

ACADÉMIE DE MONTPELLIER
UNIVERSITÉ MONTPELLIER II
— SCIENCES ET TECHNIQUES DU LANGUEDOC —

Mémoire de Stage de Master

SPÉCIALITÉ : **Recherche en Informatique**
Mention : **Informatique, Mathématiques, Statistiques**

effectué au laboratoire LIRMM/ROB

—
sous la direction de MARC CHAUMONT, WILLIAM PUECH

**Protection et enrichissement de régions
d'intérêts dans des séquences vidéos**

par

Peter MEUEL

Soutenu le 19 juin 2006

Table des matières

1	Abstract	3
2	Introduction	4
2.1	Présentation	4
2.2	Approche considérée	5
3	Compression vidéo/H.264-AVC	7
3.1	Principes généraux	8
3.2	Prédictions	10
3.2.1	Prédiction spatiale	10
3.2.2	Prédiction temporelle	13
3.3	Transformée et Quantification	13
3.3.1	Transformation	13
3.3.2	Quantification	15
3.4	Encodage	15
4	Insertion dans des séquences vidéos	17
5	Travaux	20
5.1	Enrichissement	20
5.1.1	Détection de la région d'intérêt	20
5.1.2	Calcul des pertes	23
5.1.3	Insertion de données	24
5.1.4	Résultats	26
5.2	Protection	27
5.2.1	Robustesse	28
5.2.2	Chiffrement des coefficients DCT	28
5.2.3	Chiffrement du flux binaire	29
6	Conclusion	30

1 Abstract

Ce mémoire de stage présente des travaux effectués dans la protection et l'enrichissement de régions d'intérêts dans des séquences vidéos. Nous présentons les bases du format vidéo utilisé, l'H.264, ainsi qu'un état de l'art de l'insertion dans les séquences vidéos. Nous présentons ensuite les méthodes développées durant ce stage visant à l'enrichissement de séquences vidéos, à savoir une méthode de détection de visage, qui constitue une Région d'Intérêt (RI), ainsi qu'une méthode d'insertion dans des séquences H.264 invisibles aux lecteurs H.264 usuels. Nous présentons également la méthode d'extraction de ces données et la technique de reconstruction permettant d'apporter une qualité supérieure à la RI. Enfin, nous présentons les difficultés qui se présentent lors du chiffrement partiel d'une séquence H.264 et présentons une première technique de protection combinant chiffrement et insertion.

In this report, we introduce the results found during a Master in Computer Science training course on protection and enhancement of Region of Interest (ROI). We will first present the H.264 video format and a state-of-the-art in video watermarking. We will then present our results about faces detection and video enhancement, invisible to standard H.264 decoding, that lead to a reconstruction of high quality in the ROI. At last, we will discuss of the difficulty of ROI encryption and present an early method combining encryption and watermarking for the protection of the ROI.

Remerciements Un grand merci à M. William PUECH et M. Marc CHAUMONT, ainsi qu'à M. José-Marconi RODRIGUES et M. Philippe AMAT pour leur aide tout au long de ce stage.

2 Introduction

2.1 Présentation

De nos jours, la surveillance par caméra vidéo est devenue un moyen privilégié de surveillance. Le faible coût de revient par rapport à une surveillance humaine, les capacités technologiques (enregistrement, caméras motrices) et les problèmes de sécurité de notre époque ont favorisé l'explosion de l'utilisation de moyens de surveillance vidéo. Les lieux sensibles (banques, bâtiments administratifs, ambassades) mais également les lieux publics sujets à risques (stades de football, parcs, métros) possèdent une ou plusieurs caméras destinées à la surveillance vidéo. À Londres par exemple, on ne trouve pas moins de 9400 caméras vidéos uniquement dans les transports en commun[6] et une étude de l'équipe UrbanEye[13] estimait en 2002 à 400.000 le nombre de caméras dans Londres.

Utilité de la compression vidéo Les caméras utilisées pour la surveillance vidéo sont souvent de qualité suffisante pour une visualisation correcte de la scène, mais il n'en va pas de même des moyens de transmissions des vidéos acquises vers un centre de surveillance. Une solution encore très répandue reste l'envoi d'image fixe à intervalles réguliers, une image toutes les 5 secondes par exemple. Ce problème est dû au débit généré par une séquence vidéo. A raison de 15 images par secondes et de 8 bits par composante, une séquence vidéo en 256 niveaux de gris de 320x240 pixels produit un débit de $320 * 240 * 8 * 15 = 9216000bits$, soit 9000 Kilo-octets, c'est-à-dire plus de 8,7 Méga-octets par seconde. Ce n'est évidemment pas un débit possible entre un bus et un centre de contrôle. On comprend mieux tout l'intérêt et toute l'importance de la compression vidéo, encore faut-il que les méthodes de compression s'effectuent en temps réel. En effet, une grande partie de l'intérêt d'une surveillance vidéo disparaît si la séquence transmise a plusieurs secondes de décalage avec la réalité. La compression en temps réel est donc un point essentiel en surveillance vidéo. Ainsi, nous nous intéresserons aux possibilités d'application en temps réel des méthodes développées dans le cadre de ce stage.

Ethique et vie privée Un autre problème, d'ordre éthique, se pose avec la surveillance vidéo : le respect de la vie privée. Une ville maillée de caméras vidéos permettrait de suivre à la trace un individu particulier. Il est donc nécessaire de garantir une protection immédiate aux personnes filmées par une caméra dans un lieu public. Protection qui peut être levée si des événements particuliers (vols, agression, attentat) nécessitent d'identifier les protagonistes d'une scène filmée. On pense ainsi à un système masquant les visages dans le cadre d'une visualisation par un agent de contrôle, masque pouvant être levé par un agent assermenté possédant une clef de déverrouillage. Pour des raisons pratiques (encombrement, coût), il n'est pas envisageable de produire deux versions différentes du même enregistrement. Nous cherchons donc à créer dans un unique flux deux niveaux d'informations. Le premier sera restreint : les visages seront masqués. Le second niveau, accessible uniquement en possession de la clef adéquate, permettra de visualiser les visages sans le masque.

Enrichissement Nous imaginons également le cas inverse : une surveillance nécessitant l'identification de toute personne passant dans le champ d'une caméra de surveillance. Dans le cas où que le débit est fixe (ou en tout cas limité), et que ce débit ne permet pas d'avoir une qualité suffisante sur l'ensemble de la scène, on peut souhaiter obtenir une image nette sur la portion du visage même si cela implique des pertes de qualité au niveau du reste de l'image.

Régions d'intérêts Nous définissons ainsi une région d'intérêt. Une région d'intérêt est une portion de l'image présentant un intérêt supérieur au reste de l'image. Que cette région d'intérêt soit fixe ou mobile, il s'avère que le reste de l'image est souvent superflu, inintéressant ou que la connaissance de la scène en dehors de la région d'intérêt ne nécessite pas une haute qualité. On peut accepter une dégradation de la qualité du reste de l'image au profit de la protection, ou d'une meilleure qualité, de la région d'intérêt. Un tel traitement est une optimisation du débit d'information. En effet, nous gardons le même débit mais en offrant des informations complémentaires (netteté de la région d'intérêt) ou une protection d'information (masquage de la région d'intérêt), au détriment du reste de l'image.

2.2 Approche considérée

Nous allons donc nous intéresser à la protection et l'enrichissement des séquences vidéos. Ces deux thématiques sont issues des problèmes de surveillance vidéos et considèrent deux approches différentes : permettre une identification fiable des personnes filmées et permettre le respect de la vie privée des personnes filmées. L'aspect temps-réel des méthodes développés ne devra pas être négligé, comme expliqué précédemment.

Enrichissement L'enrichissement d'une séquence vidéo se fait en deux temps. A l'encodage (voir figure 1), il s'agit de :

- déterminer la région d'intérêt
- déterminer les pertes engendrées par la compression dans la zone d'intérêt
- insérer de façon transparente les pertes dans le flux vidéos

A la décompression, l'opération s'effectue dans le sens inverse (voir figure 2).

Il s'agit alors :

- d'extraire les pertes de la région d'intérêt
- reconstruire les pertes de la région d'intérêt
- rajouter ces pertes à la région d'intérêt afin de retrouver la région d'intérêt telle qu'elle l'était avant compression.

L'insertion dite transparente des données d'enrichissement implique que les données insérées soient invisibles à un décodeur normal. Seul un décodeur spécifique saura comment extraire et reconstituer les données. Ceci nous permet de conserver une compatibilité avec les décodeurs H.264 existants.

