MÉMOIRE DE MASTER DE RECHERCHE
« ARCHITECTURES LOGICIELLES DISTRIBUÉES »

Pré-analyse de plans vidéos Haute Définition

Application à l'optimisation du codage en flux H.264

Jérôme GORIN

20 Avril 2007

encadré par Vincent RICORDEL

— ÍVC —

INSTITUT DE RECHERCHE EN COMMUNICATIONS ET EN CYBERNÉTIQUE DE NANTES

Pré-analyse de plans vidéos Haute Définition

Application à l'optimisation du codage en flux H.264

Jérôme GORIN

Résumé

H.264/MPEG-4 "AVC" est une norme utilisant les toutes dernières innovations de la technologie en matière de compression vidéo pour fournir un rapport débit/qualité encore jamais atteint avec d'autres standards. Son domaine d'application est large car ce standard est adapté a une très grande variété de réseaux et de systèmes (vidéophonie, streaming, télévision, mobile). Cependant, en raison de sa complexité, le codeur H.264 ne peut, pour l'instant pas, répondre de façon optimale à notre applications, à savoir l'implantation de cette norme dans une plateforme de codage temps-réel d'un flux Haute Définition. Ce document a pour but de présenter les possiblités d'optimisation de codage en utilisant le concept de pré-analyse vidéo.

Catégories et descripteurs de sujets : I.4.2 [**Compression (Coding)**]: Approximate methods; I.4.6 [**Segmentation**]: Pixel classification; I.4.8 [**Scene Analysis**]: Motion, Color, Tracking; I.5.3 [**Clustering**]: Algorithms, Similarity measures; I.4.10 [**Image Representation**]: Statistical

Termes généraux : Algorithmes, H.264, segmentation, Markov

Mots-clés additionnels et phrases : Optimisation H.264, Champs de markov, Tubes spatio-temporels

Remerciements

Je remercie en tout premier lieu M. Vincent Ricordel pour avoir encadré mon travail depuis mon projet technique de dernière année d'école d'ingénieur et pendant ces 6 mois de stage de D.E.A.. Je lui dois de mon futur emploi en temps qu'ingénieur dans le domaine de la vidéo grâce à sa formation et aux sujets sur lesquels j'ai pu travailler.

Je remercie également M. Olivier Brouard et M. Fabrice Delannay de m'avoir intégré au sein de leur équipe. J'ai passé 6 mois fort enrichissant tant au point de vue technique que humain. Je les remercie de m'avoir accordé de leur temps afin que je réalise ce mémoire dans les meilleures conditions possible.

Je remercie enfin toute l'équipe du laboratoire d'IVC pour leur accueil.

Table des matières

Introduction générale

L'évolution des technologies des terminaux, des algorithmes de codage de la vidéo et des réseaux de communication vont permettre la mise en exploitation de nouveaux services audiovisuels tels la Télévision Numérique Terrestre et les réseaux mobiles de troisième génération. Les bandes passantes disponibles sur ces réseaux restent une ressource d'autant plus rare que l'étendue et le type des services visés sur ces plates-formes se diversifient. En effet, aux services de diffusion (un vers tous) s'ajoutent des services personnalisés (à la demande). Par ailleurs, la télévision haute définition va progressivement prendre le pas sur la télévision en dénifition standard. La télévision sur terminaux mobiles (téléphones de troisième génération) va également faire son apparition. Ainsi, bien que la bande passante globalement disponible augmente, il est plus que jamais nécessaire d'optimiser la bande passante utilisée par chacun de ces nouveaux services.

H.264/AVC MPEG-4 part 10[MHM03], est une norme de codage vidéo développée conjointement par l'UIT-T Q.6/SG16 Video Coding Experts Group (VCEG) et l'ISO/CEI Moving Picture Experts Group (MPEG) et est le produit d'un effort de partenariat connu sous le nom Joint Video Team (JVT). La norme UIT-T H.264 et la norme ISO/CEI MPEG-4 Part 10 (ISO/CEI 14496-10) sont techniquement identiques, et la technologie employée est aussi connue sous le nom AVC, pour Advanced Video Coding. La première version de la norme a été approuvée en mai 2003 et la plus récente date de mars 2005.

Ces nouveaux algorithmes de compression sont de plus en plus complexes et nécessitent des capacités de traitement de plus en plus lourdes. De plus, les nouveaux services de Télévision Haute Définition introduisent un nouvel ordre de grandeur dans la complexité de traitement.

Le projet ArchiPEG[BDR06] vise à développer une plateforme temps réel d'encodage H.264 pour la vidéo HD. Ce projet est partagé entre deux partenaires l'IRCCyN et VITEC multimédia.

Ce stage s'inscrit dans le sous-projet de pré-analyse et de conditionnement du flux vidéo en haute définition développé par le laboratoire IRCCyN depuis Mars 2006. Ce travail, appelé pré-analyse, vise à déterminer une meilleure stratégie de codage (choix de prédicteur et de contexte de codage) et de conditionner l'information avant le codage. Mon travail devra s'intégrer à l'outil de pré-analyse basé sur des tubes spatio-temporels déjà développés par le projet

Le codage optimal d'une vidéo dans le format H.264 est un procédé très lourd en matière de calcul. Le travail à effectuer devant être intégré dans une plateforme temps réel disposant d'unités de calcul limitées, une pré-analyse permettrait de détecter les éléments importants d'une scène afin de mieux les coder et ainsi améliorer le rendu global de cette vidéo. C'est ici qu'apparaît l'intérêt de la caractérisation d'un macrobloc. Une connaissance à priori des objets composant une scène permettrait de diriger et d'optimiser leur codage. Par exemple, un objet ayant un contenu

riche en informations pourrait disposer d'une qualité meilleure que celle des macroblocs n'ayant que peu d'intérêt pour la scène. Il est donc important de déterminer des techniques d'analyse de scène vidéo, les critères qui vont révéler l'importance d'un objet dans une image et sur quels paramètres ces informations doivent influer.

La première partie de ce rapport pose la problématique du codage H.264 en exposant les différentes propriétés du codage H.264 sur lesquels nous devont agir, ainsi que les différentes techniques d'optimisation de l'actuel codeur donnant une alternative au problème de codage. Nous verrons que la complexité de calcul du codage H.264 "'optimal"' actuel interdit l'utilisation de ce codeur pour une application temps-réel. Nous introduirons alors, dans la deuxième partie de ce rapport, le concept de pré-analyse de vidéo comme une voie possible d'optimisation de ce codeur. La troisième partie de ce rapport présente les spécifications et la méthodologie de l'outil de pré-analyse développé par le projet ArchiPEG. La quatrième partie du rapport détaille l'implantation de l'outil développé durant le stage pour affiner la pré-analyse des vidéos. Enfin, la dernière partie va présenter les résulats experimentaux obtenues grâce à notre outil et conclut sur les améliorations pouvant encore être apportées.

Chapitre 1

Problématique du codage H.264

Le dernier standard de codage vidéo, à savoir MPEG-4 Part 10 (ou encore AVC ou H.624), vise à gagner jusqu'à 50% de la bande passante actuellement utilisée par MPEG-2 pour une qualité visuelle équivalente. On s'accorde donc à décrire ce standard [IR03] comme le futur de la compression des signaux TV capable de transmettre un programme HD[1] à des débits allant de 6 à 9 Mbits/s.

De telles performances ne peuvent être atteintes qu'au prix d'une estimation et d'une compensation de mouvement complexes, afin d'exploiter de façon optimale les redondances spatiales et temporelles présentes au sein des vidéos. Le standard H.264 offre donc une palette large et complexe de possibilités pour l'estimation et la compensation de mouvement, notamment au niveau de la précision des vecteurs déplacement,la taille variable des blocs estimés, les modes pour la prédiction inter et intra et la sélection des images de référence [Har03].

1.1 Présentation de la norme H.264

En 1998, Video Coding Expert Group (VCEG) se lance dans un projet appelé H.26L, dont le but est de multiplier par deux l'efficacité du codage vidéo par rapport à n'importe quelle norme existant alors. En 2001, MPEG rejoint VCEG et le JVT est créé pour concrétiser la nouvelle norme. En 2003, la norme H.264/MEG-4 Part 10 est publiée. L'objectif de H.264 est un codage efficace et robuste pour le transport de la vidéo. Les applications sont diverses : la communication vidéo (vidéoconférence et vidéotéléphonie), le codage haute qualité pour la diffusion vidéo et la vidéo en temps réel sur des réseaux par paquets (Internet). Ce chapitre présente les principales caractéristiques de la norme H.264.

1.2 Structure du codeur

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 les nouveaux termes introduits par ce standard.

[1] TVHD : télévision haute-définition.

1.2.1 Terminologie

Une **trame** (ou une **frame**) est une image d'une séquence vidéo (une trame correspond à une vidéo entrelacée et une frame à une vidéo progressive). Á chacune de ces images est assigné un numéro (signalé dans le flux binaire), ne correspondant pas forcément à l'ordre du décodage. Les **images de référence** sont des images précédemment codées et décodées qui pourront être utilisées pour le codage d'images suivantes (prédiction inter-frame). Ces images de référence sont organisées en deux listes.

Une image, avant d'être codée, est quadrillée en blocs de pixels (appelés **macroblocs**) contenant 16x16 échantillons de luminance (information sur la luminosité) et deux fois 8x8 échantillons de chrominance (information sur la couleur)[2].

Il existe trois types de codage pour un macrobloc :

Les **macroblocs I** sont obtenus par **prédiction intra** (codage sans image de référence) à partir d'échantillons décodés dans la tranche courante. Une prédiction est formée soit pour un macrobloc complet, soit pour chaque bloc 4x4 de luminance (et les blocs de chrominance associés) du macrobloc.

Les **macroblo cs P** sont obtenus par **prédiction inter**, c'est-à-dire à partir d' images de référence. Un macrobloc codé inter peut être divisé en partitions de macroblocs : des blocs de taille 16x16, 16x8, 8x16 ou 8x8 de luminance (et les blocs de chrominance associés). En choisissant la partition 8 x 8, chaque sous macrobloc peut être de taille 8x8, 8x4, 4x8 ou 4x4. La prédiction se fait depuis une image de référence prise dans une des deux listes.

Les **macroblocs B** sont obtenus par **prédiction inter** à partir de plusieurs images de référence. Chaque partition de macrobloc peut utiliser une ou deux images de référence.

Les macroblocs sont arrangés en **tranches** (slice en anglais), chaque tranche contenant un nombre de macroblocs compris entre un et l'intégralité des macroblocs de l'image. 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.

Un **GOP** (ou groupe d'images) est un ensemble d'images constituant une séquence. Il commence par une image I, suivie d'une image P et/ou B. Le prochain GOP commence à l'image I suivante (et ainsi de suite jusqu'à la fin de la séquence).

1.2.2 Le codec H.264

Tout comme les autres standards de codage vidéo, **H.264 ne définit pas un codec** mais la syntaxe d'un flux binaire codé de la vidéo et la procédure de décodage de ce flux. En pratique, un codeur/décodeur conforme à la norme inclut les fonctionnalités décrites sur les figures 1.1 et 1.2.

Le codeur (figure 1.1) 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 le chemin avant de la figure 1.1, une trame ou une image Fn est partitionnée en macroblocs, une prédiction *P* est formée en fonction des échantillons reconstruits. Dans le mode intra,

[2]En mode de codage 4 :2 :0 correspondant au mode d'échantillonnage des couleurs le plus utilisé

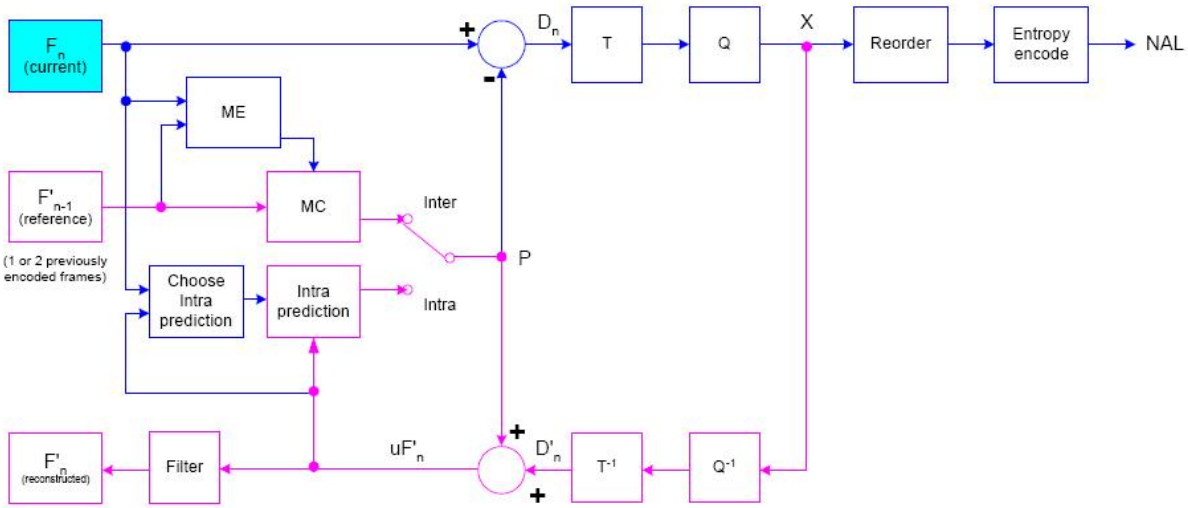

FIG. 1.1 – Schéma du codeur H.264

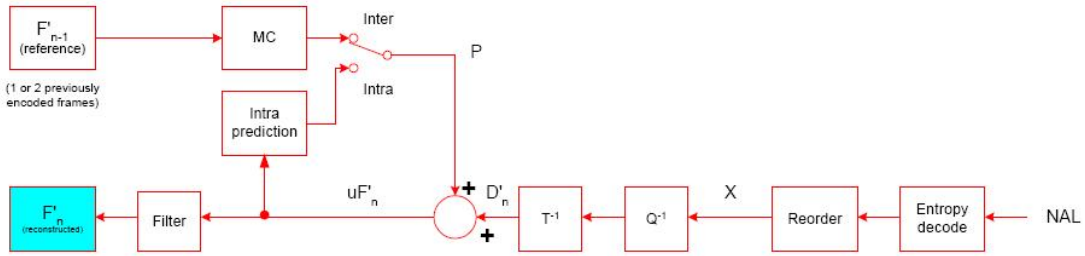

FIG. 1.2 – Schéma du décodeur H.264

P est obtenue à partir d'échantillons de la tranche courante F'_n ayant déjà été codée, décodée et reconstruite. En mode inter, p est obtenue par prédiction et compensation de mouvement à partir d'une ou deux images de référence F'_{n-1}.

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 canal (NAL) pour transmission ou stockage.

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 ré-échelonnement (Q^{-1}) et une transformée inverse (T^{-1}) pour produire un bloc D'_n et créer le bloc reconstruit uF'_n, c'est-à-dire, la version décodée du bloc original. Enfin, un filtre peut être 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 (figure 1.2), le même processus est appliqué. Les macroblocs reçus par le NAL son décodés, réarrangés pour obtenir les coefficients X. Ré-échelonnement et transformée inverse sont appliqués pour obtenir D'_n (identique au D'_n du schéma d'encodage). Grâce aux informations d'entête 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.

1.2.3 Les profils et niveaux

Le standard H.264 propose trois profils (*baseline profile*, *main profile*, *extended profile*) et onze niveaux afin d'établir les sites de conformité. Les premiers indiquent un ensemble de codage pouvant être utilisé pour générer un flux compatible. Les seconds imposent des contraintes à certains paramètres clés du flux (par exemple la taille maximale d'une image). La figure 1.3 résume les fonctionnalités incluses dans chaque profil.

Ces différents profils permettent de produire un codage adapté à de multiples 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.

1.3 Prédiction inter-frame

H.264 permet d'utiliser une ou deux images de référence pour la prédiction inter-image. La prédiction peut se faire à partir du passé, du futur, ou en bi-directionnel. Pour cela, le codeur

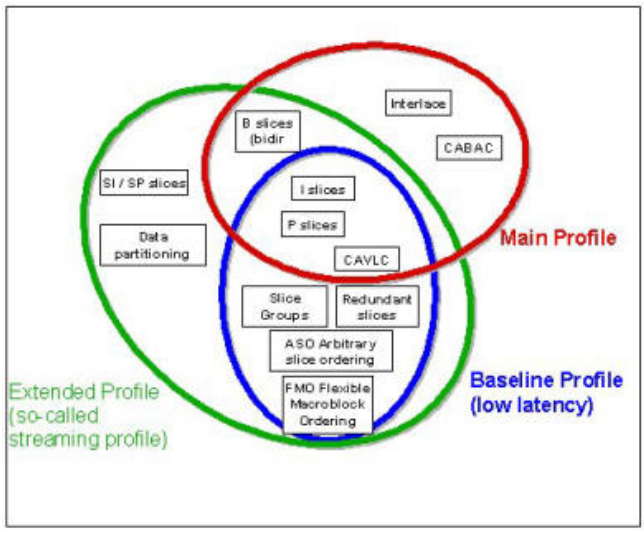

FIG. 1.3 – Les profils dans H.264

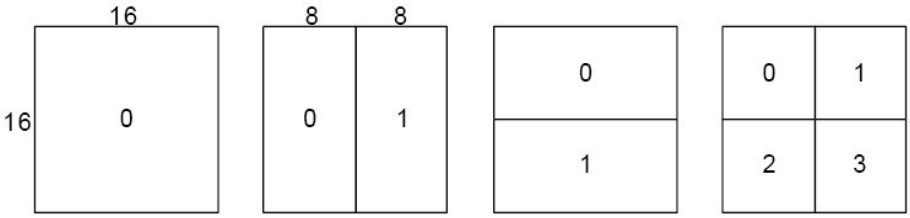

FIG. 1.4 – Partition de sous macroblocs : 16x16, 8x16, 16x8 et 8x8

et le décodeur maintiennent une ou deux listes d'images de référence précédemment décodées, selon qu'il s'agisse d'une tranche *P* ou d'une tranche *B*. Les techniques mises en oeuvre dans la prédiction inter-image sont présentées par la suite.

1.3.1 Les tranches *P*

La prédiction inter par **tranche P** permet de créer un modèle de prédiction d'un macrobloc à partir d'une **frame précédemment décodée**, sélectionnée parmi une liste d'images de référence. Cette prédiction est générée par une compensation de mouvement basée bloc, issue d'une technologie d'exploration de la position d'un bloc dans une fenêtre de recherche de l'image de référence minimisant la quantité d'informations à coder.

Compensation de mouvement à structure d'arbre

Le codage H.264 supporte une compensation de mouvement sur des blocs de luminance de taille 16x16 jusqu'à 4x4 avec de nombreuses options entre ces deux tailles. La composante de luminance de chaque macrobloc 16x16 peut être scindée de 4 manières (figure 1.4).

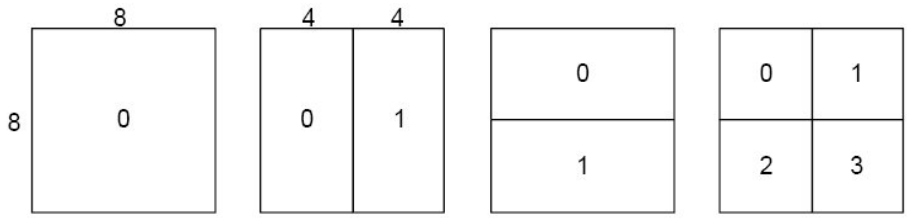

FIG. 1.5 – Partitions de sous macroblocs : 8x8, 4x8, 8x4 et 4x4

En choisissant le mode 8x8, les quatre sous-macroblocs peuvent encore une fois être divisés suivant la topologie de la figure 1.5. Ces partitions et sous-partitions permettent un grand nombre de combinaisons pour chaque macrobloc. Cette méthode de partitionnement de macroblocs par compensation de mouvement de sous-macroblocs de tailles variables est appelée **compensation à structure d'arbre**.

Un vecteur de mouvement est nécessaire à la compensation de chaque partition ou sous-partition. Le codeur doit donc trouver un compromis entre grande partition à coder (peu de bits pour coder le choix de partition et le vecteur résultant mais énergie importante dans les zones riches) et petite partition (diminition de l'énergie résiduelle mais complexité de codage des partitions et des vecteurs associés). En règle générale, une grande taille est appropriée pour les zones homogènes et une petite pour les détails.

Les composantes de chrominance du macrobloc (de résolution moitié moindre que celle de la luminance en mode d'échantillonnage 4 :2 :0) seront codées de manière analogue à la composante de luminance.

Les vecteurs de mouvement

Chaque partition ou sous-partition d'un macrobloc codé inter est prédite à partir d'une zone de taille équivalente dans l'image de référence. La différence de position entre les deux zones (le vecteur de mouvement) a une résolution au quart de pixel pour la luminance et au huitième d'échantillon pour la chrominance. Les échantillons de ces composantes n'existant pas à ces positions sous-échantillonnées dans l'image de référence, il est nécessaire de les approximer par interpolation des échantillons voisins. Cette interpolation passe par différentes étapes et formules décrites en [MHM03].

Prédiction des vecteurs de mouvement

Le codage des vecteurs de mouvement est allégé en ne codant que la différence entre le vecteur de mouvement réel et le vecteur de mouvement prédit. Cette prédiction est basée sur l'hypothèse que les vecteurs de mouvements des partitions voisines sont souvent corrélés. Ainsi, chaque vecteur de mouvement est tout d'abord prédit à partir des vecteurs des partitions voisines déjà codés. La méthode de prédiction utilisée varie en fonction de la taille de la partition pour la compensation de mouvement et de la disponibilité des vecteurs à proximité.

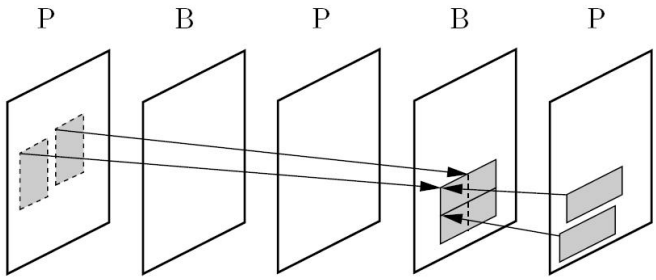

FIG. 1.6 – Prédiction bi-directionnelle de la tranche B

Outre ces prédictions, les macroblocs peuvent être également codés en mode ***SKIP***. Dans ce cas, ne sont transmis ni signal d'erreur de prédiction, ni d'index de référence, ni de vecteur de mouvement. Le macrobloc est reconstitué comme s'il s'agissait d'un macrobloc 16x16 en tout point identique au macrobloc pointé par le vecteur prédit dans l'image de référence.

1.3.2 Les tranches *B*

Les **tranches *B***, a contrario des tranches P, utilisent des signaux de prédiction générés à partir de la moyenne pondérée de **deux valeurs de prédiction**. Quatre techniques de prédiction sont possibles : mode direct, prédiction avec compensation de mouvement à partir d'une des listes d'images de référence ou avec double compensation de mouvement. Différents modes peuvent être choisis pour chaque partition. Cependant, dans le cas d'une partition 8x8 le mode de prédiction choisi sera appliqué à toutes les sous-partitions de celle-ci.

La **bi-prédiction** ou **double prédiction** recherche deux blocs de tailles identiques à la partition ou sous-partition du macrobloc courant à partir des listes des images de référence minimisant l'information à coder. On obtient ainsi deux zones de référence, ce qui implique deux vecteurs de mouvement (figure 1.6). La prédiction finale est alors formée par la moyenne pondérée des blocs référencés par les deux vecteurs. Le résidu est ensuite soustrait au macrobloc courant puis codé. Chaque vecteur est prédit à partir des vecteurs de mouvement voisin ayant la même direction temporelle.

Le mode de **prédiction directe** n'utilise aucun vecteur de mouvement pour coder un bloc. Le décodeur, pour lire ce bloc, utilise le vecteur prédit pour effectuer une bi-prédiction des échantillons résiduels.

