une image référence précédemment encodée/décodée permet d'exploiter la corrélation temporelle entre les images d'une séquence vidéo et d'atteindre des taux de compression importants tout en maintenant la qualité de la vidéo décodée. De nombreuses techniques d'optimisation de type adaptatif ont été proposées afin d'améliorer la prédiction inter-

[5] Mean Square Error
[6] Signal to Noise Ratio

image de l'image courante, celles-ci sont décrites ci-après. L'objectif est de réduire l'erreur de prédiction de l'image courante, D, tout en limitant le nombre (R) de bits nécessaires au codage de celle-ci.

1.2.1.1.1 Influence de la taille des blocs

Deux images successives d'une séquence vidéo sont montrées à la figure 1.10 ((a) et (b)). Les énergies résiduelles obtenues après estimation et compensation du mouvement en fonction de différentes tailles de blocs sont présentées dans les sous figures (c), (d) et (e) de la figure 1.10. On constate que l'énergie résiduelle obtenue à l'aide de blocs de taille 8×8 est inférieure à celle obtenue avec des blocs de taille 16×16. Les blocs de taille 4×4 réduisent le plus l'énergie résiduelle. Ces exemples montrent que de petites tailles de blocs pour l'estimation et la compensation de mouvement minimisent l'énergie résiduelle. Cependant, des blocs de petite taille vont avoir tendance à augmenter la complexité calculatoire (plus de recherches sont nécessaires) et le nombre de vecteurs de mouvement à transmettre. Ceci implique un sur-coût en termes de bits et va diminuer le bénéfice de la réduction de l'énergie résiduelle.

La figure 1.11 représente deux versions de la même image de la séquence vidéo *Vectra*, pour laquelle on a réalisé deux estimations de mouvement, d'abord avec des blocs de taille 4×4 pixels et ensuite avec des blocs de 16×16 pixels. Les vecteurs de mouvement obtenus sont superposés sur l'image originale, pour l'image de droite de la figure 1.11, on observe seize fois plus de vecteurs de mouvement.

Ainsi il existe une relation entre l'efficacité de la compression et la complexité de la méthode d'estimation du mouvement, puisqu'une estimation plus précise (blocs de petites tailles) nécessite plus de bits pour coder le champ de vecteurs de mouvement et moins de bits pour coder le résidu, alors qu'inversement une estimation de mouvement moins précise (taille de blocs plus grande) requiert moins de bits pour encoder le champ de vecteurs de mouvement et plus de bits pour coder le résidu.

Un compromis effectif est d'adapter la taille des blocs aux caractéristiques de l'image, en choisissant par exemple des blocs de grande taille pour les régions homogènes de l'image et en choisissant des blocs de taille inférieure pour les zones complexes contenant beaucoup de détails. Le standard H.264 utilise une compensation de mouvement adaptative pour la taille des blocs [ISO03] (décrite dans le chapitre 4).

1.2.1.1.2 Estimation du mouvement sous-pixélique

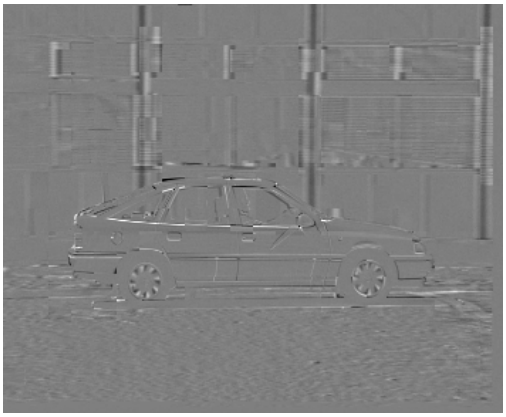
Un moyen permettant d'améliorer les performances de l'estimation de mouvement est de pouvoir prendre en compte des mouvements à composantes non entières quand cela est nécessaire, avec une certaine précision. Il est évident que cette précision est arbitraire car limitée par la structure de la grille pixélique résultante de l'échantillonnage spatial réalisé lors de l'acquisition de l'image par le capteur vidéo.

Dans certains cas, une meilleure prédiction du mouvement compensé peut être obtenue en réalisant celle-ci à partir de positions interpolées dans l'image de référence. L'estimation et la compensation du mouvement sous-pixéliques impliquent de réaliser une recherche pour les positions sous-pixéliques interpolées tout comme pour les positions au pixel près, et de choisir la position qui donne le meilleur résultat, c'est-à-dire qui minimise le critère utilisé (SAD, MSE...) et ensuite utiliser la valeur de cette position (au pixel près ou sous-pixélique) pour la prédiction du mouvement compensé.

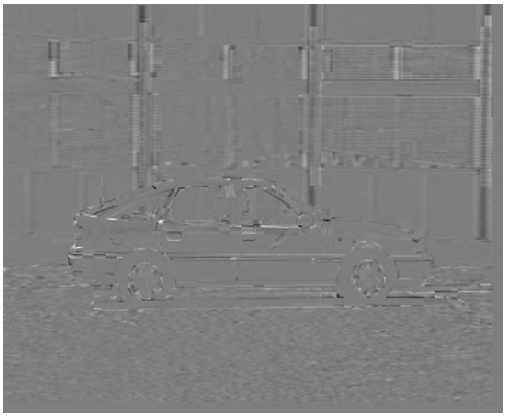
En général, une prédiction plus fine (par exemple au quart de pixel) permet d'obtenir de meilleures performances pour l'estimation/compensation de mouvement au détriment d'une augmentation de la complexité calculatoire. Les gains de performance ont tendance à se resserrer lorsque la partie fractionnaire du déplacement augmente en précision. Plus précisément, une interpolation au demi-pixel permet d'obtenir en moyenne un gain significatif par rapport à la compensation de mouvement au pixel près, une interpolation au

(a) Image 20 de la séquence *Vectra.*

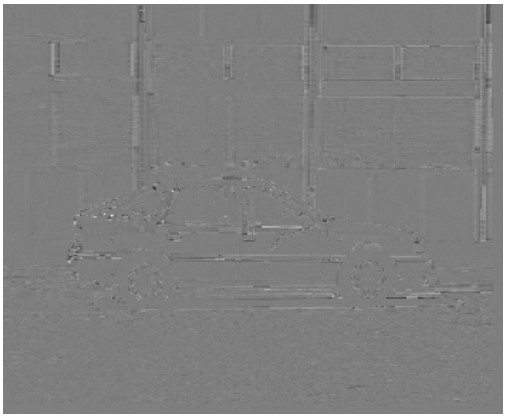
(b) image 21 de la séquence *Vectra.*



(c) énergie résiduelle avec des blocs 16×16.



(d) énergie résiduelle avec des blocs 8×8.



(e) énergie résiduelle avec des blocs 4×4.

FIG. 1.10 – Illustration de la réduction de l'énergie résiduelle après compensation du mouvement en fonction de la taille des blocs.

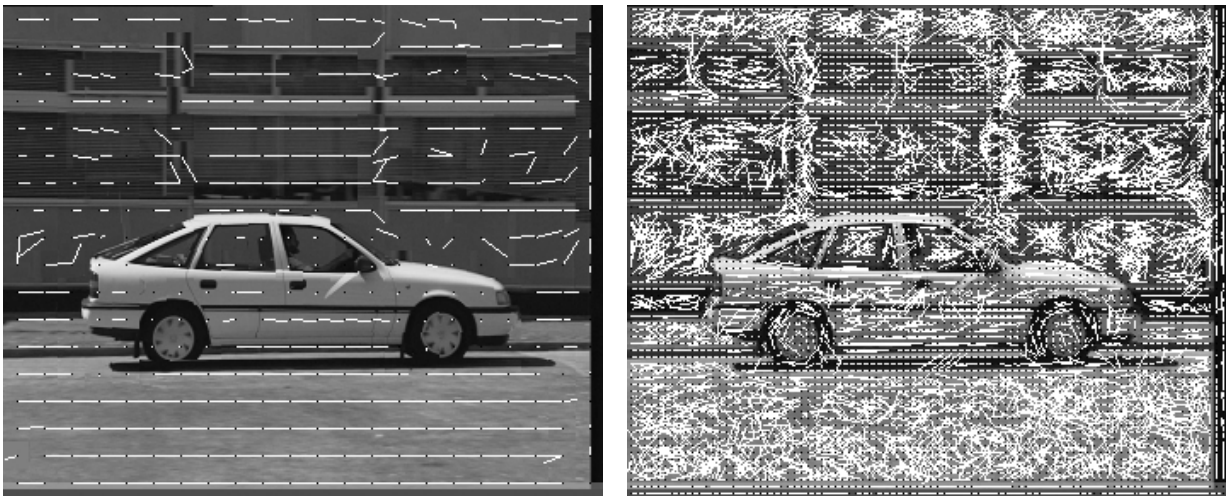


FIG. 1.11 – Vecteurs de mouvement obtenus pour une estimation de mouvement avec des blocs de 16×16 pixels (gauche) et des blocs de 4×4 pixels (droite).

quart de pixel améliore modérément encore les résultats. Les gains obtenus par une interpolation au huitième de pixel sont cas particuliers très réduits.

Il y a donc un compromis à trouver entre l'efficacité de la compression et la complexité des procédés d'estimation/compensation du mouvement, puisque si cette dernière est plus précise, elle exige plus de calculs et plus de bits pour coder le champ de vecteurs de mouvement, mais moins de bits pour coder les résidus. *A contrario*, une estimation/compensation de mouvement moins précise est réalisée plus rapidement et nécessite moins de bits pour coder le champ de vecteurs de mouvement, et plus pour le résidu.

1.2.1.1.3 Estimation du mouvement multi-références

Généralement, l'estimation de mouvement d'une image est réalisée en utilisant comme référence l'image codée/décodée immédiatement précédente qui est disponible au codeur et au décodeur. Les dépendances statistiques à long-terme entre les images de la vidéo, qui pourraient être utilisées pour améliorer l'efficacité de l'estimation/compensation de mouvement, ne sont pas exploitées par la plupart des standards de codage vidéo actuels.

Des améliorations pourraient être envisagées par l'utilisation d'une estimation du mouvement à moyen/long terme lorsque par exemple le contenu d'une séquence vidéo se répète sur plusieurs images. Des exemples de telles répétitions visuellement significatives sont variés, lorsque le contenu des images en mouvement provoque des répétitions dans l'orientation et la forme, des découvrements et recouvrements d'objets, des secousses en avant et en arrière de la caméra, etc. Les gains peuvent être obtenus si les blocs de taille 16×16 ou 8×8 pixels dans les images précédentes (même à long terme) coïncident parfaitement avec le bloc courant.

Une telle approche d'estimation de mouvement à long-terme pouvant utiliser jusqu'à 50 images références a été proposée [WZG99]. Les gains obtenus sont très significatifs dans les types de situations précédentes, particulièrement lorsque l'estimation de mouvement est contrainte au pixel près.

1.2.1.2 Méthodes prédictives

Les méthodes adaptatives décrites dans la section précédente permettent d'améliorer la prédiction des images *P*. Cependant, d'autres types d'information sont fortement corrélés au sein de régions d'images. Par

exemple, les valeurs moyennes (DC) des pixels de blocs voisins codés en intra, peuvent être très similaires ou les composantes horizontales et verticales de vecteurs de mouvement de blocs voisins peuvent être très proches. L'efficacité du codage peut alors être améliorée en prédisant des éléments du bloc courant à partir des données précédemment codées des blocs proches et en encodant la différence entre la prédiction et la valeur réelle.

1.2.1.2.1 Prédiction intra

On veut donc prédire le contenu d'une image en fonction des échantillons précédemment codés de cette même image. C'est la prédiction intra-image très généralisée en compression. Sur la figure 1.12, le pixel courant X doit être encodé. Dans un système classique, où l'encodage des pixels est effectué via un balayage *raster scan* (de la gauche vers la droite et de haut en bas), les pixels voisins codés/décodés A, B, C sont déjà disponibles pour l'encodeur et le décodeur.

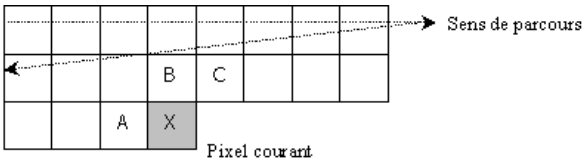


FIG. 1.12 – Prédiction spatiale ou intra.

L'encodeur génère alors une prédiction de X par une combinaison de A, B, C, soustrait cette prédiction à X et encode le résidu. Le décodeur pourra créer la même prédiction, et ajouter le résidu décodé pour reconstruire le pixel. L'efficacité de compression d'une telle méthode dépend de la précision des prédicteurs. Un prédicteur complexe, c'est-à-dire qui utilise plusieurs combinaisons (celles-ci étant appelées « modes » au sein de la norme H.264) des pixels voisins (A, B, $C\ldots$) permettra de réduire l'énergie résiduelle, mais ce gain en termes de compression est obtenu au détriment d'une combinatoire calculatoire plus importante. L'encodeur devra également transmettre le mode retenu ce qui engendrera un coût supplémentaire en termes de bits. Un compromis est alors nécessaire afin de sélectionner le meilleur mode, en utilisant par exemple une méthode des multiplicateurs de Lagrange (cf équation 1.13).

1.2.1.2.2 Prédiction du vecteur de mouvement

Le vecteur de mouvement d'un bloc cible indique le déplacement entre le bloc référence d'une image précédemment encodée vers ce bloc cible de l'image courante. Les vecteurs de mouvement des blocs voisins sont souvent corrélés si l'objet en mouvement recouvre une région importante (formée de blocs connexes) de l'image. Ceci est particulièrement vrai pour des blocs de petite taille et les objets de taille importante en mouvement. La compression du champ de vecteurs de mouvement peut être améliorée en prédisant chaque vecteur de mouvement à partir des vecteurs de mouvement précédemment encodés, en considérant un encodage *raster scan*. Une simple prédiction du vecteur de mouvement pour le bloc courant B_X est le bloc horizontalement adjacent B_A, comme illustré dans la figure 1.13. Alternativement, trois vecteurs de mouvement précédemment encodés ou plus, peuvent être utilisés pour prédire le vecteur de mouvement du bloc courant B_X (par exemple, les blocs B_A, B_B, et B_C de la figure 1.13). La différence entre le vecteur de mouvement prédit et le vecteur de mouvement réel (ou effectif) obtenu après l'estimation du mouvement est encodée et transmise.

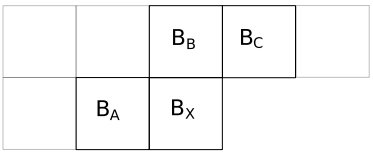


FIG. 1.13 – Candidats pour la prédiction du vecteur de mouvement.

1.2.1.2.3 Prédiction du pas de quantification

La taille du pas de quantification contrôle le compromis entre l'efficacité de compression et la qualité de l'image, ou autrement dit entre le débit et la distorsion. Pour un encodeur vidéo temps réel, il peut être nécessaire de moduler la quantification au sein même d'une image et d'utiliser différents pas de quantification pour coder séparément les résidus des macroblocs (par exemple, un pas de quantification par macrobloc). Il est généralement suffisant de maintenir, ou d'incrémenter ou de décrémenter le pas de quantification entre deux blocs codés successifs. La modification du pas de quantification doit être signalée au décodeur et au lieu de transmettre la nouvelle valeur du pas , il est préférable d'envoyer seulement la différence entre les deux valeurs.

1.2.1.3 Conclusion

Nous venons de voir les approches dites de « bas niveau » permettant d'améliorer la prédiction des images d'une séquence vidéo. Ces approches sont regroupées en deux catégories avec les méthodes adaptatives et celles prédictives. Les méthodes adaptatives visent à optimiser l'estimation du mouvement réalisée pour la prédiction de l'image courante par rapport à une image précédemment encodée/décodée. Pour cela, plusieurs techniques peuvent être utilisées, telles que réaliser l'estimation du mouvement avec des blocs de taille inférieure, également à partir de plusieurs images références (pas seulement l'image précédente), ou avec une précision plus fine (estimation du mouvement sous-pixélique). Ces méthodes essaient donc de simplement réduire l'énergie résiduelle (si critère énergétique) en réalisant une prédiction plus fine de l'image (mais ce qui a aussi pour conséquence d'augmenter la complexité calculatoire). La deuxième famille de méthodes décrite, propose d'étendre cette notion de prédiction aux autres éléments d'un codeur vidéo, à savoir le codage intra et les informations à transmettre nécessaires au décodage de la vidéo (vecteurs de mouvement et pas de quantification). Là aussi, l'objectif est seulement de réduire la quantité d'information à coder et à transmettre. Aucune considération perceptuelle n'est prise en compte dans la mise en œuvre de ces méthodes, seule la réduction de l'information en termes « d'énergie » du signal résiduel constitue la tâche qui peut donc être qualifiée à juste titre de « bas niveau ». Étant donné les performances de ces méthodes en termes d'optimisation débit-distorsion, elles sont pour la plupart maintenues (étendues) au sein de la norme actuelle H.264. Comme nous le verrons au chapitre 4, l'estimation du mouvement au sein du codeur est réalisée pour différentes tailles de bloc et à partir de plusieurs images références, et de nouvelles méthodes de prédiction ont été intégrées telles que la prédiction spatiale (pour les images I), la prédiction du vecteur de mouvement et du pas de quantification, ainsi qu'une transformation et un codeur entropique optimisés.

Cependant, aucune de ces optimisations ne considère les conséquences de ces décisions issues de ces analyses uniquement menées à court terme. En effet, les méthodes proposées risquent d'amplifier le phénomène des effets de papillotement dûs aux choix hétérogènes du codeur. Puisqu'afin de prédire le macrobloc courant, le codeur réalise une estimation/compensation de mouvement pour différentes tailles de blocs et à partir de plusieurs images références. Ainsi l'image de référence choisie et la taille de bloc sélectionnée

sont celles qui minimisent l'erreur de prédiction du macrobloc courant, mais peuvent être différentes de celles retenues pour la prédiction de ce macrobloc à la même position spatiale dans les images successives (précédentes et/ou futures). Cette incohérence temporelle de codage peut se répercuter au niveau de l'affichage et provoquer l'apparition d'effets de blocs (frontières des différentes tailles de bloc) et d'effets de papillotement.

1.2.2 Approches haut niveau

Même si le SVH est de fonctionnement très complexe, de nombreux travaux ont permis d'identifier très tôt certaines de ses propriétés (sensibilité à certains types d'information : luminance, fréquences spatiales, mouvement...). Celles-ci permettent d'isoler des paramètres importants à prendre en compte au sein d'un codeur vidéo. Il est alors envisageable de commencer à paramétrer notre schéma de codage vidéo basique en fonction de critères de plus haut niveau. Cette mise en œuvre des propriétés du SVH au sein du codeur vidéo basique se fait essentiellement pour les deux étapes : la quantification et la mesure de distorsion.

1.2.2.1 Quantification perceptuelle

Il existe une relation entre la qualité de l'image et le degré de quantification. Un pas de quantification important peut produire des dégradations visuelles importantes sur l'image. Malheureusement, réaliser une quantification plus fine conduit à diminuer le taux de compression. La question est de savoir comment quantifier efficacement les valeurs en entrée du quantificateur, à savoir, dans le cas de notre codeur basique, les coefficients de la TCD. Comme le SVH est moins sensible aux hautes fréquences (cf chapitre 2), ces fréquences jouent un rôle moins important que les basses fréquences. Donc les codeurs d'images fixes et de vidéos utilisent des pas de quantification plus importants pour coder les coefficients des hautes fréquences, produisant des dégradations seulement légèrement perceptibles.

1.2.2.1.1 Matrices de quantification

Le SVH est sensible aux faibles variations de l'intensité lumineuse sur une zone relativement importante à variation spatiale lente, mais distingue plus difficilement les variations exactes des intensités lumineuses de régions comportant des hautes fréquences. Ce fait permet de réduire grandement la quantité d'information pour représenter les coefficients des hautes fréquences issus de la TCD. La quantification a donc pour but d'annuler la plupart des coefficients hautes fréquences ou a des valeurs proches de zéro.

Une matrice de quantification est associée à la matrice des coefficients de la TCD, indiquant les pas à utiliser en fonction des domaines de fréquence. Les matrices de quantification sont généralement conçues pour conserver certaines fréquences afin d'éviter une perte de qualité. De nombreux codeurs vidéos, tels que H.264 [Ric03], permettent l'utilisation de matrices spécifiques.

Une matrice de quantification typique est donnée par :

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Les coefficients quantifiés de la TCD sont obtenus par :

$$Z_{i,j} = \left\lfloor \frac{F_{i,j}}{Q_{i,j}} \right\rfloor \text{ pour } i = 0,1,2,\cdots,N-1; j = 0,1,2,\cdots,N-1$$

où F représente le coefficient de la TCD non quantifié à la position (i, j) dans le bloc de taille $N \times N$, Q_{ij} le pas issu de la matrice de quantification, Z_{ij} le coefficient de la TCD quantifié et $\lfloor . \rfloor$ représente l'opérateur permettant d'obtenir la partie entière d'un nombre.

Si on rapproche cette matrice de quantification de la figure 1.8, on constate que les pas situés dans la partie supérieure gauche ont des valeurs inférieures par rapport aux autres pas. Ce qui implique que les coefficients de la matrice de la TCD correspondant aux basses fréquences seront quantifiés plus finement que ceux correspondant aux hautes fréquences, on exploite ainsi la propriété de la variation de la sensibilité aux fréquences spatiales du SVH.

1.2.2.1.2 Quantification pondérée en fonction de la fréquence spatiale

La variation de la sensibilité du SVH aux fréquences spatiales est également exploitée pour les codeurs qui n'utilisent pas une transformation de bloc de pixels. En effet, il a été démontré [ZDL00, ZDL02, DZLL00] qu'une pondération des fréquences avant quantification pouvait offrir une importante amélioration visuelle (à débit équivalent), particulièrement à une résolution supérieure à 200 points par pouce ($127\,\mu m/pixel$) pour les images fixes encodées à l'aide du standard JPEG2000.

Plus précisément, ce standard recommande l'usage de trois tables de « poids » pour respectivement trois distances de visualisation. Les poids des sous-bandes basses fréquences sont plus grands que ceux des sous-bandes hautes-fréquences, et il y a aussi une différence entre les poids de la composante de luminance et des deux composantes de chrominance Cb et Cr. En général, la pondération des fréquences serait plus efficace pour de grandes distances de visualisation ou de grandes résolutions d'images.

Au codeur, le pas de quantification Q_b des coefficients de la sous-bande b est ajusté pour être inversement proportionnel au poids P_b, comme indiqué dans l'équation suivante [TM01] :

$$Q_b = \alpha^{-1}.T_b.Q = \frac{Q}{\sqrt{P_b}} \tag{1.18}$$

où α est une constante positive, T_b est le seuil de détection moyen pour les signaux dans cette sous-bande.

1.2.2.2 Codage perceptuel

Les méthodes d'estimation du mouvement utilisent en général la somme des valeurs absolues des différences (SAD) ou l'erreur quadratique moyenne (MSE) pour mesurer l'erreur de mise en correspondance entre le bloc cible courant et le bloc de référence correspondant.

La SAD, la MSE ou le PSNR sont les métriques objectives les plus couramment utilisées en raison de leur faible complexité de mise en œuvre. Cependant, elles ont été largement critiquée pour leur faible corrélation avec le jugement humain [MS74]. Par exemple, dans certaines conditions, la qualité subjective d'une image peut être améliorée en ajoutant du bruit et donc en réduisant le PSNR. De plus, la visibilité des distorsions est intimement liée aux effets de masquage (cf chapitre 2), qui eux dépendent du contenu fréquentiel des images. Les métriques proposées ici étant basées pixel ne peuvent rendre compte de ces aspects. Les qualités visuelles de deux images ayant le même PSNR peuvent donc s'avérer être très différentes. Dans les dernières décennies, beaucoup d'efforts ont été fournis pour développer de nouvelles métriques d'évaluation de la qualité des images basées sur la théorie de la sensibilité aux erreurs du SVH. La plupart de ces modèles sont soit trop complexes pour être implantés au sein d'applications temps-réel, soit dédiés à des applications trop spécifiques pour être réutilisés [Wat93, Wat94, WJP93, Lam96, Win99, OLL+03].

Beaucoup de ces méthodes sont fondées sur la notion de détection de la distorsion. L'idée est que le SVH peut tolérer sans gêne visuelle une certaine quantité de bruit (dans un région donnée d'une image donnée), ceci en fonction de la sensibilité du SVH au signal source et du type de bruit. De nombreuses méthodes utilisant ce concept sont proposées pour l'allocation de bits et la quantification perceptuelle. La plupart de ces méthodes sont basées sur la réponse en fréquences spatio-temporelles du SVH [Wat92, CLCB93, OHB97, PM05]. Par exemple, Osberger et al. [OHB97, OMB98] proposent un codage MPEG qui intègre une quantification adaptative basée sur des propriétés du SVH. Partant du principe que les régions de textures et de contours ont des propriétés de masquage différentes, le quantificateur proposé les distingue et prend ainsi compte du masquage spatial. C'est un phénomène connu qui traduit la modification de la visibilité d'un signal par la présence d'un autre signal dit masquant (cette propriété du SVH sera explicitée au chapitre 2). L'image est d'abord découpée en blocs 8×8 et chaque bloc est classé comme étant soit uniforme, soit contenant un contour, soit texturé. L'activité act_b est ensuite mesurée pour chaque bloc en calculant sa variance. Cette valeur de l'activité est ensuite ajustée en fonction de la classe du bloc :

$$act\acute{}_b = \begin{cases} \min(act_b, act_{seuil}) & \text{si } b\text{est un bloc uniforme} \\ act_{seuil} \cdot \left(\frac{act_b}{act_{seuil}}\right)^{\varepsilon} & \text{si } b\text{est un bloc texturé ou contenant un contour} \end{cases} \tag{1.19}$$

où $act\acute{}_b$ est l'activité ajustée du bloc b, $act_{seuil} = 0.5$ est le seuil de visibilité de la variance, avec $\varepsilon = 0.7$ pour les zones de contours et $\varepsilon = 1$ pour les zones texturées. La valeur ajustée de l'activité est ensuite utilisée pour contrôler la quantification :

$$Nact_b = \frac{2 \cdot act\acute{}_b + \overline{act}}{act\acute{}_b + 2 \cdot \overline{act}} \tag{1.20}$$

où $Nact_b$ est l'activité normalisée pour le bloc b et $\overline{act}$ est la valeur moyenne de $act\acute{}_b$ pour l'image précédente. La valeur $Nact_b$ est ainsi comprise entre $[0.5, 2]$. Cette technique permet de minimiser les erreurs de quantification dans les régions uniformes et d'augmenter graduellement en fonction de l'activité la quantification le long de contours et encore plus pour les régions texturées.

1.2.2.3 Conclusion

Les méthodes qualifiées de « haut niveau » qui permettent d'optimiser en termes de débit-distorsion les codeurs vidéos et d'images fixes, se basent sur les propriétés du SVH. Cependant, ces propriétés sont encore très peu mises en œuvre au sein des codeurs. En effet, seule la propriété de la variation de la sensibilité aux fréquences spatiales est exploitée au sein des étapes de transformation et de quantification. En parallèle,

les recherches menées depuis quelques dizaines d'années pour mettre en évidence et modéliser certaines propriétés du SVH ont contribué à l'élaboration de métriques subjectives de qualité. Mais ces modèles de qualité restent pour la plupart trop complexes pour être intégrés au sein des schémas de codage. Quelques travaux se basant sur la réponse en fréquence spatio-temporelle du SVH [Wat92, CLCB93, OHB97, PM05] ont été proposés pour l'allocation de bits et la quantification perceptuelle, confirmant ainsi l'idée que les propriétés du SVH sont les paramètres importants à prendre en considération directement au sein d'une chaîne de codage et non seulement à la fin de celle-ci pour l'évaluation de la qualité perçue.

Conclusion

Nous avons décrit la forme d'un codeur vidéo basique afin de présenter les étapes de la chaîne de codage et l'objectif est de cerner à quel niveau des optimisations ont été possibles. La première partie de ce chapitre nous a permis d'aborder les différentes étapes du codeur vidéo basique : la décorrélation temporelle, la décorrélation spatiale, la quantification et le codeur entropique. Cette étude des différents blocs de traitement, nous a conduit à mettre en évidence une lacune importante dans cette chaîne de codage : dans son principe même de fonctionnement, ce codeur vidéo ne dispose d'aucune information quant à l'évolution temporelle de la séquence (excepté pour le codage des images B). Celui-ci est donc dans l'incapacité de prendre des décisions à moyen/long terme afin d'assurer une certaine cohérence de codage. Par exemple, l'hétérogénéité temporelle des décisions prises pour coder un macrobloc à la même position spatiale entre les images successives peut occasionner l'apparition de dégradations visuellement perceptibles. C'est l'effet de papillotement (*flickering effect* en anglais). Pour ce codeur vidéo classique, notre solution s'oriente vers le positionnement en amont d'une méthode de pré-analyse de la vidéo. Cette étape de pré-analyse de la vidéo devra par exemple extraire des informations sur le contenu et l'évolution de la séquence vidéo, afin de les transmettre au codeur et de le guider dans ses décisions pour garantir la cohérence du codage et notamment améliorer la qualité perceptuelle de la vidéo décodée.

Dans la deuxième partie du chapitre, nous avons étudié des approches présentes dans la littérature permettant d'optimiser les différentes étapes du codeur vidéo classique, afin d'améliorer la qualité perçue pour un débit donné (ou inversement). La première catégorie de méthodes a présenté les approches basiques qui permettent de réduire les informations à transmettre au codeur en réalisant soit un codage adaptatif ou un codage prédictif. Ces méthodes restent limitées, puisqu'elles ne traitent l'information à réduire que d'un point de vue signal et sont donc « bas niveau ». Dans ce cas, aucun *a priori* sur le contenu des images successives n'est pris en considération lors du codage. Cependant, comme nous le verrons lors de la description de la norme H.264 au chapitre 4, ces méthodes du fait de leur grande efficacité ont été adoptées. Au contraire, les méthodes présentées dans la dernière partie du chapitre exploitent certaines propriétés du SVH afin d'optimiser le codage au sens débit-distorsion, elles sont qualifiées de « haut niveau ». Des codeurs exploitent simplement la sensibilité variante du SVH face aux fréquences spatiales, en adaptant conjointement les étapes de transformation et de quantification. Le SVH étant plus sensible aux basses fréquences qu'aux hautes fréquences spatiales, l'utilisation de matrices de quantification permet de coder plus finement les coefficients des basses fréquences par rapport à ceux des hautes fréquences. Prévenant ainsi l'apparition d'artéfacts au sein des basses fréquences. D'autres approches réalisent un codage perceptuel en fonction du contenu spatio-temporel des macroblocs.

De nombreux travaux intégrant des propriétés du SVH sont proposés pour l'évaluation de la qualité, ou améliorer les techniques de codage. Cependant, ces méthodes sont peu retenues au niveau des standards de

compression vidéo. Les efforts de recherche réalisés se portent davantage sur l'enrichissement des nouvelles normes de codage. Par exemple, nous verrons au chapitre 4 que le standard H.264 rassemble de nombreuses techniques de prédiction ainsi qu'une optimisation du codage entropique, lui permettant d'obtenir des performances supérieures en termes de compression par rapport aux standards existants. Ces performances sont obtenues par une minimisation conjointe du débit et de la distorsion à partir de critères « bas niveau ». Cependant, les méthodes « haut niveau » présentées dans ce chapitre confirment leur intérêt pour le codage. Les standards sont compétitifs et visent à offrir de nombreuses décisions qui pourraient aussi être prises à partir d'informations « haut niveau ». Notre méthode de pré-analyse de la vidéo devra donc prendre en considération les différentes propriétés du SVH afin de transmettre au codeur des informations « haut niveau » pour le piloter et prendre rapidement les bonnes décisions (parmi toutes celles possibles). Dans la suite du document nous étudierons les propriétés et les modélisations du SVH liées au codage.

Chapitre 2

Caractéristiques et modèles du système visuel humain et de l'attention visuelle

Sommaire

Introduction **33**

2.1 Propriétés et modélisation du système visuel humain **34**

2.1.1 Introduction 34

2.1.2 La perception de la luminance 34

2.1.3 La perception des couleurs 36

2.1.4 La sensibilité aux contrastes 36

2.1.5 L'organisation multi-canal 39

2.1.6 Les effets de masquage 40

2.1.7 Conclusion 41

2.2 Mouvements oculaires et attention visuelle **42**

2.2.1 Mouvements oculaires 42

2.2.2 Attention visuelle sélective 43

2.2.3 Conclusion 46

2.3 Modélisation de l'attention visuelle pré-attentive **47**

2.3.1 Introduction 47

2.3.2 Modèles empiriques et statistiques 47

2.3.3 Modèles psycho-visuels 48

2.3.4 La dimension temporelle dans la modélisation 52

2.3.5 Conclusion 53

2.4 Applications possibles d'un modèle de l'attention visuelle au sein d'un codeur vidéo . **55**

Conclusion **56**

Introduction

Les efforts réalisés ces dernières années afin d'optimiser les codeurs vidéo en termes de débit-distorsion, ont abouti au développement de nouvelles méthodes de prédiction. Mais comme nous l'avons évoqué précédemment, celles-ci souffrent de carences. En effet, les codeurs vidéo actuels ne réalisent pas une analyse à

moyen/long terme de la séquence vidéo et prennent donc des décisions à court terme pouvant provoquer des incohérences de codage et l'apparition de défauts visuellement perceptibles. Souvent aussi, ces décisions ne sont prises qu'à partir d'informations dites de bas niveau. Cependant, quelques méthodes haut niveau ont été proposées et implantés au sein des codeurs vidéo démontrant l'intérêt d'exploiter les propriétés du SVH . Les nouvelles normes de codage vidéo sont constituées d'un grand ensemble d'étapes auxquelles sont associés des outils qu'il devient possible de paramétrer. Nos travaux de thèse visent à montrer l'intérêt de réaliser une méthode de pré-analyse de la vidéo en tenant compte d'informations dites de haut niveau, afin de mieux paramétrer le codeur pour assurer une certaine cohérence du codage.

Il est donc nécessaire de posséder les connaissances sur les propriétés et les modélisations du SVH, ainsi que les étapes biologiques du traitement de l'information visuelle, avant de penser à la mise en œuvre d'une telle étape de pré-analyse. Il existe d'ailleurs une littérature abondante et très ardue sur le sujet, mais étant donné que notre contexte porte sur l'exploitation des modèles, nous nous concentrerons dans ce chapitre sur la modélisation du SVH et sur celui de l'attention visuelle.

D'abord, nous nous intéresserons à la modélisation des propriétés dites de bas niveau qui peuvent également être appelées passives. Les propriétés seront sélectionnées pour leur intérêt dans la conception d'une étape de pré-analyse de la vidéo. Ensuite, nous nous intéresserons à la description de l'attention visuelle qui est primordiale pour comprendre et appréhender notre environnement. Dans la troisième partie de ce chapitre, des modèles mathématiques de l'attention visuelle pré-attentive seront détaillés. L'objectif est de prédire, à partir d'attributs de bas niveau, les positions des zones visuellement importantes d'une image ou d'une séquence vidéo. Les modèles présentés sont regroupés en deux catégories. La première est constituée d'une part de modèles dit empiriques (c'est-à-dire élaborés à partir de connaissances du système visuel et validés via des expériences) d'autre part aussi de modèles statistiques, qui sont relativement éloignés du SVH. La seconde catégorie, quant à elle, regroupe les modèles basés sur une architecture biologiquement plausible. Cette architecture proposée par C. Koch et S. Ullman [KU85] est d'abord décrite, introduisant la notion de carte de saillance. Les grandes caractéristiques des modèles s'inspirant de cette architecture et leurs caractères innovants sont ensuite examinés. Enfin, la dernière partie présente les mises en œuvre possibles au sein d'un codeur vidéo, permettant d'exploiter les cartes de saillance issues des modélisations de l'attention visuelle pré-attentive.

2.1 Propriétés et modélisation du système visuel humain

2.1.1 Introduction

Les recherches réalisées sur la vision durant les quarante dernières années ont permis un rapide progrès dans la connaissance du fonctionnement du SVH. Cependant, ce n'est que récemment que ces connaissances ont été mises en pratique au sein d'algorithmes de traitement d'images. Cette section présente les propriétés du SVH qui nous semblent importantes pour élaborer ensuite un modèle de l'attention visuelle. Nous verrons qu'outre la couleur et le mouvement, le contraste est la notion importante à prendre en compte pour la conception d'un modèle du SVH.

2.1.2 La perception de la luminance

Trois types de sensations sont engendrés par la perception d'une zone d'une image. La perception de la chromacité de la zone observée provoque les sensations de teinte et de saturation, alors que la luminance

perçue produira la sensation de luminosité.

Le SVH, qui reçoit des variations importantes de l'intensité lumineuse, met en place des mécanismes d'adaptation lui permettant de conserver sa sensibilité pour différentes conditions d'illumination, quelles soient importantes (conditions photopiques) ou faibles, (conditions stocopiques).

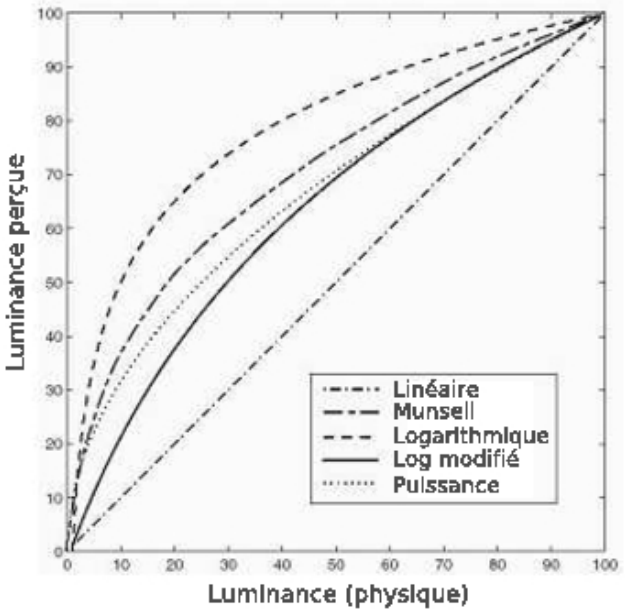


FIG. 2.1 – Relations liant la luminance perçue et la luminance réelle.

En plus de la capacité d'adaptation, la relation entre la luminance perçue (brillance) et la luminance réelle (luminance physique) n'est pas linéaire. Afin de déterminer cette relation, des expérimentations ont été réalisées, où des observateurs doivent classer des nuances de gris. Par ce biais, Munsell [NNJ43] a pu déterminer la relation entre la luminance originale et la luminance perçue. Dans la figure 2.1, la courbe de Munsell ainsi que celles issues de différentes modélisations mathématiques sont présentées. On observe que la fonction logarithmique surestime la luminance perçue, alors qu'au contraire, la fonction linéaire la sous-estime. La fonction logarithme modifiée a tendance à sous-estimer la luminance perçue pour les basses valeurs de luminance.

Finalement, la fonction puissance, plus connue sous le nom de loi Gamma, semble être la relation la plus adaptée pour modéliser la non linéarité de la perception de la luminance. La dynamique des valeurs de faible niveau de gris est augmentée, alors que les valeurs importantes de niveau de gris sont atténuées. La luminance perçue, L_p, est obtenue à partir de la luminance originale, L_o, par la relation suivante :

$$L_p = a \times L_o^e - L_0 \tag{2.1}$$

Les expérimentations réalisées par Bodman et al. [BHM80] ont permis de déterminer la valeur de l'exposant e, celui-ci étant égal à $0,31 \pm 0,03$. Les deux autres paramètres, a et L_0, permettent de s'adapter à différentes échelles . Classiquement $a = 1$ et $L_0 = 0$.

En règle générale et dans un contexte d'affichage d'images sur un moniteur à tubes cathodiques (CRT), la fonction Gamma d'exposant $e \times n$, avec $e = 0,31$ et $n = 2,3$ est utilisée pour modéliser la transduction photoélectrique et pour compenser le comportement non linéaire des écrans.

2.1.3 La perception des couleurs

Les études sur les cônes, qui sont des cellules photo-réceptrices de la rétine, ont permis de dégager trois grandes populations qui se distinguent par leur sensibilité spectrale :

- les cônes S (*Small*) ont une sensibilité maximale autour de $420\,nm$ (bleu) ;
- les cônes M (*Medium*) ont une sensibilité maximale autour de $531\,nm$ (vert-jaune) ;
- enfin, les cônes L (*Large*) ont leur maximum de sensibilité autour de $558\,nm$ (jaune-rouge).

De nombreux modèles, basés sur la théorie des signaux antagonistes ont été établis à partir des travaux réalisés sur la perception des couleurs. Selon ces modèles, les signaux issus des trois types de cônes L, M, S sont combinés à partir d'une transformation linéaire permettant de définir trois composantes perceptives : une composante achromatique, A, et deux composantes chromatiques, $Cr1$ et $Cr2$:

$$\begin{pmatrix} A \\ Cr1 \\ Cr2 \end{pmatrix} = [T] \times \begin{pmatrix} L \\ M \\ S \end{pmatrix} \tag{2.2}$$

Des expériences physiologiques ont conduit à proposer des modèles de construction des trois composantes [DVDV92, Fau76]. De Valois se base sur les signaux incidents sur les zones excitatrices et inhibitrices des champs récepteurs :

$$[T]_{DeValois} = \begin{pmatrix} 0,375 & 0,6875 & 0,00625 \\ 0,5625 & -0,7187 & 0,1562 \\ -0,8125 & 0,5938 & 0,2187 \end{pmatrix} \tag{2.3}$$

Le modèle définit par Faugeras utilise les différentes courbes d'absorption des différents types de cônes :

$$[T]_{Faugeras} = \begin{pmatrix} 13,63 & 8,33 & 0,42 \\ 64 & -64 & 0 \\ -5 & -5 & -10 \end{pmatrix} \tag{2.4}$$

D'autres modèles sont entièrement déduits d'expériences psychophysiques [WDVS90, FCF90, KW82]. Ces modèles étant très proches, nous ne donnons que la matrice de transformation proposée par Krauskopf :

$$[T]_{Krauskopf} = \begin{pmatrix} 1 & 1 & 0 \\ 164 & -1 & 0 \\ -0,5 & -0,5 & 1 \end{pmatrix} \tag{2.5}$$

2.1.4 La sensibilité aux contrastes

Une des propriétés importantes de notre système visuel est sa sensibilité différenciée aux contrastes : certaines cellules du cortex visuel répondent aux contrastes forts, d'autres aux contraste faibles. Cette sensibilité différenciée est basée sur notre aptitude à détecter les contrastes de luminance, de couleur, de mouvement... Cette propriété nous intéresse particulièrement car elle sera prise en considération dans notre contribution pour la conception des cartes de saillance. Des expériences de psychophysiques réalisées dans un contexte strict permettent de déterminer la sensibilité du SVH pour la luminance. Généralement les modèles proposés utilisent la définition du contraste de Michelson :

$$C = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} \tag{2.6}$$

où L_{max} et L_{min} représentent respectivement la luminance maximale et minimale du stimulus utilisé.

Pour un contexte particulier (fréquence spatiale et distance de visualisation données), le contraste minimal a la valeur minimale pour laquelle le stimulus devient juste détectable par l'observateur. Le seuil de sensibilité au contraste ou seuil de visibilité est obtenu en calculant l'inverse de la valeur du contraste minimal défini ci-dessus. Cette sensibilité sera donc d'autant plus élevée que le contraste seuil sera faible et inversement. Le seuil de sensibilité est également influencé par d'autres facteurs tels que, la fréquence spatiale, l'orientation, la fréquence temporelle, la couleur... Ces dépendances sont ensuite modélisées par une fonction de sensibilité au contraste (FSC, ou CSF[1] en anglais). Des CSF modélisant la sensibilité du SVH aux contrastes spatiaux et spatio-temporels sont présentés ci-après.

2.1.4.1 Sensibilité aux fréquences spatiales

Dans la littérature, un ensemble assez complet de courbes de CSF pour des signaux achromatiques et pour différentes configurations de stimuli est fourni [PAYG93]. P. Barten [Bar04] a proposé plus récemment une formulation plus complète des CSF. Un exemple de CSF classique isotrope proposé par J. Mannos et D. Sakrison [MS74] est illustré à la figure 2.2. Son équation est la suivante :

$$CSF(f) = 2,6 \times (0,0192 + 0,114f)e^{(-(0,114f)^{1,1})} \tag{2.7}$$

où f est exprimé en cycle par degré (cpd).

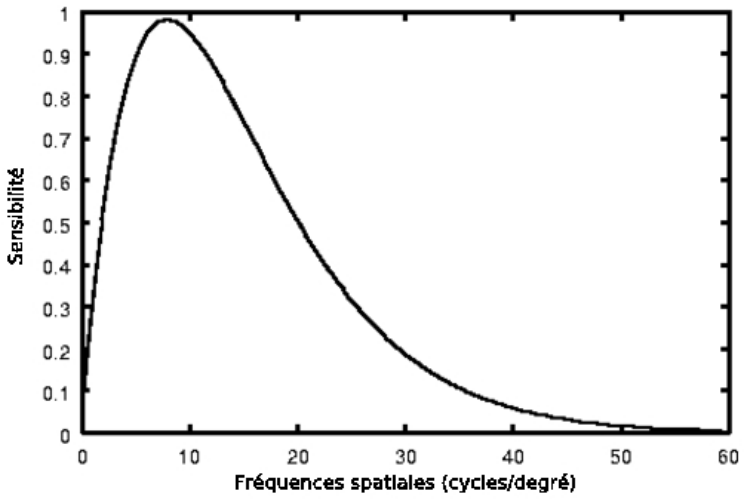


FIG. 2.2 – Fonction normalisée de sensibilité au contraste proposée par J. Mannos et D. Sakrison [MS74].

Cette courbe de sensibilité montre que nous sommes plus sensibles aux fréquences spatiales intermédiaires (entre 4 et 13 cpd) qu'aux basses et hautes fréquences spatiales. Au delà de 50 cpd, l'œil ne détecte plus rien. L'œil a donc un comportement presque passe-bande vis à vis des fréquences spatiales de la composante luminance. En d'autres termes, pour une zone de l'image où les fréquences spatiales significatives sont basses, l'apparition d'un signal (une dégradation ou artéfact de codage vidéo par exemple) même à faible contraste risque d'être gênante pour l'observateur. A contrario, pour une zone de l'image où les fréquences spatiales sont élevées, l'apparition d'un signal ayant un contraste moyen engendre une perception

[1] *Contrast Sensitivity Function*

plus faible. Comme nous l'avons évoqué au chapitre précédent, cette sensibilité variable du SVH vis à vis des fréquences spatiales est exploitée simplement au sein du codeur vidéo classique. Après l'étape de transformation, l'usage de matrices de quantification exploite la sensibilité aux fréquences spatiales du SVH : les coefficients hautes fréquences sont quantifiés plus grossièrement que les coefficients basses et moyennes fréquences.

2.1.4.2 Sensibilité aux fréquences temporelles

Par analogie au domaine spatial, il est possible de définir des CSF temporelles. Autrement dit, si on considère une zone d'une image dont le contraste varie temporellement de façon sinusoïdale et avec une amplitude maximale constante, pour certaines fréquences temporelles les variations seront visibles alors qu'à d'autres les variations ne seront pas perçues. Les premières expérimentations, menées par H. De Lange [DL58] montrent que la sensibilité est maximale (200) à environ $8\,Hz$. Au dessus, la sensibilité décroît très rapidement et atteint une valeur de 1 pour une fréquence comprise entre $50\,Hz$ et $70\,Hz$. Cette fréquence est appelée CFF[2], et représente la transition entre un scintillement de lumière et une lumière continue. Cette sensibilité décroît également pour les basses fréquences temporelles, mais de façon plus modérée.

Il a été montré que la CFF varie en fonction de l'intensité, plus précisément elle est proportionnelle au logarithme de la luminance du stimulus clignotant [KFN02] :

$$CFF = a \cdot log(L) + b \tag{2.8}$$

Au fur et à mesure que l'intensité du stimulus augmente, nous sommes plus sensibles au clignotement. Ainsi pour des lumières intenses, la CFF peut atteindre $100\,Hz$. Les expérimentations menées par l'Organisation Internationale de Normalisation (ISO) [fS91], montrent que pour une luminance de $100\,cd/m^2$ (candela par mètre carré), un écran à tube cathodique doit avoir une fréquence de $70\,Hz$ pour que le clignotement ne soit pas visible par l'observateur moyen.

2.1.4.3 Interactions spatio-temporelles

Bien qu'il soit plus simple de considérer la CSF temporelle indépendamment des autres dimensions, il apparaît cependant que la sensibilité temporelle est intimement liée à la fréquence spatiale du stimulus visuel. La séparabilité spatiale et temporelle serait obtenue si la sensibilité était obtenue via le produit des fonctions $S_s(f_s, \theta)$ et $S_T(f_t)$ représentant respectivement la sensibilité aux contrastes spatiaux et celle aux contrastes temporels :

$$S(f_s, f_t) = S_s(f_s, \theta) \times S_T(f_t) \tag{2.9}$$

où, f_s, f_t et θ représentent respectivement la fréquence spatiale, la fréquence temporelle et l'orientation.

Pour la vision humaine, la sensibilité spatio-temporelle n'est pas une fonction séparable. De nombreuses études ont montré qu'il y avait une forte interaction entre la perception spatiale et celle temporelle. Les facteurs influençant la sensibilité temporelle sont notamment :

– la taille de la cible : une cible de taille importante tend à réduire la sensibilité temporelle à basse fréquence temporelle. À haute fréquence, la sensibilité est quasiment inchangée ;

[2] *Critical Flicker Frequency*

– les contours : une cible présentant des contours contrastés augmente la sensibilité en basses fréquences temporelles. Il n'y a pas d'effet en hautes fréquences.

En résumé, pour les fortes fréquences spatiales et temporelles, les deux fonctions de sensibilité au contraste peuvent être considérées comme indépendantes. La formule 2.9 est donc utilisable. Par contre, pour les faibles et moyennes fréquences spatiales et/ou temporelles, la CSF n'est pas séparable.

Cependant, le SVH ne doit pas être considéré comme un système mono-canal dont les caractéristiques seraient uniquement données par une CSF. En effet, le SVH est un système multi-canal avec des canaux non indépendants. Il faut considérer cette structure multirésolution du SVH afin d'étudier convenablement les interactions à l'intérieur d'un canal et entre les différents canaux, interactions de type effets de masquage.

2.1.5 L'organisation multi-canal

La sensibilité des cellules du SVH à certains types d'informations telles que la couleur, l'orientation ou la fréquence, suggère qu'il existe des regroupements de l'information préalablement à son traitement. Les résultats de plusieurs expérimentations psychophysiques confortent cette idée et présentent le SVH comme un système multi-canal [BCA78].

2.1.5.1 Décomposition spatiale de l'information

La décomposition spatiale de l'information du SVH en différents canaux s'effectue selon une sélectivité radiale (de 1 à 2 octaves) et une sélectivité angulaire (de 20° à 60°). Dans la littérature, on trouve plusieurs décompositions. Nous pouvons citer la transformée Cortex de Watson [Wat87] qui décompose le signal en cinq couronnes de fréquences radiales ayant chacune une largeur de bande d'une octave et présentant une sélectivité angulaire constante de 45° (sauf pour la couronne des plus basses fréquences spatiales). Cette décomposition est réalisée au moyen de filtres dit Cortex dont les paramètres ont été affinés à partir de nombreuses expérimentations psychophysiques [Sén96, Bed98, LC01] à la fois pour la luminance et la couleur. On parle alors de décomposition en canaux perceptuels. Nous pouvons également citer les filtres de Gabor qui reposent sur la ressemblance entre la forme des champs récepteurs corticaux et la transformation bidimensionnelle de Gabor. Cependant, dans la pratique il faudrait considérer un grand nombre de filtres pour couvrir tout le pavage fréquentiel du SVH. Nous pouvons aussi citer des approches multi-résolutions, moins fidèles au SVH mais plus directes à mettre en œuvre, comme par exemple les transformations pyramidales [BA83] ou celles en ondelettes classiques. L'avantage de ces transformations est la bonne localisation spatiale, les inconvénients sont les sélectivités fréquentielles et angulaires approximatives. Dans le chapitre 3, nous utiliserons une approche multi-résolution, proche des transformations pyramidales proposées par Burt et Adelson [BA83], afin de réduire les temps de calculs de notre méthode d'estimation du mouvement.

2.1.5.2 Décomposition temporelle de l'information

Tout comme la dimension spatiale se caractérise par un ensemble de sous bandes (ou canaux) sélectives en orientation et en fréquence spatiale, des études ont montré qu'il existait des cellules du SVH sélectives en fréquences temporelles [DVCM+00]. Cependant, les avis diffèrent pour la forme de la décomposition temporelle de l'information. Le modèle le plus courant est une décomposition à partir de deux canaux temporels, l'un passe-bas et l'autre passe-bande [FH98]. Il est aussi fait mention des canaux *sustained* et *transient* dont les réponses fréquentielles sont illustrées figure 2.3. D. Parkhurst [Par02] détermine une

carte de saillance, représentant les régions d'intérêt du champ visuel, qu'à partir du canal *transient* afin de modéliser le traitement de l'information temporelle par le SVH.

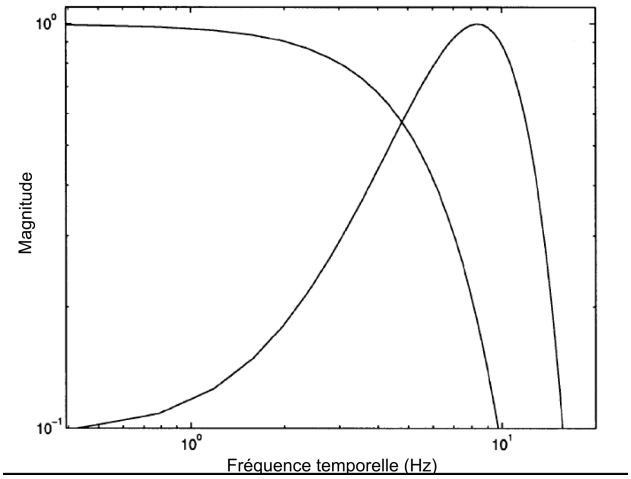


FIG. 2.3 – Réponse fréquentielle des canaux visuels *sustained* (passe-bas) et *transient* (passe-bande) de la décomposition temporelle de l'information [Lam96].

2.1.6 Les effets de masquage

2.1.6.1 Le masquage spatial

Précédemment, nous avons vu le concept de fonction de sensibilité au contraste (CSF). Ces fonctions modélisent la stimulation des cellules visuelles face à un stimulus présentant un contraste supérieur à une valeur seuil appelé seuil différentiel de visibilité. Cependant, cette modélisation mono-canal du système visuel n'est cohérente que pour des signaux simples. En effet, le seuil de visibilité ne dépend pas que de la valeur du contraste du stimulus, mais également des stimuli voisins. Cet effet de modulation entre signaux voisins est communément appelé effet de masquage visuel. Il exprime soit la réduction (*masking effect*) de la visibilité d'un signal (appelé signal masqué) par un autre signal (appelé signal masquant) soit l'augmentation de la visibilité ou facilitation (*pedestal effect*). Des études ont montré qu'il existait trois types d'effet de masquage [Hee93, Dal93, FB94, LC01] :

- le plus important est le masquage intra-canal se traduisant par une interaction entre stimuli de mêmes caractéristiques (fréquence, orientation, composante) ;
- le masquage entre stimuli de caractéristiques différentes c'est à dire n'appartenant pas au même canal. Cet effet est appelé masquage inter-canal ;
- le masquage entre différentes composantes (c'est-à-dire entre la composante de luminance et les deux composantes de chrominances) qui est appelé masquage inter-composante.

Cette brève énumération montre une nouvelle fois l'intérêt de procéder à une décomposition en canaux visuels. L'effet de masquage est souvent représenté par une courbe caractéristique comme celle présentée à la figure 2.4. L'axe horizontal représentant le contraste du signal masquant C_M, et l'axe vertical représentant le contraste du stimulus au seuil de visibilité Δ, appelé aussi contraste seuil. Le seuil Δ en l'absence de signal masquant est noté Δ_0, dans ce cas, cela signifie que le signal masquant correspond à un signal constant (signal uniforme). Pour des valeurs de contraste du signal masquant supérieures à C_{M_0}, le contraste seuil augment en même temps que le contraste du signal masquant. La courbe de masquage illustrée à la figure 2.4 (b), laisse aussi apparaître un effet de facilitation limitée à une zone où $C_M < C_{M_0}$.

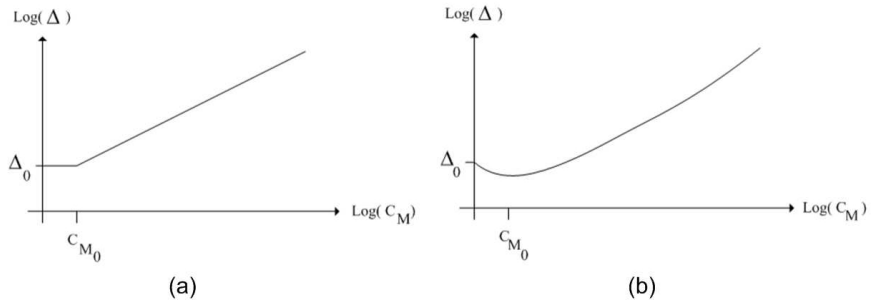


FIG. 2.4 – Caractéristique du masquage d'un stimulus par un autre : (a) sans facilitation, (b) avec facilitation.

Dans le contexte de la compression, les effets de masquage sont particulièrement intéressants car ils peuvent permettre, entre autre, d'exploiter le contenu d'une image pour masquer le bruit de quantification.

2.1.6.2 Le masquage temporel

Comparativement au masquage spatial, les effets de masquage temporel sont moins connus et la littérature les concernant est bien moins abondante. Le masquage temporel module le seuil de visibilité d'un signal en fonction d'évènements temporels. Comme pour le masquage spatial, cette modulation peut faciliter ou atténuer la visibilité d'un signal. On distingue dans la littérature deux types de masquage temporel. Le premier type de masquage qui est le plus intuitif, est le masquage avant (*forward* en anglais) apparaissant après une discontinuité temporelle (typiquement un changement de plan au sein d'une séquence vidéo) [SB59, SB65]. La perception des images situées après le changement de plan se trouve alors réduite pendant une durée d'environ 100 *ms* [Tam95]. Le deuxième type de masquage est le masquage arrière (*backward* en anglais) correspondant à une réduction de la visibilité d'images avant le changement de plan. Ce type de masquage est plus éphémère que le précédent puisque sa durée est de l'ordre de 10 *ms*. Une façon de l'expliquer est liée à la variation de latence des signaux neuronaux en fonction de leur intensité [AAN93].

Les articles de Girod [Gir89] et Watson [Wat98] présentent des exemples de mise en œuvre de masquage temporel. Takeuchi et DeValois [TDV05] exploitent une forme de masquage temporel en insérant de manière alternée, des images filtrées parmi des images non filtrées, celles-ci réduisant la perception de l'absence de détail dans les images filtrées. On parle alors de *quasi-motion sharpening* qui est une forme de *motion sharpening*. Le phénomène de *motion sharpening* est une caractéristique temporelle du SVH mettant en évidence une illusion d'optique dans laquelle un objet en mouvement dans une séquence vidéo paraît plus net, bien que chaque image prise séparément soit d'un niveau de netteté perçue moindre que celui de la séquence animée.

2.1.7 Conclusion

Dans cette section nous avons présenté les modélisations de certaines propriétés du SVH. Les propriétés qui nous intéressent sont celles présentant un intérêt pour la conception d'un modèle de l'attention visuelle : la perception de la luminance, la perception des couleurs, la sensibilité au contraste, l'organisation multi-canal et les effets de masquage. De plus, le SVH est davantage sensible aux contrastes qu'aux luminances et cette sensibilité varie avec la nature des signaux. Une modélisation du système visuel doit donc considérer ces différentes propriétés. Nos travaux portant sur la pré-analyse de séquences vidéo, la dimension temporelle est un paramètre important à prendre en compte dans notre étude. D'ailleurs, les expériences de recherches visuelles réalisées par J. Wolfe [WCF89, WH04] ont permis de mettre en évidence le phéno-

mène d'attraction visuelle des objets en mouvement ou plus particulièrement en contraste de mouvement par rapport aux autres objets. Le contraste, qu'il soit de luminance, de couleur ou de mouvement, est donc l'information pour laquelle notre SVH est le plus sensible, c'est l'information qui conduira notre pré-analyse avant codage de la vidéo.

2.2 Mouvements oculaires et attention visuelle

2.2.1 Mouvements oculaires

Notre système visuel étant de capacité de traitement malgré tout limité, celui-ci utilise des mouvements oculaires afin d'optimiser l'usage de ses ressources de traitement de l'information visuelle [Car98]. Ces mouvements oculaires sont de différents types. Ils se décomposent en mouvements de poursuite, de convergence, de saccades ou encore de fixations. Les deux mouvements oculaires principaux, associés à la focalisation dite *overt*, rendant l'effet attentionel observable, sont les saccades et les fixations. Ces deux types de mouvements sont décrits maintenant.

2.2.1.1 Les saccades

Les saccades sont des mouvements oculaires très rapides dont la vitesse angulaire est comprise entre 100 et 700 degrés par seconde [Sal99]. Ce type de mouvement permet de déplacer l'attention visuelle d'un endroit à un autre afin de les inspecter par la zone la plus performante (en termes de résolution spatiale) de la rétine : la fovéa. Les saccades sont souvent considérées comme un mécanisme favorisant la sélection des informations visuelles pertinentes de notre champ visuel. L'observation de notre environnement visuel se fait donc par une série de sauts permettant le déplacement rapide de nos ressources sensorielles d'un point à un autre. Le passage d'un point à un autre ne se réalise par forcément par le plus court chemin, c'est-à-dire la ligne droite. En effet, la trajectoire peut être incurvée. De plus, la précision d'une saccade peut ne pas être satisfaisante, dans ce cas, une seconde saccade ajuste le déplacement pour atteindre la cible désirée. Durant ces déplacements, le pouvoir d'analyse du système visuel est très faible et donc peu d'informations visuelles ne sont traitées. Les saccades sont généralement séparées par des phases de fixations.

2.2.1.2 Les fixations

Une phase de fixation se produit lorsque l'œil de l'observateur fixe un objet de son environnement visuel. Le terme fixation vient de l'apparente position de l'œil durant cette phase. Cependant, les fixations comportent également des mouvements oculaires en raison de mouvements résiduels de l'œil. Ces légers mouvements permettent de modifier constamment la zone examinée par la fovéa et donc d'exciter celle-ci en continu. Si l'œil était réellement stationnaire, c'est-à-dire en vision stabilisée, la perception visuelle disparaîtrait progressivement à cause du mécanisme inhibiteur de l'attention (voir le paragraphe 2.2.2.4).

2.2.1.3 Autres types de mouvement

Les autres mouvements oculaires sont brièvement décrits ci-dessous :

– les mouvements de poursuite : ce type de mouvement permet de suivre un objet en mouvement et de le stabiliser sur la rétine. L'image de cet objet peut alors être examinée par la fovéa avec un fort pouvoir de résolution. La vitesse angulaire limite de poursuite visuelle est d'environ $30°/s$;

– les mouvements de vergence (convergence et divergence) : pour lesquels les yeux se déplacent dans des directions horizontales opposées. Ce type de mouvement permet de conserver une vision binoculaire, c'est-à-dire le mécanisme réalisé par le cerveau permettant d'obtenir une perception tridimensionnelle d'un objet à partir des deux images différentes venant respectivement des deux yeux. Par exemple, lorsqu'un objet fixé par un observateur se rapproche de lui, ses yeux se rapprochent pour conserver une vision binoculaire (mouvement de convergence), alors qu'ils auront plutôt tendance à s'écarter (mouvement de divergence) si l'objet s'éloigne.

2.2.2 Attention visuelle sélective

2.2.2.1 Définition

L'attention visuelle désigne le mécanisme de sélection des informations visuelles spatio-temporelles pertinentes du monde visible. La quantité d'informations visuelles provenant de notre environnement étant supérieure aux capacités de traitement du SVH, celui-ci s'est adapté en utilisant différentes stratégies bien particulières pour réduire la quantité d'informations à traiter et ne conserver que les plus pertinentes. L'attention visuelle nous permet d'utiliser de façon optimisée nos ressources biologiques et ainsi seule une partie des informations incidentes est transmise aux aires supérieures de notre cerveau [Bal91]. En 1993 R. Milanese [Mil93], puis plus tard en 1995 J. K. Tsotos et al. [TCW+95], décrivent le mécanisme d'attention visuelle comme étant des répétitions de phases de sélection (détection et localisation) et de focalisation (mouvement oculaire ou focalisation interne).

2.2.2.2 Les mécanismes de sélection dits passifs

Les mécanismes de sélection passifs sont liés aux caractéristiques intrinsèques du SVH. Les principaux mécanismes sont rappelés ci-dessous :

– le premier mécanisme et le plus évident concerne la transduction photoélectrique (transformation de la lumière en un signal interprétable par le cerveau). Cette transformation ne concerne qu'une bande étroite du spectre global de la lumière incidente, appelée la lumière visible ;
– l'information est échantillonnée par les cellules photosensibles de façon non uniforme : au centre de la rétine, c'est-à-dire la fovéa, la restitution de la résolution spatiale est plus importante que sur le reste de la rétine ;
– les cellules visuelles présentent une sensibilité aux fréquences spatiales, ce qui implique que nous ne sommes pas en mesure d'apprécier tous les détails de notre environnement visuel avec le même degré de précision ;
– les cellules rétiniennes et corticales suppriment la redondance d'informations ; elles répondent presque exclusivement qu'aux variations d'amplitude suffisantes (contraste).

2.2.2.3 Les mécanismes de sélection dits actifs

Afin d'illustrer le concept de l'attention visuelle, celle-ci est souvent représentée par un faisceau lumineux [Nei67] (*spotlight of attention*) qui illumine les zones de notre champ visuel qui sont inspectées. La focalisation d'attention, c'est-à-dire l'inspection d'une zone particulière peut être réalisée de deux façons : par une focalisation dite *overt* ou une focalisation dite *covert*. Le premier type de focalisation se concrétise par un mouvement oculaire. La deuxième utilise la vision périphérique, c'est-à-dire la vision en dehors de la fovéa et des champ visuel qui l'entoure (zone parafovéale). Cette forme d'attention est particulièrement bien

mise en évidence chez les malentendants [BTH+00, MRL03] : des expériences oculométriques[3] présentant des séquences vidéo où une personne traduit un discours en langage des signes, ont montré que l'attention fovéale des malentendants se portait essentiellement sur le visage de la traductrice. Bien qu'ils ne fixaient pas directement les signes, ils étaient tout à fait capables de retranscrire le discours.

L'attention visuelle repose sur deux mécanismes distincts [BS90] :

- un mécanisme exogène (pré-attentif) [Pos80] ou plus communément appelé *Bottom-Up* sélectionnant les informations visuelles selon leur saillance. C'est un mécanisme éphémère et involontaire guidé par les informations de notre champ visuel (déplacement oculaire vers les zones capturant notre attention). Ce mécanisme de sélection s'effectue donc sans aucune connaissance a priori (c'est une vue cependant simplifiée de la réalité) ;
- un mécanisme endogène (attentif) [Pos80] ou *Top-Down*. Notre attention et le déplacement oculaire s'effectuent sous un contrôle volontaire et cognitif. En d'autres termes, ce mécanisme est guidé par la tâche à accomplir. La figure 2.5 visualise pour un même observateur et pour sept tâches différentes à effectuer les explorations visuelles associées.

Ces mécanismes et plus particulièrement le mécanisme *Bottom-Up*, nous amènent à parler de la théorie de l'intégration de caractéristiques (FIT[4] en anglais) [TG80]. Ces travaux reposent sur des expériences de recherche visuelle où le principe consiste à mesurer le temps de réaction nécessaire pour discriminer un objet cible enfoui parmi d'autres objets communément appelés distracteurs. Les objets peuvent être simples, c'est-à-dire constitués d'une seule dimension visuelle (la couleur, l'orientation, la forme...) ou composés de plusieurs dimensions (objets colorés et orientés par exemple). Les résultats obtenus révèlent des comportement distincts :

- si la cible diffère des distracteurs d'au moins une caractéristique visuelle, cas disjonctif (exemple de la figure 2.6(a)), alors le temps de réaction nécessaire pour résoudre la recherche visuelle est constant et cela quel que soit le nombre de distracteurs. On considère généralement que la cible saute aux yeux (dans la littérature scientifique de langue anglaise, le verbe *to pop-out* est souvent utilisé) ;
- si la cible est une combinaison de caractéristiques, cas conjonctif (exemple de la figure 2.6(b)), le temps de réaction augmente quasi linéairement avec le nombre de distracteurs. La recherche de la cible est séquentielle puisque tous les objets sont examinés afin de déterminer la cible.

Ainsi, le cas disjonctif est à rapprocher du mécanisme *Bottom-Up* qui permet de traiter les caractéristiques visuelles d'une scène rapidement et d'une façon massivement parallèle. Le cas conjonctif, quant à lui, est à rapprocher du mécanisme *Top-Down* qui est un mécanisme lent et traitant les informations visuelles de façon séquentielle ou série. On parle également de la dichotomie attentive/pré-attentive [TG80, WH04], qui suppose un premier traitement automatique sur l'ensemble du champ visuel suivi d'un traitement localisé déployé par l'observateur.

2.2.2.4 Mécanisme inhibiteur de l'attention visuelle

La fonction première de l'attention visuelle sélective est de diriger notre regard vers des objets d'intérêt contenus dans notre environnement. En plus des deux types d'attention pouvant être qualifiés de volontaire ou d'involontaire, un autre mécanisme appelé inhibition de retour (IOR[5] en anglais) est à prendre en considération, qui consiste à inhiber une zone inspectée afin d'éviter que notre attention visuelle se porte

[3] Elles consistent à enregistrer en continu les pauses et les trajectoires oculaires des observateurs.
[4] *Feature Integration Theory*
[5] *Inhibition Of Return*

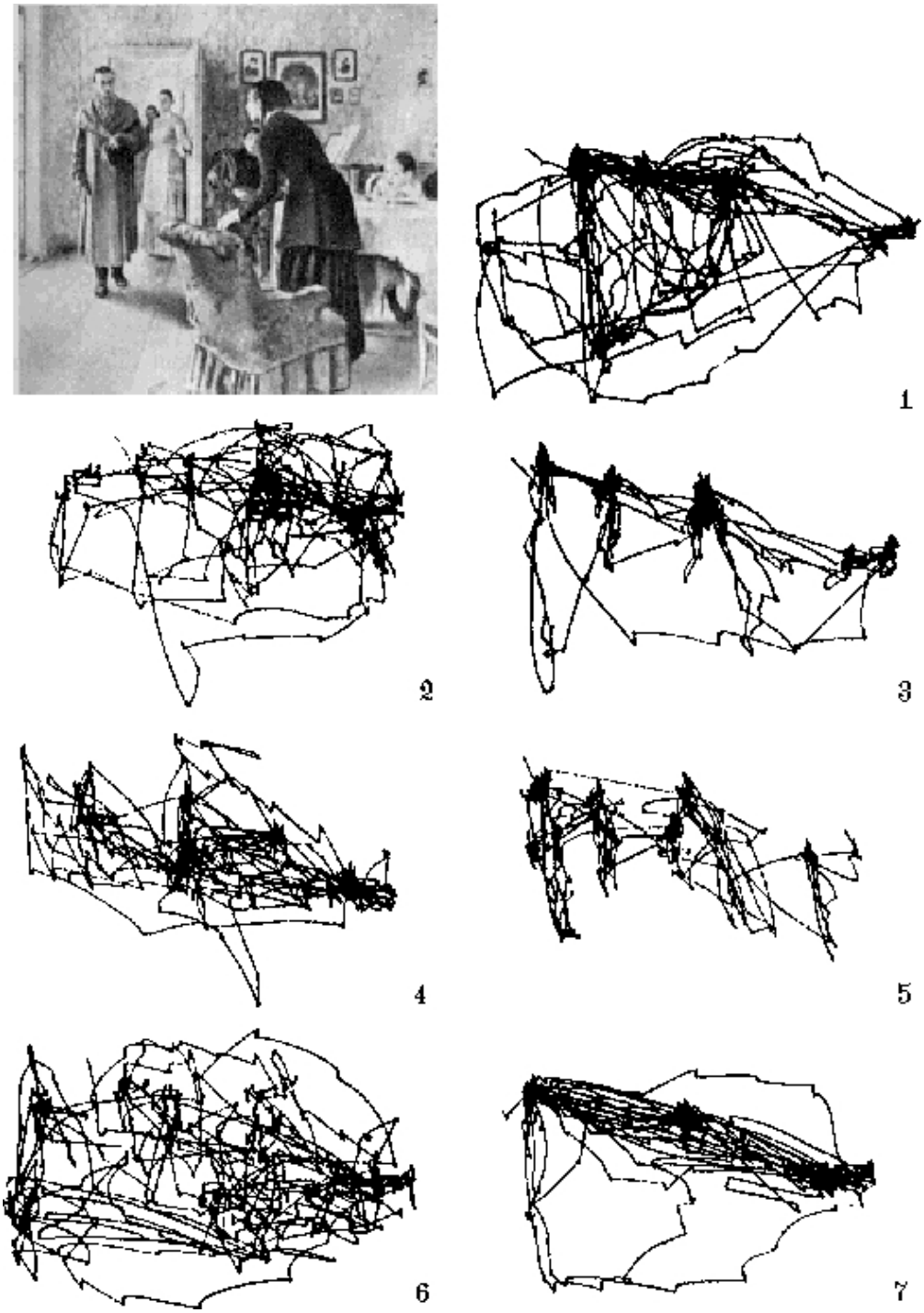
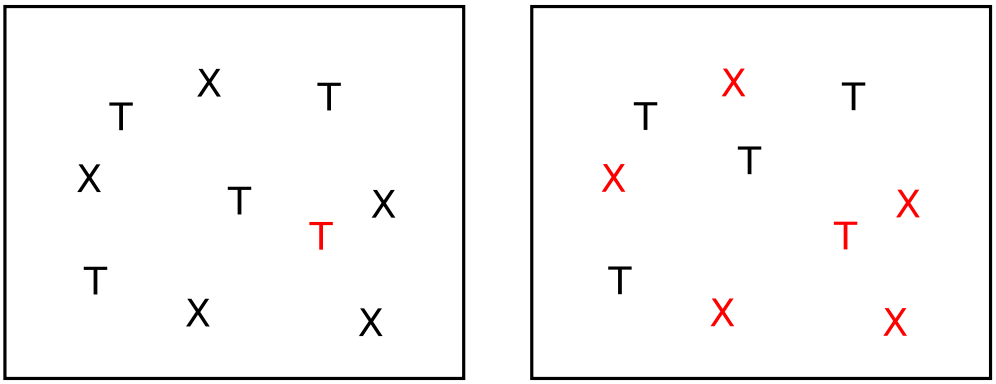


FIG. 2.5 – Impact de la tâche à effectuer sur le trajet oculaire lors de la visualisation d'une image par le même observateur : (en haut à gauche) image originale (tableau de I. E. Repin intitulé *unexpected return*) ; (1) aucune tâche n'est donnée (*Free-viewing mode*) ; (2) estimer le niveau social des personnages ; (3) évaluer leur âge ; (4) que faisaient les personnages avant l'arrivée du visiteur ? ; (5) souvenez-vous des habits portés par les différents personnages (6) mémoriser la position des personnages et des objets ; et (7) estimer depuis combien de temps le visiteur était parti (Extrait de [Yar67]).

continuellement sur cette zone. Grâce à ce mécanisme notre attention visuelle se porte séquentiellement sur les régions de notre environnement en fonction de leur saillance. L'inhibition de retour joue un rôle primordial, sans le cadre d'une recherche visuelle de type conjonctive, puisqu'elle évite à l'observateur de re-tester continuellement les mêmes objets [KMI99]. D'après les études de M. Posner et Y. Cohen [PC84], la durée d'inspection d'une zone doit être supérieure à 300 *ms* pour que le mécanisme d'inhibition de retour ait lieu.

2.2.2.5 Les caractéristiques visuelles attirant l'attention

Comme nous venons de le voir, l'humain possède une attention sélective signifiant que notre système visuel répond de façon privilégiée à un certain nombre de signaux provenant des objets et des évènements de notre environnement, le plus intuitif étant certainement l'apparition soudaine d'un objet dans une scène [YJ96]. De façon plus générale, l'attention visuelle réagit aux singularités locales [TG80]. La figure 2.6 (a) illustre un exemple classique de singularité locale « sautant aux yeux ». Par ailleurs, la sémantique joue un rôle important dans le déploiement de l'attention visuelle. Lorsqu'un objet est incohérent avec la scène, les observateurs ont tendance à faire des fixations plus longues et plus fréquentes sur cet objet saillant sémantiquement [HWH99]. Différentes études ont été menées afin d'estimer, à partir de points de fixation réels, les similarités des caractéristiques visuelles attirant notre regard [MR97, RZ99]. Ces études montrent d'une part que les régions fixées présentent un contraste (de luminance, de couleur, de texture [PN04], de mouvement, etc.) plus important que les autres régions et d'autre part que les régions fixées diffèrent de leur voisinage. Ces mesures tendent à montrer que le système visuel essaie de maximiser l'information à transmettre au cerveau en minimisant la redondance spatio-temporelle de celle-ci.



(a) Trouver la lettre T rouge. (b) Trouver la lettre T rouge.

FIG. 2.6 – Exemples d'expériences de recherche visuelle : (a) une seule lettre T rouge et les autres lettres noires : cas disjonctif (traitement parallèle) ; (b) mélange de lettres T et X rouges et noires : cas conjonctif (traitement série).

2.2.3 Conclusion

Face à la quantité d'informations visuelles de notre environnement, notre SVH s'est adapté et a adopté des stratégies particulières afin de réduire et de sélectionner les informations à traiter. Ce mécanisme de sélection des informations visuelles spatio-temporelles pertinentes constitue l'attention visuelle. Celle-ci est étroitement liée aux mouvements oculaires que nous avons décrits. Elle nous permet d'utiliser de façon optimisée nos ressources biologiques. Notre SVH répond donc de façon privilégiée à certains signaux provenant des objets et des événements de notre environnement. Considérant cette propriété de sélection de l'informa-

tion de notre SVH, des opportunités se présentent pour le codage et plus particulièrement la compression des images et des séquences vidéos. En effet, puisque les différentes zones d'une image ne seront pas traitées de la même façon par notre SVH, il est envisageable de réaliser ce traitement dès le codeur. Ainsi afin de réduire le débit tout en maintenant la même qualité perceptuelle, on peut envisager de compresser plus fortement les zones moins pertinentes d'une image ou d'une séquence d'images, et optimiser la qualité des zones significatives pour notre SVH. Cette compression adaptative des différentes régions d'une image ou d'une séquence vidéo peut être réalisée à partir d'un modèle simulant l'attention de notre SVH et indiquant les zones plus ou moins significatives.

2.3 Modélisation de l'attention visuelle pré-attentive

2.3.1 Introduction

Nous l'avons vu, le système visuel développe des stratégies particulières pour traiter l'énorme quantité d'informations à laquelle il est confronté. L'une d'elles est l'attention visuelle permettant de concentrer les ressources sensorielles sur des zones particulières de notre environnement. La concentration de ces ressources de traitement est effectuée soit de façon volontaire soit de façon involontaire. Le premier est un traitement automatique très rapide réalisé inconsciemment. Le deuxième est un processus contrôlé nécessitant un effort cognitif important et nécessitant la quasi totalité des ressources attentionnelles. Ce mécanisme est déployé lorsqu'une tâche particulière doit être effectuée. Du fait de la grande complexité des mécanismes et des interactions, des interdépendances existantes entre ces mécanismes, modéliser entièrement et finement l'attention visuelle est une tâche aujourd'hui trop complexe. De plus, le cadre d'utilisation est limité, puisqu'il consiste à la simple visualisation (aucune tâche demandée) de la télévision. Nous nous intéresserons donc ici aux zones de notre environnement visuel qui attire notre attention de façon involontaire. Une mise en œuvre possible est de modéliser seulement l'attention visuelle pré-attentive, qui est le mécanisme sélectionnant les zones pertinentes de notre environnement visuel de façon involontaire. Dans la suite du document, nous présentons quelques types de modélisation parmi les importantes.

2.3.2 Modèles empiriques et statistiques

2.3.2.1 Modèles empiriques

Les modèles que nous appelons ici empiriques se basent sur des méthodes de traitements classiques d'images, relativement éloignées des propriétés du SVH. Nous ne donnons ici que les caractéristiques principales des modèles les plus connus. W. Osberger et A. Maeder [OM98] déterminent la carte d'importance d'une image donnée en effectuant préalablement une segmentation de la source en régions homogènes en fonction de la variance locale des régions. Une importance est donnée à chaque région relativement aux critères très intuitifs tels que la taille, le contraste, la forme... Ils combinent ensuite les différents facteurs associés à chaque région afin d'obtenir une carte d'importance globale pour l'image. Chaque facteur est traité comme étant d'importance équivalente. Évidemment, la qualité de cette hiérarchisation dépend fortement de la qualité de la segmentation. Dans le même ordre d'idée, J. Luo et A. Singhal [LS00] définissent un ensemble d'éléments visuels susceptibles d'attirer l'attention. Ces derniers sont extraits afin de piloter une segmentation favorisant l'apparition de régions d'intérêt. Enfin, citons les méthodes de X. Marichal et al. [MDDV+96] et de J. Zhao et al. [ZSO+96] qui dérivent des méthodes précédemment exposées. Ces différentes méthodes n'ont pas été validées via des expérimentations oculométriques. Une autre étude in-

fluente est celle de C. Privitera et L. Stark [PS00]. Ces derniers ont évalué la pertinence de dix algorithmes de traitement d'images pour la détection de régions d'intérêt. L'évaluation de la pertinence se fait par comparaison des régions d'intérêt dites « algorithmiques et humaines ». Ils montrent que chacun des algorithmes considérés est pertinent pour un ensemble restreint d'images et présente de mauvais résultats pour les autres images. Ce résultat suggère qu'il n'est pas envisageable d'utiliser un seul procédé de traitement d'images pour détecter des régions d'intérêt de façon fiable.

2.3.2.2 Modèles statistiques

Le modèle statistique le plus connu est celui proposé par A. Oliva et al. [OTCH03]. Bien qu'a priori purement statistique, ce type de modélisation utilise une propriété du SVH vérifiée par différents travaux et citée dans le paragraphe 2.2.2.5. Le principe consiste à conjecturer que la capacité d'attraction d'une zone est inversement proportionnelle à sa probabilité d'apparition. En d'autres termes, notre attention visuelle est attirée par les zones en contraste avec leurs voisinages. Dans cette approche, chaque site (pixel ou autre...) de l'image source se caractérise par un vecteur $v_l(x) = \{v_l(x,k)\}_{k=1,..,N}$ obtenu via une décomposition hiérarchique. À partir de la propriété énoncée préalablement (la saillance est d'autant plus importante que les mesures locales sont incongrues), la carte de saillance S est obtenue en prenant l'inverse de la probabilité d'apparition d'un élément :

$$S(x) = \frac{1}{p(v_l)} \tag{2.10}$$

La probabilité $p(v_l)$ est approximée par une densité de probabilité Gaussienne :

$$p(v_l) = \frac{1}{(2\pi)^{N/2}|X|^{N/2}} e^{-1/2(v_l-\mu)^T X^{-1}(v_l-\mu)} \tag{2.11}$$

Dans l'article [OTCH03], les auteurs montrent que les performances de ce modèle sont comparables à celles du modèle de L. Itti décrit dans la section ci-après.

Le modèle de B. Bruce et E. Jernigan [BJ03] inspiré du modèle de L. Itti a la particularité d'intégrer dans une architecture classique de type C. Koch et S. Ullman, un opérateur statistique basé sur le même principe que celui utilisé par A. Oliva et al [OTCH03]. B. Bruce et E. Jernigan utilisent la mesure d'incertitude (c'est-à-dire l'espérance mathématique de l'entropie) pour définir la saillance d'un évènement.

2.3.3 Modèles psycho-visuels

2.3.3.1 L'architecture de base

Le premier modèle d'attention visuelle basé sur une architecture biologiquement plausible fut définit par C. Koch et S. Ullman [KU85] à partir des travaux sur la théorie de l'intégration des caractéristiques visuelles (*Feature Integration Theroy*) de A. Treisman et G. Gelade [TG80]. Cette architecture est présentée figure 2.7.

Un ensemble de caractéristiques visuelles pré-attentives est extrait de l'image source. Ces caractéristiques correspondent à des primitives visuelles rapidement extraites par le SVH. Il n'existe aucune liste exhaustive établie de ces caractéristiques, mais les auteurs semblent s'accorder sur un certain nombre d'entre elles [WH04] : le contraste, le mouvement, la couleur, l'orientation...

Les caractéristiques pré-attentives peuvent être identifiées à partir d'expériences psychophysiques, en mesurant le temps de réaction nécessaire pour détecter une cible parmi un ensemble d'éléments perturba-

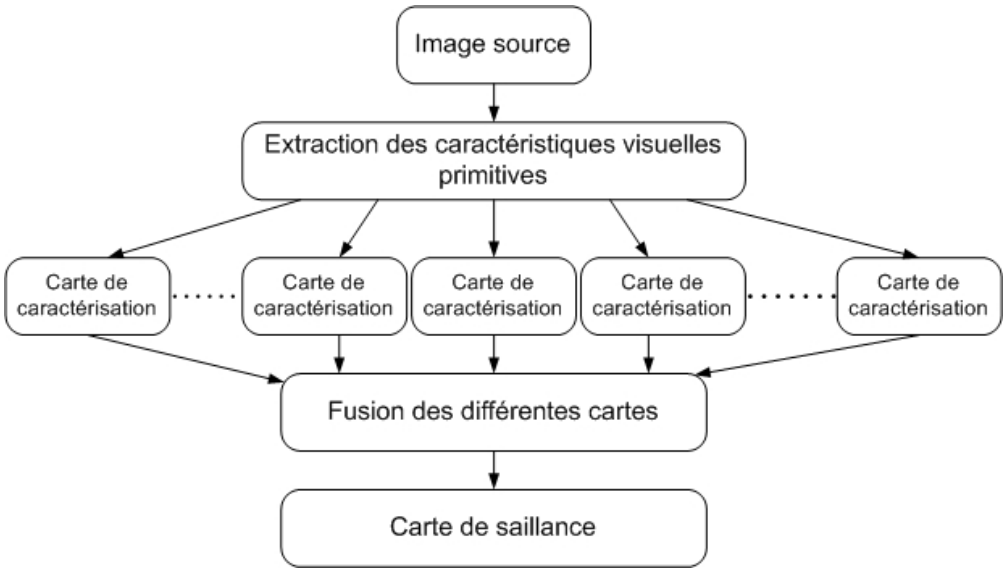


FIG. 2.7 – Architecture biologiquement plausible d'un modèle d'attention visuelle proposée par C. Koch et S. Ullman [KU85].

teurs (la figure 2.6 présente un exemple d'expériences psychophysiques et le paragraphe 2.2.2.1 définit la notion de temps de réaction). Ces caractéristiques pré-attentives sont extraites en parallèle, générant ainsi des cartes de caractérisation respectant la topologie de la source. Des mécanismes d'inhibition latérale, simulant les champs récepteurs des cellules visuelles, permettent d'isoler les zones de l'image présentant des caractéristiques visuelles différentes de leurs voisinages. Ces mécanismes agissent comme des détecteurs de contraste. Enfin, la carte de saillance, qui est l'un des caractères les plus novateurs de ces travaux, est construite en combinant les différentes cartes de caractérisation. Cette carte représente la saillance pour toutes les positions de la scène visuelle. La définition originelle donnée par C. Koch et S. Ullman est la suivante : la carte de saillance est une représentation de l'environnement accentuant les régions d'intérêt du champ visuel.