Protection La protection d'une région d'intérêt dans une séquence vidéo suppose un caractère confidentiel de cette région d'intérêt. Nous souhaitons restreindre la visualisation des régions d'intérêt, dans notre cas des visages, aux personnes autorisées. Nous utilisons donc, comme en chiffrement, une clef pour chiffrer les données de la région d'intérêt. Les données de la région d'intérêt sont

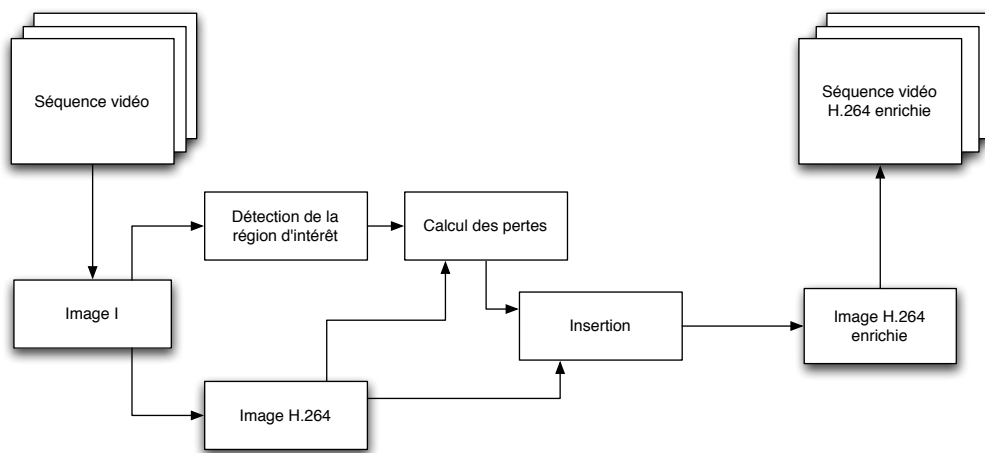


FIG. 1 – Enrichissement d’une séquence vidéo

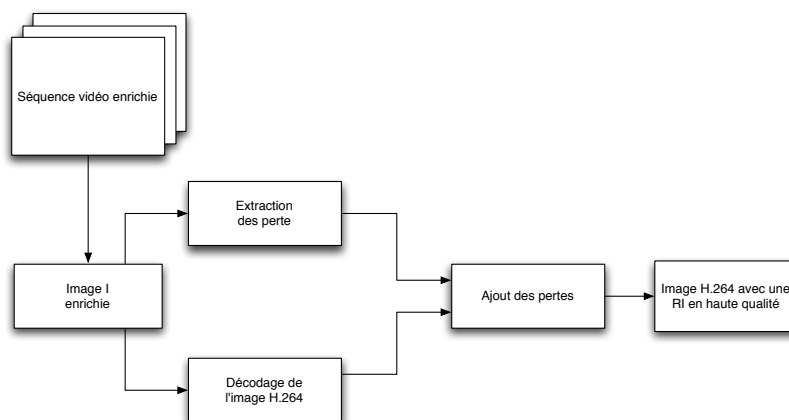


FIG. 2 – Décompression d’une séquence vidéo enrichie

chiffrés par des méthodes courantes puis nous substituons les données chiffrées aux données originales (voir figure 3).

A la visualisation de la vidéo, un décodeur normal décodera les données chiffrées de la région d’intérêt et affichera ainsi une zone brouillée. Un lecteur spécifique, moyennant la possession de la clef, déchiffrera d’abord les données de la région d’intérêt, ce qui permettra le décodage et l’affichage de la région d’intérêt en clair(voir figure 4).

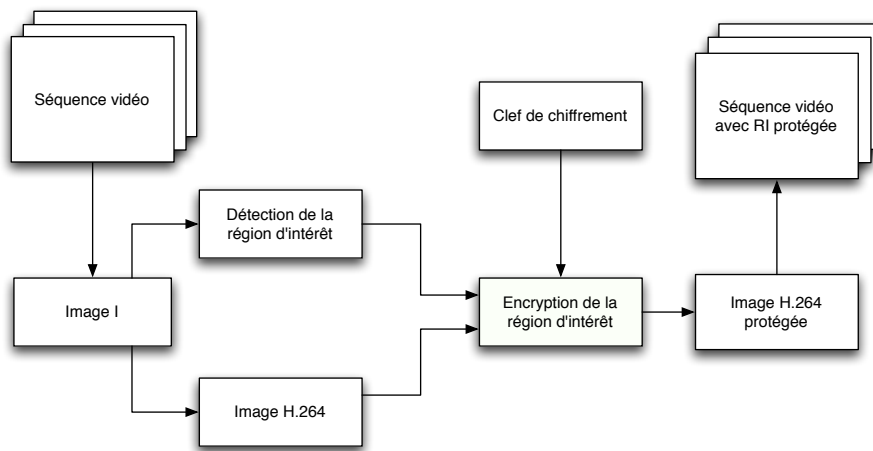


FIG. 3 – Chiffrement d'une séquence vidéo

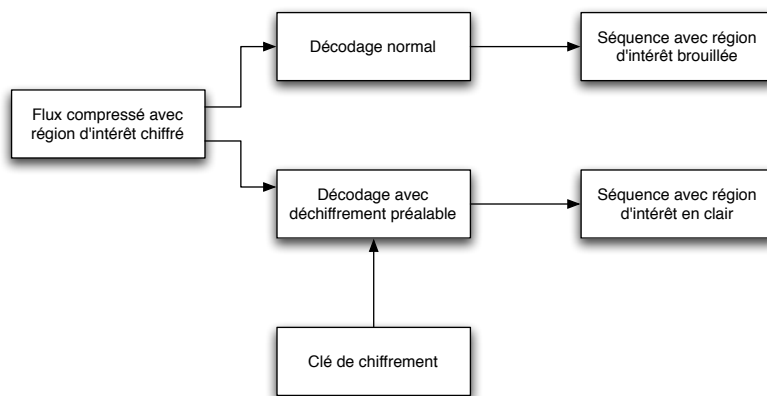


FIG. 4 – Déchiffrement d'une séquence vidéo

3 Compression vidéo/H.264-AVC

L'H264-AVC est le format de compression le plus performant à l'heure actuelle. Il constitue donc le codeur de référence concernant les travaux de compression vidéo. Il a été développé conjointement par le Moving Picture Experts Group (MPEG) et le Video Coding Experts Group (VCEG). Sa dénomination officielle est l'AVC et il a été publié simultanément sous les spécifications MPEG-4 Part 10 et sous la recommandation H.264 de l'ITU-T[16] en 2003.

L'H264-AVC définit plusieurs techniques de compression, toutes ne sont pas forcément applicables en même temps pour les différents types de vidéos (vidéos de haute qualité, streaming etc...). Il existe donc 3 profils différents pour l'H.264, selon le type d'application visé :

- Le profil Baseline, qui comprend les prédictions I et P et le codage CAVLC (notions détaillées dans les paragraphes suivants).
- Le profil Main, qui est un profil Baseline mais qui permet également l'utilisation des prédictions B et du codage CABAC.
- Le profil Extended, qui est également un profil Baseline, mais qui autorise les prédictions SP et SI[9], particulièrement adaptées au streaming.

3.1 Principes généraux

Une séquence vidéo est composée d'une succession d'images. Chaque image est composé de macroblocks. Chaque macroblock est un carré (qu'on peut voir comme un tableau ou une matrice carrée) de $16 * 16$ pixels.

Echantillonnage Les séquences vidéos sont fournies à l'encodeur dans une représentation YUV. La composante Y représente l'information de luminance, les composantes U et V représentent les informations de chrominance. Parce que le système visuel humain (SVH) est plus sensible aux informations de luminance que de chrominance, les composantes U et V sont très souvent sous-échantillonnées par rapport à la composante Y. On utilise principalement 4 modes d'échantillonnage en compression vidéo :

- 4 :0 :0 : seule l'information de luminance est transmise. C'est donc de la vidéo en niveaux de gris
- 4 :2 :0 : Pour un échantillon de $16*16$ pixels, un macroblock Y est un tableau de $16*16$ valeurs, et les macroblocks U et V correspondants sont chacun des tableaux de $8*8$ valeurs.
- 4 :2 :2 : Les tableaux des macroblocks U et V ont la même hauteur que celui des valeurs Y mais une largeur divisée par 2
- 4 :4 :4 : Les 3 composantes possèdent des tableaux de valeurs de même taille. Cet échantillonnage est utilisé pour obtenir des vidéos de très bonne qualité.

Redondance Afin de compresser la vidéo, c'est-à-dire réduire la quantité d'information sauvegardée pour une séquence vidéo, on se base sur la redondance spatiale et la redondance temporelle que l'on trouve fréquemment dans une séquence vidéo.

Redondance spatiale La redondance spatiale exprime le fait qu'on trouve souvent dans chaque image de la séquence une ou plusieurs portions présentant les mêmes caractéristiques visuelles (couleurs, textures). On va donc supposer qu'on peut déduire les valeurs des pixels d'un macroblock à partir des valeurs des pixels des macroblocks voisins. C'est ce qu'on appelle une prédiction spatiale ou prédiction I.

Redondance temporelle La redondance temporelle exprime le fait qu'entre deux, voire plusieurs, scènes consécutives d'une séquence, il n'y a pas énormément de changement dans l'image. On comprend tout de suite cette idée en pensant à une scène d'un film dans laquelle un acteur parle sans bouger dans l'image. Tout au long de la séquence, seule la portion délimitant la bouche de l'acteur variera, le reste de l'image demeurant inchangé. On suppose qu'on peut déduire

les valeurs des pixels d'un macroblock à partir de macroblocks appartenant à des images antérieures ou ultérieures (dans l'ordre d'affichage) de la séquence. C'est ce qu'on appelle une prédiction temporelle ou prédiction P ou B. Nous détaillons les différences entre prédiction P et B au paragraphe 3.2.

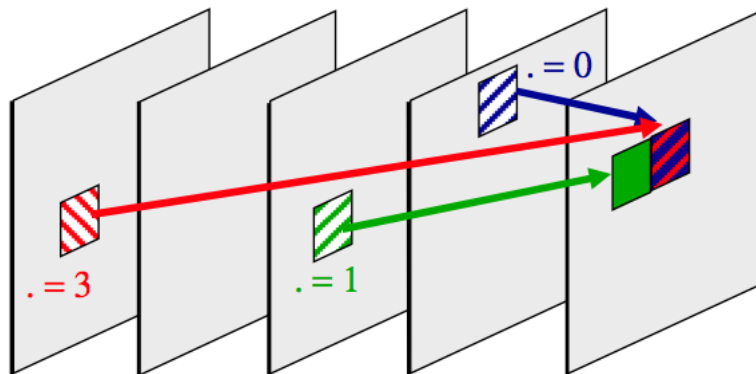


FIG. 5 – Prédictions temporelles

Remarquons qu'en compression vidéo, on distingue l'ordre d'encodage de l'ordre d'affichage d'une image dans une séquence vidéo. Cette distinction prend tout son sens pour les prédictions temporelles dans le cas où une image prédite temporellement pourra être plus compacte si elle se réfère à une image ultérieure qu'antérieure dans l'ordre d'affichage (nous détaillons en quoi une image d'une séquence codée est compacte dans les paragraphes suivants).

