

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.



**Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.**

Option : Intelligence Artificiel

Sujet :

**Protection de copyright dans la
norme de compression Vidéo
H.264/AVC**

Présenté par : Baouche Yasmina
Boudissa Razika

Promotrice: M^{me} Ait Saadi Karima

Organisme d'accueil : Centre de Développement des Technologies Avancées CDTA

Soutenu le: 23/10/2007

Devant le jury composé de :

M. BENNOUARE

M^{me}. BENBLIDIA

M. WOULD AISSA

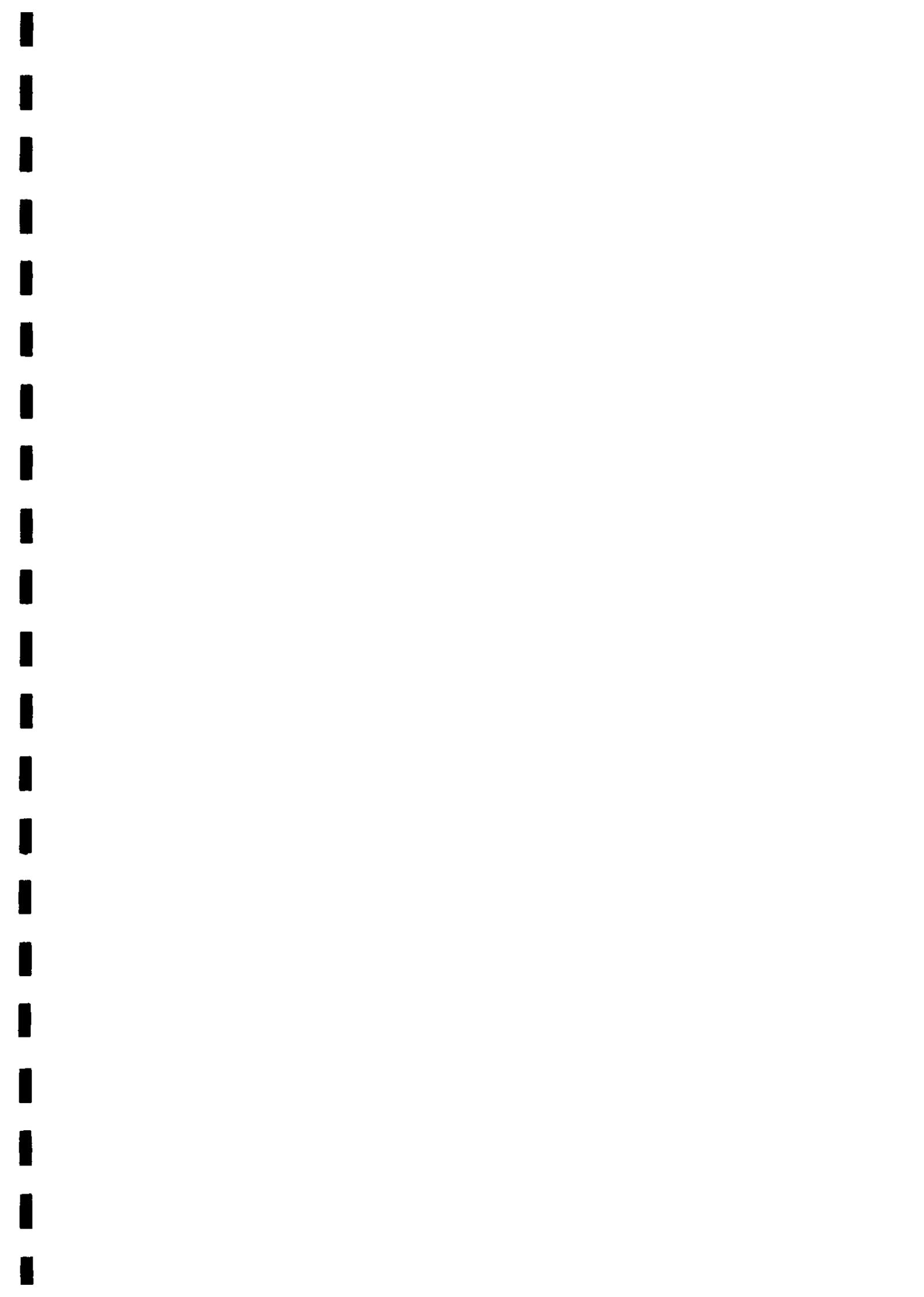
Président

Examinatrice

Examineur

- promotion 2006/2007-

MIG-004-170-4





Remerciements



Tout d'abords, nous remercions الله عزّ وجلّ de nous avoir donné la santé et la volonté pour mener a bien ce modeste travail.

Nous remercions vivement notre promotrice M^{me} AIT SAADI d'avoir accepté de nous encadré et proposé ce sujet, nous exprimons toutes notre gratitude et nos s'incère remerciements pour tout leur effort, coopération, aide, patience, encouragement et vaillance pour la réalisation de ce travail, prient dieu de leur donné toute la force pour continuer leur noble mission.

Nous tenons à remercier les membres de jury de nous avoir fait l'honneur d'être examinateurs tous en apportant leurs remarques et leurs contributions à l'enrichissement de ce mémoire.

Nous remercions encore tous les enseignants du département d'informatique, pour nous avoir formé tout au long de notre cursus.

Merci à monsieur MERAOUBI pour ses conseils et son aide.

Merci pour tous ceux que d'une manière ou autre, nous avons apportés de l'aide pour la réalisation de ce travail.

Razika et Yasmina



Dédicace



Je dédie ce travail à :

Ma très chère mère qui ma désire toujours le bonheur.

Mes chères sœurs : Fatiha, Hassina, F.zohra, Farida, Khadidja, Faiza.

Mes frères : Mohamed et Abdennour et je les remercie pour leur soutien tout au long de mes études.

Mes neveux et nièces.

A mon binôme et amie Razika et toute sa famille.

A mes amis avec qui j'ai passé des moments formidables durant mes cursus.

BAOUCHE Yasmina



Dédicace



Je dédie ce travail :

Bien évidemment, à ma mère et ma famille pour leur confiance et leur soutien sans faille au cours de toutes ces années, et pour m'avoir supporter pendant mes études.

A mes amis avec qui j'ai passé des moments formidables durant ces quelque années à l'université.

A Yasmina et toute la famille BAOUCHE, et je la remercie d'avoir supporter mes allés et venus.

BOUDISSA Razika

RÉSUMÉ

L'essor du multimédia et des supports numériques, qui permettent l'obtention d'une copie parfaite, conduit de plus en plus les fournisseurs de contenus à se poser le problème du piratage des données. La protection et le tatouage des media numériques sont rapidement devenus un axe majeur de recherche dans le domaine du traitement d'images.

La problématique que nous posons dans ce mémoire est de proposer une solution de protection de copyright dans la norme H.264/AVC. Pour ce faire, nous proposons deux algorithmes de tatouage dans le domaine compressé basés respectivement sur les coefficients DCT 4x4 et coefficients DCT quantifiés, afin d'insérer une marque invisible et robuste.

Mots clés : traitement d'images, compression vidéo, tatouage vidéo, sécurité multimédia.

ABSTRACT

The recent development of digital multimedia leads the providers to deal with the hacking of multimedia contents. The digital protection of the original media has become a major research issue in the image processing area.

In order to protect a video, three main techniques could be used:

- Cryptography.
- Scrambling.
- Watermarking.

In this thesis, we focus essentially on watermarking techniques. Our contribution exposed in this thesis is the copyright protection in the novel codec video H.264/AVC. For this, we propose two watermarking algorithms: working during the video compression H.264.

The first one is based on DCT 4x4 coefficients and the second use the quantized coefficients to insert an invisible watermark in the digital video content. *

Key words: Image Processing, Video Compression, Watermarking video, Multimedia Security.

TABLE DE MATIERES

Introduction générale	1
-----------------------------	---

Chapitre I : Généralité

1. Introduction.....	3
2. Notions de base du tatouage	3
2.1 Définition	3
2.2 Caractéristiques de la marque	4
2.3 Différents types de tatouage	5
2.4 Attaques	5
3. Notion de base du signal vidéo et sa compression	7
3.1 Echantillonnage	8
3.2 Formats vidéo	9
3.3 Compression vidéo	9
4. Application de tatouage vidéo	11
5. Conclusion	12

Chapitre II : La norme de compression vidéo H.264/AVC

1. Introduction	13
2. Historique de la norme de compression vidéo	13
3. Principales caractéristiques de la norme H.264/AVC	15
4. Fonctionnement du codeur H.264/AVC	17

4.7 Le filtre de déblocage	40
5. Les profils	42
6. Conclusion	43

Chapitre III : Le tatouage vidéo

1. Introduction	44
2. Principe général du tatouage	44
3. Tatouage vidéo	46
4. Les différents algorithmes de tatouage vidéo	47
4.1 Techniques de tatouage vidéo provenant des schémas des images fixes ...	47
4.2 Techniques adaptées à la vidéo	48
5. Conclusion	54

Chapitre IV : Conception et implémentation

1. Introduction	55
2. Méthode 1 : Tatouage appliqué aux coefficients transformés	57
2.1 Algorithme proposé	57
2.1.1 Traitement de la marque	57
2.1.2 Insertion de la marque	58
2.1.3 Extraction de la marque	61
2.1.4 Reconstruction de la marque	62
2.2 Choix expérimentaux et résultats	62
2.2.1 Choix de la taille de la marque.....	62

2.2.2	Localisation de la position d'insertion.....	63
2.2.3	Choix du n ^{ième} bit à substituer.....	66
3.	Méthode 2 : Tatouage appliqué aux coefficients transformés quantifiés	69
3.1	Algorithme proposé	69
3.1.1	Traitement de la marque	71
3.1.2	Insertion de la marque	71
3.1.2.1	La recherche des blocs d'insertion.....	71
3.1.2.2	Le choix du bloc d'insertion.....	73
3.1.2.3	La sélection des coefficients à tatouer	73
3.1.2.4	Le processus d'insertion.....	75
3.1.3	Extraction et reconstruction de la marque	76
3.2	Choix expérimentaux et résultats.....	77
3.2.1	Le choix des paramètres de quantification.....	77
3.2.2	Le choix de la position d'insertion.....	79
4.	Conclusion	80

Chapitre V : Robustesse face aux attaques

1.	Introduction	82
2.	Résultats expérimentaux	82
2.1	Attaque par compression H.264.....	82
2.2	Attaque par compression H.263.....	86
2.3	Attaque par transcodage YUV/RGB	89

2.4 Attaque par redimensionnement	92
3. Étude comparative des deux algorithmes.....	94
4. Présentation du logiciel	96
Conclusion générale et perspectives.....	101
Bibliographies.....	103

LISTE DE FIGURES

Figure I.1 : Caractéristiques de tatouage	5
Figure I.2 : Bloc diagramme d'un codeur vidéo	10
Figure II.1 : Bloc diagramme du codeur H.264/AVC	18
Figure II.2 : Bloc diagramme du décodeur H.264/AVC	19
Figure II.3 : Structure du GOP	20
Figure II.4 : Découpage d'un macrobloc pour la compensation de mouvement	20
Figure II.5 : Images de références pour la prédiction à compensation de mouvement.....	23
Figure II.6 : Exemple de prédiction de pixel et sous-pixel	24
Figure II.7 : Labellisation des échantillons de prédiction 4×4	27
Figure II.8 : Huit de neuf modes de prédiction intra	27
Figure II.9 : Modes de prédiction des blocs 4×4 de luminance	28
Figure II.10 : Modes de prédiction des blocs 16x16 de luminance	29
Figure II.11 : Ordre de parcours d'un macrobloc 16×16	31
Figure II.12 : Ordre de balayage des blocs de résiduel dans un macrobloc	31
Figure II.13 : Parcours en zigzag d'un bloc 4×4	38
Figure II.14 : La performance du filtre anti-blocs sur des images compressées, sans et avec filtre	40
Figure II.15 : Visualisation unidimensionnelle d'un bord de bloc	41
Figure II.16 : Profils de H.264/AVC	43
Figure III .1 : Schéma d'insertion	45
Figure III .2 : Schéma de détection	45
Figure IV.1 : La marque à insérer	56

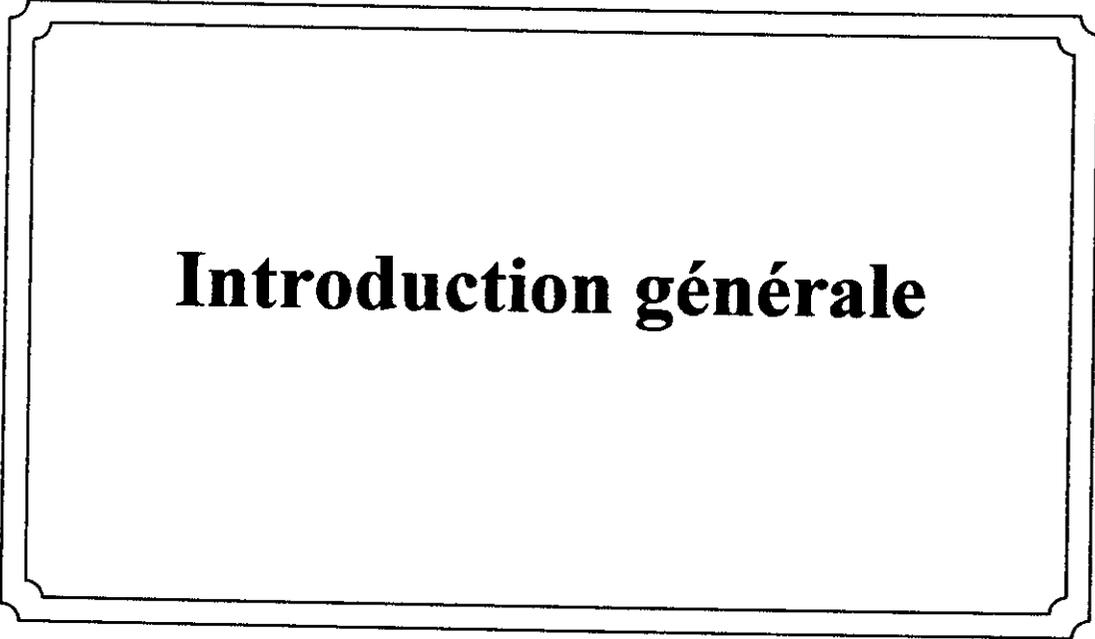
Figure IV.2 : Schéma synoptique du système de tatouage vidéo développé	57
Figure IV.3 : Schéma synoptique du processus d'insertion basé sur la DCT 4x4 du codeur H.264.....	59
Figure IV.4 : Schéma d'insertion de la marque	60
Figure IV.5 : Exemple de sélection d'un bloc d'insertion dans un macrobloc.....	61
Figure IV.6 : Schéma global d'extraction dans le décodeur H.264/AVC	61
Figure IV.7 : Parcours en zigzag des coefficients transformés dans un bloc	63
Figure IV.8 : Séquence « claire » compressée et tatouée à différentes positions	64
Figure IV.9 : Séquence « container » compressée et tatouée à différentes positions	65
Figure IV.10 : Insertion dans le 5 ^{ème} bit MSB du coefficient choisi a (3,3).....	66
Figure IV.11 : Séquence « claire » tatouée en différentes positions du coefficient binaire à tatouée	67
Figure IV.12 : Séquence « container » tatouée en différentes positions du coefficient binaire à tatouée	68
Figure IV.13 : Schéma globale d'insertion dans le module de quantification du standard H.264/AVC.....	70
Figure IV.14 : Deux images extraites d'une vidéo, (a) zone texturée, (b) zone homogène	72
Figure IV.15 : Le parcours en zigzag dans un bloc 4x4	74
Figure IV.16 : Sélection des coefficients de moyenne fréquence dans un bloc 4x4 ...	74
Figure IV.17 : Schéma d'extraction de la marque dans le décodeur H.264/AVC	76
Figure IV.18 : La marque extraite pour différentes valeurs de QP	77
Figure IV.19 : Séquence « claire » tatouée par différentes valeurs de QP	78
Figure IV.20 : Séquence « container » tatouée par différentes valeurs de QP	79

Figure V.1 : Marques extraites de la vidéo « claire » : (a) méthode basée sur la DCT 4×4 et (b) méthode basée sur quantification	83
Figure V.2 : Marques extraites de la vidéo « container » : (a) méthode basée sur la DCT 4×4 et (b) méthode basée sur la quantification.....	83
Figure V.3 : Qualité visuelle de la séquence « claire » attaquée par le codeur H.264	84
Figure V.4 : Qualité visuelle de la séquence « container » attaquée par le codeur H.264	85
Figure V.5 : Marques extraites des vidéos basée sur la DCT 4×4 et attaquée par le codeur H.263 : (a) de la séquence « claire » et (b) de la séquence « container »	86
Figure V.6 : Qualité visuelle de la séquence « claire » attaquée par le codeur H.263.....	86
Figure V.7 : Qualité visuelle de la séquence « container » attaquée par le codeur H.263	87
Figure V.8 : Marque extraite après l'attaque par le codeur H.263 : (a) pour « claire » et (b) pour « container »	87
Figure V.9 : Qualité visuelle après l'attaque par H.263 de la séquence « claire » tatouée par la deuxième méthode	88
Figure V.10 : Qualité visuelle après l'attaque par H.263 de la séquence « container » tatouée par la deuxième méthode	88
Figure V.11 : Marques extraites de la vidéo « claire » après le transcodage: (a) basée sur la méthode DCT 4×4 et (b) basée sur la méthode de quantification	89