2.3.3.2 Exemples de modèles psycho-visuels de l'attention visuelle

Le modèle le plus connu est incontestablement celui développé par L. Itti, C. Koch et E. Niebur. Les nombreuses parutions scientifiques [IKN98, IK00, IKB00, IK01] ainsi que la disponibilité des codes source ont favorisé cette popularité. Leur modèle est basé sur l'architecture proposée par Koch et Ullman et est divisé en plusieurs étapes :

- la première consiste à créer trois canaux à partir d'une image (r,g,b) :
  1. un canal lié à l'intensité : $I = (r+g+b)/3$;
  2. un canal couleur constitué de quatre composantes et donc lié à la théorie des couleurs antagonistes : rouge $R = r-(g+b)/2$, vert $G = g-(r+b)/2$, bleu $B = b-(g+r)/2$ et jaune $Y = (g+r)/2-|r-g|/2-b$;
  3. un canal pour les composantes orientées, obtenu à partir du canal intensité I et d'une pyramide de Gabor orientée $O(\sigma,\theta)$, où σ indique le niveau de la pyramide et $\theta\varepsilon\ \{0°,45°,90°,135°\}$ l'orientation exprimée en degré.
- une décomposition hiérarchique sur neuf niveaux à l'aide de pyramides Gaussiennes est effectuée sur chaque composante. Ces pyramides représentent une approximation du pavage fréquentiel des

cellules visuelles ;

- un mécanisme permet ensuite d'extraire les informations pertinentes contrastant avec leur voisinage des différents niveaux de la pyramide. Pour le canal de luminance (I) six cartes sont calculées. Pour le canal couleur, deux ensembles de cartes (douze au total) sont construits représentant respectivement la double opposition rouge/vert et vert/rouge, et la double opposition bleu/jaune et jaune/bleu. Pour le canal orientation, la convolution de la pyramide Gaussienne issue de la composante intensité I avec la pyramide de Gabor orientée, fournit 24 cartes ;
- une étape de normalisation notée $N(.)$ est appliquée sur chaque carte indépendamment les unes des autres pour créer une carte de saillance par canal. Ils proposent trois méthodes différentes de normalisation :
  1. une sommation naïve consistant à normaliser toutes les cartes sur la même dynamique et à moyenner la somme des différentes cartes ;
  2. une amplification non linéaire globale dépendante du contenu de la carte, et consistant à multiplier la carte normalisée dans une dynamique $[0\ldots M]$ par le facteur $(M-\overline{m})^2$, où M représente le maximum global de la carte et $\overline{m}$ représente la moyenne de tous les autres maxima locaux. Par conséquent, lorsque la carte normalisée ne contient que quelques pics de saillance, ces derniers sont amplifiés. Par contre, lorsque la distribution de contraste est quasi uniforme, les valeurs de saillance sont atténuées ;
  3. une méthode de normalisation plus proche de la réalité biologique en tentant d'une part de reproduire le comportement des champs récepteurs non classiques et d'autre part de modéliser les connections horizontales étendues.
- à partir des cartes normalisées, une carte de saillance est construite pour chaque canal. Ils suggèrent en fait que pour un canal donné, toutes les cartes soient mises à l'échelle du niveau quatre de la pyramide et cumulées point à point. Ils obtiennent ainsi $\overline{I}$, $\overline{C}$ et $\overline{O}$ respectivement pour la carte de saillance en intensité, en couleur et en orientation ;
- finalement la carte de saillance finale S est obtenue de la façon suivante :

$$S = \frac{1}{3}(N(\overline{I}) + N(\overline{C}) + N(\overline{O})) \tag{2.12}$$

D'après les travaux de L. Itti, ce sont les deux dernières méthodes qui offrent les meilleurs résultats. Le tableau 2.1 présente les caractéristiques principales de plusieurs modèles psycho-visuels de l'attention visuelle.

Les différents modèles d'attention visuelle sont pour la plupart basés sur l'architecture de C. Koch et S. Ullman. Le plus célèbre est celui de L. Itti. Le défaut majeur du modèle de A. Chauvin [CHMP00, Cha02] réside certainement sur le fait que la couleur ne soit pas traitée. Le fait de se contenter de la luminance pour déterminer les régions visuellement importantes réduit les performances et contourne la difficile étape de fusion de cartes provenant d'origines différentes. L. Itti a proposé des méthodes de fusion dont une tout à fait intéressante. La méthode de fusion itérative reproduit l'aspect temporel de la vision. En dépit de l'aspect intéressant de la méthode proposée par L. Itti, la fusion reste un problème paraissant mal abordé, particulièrement lorsqu'il y a plusieurs cartes de saillance (luminance, orientation, couleur...). Seul R. Milanese [MBP92, Mil93] a proposé une méthodologie de fusion qui nous semble très pertinente mais peu exploitée actuellement. L'algorithme de fusion défini doit répondre aux principes suivants :

1. la carte de saillance finale doit être un « résumé » de toutes les cartes de saillance intermédiaires ;

Modèles	**Caractéristiques visuelles**	**Opérations**
Milanese et al. [MBP92, Mil93]	Intensité, couleur, amplitude des contours et la courbure.	Filtrage des caractéristiques visuelles par un filtre passe bande orienté, filtre passe-bas isotrope Gaussien sur les différentes cartes de saillance, relaxation par descente de gradient d'une fonction énergétique, seuillage pour former une carte binaire.
Chauvin et al.[CHMP00, Cha02]	Intensité, orientations.	Extraction des primitives visuelles pré-attentives à partir de 32 ondelettes de Gabor (quatre bandes de fréquences et huit orientations), normalisation de type inhibition par division, filtrage des canaux (produit d'une gaussienne par un masque en forme de papillon), application itérative d'une différence de deux Gaussiennes, fusion par combinaison linéaire des différentes sous-bandes fréquentielles.
Canosa et al. [Can03]	Intensité, couleur, orientations.	Décomposition hiérarchique de la carte d'importance en intensité et via des filtres orientés passe-bande de Gabor, partition de l'image en régions de premier plan et d'arrière plan, fusion par combinaison linéaire des différentes cartes.
Le Meur et al.[LM05, LMLCBT06]	Luminance, deux composantes de chromatiques.	Décomposition en sous-bandes orientées dans le domaine de Fourier, Fonctions de Sensibilité au Contraste, Masquage visuel, renforcement achromatique, interactions centre/pourtour, interactions facilitatrices, normalisation des caractéristiques visuelles en fonction de leur seuil différentiel de visibilité, fusion.
Bur et al. [BH07]	Intensité, deux composantes chromatiques, orientations.	Pyramides Gaussiennes diadiques et de Gabor, interactions centre/pourtour, normalisation de chaque composante en fonction d'une valeur maximale significative, fusion.
Marat et al. [MHPG+09, MGP09, Mar10]	Intensité, orientations, détection de visages, mouvement.	Modèle à trois voies comprenant : une voie statique, une voie dynamique et une voie dédiée aux visages. Pré-traitement de l'information visuelle par un filtre « rétinien », puis décomposition par des filtres « corticaux » (filtres orientés passe-bande de Gabor), prise en compte des interactions entres les orientations, normalisation et sommation des cartes intermédiaires pour la voie statique ; compensation du mouvement de la caméra et estimation du mouvement pour la voie dynamique ; détection des visages et normalisation pour la voie visage ; fusion.

TAB. 2.1 – Caractéristiques principales des modèles psycho-visuels de l'attention visuelle pré-attentive.

2. la carte de saillance doit isoler clairement les régions perceptuellement importantes (idéalement, celle-ci doit être binaire) ;
3. les régions perceptuellement importantes doivent correspondre aux objets les plus saillants physiquement et sémantiquement. Les composantes de chaque objet sont de moindre importance ;
4. les formes des régions détectées doivent être compactes et approximer la forme des objets.

Le modèle proposé par Le Meur [LM05, LMLCBT06] s'appuie également sur l'architecture de C. Koch et S. Ullman, mais la philosophie sous-jacente est différente. À partir d'une image incidente, un espace psycho-visuel est construit. Il est constitué des composantes naturelles de notre environnement, c'est-à-dire d'une composante achromatique et de deux composantes chromatiques. La différence fondamentale vis à vis de l'état de l'art se situe dans la normalisation de ces composantes. En effet, afin d'obtenir des caractéristiques visuelles homogènes et comparables, elles sont toutes normalisées par rapport à leur seuil différentiel de visibilité. Ainsi, que ce soit une valeur liée à une composante achromatique ou à une composante chromatique, elle s'exprime en fonction de leur seuil de visibilité. Comme évoqué précédemment, le premier intérêt est d'avoir des données homogènes, donc comparables. Le deuxième concerne la hiérarchisation des données. Les informations inférieures au seuil de visibilité sont négligées alors que d'autres sont mises en exergue. Il ne s'intéresse donc qu'aux données perceptibles par le système visuel. À partir de ce cadre conceptuel cohérent, la mesure de saillance de chaque site est déterminée en transformant les valeurs exprimées en terme de visibilité en valeurs de saillance. Ce calcul est fondé sur des modèles mathématiques déduits d'expériences psychophysiques. Après cette étape, l'ensemble des données de l'espace psycho-visuel, exprimé en terme de visibilité, est utilisé pour déterminer les zones saillantes.

Le modèle proposé par Marat [Mar10] reprend les études menées par Chauvin [Cha02] et Guironnet [GGPL05] en y apportant un certain nombre d'améliorations. Le modèle proposé est inspiré des premières étapes de traitement du SVH, à savoir, à partir des cellules de la rétine jusqu'aux cellules complexes du cortex visuel primaire. L'information visuelle est pré-traitée par un filtre « rétinien », puis décomposée par des filtres « corticaux ». La rétine extrait deux signaux de chaque image qui correspondent aux deux sorties principales de la rétine. Chaque signal est ensuite décomposé en caractéristiques élémentaires (fréquences spatiales et orientations) à l'aide un banc de filtres « corticaux » (filtres orientés passe-bande de Gabor). Ces filtres sont utilisés pour extraire à la fois les informations statique et dynamique, selon leur sélectivité fréquentielle, et fournissent deux cartes de saillance : une statique et l'autre dynamique. Les deux cartes de saillance sont ensuite combinées pour obtenir une carte de saillance spatio-temporelle pour chaque image de la séquence vidéo.

2.3.4 La dimension temporelle dans la modélisation

La modélisation de l'attention visuelle pour des séquences vidéo n'a pas encore fait l'objet de nombreuses études. L'intégration de cette dimension temporelle est cependant de plus en plus abordée dans quelques articles [DI03, YPG01] ou thèses [Par02, LM05]. Certains modèles spatio-temporels proposés sont une extension relativement simple du modèle de L. Itti. D'ailleurs, celui-ci avait déjà suggéré une généralisation possible de son modèle à de nombreuses dimensions visuelles. Évidement, cette extension au temporel souffre des mêmes défauts que ceux du modèle spatial. En effet, la carte de saillance temporelle est déterminée de la même façon que celle décrite au paragraphe 2.3.3.2. Dans ses travaux de thèse [Par02], D. Parkhurst propose un modèle intéressant et biologiquement plausible pour le traitement de l'information temporelle. Dans le paragraphe 2.1.5.2, nous avons vu comment l'utilisation de deux canaux, un canal dit

sustained et l'autre dit *transient*, modélisent respectivement les fréquences temporelles faibles et élevées. À partir du canal *transient*, une carte de saillance est déterminée en appliquant les mêmes procédés que ceux de L. Itti. On pourra regretter que seulement une comparaison qualitative apparaisse dans ses travaux. Dans ses travaux de thèse [LM05], O. Le Meur détermine une carte de saillance temporelle. Alors que pour la saillance spatiale, il utilise de nombreuses dimensions, la détermination de la saillance temporelle est plus directe. En effet, cette saillance est directement liée au contraste et aux singularités de mouvement. La saillance spatio-temporelle est ensuite obtenue en fusionnant la densité de saillance spatiale et la densité de saillance temporelle. Les performances de son modèle de saillance spatio-temporelle sont évaluées sur plusieurs séquences et à partir de la vérité de terrain via des tests oculométriques en utilisant deux métriques (fonction de probabilités cumulées et de matrices de confusion). Le modèle proposé par Marat [Mar10], combine les cartes de saillance issues d'une voie statique et d'une voie dynamique. Pour la voie dynamique, le mouvement de la caméra est d'abord compensé, ensuite comme pour la voie statique, un filtre « rétinien » est appliqué aux images. À la sortie de ce filtre, la vitesse des régions en mouvement par rapport au fond est calculée en utilisant un estimateur de mouvement. Plusieurs méthodes de fusion des deux cartes (statique et dynamique) sont évaluées à partir des véritables positions oculaires et en utilisant différents critères de comparaison. Après avoir constaté que les visages étaient un attracteur visuel, ils ont proposé un modèle à trois voies [MGP09] qui regroupe : une voie statique, une voie dynamique et une voie dédiée aux visages. Ce nouveau modèle à trois voies a ensuite été validé à partir d'expériences oculométriques.

La figure 2.9 donne des exemples de cartes de saillance obtenues pour trois des modèles présentés dans le tableau 2.1. La figure 2.9 présente les aires obtenues sous les courbes après une analyse ROC (*Receiver Operating Characteristic*) entre les cartes de saillance prédites par les modèles de Itti [IKN98], Le Meur [LMLCBT06] et Bur [BH07] et les cartes de fixations obtenues après des expériences oculométriques [LMLCBT06].

2.3.5 Conclusion

Les différents travaux effectués pour modéliser l'attention visuelle sont généralement basés sur l'architecture de C. Koch et S. Ullman, le plus célèbre car étant parmi les premiers est celui de L. Itti qui obtient des résultats satisfaisants. Le modèle statistique de A. Oliva [OTCH03] obtient des résultats similaires à ceux du modèle de L. Itti. Notons que les différents modèles proposés se distinguent également par leurs méthodes de fusion des cartes provenant d'origines différentes. Cependant très peu de travaux font référence à la dimension temporelle qui est primordiale dans la modélisation de l'attention visuelle. En effet, dans un contexte de recherche visuelle, J. Wolfe [WCF89] a clairement identifié le mouvement comme un attracteur visuel. D'ailleurs, W. Osberger [OMB98] considère la carte de saillance temporelle comme étant cinq fois plus importante que n'importe quelle autre carte de saillance issue de la dimension spatiale (couleur, intensité…). Enfin, dans ses travaux, R. Milanese [Mil93] considère que les régions perceptuellement importantes doivent correspondre aux objets les plus saillants physiquement et sémantiquement, et que les formes des régions détectées doivent être compactes et approximer la forme des objets.

Dans ce contexte d'étude, un objet en contraste de mouvement est l'élément déterminant qui attire notre attention visuelle. L'une des premières étapes est donc de détecter les objets en mouvement afin d'obtenir des informations pertinentes le long de la dimension temporelle et les intégrer au sein du modèle de l'attention visuelle. Un tel modèle permettrait donc d'identifier les zones (les objets) les plus significatives d'un point de vue perceptuel. Ayant connaissance de ces informations, on pourrait ensuite envisager de les transmettre au codeur afin de le guider dans ses choix pour coder les images en fonction de leur contenu

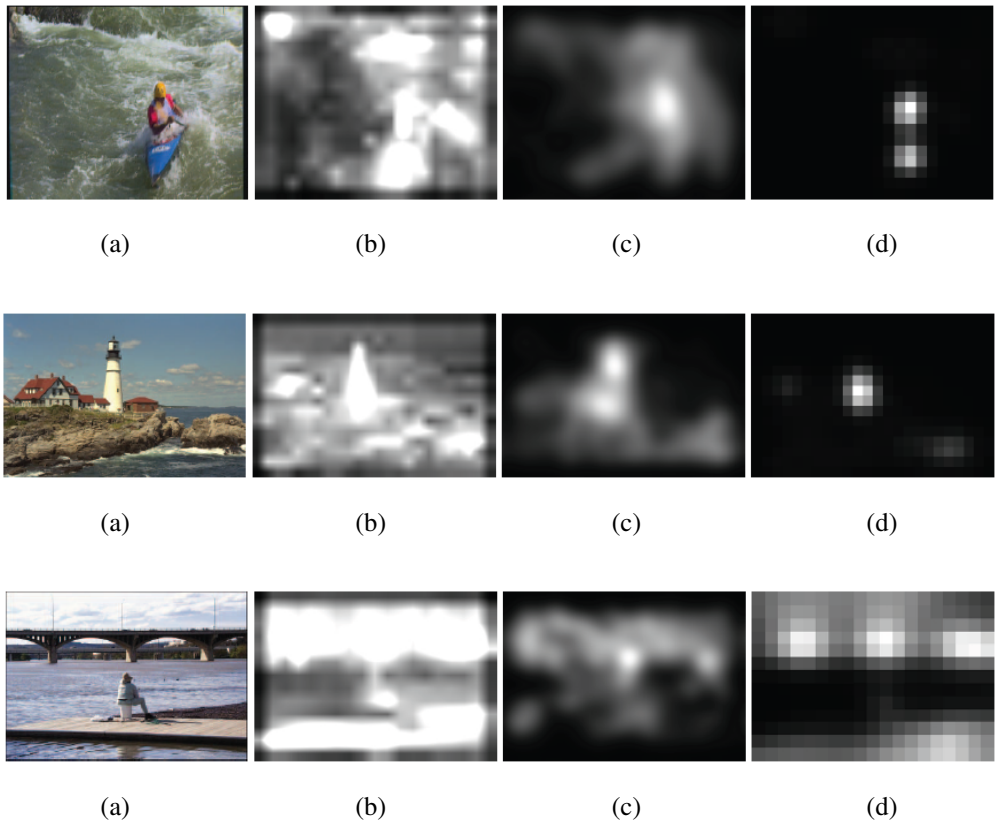


FIG. 2.8 – Cartes de saillances de différents modèles d'attention visuelle. (a) images originales (haut : *Kayak*, milieu : *Lighthouse* et bas : *Manfishing*) et leurs cartes de saillance obtenues à partir du modèle de Itti [IKN98] (b), Le Meur [LMLCBT06] (c) et Bur [BH07] (d).

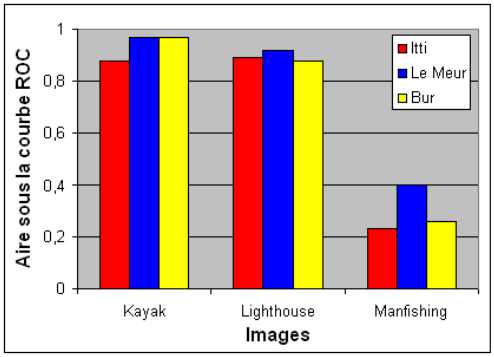


FIG. 2.9 – Aire sous la courbe après une analyse ROC des modèles de Itti [IKN98], Le Meur [LMLCBT06] et Bur [BH07] pour les images *Kayak*, *Lighthouse* et *Manfishing*.

et de la saillance des objets les constituants. Notre méthode de pré-analyse avant codage vidéo doit donc être capable de détecter les objets ou les régions les plus saillants en fonction de leur mouvement et de leurs caractéristiques spatiales, pour ensuite transmettre des informations au codeur et de réaliser un codage en fonction de l'importance visuelle des différentes régions. La gestion des ressources de codage sera ainsi réalisée en fonction d'informations extraites par la pré-analyse de la vidéo, et non plus par les critères classiques d'optimisation débit-distorsion.

2.4 Applications possibles d'un modèle de l'attention visuelle au sein d'un codeur vidéo

Nous l'avons vu (chapitre précédent), les ressources de codage au sein d'un codeur vidéo sont souvent simplement distribuées au vu des critères classiques d'optimisation débit-distorsion. Dans un contexte de codage avec pertes, la distribution adaptée des ressources de codage permettrait d'accroître substantiellement la qualité globale perçue. L'idée est simple puisqu'elle consiste à favoriser la qualité des zones les plus importantes visuellement. On parle alors de compression sélective ou de compression avec régions d'intérêt. Cela nécessite de disposer d'informations a priori sur la scène à coder, qui peuvent être obtenues via une modélisation de l'attention visuelle.

Nous distinguons deux façons, qui peuvent être complémentaires, d'effectuer une compression sélective : la première dite indirecte consiste à altérer judicieusement le signal à coder de façon à réduire la quantité d'information sur les zones de moindre intérêt. La deuxième façon, dite directe, modifie directement le coeur de codage en fonction de la connaissance des régions d'intérêt.

La compression sélective indirecte consiste donc à réduire la quantité d'information de la séquence vidéo à coder tout en conservant l'information originale sur les régions d'intérêt. Un prétraitement est réalisé en amont en fonction des données issues de la carte de saillance. Si par exemple l'objectif est de diminuer le débit global de codage, elle permet de maintenir une bonne qualité sur les zones visuellement importantes de la séquence vidéo. Le choix du type de prétraitement est important car il doit être complémentaire du codage et plus particulièrement de la quantification. Un pré-traitement non-linéaire semble être le choix le plus adéquat puisqu'il est peu redondant avec une quantification. Dans sa thèse [LM05], O. Le Meur propose d'utiliser un filtre appelé nivellement [Mey98] qui est la combinaison de deux opérateurs morphologiques (une ouverture par reconstruction et fermeture par reconstruction). Ce filtre est intéressant à la fois pour sa capacité à conserver les structures.

L'objectif de la compression sélective directe est de contrôler la distribution des ressources de codage en fonction de l'intérêt visuel de chaque macrobloc encodé afin d'accroître la qualité visuelle perçue. A. Bradley [Bra03] a d'ailleurs montré qu'une compression sélective sur images fixes permettait d'améliorer la qualité subjective d'une part lorsque les zones d'intérêt sont de tailles relativement faibles et d'autre part, lorsque pour une approche classique de codage, le débit fixé provoque l'apparition d'artéfacts sur les zones saillantes. La plupart du temps, le paramètre sur lequel on agit est le pas de quantification. En d'autres termes, un macrobloc présentant un intérêt visuel faible sera quantifié plus grossièrement qu'un macrobloc ayant un intérêt visuel important. W. Osberger [OMB98] propose une méthode pour contrôler la quantification au sein d'un codeur MPEG en fonction des cartes de saillance calculées. Il utilise également un modèle de masquage spatial pour redistribuer les erreurs de codage. Ainsi, les régions d'intérêt et les zones pour lesquelles les artéfacts de codage sont facilement détectables sont quantifiées plus finement alors que les régions qui sont visuellement moins importantes et les zones capables de fort masquage spatial sont

quantifiées plus grossièrement. Leur méthode permet d'améliorer la qualité subjective des séquences vidéo décodées par rapport à un codage classique. O. Le Meur [LM05] a également proposé une approche permettant de réaliser une compression sélective directe. Pour cela il estime d'abord la courbe débit-distorsion de chaque macrobloc et ensuite il réalise la modification du codage en deux étapes. La première étape consiste à déterminer le pas de quantification de chaque macrobloc satisfaisant un coût minimal imposé. La détermination des pas de quantification doit permettre, dans la mesure du possible, d'obtenir une qualité homogène sur l'image. La seconde étape consiste à redistribuer le surplus de débit en favorisant les zones saillantes de l'image à coder. L'approche proposée permet d'améliorer la qualité en terme de PSNR des régions d'intérêt ($+2dB$ en moyenne). Cependant, il observe les mêmes résultats que ceux énoncés par A. Bradley [Bra03] et L. Huguenel [Hug05]. C'est-à-dire qu'une approche de compression sélective directe est pertinente pour des régions d'intérêt de petites tailles et lorsque l'arrière plan contient des informations spatiales présentant des capacités de masquage visuel du bruit de quantification. L. Huguenel [Hug05] a montré qu'il était possible de réduire de 30% le débit du codage de séquences de sport (intégrant des tribunes) tout en conservant une qualité subjective constante.

Ces résultats montrent qu'intégrer un modèle de l'attention visuelle au sein d'une chaîne de codage (ou en amont de celle-ci) permet d'obtenir des gains en termes de compression. C'est pourquoi notre méthode de pré-analyse de la vidéo doit comprendre un tel modèle afin de déterminer les zones saillantes de la séquence d'images, et transmettre des informations au codeur afin que celui-ci puisse distribuer judicieusement ses ressources.

Conclusion

L'objectif de ce chapitre était d'étudier les différentes propriétés du SVH ainsi que leurs modèles mathématiques afin d'identifier les possibilités d'intégrer de tels modèles pour notre méthode de pré-analyse. Cette étude a permis de discerner les propriétés présentant un intérêt dans la conception d'un modèle de l'attention visuelle : la perception de la luminance, la perception des couleurs, la sensibilité au contraste, l'organisation multi-canal et les effets de masquage. Nous avons particulièrement souligné que le SVH est davantage sensible aux contrastes qu'aux luminances, et cette sensibilité varie selon la nature des signaux. La deuxième partie était consacrée à la description de l'attention visuelle, également appelée mécanismes actifs. L'attention visuelle est intimement liée aux mouvements oculaires et permet de réduire et de sélectionner les informations à traiter de notre environnement. Utiliser une telle propriété permet de trier efficacement l'information entre celle visuellement importante, et celle moins utile, l'idée peut alors être de maintenir une qualité perceptuelle constante, pour un coût en bits acceptable. La modélisation de l'attention visuelle est un domaine en plein essor. L'intégration de propriétés de bas niveau (mécanisme *Bottom-Up*) dans la modélisation permet d'obtenir des modèles de plus en plus performants. Bien que J. Wolfe [WCF89, WH04] ait clairement identifié le mouvement, ou plus précisément le contraste de mouvement, comme un attracteur visuel, encore peu de travaux intègrent la dimension temporelle dans la modélisation de l'attention visuelle. Dans ses travaux, R. Milanese [Mil93] considère que les régions perceptuellement importantes doivent correspondre aux objets les plus saillants physiquement et sémantiquement et que les formes des régions détectées doivent être compactes et approximer la forme d'un objet, ce que confirment les travaux de S. Frintrop [FBR05] et V. Navalpakkam [NI06].

La dernière partie de ce chapitre introduit rapidement deux principes de mise en œuvre de modèles de l'attention visuelle au sein de codeurs vidéo via les cartes de saillance, on parle alors de compression

sélective. La première étant indirecte car les cartes de saillance permettent de réduire directement l'information (en filtrant les images) des régions moins importantes tout en conservant celle des régions d'intérêt. La deuxième réalisation est la compression sélective directe où les cartes de saillance sont utilisées pour répartir les ressources de codage en fonction de l'intérêt de chaque macrobloc. Cette distribution des ressources de codage permet d'améliorer la qualité globale perçue.

Notre méthode de pré-analyse de la vidéo doit donc intégrer un modèle de l'attention visuelle qui prenne en considération le contraste spatio-temporel afin de déterminer la saillance des différentes régions. Les formes des régions saillantes doivent approximer les formes des objets contenus dans la séquence vidéo. À partir de ces données « haut niveau », la méthode de pré-analyse transmettrait des informations afin de distribuer les ressources de codage et guider le codeur dans ses décisions de prédiction et de quantification parmi l'ensemble des possibilités offertes par le codeur vidéo.

Chapitre 3

Pré-analyse spatio-temporelle de la vidéo en vue du codage

Sommaire

Introduction **59**

3.1 Principe général de l'outil de pré-analyse proposé **60**

3.2 Segmentation spatio-temporelle et suivi d'objets **61**

3.2.1 Introduction 61

3.2.2 État de l'art des techniques de segmentation spatio-temporelle et de suivi d'objets . 62

3.2.3 Segmentation basée mouvement 67

3.2.4 Segmentation spatio-temporelle multi-critères 86

3.3 Détermination des cartes de saillance **96**

3.3.1 Saillance spatiale basée sur le contraste de couleur 96

3.3.2 Saillance temporelle 100

3.3.3 Saillance spatio-temporelle 102

3.3.4 Résultats qualitatifs 102

3.4 Temps de calculs **103**

Conclusion **106**

Introduction

L'objet de ce chapitre est de présenter notre méthode de pré-analyse de la vidéo. Elle intègre un modèle de l'attention visuelle afin de réaliser un codage optimisé visant, par exemple pour un débit fixé, à améliorer le rendu de la vidéo décodée. Le but est donc d'analyser le contenu de la vidéo en tenant compte des informations haut niveau décrites au chapitre précédent, afin de transmettre au codeur le jeu de paramètres optimal permettant d'exploiter au mieux les différents outils de codage (prédiction et quantification). Comme nous l'avons vu dans le chapitre précédent, les études réalisées pour modéliser l'attention visuelle ont mis en évidence le caractère primordial du contraste de mouvement. Les régions identifiées comme les plus importantes visuellement doivent aussi correspondre aux formes des objets. Notre méthode de pré-analyse procédera donc en détectant d'abord les objets qui ont un mouvement propre différent de celui engendré par

la caméra puis en calculant les cartes de saillance permettant de déterminer les zones visuellement importantes pour notre modèle d'attention visuelle.

Dans la première partie du chapitre, nous présentons notre méthode de segmentation spatio-temporelle et de suivi des objets dans la séquence vidéo. Nous réalisons premièrement une segmentation basée mouvement en estimant le mouvement local et en tenant compte du mouvement global. Afin d'améliorer les résultats de cette segmentation basée mouvement et d'assurer le suivi des objets temporellement, de nouveaux critères (couleur, texture et connexité) sont intégrés. Dans la dernière partie, nous présentons notre modèle de l'attention visuelle pré-attentive.

3.1 Principe général de l'outil de pré-analyse proposé

Le système de pré-traitement est positionné en amont du codeur afin de conditionner le flux vidéo et de fournir au codeur un ensemble de paramètres adaptés à la vidéo traitée[1]. L'outil de pré-analyse est présenté à la figure 3.1. En réalité, la décomposition est un peu plus complexe puisque la vidéo présentée en entrée de l'outil de pré-traitement est découpée en plans homogènes. Un outil de détection des *scene cuts* est donc implicitement utilisé.

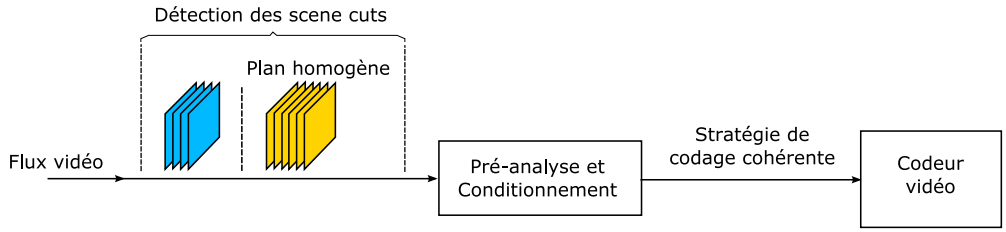


FIG. 3.1 – Spécification de l'outil de pré-analyse et de conditionnement d'un flux vidéo.

Afin d'appréhender de façon juste le mouvement des objets et leur ancrage temporel, l'analyse doit porter sur une fenêtre temporelle suffisamment large. Pour fixer la taille de cette fenêtre temporelle, nous nous sommes basés sur le temps de fixation du système visuel humain qui est sensiblement égal à $200\,ms$ [LM05]. Par exemple, pour la prochaine génération de télévision Haute Définition utilisant une définition de 1920×1080 pixels en mode progressif et une cadence de 50 images par seconde, le plan homogène en entrée sera découpé en segments temporels de neuf images (l'estimation du mouvement étant basée sur des tubes spatio-temporels définis pour un nombre impaire d'images, cf section 3.2.3.2.1), chaque segment temporel représentera ainsi $180\,ms$ de vidéo.

Il s'agit alors de déterminer les différents objets spatio-temporels qui composent chaque segment. Pour cela, le segment temporel courant doit d'abord bénéficier d'un traitement intra-segment, puis d'un traitement inter-segment afin de suivre les objets dont le cycle de vie s'étend sur plusieurs segments temporels et envoyer au codeur vidéo les paramètres pour traiter ces objets de façon cohérente temporellement.

Une fois les traitements intra et inter segment réalisés, on obtiendra pour chaque segment temporel, une carte de segmentation et une description détaillée (caractéristiques spatiales et temporelles) des objets présents dans le segment : délimitation spatiale, suivi temporel, couleur, texture, cycle de vie... À partir de ces informations, une carte de saillance spatio-temporelle pourra être déterminée (une carte par segment temporel), déterminant l'importance visuelle spatio-temporelle de chaque site, illustrant ainsi l'attractivité

[1] Le contexte du projet ANR RIAM (appel 2005) ArchiPEG délimite certaines spécificités de notre outil de pré-analyse, puisque le flux vidéo en entrée est une vidéo Haute Définition et en bout de chaîne la vidéo sera codée au format H.264. Notre méthode de pré-analyse de la vidéo est cependant générique pour les normes de codage MPEG.

visuelle d'un site. Ces informations (la carte de segmentation et sa carte de saillance visuelle associée) pourront être transmises afin de déterminer les paramètres de codage les plus adaptés (pour chaque objet).

Les spécificités exprimées ci-dessus, nécessaires à la réalisation de l'outil de pré-analyse et de conditionnement du flux vidéo, nous ont mené à décomposer ce système en un ensemble de fonctions, agencées les unes avec les autres selon le schéma bloc présenté en figure 3.2. Nous allons détaillé ces blocs de traitement au cours de ce chapitre.

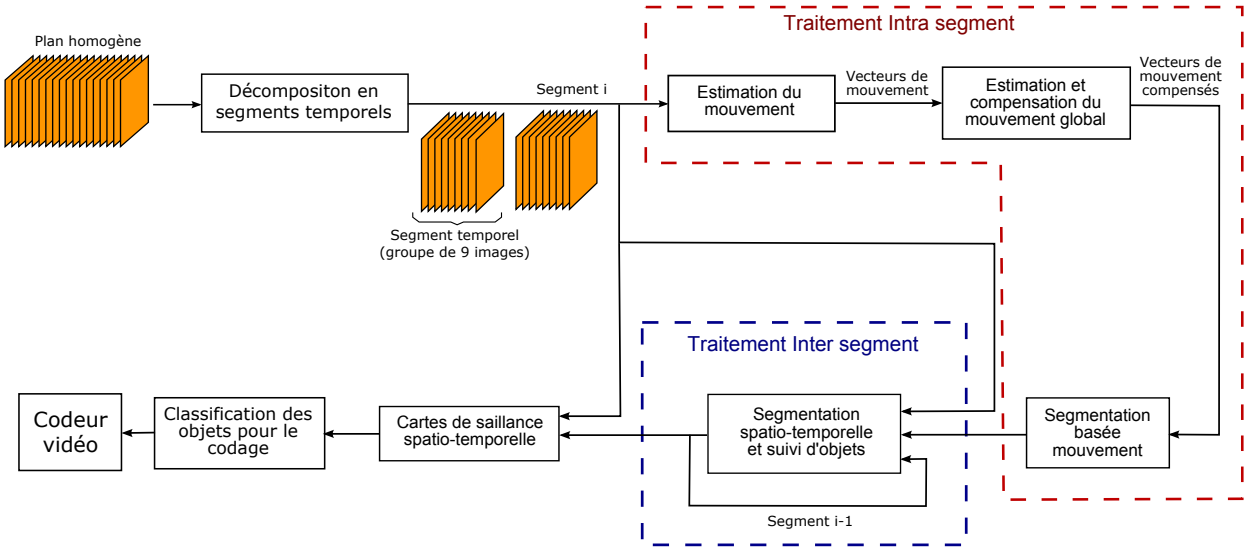


FIG. 3.2 – Synoptique global de notre outil de pré-analyse et de conditionnement du flux vidéo.

3.2 Segmentation spatio-temporelle et suivi d'objets

3.2.1 Introduction

Il n'existe pas de définition unique d'un objet vidéo. Cependant, on peut considérer que celui-ci peut être défini par une forme homogène en textures et en mouvement, il ne correspond pas exactement à un objet du monde réel. Car il est le résultat de l'analyse de la projection d'un monde 3D sur un plan. Ainsi, un objet vidéo est défini comme une région de la vidéo satisfaisant un modèle, par exemple un modèle de mouvement ou un modèle d'objet physique (modèle de visage, de corps humain, etc.). La notion d'objet est alors entièrement définie par le modèle utilisé. Afin de détecter automatiquement des objets vidéo, il est nécessaire de proposer le modèles auquel ces objets répondront, on parle de la segmentation automatique en objets vidéo.

Nous voulons détecter les objets présents dans la vidéo afin de les coder, cette opération sera réalisée en amont du codeur. Plus nécessairement, cet outil doit être capable de détecter les objets puis de les suivre à travers les différentes images de la séquence vidéo. Il existe de nombreuses techniques de segmentation et de suivi d'objets présentes dans la littérature. Un objet vidéo étant caractérisé par sa forme, sa texture et son mouvement, les méthodes de segmentation utilisent ces critères pour le détecter et le suivre temporellement. Tout d'abord après une segmentation (basée régions ou contours) d'une image ou d'un groupe images à partir de critères spatiaux, temporels ou spatio-temporels, le suivi (*tracking*) des objets peut être réalisé en utilisant des approches de mise en correspondance ou de projection/initialisation permettant d'étendre le résultat de la segmentation d'une image aux suivantes. Ces méthodes sont présentées ci-après, elles n'utilisent donc qu'un nombre réduit d'images successives pour réaliser la segmentation qui peut alors s'avérer

inexacte. L'idéal est plutôt de réaliser celle-ci à long terme. Nous introduisons alors la notion de tubes spatio-temporels, qui permettront ce suivi long terme.

3.2.2 État de l'art des techniques de segmentation spatio-temporelle et de suivi d'objets

La segmentation spatio-temporelle d'une vidéo peut être conduite par la combinaison de deux étapes, avec d'abord la segmentation d'une image ou d'un groupe d'images (cas de la segmentation 3D) par une approche basée contour [TPBF87, KWT88, CKS97] ou région [KIK85, MB90, BB92, WA94, PS99], puis le suivi des objets issus de cette segmentation sur les images suivantes. Le suivi permet donc d'étendre sur plusieurs images la segmentation. Bien entendu, le suivi doit être robuste, c'est-à-dire permettre de suivre une région même si elle change de forme, de texture ou bien est occultée. Deux approches sont possibles :

- une par mise en correspondance de deux cartes de segmentation successives. La mise en correspondance de l'image au temps $t-1$ et de celle au temps t se fait en identifiant les régions présentes au temps $t-1$ dans la carte de segmentation au temps t. On utilise pour cela des critères de recouvrement ;
- une autre par projection suivie d'une re-segmentation avec utilisation de la carte projetée comme initialisation ou bien comme référence. Ce suivi revient a recalculer les nouvelles segmentations avec l'*a priori* de connaître approximativement la partition solution. En effet, on utilise la projection de la carte de segmentation du temps $t-1$ vers le temps t comme initialisation ou référence du problème de segmentation de l'image I_t.

3.2.2.1 Suivi par mise en correspondance

De nombreuses méthodes pour suivre les objets de la séquence vidéo sont basées sur la mise en correspondance de régions obtenues après deux segmentations aux temps $t-1$ et t. Pour Marquès et Llach [ML98], la partition obtenue au temps $t-1$ est d'abord re-segmentée en utilisant un critère basé texture afin de vérifier l'homogénéité spatiale des régions. Pour cela, les auteurs utilisent une technique de partage des eaux dans l'espace de couleur. Ensuite le mouvement entre les deux images $t-1$ et t est estimé, et les régions obtenues au temps $t-1$ sont compensées et projetées sur l'image courante t. Pour finir, la méthode associe les régions obtenues au temps $t-1$ aux régions connues au temps t. Cela est réalisé en trois étapes :

- dans un premier temps, seules les régions de t recouvrant entièrement les régions de $t-1$ compensées et projetées au temps t sont appariées ;
- ensuite, les régions de t qui sont recouvertes à plus de 50% par une région de $t-1$ compensée et projetée au temps t sont associées tout en considérant que celles-ci sont voisines et que la distance de couleur entre les deux régions est faible ;
- pour finir, les régions restantes sont affectées à un des objets en utilisant la technique de partage des eaux dans l'espace de couleur en prenant pour germes les régions existantes.

Une autre méthode de suivi d'objets par mise en correspondance a été proposée par Alatan et al. [AOW+98]. Elle considère plusieurs cartes de segmentation initiales pour chaque image. La segmentation initiale d'une image est obtenue à l'aide d'un arbre RSST [2] en utilisant seulement l'information de couleur. Le but étant d'obtenir des régions dont les limites correspondent aux limites des objets réels (sémantiques) présents dans la scène. Afin d'obtenir une prédiction de la segmentation pour l'image t, ils estiment le mouvement entre deux images consécutives et utilisent également l'algorithme d'arbre RSST pour obtenir une segmentation

[2] *Recursive Shortest Spanning Tree*

basée mouvement de l'image $t-1$. Ensuite, ils calculent la différence inter-images. La combinaison des différentes segmentations est alors effectuée pour obtenir une segmentation optimale entre les temps $t-1$ et t. La mise en correspondance des objets aux temps $t-1$ et t , utilise trois règles, chacune pour chaque type d'objets détectés : une pour suivre les objets précédemment détectés, une pour les nouveaux objets commençant à se déplacer et une permettant de distinguer les objets se divisant (mouvements différents).

Les règles sont basées sur l'intersection des régions du temps $t-1$ projetées vers t avec les régions du temps t. L'intérêt de ces règles réside dans le fait que le suivi est évolutif, puisque le nombre d'objets peut varier.

Certains auteurs proposent d'utiliser d'autres critères que le mouvement pour suivre les objets [WDVD00]. Ils estiment d'abord le mouvement global (correspondant typiquement à celui produit par la caméra) et ensuite ils distinguent les objets en mouvement différent de celui du fond. Une fois qu'ils ont obtenu la carte de segmentation pour chaque image de la séquence vidéo, ils calculent un ensemble de quatre variables à partir des positions, des tailles, des niveaux de gris et des textures des objets. Leur méthode de suivi des objets repose sur un ensemble de règles définies en fonctions des valeurs des variables calculées auparavant. Ces règles permettent ainsi de détecter les objets qui commencent à se déplacer ou à l'inverse, qui stoppent leur mouvement.

Bilan :

Ces approches empiriques utilisant des seuils difficiles à régler rencontrent des difficultés qui résultent de la superposition d'une région projetée sur plusieurs régions, de l'apparition de régions lors de découvrements, de la disparition de régions lors de recouvrements, de la sensibilité des segmentations spatiales et, enfin, de la variation du nombre de régions d'une segmentation à l'autre.

3.2.2.2 Suivi par projection et initialisation

Ces approches utilisent la projection de la segmentation obtenue au temps $t-1$ comme initialisation ou référence pour une nouvelle segmentation. En effet, la carte de segmentation projetée par estimation et compensation du mouvement du temps $t-1$ vers t est proche de la segmentation solution au temps t. La solution est donc atteinte plus rapidement et peut être contrainte pour être proche de la segmentation initiale.

Par exemple, Castagno et Sodomaco [CS98] proposent une méthode de segmentation fondée sur des caractéristiques de l'image telles que le mouvement, la luminance, la chrominance et la texture. Leur technique de segmentation utilise un algorithme de « clustering flou » réalisé en deux étapes. Dans la première, une carte de probabilité d'affection *a priori* basée sur les différentes caractéristiques de l'image est calculée. Une première segmentation de l'image est ainsi obtenue. Ils calculent ensuite pour chaque région et chaque caractéristique de l'image une carte de probabilité d'affection *a posteriori*. Ces mesures servent ensuite d'initialisation pour l'algorithme de « clustering flou ».

Une autre méthode [MOK00], repose sur la minimisation de la projection de la région géodésique (forme et intensité) du temps $t-1$ au temps t. L'algorithme proposé utilise les critères :

- la forme et la luminance de la région au temps $t-1$ et au temps t,
- les bords de la région qui doivent correspondre à des zones de gradients forts.

Bilan :

Les approches par projection sont intéressantes car les cartes de segmentation varient peu au cours du temps. Cependant, les zones de découvrement et de recouvrement posent encore problème. Benois-Pineau

et al [BPMBS98, BPN02, WBPB96] tentent de traiter ces zones en calculant la DFD[3] pour obtenir un ordre de profondeur. Mais la segmentation étant réalisée seulement qu'à partir de deux images successives, les informations disponibles ne suffisent pas pour résoudre les problèmes d'appariement de ces zones. La section suivante présente des méthodes de segmentation basées sur plusieurs images.

3.2.2.3 Approches plus long terme

La plupart des méthodes présentées ci-avant ne travaillent successivement que sur des paires d'images pour réaliser la segmentation vidéo. Il est alors très difficile d'obtenir une segmentation précise dans certains cas, du fait des zones de recouvrement et découvrement qui sont difficilement assimilables à une région. Dans le cas d'une segmentation basée mouvement, si les mouvements sont faibles entre deux images successives, on ne peut distinguer les régions.

Il est alors préférable de mener une segmentation à plus long terme, c'est-à-dire, qui ne se limite pas qu'à deux images. En utilisant un contexte temporel de plusieurs images, on améliore la stabilité et la cohérence des résultats. Ces techniques basées moyen terme exploitent l'homogénéité (en texture, mouvement ou couleur...) des régions sur plusieurs images successives. On parlera alors de tubes spatio-temporels (cf figure 3.3).

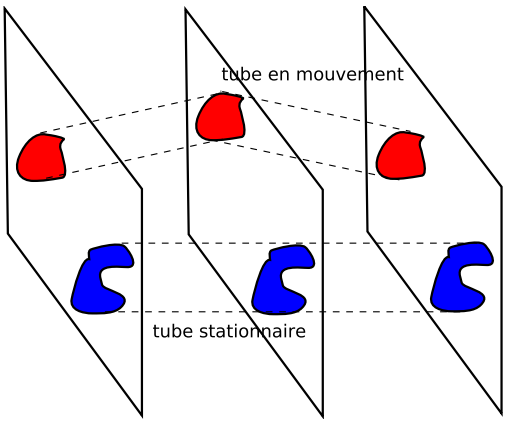


FIG. 3.3 – Illustration de la notion de tube spatio-temporel.

L'analyse d'un groupe d'images doit être faite de manière conjointe spatialement et temporellement. Une revue détaillée de ces approches de segmentation spatio-temporelle long terme avec priorité spatiale, priorité temporelle ou de manière conjointe, a été proposée par Megret et DeMenthon [MD02]. Les techniques de **segmentation avec priorité spatiale** sont les plus répandues. Elles consistent à d'abord segmenter indépendamment chaque image de la vidéo puis à suivre les régions segmentées sur deux images. On ne bénéficie alors pas totalement des avantages du moyen/long terme.

Afin de prendre en considération des informations à plus long terme, une autre catégorie de méthodes a été développée. Les trajectoires des objets en mouvement sont estimées à partir de **techniques de mise en correspondance temporelle** des points ou de suivi de zones texturées. Ensuite, les trajectoires correspondantes au même objet en mouvement sont fusionnées via la segmentation basée mouvement. Ces méthodes peuvent être à leur tour divisées en deux grandes familles selon qu'elle utilisent des similarités de mouvement [AD93, MJ02, Kan02], ou une estimation globale du mouvement [TZ98, BCB99]. Cependant,

[3] *Displaced Frame Difference* (différence inter-images déplacée).

il est nécessaire de connaître le nombre de trajectoires, sinon les frontières spatiales risquent d'être erronées.

La dernière catégorie de **segmentation dite conjointe** traite simultanément les dimensions spatiales et temporelles en considérant des blocs de pixels spatio-temporels. Cette approche est plus en corrélation avec le système visuel humain qui suit conjointement les objets dans l'espace et le temps [GK00].

Porikli et Wang [PW04] proposent un algorithme de segmentation automatique qui exploite les avantages des techniques de segmentation basée couleur, texture, forme et mouvement. Ils essaient d'obtenir des tubes spatio-temporels en utilisant une technique de croissance de régions à partir de marqueurs (germes). D'abord, ils calculent un vecteur de caractéristiques (couleur, différence entre images successives, texture, etc.) pour chaque pixel. Ensuite, les minima locaux des gradients de couleur calculés dans les trois directions (x, y et t) sont choisis comme marqueurs. Un tube se forme alors en agrégeant les sites voisins ayant des caractéristiques spatiales proches (couleur et texture). Une fois les tubes spatio-temporels obtenus, plusieurs critères sont considérés pour fusionner ou non deux tubes voisins entre eux : la similarité de mouvement entre tubes, la compacité avant et après regroupement des tubes, le ratio de la taille des frontières avant et après fusion des tubes, le nombre d'images contenant les deux tubes, enfin la similarité de couleur et de texture.

Cette approche est très intéressante car elle propose d'obtenir des objets vidéo et ceci en traitant non plus des pixels ou des régions mais directement des tubes spatio-temporels. Ceux-ci ayant des propriétés de stabilité et de cohérence importantes, ils permettent d'obtenir des objets vidéo. Cependant la notion d'objet vidéo n'est pas bien définie et les critères de fusion des tubes sont assez empiriques.

Dans ses travaux de thèse, Chaumont [Cha03] désire mieux formaliser la notion d'objet vidéo et propose une technique de **segmentation adaptée**. Pour son modèle, il fait l'hypothèse qu'un objet vidéo est défini par un mouvement propre long terme et une texture stable. Le modèle définit un objet k par son mouvement Θ_k et sa texture mosaïque M_k (voir l'illustration d'une mosaïque sur la figure 3.4). Chacune des images est recomposée de telle sorte qu'un pixel i de l'image I_t à un instant t soit définit par le pixel $\Theta_k^{t\rightarrow t_{ref}}(i)$ de la mosaïque M_k de l'objet k ;

$$\forall t, \forall i, \exists k, I_t(i) = M_k(\Theta_k^{t\rightarrow t_{ref}}(i)) + \eta \tag{3.1}$$

où η correspond à un bruit de modèle (supposé être ici un bruit blanc gaussien). On note $\Theta_k^{t\rightarrow t_{ref}}$ le mouvement associé à un objet k du temps t vers le temps t_{ref}. Un maillage actif [MPL00] permet d'obtenir une description fine du mouvement par objet. Lors de la segmentation qui a pour but de rechercher les différents objets, une phase de mise en concurrence est réalisée afin d'affecter les pixels aux objets les plus probables, l'approche est réalisée en deux étapes :

- Initialisation : localisation grossière des objets (germes) et estimation des mouvements des objets, Θ_k ;
- mise en concurrence des mouvements Θ_k et des textures M_k afin d'obtenir la segmentation finale selon le modèle d'objet de l'équation 3.1.

Contrairement à l'approche de mise en concurrence de mouvements affines proposée par Wang et al. [WA94], Chaumont utilise donc un modèle de mouvement plus complexe, obtenu par un maillage. Le problème de segmentation est abordé comme la minimisation d'une fonctionnelle E qui prend en considération les probabilités d'appartenance des pixels aux objets. Cela nécessite de connaître le nombre d'objets ainsi que le mouvement de chacun d'eux. Les germes ainsi que leurs mouvements sont obtenus via un al-

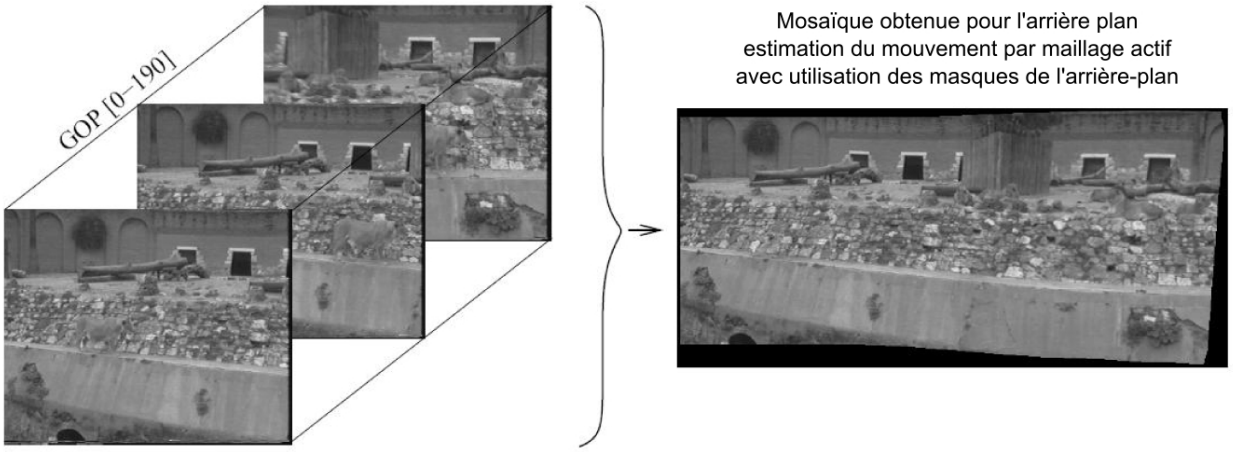


FIG. 3.4 – Mosaïque de la séquence *Lion* pour les images 0 à 190 [Cha03].

gorithme de « clustering flou affine ». Ensuite une méthode de « clustering 3D » résout l'équation d'énergie, permettant d'obtenir les probabilités d'affectation pour chaque pixel. Les résultats obtenus ne sont pas très éloignés des frontières de texture, mais dépendent fortement de la phase d'initialisation. L'utilisation du long terme et l'utilisation d'une classe rejet permettent d'obtenir des résultats intéressants. En effet, pour la séquence *Stefan*, les résultats du long terme permettent plus facilement d'affecter les zones occultées. La classe rejet quant à elle permet de rejeter les régions ne correspondant à aucun modèle objet. Par contre, la phase d'initialisation nécessite d'obtenir le bon nombre de germes ainsi que des germes pas trop éloignés des frontières d'objets.

3.2.2.4 Conclusion

Cette section a décrit les grandes familles de méthodes de segmentation spatio-temporelle et de suivi des objets. Dans un premier temps, nous avons vu celles qui visent à regrouper les régions/objets entre deux images successives en réalisant une mise en correspondance des régions obtenues après deux segmentations initiales, ou en utilisant la projection de la segmentation de l'image précédente vers l'image courante. Cependant cette segmentation n'est réalisée qu'à partir de deux images successives et les informations disponibles peuvent ne pas suffire pour résoudre les problèmes d'appariement pour les zones de découvrement et de recouvrement. Une des évolutions actuelles en segmentation vidéo est de prendre en compte l'aspect temporel à plus long terme. On recherche à segmenter directement un groupe d'images en manipulant des tubes. Cette approche se justifie car elle permet une meilleure gestion des zones de recouvrement/découvrement. Mais, la fusion des tubes pour obtenir les objets vidéos n'est basée que sur des critères empiriques. Finalement nous avons vu un modèle d'objet vidéo basé sur un mouvement long terme et une texture mosaïque par objet, où les probabilités d'appartenance des pixels aux objets sont mis en concurrence afin de les affecter correctement.

Afin de réaliser une segmentation spatio-temporelle satisfaisante, il est donc nécessaire de définir le modèle auquel les objets détectés doivent répondre. De plus, il est important de travailler sur une fenêtre temporelle suffisamment importante pour ne pas rencontrer de problèmes d'appariement dans les zones de recouvrement/découvrement. Dans la suite du chapitre, nous présentons notre méthode de segmentation spatio-temporelle et de suivi des objets sur des segments temporels de neuf images.

3.2.3 Segmentation basée mouvement

Segmenter une séquence d'images en fonction du mouvement des objets qui la composent, nécessite au préalable d'estimer le mouvement. Le mouvement relatif entre les objets de la scène et la caméra induit un mouvement apparent dans la séquence d'images. Il va donc falloir estimer ce mouvement apparent afin de pouvoir segmenter la vidéo par la suite.

3.2.3.1 Mouvement apparent

Il est nécessaire de définir le mouvement apparent et de le distinguer du mouvement projeté. On appelle mouvement projeté le mouvement 2D qui est la projection sur le plan image du mouvement 3D des objets de la scène. Le mouvement projeté[4] n'est en général pas atteignable par la mesure. On n'accède qu'au mouvement apparent qui est le mouvement 2D perçu dans l'image au travers des variations temporelles de l'intensité lumineuse dues aux mouvements 3D relatifs dans la scène. Il est clair que le champ de vitesse apparent n'est généralement pas identique au champ de vitesse projeté. Si mouvements apparent et projeté constituent deux champs différents, ils sont étroitement liés. Il a d'ailleurs été montré que ces deux champs ont les mêmes propriétés qualitatives [VP89]. On peut distinguer trois approches pour extraire le mouvement apparent. La première consiste à chercher à mettre en correspondance entre deux images successives des points ou des primitives particulières (contours, coins...). Dans ce dernier cas, le champ obtenu, n'est pas un champ dense, puisque l'information de mouvement n'est mesurée qu'en certains endroits de l'image. Le deuxième type d'approche repose sur l'hypothèse d'invariance de la luminance d'un point de l'image lors de son déplacement. Les variations d'intensité de l'image sont alors supposées seulement dues au mouvement. Cette hypothèse permet de relier de façon explicite les variations spatio-temporelles de la fonction intensité lumineuse au vecteur vitesse apparent. On parle ici de méthodes différentielles, car elles reposent sur la mesure des dérivées de la fonction intensité. Ces méthodes fournissent généralement un champ dense, à moins que l'estimation ne soit menée que sur certaines zones de l'image (le long des contours par exemple). Le troisième type d'approche regroupe les méthodes par transformées. Leur principe est d'accéder à des informations reliées au mouvement dans un autre espace que l'espace spatio-temporel d'origine (x,y,t). L'information de mouvement apparent est alors mesurée à partir du signal transformé.

Ces trois approches sont décrites succinctement dans la suite.

3.2.3.1.1 Méthodes par mise en correspondance

Elles sont basées soit sur une mesure de similarité entre blocs de deux images successives, soit sur une mise en correspondance de primitives (coins, lignes, etc.). La recherche du déplacement qui apparie au mieux les éléments (régions, blocs, primitives) de l'image entre deux instants successifs s'effectue sur une gamme de valeurs discrètes [Bru01]. Les standards de codage MPEG utilisent un tel procédé par mise en correspondance de blocs (*block matching*).

3.2.3.1.2 Méthodes différentielles

Elles sont donc basées sur une hypothèse d'invariance de l'intensité de la luminance d'un point lors de son déplacement dans l'espace (x,y,t). Horn et Schunck [HS81] formulent plusieurs restrictions afin de respecter cette hypothèse :

[4] On parlera aussi dans la suite du document de mouvement réel.

- la surface qui reflète la lumière est plate afin d'éviter les variations d'intensité de la luminosité dues à des effets d'ombre ;
- l'illumination incidente est uniforme sur la surface ;
- la réflectance varie légèrement et sans discontinuité spatiale ;
- les situations de recouvrements d'objets sont exclues car celles-ci entraînent des discontinuités de la réflectance aux frontières des objets.

Ils proposent une équation qui relie les variations d'intensité de la luminosité d'un point et son vecteur vitesse $\vec{\omega} = (u, v)$. Soit l'intensité de la luminosité au point(x, y) notée $I(x, y, t)$. L'intensité de la luminosité d'un point particulier subissant un déplacement (dx, dy) au cours de l'intervalle de temps dt est constante donc :

$$\frac{dI}{dt} = 0 \text{ soit encore } I(x+dx, y+dy, t+dt) = I(x, y, t)$$

Un développement de Taylor au premier ordre du premier terme de cette équation permet de déduire la relation suivante :

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

On note $u = \frac{dx}{dt}$ et $v = \frac{dy}{dt}$, les deux composantes du vecteur vitesse, on obtient finalement l'équation :

$$I_x u + I_y v + I_t = 0$$

où $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$ et $I_t = \frac{\partial I}{\partial t}$. Cette équation est connue sous le nom d' « équation de contrainte du mouvement apparent » (ECMA). Cette équation peut se réécrire sous la forme : $(I_x, I_y).(u, v) = -I_t$

On ne peut alors calculer que la composante du vecteur vitesse dans la direction du gradient d'intensité lumineuse (I_x, I_y). Pour calculer la composante du vecteur vitesse perpendiculaire à ce gradient d'intensité lumineuse, ils introduisent une contrainte supplémentaire : une contrainte de lissage.

En effet, chaque point d'intensité lumineuse peut bouger indépendamment, il devient alors difficile d'obtenir le champ de vecteurs de vitesse. Dans le cas d'objets de taille finie ayant un mouvement rigide ou subissant des déformations, les points voisins composant ces objets doivent posséder des vecteurs de vitesse similaires, et le champ de vecteurs de vitesse correspondant doit varier faiblement spatialement. Les discontinuités dans ce champ sont alors dues qu'aux occultations entre objets. Pour formuler cette contrainte de lissage entre point voisins, ils minimisent le carré des gradients des vecteurs de vitesse : $\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2$ et $\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2$.

Nous pouvons distinguer deux types d'approches différentielles [Agn01] :

- la première regroupe des méthodes estimant le flot optique en lissant le champ de vecteurs de vitesse, tout en faisant attention aux frontières de mouvement (lieu de discontinuité dans le champ de vecteurs de vitesse ;
- à l'inverse de la première, la deuxième catégorie correspond à l'estimation du mouvement individuellement dans chaque région. Les régions sont délimitées grâce à une détection préalable des discontinuités de mouvement.

3.2.3.1.3 Méthodes par transformation

Elles utilisent la représentation dans un domaine transformé (Fourier, Gabor, Wigner, etc.). Elles exploitent l'information d'énergie en sortie d'un filtre au sens du mouvement [SPH98], ou la variation de la phase de la transformée [FJ90]. D'autres recherches ont permis de développer des filtres spatio-temporels

spécialisés en considérant les changements de l'intensité comme une fonctionnelle de l'espace et du temps [Bru01].

Nous avons vu précédemment, qu'afin de réaliser une segmentation spatio-temporelle satisfaisante, il est important de travailler sur une fenêtre temporelle suffisamment importante pour obtenir des informations précises sur le mouvement des objets et pour ne pas rencontrer de problèmes d'appariement dans les zones de recouvrement/découvrement. Le but de notre segmentation temporelle est d'obtenir des informations sur les objets qui composent la séquence vidéo avant son codage de type MPEG. L'unité de base des standards vidéo MPEG étant le macrobloc, nous avons décidé d'estimer le mouvement par mise en correspondance de blocs de pixels sur plusieurs images successives.

3.2.3.2 Estimation optimisée du mouvement apparent

Nous proposons une méthode d'estimation optimisée du mouvement. En nous basant sur une propriété du système visuel humain, nous posons des hypothèses simplificatrices permettant de contraindre l'estimation du mouvement de façon à générer un champ de vecteur lisse et représentant si possible le mouvement réel (3D relatif) des objets. De plus, afin de réduire les temps de calcul, le mouvement est estimé par une approche multi-résolution.

3.2.3.2.1 Hypothèses pour le calcul des tubes spatio-temporels

Pour obtenir une information de mouvement plus corrélée avec le mouvement 3D relatif des objets de la séquence vidéo, nous utilisons plusieurs images de référence et considérons un mouvement uniforme entre elles. Nous retenons un segment temporel constitué de neuf images (équivalent à 180 *ms* de vidéo pour un taux d'affichage à 50 images par seconde). Le temps de fixation du système visuel humain étant d'environ 200 *ms* [LMLCBT06], nous obtenons ainsi un mouvement uniforme par tube au sein du segment temporel sur une durée perceptuellement significative. L'image courante est située au centre du segment temporel. Ainsi, quatre images passées et quatre images futures entourent celle-ci. Nous utilisons les informations de ces neuf images pour contraindre l'estimation de mouvement et obtenir des vecteurs de mouvement plus lisses. Pour accélérer les calculs, nous retenons cinq images pour évaluer les vecteurs de mouvement, comme ceci est illustré à la figure 3.5. Nous considérons un mouvement uniforme, ainsi apparaît la notion de tube entre les images. Un tube suit et aligne un macrobloc donné sur plusieurs images successives. Le champ de vecteurs de mouvement contraint ainsi obtenu est plus homogène et donc plus corrélé avec le mouvement réel.

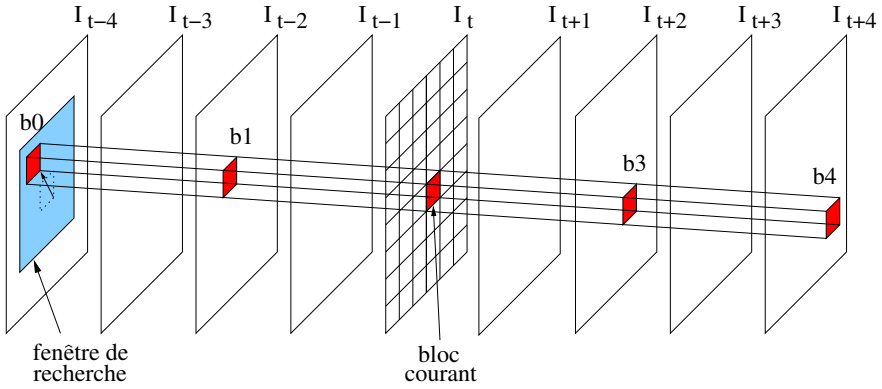


FIG. 3.5 – Tube spatio-temporel.

3.2.3.2.2 Estimation multi-résolution du mouvement

Pour les images HD, nous effectuons une estimation de mouvement multi-résolution plus rapide. Les images HD sont filtrées et sous-échantillonnées spatialement par un facteur six (plus précisément les images sont sous-échantillonnées par un facteur deux puis par un facteur trois). Avant chaque étape de sous-échantillonnage, un filtre passe-bas ad' hoc est appliqué afin d'éviter tout recouvrement de spectres. À partir des images filtrées et sous-échantillonnées, nous réalisons l'estimation du mouvement. Chaque bloc est simultanément comparé aux blocs potentiellement correspondants des images précédentes et futures, comme cela est illustré à la figure 3.5. L'erreur globale, EQM_G, est obtenue par la somme des quatre erreurs quadratiques moyennes (EQM) chacune entre le bloc courant et ses blocs correspondants des images passées et futures (voir équation 3.2).

$$EQMG = \sum_k EQM_k,\ k = -4, -2, +2, +4 \tag{3.2}$$

EQM_k prend en compte les trois composantes YUV de chaque bloc (voir équation 3.3),

$$EQM_k = \begin{cases} \frac{\sum_{i,j=0}^{N-1}(C_Y(i,j)-Rk_Y(i+\lambda_k.m,j+\lambda_k.n))^2}{N\times N} \\ + \quad \frac{\sum_{i,j=0}^{N-1}(C_U(i,j)-Rk_U(i+\lambda_k.m,j+\lambda_k.n))^2}{N\times N} \\ + \quad \frac{\sum_{i,j=0}^{N-1}(C_V(i,j)-Rk_V(i+\lambda_k.m,j+\lambda_k.n))^2}{N\times N} \end{cases} \tag{3.3}$$

avec $\lambda_{-4} = 4$, $\lambda_{-2} = 2$, $\lambda_2 = -2$ et $\lambda_4 = -4$ qui sont les facteurs d'amplitude tenant de l'écart temporel entre l'image courante (image centrale du tube) et les différentes images du tube utilisées comme références. (m,n) est le vecteur de mouvement entre l'image courante I_t et l'image précédente I_{t-1}. C_Y, C_U, C_V, Rk_Y, Rk_U et Rk_V représentent respectivement les trois composantes YUV de l'image courante et celles de l'image utilisée comme référence pour l'estimation de mouvement, avec des blocs de taille $N \times N$ (typiquement 16×16). Le vecteur de mouvement retenu est celui qui minimise EQM_G entre le bloc courant et les blocs correspondants du tube des quatre images. Les vecteurs de mouvement sont estimés à la plus faible résolution, ensuite ils sont multipliés par un facteur approprié à la résolution supérieure afin d'être utilisés comme point de recherche initial pour la recherche du vecteur de mouvement à la résolution donnée.

3.2.3.2.3 Résultats de l'estimation de mouvement

Les figures 3.6 et 3.7 présentent les vecteurs de mouvement obtenus pour la séquence vidéo *Tractor* avec la méthode d'estimation de mouvement classique du codeur H.264, et notre méthode d'estimation du mouvement basée sur des tubes spatio-temporels. Les vecteurs obtenus avec la méthode d'estimation de mouvement classique du codeur H.264 (figure 3.6) sont dispersés et ne reflètent pas correctement le mouvement réel (3D relatif) de la séquence vidéo. Le champ de vecteurs de mouvement obtenu avec notre méthode d'estimation du mouvement basée tubes spatio-temporels (figure 3.7) est plus lisse et corrélé avec le mouvement réel observé de la séquence vidéo.

3.2.3.3 Estimation du mouvement global

Les déplacements perçus dans une séquence d'images peuvent être dus soit aux mouvements des objets de la scène, soit à celui de la caméra. Si cette dernière est en mouvement (déplacement latéral ou dans l'axe de la caméra) ou opère différentes variations (panoramique, zoom ou rotation) cela engendre un mouvement dans la scène appelé mouvement global. Le mouvement apparent des objets se déplaçant est alors la combi-

FIG. 3.6 – Vecteurs de mouvement obtenus pour une image extraite de la séquence vidéo *Tractor* avec l'estimation de mouvement classique du codeur H.264.

naison des mouvements locaux des objets et de celui de la caméra. L'estimation du mouvement des objets (réels) se déplaçant doit s'affranchir du mouvement global. Afin de pouvoir réaliser cette estimation, il est nécessaire d'estimer et de compenser le mouvement global de la scène, ainsi seuls les objets réellement en mouvement subsisteront. L'étape suivante sera d'estimer les mouvements locaux des objets en déplacement, puis de réaliser la segmentation basée sur le mouvement.

Dans la suite de cette section, nous présentons différents modèles utilisés pour décrire le mouvement de la caméra ainsi que les techniques d'estimation les plus répandues. Pour finir nous détaillons notre méthode d'estimation du mouvement global par accumulation.

3.2.3.3.1 Les modèles de mouvement

La caméra peut effectuer des mouvements dans un espace à trois dimensions, alors que les déplacements affichés dans une séquence vidéo n'ont que deux dimensions. De fait, les images acquises correspondent à la projection de la scène réelle dans le plan focal de la caméra. Ici, si la caméra est complètement libre de ses déplacements, plusieurs mouvements ne pourront pas être estimés correctement. Le but de ces modèles est d'estimer au mieux n'importe quel mouvement à trois dimensions (pour un objet rigide) projeté dans un espace à deux dimensions. Il existe des modèles plus ou moins représentatifs de la réalité et dont les plus complets rendent l'estimation assez coûteuse. Nous présentons ici quelques modèles de complexité acceptable.

Modèle quadratique :

Le modèle de mouvement quadratique est obtenu en effectuant un développement linéaire jusqu'au second ordre du déplacement $\overrightarrow{V} = (V_x, V_y)^t$ autour du centre de gravité de la région considérée. Ce modèle est essentiellement utilisé pour des applications de robotique qui ont besoin d'une bonne précision.

FIG. 3.7 – Vecteurs de mouvement obtenus pour une image extraite de la séquence vidéo *Tractor* avec notre méthode d'estimation basée sur des tubes spatio-temporels.

$$\begin{pmatrix} V_x \\ V_y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} . \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a_5 & a_6 \\ a_7 & a_8 \end{pmatrix} . \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} + \begin{pmatrix} a_9 \\ a_{10} \end{pmatrix} .xy \tag{3.4}$$

Modèle affine complet :

Dans la plupart des cas, les applications n'ont pas besoin de la précision du modèle quadratique, et ce limitent à l'utilisation de modèle purement affines (en ignorant donc les termes quadratiques) :

$$\begin{pmatrix} V_x \\ V_y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} . \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.5}$$

ou encore, en posant $T = \begin{pmatrix} t_x \\ t_y \end{pmatrix}$ et $A = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$: $\begin{pmatrix} V_x \\ V_y \end{pmatrix} = T + A. \begin{pmatrix} x \\ y \end{pmatrix}$

Les paramètres a_1, a_2, a_3, et a_4 peuvent être interprétés comme les dérivées partielles du vecteur de déplacement $a_1 = \frac{\partial V_x}{\partial x}$, $a_2 = \frac{\partial V_x}{\partial y}$, $a_3 = \frac{\partial V_y}{\partial x}$ et $a_4 = \frac{\partial V_y}{\partial y}$, et t_x, t_y sont les composantes du vecteur de déplacement du centre de gravité. Ce modèle permet d'estimer généralement les mouvements avec une bonne approximation et est un bon compromis entre représentativité (translation, déformation linéaire) et complexité opératoire.

Modèle affine simplifié :

Lorsqu'encore moins de précision est requise, un modèle affine simplifié peut être suffisant en laissant tomber les termes hyperboliques, dans ce cas $a_1 = a_4$ et $a_2 = -a_3$:

$$\begin{pmatrix} V_x \\ V_y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{pmatrix} div & -rot \\ rot & div \end{pmatrix} . \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.6}$$

Le modèle affine simplifié prend en compte quatre paramètres de mouvement : t_x, t_y pour les translations, *div* correspondant au paramètre de divergence et *rot* lié à l'angle de rotation autour du centre de gravité.

En résumé, la plupart des applications d'estimation du mouvement global sont issues du modèle quadratique, mais n'utilisent pas nécessairement tous les paramètres (quand moins de précision est exigée). Un champ de vecteurs de mouvement par bloc est moins dense et moins précis que des informations de mouvement basées pixel. Dans ce contexte il n'est pas nécessaire d'utiliser un modèle trop complexe, nous utilisons donc un modèle affine à six paramètres pour représenter le mouvement global (voir équation 3.5), où a_i ($i = 1...4$), t_x et t_y sont donc respectivement les paramètres de déformation et de translation. V_x et V_y sont les composantes horizontale et verticale rapportées pour chaque bloc, (x, y) étant la position du bloc.

3.2.3.3.2 Les méthodes d'estimation du mouvement global

Pour estimer le mouvement global, plusieurs approchent sont envisageables : soit calculer un champ de vecteurs de mouvement puis n'utiliser que celui-ci pour estimer les paramètres du mouvement global, soit utiliser directement les données des images pour l'estimation. Dans la première étape de notre méthode de pré-analyse, nous avons estimé le mouvement local de chaque tube spatio-temporel. Afin de ne pas accroître la complexité calculatoire de notre méthode, nous voulons estimer les paramètres du mouvement global à partir de ces vecteurs de mouvement. C'est pourquoi, nous présentons d'abord brièvement quelques méthodes d'estimation du mouvement global par mise en correspondance, et par méthodes différentielles. Ensuite nous décrivons plus longuement les méthodes utilisant un champ de vecteur pour estimer les paramètres du mouvement global.

Méthodes par mise en correspondance

Nous retrouvons les techniques décrites au paragraphe 3.2.3.1.1. Les points d'intérêt (angles) sont premièrement détectés dans chaque image, puis les correspondances entre les points d'intérêt d'une image et de l'image suivante sont établies en calculant les corrélations croisées, enfin les paramètres de la caméra sont estimés à partir des vecteurs de déplacement des points d'intérêt entre les deux images [BG99].

Méthodes différentielles

Odobez et Bouthemy [OB95] ont proposé une méthode d'estimation robuste qui analyse le mouvement en identifiant des modèles paramétriques 2D à partir du flot optique. Ils utilisent en particulier des modèles polynomiaux du point de coordonnées (x, y) dans le plan de l'image. Ces modèles incluent le modèle constant, celui affine et celui quadratique. Wang, Doherty et Van Dyck [WDVD00] utilisent quant à eux un modèle bi-linéaire et l'estimation est basée sur le flot optique 2-D.

Méthodes d'estimation du mouvement global à partir d'un champ de vecteurs :

Pour ces méthodes, les paramètres du mouvement sont estimés de manière statistique à partir d'informations locales. Les vecteurs de mouvement issus de la mise en correspondance des blocs (*block matching*) sont plus ou moins fiables vis à vis du mouvement réel au sein de la séquence. En effet, les méthodes de *block matching* exploitent la contrainte de l'invariance intrinsèque de la luminance d'un bloc au cours du temps (cf. ECMA). Pour un bloc situé dans une zone uniforme, le vecteur de mouvement, qui minimise la variation de la luminance, ne reflète pas le mouvement réel du bloc. Pour obtenir un modèle robuste d'estimation du mouvement global à partir des vecteurs de mouvement, c'est-à-dire, qui soit en mesure d'identifier et de

rejeter les vecteurs de mouvement issus des zones uniformes et susceptibles de ne pas représenter correctement le mouvement réel, ces considérations doivent être prises en compte afin de ne pas fausser l'estimation des paramètres du mouvement global.

Le mouvement global de translation peut être estimé en minimisant l'erreur entre les deux images (l'image courante et l'image précédente compensée). Ensuite, à partir de ce vecteur de translation globale, le vecteur de translation pour chaque bloc est calculé. Les paramètres du modèle quadratique sont calculés à l'aide d'un M-estimateur (maximum de vraisemblance) à partir des vecteurs de translation de chaque bloc qui sont pondérés par une mesure de confiance. Celle-ci est calculée en fonction de la texture du bloc et de l'erreur entre le bloc de référence et le bloc compensé [GT02].

Heuer et Kaup [HK99] proposent un algorithme d'estimation du mouvement global, où ils utilisent la SAD (calculé lors de la mise en correspondance des blocs) pour calculer la fiabilité du vecteur de mouvement. Mais comme mentionné auparavant, la valeur de la SAD obtenue pour le meilleur bloc dépend de la variation de la luminance. Ils se basent sur la variance de la SAD, leur permettant ainsi d'éliminer les vecteurs de mouvement des petits objets et les mauvaises mises en correspondances de blocs. Les régions ne possédant pas un mouvement homogène sont rejetées. L'homogénéité du mouvement se calcule grâce à la matrice de Jacobi qui doit être constante :

$$\begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix} = C \tag{3.7}$$

Un histogramme des composantes des gradients des vecteurs de mouvement est utilisé pour calculer la matrice de Jacobi la plus probable. Ils comparent ensuite les paramètres résultants du modèle de mouvement global (t_x et t_y) entre les différentes régions pour les fusionner ou non. Les régions de mouvement homogène contenant le plus grand nombre de blocs sont alors considérées comme le fond et représentent le masque de segmentation pour la régression linéaire.

Disposant de vecteurs de mouvement (un vecteur par tube spatio-temporel), les méthodes d'estimation du mouvement à partir d'un champ de vecteurs sont les plus adaptées à notre situation. Il faut aussi être robuste face au risque de mouvements locaux présents dans les séquences vidéo et être capable d'écarter les données erronées (vecteurs de mouvements issus des zones uniformes). Nous proposons donc une méthode robuste d'estimation du mouvement global par accumulation.

3.2.3.3.3 Estimation du mouvement global par accumulation

Précédemment nous avons réalisé l'estimation du mouvement, les tubes obtenus reflètent plus fidèlement le mouvement local 2D des objets (figures 3.6 et 3.7). La prochaine étape est d'estimer à partir du champ de vecteurs de mouvement obtenu (un vecteur par tube) les paramètres du modèle affine choisi pour représenter le mouvement global du plan vidéo.

Pour estimer les paramètres du modèle affine, nous adaptons la méthode d'accumulation des vecteurs de mouvement décrite par Coudray [Cou05]. Dans ses travaux, Coudray extrait les vecteurs de mouvements du GOP d'une séquence MPEG et estime les paramètres du modèle affine. Cette méthode repose sur le cumul de dérivées orientées (gradients) des vecteurs de mouvement. Ce cumul est réalisé au sein d'histogrammes qui permettent ainsi d'extraire les paramètres relatifs au déplacement majoritaire. Le mode majoritaire de chaque histogramme représente la valeur du paramètre global. Les vecteurs de mouvement sont d'abord compensés en utilisant les quatre paramètres de déformation, la dernière étape est l'accumulation des vec-

teurs de mouvement ainsi compensés dans un dernier histogramme à deux dimensions. Les paramètres de translation du modèle sont alors identifiés à partir de celui-ci (les calculs seront détaillés ultérieurement dans cette section).

Indices de confiance pour une estimation robuste des paramètres

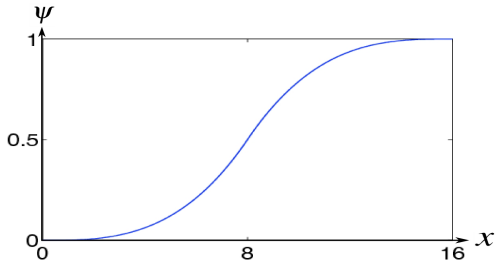
Nous avons vu que pour un macrobloc donné, si celui-ci est situé au sein d'une zone uniforme des images, l'appareillement de blocs n'est pas fiable. Ainsi, le vecteur de mouvement ne reflète pas le mouvement réel du bloc. Pour être robuste, de tels vecteurs de mouvement ne doivent pas contribuer à l'estimation du mouvement global de la vidéo. Nous proposons de pondérer la contribution des vecteurs de mouvement en fonction du contenu spatial du macrobloc situé au centre du tube. Les vecteurs de mouvement de macroblocs relatifs à des zones orientées donnent eux des informations plus fiables sur le mouvement apparent 2D que ceux associés à des zones uniformes. D'où l'idée d'utiliser l'activité spatiale du tube pour le qualifier. Pour calculer l'activité spatiale des macroblocs, nous utilisons les gradients spatiaux. Plus le gradient spatial d'un macrobloc est élevé, plus nous pouvons donner du poids (de confiance) au vecteur de mouvement de ce macrobloc. Nous utilisons les deux gradients spatiaux moyens, $\overline{\Delta V}$ et $\overline{\Delta H}$, qui sont respectivement le gradient vertical moyen et le gradient horizontal moyen. À partir du calcul de ces gradients, un macrobloc peut-être identifié comme étant d'une zone uniforme, d'une zone moyennement texturée ou d'une zone fortement texturée. Un macrobloc identifié comme étant d'une zone fortement texturée, peut l'être seulement dans une seule direction, c'est-à-dire, que l'un des gradients est élevé et l'autre faible. Si le mouvement global est une translation dans la même direction (verticale ou horizontale) que la zone texturée, les vecteurs de mouvement des macroblocs situés dans cette région ne sont pas fiables. C'est pourquoi nous distinguons les deux gradients spatiaux. En pratique, la confiance accordée aux deux composantes des vecteurs de mouvement est calculée de façon appropriée en fonction du gradient spatial $\psi(\overline{\Delta H})$ et $\psi(\overline{\Delta V})$. La fonction ψ pour obtenir les indices de confiance fonction des gradients spatiaux que nous avons utilisé est la suivante :

$$\psi(x) = \begin{cases} (\frac{x}{8})^3/2, & x \leq 8 \\ 1 - \left((\frac{16-x}{8})^3/2\right), & 8 < x < 16 \\ 1 & \text{sinon} \end{cases} \tag{3.8}$$

La figure 3.8 donne les indices de confiance obtenus pour une image de la séquence vidéo HD *Knightshields*. Les zones très texturées et correctement orientées sont blanches (indice de confiance proche de 1) tandis que les zones homogènes sont sombres (indice de confiance proche de 0). Comme on peut l'observer dans la figure 3.8, la bande verticale du mur apparaissant entre les blasons présente un fort contraste générant des contours verticaux. Les indices étant calculés pour les deux directions (horizontale et verticale), on constate que seules les valeurs des indices obtenues pour les gradients horizontaux sont élevées pour ces zones de contours. En effet, ceux ci étant verticaux, les valeurs des gradients verticaux sont faibles voire nulles. Les macroblocs situés sur ces contours auront donc un indice de confiance élevé dans la direction horizontale et faible dans la direction verticale. Ainsi les valeurs estimées des paramètres du mouvement global obtenues à partir des vecteurs de mouvement issus de ces zones peu texturées ou orientées dans une seule direction (indices de confiance faibles) ne risqueront pas de fausser l'estimation des paramètres du mouvement global.

Estimation robuste des paramètres du mouvement global

L'information de base utilisée pour estimer le mouvement global est un champ de vecteurs de mouvement



(a) Fonction de confiance.

(b) Image de la séquence vidéo *Knightshields*.

(c) Indices pour les gradients horizontaux (blanc = 1, noir = 0).

(d) Indices pour les gradients verticaux (blanc = 1, noir = 0).

FIG. 3.8 – Indices de confiance des vecteurs de mouvement.

(un vecteur par tube). La dérivée directionnelle $\nabla_u f(x_0, y_0)$ est la vitesse à laquelle la fonction $f(x,y)$ change pour le point $X_0 = (x_0, y_0)$ dans la direction u. L'estimation des paramètres du mouvement global est donc obtenue par le calcul des dérivées spatiales des vecteurs de mouvement :

$$\text{en discret} \quad \nabla_u f(x_0, y_0) = \frac{f(X_0+u)-f(X_0)}{u} \tag{3.9}$$

On note ∇_x la dérivée spatiale dans la direction horizontale, c'est-à-dire, $u = (1,0)$. Et on note ∇_y la dérivée spatiale dans la direction verticale, c'est-à-dire, $u = (0,1)$. Les composantes horizontale et verticale du vecteur de mouvement à la position (x,y) sont notées respectivement $V_x(x,y)$ et $V_y(x,y)$. Si tous les vecteurs suivent le même mouvement, chaque dérivée est égale à l'un des paramètres de la matrice de déformation A :

$$\begin{aligned}
\nabla_x V(x,y) &= V(x+1,y) - V(x,y) \\
&= A \cdot \begin{pmatrix} x+1 \\ y \end{pmatrix} + T - A \cdot \begin{pmatrix} x \\ y \end{pmatrix} - T \\
&= \begin{pmatrix} a_1(x+1)+a_2 y \\ a_3(x+1)+a_4 y \end{pmatrix} - \begin{pmatrix} a_1 x + a_2 y \\ a_3 x + a_4 y \end{pmatrix} \\
&= \begin{pmatrix} a_1 \\ a_3 \end{pmatrix}
\end{aligned} \tag{3.10}$$

$$\begin{aligned}
\nabla_y V(x,y) &= V(x,y+1) - V(x,y) \\
&= A \cdot \begin{pmatrix} x \\ y+1 \end{pmatrix} + T - A \cdot \begin{pmatrix} x \\ y \end{pmatrix} - T \\
&= \begin{pmatrix} a_1 x + a_2 (y+1) \\ a_3 x + a_4 (y+1) \end{pmatrix} - \begin{pmatrix} a_1 x + a_2 y \\ a_3 x + a_4 y \end{pmatrix} \\
&= \begin{pmatrix} a_2 \\ a_4 \end{pmatrix}
\end{aligned} \tag{3.11}$$

Ce qui donne en résumé :

$$A = \begin{pmatrix} a_1 = \nabla_x V x & a_2 = \nabla_y V x \\ a_3 = \nabla_x V y & a_4 = \nabla_y V y \end{pmatrix} \tag{3.12}$$

Les paramètres de translation sont estimés à partir du champ de vecteurs de mouvement de la manière suivante :

$$\begin{cases} t_x &= V_x - a_1 x - a_2 y \\ t_y &= V_y - a_3 x - a_4 y \end{cases} \tag{3.13}$$

Nous avons vu que pour un modèle affine du mouvement global (équation 3.5), les déplacements peuvent être des combinaisons de trois déplacements élémentaires de natures différentes : zoom (a_1, a_4), rotation (a_2, a_3) ou translation (t_x, t_y). L'équation 3.13 indique que les paramètres de déformation a_1, a_2, a_3 et a_4 affectent les valeurs des paramètres de translation. L'estimation du mouvement global est donc réalisée en deux étapes. Dans un premier temps, nous estimons les paramètres relatifs au mouvement global de déformation. Chaque tube fournit une information locale sur ces quatre paramètres de déformation (équation 3.12), chaque dérivée calculée sur un vecteur mouvement va donc être une valeur possible (hypothétique) pour un des paramètres de déformation.