1.4 Prédiction intra-frame

A l'inverse de la prédiction temporelle, la **prédiction intra** n'utilise pas d'images de référence. Elle **exploite la redondance spatiale d'une image**, c'est-à-dire le fait que dans une image naturelle les valeurs des pixels proches spatialement soient très corrélées. Le codeur H.264 a la particularité de travailler dans le domaine transformé comme c'est le cas pour les codeurs MPEG-2 et MPEG-4. Dans tous les types de codage de tranches, le codage H.264 permet **deux modes**

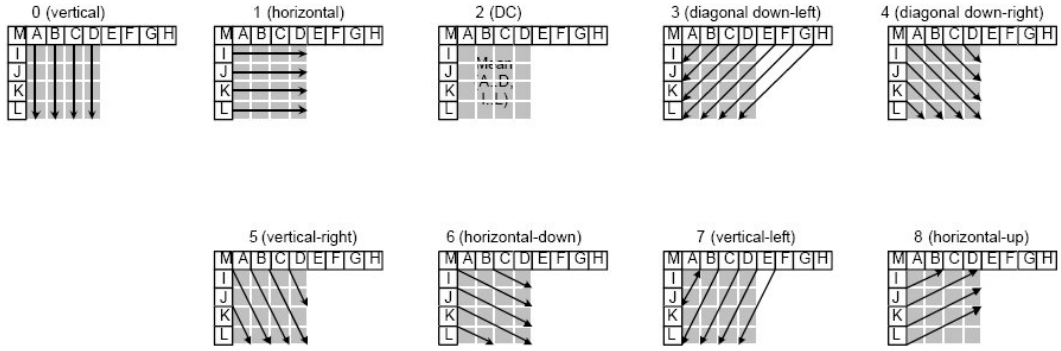

FIG. 1.7 – Les modes de prédiction de blocs 4 x 4 de luminance

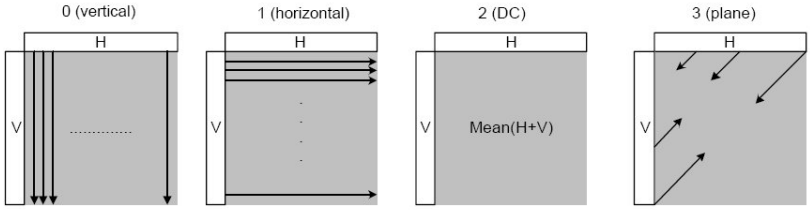

FIG. 1.8 – Les modes de prédictions des blocs 16x16 de luminance

de prédictions intra de bloc 4x4 et 16x16. La prédiction est dépendante de la taille de l'échantillon considéré. Pour le mode de prédiction de bloc 4x4, chaque bloc de luminance recourt à un des **neuf modes de prédiction**. Outre la prédiction DC, huit modes de prédiction directionnelle sont spécifiés. La figure 1.7 illustre la direction des prédictions de chacun de ces modes.

Dans le **mode de prédiction 16x16**, recommandé dans le cas de zones d'images régulières, une prédiction uniforme est obtenue pour l'ensemble de la luminance d'un macrobloc. **Quatre modes de prédiction** sont disponibles : extrapolation verticale ou horizontale, le mode DC et une prédiction linéaire oblique (figure 1.8). La prédiction des échantillons de chrominance d'un macrobloc repose toujours sur une technique semblable à celle utilisée pour la luminance des macroblocs 16x16. La prédiction intra au-delà des limites de la tranche n'est pas autorisée afin de maintenir l'indépendance des tranches les unes par rapport aux autres.

1.5 Transformation et quantification

Après prédiction et soustraction de la prédiction au bloc original, le résidu est **transformé** puis **quantifié**.

Le codeur H.264 utilise trois **transformées** selon le type de résidu à coder : une transformée pour les matrices 4x4 des coefficients des composantes continues de luminance des macroblocs intra (prédiction en mode 16x16), une transformée sur les matrices des coefficients des composantes continues de chrominance (pour tous les macroblocs) et une transformée en cosinus

discrète (DCT) pour tous les autres macroblocs 4x4. La transformée inverse est réalisée par des opérations exactes sur des entiers pour éviter les discordances.

La **quantification** des coefficients de la transformée entraîne des pertes mais produit une compression importante des données. Le codeur H.264 utilise une quantification scalaire en calculant l'arrondi d'un nombre fractionnel entier. Pour chaque macrobloc, le paramètre de quantification (QP) sélectionne un des 52 quantificateurs. Les quantificateurs sont disposés de manière à obtenir une augmentation approximative de 12,5% de la taille du pas de quantification lorsque QP augmente de 1. Le document de Manjunath et al[MHM03] donne plus de précision sur cette transformée et cette quantification.

1.6 Codage Entropique

Les données arrivant à l'unité de codage entropique sont essentiellement les données résiduelles issues de la quantification mais peuvent être également des entêtes, des vecteurs de mouvement, des méthodes de prédiction, des index des images de référence, etc... Après une étape de réarrangement des données dans un tableau, le codeur H.264 dispose de plusieurs méthodes de codage entropique, pour les éléments syntaxiques et pour les coefficients de transformée quantifiés.

Le **codage Exp-Golomb** est un codage à longueur variable utilisant un ensemble illimité de mots-code définis 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 de données.

Le **codage entropique Context-based Adaptative Variable Length Coding** (CAVLC) est utilisé pour le codage des coefficients de transformée quantifiés. Ce codage tire avantage des blocs 4x4 quantifiés en représentant de manière compacte les blocs contenant principalement des zéros, des longues séquences de coefficients de +/- 1 et en adaptant les tables de correspondance en fonction des amplitudes déjà codées.

La **méthode Context-Adaptative Binary Arithmetic Coding** (CABAC), disponible dans le profil principal, permet d'améliorer encore le codage entropique. Cette méthode va effectuer un codage arithmétique en fonction de modèles de probabilité d'apparition d'une séquence. Cette méthode permet une réduction de 10% à 15% par rapport au codage CAVLC.

1.7 Les méthodes d'optimisations du codeur

Les différentes étapes de compression du codage H.264 permettent essentiellement plus de précision et une meilleure prédiction des différentes unités de compression selon trois facteurs : plus de partitionnement des macroblocs pour la prédiction inter, réduction de la taille des blocs dans la prédiction intra et meilleure précision des vecteurs de mouvement. Ces techniques permettent d'améliorer la compression et de faire de H.264 le meilleur codec existant (50% de débit en moins par rapport à MPEG-2 pour une qualité égale). Mais ce gain obtenu en terme de débit est réalisé au détriment des temps de calcul qui, malgré l'augmentation des performances des

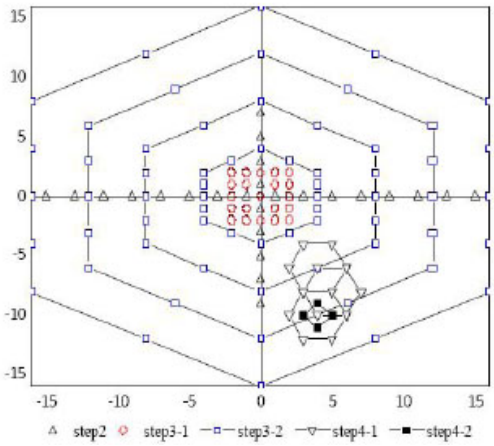

FIG. 1.9 – Processus de recherche partielle

machines actuelles, deviennent très importants. Un macrobloc peut, selon le procédé de codage H.264, être codé avec 16 sous-partitions en inter avec deux modes de prédictions possibles et une quarantaine de possibilité de codage en intra. Si l'on multiplie ces possibilités par le nombre de macroblocs contenus dans une séquence HD, il devient impossible de tester toutes ces possibilités en respectant des contraintes temps réel [RMC02]. La partie suivante effectue un listing de quelques méthodes d'optimisation proposées dans la littérature. Nous verrons que ces optimisations seront toujours réalisées au détriment de la qualité de la vidéo finale.

1.7.1 Estimation de mouvement

Parmi tout le processus de codage, la prédiction inter est la partie la plus couteuse en calculs et peut représenter 60% (1 seule image référence) à 80% (5 images références) du temps d'encodage [MHM03]. De nombreux algorithmes d'estimation ont été proposés afin d'optimiser le temps de recherche des macroblocs lors d'un codage de tranche P. Ces algorithmes peuvent être séparés en trois classes.

Les **méthodes d'estimation de mouvement multi-résolutions** [CHC+05] débutent la recherche à la plus faible résolution spatiale. Les vecteurs résultants de la recherche initiale seront ensuite multipliés par un facteur proportionnel au changement de résolution.

Les **méthodes d'estimation du mouvement avec recherches partielles** [TTB03] proposent de ne pas tester tous les points d'une fenêtre de recherche. Le processus de recherche typique de ces algorithmes est illustrés en figure 1.9. Cette recherche procède en quatre étapes : sélection du mode de prédiction, recherche sur une croix asymétrique, recherche sur des grilles hexagonales de tailles variables, recherche localisée de forme hexagonale autour du vecteur optimal.

Les algorithmes précédents appliquent des méthodes d'accélération du processus d'estimation au détriment du débit et de la qualité. En effet, la méthode classique de recherche exhaustive obtient de meilleurs résultats, puisque tous les points sont testés. Les **méthodes d'estimation avec nouvelles métriques** [MYKP06] diffèrent car elle réduisent le flux de données tout en préservant la qualité perceptuelle. Ces algorithmes, basés sur des métriques de qualité prenant

compte des dégradations des informations de luminance, de contraste et de structure, permettent, en moyenne, de réduire le flux de données de 20% pour un temps de calcul diminué de 2,5% par rapport au codeur de référence, tout en maintenant une même qualité perceptuelle de la vidéo reconstruite.

1.7.2 Réduction des modes de prédiction

Les méthodes de réduction des modes de prédiction proposent de réduire les possibilités de codage d'un macrobloc en effectuant une prédiction pour les codages intra et inter.

Les algorithmes de **sélection rapide de modes** [HLC+06] consistent à prédire quelques modes possibles de codage du macrobloc courant en fonction des images et des macroblocs précédents.

Le codage des tranches intra nécessite un test sur les huits modes de prédiction disponibles. Les algorithmes de **décision pour les modes intra** [XPR+05] réalisent une détection de la direction de la texture du macrobloc afin de sélectionner au mieux le mode de prédiction à utiliser sans tester les huit autres possiblités.

Le codage H.264 offre, pour les tranches inter, sept modes de codage possibles. De nombreuses méthodes ont été proposées afin de réduire et sélectionner les **modes inter**. Ces algorithmes [Yu04] recherchent les probabilités de mode de codage pour le macrobloc courant en fonction de sa complexité et en fonction du mode de codage utilisé par ce macrobloc sur l'image précédente. D'autres algorithmes [YM04] introduisent des mesures afin d'identifier les macroblocs devant être codés en SKIP ou en mode inter de grande taille (supérieur à 8x8). En identifiant correctement ces deux catégories de macrobloc, le codeur est exempté de les examiner avec tous les modes inter possibles.

1.7.3 Choix des images de référence

La dernière catégorie de techniques permettant d'optimiser le codeur H.264/AVC, améliore la sélection des images de référence. Par défaut, le codeur de référence utilise cinq images pour la prédiction des macroblocs inter de l'image courante. Or, 80 % des vecteurs de déplacement optimaux pour coder cette image sont obtenus à partir de l'image précédente. Les **méthodes de recherche partielle** [HHW+03] propose d'effectuer directement une recherche appronfondie pour les modes intra et inter sur l'image précédente. Ils étudient ensuite les informations disponibles, telle que, le mode choisi, l'erreur de prédiction intra, l'erreur de prédiction inter et les vecteurs de mouvement, afin de déterminer s'il est utile d'effectuer la recherche sur les autres images de référence plus éloigné temporellement.

Les **méthodes de sélection des images de référence** [OT05] proposent d'utiliser des techniques réalisant un choix rapide des images de référence. Il convient tout d'abord de coder les images afin qu'elles puissent être utilisées comme références pour le codage inter. Ces méthodes sont intéressantes car, avec un gain sur le temps de calcul en moyenne de 23%, elles montrent l'importance du choix des images de référence dans le codage.

1.8 conclusion

Dans ce chapitre, nous avons traité des terminologies et des différentes étapes de compression introduites par le nouveau standard H.264. On constate que ce standard permet essentiellement plus de précision et une meilleure prédiction dans les différentes unités de compression. Ces différentes étapes nécessitent des temps de calcul très importants qui ont nécessairement besoin d'être optimisés. Les méthodes d'optimisation pour le codage H.264 restent encore rare du fait de sa relative jeunesse et nécessitent beaucoup de recherches afin de produire une codage optimal. Le principal défaut de ces méthodes d'optimisation est une complexité de calcul moindre et généralement une baisse de qualité de la vidéo finale codée. Le deuxième chapitre introduit donc une nouvelle méthode d'optimisation de codage se basant sur la pré-analyse de séquences vidéos. Ce pré-traitement se base sur une connaissance de la scène pour en adapter son codage. Ce pré-traitement procéder sur une segmentation d'une séquence en régions, puis sur l'analyse de ces régions afin de sélectionner le meilleur codage à leur appliquer. Nous allons donc faire une rapide présentation des techniques de segmentation existantes dans la littérature pour ensuite avoir une première approche d'optimisation du codeur H.264 qui suivra cette pré-analyse.

Chapitre 2

Pré-analyse d'une vidéo

Le codeur H.264 réalise, lors de la phase de codage d'une séquence vidéo, une optimisation débit-distorsion pour chaque macrobloc afin d'obtenir le meilleur mode de codage (intra ou inter, taille des sous-partitions de macrobloc). Lors de cette optimisation débit-distorsion, le codeur doit réaliser une estimation de mouvement sur tous les modes inter en testant toutes les images de référence précédemment codées-décodées stockées dans un buffer. Cette phase est donc très coûteuse en temps de calcul, alors qu'elle ne garantie pas la cohérence avec le contenu spatio-temporel de la séquence vidéo.

Cette observation indique qu'une connaissance *a priori* sur le contenu spatio-temporel de la séquence vidéo à coder, permettrait de réduire significativement la charge de calculs du codeur. Il apparaît donc nécessaire de placer, en amont du codeur, un phase de pré-analyse dédiée au mouvement au sein de la vidéo. Il sera possible d'appréhender de façon plus juste le mouvement des objets[1] et leur ancrage temporel. Cette analyse doit pouvoir caractériser le mouvement physique ainsi que la complexité locale de l'image dans le but d'accélérer le codage, en choisissant la meilleure stratégie offerte par le codeur H.264 (i.e. le meilleur jeu de paramètres du codeur). La connaissance approfondie des objets (cycle de vie, suivi spatio-temporel, texture, ...) présents dans une scène permettra notamment de décider, pour chacun d'entre eux, quelles sont les meilleures images de référence pour la prédiction et les modes les mieux adaptés à leur codage.

Ce chapitre est divisé en trois étapes. La première partie introduit les outils nécessaires et les domaines d'application de la pré-analyse vidéo. Nous verrons que la segmentation est la base de la pré-analyse. La deuxième partie présente donc les outils existants dans la littérature afin de classer et regrouper les régions d'une vidéo selon des caractéristiques communes. La dernière partie donnent des clés de réponse afin d'exploiter cette segmentation dans le codage H.264.

[1] Un objet désigne ici et par la suite un ensemble de macroblocs dont le mouvement, la couleur et la texture sont homogènes.

2.1 Intérêt de la pré-analyse d'une vidéo

L'efficacité des codeurs actuels (en particulier du codeur H.264) est basée, comme nous l'avons vu précédemment, sur des interactions entre différentes possibilités de représentation de mouvement, du codage des différences entre blocs d'images (l'original et sa prédiction).

Par conséquent, le problème majeur d'un encodage réside dans la sélection des différentes méthodes de prédicition pour le signal liées à la multiplicité de contenus et de mouvements d'une séquence vidéo. De plus, la notion de durée de vie d'un objet, c'est-à-dire qu'un objet apparaît généralement sur plusieurs images successives d'une séquence, n'est pas suffisament prise en compte par le codeur. Elle est pourtant une propriété importante du codage, car appliquer un même codage des macroblocs contenant cet objet au cours d'une séquence permettrait d'obtenir un rendu uniforme au cours du temps et d'éviter une recherche de la meilleure décision de codage, réduisant ainsi le temps de calcul pour l'encodage de l'objet. Il est donc indispensable de pouvoir ajuster les modes de codage en fonction de l'information contenue dans la séquence, afin prédire au mieux la bonne décision à appliquer.

Cet ajustement se déroule en effectuant une pré-analyse de la séquence en segmentant et en classant les informations susceptibles d'être déterminantes pour un codage adéquat.

2.1.1 Pré-analyse et segmentation

Une segmentation adéquate d'une image ouvre la voie a de nombreux types d'applications, notamment pour améliorer les performances d'un codeur en utilisant le schéma de principe de **codage orienté-régions**, c'est-à-dire mettant à profit le **découpage en régions** pour le **codage d'une séquence**. Ce codage nécessite l'utilisation d'une carte de segmentation afin de transmettre les informations de régions au codeur. Une carte de segmentation est en fait une image d'étiquettes définissant des régions uniformes selon des critères pré-définis. Une région est alors définie comme étant un ensemble connexe de pixels étiquetés par un label (numéro). Cette carte de segmentation permet de mettre à profit les corrélations existantes entre les régions similaires placées dans les images d'une séquence. Les travaux de Pateux et Labit [PL97] montrent qu'il existe deux types de représentation de cartes de segmentation : coder la carte des étiquettes des différentes régions comme une image à niveaux de gris ou représenter les différentes régions par l'intermédiaire de leurs contours, plus complexe à coder mais moins gourmandes en ressource.

2.1.2 Application de la pré-analyse

Le principal avantage d'un codage orienté-régions est que le découpage en régions, s'il est bien effectué, permet de définir les objets dans la scène. Il s'en suit alors un codage adapté des régions en fonction des caractéristiques détectées lors de la segmentation. Par exemple, en utilisant une technique d'allocation de débit en fonction de l'importance psycho-visuelle de chaque région, il devient possible d'adapter la qualité de reconstruction de chaque objet et d'améliorer la qualité sur les zones importantes de la scène (exemple en figure 2.1 pour la visiophonie, le fond peut être dégradé afin d'améliorer la qualité du visage de l'intervenant). Le partitionnement en régions peut également être utilisé dans une optique d'optimisation du codage, en retenant

(a) H261 R=39Kb/f
Q_{fond}=25 - $PSNR_{fond}$=29,45
Q_{visage}=25 - $PSNR_{visage}$=30,91

(b) H261 R=39Kb/f
Q_{fond}=31 - $PSNR_{fond}$=28,45
Q_{visage}=11 - $PSNR_{visage}$=34,87

FIG. 2.1 – Illustration d'un codage sans répartition de débit à gauche et avec répartition de débit (visage - reste de l'image) à droite.

pour chacune d'elles la technique de codage la plus performante et fournir plusieurs niveaux de restitutions pour chaque région afin de l'adapter à des contraintes de service ou de qualité. Le document [Don97] établit une étude comparative de diverses techniques de codage adaptées à la forme d'une région telle que le Shape Adaptive Discrete Cosine Transform (SADCT) et la transformée linéaire de Gilge. Ces techniques, bien qu'adaptées au codage orienté-régions, obtiennent des résultats de qualité très dépendantes de la source.

Le choix des techniques de segmentation pour le codage doit tenir compte des paramètres de l'environnement de l'application à savoir le type de vidéo (image, vidéo, images stéréo), le débit (faible, moyen, haut), le type d'application (temps réel ou non), la puissance de calcul disponible et les fonctionnalités (hiérarchisation, progressivité ...). Ces paramètres joueront un rôle prédominant dans la sélection des critères d'évaluation du mode de partition et de codage, ainsi que sur la procédure de codage et de segmentation (manière successive sur des applications "off-line" avec le principe du "multi-pass" ou de manière conjointe pour des applications temps-réel). La thèse de Castagno [Cas98] présente les techniques de pré-analyse utilisées pour le MPEG-4.

Des méthodes conjointes entre les différentes méthodes de segmentation citées précédement permettent de donner une segmentation fiable tout en prenant en compte le coût de codage de cette segmentation (formalisme Minimum Description Length) ou les contraintes de débit et de distorsion (formalisme débit-distorsion). Ces techniques sont présentées plus en détail dans la thèse de Marc Chaumont [Cha03]. D'autres propositions de répartitions de débit entre objets ont été proposées, mettant en avant la notion de régions d'interêt ou bien de régions de focalisation visuelle. Le problème de répartition des débits nécessiterait une connaissance plus approfondie du système de vision humain (SVH). On pressent tout de même que les critères de mouvement, d'ordre de profondeur et de taille sont pertinents pour le SVH [Cha03].

De nombreuses méthodes de recherche et de suivi d'objets ont été proposées dans la littérature se basant sur des critères de trajectoires, de tailles, de textures, de bords et de couleurs d'un objet. La suite de ce document établit un liste des outils de segmentation utilisés par ces méthodes, pour ensuite adapter les informations sur le codage.

2.2 Segmentation/classification

Les techniques de segmentation, qu'elles soient spatiales, temporelles ou spatio-temporelles ont comme objectif de partitionner une image ou une vidéo en régions ayant des caractéristiques proches.

Il existe de nombreuses techniques de segmentation basées sur diverses méthodes mais la mise au point d'algorithmes est encore un thème de recherche courant en traitement d'image. Cependant toutes ces techniques ont en commun la minimisation d'une fonctionnelle d'énergie.

Il n'existe généralement pas de solution unique à la division d'une image en régions "pertinentes". La pertinence d'une région est en effet une notion éminemment dépendante de l'application. Ainsi aucun des algorithmes utilisés ne peut être considéré comme plus efficace qu'un autre.

L'objectif de cette section n'est pas d'effectuer une segmentation de haut niveau d'une image (rassemblement de régions pour la détection d'un objet, compréhension de la scène) mais de pouvoir rassembler les régions d'une image ou d'une vidéo selon des caractéristiques communes.

Les techniques présentées dans un premier temps ne sont que des segmentations spatiales, c'est-à-dire n'effectuant le traitement sur une seule image. Les approches frontières et régions ne sont adaptées à notre utilisation. Nous verrons dans la suite de ce document que les approches statistiques offrent un cadre tout à fait adaptées à notre propos. La suite de cette section nous permettra de comprendre comment étendre cette segmentation sur toutes les images d'une séquence.

2.2.1 Les différentes approches de segmentation

La segmentation fait référence aux notions de différence et de similarité perçues par le système visuel humain. Les segmentations spatiales[2] donnent naissance à deux approches couramment qualifiées d'approche 'frontière' et d'approche 'région'. L'approche région s'attache à faire apparaître des régions homogènes selon un critère (niveaux de gris ou texture), alors que l'approche frontière tente de trouver des contours ou frontières présentant une variation rapide du même critère. Un algorithme de segmentation s'appuie directement sur :

1. la recherche de discontinuités afin de mettre en évidence les contours ;
2. la recherche d'homogénéité locale pour définir les régions ;
3. la coopération des deux principes par approche conjointe ou statistique.

Les segmentations spatio-temporelle[3] utilisent ces méthodes de segmentation spatiale en incluant des principes de suivi d'objet sur le domaine temporel. Ces méthodes sont présentées dans la thèse de Marc Chaumont[Cha03], les lecteurs pourront s'y référer pour une description plus détaillée. Des exemples de segmentation sont illustrés en figure 2.2.

[2] Spatiale induit un traitement sur des caractéristiques spatiales (image) et non temporelle(vidéo)
[3] Segmentation incluant des caractéristiques spatiales et temporelles

FIG. 2.2 – Exemples de résultats de segmentation respectivement par : seuillage, détection de contours et région.

2.2.2 Approches frontière

Les approches frontières, sont parmi les méthodes les plus classiques en segmentation d'images. Ces méthodes supposent généralement un modèle *a priori* des discontinuités recherchées et opèrent de manière très localisée. Les approches frontière peuvent aussi être classées en plusieurs catégories [Cha03] à savoir : les méthodes dérivatives, surfaciques, morphologiques et variationnelles[4]. Les trois premières classes sont adaptées aux régions uniformes. Les résultats de ces techniques sur des images "'naturelles"' sont généralement peu exploitables car elles génèrent principalement des contours non-fermés, des faux contours ou des contours non détectés. En revanche, les techniques variationnelles produisent des contours fermés grâce à la prise en compte d'une information globale sur l'image, generalement issue d'un modèle *a priori* de contour.