Figure V.12 : Marques extraites de la vidéo « container » après le transcodage :	
(a) basée sur la méthode de DCT 4×4 et (b) basée sur la	
méthode de quantification.....	89
Figure V.13 : Qualité visuelle des vidéos « claire » face à l'attaque de transcodage ...	90
Figure V.14 : Qualité visuelle des vidéos « container » face à	
l'attaque de transcodage	91
Figure V.15 : Marque extraite après l'attaque de redimensionnement de	
la vidéo tatouée par la première méthode	92
Figure V.16 : Marque extraite après l'attaque de redimensionnement de	
la vidéo tatouée par la deuxième méthode	92
Figure V.17 : Qualité visuelle des vidéos « claire » face à l'attaque de	
redimensionnement.....	93
Figure V.18 : Qualité visuelle des vidéos « contenair » face à l'attaque de	
redimensionnement.....	94

LISTE DE TABLES

Tableau I.1 : Formats de trame vidéo	9
Tableau II.1 : Modes de prédiction des blocs 4x4 de luminance	28
Tableau II.2 : Modes de prédiction des blocs 16x16 de luminance	29
Tableau II.3 : Modes de prédiction des blocs 8x8 de chrominance	30
Tableau II.4 : Taille du pas de quantification dans le codec H.264	34
Tableau II.5 : Mots de codes d'Exp-Golo mb	37
Tableau II.6 : Type de codage du paramètre k	37
Tableau IV.1 : Paramètres de la norme de compression vidéo H.264/AVC utilisés pour le tatouage	56
Tableau IV.2 : Qualité et taux d'extraction pour différentes positions d'insertion	65
Tableau IV.3 : Résultats du choix du $n^{\text{ième}}$ bit MSB	69
Tableau IV.4 : Les QP utilisés pour la quantification de la marque.....	71
Tableau IV.5 : Résultats d'insertion suivant le parcours en zigzag	80
Tableau IV.6 : Table comparative des algorithmes développés.....	81
Tableau V.1 : Résultat de tatouage de la séquence « claire » par les deux algorithmes	95
Tableau V.2 : Résultat de tatouage de la séquence « container » par les deux algorithmes	95



Introduction générale

Introduction Générale

Avec l'apparition et le développement des nouvelles technologies numériques, les fraudes sont multipliées, soulignant le manque de méthodes concernant la protection des données multimédias (images audio, vidéo). Ces données sont en effet très faciles à pirater : on peut les stocker, les modifier et en fin les diffuser illégalement sans qu'elles perdent de leur qualité. Pour répondre à ces besoins, un nouvel axe de recherche se développe très rapidement : le tatouage ou watermarking. Le principe des techniques dites de tatouage est d'insérer une marque imperceptible dans les valeurs des données. Dans le cadre de la protection des droits d'auteurs, la marque insérée correspond au code du copyright. Ce type de tatouage doit répondre à des contraintes fortes en termes de robustesse. En effet, quelles que soient les transformations que la donnée tatouée subit, la marque doit rester présente tant que la donnée reste exploitable. De plus, la présence de la marque ne doit être détectée que par des personnes autorisées (possédant une clé de détection privée).

Bien que l'effort considérable de recherche dans le tatouage a été consacré aux données image et audio, le tatouage vidéo a également eu plus d'importance ces dernières années. Etant donné qu'un flux vidéo est composé d'une succession d'images, le tatouage vidéo peut donc être vu comme une extension du tatouage appliqué à une série d'images fixes. En effet, il est possible de tatouer uniquement les images intra de la séquence. De nombreux schémas développés pour les images fixes peuvent alors être appliqués aux séquences vidéo. Cependant les séquences vidéo possèdent d'autres propriétés qui peuvent être exploitées pour l'insertion de la marque telles que : la taille brute d'une séquence vidéo est beaucoup plus importante (l'espace d'insertion de la marque en est considérablement augmenté), la dimension temporelle du signal traité peut être utilisée pour l'insertion de la marque.

Les séquence vidéo présentent aussi des contraintes différentes de celles des images fixes : la complexité du schéma de tatouage doit être faible, le mouvement des objets augmente souvent la visibilité de la marque, la présence de la marque dans une séquence vidéo peut permettre d'autres attaques que celles liées aux images fixes.

Dans la plupart des applications telles que la protection de copyright (tatouage robuste), l'authentification vidéo (tatouage fragile) et les applications de traçage du contenu (empreinte digital), l'insertion et la détection de la marque sont nécessaires. Enfin, le flux vidéo est souvent compressé afin de réduire la taille brute des séquences. Les standards de compression utilisés sont Mpeg-1, Mpeg-2, H.263 et H.264. Par conséquent, la protection du contenu dans ces normes est indispensable.

La problématique que nous posons dans ce mémoire est de proposer une solution de protection de copyright dans la plus récente et future norme de compression vidéo H.264/AVC.

Le tatouage utilisé est un schéma aveugle, l'extraction de la marque ne nécessite pas la vidéo d'origine, et l'insertion est effectuée au cours de la compression.

Ce mémoire est composé des chapitres suivants :

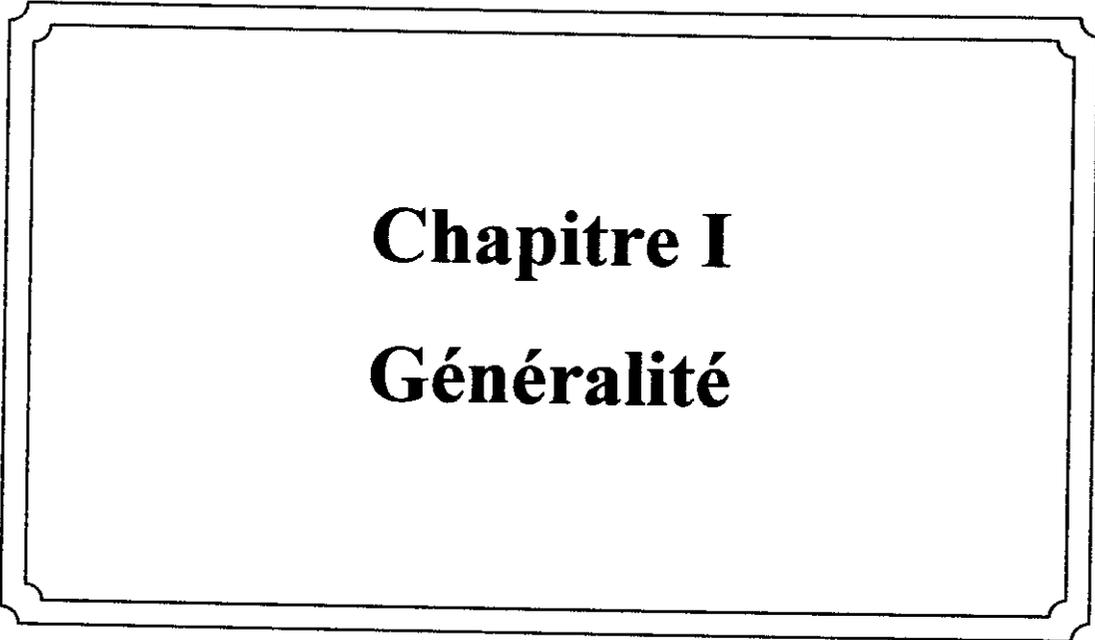
Chapitre 1 : nous présentons les définitions et quelques concepts de base concernant respectivement les domaines de tatouage et de compression vidéo.

Le deuxième chapitre est une étude détaillée du codeur vidéo H.264/AVC, en présentant les différents modules le constituant, les caractéristiques aussi ses performances par rapport aux précédents codeurs.

Le troisième chapitre est consacré à quelques méthodes de tatouage vidéo tout en évoquant leurs avantages et inconvénient.

Dans le quatrième chapitre, nous présentons de manière détaillée, les algorithmes de tatouage que nous avons élaboré au cours de ce mémoire afin de pouvoir concevoir la protection de copyright dans le standard H.264/AVC. Nous exposons en premier lieu le principe fondamental des deux méthodes, les tests et le choix des blocs et les coefficients d'insertion afin de trouver un bon compromis entre robustesse et invisibilité.

Le cinquième chapitre est destiné aux tests de performances des algorithmes face à différentes attaques bienveillantes à savoir, attaques géométriques, compression et transcodage.



Chapitre I
Généralité

1. Introduction

Notre ère numérique a ouvert une grande capacité de stockage et une facilité de circulation de données, mais ceci a engendré une facilité de piratage. En effet, des données peuvent être dupliquées de nombreuses fois sans subir de détérioration et le développement des réseaux rend très difficile la poursuite des fraudeurs. Une image numérique diffusée par exemple sur Internet peut aisément être récupérée par quiconque puis rediffusée soit sur un réseau, soit sur un CD-ROM. Une question se pose alors : Comment protéger les documents numériques contre le piratage et comment protéger leurs ayant droit ?

Les techniques cryptographiques classiques telle que le chiffrement peuvent s'appliquer à la protection des droits d'auteurs. Cette technique permet de protéger le document pendant sa transmission. Mais, une fois le document décrypté, il ne présente alors plus aucune protection. La stéganographie, « l'art de cacher un message dans un autre », est une autre solution de protection. Contrairement à la cryptographie, cette technique est invisible et permet de protéger le document même lorsque celui-ci est diffusé. Sa faiblesse réside dans le manque de robustesse. En effet, il est facile d'effacer le message inséré par changement systématique du document. De ce fait, la nécessité de recourir à des procédés plus performants de protection du copyright devient un besoin primordial. D'où l'apparition récente de la notion de tatouage. En quoi consiste donc ce nouveau concept ?

2. Notions de base du tatouage

2.1 Définition

Le tatouage (en anglais watermarking) est une technique de marquage qui consiste à insérer une marque invisible et permanente à l'intérieur d'un fichier multimédia : image, son et vidéo afin de lutter contre la fraude et le piratage et d'assurer la protection des droit de propriété intellectuelle. La marque peut permettre d'identifier soit le propriétaire du document, soit son ayant droit, soit les permissions attachées. Elle doit être non seulement indélébile, mais également robuste [1]. La marque (signature) est l'information que l'on cache dans le cadre du tatouage, elle peut être :

- Un numéro d'identification pour mettre en œuvre un système de copyright.
- Une description du contenu permettant son indexation.
- Une information « fragile » qui permettra son authentification.

Il existe deux principes contraints en tatouage :

- La marque à insérer ne doit pas surcharger l'image et donc doit être invisible.
- La marque doit être indélébile : une fois insérée, il doit être difficile de l'enlever.

2.2 Caractéristiques de la marque

Que ce soit pour l'image fixe, pour l'audio ou pour la vidéo, le tatouage fait appel aux principes de base suivants :

- Robustesse : c'est la capacité que possède un algorithme de tatouage à résister aux attaques extérieures, qu'elles soient bienveillantes ou malveillantes. Pour la vidéo, il peut s'agir d'attaques simples comme le changement de format de compression, le changement de débit ou tout autre traitement classique (il s'agit ici de traitements bienveillants qui ne visent pas forcément à retirer la marque). On peut aussi avoir des attaques plus élaborées, qui ont pour seul but de retirer la marque, comme des attaques statistiques ou des attaques basées sur la connaissance de l'algorithme utilisé.
- Capacité : c'est la quantité d'information que l'on peut cacher au sein du médium. Il paraît évident que plus on augmente la capacité, plus la marque sera perceptible, et plus la robustesse diminuera (dans le cas où on veut retrouver exactement la marque).
- Invisibilité : c'est l'impact que peut avoir la marque sur le médium. Plus le marquage sera fort, plus elle sera visible (et inversement) [2].

Concevoir un algorithme de tatouage revient à trouver le meilleur compromis entre ces trois principes, en fonction de l'application visée. La figure I.1 présente une bonne illustration de ces caractéristiques.

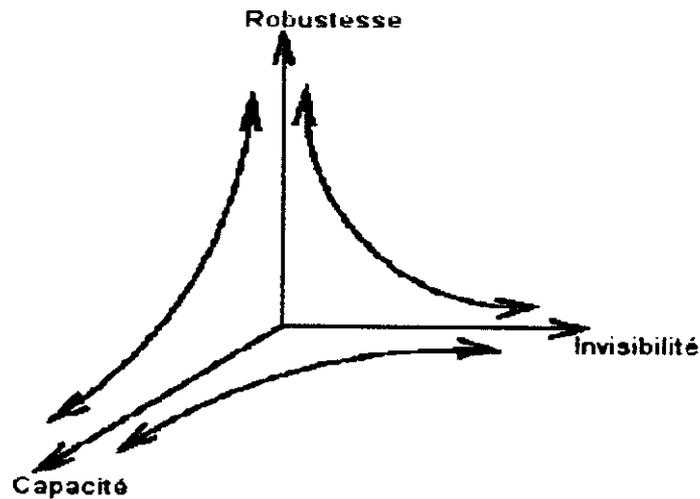


Figure I.1 Caractéristiques de tatouage

2.3 Différents types de tatouage

Différents types de tatouage existent [3] :

- Tatouage privé : le média d'origine est nécessaire à l'étape d'extraction.
- Tatouage semi-privé : Dans ce cas, on n'utilise pas le média d'origine, on se sert uniquement de la marque et d'une clé.
- Tatouage public (on parle aussi de tatouage aveugle) : Ils n'utilisent ni le média d'origine ni la marque. Seule la clé secrète est nécessaire pour extraire la marque.
- Tatouage asymétrique : l'extraction de la marque ne nécessite pas la connaissance d'un secret. Ceci implique que tout utilisateur est capable de lire la (ou les) marque(s) du médium sans pouvoir les effacer. Ceci pourrait être réalisé par un marquage sans clé ou par un tatouage avec clé secrète et par une extraction en utilisant la clé publique correspondante.

2.4 Attaques

Les attaques sont, le plus souvent, des traitements classiques qu'une personne effectue sur le support qu'elle utilise. Elles peuvent être des traitements visant soit à brouiller soit à enlever la marque de protection dans la vidéo.

On peut distinguer deux grandes familles d'attaques, les bienveillantes et les malveillantes [4] :

- **Attaques bienveillantes** : Il s'agit de traitement qui n'a pas initialement pour objectif d'empêcher la détection de la marque. Il peut s'agir des dégradations dues à une compression, à un changement de type de compression, à des filtrages (réduction de bruit), à un changement de résolution, etc. Un autre traitement couramment utilisé en vidéo est la conversion analogique/numérique, et inversement. Enfin, certaines distorsions géométriques peuvent être utilisées : flip vertical, perte d'une ligne ou d'une colonne, etc.
- **Attaques malveillantes** : vise explicitement à rendre le tatouage inopérant. Ces attaques, comme souvent dans le domaine numérique, sont difficile à prouver d'un point de vue juridique. Toutefois, une attaque malveillante qui a réussi devra produire un contenu à la fois lavé de son tatouage et encore exploitable.

Nous allons maintenant citer quelques différentes attaques :

1. **L'attaque par surmarquage** : consiste à tatouer à nouveau un média déjà tatoué. Pour certains schémas, en particulier si les lieux de tatouages sont fixés, cette attaque peut être très dangereuse. Certains algorithmes de tatouage se protègent en vérifiant, avant de distribuer une clef que le média d'origine proposé n'est pas tatoué. Cette protection n'est utile que si le schéma de tatouage demeure inconnu. En effet, s'il est connu, un pirate peut ajouter une marque de sa fabrication qui invalidera la détection.
2. **L'ajout d'un bruit** : le bruit additif et le bruit multiplicatif non corrélatif ont été en grande partie adressés dans la littérature de théorie de traitement des signaux et de théorie de communication.
3. **Attaque par cropping** : elle consiste à extraire un morceau non tatoué d'un flux média pour le réutiliser. Pour être résistant à ce type d'attaque, le tatouage doit être présent sur tout le média. La même situation se produit dans le domaine fréquentiel du média où la marque doit être partout présente afin d'éviter une destruction par filtrage.

4. **Attaque par collusion** : L'attaquant possède plusieurs copies du même contenu avec quelques différences provenant de l'individualisation des marques. Il les combine ensuite pour obtenir des documents qui ne contiennent plus aucun signal de tatouage. Cette attaque peut être utilisée pour perturber les marques individualisées de type fingerprint qui permettent d'identifier le client.
5. **L'attaque par recopie** : consiste à recopier une marque obtenue préalablement (par exemple par estimation) sur un média non marqué. Le détecteur validera alors le nouveau média comme étant tatoué.
6. **Moyennage** : Lorsqu'on dispose d'un nombre élevé d'images de contenu identique, mais avec des marques différentes (par exemple des numéros de série ou des identifications d'utilisateur), le pirate peut les moyenner, afin de produire une image sans marque détectable. De même, dans certains cas, le pirate peut extraire la marque, cela peut être réalisé aisément dans les applications vidéo, où la même marque est ajoutée à un ensemble d'images successives. En effet, lorsque l'on insère une marque identique sur l'ensemble des images, et si on les additionne ensuite, on sait que l'espérance de la valeur nous permet de déterminer la marque. Afin de résoudre ce problème, il suffit d'insérer une autre marque dépendante de la première (cependant cela peut engendrer des artefacts visibles).

Vue que notre travail consiste à réaliser un système de protection de copyright vidéo, il est nécessaire de donner quelques notions de bases du signal vidéo et sa compression.

3. Notion de base du signal vidéo et sa compression

Une séquence vidéo est une suite d'images fixes, qui peut être caractérisée par trois paramètres : sa résolution en luminance, sa résolution spatiale et sa résolution temporelle. La résolution en luminance détermine le nombre de couleurs possibles pour un pixel. Celle-ci est généralement de 8 bits pour les niveaux de gris et de 24 bits pour les séquences en couleurs. La résolution spatiale définit le nombre de lignes et de colonnes de la matrice de pixels. Enfin, la résolution temporelle est le nombre d'images par seconde. La valeur de ces trois paramètres détermine l'espace mémoire nécessaire pour stocker chaque image de la séquence. Cet espace mémoire est caractérisé par le

débit, qui est le coût de stockage pour une seconde (capacité mémoire nécessaire pour stocker une seconde de vidéo).