Les images d'une séquence s'ordonnent donc dans le flux par ordre de décodage. On appelle GOP (Group of Picture) un ensemble d'images composé au moins d'une image I (tous ses macroblocks sont encodés à partir de prédictions spatiales) et d'images P et B (macroblocks codés à partir de prédictions temporelles). Nous voyons dans la figure 6 comment peuvent s'articuler des prédictions temporelles : la première image P est prédite à partir de l'image I, les images P suivantes sont prédites à partir d'images P, les images B sont prédites à partir d'images I ou P.

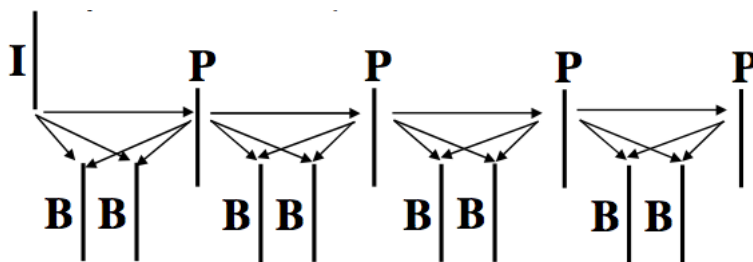


FIG. 6 – Un GOP et ses dépendances de prédictions

Erreur de prédiction Pour chaque macroblock, l'encodeur calcule une prédiction à partir d'échantillons spatiaux ou temporels. C'est donc le mode de prédiction qui sera transmis dans un premier temps. Le décodeur réceptionne le mode de prédiction et le calcule pour obtenir la prédiction du macroblock. Bien sûr, la prédiction n'est pas exacte. Le codeur doit donc calculer l'erreur de prédiction qui est la différence entre le macroblock à encoder et la prédiction calculée. La prédiction est bien sûr choisie de façon à minimiser le bloc d'erreur, afin, toujours, de minimiser la quantité d'information à sauvegarder. Si M est un macroblock à encoder et P sa prédiction calculée par le codeur, alors l'erreur de prédiction E s'écrit :

$$E_{ij} = M_{ij} - P_{ij}$$

Transformée et quantification Une fois l'erreur de prédiction obtenue, elle est transformée par une transformation en cosinus discrète puis elle est quantifiée. Pour chaque macroblock de l'image, nous transmettons donc un mode de prédiction (qui peut être spatial ou temporel) ainsi que l'erreur de prédiction.

Encodage entropique Les données issues de l'encodage d'un macroblock ne sont pas transmises telles quelles, c'est-à-dire sous forme d'entier. La dernière étape consiste en un encodage entropique des valeurs. Un encodage est dit entropique lorsqu'il tient compte des valeurs préalablement codées pour définir la manière de coder les valeurs suivante. Ces codes permettent ainsi une compacité accrue en échange d'une complexité analogue.

En résumé Nous encodons l'image F_n . Dans un premier temps, le codeur détermine si l'image est encodée en prédiction spatiale ou en prédiction temporelle. Ce choix s'opère en tenant compte des erreurs de prédiction générées par les différentes prédictions. Les erreurs de prédiction sont ensuite transformées par une transformation en cosinus discrète puis quantifiées. Enfin, les modes de prédictions et les erreurs de prédictions transformées et quantifiées sont encodés via un code entropique. Toutes ces étapes sont représentées dans la figure 7.

Nous allons maintenant détailler chacune des étapes.

3.2 Prédictions

Nous avons vu qu'il existait deux types de prédictions pour calculer une approximation d'un macroblock à encoder. Il existe en tout 13 prédictions spatiales et 4 prédictions temporelles. Elles permettent de calculer des approximations des macroblocks. L'approximation la plus fidèle, c'est-à-dire celle qui minimise l'erreur de prédiction est retenue pour l'encodage du macroblock.

Il existe également un mode dit IPCM (Intra-frame Pulse Code Modulation) qui ne fait aucune prédiction sur le macroblock mais qui permet de transmettre un macroblock sans aucune modification. Ce mode génère évidemment des débits bien plus importants que les prédictions spatiales.

3.2.1 Prédiction spatiale

Pour le macroblock X, toutes les prédictions spatiales se calculent à partir des valeurs des pixels limitrophes des macroblocks A, B et C (voir figure 8).

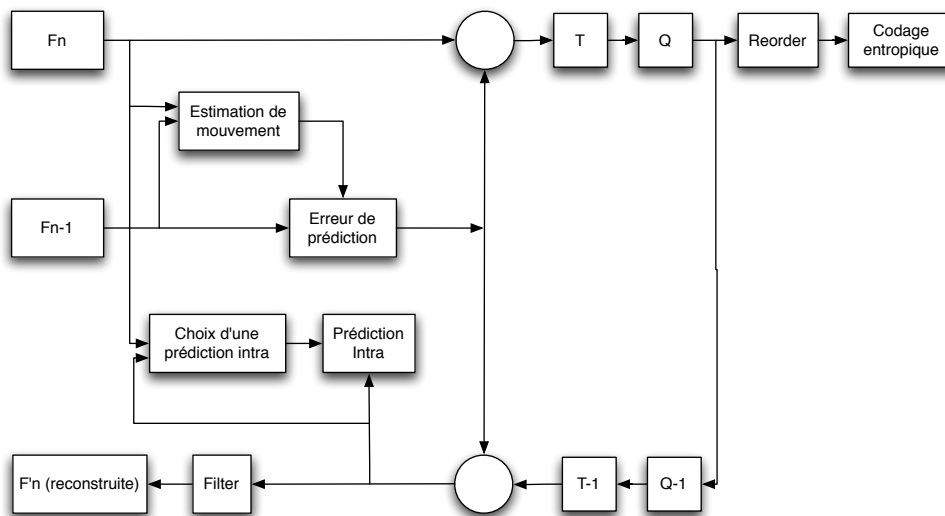


FIG. 7 – Schéma récapitulatif

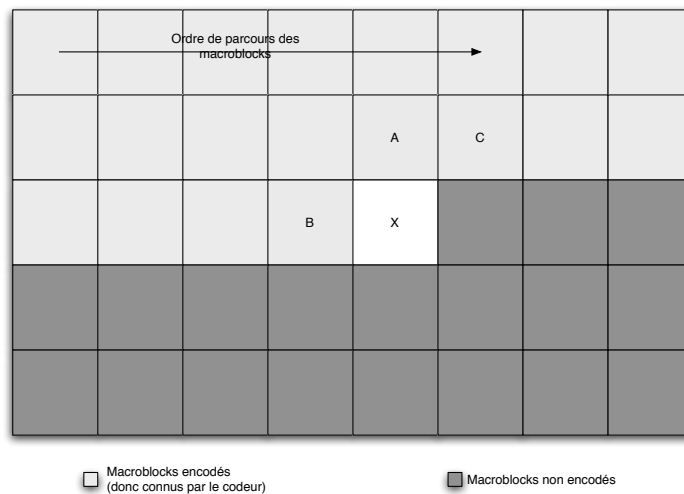


FIG. 8 – macroblocs utilisés pour la prédiction spatiale du macroblock X

L'H.264 offre deux types de prédictions spatiales. Les prédictions couvrant les 16×16 pixels d'un macroblock et les prédictions sur des sous-blocs de 4×4 pixels (voir figure 9).

les 4 prédictions spatiales s'appliquant à un macroblock entier sont exactement les 4 premiers types de prédictions pour les sous-macroblocks de 4×4 pixels, à savoir : une prédiction horizontale, une prédiction verticale, une prédiction moyenne et une prédiction en diagonale.

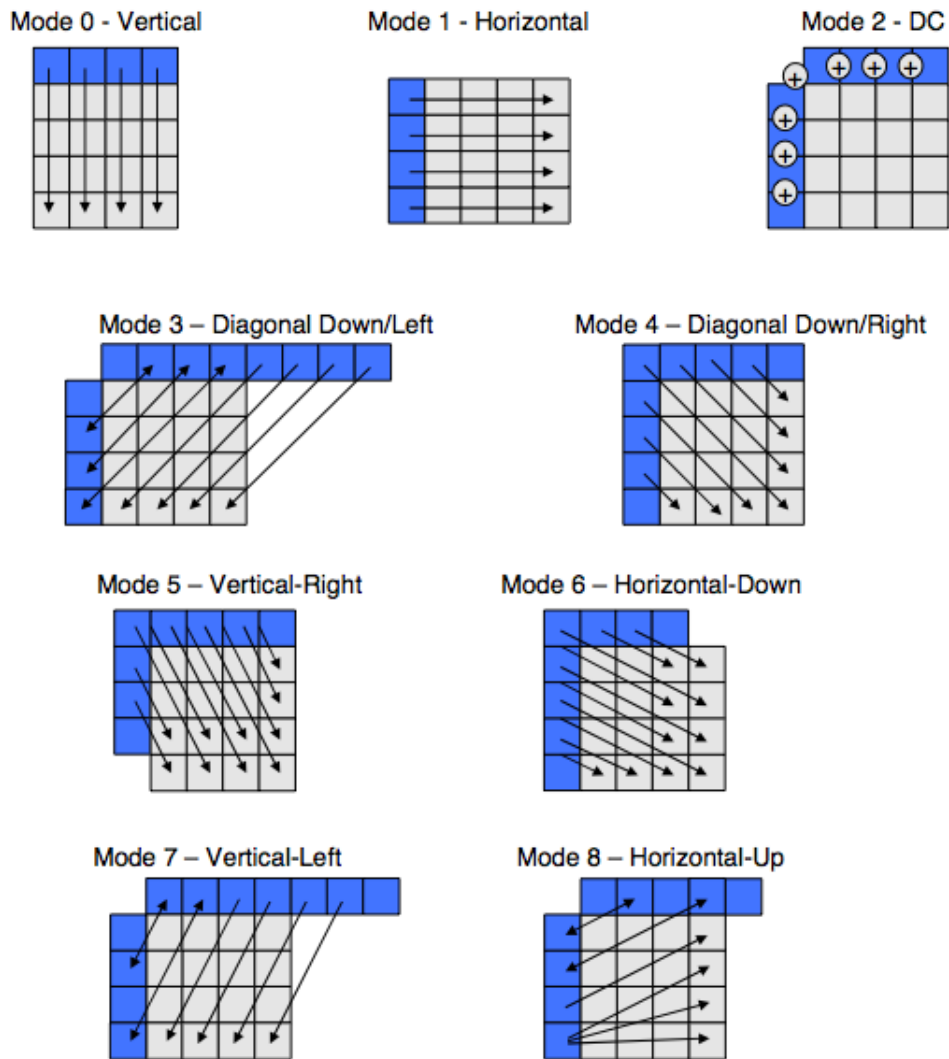


FIG. 9 – Prédictions spatiales pour des sous-blocs de 4x4 pixels

Un macroblock compressé à partir d'une prédiction spatiale est appelé macroblock I. Une image codée uniquement par des macroblocks I est appelée image I. Ces images servent souvent d'image de référence pour les prédictions temporelles ou de synchronisation dans une séquence : toutes les x images, le codeur peut choisir d'encoder une image en I afin de garantir que si un problème survient au décodage, il suffira de x images au plus pour retrouver un flux normal. Les images I_s permettent aussi d'accéder directement à un endroit de la séquence. En effet, si toutes les images suivant la première n'utilisaient que des prédictions temporelles, il faudrait pour décoder la i ème image décoder toutes les images précédentes. Par contre, si cette i ème image est de type I, elle contient

toutes les données nécessaires à sa décompression.

