Université de Nantes

–

École polytechnique de l'université de Nantes

Projet RIAM ArchiPEG

(convention ANR05RIAM014xx)

Lot 4.2 : Rapport sur les définitions d'algorithmes de filtrage et de pré-analyse du flux vidéo

F. Delannay, O. Brouard, V. Ricordel et D. Barba

Laboratoire IRCCyN - équipe IVC

avril 2007

Table des matières

Table des figures

Introduction générale

Les travaux présentés dans ce rapport ont été réalisés dans le cadre du projet RIAM ArchiPEG qui relève de la convention ANR05RIAM014xx. Ils correspondent à la seconde tâche du sous-projet 4 intitulé : Pré-analyse et conditionnement du flux vidéo en haute définition.

Le dernier standard de codage vidéo développé par le JVT (Joint Video Team) regroupant les experts MPEG et ITU, 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 [1, 2] 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. Le schéma du codeur H.264 est présenté en figure 1.

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 : elle peut aller jusqu'au au quart de pixel pour la luminance et jusqu'au huitième de pixel pour la chrominance ;
- la taille variable des blocs estimés : 7 modes pour la prédiction inter (16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4) et 2 modes pour la prédiction intra (16×16, 4×4) ;
- la sélection des images de référence : le choix de l'image de référence intervient au niveau macrobloc et sous-macrobloc contrairement aux normes précédentes telles que MPEG-2.

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[2] 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

[1] TVHD : télévision haute-définition.

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

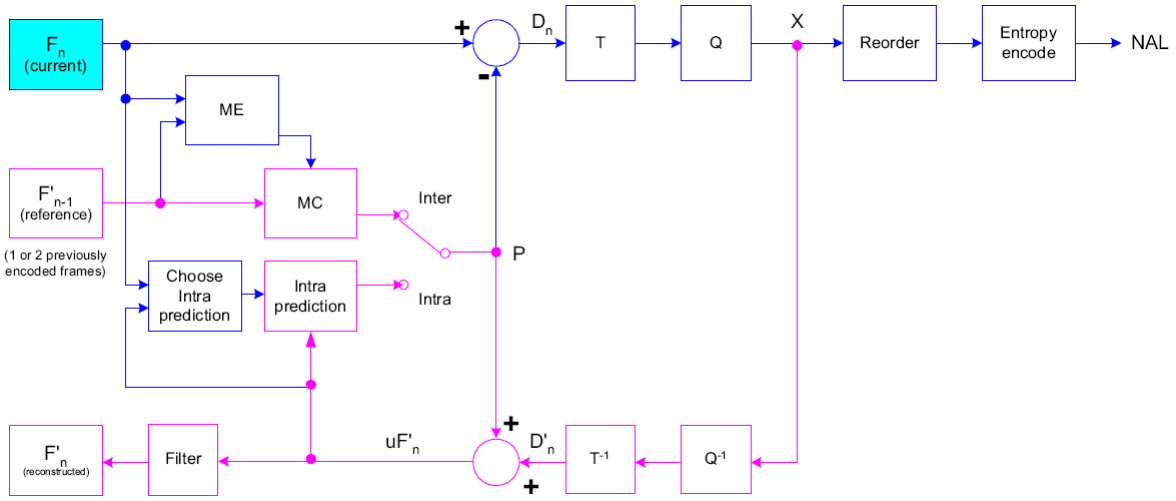


FIG. 1 – Schéma du codeur H.264/AVC [2].

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 document présente les algorithmes adaptés à la pré-analyse de flux vidéo haute définition en vue d'un encodage en temps réel sous le standard H.264. L'objectif 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. Le premier chapitre de ce rapport présentera les spécifications fonctionnelles de l'outil de pré-analyse pour le conditionnement du flux vidéo. Chacune des méthodes présentées et assemblées dans ces spécifications, fera l'objet d'un chapitre dans lequel elle sera entièrement détaillée.

Chapitre 1

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

1.1 Introduction

Afin de privilégier une stratégie de codage H.264 cohérente avec l'activité spatio-temporelle présente dans la séquence vidéo à coder, 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 adaptés à la vidéo traitée. Idéalement, cet outil devra fournir au codeur les informations nécessaires pour réduire considérablement le nombre de modes testés (inter et intra), et indiquer les images qui doivent être marquées comme références (pour les modes inter). On pourra également envisager qu'un tel outil soit capable d'attribuer à chaque objet un paramètre de quantification (QP) adapté. 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.

1.2 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 1.1.

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'. La définition et la conception d'un tel outil ne seront pas abordées dans le présent document.

Après avoir définit le comportement de l'outil de pré-analyse vis-à-vis de son environnement, il convient de définir son comportement interne. C'est l'objet de la section suivante.

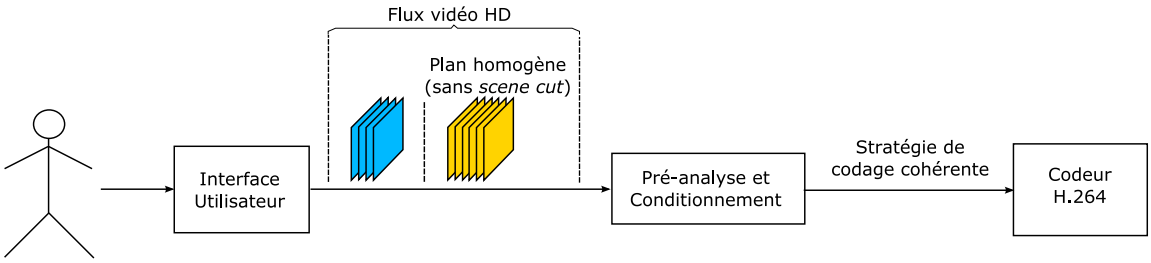


FIG. 1.1 – Spécification externe de l'outil de pré-analyse et de conditionnement d'un flux vidéo.

1.3 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, nous nous basons sur le temps de fixation du système visuel humain qui est sensiblement égal à 200ms [3]. 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 soit d'un traitement intra, soit 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). Afin que le système de pré-analyse puisse s'assurer du bien fondé de l'utilisation d'un mode inter plutôt que d'un mode intra, le segment temporel courant devra bénéficier périodiquement des deux traitements conjoints (intra puis inter). Ce double traitement permettra de vérifier que la redondance temporelle entre le segment courant et le segment précédent est assez importante pour employer le mode de traitement inter. Dans le cas contraire, le mode intra permettra donc de détecter une corrélation trop faible entre le segment précédent et le segment courant pour initialiser efficacement le traitement inter, les résultats du mode intra seront alors les seuls retenus.

Une fonction "switch" permettra, en fonction de l'analyse du segment temporel précédent, de sélectionner le mode le plus probable pour le traitement du segment courant. Que le mode de traitement sélectionné soit intra ou inter, ce dernier devra être capable de fournir, pour chaque segment temporel, 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 1.2.

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

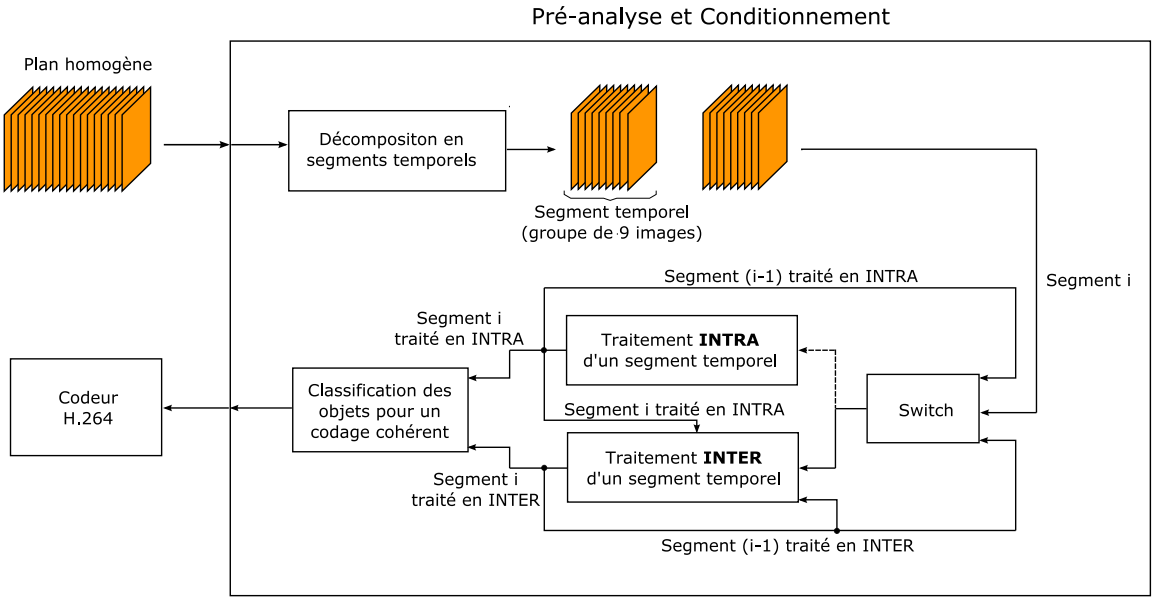


FIG. 1.2 – Conception détaillée de l'outil de pré-analyse et de conditionnement d'un flux vidéo.

1.4 Conclusion

La décomposition fonctionnelle de notre outil de pré-analyse présente principalement trois fonctions importantes : le traitement intra, le traitement inter et la classification des objets pour un codage cohérent (figure 1.2). Le traitement intra et le traitement inter d'un segment temporel fourniront au bloc de classification des informations de natures similaires. La différence majeure réside dans la capacité du bloc de traitement inter à exploiter la redondance temporelle existante entre les segments temporels successifs et donc d'envoyer des informations relatives au suivi temporel d'objets sur plus de 180ms.

Dans les chapitres suivants du présent rapport, nous présenterons donc une description détaillée de chacun de ces trois blocs de base. L'état d'avancement actuel de nos travaux, nous permet de définir fonctionnellement entièrement le bloc de traitement intra et de présenter les différents algorithmes utilisés. Les blocs de traitement inter et de classification seront eux présentés de façon moins détaillée. Leur cahier des charges, et les méthodes envisagées pour leur mise en oeuvre seront présentés.