2.2.3 Approches région

Contrairement aux approches frontière qui recherchent les dissimilarités, les approches région recherchent plutôt la similarité. Ces approches fournissent une carte de régions fermées. Cependant la localisation des frontières reste généralement peu précise. Parmi les approches régions, on trouve essentiellement trois types de méthodes :

Les méthodes de classification : les classes sont construites à partir des attributs des pixels. Chaque pixel est affecté à une classe unique. Les méthodes de classification plus classiques en segmentation région sont les méthodes de seuillage ainsi que la méthode des nuées dynamiques (ou la méthode des k-moyennes). Ces techniques sont très sensibles à d'autres propriétés de l'image, comme la variation de luminosité et le bruit de caméra. L'avantage de ces techniques est que les résultats peuvent être facilement corrigé par des opérateurs morphologiques, principalement la dilatation et l'érosion [Ser06]. On retrouve également toutes les méthodes de classification floue [LM01] et les méthodes neuronales [CJS98]. La segmentation par les techniques de classification se heurte au problème du choix du nombre de classes (approches non supervisées) et au problème du choix des attributs.

[4] principalement composée de modèles déformables ou contours actifs.

Les méthodes de type croissance de régions : également appelées agrégation de pixels, ces méthodes intègrent implicitement l'information spatiale dans le processus de segmentation. Les régions sont créées les unes après les autres avec pour chaque région, une phase d'initialisation et une phase itérative. La phase d'initialisation est la phase du choix d'un nouveau germe (point de départ d'une nouvelle région). La phase itérative est la phase d'agrégation des pixels voisins au germe selon un critère d'homogénéité jusqu'à convergence.

Les méthodes de type division-fusion "split-merge" : après une étape d'initialisation, le processus de segmentation est itératif et alterne deux phases : une phase de division de toutes les régions non homogènes et une phase de fusion de toutes les régions adjacentes de sorte que la région résultante respecte toujours le critère d'homogénéité. Ces méthodes font appel à la théorie des graphes, ainsi elles peuvent être classées selon la structure du graphe utilisé ; on retrouve le partionnement de Voronoï, l'arbre quaternaire et les approches pyramidales [CP95].

2.2.4 Les approches coopératives

Nous avons pu remarquer que les approches frontières et régions s'appuient sur des informations différentes et complémentaires. Cela a incité plusieurs chercheurs à développer des systèmes de segmentation par coopération de méthodes. D'après Kermad [Ker97], trois types de coopération sont distingués dans la littérature :

La coopération série : ce type de coopération, en juxtaposant les procédés, donne naissance à des algorithmes de segmentation puissants. Cependant le résultat dépend souvent de l'ordonnancement de la coopération.

La coopération parallèle : elle fait appel à trois mécanismes, la fusion,l'adaptation et la correction. Généralement la modélisation utilisée favorise un de ces trois mécanismes, ce qui ne contribue pas d'une manière générale à des résultats très probants.

La coopération hybride : permet de solutionner quelques lacunes des deux coopérations précédentes.

2.2.5 Les approches statistiques

Les approches statistiques sont à rapprocher des méthodes de coopération hybride. Leurs intérêts sont de pouvoir prendre en compte un grand nombre d'informations contextuelles (basées sur différents critères pouvant être indifféremment frontière ou région) dans une image, de manière mathématiquement rigoureuse. Les méthodes statistiques sont le plus couramment utilisées par champs de Markov [Bes74]. Le succès de ce modèle est principalement dû à l'aptitude du modèle de prendre en compte des dépendances spatiales des variables aléatoires, même lorsqu'elles sont en très grand nombre. Les premières applications en traitement d'images furent réalisées pour l'analyse-synthèse de textures [HS80] et pour la segmentation [The83]. Les domaines d'applications sont désormais nombreux et variés. Les modèles Markoviens ont été utilisés en analyse de textures [HS80], en détection de contours [GGG87], en segmentation [The83], en analyse d'images multisources [Rou97] ou en analyse du mouvement dans une séquence d'images

[PCBY04]. Les approches markoviennes sont couramment utilisées pour ajouter plus de régularité dans la segmentation en imposant des contraintes sur le voisinage d'un pixel. Ainsi, l'intérêt de l'utilisation des champs de Markov en segmentation d'images est de mieux modéliser l'image car ces méthodes ne créent pas de nouvelles régions mais se "'contentent"' d'affiner une segmentation en utilisant d'autres informations contextuelles. La description des champs de markov sera approfondie dans le chapitre 4.

2.2.6 Le suivi de segmentation : le "'tracking"'

La segmentation vidéo peut être vue en deux grandes étapes :

- la segmentation d'une image ou d'un groupe d'images (segmentation 3D[5]) par une approche basée contour ou région ;
- puis le suivi 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 zone même si elle change de forme, de texture ou bien est occultée. Deux approches [MD94] sont possibles :

- une approche par **mise en correspondance de deux cartes de segmentation consécutives**. La mise en correspondance de l'image au temps t et de celle au temps $t+1$ se fait en identifant les régions présentes au temps t dans la carte de segmentation au temps $t+1$. On utilise alors des critères de recouvrement ;
- une approche par **projection suivie d'une re-segmentation** avec utilisation de la carte projetée comme initialisation ou bien comme référence. Ce suivi revient à calculer de 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 vers le temps $t+1$ comme initialisation ou référence du problème de segmentation de l'image I_{t+1}.

Le résultat obtenu peut être considérablement amélioré si l'arrière-plan de la scène est soustrait : l'utilisation de caméras fixes offre un cadre privilégié au suivi. Lorsque la caméra est mobile, des méthodes d'estimations robustes permettent d'estimer le mouvement global afin de se ramener à la situation précédente après compensation. Cependant la modélisation précise d'un mouvement perspectif de la caméra[6] est très ardue, rendant délicate son utilisation.

La segmentation constitue la base de notre pré-analyse. La qualité de celle-ci va directement influer sur l'efficacité de la pré-analyse du résultat final. Cette partie établie donc un listing non-exhaustif des grandes familles de segmentation sur les différentes informations contenues dans une image. Seules les méthodes de segmentation automatique ont été ici présentées, les méthodes de segmentation manuelles ou semi-automatiques, nécessitant l'intervention d'un opérateur humain, n'ont aucun intérêt pour notre application. Le choix d'une technique de segmentation pour une pré-analyse de vidéo doit s'effectuer selon les critères d'environnement et ses contraintes associées. La partie suivante va maintenant présenter la manière dont pourrait être utilisée cette segmentation sur le codage H.264.

[5] Segmentation sur les axes x, y et l'axe temporel.
[6] Représentation 2D d'un mouvement 3D.

2.3 Application de la segmentation au codage H.264

Le problème du codeur de référence H.264 dans une contraire temps réel est qu'il teste tous les modes de prédiction pour chaque nouveau macrobloc à coder mais retient le mode qui optimise la relation débit-distorsion. Le codeur fais donc appel à une charge de calcul excessive pour compenser le fait qu'il ne dispose d'aucune visibilité intelligible de la scène. De plus, cette technique ne lui permet pas de coder un même objet spatio-temporel de façon stable. Ainsi, l'estimation de mouvement peut représenter de 60% (une seule image référence) à 80% (cinq images références) des temps de calcul[7], alors que le codage qui en résulte n'assure aucune cohérence avec le contenu spatio-temporel de la scène. L'application du codage dynamique (ou adaptatif) sur le codage H.264 permet d'effectuer une pré-analyse d'une vidéo afin de pouvoir régler de manière continue les meilleurs paramètres d'encodage. Dans l'optique de réduire ces temps de calcul et de donner au codeur une approche cohérente du contenu de la séquence vidéo, trois exemples de stratégies d'optimisation de codage, à l'aide de la décomposition spatio-temporelle donnée par notre outil de pré-analyse du flux vidéo vont être présentés à savoir :

- choix judicieux du paramètre de quantification,
- choix des modes,
- choix des images références.

Le choix du mode de codage en fonction du contenu ne sera pas aborder dans ce rapport. Il existe de nombreuses possilités d'optimisation du codeur en utilisant le principe de pré-analyse et une utilisation efficace de la pré-analyse nécessiterait une recherche approfondie. Les trois exemples décrits dans les sections suivantes sont seulement ici pour avoir un premier aperçu sur ces possibilités.

2.3.1 Choix du mode de prédiction

L'objectif d'un modèle temporel est de réduire la redondance entre les images en formant une image prédite et en la soustrayant à l'image courante. La sortie de ce traitement est une image résiduelle. Plus la prédiction de l'image courante est précise, moins l'image résiduelle contient d'énergie et donc, moins le débit nécessaire à sa transmission est important. Or le choix du mode de codage temporel est réalisé qu'avec une vision très partielle du contenu spatio-temporel de la séquence vidéo et n'assure pas de stabilité pour le codage d'un même objet au cours du temps. C'est l'un des aspects que nous souhaitons modifier en utilisant notre outil de pré-analyse.

Le codeur H.264 ne dispose pas d'une vue d'ensemble de la scène et fait donc souvent des choix inadaptés, dictés à court-terme et basés uniquement sur des critères débit-distorsion. La pré-analyse pourrait guider les choix du codeur en lui apportant une approche plus intelligente du contenu spatio-temporel la séquence vidéo. Chaque objet de la séquence doit être caractérisé en terme de cycle de vie, de mouvement, de couleur et de texture, ces informations pourraient être exploitées pour influencer le codeur quant à ses choix sur la taille des partitions et le mode de codage d'un macrobloc. Le cycle de vie d'un objet renseigne sur l'apparition et la disparition de ce dernier dans la séquence. Le codeur utiliserait donc un mode intra pour coder les macroblocs

[7] Cela peut augmenter significativement si on utilise une fenêtre de recherche plus importante.

de l'objet dans les images où ce dernier apparaîtrait ou disparaîtrait, et utiliserait un mode inter dans les autres cas. L"information de texture disponible pour chaque objet, pourrait orienter le codeur sur le choix de la taille des partitions : des petites partitions permettrait de coder plus efficacement des zones fortement détaillées, alors que des partitions plus grossières suffiraient pour coder convenablement des zones homogènes. Enfin, grâce au suivi des objets réalisé à l'étape de traitement inter-segment, un même mode de codage (intra ou inter, taille de partition) serait utilisé pendant toute la représentation temporelle d'un même objet. Cette méthode permettrait, d'une part, d'alléger fortement la charge de calcul du codeur en lui évitant de tester tous les modes de codage pour chaque nouvelle image à coder et, d'autre part, d'éviter à l'œil d'un observateur d'être dérangé par des changements de qualité visuelle lorsqu'il suit les déplacements d'un même objet au cours du temps.

Le pré-traitement de la vidéo peut donc apporter d'énormes avantages dans la perspective des choix de mode de codage. En effet, l'utilisation des informations spatio-temporelles (calculées lors des traitements intra ou inter-segment) va fournir au codeur, qui ne disposait jusque là d'aucune visibilité moyen ou long-terme du contenu de la scène, une approche intelligente pour le codage de la séquence vidéo. Grâce à cette nouvelle approche, le codeur ne réalisera plus des tests exhaustifs pour chaque nouvelle image à coder.

2.3.2 Choix du paramètre de quantification

Dans la chaîne de codage (voir figure 1.1), après la phase de prédiction et de compensation du bloc courant, le bloc résiduel D_n est transformé par une transformation 4 X 4 ou 8 X 8. Cette transformation est une transformation entière[8] basée sur une forme modifiée de la Transformée en Cosinus Discrète (TCD). Cette transformation fournit un ensemble de coefficients qui une fois combinés, recréé le bloc résiduel original. La figure 2.3 montre comment la transformée inverse de la TCD crée un bloc en pondérant, et en combinant les blocs (i.e. les motifs) de base. La sortie de la TCD, un bloc de coefficients transformés, est quantifiée, c'est-à-dire que chaque coefficient est divisé par une valeur entière. La quantification réduit la précision des coefficients TCD selon un paramètre de quantification (QP). Typiquement, le résultat est un bloc comportant de nombreux coefficients quantifiés à zéro. Fixer un QP élevé signifie que la plupart des coefficients sont mis à zéro, ce qui entraîne une forte compression mais une faible qualité de l'image décodée. Inversement, fixer un QP à une faible valeur signifie que de nombreux coefficients seront non-nuls après quantification, ce qui entraîne cette fois une meilleure qualité des images reconstruites mais une faible compression.

Néanmoins, après transformation et quantification d'une image, la qualité de l'image reconstruite ne dépend pas uniquement du paramètre QP. En effet, pour deux images différentes auxquelles on associe le même paramètre de quantification, la qualité de reconstruction varie fortement en fonction du contenu spatial (texture) de l'image originale. Par exemple, la figure 2.4 présente les cas d'une image uniforme et d'une image avec une forte activité spatiale. On observe qu'avec un même paramètre de quantification QP, la qualité de l'image reconstruite dans le cas homogène (b) est parfaite, alors que l'image reconstruite dans le cas d'un bloc texturé

[8] C'est-à-dire une transformation de $\mathbb{N}$ vers $\mathbb{N}$.

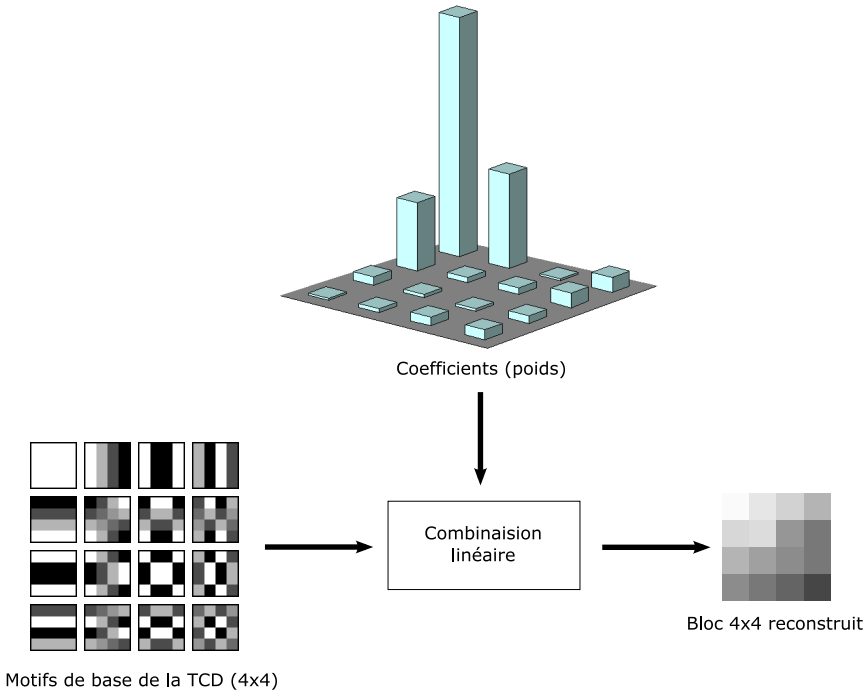

FIG. 2.3 – Transformée inverse : combinaison linéaire des blocs de base pour reconstruire le bloc original.

(a) est dégradée. Ce résultat s'explique par le fait que la TCD concentre principalement l'énergie d'un bloc sur les coefficients dits "basse-fréquence TCD". Les coefficients basse-fréquence TCD représentent les zones homogènes d'un bloc alors que les coefficients haute-fréquence représentent les contours et les textures. Lors de la quantification avec un QP élevé, la plupart des coefficients haute-fréquence sont mis à zéro et donc seuls les blocs au contenu homogène peuvent être reconstruits avec une bonne qualité.

Cette propriété peut être exploitée à l'aide d'une pré-analyse, qui caractériserait l'activité spatiale de chaque objet spatio-temporel. Ainsi, pour les objets homogènes, il serait possible d'indiquer au codeur l'utilisation d'un QP élevé afin de gagner en compression, et dans le cas d'objets à fortes textures, nous pourrons recommander au codeur d'utiliser un QP plus faible afin d'obtenir une meilleure qualité des objets reconstruits en conservant les détails. Des stratégies basées sur des considérations psychovisuelles pourraient également être envisagées afin d'exploiter les caractéristiques du système visuel humain, en quantifiant plus fortement les zones pour lesquelles l'œil est peu sensible aux erreurs. Les paramètres de quantification des différents objets seraient conservés sur plusieurs images consécutives.

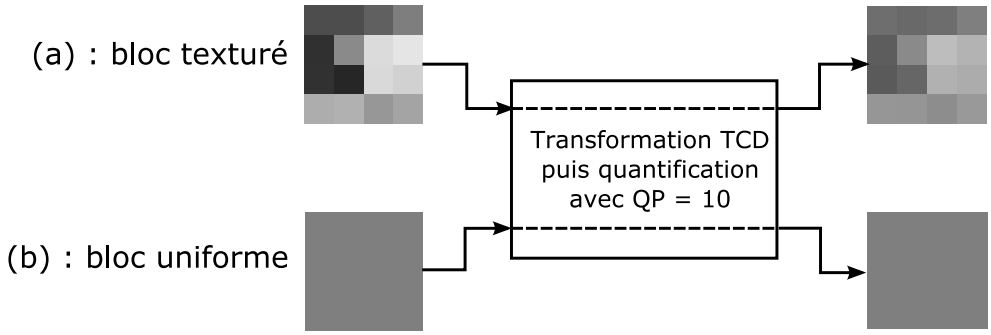

FIG. 2.4 – Comparaison des quantifications pour un bloc texturé et un bloc uniforme.

2.3.3 Choix des images références

Par défaut, le codeur H.264/AVC de référence utilise les cinq images codées précédemment à l'image courante pour remplir la mémoire tampon relative aux images références. Yuan et ses collaborateurs [YFZ03] ont montré l'intérêt du choix des images de référence. En utilisant une méthode adaptée de choix des références, ils ont noté des augmentations de la mesure objective (en terme de PSNR) et de l'évaluation subjective.

Des méthodes de sélection des images références basées sur des similarités d'histogrammes de couleur ont été proposées [NT05]. L'utilisation de ces méthodes pourraient s'avérer appropriée dans le cadre d'une pré-analyse, en segmentant les objets spatio-temporels sur des critères de couleur et de texture (section 4.2.2).

Ces méthodes interviennent directement sur la gestion de la mémoire tampon des images de référence. Le but est de conserver les images clés choisies dans la mémoire tampon et de modifier l'ordre de codage. Il est nécessaire de coder et décoder ces images en premier afin qu'elles puissent être utilisées comme références. Les meilleurs résultats peuvent réduire les temps de calcul de 23% avec également des variations de débit. Ces résultats indiquent donc, qu'avec une gestion intelligente des images références, le codage vidéo H.264/AVC rapide peut être atteint avec une qualité et un débit identique à la méthode de référence.

2.3.4 Conclusion

Le chapitre 1 de ce document permet de comprendre que l'unité de base pour le codeur H.264/AVC est le macrobloc (de taille 16 X 16 allant jusqu'à des partitions de taille 4 X 4). Quelque soit le contenu de la vidéo, le but de la prédiction est de minimiser l'erreur sur chacun de ces macroblocs. Or pour les plans de la vidéo qui contiennent les mêmes objets, il est indispensable de conserver une uniformité et une cohérence dans les modes de codage choisis pour les objets contenus dans la vidéo. Le codeur H.264/AVC en lui même ne possède aucun outil lui permettant d'obtenir des informations sur le contenu de la vidéo. En effet, si l'on souhaite réaliser un codage fonction des objets présents dans la vidéo, il est nécessaire de développer une méthode d'analyse en amont du codeur pour lui transmettre un jeu de paramètres qui soit cohérent avec le contenu spatio-temporel de la séquence vidéo. Ce jeu de paramètres repose sur l'utilisation

de plusieurs familles de méthodes d'optimisation. Toutes ces méthodes peuvent être mises en oeuvre à partir des grandeurs et des caractéristiques des objets, qui seront calculées lors d'une phases de segmentation. L'utilisation des données fournies par cette classification permettra au codeur H.264 d'effectuer un codage cohérent avec l'analyse de la séquence vidéo, les temps de calcul devraient donc diminuer sans engendrer de perte notable de qualité visuelle.

Maintenant que tous les outils nécessaires à la pré-analyse ont été présentés, la suite de ce rapport va maintenant détailler la réalisation du bloc de pré-analyse. La partie suivante va tout d'abord présenter les spécifications de l'outil à produire ainsi qu'une rapide présentation de la première pré-analyse développée par l'équipe du projet ArchiPEG. Cette première pré-analyse est basée sur l'utilisation de tubes spatio-temporels. Elle ne sera donc effectuée que sur des informations mouvement. Comme nous l'avons vu dans cette partie, l'utilisation de critère de textures et de couleurs se révélent être un facteur déterminant pour l'optimisation. La dernière partie de ce rapport sera consacrée au développement d'algorithmes pour une nouvelle pré-analyse devant prendre en compte les critères de textures et de couleurs.

Chapitre 3

Spécification de l'outil de pré-analyse

L'objectif de ce projet est donc de fournir au codeur H.264 un jeu de paramètres adapté au codage d'une séquence vidéo et présentant une cohérence spatio-temporelle fonction des objets présents dans la scène. Cette partie du rapport présente les spécifications fonctionnelles de l'outil de pré-analyse pour le conditionnement du flux vidéo puis détaille l'analyse par tubes spatio-temporels développés par le laboratoire IVC.

3.1 Spécification de l'outil de pré-analyse et de conditionnement du flux vidéo

Un système de pré-traitement doit être positionné en amont du codeur afin de conditionner le flux et de fournir au codeur un ensemble de paramètres adapté à la vidéo. Idéalement, cet outil devra fournir au codeur les informations nécessaires pour réduire considérablement le nombre de modes testés et indiquer les images qui doivent être marquées comme références. Ce chapitre présente dans un premier temps les spécifications externes du système à concevoir, en définissant l'outil de pré-traitement par rapport à son environnement, et dans un second temps, la décomposition interne du système envisagé.

3.1.1 Spécification externe de l'outil de pré-analyse

La définition de l'environnement du système de pré-analyse et de conditionnement du flux vidéo est simple, elle est en fait limitée à deux entités : l'utilisateur qui à l'aide d'une interface enverra en entrée un flux vidéo à l'outil de pré-traitement et le codeur qui recevra les informations de codage synthétisées après analyse de ce flux. L'outil et son environnement sont présentés sur 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. Cet outil est supposé intégré à la partie 'Interface utilisateur' et sa conception n'est pas inclue dans ce projet.

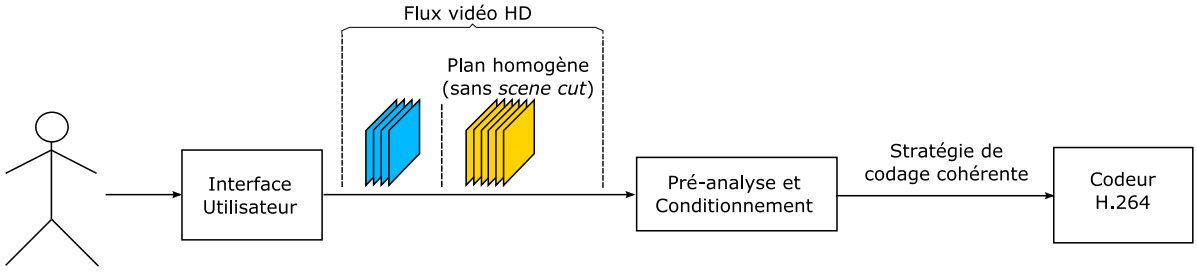

FIG. 3.1 – Spécification externe de l'outil de pré-analyse et de conditionnement d'un flux vidéo.

Après avoir définit le comportement de l'outil de pré-analyse vis-à-vis de son environnement, il convient de définir son comportement interne.

3.1.2 Spécification interne de l'outil de pré-analyse

Afin d'appréhender de façon juste le mouvement des objets et leur ancrage temporel, l'analyse doit porter sur un fenêtre temporelle suffisamment large. Pour fixer la taille de cette fenêtre temporelle, le système se base sur le temps de fixation du système visuel humain qui est sensiblement égal à 200ms [LM05]. Comme la prochaine génération de TVHD utilisera une définition de 1920×1080 pixels en mode progressif et une cadence de 50 images par seconde, le plan homogène d'images en entrée sera découpé en segments temporels de 9 images, chaque segment temporel représentera ainsi 180ms de vidéo.