Les images I peuvent être ainsi comparées à une image compressée en JPEG : l'intégralité des données nécessaires à la décompression de l'image se trouve dans cette dernière, contrairement aux images prédites temporellement, qui nécessitent le décodage préalable des images utilisées pour les prédictions temporelles.

Nous avons également évoqué les prédictions SP et SI. Nous dirons simplement qu'elles sont conçues pour la diffusion sur Internet (streaming). Les images SP ou SI sont des ponts entre plusieurs flux encodés à des débits différents d'une même vidéo[9]. Cet aspect de l'H.264 n'entrant pas en compte dans ce stage, nous ne le détaillerons pas plus.

3.2.2 Prédiction temporelle

Une prédiction temporelle utilise un macroblock d'une autre image comme base de sa prédiction. L'encodeur définit alors un vecteur de mouvement. Ce vecteur de mouvement permet la construction d'une prédiction. Puis, comme dans le cas d'une prédiction spatiale, une erreur de prédiction est calculé, transformée, quantifiée, encodée puis transmise. L'H.264 permet des vecteurs de mouvements d'une précision allant jusqu'au 1/8 de pixel. Il s'agit alors d'interpoler la vidéo afin d'obtenir une résolution triple de la résolution de départ.

Un macroblock compressé de telle façon est dit macroblock de type P (ou macroblock P). Il existe une deuxième prédiction temporelle où ce n'est plus un, mais deux macroblocks de deux image différentes de la séquence qui servent de base à la prédiction. Un macroblock compressé à partir d'une telle prédiction est dit macroblock de type B (ou encore macroblock B) (voir figure 5).

Il existe une deuxième prédiction B, dite *weighted bi-prediction* où les deux macroblocks servant à calculer la prédiction sont pondérés par des coefficients différents (et non pas égaux comme c'est le cas dans une prédiction B classique).

Ainsi, une image compressée avec au moins un macroblock P, respectivement B, sera une image de type P, respectivement B.

3.3 Transformée et Quantification

3.3.1 Transformation

L'H.264 possède 3 types de transformations (voir figure 10).

- Une transformation d'Hadamard utilisée sur le bloc 4x4 des coefficients DC (coefficients de plus basse fréquence) pour les prédictions spatiales sur un macroblock entier.
- Une transformation d'Hadamard utilisée sur les bloc 2x2 des coefficients DC pour les prédictions spatiales sur un macroblock entier.
- Une transformation basée sur la transformation en cosinus discrète pour tous les autres blocs 4x4 de données.

Soit un macroblock Mb_{ij} pour lequel l'encodeur a calculé la prédiction Pre_{ij} l'erreur de prédiction se présente sous la forme d'une matrice X_{ij} telle que :

$$X_{ij} = Mb_{ij} - Pre_{ij}$$

Afin de réduire encore les données représentant le macroblock, l'erreur de prédiction est transformée par une transformation en cosinus discrète. La transformation en cosinus discrète est une transformation proche de la transformée de

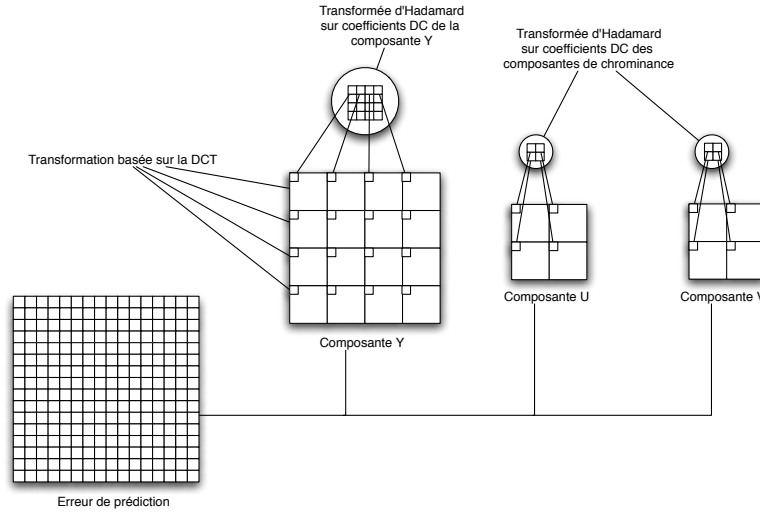


FIG. 10 – Prédications spatiales

Fourier discrète (DFT). Le noyau de projection est un cosinus et génère donc des coefficients réels, contrairement à la DFT, dont le noyau est une exponentielle complexe et qui génère donc des coefficients complexes.

Pour les images naturelles, la DCT est la transformation qui se rapproche le plus de la transformée de Karhunen-Loève[10] qui fournit une décorrélation optimale des coefficients pour un signal markovien[8].

La transformation en cosinus discrète se calcule par la formule suivante :

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

La transformation en cosinus discrète peut cependant se calculer comme un produit de matrice. La matrice de transformation sur un block de taille 4*4 s'écrit :

$$A = \begin{pmatrix} \frac{1}{2} \cos(0) & \frac{1}{2} \cos(0) & \frac{1}{2} \cos(0) & \frac{1}{2} \cos(0) \\ \sqrt{\frac{1}{2}} \cos(\frac{\pi}{8}) & \sqrt{\frac{1}{2}} \cos(\frac{3\pi}{8}) & \sqrt{\frac{1}{2}} \cos(\frac{5\pi}{8}) & \sqrt{\frac{1}{2}} \cos(\frac{7\pi}{8}) \\ \sqrt{\frac{1}{2}} \cos(\frac{2\pi}{8}) & \sqrt{\frac{1}{2}} \cos(\frac{6\pi}{8}) & \sqrt{\frac{1}{2}} \cos(\frac{10\pi}{8}) & \sqrt{\frac{1}{2}} \cos(\frac{14\pi}{8}) \\ \sqrt{\frac{1}{2}} \cos(\frac{3\pi}{8}) & \sqrt{\frac{1}{2}} \cos(\frac{9\pi}{8}) & \sqrt{\frac{1}{2}} \cos(\frac{15\pi}{8}) & \sqrt{\frac{1}{2}} \cos(\frac{21\pi}{8}) \end{pmatrix}$$

La transformation s'écrit alors :

$$Y = AXA^T$$

La transformation utilisé dans l'H.264 est basé sur une DCT avec cependant certaines différences fondamentales :

- La transformation est en nombre entier. Elle s'effectue donc par des opérations entières.

- Il est possible de garantir une erreur de transformation nulle
- Le coeur de la transformation est implémentable à partir d'addition et de décalage binaire

3.3.2 Quantification

La quantification est le seul processus dans la compression H.264 qui engendre des pertes de qualité dans la vidéo. L'H.264 utilise deux paramètres de quantification : Q_P et Q_δ . Q_P est le paramètre de quantification pour les données vidéos de la composante de luminance et Q_δ est l'écart entre Q_P et le paramètre de quantification pour les données vidéo des composantes U et V. Un paramètre de quantification est associée à une valeur de quantification via la table de quantification de l'H.264. Un paramètre de quantification a une valeur allant de 0 à 51. On appelle QStep la valeur associée à un paramètre de quantification QP donné.

QP	0	1	2	3	4
QStep	0.625	0.685	0.8125	0.875	1
	5	6	7	8	9
	1.125	1.25	1.135	1.625	1.75
	10	11	12	...	18
	2	2.25	2.5	...	5
	24	...	30	...	36
	10	...	20	...	40
	42	...	48	...	51
	80	...	160	...	224

TAB. 1 – Paramètre de quantification et valeur associée

Nous remarquons que les valeurs associées doublent lorsque les paramètres de quantification sont incrémentés de 6 unités. La matrice des coefficients DCT quantifiés Z_{ij} se calcule de la façon suivante :

$$Z_{ij} = \text{round} \left(\frac{W_{ij}}{QStep} \right)$$

Ce sont les coefficients de la matrice W_{ij} qui seront au final encodés puis transmis comme flux H.264.

3.4 Encodage

Chaque donnée transmise dans un flux H.264 est codée afin de réduire au maximum le nombre de bits transmis. L'H.264 utilise 3 codes : un code pour les données non-vidéos et deux codes pour les données de vidéos.