Accumulation au sein d'histogrammes :

Pour extraire la valeur qui représente le mieux le paramètre de mouvement global, nous procédons par accumulation au sein d'histogrammes (un histogramme par paramètre de déformation). Dans chaque histogramme, pour fusionner les valeurs proches de dérivées, nous procédons exactement an associant à chaque valeur dérivée x la gaussienne $G(x)$:

$$G(x) = \sqrt{\frac{1}{2\pi\sigma^2}} e^{-x^2/2\sigma^2} \tag{3.14}$$

Dans la pratique, l'écart type de la fonction est fixé à trois ($\sigma = 3$).

Considérons le macrobloc central d'un tube donné (n,m) dont l'activité spatiale est caractérisée par les gradients moyens $\overline{\Delta V}(n,m)$ et $\overline{\Delta H}(n,m)$. Ce macrobloc produit respectivement quatre valeurs (dérivées) $a_1(n,m)$, $a_2(n,m)$, $a_3(n,m)$ et $a_4(n,m)$ qui sont respectivement accumulées dans les histogrammes h_1, h_2, h_3 et h_4 selon les relations suivantes :

$$\begin{cases} h_i(x) \leftarrow h_i(x) + \psi(\overline{\Delta H}(n+1,m)).\psi(\overline{\Delta H}(n,m)).\sqrt{\frac{1}{2\pi\sigma^2}}e^{-(x-a_i(n,m))^2/2\sigma^2}, \ i=1,2 \\ h_j(x) \leftarrow h_j(x) + \psi(\overline{\Delta V}(n+1,m)).\psi(\overline{\Delta V}(n,m)).\sqrt{\frac{1}{2\pi\sigma^2}}e^{-(x-a_j(n,m))^2/2\sigma^2}, \ j=3,4 \end{cases} \tag{3.15}$$

Localisation du mode :

Dans chaque histogramme, la localisation du mode principal donne la valeur retenue pour le paramètre global de déformation considéré. Pour affiner la localisation de ce mode, nous estimons la courbure autour de la position du maximum à l'aide de la méthode des Moindres Carrés (figure 3.9) : neuf valeurs autour de la position du mode, notée m sont utilisées pour l'estimation de la courbure de l'équation 3.16, où y représente la valeur de l'histogramme à la position x. Ces neuf valeurs forment le vecteur h et les indices des positions autour du mode m sont renseignés par le vecteur q (équation 3.17). Les paramètres de l'équation du second ordre sont estimés par le calcul de l'équation 3.19, où X est obtenue à partir de l'équation 3.18.

$$y = a + bx + cx^2 \tag{3.16}$$

$$q = (\ -4 \quad -3 \quad -2 \quad -1 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4\ )^t \tag{3.17}$$

$$X = [\ 1 \quad q \quad q^2\ ] \tag{3.18}$$

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = (X^t X)^{-1} X^t h^t \tag{3.19}$$

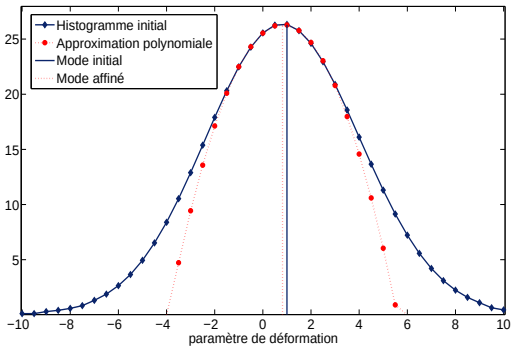
Une fois l'équation de la courbure connue, la position du maximum, notée p, se situe à la position où la dérivée $\frac{\partial y}{\partial x} = b + 2cx$ est nulle ($0 = b + 2cp$). Une fois la valeur de p obtenue ($p = \frac{-b}{2c}$), la valeur du paramètre du mouvement global devient $m + p$.

Compensation des vecteurs de mouvement et estimation des paramètres de translation :

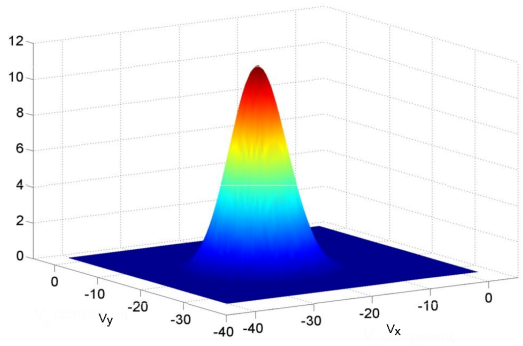
Les quatre paramètres de déformation calculés vont permettre de compenser le champ de vecteurs initial des mouvements de déformation (équation 3.13). Les vecteurs ainsi compensés représentent théoriquement les seuls mouvements de translation de la caméra et les déplacements locaux des objets. En conservant le même principe que pour l'estimation des paramètres de déformation, nous voulons extraire la translation (la plus redondante) du mouvement global. Les vecteurs compensés sont alors accumulés dans un histogramme à deux entrées. L'accumulation de chaque vecteur compensé est faite proportionnellement au minimum des indices de fiabilité de ses composantes, c'est-à-dire, que les deux composantes du vecteur sont pondérées par l'indice de confiance minimal obtenu pour le calcul des deux gradients spatiaux (horizontal et vertical). Une distribution gaussienne à deux dimensions est utilisée pour l'accumulation des données, toujours afin de regrouper les valeurs proches. Les valeurs des paramètres de translation globale sont alors données par la position du maximum dans l'espace d'accumulation (figure 3.9).

Résultats expérimentaux de l'estimation des paramètres du mouvement global

La figure 3.10 montre les vecteurs de mouvement après les différentes étapes de l'estimation du mouve-



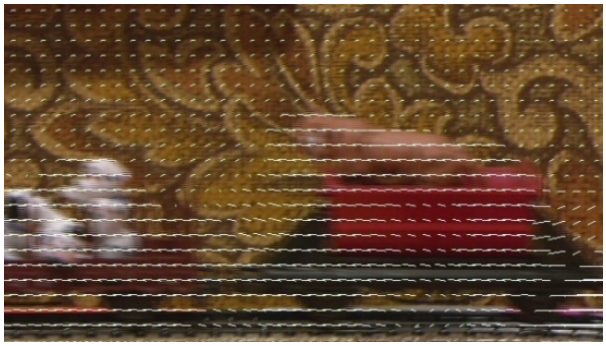
(a) Histogramme d'accumulation pour estimer un paramètre de déformation.



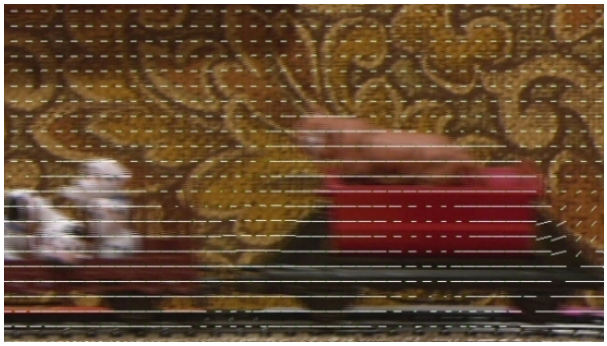
(b) Histogramme d'accumulation pour estimer un paramètre de translation.

FIG. 3.9 – Espaces d'accumulation pour l'estimation des paramètres du mouvement global (pour un segment temporel extrait de la séquence *Knightshields*).

ment global pour une image de la séquence *New mobile and Calendar*. La première image (a) illustre les vecteurs de mouvement bruts obtenus par notre méthode d'estimation du mouvement multi-résolutions. Les deux dernières images (b) et (c) contiennent les vecteurs de mouvement compensés après les deux étapes de l'estimation du mouvement global. Nous pouvons voir que le mouvement global est correctement compensé puisque les vecteurs de mouvement du fond sont presque tous nuls, seuls les vecteurs de mouvement du train se déplaçant subsistent et reflètent son mouvement de translation horizontale.



(a) Sans l'estimation du mouvement global.



(b) Zoom et rotation compensés.

(c) Mouvement global compensé.

FIG. 3.10 – Vecteurs de mouvement de la séquence vidéo *New mobile and Calendar*.

Afin d'évaluer la qualité des résultats numériques de notre méthode, nous l'avons comparée avec les résultats obtenus avec le logiciel robuste Motion2D [IRI]. Ce logiciel a été développé au laboratoire IRISA de Rennes et utilise les résultats exposés par Odobez et Bouthemy [OB95]. Motion2D procède à partir d'un

champ de vecteurs de mouvement dense et précis, et opère une estimation robuste des paramètres du mouvement global (c'est actuellement la méthode de référence pour l'estimation du mouvement global). Comme Motion2D calcule les paramètres du mouvement global entre deux images consécutives, nous combinons les paramètres obtenus pour neuf images consécutives pour les comparer à ceux obtenus avec notre estimateur (qui travaille avec un segment temporel de neuf images). Les figures 3.11 et 3.12 illustrent les paramètres de déformation pour deux vidéos synthétiques contenant seulement une rotation ou un zoom, donc nous disposons des paramètres effectifs globaux utilisés. Nous pouvons voir que notre méthode d'estimation du mouvement global se comporte correctement, mais les résultats obtenus avec Motion2D sont plus proches des paramètres de déformation effectifs utilisés pour créer les séquences vidéo. En effet, le logiciel Motion2D estime les paramètres du modèle de mouvement global pour une restriction temporelle et spatiale plus fine (estimation du mouvement basée pixel entre deux images successives) et ainsi, ceux-ci sont plus précis.

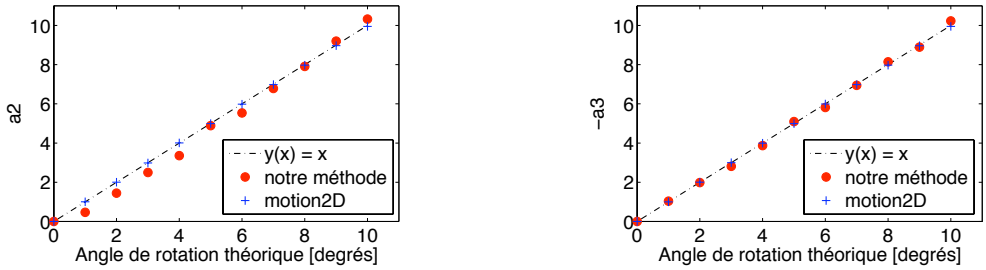


FIG. 3.11 – Paramètres globaux de déformation pour une rotation synthétique.

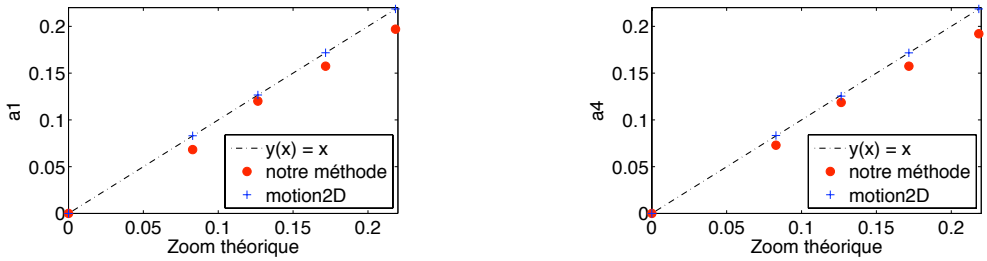


FIG. 3.12 – Paramètres globaux de déformation pour un zoom synthétique.

Résultats expérimentaux de l'estimation des paramètres du mouvement global sur des séquences réelles

Nous avons utilisé trois séquences HD 1080p (*Blue sky*, *Station*, *Tractor*) et deux séquences HD 720p (*Knightshields* et *New mobile and Calendar*) du SVT « corporate development technology » [SVT02]. Ces vidéos présentent différents mouvements de caméra :

- *Blue sky* : rotation et translation,
- *Station* : zoom arrière et légères translations,
- *Tractor* : translation horizontale, zooms avant et arrière,
- *Shields* : translation horizontale et zoom avant,
- *New mobile and Calendar* : translations verticale et horizontale et zoom arrière.

Dans les graphiques suivants (figures 3.13, 3.14, 3.15, 3.16 et 3.17), nous comparons les résultats obtenus sur séquences réelles avec notre méthode (•) et Motion2D (+). Sur l'axe des abscisses est représenté le numéro du segment temporel (à un indice de segment temporel correspond un groupe de neuf images suc-

cessives). Pour les séquences vidéo *Blue sky* et *Station* qui ne contiennent respectivement qu'une rotation et un zoom arrière, les paramètres sont estimés correctement par notre méthode et Motion2D. Les résultats obtenus pour le paramètre de translation verticale du mouvement global, t_y, pour la séquence *Station* semblent erronés pour l'une des méthodes, car on constate une différence de mouvement de près de 20 pixels entre les deux méthodes pour le deuxième segment temporel. L'estimation du mouvement global étant réalisée pour des segments temporels de neuf images, cela ne représente qu'une erreur de deux pixels entre chaque image pour les deux méthodes. Pour les trois autres séquences vidéos (*Tractor*, *Knightshields* et *New Mobile and Calendar*), les paramètres estimés du mouvement global sont très proches pour les deux méthodes. Cependant, Motion2D paraît être sensible aux mouvements locaux. En effet, les objets en mouvement au sein des séquences vidéos portent préjudice à l'estimation des paramètres du mouvement global pour cette méthode. Cette influence peut être observée dans les figures 3.15 et 3.17. Tout d'abord, pour la séquence *Tractor* (figure 3.15), lorsque le tracteur tourne légèrement au début de la séquence (segments temporels 1 à 8), Motion2D détecte à tort un zoom et une légère rotation, ce qui n'est pas le cas de notre méthode. Ce phénomène peut également être observé pour la séquence *New Mobile and Calendar* (figure 3.17) à partir du segment temporel 31. Les mouvements de translation du calendrier et du train sont estimés comme la conséquence d'une rotation globale de la scène. Cette estimation à tort d'une rotation a une incidence sur l'estimation des paramètres de translation qui sont du coup incorrectement estimés. Ce résultat plus robuste est du à notre estimation du mouvement initiale (estimation du mouvement multi-résolutions) plus cohérente. Dans la séquence vidéo *Knightshields*, bien qu'un homme soit présent dans la scène et se déplace, les paramètres sont correctement estimés par les deux méthodes.

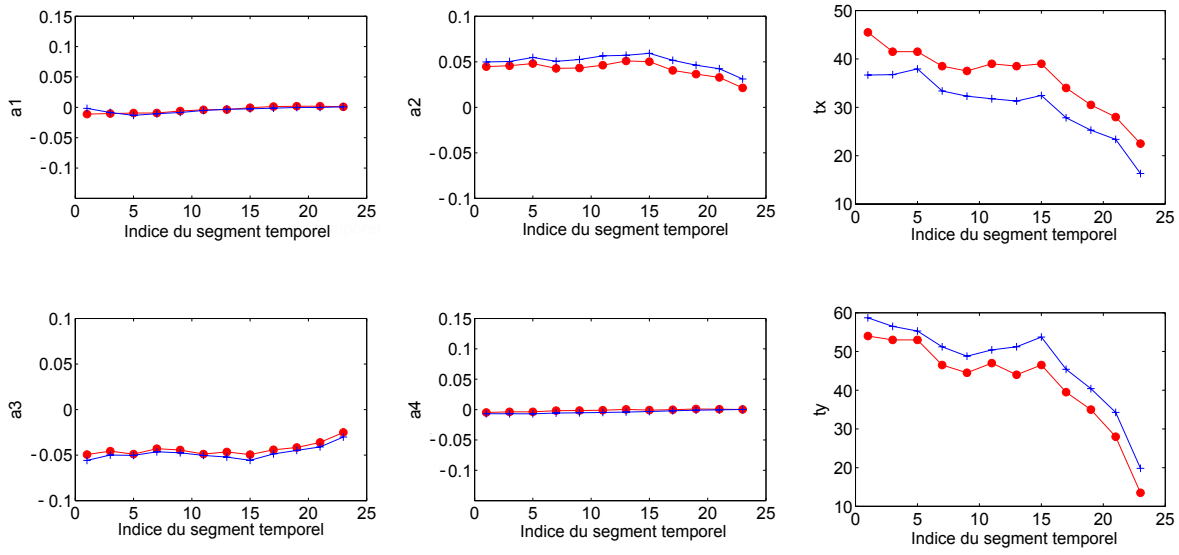


FIG. 3.13 – Paramètres estimés du mouvement global pour la séquence *Blue sky* (Motion2D (+), notre méthode (●)).

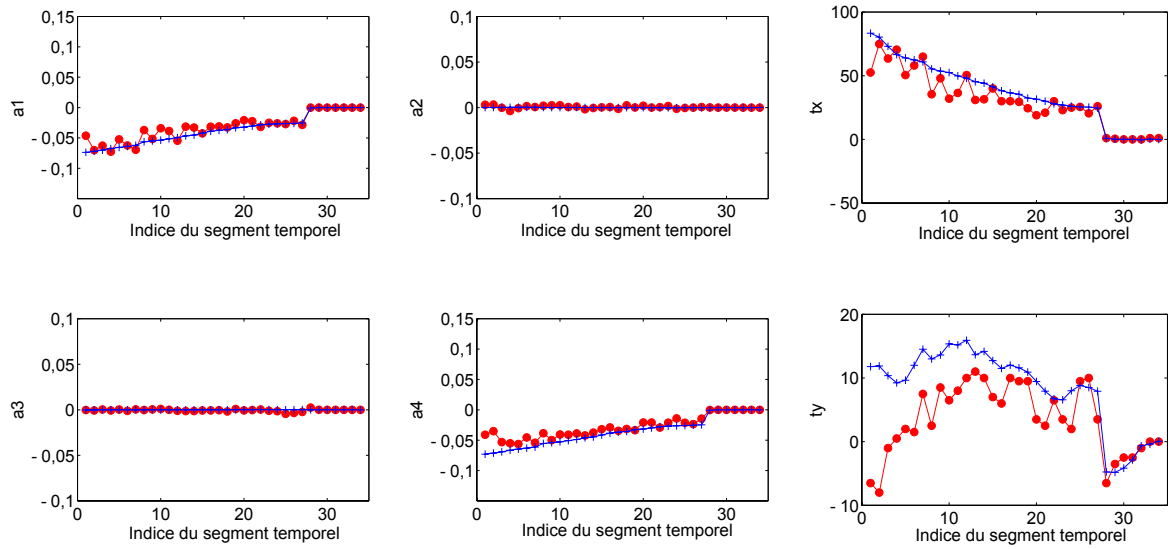


FIG. 3.14 – Paramètres estimés du mouvement global pour la séquence *Station* (Motion2D (+), notre méthode (•)).

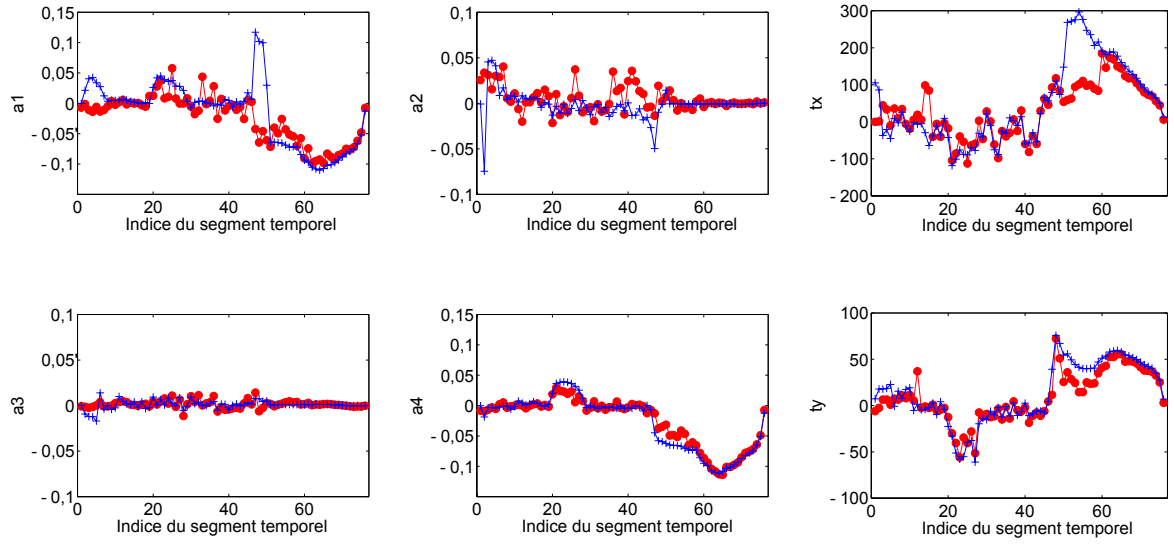


FIG. 3.15 – Paramètres estimés du mouvement global pour la séquence *Tractor* (Motion2D (+), notre méthode (•)).

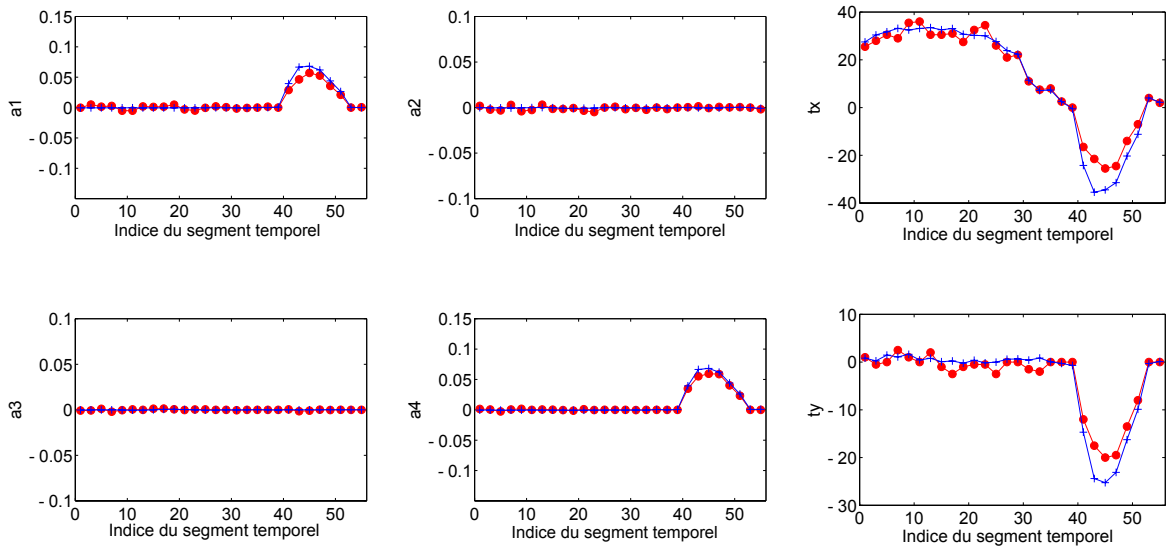


FIG. 3.16 – Paramètres estimés du mouvement global pour la séquence *Knightshields* (Motion2D (+), notre méthode (•)).

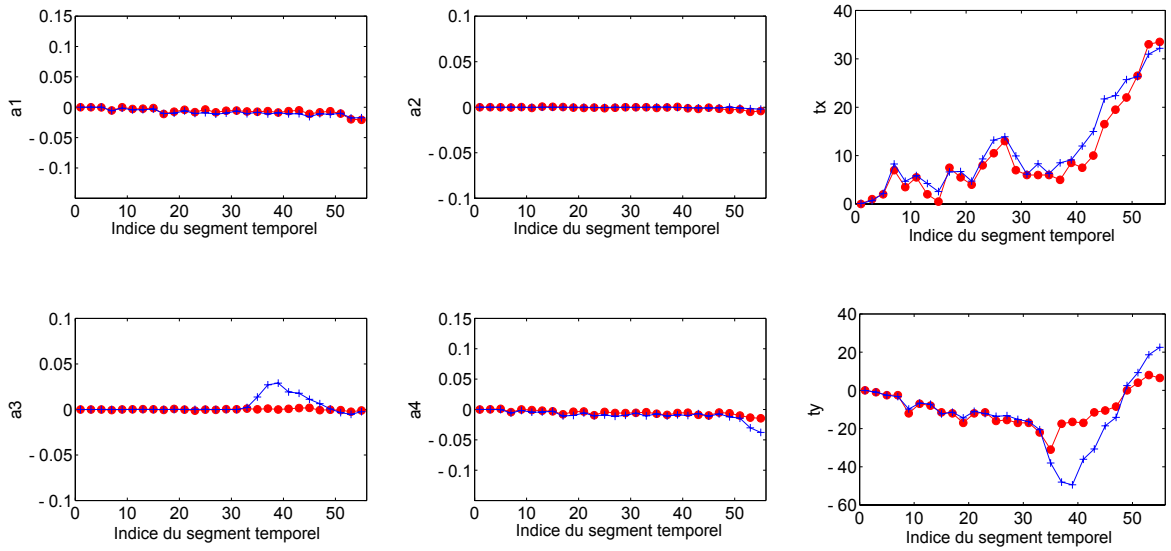


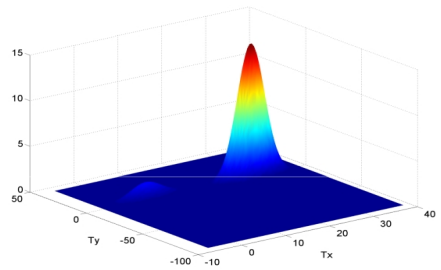
FIG. 3.17 – Paramètres estimés du mouvement global pour la séquence *New Mobile and Calendar* (Motion2D (+), notre méthode (•)).

3.2.3.4 Segmentation spatio-temporelle

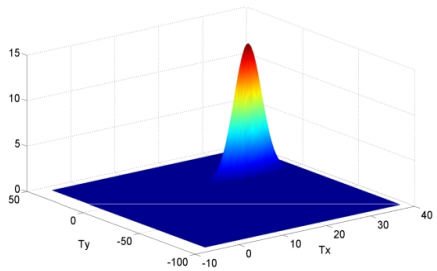
Lors de l'estimation du mouvement global, nous avons déterminé les paramètres de translation en localisant le maximum de l'histogramme d'accumulation des vecteurs compensés par les paramètres de déformation. Si nous n'étudions plus uniquement que le mode le plus important mais tous, alors une segmentation au sens du mouvement, en plus de l'estimation du mouvement global, aura été effectuée avec l'hypothèse que chaque mode représente le mouvement d'un objet.

3.2.3.4.1 Segmentation de l'espace d'accumulation

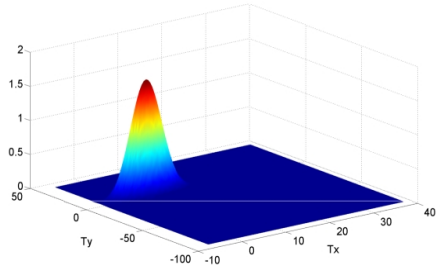
La première étape consiste à éliminer le bruit. Un seuil de rejet est défini empiriquement et toutes les cellules de l'histogramme représentant une accumulation inférieure à ce seuil sont mises à zéro. Afin de ne pas utiliser une méthode de segmentation trop coûteuse en termes de complexité de calcul, nous utilisons un algorithme récursif qui va traiter les modes par ordre décroissant. Le premier mode détecté est donc celui qui correspond au maximum global de l'espace d'accumulation. Pour toutes les positions connexes à ce mode, le gradient[5] en direction du maximum est calculé. Tant que le gradient est positif, la position testée est considérée comme appartenant au mode et l'algorithme est répété pour les cellules connexes. Pour le calcul du gradient d'un point, la différence entre sa valeur et la valeur du point connexe qui est dans la direction de la position du maximum, est prise en compte. Á la fin, toutes les positions appartenant au mode principal ont été marquées. Un nouveau maximum est détecté parmi toutes les cellules non marquées et l'algorithme est réitéré tant qu'il reste des cellules non nulles n'appartenant à aucun mode. Au final, une cellule peut être marquée comme appartenant à plusieurs modes. Dans ce cas, elle est définitivement rattachée au mode dont la position du maximum est la plus proche. La figure 3.18 présente la séparation entre modes de l'espace d'accumulation pour un segment temporel extrait de la séquence *Knightshields*. Ce segment est extrait lors de la phase de *traveling* qui a lieu au début de la séquence. Deux modes sont détectés, le mode majoritaire représente le mouvement global du fond, et le second représente le mouvement local du personnage.



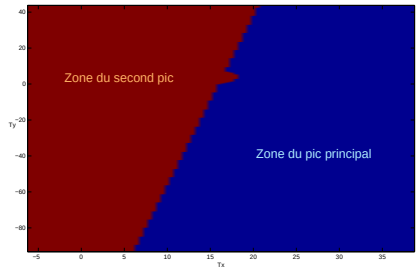
(a) Espace d'accumulation.



(b) Mode principal (mouvement global).



(c) Second mode (mouvement local).



(d) Espace d'accumulation segmenté.

FIG. 3.18 – Analyse récursive de l'espace d'accumulation.

La dernière étape consiste à segmenter le champ de vecteurs compensés par les paramètres de déformation, à partir de la séparation des différents modes. L'espace d'accumulation segmenté devient un tableau à deux entrées : les deux composantes de chaque vecteur déplacement compensé de l'image. Le contenu du tableau correspond alors à l'étiquette donnée au macrobloc associé au vecteur.

[5] Le gradient est ici une différence entre les populations de deux cellules de l'espace d'accumulation.

FIG. 3.19 – Image segmentée de la séquence *Knightshields*.

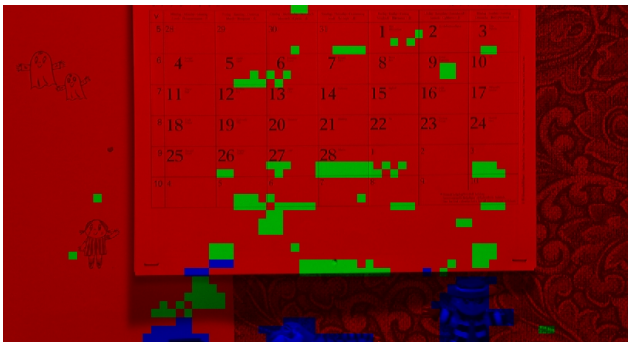


FIG. 3.20 – Image segmentée de la séquence *New Mobile and Calendar*.

3.2.3.4.2 Résultats expérimentaux de la segmentation basée mouvement

Les résultats de la segmentation basée mouvement après notre méthode d'estimation du mouvement global pour trois segments temporels des séquences vidéo *Tractor* et *New Mobile and Calendar* sont présentés en annexe B.2. À partir des deux modes présentés en figure 3.18, nous créons l'image segmentée (issue de la séquence *Knightshields)* présentée en figure 3.19. La zone rouge correspond aux macroblocs dont les vecteurs déplacement appartiennent au mode principal (mouvement global), tandis que la zone bleue correspond aux macroblocs dont les vecteurs appartiennent au second mode (mouvement local). La segmentation au sens du mouvement réalisée ici donne des résultats encourageants, cohérents avec la segmentation qu'effectuerait un humain.

Cependant, une telle qualité de segmentation, avec des critères basés sur le mouvement uniquement, ne peut être obtenue que pour des segments temporels au contenu relativement peu complexe. En effet, le segment temporel utilisé en exemple ici est assez simple : la caméra n'engendre aucune déformation de zoom ou de rotation, et le contenu spatial de la scène est suffisamment texturé pour que les vecteurs déplacement calculés soient représentatifs des mouvements des objets. Inversement, la séquence *New Mobile & Calendar* offre un contenu complexe. La caméra effectue un mouvement de *zoom out* sur une tapisserie et un calendrier uniformes. La segmentation au sens du mouvement est donc moins probante que celle réalisée précédemment pour la séquence *Knightshields* (figure 3.20). De fait, sur les zones uniformes du calendrier et de la tapisserie, nous observons des régions vertes déconnectées. Ces régions qui devraient théoriquement être englobées dans la zone rouge de l'image segmentée, correspondent à une sur-segmentation du fond : les vecteurs déplacement calculés pour les zones uniformes du fond sont différents de ceux calculés pour les zones texturées, le champ de vecteurs déplacement relatifs au fond de la scène n'est donc pas totalement homogène et certaines zones de disparité apparaissent (zones vertes).

Les figures 3.21 et 3.22 présentent les résultats de la segmentation basée mouvement après notre mé-

thode d'estimation du mouvement global pour trois segments temporels des séquences vidéo *Tractor* et *New Mobile and Calendar*. Le tracteur en mouvement dans la séquence *Tractor* est détecté correctement ainsi que le semoir fixé à l'arrière de celui-ci (figure 3.21). Les roues ayant un mouvement de rotation sont également détectées, cependant la partie avant de celles-ci appartiennent à la même région que le semoir. Ceci s'explique par le fait que les vecteurs de mouvement relatifs à l'avant des roues pointent vers le bas haut et que ceux relatifs au semoir, qui descend vers le sol, pointent vers le bas. Dans la figure 3.22, les objets en mouvement, c'est-à-dire, le calendrier et le train sont détectés par notre méthode de segmentation basée mouvement.

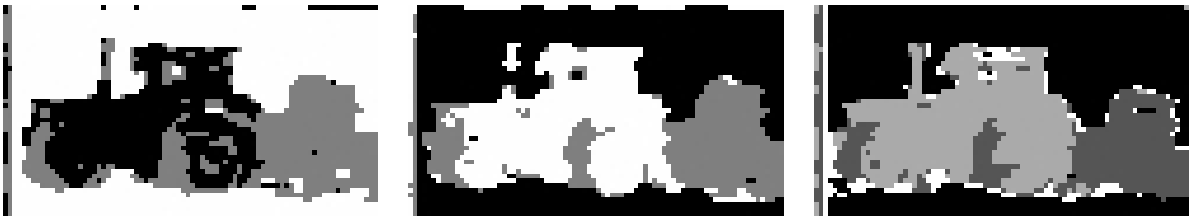


FIG. 3.21 – Cartes de segmentation basée mouvement obtenues pour trois segments temporels successifs (11, 12 et 13) de la séquence *Tractor*.

FIG. 3.22 – Cartes de segmentation basée mouvement obtenues pour trois segments temporels successifs (53, 54 et 55) de la séquence *New Mobile and Calendar*.

Cependant, les deux figures 3.21 et 3.22 illustrent les défauts de notre méthode de segmentation basée mouvement. En effet, pour la séquence *Tractor,* la caméra opérant un mouvement de translation de la droite vers la gauche, des zones découvertes apparaissent sur la gauche de l'écran et celles-ci ne sont pas segmentées correctement. Ce phénomène est illustré également sur la deuxième carte présentée à la figure 3.22. Bien que les objets en mouvement soient détectés, leurs frontières dans les cartes de segmentation ne sont pas aussi nettes qu'en réalité et des blocs isolés appartenant à des zones uniformes sont mal étiquetés. En effet, bien que les vecteurs de mouvement appartenant aux zones uniformes soient pondérés par un indice de confiance en fonction de leur activité spatiale pour l'estimation du mouvement global, ils sont la seule information utilisée par notre méthode de segmentation basée mouvement et cela ne semble pas suffisant.

Afin de supprimer ces disparités au sein de zones homogènes au sens du mouvement, nous allons introduire de nouveaux critères dans la segmentation afin de prendre en compte les caractéristiques de couleur et de texture des objets. Ces nouveaux critères doivent permettre également de lisser les frontières afin d'obtenir les formes plus précises et réalistes des objets en mouvement.

3.2.4 Segmentation spatio-temporelle multi-critères

La segmentation au sens du mouvement uniquement ne suffit pas pour créer une décomposition satisfaisante d'un segment temporel en objets spatio-temporels. En effet, pour les scènes présentant des mouvements de caméra complexes (zoom, rotation) et des contenus spatiaux uniformes, les vecteurs déplacement

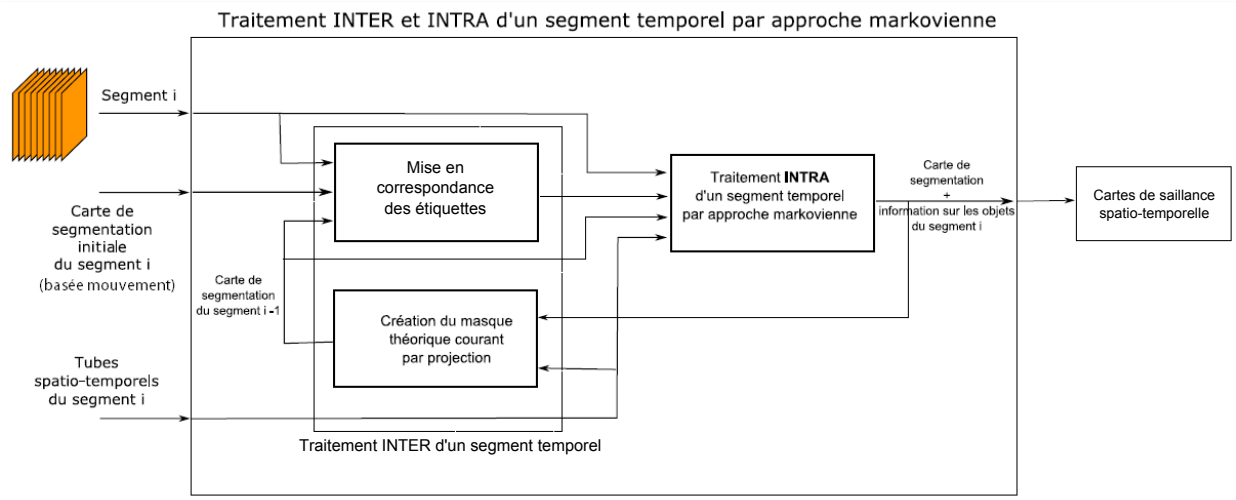


FIG. 3.23 – Bloc de traitement spatio-temporel d'un segment temporel par approche markovienne.

calculés ne reflètent pas suffisamment bien les mouvements réels des objets et ne sont pas assez précis pour être rattachés efficacement à l'un des objets détectés avec des critères basés mouvement. Pour obtenir une segmentation cohérente dans de tels cas de figure, des critères spatiaux et temporels supplémentaires vont être calculés, et intégrés afin d'affiner la carte de segmentation initiale basée sur le mouvement. Imaginons par exemple, une séquence dans laquelle une zone de découvrement ne permettrait pas d'estimer correctement les mouvements, l'ajout de critères purement spatiaux permettra de mieux faire l'appariement de chaque élément de cette zone découverte, dont le mouvement n'est pas fiable, à l'objet spatio-temporel qui lui correspond. Les critères supplémentaires choisis sont la connexité spatio-temporelle intra-segment, la couleur, la texture, et le voisinage temporel.

Le calcul et l'intégration de ces critères à l'outil de pré-analyse permettront non seulement de corriger la carte de segmentation initiale, mais aussi d'assurer le suivi des objets d'un segment temporel à l'autre. En effet, le calcul de ces critères spatiaux-temporels va fournir une description très précise des objets détectés. Le niveau de détail atteint quant à la caractérisation des objets permettra donc d'assurer le suivi des objets entre plusieurs segments temporels successifs.

Les spécifications présentées ci-avant nous ont mené à décomposer le système de raffinement de la segmentation en un ensemble de fonctions, agencées les unes avec les autres selon le schéma-bloc présenté à la figure3.23. Les différents blocs seront détaillés par la suite. Les approches statistiques étant couramment utilisées pour aborder la construction de masques d'objets, et comme le type de connaissances *a priori* que l'on veut inclure s'exprime principalement en termes de contextes spatiaux et temporels, les critères spatiaux-temporels seront intégrés au système initial via une approche markovienne. Celle-ci est composé d'un traitement intra image permettant d'affiner les résultats de la segmentation au sein d'un segment temporel, et d'un traitement inter image permettant le suivi des objets entre les segments temporels successifs.

3.2.4.1 Mise en forme du problème d'estimation

Un champ de Markov est caractérisé par ses propriétés locales, tandis qu'un champ de Gibbs est caractérisé par sa propriété globale (distribution de Gibbs). Besag [Bes74] a reformulé la relation entre champs markoviens et distributions de Gibbs initialement démontrée par Hammersley et Clifford en 1971. La possibilité d'exprimer par une distribution explicite les propriétés markoviennes d'un champ a permis l'essor

du développement de modèles markoviens. Nous allons dans un premier temps reprendre les principaux aspects mathématiques de ce type de modélisation. Les notations suivantes sont adoptées pour la résolution de notre problème :

- $E = \{e_s, s \in S\}$ est le champ d'étiquettes sur l'ensemble S des sites s. Dans notre cas, un site est un tube spatio-temporel, et les sites d'une région segmentée (correspondant à un objet en mouvement à travers les segments temporels) ont le même label ;
- $O = \{O_s, s \in S\}$ est le champ des observations. Les réalisations des champs O seront notées $o = \{o_s, s \in S\}$;
- Λ (respectivement Ω) est l'ensemble des réalisations possibles de E (respectivement toutes les configurations d'étiquettes possibles de e) ;
- $\eta = \{\eta_s,\ s \in S\}$ est une structure de voisinage définie sur s.

(E, O) est modélisé par un champ de Markov aléatoire. Dans ce cas, le champ d'étiquettes optimal $\hat{e}$ est obtenu selon un critère MAP (*Maximum A Posteriori*). Le théorème de Hammersley et Clifford établit l'équivalence entre les champs markoviens et les distributions de Gibbs [Bes74], la configuration optimale du champ des étiquettes est alors obtenue en minimisant une fonction d'énergie globale $U(o,e)$:

$$\hat{e} = arg\min_{e\in\Omega} U(o,e) \tag{3.20}$$

Les propriétés markoviennes du champ d'étiquettes permettent d'écrire cette fonction d'énergie comme étant la somme de fonctions de potentiels élémentaires. Ces fonctions de potentiel sont définies localement sur des structures appelées cliques [GG84] :

$$U(o,e) = \sum_{c\in C} V_c(o,e) \tag{3.21}$$

où C est l'ensemble des cliques c de S relatives au voisinage η. Une clique est un sous-ensemble de sites de S tel que, si s_i et s_j sont deux sites quelconques de cette clique, s_i et s_j sont voisins au sens de η. Des exemples de systèmes de voisinages et de cliques associées sont présentés en figure 3.24.

La fonction de potentiel V_c est définie localement sur la clique c et donne les interactions locales entre les différents sites qui composent la clique. L'expression analytique de la fonction V_c est dépendante du problème posé et des résultats souhaités, elle définit les propriétés locales et globales du problème.

3.2.4.2 Fonctions de potentiel

Les fonctions de potentiel vont permettre de définir, en fonction de chaque nouveau critère (couleur, texture, ...), la probabilité pour un site donné s d'être étiqueté avec e. D'après l'équation 3.20, le champ d'étiquettes le plus probable sera celui qui minimisera l'énergie globale $U(o,e)$. Chaque critère va contribuer à la valeur de cette énergie, qui s'exprime donc sous la forme :

$$U(o,e) = \sum_{i=1}^{5} \alpha_i.W_i \tag{3.22}$$

où chaque α_i est le poids de l'énergie élémentaire W_i qui représentent respectivement les critères de voisinage spatial, de couleur, de texture, de mouvement et de voisinage temporel. Les énergies élémentaires $\{W_i\}_{i=1...5}$ sont calculées comme la somme de fonctions de potentiels élémentaires (cf. équation 3.21). Afin de pouvoir comparer ces énergies et obtenir des ordres de grandeur homogènes, nous allons normaliser toutes les fonctions de potentiels sur l'intervalle centré $[-1;1]$. Ainsi, seuls les poids $\{\alpha_i\}_{i=1..5}$ attachés à

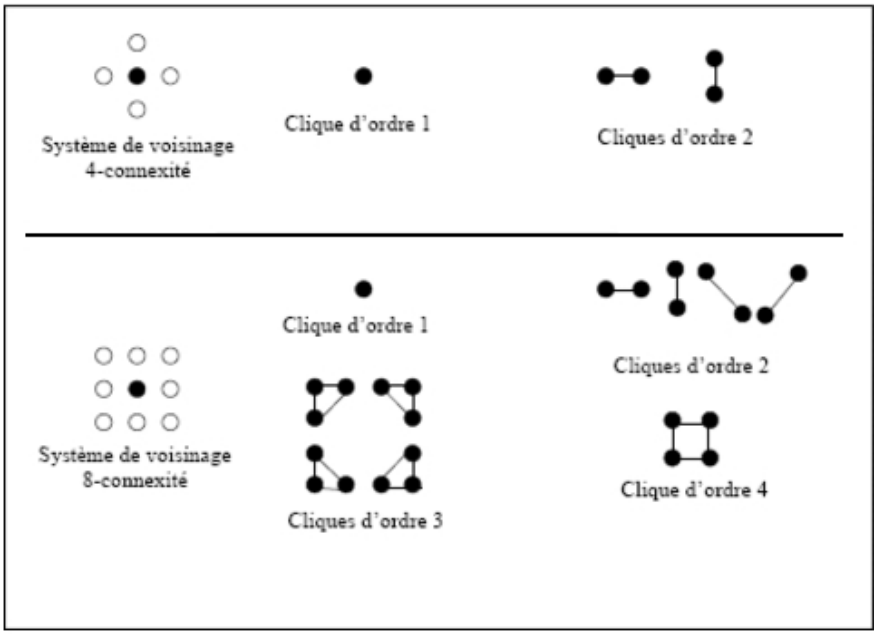


FIG. 3.24 – Cliques associées à des systèmes de voisinage en 4-connexité (en haut) et 8-connexité (en bas).

ces énergies, permettront de pondérer chaque critère dans le calcul de l'énergie globale $U(o,e)$.

3.2.4.2.1 Caractéristiques spatiales

Pour un segment temporel donné de neuf images, une région segmentée doit respecter une cohérence spatiale, c'est-à-dire que la région segmentée (constituée d'une fusion de tubes spatio-temporels) doit être localement homogène et compacte. L'énergie à minimiser $U(o,e)$ sera donc composée d'une énergie élémentaire chargée d'assurer l'homogénéité des étiquettes pour des sites voisins spatialement. Dans notre cas, le système de voisinage choisi est 8-connexe, et les cliques retenues sont les cliques d'ordre 2. Ce système de voisinage est représenté à la figure 3.25. Le modèle choisi pour favoriser la création de régions homogènes est tel que sa fonction de potentiel s'écrit :

$$\forall t \in \eta_s \begin{cases} V_{c_s} = \beta_s & \text{si } e_t \neq e_s \\ V_{c_s} = -\beta_s & \text{si } e_t = e_s \end{cases}$$

avec $\beta_s > 0$. Dans notre cas, chaque clique correspond à une paire de tubes spatio-temporels voisins et connectés au sens d'un voisinage 8-connexe. Afin de normaliser cette fonction de potentiel sur l'intervalle $[-1;1]$, le paramètre β_s sera fixé à $1/8$ pour un voisinage 8-connexe. L'énergie élémentaire $W_1(e_s)$ liée au voisinage spatial, s'exprime donc sous la forme :

$$W_1(e_s) = \sum_{c_s \in C_s} V_{c_s}(e_s, e_t)$$

où C_s représente l'ensemble de toutes les cliques spatiales de S.

3.2.4.2.2 Caractéristiques de couleur

Afin de savoir si un site est étiqueté de manière cohérente dans un segment temporel, nous souhaitons pouvoir comparer les distributions de couleur de ce site avec celles des différentes régions existantes. Plusieurs méthodes sont adaptées au cas discret (intersection, L_2, χ_2, ...), nous avons opté pour l'utilisation

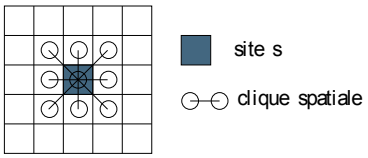


FIG. 3.25 – Ensemble des cliques spatiales d'ordre 2 associées à un voisinage 8-connexe.

du coefficient de Bhattacharyya qui permet de mesurer la similarité entre deux distributions de m couleurs possibles :

- soit $\hat{s} = \{\hat{s_u}\}_{u=1..m}$ la densité de probabilité discrète de couleur du site courant s ;
- soit $\widehat{R(e_s)} = \{\widehat{R(e_s)}_u\}_{u=1..m}$ la densité de probabilité discrète de couleur de la région $R(e_s)$ constituée des sites étiquetés e_s.

Le coefficient de Bhattacharyya qui permet de comparer ces densités est défini par :

$$\rho_{couleur} = \rho_{couleur}(\widehat{R(e_s)}, \hat{s}) = \sum_{u=1}^{m} \sqrt{\widehat{R(e_s)}_u \times \hat{s_u}}$$

Les densités de probabilité discrètes de couleur sont calculées à partir des histogrammes de couleur correspondants. Pour diminuer la complexité des calculs et regrouper les couleurs proches, chaque composante couleur est uniformément quantifiée sur 16 niveaux (donc $m = 16^3 = 4096$ couleurs possibles). Les histogrammes de couleur sont ensuite calculés en considérant uniquement l'image centrale du segment temporel courant : l'histogramme couleur d'un site est donc calculé à partir du macrobloc central du tube correspondant, et non à partir des neuf macroblocs qui constituent ce tube. Ces histogrammes sont alors normalisés par le nombre d'éléments qui ont y contribué, afin d'obtenir les densités de probabilité discrètes de couleur.

Le coefficient de Bhattacharyya varie de 0 (distributions totalement différentes) à 1 (distributions identiques). Afin de normaliser la fonction de potentiel associée à la couleur sur l'intervalle $[-1;1]$, nous utilisons la transformation linéaire : $V_{couleur} = 1 - 2 \times \rho_{couleur}(\widehat{R(e_s)}, \hat{s})$. Notons que la fonction utilisée inverse le signe initial du coefficient de Bhattacharyya, afin que deux distributions proches (coefficient de Bhattacharyya fort) aient une énergie faible. L'énergie élémentaire W_2 pour le critère de couleur est donc définie par :

$$W_2(e_s, o_s, o(R(e_s)) = \sum_{s \in S} V_{couleur}$$

3.2.4.2.3 Caractéristiques de texture

Il s'agit ici de comparer la similarité entre les textures d'un site et celles des différentes régions existantes. Comme dans le cas de la fonction de potentiel associée à la couleur, l'information de texture va être représentée sous la forme d'une distribution. Le même système de notation est conservé, où n représente le nombre de « valeurs » possibles de textures :

- $\hat{s} = \{\hat{s_v}\}_{v=1..n}$ est la densité de probabilité discrète de texture du site courant s ;
- $\widehat{R(e_s)} = \{\widehat{R(e_s)}_v\}_{v=1..n}$ est la densité de probabilité discrète de texture de la région $R(e_s)$ constituée des sites étiquetés e_s.

Par soucis de simplification, les distributions pour la texture seront calculées en ne considérant que l'image centrale du segment temporel courant. Chaque pixel de l'image centrale va donner une information de texture représentée sous la forme d'un couple de gradients (ΔH, ΔV) (respectivement le gradient spatial horizontal et le gradient spatial vertical). Afin de réduire l'importance du bruit d'acquisition dans le calcul

des textures, le gradient ΔH (respectivement ΔV) de chaque pixel est obtenu en filtrant l'image centrale du segment temporel avec un filtre de Sobel horizontal (respectivement vertical)[6]. Les gradients correspondent alors aux valeurs filtrées (en valeurs absolues) et quantifiées selon la loi suivante :

$$\begin{cases} \Delta H = \lfloor |x|/4 \rfloor & \text{si } x < 64 \\ \Delta H = 15 & \text{sinon} \end{cases}$$

où x représente la valeur filtrée d'un pixel de l'image avec un noyau de Sobel horizontal (la même loi est utilisée dans le cas vertical). D'après cette équation, chacun des deux gradients qui composent la texture est quantifié sur 16 niveaux, il y a donc $n = 16^2 = 256$ « valeurs » possibles de textures. Une fois les distributions 2D de texture calculées pour un site et pour une région, nous les comparons en utilisant de nouveau le coefficient de Bhattacharyya :

$$\rho_{texture} = \rho_{texture}(\widehat{R(e_s)}, \hat{s}) = \sum_{v=1}^{n} \sqrt{\widehat{R(e_s)}_v \times \hat{s_v}}$$

Comme dans le cas de la couleur, une fonction de potentiel à valeurs dans l'intervalle $[-1;1]$ est déduite de ce coefficient : $V_{texture} = 1 - 2 \times \rho_{texture}(\widehat{R(e_s)}, \hat{s})$. L'énergie élémentaire W_3 pour le critère de texture est donc définie par :

$$W_3(e_s, o_s, o(R(e_s))) = \sum_{s \in S} V_{texture}$$

3.2.4.2.4 Caractéristiques de mouvement

Dans un segment temporel, nous avons vu que le critère principal pour la segmentation est le mouvement : pour une région donnée, les vecteurs mouvement associés à des tubes spatio-temporels doivent avoir des valeurs proches. C'est le critère qui permet de créer un premier champ d'étiquettes avant son raffinement par modèles markoviens. Ce critère doit donc conserver une importance dans le calcul de l'énergie globale $U(o,e)$. Une énergie élémentaire liée au mouvement est alors définie, de manière à mesurer la ressemblance entre le mouvement d'un site et celui d'une région. La fonction de potentiel qui mesure cette ressemblance est définie par :

$$V_{mouvement} = -\frac{\overrightarrow{VM_s} \times \overrightarrow{VM_{R(e_s)}}}{max(\left\|\overrightarrow{VM_s}\right\|, \left\|\overrightarrow{VM_{R(e_s)}}\right\|)^2}$$

Où $\overrightarrow{VM_s}$ et $\overrightarrow{VM_{R(e_s)}}$ sont respectivement les vecteurs de mouvement associés au site s, et à la région $R(e_s)$ formée des sites étiquetés e_s. Le produit scalaire normalisé, présenté dans l'équation ci-dessus, fournit une valeur de ressemblance pour les vecteurs qui varie entre -1 et 1. L'inversion du signe de ce produit scalaire permet d'attribuer un potentiel faible lorsque les mouvements sont proches et plus important lorsqu'ils sont différents. L'énergie élémentaire W_4 pour le critère de mouvement est donc définie par :

$$W_4(e_s, o_s, o(R(e_s))) = \sum_{s \in S} V_{mouvement}$$

[6] noyau de Sobel du filtre horizontal $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$, du filtre vertical $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

3.2.4.2.5 Caractéristiques temporelles

La durée de vie d'un objet spatio-temporel est généralement plus grande que la durée d'un segment temporel (*180 ms*) : le « cycle de vie » d'un objet s'étend typiquement sur plusieurs segments successifs. Une région segmentée dans un segment temporel t doit donc présenter une cohérence temporelle avec la région correspondante dans le segment $t-1$ (si elle existe), c'est-à-dire que la forme d'une région segmentée doit rester temporellement homogène et compacte. Afin d'assurer cette propriété entre deux segments successifs, nous utilisons la projection temporelle du segment $t-1$ à l'instant t. Cette projection tient compte du mouvement global de la caméra et des mouvements locaux des objets segmentés. Une clique temporelle peut alors être définie entre le segment courant et le segment précédent projeté pour maintenir l'homogénéité de la forme de la région segmentée. Ce système est présenté à la figure 3.26. Sur ce schéma, une région, représentée en couleur, est poursuivie d'un segment à un autre et change sensiblement de forme au cours du temps.

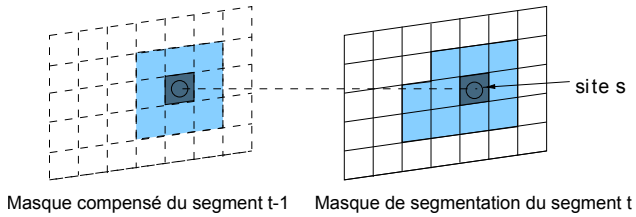


FIG. 3.26 – Clique temporelle entre deux segments successifs.

La fonction de potentiel associée au critère temporel est définie par :

$$\begin{cases} V_{c_t} = \beta_t & \text{si } e_s(t) \neq e_s(t-1) \\ V_{c_t} = -\beta_t & \text{si } e_s(t) = e_s(t-1) \end{cases}$$

avec $\beta_t = 1$, et où $e_s(t)$ et $e_s(t-1)$ sont respectivement les étiquettes du site du segment courant et du site du segment précédent projeté. L'énergie élémentaire W_5 pour le critère temporel est alors définie par :

$$W_5(e_s(t)) = \sum_{c_t \in C_t} V_{c_t}(e_s(t), e_s(t-1))$$

où C_t est l'ensemble de toutes les cliques temporelles de S. Notons que pour utiliser cette fonction de potentiel, il faut qu'un même objet garde la même étiquette d'un segment temporel à l'autre, il faut donc assurer le suivi temporel des objets à travers les segments successifs. C'est l'objet du prochain paragraphe.

3.2.4.3 Suivi d'objets

Un même objet peut avoir un cycle de vie qui s'étend sur plusieurs segments temporels successifs. Dans une perspective de codage cohérent d'un objet, il est donc intéressant de réussir à suivre un objet sur plusieurs segments contigus. Pour réaliser ce suivi entre un objet du segment $t-1$ et un objet du segment t, nous compensons en mouvement la carte de segmentation du segment $t-1$ à l'instant t, puis nous mettons en correspondance cette carte compensée avec la carte de segmentation du segment courant à l'instant t.

3.2.4.3.1 Compensation en mouvement de la carte de segmentation du segment *t-1*

Considérons la carte de segmentation disponible du segment temporel $t-1$, pour pouvoir la comparer spatialement avec le segment courant t, il faut la projeter temporellement à cet instant t. Une nouvelle carte

de segmentation projetée est alors disponible. Cette projection est décomposée en plusieurs étapes : détection de l'objet « fond », initialisation de la carte de segmentation projetée par l'étiquette du fond, et projection des objets avec leurs vecteurs de déplacement. Un exemple de projection est présenté en figure 3.27.

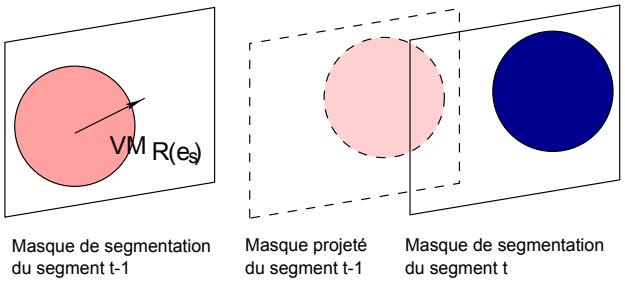


FIG. 3.27 – Suivi d'objets entre des segments temporels successifs.

3.2.4.3.2 Étiquetage et suivi des objets

Pour effectuer l'appariement des objets d'un segment au segment suivant, une métrique basée sur la similarité des couleurs, des textures, et sur le taux de recouvrement est utilisée. La similarité des couleurs et des textures est mesurée de nouveau à l'aide du coefficient de Bhattacharyya. Chaque objet du segment courant est comparé aux objets présents dans la carte compensée du segment précédent. L'objet du segment courant prend alors l'étiquette de l'objet le plus proche, en accord avec la métrique utilisée, à condition que leur similarité soit « assez forte ». En pratique, on fixe des seuils expérimentaux pour le coefficient de Bhattacharyya sur la couleur, celui sur la texture, et le taux de recouvrement. Si les deux objets les plus proches présentent, pour chacun de ces trois seuils, une similarité assez forte, alors on considère qu'il s'agit du même objet sur les deux segments successifs.

D'autre part, il peut arriver que la méthode de segmentation au sens du mouvement ne distingue aucun objet (tous les objets ont des mouvements proches ou trop difficiles à estimer). Dans ce cas, la carte de segmentation initiale du segment courant pour le traitement avec approche markovienne est vide (en fait elle ne comprend qu'un seul objet : le fond), et donc notre approche markovienne n'aurait aucun effet. Nous choisissons alors pour ces configurations particulières, d'initialiser la méthode de segmentation markovienne avec la carte de segmentation projetée du segment précédent. Cette technique permet de rester efficace dans le suivi des objets à travers les segments temporels successifs.

3.2.4.4 Minimisation de l'énergie globale

Nous avons montré que pour obtenir le champ d'étiquettes optimal, il faut minimiser la fonction d'énergie $U(o,e)$ donnée par l'équation 3.22. La carte issue de la segmentation basée mouvement va servir d'initialisation pour la segmentation par approche markovienne. Nous calculons pour chaque site un degré de stabilité $\Delta U(s)$, qui correspond à la variation entre l'énergie associée au site pour l'étiquette courante e_c et l'énergie minimale qu'aurait ce site avec une étiquette optimale e_s : $\Delta U(s) = U(s,e_c) - U(s,e_s)$, si $\Delta U(s)$ est non nul alors le site est instable. Nous mettons en œuvre une pile d'instabilité : le site le plus instable est traité en premier et ainsi de suite de façon itérative, jusqu'à ce que tous les sites soient stables. Chaque site instable est traité de la façon suivante :

- si le site a déjà été traité plusieurs fois, il empêche la solution de converger, il est donc retiré des sites à traiter ;
- le site courant prend l'étiquette qui minimise son énergie ;

– l'énergie des sites voisins est modifiée en fonction de la nouvelle étiquette du site courant.

Typiquement, les sites les plus instables sont ceux situés sur les bords de la carte de segmentation et ceux situés sur les bords des objets segmentés. Le traitement est terminé lorsque tous les sites sont stables, ou que les seuls sites non-stables restants sont ceux qui empêchent la solution de converger.

Notons, que la méthode utilisée ici vise à minimiser l'énergie globale $U(o,e)$ du champ des étiquettes, en minimisant successivement et localement les énergies de chaque site. Cette méthode simple est une méthode de relaxation déterministe. Elle assure la convergence vers le premier minimum d'énergie trouvé qui n'est pas forcément le minimum global. À l'inverse, les méthodes de relaxation stochastiques, telles que l'algorithme de Metropolis [GG84], l'échantillonneur de Gibbs [CR87] ou le recuit simulé [KGV83], autorisent des configurations qui augmentent provisoirement l'énergie du système, afin de converger vers un minimum global [Lal90]. Cependant ces méthodes sont complexes et peu adaptées à notre contexte d'utilisation.

3.2.4.5 Facteur d'importance des critères ajoutés

Un objet vidéo est une forme spatio-temporelle caractérisée par sa texture, sa couleur et son mouvement qui, souvent, diffère du mouvement global de la scène. Nous avons choisi de poser l'hypothèse selon laquelle le mouvement est le critère le plus déterminant pour segmenter les objets d'un segment temporel. Cependant, les informations de mouvement sont obtenues à partir d'une méthode d'estimation dont la précision dépend fortement des contenus vidéos. Ainsi, pour certaines séquences, le mouvement sera un critère fiable, alors que pour d'autres séquences la segmentation devra s'appuyer plus fortement sur les critères de couleur, de texture ou de voisinage. Ce constat nous a amené à créer deux jeux de paramètres $\{\alpha_i\}_{i=1..5}$ pour calculer l'énergie globale $U(o,e)$ selon que l'estimation des mouvements soit considérée fiable ou non. Dans le cas où le mouvement sera considéré comme fiable, le paramètre α_4 qui représente l'importance de l'énergie liée au mouvement dans l'énergie globale $U(o,e)$ sera augmenté, tandis que les poids liés aux autres énergies seront plus faibles. Inversement, si l'estimation de mouvement est jugée trop peu précise, ce poids sera diminué et les autres poids augmentés.

Pour caractériser la précision de l'estimation de mouvement, nous calculons pour chaque objet spatio-temporel une pseudo-variance des vecteurs de mouvement associés aux tubes qui constituent cet objet. Soit $\overrightarrow{MV_{objet}}$ le vecteur mouvement représentant le déplacement d'un objet, et $\{\overrightarrow{MV_i}\}_{i=1..N}$ l'ensemble des vecteurs déplacement rattachés à cet objet, la pseudo-variance $\overline{\sigma}^2$ sera alors définie par : $\overline{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(\overrightarrow{MV_{objet}} - \overrightarrow{MV_i})^2$, où N représente le nombre de tubes qui constituent l'objet. Si cette pseudo-variance est inférieure à un certain seuil qui empiriquement a été fixé à 40, nous considérons que la segmentation au sens du mouvement est assez précise : les mouvements associés aux tubes qui composent l'objet sont proches, les poids liés aux autres critères (couleur, texture, ...) seront donc moins importants :

$$\begin{cases} \alpha_1 = \alpha_2 = \alpha_3 = 0.4,\ \alpha_4 = 1,\ \alpha_5 = 0.6 & \text{si } \overline{\sigma}^2 < 40 \\ \alpha_1 = \alpha_2 = 1,\ \alpha_3 = 0.4,\ \alpha_4 = \alpha_5 = 0.6 & \text{si } \overline{\sigma}^2 \geqslant 40 \end{cases}$$

3.2.4.6 Résultats expérimentaux

Les résultats obtenus après une segmentation basée mouvement seule, et ceux obtenus en couplant cette segmentation à l'approche markovienne présentée avant montrent que les objets spatio-temporels sont plus fidèlement détectés avec l'approche markovienne. Le tableau 3.1 présente, pour chacune des deux méthodes, le ratio entre le nombre d'objets en mouvement détectés et le nombre réel d'objets en mouvement au sein de

séquences HD. Bien que le taux de détection soit augmenté, nous remarquons que l'approche markovienne n'assure pas la détection de tous les objets. Par exemple, à la fin de la séquence *Tractor*, le tracteur est trop petit pour être détecté à cause du zoom sortant de la caméra.

Séquence	Segmentation basée mouvement	Segmentation spatio-temporelle
Tractor (690 images)	33%	84%
New Mobile and Calendar (500 images)	85%	92%
Shields (500 images)	94%	100%

TAB. 3.1 – Ratio des objets en mouvement détectés.

Les figures 3.28 et 3.29 présentent, pour quatre segments temporels successifs des séquence *Tractor* et *New Mobile and Calendar*, les cartes de segmentation obtenues avec une segmentation basée mouvement uniquement (ligne du milieu) et une approche markovienne (ligne du bas). Les objets en mouvement sont correctement détectés avec la segmentation basée mouvement, mais le suivi des objets en mouvement entre les segments n'est pas assuré (un même objet peut avoir des étiquettes différentes d'un segment à l'autre). Avec l'approche markovienne, les bords des objets en mouvement sont plus réguliers et les contenus plus homogènes, de plus le suivi entre segments est assuré : par exemple, l'étiquette du tracteur reste la même sur les quatre segments temporels.

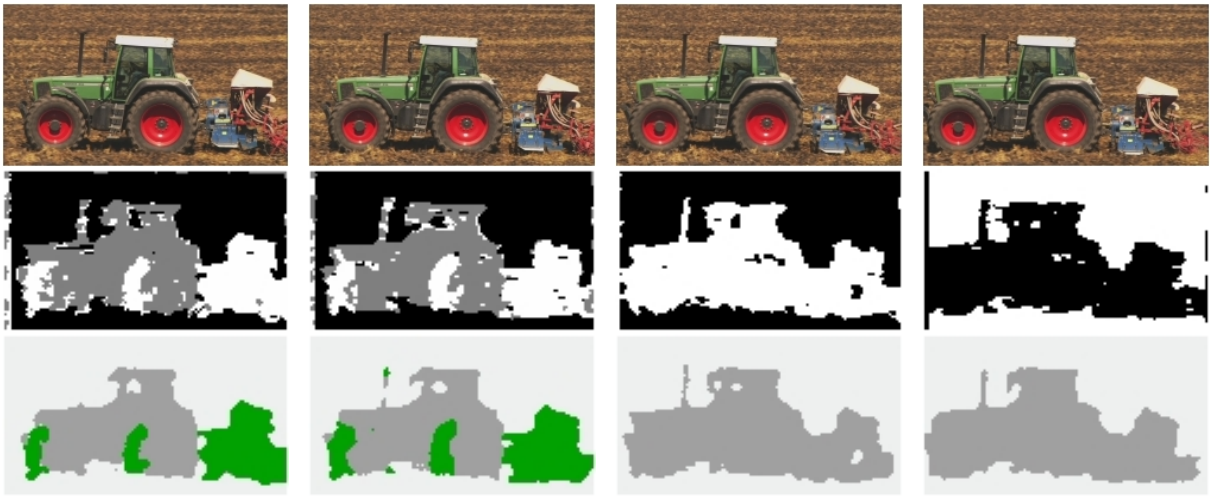


Figure 3.28: Cartes de segmentation pour *Tractor* (segments 13 à 16) : segmentation basée mouvement (ligne du milieu) et approche markovienne (ligne du bas).

L'ajout de critères spatiaux-temporels par approche markovienne a donc permis, d'une part, d'améliorer la qualité de la segmentation basée mouvement initiale et, d'autre part, d'assurer le suivi des objets sur plusieurs segments temporels successifs. De plus, nous savons que les objets en mouvement correspondent aux régions les plus attractives visuellement (voir le chapitre 2). À partir de ces informations nous allons pouvoir modéliser l'attention visuelle. En effet, la dernière étape de note méthode de pré-analyse détermine des cartes de saillance représentant les positions des zones visuellement importantes.

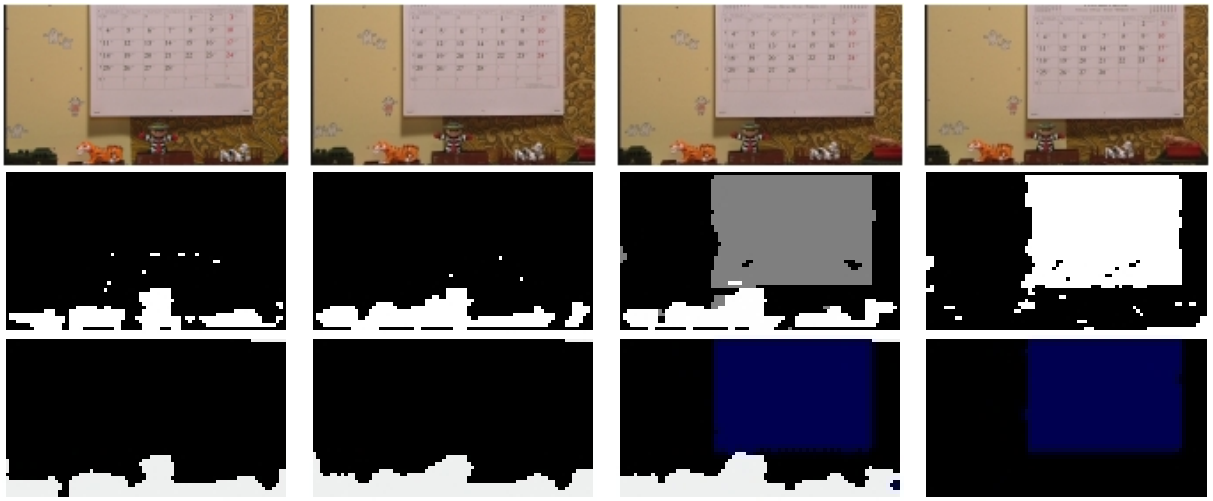


Figure 3.29: Cartes de segmentation pour *New Mobile and Calendar* (segments 50 à 53) : segmentation basée mouvement (ligne du milieu) et approche markovienne (ligne du bas).

3.3 Détermination des cartes de saillance

Comme nous l'avons vu dans le chapitre précédent, afin de faire face à l'énorme quantité d'informations visuelles, le SVH possède la faculté de sélectionner l'information pertinente localisée spatialement dans le champ visuel : on parle d'attention visuelle. Du fait de la grande complexité des inter-actions et des inter-dépendances existantes entre les mécanismes du SVH, modéliser l'attention visuelle dans son ensemble demeure pour l'instant trop complexe. Une voie réaliste est de modéliser l'attention visuelle pré-attentive, c'est-à-dire, de prédire à partir d'attributs de bas niveau, les positions des zones visuellement importantes d'une image ou d'une séquence d'images. Le modèle proposé doit être capable de déterminer les zones visuellement importantes d'une image, et dans notre cas d'une séquence vidéo.

De nombreux facteurs influençant l'attention visuelle ont été identifiés [WH04] et sont regroupés en deux catégories. La première concerne toutes les informations spatiales [AM08] les plus susceptibles de stimuler l'attention visuelle comme la couleur, l'orientation et la taille. La deuxième catégorie concerne les informations temporelles [WCF89]. Une séquence vidéo contient ces deux types d'informations susceptibles de stimuler l'attention visuelle. C'est pourquoi notre modèle d'attention visuelle pré-attentive doit les prendre en compte. Celui-ci se décompose en deux parties, l'une modélise l'attention visuelle à partir des informations spatiales et l'autre à partir des données temporelles. La dernière étape combine ces deux parties afin d'obtenir une carte de saillance spatio-temporelle.

3.3.1 Saillance spatiale basée sur le contraste de couleur

Des informations importantes peuvent être trouvées dans la littérature sur la théorie des couleurs et plus particulièrement sur les caractéristiques des couleurs qui contribuent à rendre un objet visuellement saillant ou non [Itt61, Mah96]. En termes de saillance de couleur, les modèles d'attention visuelle se sont concentrés uniquement sur les propriétés des couleurs qui ont été signalées en psychologie, de ce fait, de nombreux aspects importants décrits sur le sujet dans la théorie des couleurs ont été négligés. Johannes Itten [Itt61] fut l'un des premiers spécialistes de la théorie des couleurs à décrire des méthodes de combinaisons de couleurs produisant des effets de contraste. Il a défini différentes situations dans lesquelles le SVH détecte un contraste dans une scène colorée. Selon ses recherches, le contraste peut se produire par la présence d'objets présentant des grandes différences d'intensité, de saturation et/ou de teinte. Les autres causes rapportées

incluent la présence de couleurs opposées et de co-occurrence de couleurs chaudes et froides.

3.3.1.1 Caractéristiques de couleur influençant l'attention visuelle

Dans leurs travaux Aziz et Mertsching [AM08] combinent ces concepts et formulent un ensemble de points possibles à mettre en œuvre. Il reste ensuite à décider quelles couleurs vont être saillantes en présence d'un contraste. Les différents cas avec mention de la couleur saillante sont énumérés ci-dessous :

1. *Contraste de Saturation* : Un contraste est produit par des couleurs faiblement et fortement saturées. La valeur du contraste est directement proportionnelle à l'amplitude de la différence de saturation. Les couleurs fortement saturées tendent à attirer l'attention dans de telles situations, à moins qu'une région faiblement saturée soit entourée par une région fortement saturée.
2. *Contraste d'Intensité* : Un contraste sera visible lorsque des couleurs sombres et lumineuses co-existent. Plus la différence d'intensité est importante, plus l'effet de contraste augmente. Les couleurs lumineuses attirent l'attention dans cette situation, à moins que la région sombre ne soit entourée par une région lumineuse.
3. *Contraste de Teinte* : La différence des angles de teinte sur le disque des couleurs (cf figure 3.30) contribue à la création d'un contraste. Une différence importante va manifestement produire un contraste fort. Du fait de la nature circulaire de la teinte, la plus grande différence entre deux valeurs de teinte est de 180°.
4. *Contraste d'Opposants* : Les couleurs situées sur les côtés opposés du disque de teinte produisent une importante valeur de contraste. Cela signifie naturellement que la différence des angles des valeurs de teinte est dans ce cas proche de 180°. Les couleurs situées dans la première moitié du disque de teinte, connues comme la gamme de couleur active, domineront sur le reste des couleurs passives.
5. *Contraste des couleurs Chaudes et Froides* : Les couleurs chaudes, c'est-à-dire rouge, jaune et orange sont visuellement plus saillantes. Ces couleurs sont situées dans les premiers 45° du disque de teinte. Les couleurs chaudes et froides créent un contraste pour lequel les couleurs chaudes restent dominantes.
6. *Dominance des Couleurs Chaudes* : Les couleurs chaudes dominent leur environnement, même si un contraste existe dans cet environnement.
7. *Dominance de la Luminosité et de la Saturation* : Les couleurs fortement lumineuses et saturées sont considérées comme étant attractives sans tenir compte de leurs valeurs de teinte. De telles couleurs ont plus de chance d'attirer l'attention.

L'effet de contraste est contrôlé par la valeur de saturation des deux couleurs impliquées dans les situations mentionnées aux points 2 à 5. Les couleurs fortement saturées impliquent des contrastes importants. Notre modèle de saillance des couleurs, basé sur les travaux de Aziz et Mertsching [AM08], combinent tous les points mentionnés ci-dessus. Nous divisons cette procédure en sept étapes, chacune contribuant à la valeur de saillance d'un site s. Les valeurs des différentes composantes de couleur utilisées dans nos calculs pour un site s, sont les valeurs moyennes du bloc situé au centre du tube considéré.

3.3.1.2 Calcul de la saillance spatiale

Les cinq premières étapes de l'algorithme utilisent un ou les deux facteurs de saturation f_{ij}^{sat} et d'intensité f_{ij}^{int} dans leurs calculs. Les indices i et j représentent respectivement la position du site courant et

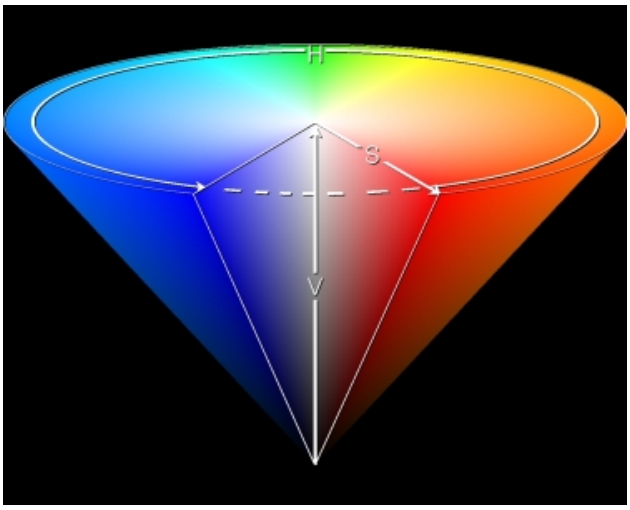


FIG. 3.30 – Représentation conique de l'espace TSV (*HSV*).

d'un site voisin (du voisinage 8-connexe). La première partie du facteur de saturation f_{ij}^{sat} est obtenue en calculant la moyenne des valeurs de saturation entre le site s_i et le site s_j, l'effet de ce facteur est donc plus important lorsque les deux blocs ont une valeur élevée de saturation et vice versa. La deuxième partie dépend seulement de la saturation du site s_i et détient une valeur minimale égale à k_{min}, afin de ne pas supprimer l'interaction des blocs avec une saturation proche de zéro. Le reste de la seconde partie est obtenu à partir de la saturation du site s_i et est pondéré par $(1-k_{min})$. Le facteur pour l'intensité est calculé de la même façon en utilisant la valeur de l'intensité de la couleur du bloc et non la valeur de la saturation. Soient $S(s_i)$ et $I(s_i)$, respectivement les valeurs de saturation et d'intensité du site s, la valeur maximale pour la saturation et l'intensité étant égale à 1, les deux facteurs de saturation et d'intensité sont définis par :

$$f_{ij}^{sat} = \frac{S(s_i)+S(s_j)}{2} \times (k_{min} + (1-k_{min}).S(s_i)) \tag{3.23}$$

$$f_{ij}^{int} = \frac{I(s_i)+I(s_j)}{2} \times (k_{min} + (1-k_{min}).I(s_i)) \quad \text{où } k_{min} = 0,21 \tag{3.24}$$

La contribution de la première étape en terme de saillance pour un site s_i est obtenue à partir des deux facteurs de saturation et d'intensité :

$$X_1(s_i) = \sum_{j=1}^{p_i} f_{ij}^{sat}.f_{ij}^{int} \ \ \forall s_j \in \eta_i \tag{3.25}$$

où p_i est la taille du voisinage (8-connexe) et η_i représente l'ensemble des sites voisins de s_i. La seconde étape collecte les contributions des sites qui ont une valeur de teinte éloignée de celle du site s_i. Le calcul de X_2^i est réalisé de la manière suivante :

$$X_2(s_i) = \sum_{j=1}^{p_i} f_{ij}^{sat}.f_{ij}^{int}.\triangle_{ij}^{teinte} \ \ \forall s_j \in \eta_i \tag{3.26}$$

où $\triangle_{ij}^{teinte}$ représente la différence de teinte entre le site s_i et le site voisin s_j. Du fait de la nature circulaire de la teinte, nous calculons la différence de teinte entre deux sites s_i et s_j de la façon suivante :

$$\triangle_{ij}^{teinte} = \begin{cases} \triangle_{ij}^{\mu} & \text{pour} \quad \triangle_{ij}^{\mu} \leq 0,5 \\ 1 - \triangle_{ij}^{\mu} & \text{sinon} \end{cases} \tag{3.27}$$

où $\triangle_{ij}^{\mu} = |H(s_i) - H(s_j)|$, $H(s_i)$ étant la valeur de teinte du site s_i, celle-ci étant comprise entre 0 et 1. Une valeur de teinte égale à 1 représente un angle de 360° et donc une valeur de $0,5$ représente un angle de 180°. Les sites voisins ayant un contraste important en terme de teinte avec le site s_i vont augmenter le poids de cette seconde contribution à la saillance finale.

Dans la troisième étape, nous étendons le principe de contraste entre couleurs chaudes et froides, au contraste entre couleurs passives et actives. Une couleur est considérée comme étant active si sa valeur de teinte est comprise dans la première moitié du disque de représentation de la teinte, c'est-à-dire, une valeur inférieure à $0,5$ (180°). Ainsi, si la couleur d'un site s_i est active, alors un site s_j avec une couleur passive va contribuer à la saillance du site s_i. Une différence importante en terme de teinte va rendre ce contraste plus saillant. Cette contribution à la saillance du site s_i s'écrit sous la forme :

$$X_3(s_i) \quad = \quad \sum_{j=1}^{p_i} f_{ij}^{sat} . f_{ij}^{int} . \triangle_{ij}^{teinte} \quad \forall s_j \in \eta_i \text{ si } H(s_i) < 0,5 \text{ et } H(s_j) \geq 0,5 \tag{3.28}$$

La quatrième étape constitue la contribution liée au contraste de saturation. Les sites possédant des différences de saturation importantes dans leur voisinage contribuent à la saillance du site s_i de la façon suivante :

$$X_4(s_i) = \sum_{j=1}^{p_i} f_{ij}^{sat} . f_{ij}^{int} . \triangle_{ij}^{sat} \quad \forall s_j \in \eta_i, \text{ où } \triangle_{ij}^{sat} \text{est la différence de saturation entre les sites } s_i \text{et } s_j. \tag{3.29}$$

La cinquième étape regroupe les contributions pour le site s_i, à partir des blocs voisins ayant une différence importante en terme d'intensité (contraste d'intensité). Le formule utilisée est similaire à celui de la quatrième étape et s'écrit :

$$X_5(s_i) = \sum_{j=1}^{p_i} f_{ij}^{sat} . f_{ij}^{int} . \triangle_{ij}^{int} \quad \forall s_j \in \eta_i, \text{ où } \triangle_{ij}^{int} \text{est la différence d'intensité entre les sites } s_i \text{et } s_j. \tag{3.30}$$

Pour chaque site s_i, p_i sites voisins ont contribué à la saillance dans les cinq premières étapes. Les contributions finales sont obtenues en fonction du nombre de voisins pour chaque site s_i :

$$V_{\varsigma}(s_i) = \frac{X_{\varsigma}^i}{p_i} \quad \forall \varsigma \in \{1..5\} \quad \text{où } p_i \text{est le nombre de voisins disponibles dans le voisinage 8-connexe.} \tag{3.31}$$

Les couleurs chaudes constituées de l'intervalle de couleurs rouge, orange et jaune produisent une contribution supplémentaire afin de renforcer leur saillance dans la sixième étape. Cet intervalle de couleur est situé dans les premiers 45° du disque de représentation de la teinte. Cette contribution se formule de la façon suivante :

$$V_6(s_i) = \begin{cases} S(s_i).I(s)_i & \text{pour} \quad 0 \leq H(s_i) < 0,125 \\ 0 & \text{sinon} \end{cases} \tag{3.32}$$

la valeur de la teinte variant entre 0 et 1, un angle de 45° correspond à une valeur de $0,125$.

Finalement, la septième étape est constituée de la contribution liée aux sites ayant une couleur fortement saturée et une intensité lumineuse importante. Ces composantes de couleur du site s_i sont combinées afin de déterminer la contribution pour la dernière étape :

$$V_7(s_i) = S(s_i).I(s_i) \tag{3.33}$$

La saillance spatiale finale est obtenue en combinant les contributions des sept étapes :

$$S^{SP}(s_i) = \frac{1}{7}\sum_{\varsigma=1}^{7} V_\varsigma(s_i) \tag{3.34}$$

Cette carte est ensuite normalisée en fonction de la saillance maximale théorique :

$$S^{SP'}(s_i) = S^{SP}(s_i)/S_{MAX} \tag{3.35}$$

où S_{MAX} est le maximum maximorum théorique de la saillance spatiale et vaut $0,5714$. Ce maximum maximorum est obtenu pour la configuration spatiale suivante : un macrobloc de teinte rouge (0°) entouré de macroblocs de teinte cyan (180°), et dont les valeurs de saturation et d'intensité sont maximales. Bien que, les contrastes de saturation et d'intensité soient nuls dans un tel cas, les autres caractéristiques de couleur influençant l'attention visuelle, énumérées dans la section 3.3.1.1, se manifestent :

- le **contraste de teinte** est maximal : 180° ;
- les **couleurs sont opposées** et cataloguées comme étant respectivement des **couleurs chaude et froide** : angle de 180° entre les teintes rouge (couleur chaude) et cyan (couleur froide) ;
- les valeurs d'**intensité et de saturation sont maximales**.

3.3.2 Saillance temporelle

Dans le chapitre précédent, nous avons conclu que l'aspect temporel et plus particulièrement le contraste de mouvement est primordial dans la modélisation de l'attention visuelle. De plus, pour la détection de zones saillantes d'une séquence d'images projetées sur un écran, il est intéressant d'avoir à l'esprit les règles en vigueur liées à la capture du film. Les mouvements de caméra influencent clairement la stratégie visuelle de l'observateur. La présence ou non de mouvements permet de hiérarchiser les différents évènements. Par ailleurs, la prise de vue est significative du « message » que le metteur en scène souhaite faire passer. Elle incite inconsciemment le téléspectateur à regarder « quelque chose à un endroit particulier ».