Une image animée est une suite d'images décrivant un mouvement. Le nombre d'images par seconde doit être suffisant pour donner à l'œil une sensation de fluidité. Le taux idéal est de 25 images par seconde. A cette fréquence, l'œil perçoit le mouvement de façon claire [5].

3.1 Echantillonnage

Les séquences vidéo 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 macrobloc Y est un tableau de 16×16 valeurs, et les macroblocs U et V correspondants sont chacun des tableaux de 8×8 valeurs.
- **4 :2 :2** : Les tableaux des macroblocs 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.

Une trame est l'échantillonnage d'un signal en un point du temps.

3.2 Formats vidéo

Il existe de nombreux formats de trame vidéo. En pratique, on utilise des formats intermédiaires tels que le CIF (Common Intermediate Format) et ses dérivés. Le format 4CIF est le standard de définition pour la télévision et le DVD (Digital Video Disc). CIF et QCIF sont utilisés en vidéoconférence et les formats QCIF et SQCIF pour les applications multimédia mobiles. Le tableau I.1 présente la taille, exprimée en bits, nécessaire à la représentation d'une trame (non compressée) pour chaque format. On considère le modèle 4:2:0 pour des échantillons de luminance et de chrominance de 8 bits.

Tableau I.1 Formats de trame vidéo

Format	Résolution de la luminance	Bits par trames (4 :2 :0)
Sub-QCIF	128×96	147456
Quarter-CIF	176×144	304128
CIF	352×288	1216512
4CIF	704×576	4866048

3.3 Compression vidéo

La compression est le processus permettant de compacter des données en un nombre réduit de bits. Une vidéo numérique non compressée nécessite un débit important. Il est donc impératif de la compresser pour pouvoir la transmettre ou la stocker.

La compression utilise deux systèmes complémentaires, un codeur et un décodeur. Le codeur convertit la source de données en une forme compressée occupant un nombre de bits inférieur, avant transmission ou stockage. Le décodeur convertit la forme compressée en une représentation de la source d'origine. La paire codeur/décodeur est appelée codec.

Un codeur comprend trois fonctionnalités principales : un modèle temporel, un modèle spatial et un codeur entropique (Figure I.2). L'entrée du système est une séquence vidéo

non compressée. Le modèle temporel réduit la redondance temporelle en exploitant les similarités entre les trames autour d'un même instant. Il construit ainsi une prédiction. Celle-ci est faite à partir d'une ou plusieurs trames futures ou passées et est améliorée en compensant les différences entre les trames (prédiction par compensation de mouvement). La sortie du modèle temporel est constituée d'une trame résiduelle (la trame courante moins la prédiction) et d'un jeu de paramètres tels que des vecteurs de mouvement décrivant la compensation.

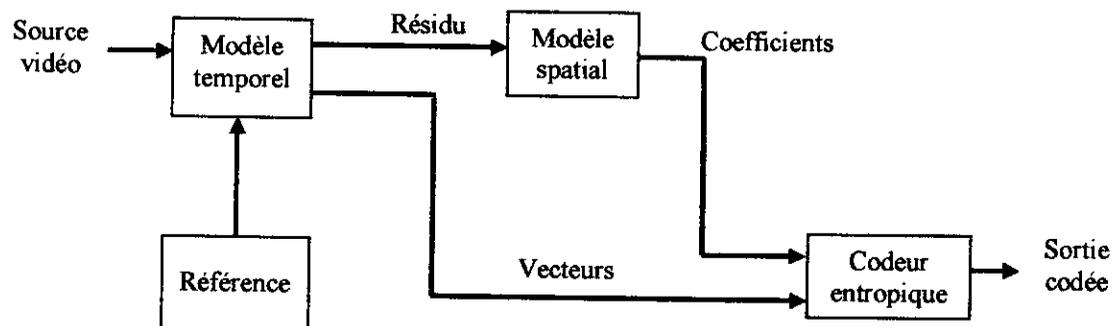


Figure I.2 Bloc diagramme d'un codeur vidéo

La trame résiduelle devient l'entrée du modèle spatial qui utilise les similarités entre les pixels d'un même voisinage pour réduire la redondance spatiale. Ceci est généralement effectué par transformation du résidu et quantification du résultat.

La transformation convertit l'image résiduelle dans un autre domaine dans lequel elle est représentée par des coefficients de transformation. Ceux-ci sont ensuite quantifiés afin d'éliminer les valeurs non significatives, laissant un nombre réduit de coefficients significatifs. On obtient ainsi une représentation plus compacte de la trame résiduelle.

La sortie du modèle spatial est un ensemble de coefficients de transformation quantifiés. Les paramètres du modèle temporel (typiquement des vecteurs de mouvement) et du modèle spatial (coefficients) sont compressés par le codeur entropique. Celui-ci retire la redondance statistique des données et produit un flux binaire compressé qui peut être transmis ou stocké. Ainsi, une séquence compressée consiste en des paramètres de vecteur de mouvement, des coefficients résiduels et un en-tête d'information. Le

décodeur vidéo reconstruit la séquence à partir du flux compressé. Les coefficients et les vecteurs de mouvement sont décodés par un décodeur entropique, les trames résiduelles sont reconstruites par le modèle spatial. Il ne reste plus qu'à recréer les prédictions à qui l'on ajoute les trames résiduelles [6].

4. Applications du tatouage vidéo

Le tatouage apparaît comme une nécessité pour la protection de contenu multimédia, que ce soit pour l'image, pour la musique ou enfin pour la vidéo. Jusqu'à présent, les systèmes dits propriétaires, ont montré leurs faiblesses. En effet, ce genre de systèmes repose sur un secret qui, une fois dévoilé, permet de passer outre la protection. Les applications classiques d'un système de tatouage vidéo sont les suivantes :

- **Protection de copyright** : le tatouage offre une alternative intéressante à la cryptographie, car il permet de protéger l'image, même lorsque celle-ci est diffusée. La protection des droits d'auteur représente l'application la plus courante aujourd'hui. L'objectif est d'insérer une information dans la donnée source afin de prévenir toute revendication frauduleuse de propriété. Cette marque ne doit être connue que de la personne ou de l'organisme de tatouage. Elle dépend donc d'une clé secrète, qui permet son insertion et sa détection. Cette application nécessite la mise en place d'un algorithme de tatouage d'un niveau de robustesse très élevé. En effet, celui-ci ne doit pas être ambigu et doit toujours déterminer l'appartenance du médium, même si d'autres parties insèrent également une marque.
- **Fingerprinting** : cette application est utilisée pour tracer les copies illégales de media (suivi des pirates). Ce type d'application engendre un marquage unique pour chaque copie distribuée (typiquement un numéro de série). Cependant, la distribution de copies composées de différentes marques, peut engendrer des problèmes de collusions. Ainsi, les marques utilisées devront satisfaire un critère de sécurisation de collusions.
- **Authentification de données** : l'objectif est de détecter toutes modifications éventuelles des données, afin de pouvoir certifier si celles-ci ont été modifiées ou

non. On aperçoit ici une problématique de contrôle d'identité de documents. Ce qui peut être obtenu avec un tatouage fragile. Le marquage pour l'authentification est donc celui qui utilise le niveau le plus faible de robustesse. Cependant, il faut noter que de nouvelles approches émergent, pour lesquelles les attributs des données sont marqués et contrôlés par le receveur, qui vérifie si les données contiennent toujours ces attributs. Ces dernières approches demandent donc un niveau de robustesse plus élevé.

- **Protection contre la copie** : un souhait des distributeurs de multimédia est l'existence d'un moyen de protection contre la copie, afin d'interdire une circulation de media illégaux. Cependant, cela est difficile à obtenir pour les systèmes ouverts, mais réalisables pour les systèmes fermés ou propriétaires. Dans ces derniers, il est possible d'utiliser des marques spécifiant le statut de la copie de la donnée.
- **Sécurité médicale** : insertion d'un « identifiant » confidentiel assurant la correspondance entre le patient et la radio, afin d'éviter toutes confusions.
- **Indexation** : le domaine de l'indexation des images consiste à classer de manière automatique des images selon leur contenu. Il permet de faciliter une recherche dans une base de données. Les techniques classiques utilisées consistent à effectuer un traitement automatique de l'image, de manière à dégager les composantes essentielles du contenu. Le tatouage d'un document permet d'insérer une information décrivant le contenu de l'image. Cela permet de qualifier sommairement l'image, ou d'insérer un pointeur vers une description plus complète.

5. Conclusion

Dans ce premier chapitre nous avons abordé les différentes définitions jugées nécessaire pour se familiariser avec les notions de tatouage vidéo appliqués à la récente norme de compression vidéo H.264/AVC qui fera l'objet du chapitre suivant.



Chapitre II
La norme de compression
vidéo H.264/AVC

1. Introduction

Les systèmes de codage vidéo peuvent avoir de multiples applications. Cependant, il est inconcevable de mettre au point un système unique capable de s'adapter à toutes les situations de débit, de qualité et/ou de coût. C'est pourquoi plusieurs voies (et donc plusieurs standards) sont explorées suivant le type d'application que doit satisfaire chaque système.

L'objectif d'un système de compression vidéo est de réduire la quantité d'information à transmettre ou à stocker, tout en préservant le contenu visuel du signal d'origine.

Certaines applications, par exemple les lecteurs DVD, la télévision numérique, la télévision par satellite, la transmission vidéo par l'Internet, la vidéo conférence, la sécurité vidéo et les caméras numériques, sont limitées par la capacité mémoire ou de bande passante pour la transmission, ce qui stimule la demande pour des rapports de compression vidéo plus élevés. Pour tenter de résoudre ce problème, plusieurs normes de codage vidéo ont été adoptées durant la dernière décennie. Mais la plus récente des systèmes de compression vidéo s'appelle « H.264/AVC » et promet des performances bien supérieures à toutes les précédentes.

2. Historique de la norme de compression vidéo H.264/AVC

Les deux organismes les plus actifs pour la normalisation des systèmes de compression vidéo sont l'ISO/IOC et l'UIT-T. Les travaux techniques de l'ISO/IEC sont menés au sein du groupe MPEG (Motion Picture Experts Group) qui a défini les standards MPEG-1, MPEG-2 et MPEG-4 pour des applications aussi variées que la télévision ou le multimédia. En parallèle des activités de MPEG, le groupe de l'UIT-T VCEG (Video Coding Experts Group) s'intéresse principalement à la définition de recommandations techniques destinées aux applications de visiophonie et de visioconférence (normes H.261 et H.263). Actuellement, ces deux organismes travaillent à l'élaboration d'une norme commune (H.264 / MPEG-4 « AVC ») depuis la date de commencement dont les performances attendues devraient permettre de réduire de moitié le débit de transmission ou de stockage pour une qualité visuelle équivalente aux normes précédentes. Ces normes sont :

- **H.261** est la première norme vidéo approuvée en 1993 destinée aux applications de visiophonie. Les formats d'image traités par la norme H.261 sont le CIF (Common Intermediate Format) de résolution 288×352 pixels et le QCIF (Quarter CIF) de résolution 144×176 pixels [7].
- **H.263** est une norme de codage vidéo destinée à la communication vidéo à très bas débit dont la première version fut adoptée en 1995. Cette norme repose sur les principes mis en place par la recommandation H.261. Les formats des images sont des multiples et sous-multiples du CIF. La deuxième version de la recommandation H.263 (1998), souvent appelée H.263+, met en oeuvre des options supplémentaires et permet de définir des formats d'image personnalisés.
- **MPEG-1** est la première norme sortie par l'ISO/IOC, et avait pour but le codage et la compression de séquences vidéo (son et images) à des débits réduits de l'ordre de 1.5Mbps [8]. Cette norme a été construite sur la base de H.261 dont elle reprend les principes en les améliorant. De nos jours cette norme n'est plus trop utilisée mis a part pour la compression des MP3.
- **MPEG-2** est le pilier technique des systèmes de télévision numérique du monde entier. Elle reprend les principes de MPEG-1 en ajoutant les outils indispensables pour les applications. Ce standard a été défini pour des applications de stockage et de transmission vidéo. Les applications les plus importantes sont la télévision numérique (satellite, câble, terrestre...) et le stockage sur DVD.
- **MPEG-4** est une norme générique de compression destinée à la manipulation d'objets multimédia, elle utilise aussi bien les fonctions du MPEG-2 que de l'H.263. Sa grande évolution est qu'elle peut séparer et coder chaque élément de l'image qui devient "objets". La norme MPEG-4 est beaucoup utilisée sur les logiciels de montages vidéo. Grâce à la technique de codage on obtient une image exceptionnelle pour un espace en mémoire de la vidéo très réduit [9].

Les deux groupes de standardisation MPEG et VCEG se mettent en commun pour créer une nouvelle norme de codage. Les deux groupes fusionnent sous le nom JVT (Joint Vidéo Team) le 6 décembre 2001. Le nouveau standard porte le nom H.26L et les deux groupes travaillent ensemble sur ce projet de nouveau standard. Ce dernier est fondé sur une architecture identique aux précédentes normes de codage.

Le nom H.264 provient de la famille de normes vidéo H.26L définies par l'UIT-T. Dans le cadre de MPEG, le nom AVC (Advanced Video Coding) a été choisi par analogie avec le codec audio AAC (Advanced Audio Coding) MPEG-2 part 7 qui avait été nommé ainsi pour le différencier du codec audio MPEG-2 part 3.

La norme est habituellement appelée H.264/AVC (ou AVC/H.264 ou H.264/MPEG-4 AVC ou MPEG-4/H.264 AVC) pour souligner l'héritage commun. H.264 utilise toutes les dernières innovations de la technologie de compression vidéo pour fournir une qualité vidéo exceptionnelle. La première version de la norme a été approuvée en mai 2003 et la plus récente date de mars 2005.

3. Principales caractéristiques de la norme H.264/AVC

H.264/AVC comprend de nombreuses techniques nouvelles qui lui permettent de compresser beaucoup plus efficacement les vidéos que les normes précédentes (H.261, MPEG-1, MPEG-2, MPEG-4) et fournit plus de flexibilité aux applications dans un grand nombre d'environnements réseau. Dans ces fonctionnalités principales sont inclus [sw1] :

- Une compensation de mouvement pouvant être effectuée par rapport à plusieurs images de référence déjà codées. Le choix de l'image de référence intervient au niveau de macrobloc (un macrobloc est un bloc carré de 16×16 pixels) et sous-macrobloc. Ceci permet d'utiliser dans certains cas jusqu'à quatre (04) références différentes pour un même macrobloc.
- Une compensation de mouvement pouvant utiliser 07 tailles de blocs différentes (16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 et 4×4) permet une segmentation très précise de zones se déplaçant.

- Une précision au quart de pixel pour la compensation de mouvement, permettant une description très précise du déplacement des zones en mouvement. Pour la chrominance, la précision de la compensation de mouvement se fait même au huitième de pixel.
- Un filtrage dit de "deblocking", effectué dans la boucle de codage et opéré sur les blocs 4×4, permettant de réduire les effets de blocs (artefacts). Ceci est dû aux caractéristiques du codage avec transformation en blocs.
- Une transformation entière effectuée sur des blocs de taille 4×4 pixels. Pour les nouveaux profils issus des extensions FExt (Fidelity Range Extensions) développé par le JVT, une transformation supplémentaire de taille 8×8 a été ajoutée.
- Une transformation de Hadamard effectuée sur les coefficients moyens de la transformation spatiale pour la luminance et la chrominance.
- Une prédiction spatiale sur le bord des blocs voisins pour un codage « intra ».
- Un codage arithmétique CABAC (Context Adaptive Binary Arithmetic Coding), qui est une technique de codage entropique qui produit d'excellents résultats en terme de compression mais possède une grande complexité.
- Un codage adaptatif de type Huffman à longueur variable CAVLC (Context Adaptive Variable Length Coding), qui est une alternative moins complexe que CABAC pour le codage des tables de coefficients de transformation. Bien que moins complexe que CABAC, CAVLC est plus élaboré et plus efficace que les méthodes habituellement utilisées jusqu'à présent pour coder les coefficients.
- Un codage à longueur variable VLC (Variable Length Coding) pour de nombreux éléments de syntaxe, considéré comme du code Golomb exponentiel (Exp-Golomb).
- Une couche d'abstraction réseau NAL (Network abstraction layer) est définie pour permettre l'usage de la même syntaxe vidéo dans de nombreux environnements réseau.

- Les tranches de commutation (appelées SP et SI) permettent à un codeur de diriger un décodeur pour que ce dernier puisse s'insérer dans un flux vidéo entrant. Quand un décodeur saute au milieu des images d'un flux vidéo en utilisant cette technique, il peut se synchroniser avec les images présentes à cet endroit malgré l'utilisation d'autres images (ou pas d'images) comme références préalables au déplacement.
- L'ordonnancement flexible des macroblocs (FMO : Flexible Macroblock Ordering) et l'ordonnancement arbitraire des tranches (ASO : Arbitrary Slice Ordering) sont des techniques de restructuration de l'ordonnancement des régions fondamentales de l'image (macroblocs). Typiquement utilisés pour améliorer la résistance aux erreurs et aux pertes.

Ces techniques, ainsi que plusieurs autres, aident H.264 à dépasser significativement les standards précédents, dans une grande variété de circonstances et dans une grande variété d'environnements d'application. H.264 peut fonctionner souvent nettement mieux que la vidéo MPEG-2 en obtenant la même qualité avec un débit binaire diminué de moitié.