Il s'agit alors de déterminer les différents objets spatio-temporels qui composent chaque segment. Pour cela, le segment temporel courant bénéficiera d'un traitement intra suivi d'un traitement inter[1] afin d'exploiter la corrélation temporelle susceptible d'exister entre deux segments temporels successifs. Le traitement inter permettra également de suivre un objet sur plus de 180ms (plusieurs segments) et d'envoyer au codeur H.264 des paramètres pour traiter cet objet de façon cohérente temporellement (e.g. éviter des phénomènes de battement). Le système devra être capable de fournir, pour chaque segment temporel, une carte de segmentation basée sur une description détaillée des objets présents dans la scène : délimitation spatiale, suivi temporel, couleur, texture, et cycle de vie. Ces informations pourront alors être transmises à une fonction de classification qui déterminera, pour chaque objet, les paramètres du codeur H.264 les mieux adaptés à sa compression.

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

Dans un premier temps, le bloc de traitement intra effectuera une estimation de mouvement long terme sur un segment temporel de 9 images. À partir des informations de mouvement déduites de cette estimation long-terme, une segmentation basée mouvement sera effectuée. Les

[1] Sous-entendu traitement intra-segment et traitement inter-segment.

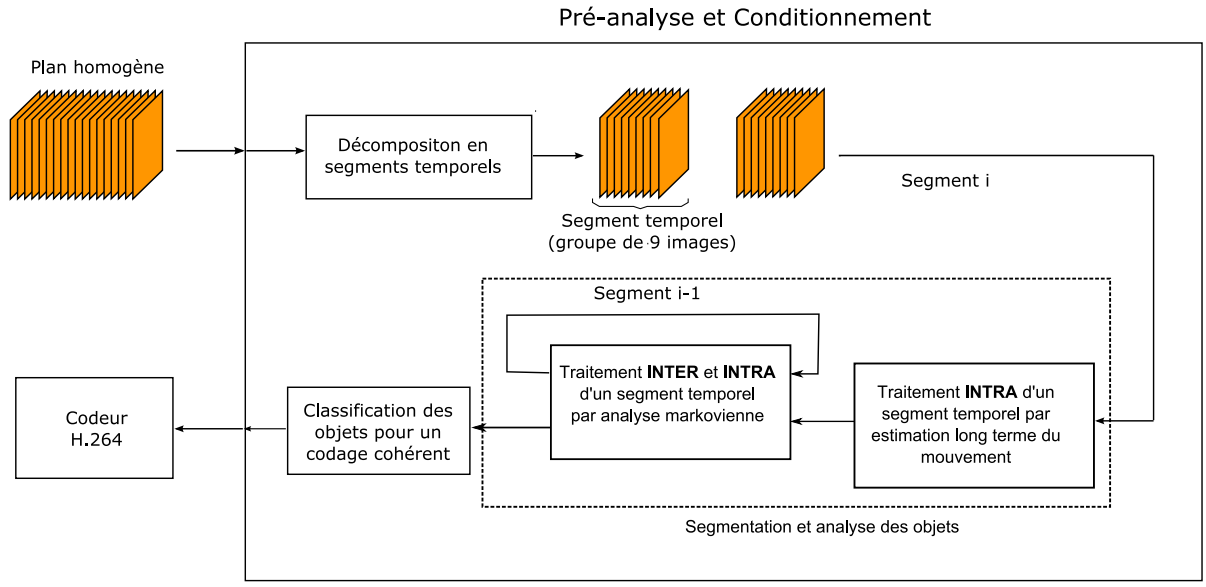

FIG. 3.2 – Conception détaillée de l'outil de pré-analyse et de conditionnement d'un flux vidéo.

résultats fournis par la segmentation basée mouvement sont ensuite affinés par approche markovienne en utilisant conjointement des critères de couleur, de texture et de corrélation avec les segments précédents (traitement inter). Ce mémoire n'inclut pas la réalisation du bloc de traitement intra correspondant à l'estimation long terme de mouvement, celui-ci ayant déjà été développé par l'équipe de recherche ArchiPEG. Cependant, la compréhension de son fonctionnement est indispensable pour une bonne implantation du bloc d'analyse suivant. La suite de ce rapport détaille donc le fonctionnement de l'estimation long terme du mouvement.

3.2 Fonctionnement de l'estimation long terme du mouvement

Le bloc d'estimation long terme du mouvement correspond à un traitement intra-segment. Il doit être capable, pour chaque segment temporel de 9 images, de fournir une décomposition en objets homogènes selon des critères de mouvement. L'objectif de ce bloc est donc de réaliser une segmentation spatio-temporelle d'un segment d'environ 180ms. Les approches couramment utilisées en segmentation vidéo se limitent à l'utilisation de deux images successives. Certaines ambiguïtés sont alors impossibles à résoudre. En effet, les zones de recouvrements (ou découvrements) sont difficilement attribuables à une région ou un objet vidéo. De plus, la distinction de régions ou d'objets par le mouvement est difficile lorsque les mouvements sont similaires.

Il est alors nécessaire de réaliser une segmentation en se plaçant dans un contexte de "mouvement long-terme", on ne se limite plus à seulement deux images successives. La stabilité et

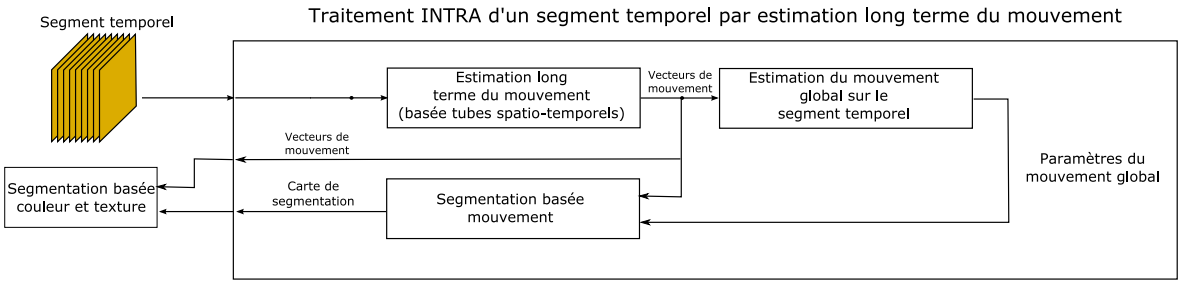

FIG. 3.3 – Traitement par estimation long terme du mouvement d'un segment temporel de 9 images.

la robustesse des résultats de la segmentation sont alors améliorées. Pour ces techniques basées long-terme, le système cherche à obtenir des **tubes spatio-temporels**, c'est-à-dire, des régions ou des objets qui a un mouvement homogène et stable sur plusieurs images, de manière à lisser les mouvements estimés et à obtenir des informations plus corrélées avec les mouvements réels au sein de la séquence vidéo. Dans notre cas, nous considérons que l'unité élémentaire à suivre temporellement est le macrobloc[2], un objet spatio-temporel sera donc composé d'un ensemble de tubes dont les propriétés de mouvement sont homogènes.

À partir des informations de mouvement déduites de l'estimation long-terme, une segmentation basée mouvement sera effectuée. Pour que cette segmentation ne soit pas biaisée par des mouvements particuliers de caméra lors de la prise de vue (e.g. zoom, rotation), une estimation et une compensation du mouvement global sont réalisées sur le segment temporel traité. La figure 3.3 présente la décomposition en schéma bloc du traitement intra d'un segment temporel. Le masque généré par la segmentation ainsi que les tubes résultants de l'estimation long terme sont finalement transmis au prochain bloc d'analyse afin d'obtenir une segmentation du segment courant sur d'autres critères que le mouvement.

La prochaine section va présenter de façon approfondie, les trois méthodes majeures qui constituent le bloc de traitement d'un segment.

3.2.1 Problématique de l'estimation long terme sur une séquence vidéo HD

Afin d'être cohérent avec les techniques utilisées par le standard H.264/AVC, des méthodes d'estimation de mouvement basées sur le *block-matching* sont utilisées dans l'objectif d'appareiller chaque macrobloc de l'image courante avec un bloc d'une image de référence. Le mouvement estimé par macrobloc est alors le déplacement mesuré entre le macrobloc courant et son correspondant dans l'image de référence. Dans une séquence vidéo, l'image courante à coder (dans un but de compression) est souvent fortement corrélée avec l'image qui la précède (ou qui la suit) immédiatement. Les mouvements d'un macrobloc de l'image courante à l'image de référence sont donc relativement faibles et, usuellement, la fenêtre de recherche d'un estimateur de

[2]Bloc de 16×16 pixels.

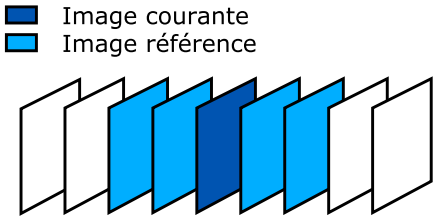

FIG. 3.4 – Image courante et images références d'un segment temporel (contexte court-terme).

mouvement est donc centrée sur la position du macrobloc de l'image courante à prédire. Cependant, dans un contexte d'estimation long-terme, les macroblocs de l'image courante, cherchés dans l'image de référence long-terme, peuvent avoir subi des déplacements très importants. Si la fenêtre de recherche est toujours centrée sur la position du bloc courant, le mouvement obtenu peut correspondre à un minimum local. Pour éviter ce phénomène, la taille de la fenêtre de recherche est agrandie, cependant cette méthode est généralement rédhibitoire, car elle alourdit significativement la charge de calculs. Une méthode alternative, moins coûteuse en calculs, est de conserver les dimensions de la fenêtre de recherche et de choisir un point d'initialisation adapté[3] pour trouver la meilleure prédiction du bloc courant dans l'image de référence long terme.

3.2.2 Estimation de mouvement multi-résolution

Une estimation de mouvement est réalisée sur des images basse-résolution par une méthode multi-résolution (méthode détaillée en partie 1.7). Pour initialiser la recherche des mouvements sur un segment temporel entier, des tubes spatio-temporels seront construits sur cinq images successives (figure 3.4). L'image centrale du segment temporel constitue l'image courante à estimer, elle possède quatre images références : les deux images qui la précèdent et les deux images qui la suivent temporellement. Chaque macrobloc de l'image courante sera considéré comme possèdant un mouvement uniforme[4] entre les cinq images considérées (figure 3.5).

3.2.3 Méthode d'estimation long-terme de mouvement appliquée

L'objectif est d'améliorer les performances d'une estimation long-terme de mouvement classique, en cherchant un point d'initialisation optimal. Une méthode d'estimation du mouvement, dans un contexte long-terme, utilise deux estimations successives. La première est une estimation court-terme multi-résolution, où l'appareillement de macrobloc ne s'effectue plus sur deux images, mais sur cinq. Cette approche par tubes spatio-temporels permet de lisser temporellement les vecteurs déplacement obtenus afin de choisir de façon optimisée un point initial de recherche sur une image référence long terme (figure 3.6). Le déplacement long terme est alors estimé autour de ce point initial à l'aide d'une méthode de minimisation rapide de fonction multi-

[3] Dans un contexte "court-terme", le point d'initialisation est obtenu implicitement pour un mouvement nul entre l'image courante et l'image de référence.

[4] Un macrobloc ayant une vitesse constante (accélération nulle).

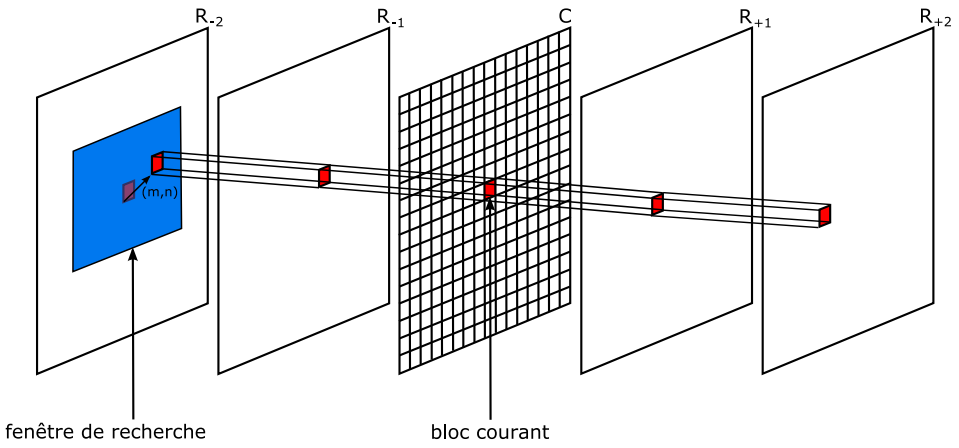

FIG. 3.5 – Représentation d'un tube spatio-temporel et du vecteur mouvement associé.

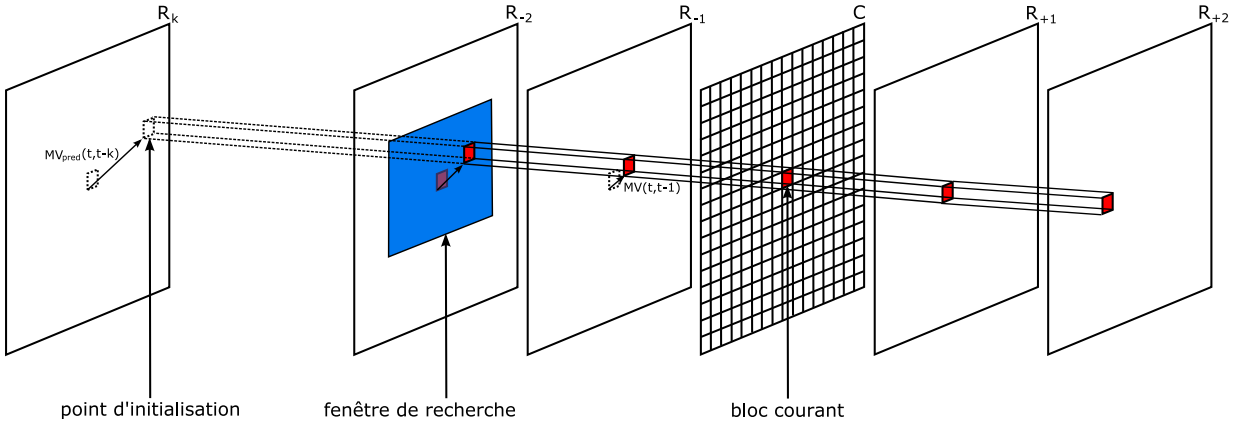

FIG. 3.6 – Initialisation de l'estimation à long terme.

dimensionnelle. Á partir du champ de vecteurs déplacement calculé dans un contexte long-terme, il est maintenant possible d'estimer le mouvement global de la caméra sur un segment temporel.

3.2.4 Estimation et compensation du mouvement global

Dans une séquence vidéo, les déplacements apparents peuvent être dus soit aux mouvements des objets de la scène, soit à celui de la caméra. Afin d'effectuer une segmentation basée uniquement sur les déplacements d'objets physiques, il est nécessaire d'estimer et compenser le mouvement de la caméra. Une segmentation basée mouvement sans cette compensation est possible, mais le résultat obtenu risque d'être fortement biaisé par un mouvement particulier de la caméra lors de la prise de vue. Par exemple, dans le cas critique d'un plan fixe sur lequel la caméra effectue un zoom, le champ de vecteurs déplacement obtenu sera épars (figure 3.7). La segmentation à partir de ces vecteurs mènera donc à la détection de nombreux objets de mouvements différents.

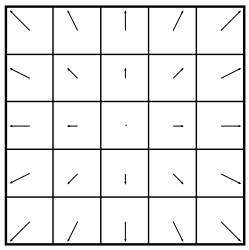

FIG. 3.7 – Champ épars de vecteurs associé à un zoom sur une image décomposée en macroblocs.

Ce qui est appelé mouvement global est en théorie le mouvement engendré par le déplacement de la caméra. En plus des difficultés de modélisation, il est impossible d'estimer le mouvement réel de la caméra en trois dimensions avec uniquement une représentation en deux dimensions de la scène. Finalement, ce qui sera estimé correspond aux déplacements de ce qui est considéré comme le fond la scène. Tous les mouvements engendrés par les déplacements des objets sont considérés comme des mouvements locaux. Cependant, dans certains cas particuliers où un objet occupe une part prépondérante du champ de vision, le mouvement global détecté sera celui de cet objet et non pas celui du fond de la scène.

Au final, les vecteurs compensés par le mouvement de caméra représenteront uniquement les déplacement locaux. Il est ainsi possible d'effectuer une segmentation des objets au sens du mouvement plus efficace.

3.2.5 Segmentation au sens du mouvement

Dans notre schéma, les vecteurs compensés, après compensation globale, représentent donc théoriquement les seuls mouvements de translation de la caméra et les déplacements locaux des objets. Ces vecteurs sont alors accumulés dans un histogramme à deux entrées correspondant aux deux composantes V_x et V_y. En localisant le maximum de l'histogramme d'accumulation des vecteurs compensés, il devient possible de localiser le mouvement global de la caméra. Si nous n'étudions plus uniquement le pic le plus important mais tous les pics, alors une segmentation au sens du mouvement, en plus de l'estimation du mouvement global, aura été effectuée avec l'hypothèse que chaque pic représente le mouvement d'un objet.

Afin de ne pas utiliser une méthode de segmentation trop coûteuse en termes de complexité de calcul, un algorithme récursif va traiter les pics par ordre décroissant. Le premier pic détecté sera donc celui correspondant au maximum global de l'espace d'accumulation. Pour toutes les positions connexes à ce pic, le gradient[5] en direction du maximum est calculé. Tant que le gradient est positif, la position testée est considérée comme appartenant au pic 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 pic principal ont été marquées. Un nouveau maximum est détecté parmi toutes les cellules non marquées et l'algorithme est réitéré

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

tant qu'il reste des cellules non nulles n'appartenant à aucun pic. Au final, une cellule peut être marquée comme appartenant à plusieurs pics. Dans ce cas, elle est définitivement rattachée au pic dont la position du maximum est la plus proche. La figure 3.8 présente la séparation des pics 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 pics sont détectés, le pic majoritaire représente le mouvement global du fond, et le second pic représente le mouvement local du personnage.

La dernière étape consiste à segmenter le champ de vecteurs compensés par les paramètres de déformation, à partir de la séparation des différents pics. L'espace d'accumulation segmenté devient un tableau à deux entrées : les deux composantes de chaque vecteur déplacement compensé de l'image sont les entrées. Le contenu du tableau correspond alors au label qu'il faut donner au macrobloc associé au vecteur. L'image présentée en figure 3.9 est alors segmentée à partir des deux pics présentés en figure 3.8. La zone rouge correspond aux macroblocs dont les vecteurs déplacement appartiennent au pic principal (mouvement global), tandis que la zone bleue correspond aux macroblocs dont les vecteurs appartiennent au second pic (mouvement local). La segmentation au sens du mouvement réalisée ici donne des résultats encourageants, cohérents avec la segmentation qu'effectuerait un œil 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 assez texturé pour que les vecteurs déplacement calculés soient représentatifs des mouvements réels. Inversement, la séquence *New Mobil & 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 3.10). 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).

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.

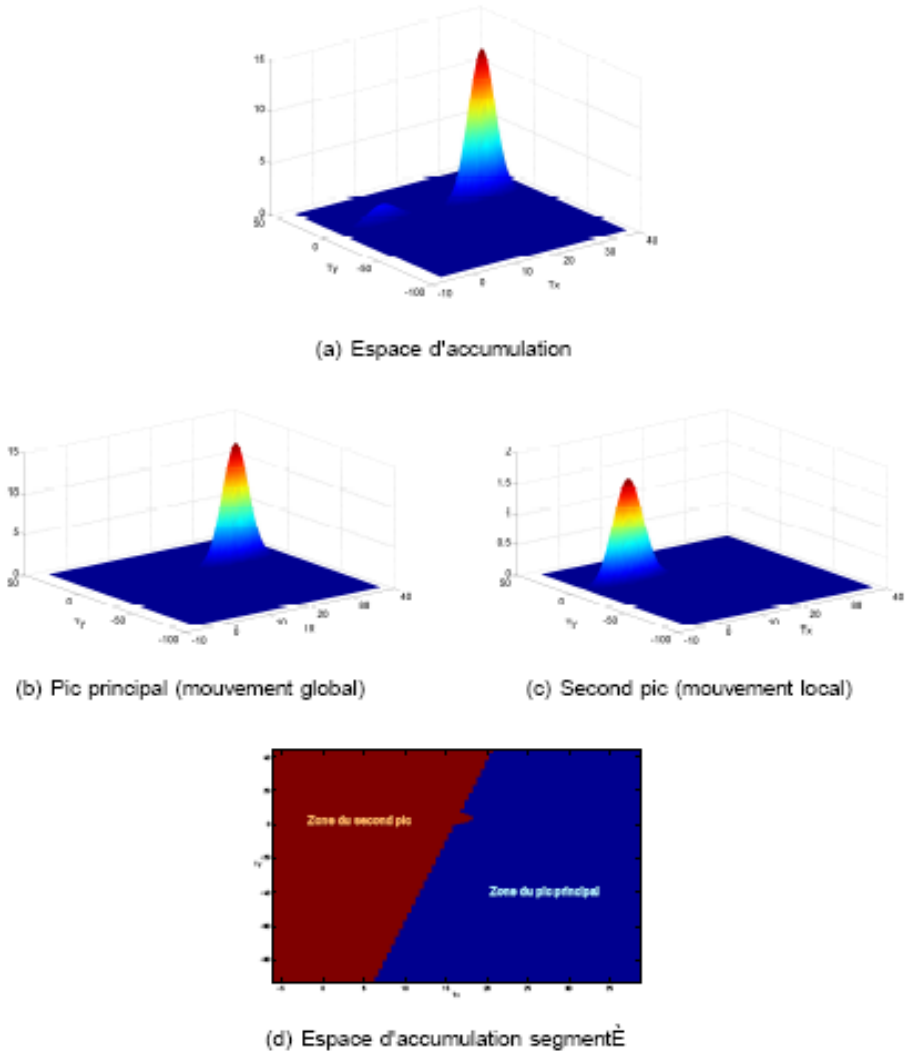

FIG. 3.8 – Analyse récursive de l'espace d'accumulation.

FIG. 3.9 – Image segmentée de la séquence *Knightshields*

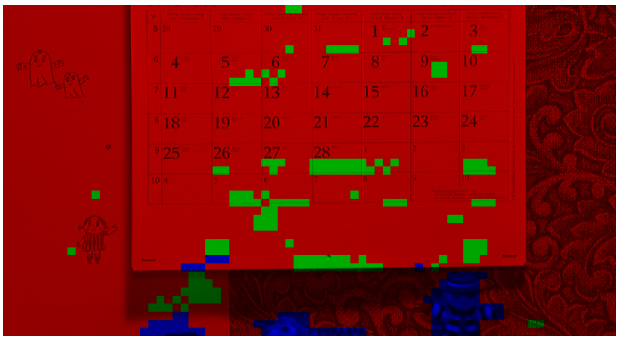

FIG. 3.10 – Image segmentée de la séquence *New Mobil & Calendar*

3.3 Conclusion

Ce chapitre est une liaison entre la théorie sur la pré-analyse et sa réalisation. Il nous a permis de comprendre comment mettre en application la pré-analyse d'un codeur H.264 en définissant son environnement et son comportement interne. Grâce à une analyse de l'existant, nous avons également pu voir que la réalisation à fournir ne concerne pas l'ensemble de l'outil de la pré-analyse pour le codeur H.264. Ce mémoire ne concerne que la partie de traitement correspondant à une prise en compte conjointe de critères supplémentaires au seul critère de mouvement développé par le laboratoire. La description du premier outil de traitement développé par le laboratoire a été nécessaire afin de comprendre les entrées du second bloc de traitement et de quelle manière la segmentation peut être améliorée. Tous les éléments sont maintenant définis pour pouvoir implanter les nouveaux algorithmes de traitement au sein du codeur. La suite de ce mémoire concerne donc le travail effectué au cours de ces six mois de stage à savoir la mise en application des traitements par des méthodes de fusion de critères de mouvement, texture et couleur, et les résultats de l'algorithme développé.