En conclusion, l'objectif est de déterminer les zones présentant un contraste de mouvement. À partir des données issues de l'estimation du mouvement global et de la segmentation spatio-temporelle, il est possible de déterminer le contraste de mouvement pour chaque objet et dans notre cas pour chaque tube. Ce contraste de mouvement étant la base de la construction de la carte de saillance temporelle.

3.3.2.1 Mouvement dominant

Afin de réaliser la segmentation spatio-temporelle, nous avons déjà estimé le mouvement global à l'aide du modèle affine à six paramètres décrit à l'équation 3.5, où (V_x,V_y) donne le déplacement d'un point à la position (x,y) en fonction des six paramètres liés au mouvement global. Le modèle affine réduit le nombre de mouvements de la caméra à trois types : les translations (t_x et t_y), les rotations (a_2, a_3) et les zooms (a_1, a_4). Nous avons exactement adapté la méthode de Coudray [CB04] pour estimer ces six paramètres.

Lors de l'estimation du mouvement global, nous avons déterminé les paramètres de translation en localisant le mode maximal de l'histogramme d'accumulation des vecteurs compensés par les paramètres de déformation. Après étude de tous les modes, une segmentation au sens du mouvement, en plus de l'estimation du mouvement global, est effectuée avec l'hypothèse que chaque mode représente le mouvement d'un objet. Cette méthode d'estimation du mouvement global possède cependant un léger défaut. En effet, les paramètres de translation du mouvement global sont détectés à l'aide du mode principal dans l'histogramme d'accumulation. Si la séquence vidéo traitée contient un objet uniforme de taille importante, c'est-à-dire, recouvrant plus de la moitié de l'image, les vecteurs de mouvement de cet objet vont alors être identifiés comme le mode principal dans l'histogramme d'accumulation. Afin de résoudre ce problème et identifier correctement le mouvement apparent dominant de la séquence, les blocs de chaque objet segmenté situés sur le bord de l'image sont comptés. L'objet possédant le plus grand nombre de blocs situés sur le bord de l'image sera identifié comme le fond de la scène. Le vecteur de translation associé à cet objet sera donc identifié comme le mouvement apparent dominant.

3.3.2.2 Mouvement relatif et saillance temporelle

À Partir de la connaissance du mouvement apparent dominant $\overrightarrow{V}_\Theta$ et du déplacement local $\overrightarrow{V}_{local}$ pour chaque site (macrobloc du tube situé sur l'image centrale du segment temporel de neuf images), le mouvement relatif $\overrightarrow{V}_{rel}$, exprimé dans le référentiel rétinien est obtenu simplement par la relation suivante :

$$\overrightarrow{V}_{rel}(s) = \overrightarrow{V}_\Theta(s) - \overrightarrow{V}_{local}(s) \tag{3.36}$$

Le mouvement relatif est nécessaire pour estimer le contraste de mouvement inhérent à un site particulier. Mais il faut prendre en compte d'autres considérations. En effet l'œil est capable de poursuivre des objets en déplacement. Cette faculté liée au mouvement oculaire de poursuite, permet de conserver l'objet suivi dans la fovéa, partie de la rétine présentant la sensibilité spatiale la plus élevée. Par conséquent, considérer directement le mouvement relatif donné par la relation 3.36 serait réducteur. Il n'est pas correct de penser que plus le mouvement relatif est important, plus la saillance est forte. Il faut en réalité prendre en compte la capacité maximale de poursuite de l'œil. S. Daly [Dal98] a montré que la vitesse de poursuite maximale de l'œil pouvait aller jusqu'à $80°/s$. Si la vélocité du mouvement relatif est supérieure à la vélocité maximale de poursuite de l'œil, alors la saillance temporelle est nulle. De plus, celle-ci sera maximale entre $\overrightarrow{v}_1 = 20\,\% \times 30°/s$ et $\overrightarrow{v}_2 = 30°/s$. Pour les vélocités inférieures à $\overrightarrow{v}_1$ et supérieures à $\overrightarrow{v}_2$, la saillance sera obtenue en fonction de la droite affine, définie ci-dessous :

$$S^T(s) = \begin{cases} \frac{1}{7}\overrightarrow{V}_{rel}(s) & \text{si} \quad 0 \leq \overrightarrow{V}_{rel}(s) < \overrightarrow{v}_1 \\ 1 & \text{si} \quad \overrightarrow{v}_1 \leq \overrightarrow{V}_{rel}(s) < \overrightarrow{v}_2 \\ \frac{1}{60}\overrightarrow{V}_{rel}(s) + \frac{8}{5} & \text{si} \quad \overrightarrow{v}_2 \leq \overrightarrow{V}_{rel}(s) < \overrightarrow{v}_{max} \\ 0 & \text{si} \quad \overrightarrow{V}_{rel}(s) \geqslant \overrightarrow{v}_{max} \end{cases} \tag{3.37}$$

où $\overrightarrow{v}_{max} = 80°/s$. L'indice de saillance temporelle obtenu en fonction de la vélocité temporelle est illustré à la figure 3.31.

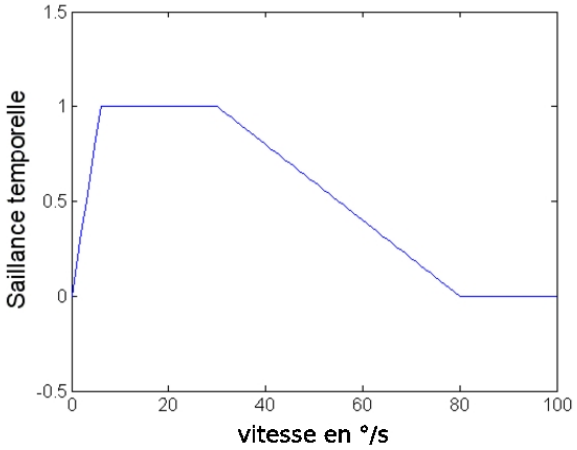


FIG. 3.31 – Fonction représentant la saillance temporelle en fonction de la vitesse relative.

3.3.3 Saillance spatio-temporelle

À partir de la saillance spatiale et de la saillance temporelle, la saillance spatio-temporelle est à déterminer. Les études réalisées par O. Lemeur [LMLCBT06, LMLCB07]] montrent que les observateurs ont tendance à favoriser le centre de l'écran. C'est pourquoi il pondère son modèle de saillance spatiale par une gaussienne bi-dimensionnelle centrée sur l'image. Son étendue spatiale a été optimisée sur une base d'images et sa valeur est de $2,5$ degrés visuels. Lors de nos tests, nous utilisons des séquences vidéos HD. Celles-ci ont une définition maximale de 1080 lignes par 1920 colonnes. C'est pourquoi, nous avons décidé d'utiliser une gaussienne bi-dimensionnelle centrée sur l'image, dont l'étendue spatiale est égale à 5 degrés visuels. La saillance spatio-temporelle est obtenue en combinant la saillance temporelle et la saillance spatiale pondérées par une gaussienne bi-dimensionnelle de la façon suivante :

$$S^{SP-T}(s) = (\frac{1}{2}S^{T}(s) + \frac{1}{2}S^{SP'}(s)) \times gauss2D(s), \tag{3.38}$$

où $gauss2D$ est la gaussienne bi-dimensionnelle d'étendue spatiale égale à 5 degrés visuels. La combinaison des différentes saillances (spatiale et temporelle) proposée ci-dessus est celle que nous avons retenu empiriquement [Fan08].

Finalement, nous obtenons une carte de saillance par segment de neuf images. Ensuite, on projette cette carte sur les images précédentes et suivantes au sein du segment temporel en la compensant à l'aide des informations issues de l'estimation du mouvement et de la segmentation spatio-temporelle.

3.3.4 Résultats qualitatifs

Les figures 3.32 présentent, pour quatre segments temporels des séquences *Tractor*, *New Mobile and Calendar*, *Knightshields* et *Parkrun* les différentes cartes de saillance obtenues.

Pour la séquence *Tractor*, la caméra suit le tracteur en mouvement, en réalisant un léger zoom avant centré sur le véhicule et plus particulièrement sur le semoir fixé à l'arrière de celui-ci. À partir du segment temporel n°50, la caméra réalise un zoom arrière afin d'obtenir une vue d'ensemble du tracteur dans le

champ. Concernant les résultats de la figure 3.32, on constate que la zone la plus saillante est le tracteur. En effet, le mouvement réel de celui-ci est détecté par notre méthode et ainsi, il devient la zone la plus saillante. On observe cependant que la saillance du tracteur n'est pas uniforme. Les caractéristiques spatiales du tracteur sont très hétérogènes en termes de couleur et ce sont ses roues de couleur rouge (couleur chaude) qui sont les plus saillantes.

Concernant les résultats de la figure 3.33 relative à la séquence *New Mobile and Calendar*, les zones les plus saillantes sont également les objets en mouvement. Au départ, seule la caméra est en mouvement, puis le calendrier réalise une translation verticale du haut vers le bas à partir du segment temporel n°34. À partir du segment n°49, le calendrier reste immobile en position haute et seul le train en mouvement (translation horizontale de la droite vers la gauche) est saillant temporellement. Dans le dernier segment temporel, le mouvement de translation verticale (du haut vers le bas) du calendrier reprend et celui-ci devient alors plus saillant. Bien qu'étant spatialement assez homogène, la saillance spatio-temporelle du calendrier n'est pas uniforme. En effet, la gaussienne bi-dimensionnelle utilisée pour produire la favorisation du centre de l'écran par les observateurs, réduit progressivement la saillance des zones éloignées du centre de l'image. De plus, les chiffres écrits en rouge sur le calendrier sont plus saillants que les zones voisines. La figurine orange représentant un tigre sur le train est détectée comme la zone la plus saillante, du fait de sa position (au centre en bas) et de sa couleur (orange).

Pour la séquence *Knightshields*, les résultats présentés à la figure 3.34, révèlent une saillance importante de l'homme se déplaçant. Le fond, bien qu'étant immobile (réellement), est très riche en informations spatiales et est saillant par endroit. Les différents blasons (*shields*) sont plus ou moins saillants en fonction de leur couleur. Les blasons possédant des couleurs chaudes (rouge, orange, jaune) sont des zones saillantes. À partir du segment temporel n°31, l'homme s'immobilise et devient alors moins saillant, puis la caméra devient immobile à son tour à partir du segment n°33 et réalise ensuite un zoom avant sur le blason pointé du doigt par l'homme.

Pour la dernière séquence testée, *Parkrun*, les résultats présentés à la figure 3.35 illustrent la saillance spatio-temporelle de l'homme courant au centre de l'écran. Lorsque celui-ci ralentit à partir du segment temporel n°34 puis s'immobilise au segment n°35, sa saillance temporelle diminue progressivement et devient nulle. La caméra qui effectuait une translation horizontale afin de suivre l'homme, devient également fixe, la saillance spatio-temporelle ainsi obtenue ne reflète alors que les informations spatiales saillantes.

3.4 Temps de calculs

Nous avons mesuré les temps de calculs des différentes étapes de notre méthode de pré-analyse. Le tableau 3.2 récapitule les résultats obtenus avec un ordinateur équipé d'un processeur Intel Xeon cadencé à $3,4GHz$ et de $2GB$ de RAM. La taille de la fenêtre de recherche pour l'estimation multi-résolution du mouvement a été fixée à ± 20 pixels (pour la plus haute résolution). Les durées obtenues exprimées en secondes sont les valeurs moyennes pour traiter un segment temporel de neuf images. L'estimation multi-résolution du mouvement basée sur des tubes spatio-temporels représente environ 90% des temps de calculs de notre méthode de pré-analyse. Nous avons également réalisé une estimation de mouvement avec une méthode de recherche exhaustive pour laquelle la taille de la fenêtre de recherche a été fixée à ± 16 pixels. Pour une séquence vidéo dont la résolution est de 1280×720 pixels, l'estimation du mouvement pour neuf images successives (taille des segments temporels utilisés pour notre méthode de pré-analyse) est réalisée en $566\,s$ contre $72\,s$ en moyenne avec notre méthode d'estimation multi-résolution du mouvement basée

FIG. 3.32 – Cartes de saillance (normalisées pour l'affichage) pour la séquence *Tractor* (segments 1, 19, 51 et 65), avec de haut en bas : images originales (au centre des segments temporels), cartes de saillance spatiale, cartes de saillance temporelle et cartes de saillance spatio-temporelle.

FIG. 3.33 – Cartes de saillance (normalisées pour l'affichage) pour la séquence *New Mobile and Calendar* (segments 1, 34, 49 et 54), avec de haut en bas : images originales (centres des segments), cartes de saillance spatiale, cartes de saillance temporelle et cartes de saillance spatio-temporelle.

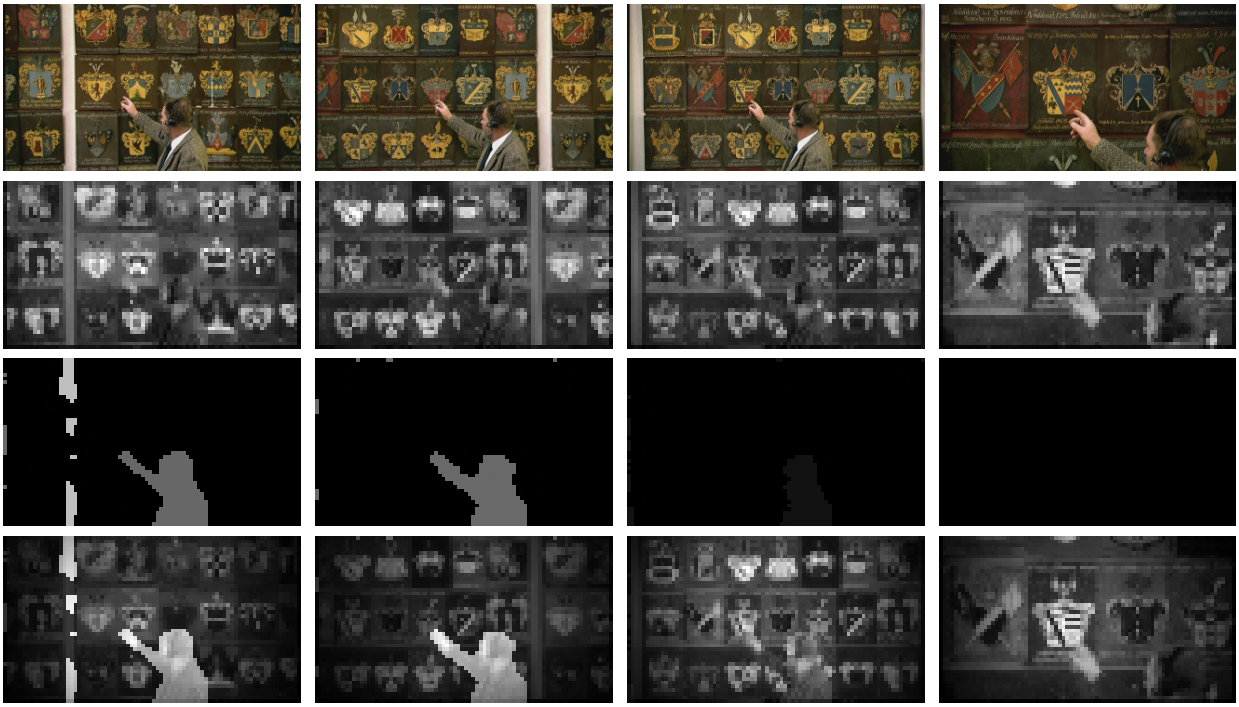


FIG. 3.34 – Cartes de saillance (normalisées pour l'affichage) pour la séquence *Knightshields* (segments 1, 20, 34 et 55), avec de haut en bas : images originales (centres des segments), cartes de saillance spatiale, cartes de saillance temporelle et cartes de saillance spatio-temporelle.

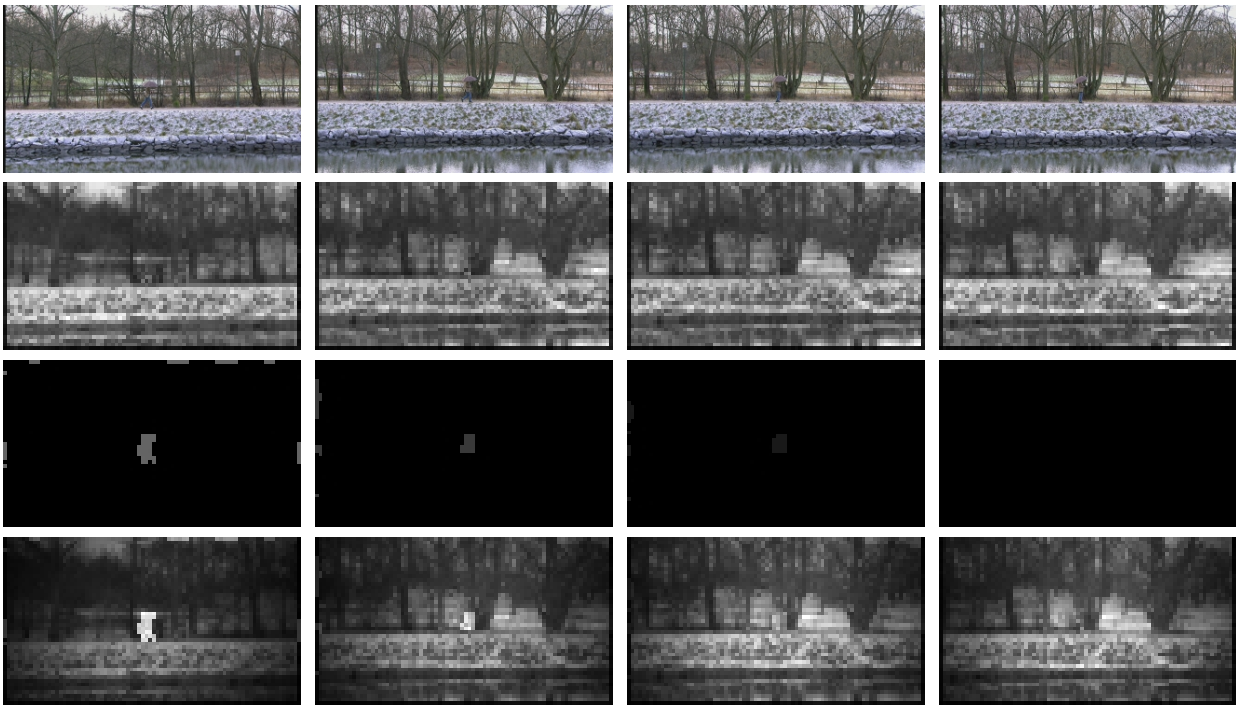


FIG. 3.35 – Cartes de saillance (normalisées pour l'affichage) pour la séquence *Parkrun* (segments 4, 34, 35 et 54), avec de haut en bas : images originales (centres des segments), cartes de saillance spatiale, cartes de saillance temporelle et cartes de saillance spatio-temporelle.

Séquence	Estimation du mouvement (tubes)	Segmentation spatio-temporelle	Détermination des cartes de saillance	Total
Knightshields (1280×720)	75,93 s	6,51 s	0,2 s	82,64 s
New Mobile and Calendar (1280×720)	79,05 s	6,67 s	0,36 s	86,09 s
Parkrun (1280×720)	62,83 s	5,52 s	0,37 s	68,72 s
Tractor (1920×1080)	206,63 s	27,43 s	3,08 s	237,14 s

TAB. 3.2 – Temps de calculs (valeurs moyennes) des différentes étapes de notre méthode de pré-analyse pour un segment temporel de neuf images.

sur des tubes spatio-temporels. Et pour une vidéo dont la résolution est de 1920×1080 pixels, l'estimation du mouvement pour neuf images successives est réalisée en $1277\,s$ contre $207\,s$ en moyenne avec notre méthode. Bien que notre méthode d'estimation du mouvement soit la plus coûteuse en termes de temps de calculs (90% des temps de calculs de la pré-analyse), celle-ci est tout de même plus de six fois plus rapide qu'une méthode de recherche exhaustive.

Conclusion

L'objectif de ce chapitre était de présenter notre méthode d'analyse spatio-temporelle des séquences vidéo. Afin de détecter les objets en mouvement qui sont importants visuellement, nous avons réalisé une segmentation spatio-temporelle de la vidéo. Tout d'abord, nous réalisons une estimation multi-résolution du mouvement basée sur des tubes spatio-temporels pour une fenêtre temporelle de neuf images, correspondant à une durée de $180\,ms$. Le temps de fixation du SVH étant sensiblement égal à $200\,ms$, notre estimation de mouvement permet de suivre le mouvement pour une durée significative perceptuellement. Avant de réaliser une segmentation basée sur le mouvement des tubes spatio-temporels, nous estimons le mouvement global de la séquence à partir du champ de vecteurs obtenu (un vecteur de mouvement par tube spatio-temporel) par accumulation au sein d'histogrammes. Une fois le mouvement global connu et compensé, nous réalisons une segmentation basée mouvement de la séquence vidéo. Nous proposons ensuite une modélisation markovienne pour affiner cette segmentation spatio-temporelle en intégrant de nouveaux critères tels que la corrélation spatiale, la couleur, la texture, le mouvement et la corrélation temporelle entre les sites. Cette modélisation markovienne permet d'une part d'améliorer les premiers résultats obtenus avec notre segmentation basée sur le mouvement et d'autre part d'assurer le suivi des régions entre les différents segments temporels. La dernière étape de notre méthode de pré-analyse propose une modélisation de notre attention visuelle pré-attentive sous la forme de cartes de saillance. La détermination de celles-ci est basée sur les contrastes de couleur et le mouvement relatif des objets, et elles mettent en évidence les zones susceptibles d'exciter notre attention visuelle. La construction de la carte de saillance spatio-temporelle est obtenue par une fusion de la carte de saillance spatiale et de la carte de saillance temporelle.

Notre méthode de pré-analyse de la vidéo nous permet d'obtenir des informations de haut niveau sur la séquence vidéo. La prochaine étape est maintenant d'exploiter ces résultats et interagir avec le codeur vidéo pour réaliser un codage cohérent en fonction de ces informations et donc du contenu de la vidéo. Dans la suite de cette thèse, après avoir détaillé le codeur H.264 et étudié les approches proposées pour optimiser

le codage, nous présenterons notre contribution au sein du codeur permettant d'exploiter les résultats de la pré-analyse.

Deuxième partie

Applications au codage

Chapitre 4

Le codeur H.264 et ses modes d'optimisations

Sommaire

Introduction 111
4.1 Description du codeur H.264 112
4.1.1 Introduction 112
4.1.2 Structure du codeur H.264 112
4.1.3 Prédiction inter-image 115
4.1.4 Prédiction intra-image 118
4.1.5 Filtre anti effet de bloc 120
4.1.6 Transformation et quantification 121
4.1.7 Conclusion 121
4.2 Optimisations du codeur H.264 122
4.2.1 Introduction 122
4.2.2 Accélération de l'estimation de mouvement 123
4.2.3 Optimisation de la prédiction intra-image 130
4.2.4 Optimisation de la prédiction inter-image 132
4.2.5 Recherche réduite pour les multiples images référence 135
4.2.6 Optimisation de la qualité 138
4.2.7 Conclusion 138
Conclusion 139

Introduction

Le chapitre précédent a présenté notre méthode de pré-analyse de la vidéo. Ainsi, nous disposons d'informations sur le contenu de la vidéo, telles que les zones susceptibles d'attirer notre attention (obtenues à partir des cartes de saillance spatio-temporelle) et le cycle de vie des objets (segmentation spatio-temporelle). L'objectif suivant est de pouvoir utiliser ces différentes informations au sein d'un codeur vidéo. En effet, comme nous l'avons évoqué dans le chapitre 1, un codeur vidéo classique ne dispose d'aucune information a priori sur le contenu de la vidéo et son évolution temporelle. Ainsi, ses adaptations ne reposent en

pratique que sur des décisions à court terme basées, concernant la prédiction avec compensation de mouvement, sur la minimisation de l'erreur de prédiction. Le dernier standard de codage vidéo adopté en Europe pour la diffusion de la Télévision Haute Définition (TVHD), à savoir H.264 (appelé encore MPEG-4/AVC ou même MPEG-4 Part 10), qui vise à gagner jusqu'à 50% de la bande passante actuellement utilisée par MPEG-2 pour une qualité équivalente, souffre des mêmes défauts que les codeurs vidéo classiques. À partir des informations haut niveau dont nous disposons, nous souhaitons guider le codeur H.264 dans ses prises de décisions afin qu'il produise un codage plus cohérent temporellement et améliore la qualité perceptuelle de la vidéo reconstruite pour un débit équivalent. Avant de transmettre de telles indications, nous devons disposer d'informations sur les modes de fonctionnement du codeur et plus particulièrement sur les outils offrant une liberté de codage et susceptibles d'être pertinents pour notre approche.

La première partie de ce chapitre présente la norme H.264, plus particulièrement les techniques de prédiction intra et inter images à partir de multiples images référence. Dans la deuxième partie, nous intéressons aux travaux déjà effectués pour optimiser les performances du codeur H.264.

4.1 Description du codeur H.264

4.1.1 Introduction

En 1998, VCEG (*Video Coding Experts Group*) se lance dans un projet appelé H.26L, dont le but est de doubler l'efficacité du codage vidéo par rapport à n'importe quelle norme existante alors. En 2001, MPEG (*Moving Picture Expert Group*) rejoint VCEG et le JVT (*Joint Video Team*) est créé pour concrétiser la nouvelle norme. En 2003, la norme H.264/MPEG-4 Part 10 est publiée [**?**]. L'objectif de H.264/AVC (*Advanced Video Coding*) est un codage efficace et robuste de la vidéo et son transport. Les applications sont diverses : la communication vidéo (vidéoconférence et vidéotéléphonie), le codage haute qualité pour la diffusion TV et la vidéo en temps réel sur les réseaux à transmission par paquets (Internet). Ce chapitre présente les principales caractéristiques de la norme H.264. L'idée n'est pas de réaliser une description exhaustive du standard (pour cela le lecteur est invité à consulter le livre écrit par Richardson [Ric03]), mais après une description sommaire, de se concentrer sur les techniques que nous pourrons exploiter après cette pré-analyse. Les autres principales caractéristiques de la norme H.264 sont détaillées en annexe B.

4.1.2 Structure du codeur H.264

Cette section présente la structure de la norme H.264. Avant de détailler le codec et les profils de compression, il est nécessaire de définir certains éléments et termes qui seront employés.

4.1.2.1 Terminologie

La norme H.264 emploie une terminologie particulière que nous allons préciser [Ric03] :

Une trame (*field*) d'une vidéo entrelacée ou une image (*frame*) d'une vidéo progressive est codée afin de produire une image codée. Un numéro lui est assigné (signalé dans le flux binaire), qui ne correspond pas forcément à l'ordre de décodage. Les images précédemment codées, les images de référence, sont utilisées pour la prédiction inter-image. Ces images de référence sont organisées en une ou deux listes : la liste 0 et la liste 1.

Une image codée est composée d'un certain nombre de macroblocs, par exemple pour le format 4:2:0 chacun d'eux contient 16×16 échantillons de luminance et deux ensembles de 8×8 échantillons, chaque

ensemble étant associé à l'une des deux composantes de chrominance. Les macroblocs sont arrangés en tranches (*slice* en anglais), chaque tranche contenant un nombre de macroblocs compris entre un (un macrobloc par tranche) et le nombre total de macroblocs d'une image (une tranche par image). H.264 utilise comme les codeurs vidéo déjà existants (voir chapitre 1), plusieurs types d'images : des images de type I (images codées en mode intra-image), des images de type P (images prédites) et des images de type B (images possiblement bi-prédites)[1]. Une tranche I ne contient que des macroblocs I, une tranche P peut contenir des macroblocs I et P, et une tranche B peut contenir des macroblocs B et I. Il existe également des tranches SI et SP, dont la notion est explicitée dans la suite du chapitre à la section 4.1.2.4.

Les macroblocs I sont obtenus par prédiction intra-image à partir d'échantillons déjà décodés dans la tranche courante. Une prédiction est formée soit pour un macrobloc complet, soit pour chaque bloc 4×4 de luminance (et les blocs associés de chrominance) du macrobloc.

Les macroblocs P sont obtenus par prédiction inter-image à partir des images de référence. La prédiction se fait depuis une image de référence.

Les macroblocs B sont obtenus par prédiction inter-image à partir de une ou deux images de référence. Chaque partition de macrobloc peut utiliser une ou deux images de référence, une image de la liste 0 et/ou une image de la liste 1.

Un GOP (ou groupe d'images) est un ensemble d'images consécutives constituant une séquence. Il commence par une image I (ne contenant que des tranches I), suivie d'images P et/ou B. Le prochain GOP commence à l'image I suivante.

Ces quelques définitions nous permettent maintenant de décrire le codec H.264.

4.1.2.2 Le codec H.264

Tout comme les autres standards de codage vidéo, H.264 [WSBL03] ne définit pas un codec mais la syntaxe d'un flux binaire codé de vidéo ainsi que la procédure de décodage de ce flux. En pratique, un codeur/décodeur conforme à la norme inclut les fonctionnalités décrites sur les figures 4.1 et 4.2.

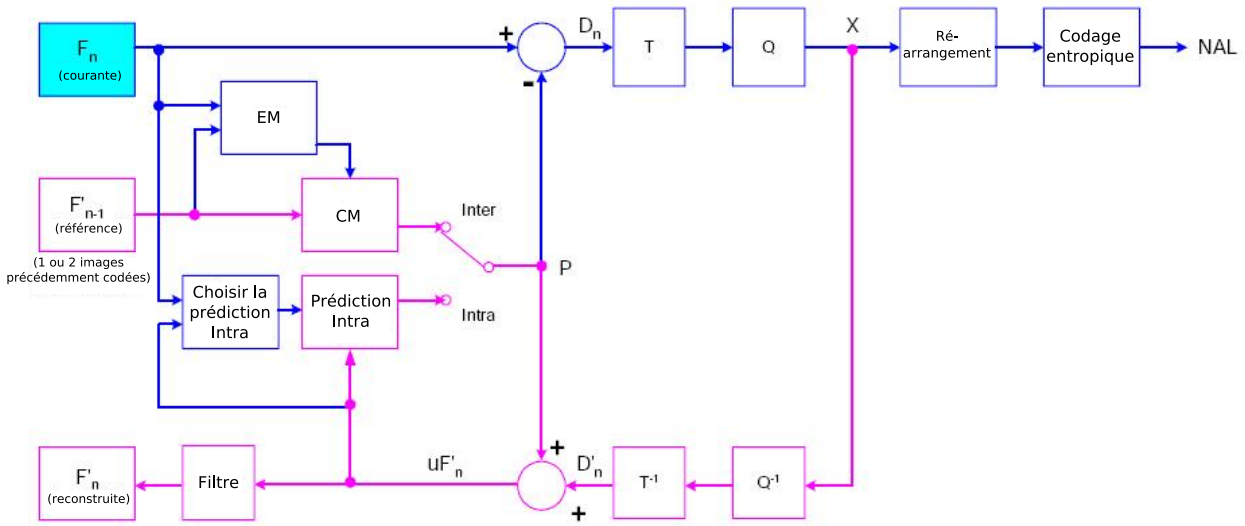


FIG. 4.1 – Schéma du codeur H.264.

Le codeur (figure 4.1) inclut deux chemins pour le flux de données, le chemin « avant » (de gauche à droite) et le chemin de reconstruction (de droite à gauche). Sur la figure 4.2, le chemin de données du

[1] Pour simplifier, nous n'utiliserons désormais plus le terme « type X » mais « X ».

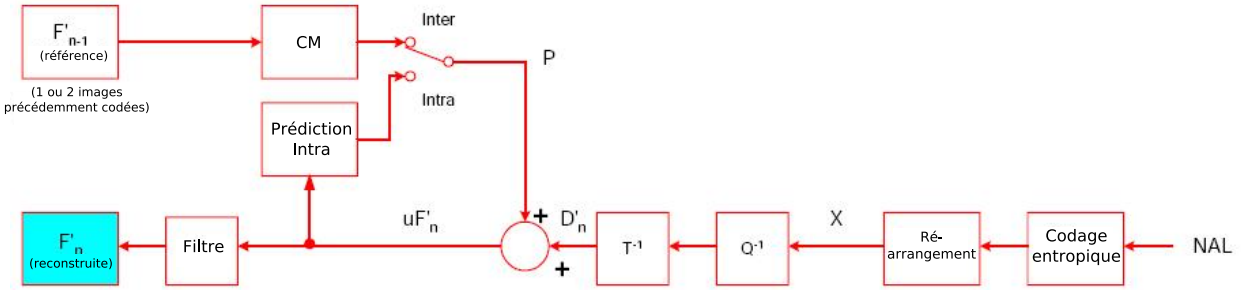


FIG. 4.2 – Schéma du décodeur H.264.

décodeur est présenté de droite à gauche, indiquant ainsi les similarités avec le codeur.

D'après la figure 4.1, sur le chemin avant, une trame ou une image F_n est partitionnée en macroblocs, une prédiction P est formée en fonction des échantillons reconstruits. Pour le mode Intra, P est obtenue à partir d'échantillons de la tranche courante ayant déjà été codés, décodés et reconstruits (uF'_n sur la figure 4.1 ; notons que les échantillons utilisés ne sont pas filtrés). En mode Inter, P est obtenue par prédiction par compensation de mouvement à partir d'une image de référence sélectionnées dans la liste 0 et/ou la liste 1. Sur les figures 4.1 et 4.2, l'image de référence est l'image précédemment codée F'_{n-1}, mais cette référence pourrait être aussi choisie parmi des images passées ou futures, ayant été codées, décodées et reconstruites.

La prédiction P est ensuite soustraite au bloc courant, générant un bloc résiduel D_n. Celui-ci est alors transformé (T) et quantifié (Q) afin de produire X, un ensemble de coefficients de transformation quantifiés. Ils sont réarrangés puis transmis au codeur entropique. Les coefficients obtenus et les informations nécessaires au décodage (mode de prédiction, table de quantification, vecteurs de mouvement...) sont codés en un flux binaire comprimé qui est passé au NAL (*Network Abstraction Layer*) pour transmission ou codage adapté aux caractéristiques du canal de transmission.

En plus du codage et de la transmission, le codeur exécute également un décodage suivant le chemin de reconstruction. Cette étape est nécessaire pour fournir une référence pour les futures prédictions. Les coefficients X subissent un déquantification (Q^{-1}) et une transformée inverse (T^{-1}) pour produire un bloc des différences D'_n. La prédiction P est ajoutée à D'_n pour créer le bloc reconstruit uF'_n, c'est-à-dire, la version décodée du bloc original. Enfin un filtre est appliqué afin de réduire les effets de bloc. L'image de référence reconstruite est ainsi créée par une série de blocs F'_n.

Au niveau du décodage, le même procédé est appliqué. Les macroblocs reçus par le NAL sont décodés, réarrangés pour obtenir les coefficients X. Déquantification et transformée inverse sont appliquées pour obtenir D'_n (identique au D'_n du schéma d'encodage). Grâce aux informations d'en-tête du flux binaire, le décodeur crée la même prédiction P que celle du codeur, ajoute D'_n pour produire uF'_n. Celle-ci est alors filtrée pour générer chaque bloc décodé F'_n.

Cette description du codage/décodage H.264 se veut générale. En effet, plusieurs possibilités existent dans l'utilisation de ces fonctions, ce sont les profils. Ceux-ci sont décrits en annexe B.1.

4.1.2.3 Les images de référence

Un codeur H.264 peut utiliser une ou deux images d'un ensemble d'images qui ont été préalablement codées, en tant que référence pour la prédiction par compensation de mouvement de chaque macrobloc ou

partition de macrobloc codé en inter. Cela permet au codeur de rechercher le meilleur correspondant pour la partition de macrobloc à partir d'un ensemble plus large d'images, plutôt qu'avec seulement l'image précédemment encodée.

Le codeur et le décodeur maintiennent chacun une ou deux listes d'images référence, contenant des images qui ont été déjà codées et décodées. Les macroblocs codés en inter et les partitions de macroblocs dans les tranches P, sont prédits à partir d'une seule liste : la **liste 0**. Les macroblocs codés en inter et les partitions de macroblocs dans les tranches B peuvent être prédits à partir de deux listes : la **liste 0** et la **liste 1**.

4.1.2.4 Les tranches

Une image vidéo est découpée et codée en une ou plusieurs tranches. Le nombre de macroblocs par tranche n'a pas besoin d'être constant à l'intérieur d'une image. L'inter-dépendance entre des tranches codées est minimale, ce qui peut contribuer à limiter la propagation des erreurs. Il existe cinq types de tranches au sein du codeur H.264 (tableau 4.1) et une image codée peut être composée de tranches de différents types. Par exemple, une image codée avec le profil de base peut contenir un mélange de tranches I et P, alors qu'une image codée avec le profil principal ou étendu peut contenir un mélange de tranches I, P et B.

Type de tranche	**Description**	**Profil(s)**
I (Intra)	Contient seulement des macroblocs I (chaque bloc ou macrobloc est prédit à partir de données précédemment codées à l'intérieur de la même tranche).	Tous
P (Prédite)	Contient des macroblocs P (chaque macrobloc ou partition de macrobloc est prédit à partir d'une image référence de la liste 0) et/ou des macroblocs I.	Tous
B (Bi-prédite)	Contient des macroblocs B (chaque macrobloc ou partition de macrobloc est prédit à partir d'images référence de la liste 0 et/ou de la liste 1) et/ou des macroblocs I.	Étendu et principal
SP (*Switching P*)	Facilite le changement entre les flux codés ; contient des macroblocs P et/ou I.	Étendu
SI (*Switching I*)	Facilite le changement entre les flux codés ; contient des macroblocs SI (un type spécial de macrobloc codé en intra).	Étendu

TAB. 4.1 – Les tranches dans H.264.

4.1.3 Prédiction inter-image

Le standard H.264 permet d'utiliser une ou deux images de référence pour la prédiction inter-images. La prédiction peut se faire à partir du passé, du futur ou de façon bi-directionnelle. Pour cela, le codeur et le décodeur maintiennent une ou deux listes d'images de référence [WZG99], selon qu'il s'agisse d'une tranche P ou d'une tranche B. Les techniques mises en œuvre dans la prédiction inter-images sont présentées par la suite.

4.1.3.1 Les tranches P

La prédiction inter crée un modèle de prédiction à partir d'une ou plusieurs images de référence en faisant de la compensation de mouvement par bloc. H.264 inclut le support d'une large gamme de tailles de

blocs et une meilleure précision pour les vecteurs de mouvement, à la différence des standards précédents.

4.1.3.1.1 Compensation de mouvement à structure d'arbre

Chaque macrobloc 16×16 peut être découpé de quatre manières différentes (voir la figure 4.3}) et la compensation de mouvement peut s'appliquer sur une partition 16×16, deux partitions 16×8, deux partitions 8×16 ou quatre partitions 8×8. En choisissant le mode 8×8, les quatre sous-macroblocs peuvent encore une fois être divisés (voir la figure 4.4) en une partition 8×8, deux partitions 8×4, deux partitions 4×8 ou quatre partitions 4×4. Ces partitions et sous-macroblocs permettent un grand nombre de combinaisons pour chaque macrobloc. C'est la méthode dite de compensation de mouvement à structure d'arbre.

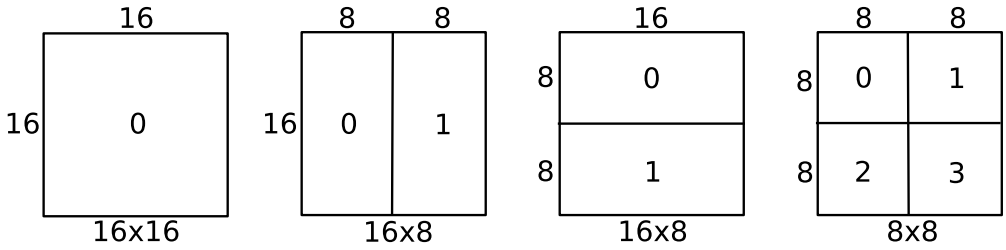


FIG. 4.3 – Partitions de macroblocs : 16×16, 8×16, 16×8 et 8×8.

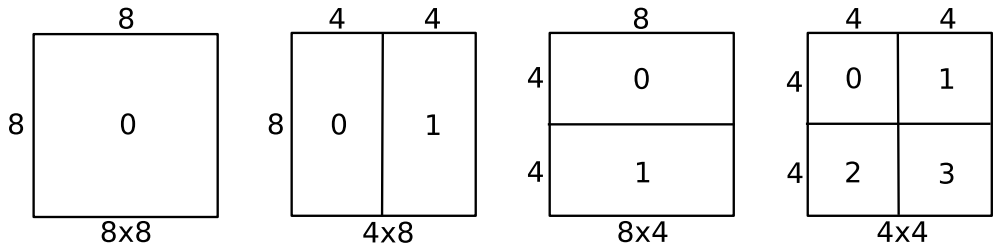


FIG. 4.4 – Partitions de sous-macroblocs :8×8, 4×8, 8×4 et 4×4.

Un vecteur de mouvement est nécessaire pour chaque partition ou chaque sous-macrobloc. Ce vecteur doit être codé et transmis et le choix de la partition doit être codé dans le train binaire. Une grande partition (16×16, 8×16 ou 16×8) ne nécessitera que peu de bits pour coder les vecteurs de mouvement et le choix de la partition. En revanche le résidu (erreur de prédiction) obtenu pourra contenir une « grande énergie » dans les zones très riches. Une petite partition (8×8, 4×8, 8×4 ou 4×4) diminuera l'énergie résiduelle, mais nécessitera plus de bits pour coder tous les vecteurs et les choix de partition. Ainsi, le choix de la taille de partition a un impact important sur les performances de compression. En règle générale, une grande taille est appropriée pour les zones homogènes et une petite pour les zones de détails.

Pour le format 4:2:0, chaque composante de chrominance (C_b et C_r) d'un macrobloc a une résolution moitié moindre que celle de la luminance. Chaque bloc de chrominance est partitionné selon la même méthode décrite précédemment, avec une diminution de la taille de deux fois deux (à une partition 8×16 de luminance correspond une partition 4×8 de chrominance). Les composantes de chaque vecteur de mouvement sont alors divisées par deux pour les blocs de chrominance.

4.1.3.1.2 Les vecteurs de mouvement

Chaque partition ou sous-partition d'un macrobloc codé en inter est prédite à partir d'une zone de même taille d'une image référence. La différence de position entre les deux zones (le vecteur de mouvement) a

une résolution du quart d'échantillon pour la luminance et donc du huitième d'échantillon pour la chrominance. Il est donc nécessaire pour les composantes non entières du vecteur de mouvement d'effectuer un ré-échantillonnage par interpolation à partir des voisins déjà codés. Cette interpolation se fait par étapes et utilise différentes techniques. Pour plus d'informations sur celles-ci, le lecteur est invité à se référer à l'article écrit par Wedi [Wed03] et le livre écrit par Richardson [Ric03].

4.1.3.1.3 Prédiction des vecteurs de mouvement

Coder un vecteur de mouvement pour chaque partition coûterait un nombre de bits significatif, surtout pour des partitions de petites tailles. Or les vecteurs de mouvement de partitions voisines sont souvent très corrélés. Ainsi, chaque vecteur de mouvement est prédit à partir des vecteurs des partitions voisines déjà codées. Un vecteur prédit MV_p est calculé à partir des vecteurs de mouvement précédemment obtenus, et la différence MVD entre le vecteur courant et le vecteur prédit est codée et transmise. La méthode de prédiction d'un MV_p dépend de la taille de la partition pour la compensation de mouvement et de la disponibilité des vecteurs voisins. Le détail de la prédiction du vecteur de mouvement est donné en annexe B.3.

4.1.3.2 Les tranches B

L'utilisation de tranches B est une possibilité supplémentaire disponible avec le profil principal, ainsi que la prédiction pondérée ou le codage entropique CABAC.

Pour les tranches B, chaque partition d'un macrobloc peut être prédite à partir d'une ou deux images de référence, situées temporellement avant ou après l'image courante. Suivant les images de référence stockées dans le décodeur et le codeur, plusieurs options deviennent possibles dans le choix des images de référence. La figure 4.5 illustre trois exemples : (a) une référence passée et une référence future, (b) deux références passées et (c) deux références futures.

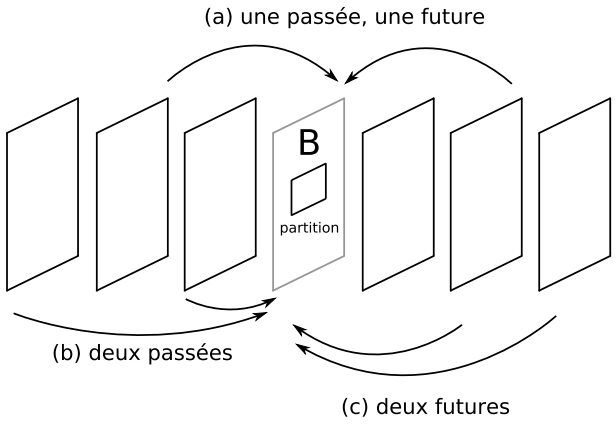


FIG. 4.5 – Exemples de prédiction pour un macrobloc de type B : (a) précédente/future, (b) précédentes, (c) futures.

4.1.3.2.1 Options de prédiction

Au contraire des normes précédentes, le concept des tranches B est généralisé dans H.264. La différence avec les tranches P est que les partitions de macroblocs d'une tranche B peuvent utiliser une moyenne pondérée de deux valeurs de prédiction à compensation de mouvement distinctes pour générer la prédiction. Les partitions de macroblocs dans une tranche B sont prédites à partir de l'une de ces méthodes, le mode direct, la prédiction à compensation de mouvement à partir d'une image référence de la liste 0, la prédiction à

compensation de mouvement à partir d'une image référence de la liste 1, ou la bi-prédiction à compensation de mouvement à partir d'images de référence des listes 0 et 1. Les différents mode des prédictions des tranches B sont détaillés en annexe B.4. Différents modes de prédiction peuvent être choisis pour chaque partition (tableau 4.2). Cependant dans le cas d'une partition 8×8, le mode de prédiction choisi sera appliqué à toutes les sous-partitions de celle-ci. La figure 4.6 montre deux exemples de combinaison de modes de prédiction. Sur la gauche, deux partitions 16×8, qui utilisent respectivement la liste 0 et la bi-prédiction et sur la droite, quatre partitions 8×8, qui utilisent respectivement le mode direct, la liste 0, la liste 1 et la bi-prédiction.

Partition	**Options**
16×16	Directe, liste 0, liste 1 ou bi-prédiction.
16×8 ou 8×16	Liste 0, liste 1 ou bi-prédiction (choisie séparément pour chaque partition).
8×8	Directe, liste 0, liste 1 ou bi-prédiction (choisie séparément pour chaque partition).

TAB. 4.2 – Options de prédiction pour les macroblocs dans les tranches B.

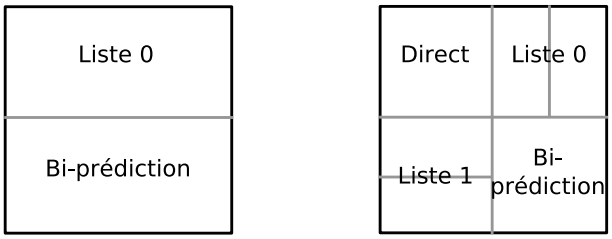


FIG. 4.6 – Exemples de modes de prédiction pour les macroblocs dans les tranches B.

Nous venons de détailler les techniques de prédiction de mouvement à compensation de mouvement dans les tranches de type P et B. Nous présentons maintenant la prédiction intra-image qui permet de coder les images de type I.

4.1.4 Prédiction intra-image

À l'inverse de la prédiction temporelle, la prédiction intra n'utilise pas d'image de référence. Elle exploite la redondance spatiale d'une image, c'est-à-dire, le fait que les pixels dans une image sont spatialement corrélés. Le codeur H.264 a la particularité de travailler dans le domaine spatial, se référant aux échantillons voisins de blocs déjà codés et non dans le domaine transformé comme c'est le cas pour les codeurs MPEG-2 et MPEG-4. La prédiction dépend bien sûr de la taille de la partition considérée. Il existe donc plusieurs méthodes en fonction du bloc : H.264 permet la prédiction intra de blocs 4×4 ou 16×16 de luminance et de blocs 8×8 de chrominance. Le codeur sélectionne le mode pour chaque bloc qui minimise la différence entre le bloc prédit P et le bloc courant.

4.1.4.1 Prédiction des blocs 4×4 de luminance

La prédiction de blocs 4×4 de luminance est une spécificité de H.264 par rapport aux autres standards qui utilisent des blocs 8×8. Cela apporte une meilleure précision dans la prédiction mais génère (ou nécessite) plus de calculs. Neuf modes de prédiction sont possibles pour les blocs 4×4 selon la direction

de prédiction. La figure 4.7 présente la façon d'étiqueter les pixels. Sur la figure 4.8, qui présente les neuf modes, les flèches indiquent la direction de prédiction. Pour les modes 3 à 8, les pixels prédits sont formés par une moyenne pondérée des pixels [A-M]. Par exemple pour le mode 4 (cf figure 4.8), le pixel d sur la figure 4.7 est prédit par l'arrondi de $(B/4+C/2+D/4)$. Le tableau 4.3 précise chaque mode [Ric03].

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

FIG. 4.7 – Étiquetage des échantillons de prédiction 4×4.

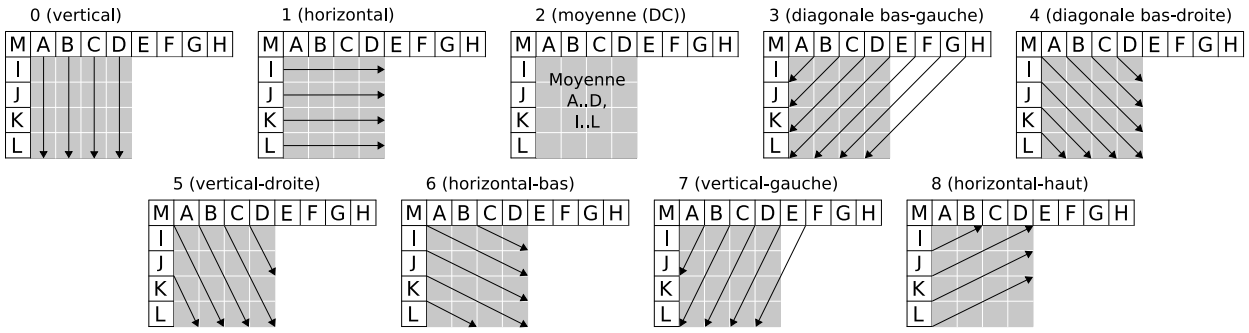


FIG. 4.8 – Les modes de prédiction des blocs 4×4 de luminance.

4.1.4.2 Prédiction des blocs 16×16 de luminance

La prédiction des blocs4×4 est plus adaptée pour les zones de détails, mais nécessite plus de calculs. Le codeur H.264 permet également la prédiction de macroblocs entiers de luminance (16×16). Quatre modes sont alors possibles, comme le montre la figure 4.9. Le tableau 4.4 précise les calculs réalisés.

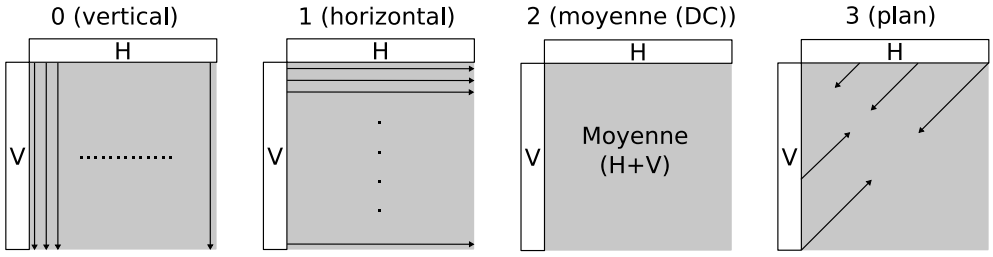


FIG. 4.9 – Les modes de prédiction des blocs 16×16 de luminance.

Mode	**Description**
Mode 0 (vertical)	Extrapolation à partir des échantillons supérieurs (H).
Mode 1 (horizontal)	Extrapolation à partir des échantillons gauches (V).
Mode 2 (DC)	Moyenne de H et V.
Mode 3 (plan)	Une fonction linéaire plane est ajustée à H et V. Cela fonctionne bien sur les zones de luminance variant doucement.

TAB. 4.4 – Les modes de prédiction des blocs 16×16 de luminance [Ric03].

Mode	Description
Mode 0 (vertical)	Les échantillons supérieurs A, B, C et D sont extrapolés verticalement.
Mode 1 (horizontal)	Les échantillons gauches I, J, K et L sont extrapolés horizontalement.
Mode 2 (DC)	Tous les échantillons [a-p] sont prédits par la moyenne de [A-D] et [I-L].
Mode 3 (diagonale bas-gauche)	Les échantillons sont interpolés avec un angle de 45˚entre le bas-gauche et le haut-droit.
Mode 4 (diagonale bas-droite)	Les échantillons sont interpolés avec un angle de 45˚vers le bas et vers la droite.
Mode 5 (vertical-droit)	Extrapolation avec un angle de 26,6˚à gauche de la verticale ($largeur/hauteur = 1/2$).
Mode 6 (horizontal-bas)	Extrapolation avec un angle de 26,6˚en-dessous de l'horizontale.
Mode 7 (vertical-gauche)	Extrapolation (ou interpolation) avec un angle de 26,6˚à droite de la verticale.
Mode 8 (horizontal-haut)	Interpolation avec un angle de 26,6˚au-dessus de l'horizontale.

TAB. 4.3 – Les modes de prédiction des blocs 4×4 de luminance.

4.1.4.3 Prédiction des blocs de chrominance

Chaque bloc 8×8 de chrominance est prédit à partir des échantillons de chrominance déjà codés au-dessus et à gauche. Les deux composantes C_b et C_r utilisent toujours la même prédiction. Quatre modes de prédiction existent, similaires aux modes 16×16 de luminance (voir la figure 4.9). Le tableau 4.5 présente ces différents modes.

Mode	Description
Mode 0 (DC)	Moyenne de H et V.
Mode 1 (horizontal)	Extrapolation à partir des échantillons gauches (V).
Mode 2 (vertical)	Extrapolation à partir des échantillons supérieurs (H).
Mode 3 (plan)	Une fonction linéaire plane est ajustée à H et V.

TAB. 4.5 – Les modes de prédiction des blocs 8×8 de chrominance.

4.1.5 Filtre anti effet de bloc

Nous venons de décrire les procédé de codage intra des images, utilisant plusieurs modes. Une étape supplémentaire, introduite dans le standard, est appliquée à tous les macroblocs avant reconstruction : il s'agit d'un filtrage anti effet de bloc [LJL+03]. Le filtrage anti effet de bloc est appliqué après la transformation inverse au niveau du codeur (voir figure 4.1) et du décodeur (voir figure 4.2). Le filtre lisse les frontières des blocs améliorant l'apparence des images décodées. L'image filtrée est utilisée pour la prédiction à compensation de mouvement des images futures. Les performances de compression peuvent être améliorées si l'image filtrée est plus « proche » de l'image originale que l'image non filtrée comportant des effets de blocs au niveau des frontières des macroblocs. La force du filtre dépend de l'index de quantification courant, des modes de codage des blocs voisins, des vecteurs de mouvement et du gradient des données de l'image situées de part et d'autre de la frontière.

La partie prédiction du standard H.264 est maintenant complète, les prochaines étapes sont la transfor-

mation et la quantification.

4.1.6 Transformation et quantification

Après soustraction de la prédiction au bloc original, le résidu est transformé puis quantifié, comme c'est le cas avec les autres standards. Cependant H.264/AVC apporte différentes nouveautés.

4.1.6.1 Transformation

Le codeur H.264 utilise trois transformées [MHKK03] selon le type de données résiduelles à coder : une transformée pour les matrices 4×4 des coefficients des composantes continues de luminance des macroblocs intra (prédits en mode 16×16), une transformée pour les matrices des coefficients des composantes continues de chrominance (pour tous les macroblocs) et une transformée basée sur la TCD (DCT en anglais, voir chapitre 1) pour tous les autres blocs 4×4. Cette dernière est une transformée en entiers séparable. La transformée inverse est réalisée par des opérations exactes sur des entiers. Ceci évite les discordances de transformée inverse. Pour plus de précision, le lecteur pourra se référer au livre de Richardson [Ric03].

4.1.6.2 Quantification

Après transformation des données résiduelles en coefficients, une quantification est appliquée. Cette opération entraîne des pertes mais permet une compression des données importante. Le codeur H.264 utilise une quantification scalaire. Les calculs de la quantification et de son inverse sont complexes du fait de contraintes : éviter les divisions et les nombres à virgule flottante, et incorporer des matrices de redimensionnement.

L'opération basique de quantification est la suivante :

$$Z_{ij} = arrondi(\frac{Y_{ij}}{Q_{step}}),$$

où Y_{ij} est un coefficient de la transformation décrite précédemment, Q_{step} est un pas de quantification et Z_{ij} est le coefficient quantifié. 52 valeurs de Q_{step} sont permises par la norme, et indexées par le paramètre de quantification QP (tableau 4.6). Q_{step} double de taille pour chaque augmentation d'un facteur 6 de QP. La vaste gamme des pas de quantification permet au codeur de contrôler avec précision et souplesse le compromis entre débit et qualité. Les valeurs de QP peut être différentes pour la luminance et la chrominance. De plus, contrairement aux codeurs vidéo existants, H.264 offre la possibilité d'utiliser un pas de quantification par macrobloc. Ainsi, il devient possible de compresser différemment les macroblocs en fonction de leur saillance ou de leur activité spatio-temporelle (effets de masquage).

La dernière étape du codeur permettant d'obtenir un gain important en termes de compression, à savoir le codage entropique, est détaillée en annexe B.5.

4.1.7 Conclusion

Cette section a traité du nouveau standard H.264. Nous avons détaillé, à partir d'une terminologie définie en premier lieu, les différentes étapes de compression. Les nombreuses étapes de prédiction permettent d'améliorer la compression et de faire de H.264 a priori un meilleur codec (50% de débit en moins par rapport à MPEG-2 pour une même qualité [WSBL03]). De plus, ces techniques de prédiction ainsi que la possibilité de modifier le pas de quantification pour chaque macrobloc offre des opportunités en terme

QP	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Q_{step}	0,625	0,6875	0,8125	0,875	1	1,125	1,25	1,375	1,625	1,75	2
QP	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**
Q_{step}	2,25	2,5	2,75	3,25	3,5	4	4,5	5	5,5	6,5	7
QP	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
Q_{step}	8	9	10	11	13	14	16	18	20	22	26
QP	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**
Q_{step}	28	32	36	40	44	52	56	64	72	80	88
QP	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**			
Q_{step}	104	112	128	144	160	176	208	224			

TAB. 4.6 – Pas de quantification du codeur H.264.

de liberté de codage, c'est-à-dire, prendre des décisions de codage en fonction des informations dont nous disposons après notre pré-analyse de la vidéo. Cependant, toutes ces techniques de prédiction ont pour inconvénients d'augmenter la complexité calculatoire du codeur. Dans la deuxième partie de ce chapitre, nous nous intéressons aux travaux réalisés sur le codeur afin d'optimiser ces techniques de prédiction et de réduire les coûts de calculs, et examinons les approches que nous pouvons exploiter après notre méthode de pré-analyse.

4.2 Optimisations du codeur H.264

4.2.1 Introduction

Nous l'avons vu, au sein du codeur H.264, de nombreuses techniques de prédiction différentes peuvent être utilisées afin de réduire la dynamique des erreurs de prédiction et donc, incidemment, le flux de données nécessaire pour conserver la qualité des séquences vidéo. Cela a néanmoins pour conséquence d'augmenter la masse de calculs. Par exemple, le logiciel de référence établi à partir de la description du standard adopte une méthode de recherche exhaustive (*Full Search*) pour prédire le contenu des blocs [Ric03]. Il existe 7 modes inter différents (16×16, 16×8,8×16, 8×8, 8×4, 4×8 et 4×4) et deux modes intra (16×16 et 4×4). Pour les modes inter, la recherche des vecteurs de mouvement s'effectue à l'aide de l'ensemble des images référence. Ces différents modes de prédiction sont illustrés à la figure 4.10. Afin de choisir le mode de prédiction optimal pour un macrobloc, le codeur H.264, utilise une optimisation débit-distorsion par minimisation Lagrangienne. La forme générale de l'optimisation débit-distorsion est définie par :

$$J_{mode} = D + \lambda_{mode} \times R, \tag{4.1}$$

où J_{mode} est le coût débit-distorsion et λ_{mode} est le multiplicateur de Lagrange ; D est la distorsion calculée (de type erreur quadratique moyenne) entre le macrobloc courant et le macrobloc reconstruit, et R est le débit, plus précisément le nombre de bits nécessaires pour coder le mode, le paramètre de quantification du macrobloc, les bits nécessaires à l'en-tête du macrobloc, ainsi que le vecteur de mouvement et tous les résidus des coefficients DCT (la transformée des erreurs de prédiction).

Alors, l'estimation du mouvement peut représenter 60% (1 seule image référence) à 80% (5 images référence) de la charge de calcul à l'encodage et cela peut augmenter si on utilise une taille de fenêtre de recherche plus importante (48 ou 64 pixels). Afin de réduire ces temps de calcul, de nombreuses techniques ont été proposées. Les méthodes varient et sont axées sur trois stratégies bien distinctes : estimation du mou-

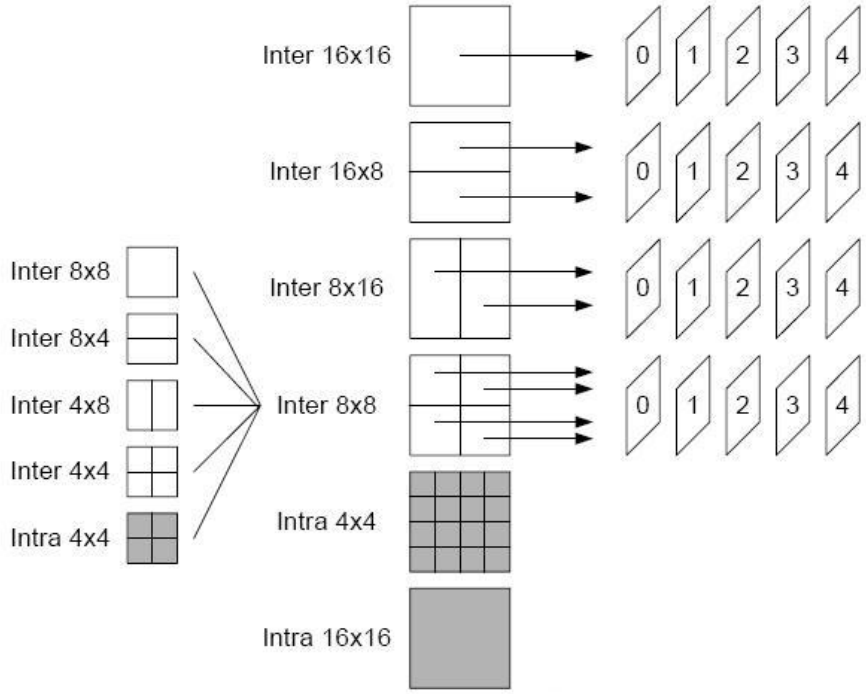


FIG. 4.10 – Étapes de recherche pour les prédictions intra et inter avec 5 images de référence pour le codeur H.264.

vement plus rapide, réduction des modes de prédiction intra et inter, et recherche partielle sur les multiples images référence.

Ces trois stratégies sont détaillées par la suite en décrivant les différentes approches proposées dans la littérature, ainsi que certaines plus particulières qui cherchent à optimiser la qualité perceptuelle de la vidéo reconstruite.

4.2.2 Accélération de l'estimation de mouvement

De nombreux algorithmes ont été proposés afin d'accélérer l'estimation du mouvement : recherche à trois étapes [KIH+81] (TSS[2]), estimateur logarithmique [JJ81] (2-D LOGS[3]), recherche par descente de gradient [LF96] BSDS[4]), recherche à quatre étapes [PM96] (FSS[5]), recherche sur une grille hexagonale [ZLC02] (HEXBS[6]), etc. Ils obtiennent de bons résultats pour de petits intervalles de recherche et pour des images de taille faible. Mais pour des signaux SDTV (*Standard Definition Television*) ou HDTV (*High Definition Television*), la taille de l'image est grande et la fenêtre de recherche doit être suffisamment large pour obtenir des résultats satisfaisants. Pour des séquences telles que *Stefan* ou *Bus*, la relaxation des algorithmes tels que HEXBS peuvent rencontrer rapidement un minimum local. Afin de résoudre ce problème de minimum local, les vecteurs de mouvement peuvent être prédits à partir de vecteurs des blocs voisins. Mais le vecteur de mouvement ainsi obtenu peut être « faux ». Les expériences montrent [CZH02] que de tels algorithmes entraînent une perte de 1 à $2dB$ pour la qualité de l'image par rapport à la méthode de

[2] *Three step search*
[3] *2-D logarithmic search*
[4] *Block based gradient descent search*
[5] *Four step search*
[6] *Hexagon-based Search*

recherche exhaustive sur les séquences *Stefan* et *Bus*.

4.2.2.1 Estimation de mouvement réalisant des recherches partielles

La plupart des méthodes d'optimisation concernant le codeur H.264/AVC essaient d'accélérer le procédé d'estimation de mouvement. Ces méthodes rapides sont composées pour la plupart de trois approches distinctes : prédiction du point de recherche initial, recherche de la meilleure position, technique d'arrêt adaptative.

Certaines d'entre elles ont été adoptées par le JVT et mises en œuvre au sein du logiciel de référence [CZH02, CTT02, YZLS05, TCT05].

4.2.2.1.1 Prédiction du point de recherche initial

De nombreux algorithmes d'estimation de mouvement privilégient le vecteur de mouvement nul $(0,0)$ et donc la fenêtre de recherche reste centrée sur la position zéro $(0,0)$. En effet, pour certaines séquences, les vecteurs de mouvement estimés restent concentrés dans une petite zone autour du centre de la fenêtre de recherche. Malheureusement, pour d'autres séquences, ce n'est plus vrai, ce qui implique que de tels algorithmes obtiendront alors des performances réduites. Afin d'améliorer les performances de l'estimation de mouvement, la fenêtre de recherche doit être centrée sur la position qui minimise l'erreur entre le bloc courant et son correspondant dans l'image référence. La prédiction du point de recherche initial est donc un paramètre important à prendre en considération. Différentes propriétés sont exploitées pour prédire le vecteur de mouvement et centrer la fenêtre de recherche sur la position minimisant l'erreur :

- **prédiction d'un mouvement initial nul** : La probabilité que le vecteur estimé soit nul $(0,0)$ étant voisine de 65% [MQ03], un déplacement nul est également choisi comme l'un des candidats pour la prédiction du point de recherche initial [CZH03, CXH03, XHFH06].
- **prédiction à partir de la partition supérieure** : Le vecteur de mouvement obtenu pour la partition de taille supérieure est utilisé comme candidat pour la couche inférieure [CZH02, CXH03, XYH03, YZLS05, TCT05, XHFH06].
- **la corrélation spatiale** : Le vecteur de mouvement prédit MV_p (médiane des déplacements des blocs spatialement adjacents) [CXH03, MQ03, XYH03, YZLS05], les vecteurs de mouvement des trois blocs adjacents [TAL01, CZH02, CZH03] ainsi que le vecteur de mouvement du bloc situé en haut à gauche du bloc courant [CTT02, TCT05] sont utilisés pour la recherche du point initial.
- **la corrélation temporelle** : Le vecteur de mouvement du bloc correspondant au bloc courant dans l'image précédente est utilisé comme candidat pour prédire le point de recherche initial [CXH03, XYH03, TCT05, XHFH06]. Les vecteurs de mouvement estimés pour la prédiction du point de recherche initial pouvant pointer vers une image référence différente de celle qui est présentement examinée, le vecteur de mouvement candidat, $\overrightarrow{MV_i'}$, calculé pour l'image référence, F_i, est pondéré par la distance temporelle relative entre l'image courante et l'image référence, F_j, qui est testée [CTT02, CXH03, XYH03, XHFH06], $\overrightarrow{MV_j'} = \frac{t-t_j}{t-t_i} \times \overrightarrow{MV_i'}$, où, $t-t_j$ et $t-t_i$ sont respectivement les distances temporelles entre l'image courante et l'image référence, F_i, et l'image référence, F_j, comme cela est illustré à la figure 4.11. Si l'image référence testée n'est pas la plus proche temporellement de l'image courante, le vecteur de mouvement estimé peut également être utilisé comme candidat pour les autres images référence, il est alors pondéré en fonction de la distance temporelle entre les deux images référence [CTT02]. Un nouveau candidat peut également être calculé en prenant en compte l'accélération d'un objet en mouvement [CTT02] (voir la figure 4.12). Ce candidat est obtenu à partir

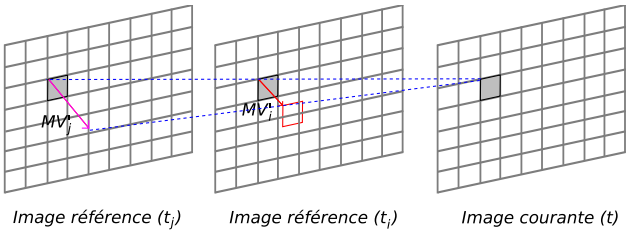


FIG. 4.11 – Prédiction du point de recherche initial à partir du vecteur de mouvement de l'image référence voisine.

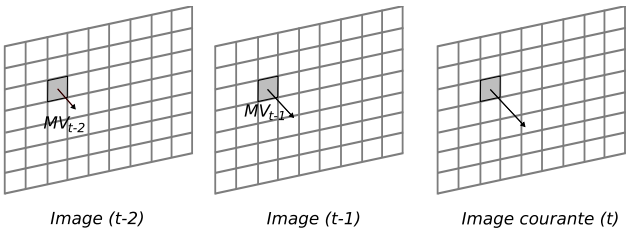
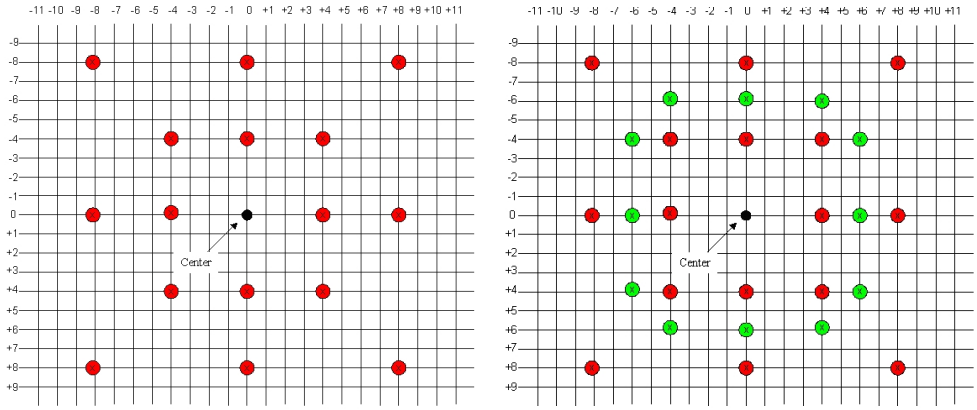


FIG. 4.12 – Prédiction du point de recherche initial en tenant compte de l'accélération du mouvement entre les deux images précédentes.

des vecteurs de mouvement du bloc correspondant dans l'image précédente $(t-1)$ et dans celle située juste avant $(t-2)$, $\overrightarrow{MV}_{acceleration} = \overrightarrow{MV}_{t-1} + (\overrightarrow{MV}_{t-1} - \overrightarrow{MV}_{t-2})$.

– **la corrélation spatio-temporelle** : Le vecteur de mouvement du bloc courant étant également corrélé avec les vecteurs de mouvement des blocs spatialement adjacents au bloc correspondant de l'image précédente [CTT02], les vecteurs de mouvement des quatre ou huit [TCT05] blocs spatialement adjacents (voisinage 4-connexe ou 8-connexe) au bloc correspondant viennent s'ajouter à l'ensemble des candidats pour la prédiction du point de recherche initial.

Tourapis et al [CTT02] ont également proposé une grille de prédicteurs fixes centrée sur le vecteur de mouvement prédit ou alors sur la position nulle $(0,0)$ (figure 4.13) qu'ils n'utilisent que pour la première image de la séquence codée en inter. Ils proposent une deuxième grille avec neuf points supplémentaires (voir figure 4.13 (b)), qui n'est utilisée que si au moins deux des voisins spatiaux du bloc courant sont codés en intra [TCT05].



(a) (b)

FIG. 4.13 – Grille de prédicteurs fixes, (a) celle utilisée pour la première image de la séquence codée en inter et (b) celle utilisée dans le cas ou deux des blocs voisins sont codés en intra.

4.2.2.1.2 Utilisation de « formes de recherche »

Plusieurs formes de recherche peuvent être utilisées pour accélérer l'estimation du mouvement :

- **une croix de dimension ± 2 pixels** centrée sur le point de recherche initial [TP06, TPC03]. Ting et al [TPC03] testent cette croix de dimension ± 2 pixels pour les cinq images référence. La position qui minimise l'erreur de mise en correspondance indique l'image référence pour laquelle une recherche exhaustive est réalisée afin d'affiner le vecteur de mouvement.
- **un diamant** [TAL01, CTT02, TCT05] : Tourapis et al [TAL01] utilisent deux diamants de taille différentes (± 2 pixels et ± 1 pixel) en fonction d'une caractérisation initiale du mouvement. Ma et Qiu [MQ03] proposent de générer une forme de recherche en fonction des vecteurs de mouvements obtenus pour les blocs voisins. Les quatre coins de la forme de recherche adaptative de leur algorithme d'estimation de mouvement sont obtenus de la manière suivante : $\overrightarrow{MV_1} = [max(MV_x), MV_{p_y}]$, $\overrightarrow{MV_2} = [min(MV_x), MV_{p_y}]$, $\overrightarrow{MV_3} = [MV_{p_x}, min(MV_y)]$, $\overrightarrow{MV_4} = [MV_{p_x}, max(MV_y)]$, où MV_x et MV_y représentent respectivement les composantes horizontales et verticales de tous les vecteurs de mouvement des blocs voisins. Les opérateurs *max* et *min* sont utilisés pour trouver les valeurs maximale et minimale parmi tous ces vecteurs de mouvement. MV_{p_x} et MV_{p_y} représentent les composantes horizontales et verticales du vecteur de mouvement prédit. Pour affiner la recherche, un diamant unitaire est centré sur le point minimisant l'erreur, le procédé est alors réitéré jusqu'à ce que le point minimisant l'erreur soit situé au centre du diamant.
- **un carré de taille ± 1 pixel** [CTT02, TCT05].
- **différentes formes hexagonales** : Pour affiner la position du point minimisant l'erreur obtenue avec une croix de dimension ± 2 pixels, Tsai et Pan [TP06] utilisent ensuite différentes formes hexagonales (figure 4.14) en fonction de la position de celui-ci. Chen et al [CZH02] ont proposé une méthode appelée UMHexagonS[7] et reprise par de nombreux auteurs [CZH03, CXH03, XYH03, YZLS05, ZPW08]. Elle est composée de quatre étapes successives combinant différentes formes de recherche : prédiction du point de recherche initial, recherche sur une croix asymétrique, recherche sur de multiples grilles hexagonales et irrégulières, et recherche basée sur des hexagones étendus (figure 4.15).

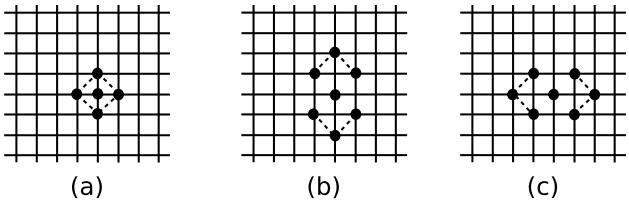


FIG. 4.14 – Hexagones de recherche pour l'estimation de mouvement. (a) Petit hexagone de recherche. (b) Grand hexagone vertical de recherche. (c) Grand hexagone horizontal de recherche.

4.2.2.2 Estimation de mouvement utilisant un sous-échantillonnage

Afin de réduire la complexité calculatoire certains auteurs proposent de réaliser un sous-échantillonnage des images [CHC+05, LSIG06]. Afin d'éviter un recouvrement des fréquences, les images subissent d'abord un filtrage passe-bas [LSIG06]. Ensuite, le sous-échantillonnage est réalisé sur les versions filtrées de ces images.

Choi et al [CHC+05] réalisent une méthode d'estimation de mouvement multi-résolution. Les vecteurs de mouvement sont estimés à la résolution la plus faible et le vecteur estimé est mis à l'échelle afin d'être

[7] *Unsymmetrical-cross Multi-Hexagon-grid Search*

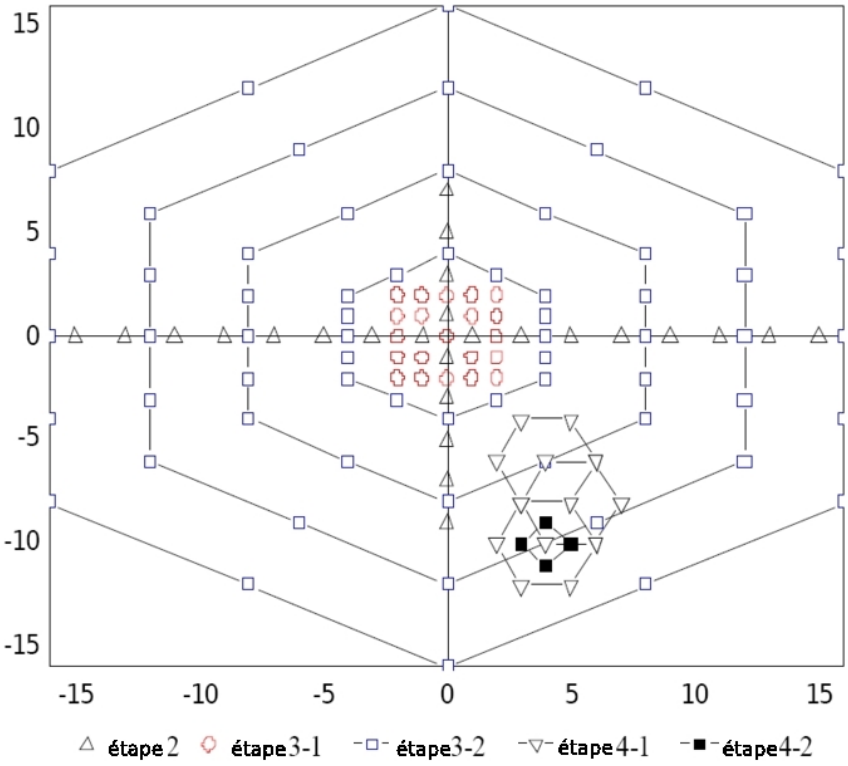


FIG. 4.15 – Les différentes étapes de recherche de l'algorithme UMHexagonS pour une fenêtre de recherche de ± 16 pixels.

utilisé comme point de recherche initial et affiné pour les résolutions supérieures. La figure 4.16 illustre les formes de recherche utilisées pour les différents niveaux ; le petit carré (SS[8]), la grille de multiples hexagones irréguliers (UMHG[9]) et l'hexagone étendu (EH[10]). Pour la plus faible résolution, la recherche du vecteur de mouvement est réalisée à l'aide de deux formes de recherche ; le petit carré et la grille de multiples hexagones irréguliers. L'hexagone étendu est utilisé pour affiner le vecteur de mouvement pour les résolutions de taille supérieure.

Les recherches multi-résolutions basées sur une décomposition à l'aide d'une transformation en ondelettes dyadique possèdent des limites. En effet, le vecteur de mouvement est obtenu avec une précision de deux pixels, il est nécessaire de réaliser une recherche supplémentaire pour obtenir une précision plus fine.

4.2.2.3 Optimisation de l'estimation de mouvement sous-pixélique

Le codeur H.264 permet de réaliser l'estimation de mouvement, non pas au pixel près, mais au $\frac{1}{8}$ de pixel, augmentant ainsi la complexité calculatoire. On trouve donc également dans la littérature, des algorithmes d'optimisation de l'estimation de mouvement sous-pixélique [CZH02, YZLS05, TCT05, HCB06].

En se basant sur l'hypothèse d'une erreur unimodale pour l'estimation de mouvement sous-pixélique, Chen et al ont proposé une méthode de recherche sous-pixélique avec prédiction du point de recherche initial [CZH02] (CBFPS[11]) qui est plus rapide et obtient les mêmes performances en terme de débit-distorsion que celle du logiciel de référence. Ils extraient la partie décimale du vecteur de mouvement prédit en sous-

[8] *Small Square*
[9] *Uneven Multi-Hexagon Grid*
[10] *Extended Hexagon*
[11] *Center Biased Fractional Pel Search*

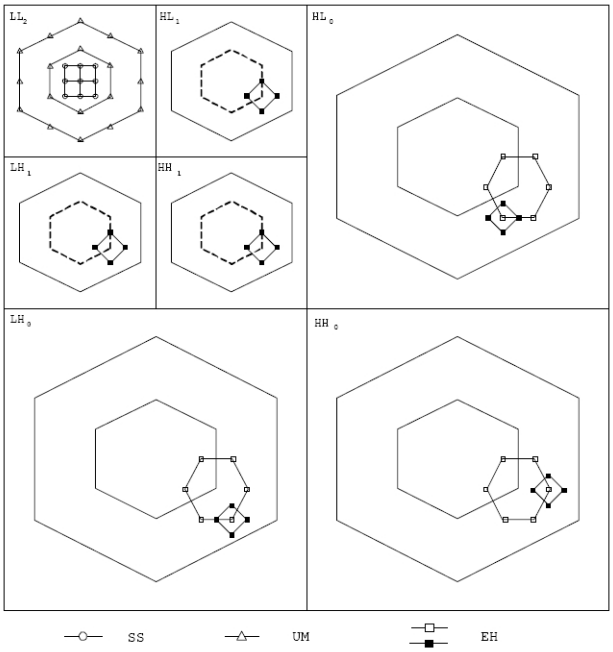


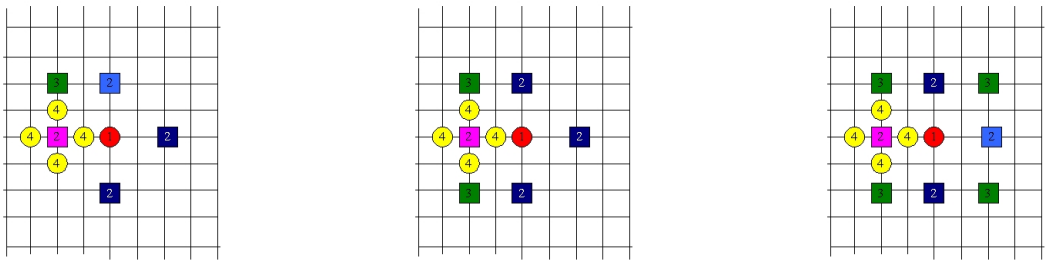
FIG. 4.16 – Formes de recherche pour l'estimation de mouvement multi-résolutions.

trayant le vecteur de mouvement MV obtenu lors de l'estimation de mouvement au pixel près :

$$Décimale_{MV_p} = (MV_p - MV),$$

Cette partie décimale du vecteur de mouvement prédit sert de point de recherche initial pour l'estimation de mouvement sous-pixélique. La position est ensuite affinée en réalisant une recherche en diamant. Le déplacement du diamant peut également être décidé en fonction de la position des deux points minimisant l'erreur de mise en correspondance [TT05] (figure 4.17). Bien qu'ayant été adoptée au sein du codeur de référence pour l'estimation de mouvement sous-pixélique, la méthode de Chen et al [CZH02] n'est appliquée qu'aux blocs de petites tailles. Cela implique que pour les partitions supérieures (16×16, 16×8 et 8×16), l'estimation de mouvement sous-pixélique est réalisée à partir d'une méthode de recherche exhaustive. Afin de palier à ce défaut, Yi et al proposent deux méthodes rapides d'estimation de mouvement sous-pixélique [YZLS05] : une méthode simple et efficace d'omission de la recherche sous-pixélique basée sur une analyse statistique, et une technique d'arrêt immédiat basée sur le coût minimum.

Afin de réduire l'utilisation de la mémoire et d'améliorer l'efficacité calculatoire, Hill et al ont proposé une méthode d'interpolation pour l'estimation de mouvement sous-pixélique [HCB06]. Ils utilisent un modèle 2D parabolique qui estime l'erreur des positions sous-pixéliques à partir de la somme des différences absolues pour la meilleure position au pixel près et ses huit voisins (voisinage 8-connexe). L'estimation de mouvement sous-pixélique est réalisée en identifiant le minimum de la surface parabolique en dimension deux. Afin de ne pas être bloqué dans un minimum local, ils calculent la divergence moyenne des sommes des différences absolues, entre les valeurs réelles (positions au pixel près) et celles estimées avec le modèle parabolique. Si la divergence moyenne est trop importante, alors la méthode d'interpolation classique est utilisée pour ce point. Plus cette méthode d'interpolation classique est utilisée, plus les performances en termes de débit-distorsion s'améliorent au détriment des coûts de calculs. Le seuil de décision utilisé sur la fonction de la divergence moyenne permet un compromis entre complexité calculatoire et qualité de l'estimation.



(a) (b) (c)

FIG. 4.17 – Exemples du raffinement de la position sous-pixélique en fonction des deux meilleures positions précédentes. Dans le cas (a) les deux meilleures positions adjacentes non nulles, ainsi une seule position est testée en supplément pour le raffinement. Dans le cas (b) l'une des deux meilleures positions est la position nulle et dans ce cas deux points supplémentaires sont testés. Finalement dans le cas (c), les deux meilleures positions sont opposées par rapport à la position nulle, ce qui suggère que l'information n'est pas suffisante et toutes les positions sous-pixéliques (à la résolution sous-pixélique courante) sont alors testées.

4.2.2.4 Utilisation de techniques d'arrêt

Le procédé de recherche peut être stoppé sans avoir testé l'ensemble des points de recherche. Cette décision d'arrêt de l'estimation du mouvement est basée sur la détection des blocs qui ne contiendront que des zéros après transformation et quantification. Dès qu'un tel bloc est détecté lors de l'estimation de mouvement, il n'est alors plus nécessaire de rechercher un éventuel autre meilleur bloc correspondant, car celui-ci n'apporterait aucun gain en termes d'efficacité de codage. La détection a priori de ces blocs peut être réalisée à partir de l'analyse théorique des coefficients de la transformation en cosinus discrète et de la quantification [CZH02, MKK05, WKK06, WLC+07, LL07]. Comme pour les vecteurs de mouvement, les distorsions des blocs adjacents tendent à être fortement corrélées. En considérant cela, une technique d'arrêt peut être mise en œuvre pour l'estimation du mouvement. La distorsion du macrobloc courant peut être prédite à partir des résultats obtenus précédemment pour les macroblocs voisins. De la même manière que pour la prédiction du point de recherche initial, l'erreur de prédiction peut être estimée de plusieurs façons : prédiction médiane, ou à partir des partitions de plus grandes tailles, ou à partir du bloc correspondant dans l'image précédente, ou à partir de l'erreur obtenue pour l'image référence voisine.