4. Fonctionnement du codeur H.264/AVC

Le codeur inclut deux chemins de données : le chemin avant (de gauche à droite) et le chemin de reconstruction (de droite à gauche). C'est pourquoi le décodeur est présenté de droite à gauche, le décodeur étant ainsi inclus dans le codeur.

Une image F_n est tout d'abord découpée en macroblocs (Figure II.1). Chacun est codé par le mode intra ou inter, et pour chaque bloc de macrobloc, une prédiction P est formée. En mode intra, P est obtenue à partir des échantillons de l'image courante ayant déjà été codés, décodés et reconstruits (uF'_n sur la figure, notons que P est formée avec des échantillons non filtrés). En mode inter, P est obtenue par estimation de mouvement (ME), prédiction et compensation de mouvement (MC) à partir d'une ou plusieurs images de référence. Sur les deux schémas (codeur, décodeur), l'image de référence est F'_{n-1} . Les images de référence peuvent aussi bien faire parti du passé que du futur de l'image courante.

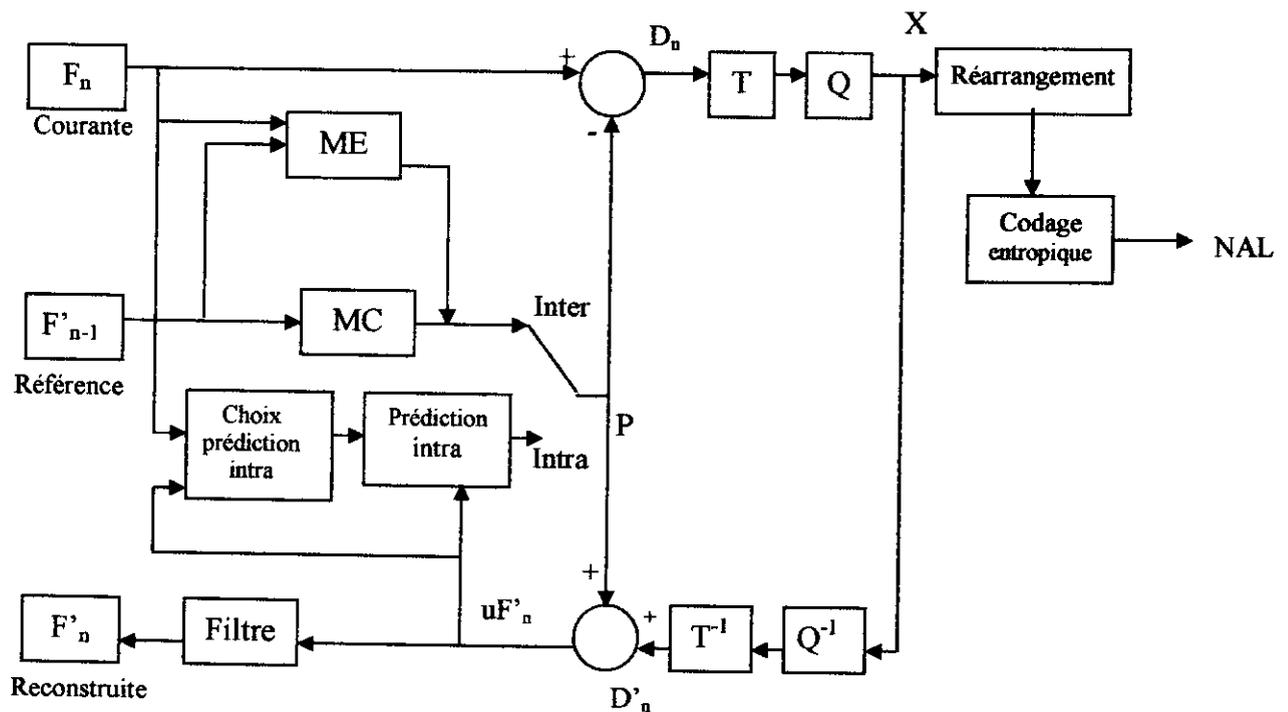


Figure II.1 Bloc diagramme du codeur H264/AVC

La prédiction P est soustraite au bloc courant. Ce qui donne un bloc résiduel D_n , transformé (T) et quantifié (Q) pour former X , un ensemble de coefficients de transformation quantifiés. Ceux-ci sont ensuite ordonnés et codés par codage entropique. Les coefficients obtenus de codage entropique et les informations nécessaires au décodage forment un bitstream compressé qui passe au NAL pour transmission ou stockage.

Le codeur effectue également un décodage local des macroblobs afin de créer les références nécessaires à la prédiction. Les prédictions X sont déquantifiés (Q^{-1}) et transformés par l'inverse (T^{-1}) pour produire un bloc différence D'_n . La prédiction P est ajoutée à D'_n . Cela permet d'obtenir le bloc reconstruit uF'_n non filtré.

Un filtre est enfin appliqué pour réduire les effets de bloc. L'image de référence reconstruite F'_n est créée et ajoutée à la liste correspondante [10].

Le même processus de codage est utilisé pour le décodage (Figure II.2). Les macroblobs fournis par le NAL sont décodés et réarrangés pour obtenir les coefficients

X à partir desquels sont obtenus les mêmes D'_n que le codeur. En utilisant les informations contenues dans le bitstream, le décodeur crée une prédiction P , l'ajoute à D'_n , et obtient uF'_n . Après filtrage, l'image décodée F'_n est retrouvée.

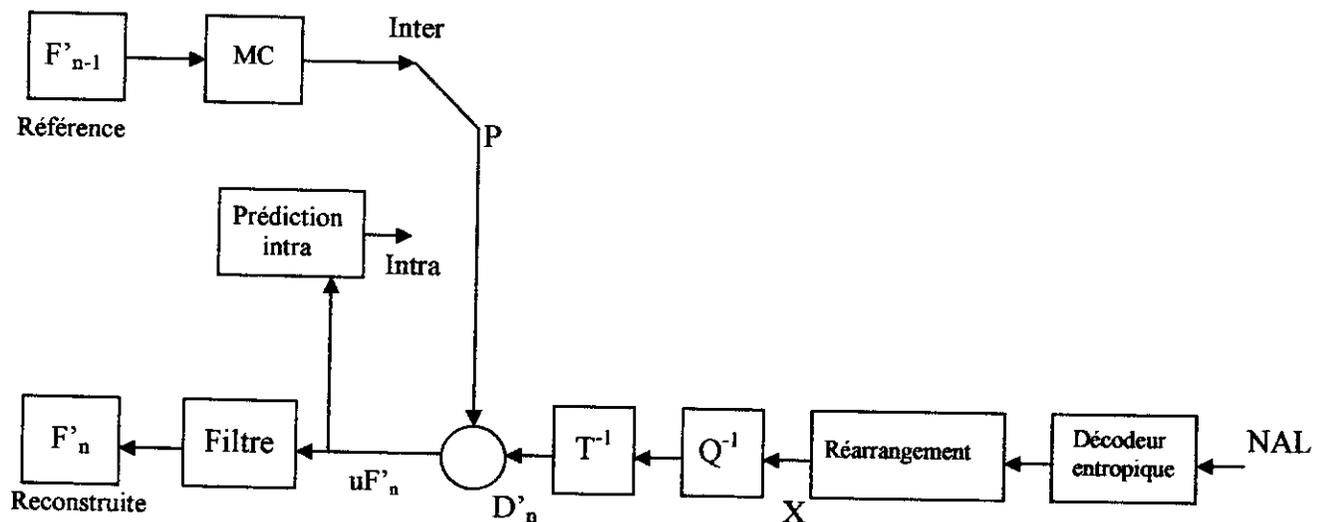


Figure II.2 Bloc diagramme du décodeur H264/AVC

4.1 Partitionnement en Groupe d'images et en Macroblocs

Une séquence vidéo est décomposée en groupes d'images GOP (Group Of Pictures) qui sont le point d'accès aléatoire dans une séquence vidéo, et chaque image de ce groupe est décomposée en bandes. La bande est un groupe de macroblocs organisés en 16 blocs dont chacun représente la plus petite entité de cette hiérarchie, il permet de réduire les redondances spatiales.

Chaque bande d'image est partitionnée en macroblocs de dimension fixe qui couvrent une zone rectangulaire de taille 16×16 pixels (Figure II.3). Les macroblocs sont organisés en tranches (Slices) qui représentent en général des sous-ensembles d'une image donnée pouvant être codés indépendamment.

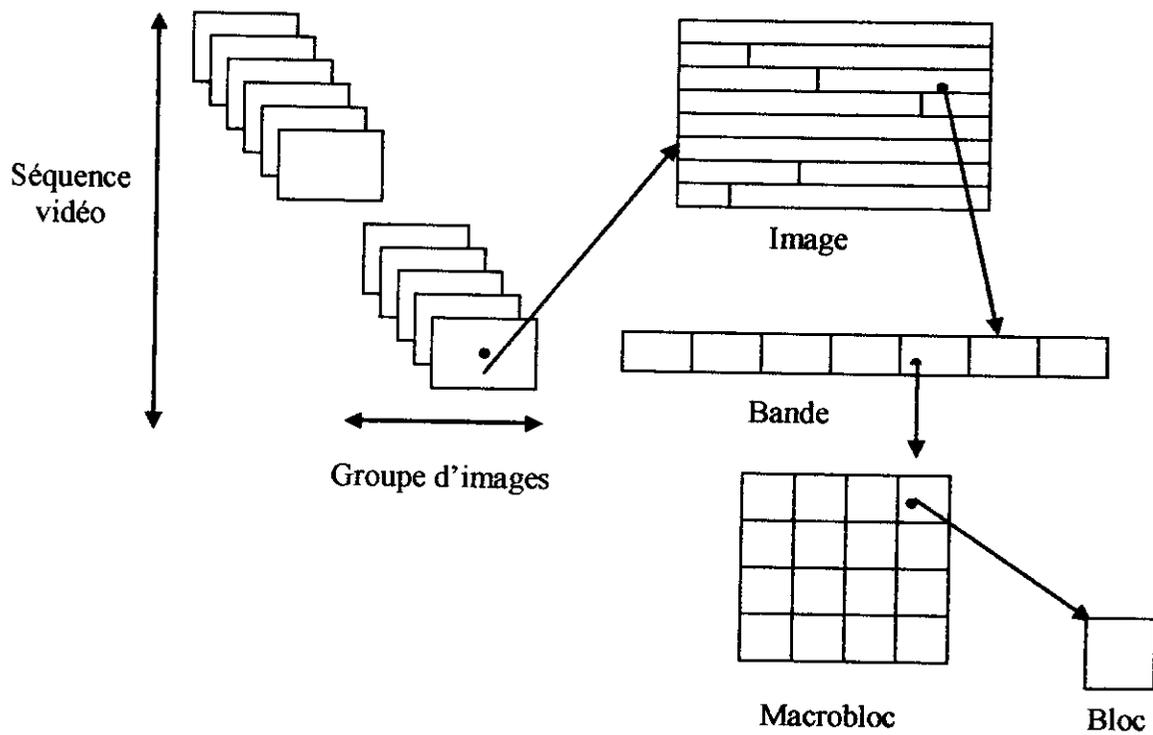


Figure II.3 Structure du GOP

Sept (07) tailles de division différentes (16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 , 4×4) sont utilisées dans la norme H.264/AVC (Figure II.4).

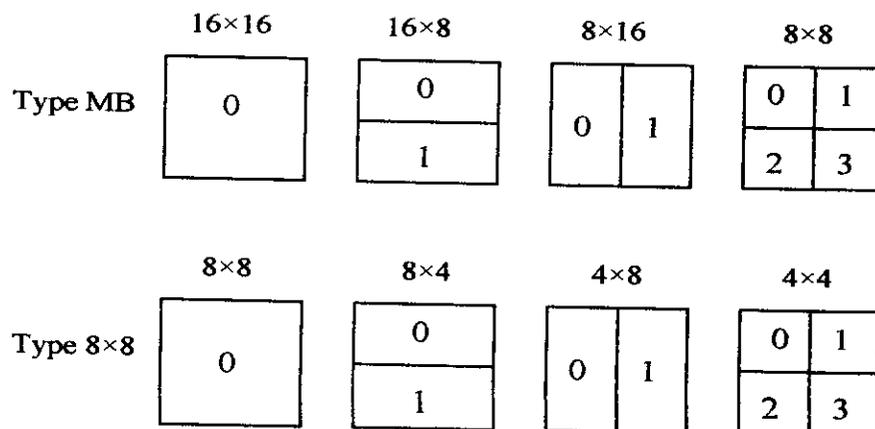


Figure II.4 Découpage d'un macrobloc pour la compensation de mouvement

4.2 Partitionnement en tranches (slices)

La norme H.264/AVC englobe deux couches, une couche de codage vidéo VCL (Video Coding Layer) qui représente de manière efficace le contenu des données vidéo, et une couche d'abstraction de réseau NAL (Network Abstraction Layer) qui est définie pour formater ces données et fournir des informations d'en-tête de manière adéquate qui seront acheminées par des couches transport ou des supports de stockage particuliers [11]. Toutes les données sont contenues dans des unités NAL, chacune contenant un nombre entier d'octets. Une unité NAL définit un format générique à utiliser avec les systèmes par paquet et les systèmes en flux. Chacun de ces paquets peut contenir une partie d'une tranche, c.-à-d., il peut y avoir une ou plusieurs unités de NAL par tranche. Les unités de NAL ne contiennent pas seulement des données de tranche, mais contiennent aussi des en-têtes. Le slice, qui est une suite de macrobloccs appartenants à la même image, constitue la plus petite unité qui est indépendamment décodable dans un flux vidéo. Chaque image dans la séquence vidéo est constituée d'un ou de plusieurs slices et chaque MB appartient à un seul slice. H.264 définit cinq types de slice : I, P, B, SI et SP. Ils sont classés selon les méthodes de prédiction utilisées, Le type le plus simple correspond à la tranche I.

- Les slices I ou les « intra » slices décrivent une pleine image immobile, contenant seulement des références à elle-même. Tous les macrobloccs sont codés sans référence à d'autres images de la séquence vidéo.
- Les slices P ou les slices « prédictifs » emploie une ou plusieurs tranches récemment codées en tant qu'une référence pour la construction d'image. Toutes les partitions de 16×16 jusqu'à 4×4 peuvent être utilisés dans ce type de tranche.
- Les slices B ou les slices « bidirectionnelles prédictifs » fonctionnent comme des tranches P à l'exception, ils nécessitent les anciens et les futurs tranches d'I ou de P comme images de référence. Deux listes de références sont tenues à jour par le processus de reconstruction. La première liste trie les éléments par rapport au numéro de trame courant ; d'abord elle sépare les trames antérieures des futures, place les antérieures devant les futures et place en première position la trame la plus proche. Les autres ne sont pas triées. Une opération similaire est faite sur la deuxième liste.

- **Tranches SP et SI** ou les tranches de « commutation » peuvent être employées pour des transitions entre deux flux différents de la vidéo H.264. Le but des tranches SP et SI est de faciliter l'accès aléatoire aux trames en phase de décompression d'une vidéo [12].

4.3 La prédiction

Le but de la prédiction est de réduire la redondance temporelle entre les trames transmises. Pour cela, elle construit une trame prédite et la soustrait à la trame courante pour obtenir une trame résiduelle. Deux types de prédiction dans la norme H.264/AVC sont prévus : la prédiction inter et la prédiction intra.

4.3.1 La prédiction inter

En mode inter, la prédiction inter crée un modèle prédictif à partir d'une ou plusieurs trames vidéo précédemment codées. H.264/AVC apporte quelques différences avec les précédents standards. Parmi les plus importantes, notons un plus large support de taille de blocs et un échantillonnage plus fin des vecteurs de mouvement, d'où une complexité calculatoire accrue. Cette prédiction est réalisée par compensation de mouvement des blocs. Des vecteurs de mouvement sont prédits et calculés suivant un certain découpage de l'image.

4.3.1.1 L'estimation de mouvement

L'estimation du mouvement est un problème incontournable dans le domaine de la compression vidéo. Le but est d'estimer le mouvement des objets contenus dans une vidéo sous forme d'un champ dense de vecteurs. Le principe général est d'exploiter les redondances temporelles existant entre des images consécutives. Pour simplifier l'estimation, le mouvement estimé est présenté entre deux images successives : l'image courante et l'image de référence [13].

Chaque image est subdivisée en blocs de taille égale et chaque bloc est considéré comme un objet indépendant. Et l'algorithme consiste, pour un bloc de l'image courante, à choisir un bloc dans l'image de référence (passée ou future) et à calculer un critère de comparaison entre ces deux blocs. Le bloc le plus semblable est identifié dans l'image

de référence pour chaque bloc de l'image courante. De cette manière, un vecteur déplacement est calculé pour chaque bloc caractérisé le mouvement de ce dernier.

4.3.1.2 La compensation de mouvement

Dans une trame de référence, on définit une zone de recherche dont la quelle le bloc à estimer est comparé à tous les blocs de même taille, le meilleur bloc retenu devient le prédicteur du bloc courant et lui est soustrait pour former un bloc résiduel qui est transmis avec le vecteur de mouvement entre les deux blocs (toutes les indications : mode de partition, vecteur de mouvement... doivent être transmises aux bitstream).

Un vecteur de mouvement séparé est exigé pour chaque partition. Chaque vecteur de mouvement doit être codé et transmis; en outre, le choix de partition doit être codé dans le bitstream. Le choix d'une grande taille de partition signifie qu'un nombre restreint de bits sont exigés de signaler le choix du vecteur de mouvement et du type de partition ; cependant, le résiduel de compensation de mouvement peut contenir une quantité significative d'énergie dans des régions de trame avec un détail élevé.