Chapitre 2

Présentation du bloc de traitement intra-segment temporel

2.1 Introduction

Le bloc de traitement intra doit être capable, pour chaque segment temporel de 9 images, de fournir une décomposition en objets homogènes selon des critères de mouvement, de texture et de couleur. 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 la robustesse des résultats de la segmentation sont alors améliorées. Pour ces techniques basées long-terme, on cherche à obtenir des tubes spatio-temporels, c'est-à-dire, des régions ou des objets qui ont une texture et un mouvement homogènes et stables sur plusieurs images, de manière à lisser les mouvements estimés et à obtenir des informations plus corrélées avec les mouvements réels de la séquence vidéo. Dans notre cas, nous considérons que l'unité élémentaire à suivre temporellement est le macrobloc[1], un objet spatio-temporel sera donc composé d'un ensemble de tubes dont les propriétés de mouvement, de couleur, et de texture sont homogènes.

Dans un premier temps, le bloc de traitement intra doit donc effectuer 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. 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. Enfin, les résultats fournis par la segmentation basée mouvement sont affinés en les utilisant conjointement avec des critères de couleur et de texture. La figure 2.1 présente la décomposition en schéma bloc du traitement intra d'un segment temporel.

[1] Bloc de 16×16 pixels.

Les prochaines sections de ce chapitre vont présenter de façon approfondie, les quatre méthodes majeures qui constituent le bloc de traitement intra.

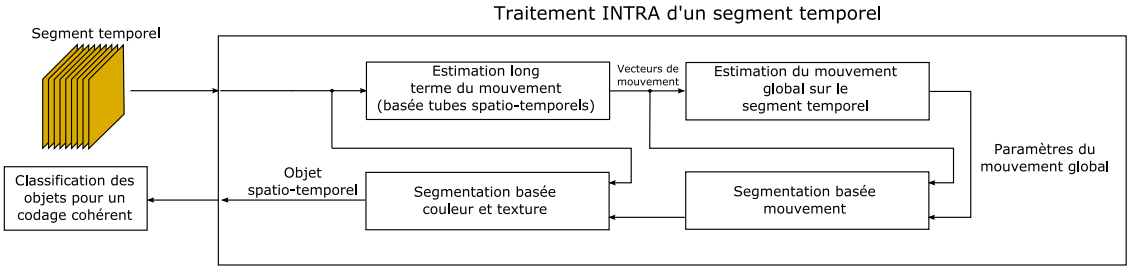


FIG. 2.1 – Traitement INTRA d'un segment temporel de 9 images.

2.2 Estimation long terme du mouvement

2.2.1 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, nous utiliserons des méthodes d'estimation de mouvement basées sur le *block-matching*, dont l'objectif est 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 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, nous pouvons agrandir la taille de la fenêtre de recherche, 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[2] pour trouver la meilleure prédiction du bloc courant dans l'image de référence long terme.

2.2.2 Estimation de mouvement multi-résolution

Nous proposons une méthode multi-résolution [4], afin d'accélérer les temps de calcul pour l'estimation de mouvement de séquences HD. Les images HD de la séquence sont sous-échantillonnées

[2]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.

spatialement d'un facteur 6. Ce sous-échantillonnage est réalisé en deux passes : application d'un sous-échantillonnage d'un facteur 2, puis d'un sous-échantillonnage d'un facteur 3. Avant chaque passe, un filtre passe-bas adapté (filtre demi-bande puis tiers de bande) est appliqué afin de réduire les problèmes de repliement (*aliasing*) liés au sous-échantillonnage.

L'estimation de mouvement est alors réalisée sur les images basse-résolution. Pour initialiser la recherche des mouvements sur un segment temporel entier, nous utilisons cinq images successives sur lesquelles seront construits des tubes spatio-temporels (figure 2.2). 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.

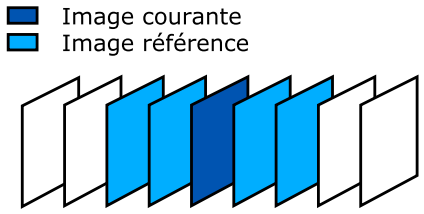


FIG. 2.2 – Image courante et images références d'un segment temporel (contexte court-terme).

Nous considérons que chaque macrobloc de l'image courante possède un mouvement uniforme[3] entre les cinq images considérées. Pour chaque macrobloc de l'image courante, le tube retenu est celui qui minimise la fonction de coût global $MSE_G(i,j)$ associée au centre (i,j) du macrobloc courant et définie par l'équation 2.1 :

$$MSE_G(i,j) = \sum_k MSE_k(i,j), \quad k = -2,-1,+1,+2 \tag{2.1}$$

La fonction de coût global est donc la somme de quatre fonctions de coût élémentaires MSE_k, chacune étant calculée entre le macrobloc de l'image courante et le macrobloc correspondant dans l'image référence indexée par k. L'indice k donne la position temporelle de l'image référence relativement à la position de l'image courante. Chaque fonction de coût élémentaire MSE_k prend en compte les trois composantes YUV d'un bloc de taille N×N selon l'équation 2.2 :

$$MSE_k(i,j) = \frac{1}{N \times N} \cdot \sum_{Z=0}^{2} \left[\sum_{i,j=0}^{N-1} \left[C_Z(i,j) - R_{k,Z}(i_c,j_c) \right]^2 \right] \tag{2.2}$$

C_0, C_1, C_2, $R_{k,0}$, $R_{k,1}$, et $R_{k,2}$ représentent respectivement les trois composantes YUV de l'image courante et de l'image k utilisée comme référence pour l'estimation de mouvement. La position (i_c,j_c) correspond à la position (i,j) compensée dans l'image de référence R_k. En considérant que le vecteur déplacement, entre l'image courante à l'instant t et l'image qui la précède à l'instant t-2, a pour

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

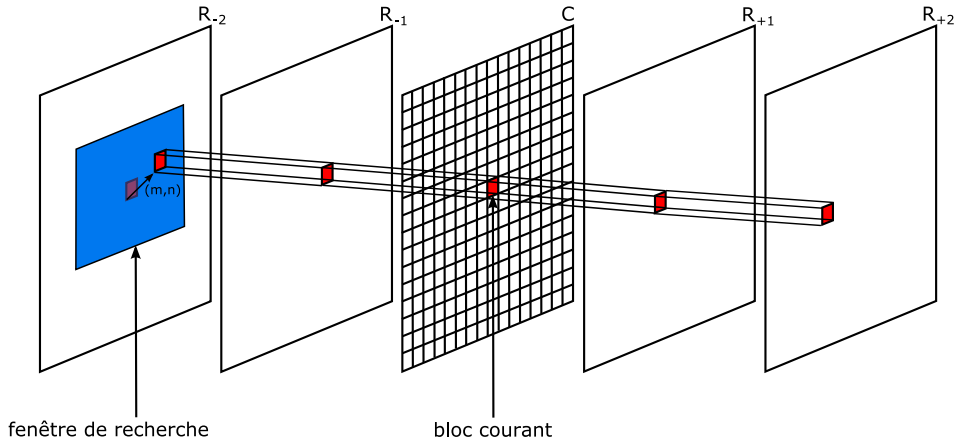


FIG. 2.3 – Représentation d'un tube spatio-temporel et du vecteur mouvement associé.

composantes (m,n) (cf. figure 2.3), la position compensée (i_c,j_c) dans une image référence R_k est donnée par l'équation 2.3 :

$$\begin{cases} i_c = \lfloor i - \frac{k}{2}.m \rfloor \\ j_c = \lfloor j - \frac{k}{2}.n \rfloor \end{cases} \tag{2.3}$$

Le vecteur mouvement retenu pour chaque macrobloc est donc celui qui minimise le coût global MSE_G. Les vecteurs de mouvement étant calculés à la plus basse résolution, un ré-échelonnage adéquat de ces derniers est nécessaire. Le champ de vecteurs mouvement à la résolution initiale (HD) est alors obtenu.

2.2.3 Notre méthode d'estimation long-terme de mouvement

L'objectif est d'améliorer les performances d'une estimation long-terme de mouvement classique, en cherchant un point d'initialisation optimal. Si l'image de référence et l'image courante sont proches temporellement (e.g. immédiatement successives), leurs contenus sont généralement très fortement corrélés. Le vecteur qui donne le déplacement en pixels d'un macrobloc de l'image courante vers son correspondant dans l'image de référence est donc faible. Dans ce cas, la fenêtre de recherche n'a pas besoin d'être trop large et est centrée sur le bloc courant à prédire. Cependant, dans le cas d'une image de référence éloignée temporellement de l'image courante, la corrélation spatiale entre les contenus des deux images peut diminuer fortement. Dans ce cas, il devient important de prédire un point initial de recherche, qui est souvent différent du centre du bloc courant. Sans initialisation appropriée, il faudrait considérablement augmenter la taille de la fenêtre de recherche autour du bloc courant, afin de ne pas tomber dans un minimum local lors de l'étape de *block matching* [2]. Pour des écarts temporels assez importants, la fenêtre de recherche pourrait même recouvrir une image HD entière et les temps de calculs exploseraient.

Notre approche repose sur l'utilisation des vecteurs de mouvement obtenus avec des références immédiates pour initialiser la recherche à long terme. Ces vecteurs initiaux sont calculés en appliquant

la méthode d'estimation multi-résolution présentée précédemment. Les vecteurs mouvement pour une référence long-terme sont alors prédits à partir d'un ré-échelonnement linéaire des vecteurs initiaux calculés pour des références proches. En notant $\mathrm{MV}(t, t-k)$ le vecteur mouvement entre l'image courante à l'instant t et l'image référence à l'instant t-k, le vecteur mouvement prédit $\mathrm{MV}_{\mathrm{pred}}$ pour une référence long terme est donné par l'équation 2.4 :

$$MV_{pred}(t, t-k) = k \times MV(t, t-1) \tag{2.4}$$

Le vecteur de mouvement prédit permet de localiser la position de recherche initiale dans l'image de référence long terme (voir figure 2.4). Dès lors, une possibilité est de centrer la fenêtre de recherche sur ce point initial pour affiner le vecteur mouvement prédit entre l'image courante et l'image référence. Cependant, cette méthode double la charge de calculs nécessaires lors d'une simple prédiction à court terme. En effet, elle met en oeuvre deux *block matching full search* successivement. Nous avons donc choisi d'affiner la prédiction du vecteur mouvement long terme avec une méthode rapide d'optimisation non linéaire afin d'éviter le calcul d'un second *block matching full search*. La méthode retenue est celle du simplex de Nealder et Mead [5]. Cette méthode tente de réduire une fonction[4] non linéaire de n variables réelles en utilisant simplement quelques valeurs de la fonction à minimiser et aucune information sur ses dérivées.