Exp-Colomb[7] C'est un code à longueur variable qui est utilisé pour toutes les données qui ne sont pas à proprement parler des données de vidéos : mode de prédictions, vecteurs de mouvements, etc... Le codage Exp-Golomb est assez simple : un mot du code s'écrit toujours sous la forme binaire : [préfixe]1[suffixe]. Le préfixe est composé uniquement de zéros. Le suffixe peut être composé de

zéros ou de uns. Le préfixe et le suffixe d'un mot ont toujours la même longueur. La valeur que code le mot se calcule de la façon suivante :

$$v = 2^{\text{taille}(\text{suffixe})} + \text{valeur}(\text{préfixe}) - 1$$

Par exemple, le mot 00000 1 10100 équivaut à :

$$2^5 + 20 - 1 = 11$$

CAVLC[1] Context-Adaptive Variable Length Coding. Comme son nom l'indique : le CAVLC est un code entropique à longueur variable. Le mot "entropique" indique que le code s'adapte au fur et à mesure des valeurs codées pour déterminer le codage le plus efficace pour les valeurs à venir. Le CAVLC a été conçu en tenant particulièrement compte des propriétés généralement constatées des blocs de coefficients DCT après quantification, à savoir :

- Les blocs de coefficients DCT après quantification sont composés en grande partie de zéros, CAVLC utilise une syntaxe run/level afin de réduire au mieux de longue séquence de zéros
- Après le parcours en zig-zag, les plus haut coefficients non nuls (c'est-à-dire les coefficients situés à la fin de la chaîne) sont souvent des +1 ou des -1. CAVLC permet de coder efficacement ces suites de +/-1
- Le nombre de coefficients non nuls dans des blocs voisins est souvent corrélé. CAVLC tient compte de ce fait et utilise des tables en fonction du nombre de coefficients non nuls des blocs voisins.
- Les coefficients de basse fréquence ont souvent des valeurs plus importantes que les coefficients de haute fréquence.

CABAC[12] Context-Adaptive Binary Arithmetic Coding. Le CABAC est également un code entropique. Le CABAC détermine un contexte d'après la valeur à encoder et les valeurs précédemment encodées et met à jour des probabilités d'encoder un 0 ou un 1. Ces statistiques sont ensuite mises à jour à chaque bit encodé. Ce code est plus performant que le CAVLC mais d'une complexité également accrue.

4 Insertion dans des séquences vidéos

[5] définit pour l'insertion dans des séquences vidéos les applications suivantes :

Technique	But
Contrôle de copie	Interdire des copies (ex : DVD vidéos) illicites
Etiquette de diffusion	Identifier la source d'émission d'une vidéo diffusée
Empreinte numérique	Retrouver un propriétaire peu scrupuleux
Authentification	Assurer que la vidéo n'a pas été modifiée
Copyright	Prouver l'appartenance d'une vidéo

TAB. 2 – Techniques et applications d'insertion dans des séquences vidéos

Beaucoup de ces techniques sont issues des demandes de l'industrie de la vidéo cherchant à se protéger du piratage. Nous détaillons chacune de ces applications, et présentons pour certaines quelques applications

Contrôle de copie Le contrôle de copie s'est réellement posé avec l'avènement du Digital Versatile Disk (DVD) et des lecteurs DVD grand public en 1996. La qualité visuelle permise par le DVD ainsi que la non-dégradation lors de la copie (contrairement aux copies entre cassettes VHS) a fini par inquiéter les distributeurs des risques de piratages à grande échelle. Le droit à la copie permet à l'utilisateur de faire une copie directe de son film sur DVD, copie dite de première génération. Afin de limiter les tentations de piratage, les distributeurs souhaitent interdire la copie de copie, ou copie dite de deuxième génération. Le Copy Protection Technical Working Group (CPTWG) a été créé à cet effet en 1996. Un système de protection a été proposé[2] en 1999. Trois de ses composantes sont déjà implémentées et trois autres sont toujours en développement.

Les 3 méthodes actuellement utilisées sont

- Content Scrambling System (CSS) : sans une clef de déchiffrement, la vidéo apparaît brouillée.
- Analog Protection System (APS) : le flux vidéo peut être lu sur une télévision mais ne peut pas être enregistré sur un magnétoscope.
- Copy Generation Management System (CGMS) : un système de bit de contrôle (*copy-always, copy-once, copy-never*) qui indique au lecteur/enregistreur DVD quels sont les droits de copie de la vidéo actuellement lue.

Etiquette de diffusion L'étiquette de diffusion est utilisée pour identifier un programme sur un canal de transmission. On prend comme exemple la diffusion de publicité lors de grands événements sportifs. Le prix de diffusion d'un spot publicitaire se compte souvent en dizaines de milliers de dollars. L'annonceur souhaite donc s'assurer qu'il paie exactement le nombre de diffusions annoncé par le diffuseur. Plutôt que d'utiliser un opérateur humain chargé de

surveiller le canal de transmission, l'étiquette de diffusion agit comme un code-barre qui permet d'identifier automatiquement la diffusion un spot particulier. Ainsi, annonceurs et diffuseurs disposent de moyens automatiques pour contrôler les diffusions de spots publicitaires. Il s'agit donc de tatouer une vidéo avec une marque unique afin de l'authentifier lors de sa diffusion. Le projet européen Visual Identity Verification Auditor (VIVA) a déjà présenté un article[3] sur ce sujet.

Empreinte numérique Le plus grand facteur de piratage vient actuellement du fait que les données informatiques (fichiers audios, vidéos ou autres) sont très facile à diffuser via, entre autres, les réseaux point-à-point (peer-to-peer, P2P). Il n'est pas rare de pouvoir trouver sur les systèmes de partage (GNutella, Kazaaa, BitTorrent) des films américains avant leur diffusion dans les salles européennes. Il est évident que la faute n'incombe pas aux réseaux P2P mais aux personnes malhonnêtes mettant illégalement des fichiers à disposition d'autrui. L'idée de l'empreinte numérique est celle de l'empreinte digitale : pouvoir identifier le possesseur d'un fichier illégalement partagé afin d'identifier l'origine de la fuite.

Plusieurs méthodes ont déjà été proposées :

- tatouer l'empreinte à la réception du fichier, ce qui implique que le logiciel de tatouage soit disponible à l'utilisateur, ce qui pose le problème du *reverse-engineering* ;
- tatouer l'empreinte selon le chemin suivi par le fichier lors du téléchargement, ce qui implique la disposition, tout le long du réseau, de matériels compatibles permettant le tatouage, au fur et à mesure, du fichier.

Une autre application de l'empreinte numérique concerne les screeners, des copies réalisées directement dans une salle de cinéma à partir d'un caméscope standard. Bien que la qualité de ces vidéos soit faible comparée à celle d'un film, l'impact économique est important. Afin d'inciter les salles de cinéma à empêcher de telles pratiques, une empreinte indiquant la date ainsi que la salle de diffusion permettrait d'identifier le cinéma "coupable" si une copie du film est disponible sur les réseaux d'échanges.

Authentification L'authentification répond au besoin grandissant de certifier la provenance d'une vidéo. Les connexions à Internet avec des débits grandissant et la puissance des ordinateurs actuels permettent des créations et des manipulations de vidéos d'une qualité en constante progression. Il n'est pas rare de trouver dans sa boîte email une vidéo envoyé par une connaissance et qui soit incroyable. Comment peut-on déterminer l'origine de la vidéo, comment être sûr que cette dernière n'a pas été modifiée ? On parle ici de certificat d'authenticité.

Une première approche se basait sur la cryptographie. L'inconvénient est que la moindre altération de la séquence vidéo invalidait totalement la séquence. Il arrive que du bruit s'ajoute au signal durant une transmission sans que cela remette en cause l'authenticité de la vidéo. Le problème se ramène donc à une vérification de contenu. Un des premiers travaux dans ce sens a été l'utilisation de marqueurs temporels insérés dans des images de la séquence à authentifier[14]. Cette méthode permet de détecter efficacement les altérations temporelles (inversions ou insertions de scènes par exemple). Par contre, cette méthode ne donne aucune garantie sur le contenu modifié. Un élément d'une scène peut être modifié : la méthode ne nous l'indique en rien.

Tenant compte de ce constat, une deuxième méthode[4] consiste à insérer dans chaque image sa carte des régions, c'est-à-dire l'ensemble des régions distinctes qui composent l'image. Ainsi, une modification de l'image (masquage d'un élément, ajout d'un élément étranger) peut être détecté. Il n'est pas à douter, pour les raisons explicitées précédemment que ce domaine connaîtra un intérêt croissant dans les prochains mois.

Copyright Le copyright représente les droits réservés à l'auteur d'une œuvre. Afin de pouvoir garantir la propriété, et donc les droits, de l'auteur, la caractéristique principale de cette insertion doit être évidemment la robustesse du signal inséré. Si une copie illégale est débusquée, l'extraction de la marque permet à l'auteur de prouver ses droits sur l'œuvre.

Dans ce cas de figure, la notion de propriété et de droits d'auteurs est la principale application, et touche à un marché très fructueux. Les manœuvres mal-intentionnées seront donc légions et la notion de robustesse prend tout son sens. Des travaux sont à ce propos spécifiques à l'insertion de marques de copyright[11].

En résumé Les travaux d'insertion ont un large champ d'application. La vidéo étant un média ludique, on ne peut bien évidemment pas faire abstraction des intérêts économiques qui découlent de la commercialisation de vidéos (Video-on-Demand, DVD). De façon générale, on remarquera que l'insertion dans une vidéo détruit partiellement le travail de compression vidéo. Alors que la compression vidéo s'évertue à garder le maximum d'informations en un minimum de bits (en considérant certaines contraintes), l'insertion, du point de vue du codeur vidéo, ne fait qu'insérer du bruit. Deux points de vues sont alors possible : chercher à profiter au maximum des spécificités d'un codeur pour obtenir une insertion optimisée, ou définir des formats vidéos qui permettent de prendre en compte les informations que nous cherchons à insérer. Nous nous intéressons bien sûr, pour ce stage, à la première perspective.

5 Travaux

Les recherches effectuées se sont exécutées en deux temps. L'enrichissement de vidéos a été la première partie. Il a fallu trouver une méthode adéquate de détection de la région d'intérêt, c'est-à-dire dans notre cas une méthode permettant la détection de visage. Nous avons ensuite déterminé la manière de construire le message à embarquer dans la séquence. Puis nous avons choisi une méthode d'insertion dans le flux vidéo. Enfin, nous avons mesuré la dégradation engendrée par l'insertion sur l'ensemble de la vidéo mais également le gain apporté sur la région d'intérêt.