Le choix d'une petite taille de partition peut donner un résiduel à moins d'énergie après la compensation de mouvement mais exige un plus grand nombre de bits pour signaler les vecteurs de mouvement et les types de partition. Le choix de la taille de partition a un impact significatif sur la performance de la compression. En général, une grande taille de partition est appropriée pour des régions homogènes et une petite taille de partition peut être spécifiée pour des régions détaillées. Dans H.264/AVC il est possible de se rapporter à plusieurs images précédentes. À cette fin, un paramètre supplémentaire de l'image de référence doit être transmis en même temps avec le vecteur de mouvement. Cette technique est dénotée aussi prédiction à compensation de mouvement à multiples trames de références. La figure illustre ce concept (Figure II.5).

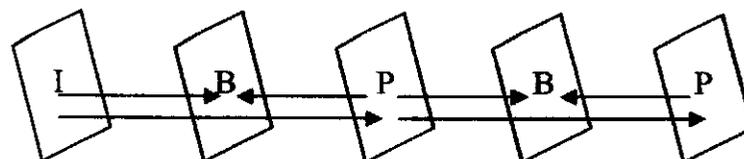


Figure II.5 Images de références pour la prédiction à compensation de mouvement

4.3.1.3 Le vecteur de mouvement

Chaque partition dans un macrobloc codé en inter est prédite d'une région de la même taille dans une image de référence. La différence entre ces deux régions (le vecteur de mouvement) a la résolution au quart de Pixel pour la luminance et au huitième de pixel pour la chrominance. Les échantillons de luminance et de chrominance aux positions échantillonnées n'existent pas dans l'image de référence. Il est donc nécessaire de les créer par interpolation à partir des échantillons des voisinages déjà codés (Figure II.6).

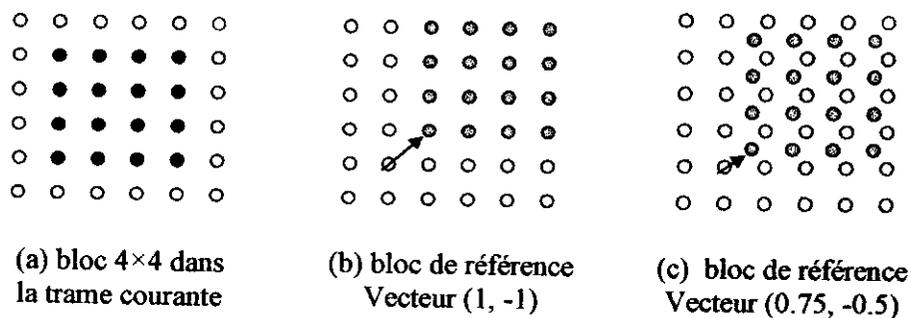


Figure II.6 Exemple de prédiction de pixel et sous-pixel

Une sous partition 4×4 dans la trame courante doit être prédite d'une région voisine de l'image de référence (Figure II.6.a). Si les composantes horizontales et verticales du vecteur de mouvement sont des nombres entiers, les échantillons appropriés dans le bloc de référence existent réellement (les points gris) (Figure II.6.b). Si un ou les deux composantes de vecteur sont des valeurs fractionnelles, les échantillons de prédiction (points gris) sont produits par interpolation entre échantillons adjacents dans la trame de référence (points blancs) (Figure II.6.c).

Dans le composant de la luminance, les échantillons aux positions de un demi pixel sont produits et interpolés des échantillons des pixels voisins à l'aide d'un filtre. Ceci signifie que chaque échantillon de un demi pixel est une somme de six échantillons voisins de pixels. Lorsque tous les échantillons de un demi pixels sont disponibles, chaque échantillon de un quart pixel est produit en utilisant l'interpolation bilinéaire entre les échantillons voisins de un ou un demi pixel.

Le codage d'un vecteur de mouvement pour chaque partition peut prendre un nombre significatif de bits, particulièrement si des petits tailles de partition sont choisies. Les

vecteurs de mouvement pour les partitions voisines sont souvent fortement corrélés ; pour cela, chaque vecteur de mouvement est prédit par des vecteurs voisins des partitions précédemment codées. Le vecteur prédit est formé à l'aide des vecteurs de mouvement précédemment calculés. La différence entre le vecteur courant et le vecteur prédit, est codé et transmis. Au décodeur, le vecteur de mouvement prédit est formé de la même manière et ajouté au vecteur différence décodé [sw2].

4.3.1.4 La compensation dans les tranches P

Dans les tranches P, la prédiction temporelle est employée, en estimant le mouvement entre les images. Un vecteur de mouvement distinct peut être envoyé pour chaque partition. Le mouvement peut être estimé à partir des multiples images de référence. Le choix de l'image de référence est employée et fait au niveau de partition de macrobloc [14]. Pour estimer le mouvement, les pixels sont d'abord interpolés pour réaliser l'exactitude de quart de pixel pour la luminance et jusqu'à huitième de Pixel pour la chrominance.

La prédiction pour chaque bloc de luminance est obtenue en déplaçant une zone de l'image de référence. Elle est définie par un vecteur de translation et un index de référence d'image. La précision de la compensation de mouvement équivaut à un quart de la distance d'échantillon. Dans les cas où le vecteur de mouvement pointe sur une position d'échantillon entier, les signaux de prédiction équivalent aux échantillons correspondants de l'image de référence; sinon, ils sont obtenus en utilisant une interpolation basée sur les positions d'échantillon. La norme H.264/AVC prend en charge la prédiction multi image à compensation de mouvement. En d'autres termes, plus d'une image préalablement codée peuvent être utilisées comme référence pour la prédiction à compensation de mouvement. Le codeur et le décodeur doivent stocker les images de référence utilisées pour la prédiction intra dans une mémoire multi image. Le décodeur reproduit la mémoire multi image du codeur selon le type de mise en mémoire de l'image de référence et de toute opération de contrôle de la mémoire qui est spécifiée dans le flux. L'index qui localise l'image de référence dans la mémoire doit être signalé.

4.3.1.5 La compensation dans les tranches B

La différence fondamentale entre les tranches B et P est que les premières sont codées pour que certains macroblocs ou blocs puissent utiliser une moyenne pondérée de deux valeurs de prédiction à compensation de mouvement distinctes pour générer le signal de prédiction.

Dans les tranches B, quatre types de prédiction inter sont possibles: liste 0, liste 1, double (interpolation) et directe. La prédiction liste 0 indique que le signal de prédiction est formé en utilisant la compensation de mouvement issu d'une image de la première mémoire, la liste 1 utilise quant à elle une image de la seconde pour générer le signal de prédiction. En mode double prédiction, le signal est formé par une moyenne pondérée d'un signal de prédiction liste 0 et liste 1 à compensation de mouvement. Le mode de prédiction directe est déduit des éléments syntaxiques précédemment transmis et peut-être une prédiction liste 0 ou liste 1 ou encore double. Les tranches B utilisent un partitionnement de macrobloc similaire à celui des tranches P [11].

4.3.1.6 La compensation dans les tranches SI et SP

Les tranches de commutation SI et SP sont les cousins étroits des tranches I et P, employer en prédiction temporel ou spatiale. Cependant, leur vertu principale est qu'elles peuvent permettre la reconstruction exactes des valeurs des échantillons, même lorsqu'en utilisant multiple images de référence. L'utilité principale de cette propriété doit permettre la commutation de bitstream. Les tranches assurent aussi d'autres fonctionnalités telles que l'accès sélectif ...

4.3.2 La prédiction intra

Contrairement à la prédiction inter, la prédiction intra n'utilise pas les images de référence mais uniquement l'image courante. Il s'agit donc d'exploiter la redondance spatiale présentée dans toute image. En effet, le principe de base est, que, dans un certain voisinage, la structure de l'image varie peu globalement. Bien sûr, ceci dépend de la taille de l'échantillon considéré, c'est pourquoi chaque taille de bloc correspond à un traitement spécifique.

De fait, pour la luminance, la prédiction est fait sur des blocs 4×4 ou des macroblocs 16×16. De plus, bien que la luminance et la chrominance n'aient pas la même résolution, la chrominance n'est codée que par blocs 8x8. Un bloc prédit est donc construit à partir de ces voisins déjà codés et reconstruits. Il est ensuite soustrait au bloc original et le bloc différence est codé et transmis. Toute la procédure de prédiction doit être transmis au bitstream, ce qui nécessite de l'information supplémentaire dont l'importance ne doit pas être négligée.

4.3.2.1 Prédiction des blocs 4x4 de luminance ou Intra 4×4

Pour la prédiction Intra, les standards précédents travaillaient généralement avec des blocs de taille 8x8. C'est donc une nouveauté de descendre jusqu'à des blocs de taille 4x4, ce qui engendre bien évidemment, une augmentation de la quantité de calculs [10]. Neuf modes de prédiction différents sont utilisés dans la norme H.264/AVC. Un mode DC, telle que tous les échantillons du bloc courant 4×4 sont prédits par la moyen de tous les échantillons voisins à gauches et juste au dessus du bloc courant qui ont été déjà reconstruits. En plus de ce mode, huit modes de prédiction dont chacun a une direction spécifique.

La figure II.7 présente la manière dont les pixels sont labellisés et la figure II.8 les directions de huit modes associés.

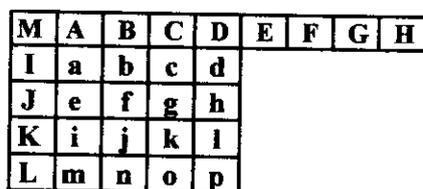


Figure II.7 Labellisation des échantillons de prédiction (4x4)

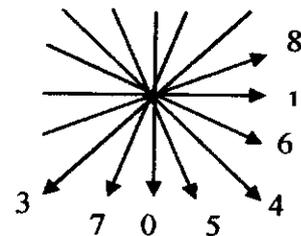


Figure II.8 Huit de neuf modes de prédiction intra

Avec a, b, c ..., p sont des pixels du bloc courant, et M, A, B..., L sont des pixels des blocs voisins.

Le mode 0 (prédiction verticale) et le mode 1 (prédiction horizontale) sont montrés explicitement sur la figure II.9. Par exemple, si le mode de prédiction verticale est appliqué, tous les échantillons au-dessous de l'échantillon A sont prédits par

l'échantillon A ; tous les échantillons ci-dessous de l'échantillon B sont prédits par l'échantillon B et ainsi de suite.

Les échantillons des autres modes de prédiction sont calculés à partir des échantillons de A à M comme montré dans le tableau II.1. Si les pixels voisins manquent (cas du premier bloc de la trame), des valeurs par défaut sont utilisées.

Tableau II.1 Modes de prédiction des blocs 4x4 de luminance

Mode	Direction
0 (vertical)	[A-D] sont extrapolés verticalement
1 (horizontal)	[I-L] sont extrapolés horizontalement
2 (composante continue)	[a-p] sont prédits par la moyenne de [A-D] et [I-L]
3 (diagonal bas-gauche)	[a-p] sont interpolés à un angle de 45° bas-gauche
4 (diagonal bas-droit)	[a-p] sont interpolés à un angle de 45° bas-droit
5 (vertical-droit)	[a-p] sont interpolés à un angle de 22.5° vertical-droit
6 (horizontal-droit)	[a-p] sont interpolés à un angle de 22.5° horizontal-droit
7 (vertical-gauche)	[a-p] sont interpolés à un angle de 22.5° vertical-gauche
8 (horizontal-haut)	[a-p] sont interpolés à un angle de 22.5° horizontal-haut

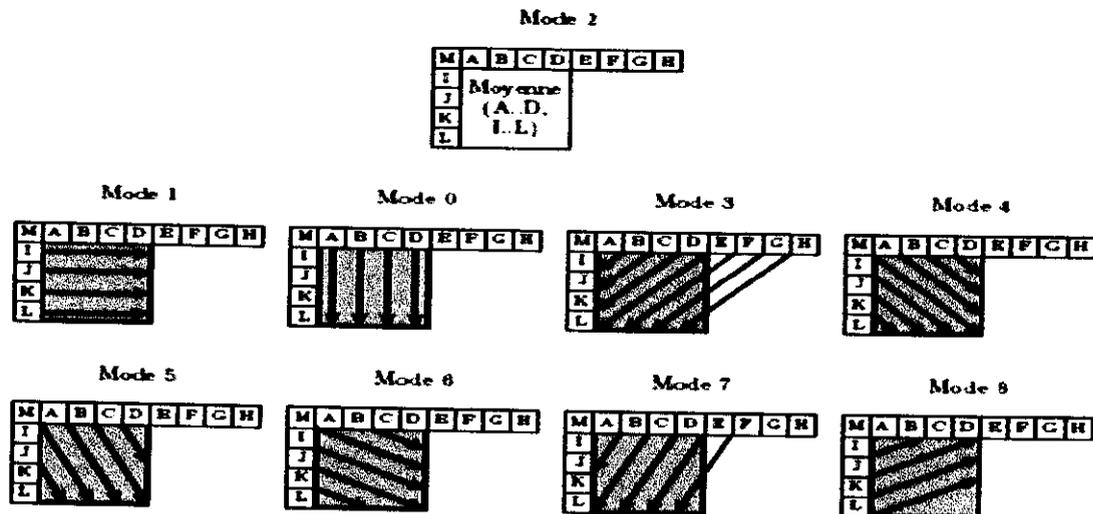


Figure II.9 Modes de prédiction des blocs 4x4 de luminance

Les flèches indiquent la direction de la prédiction de chaque mode (Figure II.9). Pour les modes de 3 à 8, les pixels prédits sont calculés par une formule appliquée aux pixels [A-M]. Une fois calculés, les neuf modes sont évalués par une

somme des erreurs absolues SAE (Sum of Absolute Errors). Le mode fournissant la SAE minimale est retenu pour la prédiction du bloc.

4.3.2.2 Prédiction des blocs 16×16 de luminance ou Intra 16×16

La prédiction par blocs 4x4 demande des calculs importants et n'est pas toujours justifiée pour des régions de faible variation. C'est pourquoi la prédiction intra image peut se faire par blocs 16x16. Cette alternative a l'avantage d'être évidemment plus rapide et moins coûteuse. Quatre modes de prédiction sont présentés dans la norme H.264/AVC dont les trois premiers (mode 0, mode 1 et mode 2) sont semblables aux modes de prédiction Intra 4×4 et un mode plan (mode 3) tel que la prédiction est une fonction linéaire entre les échantillons voisins gauche et au dessus afin de prédire les échantillons courants (les échantillons sont calculés par moyenne des valeurs obliques des deux sens de direction). Les quatre modes sont représentés sur la figure II.10 et le tableau II.2 les explicite [sw3].

Tableau II.2 Modes de prédiction des blocs 16x16 de luminance

Mode	Direction
0 (vertical)	Extrapolation à partir de H (Horizontal)
1 (horizontal)	Extrapolation à partir de V (Vertical)
2 (composante continue)	Moyenne de H et V
3 (plan)	Une fonction linéaire plan est ajustée à partir de H et V

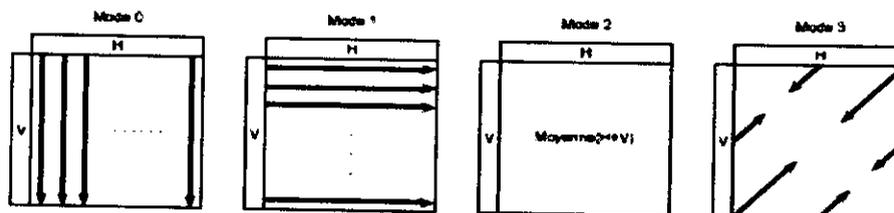


Figure II.10 Modes de prédiction des blocs 16x16 de luminance

Le critère de sélection du mode choisi est à nouveau une SAE calculée au niveau du macrobloc.

4.3.2.3 Prédiction des blocs 8×8 de chrominance

Les blocs de chrominance sont prédits à partir des blocs de chrominance déjà codés à gauche et en haut du bloc courant. Les deux composantes (Cb et Cr) utilisent les mêmes modes de prédiction. Les quatre modes sont similaires à ceux de la prédiction des blocs 16x16 de luminance, seule la numérotation des modes diffèrent (Tableau.II.3).

Tableau.II.3 Modes de prédiction des blocs 8x8 de chrominance

Mode	Direction
0 (composante continue)	Moyenne de H et V
1 (horizontal)	Extrapolation à partir de V
2 (vertical)	Extrapolation à partir de H
3 (plan)	Une fonction linéaire et ajustée à partir de H et V

Encore une fois, le critère de sélection du mode choisi est une SAE.

Remarque

Toutes ces informations (choix de prédiction, mode de prédiction, vecteur de mouvement) vont transmettre au bitstream.

4.4 La transformation

La transformation est réalisée par blocs et est appliquée sur le résidu obtenu après prédiction. H.264/AVC utilise trois transformations selon le type de données résiduelles qui doivent être codées:

- Pour les macroblocs prédits en mode 16x16, les coefficients DC (coefficients de plus basse fréquence) sont placés dans une matrice 4x4 et une transformation de Hadamard lui est appliquée.
- Pour tous les macroblocs, les coefficients continus de chrominance sont placés dans une matrice 2x2 et une transformation de Hadamard lui également appliquée.
- En fin, pour le reste des blocs 4x4, une transformation DCT est appliquée.

La figure II.11 montre l'ordre dans lequel les macroblocs sont parcourus. Si un macrobloc est codé en mode intra 16x16, le bloc contenant les composantes continues de chaque bloc de luminance 4x4 est labellisé « -1 » et est transmis en premier. Ensuite, les blocs résiduels de luminance [0-15] sont transmis dans le même ordre (Figure II.12.a). Puis, les blocs contenant les composantes continues de la chrominance (bloc 16 pour Cb et 17 pour Cr) sont envoyés (Figure II.12.b) (Figure II.12.c). Enfin, les blocs [18-25] de chrominance sont transmis.