Notre méthode d'estimation du mouvement, dans un contexte long-terme, repose donc sur 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. Le déplacement long terme est alors estimé autour de ce point initial à l'aide d'une méthode de minimisation rapide de fonction multi-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, c'est l'objet de la prochaine section.

2.3 Estimation et compensation du mouvement global

2.3.1 Notion de 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, nous souhaitons 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 2.5). La segmentation à partir de ces vecteurs mènera donc à la détection de nombreux objets de mouvements différents.

[4] Dans le cas du *block-matching*, la fonction à minimiser est l'erreur entre le bloc courant et sa prédiction dans l'image de référence.

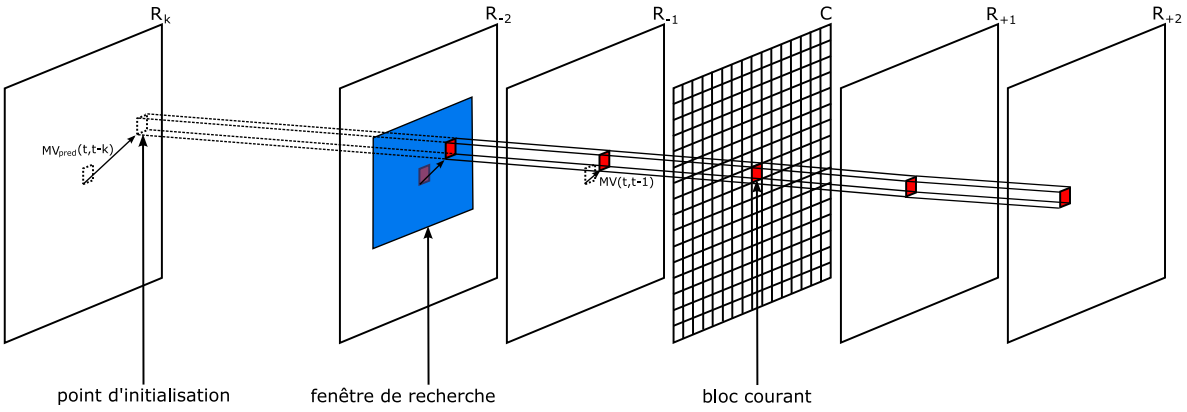


FIG. 2.4 – Initialisation de l'estimation à long terme.

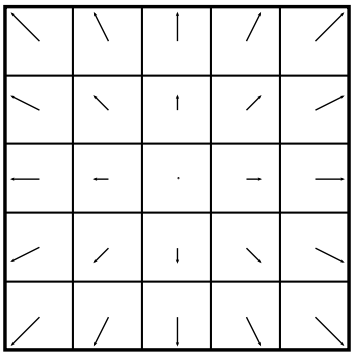


FIG. 2.5 – Champ épars de vecteurs associé à un zoom sur une image décomposée en macroblocs.

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 est 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.

2.3.2 Estimation du mouvement global par accumulation

2.3.2.1 Modèles paramétriques de mouvement global

La caméra peut effectuer des mouvements dans un espace à trois dimensions, alors que les déplacements perçus dans une séquence vidéo n'ont que deux dimensions. De fait, les images acquises

correspondent à la projection de la scène réelle dans le plan focale de la caméra. Dans ce contexte, si la caméra est complètement libre dans ses déplacements, plusieurs mouvements ne pourront pas être estimés correctement. Le modèle quadratique (équation 2.5) est le modèle paramétrique qui permet d'estimer au mieux n'importe quel mouvement à trois dimensions pour un objet rigide, projeté dans un espace à deux dimensions.

Les vecteurs déplacement fournis par notre méthode d'estimation de mouvement long terme sont corrélés avec les déplacement réels des objets dans la scène, mais ces vecteurs ne sont pas assez précis pour mettre en oeuvre un modèle d'estimation aussi complexe que le modèle quadratique. Nous choisissons donc d'utiliser le modèle affine à six paramètres (équation 2.6).

$$\begin{pmatrix} V_x \\ V_y \end{pmatrix} = \begin{pmatrix} a_5 + a_1 x + a_2 y + a_7 x^2 + a_8 xy \\ a_6 + a_3 x + a_4 y + a_8 y^2 + a_7 xy \end{pmatrix} \tag{2.5}$$

$$\begin{pmatrix} V_x \\ V_y \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{2.6}$$

L'équation 2.6 donne le déplacement (V_x,V_y) d'un point à la position (x,y) en fonction de six paramètres liés au mouvement global. Le modèle affine réduit le nombre de mouvements de la caméra à trois types : les translations (t_x et t_y), les rotations (a_2, a_3) et les zooms (a_1, a_4). Nous allons adapter la méthode de Coudray [6] pour estimer ces six paramètres.

2.3.2.2 Indices de confiance pour une estimation robuste

Pour chaque macrobloc, le vecteur déplacement associé est celui qui minimise une fonction de coût[5] lors de l'estimation de mouvement long-terme. Dans le cas de macroblocs situés dans des zones de faibles textures et de couleur uniforme, cette minimisation ne permet généralement pas d'estimer le mouvement réel des objets. En effet, tous les macroblocs de la zone uniforme dans l'image de référence minimisent la fonction de coût et c'est généralement le macrobloc le plus proche du macrobloc courant qui est retenu. Pour donner plus d'importance aux vecteurs déplacement situés dans des zones fortement texturées qu'à ceux situés dans des zones homogènes, nous utilisons une fonction de pondération qui attribuera un indice de confiance au vecteur déplacement selon l'activité spatiale de la zone à laquelle il est associé. L'activité spatiale d'un macrobloc sera évaluée par le calcul de gradients orientés horizontalement ($\overline{\Delta H}$) et verticalement ($\overline{\Delta V}$)[6]. Plus le gradient horizontal (resp. vertical) sera fort, plus la confiance accordée à la composante V_x (resp. V_y) du vecteur déplacement sera importante. L'indice de confiance associé à une composante d'un vecteur déplacement varie entre 0 et 1. L'application ψ (équation 2.7) est utilisée pour attribuer les valeurs de fiabilité en fonction de la force gradient, elle est illustrée en figure 2.6.

$$\psi(x) = \begin{cases} \left(\frac{x}{8}\right)^3 /2, & x \leq 8 \\ 1 - \psi(16 - x), & 8 < x \leq 16 \\ 1, & sinon \end{cases} \tag{2.7}$$

[5] Classiquement une somme des différences absolues ou une erreur quadratique moyenne, dans notre cas MSE_G (cf. section2.2).

[6] Le calcul de ces gradients est présenté en section 2.5.2.

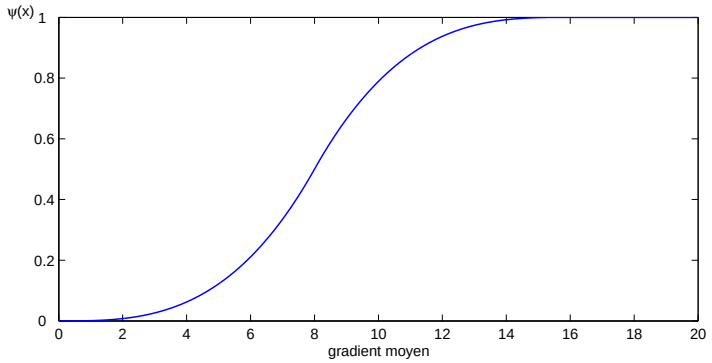


FIG. 2.6 – Évolution des indices de confiance en fonction de la valeur du gradient spatial.

La figure 2.7 donne une représentation visuelle des indices de confiance associés aux composantes V_x et V_y pour une image de la séquence vidéo HD *Shields*. Les zones très texturées et correctement orientées sont blanches (indice de confiance proche de 1) tandis que les zones homogènes sont sombres (indice de confiance proche de 0).

(a) Image originale

(b) Indices de confiance pour Vx

(c) Indices de confiance pour Vy

FIG. 2.7 – Illustration du calcul des indices de confiance sur une image de la séquence *Shields*.

2.3.2.3 Accumulation de paramètres pondérés

L'information élémentaire utilisée pour estimer le mouvement global est un champ de vecteurs mouvement (un vecteur par macrobloc). Nous estimons les paramètres du modèle affine à partir du champ initial de vecteurs en utilisant les équations 2.8 et 2.9 :

$$\begin{cases} a_1 & = & \partial V_x/\partial x \\ a_2 & = & \partial V_x/\partial y \\ a_3 & = & \partial V_y/\partial x \\ a_4 & = & \partial V_y/\partial y \end{cases} \tag{2.8}$$

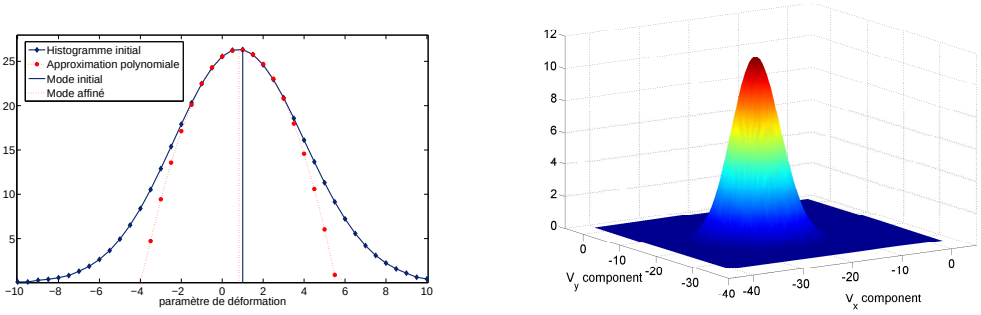
$$\begin{cases} t_x & = & V_x - a_1 x - a_2 y \\ t_y & = & V_y - a_3 x - a_4 y \end{cases} \tag{2.9}$$