Chapitre 4

Segmentation par approche markovienne

Comme il a été présenté dans la section précédente, une segmentation au sens du mouvement seule peut ne pas suffire pour créer la décomposition cohérente d'un segment en objets spatio-temporels. En effet, pour certaines vidéos avec des mouvements de caméra complexes (zoom ou rotation) et des contenus spatiaux uniformes, les vecteurs déplacement calculés ne reflètent pas suffisamment les mouvements réels des objets et ne sont pas assez précis ou fiable pour être rattachés efficacement à l'un des objets détectés par la segmentation au sens du mouvement (figure 3.10). Dans ces cas particuliers, des critères de texture et de couleur permettraient d'assigner une étiquette à des objets dont le vecteur déplacement ne peut être rattaché à aucun pic dans l'espace d'accumulation des vecteurs.

Ces critères de texture et de couleur doivent également être calculés pour les objets dont la segmentation au sens du mouvement est cohérente. En effet, en plus d'être caractérisé par une information de mouvement, chaque objet sera également défini par son contenu spatial. Ce traitement correspond à une analyse intra du segment, celle-ci doit être couplée à un traitement inter permettant d'obtenir plus de cohérence dans l'analyse avec les segments précédent. Cette cohérence par traitement inter doit passer par :

- un bon suivi des étiquettes des objets [1] tout au long de la séquence.
- l'utilisation du masque précédent afin d'avoir un maximum de probabilité de garder une forme stable des objets au cours de la séquence. Cette étape passe par le calcul d'un masque théorique par projection du masque du segment précédent avec leurs vecteurs correspondants. Ce masque théorique sera utilisé comme un critère supplémentaire à celui du mouvement, de la texture et de la couleur.

Les spécificités exprimées ci-dessus nous ont menés à décomposer le système de traitement par approche markovienne en un ensemble de fonctions, agencées les unes avec les autres selon le schéma bloc présenté en figure 4.1.

La suite de ce mémoire détaille le fonctionnement de chacun des blocs constituant le système de traitement par aproche markovienne. Il convient dans un premier temps de bien comprendre le fonctionnement de l'approche markovienne avant de l'intégrer à l'outil de pré-analyse.

[1] sous-entendu que les objets gardent le même numéro d'étiquette tout au long de leur apparition dans les segments

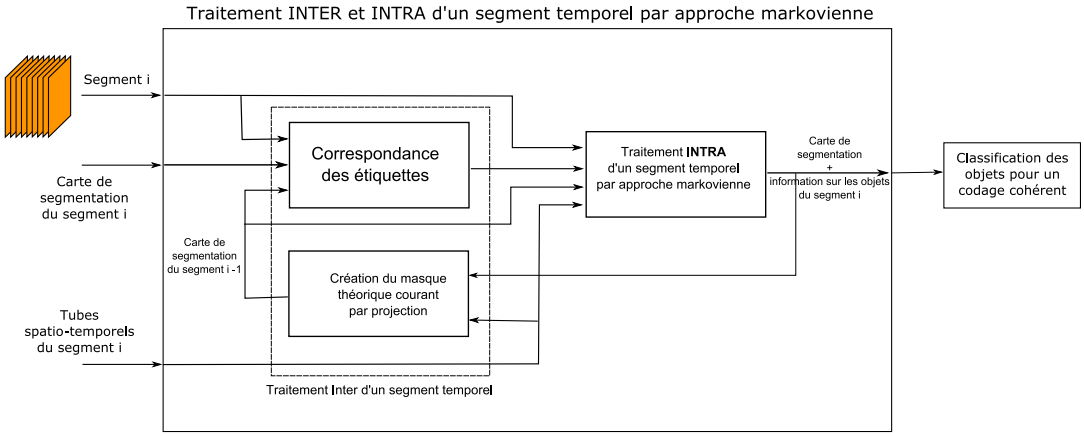

FIG. 4.1 – Bloc de traitement INTER et INTRA du segment i par approche markovienne

4.1 Segmentation par approche Markovienne

Nous avons vu en 2.2.5 que les approches stastistiques sont couramment utilisées pour aborder la construction des masques des objets. Comme le type de connaissances *a priori* que l'on veut inclure s'exprime principalement en termes de contextes spatial (intra) et temporel (vecteurs mouvement + inter), l'utilisation de modèles markoviens semble particulièrement judicieuse.

Il nous faut donc maintenant parler d'observations (reliées aux données disponibles) et de primitives (fournies par la carte de segmentation originale). La décision prise au niveau d'un site ne se fera pas uniquement sur la seule observation en ce site ou même sur un voisinage de ce site, mais en tenant compte d'informations *a priori* sur les propriétés en espace et en temps du masque d'objets mobiles. Un modèle, prenant en compte ces propriétés, devra alors être défini.

Un cadre de modélisation statistique bien élaboré permet de plus de prendre en compte différents types de situations (c.à.d diverses informations à priori) en jouant sur les paramètres du modèle de façon maîtrisée.

Enfin, un avantage d'une technique bayésienne d'estimation [Bay63] jointe à une modélisation markovienne est de pouvoir utiliser un critère global sur l'image tout en effectuant que des calculs à un niveau local (le voisinage spatio-temporel). Nous allons présenter la méthodologie utilisée pour résoudre ce problème en rappelant, dans un premier temps, le principe de la modélisation par champ markovien.

4.1.1 Modélisation par champ Markovien

Un champ de Markov (CM) est caractérisé par sa propriété locale tandis qu'un champ de Gibbs (CG) 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 ont permis l'essor de l'utilisation

de tels modèles.

Nous allons dans un premier temps reprendre les principaux aspects mathématiques de ce type de modélisation et d'estimation associée. Les notations suivantes sont adoptées pour la résolution de notre problème :

Soit S le treillis rectangulaire des sites s étant, dans notre cas, un macrobloc, indicé par i, j avec i indice de colonne, j indice de ligne.

Soit $E = \{E(s), s \in S\}$ un champ des étiquettes ou primitives à estimer, indexé par s.

$(E = e)$ désigne l'événement $\{E(s_1) = e(s_1), E(s_2) = e(s_2), ..., E(s_n) = e(s_n)\}$.

$\Lambda = e$ est l'ensemble des réalisations possibles du champ des étiquettes.

$\Omega = \{\omega_i\}$ est l'ensemble des valeurs que peut prendre une étiquette.

Soit enfin $\eta = \{\eta_s, s \in S\}$ une structure de voisinage définie sur s.

Le système de voisinage d'un site (i, j), peut être par exemple, l'ensemble des quatre sites les plus proches soit : $(i-1, j)$,$(i+1, j)$, $(i, j-1)$, $(i, j+1)$.

Le champ E associé au système de voisinage η est alors un champ de Markov si :

– condition de positivité

$$\forall e \in \Lambda, P(E = e) > 0 \tag{4.1}$$

– condition markovienne

$$P(E(s) = e(s)|E(r) = e(r), r \in S, r \neq s) = P(E(s) = e(s)|E(r) = e(r), r \in \eta) \tag{4.2}$$

L'étiquette affectée au site s de S est donc conditionnée uniquement par les étiquettes des voisins de s et non par l'image entière. Cette relation permet donc de passer d'une situation globale à une situation locale.

Un aspect primordial de la modélisation résulte du théorème de Hammersley et Clifford établissant l'équivalence entre les champs markoviens et les distributions de Gibbs [Bes74], il est ainsi possible de manipuler explicitement la distribution *a priori* du champ des étiquettes :

Un champ est markovien relativement à η, si et seulement si il existe une fonction U telle que :

$$P(E = e) = \frac{1}{Z} exp - U(e), \tag{4.3}$$

où Z est une constante de normalisation et U est appelée fonction d'énergie et s'écrit comme la somme de potentiels élémentaires définis sur des structures locales appelées cliques :

$$U(e) = \sum_{c \in C} V_c(e), \tag{4.4}$$

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 étant donnés deux sites quelconques s_i et s_j 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 (figure 4.2). La distribution de probabilité définie par les équations 4.3 et 4.4 est appelée distribution de Gibbs relativement au système de voisinage η.

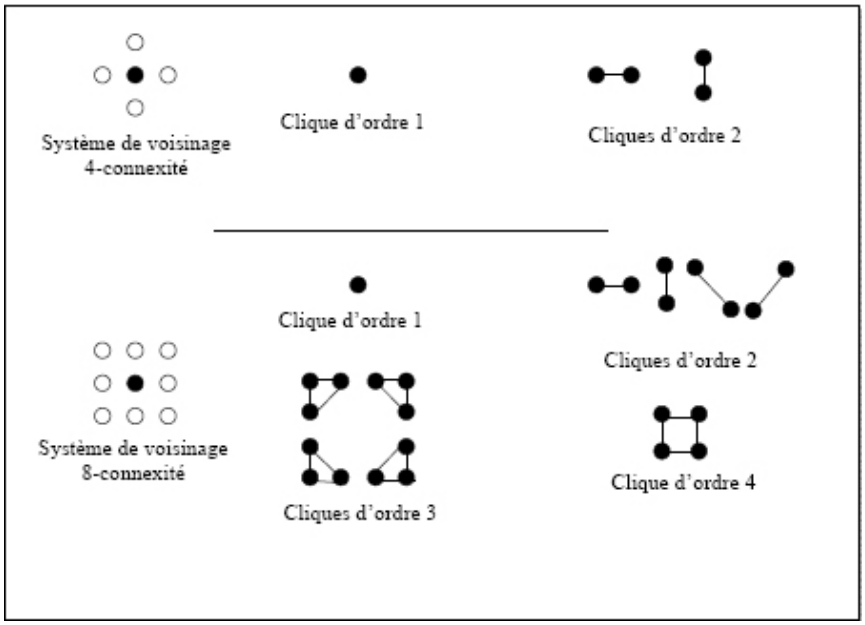

FIG. 4.2 – Cliques associées à un système de voisinage en 4-connexité et en 8-connexité.

Il est important de noter que la configuration des étiquettes e la plus probable est celle ayant l'énergie U la plus faible. Le choix de U repose sur une connaissance globale du problème et des *a priori* sur le résultat désiré.

4.1.2 Mise en forme du problème d'estimation

La formulation complète d'un problème d'analyse d'images à l'aide de champs markoviens oblige à définir précisément certains sites.

Le premier site est de définir à la fois les observations et les primitives ainsi que les relations les liant. La seconde étape est de définir les différents éléments du modèle :

– les interactions locales au niveau des primitives
– toutes les relations liant les primitives aux observations

A ce stade tous les éléments du modèle sont définis. Il reste alors à déterminer le critère de décision et de choisir une politique d'optimisation de ce dernier.

Choix du critère

Le critère le plus souvent retenu est du type MAP, maximisation *a posteriori* de la distribution du champ d'étiquettes E étant donnée la matrice d'observations O :

$$max_e P(E = e|O = o). \tag{4.5}$$

En utilisant la régle de Bayes, on obtient facilement que cela revient à maximiser :

$$max_e \frac{P(O=o|E=e)P(E=e)}{P(O=o)}, \tag{4.6}$$

soit en négligeant le facteur, constant au regard du critère, $P(O=o)$

$$max_e P(O=o|E=e)P(E=e), \tag{4.7}$$

Relation observations/primitives

Pour définir la probabilité conditionnelle des observations, ou vraisemblance des observations, il est nécessaire d'établir la relation qui lie champ d'observations O (mesure physique), et champ des étiquettes E. Cette relation peut être formalisée en termes généraux de la façon suivante :

$$O = \Psi(E) + bruit, \tag{4.8}$$

où Ψ dépend du problème traité. En général, le bruit sera supposé blanc, gaussien, centré et de variance, σ^2.

Si nous ramenons cette équation à la résolution de la pré-analyse du flux, à l'étape actuelle, chaque objet, attribué à une étiquette e, est constitué d'un ensemble de macroblocs dont les mouvements sont proches. Pour agglomérer de nouveaux macroblocs à un objet, ou encore diviser un objet en sous-objets, chaque macrobloc doit être caractérisé par son activité spatiale en plus de son activité temporelle. Par activité spatiale est désigné le contenu en couleur du macrobloc et l'orientation de ses textures.

Pour estimer au mieux cette activité spatiale et temporelle intra-segment, il est alors nécessaire de calculer pour chaque macrobloc les informations suivantes :

- les gradients des textures selon quatre directions : 0˚, 45˚, 90˚et 145˚[PLCCB07]
- la distribution des couleurs.
- le mouvement du macrobloc dans le segment, déterminé par l'étape d'estimation long terme (voir 3.2). Cette information est nécessaire pour conserver un résultat proche de la première segmentation basée mouvement.
- les connexités spatiales et temporelles des étiquettes des macroblocs. Ces informations permettent d'apporter plus de régularité pour la segmentation.

Ces informations sont caractéristiques d'un macrobloc. Les étiquettes devront être caractérisées par le même type d'informations. Les informations sur les gradients de textures, de distributions de couleurs et de mouvements seront des critères globaux à l'étiquettes. Les informations de connexité locale et temporelle seront des caractéristiques locales aux macroblocs.

Les observations O seront définies selon ces quatres critères. Le choix de $E = e$ parmi Λ pour un site s se fera en fonction de l'étiquette ayant les caractéristiques, selon les critères donnés, les plus similaires au macrobloc, c'est-à-dire ayant la probabilité $P(E = e|O = o)$ la plus forte.

4.1.3 Fonctions de potentiel

Le choix de ces fonctions de potentiel dépend principalement des propriétés des champs d'étiquettes que l'on cherche à modéliser. Le choix permet alors de définir complètement, au paramétrage près, l'énergie associée à un champ. Dans le cas de calculs sur des caractéristiques locales de s, l'énergie associée à ce champ est définie par :

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

où C est l'ensemble des cliques c de S relatives au voisinage η.

Différentes politiques peuvent être envisagées pour établir la forme de ces potentiels $Vc(e)$ [Lal90].

Pour notre application, les potentiels seront égaux à des constantes prédéfinies dépendant des configurations examinées :

$$V(e(s)) = \alpha_m \;\; si \;\; e(s) = m \tag{4.9}$$

$$\begin{cases} V(e(s), e(p)) = \beta_{sp} \;\; si \;\; e(s) \neq e(p) \\ V(e(s), e(p)) = -\beta_{sp} \;\; si \;\; e(s) = e(p) \end{cases} \tag{4.10}$$

avec β_{sp} positif et en utilisant uniquement des cliques binaires.

Les paramètre α_m indique le degré de confiance dans la valeur m d'une étiquette. C'est en fait une sorte de probabilité *a priori* des différentes classes.

Les potentiels β_{sp} correspondent aux paramètres d'interaction entre les différents sites.

Les potentiels β_{sp} choisis ici favorisent plutôt la création de zones où les sites voisins auront la même étiquette, autour des zones homogènes. Choisir β_{sp} négatif aurait favorise la création de cliques avec deux sites voisins étiquetés de manière différente, c.à.d des zones fortement inhomogènes (des zones de texture à "'grain fin"' par exemple).

Ce choix de potentiels ne tient compte que de l'égalité ou de l'inégalité des étiquettes de deux sites voisins. Ils seront donc utiles pour mesurer l'énergie locale d'un site en fonction de son entourage.

Optimisation du champ des étiquettes

Le problème central à résoudre est de trouver le champ d'étiquettes E maximisant la probabilité $P(E = e|O = o)$.

De façon équivalente, on peut s'intéresser à la probabilité jointe $P(E = e, O = o)$. Cette probabilité suit en général, elle aussi, une distribution de Gibbs :

$$P(E = e, O = o) = \frac{1}{z} exp^{-U(e,o)}, \tag{4.11}$$

Les critères 4.5 et 4.7 sont équivalents au critère suivant :

$$max_e P(E = e, O = o), \tag{4.12}$$

D'après 4.11, maximiser cette probabilité revient à minimiser l'énergie $U(e, o)$. Ce minimum peut être obtenu en utilisant des méthodes de relaxations soit stochastique soit déterministes. Ces méthodes proviennent de la thèse [Lal90], la suite de ce rapport va tout d'abord détailler les méthodes stochastiques plus rigoureuses mais aussi complexes, pour s'attarder ensuite sur les méthodes déterministes, plus adaptées à notre utilisation.

4.1.4 Méthodes de relaxation stochastique

Les méthodes de relaxation stochastique sont couramment utilisées pour l'optimisation de systèmes complexes. Elles consistent à tirer aléatoirement une nouvelle configuration e suivant une certaine distribution dépendant de la fonction d'énergie à minimiser. La particularité de sa relaxation stochastique est d'autoriser des clan men de configuration même si ceux-ci correspondent à une augmentation provisoire de l'énergie du système. Cette propriété permet d'assurer une couvhérence théorique vers le minimum global d'énergie, et de ne pas rester bloqué dans un minimum local.

Nous allons d'abord présenter deux méthodes de simulation de champs de Gibbs, l'algorithm de Metropolis et l'échantillonneur de Gibbs, sur lesquels reposent ses techniques de relaxation stochastique, de type "recuit simulé".

Algorithme de Metropolis

Soit $E_k = e$ l'état du système à l'itération k. La transition vers l'état $k + 1$ se fait en deux étapes :

- on choisit une nouvelle configuration e' proche de la configuration e : e' ne différe généralement de e qu'en un site s où une nouvelle valeur d'étiquette $e'(s)$ est choisie aléatoirement,
- une décision est alors prise en calculant la variation d'énergie $\Delta U = U(e') - U(e)$
  - si $\Delta U < 0$ alors $E_{k+1} = e'$
  - sinon
    - $E_{k+1} = e'$ avec une probabilité $e^{-\Delta U}$
    - $E_{k+1} = e'$ avec une probabilité $1 - e^{-\Delta U}$

En pratique, une valeur λ est tirée aléatoirement selon une loi uniforme sur $[0, 1]$; E_{k+1} est alors égale à e' si $e^{-\Delta U} < \lambda$ à E_k sinon.

Cette condition permet des changements d'étiquetage entrainant des augmentations provisoires de l'énergie du système.

Cette méthode est générale quelque soit le système considéré. Dans le cas d'images, où le champ à estimer est considéré Markovien et suit une distribution de Gibbs,la diminution de l'énergie sera calcule alors uniquement dans le voisinage du site où se place la modification et noin sur toute l'image.

Echantillonneur de Gibbs

De la même manière, l'échantillonneur de Gibbs fonctionne de manière itérative jusqu'à obtenir la stabilité du système, soit :

– Choisir éatoirement un site s,

– Modifier l'étiquette du point selon la loi conditionnelle : $P(E_{k+1}(s)/E_k(r), r \in \eta_s)$

En pratique, la probabilité :

$$p(E_{k+1}(s) = \omega_i / E_k(r), r \in \eta_s) \tag{4.13}$$

est calculée pour toutes les étiquettes ω_i possibles de Ω. On tire comme précédement un nombre λ compris dans $[0, 1]$; la nouvelle étiquette attribuée à s est ω_j correspondant, au plus petit j tel que :

$$\sum_{k=1}^{j} p_k > \lambda \tag{4.14}$$

Algorithme de recuit simulé

La minimisation globale de la fonction énergie est menée en introduisant un "facteur de température" T : $\frac{U}{T}$. Les méthodes de simulation évoquées ci-dessus sont alors utilisées en faisant décroître la température, (recuit simulé). Les méthodes de recuit ont directement été inspirées par la physique. La procédure d'optimisation est lancée avec une température élevée. De ce fait, le processus autorise de grandes variations de l'énergie,à la fois positive et négative. L'optimisation est obtenue en faisant diminuer progressivement la température. Plus précisément on montre que si la température tend vers zéro à l'infini et qu'à chaque itération k :

$$T_k \geq \frac{\alpha}{\log k} \tag{4.15}$$

où α est une constante, alors la configuration finale est optimale.

Cependant les méthodes stochastiques se trouvent limitées par l'importance des calculs qu'elles nécessitent à la fois pour converger à une température donnée T et pour arriver à une température proche de zéro, (la loi de décroissance théorique s'avérant extrément lente en pratique).

4.1.5 Méthodes de relaxation déterministe

Une méthode simple, connue sous le nom d'ICM [Bes86] (Iterated Conditional Modes), consiste à chercher en chaque site s de l'image, l'étiquette $e(s)$ qui maximise :

$$p(E(s) = e(s) | o(s), e(r), r \in \eta_s) \tag{4.16}$$

Cet algorithme correspond à l'algorithme de recuit simulé à température nulle, et une étiquette ne peut donc être modifiée que si cela entraîne une diminution de l'énergie.

Cette méthode assure une convergence vers le premier minimum d'énergie trouvé qui n'est pas forcément le miminimum global. L'initialisation e^0, liée à la qualité du masque initial, est donc très importante puisqu'elle influera sur le minimum sur lequel le processus va converger.

La procédure d'identification s'effectue alors de la manière suivante :

– choisir un site s ;
– calculer l'énergie U pour toutes les étiquettes ω_i possibles ;
– l'étiquette retenue est celle qui correspond à la plus grande diminution locale de l'énergie, c'est à dire, la plus gande valeur de la probabilité $p(E(s) = e(s)|o(s), e(r), r \in \eta_s)$.

Différentes politiques de choix des sites sont envisageables :

– balayage séquentiel de l'image,
– tirage aléatoire,
– utilisation d'une pile cite d"'instabilité"' [CR87].

Le balayage séquentiel n'est pas toujours une bonne solution puisqu'il entraîne une causalité dans le tirage des sites. Il est préférable d'avoir recours à la dernière solution, la gestion par pile étant une bonne solution dans notre cas où le nombre de sites critiques (c.à.d ceux qui ne sont pas dans un état stable) est faible par rapport à la taille de l'image. Cette hypothèse est admise en utilisant l'*a priori* que la segmentation initiale basée sur l'estimation long terme est fiable. Nous exposerons en détail le fonctionnement de la pile d'instabilité dans la section 4.2.3.

Pour résoudre un problème modélisé à l'aide de champs markoviens, plusieurs options doivent être concrètement définies :

– système de voisinage et jeu de cliques d'intérêt (4.2.2),
– fonctions de potentiels (relation 4.4),
– probabilité condition des observations, (d'après la relation 4.8),
– politique de visite des sites dans la phase d'optimisation 4.2.3.

La suite de ce rapport est donc consacrée à la mise en application des champs markoviens pour le traitement inter et intra segment.

4.2 Traitement intra-segment temporel

À partir de l'estimation globale du mouvement et de la segmentation basée mouvement, on obtient une initialisation e^0 de tous les sites $s \in S$. Où S représente l'image pivot du tube spatio-temporel de 9 images et s représente un bloc 16×16.

La sortie du bloc de traitement intra sera donc une carte de segmentation améliorée grâce à la prise en compte supplémentaire pour chaque objet de la connexité, de la couleur et de la texture par analyse markovienne. Chaque segment aura pour résultat une carte de segmentation, les caractéristiques de chacun des objets selon les critères cités précédemment et leurs mouvements associés.

L'environnement de résolution par champs de markov étant maintenant défini, il est nécessaire d'associer une énergie à chacun de ces critères de façon conforme à l'équation de Hammersley et Clifford 4.3. La résolution de notre problème revient à trouver l'étiquette e minimisant l'énergie U associée au site s.

FIG. 4.3 – Exemple d'initialisation de la segmentation fourni par l'estimation long terme.

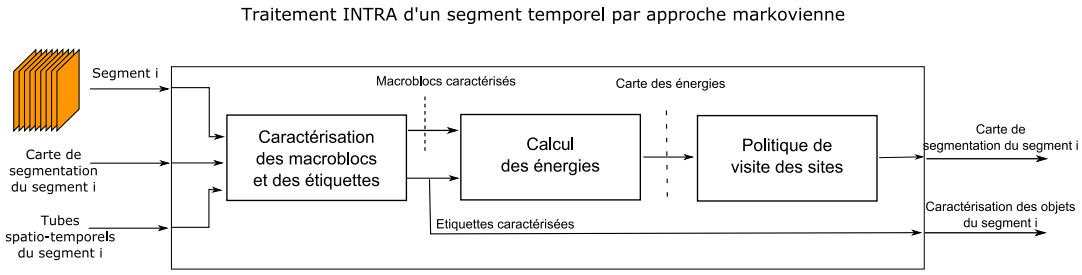

FIG. 4.4 – Décomposation du bloc de traitement INTRA segment par analyse markovienne.