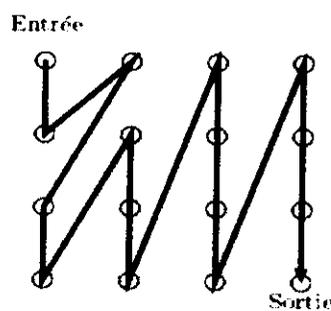


Figure II.11 Ordre de parcours d'un macrobloc 16x16

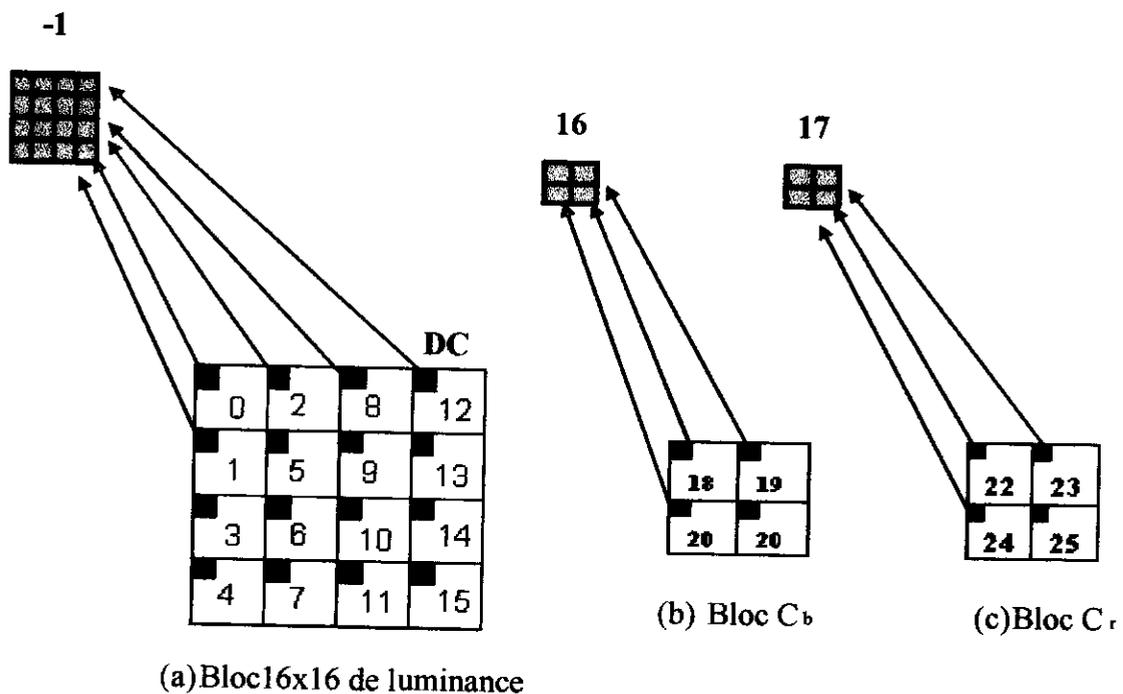


Figure II.12 Ordre de balayage des blocs de résiduel dans un macrobloc

4. 4. 1 La transformation DCT 4x4

Cette transformation est appliquée sur les blocs 4x4 de données résiduelles marquées de 0 à 15 et de 18 à 25 sur la figure II.12. Elle est basée sur la DCT classique mais avec certaines différences fondamentales [sw4] :

1. c'est une transformation entière (toutes les opérations peuvent être effectuées avec l'arithmétique de nombre entier).
2. La transformation inverse est entièrement indiquée dans le standard H.264 et si elle est suivie correctement, la disparité entre le codeur et le décodeur ne pas se produire.
3. La transformation exige seulement des additions, des multiplications et des décalages.

La DCT 4x4 d'un bloc d'entrée X est donné par :

$$Y = AXA^T = \begin{pmatrix} a & a & a & a \\ b & c & -c & -c \\ a & -a & -a & -b \\ c & -b & b & -c \end{pmatrix} (X) \begin{pmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & b & a & -c \end{pmatrix} \quad (\text{II.1})$$

Avec

$$\begin{aligned} a &= \frac{1}{2} \\ b &= \sqrt{\frac{1}{2}} \cos \frac{\pi}{8} \\ c &= \sqrt{\frac{1}{2}} \cos \frac{3\pi}{8} \end{aligned} \quad (\text{II.2})$$

La matrice Y peut être factorisée à la forme équivalente suivante :

$$Y = (CX C^T) \otimes E = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & 1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{pmatrix} (X) \begin{pmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & 1 & 1 & -d \end{pmatrix} \otimes \begin{pmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{pmatrix} \quad (\text{II.3})$$

CXC^T est le coeur de transformation. E est une matrice des facteurs de graduation et le symbole \otimes indique la multiplication terme à terme et non matriciel. Les constantes a et b sont en tant qu'avant ; et d est approximé par $1/2$.

La transformation inverse est donnée par :

$$X = C_1^T (Y \otimes E) C_1 = \begin{pmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{pmatrix} \left((Y) \otimes \begin{pmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{pmatrix} \right) \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{pmatrix} \quad (\text{II.4})$$

4. 4. 2 La transformation de Hadamard 4×4

Si un macrobloc est codé en mode intra 16×16, chaque bloc résiduel 4×4 est d'abord transformé par la transformation DCT 4×4. Les DC (coefficients continus des bloc) sont placés dans une matrice 4×4 et transformés par une transformation de Hadamard 4×4 tel que :

$$Y = \left(\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} (W) \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \right) \quad (\text{II.5})$$

W est le bloc 4×4 de coefficients DC et Y est le bloc après transformation.

Au décodeur, la transformation inverse de Hadamard est donnée par :

$$W = \left(\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} (Z) \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \right) \quad (\text{II.6})$$

Avec Z la matrice de quantification.

4.4.3 La transformation de Hadamard 2×2

Les coefficients continus de chaque bloc 4×4 de chrominance sont groupés dans un bloc W de taille 2×2 et transformés par une transformation de Hadamard comme suit :

$$Y = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} (W) \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (\text{II.7})$$

La transformation inverse est comme suit:

$$W = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} (Z) \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (\text{II.8})$$

4.5 La quantification

H.264/AVC utilise une quantification scalaire dont l'opération de base est :

$$Z_{ij} = \text{round} \left(Y_{ij} / Q_{step} \right) \quad (\text{II.9})$$

Avec Y_{ij} le coefficient de la transformation, Q_{step} la taille du pas de quantification et Z_{ij} le coefficient quantifié. Un total de cinquante-deux (52) valeurs de Q_{step} sont supportés par la norme, et indexées par un paramètre de quantification QP.

Les valeurs de Q_{step} correspondant à chaque QP sont montrées dans le tableau II.4 suivant :

Tableau. II.4 Taille du pas de quantification dans le codec H.264

QP	0	1	2	3	4	5	6	7	8	9	10	11	12
Qstep	0,625	0,6875	0,8125	0,875	1	1,125	1,25	1,375	1,625	1,75	2	2,25	2,5	
QP	18	24	30	36	42	48	51
Qstep		5		10		20		40		80		160		224

Q_{step} double pour chaque incrément de 6 dans QP et augmente de 12.5% pour chaque incrément de 1 dans QP. Les valeurs de QP peuvent être différentes pour la luminance

et la chrominance. Les deux paramètres varient entre 0 et 51 mais le paramètre QP de la chrominance est dérivé de paramètre QP de luminance.

L'opération de base de quantification inverse est donnée par:

$$Y'_{ij} = Z_{ij} \cdot Q_{step} \quad (\text{II.10})$$

Nous venons de décrire les étapes de transformation et de quantification telles qu'elles sont actuellement utilisées par H.264/AVC. Et maintenant, les coefficients de transformation quantifiés obtenus après quantification sont codés afin de diminuer la quantité de données occupée.

4. 6 Le codage entropique

H.264/AVC indique deux méthodes alternatives de codage entropique : une technique de faible complexité basée sur l'utilisation de codage à longueur variable VLC, et une technique algorithmique adaptative CABAC. Les deux méthodes représentent l'amélioration principale en termes d'efficacité de codage comparée aux normes précédentes.

4. 6. 1 Codage à longueur variable VLC

Un codage à longueur variable tel que les différents mots de code n'ont pas nécessairement la même longueur, en bits. Donc, ce codage utilise une table de mot de code défini pour tous les éléments syntaxiques. Les éléments de syntaxe incluent [15] :

1. L'élément de syntaxe de l'image, de slice, et les en-têtes de slice.
2. les indicateurs de macroblocs sautés.
3. Le type de macrobloc.
4. Paramètres de quantification.
5. Index de trames de références.
6. Vecteur de mouvement, transmis comme différence de prédiction de vecteur de mouvement.

7. Les coefficients de transformations quantifiées (de transformations 8×8 ou 4×4 , ou de transformations de Hadamard 4×4 appliquées aux coefficients DC).

Le codage par défaut utilise un ensemble de mots de codes défini pour tous les éléments syntaxiques, à l'exception des coefficients de transformation quantifiés. Ces éléments sont codés par le codage exponentiel de Golomb. Pour transmettre les coefficients de transformation quantifiés, on utilise la méthode CAVLC.

4. 6. 1. 1 Codage de Golomb exponentiel (Exp-Golomb)

Les codes exponentiels de Golomb sont des codes de longueur variable avec une construction régulière. Le tableau II.6 énumère les 9 premiers mots de codes, il est clair de cette table que les mots de codes progressent dans un ordre logique. Chaque mot de code est construit comme suit :

$$[M \text{ zéros}] [1] [\text{INFO}] \quad (\text{II.11})$$

Où INFO est un champ de M bits contenant l'information.

Le premier mot de code ne comporte ni zéro ni information. Les mots de codes 1 et 2 ont un champ à bit unique d'information, les mots de codes de 3 à 6 ont un champ de 2 bits d'information, et ainsi de suite. La longueur de chaque mot de code est de $2M + 1$ bits et chaque mot de code peut être construit à partir de numéro de code comme suit [sw6] :

$$M = \log_2 (\text{Num} - \text{code} + 1) \quad (\text{II.12})$$

$$\text{INFO} = \text{Num} - \text{code} + 1 - 2^M \quad (\text{II.13})$$

Un mot de code est décodé de la manière suivante (pour le premier mot de code, INFO et M sont nuls) (Tableau.II.5):

1. Lecture des M zéros suivis par 1.
2. Lecture des M bits d'INFO.
3. $\text{Num-code} = 2^M + \text{INFO} - 1$.

Tableau.II.5 Mots de codes d'Exp-Golomb

Num-code	Mot de code
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001
...	...

Un paramètre k est associé à Num-code de la manière suivante (Tableau.II.6) :

Tableau.II.6 Type de codage du paramètre k

Type	Description
ue	Association direct : Num-code = k .
te	Version du codage où les mots de codes courts sont tronqués
se	Association signée : Num-code = $2 k $ si $k \leq 0$ et Num-code = $2 k - 1$ si $k > 0$
me	Associations spécifiques, k est associé à Num-code d'après le tableau II.8

Chacune de ces associations (ue, te, se et me) est conçue pour produire des mots de codes courts pour les valeurs les plus apparentes et des mots de codes longs pour les valeurs les moins communes.

Ce codage à longueur variable est utilisé dans H.264/AVC pour tous ce qui n'est pas données résiduelles.

4. 6. 1. 2 Codage adaptatif à longueur variable CAVLC

Le CAVLC est une procédure de codage utilisé pour coder le résiduel : ce codage entropique adaptatif à longueur variable parcourt les blocs 4x4 en zigzag (Figure II.13).

Il est conçu pour tirer l'avantage de plusieurs caractéristiques des blocs 4x4 quantifiés :

3. Coder les coefficients différents de zéro : chaque valeur de coefficient non nul est codé dans l'ordre inverse, commençant par les haute fréquence jusqu'à le coefficient DC.
4. Coder tous les zéros avant le dernier coefficient (Total Zéro) : est la somme de tous les zéros précédant le dernier coefficient différent de zéro.
5. Coder chaque coefficient zéro (Run Before) : celui-ci, code le nombre de zéros précédent chaque coefficient non nul, dans l'ordre inverse du parcours en zigzag.

4.6.2 Codage arithmétique CABAC

L'efficacité du codage entropique peut être plus améliorée si le codage arithmétique CABAC est utilisé [16]. Par rapport au CAVLC, le CABAC garantit en général une réduction du débit binaire de 10 à 15% lors du codage de signaux TV d'une même qualité. Le codage d'un symbole de données implique les étapes suivantes:

1. Binarisation : Le CABAC utilise le codage arithmétique binaire qui signifie que seulement des décisions binaires (1 ou 0) sont codés. Un symbole non binaire (par exemple un coefficient de transformation ou un vecteur de mouvement) est converti en code binaire avant le codage arithmétique. Les étapes 2, 3 et 4 sont répétées pour chaque bit de symbole binaire.
2. Sélection de modèle de contexte : Un modèle de contexte est un modèle de probabilité pour un ou plusieurs bits de symbole binaire. Ce modèle peut être choisi à partir des modèles disponibles selon les symboles statistiques de données précédemment codés. Le modèle de contexte stocke la probabilité de chaque bit étant 1 ou 0.
3. Codage arithmétique : Un codeur arithmétique code chaque bit selon le modèle de probabilité choisi.
4. Mise à jour de probabilité : Le modèle de contexte choisi est basé sur la valeur actuel à coder (par exemple si la valeur de bit est 1, le nombre de fréquence de 1 est augmenté).

Les modèles de contexte et les schémas de binarisation pour chaque élément de syntaxe sont définis dans la norme. Au début de chaque tranche codée, les modèles de contexte sont initialisés selon la valeur initiale du paramètre de quantification QP.

La définition du processus de décodage est conçue pour faciliter l'implémentation de codage et décodage arithmétique. De façon générale, CABAC fournit une amélioration efficace comparé avec VLC dépend d'une plus grande complexité informatique.

4.7 Le filtre de déblocage

Un défaut du codage axé sur le bloc est la visibilité de la structure en blocs. Les bords sont en général reconstitués avec moins de précision que les pixels intérieurs : la pixellisation est l'un des artefacts les plus visibles des méthodes de compression actuelles. Pour cette raison, la norme H.264/AVC définit un filtre de «déblocage» adaptatif en boucle où la puissance du filtrage est contrôlée par les valeurs de plusieurs éléments syntaxiques. La pixellisation est réduite sans affecter outre mesure la clarté du contenu et la qualité subjective est considérablement améliorée.

En même temps, le filtre réduit le débit binaire de 5 à 10% tout en produisant la même qualité objective que la vidéo non filtrée. La figure II.14 illustre les performances du filtre anti-blocs.

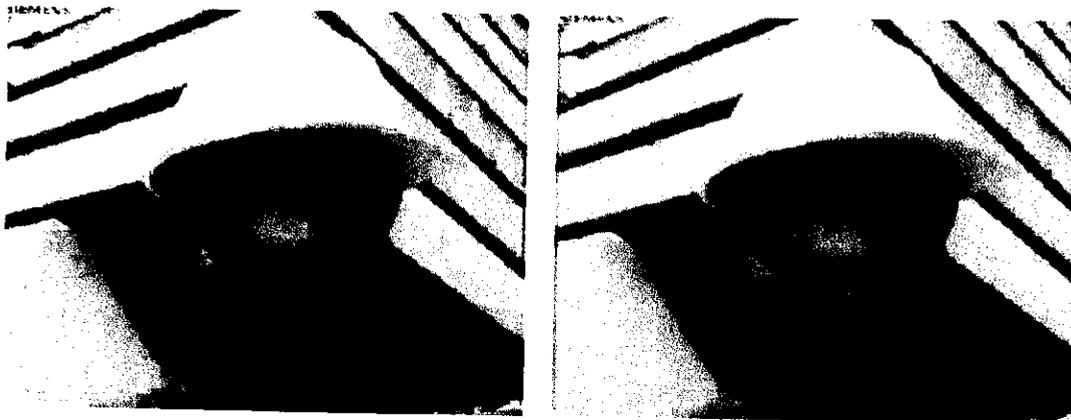


Figure II.14 La performance du filtre anti-blocs sur des images compressées, sans et avec filtre.

Plusieurs paramètres et seuils et également caractéristiques locales de l'image elle-même contrôlent la force du processus de filtrage. Le déblocage de H.264/AVC est adaptatif à trois niveaux :

1. Au niveau de tranche : la force de filtrage globale peut être ajustée sur les différentes caractéristiques de séquence vidéo.
2. Au niveau de bord du bloc : la force de filtrage dépend de la décision de la prédiction Inter/Intra, de différence de mouvement, et de la présence des résiduels codés. À partir de ces variables, un paramètre de filtrage est calculé.
3. Au niveau d'échantillon : il est crucialement important de pouvoir distinguer entre les bords exacts dans l'image et ceux créés par les coefficients de transformation quantifiés. Des bords exacts devraient être laissés non filtré autant que possibles. Afin de séparer les deux cas, les valeurs des échantillons à travers chaque bord sont analysées. Pour une explication, dénotée les valeurs des échantillons à l'intérieur de deux blocs 4×4 voisins comme $p_3, p_2, p_1, p_0, q_0, q_1, q_2, q_3$ avec la frontière entre p_0 et q_0 comme représenté sur la figure II.15. Le filtrage de deux Pixels p_0 et q_0 est la différence absolue entre ces deux dernier, qui égale à α . De même, la différence absolue entre deux pixels de chaque côté du bord ($|p_1 - p_0|$ et $|q_1 - q_0|$) est égale à β , qui est plus petit que α . En plus, la différence absolue entre p_0 et p_2 (p_0 et q_2) doit être plus petits que β .