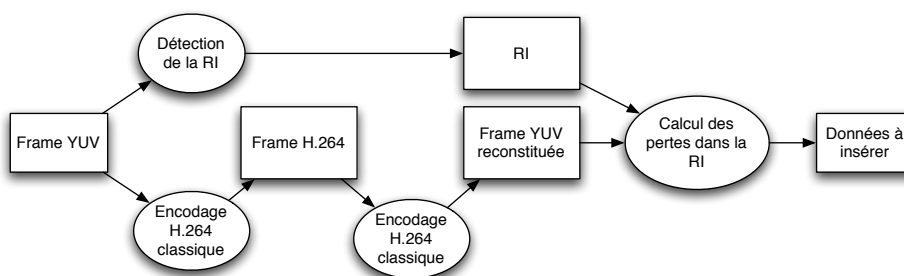


FIG. 11 – Calculs des pertes

En ce qui concerne la protection de la région d'intérêt, nous avons déjà, par le travail précédent, une méthode de détection de la région d'intérêt. Il fallait ensuite déterminer les différentes méthodes de protection possibles puis choisir la plus adaptée à notre application. Le travail sur la protection de la région d'intérêt n'a malheureusement pas pu être mené à son terme.

5.1 Enrichissement

5.1.1 Détection de la région d'intérêt

Il existe autant de calculs possible des régions d'intérêts que de nature de ces régions d'intérêts. Dans le cadre du stage, nous nous sommes restreint à la détection des visages. Pour la détection de visages, nous avons deux options :

- la détection par mouvement
- la détection par couleur

Nous avons privilégié la deuxième solution pour plusieurs raisons :

- la détection d'une région d'intérêt par la couleur se fait avec une seule image, alors que la détection par mouvement nécessite de garder des informations d'une image précédente.
- le calcul est simple, la méthode est donc rapide
- nous avons constaté empiriquement que la méthode est suffisamment fiable.

Dans un soucis d'optimisation, nous pouvons bien sûr coupler les deux méthodes de détection. Sur chaque image I (qui correspond souvent à des changements de scènes) nous lançons une détection par couleur et sur les images P et B, nous utilisons une détection par mouvement en tenant compte des positions précédentes de la région d'intérêt.

Supposons que nous souhaitons déterminer si un pixel P, qui possède 3 composantes : une de luminance P_Y et deux de chrominances P_U et P_V , est un pixel "de peau", c'est-à-dire qu'il ait une couleur de visage. La formule utilisée pour la détection de visage par couleur est :

$$\sqrt{(P_U - referenceU)^2 + (P_V - referenceV)^2} < seuil$$

Après expérimentation, nous avons déterminé que les valeurs donnant les meilleurs résultats étaient :

- $referenceU = 140$,
- $referenceV = 110$,
- $seuil = 10$.

Nous voyons ici un des avantages d'utiliser un espace YUV. Pour une surface donnée, il suffit d'un simple calcul sur les composantes U et V afin de déterminer si nous sommes en présence d'une partie de peau. Ce résultat exprime que les différences de "couleur" de peau ne sont en fait que des différences de luminosité. Des peaux blanches, noires, jaunes ou autres ont en fait la même couleur, ce n'est que l'intensité lumineuse réfléchie qui varie. Cette méthode n'est pas utilisable dans un espace RGB qui ne considère pas la couleur sous un aspect luminance-chrominance. Il existe toutefois des formules de conversion RGB-YUV qui permettent de travailler à partir de séquences RGB.

Nous allons maintenant détailler une détection de visage pas-par-pas. En premier lieu, nous appliquons la formule colorimétrique à chaque pixel. Nous marquons tous les pixels satisfaisant cette formule. Nous dirons qu'un pixel est "de peau" s'il satisfait la formule précédente.



FIG. 12 – Pixels marqués "de peau"

Le premier traitement (voir figure 12) nous permet de détecter tous les pixels appartenant à un visage mais également des pixels parasites ayant des composantes U et V satisfaisant la formule. Ainsi, un deuxième traitement, consistant en une succession d'ouvertures et de fermetures permet d'éliminer les-dits artefacts.

Nous appelons ouverture l'opération qui consiste en une contraction puis une dilatation. La fermeture est l'opération inverse : une dilatation puis une contraction.

La contraction est une opération consistant à supprimer les pixels ayant moins de x (ici, $x = 3$) pixels voisins de peau de l'ensemble des pixels de peau (voir figure 13).

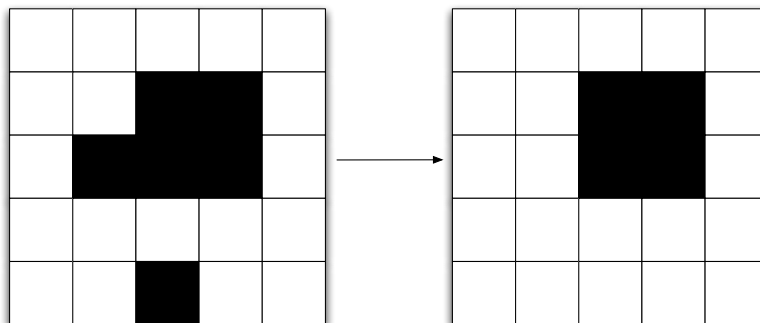


FIG. 13 – Contraction

Par opposition, nous appelons dilatation l'opération consistant à marquer des pixels non-marqués ayant plus de x (ici, $x = 3$) voisins dits de peau (voir figure 14).

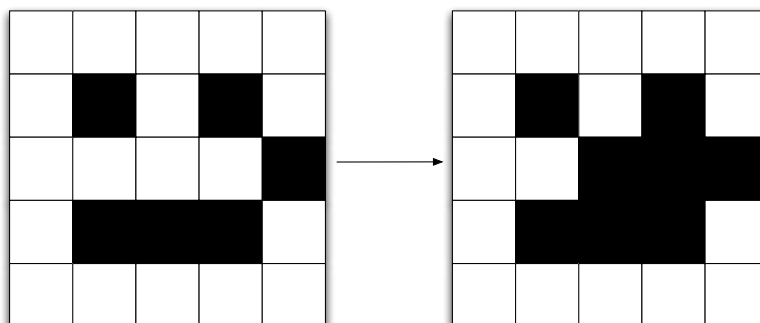


FIG. 14 – Dilatation

Nous appliquons une succession d'ouvertures-fermetures ce qui nous permet de supprimer les pixels (ou groupes de pixels) isolés de l'ensemble des pixels marqués comme pixels de peau. Nous obtenons uniquement les pixels de la région d'intérêt (voir figure 15).

Nous délimitons ensuite la région d'intérêt en tenant compte de la division de l'image en macroblocks. Nous obtenons donc notre région d'intérêt finale (voir figure 16), celle qui servira au calcul du message destiné à enrichir l'image.

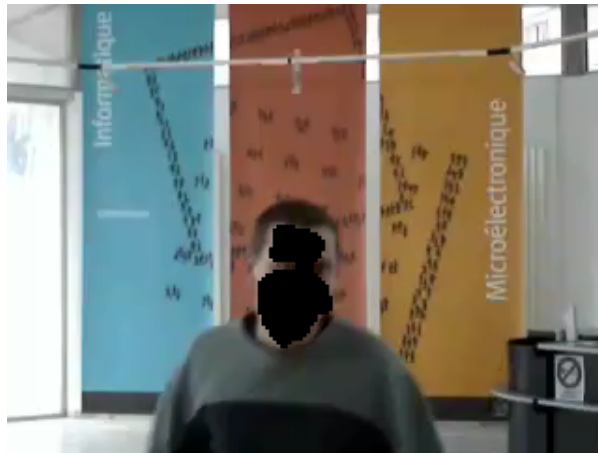


FIG. 15 – Ouvertures et fermetures successives éliminent les pixels isolés

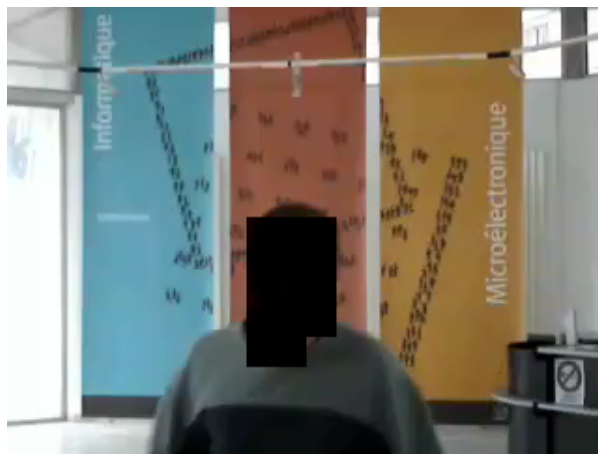


FIG. 16 – Région d'intérêt délimitée en macroblock

5.1.2 Calcul des pertes

Maintenant que la région d'intérêt est définie, nous devons construire le message qui sera embarqué dans l'image. En tenant compte du fait que l'oeil humain est plus sensible à la luminosité qu'à la chrominance, nous décidons de ne calculer que les pertes sur la composante de luminance. Le calcul des pertes est simple : c'est la différence entre la région d'intérêt originale et la région d'intérêt encodée. Nous faisons une soustraction pixel par pixel sur la valeur de la composante L de l'image. Nous calculons d'abord la matrice $Diff$, différence entre un macroblock M et sa reconstruction Rec , c'est-à-dire sa compression/décompression en H.264.

$$Diff_{i,j} = M_{i,j} - Rec_{i,j}$$

Nous construisons ensuite une chaîne C_{Diff} constitués des valeurs de la matrice $Diff$ réordonnées selon le parcours présenté dans la figure 17.