Nous avons vu que pour un modèle affine du mouvement global (équation 2.6), les déplacements peuvent être de trois natures différentes : zoom (a_1, a_4), rotation (a_2, a_3) ou translation (t_x, t_y). L'équation 2.9 indique que les paramètres de déformation a_1, a_2, a_3 et a_4 affectent les valeurs des paramètres de translation. L'estimation du mouvement global est donc réalisée en deux étapes. Dans un premier temps, nous estimons les paramètres relatifs au mouvement global de déformation. Chaque macrobloc fournit une information locale sur ces quatre paramètres de déformation (équation 2.8), chaque dérivée calculée sur un vecteur mouvement va donc apporter une hypothèse pour un des paramètres de déformation. En pratique, les quatre hypothèses de chaque macrobloc sont évaluées en calculant la différence entre les composantes du vecteurs du macrobloc courant avec celles de son voisin direct (équation 2.10).

$$\begin{cases} a_1 = V_x(n+1,m) - V_x(n,m) \\ a_2 = V_x(n,m+1) - V_x(n,m) \\ a_3 = V_y(n+1,m) - V_y(n,m) \\ a_4 = V_y(n,m+1) - V_y(n,m) \end{cases} \tag{2.10}$$

où (n, m) désigne la position du macrobloc courant sous la forme (abscisse, ordonnée) du macrobloc. Pour connaître l'hypothèse la plus redondante et donc celle qui en probabilité représente le mieux le paramètre de mouvement global, toutes les hypothèses sont accumulées dans un histogramme (un histogramme par paramètre de déformation). Les hypothèses émises par le calcul des dérivées participent à l'accumulation proportionnellement à la confiance que l'on peut leur donner. Comme chaque dérivée est calculée par la différence entre deux composantes de deux vecteurs, la confiance attribuée à une hypothèse est le résultat du produit des indices de fiabilité de chaque composante. Afin de rassembler les hypothèses proches, chaque hypothèse est accumulée dans l'histogramme avec une distribution gaussienne.

Considérons un macrobloc (n, m) dont l'activité spatiale est caractérisée par les gradients moyens $\overline{\Delta V}(n, m)$ et $\overline{\Delta H}(n, m)$. Ce macrobloc émet respectivement quatre hypothèses a_1(n, m), a_2(n, m), a_3(n, m) et a_4(n, m) sur les valeurs globales des quatre paramètres de déformation a_1, a_2, a_3 et a_4. Ces quatre hypothèses participent respectivement aux accumulations dans les histogrammes h_1, h_2, h_3 et h_4 selon les relations données en équation 2.11.



(a) accumulation pour estimer un paramètre de déformation

(b) accumulation pour estimer un paramètre de translation

FIG. 2.8 – Espaces d'accumulation pour l'estimation des paramètres du mouvement global (segment extrait de la séquence *Shields*).

$$\begin{cases} h_1(x) \leftarrow h_1(x) + \psi(\overline{\Delta H}(n+1,\,m)).\psi(\overline{\Delta H}(n,\,m)).\sqrt{\frac{1}{2\pi\sigma^2}}e^{-(x-a_1(n,\,m))^2/2\sigma^2} \\ h_2(x) \leftarrow h_2(x) + \psi(\overline{\Delta H}(n,\,m+1)).\psi(\overline{\Delta H}(n,\,m)).\sqrt{\frac{1}{2\pi\sigma^2}}e^{-(x-a_2(n,\,m))^2/2\sigma^2} \\ h_3(x) \leftarrow h_3(x) + \psi(\overline{\Delta V}(n+1,\,m)).\psi(\overline{\Delta V}(n,\,m)).\sqrt{\frac{1}{2\pi\sigma^2}}e^{-(x-a_3(n,\,m))^2/2\sigma^2} \\ h_4(x) \leftarrow h_4(x) + \psi(\overline{\Delta V}(n,\,m+1)).\psi(\overline{\Delta V}(n,\,m)).\sqrt{\frac{1}{2\pi\sigma^2}}e^{-(x-a_4(n,\,m))^2/2\sigma^2} \end{cases} \tag{2.11}$$

Pour chaque histogramme, la localisation du maximum donne la valeur retenue pour le paramètre global de déformation étudié. Pour affiner la localisation du mode, nous estimons la courbure autour de la position du maximum à l'aide de la méthode des Moindres Carrés (figure 2.8).

Les quatre paramètres de déformation calculés permettent de compenser le champ de vecteurs initial (équation 2.9). Les vecteurs ainsi compensés représentent donc théoriquement les seuls mouvements de translation de la caméra et les déplacements locaux des objets. En conservant le même principe que pour l'estimation des paramètres de déformation, nous considérons que la valeur de translation la plus redondante représentera le mouvement global. Les vecteurs compensés sont alors accumulés dans un histogramme à deux entrées pour ne pas fausser la représentativité des mouvements[7]. L'accumulation pour chaque vecteur compensé est faite proportionnellement au minimum des indices de fiabilité de ses composantes. Une distribution gaussienne en deux dimensions est également utilisée lors de l'accumulation des données, toujours afin de regrouper les hypothèses proches. Les valeurs des paramètres de translation sont alors données par la position du maximum dans l'espace d'accumulation (figure 2.8). Les vecteurs déplacement initiaux peuvent donc à présent être compensés par les paramètres de déformation et de translation de la caméra. Ainsi, les vecteurs compensés par les quatre paramètres de déformation et les deux paramètres de translation, représentent uniquement les déplacement locaux, nous pouvons donc effectuer une segmentation des objets au sens du mouvement plus efficace.

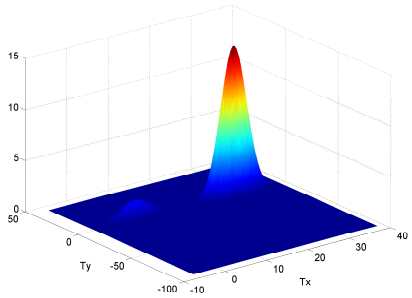
[7] Deux vecteurs peuvent avoir une de leurs deux composantes identique mais ne pas représenter le même mouvement.

2.4 Segmentation au sens du mouvement

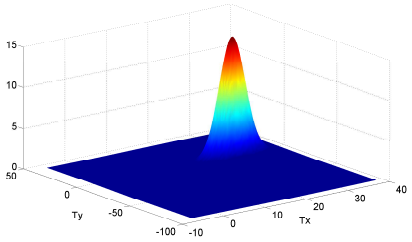
Lors de l'estimation du mouvement global, nous avons déterminé les paramètres de translation en localisant le maximum de l'histogramme d'accumulation des vecteurs compensés par les paramètres de déformation. Si nous n'étudions plus uniquement le pic le plus important mais tous les pics, alors une segmentation 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.

La première étape consiste à éliminer le bruit. Un seuil de rejet est défini empiriquement et toutes les cellules représentant une accumulation inférieure à ce seuil sont mises à zéro. Afin de ne pas utiliser une méthode de segmentation trop coûteuse en termes de complexité de calcul, nous utilisons un algorithme récursif qui va traiter les pics par ordre décroissant. Le premier pic détecté est donc celui qui correspond au maximum global de l'espace d'accumulation. Pour toutes les positions connexes à ce pic, le gradient[8] 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 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 2.9 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.

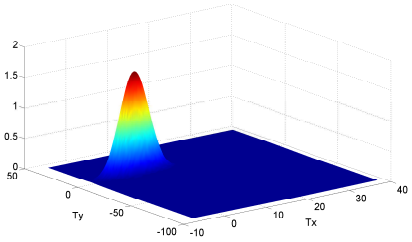
[8] Le gradient est ici une différence entre les populations de deux cellules de l'espace d'accumulation.



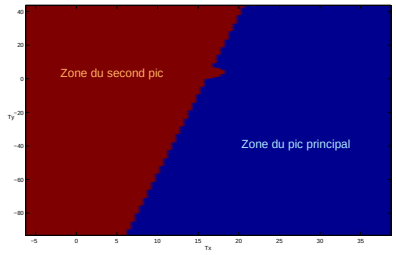
(a) Espace d'accumulation



(b) Pic principal (mouvement global)



(c) Second pic (mouvement local)



(d) Espace d'accumulation segmenté

FIG. 2.9 – Analyse récursive de l'espace d'accumulation.

La dernière étape consiste à segmenter le champ de vecteurs compensés par les paramètres de déformation, à partir de la séparation des différents 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. Á partir des deux pics présentés en figure 2.9, nous créons l'image segmentée présentée en figure 2.10. 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

FIG. 2.10 – Image segmentée de la séquence *Shields*

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 Mobile & Calendar* offre un contenu complexe. La caméra effectue un mouvement de *zoom out* sur une tapisserie et un calendrier uniformes. La segmentation au sens du mouvement est donc moins probante que celle réalisée précédemment pour la séquence *Knightshields* (figure 2.11). 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.