Un seuil de décision peut alors être évalué et si l'erreur de prédiction du macrobloc courant est inférieure à celui-ci, les autres points ne sont pas évalués [CZH03, CXH03, TAL01, CTT02, TCT05].

4.2.2.5 Discussion

Les différentes méthodes d'optimisation de l'estimation de mouvement que nous venons de décrire permettent de réduire les coûts de calculs, tout en maintenant des performances en termes de débit-distorsion proches de celles obtenues avec la méthode de recherche exhaustive pour la mise en correspondance des blocs. Cependant l'approche de mise en correspondance étant basée sur la minimisation de l'erreur entre le bloc courant et le bloc le plus proche spatialement (en termes de signal) dans l'image de référence, l'estimation de mouvement ne reflète pas le mouvement réel (projeté) des blocs (objets). Ce qui à bas débit, peut favoriser l'apparition d'artéfacts de codage très visibles. Les différentes méthodes d'optimisation de l'estimation de mouvement n'ont pour objectif que de réduire la complexité calculatoire rendue très importante par les différents modes de prédiction et les multiples images référence et aucune considération de plus

haut niveau n'est prise afin de « mieux » comprendre le contenu et d'optimiser ces méthodes dans ce sens. Connaissant le cycle de vie des objets présents dans la séquence vidéo après notre pré-analyse, plusieurs options peuvent être envisagées afin d'optimiser l'estimation du mouvement en termes de cohérence de codage et de qualité perceptuelle. Afin d'éviter les défauts d'effet de papillotement dus à des choix hétérogènes temporellement pour coder le même macrobloc (même position spatiale) entre des images successives, l'estimation de mouvement pourrait être contrainte : choix de l'image référence, sélection du point de recherche initial en fonction du mouvement estimé de l'objet (segmentation spatio-temporelle).

4.2.3 Optimisation de la prédiction intra-image

Au sein du codeur vidéo H.264, la prédiction intra-image exploite la redondance spatiale. Au total, treize modes (neuf modes pour les blocs 4×4 et quatre modes pour les blocs 16×16) sont testés pour prédire un macrobloc (voir la section 4.1.4). Il existe aujourd'hui dans la littérature, plusieurs méthodes d'optimisation de la prédiction intra, permettant de sélectionner les modes de prédiction. Celles-ci se basent sur des analyses du domaine spatial, ou exploitent la corrélation spatio-temporelle entre les macroblocs des images successives de la vidéo.

4.2.3.1 Détection de l'orientation des contours

Les pixels situés d'un même coté d'un contour local ont généralement des valeurs similaires, ainsi une bonne prédiction peut être réalisée à partir de pixels voisins bien choisis. Il existe plusieurs méthodes pour obtenir l'information locale de la direction d'un contour, par exemple l'histogramme des directions des contours après un simple algorithme de détection des contours, des gradients locaux ou un filtre de Sobel[PLR+05]. Après avoir appliqué un filtre de Sobel sur l'image courante, Pan et al [PLR+05] calculent un histogramme de direction des contours pour chaque mode de prédiction : des blocs de luminance 4×4 pixels, de ceux 16×16 pixels, et de chrominance 8×8 pixels. Un ensemble réduit de modes est déterminé pour les différentes tailles de blocs de la prédiction intra :

- blocs de luminance 4×4 pixels : le mode correspondant au maximum dans l'histogramme, les deux modes voisins ainsi que le mode DC ;
- blocs de luminance 16×16 pixels : le maximum de l'histogramme et le mode DC ;
- blocs de chrominance 8×8 pixels : les deux maximums (un pour la composante U et l'autre pour la composante V) et plus le mode DC.

Qiong et al [LHZ+06] utilisent également la méthode d'accumulation des directions des contours dans des histogrammes. Si l'amplitude du maximum détecté dans l'histogramme des directions des contours a une valeur importante, le bloc correspondant sera codé en intra 4×4 pixels. Dans le cas contraire, l'ensemble réduit des modes à tester est constitué du mode représenté par le maximum dans l'histogramme, le mode le plus représenté dans les blocs voisins et le mode DC. Le mode optimal peut également être estimé après analyse d'une matrice des contours pour chaque bloc [HZL07] définie par :

$$H = \sum_{bloc} \begin{bmatrix} dx_{i,j}^2 & dx_{i,j}dy_{i,j} \\ dx_{i,j}dy_{i,j} & dy_{i,j}^2 \end{bmatrix}$$

où $dx_{i,j}$ et $dy_{i,j}$ représente respectivement les valeurs des composantes horizontale et verticale des gradients du bloc de coordonnées (i, j).

Afin de ne pas réaliser de trop nombreuses opérations dues à l'utilisation d'un filtre de Sobel, une transformation d'entiers peut être appliquée sur l'image originale pour détecter les directions des textures locales [SLZ06]. Cette méthode obtenant des résultats moins précis que celle utilisant un filtre de Sobel, un seuil adaptatif de décision est employé. Après application de la transformation d'entiers, un ensemble restreint de modes est testé et si la somme des différences absolues du bloc reconstruit avec le mode optimal est suffisamment faible, il n'est pas nécessaire de tester de modes supplémentaires.

La détection des directions des contours peut être également réalisée à partir de l'analyse des coefficients obtenus après une TCD (DCT en anglais) [LEC07, TNHT05].

4.2.3.2 Décision en fonction de la corrélation spatiale

Les macroblocs au sein d'une même image sont fortement corrélés, de telle sorte que pour de nombreux macroblocs d'une image intra, les modes de prédiction retenus sont les mêmes [WTST06]. Si le coût débit-distorsion obtenu avec l'un des modes des macroblocs voisins est suffisamment faible, il n'est alors pas nécessaire de tester les autres modes. Zhang et al [ZWH+07] ont analysé les modes retenus en fonction du contexte du voisinage pour plusieurs séquences vidéo et classent par ordre croissant les probabilités des modes. Les probabilités et les rangs des modes sont stockés dans deux tables indexées par le contexte du voisinage. Lorsqu'un macrobloc doit être prédit pour un contexte de voisinage spécifique, les coûts débit-distorsion de chaque mode sont évalués à partir de la table des rangs jusqu'à ce que la probabilité de ne pas avoir testé le meilleur mode soit suffisamment faible. Si le coût débit-distorsion du meilleur mode testé est inférieur au seuil défini en fonction du paramètre de quantification, le procédé de prédiction s'arrête. Dans le cas contraire, les autres modes restants sont testés. Finalement, une étape de mise à jour des tables et du seuil de décision est réalisée en fonction du mode optimal.

4.2.3.3 Décision en fonction de la corrélation temporelle

Afin de mesurer le degré de corrélation entre macroblocs, Xin et Vetro [Xin06] calculent la différence entre deux blocs b_2 et b_1 :

$$D(b_2,b_1) = \sum_{j=b_y-1}^{b_y+15} \sum_{i=b_x-1}^{b_x+15} |p_2(j,i) - p_1(j,i)| + \sum_{i=b_x+16}^{b_x+23} |p_2(b_{y-1},i) - p_1(b_y-1,i)|,$$

où p_2 et p_1 sont les deux images contenant respectivement b_2 et b_1. b_x et b_y sont les coordonnées horizontales et verticales de b_2 et b_1. Cette différence est calculée à partir de tous les pixels pouvant intervenir dans la prédiction intra du macrobloc courant. Hwang et al [HCK+05] ont proposé d'utiliser les informations obtenues à partir de la prédiction inter pour le macrobloc courant. Le mode intra optimal du macrobloc pointé par le vecteur de mouvement estimé lors de la prédiction inter, est ajouté à l'ensemble réduit des modes intra. En plus du mode intra du macrobloc pointé par le vecteur de mouvement, les deux modes voisins en termes d'angle de prédiction (pour la prédiction intra 4×4 pixels) et le mode DC sont ajoutés à l'ensemble réduit des modes.

4.2.3.4 Discussion

Cette section a présenté les différentes méthodes d'optimisation de la prédiction intra du codeur H.264. Les méthodes proposées réalisent une analyse du domaine spatial, en appliquant des algorithmes de détection de contours à l'aide de filtres de Sobel ou d'analyse des coefficients de la transformée en cosinus

discrète. Bien que certaines méthodes exploitent la corrélation spatio-temporelle entre les macroblocs pour la sélection du mode du macrobloc courant, aucune information sur les macroblocs appartenant à la même région ou au même objet ne sont prises en considérations. Ainsi, des macroblocs appartenant au même objet et ayant une activité spatiale similaire, peuvent être codés avec des modes différents. Ces choix hétérogènes pour le codage intra des macroblocs appartenant à la même région spatiale peuvent provoquer des défauts visuellement gênants, tels que des effets de blocs. Il serait plus judicieux de contraindre la sélection des modes pour les macroblocs appartenant à une région spatiale ayant une texture homogène. En effet, suite à notre pré-analyse nous disposons des cartes de segmentation spatio-temporelle, il est alors envisageable de choisir un seul mode ou un ensemble restreint de modes pour chacune des régions segmentées (en fonction de leur texture : activité et orientation) et coder tous les macroblocs appartenant à cette région avec ce mode ou cet ensemble restreint.

4.2.4 Optimisation de la prédiction inter-image

De nombreuses méthodes ont été proposées dans la littérature, celles-ci exploitant certaines caractéristiques intra et/ou inter images, afin de sélectionner un ensemble réduit de modes à tester. Ainsi les partitions non adaptées aux caractéristiques du macrobloc ne seront pas testées.

4.2.4.1 Détection des macroblocs codés en mode SKIP

Le mode SKIP est généralement assigné aux macroblocs contenant des informations spatiales quasi identiques au macrobloc correspondant dans l'image précédente. Si le codeur peut détecter de tels macroblocs sans connaissance a priori, il est alors inutile de tester les autres modes. Les densités de probabilité des différences des coûts débit-distorsion entre le mode retenu et le mode SKIP (et pour des séquences références), ont été modélisées comme fonctions du paramètre de quantification [ZBR06]. À partir de ces modèles analytiques, des seuils de décision pour le coût débit-distorsion du macrobloc courant sont évalués. Alors pour tous les macroblocs, le mode SKIP est d'abord estimé et si le coût débit-distorsion est inférieur au seuil de décision, les autres modes de prédiction ne sont pas testés. L'approche proposée par Nieto et al [NSJ06] diffère légèrement de celle proposée par Zhao [ZBR06]. En effet, ils testent le mode SKIP pour tous les macroblocs, et la décision de poursuivre l'estimation de mouvement avec les autres modes de prédiction est réalisée en fonction d'un seuil basé sur la distorsion moyenne des images précédemment codées.

4.2.4.2 Décision en fonction de l'activité spatiale

On peut observer qu'il existe de nombreuses régions homogènes qui appartiennent au même objet vidéo ou au fond de la scène. Lorsque cet objet ou le fond bouge, les régions homogènes se déplacent de la même façon. Ces régions homogènes seront rarement codées à l'aide de partitions de petites tailles. De la même façon, lorsqu'un macrobloc contient des contours, ceux-ci peuvent appartenir au même objet ou pas. Même s'ils appartiennent au même objet, ils peuvent se déplacer (bouger) dans des directions différentes et seront alors rarement codés avec des partitions de grandes tailles. Généralement, les régions homogènes sont codées à l'aide de grandes partitions et les macroblocs contenant des contours sont codés à l'aide de petites partitions. Pan et Tsai [PT07] exploitent ces observations en mesurant l'activité spatiale au sein du macrobloc (gradient par filtre de Sobel) . Ils définissent ensuite des seuils adaptatifs en fonction du paramètre de quantification pour évaluer la valeur obtenue du gradient et décider de la partition adéquate pour le

macrobloc. En effet, à bas débit, le nombre de macroblocs codés avec le mode 16×16 augmente et lorsque le débit croît le nombre de macroblocs codés avec le mode 16×16 diminue. L'algorithme de sélection rapide du mode inter adopté par le JVT utilise également le gradient spatial (filtre de Sobel) afin d'identifier les blocs homogènes et favoriser les grandes partitions dans de tels cas [LWW+03]. Dans ses différents travaux, Yu utilisent un facteur prédictif basé sur la complexité intrinsèque du macrobloc. Il calcule la somme des coefficients AC de chaque bloc après la TCD. Le critère pour évaluer la complexité spatiale d'un macrobloc est obtenu de la façon suivante :

$$R_B = \frac{ln(E_{AC})}{ln(E_{max})} \tag{4.2}$$

où, E_{max} est la somme maximale possible des coefficients du macrobloc (cas du « damier blanc et noir »), et E_{AC} est la somme des coefficients des hautes fréquences. À partir de la valeur obtenue pour cette mesure de l'activité spatiale, il classe les macroblocs dans trois catégories distinctes [Yu04] :

- MD16 : mode 16×16, mode SKIP et les modes intra,
- MD8 : mode 16×16, mode 16×8, mode 8×16, mode 8×8, mode SKIP et les modes intra,
- MD4 : toutes les partitions.

La catégorie sélectionnée est ensuite révisée en fonction du mode optimal retenu pour le macrobloc correspondant dans l'image précédente. Par la suite, ils ont proposé de détecter les macroblocs pouvant être codés à l'aide du mode SKIP, en fonction du mode utilisé pour les quatre macroblocs voisins [YM04]. Si le macrobloc courant n'est pas codé à l'aide du mode SKIP, ils mesurent alors la complexité spatiale du macrobloc à l'aide de la formule décrite dans l'équation 4.2 et sélectionnent certains modes en fonction de la valeur obtenue. Finalement, les modes sélectionnés sont révisés en fonction des coûts débit-distorsion calculés [YMP05]. Dans leurs travaux, Lee et Lin [LL06] utilisent également la mesure de complexité spatiale de Yu et la compacité des vecteurs de mouvement pour décider de tester les partitions de tailles inférieures.

4.2.4.3 Décision en fonction de l'activité spatio-temporelle

De petites tailles de partitions sont préférées lorsque les caractéristiques spatio-temporelles du macrobloc sont significatives, et des partitions de grandes tailles sont choisies pour les macroblocs homogènes (avec peu de mouvement). Cette information spatio-temporelle peut être évaluée à l'aide d'un gradient spatio-temporel [AKA06, BAA06]. Afin d'évaluer le mode optimal pour chaque macrobloc, Ates et al calculent le gradient spatio-temporel de la façon suivante [AKA06] :

$$D = D_T + Dx + D_y$$

où

$$D_T = \sum_{x=1}^{16}\sum_{y=1}^{16}|c(x,y) - r(x,y)|;\ D_x = \sum_{x=1}^{16}\sum_{y=1}^{16}|c(x,y) - c(x+1,y)|;\ D_y = \sum_{x=1}^{16}\sum_{y=1}^{16}|c(x,y) - c(x,y+1)|$$

Ici, c et r représentent respectivement le macrobloc courant et le macrobloc correspondant dans l'image précédente. Les décisions sont prises en fonction de la valeur du gradient spatio-temporel calculé qui indique l'activité spatiale et temporelle du macrobloc. Les petites partitions sont préférées pour les macroblocs ayant un gradient élevé, au contraire lorsque celui-ci est moins important les grandes partitions sont choisies. La décision de garder une partition est réalisée afin de minimiser le coût total des erreurs d'estimation pour une image donnée. Ainsi, un mode est supprimé si seulement la réduction des calculs justifie la possible perte

en efficacité de codage. Considérant le temps nécessaire alloué pour coder une image, le critère de sélection des modes est adapté de façon telle que l'estimation de mouvement pour l'image entière soit réalisée avec le nombre de cycles d'horloge prévu. Ates et al [AKA06] calculent la probabilité de chaque partition à partir de gradients spatio-temporels évalués durant un procédé d'apprentissage. Ils utilisent une fonction rationnelle quadratique afin de modéliser ces probabilités et obtiennent ainsi les paramètres de leur fonction pour tous les modes et pour différents paramètres de quantification (32, 28, 24, 20, et 16). Les probabilités du macrobloc courant sont obtenues en fonction de la valeur du gradient spatio-temporel, les trois partitions ayant les plus fortes probabilités sont alors sélectionnées.

4.2.4.4 Décision en fonction de la corrélation spatio-temporelle

L'information contextuelle des modes optimaux pour les macroblocs voisins peut être étudiée, afin de sélectionner un ensemble réduit de modes possibles pour le macrobloc courant. Certaines méthodes proposent d'étudier les modes retenus pour les macroblocs voisins spatialement (et déjà codés) ainsi que le macrobloc correspondant dans l'image précédente et ses huit voisins [KSC06, HLC+06]. Ensuite, le macrobloc est classé dans une catégorie correspondante. Kim et al [KSC06] considèrent seulement deux catégories : les partitions (16×16 , 16×8 et 8×16 pixels), et les sous-partitions (8×8 , 8×4, 4×8 et 4×4 pixels).

Si le macrobloc est classé dans la catégorie des sous-partitions, les coûts débit-distorsion des modes 8×8 et 16×16 sont comparés. Si le coût du mode 8×8 est inférieur à celui du mode 16×16, les autres sous-partitions sont testées, dans le cas contraire, seules les partitions 16×8 et 8×16 sont évaluées. Huang et al [HLC+06] ont proposé de distinguer deux nouvelles catégories : le mode SKIP et le mode Intra. La catégorie du macrobloc est choisie en fonction du mode le plus représenté parmi les modes retenus des treize voisins (neuf voisins temporels et quatre voisins spatiaux). Le raffinement de la décision est réalisé à partir de tests bayésiens et d'un réseau de neurones entraîné sur cinq vidéos.

4.2.4.4.1 Décision en fonction de la corrélation temporelle

Certaines méthodes n'utilisent que les informations du macrobloc correspondant (de l'image précédente). En effet, le coût débit-distorsion obtenu pour le mode retenu du macrobloc correspondant peut être utilisé comme seuil de décision pour le macrobloc courant [SN06, Kim07]. Les modes sont testés dans un ordre hiérarchique et la décision de tester les partitions inférieures est prise en fonction du coût débit-distorsion comparé à celui du macrobloc correspondant.

4.2.4.4.2 Décision en fonction de la corrélation spatiale

Les expériences menées par Chang et al montrent que dans plus de 60% des cas, si les quatre voisins spatiaux ont été codés avec le même mode, le macrobloc courant est alors également codé avec ce mode [CPC04]. Leur méthode exploite l'information disponible des quatre macroblocs voisins, c'est-à-dire, les modes de codage retenus pour ceux-ci. Si au moins trois de ceux-ci ont été codés avec le même mode de prédiction, on assigne ce mode au macrobloc courant. Les autres modes ne seront testés que si le coût débit distorsion du macrobloc courant pour le mode désigné est supérieur à la moyenne des coûts débit-distorsion des trois ou quatre macroblocs voisins codés avec ce mode. Afin de déterminer des régions homogènes, Kamnoonwatana et al [KAC07] utilise le descripteur de textures de MPEG-7. Ils obtiennent ainsi des informations de la texture de chaque macrobloc. Afin de regrouper les macroblocs en régions homogènes, ils utilisent un algorithme d'agglomération hiérarchique. Le choix du mode de prédiction pour le macrobloc courant est décidé en fonction du mode retenu pour le macrobloc précédemment codé et appartenant à la

même région. Les modes utilisés pour les macroblocs appartenant à la même région dans l'image précédente sont analysés afin d'affiner la décision [KAC08].

4.2.4.5 Décision en fonction des coûts débit-distorsion

Afin de réduire le nombre de modes de prédiction testés lors de l'estimation de mouvement, les propriétés statistiques des coûts débit-distorsion des modes retenus pour des séquences tests ont été étudiées [MEdlPADdM07]. L'analyse de ces résultats montre que les grandes partitions sont préférées lorsque le coût débit-distorsion est faible. À partir de ces résultats des seuils de décision peuvent être construits afin de sélectionner les modes optimaux pour chaque macrobloc. L'estimation de mouvement est réalisée dans un ordre hiérarchique en commençant par les plus grandes partitions (SKIP, 16×16 , 16×8 et 8×16 pixels), les partitions inférieures ne sont testées que si les coûts débit-distorsion obtenus pour les modes testés ne sont pas satisfaisants, c'est-à-dire, s'ils sont supérieurs aux seuils estimés [KKKH05, BLCZ06, CMB+05]. La monotonie de l'erreur de prédiction peut également renseigner sur les modes optimaux pour le macrobloc courant [YCTB03].

4.2.4.6 Discussion

Cette section a présenté les différentes méthodes d'optimisation de la prédiction inter du codeur H.264. Les algorithmes proposés exploitent certaines propriétés spatiales et ou temporelles des séquences vidéos, telles que l'activité spatio-temporelle des images et la corrélation spatio-temporelle entre les images, afin de supprimer les modes les moins à même de produire des gains en termes de débit-distorsion. À partir d'expériences réalisées sur des séquences tests, ils mettent en évidence des relations entre les propriétés spatio-temporelles des macroblocs et les modes optimaux retenus. Cependant, même si certaines méthodes exploitent quelque peu la corrélation spatio-temporelle entre les macroblocs pour la sélection du mode de prédiction inter optimal, aucune cohérence de codage n'est assurée pour coder de façon optimale un objet dans l'image courante puis sur les images successives. Seule l'approche de Kamnoonwatana et al [KAC07, KAC08] assure une certaine cohérence pour le codage de régions ayant des caractéristiques spatio-temporelles similaires. En effet, les macroblocs appartenant au même objet présent sur plusieurs images successives risquent d'être codés avec des modes différents. Ce qui peut avoir pour effet de provoquer l'apparition de défauts visuellement gênants, tels que des effets de blocs ou des effets de papillotement. Afin d'éviter l'apparition de tels défauts, les macroblocs voisins spatialement et temporellement qui appartiennent au même objet doivent être codés avec des partitions similaires. Une telle approche est envisageable après notre étape de pré-analyse. En effet, grâce aux résultats de notre segmentation temporelle, nous pouvons suivre un objet sur plusieurs segments temporels et connaître ainsi son cycle de vie et définir la partition ou ensemble de partitions en fonction des caractéristiques de cet objet pour coder les macroblocs appartenant à celui-ci.

4.2.5 Recherche réduite pour les multiples images référence

Comme nous l'avons vu dans la section 4.1.2.3, le codeur H.264 utilise plusieurs images référence pour ancrer la prédiction de mouvement. Cette estimation de mouvement à partir de plusieurs images référence permet de réduire le débit pour une même qualité. Mais cette réduction des erreurs de prédiction ne dépend pas seulement du nombre d'images référence utilisées, mais également de la nature des séquences vidéo.

4.2.5.1 Prépondérance de l'image précédente

Des études ont montré que 60% des vecteurs de mouvement déterminés par le codeur de référence pointaient sur l'image référence précédente à l'image courante [HHW+03, ZPW08]. Plusieurs auteurs proposent des méthodes exploitant ces observations. Une recherche complète [HHW+03] est d'abord réalisée pour l'image référence la plus proche temporellement de l'image courante. Ensuite, les informations obtenues, telles que le mode retenu (inter ou intra), le vecteur de mouvement et les erreurs de prédiction (inter et/ou intra), sont analysées pour déterminer si l'estimation de mouvement doit être réalisée sur des images référence supplémentaires. D'autres études menées sur la meilleure image référence et le mode retenu lors d'une estimation de mouvement multi-référence [ZPW08], ont montré que dans 91% des cas, le mode optimal obtenu lors de l'estimation multi-référence est le même que celui obtenu pour l'image référence la plus proche temporellement de l'image courante. Ils réalisent d'abord une estimation de mouvement rapide [YZLS05] sur l'image référence précédente et testent ensuite le meilleur mode (obtenu pour l'image précédente) sur les autres images référence. Pour les autres partitions, ils testent seulement trois positions (le vecteur de mouvement prédit, la position nulle et le vecteur de mouvement obtenu pour le mode 16×16), puis raffinent la recherche à l'aide d'un hexagone et d'un diamant. Ting et al [TPC03] ont proposé quant à eux de réaliser d'abord une recherche à l'aide d'une fenêtre de recherche de taille ± 2 pixels sur chaque image référence. La position minimisant l'erreur sert d'indication sur la meilleure image référence. Une estimation de mouvement avec une fenêtre de recherche plus importante est alors réalisée sur l'image référence ainsi sélectionnée. Dans leur méthode, Kapotas et Skodras [KS07] ne testent seulement que deux positions (le vecteur de mouvement prédit et la position nulle $(0,0)$) sur chaque image référence. L'image référence pour laquelle le coût est minimum devient la référence optimale, et une estimation de mouvement complète est alors réalisée pour cette image seulement. Li et al [LLC04] ont proposé une méthode d'estimation multi-référence où l'estimation de mouvement est d'abord réalisée sur les trois premières images référence et ensuite la décision de continuer la recherche sur les deux dernières images référence est prise en fonction des vecteurs de mouvement et des coûts obtenus pour ces trois premières images référence.

4.2.5.2 Corrélation temporelle du vecteur de mouvement

La continuité du mouvement peut également être exploitée afin de faciliter l'estimation de mouvement multi-référence. En supposant qu'un objet (bloc) soit en mouvement dans une séquence vidéo, et garde une apparence similaire entre les différentes images. La continuité de ce bloc en mouvement entre les images va engendrer un champ de vecteurs de mouvement fortement corrélés qui peut être exprimé de la façon suivante :

$$\overrightarrow{MV}_n^{-k} \approx \overrightarrow{MV}_n^{-k_1} + \overrightarrow{MV}_{n-k_1}^{-(k-k_1)}, \tag{4.3}$$

et illustré à la figure 4.18. Dans l'équation 4.3, $\overrightarrow{MV}_n^{-k}$ représente le vecteur de mouvement de l'image n pointant sur l'image référence $(n-k)$.

Plusieurs méthodes exploitent cette corrélation pour réduire les calculs lors de l'estimation de mouvement multi-référence [SS04, SS06, CCLC04]. Dans leur algorithme, Su et Sun [SS04, SS06] réalisent d'abord l'estimation de mouvement complète sur l'image référence la plus proche temporellement de l'image courante. Pour les autres images référence, les vecteurs de mouvement sont estimés à partir de la formule exprimée dans l'équation 4.3. La dernière étape consiste à raffiner la position autour de ces vecteurs de mouvement estimés (± 1 pixel). La méthode proposée par Chen et al [CCLC04] ne diffèrent que dans la

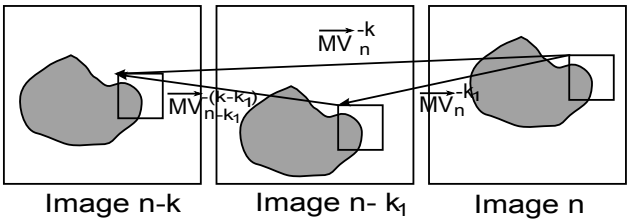


FIG. 4.18 – Illustration de la continuité du mouvement.

dernière étape de raffinement des vecteurs de mouvement composés.

4.2.5.3 Relation entre la fenêtre de recherche temporelle et la taille des partitions

Afin d'exploiter la relation entre la fenêtre de recherche temporelle et la taille des partitions, des tests ont été réalisés [KMJ06] où le nombre d'images référence assignées pour les partitions de petites tailles est plus important que pour celles plus grandes. Ils utilisent un filtre adaptatif (aux moindres carrés) afin de prédire la qualité et le débit. À partir de ces estimations, le gradient débit-distorsion est obtenu et ainsi ils peuvent déterminer le nombre d'images référence optimales, c'est-à-dire, le nombre d'images référence qui maximisent la réduction des calculs tout en situant l'augmentation du débit inférieure au seuil fixé. Les études réalisées par Jiao et al [JZB06] montrent que dans plus de 75% des cas, le vecteur de mouvement estimé pointe vers l'une des trois images référence les plus proches temporellement de l'image courante. Ils étudient également la relation entre la meilleure image référence et les différentes partitions. La meilleure image référence obtenue avec une partition de taille 16×16 pixels est la même que celle obtenue avec une partition de taille 16×8 pixels dans plus de 67% des cas. Cette probabilité atteint plus de 83% entre les partitions de taille 8×8 et 8×4 pixels. En se basant sur ces résultats, ils proposent une méthode d'estimation de mouvement où pour chaque partition seules les trois images référence les plus proches temporellement de l'image courante sont testées, plus une liste adaptative d'images référence construite en fonction des résultats obtenus pour les partitions supérieures.

4.2.5.4 Sélection de l'image référence

Une séquence vidéo étant une représentation discrète d'une scène naturelle continue, la valeur d'un pixel est l'intensité de la lumière pendant l'intervalle de détection du capteur discret. Entre les images successives de la vidéo, le mouvement d'un objet peut correspondre exactement à la grille de sensibilité du capteur ou alors correspondre à un mouvement sous-pixélique. Dans le cas où le mouvement de l'objet correspond à la grille de sensibilité du capteur, les contours de l'objet seront nets. Et lorsque l'objet se déplace d'un nombre entier de pixels, l'objet sera identique entre les deux images et pourra être prédit en utilisant l'estimation de mouvement au pixel près. Dans le cas d'un mouvement sous-pixélique (par exemple au demi-pixel), les contours risquent d'être flous et les pixels correspondants n'auront pas la même intensité que les pixels originaux, ce qui rendra l'estimation de mouvement plus difficile. Les expériences réalisées par Chang et al [CAY03] montrent que le coût de l'estimation de mouvement à tendance à diminuer lorsque l'image référence et l'image courante sont discrétisées de la même façon, c'est-à-dire, que les objets ont les mêmes positions sous-pixéliques. Ils proposent une méthode d'estimation de mouvement dans laquelle ils identifient les images référence pour lesquelles les correspondants du macrobloc courant ont les mêmes positions sous-pixéliques. Si plusieurs images référence sont ainsi identifiées, l'estimation de mouvement est réalisée sur celle étant la plus proche temporellement de l'image courante.

4.2.5.5 Discussion

L'estimation de mouvement multi-référence permet de réduire la redondance temporelle entre les images successives d'une séquence vidéo. Le codeur H.264 autorise une telle estimation de mouvement pour les sept modes de prédiction inter. La prédiction entre les blocs s'en trouve significativement améliorée, cependant la complexité calculatoire augmente avec le nombre d'images référence utilisées. Les méthodes proposant d'optimiser cette estimation du mouvement multi-référence privilégient l'estimation de mouvement sur l'image référence la plus proche temporellement de l'image courante, ou réalisent des recherches partielles sur toutes les images référence stockées et affinent la recherche pour celle ayant le coût minimal. Aucune méthode ne propose de réaliser une estimation de mouvement multi-référence avec des images référence sélectionnées en fonction des caractéristiques spatiales du bloc courant à estimer, afin d'améliorer la qualité. Seule la méthode proposée par Chang et al [CAY03] choisit la meilleure référence parmi celles qui sont stockées, en fonction de la position sous pixélique de l'objet dans l'image courante, dans l'optique de ne pas réaliser l'estimation de mouvement pour les autres images référence. Afin de prédire au mieux le macrobloc courant, il serait judicieux de choisir comme image de référence celle qui représente celui-ci le plus fidèlement. Or généralement, seules les images précédentes à l'image courante, exceptées pour les images B, sont utilisées comme référence. Suite à la pré-analyse de la vidéo et plus particulièrement de la segmentation spatio-temporelle, nous disposons d'informations sur le cycle de vie des objets (apparition, début et/ou fin du mouvement). Il est alors envisageable de sélectionner les images (futures ou passées) représentant au mieux ces objets et les utilisées comme référence pour la prédiction de l'objet contenu dans l'image courante.

4.2.6 Optimisation de la qualité

La plupart des algorithmes proposés dans la littérature essaient d'accélérer le procédé d'estimation de mouvement au détriment du débit et de la qualité. En effet, la méthode classique de recherche exhaustive produit les meilleurs résultats, puisque tous les déplacements sont testés. Au contraire, Mai et al [MYKP06] proposent une méthode afin de réduire le flux de données tout en préservant la qualité perceptive. Leur approche est une méthode d'estimation de mouvement basée sur SSIM (*Structural Similarity*) [WBSS04]. SSIM est une métrique de qualité basée sur la mesure de la dégradation de l'information structurelle composée de la luminance, du contraste et de la structure. SSIM est intégrée au sein de leur codeur H.264 en lieu et place de la traditionnelle somme des différences absolues (SAD). En moyenne, le flux de données est réduit de 20% et les temps de calculs diminuent de $2,5\%$ par rapport au codeur de référence, tout en maintenant la même qualité perceptive de la vidéo reconstruite.

La décision de sélection du mode de prédiction inter peut également être basée sur un facteur de sensibilité spatio-temporel [BHT+07]. Dans cette méthode, un facteur de pondération est introduit au sein de la métrique de distorsion. La somme des erreurs quadratiques au carré entre le bloc courant et le bloc prédit est pondérée par un facteur de sensibilité spatio-temporel. Celui-ci est obtenu en appliquant la fonction de sensibilité au contraste (CSF) au déplacement de chaque bloc. La méthode permet de réduire le débit de près de 5% pour une qualité subjective équivalente.

4.2.7 Conclusion

Cette section a présenté un aperçu de plusieurs familles de méthodes d'optimisation de codage. Nous avons vu qu'il existe trois grandes familles de techniques permettant d'optimiser le codeur H.264. La pre-

mière d'entre elles regroupe les méthodes d'estimation de mouvement qui accélèrent la recherche du vecteur de mouvement. Celles-ci permettent de réduire la complexité calculatoire tout en maintenant la qualité perceptive par rapport à la méthode de recherche exhaustive du codeur H.264. La deuxième famille regroupe les techniques qui permettent de réduire les modes de prédiction testés (intra et/ou inter). Ces méthodes constituent un ensemble réduit de modes (parmi tous ceux possibles pour la prédiction inter et/ou intra du codeur H.264), en fonction des caractéristiques spatio-temporelles du macrobloc courant et de la corrélation avec son voisinage. Ainsi, seuls les modes les plus probables et susceptibles d'être retenus par la méthode de prédiction classique sont évalués. Cependant, aucune cohérence en termes de codage pour les macroblocs appartenant à un même objet n'est assurée. La dernière famille présente les techniques de sélection des images de référence (par défaut, le codeur utilise les cinq images précédant l'image courante). Les expériences réalisées permettent de mettre en évidence la prépondérance de l'image référence la plus proche temporellement de l'image courante. Ainsi, l'optimisation de l'estimation de mouvement multi-référence se consacre à réduire les recherches parmi les multiples images référence. Aucune méthode existante ne propose une sélection exacte de ou des images référence pour le macrobloc courant.

Le codeur H.264 ne possède donc pas les outils nécessaires pour prendre des décisions permettant d'assurer un codage cohérent en fonction du contenu spatial des images et de leur évolution temporelle. Cependant la diversité des outils de prédiction du codeur offre de réelles possibilités à condition d'être orientées en fonction d'informations de plus haut niveau. En effet, après notre étape de pré-analyse nous disposons d'informations variées sur la séquence vidéo. Les résultats des estimations du mouvement global et du mouvement local (tubes spatio-temporels) peuvent être utilisées pour détecter des changements importants dans la séquence vidéo et nécessitant la modification de la structure du GOP, telles que l'insertion d'une nouvelle image I, ou augmenter le nombre d'images B entre deux images P lorsque le mouvement estimé est faible. Les cartes de segmentation spatio-temporelle nous renseigne sur le cycle de vie des objets en mouvement. Ces informations peuvent ainsi être exploitées pour assurer une cohérence de codage pour le codage intra des macroblocs adjacents et appartenant à une même région et/ ou pour le choix des partitions lors du codage inter des macroblocs appartenant au même objet sur plusieurs images successives. De plus, les cartes de saillance renseignant sur l'attraction visuelle des différentes régions de l'image peuvent être utilisées pour adapter le pas de quantification en fonction de chaque macrobloc ou plutôt de sa saillance.

Conclusion

Ce chapitre a présenté le nouveau standard de codage vidéo H.264, norme de codage vidéo développée conjointement par VCEG et MPEG. Le codeur H.264 combine de nombreuses techniques nouvelles qui lui permettent de compresser beaucoup plus efficacement les vidéos que les normes précédentes et fournit plus de flexibilité aux applications dans un grand nombre d'environnements réseau. Dans ces fonctionnalités principales sont inclus :

- une compensation de mouvement pouvant être effectuée par rapport à plusieurs (jusqu'à seize) images de référence déjà codées ;
- une compensation de mouvement pouvant utiliser sept tailles de blocs différentes (16×16, 16×8, 8×16, 8×8, 8×4, 4×8 et 4×4) ;
- une précision au quart de pixel pour la compensation de mouvement, permettant une description très précise du déplacement des zones en mouvement ;
- un filtrage anti effet de bloc effectué dans la boucle de codage permettant de réduire les artefacts

caractéristiques du codage ;

– un codage arithmétique (CABAC), technique sophistiquée de codage entropique, qui produit d'excellents résultats en termes de compression mais possède une grande complexité.

Ces techniques, ainsi que plusieurs autres, conduisent le standard H.264 à dépasser de façon significative les performances des standards précédents, dans une grande variété de contextes et d'environnements d'applications. H.264 peut fonctionner souvent nettement mieux que le standard de codage vidéo MPEG-2 en obtenant la même qualité avec un débit réduit de moitié, voire plus. De nombreuses méthodes ont été proposées dans la littérature permettant d'optimiser les différentes étapes de prédiction du codeur :

– l'estimation de mouvement,
– la prédiction inter et intra,
– l'estimation de mouvement multi-référence.

Bien que la complexité calculatoire puisse être quelque peu réduite par l'utilisation de ces méthodes, les défauts du standard H.264 qui est un codeur fondé sur la partition spatiale des images en blocs sont toujours présents. Le problème vient du fait que le codeur H.264, tout comme ses prédécesseurs, prend des décisions à court terme et seulement selon des considérations de type signal (SNR). En effet, les différentes partitions possibles lors de la prédiction inter, peuvent à bas débit faire apparaître des artéfacts de codage. Si aucune cohérence de codage n'est appliquée pour coder un macrobloc au cours du temps, des effets de papillotement peuvent apparaître. Ceci étant dû à l'utilisation de différentes partitions ou d'une quantification variable faisant apparaître les frontières des macroblocs ou dégradant la texture du macrobloc par intermittence. L'issue possible à de telles dégradations est de contraindre le codeur à ne plus prendre simplement des décisions à court terme mais de le guider dans ses choix afin d'assurer la cohérence du codage d'un objet tout le long de sa durée de vie. Or, le codeur H.264 actuel ne possède pas les outils nécessaires pour analyser la vidéo et son contenu afin de prendre de telles décisions, l'obtention de telles informations doit être réalisée par une étape de pré-analyse avant codage de la vidéo. Les travaux présentés dans la première partie de ce rapport ont décrit notre méthode de pré-analyse, celle-ci permet d'obtenir des informations contextuelles sur la vidéo à coder. Ainsi, ces informations peuvent être exploitées et transmises au codeur afin de le guider dans ses choix de codage :

– structurer le GOP selon les caractéristiques spatio-temporelles de la vidéo ;
– assurer la cohérence de codage intra pour les macroblocs appartenant à la même région ;
– assurer la cohérence de codage inter pour les macroblocs appartenant à la même région (spatialement et temporellement) ;
– sélectionner les images référence optimales pour la prédiction temporelle ;
– quantifier en fonction de l'attraction visuelle des différentes régions (objets).

Le dernier chapitre de cette thèse présente la mise en œuvre au sein du codeur H.264 des stratégies de codage en fonction de certaines informations disponibles après notre méthode de pré-analyse.

Chapitre 5

Adaptation des paramètres du codage H.264 de la vidéo : méthodes et performances comparées avec le codeur de référence

Sommaire

Introduction **141**

5.1 Modification adaptative de la structure des GOP **142**

5.1.1 Approches de modification adaptative de la structure du GOP 143

5.1.2 Variation dynamique du nombre d'images B 144

5.1.3 Adaptation dynamique de la taille du GOP en fonction de l'évolution de l'activité temporelle 153

5.2 Codage adaptatif basé sur la saillance visuelle **157**

5.2.1 Principe général de la compression sélective 157

5.2.2 Objectif de la compression sélective directe 159

5.2.3 Carte de saillance dédiée pour le codage 159

5.2.4 Modification du coeur de codage 160

5.2.5 Résultats 161

5.3 Évaluation subjective de la qualité **163**

5.3.1 Méthodologie 163

5.3.2 Contenus évalués 168

5.3.3 Conditions d'observation 169

5.3.4 Observateurs 171

5.3.5 Résultats 171

Conclusion **178**

Introduction

Dans le chapitre précédent, nous avons vu que le codeur H.264 exploite de nombreuses techniques de prédiction lui permettant de dépasser de façon significative les performances des standards de codage vidéo

précédents. Mais tout comme ses prédécesseurs, le standard de codage H.264 pâtit des mêmes défauts. En effet, dans son principe même de fonctionnement, il ne dispose d'aucune information quant à l'évolution temporelle de la séquence, et est donc limité à prendre des décisions à court terme et seulement selon des considérations de type signal. Fondamentalement, le codeur H.264 est incapable d'analyser la vidéo et son contenu afin de prendre des décisions à moyen/long terme et d'assurer une cohérence de codage. Nous proposons de réaliser une telle analyse pour pouvoir ensuite exploiter les résultats de cette analyse dans le cadre du codage H.264 de la vidéo.

La première partie de ce mémoire a présenté notre méthode de pré-analyse de la vidéo. Celle-ci permet d'obtenir des informations de haut niveau caractérisant la séquence vidéo. Nous voulons donc dans ce chapitre illustrer l'intérêt en codage vidéo de cette approche. Parmi la multitude des applications possible de la pré-analyse, nous allons en présenter deux qui nous paraissent pouvoir être intéressantes. Nous envisageons à chaque fois une configuration « classique » de codage : après son analyse, le codeur doit encoder en une passe le plan vidéo. Ce chapitre décrit deux applications de codage en fonction des informations obtenues après notre méthode de pré-analyse de la vidéo, ainsi que l'évaluation de leurs performances. La première propose de modifier adaptativement la structure du GOP en fonction du contenu spatio-temporel de la séquence vidéo. Elle sera détaillée dans la première partie de ce chapitre. La deuxième, quant à elle, concerne une application de compression de la vidéo avec une qualité visuelle différenciée guidée par les cartes de saillance. Elle sera décrite en deuxième partie. La troisième et dernière partie de ce chapitre analysera les performances de ces deux méthodes, obtenues à partir de tests d'évaluation subjective de la qualité visuelle.

5.1 Modification adaptative de la structure des GOP

Comme nous l'avons vu dans le premier chapitre, les codeurs vidéo utilisent plusieurs modes de codage des d'images (*frames* en anglais) : les images I (images codées en mode intra), les images P (images codées par prédiction avec compensation de mouvement) et les images B (images bi-prédites). La succession des images I, P et B constitue un GOP. Celui-ci débute toujours par une image I. Ensuite, plusieurs images P suivent à des intervalles réguliers. Dans les « espaces » entre deux images P ou entre une image P et une image I, aucune, une ou plusieurs images B sont intercalées. Cette construction du GOP a une influence directe sur le débit du flux compressé car en moyenne, une image codée en mode I nécessite plus de bits pour son codage qu'une image codée en mode P, cette dernière nécessitant elle-même plus de bits qu'une image codée en mode B. Le choix du mode de codage (I,P ou B) pour les images est donc un compromis entre débit et qualité de reconstruction.

Les images I également appelées images clés (*keyframes* en anglais) sont utilisées comme images référence et sont la base de la prédiction par compensation de mouvement. Si certaines régions de l'image courante ne peuvent être reconnues dans les images clés correspondantes, l'efficacité de leur prédiction sera faible. Si l'intervalle entre les images clés (images I) est fixe (cas classique des codeurs vidéo), ce problème est inévitable. Nous avons vu dans le chapitre précédent que le standard H.264 réalise l'estimation/compensation du mouvement à partir de plusieurs images référence, ceci permet de réduire légèrement le problème. Un codeur vidéo qui utilise une taille de GOP fixe ne peut s'adapter aux changements de scènes et aux variations temporelles de la séquence vidéo et donc utiliser plus de bits pour encoder ces images. Afin d'améliorer les performances de codage, la taille du GOP doit être dynamique de telle sorte que les images I soient insérées aux instants adéquats (présentant des images les plus riches en contenu qui pourront donc servir de référence). Nous ne nous intéressons pas ici à la problématique de la détection des changements de

scènes (des outils efficaces existent déjà), mais à la modification adaptative du GOP au sein du plan vidéo. Pour ce faire, il faut être capable de détecter les changements significatifs au sein de la séquence vidéo qui nécessitent l'insertion d'une image I.

La sélection du nombre d'images B entre une image I et une image P ou entre deux images P[1] est une décision prise au niveau du codeur qui affecte significativement le débit du flux compressé. Les codeurs fixent classiquement le nombre d'images B à un ou deux. Ce compromis est motivé par des travaux expérimentaux, qui montrent qu'en moyenne, une telle décision réduit le débit sans affecter négativement la qualité visuelle des séquences décodées. On pourrait cependant réduire encore plus le débit pour des séquences ayant peu de mouvement ou avec un mouvement panoramique de la caméra, en augmentant le nombre d'images B. La gestion dynamique du mode de codage (I, P ou B) pour une image est trop complexe pour les systèmes de codage classiques (usuels), ce qui les empêche de prendre pleinement avantage du codage avec un nombre variable d'images B. En effet, ce nombre approprié d'images B dépend des caractéristiques temporelles et spatiales du plan, il varie donc lorsque les caractéristiques de mouvement évoluent. Un autre aspect est l'augmentation de la complexité du calcul au niveau de l'encodeur si celui-ci doit déterminer le meilleur mode de codage (I, P ou B) pour l'image courante. L'approche radicale (exhaustive) qui teste toutes les combinaisons d'images B et choisit la combinaison qui minimise le débit, est bien entendue trop complexe pour être utilisée de façon opérationnelle.

En codage vidéo classique, le choix des images I et du nombre d'images B sont des paramètres déterminés avant le procédé d'encodage du plan vidéo, et ils ne sont donc pas évalués en fonction du contenu de la vidéo. Nous voulons donc montrer que si ces paramètres peuvent s'adapter dynamiquement durant le codage de la vidéo, des gains en termes d'efficacité de compression et/ou de qualité peuvent être obtenus.

5.1.1 Approches de modification adaptative de la structure du GOP

Nous pouvons trouver dans la littérature différentes approches de modification de la structure du GOP. Afin de détecter les changements importants au sein de la séquence vidéo plusieurs auteurs proposent d'exploiter les caractéristiques de mouvement issues de la phase d'estimation. Yuan et al. [YFZ04] proposent d'étudier les variations des vecteurs de mouvement. Ils calculent la différence (norme euclidienne au carré) des vecteurs de mouvement entre deux images et suivent leur évolution. À partir de seuils définis empiriquement, ils sélectionnent les modes de codage des images et/ou diminuent leur taux (*frame rate*). Gu et Zhang [GH03] évaluent l'importance temporelle d'une image et donc son mode de codage, à partir de la moyenne des vecteurs de mouvement et de leur homogénéité (translation globale ou zoom avant de la caméra). Lan et al. [LH97] teste aussi la moyenne des vecteurs de mouvement pour sélectionner le mode de codage. Ding et al. [DLY06] utilisent la somme des modules de vecteurs de mouvement pour détecter les changements importants au sein des séquences et y insérer les images I, plus précisément ils calculent (pour les comparer à des seuils) le rapport entre la somme des modules des vecteurs de mouvement de l'image courante et de l'image suivante, ainsi que le rapport entre les erreurs de prédiction (somme des différences absolues transformées) de ces deux images. Dumitras et Haskell [DH04] évaluent la similarité du mouvement en termes de vitesse et de direction pour décider du nombre d'images B à insérer. Ils mesurent également la similarité entre images successives pour détecter les changements de scène.

Les changements au sein des séquences vidéo peuvent également être détectés à partir de l'erreur de prédiction après estimation de mouvement [LH97]. Wang et al. [WWLS06] utilisent quant à eux la MAD et

[1] Dans la suite du chapitre, afin de ne pas surcharger les explications, l'expression « entre une image I et une image P ou entre deux images P » sera omise lorsque l'on mentionnera les termes « nombre d'images B ».

la SAD pour décider du mode de codage adéquat.

Lee et Dickinson [LD94] utilisent une méthode d'« allocation réduite de débit ». Les images P avec allocation réduite de bits sont insérées lorsque la différence (MSE) entre les images est supérieure à un seuil prédéfini. Un critère haut niveau est évoqué à ce niveau : l'exploitation du masquage temporel effectué par le SVH.

Bilan :

Les différentes méthodes de modification adaptative de la structure du GOP restent assez rudimentaires. Une première estimation/compensation du mouvement est d'abord réalisée. Ensuite à partir des vecteurs de mouvement et des erreurs de prédiction, le mode de codage (I,P ou B) de l'image concernée est arrêté. Nous avons vu dans le chapitre 3, qu'une méthode classique d'estimation de mouvement reste limitée. En effet, l'objectif de l'étape d'estimation de mouvement d'un codeur classique n'est pas de déterminer le mouvement des macroblocs mais plutôt de déterminer le bloc le plus semblable dans l'image de référence (dans les limites de la fenêtre de recherche). Les vecteurs de mouvement obtenus avec une telle méthode d'estimation de mouvement ne reflètent pas correctement le mouvement réel (3D relatif) de la séquence vidéo.

Par l'utilisation de notre méthode de pré-analyse spatio-temporelle des séquences vidéo, nous disposons d'informations pertinentes. En effet, le champ de vecteurs de mouvement obtenu avec notre méthode d'estimation du mouvement basée tubes spatio-temporels est plus lisse et corrélé avec le mouvement réel observé de la séquence vidéo. De plus, nous avons estimé les paramètres du mouvement global (modèle affine) à partir de ce champ de vecteurs. La modification adaptative de la structure du GOP peut donc être réalisée à partir de ces informations qui reflètent plus efficacement l'évolution spatio-temporelle du contenu des séquences vidéo.

5.1.2 Variation dynamique du nombre d'images B

5.1.2.1 Objectif

L'objectif de notre méthode est de faire varier dynamiquement le nombre d'images B en fonction de l'évolution spatio-temporelle du contenu de la séquence vidéo. Pour cela, il est nécessaire au préalable d'estimer les caractéristiques spatio-temporelles de la vidéo. Une vidéo qui possède peu ou pas de mouvement pourra être codée avec un nombre plus important d'images B, alors que dans le cas contraire (de forts mouvements), il sera nécessaire de diminuer le nombre d'images B. Il est important également de prendre en considération les caractéristiques spatiales de la séquence vidéo. En effet, une vidéo possédant un contenu spatial très riche (par ex. filmée en extérieur), même avec peu de mouvement, est susceptible de présenter des variations importantes. Une telle vidéo ne devra donc pas être encodée avec un nombre trop important d'images B. Pour ces différentes raisons, proposer une méthode optimale permettant de faire varier dynamiquement le nombre d'images B, implique de mesurer le mouvement au sein de la séquence vidéo, ainsi que ses caractéristiques spatiales. Avant de mettre en œuvre une telle approche, il est nécessaire de vérifier si la configuration optimale du GOP varie en fonction des séquences ou plus précisément en fonction de leur contenu. Pour cela, les séquences vont être premièrement encodées pour différents nombres d'images B, afin de déterminer la configuration optimale du GOP pour chaque séquence. Ensuite, à partir de ces résultats et de l'analyse spatio-temporelle des segments temporels, nous allons définir notre approche de variation dynamique du nombre d'images B.

5.1.2.2 Analyse des séquences

5.1.2.2.1 Tests de différentes configurations du GOP

Avant de déterminer les paramètres de notre méthode de variation dynamique du nombre d'images B, nous avons d'abord réalisé des tests avec le codeur classique pour différents nombres d'images B : aucune image B, une image B, deux images B et trois images B. Notre intention est ainsi de vérifier si notre hypothèse de départ (faire varier le nombre d'images B) s'avère juste. De plus, nous disposerons ainsi d'une vérité de terrain quant au nombre optimal (global pour toute la séquence) d'images B, qui va nous aider à construire notre approche.

Nous avons testé cinq séquences ayant une résolution de 1280×720 pixels (*Crew*, *Knightshields*, *New Mobile and Calendar*, *Night* et *Parkrun*) et trois séquences ayant une résolution de 1920×1080 pixels (*Parkjoy*, *Tractor* et *Umbrella*) pour quatre débits (cf section 5.3.2.2) avec le codeur x264 [Vid06]. Les paramètres de codage sont les suivants :

- taille du GOP fixée à 25 images ;
- cinq images de référence pour l'estimation du mouvement ;
- fenêtre de recherche pour l'estimation de mouvement : 16×16 ;
- taille des partitions et des sous partitions pour l'estimation de mouvement : 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 et 4×4 ;
- codeur entropique : CABAC.

La figure 5.1 présente les courbes débit-distorsion en fonction du nombre d'images B pour les huit séquences testées (*Crew*, *Knightshields*, *New Mobile and Calendar*, *Night*, *Parkrun*, *Parkjoy*, *Tractor* et *Umbrella*). Le tableau 5.1 récapitule pour chaque séquence, la configuration du GOP optimale, c'est-à-dire, le nombre d'images B pour lequel la courbe débit-distorsion est supérieure aux autres. Ces premiers résultats soulèvent plusieurs questions :

- la première réside dans le fait que contrairement aux recommandations, l'utilisation de deux images B au sein du GOP, ne se révèle optimale que pour seulement trois séquences vidéo : *New Mobile and Calendar*, *Night* et *Knightshields.* Une première visualisation de ces trois séquences permet de constater qu'elles présentent des caractéristiques semblables. En effet, elles contiennent peu ou pas de mouvement (faibles mouvements de la caméra ou caméra fixe). De plus, les séquences *Knightshields* et *New Mobile and Calendar* ont été tournées en intérieur ainsi elles ne sont pas sujettes aux changements dus aux variations de la luminance. La séquence *Night* quant à elle a été prise de nuit dans une rue où l'arrière plan est constitué de bâtiments et d'un éclairage artificiel. Ainsi, pour ces trois séquences, les changements entre les images successives sont principalement dus aux mouvements des objets ou des personnes.
- la seconde concerne le nombre d'images B optimal qui apparaît dépendant du contenu de la séquence. De plus, notons que sur les huit séquences vidéo testées, aucune configuration particulière ne se dégage des autres.

Ces premiers tests effectués avec le codeur x264 semblent montrer que l'approche envisagée de modification dynamique du nombre d'images B peut être pertinente. En effet, les résultats obtenus indiquent que le nombre d'images B optimal dépend du contenu de la séquence. C'est pourquoi, à partir des informations obtenues après notre méthode de pré-analyse, nous voulons modifier dynamiquement le nombre d'images B, non plus globalement pour toute la séquence, mais localement pour chaque segment temporel de neuf

images.

Nom	Configuration optimale du GOP
New Mobile and Calendar	2 images B
Night	2 images B
Knightshields	2 images B
Crew	1 image B
Parkrun	1 image B
Parkjoy	aucune image B
Tractor	aucune image B
Umbrella	aucune image B

TAB. 5.1 – Tableau récapitulatif des configurations optimales du GOP obtenues avec le codeur classique x264 pour les huit séquences vidéo testées.

5.1.2.2.2 Caractérisation spatio-temporelle des séquences et des segments temporels

Nous voulons faire évoluer dynamiquement le nombre d'images B au niveau des segments temporels (neuf images). Nous souhaitons ici récapituler les informations fournies par notre méthode de pré-analyse, qui vont nous être utiles (pour caractériser spatio-temporellement le segment).

Dans le chapitre 3, nous avons présenté cette méthode de pré-analyse réalisant une estimation de mouvement basée sur des tubes spatio-temporels. Nous disposons alors d'informations, telles que le vecteur de mouvement de chaque tube spatio-temporel, ainsi que l'erreur globale EQM_G obtenue par la somme des quatre erreurs quadratiques moyennes (EQM), chacune entre le bloc courant et ses blocs correspondants des images passées et futures (équation 3.1).

Chaque segment temporel est donc caractérisé par deux mesures qui reflètent son activité spatio-temporelle :

– l'information temporelle du segment ST est mesurée par la moyenne des vecteurs de mouvement de ses tubes spatio-temporels :

$$IT(ST) = \frac{1}{nb_{Tubes} \times taille_{ST} \times fr} \sum_{t=1}^{nb_{Tubes}} \sqrt{V_x(t)^2 + V_y(t)} \quad \text{en pixels/s} \tag{5.1}$$

où :

– nb_{Tubes} est le nombre de tubes spatio-temporels du segment temporel ST,
– $taille_{ST}$ est la taille du segment temporel et ici est égal à neuf images,
– fr est le taux d'affichage des images de la séquence,
– V_x, V_y sont les composantes horizontale et verticale des vecteurs de mouvement.

– l'information spatiale IS d'un segment temporel ST est mesurée par la moyenne des erreurs globales EQM_G obtenues lors de l'estimation de mouvement des tubes spatio-temporels du segment temporel ST considéré :

$$IS(ST) = \frac{1}{nb_{Tubes} \times nb_{IR}} \sum_{t=1}^{nb_{Tubes}} EQM_G(t) \quad \text{en EQM/pixel}$$

où nb_{IR} est le nombre d'images de référence utilisée lors de l'estimation du mouvement des tubes spatio-temporels. Dans notre cas, $nb_{IR} = 4$ (voir section 3.2).

Nous avons calculé les valeurs de IS et IT de chaque segment temporel des huit séquences tests. Les

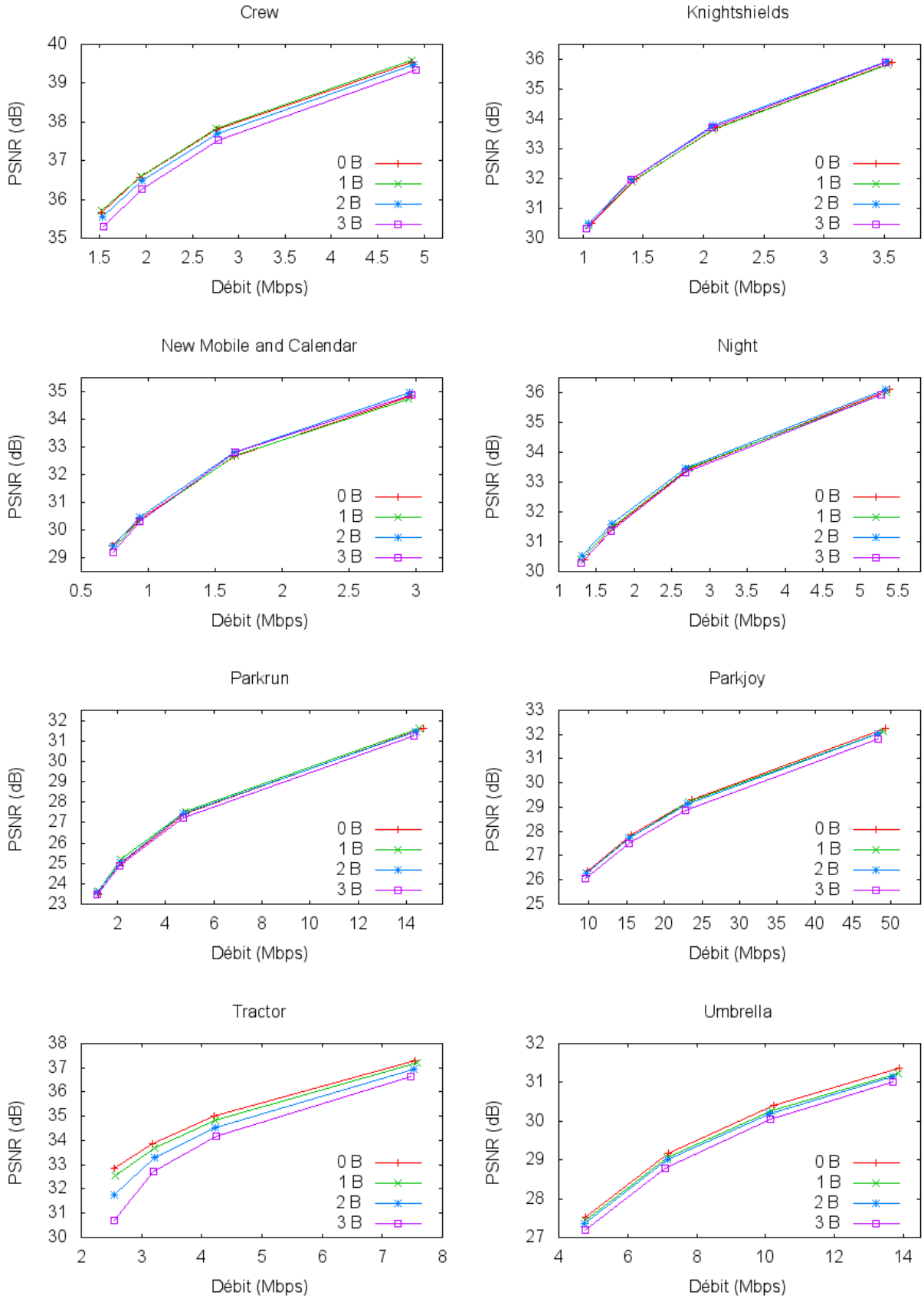


FIG. 5.1 – Courbes débit-distorsion des huit séquences vidéo testées et obtenues avec le codeur x264 en fonction du nombre d'images B insérées entre deux images P.

Séquence	IS (EQM/pixel)	IT (pixels/s)
Night	12.1	132
Knightshields	13	140.5
New Mobile and Calendar	18.5	138
Crew	12.6	279.5
Tractor	18.3	669.5
Parkrun	47.2	107
Umbrella	50.1	76
Parkjoy	87.1	314.5

TAB. 5.2 – Valeurs de IS et IT pour les huit séquences vidéo.

moyennes des valeurs de IS et de IT de tous les segments temporels des séquences sont données dans le tableau 5.2. Afin de ne pas être dépendant du format des séquences (dans notre cas, 1280×720 ou 1920×1080), les valeurs de IT des séquences vidéo au format 1920×1080 ont été normalisées (divisées par $1,5$). Les résultats des indices correspondent bien à ce que nous constatons visuellement dans les séquences. Une première lecture de ce tableau nous permet de constater que la séquence *Parkjoy*, qui possède en effet un contenu très détaillé, a la valeur d'IS la plus grande $(87,1)$. La séquence *Parkrun* qui est également très détaillée a une valeur d'IS égale à $47,2$. La séquence *Tractor* ayant la valeur d'IT la plus élevée, a effectivement une très forte activité temporelle.

Nous illustrons figure 5.2 a) les couples (IS, IT) des segments temporels et les couples (IS, IT) des séquences entières à la figure 5.2 b).

On effectue à présent une analyse en recoupant les résultats du tableau 5.1 et de la figure 5.2. On constate que les séquences *Knightshields*, *New Mobile and Calendar* et *Night,* pour lesquelles la configuration optimale du GOP est avec deux images B, ont des valeurs de IS et IT faibles.

Inversement, les séquences *Tractor* et *Parkjoy*, pour lesquelles la configuration optimale du GOP est avec zéro image B, ont les valeurs du couple (IS, IT) les plus élevées.

La séquence *Crew*, qui obtient les meilleurs performances avec une seule image B, a une valeur de IS faible et une valeur de IT moyenne.

Enfin, les séquences *Parkrun* et *Umbrella* qui ont des valeurs de IS et IT très proches, ont des configurations optimales du GOP différentes. Pour la séquence *Parkrun* les meilleures performances sont obtenues avec une image B, alors que pour la séquence *Umbrella* la configuration optimale du GOP consiste à n'insérer aucune image B.

L'analyse d'une séquence vidéo est donc réalisée à partir de deux mesures qui reflètent son activité temporelle IT et son activité spatiale IS. La caractérisation globale des séquences vidéo (un seul couple (IS, IT) par séquence vidéo), nous a permis d'identifier les relations entre l'activité spatio-temporelle et la configuration optimale de la structure du GOP. Le but est maintenant de proposer une classification des segments temporels en fonction de leurs caractéristiques spatio-temporels, permettant ainsi de déterminer pour chaque segment temporel le nombre d'images B optimal.

5.1.2.3 Classification des segments temporels

Nous disposons désormais de mesures permettant d'évaluer l'activité temporelle au sein d'un segment temporel ainsi que la richesse de son contenu spatial. Nous définissons neuf classes caractérisant l'activité spatio-temporelle et auxquelles correspondent un nombre d'images B : Ci, avec $i = 0..8$ indiquant le nombre d'images B. Chaque segment temporel peut à présent être classé en fonction de la valeur des mesures de son

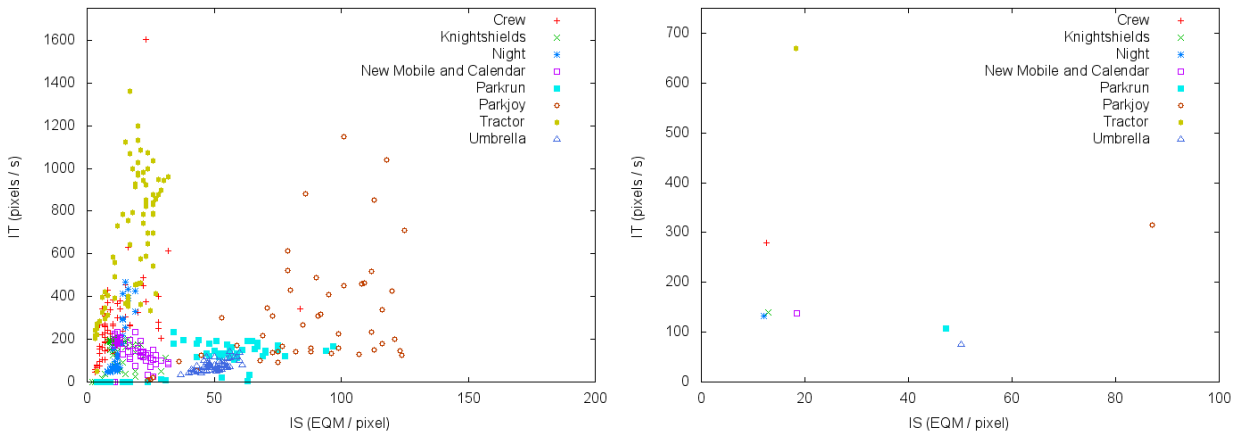


FIG. 5.2 – Couples (IS, IT) des huit séquences vidéo testées. a) un couple (IS, IT) pour chaque segment temporel. b) un couple (IS, IT) pour chaque séquence vidéo (moyenne des valeurs de IS et IT des segments temporels de la séquence).

activité temporelle $IT(ST)$ et de son activité spatiale $IS(ST)$. On utilise pour cela une partition de l'espace (IS, IT) telle que présentée figure 5.3. Les bornes des intervalles de l'activité temporelle ont été déterminées afin que l'activité temporelle entre deux images P soit constante :

$$\begin{cases} \text{si} \quad IT(ST) \geq \mu_{IT} & \text{alors } ST \in C_0 \\ \text{si} \quad \frac{\mu_{IT}}{i} > IT(ST) \geq \frac{\mu_{IT}}{i+1} & \text{alors } ST \in C_i,\ i = 1..7 \\ \text{si} \quad \frac{\mu_{IT}}{i} > IT(ST) & \text{alors } ST \in C_8 \end{cases}$$

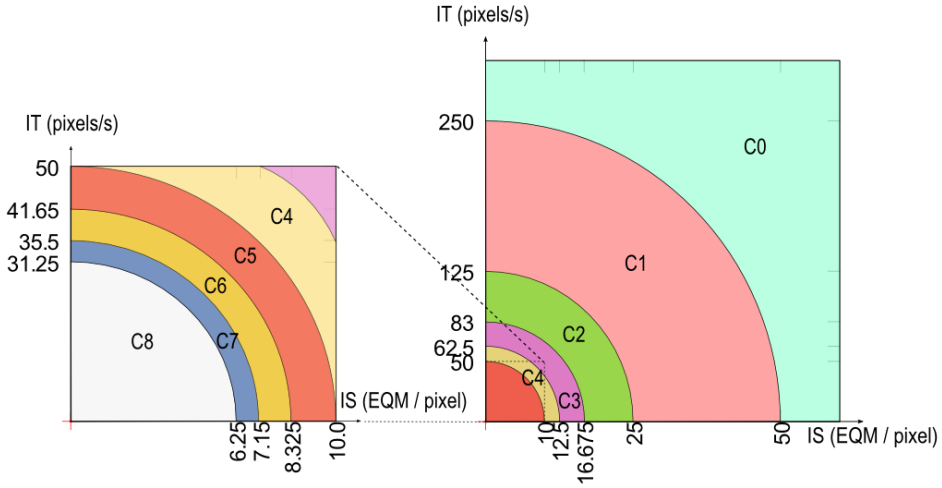


FIG. 5.3 – Classification des segments temporels afin d'adapter le nombre d'images B en fonction de leur activité spatio-temporelle (cas d'une séquence au format 1280×720 pixels, $\mu_{MV} = 5$).

Cependant, la valeur de l'activité spatiale est obtenue à partir de l'erreur de prédiction EQM_G lors de l'estimation de mouvement des tubes spatio-temporels. Ainsi, la valeur de IS traduit l'activité spatiale mais également l'efficacité ou non de l'estimation de mouvement. De ce fait, la variation temporelle de l'activité

spatiale n'est pas linéaire et dépend du type de texture. Malgré ces remarques, nous avons choisi d'adopter le même principe de délimitation des intervalles utilisé pour l'activité temporelle, c'est-à-dire, que la valeur de l'activité spatiale entre deux images P soit constante.

Ainsi les classes sont délimitées par des couronnes elliptiques. L'appartenance d'un segment temporel ST à une classe C_i est alors obtenue par :

$$\begin{cases} \text{si} & \frac{IT(ST)}{(\mu_{IT})^2} + \frac{IS(ST)}{(\mu_{IS})^2} \geq 1 & \text{alors } ST \in C_0 \\ \text{si} & \frac{IT(ST)}{(\frac{\mu_{IT}}{i+1})^2} + \frac{IS(ST)}{(\frac{\mu_{IS}}{i+1})^2} < 1 \quad \text{et} \quad \frac{IT(ST)}{(\frac{\mu_{IT}}{i+2})^2} + \frac{IS(ST)}{(\frac{\mu_{IS}}{i+2})^2} \geq 1 & \text{alors } ST \in C_i,\ i = 1..7 \\ \text{si} & \frac{IT(ST)}{(\frac{\mu_{IT}}{7+1})^2} + \frac{IS(ST)}{(\frac{\mu_{IS}}{7+1})^2} < 1 & \text{alors } ST \in C_8 \end{cases}$$

Les valeurs μ_{IT} et μ_{IS} ont été déterminées empiriquement sur l'ensemble des séquences :

$$\begin{cases} \mu_{IT} = 250 & \text{pour les vidéos au format } 1280 \times 720 \text{ pixels} \\ \mu_{IT} = 250 \times 1.5 & \text{pour les vidéos au format } 1920 \times 1080 \text{ pixels} \end{cases}, \quad \text{et } \mu_{IS} = 50$$

Ainsi est donc obtenue la classification des segments temporels, chaque segment temporel ST appartenant à une classe C_i ($i \in \{0;1;2;3;4;5;6;7;8\}$).

5.1.2.4 Résultats

5.1.2.4.1 Tests réalisés

Nous avons testé cette approche sur les cinq séquences ayant une résolution de 1280×720 pixels (*Crew*, *Knightshields*, *New Mobile and Calendar*, *Night* et *Parkrun*) et les trois séquences ayant une résolution de 1920×1080 pixels (*Parkjoy*, *Tractor* et *Umbrella*) pour quatre débits. La gamme de débits est différente d'un contenu à un autre (cf section 5.3.2.2). Nous avons directement utilisé le codeur x264 [Vid06]. Nous avons également une version modifiée du codeur x264 afin que celui-ci puisse appliquer les directives de codage issues de notre outil (faire varier dynamiquement le nombre d'images B). Les paramètres de codage sont les suivants :

- taille du GOP fixée à 25 images ;
- deux images B (entre deux images P ou entre une I et une P) pour le codage par défaut ;
- cinq images de référence pour l'estimation du mouvement ;
- fenêtre de recherche pour l'estimation de mouvement : 16×16 ;
- taille des partitions et des sous partitions pour l'estimation de mouvement : 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 et 4×4 ;
- codeur entropique : CABAC.

5.1.2.4.2 Évaluation de l'approche

La métrique de qualité utilisée est le PSNR (*Peak Signal Noise Ratio*). Le seuls avantage de cette métrique est sa simplicité. Les inconvénients sont nombreux et bien connus : corrélation réduite avec les tests visuels de qualité, incapacité de la métrique à prendre en compte des effets de masquage des dégradations... Bien que cette métrique ne soit pas une mesure absolue de la qualité, elle est suffisante pour effectuer une comparaison entre méthodes. La validation du fonctionnement de l'algorithme a donc été premièrement effectuée avec ce type de métrique. Le tableau 5.3 présente les gains en réduction de débit et en augmentation du

PSNR calculés avec la métrique de Bjontegaard [Bjo01], qui permet de calculer une différence moyenne entre deux courbes débit-distorsion :

- d'abord, une approximation des courbes passant par les points (Débit, Distorsion) est réalisée. La différence entre les courbes pouvant être dominée par les hauts débits, il est plus approprié d'utiliser une échelle logarithmique pour les débits ;
- les intégrales des deux courbes (log(Débit), Distorsion) sont donc déterminées ;
- la différence moyenne est obtenue en calculant la différence entre les deux intégrales divisée par l'intervalle d'intégration.

De cette façon, on obtient :

- la différence moyenne du PSNR en dB pour toute la gamme des débits considérés ;
- la différence moyenne du débit exprimée en % pour toute la gamme de PSNR.

La réduction de débit pour la méthode de variation dynamique du nombre d'images B (VDNIB) est en moyenne de $2,91\%$ par rapport au codage classique (x264) et le gain en PSNR est en moyenne de $0,11\,dB$. Les gains sont systématiques pour l'ensemble des séquences que l'on a utilisées, à l'exception des séquences *Night* et *Parkrun*. Les « gains en réduction du débit » pour ces deux séquences sont respectivement de $-3,03\%$ et $-2,11\%$, et les « gains en augmentation » du PSNR sont respectivement de $-0,11\,dB$ et de $-0,06\,dB$. En effet, le contenu de la séquence *Parkrun* tournée en extérieur est très détaillé (feuillages, reflets sur l'eau...) et par conséquent, la valeur moyenne des MSE obtenues lors de l'estimation des tubes spatio-temporels est élevée. Dans ce cas, notre méthode VDNIB va donc réduire à tort le nombre d'images B. Pour la séquence *Night*, la scène se déroule également en extérieur. Une caméra fixe filme une rue la nuit, où sont présentes de nombreuses enseignes lumineuses clignotantes. Le mouvement entre les images successives étant donc faible, notre méthode VDNIB augmente le nombre d'images B. Mais la variation de la luminance due aux enseignes lumineuses engendre des variations entre les images successives qui pénalisent la compression avec les décisions (augmentation du nombre d'images B) prises par notre méthode. Les gains maximaux en termes de réduction du débit et d'augmentation du PSNR sont obtenus pour la séquence *Tractor* et sont respectivement de $12,67\%$ et de $0,55\,dB$. Cette séquence est également tournée en extérieur, et possède un contenu très détaillé, mais contrairement à la séquence *Parkrun*, elle présente de forts mouvements et tire pleinement avantage de notre méthode VDNIB, puisque le nombre d'images B va être réduit. Néanmoins, en moyenne notre méthode VDNIB fournit des améliorations. Les figures illustrant la variation dynamique du nombre d'images B en fonction de l'information temporelle IT et de l'information spatiale IS des segments temporels pour les huit séquences sont présentées en annexe C.1.

La figure 5.4 présente les courbes débit-distorsion pour les huit séquences vidéo testées.

5.1.2.5 Limitations de l'approche

Notre méthode de variation dynamique du nombre d'images B obtient en moyenne de meilleures performances relativement à une approche de codage classique. Cependant, plusieurs limites de notre approche sont identifiées :

- l'activité temporelle est mesurée en calculant la moyenne des vecteurs de mouvement (des tubes spatio-temporels) du segment temporel. La valeur ainsi obtenue nous renseigne sur le mouvement moyen au sein du segment temporel. Les spécificités locales du mouvement ne sont pas prises en compte. En effet, les paramètres du mouvement global estimés lors de la pré-analyse de la séquence ne sont pas considérés pour la variation dynamique du nombre d'images B. On pourrait envisager de

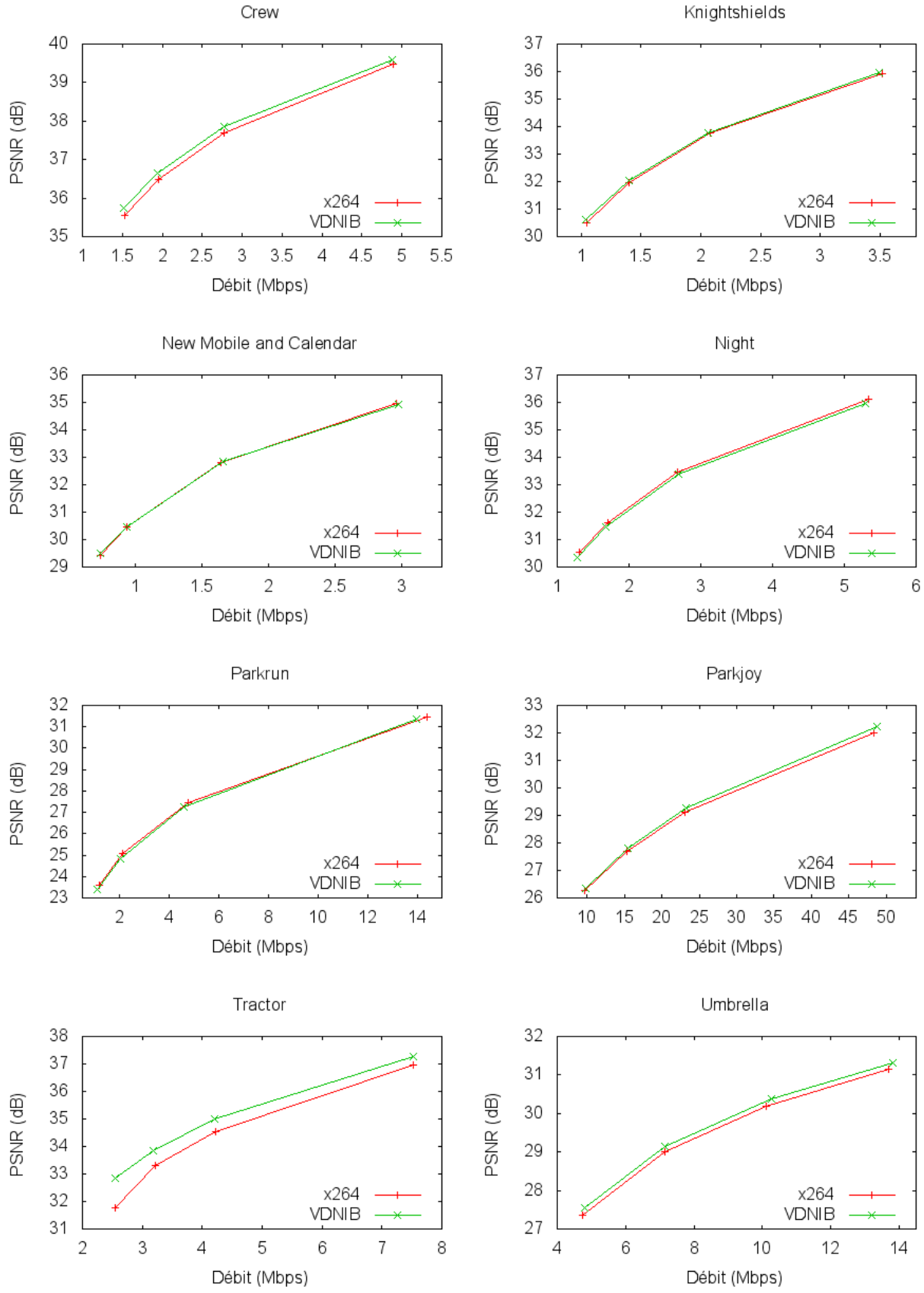


FIG. 5.4 – Prise en compte de l'activité spatio-temporelle : courbes débit-distorsion (PSNR en dB) pour les huit séquences vidéo testées avec notre méthode de variation dynamique du nombre d'images B (VDNIB) et un codage classique (x264).

Nom	Gain débit (%)	Gain PSNR (dB)
New Mobile and Calendar	0,41	0,02
Night	−3,03	−0,11
Knightshields	1,80	0,07
Parkrun	−2,11	−0,06
Umbrella	4,55	0,18
Parkjoy	3,55	0,11
Crew	5,43	0,18
Tractor	12,67	0,55
Moyenne	2,91	0,11

TAB. 5.3 – Prise en compte de l'activité spatio-temporelle : gains en réduction du débit et en augmentation du PSNR calculés avec la métrique de Bjontegaard [Bjo01] obtenus pour notre méthode de variation dynamique du nombre d'images B (VDNIB) par rapport au codage classique (x264).

découper les images en tranches (*slices*) et de faire varier localement (pour chaque tranche) le nombre d'images en fonction des caractéristiques locales du mouvement. Par exemple, dans le cas d'un zoom avant de la caméra, on pourrait découper l'image en tranches concentriques centrées sur le centre du zoom et faire varier le nombre d'images B pour chaque tranche (augmenter le nombre d'images B pour les tranches extérieures) ;

– la classification proposée n'est pas appropriée pour toutes les séquences, ou plus particulièrement la mesure utilisée pour caractériser l'activité spatiale. En effet, les séquences *Parkrun* et *Umbrella* qui ont des configurations optimales du GOP différentes (respectivement 1 et 0 image B dans le cas d'un codage classique, c'est-à-dire, pour un nombre d'images B fixe), ont des valeurs de IT et IS proches. Ainsi, on a constaté que pour la séquence *Parkrun*, notre méthode de variation dynamique du nombre d'images B déterminait un nombre insuffisant d'images B, puisque les performances en termes de débit-distorsion étaient supérieures avec le codage classique pour un nombre fixe d'images B.

5.1.3 Adaptation dynamique de la taille du GOP en fonction de l'évolution de l'activité temporelle

5.1.3.1 Objectif

Notre méthode d'adaptation dynamique de la taille du GOP modifie les tailles en fonction de l'activité au sein du plan vidéo. Son objectif est de tirer avantage des caractéristiques de la séquence vidéo pour améliorer les performances de codage, en identifiant les changements au sein du plan qui nécessitent l'introduction d'une nouvelle image I, car les images I servent d'ancrage (de référence) pour prédire (par compensation de mouvement) les autres. Le codage d'un plan débute toujours par une image I, on veut ensuite adapter la taille des GOP successifs de telle sorte que les images I soient placées judicieusement. Plus précisément, lorsque des changements sont détectés, il faut insérer une nouvelle image I pour améliorer la prédiction des images prédites après ce changement. L'analyse du mouvement au sein du plan, se fera au niveau de ses segments temporels.

5.1.3.2 Détection des changements au sein du plan

Rappelons que la séquence traitée a été préalablement découpée en plans homogènes. L'analyse du mouvement du plan permet de distinguer plusieurs cas :

- le plan contient de forts mouvements (les changements sont notables entre les images successives), il est alors préférable de réduire l'intervalle entre les images I ;
- le plan contient peu de mouvement (peu de variation entre les images), il est alors envisageable d'espacer davantage les images I. Ainsi, mécaniquement, la taille des GOP augmente ;
- le plus souvent le plan contient des mouvements modérés, dans ce cas, l'approche classique avec une taille fixe de GOP est conservée.

Après notre étape de pré-analyse nous disposons des caractéristiques de mouvement des segments temporels du plan (les vecteurs de mouvement des tubes spatio-temporels) et l'activité temporelle au sein de chaque segment est traduite par $IT(ST)$ (cf équation 5.1). Les graphiques représentant l'évolution des $IT(ST)$ des segments temporels des différentes séquences testées sont présentés en annexe C.2. À partir de l'analyse de ces graphiques, nous avons déterminé empiriquement deux seuils afin de détecter les changements critiques des IT des segments temporels successifs. Un seuil s_h pour distinguer les segments temporels qui contiennent de forts mouvement, et un seuil s_b pour ceux qui contiennent peu de mouvement. Ces seuils prennent les valeurs suivantes en fonction du format des séquences vidéos :

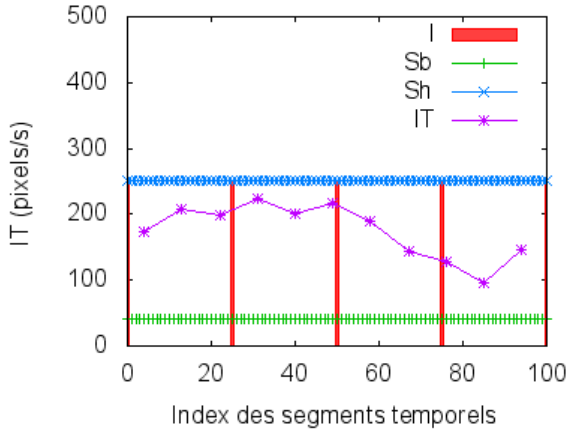
$$\begin{cases} s_h = 250\, pixels/s & \text{(séquences au format } 1280 \times 720\text{pixels)} \\ s_h = 250 \times 1,5 = 375\, pixels/s & \text{(séquences au format } 1920 \times 1080 \text{ pixels)} \end{cases}$$

$$\begin{cases} s_b = 40\, pixels/s & \text{(séquences au format } 1280 \times 720\text{pixels)} \\ s_b = 40 \times 1,5 = 60\, pixels/s & \text{(séquences au format } 1920 \times 1080\text{pixels)} \end{cases}$$

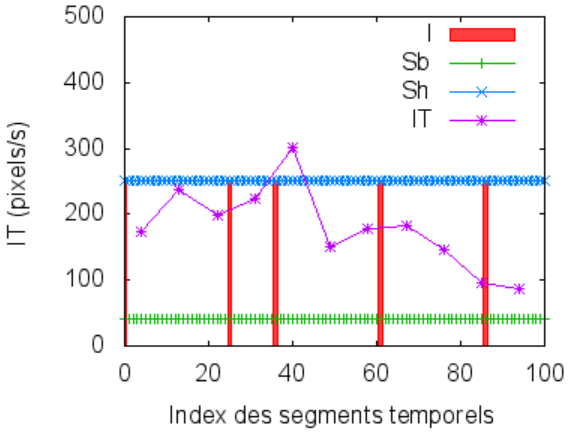
5.1.3.3 Analyse de l'évolution de l'activité temporelle

L'adaptation dynamique de la taille du GOP prend en considération différentes situations :

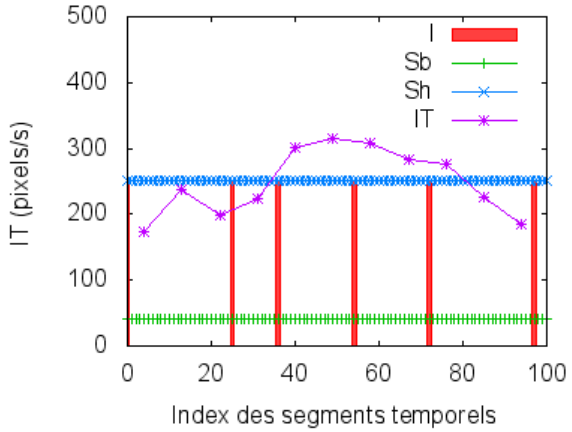
- $s_b > IT(ST) < s_h$: activité temporelle modérée, la taille des GOP reste inchangée (typiquement 25 images, cas de la figure 5.5 a)).
- $IT(ST) > s_h$: pour un segment temporel isolé. Un changement important s'est produit et, afin de prédire efficacement les images suivantes, le GOP suivant débutera à partir de ce segment temporel (figure 5.5 b)).
- $IT(ST) > s_h$: pour plusieurs segments temporels successifs. Il faut réduire la taille des GOP (typiquement à 18 images, cf figure 5.5 c)).
- $IT(ST) < s_b$: pour un segment temporel isolé. Le mouvement de ce segment temporel est faible (peu de changements entre les images). Le nouveau GOP commencera au niveau de ce segment temporel (figure 5.5 d)).
- $IT(ST) < s_b$: pour plusieurs segments temporels successifs (mouvement faible au sein du plan). A priori, il serait envisageable d'augmenter la taille des GOP dans une telle situation. Cependant, l'application concernée (codage de la TVHD) implique que l'intervalle entre deux images I ne peut pas être trop important (ex. quand le spectateur change de chaîne, le décodeur attend la prochaine I avant de recommencer à décoder la vidéo). Dans ce contexte (pour une qualité de service suffisante), nous avons fixé le temps de latence maximal à une demi seconde, ce qui correspond à une taille de GOP maximale de 25 images. Finalement, dans le cas considéré, le nouveau GOP débute lorsque $IT(ST)$ devient inférieur au seuil, mais ensuite la taille du GOP reste fixe (typiquement 25 images) (figure 5.5 e)).