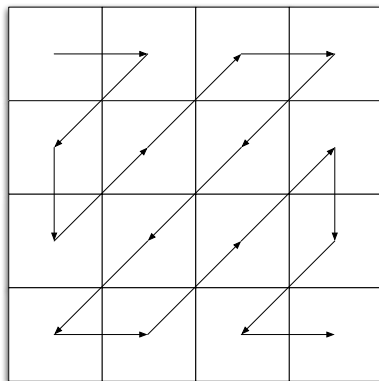


FIG. 17 – Parcours des coefficients DCT dans un bloc 4x4

Si la région d'intérêt est constitué de $m * n$ macroblocks, nous concaténons les $m * n$ C_{Diff_i} pour obtenir une chaîne $C_{Diff_{global}}$. Pour cela, nous parcourons la région d'intérêt dans l'ordre de lecture du codeur, à savoir de gauche à droite et de haut en bas (voir figure 18).

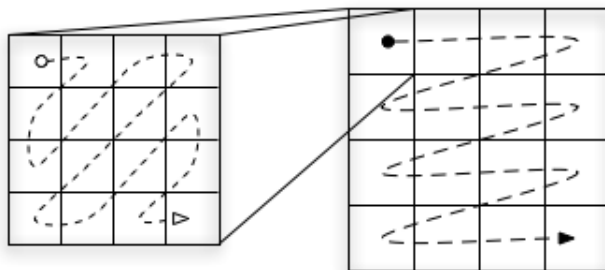


FIG. 18 – Ordre de parcours des coefficients d'une région d'intérêt de taille 4x4

Afin d'accélérer la reconstruction, nous construisons un message dont les 4 premiers entiers seront les coordonnées (abscisse puis ordonnée) et les dimensions (largeur puis hauteur) de la région d'intérêt. Ainsi le message final sera :

$$x_{r_1} y_{r_1} largeur_{r_1} hauteur_{r_1} C_{Diff_{global}}$$

Nous disposons maintenant du message à insérer.

5.1.3 Insertion de données

Nous avons construit le message à insérer dans l'image, nous devons maintenant définir les conditions et méthodes d'insertions.

Constitution du message Nous avons vu précédemment que le message à insérer est constitué des coefficients DCT quantifiés des macroblocks de la région d'intérêt placés dans un ordre allant de gauche à droite et du haut vers le bas. Cet ordre est choisi pour faciliter la reconstitution en Haute Qualité de la région d'intérêt au décodage.

Robustesse temporelle Dans le but de ne pas avoir de problème de fragmentation temporelle, nous insérons le message d'une région d'intérêt uniquement dans l'image contenant cette région d'intérêt. Ainsi, à la fin de la réception d'une image, nous pouvons reconstruire la région d'intérêt. Si une image est perdue lors de la transmission ou qu'un bruit excessif apparaît, cela ne gêne pas le décodage et l'enrichissement des images suivantes. Mais si l'insertion du message d'enrichissement d'une image dans elle-même présente l'avantage d'être robuste aux attaques temporelles (pertes d'images, images non décodables), l'inconvénient majeur est le problème de place.

Dans notre cas, la capacité d'une image est déterminée par le nombre de coefficient DCT non nuls de chaque macroblock. Nous arrêtons tout simplement l'insertion lorsque la capacité de l'image est atteinte. Nous ne pouvons donc pas garantir une reconstruction totale de la région d'intérêt en Haute Qualité mais on peut envisager une compression des données afin d'embarquer la totalité ou sinon le maximum de données pour la reconstruction de la région d'intérêt.

Insertion Le message et le protocole d'insertion sont maintenant fixé, il nous reste à définir la méthode d'insertion à proprement parler. Nous avons choisi une insertion sur les bits de poids faible. Cette méthode présente une capacité d'insertion satisfaisante (1/8ème des données modifiables) ainsi qu'une facilité de mise en œuvre appréciable. Nous avons décidé d'insérer les bits du message sur chaque coefficient non nul de chaque macroblock hormis ceux de la région d'intérêt.

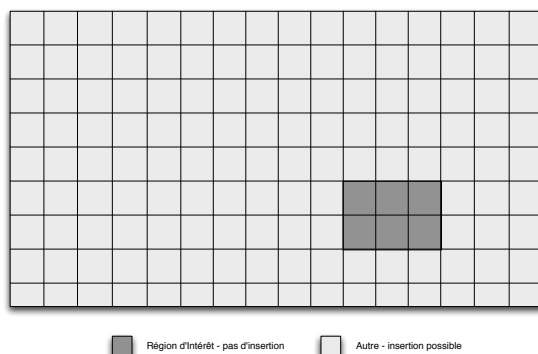


FIG. 19 – Macroblocks modifiables ou pas

Prédiction rétroactive Si on modifie les coefficients DCT d'un macroblock après l'encodage complet de l'image (c'est-à-dire qu'on fait une insertion sur le bit de poids faible de ces coefficients), au décodage toute prédiction utilisant ce

macroblock s'en verra affectée. Et par un effet de cascade (prédiction utilisant un macroblock utilisant le macroblock modifié), des modifications imprévues risquent d'affecter l'ensemble de l'image au décodage. Ce problème est résolu en calculant les prédictions à partir des blocs modifiés. Ainsi, les prédictions seront les mêmes au décodage qu'à l'encodage. Ceci est rendu possible par le fait que toutes les prédictions n'utilisent que des blocs préalablement encodés.

En effet, si une prédiction utilisait un macroblock m non encodé (i.e. non modifié), lors de l'insertion sur les LSBs des coefficients DCT de m , la prédiction au décodage ne serait pas la même que celle calculée à l'encodage.

Région d'intérêt On recherche sur la région d'intérêt une Haute Qualité, on ne peut donc pas se permettre d'insérer du bruit supplémentaire, en plus de la dégradation imposée par la compression. On évite donc toute insertion sur les coefficients de la région d'intérêt.

5.1.4 Résultats

Afin d'évaluer les performances de notre méthode, nous nous basons sur trois mesures :

- la dégradation engendrée par l'insertion,
- la hausse de débit engendrée par l'insertion,
- le temps de calcul nécessaire à l'insertion.

Dégradation La dégradation est mesurée par le Peak Signal Noise Rate (PSNR). Le PSNR est une mesure de distorsion utilisée en compression d'image. Il s'agit de quantifier la performance des codeurs en mesurant la qualité de reconstruction de l'image compressée par rapport à l'image originale.

Le PSNR se calcule de la façon suivante :

$$PSNR = 10 \log \left(\frac{d^2}{EQM} \right)$$

d représente la dynamique du signal, c'est-à-dire le nombre de niveaux du signal. Par exemple, pour une image en niveaux de gris dont chaque pixel est codé sur 8 bits, $d = 255$.

EQM est l'Erreur Quadratique Moyenne. L' EQM entre deux images I_o et I_r , toutes les deux de tailles $m * n$ se calcule de la façon suivante :

$$EQM = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I_o(i, j) - I_r(i, j)\|^2$$

Nous comparons le PSNR obtenu par une compression sans insertion et une compression avec insertion (voir tableau 3).

Nous remarquons que les PSNR des composantes U et V ne varient pas, ce qui est normal puisque l'insertion se fait uniquement sur la composante Y. La dégradation semble donc superficielle, les PSNR ne varient quasiment pas entre les deux méthodes. Nous allons comprendre pourquoi dans le paragraphe suivant.

	Séquence H.264 normale	Séquence H.264 enrichie
PSNR Y (dB)	37.98	36.47
PSNR U (dB)	40.42	40.42
PSNR V (dB)	39.99	39.99

TAB. 3 – Comparaison des PSNR d’une séquence non modifiée et d’une séquence enrichie

	Séquence H.264 normale	Séquence H.264 enrichie
Débit (kbts/s)	355.90	425.93

TAB. 4 – Comparaison des débits d’une séquence non modifiée et d’une séquence enrichie

Hausse de débit Voici le tableau comparatif (voir tableau 4), pour les mêmes séquences, des débits générés.

Le débit, pour la même séquence et avec une qualité visuelle similaire (nous avons vu que le PSNR varie très peu) passe de 355.90 kbts/s à 425.93 kbts/s, soit une augmentation de plus de 19%.

Temps de calcul Nous n’avons pas de mesure précise concernant les temps de calcul. Néanmoins, les compressions avec insertion ont été effectués sur une machine grand public (un Macbook Core Duo 2GHz) et ont atteint la vitesse de compression de 7 images par seconde, sachant que seule la première image était une image I, toutes les images suivantes étaient des images P ou B.

On peut donc supposer que sur un matériel dédié et optimisé, le temps réel soit atteignable.

5.2 Protection

La deuxième partie du stage à consister en l’étude des moyens de protection de la région d’intérêt. Nous souhaitons développer une méthode de protection automatique de la région d’intérêt. Grâce aux travaux effectués lors de l’enrichissement de séquence vidéos, nous étions déjà en possession d’une méthode de détection de visages. Nous pouvions attaquer directement la partie protection à proprement parler.

La protection de la région d’intérêt présente deux avantages. Par rapport à un cryptage total de l’image, crypter uniquement la région d’intérêt permet de réduire les temps de calculs à la compression comme à la décompression, et permet de garder possible l’éventualité d’utilisation en temps réel.

Le deuxième avantage d’un cryptage partiel (ou sélectif, selon les points de vue) de la vidéo est de laisser visible le reste de l’image. On dispose ainsi de deux niveaux d’informations, dont un protégé, dans un même flux.

Dans le cadre de la protection de la région d’intérêt d’intérêt, nous nous intéressons à deux critères :

- la robustesse. Dans le cas du cryptage vidéo, la faille la plus importante de la compression vidéo vient de son plus grand atout : la prédiction. On peut procéder par attaques statistiques sur les prédictions pour deviner

les zones prédites et en déduire des pistes pour casser le cryptage. Ce problème ne se pose pas dans le cas d'un cryptage total de l'image mais dans notre cas, le cryptage partiel de l'image fait apparaître cette faille.