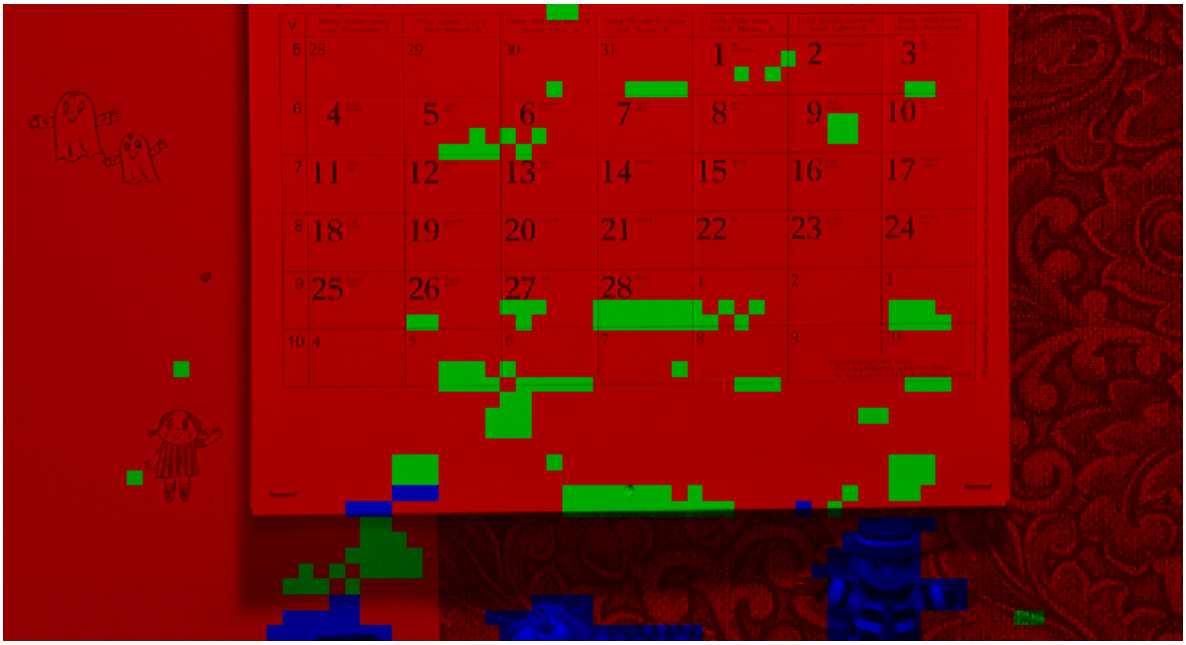


FIG. 2.11 – Image segmentée de la séquence *New Mobile & Calendar*

2.5 Critères de couleur et de texture

2.5.1 Contexte

Comme nous l'avons vu dans la section précédente, une segmentation au sens du mouvement seule peut ne pas suffire pour créer la décomposition 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 pour être rattachés efficacement à l'un des objets détectés par la segmentation au sens du mouvement (figure 2.11). 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. Ces informations supplémentaires permettront par exemple au bloc de classification de calculer, pour chaque objet, un pas de quantification adapté à son contenu spatial (homogène ou texturé) qui sera transmis au codeur H.264 (cf. section 4.2).

2.5.2 Caractérisation de la texture d'un macrobloc

Á cette étape de la pré-analyse du flux, chaque objet est constitué d'un ensemble de macroblocs dont les mouvements sont proches. Pour agglomérer de nouveaux macroblocs à un objet, ou encore

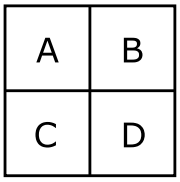


FIG. 2.12 – Bloc de 2×2 pixels.

diviser un objet en sous-objets, nous caractérisons chaque macrobloc par son activité spatiale. Par activité spatiale, nous désignons le contenu en fréquences spatiales du macrobloc et l'orientation de ses textures.

Pour estimer cette activité spatiale, quatre gradients sont calculés sur chaque pixel d'un macrobloc [4] :

- 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, un gradient moyen est alors calculé en fonction des gradients calculés pour chaque pixel. Par exemple, pour le bloc 2×2 présenté en figure 2.12, les quatre gradients moyens $\overline{\Delta V}$, $\overline{\Delta H}$, $\overline{\Delta D_{45}}$ et $\overline{\Delta D_{135}}$ sont calculés selon l'équation 2.12 :

$$\begin{cases} \overline{\Delta V} = \frac{|A-B|+|C-D|}{2} \\ \overline{\Delta H} = \frac{|A-C|+|B-D|}{2} \\ \overline{\Delta D_{45}} = |B-C| \\ \overline{\Delta D_{135}} = |A-D| \end{cases} \tag{2.12}$$

La catégorie d'un macrobloc est déterminée en utilisant les valeurs de ses gradients moyens $\overline{\Delta H}$ et $\overline{\Delta V}$. Pour cela, nous utilisons la partition plane P = ($\overline{\Delta H}$,$\overline{\Delta V}$) présentée en figure 2.13. Conformément à sa position dans le plan P = ($\overline{\Delta H}$,$\overline{\Delta V}$), un bloc situé dans la zone C_0 est un bloc au contenu spatial lisse et homogène, un bloc situé dans la zone C_1 est un bloc faiblement texturé, un bloc situé dans la zone C_3 est un bloc qui contient des contours horizontaux, un bloc situé dans la zone C_4 est un bloc qui contient des contours verticaux et un bloc situé dans la zone C_2 est un bloc au contenu indéterminé. Afin de caractériser un bloc de la zone C_2, nous utilisons les gradients moyens $\overline{\Delta D_{45}}$ et $\overline{\Delta D_{135}}$ de la même manière avec la même partition plane nommée P' = ($\overline{\Delta D_{45}}$,$\overline{\Delta D_{135}}$). Dans ce nouveau plan, un bloc situé dans la zone C_0 est un bloc au contenu spatial lisse et homogène, un bloc situé dans la zone C_1 est un bloc faiblement texturé, un bloc situé dans la zone C_2 est un bloc fortement texturé, un bloc situé dans la zone C_3 est un bloc qui contient des contours diagonaux orientés à 135° et un bloc situé dans la zone C_4 est un bloc qui contient des contours diagonaux orientés à 45°.

La description de la texture d'un macrobloc est donc établie à l'aide de quatre gradients moyens qui vont permettre de définir deux plans de caractérisation P et P'. Notons que le plan P a déjà été défini lors de l'étape d'estimation et de compensation du mouvement global (section 2.3). En effet, pour assigner un indice de confiance au vecteur déplacement d'un macrobloc, nous avons déjà calculé les gradients moyens $\overline{\Delta H}$ et $\overline{\Delta V}$. La méthode de caractérisation de texture d'un macrobloc proposée ici, ne nécessite donc que le calcul supplémentaire des gradients moyens $\overline{\Delta D_{45}}$ et $\overline{\Delta D_{135}}$.

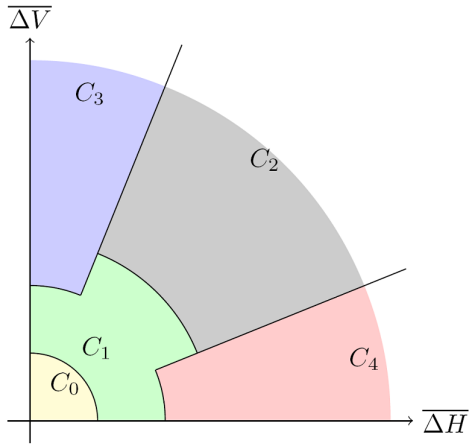


FIG. 2.13 – Plan pour la caractérisation de l'activité spatiale.

2.5.3 Caractérisation de la couleur d'un macrobloc

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 donc un troisième critère de segmentation des macroblocs : la couleur. La couleur d'un macrobloc est représentée de façon condensée par un histogramme à trois entrées. Les trois entrées correspondent aux trois composantes couleur de chaque pixel d'un macrobloc[9]. Chacune des trois composantes couleur est quantifiée sur N niveaux afin de réduire la taille de l'histogramme[10]. L'histogramme couleur est calculé en utilisant une fonction de profil convexe k[11], monotone et décroissante, qui attribue un poids plus faible aux coordonnées éloignées du centre du macrobloc [7]. Ce procédé de pondération permet d'augmenter la robustesse de la densité colorimétrique calculée pour chaque macrobloc.

Soit $\{x_i^*\}_{i=1..n}$ l'ensemble des coordonnées des n pixels du macrobloc courant, centré en zéro, et normalisé par les demi-tailles de la longueur et de la largeur d'un macrobloc. On note b la fonction de $\mathbb{R}^2 \rightarrow \{1...N\}^3$ qui associe à chaque pixel de coordonnées x_i^* l'indice de sa couleur dans l'histogramme couleur. La densité colorimétrique $q_{(u_1,u_2,u_3)}$ du macrobloc est alors donnée par l'équation 2.13 :

$$q_{u_1,u_2,u_3} = C.\sum_{i=1}^{n} k(\| x_i^* \|^2).\delta[b(x_i^*),(u_1,u_2,u_3)] \tag{2.13}$$

où δ est la fonction de Kronecker et C est une constante de normalisation. Chaque macrobloc peut à présent être caractérisé par sa densité colorimétrique, il convient donc d'établir une mesure de vraisemblance entre deux densités colorimétriques afin d'estimer la ressemblance entre deux macroblocs.

[9] Pour une image en niveaux de gris, l'histogramme ne possède qu'une entrée.
[10] Typiquement, une composante couleur représentée sur 256 niveaux, est quantifiée sur 8 ou 16 niveaux.
[11] Classiquement, on utilise le noyau d'Epanechnikov.

Pour cela, nous nous appuyons sur les travaux de Comaniciu [7] qui effectue un suivi d'objets déformables dans une vidéo à l'aide d'histogrammes de couleur. Comaniciu utilise une métrique dérivée du coefficient de Bhattacharyya (équation 2.14) afin de mesurer la similarité des couleurs de deux objets.

$$\rho(q_1, q_2) = \sum_{u_1=1}^{N} \sum_{u_2=1}^{N} \sum_{u_3=1}^{N} \sqrt{q_1(u_1, u_2, u_3).q_2(u_1, u_2, u_3)} \tag{2.14}$$

Nous pouvons déduire de ce coefficient la distance exprimée en équation 2.15 :

$$d(MB1, MB2) = \sqrt{1 - \rho(q_1, q_2)} \tag{2.15}$$

Nous considérons donc que deux blocs sont similaires en couleur lorsque la distance d est faible, c'est-à-dire lorsque le coefficient de Bhattacharyya est fort. Notons que les changements d'illumination au sein d'un même objet affaiblissent la similarité de deux macroblocs issu de cet objet. Les chances d'associer les deux macroblocs sont donc réduites. Pour pallier ce problème, différentes combinaisons d'espaces colorimétriques ont été testées [8]. 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 pourront donc être nécessaires dans le cas de vidéo à fort changement d'illumination.

2.6 Conclusion

Dans ce chapitre, nous avons décrit le bloc de traitement intra-segment temporel de l'outil de pré-analyse d'un flux vidéo. Ce bloc doit fournir une description détaillée des différents objets d'un segment de 180ms. La description détaillée d'un objet est constituée de trois informations : son mouvement, sa couleur et sa texture.