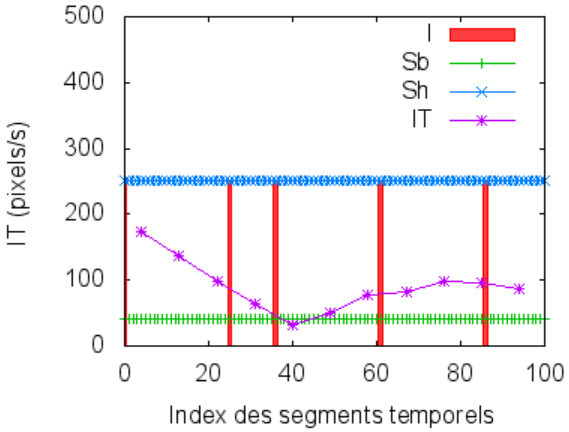
a) cas normal : taille du GOP fixe = 25 images



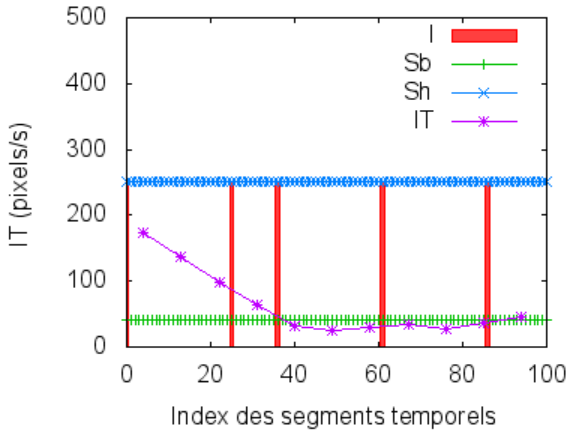
b) $IT(ST) > s_h$ pour un segment temporel isolé : un nouveau GOP débute à l'origine (image 36) de ce segment



c) $IT(ST) > s_h$ pour plusieurs segments temporels successifs : taille du GOP réduite = 18 images



d) $IT(ST) < s_b$ pour un segment temporel isolé : un nouveau GOP débute à l'origine (image 36) de ce segment



e) $IT(ST) < s_b$ pour plusieurs segments temporels successifs : taille du GOP fixe = 25 images

FIG. 5.5 – Évolution de l'indice d'activité temporelle : adaptation dynamique de la taille du GOP (en fonction de $IT(ST)$).

5.1.3.4 Résultats

5.1.3.4.1 Tests réalisés

Nous retrouvons nos cinq séquences *Crew*, *Knightshields*, *New Mobile and Calendar*, *Night* et *Parkrun* et les trois séquences *Parkjoy*, *Tractor* et *Umbrella*, le codeur x264 classique et un autre modifié gérant l'adaptation dynamique de la structure du GOP (MASGOP), c'est-à-dire, avec variation dynamique du nombre d'images B en fonction de l'activité spatio-temporelle du plan, ainsi que l'adaptation dynamique de la taille du GOP en fonction de l'évolution de l'activité temporelle. Les conditions de tests sont les mêmes que celles décrites à la section 5.1.2.4.1.

5.1.3.4.2 Évaluation de l'approche

Le tableau 5.4 présente les gains en réduction de débit et augmentation du PSNR calculés avec la métrique de Bjontegaard [Bjo01] entre le codeur classique x264 et celui modifié intégrant la modification adaptative de la structure du GOP (MASGOP).

La réduction du débit pour la méthode MASGOP est en moyenne de $4,45\%$ par rapport au codage classique, et le gain en PSNR est en moyenne de $0,16 dB$. Les gains sont systématiques sur l'ensemble des séquences utilisées, à l'exception de *Parkrun* dont les « gains » en réduction du débit et de PSNR sont respectivement de $-0,1\%$ et $-0,01 dB$. Cela est toujours dû au contenu très riche de la séquence *Parkrun*, par conséquent, la valeur moyenne des MSE obtenues lors de l'estimation des tubes spatio-temporels est donc élevée. Les gains maximaux sont encore obtenus pour la séquence *Tractor* et sont respectivement de $10,94\%$ et de $0,48 dB$. Cette séquence présente de forts mouvements et tire donc pleinement avantage de la méthode MASGOP, puisque le nombre d'images B va être réduit ainsi que l'intervalle entre deux images I permettant une meilleure estimation/compensation de mouvement des images prédites.

Nom	Gain débit (%)	Gain PSNR (dB)
New Mobile and Calendar ;	9,15	0,32
Night	2,45	0,09
Knightshields	1,68	0,06
Parkrun	−0,1	−0,01
Umbrella	4,11	0,13
Parkjoy	2,83	0,09
Crew	4,5	0,13
Tractor	10,94	0,48
Moyenne	4,45	0,16

TAB. 5.4 – Prise en compte de l'évolution de l'activité temporelle : gains en réduction de débit et augmentation du PSNR calculés avec la métrique de Bjontegaard [Bjo01] obtenus pour notre méthode de modification adaptative de la structure du GOP (MASGOP) par rapport au codage classique (x264).

La figure 5.6 présente les courbes débit-distorsion pour les huit séquences vidéo testées. Pour la séquence *Parkrun*, on constate que malgré les pertes obtenues en termes de réduction de débit ($-0,1\%$) et d'augmentation du PSNR ($-0,01 dB$) avec la métrique de Bjontegaard, notre méthode MASGOP semble être plus performante que le codage classique x264 à haut débit. Les graphiques illustrant les types des images (I, P ou B) codées par notre méthode de modification adaptative de la structure du GOP sont présentés en annexe C.3.

Les résultats obtenus avec le PSNR permettent de valider provisoirement le fonctionnement de notre méthode de modification adaptative de la structure du GOP, en attendant la confirmation ou non par des

tests d'évaluation subjective de la qualité.

5.1.3.5 Limitations de l'approche

Bien que les performances de notre méthode de modification adaptative de la structure du GOP soient supérieures à celles d'une méthode classique de codage, elle soulève plusieurs questions :

- la décision d'insérer une nouvelle image I est prise en considérant seulement les informations des segments temporels précédents et du segment courant. On pourrait envisager d'utiliser une fenêtre d'analyse temporelle plus large (par exemple, trois segments temporels) centrée sur le segment temporel courant afin de tenir compte des évolutions passées et futures pour la prise de décision ;
- la position d'une nouvelle image I est arbitrairement au début du segment temporel. Nous aurions pu tout aussi bien la placer au centre du segment temporel. Cependant, l'analyse du mouvement effectuée de chaque chaque segment temporel détecte les changements mais ne donne pas d'indication sur l'instant précis de celui-ci. Il faudrait par exemple, étudier les EQM locales (équation 3.1), calculées entre les différentes images afin de placer correctement la nouvelle image I au sein du segment ;
- même si le mouvement au sein de plusieurs segments temporels successifs est faible, nous avons choisi de ne pas augmenter l'intervalle entre deux images I, afin de ne pas augmenter le temps de latence. Si cette contrainte est levée, on pourrait augmenter la taille des GOP.

5.2 Codage adaptatif basé sur la saillance visuelle

L'objet de cette section concerne l'application de la compression vidéo avec une qualité visuelle différenciée pilotée par les cartes de saillance. Ce type de compression est communément appelée compression sélective ou compression avec régions d'intérêt. Contrairement aux approches conventionnelles distribuant de façon homogène les ressources, la compression sélective répartit les ressources en bits de façon adaptée directement ou indirectement. Dans un contexte de compression avec pertes, la distribution adaptée des ressources de codage vise à accroître substantiellement la qualité globale perçue. L'idée est simple puisqu'elle consiste à favoriser la qualité des zones les plus importantes visuellement. Bien évidemment, cela nécessite de disposer d'informations a priori sur la scène à coder. Dans ce contexte d'études, nous couplons un schéma de compression sélective directe exploitant notre modèle de saillance visuelle décrit dans le chapitre 3.

5.2.1 Principe général de la compression sélective

Le principe d'un codage sélectif ainsi que ses grandes étapes ont été définis par E. Nguyen [Ngu95]. Il introduit deux notions, un a priori de sélection et un a priori de compression :

- l'a priori de sélection permet de définir les zones de l'image à privilégier. Ces zones peuvent être définies implicitement via des connaissances a priori sur le contenu des images (par exemple dans le cadre de la visiophonie, les images sont de type « tête et épaules »), ou de façon explicite, c'est-à-dire, les zones sont déterminées à partir de caractéristiques locales basées sur le signal image, d'opérateurs d'analyse (segmentation, classification, reconnaissance de formes,...), de critères psycho-visuels...
- l'a priori de compression caractérise d'une part la nature du codage et d'autre part le critère d'allocation des ressources de codage. Autrement dit, il faut mettre en œuvre cet a priori de sélection au niveau de la compression.

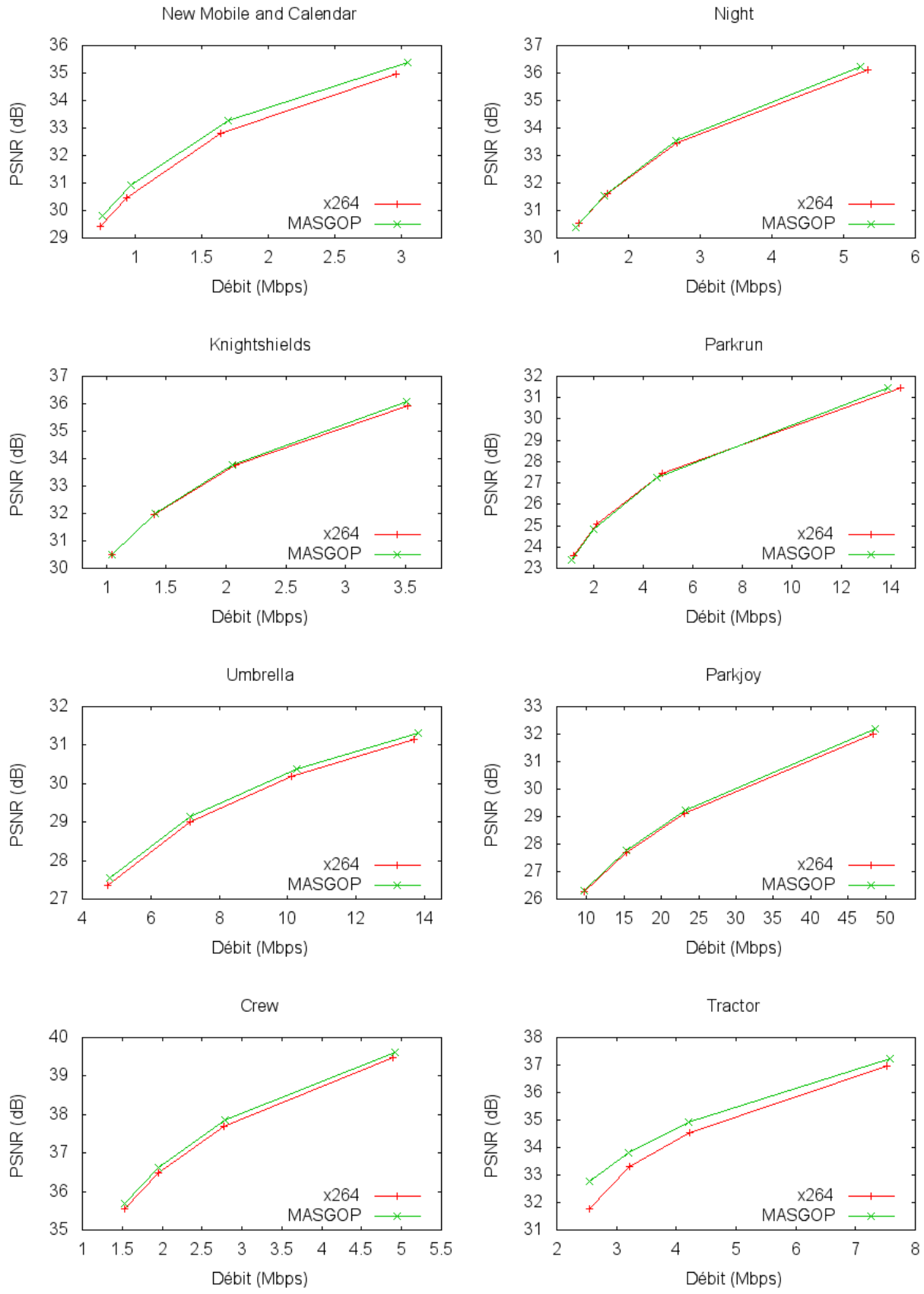


FIG. 5.6 – Prise en compte de l'évolution de l'activité temporelle : courbes débit-distorsion (PSNR en dB) pour les huit séquences vidéo testées avec notre méthode de modification adaptative de la structure du GOP (MASGOP) et un codage classique (x264).

On distingue également deux façons, qui peuvent être complémentaires, d'effectuer une compression sélective : la première dite indirecte consiste à modifier préalablement le signal à coder de façon à réduire la quantité d'information sur les zones de non intérêt. La deuxième dite directe, modifie directement le cœur de codage en fonction de la connaissance des régions d'intérêt. Une mise en œuvre de compression sélective directe est détaillée dans les sections suivantes.

5.2.2 Objectif de la compression sélective directe

L'objectif de la compression sélective directe est donc de contrôler la distribution des ressources binaires en fonction de l'intérêt visuel de chaque macrobloc afin d'accroître la qualité visuelle perçue. Il a été montré [Bra03] qu'une telle approche sur images fixes permet d'améliorer la qualité subjective, si les zones d'intérêt sont de tailles relativement faibles et si (pour le codage classique) la contrainte en débit provoque l'apparition d'artefacts sur les zones saillantes. La plupart du temps, le paramètre sur lequel on agit est la consigne de quantification : un macrobloc présentant un intérêt visuel faible sera quantifié plus grossièrement qu'un macrobloc ayant un intérêt visuel important.

5.2.3 Carte de saillance dédiée pour le codage

Il est nécessaire d'adapter la carte de saillance S^{SP-T} (équation 3.39) aux contraintes du codeur.. Le standard H.264 permet l'utilisation d'un pas de quantification par macrobloc. Seule la variation du pas de quantification entre deux macroblocs est codée et transmise, le surplus de bits nécessaire pour ce codage s'effectue au détriment du nombre de bits nécessaire pour coder les erreurs de prédiction, cela peut donc entraîner une perte de la qualité (visuelle). Des macroblocs voisins spatialement et/ou temporellement peuvent aussi avoir des indices de saillance différents. Cette hétérogénéité spatio-temporelle des indices de saillance (parfois d'un même objet) peut conduire après décodage à des artéfacts et des effets de papillotement. Si ces erreurs se situent au niveau d'un objet d'intérêt, elle seront gênantes pour les observateurs et pénaliseront la qualité globale perçue. Nous avons donc adapter la carte de saillance afin de prendre en considération ces différentes contraintes et aboutir à la carte de saillance modifiée, notée S^{SP-T}_{Mod}.

5.2.3.1 Modification de la carte de saillance

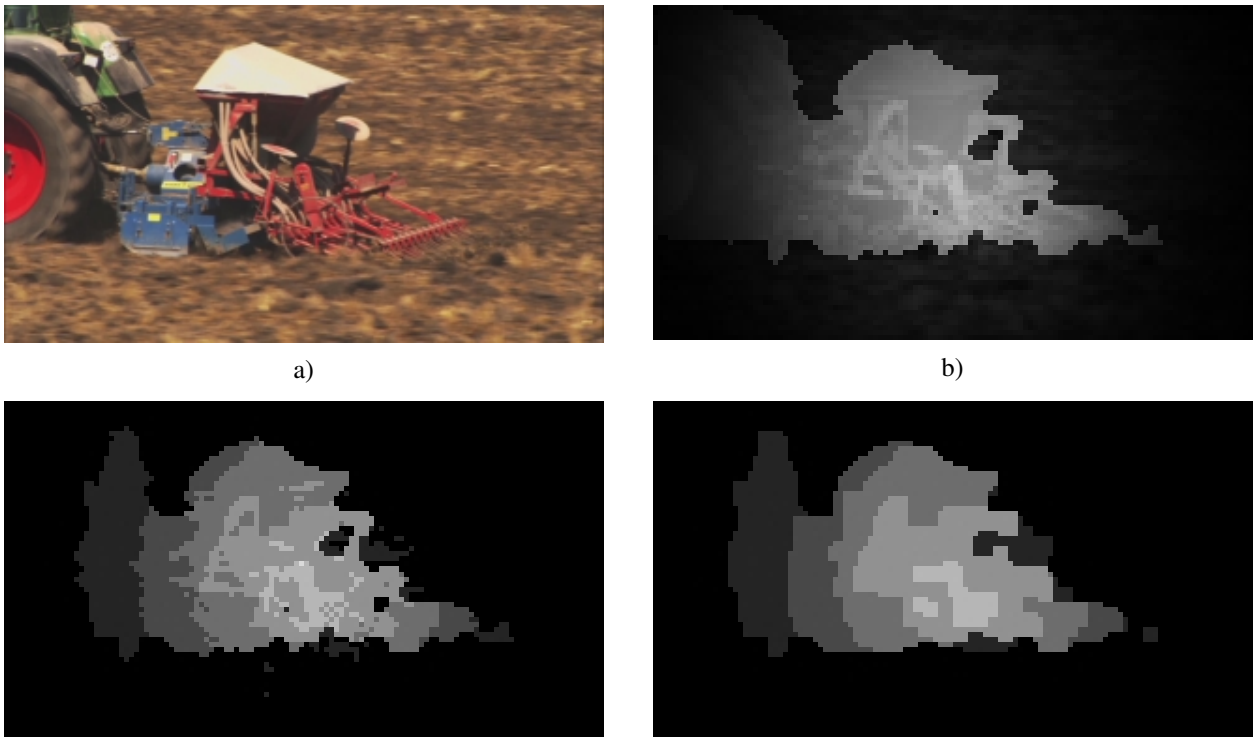
La simplification et l'homogénéisation de la carte de saillance s'effectuent en réalisant une quantification sur huit niveaux et suivie d'un filtrage morphologique via la combinaison de deux opérateurs morphologiques (ouverture puis fermeture), avec B un élément structurant 3×3. En adoptant la terminologie classique des filtres morphologiques, les notations sont :

- f : fonction originale,
- B : élément structurant,
- ε_B : érosion par l'élément structurant B,
- δ_B : dilatation par l'élément structurant B.

Les opérations morphologiques d'ouverture, érosion suivie d'une dilatation $\delta_B(\varepsilon_B(f))$, et de fermeture, dilatation suivie d'une érosion $\varepsilon_B(\delta_B(f))$, sont classiques. L'ouverture est croissante, anti-extensive et idempotente, ce qui lui confère sa nature de filtre morphologique. Sur une image à niveaux de gris, elle a pour effet de supprimer les parties claires trop petites pour contenir l'élément structurant. La fermeture sur une image

à niveaux de gris est croissante, extensive et idempotente, elle a l'effet dual de l'ouverture en supprimant les parties sombres de l'image trop petite vis à vis de la taille de l'élément structurant.

La figure 5.7 présente une image de la séquence *Tractor*, ainsi que les cartes de saillance obtenue après les différentes étapes de pré-traitement décrites.



a) b)

c) d)

FIG. 5.7 – Résultats de la modification de la carte de saillance pour une image de la séquence *Tractor* : (a) image originale, (b) carte de saillance originale S^{SP-T}, c) carte de saillance quantifiée sur huit niveaux, d) carte de saillance quantifiée et filtrée (combinaison d'une ouverture et fermeture) S^{SP-T}_{Mod}.

5.2.4 Modification du coeur de codage

Il s'agit de déterminer l'index de quantification de chaque macrobloc en fonction de la carte de saillance spatio-temporelle. Plus une zone est saillante et plus elle sera quantifiée finement et inversement. La carte de saillance S^{SP-T}_{Mod} nous indique la saillance de chaque macrobloc de l'image centrale d'un segment temporel (neuf images). Les cartes de saillance respectives des autres images du segment temporel (quatre images précédentes et quatre images suivantes) sont déduites à partir des informations de mouvement (projection temporelle dans le sens du mouvement des tubes de la carte de saillance centrale).

L'indice de saillance calculé pour un macrobloc varie entre 0 (saillance nulle) et 1 (très saillant). Afin de quantifier les macroblocs en fonction de leur indice de saillance, le pas de quantification doit être modifié en fonction de cet indice. Pour cela, on module la valeur du pas de quantification initialement calculé par le codeur. Si le macrobloc est saillant, on diminue la valeur initiale du pas de quantification. Dans le cas contraire, on conservera ce pas initial. Le contexte dans lequel nous nous situons (codage de la TVHD), implique que l'on doive respecter une consigne globale de débit. Le surplus éventuel de bits dû à la quantification fine des zones saillantes est contrôlé par l'optimisation débit-distorsion du codeur : par exemple, si le codage des macroblocs de la zone saillante a consommé trop de bits, le codeur ajuste le pas de quantification des images suivantes, afin de respecter cette consigne de débit.

Les notations que nous allons utiliser sont les suivantes :
- $QP_{final}(i)$, est le pas de quantification modulé du macrobloc i ;
- $QP_{init}(i)$, est le pas de quantification initial déterminé par le codeur pour l'image à laquelle appartient le macrobloc i ;
- QP_{min}, est le pas de quantification minimal ;
- $\overline{S_{Mod}^{SP-T}}$, est la valeur moyenne des indices de saillance des macroblocs de l'image à laquelle appartient le macrobloc i ;
- $S_{Mod}^{SP-T}(i)$, est l'indice de saillance du macrobloc i.

Le pas de quantification initial $QP_{init}(i)$ déterminé par le codeur est modulé en fonction de l'indice de saillance $S_{Mod}^{SP-T}(i)$ du macrobloc i. Chaque macrobloc i est donc associé à un pas de quantification modifié $QP_{final}(i)$. Le pas de quantification des macroblocs dont l'indice de saillance $S_{Mod}^{SP-T}(i)$ est supérieur à la valeur moyenne des indices de saillance de l'image $\overline{S_{Mod}^{SP-T}}$ sera diminué. Alors que celui des macroblocs dont l'indice de saillance est inférieur à la valeur moyenne des indices de saillance de l'image restera inchangé. Ainsi, les macroblocs dont l'indice de saillance est élevé seront quantifiés plus finement. La modulation du pas de quantification au sein du codeur est réalisée de la façon suivante :

$$\begin{cases} QP_{final}(i) = QP_{init}(i) + \left(\frac{\overline{S_{Mod}^{SP-T}} - S_{Mod}^{SP-T}(i)}{1-\overline{S_{Mod}^{SP-T}}}\right).\alpha.(QP_{init}(i) - QP_{min}) & \text{si} \quad \overline{S_{Mod}^{SP-T}} < S_{Mod}^{SP-T}(i) \leq 1 \\ QP_{final}(i) = QP_{init}(i) & \text{si} \quad 0 \leq S_{Mod}^{SP-T}(i) \leq \overline{S_{Mod}^{SP-T}} \end{cases} \tag{5.2}$$

où $0 \leq \alpha \leq 1$. Nous avons testé différentes valeurs du paramètre α : $\frac{1}{8}$, $\frac{1}{4}$ et $\frac{1}{2}$. La valeur retenue et offrant les meilleures performances en termes de débit-distorsion (PSNR en dB) est : $\alpha = \frac{1}{2}$.

5.2.5 Résultats

5.2.5.1 Tests réalisés

Cette approche a été testée sur les séquences *Crew*, *Knightshields*, *New Mobile and Calendar*, *Night* et *Parkrun* , et les séquences *Parkjoy*, *Tractor* et *Umbrella*. Les conditions de tests sont les mêmes que ceux réalisés à la section 5.1.2.4.1. Il s'agit ici de comparer la qualité de la zone d'intérêt lorsqu'on utilise un codage classique ou un codage adapté (compression sélective directe).

5.2.5.2 Évaluation de l'approche

Le tableau 5.5 présente les gains en réduction de débit et augmentation du PSNR calculés avec la métrique de Bjontegaard [Bjo01] entre le codeur classique x264 et le codeur modifié permettant la quantification adaptée en fonction des cartes de saillance (QA). Il indique les résultats obtenus avec la métrique de Bjontegaard pour la séquence entière, ou pour la région d'intérêt constituée des macroblocs dont l'indice de saillance $S_{Mod}^{SP-T}(i)$ est supérieur à la saillance moyenne de l'image $\overline{S_{Mod}^{SP-T}}$ (à laquelle ils appartiennent).

La qualité globale en terme de PSNR est diminuée lorsqu'on considère l'approche adaptée, $-0,02\,dB$ en moyenne. En contrepartie, la qualité des zones d'intérêt est améliorée de $0,03\,dB$ en moyenne. En réalité, la qualité des zones d'intérêt est améliorée pour seulement les séquences *Parkrun* ($0,01\,dB$), *Umbrella* ($0,10\,dB$), *Parkjoy* ($0,12\,dB$), *Crew* ($0,03\,dB$) et *Tractor* ($0,14\,dB$). Cela peut s'expliquer par la taille de la région saillante considérée ici, qui est constituée de tous les macroblocs dont l'indice de saillance $S_{Mod}^{SP-T}(i)$

Nom (taille de la région saillante)	Séquence entière		Région d'intérêt ($S_{Mod}^{SP-T}(i) > \overline{S_{Mod}^{SP-T}}$)	
	Gain débit (%)	Gain PSNR (dB)	Gain débit (%)	Gain PSNR (dB)
New Mobile and Calendar (20,92%)	−2,49	−0,09	−0,90	−0,04
Night (30,48%)	−3,38	−0,12	−1,89	−0,07
Knightshields (10,72%)	−3,02	−0,12	−1,46	−0,06
Parkrun (0,23%)	−0,81	−0,03	−0,03	0,01
Umbrella (24,3%)	2,34	0,07	3,04	0,10
Parkjoy (7,05%)	2,68	0,09	3,77	0,12
Crew (35,1%)	−0,36	−0,01	0,96	0,03
Tractor (24,20%)	0,89	0,05	2,81	0,14
Moyenne	−0,52	−0,02	0,79	0,03

TAB. 5.5 – Résultats du codage adapté (la région d'intérêt est constituée de tous les macroblocs dont l'indice de saillance modifiée $S_{Mod}^{SP-T}(i)$ est supérieur à la saillance moyenne de l'image $\overline{S_{Mod}^{SP-T}}$) : différence de Bjontegaard [Bjo01] entre les courbes débit-distorsion obtenues pour le codage classique (x264) et notre méthode de quantification adaptée (QA).

est supérieur à la saillance moyenne de l'image $\overline{S_{Mod}^{SP-T}}$, et représente donc en moyenne (pour les huit séquences) 19,1% des macroblocs. Si on considère une région d'intérêt trop importante, sa qualité ne sera pas forcément améliorée par rapport à une approche classique.

Le tableau 5.6 indique les résultats obtenus avec la métrique de Bjontegaard pour la séquence entière, ou pour la région d'intérêt constituée des macroblocs vérifiant : $S_{Mod}^{SP-T}(i) = S_{max}$, où S_{max} est la valeur maximale de la saillance de l'image à laquelle appartient le macrobloc i. Sur les huit séquences testées, la région d'intérêt réduite ($S_{Mod}^{SP-T}(i) = S_{max}$) ne représente plus que 4,1% des macroblocs.

Nom (taille de la région saillante)	Séquence entière		Région d'intérêt ($S_{Mod}^{SP-T}(i) = S_{max}$)	
	Gain débit (%)	Gain PSNR (dB)	Gain débit (%)	Gain PSNR (dB)
New Mobile and Calendar (10,4%)	−2,49	−0,09	−0,67	−0,03
Night (4,5%)	−3,38	−0,12	−0,39	−0,02
Knightshields (4%)	−3,02	−0,12	−0,84	−0,03
Parkrun (0,21%)	−0,81	−0,03	0,25	0,01
Umbrella (1,8%)	2,34	0,07	4,17	0,14
Parkjoy (1,8%)	2,68	0,09	4,42	0,14
Crew (6,9%)	−0,36	−0,01	2,74	0,09
Tractor (3,3%)	0,89	0,05	4,35	0,20
Moyenne	−0,52	−0,02	1,75	0,06

TAB. 5.6 – Résultats du codage adapté avec une région d'intérêt de taille réduite ($S_{Mod}^{SP-T}(i) = S_{max}$, où S_{max} est la valeur maximale de la saillance) : différence de Bjontegaard [Bjo01] entre les courbes débit-distorsion obtenues pour le codage classique classique (x264) et notre méthode de quantification adaptée (QA).

La qualité en terme de PSNR de ces zones d'intérêt réduites est améliorée de $0,06\,dB$, contre $0,03\,dB$ dans le cas précédent, où la région d'intérêt était constituée des macroblocs vérifiant $S_{Mod}^{SP-T}(i) > \overline{S_{Mod}^{SP-T}}$. La figure 5.8 illustre les densités de probabilité des indices de saillance modifiés S^{SP-T} pour les huit séquences. On constate que sur l'ensemble des séquences, la distribution des indices de saillance n'est pas uniforme, et que la dynamique des valeurs n'est pas totalement exploitée. En effet, les indices de saillance des ré-

gions d'intérêt restent modérés et de ce fait, la différence entre les régions saillantes et non saillantes n'est pas assez marquée. Le gain en PSNR pour ces régions d'intérêt n'est donc pas important, puisque le pas de quantification varie peu par rapport aux régions non saillantes. Les régions susceptibles d'attirer notre attention ne sont donc pas suffisamment mises en exergue par notre modèle d'attention visuelle, indiquant que la normalisation utilisée pour la carte de saillance spatiale et/ou l'étape de fusion des cartes spatiale et temporelle doivent être reformulées en se basant par exemple sur les résultats d'expériences oculométriques.

De plus, si l'on observe les résultats du tableau 5.6 individuellement, on constate que pour les séquences *New Mobile and Calendar*, *Night* et *Knightshields*, la qualité en terme de PSNR est diminuée non seulement globalement (séquence entière), mais également pour les zones d'intérêt. Il est intéressant de noter que ses trois séquences ont des valeurs de IT et IS faibles (voir le tableau 5.2), ce qui implique que les changements entre les images successives de la séquence sont faibles. Le surcoût engendré par la quantification plus fine des zones d'intérêt est alors trop important par rapport au gain produit par la quantification plus forte résultante de l'optimisation débit-distorsion réalisée par le codeur. Au contraire, pour les séquences *Umbrella*, *Tractor* et *Parkjoy*, le gain engendré par cette quantification plus forte des zones de non intérêt est plus profitable que le surplus de bits nécessaire pour coder les zones d'intérêt. En effet, ces séquences ont des valeurs élevées pour IT ou IS, traduisant respectivement une activité temporelle ou spatiale importante. Les séquences vidéo avec une activité spatiale importante présentent deux intérêts : tout d'abord, le codage de ces zones possédant un contenu détaillé est fortement consommateur de débit. Il est donc possible de venir puiser du débit dans les zones de non intérêt pour l'affecter à la zone saillante. Le second point intéressant est que ce type de contenu a une capacité intrinsèque à masquer le bruit de quantification et donc par voie de conséquence peut supporter une quantification plus forte sans introduire de défauts perceptibles.

Les figures 5.9 et 5.10 présentent les courbes débit-distorsion pour la séquence entière ou pour la région d'intérêt constituée des macroblocs vérifiant : $S_{Mod}^{SP-T}(i) = S_{max}$, où S_{max} est la valeur maximale de la saillance de l'image à laquelle appartient le macrobloc i.

Les résultats de compression obtenus par un codage classique et un codage adapté (QA) sont donnés figure 5.11 pour la séquence *Crew*, codée à $1{,}4$ Mbps. Il est particulièrement intéressant de considérer les erreurs engendrées par un codage classique (d) et un codage adapté (f). On constate, en effet, que pour le premier cas, l'erreur est uniforme alors que pour le second la distribution de l'erreur de codage est fonction de la carte de saillance. L'objectif que nous souhaitions est donc atteint : favoriser la qualité des zones d'intérêt au détriment des zones de non intérêt.

La figure 5.12 présente un résultat de codage obtenu sur la séquence *Tractor*. Les cartes d'erreurs montrent bien de l'amélioration du rendu de la zone saillante. Le semoir fixé à l'arrière du tracteur est mieux codé par la méthode proposée que par la méthode classique. La séquence *Tractor* est une séquence type pour laquelle le codage sélectif peut grandement améliorer la qualité perçue. En effet, cette séquence est composée d'un arrière plan contenant des zones fortement texturées, difficile à coder. Cependant ce type d'arrière plan est intéressant, car cette texture a une capacité intrinsèque à masquer le bruit de quantification et peut donc supporter une quantification plus forte sans introduire de défauts visuellement perceptibles.

5.3 Évaluation subjective de la qualité

5.3.1 Méthodologie

L'évaluation subjective de la qualité visuelle des séquences vidéo menée durant cette campagne de tests s'est déroulée en utilisant le protocole d'évaluation ACR (Absolute category rating). Ce protocole consiste

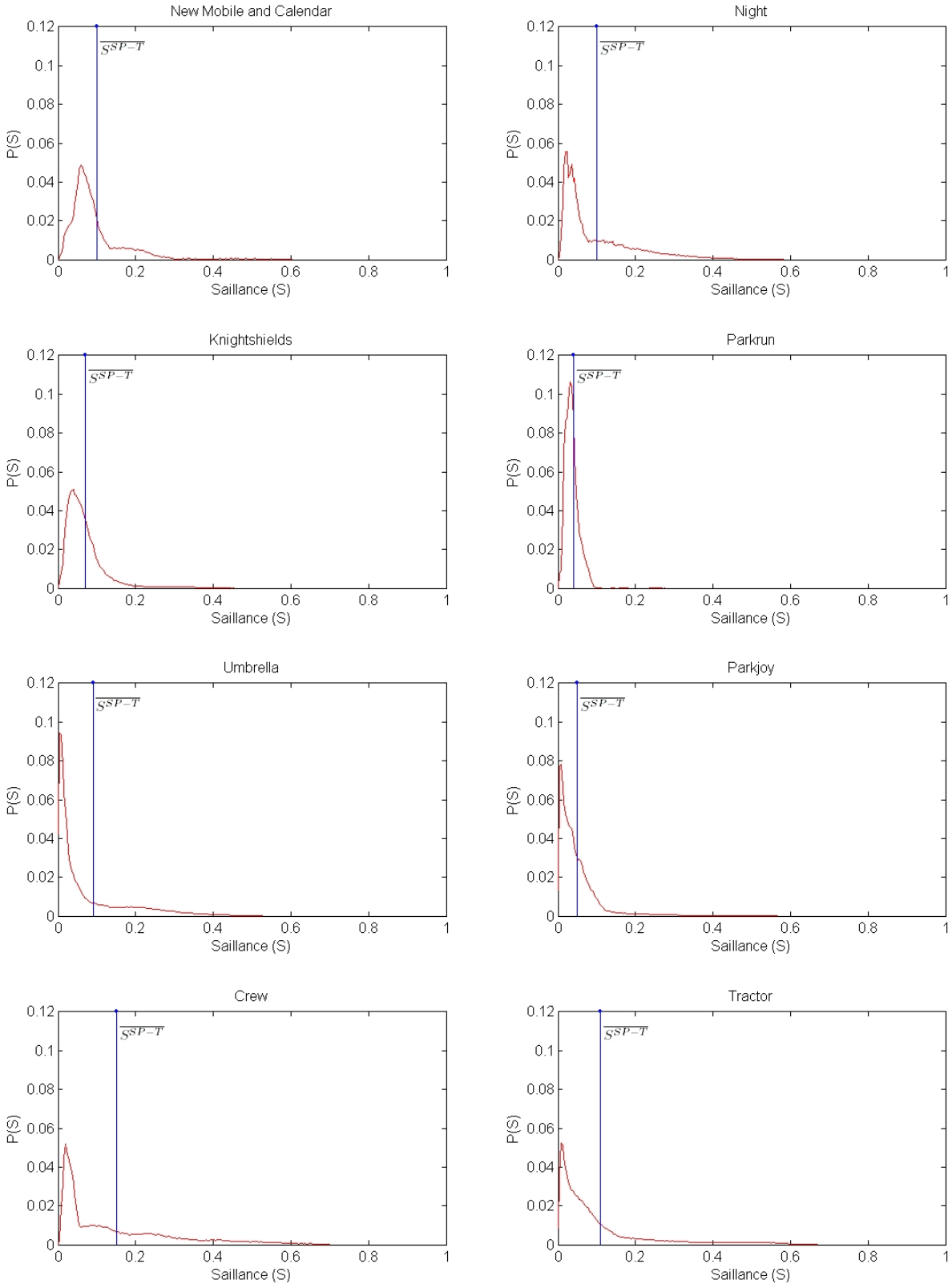


FIG. 5.8 – Fonctions de densité de probabilités des indices de saillance S^{SP-T} pour les huit séquences testées.

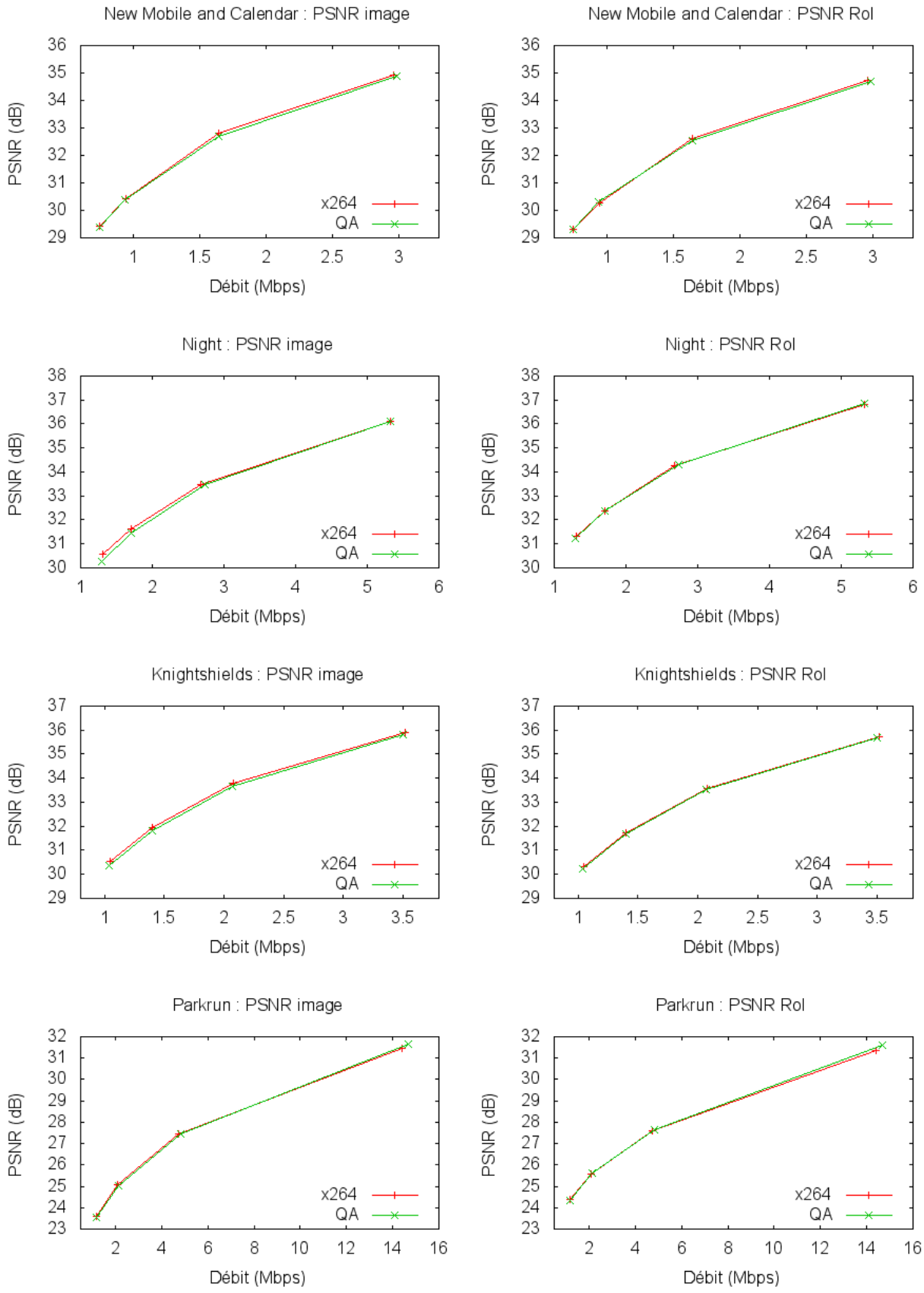


FIG. 5.9 – Résultats du codage adapté avec une région d'intérêt de taille réduite ($S_{Mod}^{SP-T}(i) = S_{max}$, où S_{max} est la valeur maximale de la saillance) : courbes débit-distorsion (PSNR) obtenues sur la séquence entière ou sur la région d'intérêt pour les séquences *New Mobile and Calendar*, *Night*, *Knightshields* et *Parkrun*.

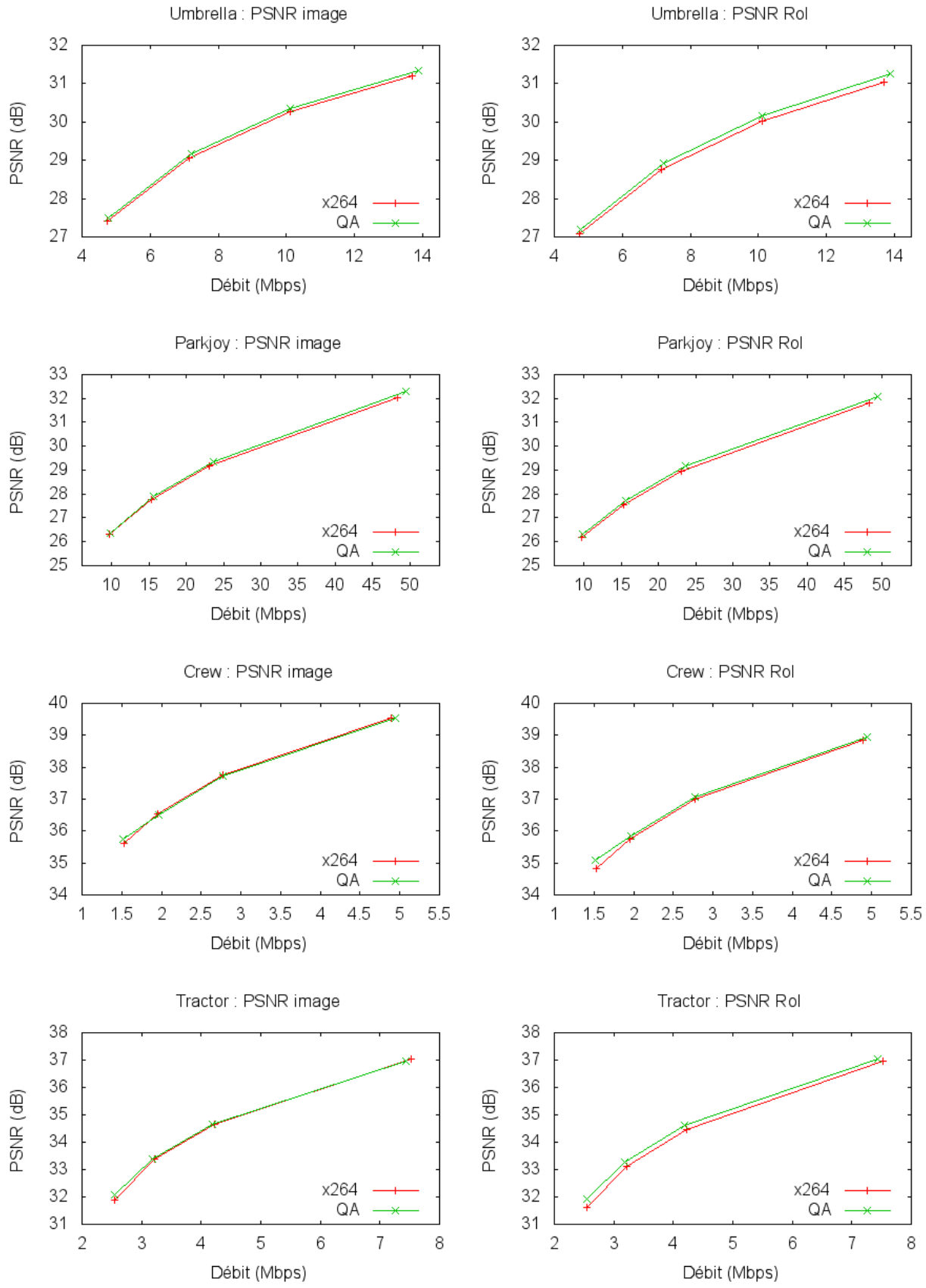
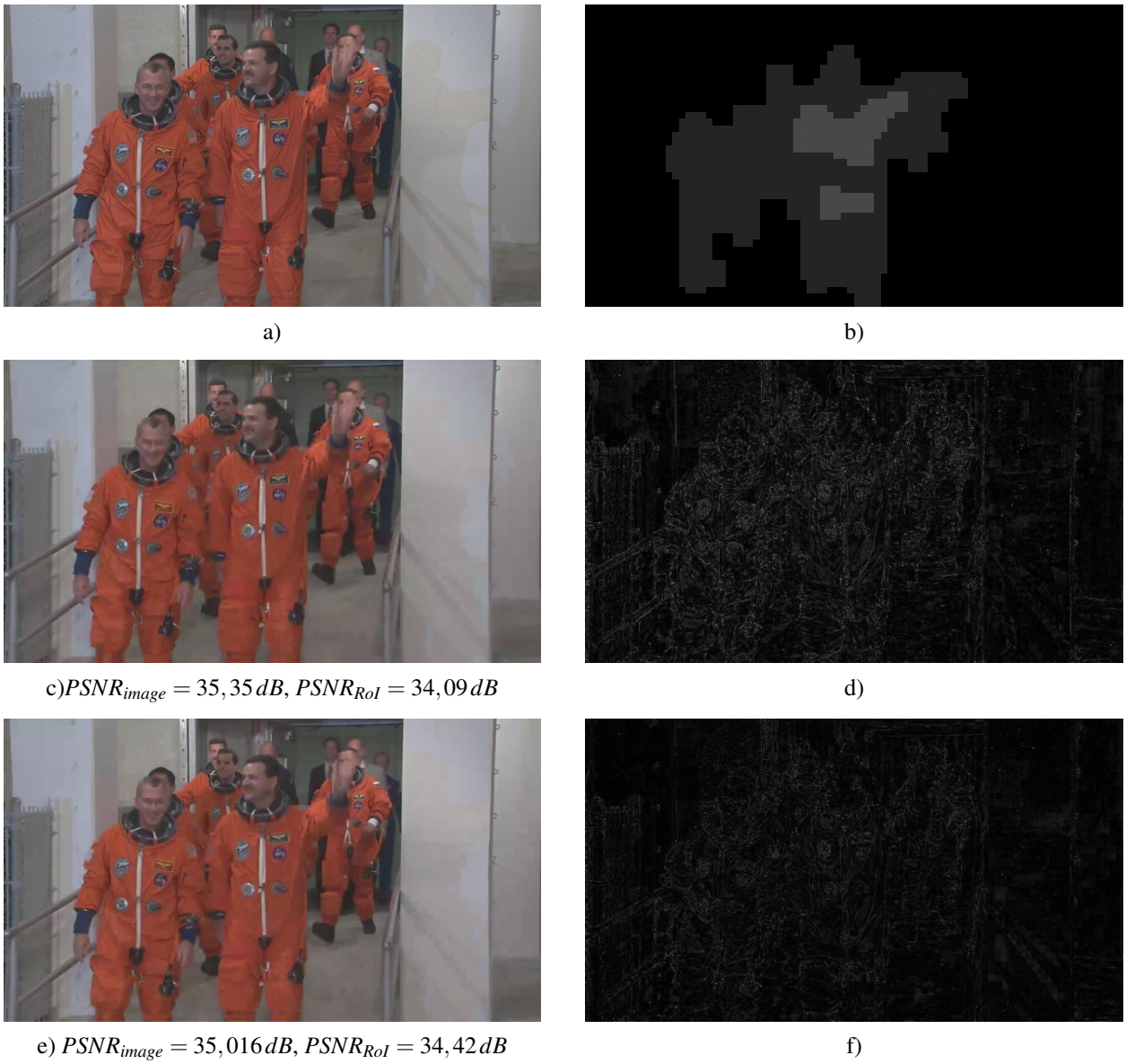


FIG. 5.10 – Résultats du codage adapté avec une région d'intérêt de taille réduite ($S_{Mod}^{SP-T}(i) = S_{max}$, où S_{max} est la valeur maximale de la saillance) : courbes débit-distorsion (PSNR) obtenues sur l'image complète ou sur la région d'intérêt pour les séquences *Umbrella*, *Parkjoy*, *Crew* et *Tractor*.



a) b)

c)$PSNR_{image} = 35,35\,dB$, $PSNR_{RoI} = 34,09\,dB$ d)

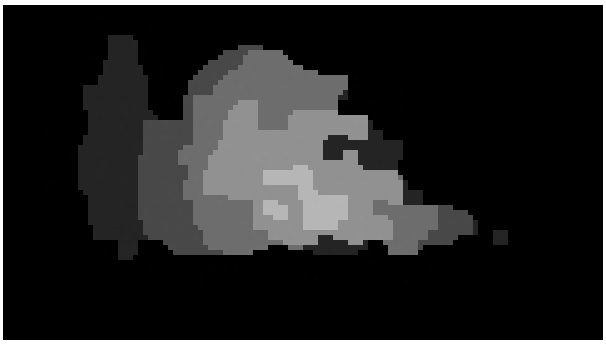
e) $PSNR_{image} = 35,016\,dB$, $PSNR_{RoI} = 34,42\,dB$ f)

FIG. 5.11 – Exemple de résultats du codage adapté pour la séquence *Night* : a) image originale, b) carte de saillance modifiée pour le codage, c) codage classique, d) erreur engendrée par un codage classique, e) codage adapté (QA), f) erreur engendrée par un codage adapté.

à porter un jugement de qualité sur des séquences présentées les unes après les autres. Une session de tests contenant N_S séquences indexées par $i \in [1; N_S]$ consiste en la visualisation V_i et l'évaluation E_i de la séquence i. La figure 5.13 représente la structure d'une telle session. Chaque séquence est notée indépendamment sur une échelle de qualité à cinq catégories comme celle présentée dans le tableau 5.7. Ce nombre de catégories permet une bonne discrimination sans trop de dispersion des scores. Avec cette échelle, l'observateur ne voit que les attributs sémantiques, et non les valeurs numériques associées.

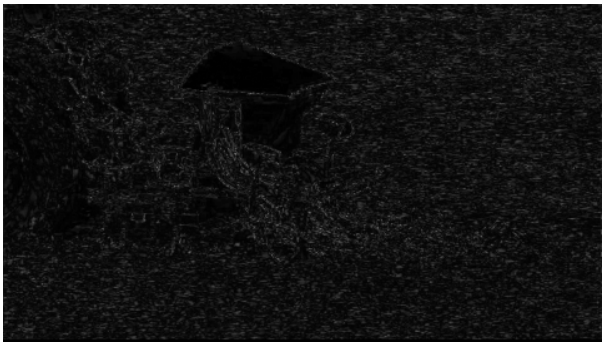
Remarquons que les séquences de référence, qu'il est recommandé d'ajouter dans la session, ne sont pas identifiées comme telles par l'observateur. Ainsi, l'ACR ne mesure pas la différence de qualité entre deux séquences, mais bien la qualité absolue de chaque séquence. Cette méthodologie est simple pour sa mise en place et particulièrement rapide, elle permet de juger un grand nombre de séquences par session. Par exemple, dans le cadre du Test Plan Multimedia [VQE07], VQEG évalue 166 séquences de huit secondes en une session de 35 minutes environ. La contrepartie de ce rendement est la moindre précision des scores,

a)



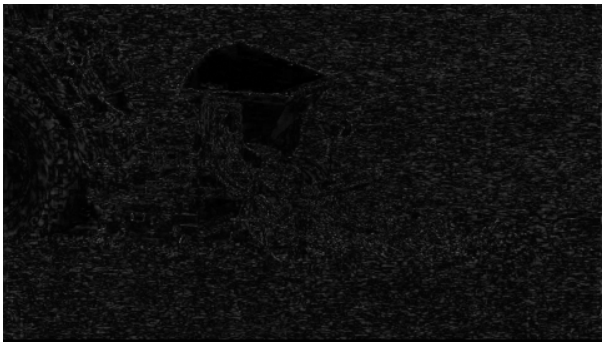
b)

c)$PSNR_{image} = 33,26\,dB$, $PSNR_{RoI} = 33,16 dB$



d)

e) $PSNR_{image} = 32,95\,dB$, $PSNR_{RoI} = 33,42\,dB$



f)

FIG. 5.12 – Exemple de résultats du codage adapté pour la séquence *Tractor* : a) image originale, b) carte de saillance modifiée pour le codage, c) codage classique, d) erreur engendrée par un codage classique, e) codage adapté (QA), f) erreur engendrée par un codage adapté.

un nombre important d'observateurs est donc nécessaire. Ainsi, VQEG préconise d'utiliser un panel d'au moins 24 observateurs.

5.3.2 Contenus évalués

5.3.2.1 Séquences vidéo

Huit séquences vidéo ont été utilisées durant cette campagne de tests. Toutes ces séquences sont au format haute-définition progressif 720p ou 1080p, cadencé à 50 trames par seconde. Les valeurs de chaque pixel sont codées dans l'espace colorimétrique YC_bC_r, avec un échantillonnage couleur 4:2:0. Ces séquences sont présentées en annexe A.

FIG. 5.13 – Structure d'une session utilisant la méthodologie ACR. V_i représente la visualisation d'une séquence et E_i le vote pour cette séquence.

note	valeur sémantique
5	excellent
4	bon
3	assez bon
2	médiocre
1	mauvais

TAB. 5.7 – Échelle d'évaluation de la qualité par catégories préconisée dans la recommandation l'ITU-R BT.500-11 [ITU04].

5.3.2.2 Génération des séquences dégradées

Pour chacune des huit séquences, 17 versions différentes ont été évaluées par les observateurs : la version non codée (et donc non dégradée) et seize versions compressées (et donc dégradées). Ces seize versions dégradées ont été obtenues en encodant chaque séquence pour quatre débits différents et avec les quatre codeurs présentés ci-dessous :

- le codeur x264 classique (x264),
- le codeur x264 modifié permettant la modification adaptative de la structure du GOP (MASGOP),
- le codeur x264 modifié permettant la quantification adaptée (QA).
- le codeur x264 modifié permettant la modification adaptative de la structure du GOP et la quantification adaptée (MASGOP + QA).

Les quatre débits utilisés pour chaque séquence ont été définis par un panel d'experts. La gamme des débits est différente d'un contenu à un autre. Ceci est dû aux différences de complexité spatio-temporelle de chaque contenu. Les valeurs de débit ont été choisies de manière à ce que la qualité des séquences générées corresponde approximativement à des niveaux de qualité prédéfinis (cf tableau 5.7). Les débits choisis, ainsi que l'ensemble des informations utiles concernant les huit séquences, sont récapitulés au tableau 5.8. Lors des tests, les observateurs ont donc évalué 17 séquences pour chacun des 8 contenus vidéo : 17 versions non étiquetées constituées des 16 versions codées et de la référence cachée (non traitée). Au total, ce sont donc 136 séquences qui ont été évaluées durant cette campagne de tests.

5.3.3 Conditions d'observation

5.3.3.1 Écran

Il s'agit d'un moniteur LCD TVLogic Multi-Format LVM-401W d'une diagonale de 40 pouces. Il possède une résolution native de 1920×1080 pixels avec un rapport de contraste de 1000 : 1 et une luminance au centre de l'écran de $450\,cd/m^2$.

Nom	Durée	Résolution	Débits des versions dégradées (Mbps)
Crew	$12\,s$	1280×720	$1,4 - 1,8 - 2,6 - 4,7$
Knightshields	$10\,s$	1280×720	$1 - 1,35 - 2 - 3,4$
New Mobile and Calendar	$10\,s$	1280×720	$0,7 - 0,9 - 1,6 - 2,9$
Night	$9,2\,s$	1280×720	$1,3 - 1,7 - 2,7 - 5,3$
Parkrun	$9,7\,s$	1280×720	$1,2 - 2,2 - 5 - 15$
Parkjoy	$8,8\,s$	1920×1080	$10 - 16 - 24 - 50$
Tractor	$13,8\,s$	1920×1080	$2,8 - 3,5 - 4,6 - 8,2$
Umbrella	$10\,s$	1920×1080	$4,7 - 7 - 10 - 13,5$

TAB. 5.8 – Tableau récapitulatif des caractéristiques des huit séquences utilisées.

5.3.3.2 Salle de tests

Les tests se sont déroulés dans une salle spécifique, dont les murs sont recouverts d'un tissu neutre. Les conditions de lumière et la distance d'observation ont été mesurées et ajustées suivant les recommandations de l'ITU [ITU98, ITU04]. Pour des contenus haute-définition la distance d'observation recommandée est égale à trois fois la hauteur de l'image, soit dans notre cas à $150\,cm$.

5.3.3.3 Adaptation des différents formats à l'écran

Les huit séquences utilisées sont aux formats 1920×1080 (*Parkjoy*, *Tractor* et *Umbrella*) et 1280×720 (*Crew*, *Knightshields*, *New Mobile and Calendar*, *Night* et *Parkrun*). Des problèmes se posaient, car l'écran LCD ne possède qu'une résolution de 1920×1080 pixels, pour pouvoir comparer les résultats de l'évaluation des séquences vidéo. Les conditions de tests doivent demeurer les mêmes, notamment la distance d'observation (ou plus précisément le rapport pixels par degré visuel, qui doit rester égal à 57 pixels/degré). La solution de redimensionnement de séquences introduit des distorsions visuelles, ce qui n'est pas souhaitable. Nous avons plutôt décidé de leur ajouter une bordure grise sur les quatre cotés, telle que la séquence finale possède une taille de 1920×1080 pixels. Cette opération est illustrée à la figure 5.14. La distance d'observation reste la même que pour l'évaluation des séquences vidéo au format 1920×1080 pixels ($150\,cm$). Cela correspond pour ces séquences à une distance d'environ $4,5$ fois la hauteur de la séquence (séquence originale sans bordure).

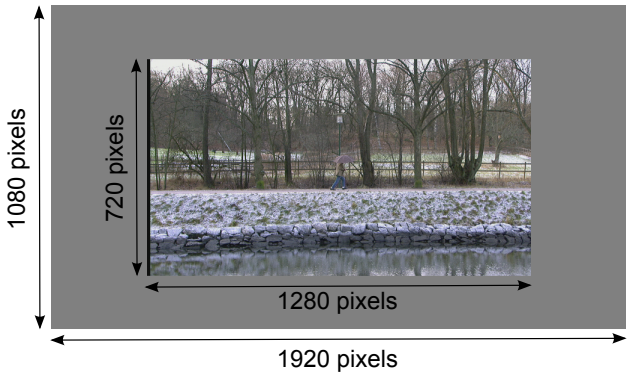


FIG. 5.14 – Séquence originale au format 1280×720 pixels insérée dans une séquence grise de 1920×1080 pixels, pour être affichée au format 1080p sur l'écran de tests.

Session	Codeurs évalués	Nb contenus	Nb obs (retenus/total)
1	x264 et MASGOP	72	27/27
2	x264, QA et MASGOP + QA	104	32/32

TAB. 5.9 – Tableau récapitulatif des deux sessions d'évaluation subjective de la qualité visuelle.

5.3.4 Observateurs

L'ensemble des évaluations a été divisé en deux sessions de tests. La première correspond à l'évaluation des séquences générées avec le codeur classique (x264) et le codeur modifié permettant la modification adaptative de la structure du GOP (MASGOP), par un panel d'observateurs dits « naïfs », c'est-à-dire, n'ayant aucune expérience de la qualité visuelle. La seconde session de tests a été réalisée également avec un panel d'observateurs « naïfs » qui ont évalué les séquences obtenues avec le codeur classique (x264), le codeur modifié permettant la quantification adaptée (QA) et le codeur modifié permettant les deux approches (MASGOP + QA). Le tableau 5.9 décrit les deux sessions de tests.

Chacun des observateurs a subi un examen de manière à vérifier son acuité visuelle par le test de Monoyer [Mon75] et l'absence de daltonisme par les tests d'Ishihara [Ish67].

La cohérence des résultats de chaque observateur est évaluée, de manière à détecter et à rejeter, le cas échéant, les observateurs non consistants. La comparaison des notes de qualité données par un observateur avec les notes moyennes de l'ensemble des observateurs est effectuée [ITU04]. Deux critères sont évalués, le coefficient de corrélation linéaire et le coefficient de corrélation de rang. Si l'un des deux coefficients est inférieur à un certain seuil, l'observateur est rejeté. Nous avons fixé le seuil à 0.85. VQEG préconise d'utiliser un panel d'au moins 24 observateurs pour la méthodologie ACR [VQE07].

5.3.5 Résultats

Les résultats de ces évaluations subjectives de la qualité sont présentées dans les figures 5.15 à 5.17. Sur chaque graphe, la note de qualité moyenne Q sur l'ensemble des observateurs est tracée en fonction du débit de la séquence codée. Les intervalles de confiance à 95% sont donnés par les segments verticaux associés à chaque point.

On note $\triangle Q$ la différence entre la note de qualité moyenne obtenue pour notre stratégie de codage et la note de qualité moyenne pour le codage classique (x264), pour chaque contenu i et pour chaque débit j :

$$\triangle Q_{M_k,i,j} = Q_{M_k,i,,j} - Q_{x264,i,j} \tag{5.3}$$

avec $k = 1,2,3$, et où les codeurs M_1, M_2 et M_3 correspondent respectivement aux méthodes MASGOP, QA et MASGOP + QA. Nous avons choisi de nous intéresser, pour chaque méthode de codage $M_{k,}$ et pour chaque contenu i, à la différence de qualité $\triangle Q$ pour les N_i débits j ayant obtenu une note de qualité supérieure ou égale à 3 pour nos différentes méthodes de codage. Cette note de qualité correspond à la catégorie « assez bon ». En effet, l'application concernée étant le codage de la TVHD, cela nécessite de proposer une qualité visuelle minimale suffisante. La moyenne des différences de qualité entre la méthode de codage M_k et le codage classique (x264) pour l'ensemble des traitements j vérifiant cette condition est notée $\triangle Q_{MOY,M_k}$, on obtient une valeur de $\triangle Q_{MOY,M_k}$ par contenu i :

$$\triangle Q_{MOY,M_k} = \frac{1}{N_i} \sum_{j|Q_{M_k,i,,j} \geq 3} (Q_{M_k,i,,j} - Q_{x264,i,j}), \quad k = 1,2,3 \tag{5.4}$$

Séquence	$\triangle Q_{MOY,M_1}$ ($M_1 \Longleftrightarrow$ MASGOP)
New Mobile and Calendar	$0,31$
Night	$-0,02$
Knightshields	$0,24$
Parkrun	$0,04$
Umbrella	$-0,09$
Parkjoy	$0,14$
Crew	$0,48$
Tractor	$0,33$
Moyenne	$0,18$

TAB. 5.10 – Différence de qualité $\triangle Q$ entre la méthodes de codage MASGOP et le codage classique (x264) pour chaque séquence, $\triangle Q_{MOY,M_1}$ est la moyenne sur les séquences ayant obtenues une note moyenne de qualité supérieure à 3 pour la méthode de codage MASGOP.

5.3.5.1 Résultats de la modification adaptative de la structure du GOP

On observe sur la figure 5.15 que des différences significatives existent, pour seulement trois contenus et pour un seul débit à chaque fois, entre les notes de qualité obtenues pour la méthode MASGOP et les notes de qualité obtenues pour le codage classique (x264) :

- pour la séquence *New Mobile and Calendar* à $2,9 Mbps$,
- pour la séquence *Crew* à $2,6 Mbps$,
- et pour la séquence *Tractor* à $2,8 Mbps$.

Les valeurs de $\triangle Q_{MOY,M_1}$ sont données pour chaque séquence dans le tableau 5.10. La valeur moyenne de cette mesure sur les huit contenus est également donnée.

Le gain de qualité moyen sur les séquences ayant obtenu un score de qualité supérieur à 3 pour la méthode MASGOP a une valeur moyenne de $0,18$. Néanmoins, on peut observer une très grande variabilité d'un contenu à l'autre. Les valeurs extrêmes étant obtenue pour la séquence *Umbrella* d'une part avec $\triangle Q_{MOY,M_1} = -0,09$ et pour la séquence *Crew* d'autre part avec $\triangle Q_{MOY,M_1} = 0,48$.

On constate que pour les deux séquences tournées en intérieur, *New Mobile and Calendar* et *Knightshields*, et pour lesquelles les valeurs de *IT* et *IS* étaient faibles, les notes de qualité pour la méthode MASGOP sont supérieures à celles obtenues pour le codage classique. Bien qu'ayant également des valeurs de *IT* et *IS* faibles, aucun gain n'est obtenu pour la séquence *Night*. Contrairement aux deux séquences précédentes, elle a été filmée en extérieur, et les variations de luminosité (enseignes clignotantes) engendrent des changements entre les images successives, pénalisant la méthode MASGOP qui augmente à tort le nombre d'images B. Pour les trois séquences ayant une valeur de *IS* élevée traduisant un contenu très texturé, *Parkrun*, *Parkjoy* et *Umbrella*, les notes de qualité pour la méthode MASGOP sont légèrement supérieures à celles obtenues pour le codage classique. Les gains les plus importants sont obtenus pour les séquences *Crew* et *Tractor* qui possèdent une valeur *IT* élevée.

Notre méthode de modification adaptative de la structure du GOP semble donc être particulièrement adaptée lorsque la séquence contient de forts mouvements.

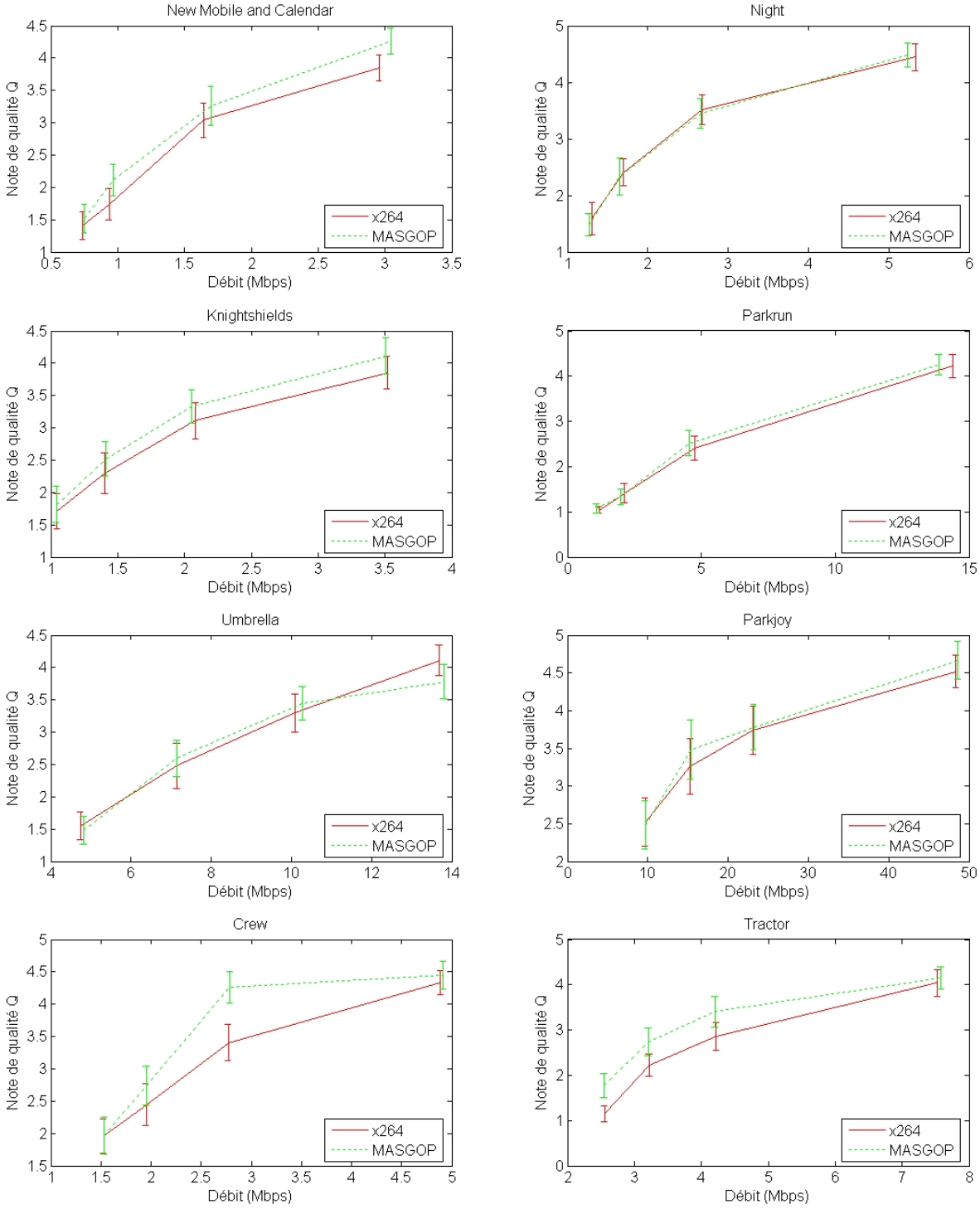


FIG. 5.15 – Résultats de l'évaluation subjective de la qualité visuelle des huit séquences pour un codage classique (x264) et notre méthode de modification adaptative de la structure du GOP (MASGOP).

5.3.5.2 Résultats du codage adaptatif basé sur la saillance visuelle

Aucune différence significative n'apparaît entre les notes de qualité obtenues pour la méthode QA et celles obtenues pour le codage classique (x264).

Les valeurs de $\triangle Q_{MOY,M_2}$ sont données pour chaque séquence dans le tableau 5.11. La valeur moyenne de cette mesure sur les huit contenus est également précisée.

Séquence	$\triangle Q_{MOY,M_2}$ ($M_2 \Longleftrightarrow$ QA)
New Mobile and Calendar	$-0,13$
Night	$0,09$
Knightshields	$-0,02$
Parkrun	$-0,06$
Umbrella	$0,06$
Parkjoy	$0,17$
Crew	$-0,06$
Tractor	$0,25$
Moyenne	$0,04$

TAB. 5.11 – Différences de qualité $\triangle Q$ entre les méthodes de codage QA et celui classique (x264) pour les différentes séquences, $\triangle Q_{MOY,M_2}$ est la moyenne sur les séquences ayant obtenues une note moyenne de qualité supérieure à 3 pour la méthode de codage QA.

On constate que pour les séquences ayant obtenu un score de qualité supérieur à 3 pour la méthode QA, le gain de qualité moyen sur les séquences n'est que de $0,04$. Néanmoins, on observe une variabilité d'un contenu à l'autre. Les valeurs extrêmes sont obtenues pour les séquences *New Mobile and Calendar* ($\triangle Q_{MOY,M_2} = -0,13$) et *Tractor* ($\triangle Q_{MOY,M_2} = 0,25$).

Pour les séquences *Umbrella*, *Parkjoy* et *Tractor*, la méthode de codage QA produit des gains de qualité moyens qui ont respectivement pour valeur $0,06$, $0,17$ et $0,25$. On avait déjà observé de tels gains de qualité pour ces trois séquences lorsque nous utilisions le PSNR (section 5.2.5.2) pour évaluer cette approche. L'analyse est la même : l'économie de bits réalisée par la quantification plus forte des zones de non intérêt est plus profitable que le surplus de bits nécessaire pour coder les zones d'intérêt. En effet, ces séquences ont des valeurs élevées pour IT ou IS, traduisant respectivement une activité temporelle ou spatiale importante. Les séquences vidéo avec une activité spatiale importante présentent deux intérêts : d'abord, le codage des zones possédant un contenu riche (hautes fréquences) est fortement consommateur de bits. Il est donc possible de consommer moins de bits pour les zones de non intérêt pour les affecter au codage de la zone saillante. Le second point intéressant est que ce type de contenu a une capacité intrinsèque à masquer le bruit de quantification et donc peut supporter une quantification plus forte sans introduire de défauts perceptibles. Par contre, pour les séquences ayant des valeurs de IT et IS faibles, telles que *New Mobile and Calendar* et *Knightshields*, la méthode de codage QA est moins performante, puisque les « gains » de qualité moyens ont respectivement pour valeurs $-0,13$ et $-0,02$. Le fait qu'il n'y ait peu de changements entre les images successives de la séquence pénalise notre méthode de codage adaptatif basé sur la saillance visuelle. Le surcoût engendré par la quantification plus fine des zones d'intérêt et par la transmission des variations du pas de quantification entre les macroblocs, est alors trop important par rapport à l'économie de bits réalisée par la quantification plus forte résultante de l'optimisation débit-distorsion réalisée par le codeur.

Ces résultats semblent montrer que notre approche de codage adaptatif basé sur la saillance visuelle n'est pas bien adaptée dans un contexte de diffusion de TVHD. En effet, la régulation de débit réalisée par

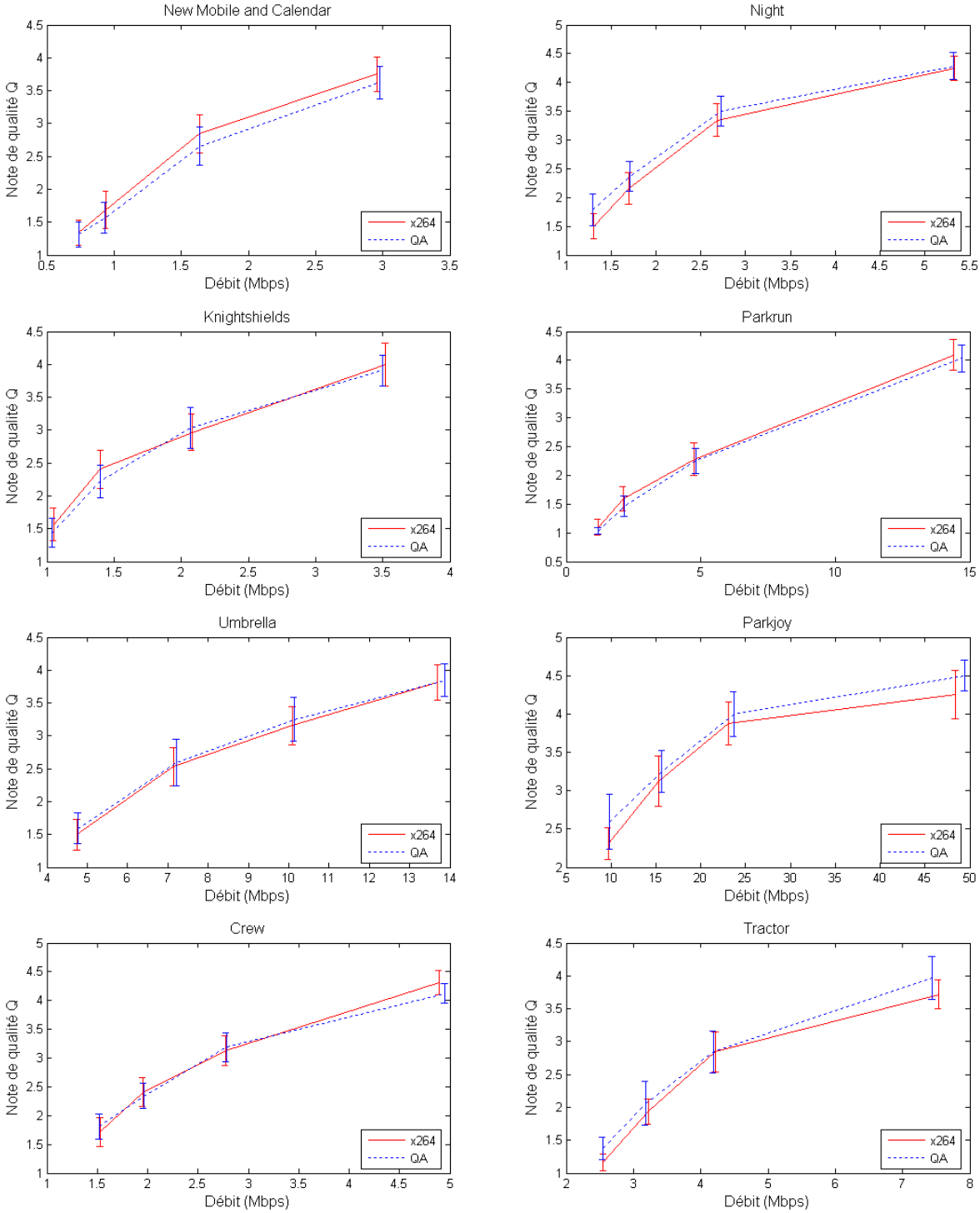


FIG. 5.16 – Évaluation subjective de la qualité visuelle des huit séquences après un codage classique (x264) ou notre méthode de codage adaptatif basé sur la saillance visuelle (QA).

Séquence	$\triangle Q_{MOY,M_3}$ ($M_3 \Longleftrightarrow$ MASGOP + QA)
New Mobile and Calendar	$0,27$
Night	$0,02$
Knightshields	$0,2$
Parkrun	$-0,13$
Umbrella	$0,25$
Parkjoy	$0,07$
Crew	$-0,08$
Tractor	$0,36$
Moyenne	$0,12$

TAB. 5.12 – Différences de qualité $\triangle Q$ entre la méthode de codage (MASGOP + QA) et celui classique (x264) pour les différentes séquences, $\triangle Q_{MOY,M_3}$ est la moyenne sur les séquences ayant obtenues une note moyenne de qualité supérieure à 3 pour la méthode de codage (MASGOP + QA).

le codeur afin de contrôler les surcoûts engendrés par la variation du pas de quantification, ainsi que par la quantification plus fine des régions saillantes, pénalise la qualité visuelle de certaines séquences (*New Mobile and Calendar*, *Knightshields, Parkrun* et *Crew*). Ces résultats ne permettent pas non plus d'identifier pour quels types de contenu la méthode de codage QA est adaptée. Il serait intéressant de pouvoir mesurer précisément le surcoût engendré par le codage de la variation du pas de quantification entre les macroblocs, afin de pouvoir mieux évaluer la pertinence de notre approche.

5.3.5.3 Résultats des deux approches utilisées conjointement

On observe sur la figure 5.17 deux différences significatives entre les notes de qualité obtenues pour la méthode (MASGOP + QA) et celles obtenues pour le codage classique (x264). Elles apparaissent pour la séquence *Tractor* à $2,8 Mbps$ et $3,5 Mbps$.

Les valeurs de $\triangle Q_{MOY,M_3}$ sont données pour chaque séquence dans le tableau 5.12. La valeur moyenne de cette mesure sur les huit contenus est également donnée. Le gain de qualité moyen sur les séquences ayant obtenu un score de qualité supérieur à 3 pour la méthode (MASGOP + QA) a une valeur moyenne de $0,12$. Néanmoins, on observe une variabilité d'un contenu à l'autre. Les valeurs extrêmes sont obtenues pour les séquences *Parkrun* ($\triangle Q_{MOY,M_3} = -0,13$) et *Tractor* ($\triangle Q_{MOY,M_3} = 0,36$).

L'utilisation conjointe des deux approches de codage adaptées (MASGOP et QA) ne semble pas profitable. En effet, sur l'ensemble des séquences le gain de qualité moyen est de $0,12$, alors que pour la méthode MASGOP d'une part et la méthode QA d'autre part, les gains de qualité moyens étaient respectivement de $0,18$ et $0,04$. Ce phénomène atteint particulièrement la séquence *Crew*, puisque les méthodes MASGOP et QA obtenaient respectivement des gains de qualité moyens de $0,48$ et de $-0,06$, alors que le gain de qualité moyen obtenue par l'utilisation conjointe des deux méthodes est de $-0,08$. Notre méthode de codage adaptatif basé sur la saillance visuelle pénalise le gain de qualité obtenu par la méthode de modification adaptative de la structure du GOP, excepté ici pour les séquences *Night*, *Umbrella* et *Tractor*. En effet, les gains de qualité moyens obtenus pour la séquence *Umbrella* étaient respectivement de $-0,09$ et de $0,06$ pour les approches MASGOP et QA, alors qu'il atteint $0,25$ lorsque les deux approches sont utilisées conjointement. On retrouve les carences que l'on a identifiées précédemment (section **??**). Donc, les résultats obtenus lors de l'utilisation conjointe des méthodes MASGOP et QA semblent montrer que notre approche de codage adaptatif basé sur la saillance visuelle n'est pas adaptée dans un contexte de diffusion de TVHD. Les gains

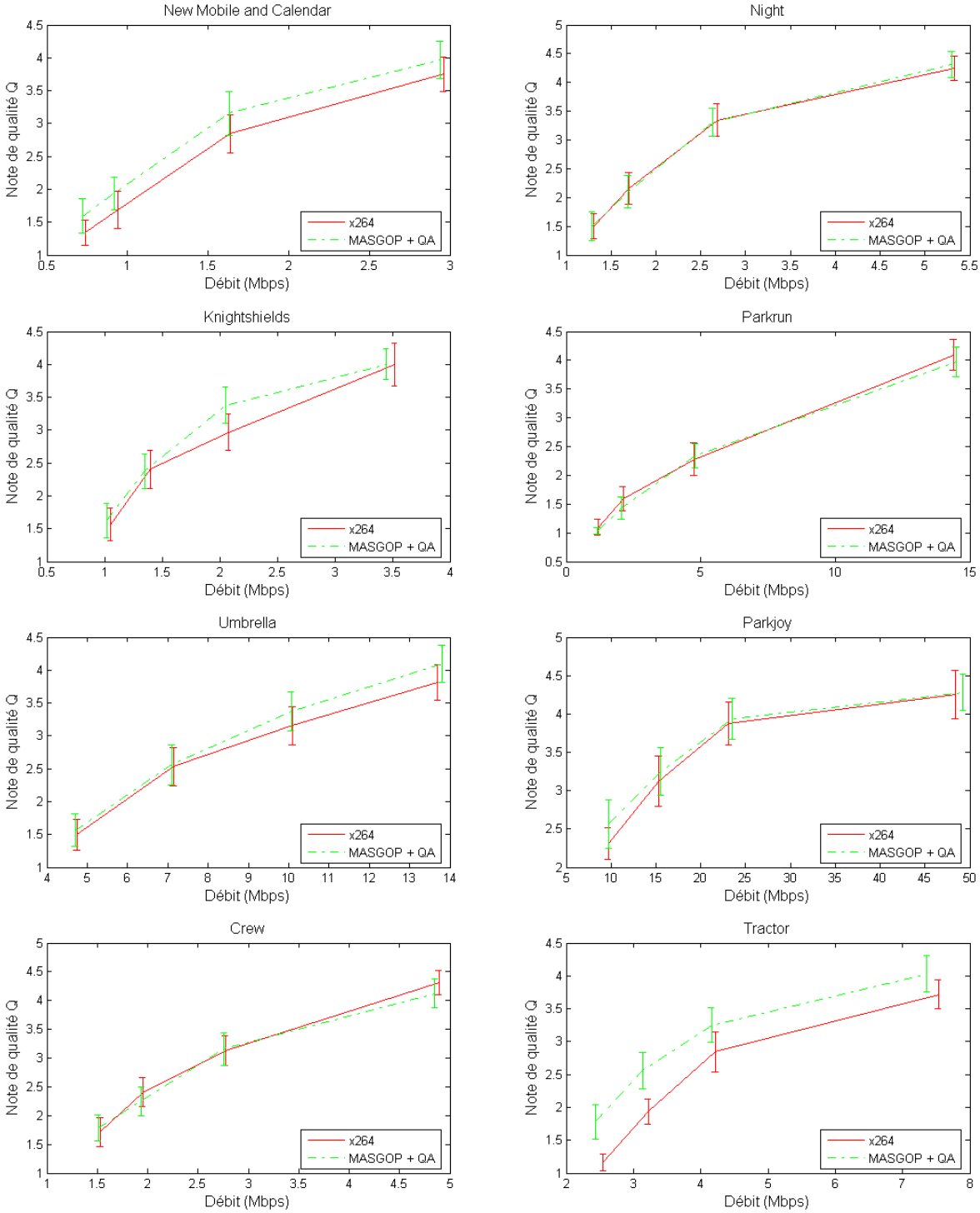


FIG. 5.17 – Évaluation subjective de la qualité visuelle des huit séquences après un codage classique (x264) ou le codeur modifié permettant les deux approches (MASGOP + QA).

de qualité moyens sont obtenus en grande partie par la méthode de modification adaptative de la structure du GOP qui est particulièrement efficace lorsque la séquence contient de forts mouvements (section 5.3.5.1).