- l'augmentation de la taille du fichier (ou du débit) suite au chiffrement des données, que l'on appellera *surpoids*.

5.2.1 Robustesse

Afin d'empêcher les prédictions des macroblocks qui forment la bordure supérieure ainsi que la bordure gauche de la région d'intérêt, nous transmettons ces macroblocks en mode ICPM. Ainsi, aucune information sûre ne peut être déduite des prédictions concernant les bords de la région d'intérêt.

Il est également évident qu'on ne permet pas à l'encodeur de prédire des macroblocks contigus à la région d'intérêt à partir de macroblock de la-dite région d'intérêt. Les possibilités de retrouver l'information étant alors aussi grandes que dans le cas précédent.

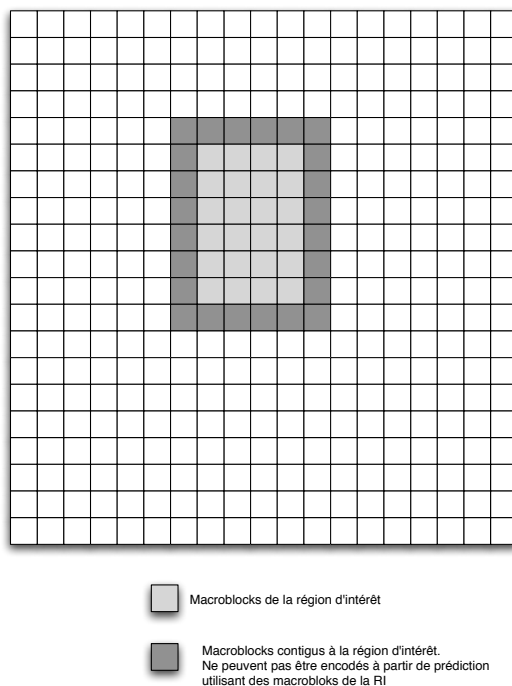


FIG. 20 – Interdiction de prédiction pour les macroblocks contigus à la région d'intérêt

5.2.2 Chiffrement des coefficients DCT

L'encryption des coefficients DCT avant encodage présente l'avantage d'être simple et facile. Néanmoins, à cause de l'optimalité des codes utilisés en H.264, le surpoids entraîné peut être non négligeable. Cette option est largement envi-

sageable si l'on ne dispose pas de contraintes de temps réel ou de ressources car à l'encodage/décodage se rajoute le chiffrement/déchiffrement de données.

Le problème de cette méthode vient du fait que le surpoids généré par le chiffrement n'est pas quantifiable.

5.2.3 Chiffrement du flux binaire

L'encryption au niveau du flux binaire, donc après encodage des coefficients DCT avec un des deux codes : CAVLC ou CABAC, permet de garantir le même nombre de bits avant et après encryption. Ainsi, le temps de calcul additionnel lié à l'encryption est le seul inconvénient. Malheureusement, on ne peut garantir la faisabilité d'une telle méthode. En effet, les deux codes sont contextuels, c'est-à-dire qu'ils évoluent en fonction du contexte.

Dans le cas du CAVLC, après chaque coefficient encodé, un calcul est effectué sur les valeurs des coefficients déjà traités pour choisir une des tables de code les plus appropriées. De plus, au décodage, le nombre de bits considérés pour le décodage d'un coefficient dépend directement de la valeur du coefficient précédent. Ainsi, pour rester valide, l'encryption devra garantir que le mot de code crypté code un coefficient compatible avec la méthode de décodage.

C'est sur cette méthode que s'est axée la deuxième partie du stage. Sur une idée originale de M. José-Marconi RODRIGUES[15] doctorant au LIRMM/ROB dans l'équipe ICAR, étudiant la protection de région d'intérêt dans des images JPEG, j'ai pu participer à l'étude de la transposition de la méthode à l'H.264. Les recherches sont restées sur un échec mais plusieurs idées sont apparues pour la réalisation d'une telle méthode.

Ainsi, une solution proposée est de vérifier à chaque chiffrement si le mot de code crypté est valide avec le décodage CAVLC, et sinon, de rechiffrer le mot crypté jusqu'à obtenir un mot de code crypté conforme avec le décodage. On pourra ensuite embarquer le nombre d'encryption pour chaque coefficient dans la vidéo par la méthode d'enrichissement.

Exemple Nous souhaitons chiffrer le mot de code CAVLC $M_1=1001011$. Ce mot de code indique au décodeur CAVLC que le prochain mot de code devra être lu sur 6 bits. Supposons maintenant qu'un premier chiffrement du mot de code $M_1=1001011$ nous donne le code $M_2=0011100$ et que ce mot de code indique au décodeur que la prochaine valeur codée doit être lue sur 7 bits. Il y aura donc un décalage d'un bit qui entraînera à coup sûr une erreur de décodage. Nous rechiffrons donc M_2 jusqu'à obtenir un mot de code M_i qui indique au décodeur de lire le mot de code suivant sur 6 bits. Nous insérons ensuite le nombre total de chiffrement par la méthode d'enrichissement précédente. Nous obtenons ainsi une méthode de protection viable.

Ce n'est là qu'une première idée, l'idéal étant bien sûr d'arriver à une méthode de chiffrement directement compatible avec la méthode de compression, c'est ce qu'on appelle l'aquamarquage.

6 Conclusion

Maîtrise du sujet Ce stage a constitué un travail complet de recherche. Toutes les facettes du travail de chercheur ont été abordées. Dans un premier temps, pour une connaissance plus approfondie, il a fallu appréhender le format vidéo utilisé, l'H.264, que je ne connaissais pas auparavant. Ce travail fut relativement rapide, quoi que peu aisé. Les concepts de la compression vidéo mais également les améliorations pointues apportées par l'H.264 ont été étudiés.

Bibliographie Dans un second temps, il a fallu constituer une bibliographie d'article et de livres permettant d'avoir une vue suffisante afin d'aiguiller les travaux et les méthodes à suivre. Les références à la fin de ce mémoire constitue une bonne base pour l'introduction et les premiers pas dans l'insertion de données dans des séquences vidéos.

Enrichissement La phase suivante du stage a consisté en le travail sur l'enrichissement de séquences. Ce travail a permis d'aborder différentes thématiques du traitement d'images (détection de régions d'intérêts, insertions) mais a également nécessité un apprentissage du code source du codeur H.264 de référence (JM 10.2). Ce codeur/décodeur, qui représente plus de 100 000 lignes de code, a permis la compréhension poussée de la majorité des mécanismes de la compression au format H.264.

Protection La dernière phase, concernant la protection de la région d'intérêt, bien qu'étant la moins fructueuse, est sûrement celle qui a procuré le plus de plaisir de recherches et d'effervescence intellectuelle. Elle a pris place dans la collaboration avec M. José-Marconi Rodrigues[15] sur un article de recherche sur le chiffrement de région d'intérêt dans une image JPEG. Devant déterminer les possibilités de porter la méthode proposée par M. Rodrigues, le travail de recherche précis (le codage CAVLC dans le processus de compression H.264) a permis un approfondissement très intéressant d'une partie du sujet. Comme expliqué précédemment, cette phase du stage ne s'est pas soldée par une réussite brillante mais a permis une compréhension avancée de l'encodage en H.264, ainsi qu'un travail en groupe stimulant.

Références

- [1] G. Bjontegaard and K. Lillevold. Context-adaptive vlc coding of coefficients. *JVT document JVT-C028*, 2002.
- [2] J. Bloom, I. Cox, T. Kalker, J-P. Linnartz, M. Miller, and C. Traw. Copy protection for dvd video. *Proceedings of the IEEE 87 (7)*, 1999.
- [3] G. Depovere, T. Kalker, J. Haitsma, M. Maes, L. de Strycker, P. Termont, J. Vandewege, A. Langell, C. Alm, P. Norman, G. Reilly, B. Howes, H. Vaanholt, R. Hintzen, P. Donnelly, and A. Hudson. The viva project : digital watermarking for broadcast monitoring. *Image Processing*, 1999.
- [4] J. Dittmann, A. Steinmetz, and R. Steinmetz. Content-based digital signature for motion pictures authentication and content-fragile watermarking. *Multimedia Computing and Systems*, 1999.
- [5] Gwenaël Doërr and Jean-Luc Dugelay. A guide tour of video watermarking. *Signal Processing : Image Communication 18*, 2003.
- [6] Jacky DURAND. <http://www.liberation.fr/page.php?article=315627>, 2005. La caméra, nouvelle arme des policiers européens.
- [7] S. W. Golomb. Run-length encoding. *IEEE Transactions on Information Theory*, 1966.
- [8] Charles M. Grinstead and J. Laurie Snell. *Introduction to Probability*. American Mathematical Society, 2003.
- [9] M Karczewicz and R Kurceren. The sp and si frames design for h.264/avc. *IEEE Transactions on Circuit and Systems for Video Technology*, 2003.
- [10] J.T. Karhunen and J. Joutsensalo. Sinusoidal frequency estimation by signal subspace approximation. *Signal Processing*, 1992.
- [11] K. Nahrstedt L. Qiao. Watermarking methods for mpeg encoded video : towards resolving rightful ownership. *Proceedings of the IEEE International Conference on Image Processing*, 1998.
- [12] D. Marpe, H. Schwarz, and T. Wiegand. Context-based adaptive binary arithmetic coding in the h.264/avc video compression standard. *IEEE Transactions on Circuit and Systems for Video Technology*, 2003.
- [13] Michael McCahill and Clive Norris. Cctv in london, 2002.
- [14] B. Mossaberi, M. Sieffert, and R. Simard. Content authentication and tamper detection in digital video. *Multimedia Computing and Systems*, 1999.
- [15] J. M. Rodrigues, W. Puech, P. Meuel, J.C. Bajard, and M. Chaumont. Face protection by fast selective encryption in a video. *The Institution of Engineering and Technology Conference on Crime and Security*, 2006.
- [16] Joint Video Team. *ITU-T Recommendation H.264 - ISO/IEC 14496-10 AVC*. JVT, 2003.