Ce traitement se décompose donc en trois étapes distinctes. La première est de caractériser toutes les étiquettes et tous les macroblocs afin de pouvoir ensuite calculer toutes les énergies entre tous les sites s et les étiquettes de l'image. A ce site du traitement, il est alors possible de fournir une carte d'énergie permettant d'indiquer à la prochaine étape les sites critiques, c'est-à-dire les sites présentant la différence d'énergie la plus importante. La dernière étape décide, à partir de la carte d'énergie, d'une politique de visite de minimisation d'énergie pour chacun des sites. La décomposition du bloc de traitement intra pour chaque segment est illustrée en figure 4.4.

4.2.1 Probabilité condition des observations

Avant de pouvoir déterminer les fonctions de potentiels permettant de lier les observations et les étiquettes, il est nécessaire de calculer l'énergie associant un macrobloc à une étiquette.

Nous cherchons à caractériser un site selon sa connexité, sa couleur, sa texture et son mouvement. Nous définirons cette fonction d'énergie, associant un site s à une étiquette e, par :

$$U(s,e) = \beta_1.W_{cliques}(s,e) + \beta_2.W_{coul}(s,e) + \beta_3.W_{text}(s,e) + \beta_4.W_{mouv}(s,e), \qquad (4.17)$$

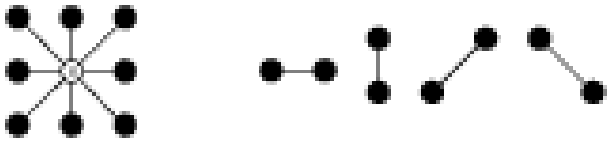

FIG. 4.5 – Voisinage 8-connexes, et les cliques d'ordre 2 associées

avec $W_{cliques}$,W_{coul}, W_{text} et W_{mouv} l'énergie associée à chacun de ces critères et β_1, β_2, β_3 et β_4 des paramètres permettant de donner plus ou moins d'importance aux différentes fonctions de potentiels.

L'énergie, associant chacune des étiquettes e présentes dans l'image à un site s, représente l'ensemble des réalisations Λ possibles du champ des étiquettes.

Maximiser la probabilité que $P(E = e/O = o)$, c'est-à-dire qu'un macrobloc appartienne à un objet, revient à minimiser la fonction d'énergie U associé au site s et à l'étiquette e.

Une décision au niveau d'un site nécessite le calcul de l'énergie entre ce site et toutes les étiquettes présentes dans l'image. Le site sera ensuite marqué par l'étiquette ω_i dont l'énergie correspondante sera la plus faible.

L'énergie d'un site s étant maintenant déterminée, la suite de cette section va détailler les mesures associées à l'obtention de chacunes de ces énergies.

4.2.2 fonctions de potentiels

Afin de déterminer la probabilité d'appartenance d'un macrobloc à un objet, il est nécessaire de pouvoir comparer les caractéristiques globales de l'étiquette e et avec les caractéristiques globales et locales du site s. La mesure des caractéristiques locales du site se passe par le calcul des cliques. Pour les caractéristiques globales, à savoir la couleur et la texture, il faut alors pouvoir déterminer comment mesurer le degré de ressemblance de distribution d'un site s avec une étiquette e en respectant l'équation 4.3 de Hammersley et Clifford.

Le mouvement, qui reste le critère le plus caractéristique d'un objet, ne nécessite pas de calcul sur la distribution. Une méthode différente d'obtention d'énergie lui sera donc associée.

Pour garder une certaine homogénéité dans les critères, nous attacherons de l'importance à ce que chacune des énergies soient bornées entre 0 et 1 (0 pour des macroblocs identiques selon les critères données et 1 pour des macroblocs en tous points différents).

Système de voisinage et jeu de cliques d'intérêt

Les objets en mouvements doivent avoir une cohérence spatiale et une certaine compacité. Les potentiels spatiaux doivent donc favoriser les zones homogènes et éliminer les petites zones parasites. Le calcul des énergies liées aux cliques revient donc à donner plus d'importance aux étiquettes situées au voisinage du site.

Après tests de plusieurs configurations de plusieurs voisinages et de cliques, nous avons déterminé que la meilleure configuration en termes de complexité et de résultat revenait à prendre un voisinage 8-connexes et des cliques d'ordre 2 (figure 4.5).

Soit s_2 un site appartenant au voisinage spatial η_s du site s, le potentiel associé à la clique est défini par :

$$\forall s_2 \in \eta_s \begin{cases} V_{c_s} = & \beta_s \quad si \quad e_t(s) \neq e_t(s_2) \\ V_{c_s} = & -\beta_s \quad si \quad e_t(s) = e_t(s_2) \end{cases} \tag{4.18}$$

avec β_s positif.

Le calcul du potentiel associé revient donc à dénombrer le nombre d'étiquettes présentes sur le voisinage du site :

L'énergie spatiale est alors :

$$W_{spatial} = \sum_{c_s \in C_s} V_{c_s}(e), \tag{4.19}$$

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

L'énergie d'un site s donné liée aux cliques spatiales varie entre $-8\beta_s$ et $+8\beta_s$. Pour obtenir une énergie variant entre 0 et 1, β_s est fixée à $1/16$ et le résultat est incrémenté de 1/2. Maintenant que les caractéristiques globales peuvent être mesurées, il est nécessaire de pouvoir comparer les distributions en couleur et en texture entre un site et une étiquette. Nous allons utiliser pour cela la distance de Bhattacharrya.

Mesure de similarité entre les distributions

On souhaite comparer les distributions de couleurs et de textures d'un site et de ses voisins. Il existe de nombreuses méthodes adaptées au cas discret (intersection, L_2, χ_2, ...). Parmi celles-ci, les évaluations seront mesurées en utilisant le coefficient de Bhattacharrya dont le calcul est donné ci-après.

Les densités discrètes du site courant $\hat{s} = \{\hat{s}_u\}_{u=1...m}$ et des étiquettes voisines $\hat{C}_c = \{\hat{C}_{cu}\}_{u=1...m}$ sont calculées à partir du m-histogramme (histogramme couleur multi-dimensionnel à m classes ou histogramme selon les m-directions de textures). C_c représente l'ensemble des sites voisins de s ($\in \eta_s$) qui portent la même étiquette que s.

Le coefficient de Bhattacharyya est défini par :

$$\rho_c = \rho_c(\hat{C}_c, \hat{s}) = \sum_{u=1}^{m} \sqrt{\hat{C}_{cu}.\hat{s}_u}. \tag{4.20}$$

Le coefficient obtenu est compris entre 0 et 1.

Démonstration :

$$\begin{aligned} \sqrt{\hat{C}_{cu}} - \sqrt{\hat{s}_u} &\geq 0 \\ \hat{C}_{cu} + \hat{s_u} &\geq 2.\sqrt{\hat{C}_{cu}.\hat{s}_u} \\ \textstyle\sum_{u=1}^{m}(\hat{C}_{cu} + \hat{s}_u) &\geq 2.\textstyle\sum_{u=1}^{m}\sqrt{\hat{C}_{cu}.\hat{s}_u} \\ \textstyle\sum_{u=1}^{m}\hat{C}_{cu} + \textstyle\sum_{u=1}^{m}\hat{s}_u &\geq 2.\textstyle\sum_{u=1}^{m}\sqrt{\hat{C}_{cu}.\hat{s}_u} \quad or \quad \textstyle\sum_{u=1}^{m}\hat{C}_{cu} = 1 \;\; et \;\; \textstyle\sum_{u=1}^{m}\hat{s}_u = 1 \\ 1 &\geq \textstyle\sum_{u=1}^{m}\sqrt{\hat{C}_{cu}.\hat{s}_u} \end{aligned} \tag{4.21}$$

À partir de ce coefficient on peut calculer une distance qui sera comprise entre 0 et 1,

$$\begin{aligned} dist &= \sqrt{1 - \sum_{u=1}^{m} \sqrt{\hat{C}_{cu}.\hat{s}_u}} \\ dist &= \sqrt{1 - \rho_c(\hat{C}_c, \hat{s})} \\ dist &= \sqrt{1 - \rho_c} \end{aligned} \tag{4.22}$$

L'énergie liée aux distributions de couleurs et de textures peut alors s'écrire sous la forme :

$$W = \sum_{s \in S} \sqrt{1 - \rho_c(e)} \tag{4.23}$$

$$W = \sum_{s \in S} dist(e) \tag{4.24}$$

Nous considérons donc que deux blocs sont similaires selon le critère donné lorsque la distance *dist* est faible, c'est-à-dire lorsque le coefficient de Bhattacharyya est fort.

La technique de mesure étant définie, il est maintenant nécessaire de créer les histogrammes qui vont caractériser tous les macroblocs et toutes les étiquettes selon les critères de textures, de couleurs et de mouvements.

Caractérisation de la distribution de texture des macrobloc et des étiquettes

Comme nous l'avons vu dans la section précédente, nous cherchons à caractériser la texture d'un macrobloc et de l'ensemble des macroblocs contenus dans une étiquette. Pour cela, quatre distributions selon l'orientation de gradient doivent être calculées :

- le gradient vertical, noté ΔV,
- le gradient horizontal, noté ΔH,
- le gradient diagonal suivant la direction de 45°, noté ΔD_{45},
- le gradient diagonal suivant la direction de 135°, noté ΔD_{135}.

Pour chaque macrobloc et chaque étiquette, une distribution de gradient est alors calculée en fonction des gradients déterminés pour chaque pixel. Pour cela, quatres filtres de Sobel (4.25) vont être appliqués à l'image.

$$\begin{gathered} \overline{\Delta V} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & 2 \\ -1 & 0 & +1 \end{bmatrix}, \quad \overline{\Delta H} = \begin{bmatrix} -1 & 2 & +1 \\ 0 & 0 & 0 \\ -1 & 2 & +1 \end{bmatrix}, \\ \overline{\Delta D_{45}} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & +1 \\ 0 & +1 & +2 \end{bmatrix}, \quad \overline{\Delta D_{135}} = \begin{bmatrix} 0 & +1 & 2 \\ -1 & 0 & +1 \\ -2 & -1 & 0 \end{bmatrix}. \end{gathered} \tag{4.25}$$

Les résultats du gradient sur la luminance de l'image peut aussi être transformés en une image en niveaux de gris qui illustre les différences locales de luminosité des pixels (figure 4.6). C'est une approximation de la norme de la dérivée première (selon les directions) de la luminosité en

FIG. 4.6 – Application du filtre de Sobel sur l'image de direction respective : 0˚, 90˚, 45˚et 145˚.

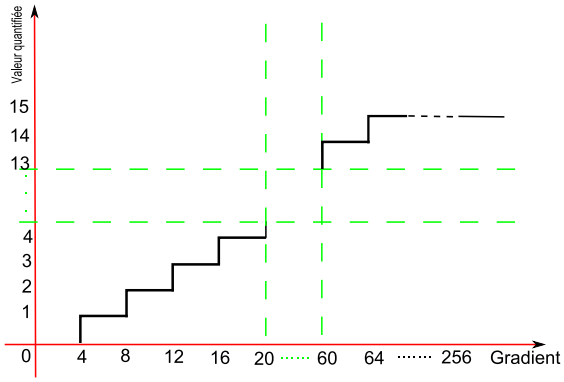

FIG. 4.7 – Quantification appliquée à chaque valeur des gradients.

chaque site. Les pixels blancs correspondent à des zones de fort gradient alors que les pixels sombres à des zones de gradient faible.

La mesure de la distance de distributions des gradients entre le site s et l'étiquette e nécessite maintenant l'accumulation des gradients des quatres directions dans les histogrammes respectifs aux macroblocs et à l'étiquette. Pour cela, chacune des valeurs est quantifiée selon la figure 4.7 . Les gradients faibles sont quantifiés de manière fines ; les gradients supérieurs à 64 sont considérés comme des textures forte et sont quantifiés à la même valeur.

La somme des coefficients de ces gradients quantifiés est ensuite égalisée à 1 afin d'obtenir une densité de probabilité des coefficients des gradients sur le macrobloc et sur la totalité de l'étiquette. Le calcul de la distance de Bhattacharyya entre les deux distributions est maintenant possible entre un site s et une étiquette e :

$$W_{text}(s,e) = dist_0(s,e) + dist_{45}(s,e) + dist_{90}(s,e) + dist_{145}(s,e) \tag{4.26}$$

Caractérisation de la distribution couleur des macroblocs et des étiquettes

Deux macroblocs appartenant à des objets différents peuvent avoir des mouvements et des textures proches. Dans ce cas, notre segmentation au sens du mouvement et notre critère sur la texture ne permettront pas de distinguer ces objets. Nous utilisons un troisième critère de segmentation des macroblocs : la couleur. Les couleurs d'un macrobloc et d'une étiquette sont représentées de façon condensée par un histogramme à trois entrées. Les trois entrées correspondent

aux trois composantes couleur de chaque pixel d'un macrobloc[2]. Chacune des trois composantes couleur est quantifiée uniformément sur 16 niveaux afin de réduire la taille de l'histogramme[3]. Une fois l'histogramme égalisé à 1, toujours pour obtenir une distribution de probabilité d'apparition de la couleur, chaque macrobloc et chaque étiquette est à présent caractérisé par sa densité colorimétrique.

Le calcul de la distance de Bhattacharyya entre le site s et une étiquette e donne :

$$W_{coul}(s,e) = dist_H(s,e) + dist_U(s,e) + dist_V(s,e) \tag{4.27}$$

Notons que les changements d'illumination au sein d'un même objet affaiblissent la similarité de deux macroblocs issus de cet objet. Les chances d'associer un macrobloc à un objet sont donc réduites. Pour pallier ce problème, différentes combinaisons d'espaces colorimétriques ont été testées [LLLP05]. La composante de teinte H de l'espace HSV et les deux composantes de chrominance U et V de l'espace YUV semblent être les mieux adaptées pour supporter les changements d'illumination. Des transformations colorimétriques seront donc appliquées sur la vidéo avant le calcul de ces histogrammes.

Caractérisation et mesure du mouvement entre un macrobloc et une étiquette

La caractérisation d'un objet par son mouvement est nécessaire pour conserver une cohérence dans les résultats avec la première segmentation. Il convient donc de pouvoir associer une énergie sur la différence entre le mouvement d'un macrobloc et le mouvement d'un objet dans le segment.

Les mouvements des macroblocs sont calculés durant la phase de traitement par estimation long terme (section 3.2) pour la première segmentation. Nous avons vu que dans cette segmentation, chacun des objets est localisé grâce à un pic sur l'histogramme associé (figure 3.8). Ce pic est donc caractéristique du mouvement d'un objet dans le segment. Etablir une distance sur la différence de mouvements entre un objet et un macrobloc revient à mesurer la différence entre le vecteur (du tube spatio-temporel) du macrobloc et celui de l'objet selon sa norme et sa direction.

Soit , $\vec{MV}_i$ le vecteur pic, déterminé par l'estimation long terme, associé à une étiquette et $\vec{MV}_{site}$ le vecteur associé à un site s.

Le calcul de distance utilisé est le suivant :

$$dist = \frac{\vec{MV}_{site}.\vec{MV}_i}{max^2(\left\|\vec{MV}_{site}\right\|, \left\|\vec{MV}_{MV_i}\right\|)} \tag{4.28}$$

Si on prend l'hypothèse $\left\|\vec{MV}_{site}\right\| > \left\|\vec{MV}_i\right\|$. On nomme θ la valeur de l'angle formée par

[2]Pour une image en niveaux de gris, l'histogramme ne possède qu'une entrée.

[3]Typiquement, une composante couleur représentée sur 256 niveaux, est quantifiée sur 16 niveaux.

$(\left\|M\vec{V}_{site}\right\|, \left\|M\vec{V}_i\right\|)$, l'équation 4.28 devient alors :

$$dist = \frac{\left\|M\vec{V}_{site}\right\| \times \left\|M\vec{V}_i\right\| \times cos\theta}{\left\|M\vec{V}_{site}\right\|^2} \tag{4.29}$$

soit

$$dist = \frac{\left\|M\vec{V}_i\right\|}{\left\|M\vec{V}_{site}\right\|} \times cos\theta \tag{4.30}$$

La différence des normes entre les deux vecteurs et leurs directions est maintenant mesurée. La norme du vecteur $\left\|M\vec{V}_{site}\right\|$ étant considérée, pour notre exemple, supérieure à $\left\|M\vec{V}_i\right\|$, la fraction $\frac{\left\|M\vec{V}_i\right\|}{\left\|M\vec{V}_{site}\right\|}$ sera bornée entre $[0, 1]$ avec 1 lorsque ces distances seront égales et 0 lorsque $\left\|M\vec{V}_i\right\| \to 0$ ou lorsque $\left\|M\vec{V}_{site}\right\| \to \infty$.

Le cosinus θ est lui borné entre $[-1, 1]$ avec 1 lorsque les vecteurs sont colinéaires et de même directions et -1 lorsqu'ils sont colinéaires et de directions opposées. L'équation générale est donc bornée entre $[-1, 1]$. Pour satisfaire à la règle de Bayes, l'énergie liée à cette distance est ramenée à :

$$W_{mouv} = \frac{1 - dist}{2} \tag{4.31}$$

L'énergie devient bornée sur $[0, 1]$ avec $U \to 0$ lorsque les caractéristiques de norme et de sens sont identiques.

A présent, les critères de caractérisation pour un site s sont définis. Nous pouvons donc calculer pour un site s donné l'énergie associée à chacune des étiquettes présente dans la scène. Nous savons que l'étiquette ayant l'énergie la plus faible aura la plus grande probabilité de correspondre au site courant. La résolution du problème est donc complète, il ne reste plus maintenant qu'à définir une politique de visite des sites.

4.2.3 Politique de visite des sites

La segmentation obtenue après l'estimation globale du mouvement et l'analyse des vecteurs de mouvement compensés sert d'initialisation pour la procédure d'optimisation. Les blocs se situant aux frontières des objets en mouvement et dans les zones uniformes ont la plus grande probabilité d'être mal étiquetés, ce qui limite le nombre de sites instables. On peut alors utiliser une pile d'instabilité pour l'ordre de visite des sites.

Dans un premier temps, on calcule, pour chaque site de l'image, l'énergie associée à chacune des étiquette :

– Si l'énergie associée à l'étiquette courante est minimum, cela veut dire que le site est stable. Sa variation d'énergie est nulle : $\Delta U(s) = 0$, le site ne doit pas être traité.

FIG. 4.8 – Exemple de carte d'instabilité sur la séquence shield avec respectivement de gauche à droite : le segment source, l'initialisation de la segmentation et la carte d'instablité associée

– Si une étiquette dispose d'une énergie plus faible que celle de l'étiquette courante, un coefficient d'instabilité est calculé, correspondant au maximum de variation d'énergie possible. Soit $e1$, l'étiquette courante du site s et $e2$, l'étiquette ayant l'énergie associée au site s la plus faible :

$$\Delta U(s) = U(s, e1) - U(s, e2)$$

Lorsque tous les ΔU associées aux sites présents dans l'image ont finalement été évaluées, il est possible de construire une carte d'instabilité localisant les sites ayant la plus forte probabilité d'étiquettage erroné.La figure 4.8 illustre un exemple de carte d'instabilité sur la séquence shield. On peut remarquer que les sites d'instabilités sont peu nombreux et généralement localisés sur les frontières des étiquettes, signe que la segmentation par estimation long terme est particulièrement fiable sur ce genre de scène.

L'énergie associé à chaque site étant maintenant définie, on construit alors une pile d'instabilité, dans laquelle les sites instables sont classés par valeur décroissante de leur instabilité. L'étiquette du premier site de la pile (le plus instable) est modifiée de façon à atteindre le minimum d'énergie locale possible. L'énergie des sites voisins mais aussi des étiquettes s'en trouvant modifiée, la pile devrait en théorie être remise à jour à chaque itération.

En effet, lors d'un changement d'étiquettes, les énergies associées aux cliques du voisinage changent et la distribution des couleurs et des gradients doit être soustraite de l'ancienne étiquette pour être associé à la nouvelle. Cette remise à jour obligerait à avoir une pile de petite taille. Le changement d'une étiquette nécessiterait le calcul à nouveau de l'intégralité de la pile ce qui deviendrait vite plus coûteux que de balayer systématiquement toute l'image.

Après différents tests réalisés sur les différentes possibilités de modification de la pile, il s'est avéré que la distribution de couleurs et de textures de chaque objet ne variait que très peu durant la phase de correction mais que la modification d'un étiquette pouvait entraîner une forte variation d'énergie sur le voisinage. Nous avons donc décidé de ne pas remettre à jour les distributions de couleurs et de textures associées aux étiquettes mais de mettre à jour dans la pile les énergies liées au voisinage du site traité.

Ce processus est ensuite itéré jusqu'à ce que la pile soit vide, c'est à dire jusqu'à ce que tous les sites deviennent stables. Afin de ne pas tomber dans un minimum local, un site traité ne pourra plus retourner dans la pile d'instabilité. Un exemple d'instabilité est illustré en figure 4.9.

A la fin de ce traitement, les distributions associées à chacune des étiquettes doivent être recalculées afin de pouvoir transmettre au bloc de classification les informations caractéristiques de celle-ci.

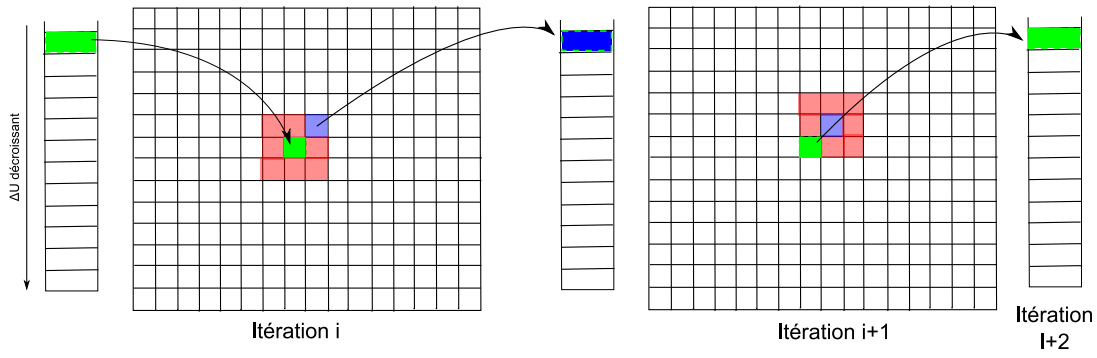

FIG. 4.9 – Illustration d'une instabilité locale : le site s (en vert) se trouve en haut de la pile d'instabilité à l'itération i, le changement d'étiquette affecte son voisinage (en rouge), nous tombons dans un minimum local lorsque le changement d'énergie dans le voisinage (illustré ici en bleu) le place directement en haut de la pile d'instabilité à l'itération i+1. Ce minimum local aura pour conséquence un traitement en boucle de ces deux seuls sites

4.2.4 Conclusion

Cette section est dédiée à la description du bloc de traitement intra-segment temporel de l'outil de pré-analyse d'un flux vidéo. L'utilité de ce bloc est de fournir une segmentation basée sur la carte intialement fournie par le premier traitement avec prise en compte de nouveaux critères et une description détaillée des différents objets d'un segment de 180ms. La description détaillée d'un objet est maintenant constituée de trois informations : son mouvement, sa couleur et sa texture.

Á ses trois informations est associé un réglage de leur importance par les paramètres β_1 pour la connexité, β_2 pour la couleur, β_3 pour la texture et β_4 pour le mouvement. Le réglage de cette importance va directement influer sur la qualité de segmentation finale. Une méthode pour ajuster ces coefficients doit être nécessairement définies avant de pouvoir implanter cette fonction dans la pré-analyse. Cette méthode n'est pas traitée dans ce mémoire, mais il sera peut-être nécessaire pour obtenir une segmentation robuste sur n'importe quel type de séquence d'ajouter une nouvelle fonction qui adapterait ses propriétés en fonction des caractéristiques prédominantes des objets dans une séquence.

Une autre caractéristique importante de notre segmentation par approche markovienne est qu'elle ne créée pas de nouvelle région mais affine le résultat final de la carte initiale de segmentation. La fiabilité de la segmentation initiale basée sur l'analyse de tubes spatio-temporels est primordiale à l'efficacité de cette méthode.