Grâce à une estimation long-terme des mouvements, basée sur l'utilisation de tubes spatio-temporels, nous pouvons, à la fois, dissocier efficacement les mouvements parfois proches d'objets différents et estimer le mouvement global de la caméra. L'information de mouvement ainsi obtenue couplée à une compensation du mouvement global sur le segment de neuf images, permet d'obtenir une segmentation au sens du mouvement performante. Cependant, une segmentation au sens du mouvement, seule, ne fournit pas une description assez complète des objets pour s'assurer de les avoir tous dissociés. Plusieurs objets distincts peuvent donc être regroupés au sein de la même zone de segmentation, il y a sous-segmentation. Inversement, les zones homogènes de la vidéo créent des informations de mouvement erronées, qui peuvent participer à la création de mouvements fictifs et générer la sur-segmentation d'un objet.

Pour pallier les défauts de la segmentation au sens du mouvement en affinant les résultats, nous avons choisi d'introduire deux critères supplémentaires pour caractériser un objet : la couleur et la texture. Ces critères devront être utilisés conjointement avec les résultats de la segmentation au sens du mouvement, afin d'obtenir une description spatio-temporelle complète des objets au sein d'un segment de 180ms. Afin de fusionner ces informations, nous pourrons par exemple utiliser une approche markovienne. La description finale, composée des informations de mouvement, de couleur et de texture, sera transmise au bloc de classification, où elle sera interprétée en vue d'un codage cohérent par le codeur H.264.

Chapitre 3

Présentation du bloc de traitement inter-segment temporel

3.1 Introduction

Le traitement inter-segment temporel doit fournir des résultats dont la nature est proche de ceux fournis par le traitement intra-segment. Néanmoins, 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, afin d'exploiter la redondance temporelle entre des segments temporels successifs et de réduire la complexité calculatoire, une méthode d'initialisation de la segmentation spatio-temporelle du segment courant peut être mise en place. Cette méthode s'appuiera sur les informations de mouvement obtenues sur les segments précédents. Ces mouvements pourront servir de germes lors des phases d'estimation de mouvement afin, par exemple, de réduire la taille des fenêtres de recherche. Ces mouvements seront alors affinés avec des méthodes de *block matching* classiques telles que le *full search* ou des méthodes d'optimisation rapides telles que l'algorithme du *simplex*.

Enfin, les résultats du traitement inter d'un segment temporel devront être périodiquement confrontés à ceux fournis par le traitement intra de ce même segment, cette comparaison permettra de vérifier la cohérence des informations données par le traitement inter et de justifier son utilisation.

3.2 Utilisation de la redondance temporelle pour le traitement inter-segment

Le bloc de traitement inter-segment et le bloc de traitement intra-segment fournissent tous deux des résultats de nature similaire, qui sont des cartes de segmentation spatio-temporelle. Les méthodes

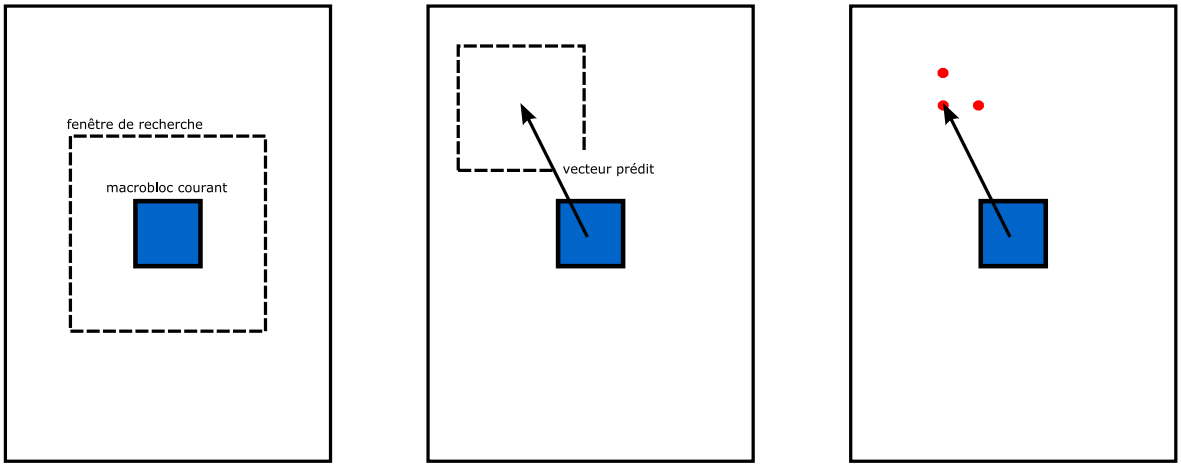


FIG. 3.1 – Méthodes pour accélérer la phase d'initialisation des mouvements long-terme.

utilisées pour construire ces cartes de segmentation sont identiques pour les deux types de traitement : estimation long-terme du mouvement, estimation du mouvement global puis segmentation spatio-temporelle basée sur le mouvement, la texture et la couleur.

Afin d'alléger la charge de calcul d'un traitement inter-segment lors de la construction d'une carte de segmentation, nous souhaitons exploiter la redondance temporelle entre des segments successifs. Nous avons observé que pour un traitement de type intra-segment, l'étape la plus coûteuse en termes de temps de calcul est celle de l'estimation long terme des mouvements (section 2.2). En effet, les deux *block matching* utilisés lors de cette étape nécessitent de nombreuses opérations et particulièrement la phase d'initialisation de la recherche long terme, qui emploie une méthode *full search*. En utilisant la corrélation entre les déplacements du segment courant et ceux du segment précédent, cette phase d'initialisation peut être fortement accélérée.

Pour cela, notre méthode consiste à privilégier une direction de recherche autour du bloc courant, afin de pouvoir réduire la taille de la fenêtre de recherche. Nous considérons que les déplacements calculés sur le segment temporel précédent sont des prédictions des déplacements du segment courant. La fenêtre de recherche n'est donc plus centrée sur le site correspondant à un mouvement nul, mais sur le site pointé par le vecteur prédit. Pour diminuer encore plus significativement les temps de calculs, nous pouvons remplacer la méthode classique de *full search* par une méthode d'optimisation rapide telle que celle du simplex de Nealder et Mead. La position du simplex initial est alors également pointée par le vecteur prédit. La figure 3.1 illustre les deux méthodes envisagées pour accélérer la phase d'initialisation de l'estimation des mouvements long-terme.

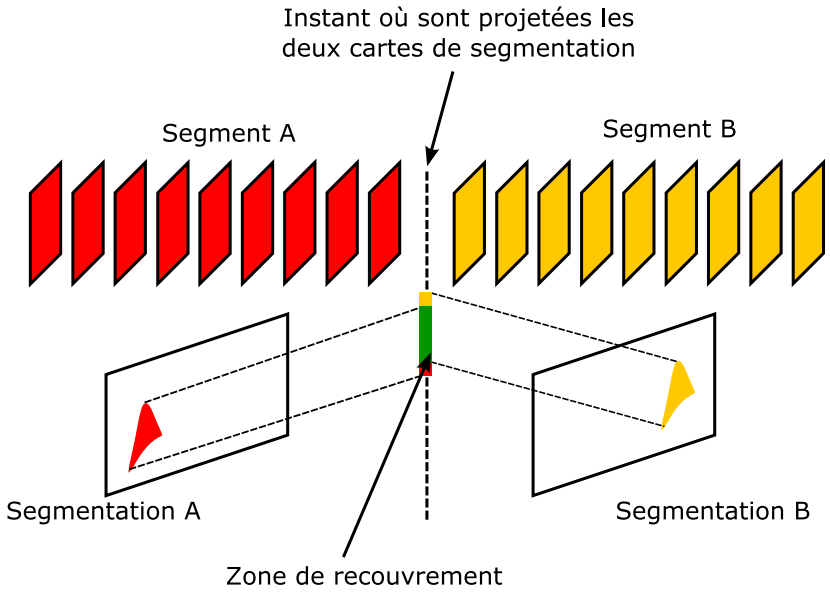


FIG. 3.2 – Recouvrement d'objets.

3.3 Suivi d'objets sur plusieurs segments temporels

3.3.1 Suivi par recouvrement de projections

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 majeures : 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 superposer les cartes de segmentation de chacun des deux segments successifs. Ainsi, si un objet de la carte de segmentation du premier segment temporel recouvre un objet de la carte de segmentation du segment suivant, nous attribueront le même label à ces deux objets afin de les fusionner en un seul objet dont le cycle de vie est supérieur à 180ms.

Pour chaque tranche temporelle de neuf images, la carte de segmentation est calculée à partir d'une estimation de mouvement réalisée sur l'image médiane du segment, les cartes de segmentation de deux segments successifs ne correspondent donc pas aux mêmes instants temporels et ne peuvent pas être comparées directement. Afin de pouvoir superposer deux cartes de segmentation successives, nous les projetons au même instant temporel à l'aide des informations de mouvement disponibles pour chaque objet. L'instant temporel choisi est un instant "fictif" puisqu'il est situé entre la dernière image du premier segment et la première image du second segment. La figure 3.2 illustre le procédé de comparaison des cartes de segmentation.

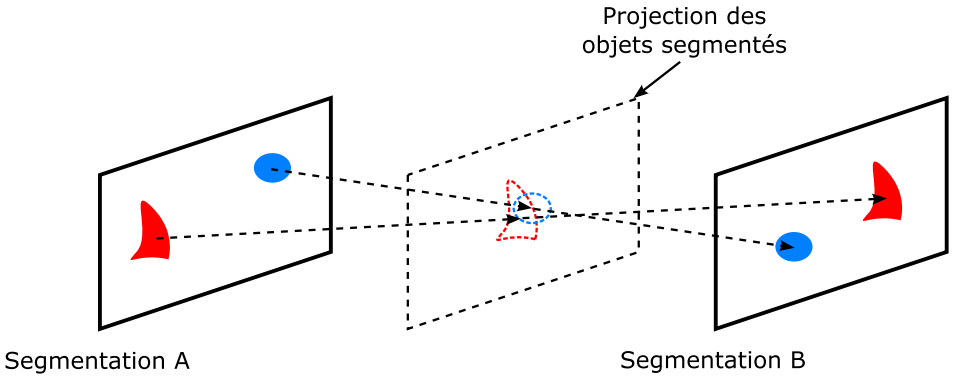


FIG. 3.3 – Illustration des problèmes de recouvrements

3.3.2 Connexions multiples lors du suivi d'objets

La méthode de suivi d'objets repose sur les recouvrements d'objets spatio-temporels translatés à partir d'une carte de segmentation (figure 3.2). Dans certains cas, plusieurs objets issus d'une même carte de segmentation peuvent être projetés sur un même site spatial, cette configuration génère un conflit lors du recouvrement avec les objets projetés depuis la carte de segmentation du segment suivant : les projections de certains objets présentent des connexions multiples. La figure 3.3 présente un exemple de ce genre. Pour assurer à chaque objet à connexions multiples, un suivi temporel cohérent, nous devons choisir un critère de discrimination entre les différents candidats dont les projections recouvrent celle de l'objet courant. Pour éviter de générer une charge de calcul trop importante, nous réutilisons l'information de couleur associée à chaque objet. Un objet dont la projection est à connexions multiples sera donc associé à l'objet dont les caractéristiques de texture et de couleur sont les plus proches.