Tous les filtres peuvent être calculés sans multiplications ou divisions pour réduire au minimum la charge de processus impliquée dans le filtrage. Seulement les additions et les décalages sont nécessaires [17].

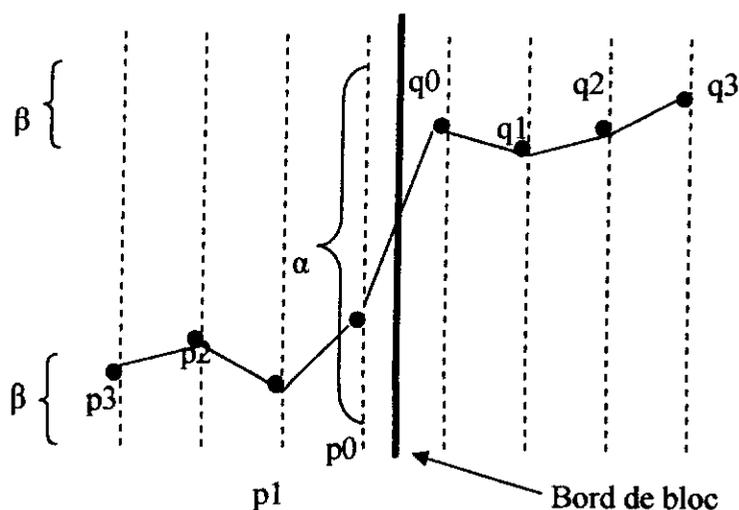


Figure II.15 Visualisation unidimensionnelle d'un bord de bloc

5. Les profils

Un profil définit un ensemble d'outils de codage ou d'algorithmes qui peuvent être utilisés pour générer un flux compatible. Tous les décodeurs conformes à un profil spécifique doivent prendre en charge toutes les fonctionnalités de ce profil. Les codeurs ne sont pas tenus d'utiliser un ensemble particulier de fonctionnalités prises en charge dans un profil, mais doivent fournir des flux compatibles. Le standard H.264/AVC définit quatre profils chacun supportant un ensemble spécifique de fonctions de codage et spécifiant la conformité d'un couple codeur/décodeur (Figure II.16) [18] :

- **Le profil de base (baseline profil) :** Ce profil vise typiquement des applications avec une résolution réduite, comportant seulement des images I et P, avec une précision de compensation au 1/4 de pixel.
- **Le profil principal (main profil) :** Inclus le module de prédiction, les tranches I, le codage inter utilisant des trames P et B, le codage entropique CAVLC et CABAC. Ce profil permet typiquement la meilleure qualité au coût d'une complexité plus élevée (essentiellement due aux tranches B et au CABAC).
- **Le profil étendu (extended profil) :** Inclus tous les outils de la norme H.264/AVC comme le profil de base excepté le CABAC, il inclus aussi la possibilité de changer de flux entre les bitstreams codés (slices SI et SP) et améliore la résilience d'erreur (Data partitioning).
- **Le profil élevé (high profil) :** Pour les usages professionnels (production, contribution).

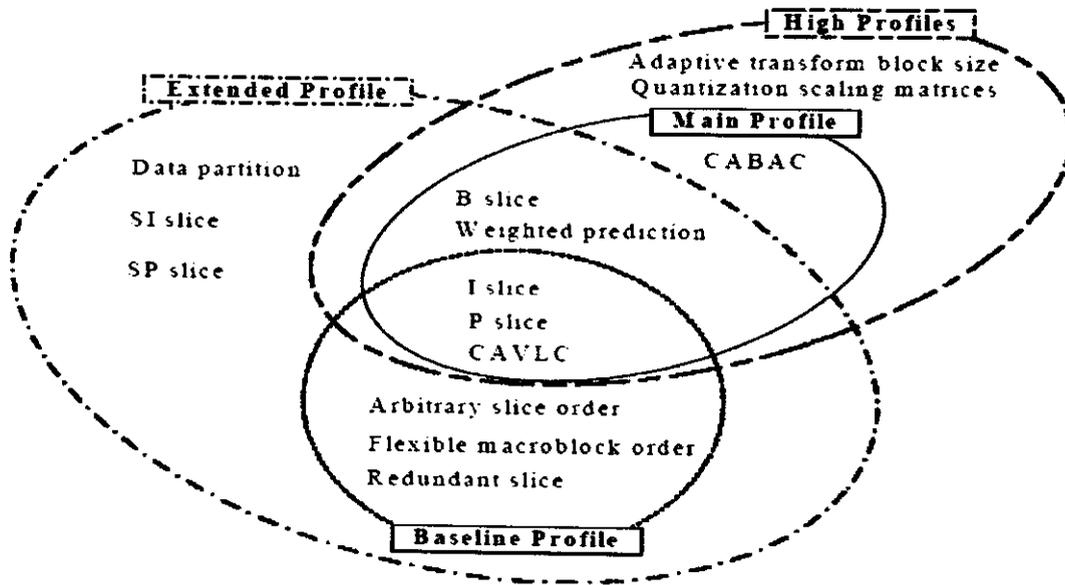
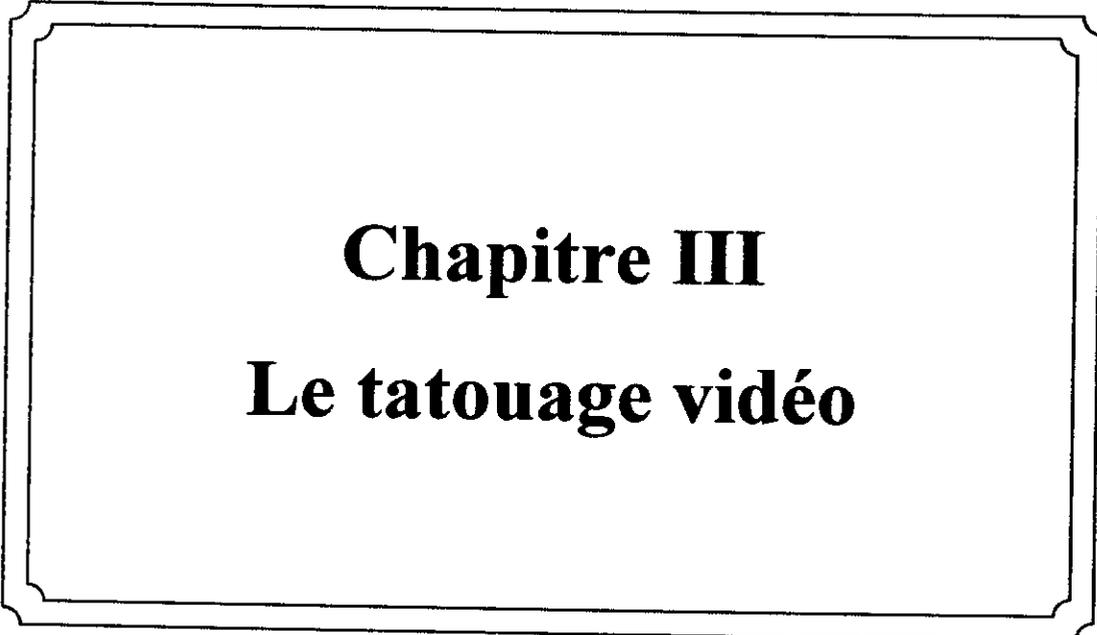


Figure II.16 Profils de H.264/AVC

Les applications visées par le profil de base sont la vidéotéléphonie, la vidéoconférence et les communications sans fil. Le profil principal est destiné à la télévision et le stockage, alors que le profil étendu serait plus adapté aux applications de diffusion de media. Cependant, chaque profil est suffisamment flexible pour supporter un large éventail d'applications.

6. Conclusion

Nous avons détaillé les principaux éléments de la norme H.264/AVC. S'appuyant sur un vocabulaire défini, ce chapitre présente les spécificités du standard en termes de modèles spatial et temporel et de codage entropique. Apportant un surcroît de précision dans la plupart des traitements effectués, H.264/AVC s'accompagne logiquement d'un accroissement non négligeable des besoins en ressources informatiques. Le tatouage vidéo dans le but de protection de copyright est un parmi leurs applications.



Chapitre III
Le tatouage vidéo

1. Introduction

Le tatouage apparaît comme étant une alternative à la protection de la propriété intellectuelle des supports multimédia. Le domaine d'application de cette technologie encore jeune, s'étend du monde 1D (l'audio), jusqu'au monde 3D, en passant par le 2D et le 2D+t (vidéo). Actuellement, il apparaît que le tatouage seul ne peut pas répondre à une protection suffisamment fiable et complète dans un milieu grand public, où les degrés de liberté (en terme de manipulation des contenus) sont trop élevés. Il semble de plus en plus évident que l'on ne pourra jamais empêcher le piratage grand public, à moins de créer des systèmes complètement propriétaires ; mais dans ce cas, il n'est encore pas assuré que la protection soit totale. La recherche en tatouage a timidement commencé au début des années 90, pour réellement débiter en 1995. Depuis, les concepts fondamentaux du tatouage ont mûri, et sont maintenant bien connus.

2. Principe général du tatouage

En général, un schéma de tatouage se découpe en trois étapes fondamentales : le traitement, l'insertion et la détection de la marque.

1. Traitement de la marque

Cette étape fait appel à des principes qui viennent aussi bien du monde de la cryptographie (chiffrement de la marque, utilisation du clé...), que du monde du traitement du signal (étalement de spectre, canal de transmission, code correcteur...).

2- La phase d'insertion

Cette phase consiste à introduire une marque dans un document afin d'identifier son propriétaire (Figure III.1). Cette insertion nécessite la disposition de l'objet d'origine et de la marque à insérer. Un troisième paramètre facultatif, une clé secrète, peut être utilisée en cas de besoin. Cette clé peut servir soit à la mise en forme de la signature soit pour localiser sur l'objet d'origine l'emplacement des pixels où il va y avoir l'insertion.

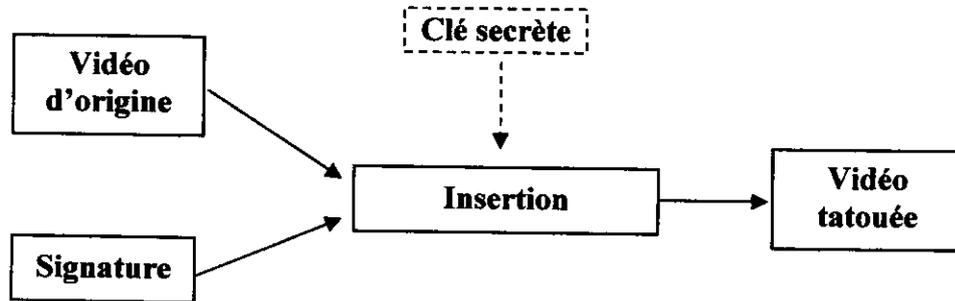


Figure III .1 Schéma d'insertion

3- La phase de détection

Cette phase permet de prouver la présence de la marque (Figure III.2). Ceci peut être réalisé par extraction complète de la marque insérée ou par détection d'un pic, supérieur à un certain seuil prédéfini, obtenu par une corrélation entre la marque d'origine et celle extraite. La détection dans le premier cas nécessite l'image tatouée, des clés et éventuellement de l'image d'origine, alors que dans le second cas, en plus des paramètres cités la marque d'origine est nécessaire.

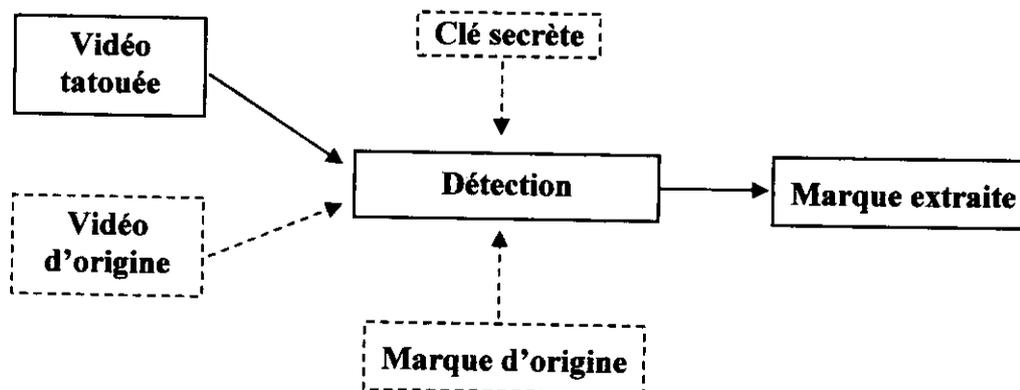


Figure III.2 Schéma de détection

3. Tatouage vidéo

Le développement rapide ces dernières années des technologies liées à l'analyse et à la transmission des vidéos numériques ainsi que des capacités de transmission associées permet non seulement l'échange de plus en plus rapide et facile des vidéos, mais également la modification de ces vidéos grâce aux techniques de post-production. Dans ce contexte, il devient alors nécessaire de définir des systèmes de tatouage vidéo permettant d'identifier l'auteur d'une vidéo, ou d'en vérifier l'intégrité.

Dans un flux vidéo, il est possible de ne tatouer que les images de type intra. De nombreux schémas développés pour les images fixes peuvent être alors appliqués aux séquences vidéos. Ces dernières présentent cependant d'autres propriétés, qui peuvent être exploitées pour l'insertion de la marque [2]:

- La taille brute d'une séquence vidéo est beaucoup plus importante que la taille d'une image fixe. L'espace d'insertion de la marque en est considérablement augmenté.
- La dimension temporelle du signal traité peut être utilisée pour l'insertion de la marque. Celle-ci peut, par exemple, être insérée dans le mouvement des différents objets de la séquence.

Les séquences vidéo présentent également des contraintes différentes de celles des images fixes :

- La complexité du schéma de tatouage doit être faible. L'insertion et la détection de la marque doivent pouvoir s'effectuer à la volée, dans la plupart des applications. La contrainte de temps réel s'applique essentiellement à la phase de détection.
- Le mouvement des objets augmente souvent la visibilité de la marque. Une marque fixe ajoutée sur un objet en mouvement, sera d'avantage perceptible que si l'objet est statique.

- Le flux vidéo est souvent compressé de manière à réduire la taille brute des séquences. L'insertion de la marque peut alors directement s'effectuer lors de la compression. L'insertion sur le format décompressé ne doit pas entraîner, après compression, une augmentation significative de la taille des données.
- la présence de la marque dans la séquence vidéo peut permettre d'autres attaques que celles liées aux images fixes. La marque doit pouvoir être détectée après une perte produite par la sélection d'une séquence précise ou la perte d'images de la séquence.

4. Les différents algorithmes de tatouage vidéo

Les méthodes de tatouage vidéo sont regroupées en deux classes :

- Les techniques issues de l'image fixe.
- Les techniques propres à la vidéo.

4.1 Techniques de tatouage vidéo provenant des schémas des images fixes

De nombreux algorithmes de tatouage vidéo sont directement adaptés des techniques appliquées à l'image fixe. Parmi eux, nous pouvons citer :

1- C.T. Hsu & J.L. Wu [19] présentent une méthode de marquage de vidéos compressées, qui constitue une extension de leur méthode, mise en oeuvre pour les images. Celle-ci modifie les fréquences moyennes des coefficients DCT en fonction des blocs voisins (spatialement pour les images intra et temporellement pour les images P et B). On force la valeur des coefficients à être inférieure ou supérieure aux coefficients des blocs voisins, suivant la valeur de l'échantillon de la marque que l'on souhaite insérer. Le signal de marquage est basé sur un motif visuel, (tel qu'un logo composé de pixels binaires). L'extraction de la marque nécessite l'utilisation de la vidéo non marquée.

2- J. Dittman & al. [20] présentent une méthode de tatouage adaptée de celle de Fridrich [21], qui consiste à insérer un motif, créé par un générateur pseudo aléatoire, sur une image. L'énergie du motif est essentiellement concentrée dans les basses fréquences, ce qui assure une plus grande robustesse face à la compression, mais rend la marque plus visible. Le principal désavantage de ce système est qu'il nécessite l'image d'origine. L'information contenue dans la marque, est très pauvre et le système permet seulement de savoir si l'image est marquée ou non. Ces deux problèmes ont été résolus par J. Dittman et al, pour être ensuite adapter à la vidéo.

Le marquage se fait sur des blocs 8×8 . Pour chaque bloc, une séquence 8×8 pseudo aléatoire est générée à partir d'une clé propre à chaque utilisateur. Un automate cellulaire est ensuite appliqué sur cette séquence, afin d'en retirer les hautes fréquences. L'insertion de la maque se fait dans les valeurs de luminance des images. Ces améliorations permettent de retrouver la marque, sans avoir recours aux images d'origine.

3- J. J. Chae & al. [22] proposent un algorithme d'insertion dans le domaine DCT. Le principe est d'insérer des images (ou une vidéo) dans une vidéo (insertion d'un volume important d'information). Les auteurs appliquent une DCT sur les blocs 8×8 de l'image à insérer et sur l'image d'origine. Les blocs sont analysés, pour obtenir un facteur multiplicatif, qui permettra d'adapter la force du marquage. La vidéo d'origine n'est pas nécessaire pour la détection.

4.2 Techniques adaptées à la vidéo

Les schémas de tatouage sont indirectement liés aux schémas de codage. D'ailleurs, de nombreuses techniques de tatouage sont directement issues des techniques de codage. On retrouve en tatouage les grands principes de la compression (domaine transformé, mouvement, aspects psychovisuels...).