Conclusion

Ce chapitre a présenté deux applications de codage en fonction des informations obtenues après notre méthode de pré-analyse de la vidéo. La première concerne la modification adaptative de la structure du GOP en fonction du contenu spatio-temporel de la séquence vidéo. Celle-ci étant elle même constituée de deux approches différentes : la variation dynamique du nombre d'images B (VDNIB) et l'adaptation dynamique de la taille du GOP. Nous avons d'abord étudié la configuration optimale de la structure du GOP, c'est-à-dire, le nombre fixe d'images B (0, 1 , 2 ou 3) qui obtient les meilleures performances en termes de débit-distorsion, pour chaque séquence vidéo testée. Cette première étude a permis d'identifier que le nombre d'images B optimal était dépendant du contenu des séquences vidéos. Afin de caractériser l'activité spatio-temporelle des séquences ou plus précisément des segments temporels, nous avons proposé deux mesures permettant d'évaluer respectivement l'activité temporelle IT et l'activité spatiale IS de ceux-ci. Une classification des segments temporels en fonction de leurs valeurs de IT et IS est ensuite réalisée déterminant ainsi le nombre d'images B pour chaque segment temporel, permettant d'obtenir en moyenne un gain de $0,11\,dB$ et jusqu'à $0,55\,dB$ pour la séquence *Tractor*. La deuxième approche modifie dynamiquement la taille du GOP en fonction de l'activité au sein du plan vidéo. L'analyse de l'évolution de l'activité temporelle IT, permet de détecter les changements au sein de la séquence qui nécessitent l'insertion d'une nouvelle image I. Cette deuxième méthode couplée avec notre méthode VDNIB permet d'obtenir en moyenne un gain de $0,16\,dB$ et de réduire le débit jusqu'à 11% pour la séquence *Tractor*.

La deuxième, quant à elle, concerne une application de compression de la vidéo avec une qualité visuelle différenciée guidée par les cartes de saillance (QA). Après avoir défini la compression sélective, la carte de saillance, moyennant quelques modifications, a été couplée au schéma de codage. On parle alors de compression sélective directe. Pour ce type de compression, c'est le cœur et la stratégie de codage qui sont modifiés. L'objectif est d'améliorer la qualité perçue comparativement à une approche classique de codage. L'approche envisagée permet d'améliorer la qualité en terme de PSNR des régions d'intérêt au détriment de la qualité globale.

La dernière partie de ce chapitre a présenté les tests d'évaluation subjective de la qualité visuelle des deux applications proposées. Ils n'ont pas permis de mettre en évidence une différence significative entre les notes de qualité moyennes obtenues par nos approches de codage adaptées et une méthode de codage classique (x264). Une différence moyenne de $0,18$ (sur une échelle de 1 à 5) en faveur de la méthode de modification adaptative de la structure du GOP a été mesurée sur l'ensemble des contenus vidéos. Les gains les plus importants sont obtenus pour les séquences qui possèdent une valeur IT élevée. Notre méthode de modification adaptative de la structure du GOP semble donc être particulièrement adaptée lorsque la séquence contient de forts mouvements. Les résultats obtenus pour notre méthode de codage adaptatif basé sur la saillance visuelle (QA) ne permettent pas d'identifier les types de contenu propices à une telle approche. Au contraire, il semblerait que celle-ci ne soit pas adaptée au codage et à la diffusion de la TVHD. L'optimisation débit-distorsion réalisée par le codeur afin de contrôler les surcoûts engendrés par la variation du pas de quantification, ainsi que par la quantification plus fine des régions saillantes, pénalise la qualité visuelle de certaines séquences. Il serait intéressant de mesurer le surcoût engendré par le codage de la variation du pas de quantification entre les macroblocs, afin de pouvoir évaluer la pertinence de notre approche de codage

adaptatif basé sur la saillance visuelle et de décider s'il est nécessaire d'approfondir les investigations dans ce domaine.

Conclusion générale et perspectives

Les travaux présentés dans ce mémoire de thèse portaient sur la pré-analyse de la vidéo en vue de son codage avancé. Nous avons développé une méthode de segmentation spatio-temporelle et de suivi des objets dans la séquence vidéo, ainsi qu'un modèle de l'attention visuelle pré-attentive. Deux applications utilisant cette méthode de pré-analyse proposée sont décrites. Dans cette conclusion générale, nous présentons d'abord une synthèse de ces travaux. Ce bilan est suivi par plusieurs perspectives tant sur la pré-analyse, les applications proposées que sur l'exploitation de la pré-analyse de la vidéo.

Synthèse des travaux réalisés

Pré-analyse de la vidéo

Afin de pallier les lacunes des codeurs vidéos classiques, nous avons proposé une méthode d'analyse spatio-temporelle de plans vidéo. Afin de détecter les objets en mouvement qui sont importants visuellement, nous avons réalisé une segmentation spatio-temporelle de chaque plan. Tout d'abord, nous réalisons une estimation multi-résolution du mouvement basée sur des tubes spatio-temporels pour une fenêtre temporelle (ou segment temporel) de neuf images, correspondant à une durée de $180 ms$. Le temps de fixation du SVH étant sensiblement égal à $200 ms$, cette estimation du mouvement permet de le suivre sur une durée significative perceptuellement. Avant de réaliser la segmentation, nous estimons le mouvement global du segment temporel à partir de l'histogramme des vecteurs obtenus (un vecteur de mouvement par tube spatio-temporel). Une fois le mouvement global calculé et compensé, nous réalisons la segmentation basée mouvement du plan vidéo. Nous proposons ensuite une approche markovienne pour affiner cette segmentation spatio-temporelle en intégrant de nouveaux critères tels que la connexité spatiale, la couleur, la texture et la connexité temporelle entre sites. Cette modélisation markovienne permet d'améliorer les résultats initiaux de segmentation, mais aussi d'assurer le suivi des régions entre segments temporels. La dernière étape de notre pré-analyse construit une modélisation de notre attention visuelle pré-attentive sous la forme de cartes de saillance. Leur détermination est basée sur les contrastes de couleur et le mouvement relatif des objets. Elles mettent en évidence les zones susceptibles d'exciter notre attention visuelle. La construction de la carte de saillance spatio-temporelle est obtenue par fusion de la carte de saillance spatiale et de la carte de saillance temporelle.

Applications de la pré-analyse

Notre méthode de pré-analyse de plans vidéo produit des informations de haut niveau pouvant être exploités dans de nombreuses nombreuses applications de codage, nous avons choisi deux de ces applications. La première concerne la modification adaptative de la structure du GOP en fonction d'une caractérisation

du contenu spatio-temporel du plan vidéo. Nous utilisons à ce niveau deux approches complémentaires : la variation dynamique du nombre d'images B (VDNIB) et l'adaptation dynamique de la taille du GOP (positionnement non régulier des images I). Après avoir étudié des configurations classiques de structures du GOP (performances en termes de débit-distorsion), où le nombre d'images B choisi est fixe (0, 1 , 2 ou 3) , et pour différentes séquences vidéo, nous avons montré que le nombre optimal d'images B était dépendant du contenu vidéo. Afin de caractériser l'activité spatio-temporelle de segments temporels, nous avons alors proposé deux mesures permettant d'évaluer respectivement leur activité temporelle IT et leur activité spatiale IS. Une classification des segments temporels en fonction de IT et IS utilisés conjointement est ensuite réalisée déterminant le nombre d'images B entre deux images P. En moyenne un gain de $0,11\,dB$ a été obtenu pour le PSNR. La deuxième approche modifie par ailleurs dynamiquement la taille du GOP en fonction de l'activité du plan vidéo. Elle ajoute par analyse de l'évolution de l'activité temporelle IT, la détection des changements qui nécessitent l'insertion d'une nouvelle image I. Cette deuxième méthode couplée avec notre méthode VDNIB permet d'obtenir en moyenne un gain de $0,16\,dB$.

La deuxième application que nous avons testée, concerne la compression vidéo avec une qualité visuelle différenciée guidée par les cartes de saillance. La carte de saillance obtenue par notre modèle d'attention visuelle est d'abord modifiée afin de s'adapter aux contraintes du codeur. Elle est ensuite couplée au schéma de codage, on parle alors de compression sélective directe, où c'est le cœur et la stratégie de codage qui sont adaptés. L'objectif est d'améliorer la qualité perçue comparativement à une approche classique de codage. Notre approche permet d'améliorer la qualité en terme de PSNR des régions d'intérêt au détriment de la qualité globale. Nous avons ensuite réalisé des tests d'évaluation subjective de la qualité visuelle des deux applications proposées. Ils n'ont pas permis de mettre en évidence une différence significative entre les notes de qualité moyennes obtenues par nos approches de codage adaptées et une méthode de codage classique (x264). Une différence moyenne de $0,18$ (sur une échelle de 1 à 5) en faveur de la méthode de modification adaptative de la structure du GOP a été mesurée sur l'ensemble des contenus vidéos. Les gains les plus importants sont obtenus pour les séquences qui possèdent une valeur IT élevée. Notre méthode de modification adaptative de la structure du GOP semble donc être particulièrement adaptée lorsque la séquence contient de forts mouvements. Les résultats obtenus pour notre méthode de codage adaptatif basé sur la saillance visuelle (QA) ne permettent pas d'identifier les types de contenu propices à une telle approche. Au contraire, il semblerait que celle-ci ne soit pas adaptée au codage et à la diffusion de la TVHD. L'optimisation débit-distorsion réalisée par le codeur afin de contrôler les surcoûts engendrés par la variation du pas de quantification, ainsi que par la quantification plus fine des régions saillantes, pénalise la qualité visuelle de certaines séquences. Il serait intéressant de mesurer le surcoût engendré par le codage de la variation du pas de quantification entre les macroblocs, afin de pouvoir évaluer la pertinence de notre approche de codage adaptatif basé sur la saillance visuelle et de décider s'il est nécessaire d'approfondir les investigations dans ce domaine.

Perspectives

Les perspectives que nous décrivons ici dans le prolongement de ces travaux de thèse sont au nombre de trois, avec d'abord la poursuite de travaux dans le domaine de la pré-analyse et plus particulièrement de la modélisation de l'attention visuelle. La seconde concerne l'amélioration des deux applications de codage vidéo proposées. La dernière perspective est relative aux multiples applications pouvant bénéficier de ces travaux.

Pré-analyse de la vidéo

Notre méthode de pré-analyse de la vidéo nous permet d'obtenir des informations de haut niveau sur la séquence vidéo. Mais elle n'est pas encore suffisamment robuste, il faudrait aussi :

- assurer la segmentation spatio-temporelle de petits objets en mouvement ;
- envisager une autre méthode de segmentation spatiale afin d'obtenir des contours francs des objets.

La modélisation de l'attention visuelle est un domaine en plein essor. Nous y avons contribué mais des voies restent à explorer pour l'améliorer, avec :

- une meilleure évaluation des performances du modèle, notamment par la construction d'une référence à partir d'expériences oculométriques sur des séquences vidéo. Cela nécessite la définition d'un protocole d'expérimentation ainsi qu'une façon de construire la vérité terrain ;
- la mise en place d'expérimentations psychophysiques pour optimiser les paramètres du modèle. Ce type d'expériences permettrait également d'affiner la technique de fusion que nous avons mis en place ;
- Le couplage de notre modèle avec des informations visuelles de haut niveau (visage, teinte chair...) pour en améliorer les performances.

Modification adaptative de la structure du GOP

Notre méthode de variation dynamique du nombre d'images B obtient en moyenne de meilleures performances relativement à une approche de codage vidéo classique, des améliorations possibles sont nombreuses en :

- tenant compte des spécificités locales du mouvement. Les paramètres du mouvement global estimés ne sont pas considérés en fonction de la variation dynamique du nombre d'images B. On pourrait envisager de découper les images en tranches (*slices*), puis d'adapter la stratégie de codage en fonction des caractéristiques locales du mouvement. Par exemple, dans le cas d'un zoom avant de la caméra, l'image serait découpée en tranches concentriques centrées sur l'origine du zoom, le nombre d'images B varierait pour chaque tranche (augmentation du nombre de B pour les tranches proches de l'origine du zoom où le mouvement est moins important) ;
- en proposant une nouvelle mesure de caractérisation de l'activité spatiale du segment temporel afin d'en améliorer la classification des segments temporels, et influençant la modification adaptative de la structure du GOP.

Bien que les performances de codage de notre modification adaptative de la structure du GOP soient supérieures à celles d'une méthode classique de codage, elles peuvent encore être améliorées en :

- utilisant une fenêtre d'analyse temporelle plus large (par exemple, trois segments temporels) centrée sur le segment temporel courant afin de tenir compte des évolutions passées et futures pour la prise de décision ;
- affinant le positionnement de la nouvelle image I. Il faudrait par exemple, exploiter les EQM locales issues de l'estimation de mouvement des tubes spatio-temporel, afin de placer précisément la nouvelle image I au sein du segment ;
- augmentant la taille des GOP lorsque le mouvement de plusieurs segments temporels successifs est faible.

Codage adaptatif basé sur la saillance visuelle

L'objectif est d'améliorer la qualité perçue comparativement à une approche classique de codage. L'approche envisagée permet d'améliorer la qualité en terme de PSNR des régions d'intérêt au détriment de la qualité globale, cependant il serait intéressant de :

- contrôler le surcoût engendré par la quantification plus fine des régions d'intérêt en codant plus grossièrement les régions de non intérêt ;
- contrôler le gain de débit engendré par une quantification plus forte ;
- distribuer les ressources de codage en fonction de la saillance, mais également en fonction des capacités de masquage de chaque macrobloc.

Applications de la pré-analyse de la vidéo

Notre méthode de pré-analyse permet d'obtenir des informations contextuelles sur la vidéo à coder, qui peuvent être exploitées et transmises au codeur afin de le guider dans ses choix de codage. Deux applications ont été présentées dans cette thèse. D'autres restent possibles En effet, un codeur vidéo classique prend des décisions à court terme et seulement selon des considérations de type signal (SNR). Les différentes partitions possibles de la prédiction inter image, peuvent à bas débit faire apparaître des artéfacts de codage. Si aucune cohérence de codage n'est assurée pour coder le macrobloc au cours du temps, des effets de papillotement peuvent apparaître. Ceci étant dû à l'utilisation de différentes partitions ou ou à une quantification inadaptée faisant apparaître les frontières des macroblocs ou dégradant les textures du macrobloc par intermittence. Une solution possible face à de telles dégradations est de contraindre le codeur à ne plus prendre des décisions à court terme, mais de le guider dans ses choix afin d'assurer la cohérence du codage de l'objet tout le long de sa durée de vie :

- la cohérence des codages intra et inter des macroblocs de la même région ;
- sélectionner les images référence optimales pour la prédiction temporelle ;

Nous pouvons également envisagé une application de codage canal. Les cartes de saillance permettent de hiérarchiser la séquence vidéo encodée, et afin de protéger plus efficacement les régions saillantes contre les erreurs de transmission, et donc afin d'améliorer la qualité d'usage. Dans cette optique, F. Boulos [Bou10] a proposé de protéger les régions d'intérêt obtenues via des tests oculométriques, contre la propagation spatio-temporelle des dégradations. Cette protection s'appuie sur le codage en mode intra (H.264) des macroblocs de la région d'intérêt dans les images B et P. Déjà une première évaluation de performances sur un nombre limité de séquences montre que cette méthode ne cause pas une diminution de la qualité visuelle à débit constant. Des évaluations objectives et subjectives de qualité indiquent aussi une amélioration de la qualité dans la région d'intérêt par rapport à un codage classique.

Annexes

Annexe A

Présentation des séquences vidéo utilisées lors des tests

Les séquences utilisées lors des tests réalisés pour les besoins de cette thèse sont disponibles via le serveur ftp ftp://ftp.ldv.e-technik.tu-muenchen.de/pub/test_sequences/. Ces séquences ont été filmées à une fréquence de 50 images par seconde avec l'équipement du SVT en octobre 2004. La plus grande attention a été donnée à la conversion des films vers un format numérique. Les détails concernant les conditions de prise de vue et les post-traitements sont présentés dans la documentation fournie par le SVT [ct06].

A.1 Les séquences 720p

Les séquences 720p utilisées ici sont des vidéos numériques acquises sous forme progressive de 720 lignes par 1280 pixels par ligne, cadencées à 50 images par seconde, la structure d'échantillonnage couleur des composantes YUV est 4:2:0.

A.1.1 New Mobile and Calendar

La séquence comporte 500 images filmées en plan rapproché. La caméra, qui subit un mouvement translationnel puis de zoom arrière, filme un calendrier avec du texte et une photo détaillée du Vasa[1]. Á partir de la 355ème image apparaît un train en mouvement translationnel avec des jouets très colorés. Le fond est composé de deux types de papiers peints, le premier est jaune, uniforme avec quelques figures dessinées et le second est très texturé. La figure A.1 a) présente une image extraite de la séquence *New Mobile and Calendar*.

A.1.2 Parkrun

La séquence comporte 500 images filmées en plan éloigné. La scène représente un homme, avec un parapluie dans sa main, qui court dans un parc puis s'arrête et reste immobile vers la 340ème image. L'arrière plan est composé d'arbres, de neige et d'une source d'eau. Le contenu est très détaillé. La figure A.1 b) présente une image extraite de la séquence *Parkrun*.

[1] Le Vasa est un vaisseau de guerre scandinave du 17ème siècle.

A.1.3 Knightshields

La séquence comporte 500 images filmées en plan rapproché. Un homme avec une barbe et une veste très texturée marche devant un mur composé de boucliers de chevaliers détaillés. Á la fin de la séquence, le capteur effectue un zoom avant de la scène. La figure A.1 c) présente une image extraite de la séquence *Knightshields*.

A.1.4 Crew

La séquence comporte 600 images filmées en plan rapproché. Des astronautes marchent et saluent des spectateurs. Plusieurs flashs d'appareils photos modifient la luminosité de la séquence. La figure A.1 d) présente une image extraite de la séquence *Crew*.

A.1.5 Night

La séquence comporte 460 images filmées la nuit en plan rapproché de l'intersection de deux routes. Au début de la séquence, un taxi jaune traverse la scène au premier plan. De nombreuses personnes se croisent sur un passage piétons devant les files de voitures à l'arrêt. A.1 e) présente une image extraite de la séquence *Night*.

A.2 Les séquences 1080p

Les séquences 1080p utilisées ici sont des vidéos numériques acquises sous forme progressive de 1080 lignes par 1920 pixels par ligne, cadencées à 50 images par seconde, la structure d'échantillonnage couleur des composantes YUV est également 4:2:0.

A.2.1 Blue Sky

La séquence comporte 250 images. La scène représente les cimes de deux arbres très détaillés, en fort contraste avec le ciel bleu uniforme. La caméra effectue une rotation. La figure A.2 a) présente une image extraite de la séquence *Blue sky*.

A.2.2 Station

La séquence comporte 313 images filmées en soirée, depuis un pont de la gare routière de Munich. La caméra effectue un long zoom arrière. La scène comporte des structures régulières avec beaucoup de détails (rails). La figure A.2 b) présente une image extraite de la séquence *Station*.

A.2.3 Tractor

La séquence comporte 690 images qui présentent un tracteur dans un champ. La séquence entière contient des zones sur lesquelles un très fort zoom avant est appliqué de manière à en obtenir une vue totale. La caméra suit le tracteur, avec un mouvement chaotique, sur la structure du champ de récolte. La figure A.2 c) présente une image extraite de la séquence *Tractor*.

a) Image 478 de la séquence *New Mobile and Calendar*

b) Image 160 de la séquence *Parkrun*

c) Image 1 de la séquence *Knightshields*

d) Image 200 de la séquence *Crew*

e) Image 50 de la séquence *Night*

FIG. A.1 – Séquences vidéo au format 1280×720 utilisées lors des tests.

A.2.4 Parkjoy

La séquence comporte 439 images filmées en plan éloigné. La scène représente un groupe de personnes qui courent sur la rive d'une rivière. La caméra suit le groupe de personnes depuis la rive opposée, la scène étant successivement occultée par deux arbres présents sur cette rive. Le contenu est très détaillé. La figure A.2 d) présente une image extraite de la séquence *Parkjoy*.

A.2.5 Umbrella

La séquence comporte 500 images qui présentent un défilé dans une rue. Un groupe d'hommes avec des parapluies défilent au milieu des spectateurs. Au premier plan, on aperçoit le toit du char motorisé qui laisse échapper des gaz d'échappements. La figure A.2 e) présente une image extraite de la séquence *Umbrella*.

a) Image 1 de la séquence *Blue sky*

b) Image 100 de la séquence *Station*

c) Image 600 de la séquence *Tractor*

d) Image 1 de la séquence *Parkjoy*

d) Image 1 de la séquence *Umbrella*

FIG. A.2 – Séquences vidéo au format 1920×1080 utilisées lors des tests.

Annexe B

Informations complémentaires sur le codeur H.264

B.1 Profils et niveaux

Le standard H.264 propose trois profils et onze niveaux afin de définir les points de conformité. Les premiers indiquent un ensemble d'outils de codage pouvant être utilisés pour générer un flux compatible. Les seconds imposent des contraintes à certains paramètres clés du flux (par exemple la taille maximale d'une image). Le profil de base (*baseline profile*) supporte le codage intra et inter des tranches I et P et le codage entropique CAVLC (*Context-Adaptive Variable-Lenght Codes*). Le profil principal (*main profile*) ajoute le support pour la vidéo entrelacée, le codage inter des tranches B, la prédiction pondérée et le codage entropique CABAC (*Context-Based Arithmetic Codin*g). Enfin, le profil étendu (*extended profile*) ne supporte ni la vidéo entrelacée ni le codage CABAC mais ajoute des modes pour permettre une commutation efficace entre les flux binaires codés (avec tranches SP et SI) et améliore la résistance du codage face aux erreurs (*data partitionning*). La figure B.1 résume les fonctionnalités incluses dans chaque profil.

Ces différents profils permettent différentes applications. Ainsi le profil de base est utilisé pour la vidéotéléphonie, la vidéoconférence et la communication sans fil. Le profil principal peut être utilisé pour la télévision et le stockage tandis que le profil étendu est destiné à la diffusion de vidéos en continu, au fur et à mesure du téléchargement. Néanmoins, chaque profil est suffisamment flexible pour supporter une large gamme d'applications.

Les différents profils permettent une large plage d'utilisation de la norme H.264.

B.2 Gestion des images de référence

Les images précédemment codées sont décodées et stockées dans une mémoire tampon appelée DPB (*Decoded Picture Buffer*), dont la taille est définie par un paramètre et l'indexage commence à 0. Le codeur et le décodeur conservent une ou deux listes d'images précédemment codées/décodées, la liste 0 et la liste 1 d'images de référence. Pour les tranches P, la liste 0 contient des images avant et après l'image courante selon l'ordre d'affichage. Ces images peuvent être des images de référence à court ou long terme (*short term* et *long term*), disponibles pour la prédiction. Les images à court terme sont directement identifiées par leur numéro d'image. Les images à long terme sont typiquement des images plus anciennes mais pouvant être utiles pour la prédiction. Elles sont identifiées par une variable *LongTermPictureNum* et restent dans le DPB

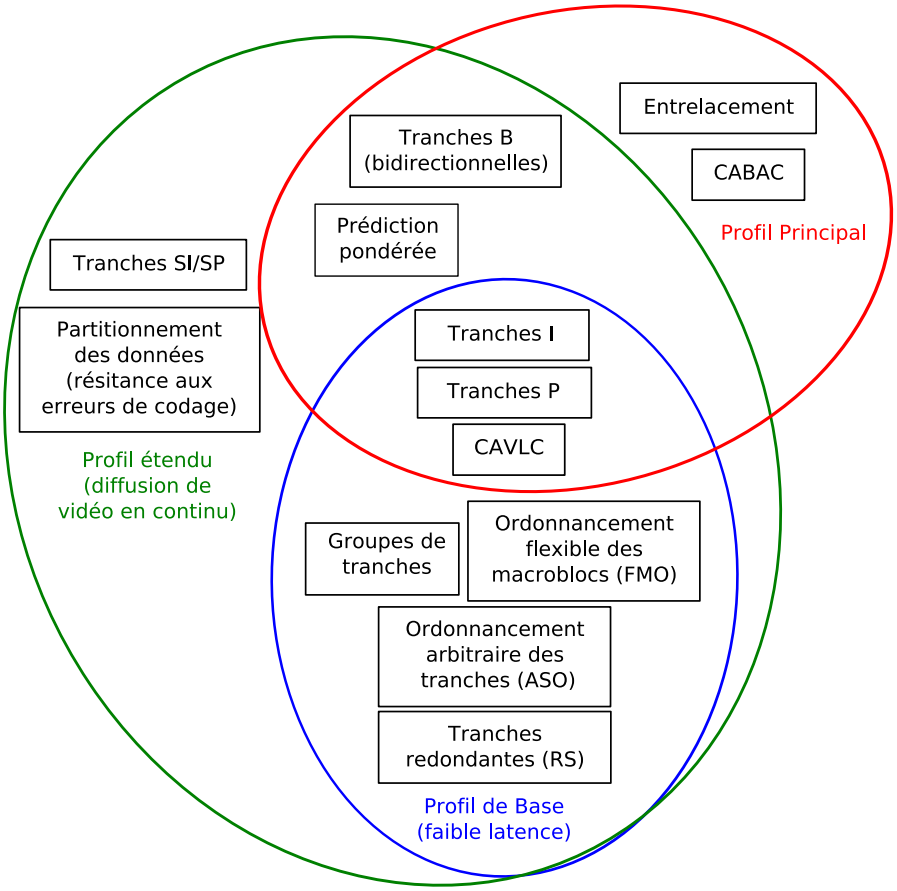


FIG. B.1 – Les profils dans H.264.

jusqu'à un ordre explicite de suppression ou de remplacement.

Une image codée puis reconstruite (au niveau du codeur) ou décodée (au niveau du décodeur) est placée dans le DPB, et est :

- soit marquée comme « inutile pour référence » (et ne sera pas utilisée pour les prochaines prédictions),
- soit marquée en tant qu'image à court terme (cas par défaut),
- soit marquée en tant qu'image à long terme,
- ou simplement envoyée à l'affichage.

Par défaut, les images à court terme de la liste 0 sont ordonnées du plus grand au plus petit *PicNum* (une variable dérivée du numéro d'image) et les images à long terme sont ordonnées depuis le plus petit vers le plus grand *LongTermPicNum*. Un changement de cet ordre peut être signalé par le codeur. Pour chaque nouvelle image à court terme ajoutée avec l'index 0, les indices des autres images à court terme sont incrémentés. Si le nombre d'images (court et long terme) est égal à la taille du DPB, l'image à court terme la plus ancienne (avec l'index le plus grand) est supprimée : le DPB agit comme un contrôleur de mémoire à fenêtre glissante.

Des commandes envoyées par le codeur permettent la gestion des index des images du DPB. Une image à court terme peut ainsi devenir une image à long terme, ou des images à court et long terme peuvent être marquées « inutile pour référence ». Dans le cas des tranches B, le codeur utilise deux listes d'images de référence : la liste 0 et la liste 1, contenant des images à court et long terme, passées ou futures selon l'ordre d'affichage. Les images à long terme se comportent de la même manière que celles à court terme. L'ordonnancement des images à court terme est le suivant :

– Liste 0 : l'image passée la plus proche (selon l'ordre du compteur d'images ou POC (*Picture Order Count*) est assignée à l'index 0, puis les autres images passées (en diminuant le POC) et les images futures (en augmentant le POC).
– Liste 1 : l'image future la plus proche est assignée à l'index 0, puis les autres images futures (POC augmentant) et les images passées (POC diminuant).

Le tableau B.1 donne un exemple de contenu d'un buffer de taille 5, l'image courante ayant un POC égal à 127.

Index	**Liste 0**	**Liste 1**
0	126	128
1	125	129
2	123	130
3	128	126
4	129	125
5	130	123

TAB. B.1 – Indice des images à court terme d'un DPB [Ric03].

Le codeur peut alors choisir une ou deux images de référence de la liste 0 et/ou de la liste 1 pour encoder chaque sous-macrobloc d'un macrobloc codé en inter. De plus, grâce à une possibilité de réarrangement de l'ordre des images dans les listes (en mettant l'image choisie à l'index 0), on réduit le coût de codage pour signaler la prédiction à partir de cette image.

B.3 Prédiction des vecteurs de mouvement

Soit E le macrobloc courant, partition de macroblocs ou de sous-macroblocs et A, B et C les partitions de macroblocs ou sous-macroblocs situées respectivement à gauche, au-dessus, au-dessus à droite de E. S'il y a plusieurs partitions à gauche de E, on choisit la plus haute pour A. De même pour la partition B, c'est celle située la plus à gauche qui sera retenue. La figure B.2 illustre ce principe.

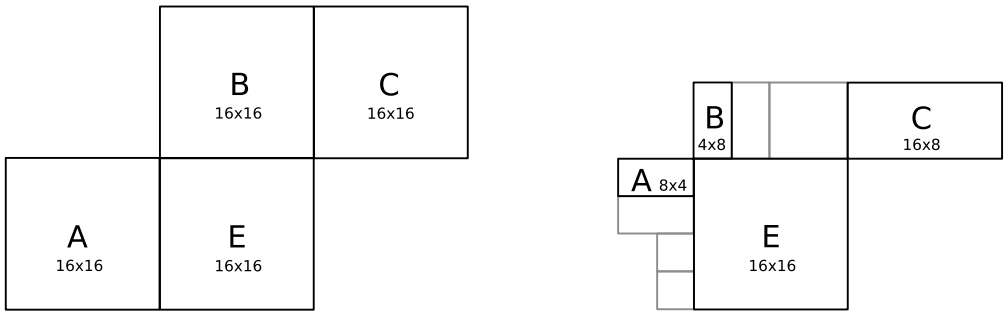


FIG. B.2 – Partitions courantes et voisines (à gauche pour des partitions de même taille et à droite pour des partitions de tailles différentes).

1. Pour les partitions transmises sauf 16×8 et 8×16, chaque composante de MV_p est la médiane respectivement des composantes des vecteurs de mouvement de A, B et C.
2. Pour les partitions de taille 16×8, MV_p pour la partition supérieure est prédit à partir de B, et pour la partition inférieure MV_p est prédit à partir de A.
3. Pour les partitions 8×16, MV_p pour la partition gauche est prédit à partir de A, et pour la partition droite à partir de C.

4. Pour les macroblocs SKIP (voir ci-dessous), MV_p est généré selon le cas 1 ci-dessus (comme si le bloc était codé en mode inter 16×16).

Si un ou plusieurs blocs ne sont pas disponibles (par exemple en dehors de la tranche), le choix du MV_p est modifié en conséquence. Du côté du décodeur, le vecteur prédit MV_p est créé de la même façon et ajouté à la différence MVD. Dans le cas des macroblocs SKIP, il n'y a pas de MVD et le bloc compensé en mouvement est obtenu en utilisant MV_p comme vecteur de mouvement.

Outre ces modes de prédiction, les macroblocs peuvent également être codés en mode SKIP. Dans ce cas, ne sont transmis ni signal d'erreur de prédiction quantifiée, ni index de référence, ni vecteur de mouvement. Le macrobloc est reconstitué comme s'il s'agissait d'un macrobloc 16×16 codé inter référençant l'image à l'index 0 du DPB.

B.4 Options de prédiction des tranches B

B.4.1 Bi-prédiction

Dans ce mode, un bloc de référence (de taille identique à la partition ou sous-partition de macrobloc courante) est créé à partir des listes 0 et 1 des images de référence. On obtient ainsi deux zones de référence compensées en mouvement obtenues respectivement à partir de la liste 0 et de la liste 1, ce qui implique deux vecteurs de mouvement. La prédiction finale est alors formée par la moyenne des blocs référencés par les deux vecteurs. À l'exception de la prédiction pondérée (voir la section B.4.3), le calcul de la bi-prédiction est réalisé de la manière suivante :

$P(i,j) = (P_0(i,j) + P_1(i,j) + 1)/2,$

où $P_0(i,j)$ and $P_1(i,j)$ sont les prédictions obtenues à partir des images références de la liste 0 et de la liste 1, et $P(i,j)$ est le résultat de la bi-prédiction. Après calcul de la prédiction, celle-ci est soustraite au macrobloc courant.

D'autre part, les vecteurs de mouvement de la liste 0 et de la liste 1 d'un macrobloc bi-prédit sont prédits à partir des vecteurs de mouvement voisins ayant la même direction temporelle. Par exemple un vecteur du bloc courant pointant sur une image passée est prédit à partir d'autres vecteurs voisins pointant également vers des images passées.

B.4.2 Prédiction directe

Dans ce mode, aucun vecteur de mouvement n'est transmis pour coder un macrobloc ou une partition de macrobloc. À la place le décodeur calcule les vecteurs pour la liste 0 et la liste 1 selon les vecteurs précédemment codés, et utilise ces vecteurs calculés pour réaliser la bi-prédiction à compensation de mouvement à partir des échantillons résiduels décodés.

Un témoin dans l'entête de la tranche indique la méthode de calcul des vecteurs pour la prédiction directe. Deux méthodes existent [FG03] :

– Le mode spatial direct : les vecteurs de mouvement prédits des listes 0 et 1 sont calculés identiquement à la prédiction des vecteurs de mouvement décrite pour les tranches P (voir la section B.3). Si le macrobloc ou la partition, co-localisé dans la première image de référence de la liste 1, a un vecteur de mouvement d'ampleur inférieure à un demi pixel (de luminance), un ou les deux vecteurs prédits sont fixés à zéro ; sinon les vecteurs prédits des listes 0 et 1 sont utilisés pour la bi-prédiction à compensation de mouvement.

– Le mode temporel direct : la méthode appliquée est la suivante :

1. Trouver l'image de référence de la liste 0 pour le macrobloc ou partition co-localisé dans l'image de la liste 1.
2. Trouver le vecteur de la liste 0, VM_C, pour le macrobloc ou partition co-localisé dans l'image de la liste 1.
3. Interpoler le vecteur VM_C en fonction de la « distance » entre l'image courante et celle de liste 1 : c'est le nouveau vecteur VM_1 de la liste 1.
4. Interpoler le vecteur VM_C en fonction de la « distance » entre l'image courante et celle de la liste 0 : c'est le nouveau vecteur VM_0 de la liste 0.

La figure B.3 illustre ce procédé. VM_0 et VM_1 sont calculés en fonction des distances temporelles DT_B et DT_D.

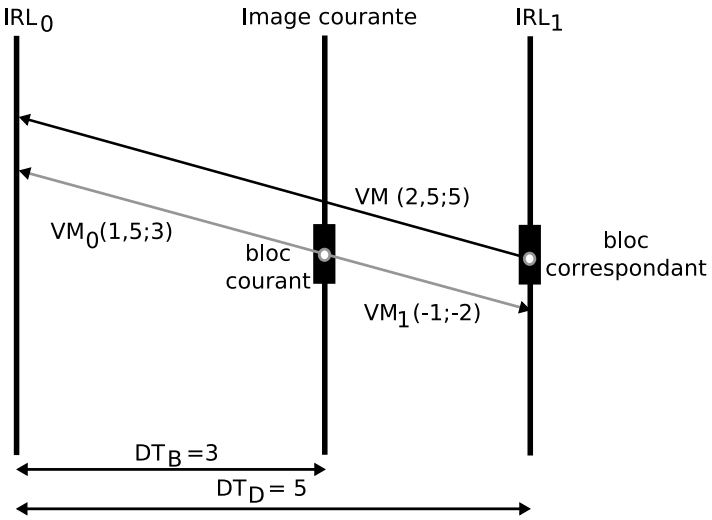


FIG. B.3 – Dans le mode temporel direct, un bloc a deux vecteurs de mouvement dérivés VM_0 et VM_1 pointant vers deux images de référence de la liste 0, IRL_0, et de la liste 1 , IRL_1.

Ces modes peuvent cependant être modifiés, par exemple si les macroblocs de référence ne sont pas disponibles ou s'ils sont codés en intra.

B.4.3 Prédiction pondérée

La prédiction pondérée est une méthode qui modifie les valeurs des données prédites des macroblocs des tranches P ou B. Il existe trois types de prédiction pondérée dans le standard H.264 :

1. prédiction pondérée « explicite » pour les macroblocs des tranches P,
2. prédiction pondérée « explicite » pour les macroblocs des tranches B,
3. prédiction pondérée « implicite » pour les macroblocs des tranches B.

Chaque échantillon prédit $P_0(i,j)$ ou $P_1(i,j)$ est pondéré par un facteur w_0 ou w_1 en fonction de la prédiction à compensation de mouvement. Pour les types « explicites », les facteurs de pondération sont déterminés par le codeur et transmis dans l'entête. Si la prédiction « implicite » est utilisée, w_0 et w_1 sont calculés en fonction des positions temporelles relatives de la liste 0 et de la liste 1 d'images référence. Si l'image de référence est temporellement proche de l'image courante, un facteur de pondération important est utilisé ; au

contraire, un facteur plus faible sera appliqué si l'image de référence est éloignée temporellement de l'image courante. La prédiction pondérée permet le contrôle explicite ou implicite des contributions relatives des images de référence lors de la prédiction à compensation de mouvement.

B.5 Codage entropique

Les données arrivant à l'unité de codage entropique sont nombreuses et variées. Ce sont essentiellement les données résiduelles après quantification mais également des entêtes, des vecteurs de mouvement, les méthodes de prédiction, les index des images de référence, ... Après une première étape de réarrangement des données dans un tableau, le codeur H.264 applique plusieurs méthodes de codage entropique, pour les éléments syntaxiques et pour les coefficients de transformée quantifiés.

B.5.1 Codage Exp-Golomb

Ce codage à longueur variable [Gol66] utilise un unique ensemble illimité de mots-codes défini pour tous les éléments syntaxiques (sauf pour les données résiduelles quantifiées). Les différentes tables de code à longueur variable (VLC) sont remplacées par une seule table, personnalisée en fonction des statistiques des données. Cette table est un code de Golomb exponentiel aux propriétés de décodage simples et régulières.

B.5.2 CAVLC

Le codage entropique CAVLC (*Context-based Adaptive Variable Length Coding*) [BL02] est utilisé pour le codage des coefficients de transformée quantifiés. Ce codage entropique parcourt les blocs 4×4 en zigzag et tire avantage des blocs 4×4 quantifiés :

1. Après prédiction, transformation et quantification, les blocs contiennent principalement des éléments nuls. CAVLC les représente de manière compacte.
2. Les coefficients non nuls après le parcours en zigzag sont souvent des séquences de 1 et -1. CAVLC signale le nombre de ces coefficients de manière compacte.
3. Le nombre de coefficients non nuls dans des blocs voisins est corrélé. Le nombre de coefficients est codé avec une table de correspondance.
4. L'amplitude des coefficients non nuls tend à être plus grande au début du tableau réarrangé (près des coefficients de la composante continue). CAVLC en tire avantage en adaptant le choix des tables de correspondances (LUT[1]) en fonction des amplitudes déjà codées.

B.5.3 CABAC

La méthode CABAC (*Context-Adaptive Binary Arithmetic Coding*) [MBW01, MSW03] disponible dans le profil principal, améliore encore le codage entropique. Le procédé de codage applique les étapes suivantes :

1. Binarisation : CABAC utilise un codage arithmétique binaire, c'est-à-dire, ne codant que les éléments binaires (0 ou 1). Les symboles non binaires sont « binarisés » ou convertis en code binaire avant le codage arithmétique. Ce procédé est semblable au procédé de transformation d'un symbole en un mot

[1] *Look Up Table*

de code de longueur variable (voir le chapitre 1) mais le code binaire est ensuite codé (par le codeur arithmétique) avant d'être transmis.

Les étapes 2, 3 et 4 sont répétées pour chaque bit ou *bin* du symbole binarisé.

2. Sélection du modèle de contexte : un « modèle de contexte » est un modèle de probabilité pour un ou plusieurs *bins* du symbole binarisé et est choisi parmi une sélection de modèles disponibles en fonction des statistiques des symboles récemment codés. Il enregistre la probabilité pour chaque *bin* du symbole converti d'être un « 1 » ou un « 0 ».
3. Codage arithmétique : il code chaque *bin* en fonction du modèle de probabilité sélectionné
4. Mise à jour de la probabilité : le modèle de contexte sélectionné est mis à jour par rapport à la valeur courante codée (si la valeur du *bin* était un « 1 » alors le compteur de « 1 » est incrémenté).

Par rapport au CAVLC, CABAC garantit en général une réduction supplémentaire du débit binaire de 10 à 15% lors du codage de signaux télévisuels pour une même qualité.

Le codage entropique, dernier élément du codeur H.264 avant la génération du flux binaire est un élément important dans le codeur. La compression peut être efficace et permet une dernière optimisation du codage des données répétitives.

Annexe C

Résultats de la modification adaptative de la structure du GOP

C.1 Variation dynamique du nombre d'images B

Les figures C.1 à C.8 illustrent la variation dynamique du nombre d'images B en fonction de l'information temporelle *IT* et de l'information spatiale *IS* des segments temporels pour les séquences *New Mobile and Calendar*, *Night*, *Knightshields*, *Parkrun*, *Umbrella*, *Parkjoy*, *Crew* et *Tractor*.

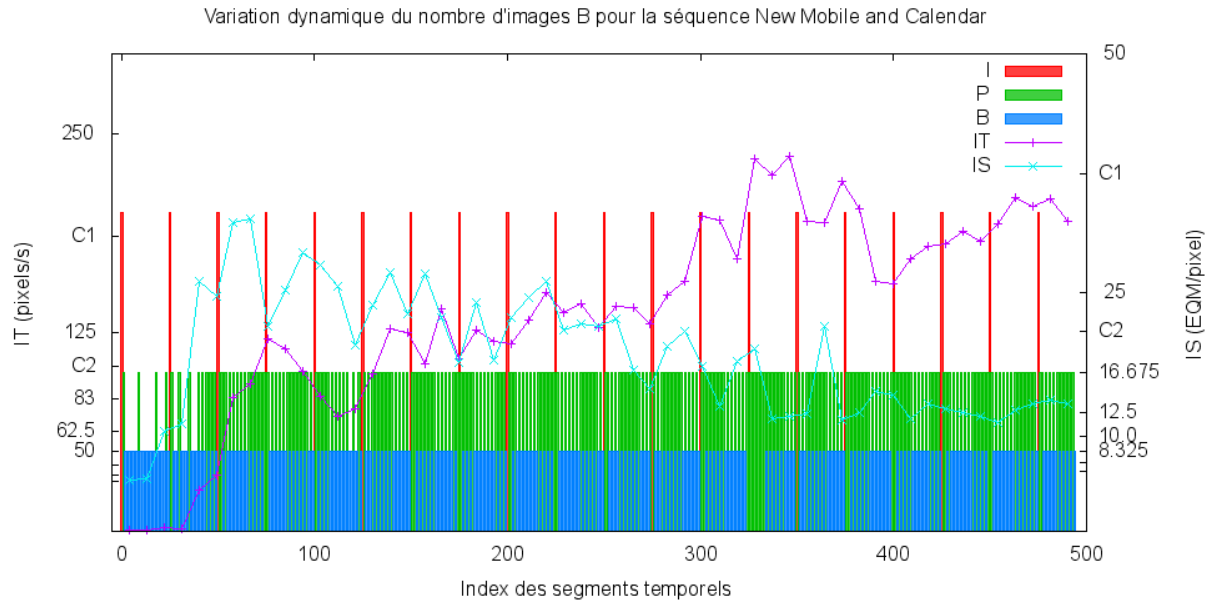


FIG. C.1 – Variation dynamique du nombre d'images B pour la séquence *New Mobile and Calendar*.

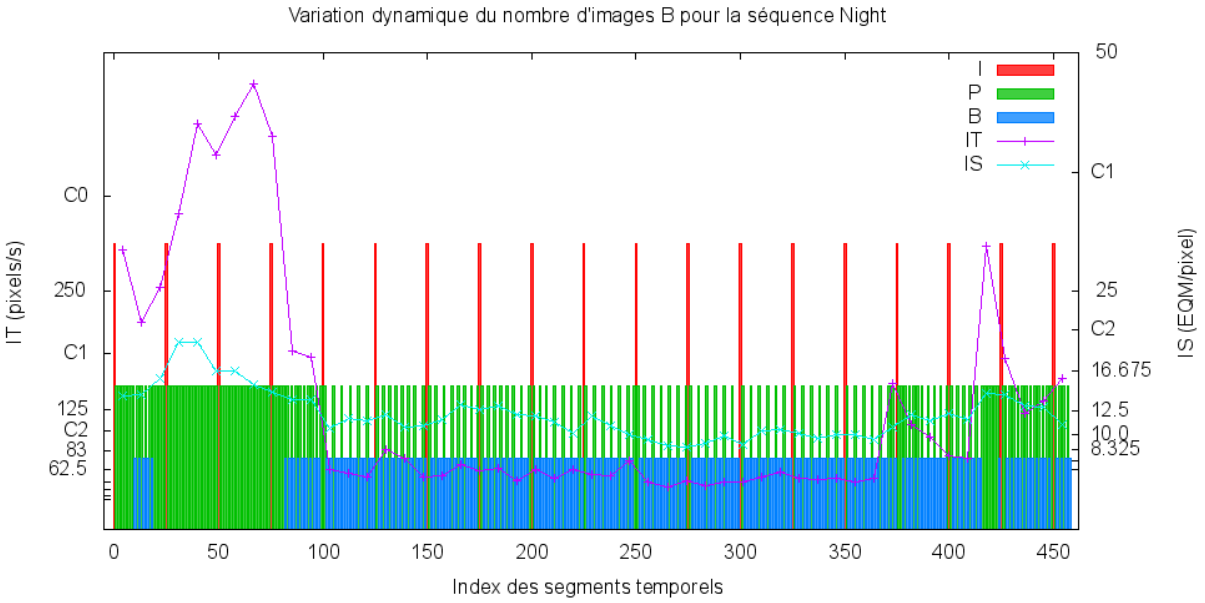


FIG. C.2 – Variation dynamique du nombre d'images B pour la séquence *Night*.

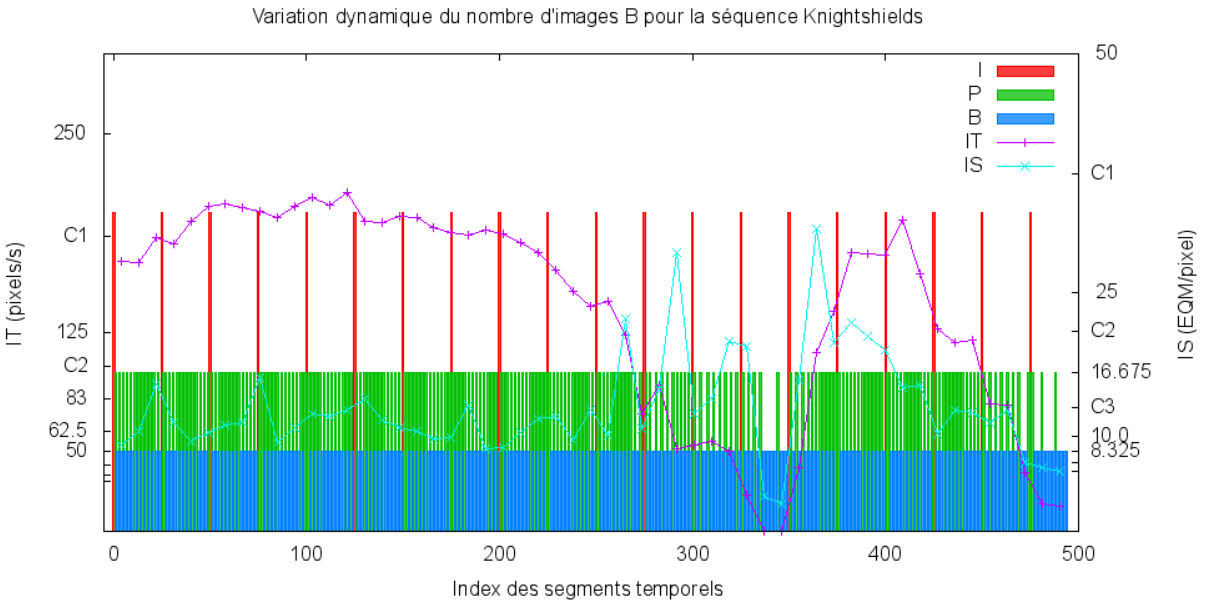


FIG. C.3 – Variation dynamique du nombre d'images B pour la séquence *Knightshields*.

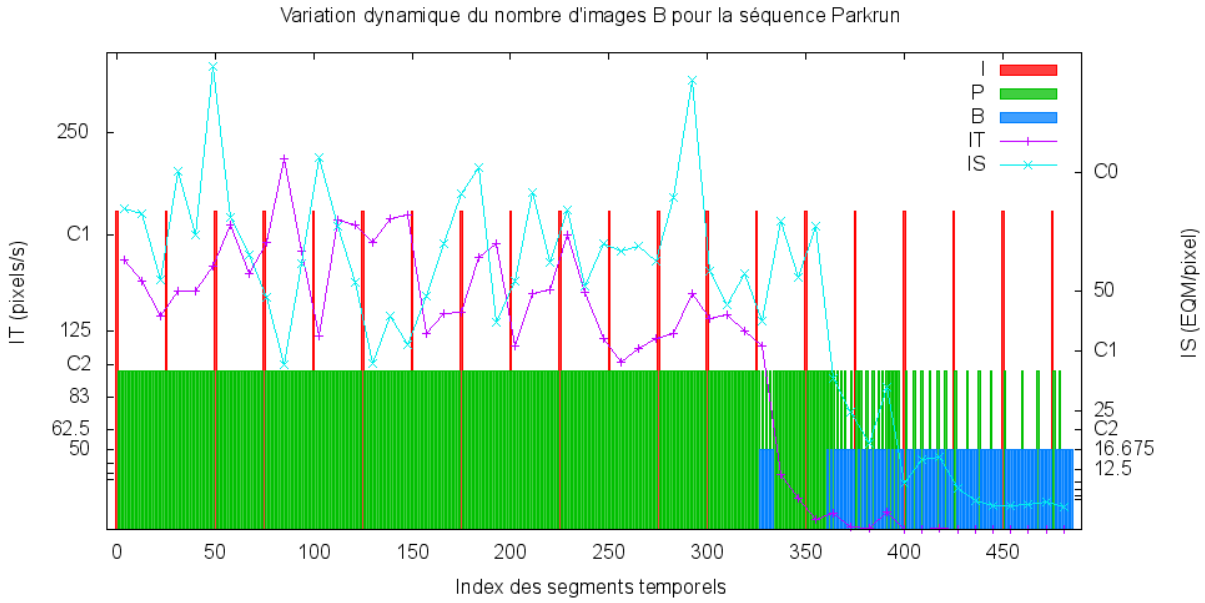


FIG. C.4 – Variation dynamique du nombre d'images Bpour la séquence *Parkrun.*

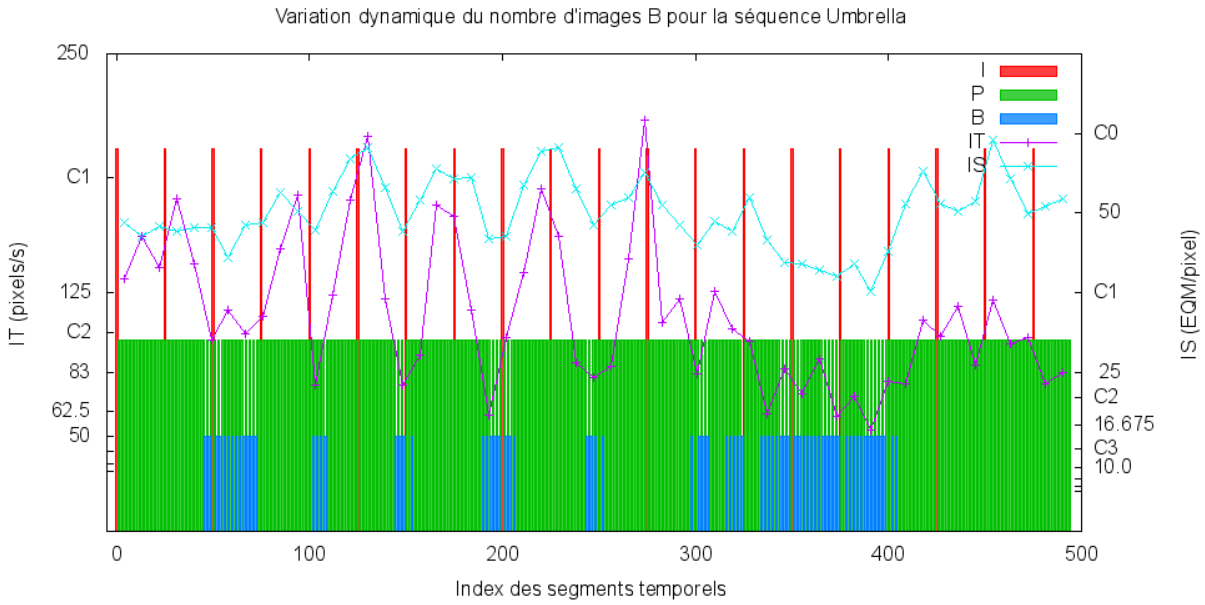


FIG. C.5 – Variation dynamique du nombre d'images B pour la séquence *Umbrella*.

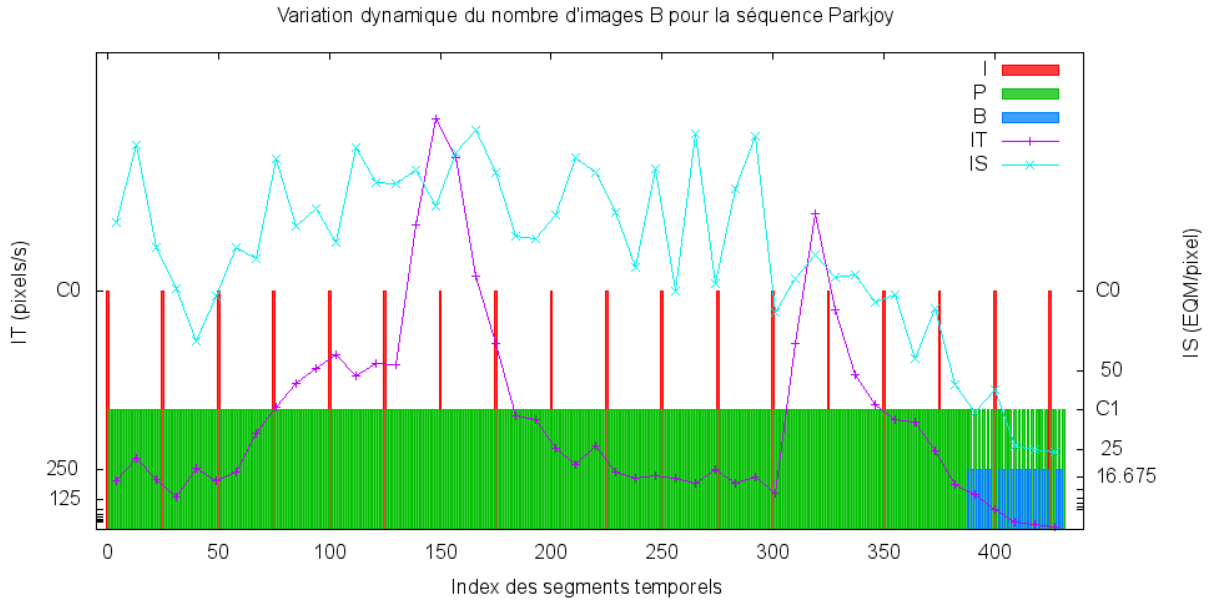


FIG. C.6 – Variation dynamique du nombre d'images B pour la séquence *Parkjoy*.

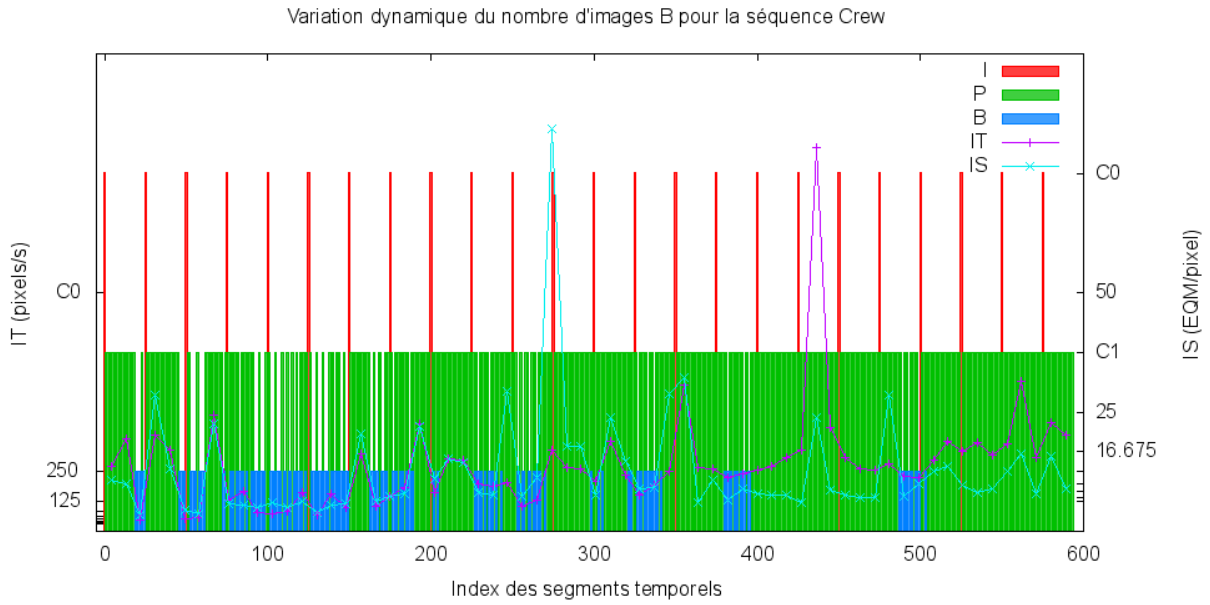


FIG. C.7 – Variation dynamique du nombre d'images B pour la séquence *Crew*.

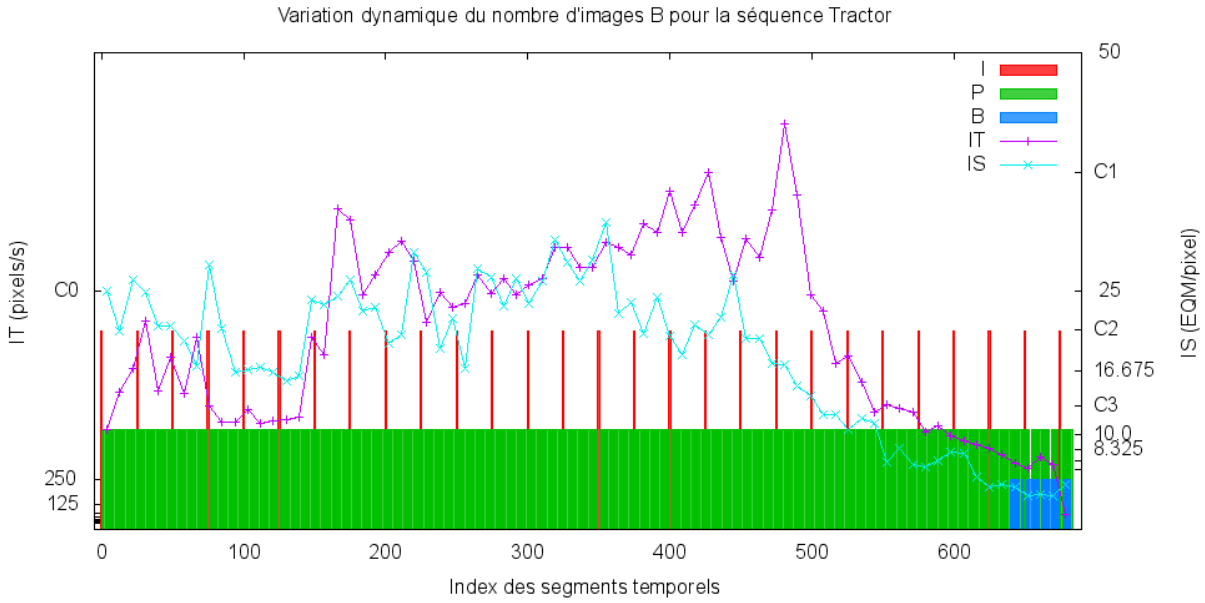


FIG. C.8 – Variation dynamique du nombre d'images B pour la séquence *Tractor*.

C.2 Évolution de l'activité temporelle

Les figures C.9 à C.16 présentent l'évolution de l'activité temporelle aux seins des segments temporels pour les séquences *New Mobile and Calendar*, *Night*, *Knightshields*, *Parkrun*, *Umbrella*, *Parkjoy*, *Crew* et *Tractor*.

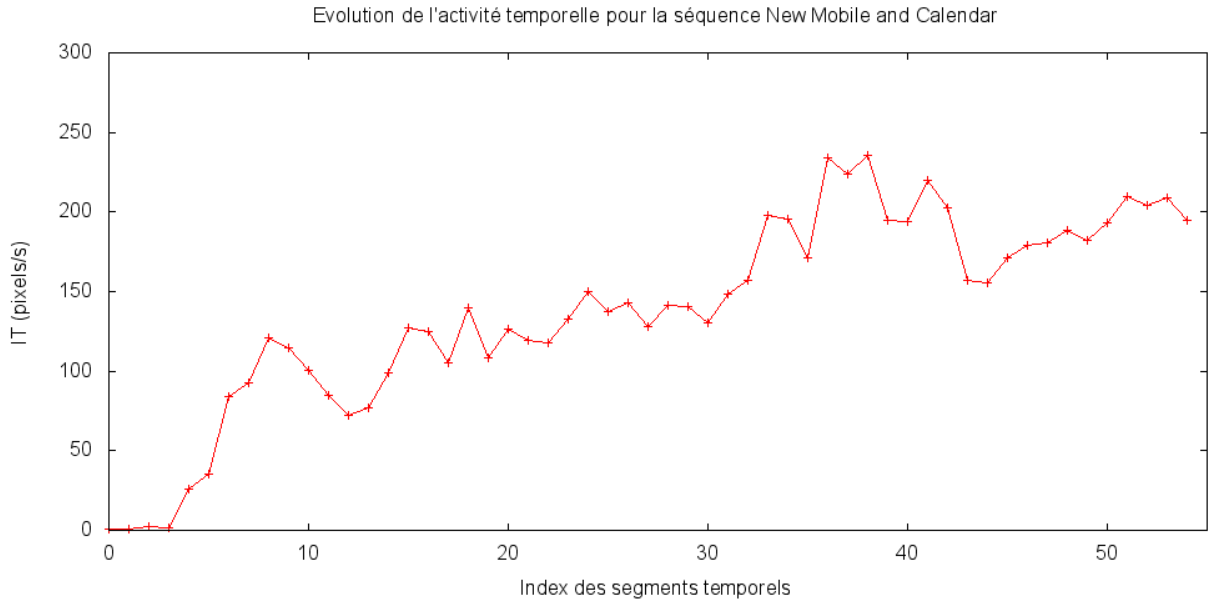


FIG. C.9 – Évolution de l'activité temporelle pour la séquence *New Mobile and Calendar*.

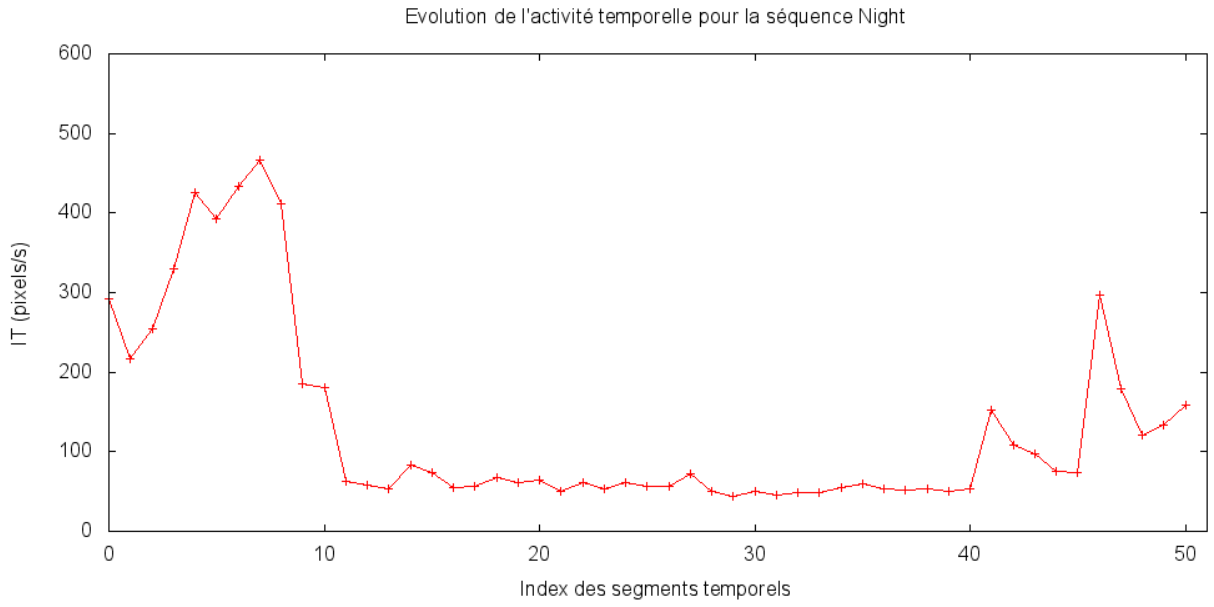


FIG. C.10 – Évolution de l'activité temporelle pour la séquence *Night*.

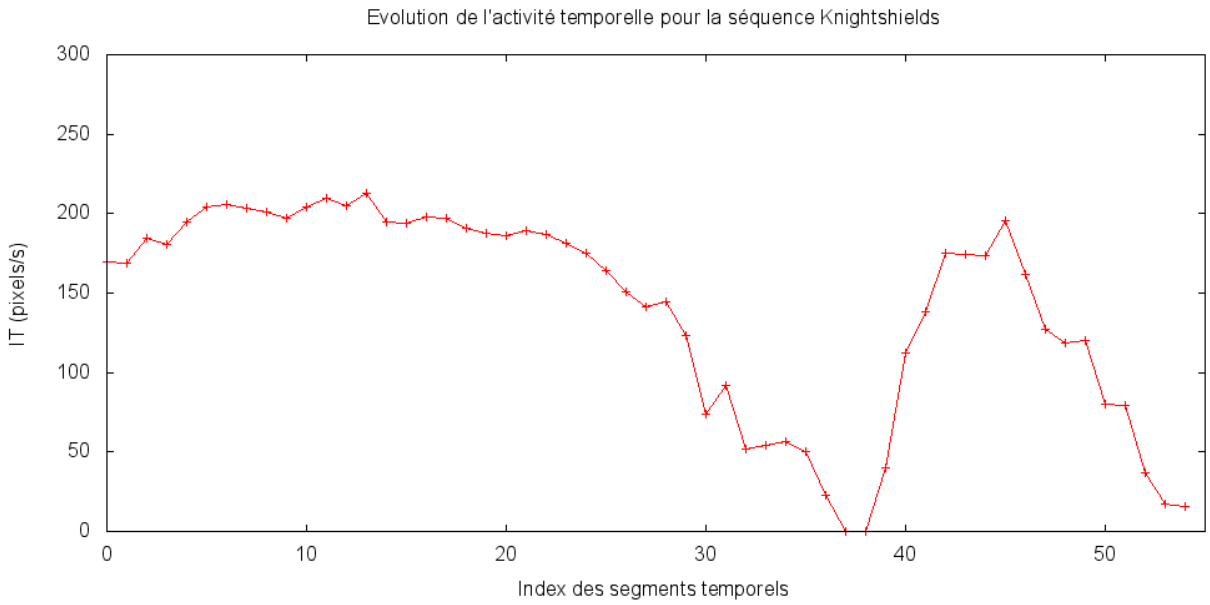


FIG. C.11 – Évolution de l'activité temporelle pour la séquence *Knightshields*.

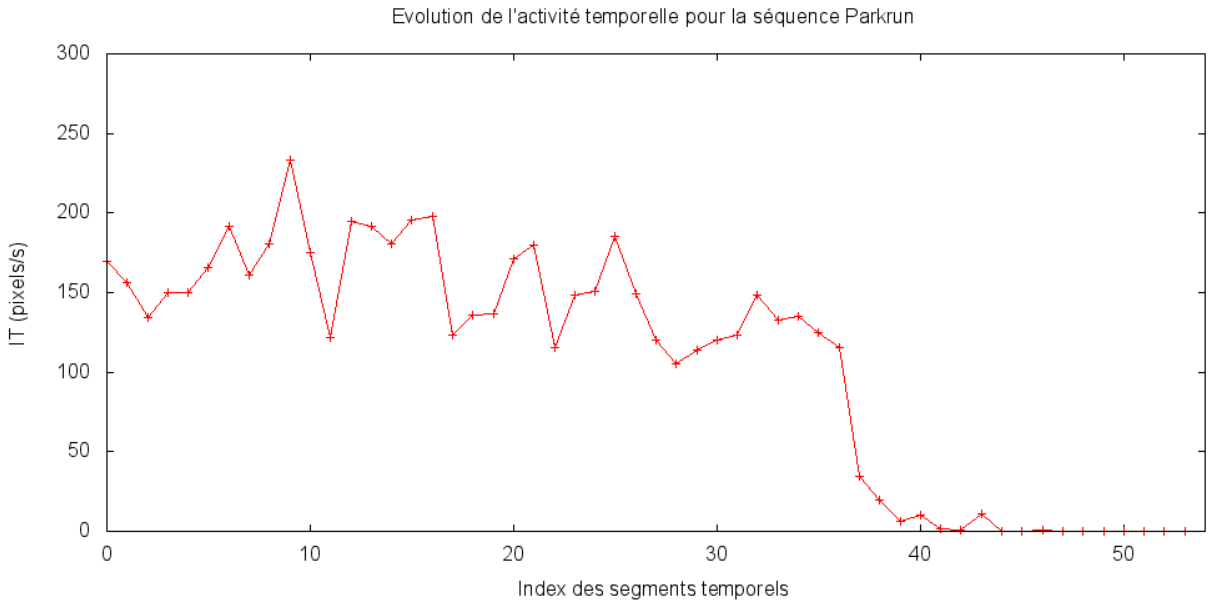


FIG. C.12 – Évolution de l'activité temporelle pour la séquence *Parkrun*.

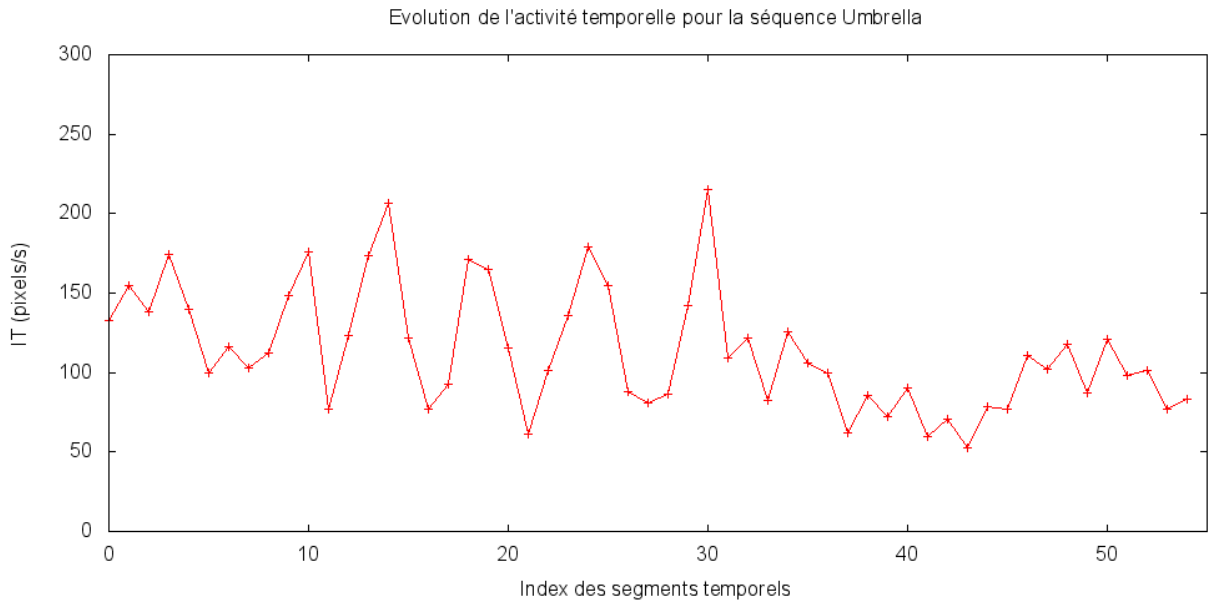


FIG. C.13 – Évolution de l'activité temporelle pour la séquence *Umbrella*.

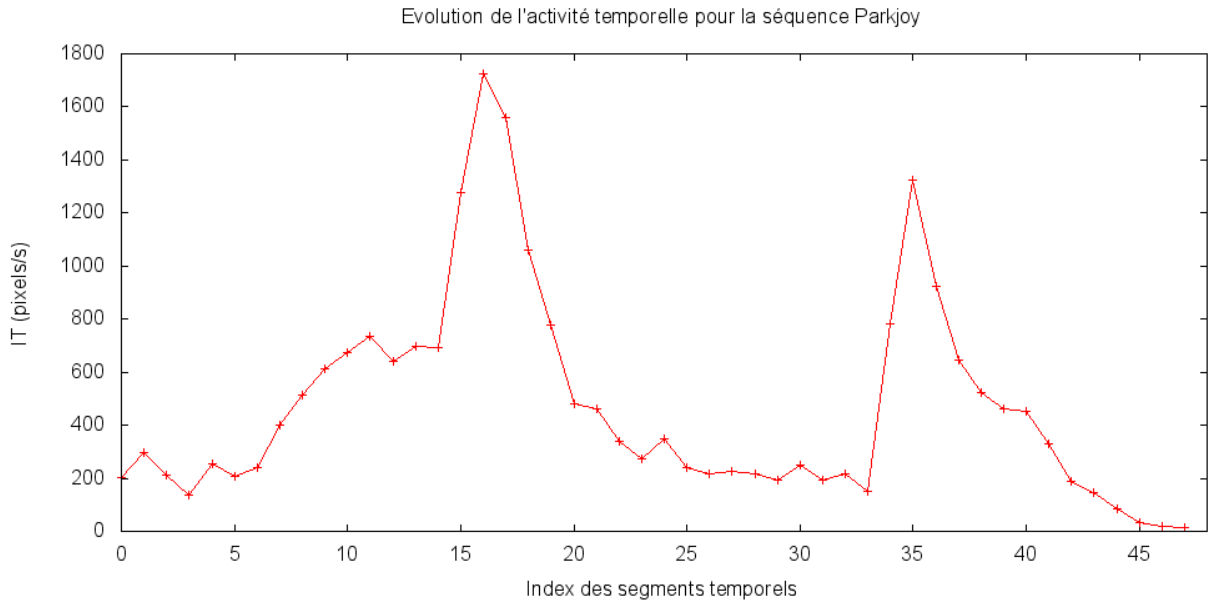


FIG. C.14 – Évolution de l'activité temporelle pour la séquence *Parkjoy*.

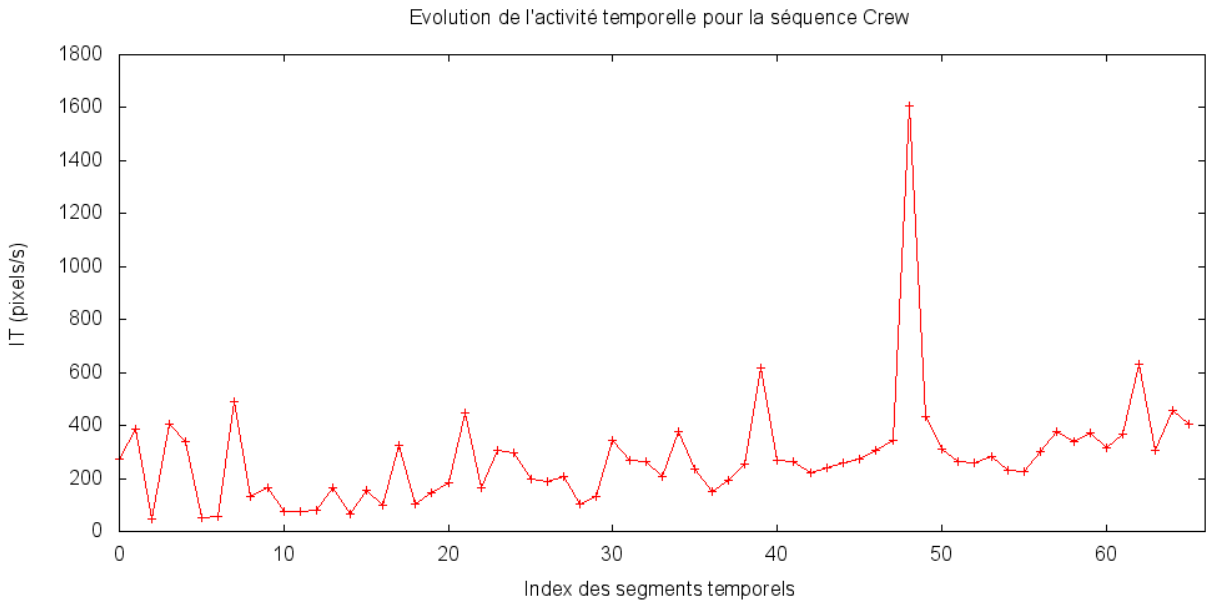


FIG. C.15 – Évolution de l'activité temporelle pour la séquence *Crew*.

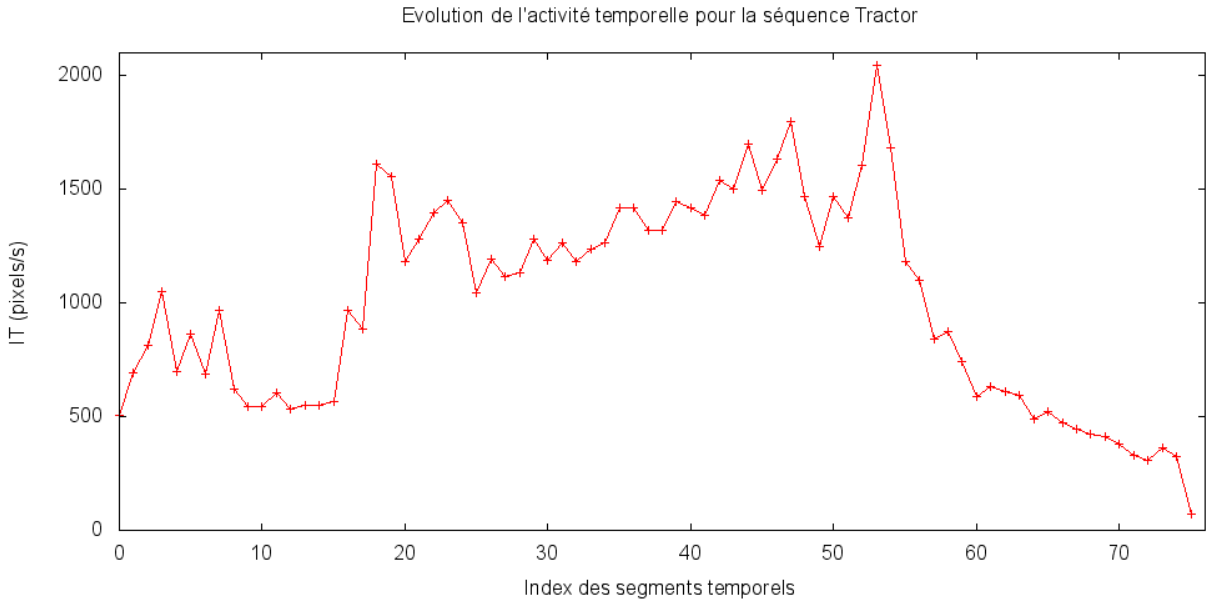


FIG. C.16 – Évolution de l'activité temporelle pour la séquence *Tractor*.

C.3 Modification adaptative de la structure du GOP

Les figures C.17 à C.19 illustrent la modification adaptative de la structure du GOP en fonction du mouvement et de l'EQM des segments temporels pour les séquences *New Mobile and Calendar*, *Night*, *Knightshields*, *Parkrun*, *Umbrella*, *Parkjoy*, *Crew* et *Tractor*.

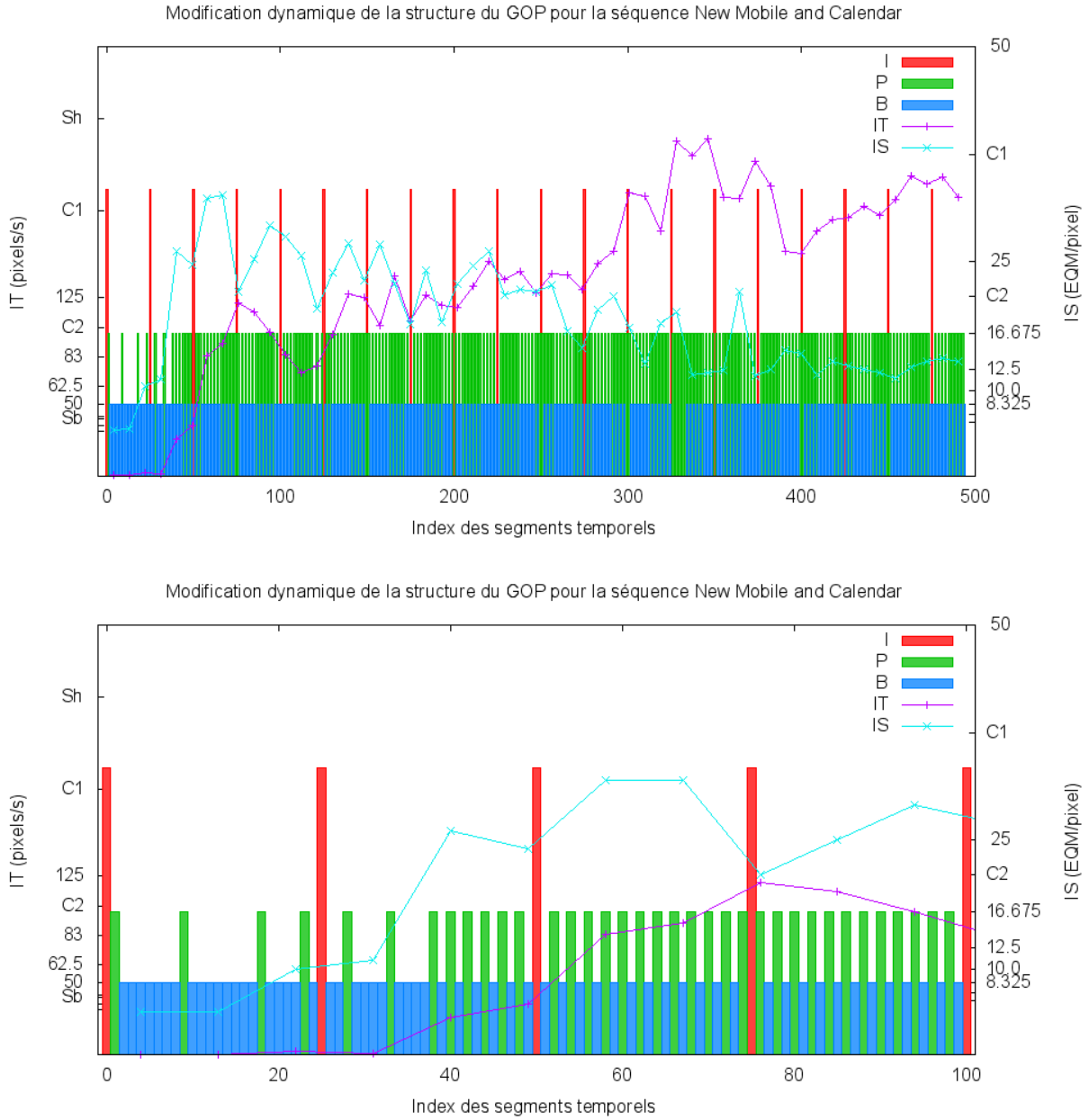


FIG. C.17 – Type des images pour la séquence *New Mobile and Calendar* : a) pour la séquence entière, b) pour les images 0 à 100.

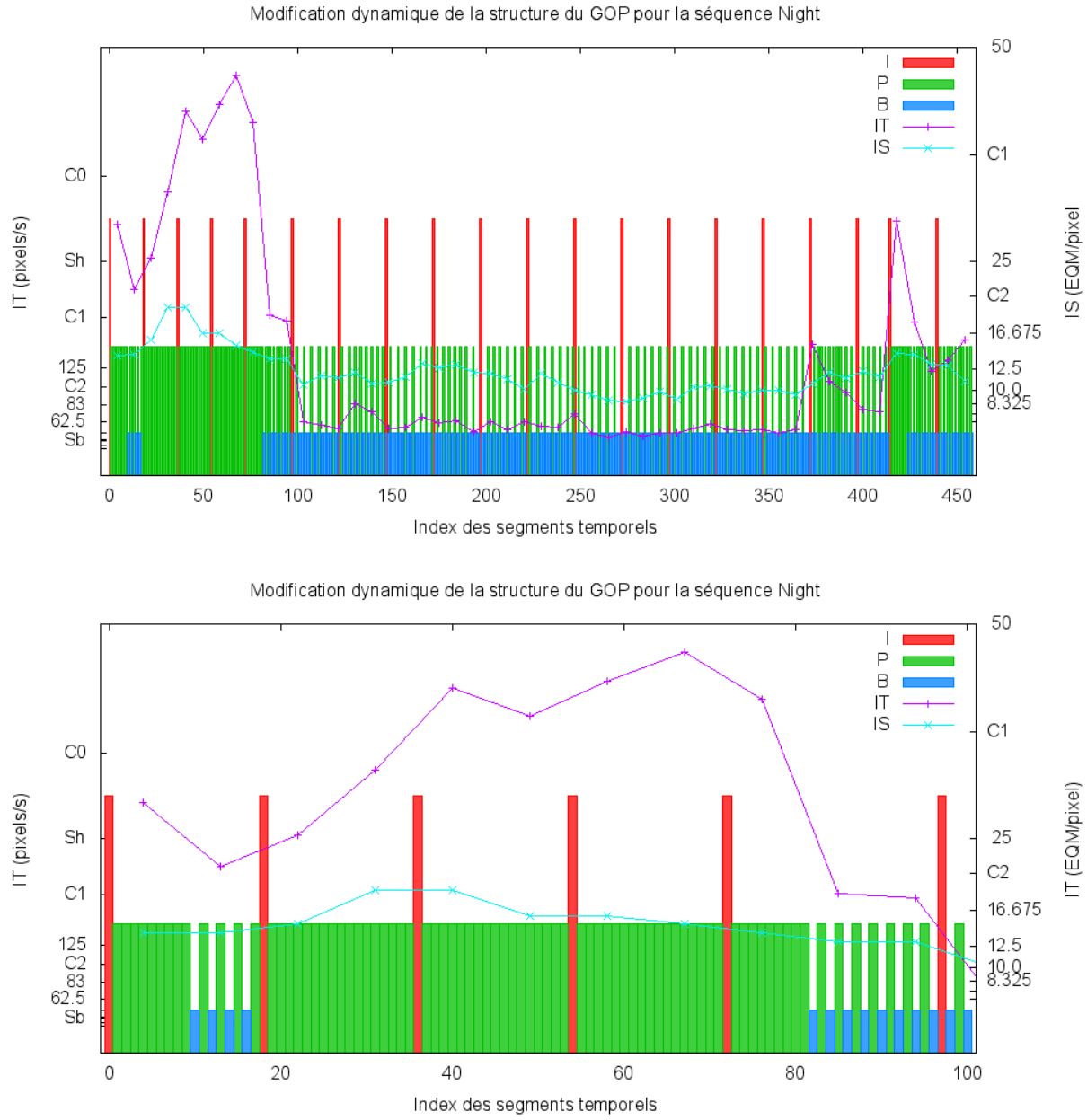


FIG. C.18 – Type des images pour la séquence *Night* : a) pour la séquence entière, b) pour les images 0 à 100.