La prochaine étape de traitement pour la pré-analyse est d'assurer une cohérence dans le traitement entre chaque segment à partir de la première segmentation, cohérence qui n'est pas prise en compte par la segmentation basée sur l'analyse long terme.

4.3 Traitement inter-segment temporel

Le traitement inter-segment temporel permet de prendre en compte le segment précédent pour fournir des résultats plus cohérents sur l'ensemble de la séquence. Au lieu de diviser simplement un segment temporel de 180ms en objets spatio-temporels caractérisés par leur mouvement, leur couleur et leur texture, le bloc de traitement inter-segment va permettre d'assurer le suivi de ces objets sur plusieurs segments successifs. Une nouvelle caractéristique sera alors disponible pour les objets : leur cycle de vie.

Cette information de suivi permettra au bloc de classification de transmettre au codeur des paramètres cohérents pour coder de la même façon un même objet à différents instants temporels afin notamment, de réduire les phénomènes de battement générés par un codage reposant exclusivement sur une minimisation débit-distorsion.

De plus, dans le but d'exploiter la redondance temporelle entre des segments temporels successifs et d'améliorer les probabilités de détections des objets présents dans la scène, le traitement inter-segment peut se baser sur les résultats du segment précédent en incluant une nouvelle fonction de potentiel ou lorsque que la segmentation du segment courant par estimation long terme échoue (aucun objet n'est détecté), se servir du résultat du segment précédent pour l'initialisation.

4.3.1 Suivi d'un objet sur plusieurs segments

L'objectif est de suivre les objets temporels sur des tranches supérieures à 180ms, il faut donc concevoir une méthode qui assure la continuité d'un segment temporel au segment suivant. Nous avons vu dans le chapitre précédent que le traitement intra d'un segment temporel génère des objets caractérisés par trois informations : leur mouvement, leur couleur et leur texture.

Ces informations permettent de créer une carte de segmentation pour chaque tranche temporelle de 180ms. Pour pouvoir suivre un objet d'un segment temporel au segment suivant, nous allons projeter la carte de segmentation du segment précédent sur la carte de segmentation du segment courant. Ainsi, si un objet de la carte de segmentation du premier segment temporel dispose de caractéristiques de textures, de couleurs et recouvre majoritairement un objet de la carte de segmentation du segment suivant, nous attribuerons le même label à ces deux objets afin de les fusionner en un seul objet dont le cycle de vie est supérieur à 180ms.

Afin de pouvoir superposer deux cartes de segmentation successives, il est nécessaire de créer un carte de segmentation théorique par projection du segment précédent à l'instant correspondant à la carte de segmentation du segment courant. Nous savons que les pics détectés lors de l'estimation long terme sont caractéristiques du mouvement d'un objet dans le segment. Nous projetons alors les vecteurs pics associés aux objets de manière à avoir pour un instant t, une carte de segmentation fournie par l'estimation long terme ainsi qu'une carte de segmentation théorique du segment précédent(figure 4.10). Il suffit maintenant de voir selon les caractéristiques d'un objet, quelles sont ceux du segment précédent qui auront le plus de probabilité de correspondre aux objets du segment courant. Pour cela, nous calculons la distance selon le même procédé que celui utilisé lors du traitement intra sur des critères de distribution de couleurs, de textures et de recouvrement entre chacune des étiquettes de la carte de segmentation théorique

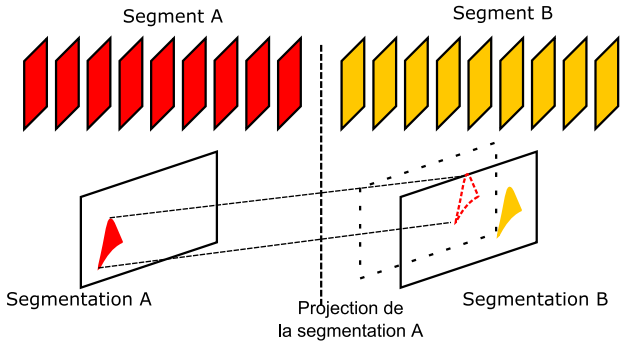

FIG. 4.10 – Recouvrement d'objets.

du segment précédent et de la carte de segmentation du segment courant. Les étiquettes de la carte de segmentation courante ayant les distances les plus courtes seront affectées à la valeur de l'étiquette correspondante.

Les objets peuvent maintenant être reconnus d'un segment à un autre.

4.3.2 Calcul de l'énergie liée à la clique temporelle inter-segment

La carte de projection théorique du segment précédent peut être aussi utilisée pour apporter plus de régularité au traitement intra. Pour cela, un nouveau critère est introduit, celui de la connexité temporelle inter-segment (figure 4.11).

Ce critère est semblable au critère de cliques du traitement intra-segment, mais va vérifier la présence de l'étiquette à la même position que le site s traité afin de lui apporter plus de poids. Le résultat obtenu permettra aux objets de garder une forme cohérente durant toute leur apparition durant la séquence. La nouvelle énergie associant un site à une étiquette devient alors :

$$U(s,e) = \beta_1.W_{cliques}(s,e)+\beta_2.W_{coul}(s,e)+\beta_3.W_{text}(s,e)+\beta_4.W_{mouv}(s,e)+\beta_5.W_{temporelle}(s,e), \tag{4.32}$$

avec :

$$\begin{cases} V_{c_t}(s,e) = \beta_t \quad si \quad e_t(s) \neq e_{t-1}(s) \\ V_{c_t}(s,e) = 0 \quad si \quad e_t(s) = e_{t-1}(s) \end{cases} \tag{4.33}$$

L'énergie totale $W_{temporelle}$ est donc :

$$\{ \ W_{temporelle} = \sum_{c_t \in C_t} V_{c_t}(e), \tag{4.34}$$

Afin de garder une cohérence avec les autres énergie, β_t est fixé à 1. L'énergie $W_{temporelle}$ est donc compris alors entre $[0,1]$. Cette énergie va permettre de favoriser les étiquettes du segment courant recouvrant les étiquettes du segment précédent.

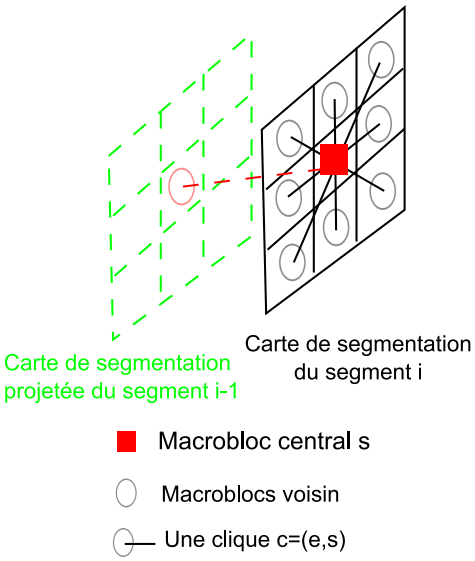

FIG. 4.11 – Illustration de toutes les cliques associées au site s

4.3.3 Utilisation du masque théorique lors d'un traitement par estimation long terme défaillant

Lorsque les mouvements présents dans le segment courant se révèlent trop faible pour être détectés lors de l'estimation long terme, aucun objet ne peut être localisé par la segmentation. Sans objet, le traitement intra ne pouvant pas créer de nouvelles régions, le traitement du segment courant devient impossible.

Une utilisation de la carte de segmentation projetée du segment précédent peut alors servir d'initialisation pour un nouveau traitement par approche markovienne et ainsi tenter de retrouver la présence des mêmes objets dans le segment courant. La segmentation intra va donc sélectionner comme initialisation, lorsque la carte de segmentation est vide[4], le projeté de la carte de segmentation du segment précédent et effectuée un traitement en fonction des étiquettes théoriques. Ce procédé permet d'obtenir un suivi d'objet fiable tout au long d'une séquence.

4.4 Conclusion

Dans ce chapitre, nous avons décrit le bloc de traitement intra et inter-segment temporel de l'outil de pré-analyse d'un flux vidéo. Nous avons vu que le traitement inter et intra se fait conjointement pour chaque segment. Au final, nous maximisons la probabilité d'avoir une carte de segmentation spatio-temporelle et des informations de mouvement, de couleur et de texture des différents objets qui composent la scène de plans vidéo. Les différents traitements effectués sur la carte de segmentation permettent essentiellement d'affiner la première segmentation et d'obtenir plus de régularité dans la détection des objets d'une séquence.

[4]Une carte de segmentation vide correspond à une scène sans objet détecté

Le bloc de traitement Inter et Intra ne fonctionne pour l'instant pas de manière autonome. En effet, pour obtenir une segmentation de qualité, l'analyse doit passer par une action de l'utilisateur sur les cinqs réglages d'importance de tous les critères. Ce qui rend pour l'instant impossible l'utilisation de cette pré-analyse au sein du codeur. Une étude sur le jeu de paramètre optimal ou le développement d'une fonction adaptant ces coefficients en fonction de la séquence source devra être développée. Hormis le réglage de ce jeu de paramêtre, la fonction, elle, est autonome. Lorsque l'on place une séquence à son entrée, elle est découpée en segment, puis analysée pour obtenir un ensemble d'objet caractérisé par sa couleur, sa texture et son mouvement.

Toutes ces informations peuvent être maintenant transmises à un bloc de classification qui permettra d'exploiter ces caractéristiques pour optimiser les choix du codeur H.264.

Dans le chapitre suivant, nous allons maintenant présenter les résultats fournies par le bloc de traitement INTER et INTRA de l'outil de pré-analyse.

Chapitre 5

Présentation des résultats de la pré-analyse

Ce chapitre illustre les résultats de la pré-analyse sur trois séquences : *Knightshields*, *new mobil & calendar* et *tractor* donc le contenu est détaillé en annexe A. Ces vidéos ont été sélectionnées car elles permettent de bien mettre en évidence les avantages et les inconvénients de notre méthode d'analyse.

Dans un premier temps, nous verrons l'impact du réglage de l'importance des critères de segmentation afin de déterminer quelles informations sont caractéristiques à la segmentation pour une séquence donnée.

Les seconds résultats vont permettre de mettre en évidence les corrections apportées par le traitement par approche markovienne sur deux vidéos tests par rapport à la première analyse par estimation long terme.

Les derniers résultats montrent les améliorations mais aussi les erreurs générées par le traitement Inter. Nous donnerons alors des clés de réponse à la correction de ce traitement.

Les informations illustrants les différents traitement seront prises à différent endroit de la pré-analyse : l'entrée de la pré-analyse représentée par un segment dans la vidéo source, l'entrée du traitement par approche markovienne représentée par la carte de segmentation de sortie de l'estimation long terme et la sortie du traitement qui est une nouvelle carte de segmentation améliorer par le traitement Inter et Intra par approche markovienne.

5.1 Influence des paramètres sur la segmentation

Cette section étudie de quelle manière vont opérer chacun des critère associé à la segmentation sur la qualité du masque final. Pour cela, chacun des paramètres β_1, β_2, β_3, β_4, β_5 va être successivement fixé à une valeur bien supérieur à autres. En fixant ces paramètres à une valeur supérieure, le choix d'une étiquette associé à un site s sera effectué sur le critère discriminant.

Nous avons choisi d'effectuer ces tests sur la vidéo Knightshields (figure 5.1) car celle-ci obtient déjà de très bon résultat avec la seule estimation long-terme (figure 5.2). Les deux éléments constituants la scène sont détectés avec le fond marqué en noir sur la carte et le personnage marqué en blanc. Quelques erreurs de segmentation sont visibles :

– les bords du personnage dans la séquence sont confondu avec le fond dans la carte de segmentation,
– deux régions situées en face et en haut du personnage, correspondant au fond dans la séquence, sont erronnées (tâche blanche sur la carte de segmentation),
– une ligne blanche suivant le bord gauche du cadre est elle aussi marquée comme n'appartenant pas au fond. Ce problème est récurrent lors de la segmentation long terme. La caméra effectuant un travelling latéral, les nouveaux pixels entrant dans le segment sont généralement mal étiquettés.

Le peu de défaut présent dans la carte de segmentation initiale va permettre de constater les critères améliorants ou non le résultat final.

FIG. 5.1 – Segment de référence correspondant à l'image 35 de la séquence *Knightshields*

Cependant, un véritable test fiable pour déterminer ces critères ne doit pas s'effectuer sur une seule séquence. En effet, la séquence choisie dispose de caractéristiques propres qui : sont un ensemble de la scène très texturé et un mouvement linéaire de la caméra avec un personnage central placé au centre ayant des couleurs assez proches du fond. L'information la plus importante ici sera donc le mouvement, c'est d'ailleurs la raison de la bonne qualité de la segmentation par estimation long terme donne.

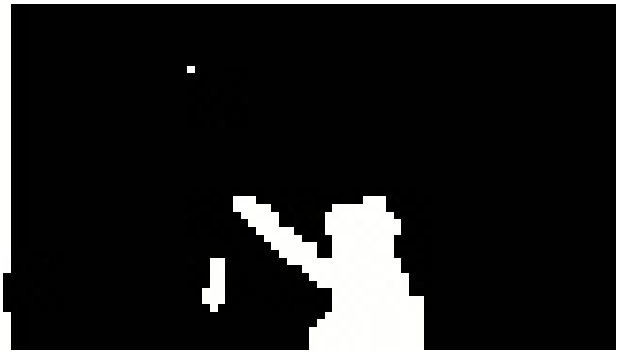

FIG. 5.2 – Résultat avec la technique d'estimation long terme

Des tests sur la séquence *new mobil & calendar*, où chacun des objets de la scène dispose d'une caractéristique en texture et en couleur qui lui est propre, donnerait des résultats très dif-

férents. Les tests sur cette scène permetrait de constater l'amélioration, mais, l'estimation long terme effectuée sur la séquence ne donnant pas des résultats d'assez bonne qualité, il serait difficile de constater les dégradation sur le résultat final.

Chacun de ces tests vont être illustré par deux images : le résultat placé à gauche avec la carte de segmentation qui se trouve en sortie du codeur et à droite une carte d'information binaire indiquant les sites traités. Cette carte d'information binaire est en faite la carte d'énergie qui est transmise pour être traitée par la pile d'instabilité. Un site blanc indique un site qui sera traité par l'analyse et un site noir indique un site qui sera non traité.

Le premier test est effectué en donnant à chacun des critères la même importance : β_1, β_2, β_3, β_4, β_5 sont fixés à 1. Le résultat obtenu, visible en figure 5.3, permet d'établir une comparaison entre les différents critères. On peut remarque que lorsque tous les coefficients sont aux mêmes niveaux, le traitement sur la séquence est minime. Quelques sites sont corrigés autour du personnage central et sur les bords de l'image mais les plus gros défauts que constituent la ligne blanche au bord du cadre et la tache du fond mal étiquettée sont toujours présents.

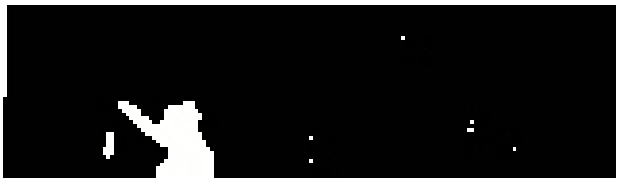

FIG. 5.3 – Résultat avec Beta1 = 1, Beta2 = 1, Beta3 = 1, Beta4 = 1 et Beta5=1

L'importance donnée à la connexité locale permet de résoudre l'ensemble des problèmes de segmentation dans le test illustré en figure 5.4. En effet, les erreurs de cette séquence sont assez localisées, donner plus d'importance dans la correction aux groupes de macroblocs permet de rattacher la plus part des erreurs soit au fond de la scène, soit au personnage. Dans cette séquence, la connexité locale est donc un critère important de correction.

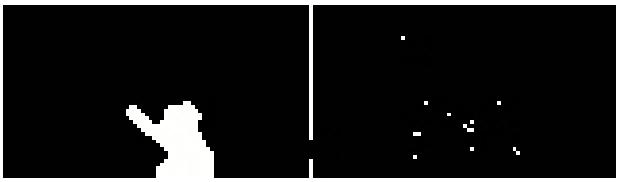

FIG. 5.4 – Résultat avec Beta1 = 10, Beta2 = 1, Beta3 = 1, Beta4 = 1 et Beta5=1

Le test effectué en figure 5.5 tente de donner plus d'importance au critère de couleur. Cependant, le personnage et le fond ayant des couleurs diverses mais finalement assez proche, ce critère ne corrige que très peu la segmentation. Quelques erreurs sont apportées au niveau de la ligne blanche longeant le cadre gauche qui, par leurs couleurs sombres, ont été associées au personnage.

Le troisième critère, en figure 5.6, donne les résultats les plus mauvais. Les deux objets, le fond et le personnage, sont très fortement texturés. Le choix dans l'appartenance d'un macrobloc

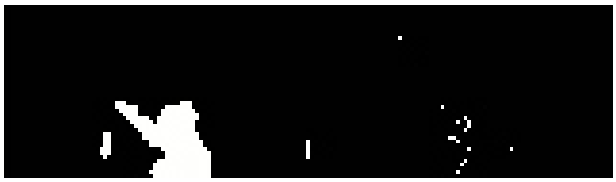

FIG. 5.5 – Résultat avec Beta1 = 1, Beta2 = 10, Beta3 = 1, Beta4 = 1 et Beta5=1

à un objet est donc totalement biaisé par une recherche sur un critère équivalent. L'information de texture n'est pas utile au traitement de cette séquence.

FIG. 5.6 – Résultat avec Beta1 = 1, Beta2 = 1, Beta3 = 10, Beta4 = 1 et Beta5=1

Le critère de mouvement, figure 5.7, permet de conserver un résultat équivalent à la segmentation initiale. Le critère de mouvement étant celui utilisé lors de la première segmentation, même si la technique de comparaison entre les vecteurs est différente, il est normal de trouver une segmentation identique. L'influence, même minime des autres critères, permet néanmoins de corriger les différentes erreurs présents dans la segmentation initiale. Le critère de mouvement est donc un facteur important pour le traitement de cette séquence.

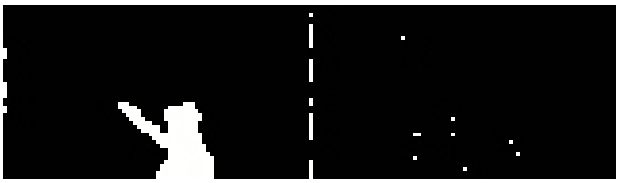

FIG. 5.7 – Résultat avec Beta1 = 1, Beta2 = 1, Beta3 = 1, Beta4 = 10 et Beta5=1

Le dernier critère va tenter au maximum de garder une cohérence avec la projection de la carte de segmentation du segment précédent. On constate sur les résultats, en figure 5.8, un envol du personnage au cours de la séquence. Ceci est la conséquence d'une carte de segmentation biaisée. Si le vecteur associé à l'objet personnage dévie un tant soit peu de sa position sur le prochain segment, il en résulte une détection des objets qui risquent de ne pas être en phase avec sa réel location dans l'image. Ce critère doit donc être utilisé avec prudence de façon à ne pas obtenir des résultats corrects mais mal localisés dans l'image.

Au vu des résultats, nous comprenons bien que la qualité de la segmentation obtenu par le bloc de traitement développé est relative au jeu de paramètres qui lui sont associés. Ce jeu de paramètre doit obligatoirement être fixé en fonction du contenu de la séquence.

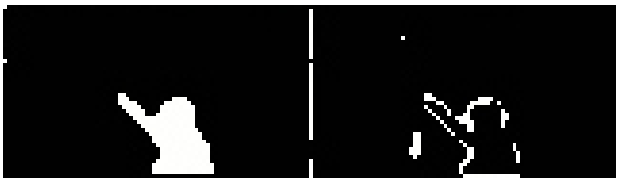

FIG. 5.8 – Résultat avec Beta1 = 1, Beta2 = 1, Beta3 = 1, Beta4 = 1 et Beta5=10

Sur la séquence testé, les critères discriminants sont la connexité et le mouvement car les deux objets contenus dans la scène dispose d'informations de couleurs et de textures assez similaire. Les tests effectués sur la séquence *new mobil & calendar* ont eux donné de meilleur résultat avec les informations de connexité et de couleur.

Ce que révèle tout cela, c'est que la connexité doit être considéré comme le critère le plus important avec le mouvement. Les autres critères devront passé par une analyse, par exemple une étude sur les variances de distribution, pour pouvoir être au mieux ajuster.

La suite des résultats permet de constater l'amélioration, avec un jeu de paramètre fixé, de la segmentation par approche markovienne sur la carte d'initialisation initiale.

5.2 Impact de la segmentation markovienne par rapport à la première segmentation

Le résultat maintenant présenté est obtenu en appliquant une analyse par notre traitement en tentant d'utiliser le jeu de paramètre le plus adapté à la séquence. Ces paramètres ont été obtenus de manière empirique en effectuant une série de test et en observant les cas où les différents objets étaient le mieux englobés par une étiquette qui leur était propre.

Cette série de test a donc été effectué sur la séquence *Knightshields* et sur la séquence *New mobil & calendar*. Les résultats de ces test seront illustrés par une figure divisé en 4 parties :

- L'image en haut à gauche correspond au segment à traiter.
- L'image en haut à droite correspond à la carte de segmentation initiale traitée par l'estimation long terme.
- L'image en bas à droite correspond à la carte d'énergie transmis à la pile d'instabilité. Cette image correspond à la localisation des points modifiés sur la carte de segmentation initiale.
- L'image en bas à gauche correspond à la carte de segmentation finale donc au résultat de notre traitement.

Le premier test est effectué sur la séquence *Knightshields*. En tenant compte des résultats précédents, nous avons décidé de porter l'importance sur les critères de connexité et de mouvement. La segmentation étant initialement de bonne qualité, le résultat de notre pré-analyse se contente de corriger les quelques erreurs présents dans la segmentation. Sur la figure 5.9, nous pouvons voir que la segmentation initiale détecte trois objets : le fond (en noir), le corps du personnage (en gris) et le bras du personnage (en blanc). On peut voir que plusieurs parties du fond sont étiquettées comme étant le bras ou le corps du personnage. Le traitement par notre procédé corrige la plupart des erreurs même si la bande gauche blanche n'arrive pas à être rattaché au fond. Ceci

est du au faite que les macroblocs placés sur le cadre gauche dispose d'une information couleur très semblable au personnage et que les vecteurs rattachés à ces macroblocs sont mal estimés. Ce problème sera récurrent dans nos résulat lors de mouvement important de la caméra, les objets apparaissant au centre d'un segment ont souvement des mouvements faussés par l'estimation long terme.

FIG. 5.9 – Amélioration sur *Knightshields* avec beta1 = 3, beta2 = 1, beta3 = 1, beta4 = 2, beta5 = 0.5

Le deuxième test tente d'améliorer le traitement sur la séquence *New mobil & calendar*. Nous avons vu précédemment que cette séquence ne donne pas de très bon résultat sur l'estimation long terme car la scène est complexe, comporte plusieurs mouvement de caméra (zoom, déplacement latéral) et plusieurs objets en mouvement. Les trois objets sont correctement détectés avec le fond (en noir), le calendrier (en gris) et le train (en blanc). On peut constater de nombreuses irrégularitées sur tous les objets.

Les objets dans cette séquence ont des caractéristiques de distributions et de couleurs qui leurs sont propres. Le mouvement est aussi un facteur important pour leur détection. Nous avons donc choisi de fixer les coefficients sur les critères a des valeurs quasiment équivalentes.

Le traitement par notre approche apporte des bords plus lisses, englobant mieux les différents objets. Des défauts subsistent sur bas du calendrier (la partie ombrée de la tapissirie), car les couleurs du fond sont plus sonmbres. La distribution est modifiée et est rattaché à l'étiquette englobant déjà cette nouvelle distribution (en l'occurence l'étiquette du calendrier).

Nous pouvons voir que notre traitement apporte plus de régularité dans le traitement d'une séquence et que les objets sont mieux rattachés aux étiquettes qui leurs sont propres. Quelques erreurs subsistent, souvent liés à une estimation long terme erronée. Les résultats pourraient cependant être améliorer en incluant un nouveau critère à la segmentation : le contour. En effet, en détectant les contours fermés d'un objet, nous pourrions donner plus de probabilité à un macrobloc d'être rattaché à un objet en étudiant les distributions inclues par ce dernier.

Les prochains test vont mettre en evidence l'impact de la prise en compte de la correlation inter-temporelle sur la séquence.