3.4 Conclusion

Dans ce chapitre, nous avons décrit le bloc de traitement inter-segment temporel de l'outil de pré-analyse d'un flux vidéo. Comme dans le cas d'un traitement intra-segment, une carte de segmentation spatio-temporelle est construite et donne des information de mouvement, de couleur et de texture des différents objets qui composent la scène d'une séquence vidéo. Pour diminuer la complexité en terme de calcul, nous ré-utilisons les informations de mouvement calculées pour le segment temporel précédent, afin d'initialiser la recherche des déplacements sur le segment courant. La phase d'estimation long-terme des mouvements, qui est la phase la plus coûteuse lors de la création d'une carte de segmentation spatio-temporelle, est donc fortement accélérée. Enfin, la carte de segmentation calculée pour le segment courant est comparée à celle du segment précédent afin de pouvoir assurer le suivi des objets spatio-temporels sur des tranches supérieures à 180ms. Dans le chapitre suivant, nous verrons que les informations de mouvement, de couleur, de texture et de cycle de vie d'un objet vont être transmises au bloc de classification, ce dernier va exploiter ces caractéristiques pour optimiser les choix du codeur H.264.

Chapitre 4

Présentation du bloc de classification pour un codage cohérent avec H.264

4.1 Introduction

Afin de réduire le flux de données tout en conservant la qualité des séquences vidéo, le codeur H.264/AVC utilise de nombreuses techniques de prédiction. Ainsi, le codeur H.264 permettrait de réduire le flux de données jusqu'à 50% par rapport au codeur MPEG-2 pour une qualité des images équivalente.

Cependant, ces techniques génèrent des temps de calcul très importants. Par exemple, le codeur de référence[1] adopte une méthode de recherche *Full Search* pour la prédiction. Il existe 7 modes inter différents (16×16, 16×8, 8×16, 8×8, 8×4, 4×8 et 4×4) et deux modes intra (16×16 et 4×4). Pour les modes inter, la recherche des vecteurs de mouvement s'effectue à l'aide de 5 images références. Ces différents modes de prédiction sont illustrés en figure 4.1.

Le codeur de référence H.264 teste tous ces modes de prédiction pour chaque nouveau macrobloc à coder, il retient le mode qui optimise la relation débit-distorsion. Le codeur utilise donc une charge de calcul excessive pour compenser le fait qu'il ne dispose d'aucune visibilité intelligible de la scène. Cependant, 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[2], alors que le codage qui en résulte n'assure aucune cohérence avec le contenu spatio-temporel de la scène. Dans l'optique de réduire ces temps de calcul et de donner au codeur une approche cohérente du contenu de la séquence vidéo, nous proposons trois stratégies pour optimiser le codage à l'aide de la décomposition spatio-temporelle donnée par notre outil de pré-analyse du flux vidéo :

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

Ces trois stratégies sont décrites dans les sections suivantes.

[1] *JM reference software* disponible sur http ://iphome.hhi.de/suehring/tml/.
[2] Cela peut augmenter significativement si on utilise une fenêtre de recherche plus importante.

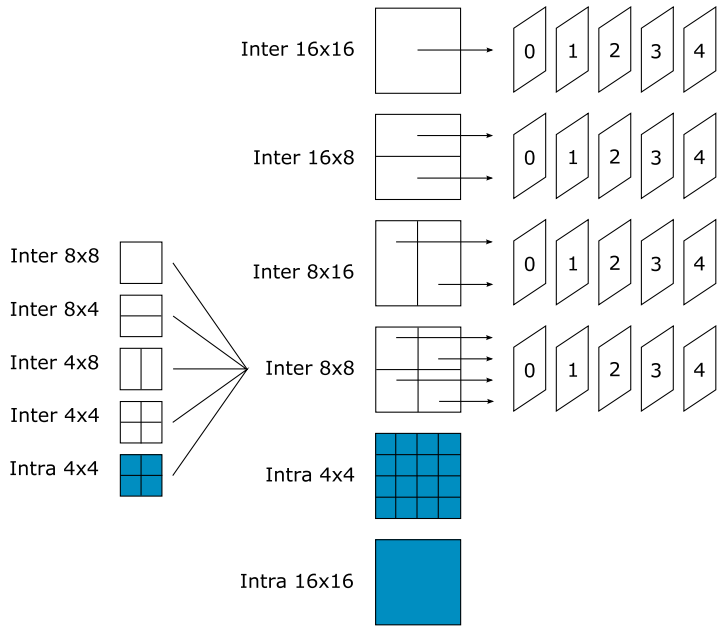


FIG. 4.1 – Modes de recherche pour les prédictions intra et inter avec 5 images de référence pour le codeur H.264/AVC.

4.2 Choix du paramètre de quantification

Dans la chaîne de codage (voir figure 1), après la phase de prédiction et de compensation du bloc courant, le bloc résiduel D_n est transformé par une transformation 4×4 ou 8×8. Cette transformation est une transformation entière[3] basée sur une forme modifiée de la Transformée en Cosinus Discrète (TCD). Cette transformation fournit un ensemble de coefficients, dont chacun est une valeur de pondération pour un motif de base. Une fois combinés, les motifs de base pondérés recréent le bloc résiduel original. La figure 4.2 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 4.3 présente les cas d'une image uniforme et d'une image avec une forte activité spatiale. On observe qu'avec un même paramètre de

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

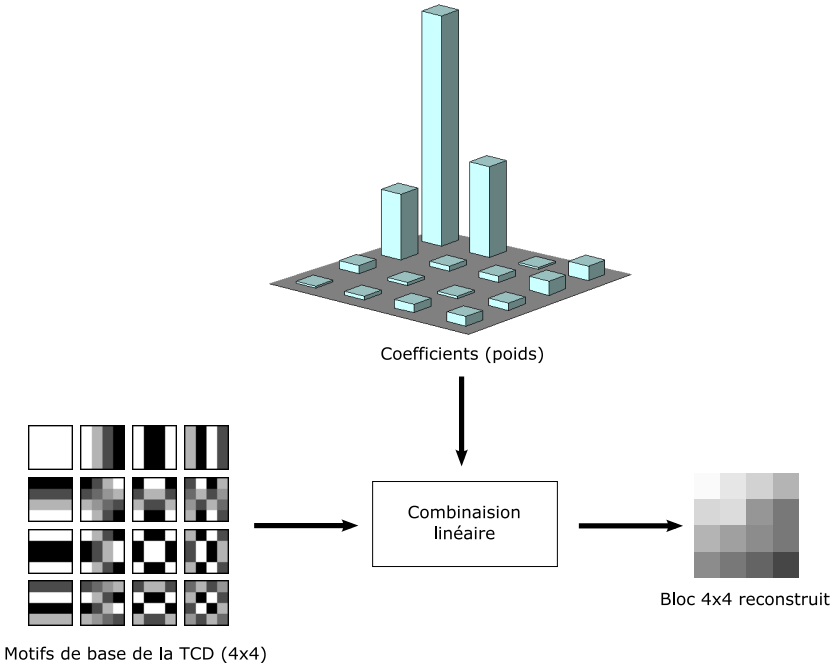


FIG. 4.2 – Transformée inverse : combinaison linéaire des blocs de base pour reconstruire le bloc original.

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 (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.

Nous pouvons exploiter cette propriété à l'aide de notre outil de pré-analyse, qui caractérise l'activité spatiale de chaque objet spatio-temporel. Ainsi, pour les objets homogènes, nous pourrons indiquer au codeur d'utiliser 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 pourront 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 pourront être conservés sur plusieurs images consécutives.

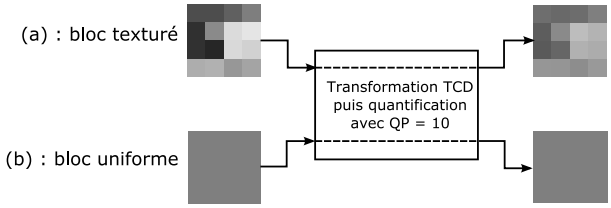


FIG. 4.3 – Comparaison des quantifications pour un bloc texturé et un bloc uniforme.

4.3 Choix du mode de codage

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. Ainsi, afin d'accroître la précision de la prédiction pour réduire le débit nécessaire, le codeur H.264/AVC teste de nombreuses combinaisons de tailles de bloc (de 16×16 à 4×4) et de mode de codage (intra ou inter). Ces tests sont réalisés de manière exhaustive pour chaque macrobloc, afin de créer une relation débit-distorsion en fonction de laquelle le codeur va choisir le mode le plus adapté à la compression. Le choix du mode est donc réalisé indépendamment 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. Nous allons donc utiliser notre outil de pré-traitement pour guider les choix du codeur en lui apportant une approche plus intelligente du contenu spatio-temporel la séquence vidéo. Chaque objet de la séquence est à présent caractérisé en terme de cycle de vie, de mouvement, de couleur et de texture, ces informations peuvent être exploitées pour influencer le codeur quant à ses choix sur la taille des partitions et le mode de codage d'un macrobloc. D'une part, le cycle de vie d'un objet renseigne sur l'apparition et la disparition de ce dernier dans la séquence. Nous pouvons donc indiquer au codeur d'utiliser un mode intra pour encoder les macroblocs de l'objet dans les images ou ce dernier apparaît ou disparaît, et d'utiliser un mode inter dans les autres cas. D'autre part, l'information de texture disponible pour chaque objet, peut orienter le codeur sur le choix de la taille des partitions : des petites partitions permettront de coder plus efficacement des zones fortement détaillées, alors que des partitions plus grossières suffiront pour coder convenablement des zones homogènes. Enfin, grâce au suivi des objets réalisé à l'étape de traitement inter-segment, nous allons indiquer au codeur H.264 d'utiliser les mêmes modes de codage (intra ou inter, taille de partition) pour les différentes représentations temporelles d'un même objet. Cette méthode va, d'une part, 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, 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, nous allons éviter au codeur de réaliser des tests exhaustifs pour chaque nouvelle image à coder.