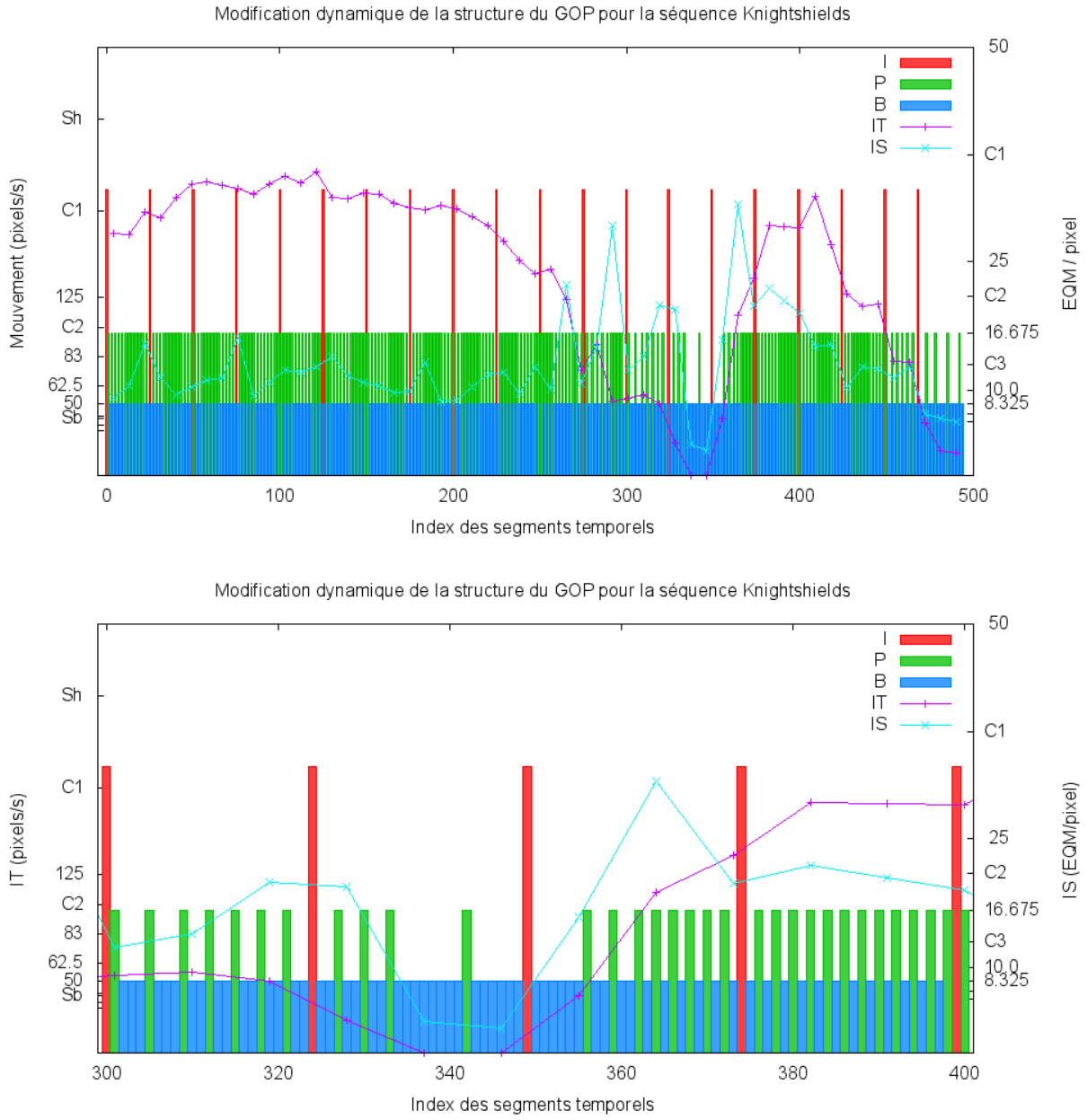


FIG. C.19 – Type des images pour la séquence *Knightshields* : a) pour la séquence entière, b) pour les images 300 à 400.

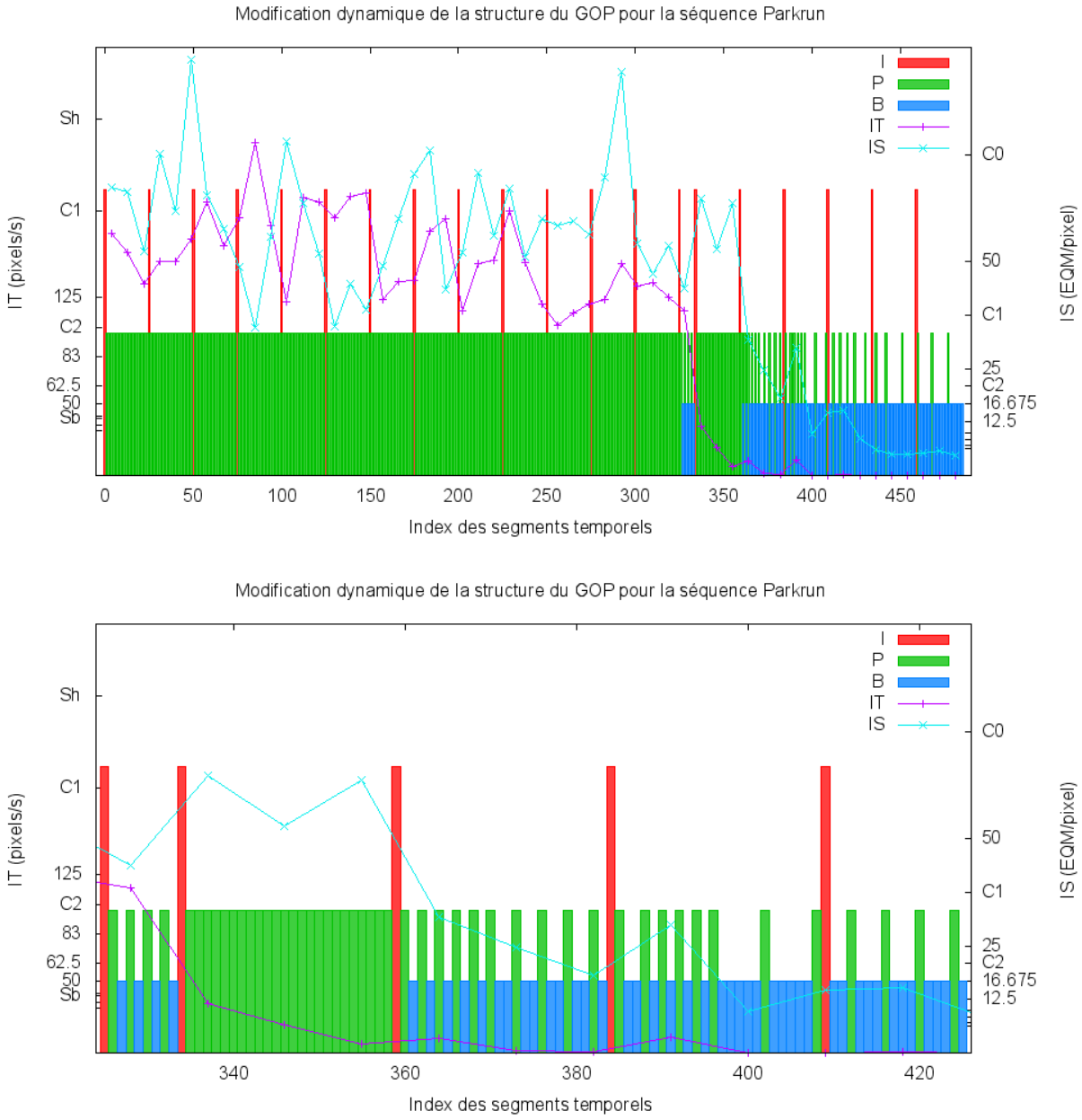


FIG. C.20 – Type des images pour la séquence *Parkrun* : a) pour la séquence entière, b) pour les images 325 à 425.

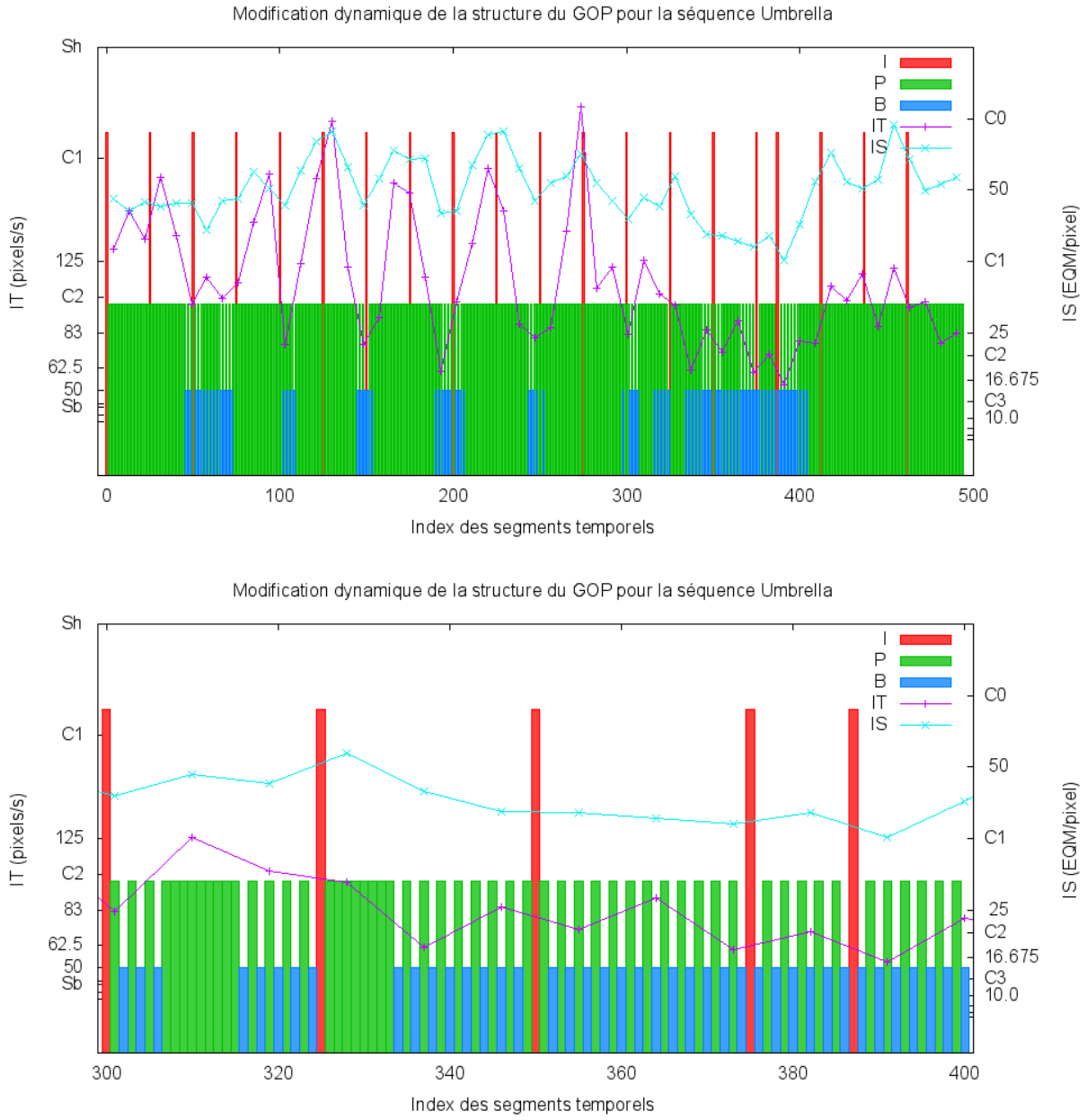


FIG. C.21 – Type des images pour la séquence *Umbrella* : a) pour la séquence entière, b) pour les images 300 à 400.

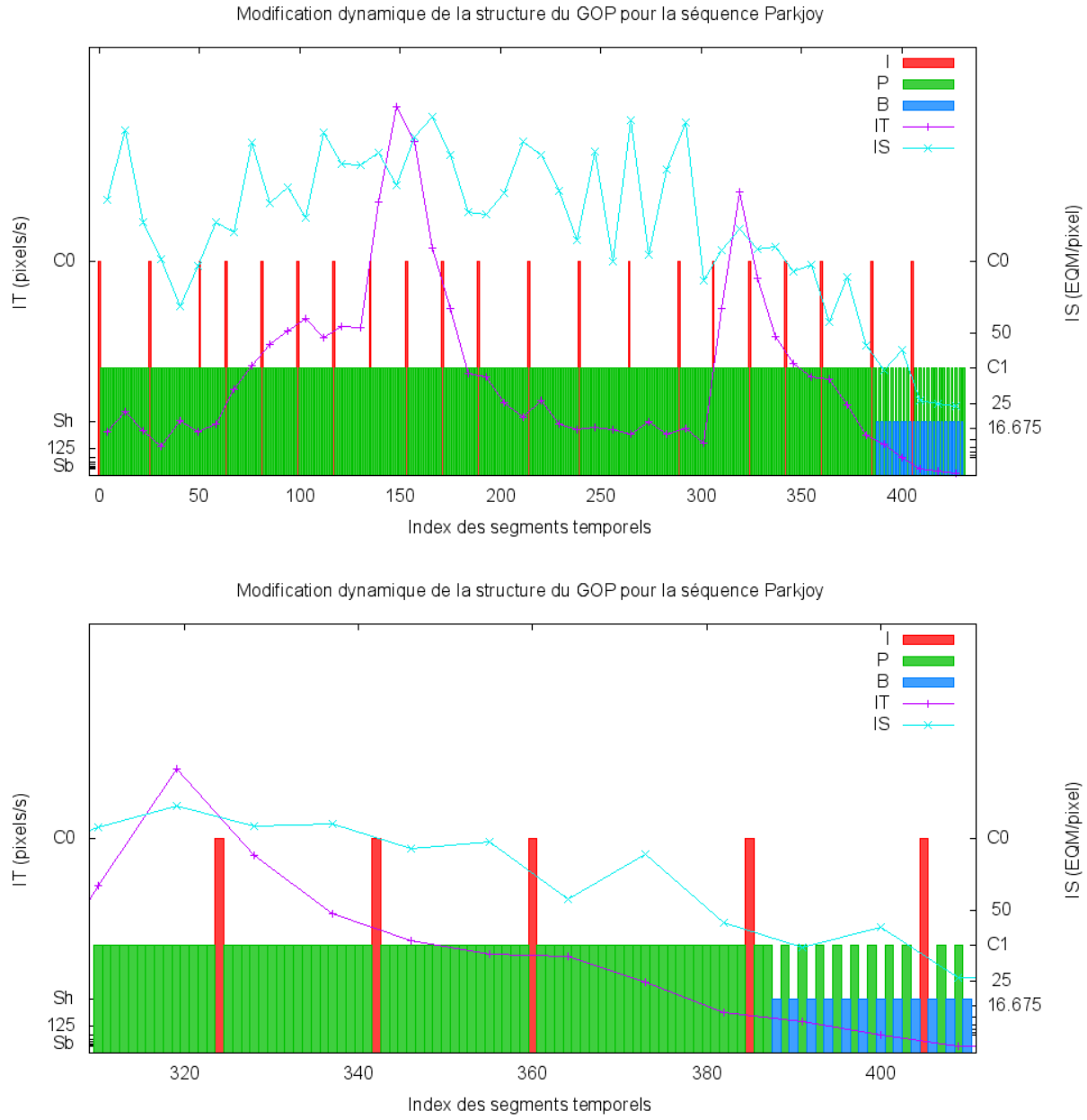


FIG. C.22 – Type des images pour la séquence *Parkjoy* : a) pour la séquence entière, b) pour les images 310 à 410.

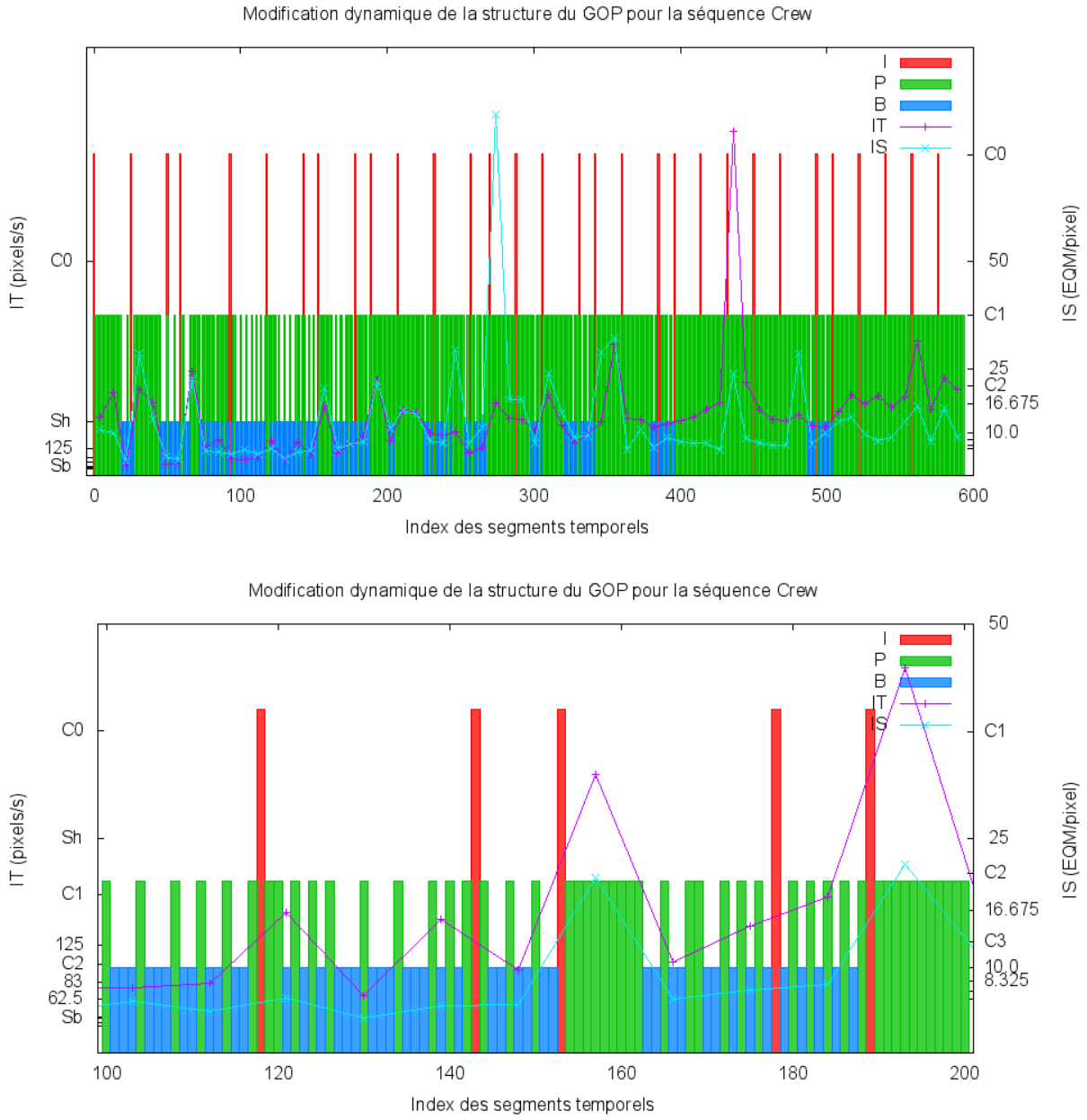


FIG. C.23 – Type des images pour la séquence *Crew* : a) pour la séquence entière, b) pour les images 100 à 200.

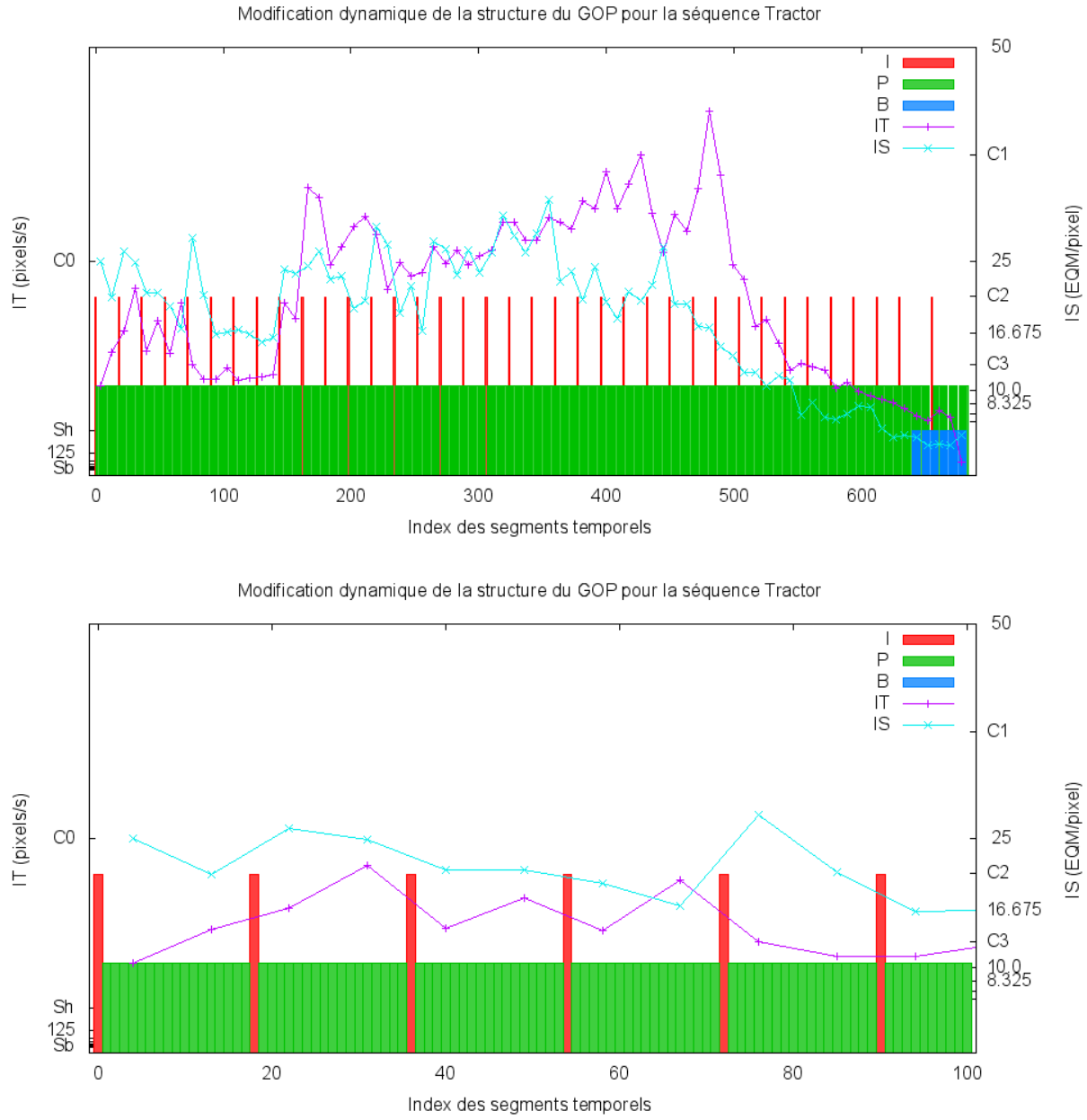


FIG. C.24 – Type des images pour la séquence *Tractor* : a) pour la séquence entière, b) pour les images 0 à 100.

Bibliographie

[AAN93] Jr. AHUMADA, J. ALBERT et Cynthia H. NULL : *Digital Images and Human Vision*, chapitre Image quality : A multidimensional problem, pages 141 – 148. MIT Press, Cambridge, MA, USA, 1993.

[AD93] M. ALLMEN et C. R. DYER : Computing Spatiotemporal Relations for Dynamic Perceptual Organization. *CVGIP : Image Understanding*, 3(58):338 – 351, 1993.

[Agn01] Vincent AGNUS : *Segmentation spatio-temporelle de séquences d'images par des opérateurs de morphologie mathématique*. Thèse de doctorat, Université de Louis Pasteur, Strasbourg, 2001.

[AKA06] Hasan F. ATES, Beakay KANBEROGLU et Yucel ALTUNBASAK : Rate Distorsion and Complexity Joint Optimization for fast Motion Estimation in H.264 Video Coding. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'2006*, Atlanta, GA, USA, octobre 2006.

[AM08] Muhammad Zaheer AZIZ et Barbel MERTSCHING : Fast and Robust Generation of Feature Maps for Region-Based Visual Attention. *IEEE Transactions on Image Processing*, 17(5):633 – 644, 2008.

[AOW+98] A. Aydin ALATAN, Levent OUNRAL, Michael WOLBORN, Roland MECH, Ertem TUNCEL et Thomas SIKORA : Image Sequence Analysis for Emerging Interactive Mutlimedia Services – The European COST 211 Framework. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(7):802 – 813, novembre 1998.

[BA83] P.J. BURT et E.H. ADELSON : The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communications*, 31:532 – 540, 1983.

[BAA06] Seydou-Nourou BA, Yucel ATLTUNBASAK et Hasan ATES : Low Complexity Inter Mode Selection for H.264. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'2006*, Atlanta, GA, USA, octobre 2006.

[Bal91] D. BALLARD : Animate vision. *Artificial Intelligence*, 86:48 – 57, 1991.

[Bar04] P.G. BARTEN : Formula for the contrast sensitivity of the human eye. *In SPIE Human Vision and Electronic Imaging*, San Jose, CA, USA, 2004.

[BB92] J. BENOIS et D BARBA : Image segmentation by region-contour cooperation for image coding. *In Proceedings of the International Conference on Pattern Recognition, ICPR'1992*, volume C, pages 331 – 334, 1992.

[BCA78] O. BRADDICK, F. W. CAMPBELL et J. ATKINSON : *Handbook of Sensory Physiology*, volume 8, chapitre Channels in vision : Basic aspects, pages 3 – 38. Springer-Verlag, Berlin, 1978.

[BCB99] G. BALDI, C. COLOMBO et Del BIMBO : A Compact and Retrieval-Oriented Video Representation Using Mosaics. *In Proceedings of the International Conference on Visual Information Systems (VISUAL)*, pages 171 – 178, 1999.

[Bed98] L. BEDAT : *Aspects psychovisuels de la perception des couleurs. Application au codage d'images couleurs fixes avec compression de l'information*. Thèse de doctorat, Université de Nantes, IRESTE, 1998.

[Ber71] Toby BERGER : *Rate Distortion Theory : A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ : Prentice-Hall, 1971.

[Bes74] Julian BESAG : Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society*, B-48(3):192 – 236, 1974.

[BG99] H. BROSZIO et SO. GRAU : Robust Estimation of Camera Parameters Pan, Tilt and Zoom for Integration of Virtual Objects into Video Sequences. *In International Workshop on Synthetic-Natural Hybrid Coding and Three-Dimensional Imaging, IWSNHC3DI'99*, Fira, Santorini, Grèce, septembre 1999.

[BH07] Alexandre BUR et Heinz HÜGLI : Optimal cue combination for saliency computation : A comparison with human vision. *In IWINAC '07 : Proceedings of the 2nd international work-conference on Nature Inspired Problem-Solving Methods in Knowledge Engineering*, pages 109 – 118, Berlin, Heidelberg, 2007. Springer-Verlag.

[BHM80] H. BODMANN, P. HAUBNER et A. MARSEDN : A unified relationship between brightness and luminance. *CIE Proceedings of Kyoto Session*, pages 99 – 102, 1980.

[BHT+07] Yukihiro BANDOH, Kazuya HAYASE, Seishi TAKAMURA, Kazuto KAMIKURA et Yoshiyuki YASHIMA : Mode Decision for H.264/AVC based on Spatio-Temporal Sensitivity. *In Proceedings of the Picture Coding Symposium, PCS 2007*, Lisbonne, Portugal, novembre 2007.

[BJ03] N. BRUCE et E. JERNIGAN : Evolutionary Design of Context-Free Attentional Operators. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2003*, Barcelone, Espagne, septembre 2003.

[Bjo01] Gisle BJONTEGAARD : Calculation of average PSNR Differences between RD-curves. ITU- Telecommunications Standardization Sector, Video Coding Expert Group (VCEG) 13ème Meeting : Austin, Texas, USA, avril 2001.

[BL02] Gisle BJONTEGAARD et Karl LILLEVOLD : *Context-adaptive VLC coding of coefficients*. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16, document JVT-C028), 3ème Meeting :Fairfax, USA, mai 2002.

[BLCZ06] Jiajun BU, Shuiyong LOU, Chun CHEN et Jingjing ZHU : A Predictive Block-Size Mode Selection for Inter Frame in H.264. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'2006*, Toulouse, France, mai 2006.

[Bou10] Fadi BOULOS : *Transmission d'images et de vidéos sur réseaux à pertes de paquets : mécanismes de protection et optimisation de la qualité perçue*. Thèse de doctorat, Université de Nantes, 2010.

[BPMBS98] J. BENOIS-PINEAU, F. MORIER, D. BARBA et H. SANSON : Hierarchical segmentation of video sequences for content manipulation and adaptive coding. *Signal Processing : special issue on Video sequence segmentation for content processing and manipulation*, 66:181 – 201, 1998.

[BPN02] Jenny BENOIS-PINEAU et Henri NICOLAS : A New Method for Region-Based Depth Ordering in a Video Sequence : Application to Frame Interpolation. *Journal of Visual Communication and Image Representation*, 13(3):363 – 385, septembre 2002.

[Bra03] A. P. BRADLEY : Can Region of Interest Coding Improve Overall Perceived Image Quality ? *In Proceedings of APRS Workshop on Digital Image Computing*, 2003.

[Bru01] Eric BRUNO : *De l'estimation locale à l'estimation globale de mouvement dans les séquences d'images*. Thèse de doctorat, Université Joseph Fourier, Grenoble, 2001.

[BS90] J. BRAUN et D. SAGI : Vision outside the focus of attention. *Perception and Psychophysics*, 48:45 – 58, 1990.

[BTH+00] D. BAVELIER, A. TOMANN, C. HUTTON, T. MITCHELL, G. LIU, D. CORINA et H. NEVILLE : Visual Attention to the Periphery Is Enhanced in Congenitally Deaf Individuals. *Journal of Neuroscience*, 20(17):1 – 6, 2000.

[Can03] R. CANOSA : *Seeing, Sensing, and Selection : Modeling Visual Perception in Complex Environments*. Thèse de doctorat, Rochester Institute of Technology, USA, 2003.

[Car98] Roger H. S CARPENTER : *Movements of the Eyes*. 2nd edition Pion, London, 1998.

[CAY03] Andy CHANG, Oscar C. AU et Y. M. YEUNG : A Novel Approach to Fast Multi-Frame Selection for H.264 Video Coding. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'2003*, Hong Kong, avril 2003.

[CB04] Renan COUDRAY et Bernard BESSERER : Global Motion Estimation for MPEG-Encoded Streams. *In Proccedings of the IEEE International Conference on Image Processing, ICIP 2004*, Singapore, Republic of Singapore, octobre 2004.

[CCLC04] Mei-Juan CHEN, Yi-Yen CHIANG, Hung-Ju LI et Ming-Chieh CHI : Efficient Multi-Frame Motion estimation Algorithms for MPEG-4 AVC/JVT/H.264. *In Proceedings of the 2004 International Symposium on Circuits and Systems*, mai 2004.

[Cha02] A. CHAUVIN : *Perception des scènes naturelles : étude et simulation du rôle de l'amplitude, de la phase et de la saillance dans la catégorisation et l'exploration des scènes naturelles*. Thèse de doctorat, Grenoble, Université Joseph Fourier, 2002.

[Cha03] Marc CHAUMONT : *Représentation en objets vidéo pour un codage vidéo progressif et concurrentiel des séquences d'images*. Thèse de doctorat, Institut de Formation Supérieur en Informatique et Communication (IFSIC), novembre 2003.

[CHC+05] Byeong-Doo CHOI, Min-Cheol HWANG, Jun-Ki CHO, Jin-Sam KIM, Jin-Hyung KIM et Sung-Jea KO : Realtime H.264 Encoding System Using Fast Motion Estimation and Mode Decision. *In* Berlin SPRINGER, éditeur : *Lecture notes in computer science ISSN 0302-9743*, pages 174–183. 2005.

[CHMP00] A. CHAUVIN, J. HÉRAULT, C. MARENDAZ et C. PEYRIN : Natural Scene Perception : Visual Attractors and Images Processing. *In 7th Neural Computation and Psychology Workshop*, 2000.

[CKS97] V. CASELLES, R. KIMMEL et G. SAPIRO : Geodesic Active Contours. *International Journal of Computer Vision*, 22:61 – 79, 1997.

[CLCB93] K. W. CHUN, K. W. LIM, H. D. CHO et Ra J. B. : An Adaptive Perceptual Quantization Algorithm for Video Coding. *IEEE Transactions on Consumer Electronics*, 39(3):555 – 558, 1993.

[CMB+05] Chun CHEN, Linjian MO, Jianjun BU, Shiyong LOU et Zhi YANG : A Novel Fast Predictive Mode Decision Algorithm For H.264. *In Proceedings of the IEEE International Symposium on Signal Processing and Its Applications, ISSPA'05*, Sydney, Australie, août 2005.

[Cou05] Renan COUDRAY : *Réutilisation des informations de compensation de mouvement d'un flux MPEG : évaluation qualitative et applications possibles*. Thèse de doctorat, Université de La Rochelle, spécialité informatique, novembre 2005.

[CPC04] Che-Yu CHANG, Chia-Ho PAN et Homer CHEN : Fast Mode Decision for P-Frames in H.264. *In Proceedings of the Picture Coding Symposium, PCS 2004*, San Francisco, USA, décembre 2004.

[CR87] P. B. CHOU et R. RAMAN : On Relaxation Algorithms Based on Markov Random Fields. Rapport technique, University of Rochester. Department of Computer Science ; Technical Report TR. 212, Rochester, New-York, 1987.

[CS98] Roberto CASTAGNO et Andrea SODOMACO : Estimation of Image Feature Reliability for an Interactive Videop Segmentation Scheme. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'1998*, volume 1, pages 938 – 942, Chicago, Illinois, USA, octobre 1998.

[ct06] SVT corporate TECHNOLOGY : The SVT High Definition Multi Format Test Set. Rapport technique, février 2006.

[CTT02] Hye-Yeon. CHEONG, Alexis Michael TOURAPIS et Pankaj TOPIWALA : *Fast Motion Estimation within the JVT codec*. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16, document JVT-E023), 5ème Meeting : Genève, Suisse, octobre 2002.

[CXH03] Zhibo CHEN, JianFeng XU et Yun HE : *Simplifications on Fast Motion Estimation*. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, document JVT-H026), 8ème meeting : Genèves, Suisse, mai 2003.

[CZH02] Zhibo CHEN, Peng ZHOU et Yuang HE : *Fast Integer Pel and Fractional Pel Motion Estimation for JVT*. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, document JVT-F017), 6ème meeting : Awaji, Island, JP, décembre 2002.

[CZH03] Zhibo CHEN, Peng ZHOU et Yuang HE : *Fast Motion Estimation for JVT*. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, document JVT-G016), 7ème meeting : Pattaya II, Thailande, mars 2003.

[Dal93] S. DALY : *Digital Images and Human Vision*, chapitre The Visible Differences Predictor : An Algorithm of Image Fidelity, pages 179 – 206. MIT Press, 1993.

[Dal98] Scott J DALY : Engineering Observations from Spatiovelocity and Spatiotemporal Visual Models. *In Proceedings of the IS&T/SPIE Conference on Human Vision and Electronic Imaging III*, volume 3299, pages 180 – 191, janvier 1998.

[DH04] Adriana DUMITRAS et Barry G. HASKELL : I/P/B Frame Type Decision by Collinearity of Displacements. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'2004*, Singapore, Republic of Singapore, octobre 2004.

[DI03] N. DHAVALE et L. ITTI : Saliency-based multi-foveated MPEG compression. *In Proceedings of the IEEE Seventh International Symposium on Signal Processing and its Applications*, 2003.

[DL58] H. DE LANGE : Research into the Dynamic Nature of the Human Fovea-Cortex Systems with Intermittent and Modulated Light. I. Attenuation Characteristics with White and Colored Light. *Journal of the Optical Society of America*, 48:777 – 784, 1958.

[DLY06] Jun-Ren DING, Ji-Kun LIN et Jar-Ferr YANG : Motion-based Adaptive GOP Algorithms for Efficient H.264/AVC Compression. *In Proceedings of the 9th Joint Conference on Information Sciences (JCIS)*, Kaohsiung, Taiwan, octobre 2006.

[DVCM+00] R. DE VALOIS, N.P. COTTARIS, L.E. MAHON, S.D. ELFAR et J.A. WILSON : Spatial and temporal receptive fields of geniculate and cortical cells and diretional selectivity. *Vision Research*, 20:3685 – 3702, 2000.

[DVDV92] R. DE VALOIS et K. K. DE VALOIS : A multi-stage color model. *Vision Research*, 33(8):1035 – 1065, 1992.

[DZLL00] Scott DALY, Wenjun ZENG, J. LI et Shawmin LEI : Visual masking in wavelet compression for JPEG2000. *In Proceedings of IS&T/SPIE Conference on Image and Video Communications and Processing*, volume 3974, San Jose, Californie, USA, janvier 2000.

[Fan08] Xiaole FANG : Importance Map Computation of HD Video For The Coding In H.264/AVC. Stage de master r2, systèmeselectroniques et génie electrique, Ecole polytechnique de l'université de Nantes, 2008.

[Fau76] O. D. FAUGERAS : *Digital Color Image Processing and Psychophysics within the Framework of a Human Visual Model*. Thèse de doctorat, University of Utah, 1976.

[FB94] John M. FOLEY et Geoffrey M. BOYNTON : A new model of human luminance pattern vision mechanisms : Analysis of the effects of pattern orientation, spatial phase and temporal frequency. *SPIE Human Vision and Electronic Imaging*, 2054:32 – 42, 1994.

[FBR05] S. FRINTROP, G. BACKER et E. ROME : Goal-Directed Search with a Top-Down Modulated Computational Attention System. *DAGM 2005, LNCS 3663. Springer*, pages 117 – 124, 2005.

[FCF90] P. FLANAGAN, P. CAVANAGH et O. E. FAVREAU : Independent orientation-selective mechanisms for cardinal directions of color space. *Vision Research*, 30(5):769 – 778, 1990.

[FG03] Markus FLIERL et Bernd GIROD : Generalized B Pictures and the Draft H.264/AVC Video Compression Standard. *IEEE Transactions on Circuits and Systems for Video technology*, 13:587 – 597, 2003.

[FH98] R.E. FREDERICKSEN et R.F. HESS : Estimating Multiple Temporal Mechanisms in Human Vision. *Vision Research*, 38(7):1023 – 1040, 1998.

[FJ90] David J. FLEET et Allan D. JEPSON : Computation of Component Image Velocity from. Local Phase Information. *International Journal of Computer Vision*, 5(1):77 – 104, 1990.

[fS91] International Organization for STANDARDIZATION : Visual Display Terminals (VDTs) used for office tasks - Ergonomic requirements - Part 3 : Visual displays, 1991.

[GG84] S. GEMAN et D. GEMAN : Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721 – 741, novembre 1984.

[GG92] Allen GERSHO et Robert M. GRAY : *Vector Quantization and Signal Compression*, volume 159. 1992.

[GGPL05] Mickael GUIRONNET, Nathalie GUYADER, Denis PELLERIN et Patricia LADRET : Spatio-temporal attention model for video content analysis. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2005*, Genoa, Italy, septembre 2005.

[GH03] Xiaodong GU et Zhang HONGJIANG : Implementing Dynamic GOP in Video Encoding. *In Proceedings of the International Conference on Multimedia and Expo, ICME '03*, volume 1, pages 349 – 352, Baltimore, MD, USA, juillet 2003.

[Gir89] B. GIROD : The Information Theoretical Significance of Spatial and Temporal Masking in Video Signals. *SPIE Human Vision and Electronic Imaging*, 1077:178 – 187, 1989.

[GK00] S. GEPSHTEIN et M. KUBOVY : The emergence of visual objects in space-time. *Proceedings of the National Academy of Sciences*, 97(14):8186 – 8191, 200.

[Gol66] Solomon W. GOLOMB : Run Length Coding. *IEEE Transactions on Information Theory*, IT-12:399 – 401, 1966.

[GP02] Christine GUILLEMOT et Stéphane PATEUX : Eléments de théorie de l'information et de la communication. *In Compression et codage des images et des vidéos*, pages 21 – 43. Hermès Science Publications, 2002.

[GT02] I. GRINIAS et G TZIRITAS : Robust pan, tilt and zoom estimation. *In Proceedings of the 14th IEEE International Conference on Digital Signal Processing*, volume 2, pages 679 – 682, Santorini, Grèce, juillet 2002.

[HCB06] P.R. HILL, T.K. CHIEW et D.R. BULL : Interpolation Free Sub-Pixel Motion Estimation for H.264. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2006*, Atlanta, GA, USA, octobre 2006.

[HCK+05] Min-Cheol HWANG, Jun-Ki CHO, Jin-Sam KIM, Jin-Hyung KIM et Sung-Jea KO : Fast Intra Prediction Mode Selection Scheme Using Temporal Correlation in H.264. *In Proceedings of the the IEEE Tencon 2005*, Melbourne, Victoria, Australie, novembre 2005.

[Hee93] D.J. HEEGER : Modeling Simple-Cell Direction Selectivity With Normalized,. Half-Squared, Linear Operators. *Journal of Neurophysiology*, 70(5), 1993.

[HHW+03] Yu-Wen HUANG, Bing-Yu HSIEH, Tu-Chih WANG, Shao-Yi CHIEN, Shyh-Yih MA, Chun-Fu SHEN et Liang-Gee CHEN : Analysis and Reduction of Reference Frames for Motion Estimation in MPEG-4/AVC/JVT/H.264. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'2003*, volume 3, pages 145 – 148, Hong Kong, avril 2003.

[HK99] Joerg HEUER et André KAUP : Global Motion Estimation in Image Sequences Using Robust Motion Vector Field Segmentation. *In Proceedings of the seventh ACM international conference on Multimedia*, pages 261 – 264, Orlando, Floride, USA, 30 octobre - 5 novembre 1999.

[HLC+06] Win-Bin HUANG, Yi-Li LIN, Hung-Wei CHENG, Alvin W.Y. SU et Yau-Hwang KUO : Two-Stage Mode Selection of H.264/AVC Video Encoding with Rate Distortion Optimization. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'2006*, Toulouse, France, mai 2006.

[HS81] B.K.P. HORN et B.G. SCHUNCK : Determining Optical Flow. *Artificial Intelligence*, 17(1 – 3):185 – 203, août 1981.

[Huf52] David A. HUFFMAN : A Method for the Construction of Minimum-Redundancy Codes. *In Proceedings of the I.R.E.*, pages 1098 – 1101, septembre 1952.

[Hug05] L. HUGUENEL : *Codage par zones d'intérêt dans le cadre d'un codeur MPEG4 AVC*. Diplôme de Recherche Technologique, 2005.

[HWH99] John M. HENDERSON, Phillip A. WEEKS et Andrew HOLLINGWORTH : The Effects of Semantic Consistency on Eye Movements During Complex Scene Viewing. *Journal of Experimental Psychology : Human Perception and Performance*, 25(1):210 – 228, 1999.

[HZL07] Chiuan HWANG, ShinShan ZHUANG et Shang-Hong LAI : Efficient Intra Mode Selection using Image Structure Tensor for H.264/AVC. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2007*, San antonio, Texas, USA, septembre 2007.

[IK00] L. ITTI et C. KOCH : A Saliency-Based Search Mechanism for Overt and Covert Shifts of Visual Attention. *Vision Research*, 40:1489 – 506, 2000.

[IK01] L. ITTI et C. KOCH : Computational modelling of visual attention. *Nature Review | Neuroscience*, 2(3):194 – 203, 2001.

[IKB00] L. ITTI, C. KOCH et J. BRAUN : Revisiting Spatial Vision : Toward a Unifying Model. *Journal of the Optical Society of America*, 17(11):1899 – 1917, 2000.

[IKN98] L. ITTI, C. KOCH et E. NIEBUR : A Model of Saliency-based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254 – 1259, 1998.

[IRI] IRISA : Motion2D. Rapport technique. http ://www.irisa.fr/Vista/Motion2D/.

[Ish67] Shinobu ISHIHARA : *Tests for Colour-Blindness*. Kanehara Shuppen Company, Ltd., Tokyo, Japan, 1967.

[ISO03] ISO/IEC 14496-10 and ITU-T Recommendation H.264 - Advanced video coding for generic audiovisual services, 2003.

[Itt61] Johannes ITTEN : *The Elements of Color*. New York USA : John Wiley & Sons Inc, 1961.

[ITU98] ITU : *Subjective assessment methods for image quality in high-definition television.* Recommendation ITU-R BT.710-4, International Telecommunication Union, 1998.

[ITU04] ITU : Methodology for the subjective assessment of the quality of television pictures. Rapport technique, Recommendation ITU-R BT.500-11, International Telecommunication Union, 2004.

[JBI] ITU-T Recommendation, Information technology – Coded representation of picture and audio information – Progressive bi-level image compression, T.82 (JBIG).

[JJ81] Jaswant R. JAIN et Anil K. JAIN : Displacement Measurement and Its Application in Interframe Image Coding. *IEEE Transactions on Communications*, COM-29:1799 – 1806, décembre 1981.

[JZB06] Liangbao JIAO, De ZHANG et Houjie BI : Inherit-Based Adaptive Frame Selection for Fast Multi-frame Motion Estimation in H.264. *In* Berlin SPRINGER, éditeur : *Lecture notes in control and information sciences ISSN 0170-8643*, volume 345, pages 938–944. 2006.

[KAC07] N. KAMNOONWATANA, D. AGRAFIOTIS et C. N. CANAGARAJH : Fast Mode Decision for H.264/AVC based on Clustering of MPEG-7 Texture Descriptor Values. *In Proceedings of the Picture Coding Symposium, PCS 2007*, Lisbonne, Portugal, novembre 2007.

[KAC08] N. KAMNOONWATANA, D. AGRAFIOTIS et C. N. CANAGARAJH : Exploiting MPEG-7 Texture Descriptors for Fast H.264 Mode Decision. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'2008*, San Diego, CA, USA, octobre 2008.

[Kan02] K. KANATANI : Motion segmentation by subspace separation : Model selection and reliability evaluation. *International Journal of Image and Graphics*, 2(2):179 – 197, 2002.

[KFN02] Helga KOLB, Eduardo FERNANDEZ et Ralph & NELSON : Webvision - The organization of the retina and visual system, 2002.

[KGV83] S. KIRKPATRICK, C. D. GELLAT et M. P. VECHI : Optimisation by Simulated Annealing. *Science*, 220:671 – 680, 1983.

[KIH+81] T. KOGA, K. IINUMA, A. HIRANO, Y. LIJIMA et T. ISHIGURO : Motion Compensated Interframe Coding for Video Conferencing. *In Proceedings of the National Telecommunications Conference*, pages G5.3.1 – G5.3.5, New Orleans, LA, novembre 1981.

[KIK85] M. KUNT, A. IKONOMOPOULOS et M. KOCHER : Second-Generation Image-Coding Techniques. *Proceedings of the IEEE*, 73:549 – 574, 1985.

[Kim07] Cho Chang-Sik KIM, Byung-Gyu and : A Fast Inter-Mode Decision Algorithm based on Macro-Block Tracking for P Slices inthe H.264/AVC Video Standard. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2007*, San Antonio, Texas, USA, septembre 2007.

[KKKH05] Geun-Yong KIM, Seung-Hwan KIM, Hee-Soon KUIM et Yo-Sung HO : Fast Mode Decision Algorithm for H.264 based on Motion Cost. *In Proceedings of the European Signal Processing Conference, EUSIPCO 2005*, Antalya, Turquie, septembre 2005.

[KMI99] Raymond M. KLEIN et W. Joseph MAC INNES : Inhibition of Return Is a Foraging Facilitator in Visual Search. *Psychological Science*, 10(4):346 – 352, juillet 1999.

[KMJ06] Changsung KIM, Siwei MA et Kuo C.-C. JAY : Fast H.264 Motion Estimation with Block-Size Adapative Referencing (BAR). *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2006*, Atlanta, GA, USA, octobre 2006.

[KS07] Spyridon K. KAPOTAS et Athanassios N. SKODRAS : Fast multiple Reference Frame Selection Method in H.264 Video Encoding. *In Proceedings of the Picture Coding Symposium, PCS 2007*, Lisbonne, Portugal, novembre 2007.

[KSC06] Byung-Gyu KIM, Suk-Kyu SONG et Chang-Sik CHO : Efficient Inter-Mode Decision Based on Contextual Prediction for the P-Slice in H.264/AVC Video Coding. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2006*, Atlanta, GA, USA, octobre 2006.

[KU85] C. KOCH et S. ULLMAN : Shifts in selective visual attention : towards the underlying neural circuitry. *Human Neurobiology*, 4(4):219 – 227, 1985.

[KW82] J. KRAUSKOPF et D. W. WILLIAMS, D. R.and Heeley : Cardinal direction of color space. *Vision Research*, 22:1123 – 1131, 1982.

[KWT88] M. KASS, A. WITKIN et D. TERZOPOULOS : Snakes : Active contour models. *International Journal of Computer Vision*, 1:321 – 332, 1988.

[Lal90] Patrick LALANDE : *Détection du mouvement apparent dans une séquence d'images selon une approche markovienne ; Application à la robotique sous-marine*. Thèse de doctorat, Université de Rennes I, Mention : Traitement du Signal et Télécommunications, mars 1990.

[Lam96] C J. Van Den Branden LAMBRECHT : *Perceptual Models And Architectures For Video Coding Applications*. Thèse de doctorat, Ecolec Polytechnique Fédérale de Lausanne, EPFL, 1996.

[LC01] P. LE CALLET : *Critères objectifs avec référence de qualité visuelle des images couleur*. Thèse de doctorat, Université de Nantes, Ecole polytechnique de l'université de Nantes, 2001.

[LD94] Jungwoo LEE et Bradley W. DICKINSON : Temporally Adaptive Motion Interpolation Exploiting Temporal Masking in Visual Perception. *IEEE Transactions on Image Processing*, 3(5):513 – 526, septembre 1994.

[LEC07] Byeongdu LA, Minyoung EOM et Yoonsik CHOE : Fast Mode Decision for Intra Prediction in H.264/AVC Encoder. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2007*, San Antonio, Texas, USA, septembre 2007.

[LF96] Lurng-Kuo LIU et Ephraim FEIG : A Block-Based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(4):419 – 422, août 1996.

[LG91] Didier LE GALL : MPEG : A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4), avril 1991.

[LH97] Austin Y. LAN et Jenq-Neng HWANG : Scene Context Dependent Reference Frame Placement for MPEG Video Coding. *In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, ICASSP 97*, volume 4, pages 2997 – 3000, Munich , Allemagne, avril 1997.

[LHZ+06] Qiong LIU, Rui-min HU, Li ZHU, Xin-chen ZHANG et Zhenyu HAN : Improved Fast Intra Prediction Algorithm of H.264/AVC. *Journal of Zhejiang University Science A*, pages 101 – 105, 2006.

[LJL+03] Peter LIST, Anthony JOCH, Jani LAINEMA, Gisle BJØNTEGAARD et Marta KARCZEWICZ : Adaptive deblocking filter. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:614 – 619, juillet 2003.

[LL06] Pei-Jun LEE et Ming-Long LIN : Fast Inter Mode Selection Algorithm for Motion Estimation in MPEG-4 AVC/JVT/H.264. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2006*, Atlanta, GA, USA, octobre 2006.

[LL07] Yu-Ming LEE et Yinyi LIN : An Improved Zero-Block Mode Decision Algorithm for H.264/AVC. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'2007*, San Antonio, Texas, USA, septembre 2007.

[LLC04] Xiang LI, Eric Q. LI et Yen-Kuang CHEN : Fast Multi-Frame Motion Estimation Algorithm With Adaptive search Strategies In H.264. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'2004*, volume 3, pages 369 – 372, Montreal, Quebec, Canada, mai 2004.

[LM05] O. LE MEUR : *Attention sélective en visualisation d'images fixes et animées affichées sur écran : Modèles et évaluaton de performances – Applications*. Thèse de doctorat, Ecole polytechnique de l'université de Nantes, 2005.

[LMLCB07] Olivier LE MEUR, Patrick LE CALLET et Dominique BARBA : Predicting visual fixations on video based on low-level visual features. *Vision Research*, vol 47(19):2483 – 2498, septembre 2007.

[LMLCBT06] Olivier LE MEUR, Patrick LE CALLET, Dominique BARBA et Dominique THOREAU : A Coherent Computational Approach to Model Bottom-Up Visual Attention. *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(5):802 – 817, mai 2006.

[LS00] J. LUO et A. SINGHAL : On Measuring Low-Level Saliency in Photographic Images. *In Proceedings of the IEEE Conference on Computer Vision and Patten Recognition*, 2000.

[LSIG06] Zhenyu LIU, Yang SONG, Takeshi IKENAGA et Satoshi GOTO : Low-Pass Filter Based VLSI Oriented Variable Block Size Motion Estimation Algorithm for H.264. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'2006*, Toulouse, France, mai 2006.

[LWW+03] K. P. LIM, S. WU, D. J. WU, S. RAHAEDJA, X. LIN, F. PAN et Z. G. LI : *Fast INTER Mode Selection*. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, document JVT-I020, 9ème Meeting : San Diego, USA, septembre 2003.

[Mah96] Frank H. MAHNKE : *Color, Environment, & Human Response*. Detroit : Van Nostrand Reinhold, 1996.

[Mal89] Stéphane G. MALLAT : A Theory for Multiresolution Signal Decomposition : The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, juillet 1989.

[Mal90] Stéphane G. MALLAT : *A Wavelet Tour of Signal Processing*. Academic Press, 1990.

[Mar10] Sophie MARAT : *Modèles de saillance visuelle par fusion d'informations sur la luminance, le mouvement et les visages pour la prédiction de mouvements oculaires lors de l'exploration de viidéos*. Thèse de doctorat, Université de Grenoble, 2010.

[MB90] F. MEYER et S. BEUCHER : Morphological segmentation. *Journal of visual communication image representation*, 1(1):21 – 46, 1990.

[MBP92] R. MILANESE, J.M. BOST et T. PUN : A Bottom-Up Attention System for Active Vision. *In Proccedings of the 10th European Conference on Artificial Intelligence, ECAI92*, pages 808 – 810, 1992.

[MBW01] Detlev MARPE, Gabi BLÄTTERMANN et Thomas WIEGAND : *Adaptive Codes for H.26L*. ITU-T Study Group 16/Question 6, Document VCEG-L13, Eibsee, Allemagne, janvier 2001.

[MD02] Remi MEGRET et Daniel DEMENTHON : A Survey of Spatio-Temporal Grouping Techniques. Rapport technique LAMP-TR-094, CS-TR-4403, UMIACS-TR-2002-83, CAR-TR-979, University of Maryland, College Park, University of Maryland, USA, octobre 2002.

[MDDV+96] X. MARICHAL, T. DELMOT, C. DE VLEESCHOUWER, V. WARSCOTTE et B. MACQ : Automatic Detection Of Interest Areas of an Image or of a Sequence of Images. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 1996*, 1996.

[MEdlPADdM07] E. MARTINEZ-ENRIQUEZ, M. deFrutos LOPEZ, J. C. PUJOL-ALCOLADO et F. Diaz-de MARIA : A Fast Motion-Cost Based Algorithm for H.264/AVC Inter Mode Decision. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2007*, San Antonio, Texas, USA, septembre 2007.

[Mey98] F. MEYER : From Connected Operators to Levelings. *In Proceedings of the fourth international symposium on Mathematical morphology and its applications to image and signal processing*, 1998.

[MGP09] S. MARAT, N. GUYADER et D. PELLERIN : *Gaze prediction improvement by adding a face feature to a saliency model, In-Tech book : Recent advances in Signal Processing*, chapitre 12, pages 195 – 210. Numéro ISBN 978-953-307-002-5. novembre 2009.

[MHKK03] Henrique S. MALVAR, Antti HALLAPURO, Marta. KARCZEWICZ et Louis KEROFSKY : Low-Complexity Transform and Quantization in H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:598 – 603, juillet 2003.

[MHPG+09] S. MARAT, T. HO PHUOC, L. GRANJON, N. GUYADER, D. PELLERIN et A. GUÉRIN-DUGUÉ : Modelling spatio-temporal saliency to predict gaze direction for short videos. *International Journal of Computer Vision (IJCV)*, 82(3):231 – 243, 2009.

[Mil93] R. MILANESE : *Detecting Salient Regions in an Image : From Biological Evidence to Computer Implementation*. Thèse de doctorat, Université de Genèves, Suisse, 1993.

[MJ02] R. MEGRET et J.-M. JOLION : Grey-level blobs tracking for video dynamic content representation. *Reconnaissance de Formes et Intelligence Artifcielle*, 2:397 – 406, 2002.

[MKK05] Yong Ho MOON, Gyu Yeong KIM et Jae Homer KIM : An Improved Early Detection algorithl for All-Zeo Blocks in H.264 Video Encoding. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(8):1053 – 1057, août 2005.

[ML98] Ferran MARQUÈS et Joan LLACH : Tracking of Generic Objects for Video Object Generation. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 1998*, Chicago, USA, octobre 1998.

[MOK00] Abdol-Reza MANSOURI, Antoine OLIVIER et Janusz KONRAD : Topology-Independent Region Tracking with Level Sets. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'2000*, Vancouver, Canada, septembre 2000.

[Mon75] Ferdinand MONOYER : Échelle typographique pour la détermination de l'acuité visuelle. *Comptes rendus 113, Académie des Sciences*, 1875.

[MPE01] 14496-2, Amendment 1, Information technology – Coding of audio-visual objects - Part 2 : Visual, 2001.

[MPL00] G. MARQUANT, S. PATEUX et C. LABIT : Mesh and "Crack Lines" : Application to Object-based Motion Estimation and Higher Scalability. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2000*, volume 2, pages 554–557, Vancouver, BC, Canada, septembre 2000.

[MQ03] Kai-Kuang MA et Gang QIU : Unequal-Arm Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation in the JVT/H.26L. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'2003*, Barcelone, Espagne, septembre 2003.

[MR97] S.K. MANNAN et D.S RUDDOCK, K.H.and Wooding : Fixation sequences made during visual examination of briefly presented 2D images. *Spatial Vision*, 11(2):157 – 178, 1997.

[MRL03] L. MUIR, I. RICHARDSON et S. LEAPER : Gaze Tracking and Its Application to Video Coding for Sign Language. *In Proceedings of the Picture Coding Symposium, PCS 2003*, 2003.

[MS74] J. L. MANNOS et D. J. SAKRISON : The Effects of a Visual Fidelity Criterion on the Encoding of Images. *IEEE Transactions of Information Theory*, 20(4):525 – 535, 1974.

[MSW03] Detlev MARPE, Heiko SCHWARZ et Thomas WIEGAND : Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):620 – 636, juillet 2003. Publication Award of ITG.

[MYKP06] Zhi-Yi MAI, Chun-Ling YANG, Kai-Zhi KUANG et Lai-Man PO : A Novel Motion Estimation Method Based on Structural Similarity for H.264 Inter Prediction. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'2006*, Toulouse, France, mai 2006.

[Nei67] Ulric NEISSER : *Cognitive Psychology*. Appleton-Century-Crofts New York, 1967.

[Ngu95] E. NGUYEN : *Compression sélective et focalisation visuelle : application au codage hybride de séquences d'images*. Thèse de doctorat, Université de Rennes 1, 1995.

[NI06] V. NAVALPAKKAM et L. ITTI : Top-down Attention Selection is Fine-grained. *Journal of Vision*, 6:1180 – 1193, octobre 2006.

[NNJ43] S. M. NEWHALL, D. NICKERSON et D. B. JUDD : Final Report of the O.S.A. Subcommittee on the Spacing of the Munsell Colors. *Journal of the Optical Society of America*, 33(7):385 – 418, 1943.

[NSJ06] Marcos NIETO, Luis SALGADO et Cabrera JULIÁN : Fast Mode Decision on H.264/AVC Main Profile Encoding Based on PSNR Predictions. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2006*, Atlanta, GA, USA, octobre 2006.

[OB95] J.M. ODOBEZ et P. BOUTHEMY : Robust Multiresolution of parametric Motion Models. *Journal of Visual Communication and Image Representation*, 6(4):348 – 365, décembre 1995.

[OHB97] Wilfried OSBERGER, Sean HAMMOND et Neil BERGMANN : An MPEG Encoder Incorporating Perceptually Based Quantisation. *In Proceedings of the IEEE TENCON - Speech and Image Technologies for Computing and Telecommunications*, pages 731 – 734, Brisbane, Australia, décembre 1997.

[OLL+03] E. ONG, W. LIN, Z. LU, X. YANG, S. YAO, X. LIN et F. MOSCHETTI : Quality evaluation of MPEG-4 and H.26L coded video for mobile multimedia communications. *In Proceedings of the International Symposium on Signal Processing and its Applications*, 2003.

[OM98] Wilfried OSBERGER et Anthony J. MAEDER : Automatic Identification of Perceptually Important Regions in an Image. *In Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 701 – 704, Brisbane, Australia, août 1998.

[OMB98] Wilfried OSBERGER, Anthony J. MAEDER et Neil BERGMANN : A Perceptually Based Quantization Technique for MPEG Encoding. *In Proceedings of the IS&T/SPIE Conference on Human Vision and Electronic Imaging III*, volume 3299, pages 148 – 159, janvier 1998.

[OTCH03] A. OLIVA, A. TORRALBA, M. S. CASTELHANO et J. M. HENDERSON : Top-Down Control of Visual Attention in Object Detection. *In Proceedings of the IEEE Inernational Conference on Image Processing, ICIP 2003*, 2003.

[Par02] D. J. PARKHURST : *Selective attention in natural vision : Using computational models to quantify stimulus-driven attentional allocation*. Thèse de doctorat, The Johns Hopkins University, Baltimore, Maryland, USA, 2002.

[PAYG93] E. PELI, L. E. AREND, G. M. YOUNG et R. B. GOLDSTEIN : Contrast sensitivity to patch stimuli : Effects of spatial bandwith and temporal presentation. *Spatial Vision*, 7(1):1 – 14, 1993.

[PC84] M. I. POSNER et Y. COHEN : *Components of Visual Orienting*. Attention and Performance X. London : Lawrence Erlbaum, 1984.

[PLR+05] Feng PAN, Xiao LIN, Susanto RAHARDJA, Keng Pang LIM, Z. G. LI, Dajun WU et Siwei WU : Fast Mode Decision Algorithm for Intraprediction in H.264/AVC Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(7):813 – 822, juillet 2005.

[PM96] Lai-Man PO et Wing-Chung MA : A Novel Four-Step Search Algorithm for Fast Block Motion Estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3):313 – 317, juin 1996.

[PM05] Branko PETLJANSKI et Oge MARQUES : A Novel Approach for Video Quantization Using the Spatiotemporal Frequency Characteristics of the Human Visual System. *In Proceedings of the British Machine Vision Conference (BMVC'2005)*, Oxford, United Kingdom, 5-8 septembre 2005.

[PN04] Derrick J. PARKHURST et Ernst NIEBUR : Texture contrast attracts overt visual attention in natrual scenes. *European Journal of Neuroscience*, 19:783 – 789, 2004.

[Pos80] M.I. POSNER : Orienting of attention. *Quaterly Journal of Experimental Psychology*, 32:3 – 25, 1980.

[PS99] D.G. PEREZ et C. SUN, M.-T.and Gu : Semantic Video Object Extraction Based on Backward Tracking of Multivalued Watershed. *In Proceedings of the International Conference on Image Processing, ICIP'1999*, volume 2, pages 145 – 149, 1999.

[PS00] C. M. PRIVITERA et L. STARK : Algorithms for Defining Visual Regions-of-Interest : Comparison with Eye Fixations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:970 – 982, 2000.

[PT07] Yu-Nan PAN et Tsung-Han TSAI : Fast Motion Estimation and Edge Information Inter-Mode Decision on H.264 Video Coding. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'2007*, San Antonio, USA, septembre 2007.

[PW04] Fatih PORIKLI et Yao WANG : Automatic Video Object Segmentation Using Volume Growing and Hierarchical Clustering. *EURASIP Journal on Applied Signal Processing*, 3:442 – 453, mars 2004.

[Ric03] Iain E. G. RICHARDSON : *H.264 and MPEG-4 Video Compression : Video Coding for Next-Generation Multimedia*. Ltd. ISBN : 0-470-84837-5. Chippenham, septembre 2003.

[RY90] K. R. RAO et P. YIP : *Discrete Cosine Transform : Algorithms, Advantages, Applications*. Academic Press, Boston, 1990.

[RZ99] Pamela REINAGEL et Anthony M. ZADOR : Natural Scene Statistics at the Center of Gaze. *Network : Computation in Neural Systems*, 10:341 – 350, 1999.

[Sal99] D. D. SALVUCCI : *Mapping Eye Movements to Cognitive Processes*. Thèse de doctorat, Carnegie Mellon University, 1999.

[SB59] A.J. SEYLER et Z.L. BUDRIKIS : Measurements of Temporal Adaptation to Spatial Detail Vision. *Nature Revue Neuroscience*, 184:1215 – 1217, 1959.

[SB65] A.J. SEYLER et Z.L. BUDRIKIS : Detail perception after scene changes in television image presentations. *IEEE Transactions on Information Theory*, 11(1):31 – 43, 1965.

[SLZ06] Rui SU, Guizhong LIU et Tongyu ZHANG : Fast Mode Decision Algorithm for Intra Prediction in H.264/AVC. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'2006*, Toulouse, France, mai 2006.

[Sén96] H. SÉNANE : *Représentation d'images en sous-bandes visuelles. Application au codage d'images de télévision sans défaut visuel*. Thèse de doctorat, Université de Nantes, IRESTE, 1996.

[SN06] Luis SALGADO et Marcos NIETO : Sequence Independent very Fast Mode Decision Algorithm on H.264/AVC Baseline Profile. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2006*, Atlanta, GA, USA, octobre 2006.

[SPH98] Adrian SPINEI, Denis PELLERIN et Jeanny HÉRAULT : Spatiotemporal energy-based method for velocity estimation. *Signal Processing*, 65(6):347 – 362, 1998.

[SS04] Yeping SU et Ming-Ting SUN : Fast Multiple Reference Frame Motion Estimation for H.264. *In Proceedings of the IEEE International Conference on Multimedia and Expo, ICME 2004*, pages 695–698, Taipei, Taiwan, juin 2004.

[SS06] Yeping SU et Ming-Ting SUN : Fast Multiple Reference Frame Motion Estimation H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(3):447 – 452, mars 2006.

[SVT02] SVT : Overall-quality assessment when targeting Wide-XGA flat panel displays. Rapport technique, SVT corporate development technology, 2002. ftp ://ftp.ldv.e-technik.tu-muenchen.de/pub/test_sequences/.

[TAL01] Alexis Michael TOURAPIS, Oscar C. AU et Ming Lei LIOU : Predictive Motion Vector Field Adaptive Search Technique (PMVFAST) - Enhancing Block Based Motion Estimation. *In Proceedings of Visual Communications and Image Processing 2001 (VCIP-2001)*, pages 883–892, San Jose, CA, USA, janvier 2001.

[Tam95] W.J. TAM : Visual Masking at Video Scene Cuts. *SPIE Human Vision and Electronic Imaging*, 2411:111 – 119, 1995.

[TCT05] Alexis Michael TOURAPIS, Hye-Yeon CHEONG et Pankaj TOPIWALA : *Fast ME in the JM reference software*. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, document JVT-P026), 16ème Meeting : Poznan, Pologne, juillet 2005.

[TCW+95] J. K. TSOTSOS, S. M. CULHANE, W. Y. K. WAI, Y. H. LAI, N. DAVIS et F. NUFLO : Modeling Visual Attention via Selective Tuning. *Artificial Intelligence*, 78:507 – 545, 1995.

[TDV05] T. TAKEUCHI et K. DE VALOIS : Sharpening image motion based on the spatio-temporal characteristics of human vision. *In Proceedings of the SPIE Conference Human Vision and Electronic Imaging X, 5666,*, pages 83 – 94, 2005.

[TG80] A. M. TREISMAN et G. GELADE : A feature-integration theory of attention. *Cognitive Psychology*, 12:97 – 136, 1980.

[TM01] D. TAUBMAN et M. MARCELLIN : *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2001.

[TNHT05] Takeshi TSUKUBAT, Isao NAGAYOSHI, Tsuyoshi HANAMURAT et Hideyoshi TOMINAGA : H.264 Fast Intra-Prediction Mode Decision based on Frequency Characteristic. *In Proceedings of the 13th European Signal Processing Conference, EUSIPCO 2005*, Antalya, Turquie, septembre 2005.

[TP06] Tsung-Han TSAI et Yu-Nan PAN : A Novel 3-D Predict Hexagon search Algorithm for Fast Block Motion Estimation on H.264 Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(12):1542 –1549, Décembre 2006.

[TPBF87] D. TERZOPOULOS, J. PLATT, A. BARR et K. FLEISCHER : Elastically Deformable Models. *Computer Graphics*, 21(4):205 – 214, 1987.

[TPC03] Chi-Wang TING, Lai-Man PO et Chun-Ho CHEUNG : Center-Biased Frame Selection Algorithms for Fast Multi-Frame Motion Estimation in H.264. *In Proceedings of the IEEE International Conference on Neural Networks and Signal Processing*, pages 1258 – 1261, Nanjing, China, Décembre 2003.

[TT05] Alexis TOURAPIS et Pankaj TOPIWALA : *Fast Subpixel Motion Estimation Support for the Enhanced Predictive Zonal Search Scheme*. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, document JVT-Q079), 17ème Meeting : Nice, France, octobre 2005.

[TZ98] P. H. S. TORR et A. ZISSERMAN : Concerning bayesian motion segmentation, model averaging, matching and the trifocal tensor. *European Conference on Computer Vision*, 1:551 – 527, 1998.

[Vid06] VIDEOLAN : x264 - a free H264/AVC encoder, snapshot-20070626-2245, 2006. ftp://ftp.videolan.org/pub/videolan/x264/snapshots/.

[VP89] Alessandro VERRI et Tomaso POGGIO : Motion Field and Optical Flow : Qualitative Properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):490 – 498, mai 1989.

[VQE07] VQEG : *Multimedia Test Plan 1.19*. VQEG, 2007.

[WA94] John Y. A. WANG et Edward H. ADELSON : Representing Moving Images with Layers. *IEEE Transactions on Image Processing*, 3(5):625 – 638, septembre 1994.

[Wat87] A. B. WATSON : The Cortex Transform : Rapid Computation of Simulated Neural Images. *Computer Vision, Graphics and Image Processing*, 39:311 – 327, 1987.

[Wat92] S. WATANABE, H. andSinbghal : Bit allocation and rate control based on human visual sensitivity for interframe coders. *In Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP 92*, volume 3, pages 521 – 524, 23-26 mars 1992.

[Wat93] A. B. WATSON : DCTune : A technique for visual optimization of DCT quantization matrices for individual images. *Society for Information Display Digest of Technical. Papers XXIV*, pages 946 – 949, 1993.

[Wat94] A. B. WATSON : Perceptual optimization of DCT color quantizations matrices. *In Proceedings of the International Conference on Image Processing, ICIP 1994*, 1994.

[Wat98] A.B. WATSON : Toward a perceptual video quality metric. *SPIE Human Vision and Electronic Imaging*, 3299:946 – 949, 1998.

[WBPB96] L. WU, J. BENOIS-PINEAU et D. BARBA : Spatio-temporal segmentation of image sequences for object oriented low bit-rate image coding. *Signal Processing : Image Communication, a EURASIP journal*, 8(6):513 – 544, 1996.

[WBSS04] Z WANG, A. C. BOVICK, H. R. SHEIKH et E. P. SIMONCELLI : Image Quality Assessment : From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600 – 612, avril 2004.

[WCF89] Jeremy M. WOLFE, Kyle R. CAVE et Susan L. FRANZEL : Guided Search : An Alternative to the Feature Integration Model for Visual Search. *Journal of experimental psychology. Human perception and performance*, 15(3):419 – 433, 1989.

[WDVD00] Y. WANG, J. F. DOHERTY et R. E. VAN DYCK : Moving Object Tracking in Video. *In Proceedings of the IEEE Applied Imagery Pattern Recognition Workshop*, Washington DC, USA, octobre 2000.

[WDVS90] M. A. WEBSTER, K. K. DE VALOIS et E. SWITKES : Orientation and spatialfrequency discrimination for luminance and chromatic gratings. *Journal of the Optical Society of America*, 7(6):1034 – 1049, 1990.

[Wed03] T. WEDI : Motion Compensation in H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:577 – 586, juillet 2003.

[WH04] J.M. WOLFE et T.S. HOROWITZ : What attributes guide the deployment of visual attention and how do they do it ? *Nature Reviews Neuroscience*, 5:1 – 7, 2004.

[Win99] S. WINKLER : A Perceptual Distortion metric for Digital Color Video. *In Human Vision and Electronic Imaging IV*, 3644:175 – 184, 1999.

[WJP93] A. A. WEBSTER, C. T. JONES et M. H. PINSON : An Objective Video Quality Assessment System Based on Human Perception. *In Human Vision, Visual Processing and Digital Display IV*, 1913:15 – 26, 1993.

[WKK06] Hanli WANG, Sam KWONG et Chi-Wah KOK : Effectively Detecting All-Zero DCT Blocks for H.264 Optimization. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'2006*, Atlanta, GA, USA, octobre 2006.

[WLC+07] D. WU, K.P. LIM, T.K. CHIEW, J.Y. THAM et K.H. GOH : An Adaptive Thresholding Technique for the Detection of All-Zeros Blocks in H.264. *In Proceedings of the IEEE*

International Conference on Image Processing, ICIP'2007, San Antonio, Texas, USA, septembre 2007.

[WNC87] I. WITTEN, R. NEAL et J. CLEARY : Arithmetic coding for data compression. *In Communications of the ACM*, volume 30, juin 1987.

[WSBL03] Thomas WIEGAND, Gary SULLIVAN, Gisle BJONTEGAARD et Ajay LUTHRA : Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560 – 576, juillet 2003.

[WTST06] Chou-Chen WANG, Chen TSUNG-SHIEN et Chi-Wei TUNG : Fast Intra Mode Decision in H.264 using Interblock Correlation. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2006*, Atlanta, GA, USA, octobre 2006.

[WWLS06] Yu-Lin WANG, Jing-Xin WANG, Yen-Wen LAI et Alvin W. Y. SU : Dynamic GOP Structure Determination for Real-Time MPEG-4 Advanced Simple Profile Video Encoder. *In Proceedings of the IEEE International Conference on Multimedia and Expo, ICME'06*, Amsterdam, Hollande, juillet 2006.

[WZG99] Thomas WIEGAND, Xiaozheng ZHANG et Bernd GIROD : Long-Term Memory Motion-Compensated Prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(Issue 1):70 – 84, février 1999.

[XHFH06] Ying-lai XI, Chong-Yang HAO, Yang-Yu FAN et Hong-Qi HU : A Fast Block-Matching Algorithm based on Adaptive Search Area and its VLSI Architecture for H.264/AVC. *Signal Processing : Image Communication*, 21(8):626 – 646, 2006.

[Xin06] Anthony XIN, Jun.and Vetro : Fast Mode Decision for Intra-only H.264/AVC Coding. *In Proceedings of the Picture Coding Symposium, PCS 2006*, Pékin, Chine, avril 2006.

[XYH03] JianFeng XU, Ping YANG et Yun HE : *Modification of Fast Motion Estimation*. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, document JVT-J027, 10ème meeting : Waikoloa, HI, USA, décembre 2003.

[Yar67] A. YARBUS : *Eye Movements and Vision*. Trans., New-York :Plenum Press, 1967.

[YCTB03] Peng YIN, Hye-Yeon CHEONG, Alexis Michael TOURAPIS et Jill BOYCE : Fast Mode Decision and Motion Estimation for JVT/H.264. *In Proceedings of the IEEE International Conference on Image Processing, ICIP 2003*, Barcelone, Espagne, septembre 2003.

[YFZ04] Yu YUAN, David FENG et Yu-Zhuo ZHONG : Three Fast Methods for Adaptive Keyframe Setting and Dynamic Frame-rate Adjusting in Video Coding. *International Journal of Computational Intelligence*, 1(1), 2004.

[YJ96] S. YANTIS et J. JONIDAS : Attentional Capture by Abrupt Onsets and Selective Attention : Evidence from Visual Search. *Journal of Experimental Psychology. Human Perception and Performance*, 20:1505 – 1513, 1996.

[YM04] Andy C. YU et Graham R. MARTIN : Advanced Block Size Selection Algorithm for Inter-Frame Coding in H.264/AVC. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'2004*, pages 95 – 98, Singapore, Republic of Singapore, octobre 2004.

[YMP05] Andy C. YU, Graham R. MARTIN et Heechan PARK : Improved Schemes for Inter-Frame Coding in the H.264/AVC Standard. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'2005*, volume 2, pages 902 – 905, septembre 2005.

[YPG01] Hector YEE, Sumanita PATTANAIK et Donald P. GREENBERG : Spatiotemporal Sensitivity and Visual Attention for Efficient Rendering of Dynamic Environments. *ACM Transactions on Graphics*, 20(1):39–65, 2001.

[Yu04] Andy C. YU : Efficient Block-Size Selection Algorithm for Inter-Frame Coding in H.264/MPEG-4 AVC. *In Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing, ICASSP'2004*, volume 3, pages 169 – 172, Montreal, Quebec, Canada, mai 2004.

[YZLS05] Xiaoquan YI, Jun ZHANG, Nam LING et Weijia SHANG : *Improved and Simplified Fast Motion Estimation for JM*. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, document JVT-P021), 16ème Meeting : Poznan, Pologne, juillet 2005.

[ZBR06] Yafan ZHAO, Maja BYSTROM et Iain RICHARDSON : A MAP Framework for Efficient SKIP/code Mode Decision in H.264. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'2006*, Atlanta, GA, USA, octobre 2006.

[ZDL00] Wenjun ZENG, Scott. DALY et Shawmin LEI : Point-wise Extended Visual Masking for JPEG-2000 Image Compression. *In Proceedings of the International Conference on Image Processing, ICIP 2000*, Vancouver Canada, septembre 2000.

[ZDL02] Wenjun ZENG, Scott. DALY et Shawmin LEI : An Overview of the Visual Optimization Tools in JPEG2000. *Signal Processing : Image Communication*, 17:85 – 104, janvier 2002.

[ZLC02] Ce ZHU, Xiao LIN et Lap-Pui CHAU : Hexagon-Based Search Pattern for Fast Block Motion Estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(5):349 – 355, mai 2002.

[ZPW08] Yinqing ZHAO, Krit PANUSOPONE et limin WANG : Fast Motion Estimation Algorithm using Unequal Search Effort for H.264|MPEG-4 AVC Encoder. *In Proceedings of the IEEE International Conference on Image Processing, ICIP'2008*, San Diego, CA, USA, octobre 2008.

[ZSO+96] J. ZHAO, Y. SHIMAZU, K. OHTA, R. HAYASAKA et Y. MATSUSHITA : An Outstandingness Oriented Image Segmentation and its Application. *In Proceedings of the International Symposium on Signal Processing and its Applications*, 1996.

[ZWH+07] Kai ZHANG, Qiang WANG, Qian HUANG, debin ZHAO et Wen GAO : A Context-based Adaptive Fast Intra 4x4 Prediction Mode Decision Algorithm for H.264/AVC Video Coding. *In Proceedings of the Picture Coding symposium, PCS 2007*, Lisbonne, Portugal, novembre 2007.

Liste des publications

Conférences Internationales avec comité de lecture et publication des actes

Olivier Brouard, Fabrice Delannay, Vincent Ricordel, and Dominique Barba,
Spatio-Temporal Segmentation and Regions Tracking of High Definition Video Sequences based on a Markov Random Field Model
Proceedings of IEEE International Conference on Image Processing (ICIP 2008), San Diego, USA, octobre 2008.

Olivier Brouard, Fabrice Delannay, Vincent Ricordel, and Dominique Barba,
Fast Long-Term Motion Estimation for High Definition Video Sequences based on Spatio-Temporal Tubes and using the Nelder-Mead Simplex Algorithm
Proceedings of Picture Coding Symposium (PCS 2007), Lisbonne, Portugal, novembre 2007.

Olivier Brouard, Fabrice Delannay, Vincent Ricordel, and Dominique Barba,
Robust Motion Segmentation for High Definition Video Sequences using a Fast Multi-Resolution Motion Estimation based on Spatio-Temporal Tubes
Proceedings of Picture Coding Symposium (PCS 2007), Lisbonne, Portugal, novembre 2007.

Conférences Nationales avec comité de lecture et publication des actes

Olivier Brouard, Fabrice Delannay, Vincent Ricordel et Dominique Barba,
Estimation Robuste du Mouvement Global au sein de Séquences Vidéo Haute Définition après une Estimation du Mouvement basée Tubes Spatio-Temporels
Compression et représentation des signaux audiovisuels (CORESA 2007), Montpellier, France, novembre 2007.

Olivier Brouard, Vincent Ricordel et Dominique Barba,
Cartes de Saillance Spatio-Temporelle basées Contrastes de Couleur et Mouvement Relatif
Compression et représentation des signaux audiovisuels (CORESA 2009), Toulouse, France, mars 2009.

Rapports de contrats

Olivier Brouard, Fabrice Delannay, Vincent Ricordel et Dominique Barba,

Rapport d'étude sur les m´ethodes de conditionnement et de pré-analyse du flux vidéo
Lot 4.1 du Livrable du projet RIAM ArchiPEG (convention ANR05RIAM01401), septembre 2006.

Fabrice Delannay, Olivier Brouard, Vincent Ricordel et Dominique Barba,
Rapport sur les définitions d'algorithmes de filtrage et de pré-analyse du flux vidéo
Lot 4.2 du Livrable du projet RIAM ArchiPEG (convention ANR05RIAM01401), avril 2007.

Olivier Brouard, Fabrice Delannay, Vincent Ricordel et Dominique Barba,
Manuel d'utilisation du logiciel de conditionnement et de pré-analyse du flux vidéo
Lot 4.3 du Livrable du projet RIAM ArchiPEG (convention ANR05RIAM01401), juin 2008.

Olivier Brouard, Fabrice Delannay, Vincent Ricordel et Dominique Barba,
Compte-rendu de test des algorithmes de conditionnement et de pré-analyse du flux vidéo
Lot 4.4 du Livrable du projet RIAM ArchiPEG (convention ANR05RIAM01401), septembre 2008.

Pré-analyse de la vidéo pour un codage adapté. Application au codage de la TVHD en flux H.264

Résumé : Les méthodes d'optimisation d'un codeur vidéo classique ne traitent l'information à réduire que d'un point de vue signal et sont donc « bas niveau ». Bien que des travaux intégrant des propriétés du SVH soient proposés pour l'évaluation de la qualité, ou améliorer les techniques de codage, ces méthodes sont peu retenues au niveau des standards. Les travaux de recherche se portent davantage sur l'enrichissement des nouvelles normes, tel que le standard H.264. Cependant, les méthodes « haut niveau » obtiennent des performances encourageantes. Nous proposons donc une méthode de pré-analyse de la vidéo, qui intègre un modèle de l'attention visuelle. Le but est d'analyser la vidéo en tenant compte des informations haut niveau, pour transmettre au codeur le jeu de paramètres optimal afin d'exploiter au mieux les outils de codage. Les études réalisées pour modéliser l'attention visuelle ont mis en évidence le caractère primordial du contraste de mouvement. Notre méthode de pré-analyse détecte d'abord les objets en mouvement (par rapport à celui de la caméra), puis calcule les cartes de saillance permettant de déterminer les zones visuellement importantes. Nous proposons deux applications de codage (qui peuvent être utilisées conjointement) en fonction des informations obtenues après la pré-analyse, ainsi que l'évaluation de leurs performances. La première propose de modifier adaptativement la structure du GOP en fonction du contenu spatio-temporel de la vidéo. La deuxième concerne une application de compression de la vidéo avec une qualité visuelle différenciée guidée par les cartes de saillance. Les performances sont analysées à partir de tests d'évaluation subjective de la qualité.

Mots-clés : compression vidéo, pré-analyse de la vidéo, attention visuelle, codage vidéo avancé, codeur H.264, TVHD.

Pre-analysis of video for its advanced coding. Application to the HDTV coding in H.264 streams

Abstract: The optimization methods of a classical video encoder process the information to reduce it only as a signal point of view, and are therefore «low level». Although some works incorporating the HVS properties have been proposed to assess the quality, or to improve the coding, these methods have not been included in the standards. Researchs focus more on the improvment of new standards, such as H.264. However, the performances of the «high level» methods are encouraging. We therefore propose a method of pre-analysis of videos, that incorporates a visual attention model. This model aims at analyzing the video taking into account the high level information, in order to transmit to the encoder an optimal set of parameters, and to exploit efficiently the coding tools. Researchs have highlighted the fundamental characteristic of the motion contrast for the visual attention. Our pre-analysis method first detects the moving objects (relative to that of the camera), then calculates the saliency maps to localize the visually important areas. We propose two coding applications (that can be exploited jointly) based on information obtained after the pre-analysis, and the assessment of their performance. The first proposes to adaptively modify the structure of the GOP, it is based on the spatio-temporal content of the video. The second concerns an application of video compression with a visual differentiated quality guided by the saliency maps. The performances of these two methods are carried out using tests of subjective quality assessment.

Keywords: video compression, video pre-analysis, visual attention, advanced video coding, H.264 encoder, HDTV.