FIG. 5.10 – Amélioration sur *New mobil & calendar* avec beta1 = 2, beta2 = 1, beta3 = 1, beta4 = 2, beta5 = 0.5

5.3 Influence du traitement inter sur le résultat de la séquence

La dernière partie de résultat permette de mettre en évidence les améliorations mais aussi les erreurs restantes du traitement inter. Pour cela, nous allons effectuer nos tests sur la séquence *tractor*.

Les résultats seront présentés de la même manière que précédemment, à savoir :

- L'image en haut à gauche correspond au segment à traiter.
- L'image en haut à droite correspond à la carte de segmentation initiale traitée par l'estimation long terme.
- L'image en bas à droite correspond à la carte d'énergie transmis à la pile d'instabilité. Cette image correspond à la localisation des points modifiés sur la carte de segmentation initiale.
- L'image en bas à gauche correspond à la carte de segmentation finale donc au résultat de notre traitement.

Les résultats sur la séquence *tractor* conduisent par estimation long terme conduisent à des objets mal ou non détectés. La prise en compte de critères suplémentaires sur une image ainsi que l'utilisation du masque précédent lorsque rien n'est détectée dans la scène permettent d'obtenir une carte de segmentation suivant correctement l'objet principal tout au long de la séquence.

La couleur est ici un critère important des objets. Le fond et le tracteur étant très texturé nous avons décidé de ne pas accordé beaucoup d'importance au critère de texture.

La première image (figure 5.11) illustre un segment où le tracteur est détecté par l'estimation long terme. On peut voir que l'étiquette correspondant à ce dernier est blanche et que le fond est marqué par du noir. On peut remarquer que le traitement intra joue un rôle important dans la correction des zones autour du tracteur.

La seconde image (figure 5.12) correspond au segment suivant la figure 5.11. Cette fois ci, l'estimation long terme détecte que le tracteur prend une place prépondérante dans la scène et le détecte donc comme le fond (marque noir). La fonction de correspondance des étiquettes, en analysant les distributions et les recouvrements, permet de corriger ce défaut et de reétiquetter le tractor avec son étiquette correspondance (en bas à gauche le tracteur est en blanc).

La dernière image (figure 5.13)illustre l'amélioration apportée par la prise du segment pré-

FIG. 5.11 – Image 33 de la séquence *tractor* avec beta1 = 3, beta2 = 2, beta3 = 1, beta4 = 2, beta5 = 0.5

FIG. 5.12 – Image 42 de la séquence *tractor* avec beta1 = 3, beta2 = 2, beta3 = 1, beta4 = 2, beta5 = 0.5

cédent lorsqu'aucun objet n'est détecté dans la séquence par l'estimation long terme. Sur neuf segments successifs, un zoom est effectué sur le tractor. La compensation de mouvement effectué par l'estimation long terme fausse la détection du tracteur, aucun objet n'est détecté dans la segmentation initiale (image en haut à gauche noir). Notre procédé prend alors le segment précédent corrigé comme initialisation. On peut constater que le suivi est correctement effectué avec un arrière de tracteur correctement détecté.

Ce traitement fonctionne correctement dans cette séquence car l'objet tracteur apparait tout au long de la scène. Cependant grâce l'exemple suivant, tiré de la séquence *New mobil & calendar*, nous allons pouvoir mettre en évidence un problème du traitement inter. La figure 5.14 illustre le traitement effectué sur l'image 400 de cette séquence. La carte de segmentation initiale détecte le calendrier en mouvement (en blanc) mais non le train situé en bas.

Lors du traitement du segment suivant (figure 5.15), le calendrier n'est plus détecter dans la carte de segmentation initiale mais le train est lui marqué comme un objet en mouvement (en blanc). Le fond est lui bien détecté en noir. Notre traitement se basant sur le segment précédent pour effectuer la comparaison d'étiquette va tenter de faire une correspondance entre la carte de segmentation résultant du masque précédent et la carte de segmentation du segment courant. La distribution des informations du calendrier, qui était on le rappelle détecter comme l'objet en

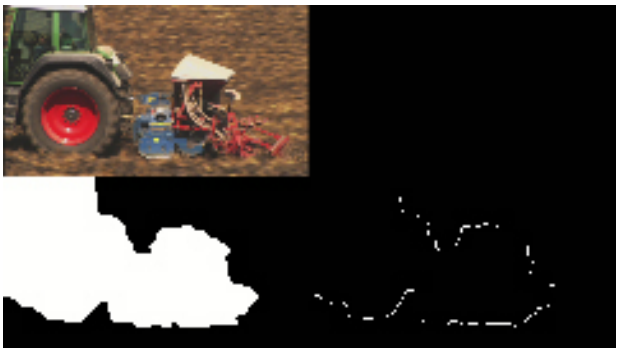

FIG. 5.13 – Image 79 de la séquence *tractor* avec beta1 = 3, beta2 = 2, beta3 = 1, beta4 = 2, beta5 = 0.5

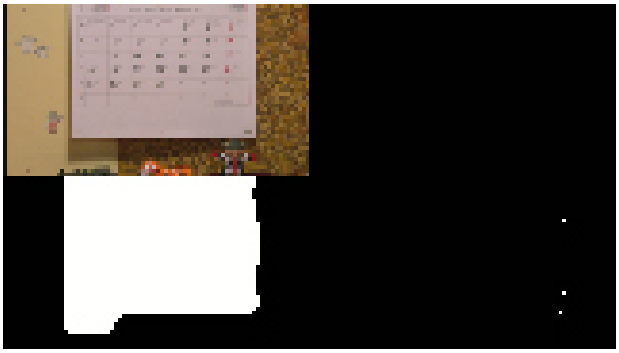

FIG. 5.14 – Image 400 de la séquence *new mobil & calendar*

mouvement, est maintenant inclue dans le fond. Notre traitement estime donc que le fond de la carte de segmentation iniatiale correspond au calendrier détecté dans le segment précédent et lui donne l'étiquette correspondante (l'étiquette blanche en bas à gauche de l'image). Le train est lui maintenant détecté comme le fond.

Ce problème, mis en évidence par la séquence *new mobil & calendar*, pourrait être résolu en ayant un dispositif de reconnaissance de nouveaux objets dans la scène. Pour cela, un seuil dans les critères de sélections d'étiquettes devrait être fixé. Lorsque l'énergie est trop faible, c'est à dire que les correspondances entre les objets sont trop éloignées, l'objet courant (dans notre exemple le train) serait alors étiquette comme un nouvel objet.

5.4 Conclusion

La dernière partie du rapport montre l'application de notre méthode de pré-analyse sur trois séquences tests. Ces traitements nous ont permi de constater les améliorations mais aussi les problèmes qui restent à résoudre pour obtenir une segmentation véritablement fiable.

Le défaut majeur de notre segmentation *intra* et *inter* est qu'elle nécessite un réglage adapté à chaque vidéo pour obtenir un résultat robuste. Il sera donc nécessaire par la suite de développer

FIG. 5.15 – Image 409 de la séquence *new mobil & calendar*

une fonction qui permettrait d'avoir un *a priori* sur la scène en détectant les critères discriminant d'un objet pour ensuite transmettre ces informations au bloc de pré-analyse.

Cependant, les résultats générés par notre traitement inter et intra avec réglage manuel, ou même sans, améliorent de manière importante la détection des objets dans la scène. A partir d'une carte de segmentation initiale correcte, nous retrouvons des objets mieux détourés et gardant la même étiquette sur plusieurs segments successifs.

Cette mise en correspondance des étiquettes entre les segments comporte elle aussi un défaut majeur. Ce défaut apparaît lorsqu'un nouvel objet apparaît sur un segment et alors qu'un objet, présent dans le segment précédent, n'est plus détecté dans le segment courant. La mise en correspondance ne sachant pas distinguer des objets entre eux, celui-ci sera directement marqué par l'objet précédent alors qu'il devrait être reconnu comme un nouvel objet de la scène.

La résolution de ce problème est simple, en fixant un seuil de probabilité de correspondance, la pré-analyse saurait ou non s'il doit associer un objet sur le segment courant à un objet du segment précédent. Or, un objet peut avoir ses distributions changeant rapidement durant son apparition au cours d'une séquence. Ce changement de distribution peut être du par exemple à un changement de luminance. Ce qui provoquerait une réorganisation dans la distribution de ses couleurs. Une étude rigoureuse doit donc être portée sur le changement maximal de propriété que peut subir un objet. Ce seuil, une fois déterminé, donnerait un suivi d'étiquette fiable.

Conclusion générale

Intérêt de la pré-analyse vidéo

Dans ce mémoire, nous avons abordé le problème de la visibilité limitée de la norme de codage H.264 pour le traitement des objets dans une séquence . Nous avons vu que cette problématique conduisait le codeur de référence à tester tous ses modes de prédictions afin de ne retenir que celui optimisant le résultat en terme de débit/distortion. Les objectifs de ce mémoire étaient : premièrement de valider l'hypothèse selon laquelle une connaissance *a priori* de la scène permet de réduire la complexité de son traitement, et deuxièmement de développer une méthode mettant à profit cette hypothèse pour optimiser le codage d'un flux H.264.

Le résultat de la deuxième partie permet d'introduire le terme de pré-analyse vidéo. Ce terme désigne le procédé de prise en compte des informations contenues dans une scène pour en effectuer, par la suite, un meilleur traitement. Cette pré-analyse vidéo permettrait d'analyser une scène dans le but de détecter la présence des différents objets la composant. La détection de ces objets doit passer obligatoirement par une segmentation/classification des informations contenues dans une scène selon des critères donnés. La segmentation permet de réunir l'unité du codage, correspondant dans le cas de la norme H.264 aux macroblocs, en groupe d'unités ayant des caractéristiques similaires. Un objet est majoritairement caractérisé par son mouvement dans la scène mais aussi par ses informations de couleurs et de textures. La classification des macroblocs devra dons s'effectuer selon ces informations.

Pour cela, une méthode de pré-analyse de contenu vidéo, détaillée dans la troisième partie du rapport, a été développée par l'équipe du projet ArchiPEG permettant de détecter le mouvement d'un macrobloc sur 9 images d'une séquence. En effectuant une classification des macroblocs selon leurs mouvements, l'équipe du projet disposait d'un outil ayant une visibilité supérieure du mouvement des objets par rapport au procédé de codage de la norme H.264. Cependant, nous l'avons vu, un objet n'est pas seulement caractérisé par son mouvement mais peut être aussi caractérisé par ses informations de textures et de couleur.

L'objectif final de cette recherche est donc de pouvoir améliorer le premier travail de détection d'objet de l'outil de pré-analyse, pour y inclure de nouveaux critères de textures et de couleurs conjointement au mouvement.

Contributions

Dans la deuxième partie de ce rapport, nous avons étudié les différents types de segmentation/classification d'informations existant dans la littérature. Le modèle statistique par approche markovienne est un outil à la fois simple et robuste pour classer des informations selon un nombre de critère illimité. Ce modèle d'analyse est tout à fait adapté à nos besoins. La quatrième partie du rapport expose donc les moyens d'application de ce modèle dans notre pré-analyse. Cette implantation passe d'abord par une bonne compréhension théorique du principe des champs markoviens et une modélisation bien élaborée du problème pour prendre au mieux en compte les différentes informations mises à disposition dans la scène. Cette modélisation doit passer par une mesure au plus juste des informations nécessaires, à savoir les textures d'un objets et ses couleurs.

De nombreux outils existent dans la littérature concernant l'application de champs markovien pour le traitement d'image. La suite de la quatrième partie du rapport s'attarde sur les choix à disposition et sélectionne les méthodes les mieux adaptées à notre application. Ces méthodes sont la mesure de similarité des critères sélectionnés mais aussi la gestion de traitement des différentes corrections à appliquer sur l'outil d'analyse développé par l'équipe du projet.

Enfin, les résultats de l'outil développé peuvent encore être améliorés en étendant la visibilité sur toute la séquence. La fin de la quatrième partie expose des méthodes exploitant les informations déjà traitées par l'analyse pour ajouter plus de cohérence au résultat final. Cette cohérence passe par la reconnaissance dans un objet tout au long de l'apparition dans un séquence, mais aussi la prise en compte du résultat du traitement précédent pour augmenter la probabilité de détection d'un objet.

Enfin, les résultats de notre analyse, présentés dans la dernière partie du rapport, valident le bon fonctionnement de l'outil finalement développé. L'amélioration des résultats effectués par la première analyse est visible. Les contours des objets sont plus net, la plupart des erreurs de détections sont corrigées et enfin les objets sont reconnus durant toutes leurs apparitions dans la séquence. Cependant, lors de séquences complexes, des erreurs de détections subsistent.

Perspectives

La dernière partie de ce mémoire montre que les résultats du travail présenté peuvent être améliorés. Nous l'avons vu précédemment, l'intérêt des approches markoviennes est une possibilité de prise de décision sur un nombre de critères illimités. Les critères de textures et de couleurs semblent être des caractéristiques importantes d'un objet, mais une prise en compte d'informations supplémentaires pourrait améliorer d'avantage la probabilité de réussite de détection d'objets. Un critère de contour, par exemple, permettrait une meilleure reconnaissance des objets sur ses frontières.

Le défaut majeur de la pré-analyse développé réside dans l'obligation de réglage de l'importance des critères de classification afin d'obtenir les meilleurs résultats de détection. Ce réglage nécessite une connaissance au préalable de la scène à analyser et donc l'intervention d'un utilisateur externe. Cette contrainte est incompatible avec l'application finale de la pré-analyse. Une

méthode robuste de réglage automatique de l'importance des critères doit donc être développée. Cette méthode devra se baser sur une première analyse d'une scène et la reconnaissance des critères prédominant à la détection des objets. Cette méthode pourrait par exemple étudier la diversité et les variations des informations contenues par les objets détectées par l'analyse du mouvement.

Un autre défaut de notre pré-analyse est d'avoir une vision des caractéristiques d'un objet limité. De ce fait, un nouvel objet ayant des caractéristiques différentes d'un autre objet pourrait être détecté comme un seul et même objet. La disparation de ce premier, simultanément avec l'apparition de ce deuxième, induirait un suivi d'objet erroné. Il est donc également nécessaire de développer une méthode de caractérisation d'un objet pouvant différencier un objet d'un autre. Le suivi d'un objet pourrait de ce fait être amélioré. Pour cela, il est nécessaire de connaitre les variations d'informations que peuvent subir un objet au cours du temps. Ainsi, la reconnaissance d'un objet passerait au préalable par une analyse de son contenu afin de déterminer la présence de l'objet dans les images précédentes.

A présent, le travail le plus important est de déterminer de quelle manière les informations fournies par notre pré-analyse doivent influer sur le codage d'une séquence en flux H.264. Ce mémoire donne quelques clés de réponse, mais il est loin de donner toutes les possibilités ouverte par l'utilisation de la pré-analyse.

Annexe A

Présentation des séquences vidéo utilisées lors des tests

Les séquences utilisées lors des tests réalisés pour les besoins de ce rapport 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 progressives de 720 lignes par 1280 colonnes, cadencées à 50 images par seconde, la structure d'échantillonnage couleur des composantes YUV est 4 :2 :0.

A.1.1 New mobil 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 présente une image extraite de la séquence *New mobil and calendar*.

A.1.2 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

[1] Le Vasa est un vaisseau de guerre scandinave du 17ème siècle.

FIG. A.1 – Image 478 de la séquence *New mobil and calendar*.

FIG. A.2 – Image 1 de la séquence *Knightshields*.

la fin de la séquence, le capteur effectue un zoom avant de la scène. La figure A.2 présente une image extraite de la séquence *Knightshields*.

A.2 La séquence 1080p

La séquences 1080p utilisée ici est une vidéo progressives de 1080 lignes par 1920 colonnes, cadencées à 25 images par seconde, la structure d'échantillonnage couleur des composantes YUV est également 4 :2 :0.

A.2.1 Tractor

La séquence comporte 761 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.3 présente une image extraite de la séquence *Tractor*.

FIG. A.3 – Image 60 de la séquence *Tractor*.

Bibliographie

[Bay63] T. Bayes. An essay towards solving a problem in doctrine of chances. *Philos. Trans. R. Soc. London*, 53 :293–315, 1763.

[BDR06] Olivier Brouard, Fabrice Delannay, and Vincent Ricordel. Rapport d'étude sur les méthodes de conditionnement et de pré-analyse du flux vidéo. *Premier délivrable du projet 'RIAM' ArchiPEG*, Septembre 2006.

[Bes74] Julian E. Besag. Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society, Series B*, 36 :196–236, 1974.

[Bes86] Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48 :No.,3 pp. 259–302, 1986.

[Cas98] Roberto Castagno. *Video segmentation based on multiple features for interactive and automatic multimedia applications*. PhD thesis, École Polytechnique Fédérale de Lausanne, Lausanne, 1998.

[Cha03] Marc Chaumont. *Représentation en objets vidéo pour un codage progressif et concurrentiel des séquences d'images*. PhD thesis, Université de Rennes I, Novembre 2003.

[CHC+05] Byeong-Doo CHOI, Min-Cheol HWANG, Jun-Ki CHO, Jin-Sam KIM, and KIM Jin-Hyung ; KO Sung-Jea. Realtime h.264 encoding system using fast motion estimation and mode decision. *Lecture notes in computer science ISSN 0302-9743*, pages 174–183, 2005.

[CJS98] H.D. Cheng, X. H. Jiang, and Y. Sung. A survey on color image segmentation. *The First International Workshop on CVPRIP*, 1998.

[CP95] J.P. Cocquerez and S. Phillip. *Analyse d'images : filtrage et segmentation*. Masson, 1995.

[CR87] Paul Chou and Rajeev Raman. On relaxation algorithms based on markov random fields. *Technical Report 212, Computer Science Department*, pages 140–15, 1987.

[ct06] SVT corporate technology. The svt high definition multi format test set. 2006.

[Don97] I. Donescu. *Modélisation de techtures invariantes par transformation plane. Codage sur des supports de forme arbitraire*. PhD thesis, Université de Rennes I, Decembre 1997.

[GGG87] Donald Geman, Stuart Geman, and Christine Graffigne. Locating texture and object boundaries. In *Proc. of the NATO Advanced Study Institute on Pattern recognition theory and applications*, pages 165–177, London, UK, 1987. Springer-Verlag.

[Har03] Harmonic. White paper on digital video solutions. *AVC + AAC The Next Generation of Compression*, 2003.

[HHW+03] Yu-Wen Huang, Bing-Yu Hsieh, Tu-Chih Wang, Shao-Yi Chen, Shyh-Yih Ma, Chun-Fu Shen, and Liang-Gee Chen. Analysis and reduction of reference frames for motion estimation in mpeg-4 avc/jvt/h.264. In *ICME '03 : Proceedings of the 2003 International Conference on Multimedia and Expo*, pages 809–812, Washington, DC, USA, 2003. IEEE Computer Society.

[HLC+06] Win-Bin Huang, Yi-Li Lin, Hung-Wei Cheng, A.W.Y. Su, and Yau-Hwang Kuo. Two-stage mode selection of h.264/avc video encoding with rate distortion optimization. *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference*, pages Volume : 2, On page : II–II, 2006.

[HS80] Martin Hassner and Jack Sklansky. The use of Markov random fields as models of texture. 12(4) :357–370, April 1980. HASSNER80.

[IR03] E. G. Iain and Richardson. *H.264 and MPEG-4 Video Compression : Video Coding for Next-generation Multimedia*. John Wiley and Sons, 2003.

[Ker97] C. Kermad. *Segmentation d'image : recherche d'une mise en oeuvre automatique par coopération de méthodes*. PhD thesis, ENSSAT, Université de Rennes 1, 1997.

[Lal90] P. Lalande. *Détection du mouvement dans les séquences d'images selon une approche markovienne ; application à la robotique sous-marine*. Thèse de doctorat, Université de Rennes I, 1990.

[LLLP05] A. Lehuger, P. Lechat, N. Laurent, and P. Pérez. Suivi de joueurs dans les séquences sportives à fort changement d'illumination : évaluation du problème et solutions. In *Proc. Journées Compression et Représentation des Signaux Visuels (CORESA'05)*, Rennes, France, November 2005.

[LM01] L. Lucchesey and S.K. Mitray. Color image segmentation : A state-of-the-art survey. *Image Processing, Vision, and Pattern Recognition*, pages Vol. 67 A, No. 2, Mar. 2001, pp. 207–221, 2001.

[LM05] Olivier LE MEUR. *Attention sélective en visualisation d'images fixes et animées affichées sur écran : modèles et évaluation des performances - applications*. PhD thesis, Université de Nantes, Ecole polytechnique de l'université de Nantes, 2005.

[MD94] R. Megret and D. DeMenthon. A survey of spatio-temporal grouping techniques, 1994.

[MHM03] B. S. Manjunath, G. M. Haley, and W. Y. Ma. *H.264 and MPEG-4 Video Compression : Video Coding for Next-generation Multimedia*. John Willey, 2003.

[MYKP06] Z.-Y. Mai, C.-L. Yang, K.-Z. Kuang, and L.-M. Po. A novel motion estimation method based on structural similarity for h.264 inter prediction. *IEEE International*

Conference on Acoustics, Speech, and Signal Processing, pages Vol. 2, pp. 913–916, 2006.

[NT05] Özbek Nükhet and A. Murat Tekalp. Fast h.264/avc video encoding with multiple frame references. In *ICIP (1)*, pages 597–600, 2005.

[OT05] N. Ozbek and A.M. Tekalp. Fast h.264/avc video encoding with multiple frame references. *Image Processing, 2005. ICIP 2005. IEEE International Conference*, pages 597–600, 2005.

[PCBY04] G. Piriou, F. Coldefy, P. Bouthemy, and J-F. Yao. Détection supervisée d'événements à l'aide d'une modélisation probabiliste du mouvement perçu. In *14ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle, RFIA 2004*, Toulouse, France, January 2004.

[PL97] Stéphane Pateux and Claude Labit. *Codage efficace de carte de segmentation pour la compression orientée régions de séquences d'images*. PhD thesis, Université de Rennes, Janvier 1997.

[PLCCB07] Stéphane Péchard, Patrick Le Callet, Mathieu Carnec, and Dominique Barba. A new methodology to estimate the impact of h.264 artefacts on subjective video quality. In *Proceedings of the Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics, VPQM2007*, Scottsdale, 2007.

[RMC02] M. Ravasi, M. Mattavelli, and C. Clerc. A computational complexity comparison of mpeg4 and jvt codecs. *Doc.JVT-D153r1-L, Joint Video Team (JVT) of ISO/CEI MPEG & UIT-T VCEG (ISO/CEI CTM1/ SC29/GT11 et UIT-T SG16 Q.6)*, July 2002.

[Rou97] Ludovic Roux. *Fusion d'informations multisource pour l'interprétation d'images* . Thèse de doctorat, Université Paul Sabatier, Toulouse, France, janvier 1997.

[Ser06] Jean Serra. A lattice approach to image segmentation. *J. Math. Imaging Vis.*, 24(1) :83–130, 2006.

[The83] Charles W. Therrien. An estimation-theoretic approach to terrain image segmentation. 22(3) :313–326, June 1983. THERRIEN83.

[TTB03] Peng Yin Tourapis, H.-Y.C. Tourapis, and A.M. Boyce. Fast mode decision and motion estimation for jvt/h.264. *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference*, 2003.

[XPR+05] Feng Xiao, Lin Pan, S. Rahardja, K.P. Lim, Z.G. Li, and Wu Dajun. Fast mode decision algorithm for intra prediction in h.264/avc. *Circuits and Systems for Video Technology, IEEE Transactions*, pages 813– 822, 2005.

[YFZ03] Yu Yuan, David Feng, and Yuzhou Zhong. Three fast methods for adaptive key frame setting and dynamic frame-rate adjusting in video coding. *International Journal of Computational Intelligence*, pages volume 1 de ISSN : 1304–4508., 2003.

[YM04] A. C. Yu and G. R. Martin. Efficient block size selection algorithm for inter-frame coding in h.264/avc. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP),Montreal, Canada*, pages vol. 3, pp. 17Ű21., 2004.

[Yu04] A.C. Yu. Efficient block-size selection algorithm for inter-frame codingin h.264/mpeg-4 avc. *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference*, 2004.

Table des figures