4.4 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 [9] 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 [10]. L'utilisation de ces méthodes pourraient s'avérer appropriée dans notre cas, puisque l'information de couleur des objets spatio-temporels a été déterminée lors du raffinement de la segmentation avec des critères de couleur et de texture (section 2.5).

Les images clés qui sont les images les plus représentatives d'une séquence vidéo sont généralement utilisées pour la récupération, l'indexage et les résumés de vidéos. Ozbek et Tekalp proposent d'utiliser les méthodes de sélection des images clés afin de réaliser un choix rapide des images références pour un groupe d'images. L'image dont l'histogramme est le plus proche de l'histogramme moyen est choisie comme image clé principale pour le groupe d'images sélectionnées. Ils définissent trois cas de choix différents :

- l'image clé principale et les deux autres images les plus proches de l'histogramme moyen,
- l'image clé principale et les deux autres images les plus éloignées de l'histogramme moyen,
- l'image clé principale et l'image la plus éloignée de l'histogramme moyen.

Au niveau du codage, la méthode intervient 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 sont obtenus avec le troisième cas (l'image clé principale et l'image la plus éloignée de l'histogramme moyen) avec 23% de temps de calcul en moins, mais 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.

4.5 Conclusion

Ce dernier chapitre a présenté le cahier des charges du bloc de classification de notre outil de pré-analyse d'un flux vidéo et quelques pistes pour sa réalisation. Ce bloc permettra de transmettre un jeu de paramètres au codeur H.264/AVC 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, que nous avons calculées lors des phases de traitement intra et/ou inter-segment temporel. La première méthode propose de fixer le paramètre de quantification QP du codeur en fonction de l'activité spatiale du macrobloc courant et de considérations psychovisuelles. La seconde méthode concerne le choix du mode de prédiction utilisé (intra et/ou inter, taille de partition). Pour cela, la

décomposition spatio-temporelle de la scène sera utilisée pour donner au codeur H.264 une vision structurée du contenu de la scène. La dernière méthode repose sur la sélection des images de référence (par défaut, le codeur utilise les cinq images codées précédemment à l'image courante). Peu de méthodes réalisant un choix des images références ont été proposées dans la littérature. Cette notion d'image de référence est pourtant importante puisqu'elle est à la base de la prédiction inter moyen et long-terme, la bonne gestion de ces images est donc primordiale. L'utilisation des données fournies par notre bloc de 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 élevée.

Conclusion

Ce rapport présente les définitions d'algorithmes de pré-analyse adaptés en vue du codage sous le standard H.264/AVC de flux vidéo haute définition. Le premier chapitre a présenté les spécifications fonctionnelles de l'outil à concevoir. Ce dernier a d'abord été spécifié par rapport à son environnement, c'est-à-dire par rapport aux entités avec lesquelles il communique. Ici, deux entités externes sont identifiées : la première (un utilisateur) transmet un flux haute définition à l'entrée de l'outil de pré-analyse, la seconde (le codeur H.264) reçoit un jeu de paramètres de codage fourni en sortie de l'outil de pré-analyse. Á partir du comportement de notre système de pré-traitement vis-à-vis de son environnement, nous avons pu définir les principales fonctionnalités qui le composent :

- une fonction de traitement intra-segment temporel ;
- une fonction de traitement inter-segment temporel ;
- une fonction de classification qui fournira le jeu de paramètres adapté au codage avec le standard H.264/AVC.

Le chapitre suivant présente les méthodes envisagées pour mettre en oeuvre la fonction de traitement intra-segment. Cette fonction s'appuie sur une estimation de mouvement long-terme, sur un segment temporel de neuf images, soit sensiblement 180ms. L'estimation long-terme du mouvement présente un double avantage par rapport à une estimation court-terme classique, d'une part, lors d'une segmentation basée sur les vecteurs déplacement, les mouvements estimés permettent de mieux distinguer des objets proches dans leurs déplacements respectifs et d'autre part, les mouvements long-termes calculés permettent d'estimer et de compenser le mouvement global de la scène sans avoir à combiner des déplacements locaux, comme c'est le cas pour des estimations à court-terme. L'estimation des déplacements long-termes et celle du mouvement global permettent de segmenter un segment temporel au sens du mouvement. Afin d'affiner la segmentation effectuée précédemment et d'obtenir des caractéristiques supplémentaires sur les objets détectés, des critères basés couleur et textures sont calculés pour chaque objet. Ainsi, le bloc de traitement intra-segment fournit la décomposition d'un segment temporel de neuf images en objets spatio-temporels caractérisés par leur mouvement, leur couleur et leur texture.

Le troisième chapitre présente la méthode de traitement inter d'un segment temporel. La nature des résultats fournis par ce bloc de traitement sera similaire à celle des résultats obtenus avec le bloc de traitement intra, cependant une caractéristique supplémentaire sera ajoutée aux objets spatio-temporels : leur cycle de vie. Les objets pourront donc être suivis sur plusieurs segments temporels successifs. D'autre part, le traitement inter permettra d'exploiter la redondance temporelle entre des segments successifs, la charge de calcul nécessaire pour décomposer le segment courant pourra donc être allégée par rapport au cas d'utilisation d'un traitement intra.

Le dernier chapitre présente le bloc de classification des objets spatio-temporels déterminés lors des phases de traitement intra et/ou inter-segment. Ce bloc va générer un ensemble de paramètres, qui permettra de coder les objets de façon cohérente en fonction de leur contenu spatial, de leur

mouvement, de leur cycle de vie et de leur environnement. Le jeu de paramètres permettra de modifier principalement trois des stratégies classiques utilisées par le codeur H.264 de référence :

- choix du paramètre de quantification,
- limitation pour le choix des modes de prédiction,
- choix des images références.

Dans le cadre du sous-projet 4 : "Pré-analyse et conditionnement du flux vidéo en haute définition" du projet ArchiPEG, les prochains travaux à réaliser, qui correspondent à la tâche 4.3 , seront le développement des algorithmes de pré-traitement du flux vidéo étudiés dans le présent rapport. Puis, les derniers travaux à réaliser, qui correspondent à la tâche 4.4, seront les phases de test des algorithmes sur prototypes et sur la plate-forme d'accueil du projet.

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 [11].

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 Parkrun

La séquence comporte 500 images filmées en plan éloigné. La scène représente un homme, avec un parapluie dans sa main, qui court dans un parc puis s'arrête et reste immobile vers la 340ème image.

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

L'arrière plan est composé d'arbres, de neige et d'une source d'eau. Le contenu est très détaillé. La figure A.2 présente une image extraite de la séquence *Parkrun*.

A.1.3 Knightshields

La séquence comporte 500 images filmées en plan rapproché. Un homme avec une barbe et une veste très texturée marche devant un mur composé de boucliers de chevaliers détaillés. Á la fin de la séquence, le capteur effectue un zoom avant de la scène. La figure A.3 présente une image extraite de la séquence *Knightshields*.

A.2 Les séquences 1080p

Les séquences 1080p utilisées ici sont des vidéos 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 Blue Sky

La séquence comporte 250 images. La scène représente les cimes de deux arbres très détaillés, en fort contraste avec le ciel bleu uniforme. La caméra effectue une rotation. La figure A.4 présente une image extraite de la séquence *Blue sky*.

A.2.2 Station

La séquence comporte 313 images filmées en soirée, depuis un pont de la gare routière de Munich. La caméra effectue un long zoom arrière. La scène comporte des structures régulières avec beaucoup de détails (rails). La figure A.5 présente une image extraite de la séquence *Station*.

A.2.3 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.6 présente une image extraite de la séquence *Tractor*.

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

FIG. A.2 – Image 160 de la séquence *Parkrun*.

FIG. A.3 – Image 1 de la séquence *Knightshields*.

FIG. A.4 – Image 1 de la séquence *Blue sky*.

FIG. A.5 – Image 100 de la séquence *Station*.

FIG. A.6 – Image 60 de la séquence *Tractor*.

Bibliographie

[1] White Paper on Digital Video solutions, AVC + AAC The Next Generation of Compression, *Harmonic*, 2003.

[2] Iain E.G. Richardson, H.264 and MPEG-4 Video Compression, John & Sons, September 2003.

[3] O. Le Meur, "Attention sélective en visualisation d'images fixes et animées affichées sur écran : modèles et évaluation des performances - applications," Université de Nantes, PhD. Thesis, École polytechnique de l'université de Nantes, 2005.

[4] S. Péchard, P. Le Callet, M. Carnec, and D. 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, VPQM 2007*, Scottsdate, 2007.

[5] J.A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308-313, 1965.

[6] R. Coudray, and B. Besserer, "Global motion estimation for MPEG-encoded streams," in Proc. IEEE International Conference on Image Processing, ICIP 2004, Singapore, Republic of Singapore, October 2004.

[7] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-Based Object Tracking," in Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 25, Issue 5, May 2003, pp. 564 - 577

[8] A. Lehuger, P. Lechat, N. Laurent and P. Perez, "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.

[9] Yuan, Y., Feng, D. and Zhong, Y.-Z. Three Fast Methods for Adaptive Key-frame Setting and Dynamic Frame-rate Adjusting in Video Coding. volume 1 de ISSN : 1304-4508. International Journal of Computational Intelligence, 2004.

[10] Ozbek, N. and Tekalp, A. M. Fast H.264/AVC Video Encoding with Multiple Frame References. Pages 597-600, Gênes, Italie. IEEE International Conference on Image Processing, ICIP'2005.

[11] The SVT High Definition Multi Format Test Set, SVT corporate technology, février 2006, `ftp://vqeg.its.bldrdoc.gov/HDTV/SVT_MultiFormat/SVT_MultiFormat_v10.pdf`.