La marque peut être insérée dans différents espaces. Il peut s'agir d'un domaine spatial, d'un domaine transformé, ou encore d'un domaine temporel. Le domaine spatial présente l'avantage d'être peu coûteux en temps de calcul, puisqu'il n'est pas nécessaire de réaliser des transformations. Cependant, ce domaine ne permet pas de gérer aisément l'invisibilité.

En outre, les domaines transformés sont utilisés en tatouage lors de l'insertion du système de tatouage dans un processus de codage ou lors de l'insertion de la marque dans un flux compressé. Le domaine DCT est le plus couramment utilisé. D'autres transformées (telle que la FFT, la WFT,...), sont plus marginales.

Dans cette partie, nous résumons quelques méthodes spécifiées pour le marquage vidéo.

1- M.D. Swanson & al proposent une méthode multi échelle de marquage appliquée à la vidéo non compressée [23], la méthode disposant des propriétés intéressantes. La première étape consiste à segmenter en scènes la vidéo à marquer. Chaque scène est considérée comme une entité à part entière. Un filtrage temporel est alors appliquée sur chaque scène vidéo. La marque utilisée n'est pas une marque arbitraire, mais plutôt un code identifiant le propriétaire. Cette marque est insérée dans chaque composante temporelle et les coefficients marqués sont transformés de manière inverse pour obtenir une vidéo marquée.

De plus, la marque est constituée de différentes composantes, certaines varient au cours du temps, tandis que d'autres restent fixes puisqu'elles sont insérées dans les coefficients représentant les fréquences temporelles basses. L'existence de ces composantes variables, permet la détection d'une image d'une scène sans la connaissance de son actuel index. La marque représente de plus un signal dépendant d'une clé afin d'éviter les problèmes.

2- N. Checcacci et al. [24] présentent un schéma de tatouage vidéo testé dans un contexte de communication multimédia sans fil. Le système de marquage est adapté à des schémas de codage tels que MPEG4 ou H263. Ce contexte impose certaines contraintes, propres à ce type de communication, et notamment la nécessité de résister à la perte des images ou à la désynchronisation temporelle. L'insertion se fait directement sur le flux compressé (MPEG-4) en sélectionnant des paires des coefficients, et peut se faire sur les objets tels qu'ils sont définis dans la norme MPEG4. La vidéo d'origine n'est pas nécessaire pour la détection. La capacité du système est de 15 bits.

3- En 97, G.C. Langelaar et al proposent une méthode de marquage complexe, mais également plus robuste [25]. Elle est basée sur la séparation de certaines parties du bitstream de la vidéo compressée. Pour chaque bit d'information à insérer, un ensemble de n blocs 8×8 est choisi pseudo aléatoirement à partir des images de la vidéo, il est divisé pseudo aléatoirement en deux sous ensembles de même taille. Le nombre n de blocs varie typiquement entre 16 et 64. Pour chaque sous-ensemble, l'énergie des coefficients DCT hautes fréquences est calculée. Afin d'insérer un bit, l'énergie des coefficients hautes fréquences d'un des sous ensembles est réduite en ôtant des coefficients hautes fréquences. Pour faciliter la compréhension de l'approche, des blocs consécutifs sont utilisés plutôt que des blocs aléatoirement choisis. Le bit d'information peut alors être extrait, en sélectionnant le même ensemble de blocs, en le divisant en sous ensembles, et en comparant l'énergie des coefficients hautes fréquences de chaque sous ensemble. Cette méthode nécessite seulement un décodage partiel. Cependant, la robustesse de la marque insérée est limitée, car le décodage augmente le taux d'erreur des bits insérés. De plus, cette méthode ne résiste pas à un nouvel codage utilisant un autre GOP (group of Pictures), différent de celui utilisé lors du marquage, puisque les coefficients DCT sont différents suivant que l'image est codée en I, P ou B.

4- Matsui et Tanaka proposent d'introduire la signature binaire lors de l'étape de quantification des coefficients DCT [26]. Plus précisément, ils modifient la fonction d'arrondi. Par rapport à la fonction classiquement utilisée dans un codeur MPEG, on ne considère plus l'entier le plus proche mais l'entier paire (respectivement impair) le plus proche lorsque l'on désire introduire un bit de signature à 1 (respectivement 0). L'erreur de quantification ainsi créée est donc directement corrélée avec la signature.

Les auteurs admettent que la dégradation engendrée par cette erreur est suffisamment faible pour ne pas entraîner de gêne visuelle. Si tel n'est pas le cas, il est possible de réduire le pas de quantification des tables de coefficients DCT pour se positionner à un niveau de dégradation acceptable. Malheureusement, cette opération entraîne une moindre résistance du tatouage. Le pas de quantification fournit donc un paramètre de réglage du compromis robustesse.

5- F. Hartung et B. Girod [27] proposent deux méthodes d'insertions. La première, dans le domaine non compressé, basée sur une règle additive dans le domaine spatial. La seconde technique, qui représente une variante de la première, est mise en œuvre au cours de compression.

Dans la première méthode, l'algorithme est basé sur l'étalement de spectre : consiste à transmettre un signal de bande étroite (watermark) par l'intermédiaire d'une large bande avec une fréquence étendue. Cette marque est tout d'abord échantillonnée, afin de créer un signal redondant, permettant de le rendre plus robuste. On module ensuite le signal obtenu par un bruit binaire pseudo aléatoire, qu'on amplifie ensuite par un facteur d'amplification afin d'ajouter au signal vidéo.

Pour l'extraction de la marque, la vidéo d'origine n'est pas nécessaire. Avant de commencer la détection, l'image est filtrée par un filtre passe bas. Le signal obtenu est démodulé en le multipliant par le bruit utilisé lors de l'insertion de la marque. On calcule ensuite la corrélation entre la marque et le signal obtenu.

Cette méthode peut résister à la plupart des attaques (filtrage, compression, ...) pourvu que les paramètres d'insertion soient choisis judicieusement. Cependant, ils ne développent pas l'aspect d'invisibilité de la marque, le marquage est résistant lorsque la force d'insertion est élevée, mais en contrepartie, les artefacts qui apparaissent deviennent probablement plus nombreux et plus visibles.

Dans la seconde technique de F. Hartung et B. Girod, l'insertion se fait dans le domaine DCT, sur les images intra. Les blocs DCT de l'image et ceux correspondant à la marque sont ajoutés. Afin de ne pas augmenter le débit de la vidéo, seuls les blocs marqués dont la taille est inférieure ou égale au bloc d'origine sont sélectionnés.

6- E. Koch et J.Zhao en 1995 proposent une méthode sur la transformation DCT [28]. Les étapes d'insertion et d'extraction peuvent se résumer comme suit :

Etape d'insertion

1. Soit une séquence de k bits (b_1, \dots, b_k) à cacher dans la vidéo.
2. Sélectionner dans la vidéo (selon une clé secrète) k blocs B (B_1, \dots, B_k) de taille 8×8 .
3. Calculer les coefficients DCT (a_{11}, \dots, a_{nn}) de chaque bloc sélectionné.
4. Pour i allant de 1 à k :

Soient (a_{k1}) et (a_{mn}) deux des coefficients DCT du bloc B_i , et b_i le bit à cacher

 - Si $\{(b_i = 1) \text{ et } (a_{k1}) \cdot i > (a_{mn}) \cdot i\}$ ou $\{(b_i = 0) \text{ et } (a_{k1}) \cdot i < (a_{mn}) \cdot i\}$, alors ne rien faire.
 - Si non modifier les valeurs de $(a_{k1}) \cdot i$ et $(a_{mn}) \cdot i$ pour que la relation précédente soit vérifiée.
5. Calculer la DCT inverse à partir des valeurs modifiées afin d'obtenir l'image marquée, et revenir dans le domaine spatial.

Etape d'extraction

1. Retrouver les blocs marqués grâce à la clé secrète ;
2. Calculer les coefficients DCT associés aux blocs sélectionnés ;
3. Comparer les valeurs des coefficients DCT afin de déterminer si le bit concerné du message était un '0' ou un '1'.

Cet algorithme est difficile à régler en termes de visibilité et robustesse. En effet, il s'appuie sur une découpe prédéfinie de la vidéo, il sera assez vite mis en difficulté face à une attaque géométrique modifiant par exemple la position relative de la grille de segmentation en blocs 8×8 par rapport à l'image.

7- M. Kutter et al proposent un algorithme de marquage dont l'insertion de la marque est faite dans les vecteurs de mouvements d'une vidéo de type MPEG [29].

La marque générée est une suite binaire de longueur 16 bits ou 32 bits. Les différentes étapes de marquage s'appliquent sur l'une des deux composantes des vecteurs de mouvement. Un bloc est sélectionné aléatoirement par trame, et son vecteur de mouvement est calculé. Deux bits de la marque sont alors insérés dans chaque composante du vecteur de mouvement.

Soit par exemple V , la composante verticale d'un vecteur de mouvement.

Soit $b = 0; 1$ est la valeur du bit à cacher.

- Si $(v \cdot q + T) \bmod [2] \cdot b$ alors $v = v + d$
- Sinon $v' = v$.
- $d = (2 \cdot n + 1) / q$.
- $n = 1$ si le vecteur de mouvement est le vecteur nul
- $n = 0$ sinon

q = facteur de modulation de l'amplitude du vecteur de mouvement.

v' est le vecteur de mouvement marqué, il est extrait de flux compressé.

$T = 2 \cdot \text{dim}$, dim = la taille de la fenêtre de recherche pour l'estimation de mouvement.

J. Zhang & al. [30] ont amélioré cette méthode, en réalisant une sélection de la composante de plus grande amplitude du vecteur de mouvement, pour déterminer la composante à marquer. Il s'agit d'insérer une marque invisible dans les vecteurs de mouvement d'un flux vidéo, cette insertion s'effectue sur les vecteurs de mouvement ayant la plus grande amplitude.

5. Conclusion

En général, les méthodes développées pour le tatouage vidéo couvrent un large champ de complexité, de la plus faible à la plus considérable, incluant les transformées et les modèles SVH. Les méthodes les plus complexes semblent insérer des marques avec un niveau de robustesse plus élevé. La plupart de ces méthodes s'appliquent dans un domaine non compressé. Seules, quelques techniques insèrent directement leur marque dans la vidéo compressée. Pour le marquage dans l'espace compressé, les marques sont insérées dans les coefficients DCT, les vecteurs de mouvements ou dans la structure GOP.

Il existe deux types de marquage : les méthodes additives, qui ajoutent une marque au modèle et les méthodes substitutives, qui remplacent des composantes du modèle par une marque.

A ce jour, très peu d'approches prennent en compte l'aspect temporel de la vidéo, la plupart des techniques utilisées étant importées de l'image fixe. Toutefois, certains auteurs exploitent les particularités de la vidéo (transformée 3D avec un axe temporel, vecteur de mouvement ...).

Notre but est de développer un système de tatouage vidéo pour la protection de copyright dans la norme de compression vidéo H.264. Dans le chapitre suivant, nous allons exposer deux algorithmes de tatouage destinés pour cette norme basés respectivement sur l'insertion d'une marque dans les coefficients transformés DCT 4x4 et les coefficient transformés quantifiés.



Chapitre IV
Conception et implémentation

1. Introduction

Avec le développement croissant des échanges de fichiers vidéo sous format numérique, la protection des droits de propriété intellectuelle est devenue un problème majeur. Le tatouage s'est imposé comme une solution potentielle à ce problème.

Dans ce chapitre, deux méthodes de tatouage vidéo ont été développées pour la récente et futur norme de compression vidéo H.264/AVC. L'une est basée sur la DCT 4×4 et la seconde sur la quantification.

Ce travail a été réalisé sur un CPU INTEL PENTIUM 4 : 2.80 GHz, 256 de RAM.

Nous avons utilisé le langage de programmation Microsoft Visual Studio.Net version 7.0 sous environnement windows. Ce langage a été choisi pour les raisons suivantes :

- La norme de compression vidéo H.264 utilisée dans notre travail a été téléchargée de l'Internet et compilé sous visual C++.Net [SW7].
- Le langage C permet de réaliser des applications professionnelles.
- Le traitement des images en générale nécessite un langage qui permet un travail à bas niveau avec la machine.

Les performances de notre logiciel de tatouage vidéo dans la norme H.264 pour l'application de protection de copyright ont été évaluées en terme de qualité vidéo (PSNR : Peak Signal to Noise Ratio), robustesse aux attaques et l'intégrité (extraction de l'information).

Le PSNR est le critère le plus usuel pour l'évaluation de la qualité. Il est basé sur des mesures issues de traitement de signal. Il est donné par la formule suivante :

$$PSNR_{dB} = 10 \log_{10} \left(\frac{I_{\max}^2}{\frac{1}{N \cdot M} \sum_{i=1}^N \sum_{j=1}^M [I_0(i, j) - I_d(i, j)]^2} \right) \quad (IV.1)$$

Où I_0 : l'image tatouée et I_d : l'image d'origine. N et M représentent les dimensions des images, I_{\max} est la valeur maximale d'un pixel ($I_{\max} = 255$ pour un codage sur 8 bits).

Deux séquences vidéo « claire » et « container » de format QCIF ont été utilisées pour les tests. Elles contiennent respectivement 300 et 500 images.

Les caractéristiques de compression vidéo par la norme H.264 version 9.0 utilisées pour l'insertion de la marque sont regroupées dans le tableau IV.1.

Tableau IV.1 Paramètres de la norme de compression vidéo H.264/AVC utilisés pour le tatouage

Paramètres de compression	valeurs	Description
Source Width	176	La largeur de l'image en pixel, doit être multiple de 16
Source Height	144	La hauteur de l'image en pixel, doit être multiple de 16
Frames To Be Encoded	10	Nombre des images codées dans une vidéo
Picture Type Sequence	I-P-B-P	La vidéo est codée selon la séquence associée
Intra Period	2	Période de trames I
QP First Frame	15	QP utilisé pour coder les trames I
QP Remaining Frame	16	QP utilisé pour coder les trames restantes (P)
QP B Picture	17	QP utilisé pour coder les trames B
Use Hadamard	1	Utilisation de la transformation de Hadamard
Number Reference Frames	2	Nombre des images de références utilisées pour la prédiction inter (1-5)
Entropy coding method	1	Utilisation de codage CABAC

La marque insérée dans la vidéo est une image monochrome de taille 16×16 pixels, elle est illustrée par la figure IV.1 (taille agrandie).



Figure IV.1 La marque à insérer

2. Méthode 1 : Tatouage appliqué aux coefficients transformés

L'algorithme décrit dans la première méthode est basé sur le marquage des coefficients DCT 4×4. L'approche utilisée dans cette première méthode est une méthode de tatouage des images fixes décrite par « Rajab, Sarhan & Habib » [31] que nous avons appliqué à la vidéo compressée par la norme H.264/AVC.

2.1 Algorithme proposé

Le système de tatouage vidéo développé comprend quatre étapes (Figure IV.2) :

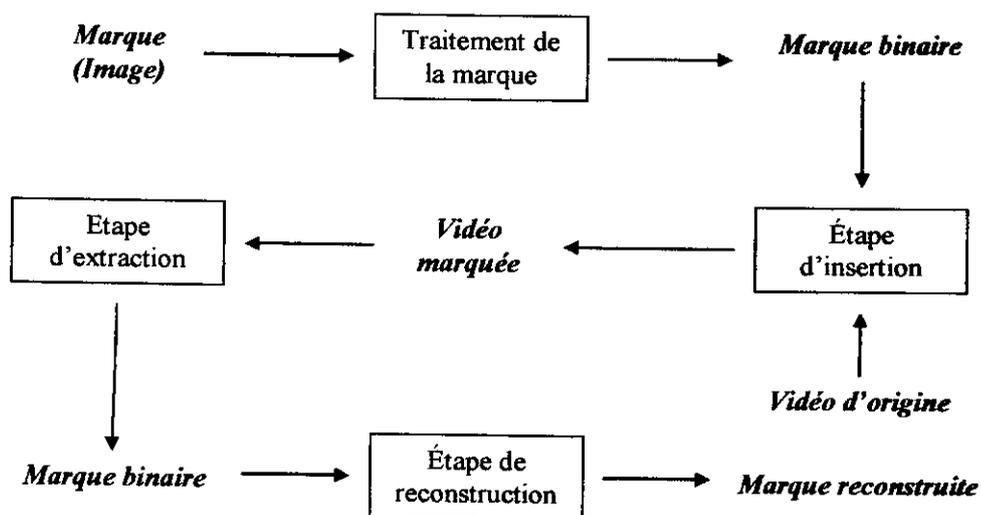


Figure IV.2 Schéma synoptique du système de tatouage vidéo développé

2.1.1 Traitement de la marque

Le tatouage consiste à insérer une marque dans une vidéo, dans notre cas la marque est traitée avant d'être insérée en appliquant une transformation de Hadamard 4×4 [32]. Le traitement est effectué en deux étapes :

- 1- Décomposition de la marque en blocs de taille 4x4 notés par $M(x, y)$ où $(0 \leq x, y \leq 3)$. Puis, chaque bloc $M(x, y)$ est transformé par Hadamard:

$$M'(x, y) = H \cdot M(x, y) \cdot H^{-1} \quad (IV.2)$$

Avec

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}$$

La marque doit être de taille $N \times M$ où N et M sont multiple de 4.

- 2- Application d'une table de binarisation pour chaque coefficient transformé. Les coefficients transformés prennent des valeurs jusqu'à 16, ce qui exige de coder chaque coefficient transformé sur 5 bits.

Le choix de la transformation de Hadamard pour le traitement de la marque est justifié par les points suivants :

- 1- Elle est utilisée dans la norme H.264/AVC.
- 2- Les valeurs de base sont +1 et -1 ce qui simplifie le calcul.
- 3- La matrice de Hadamard est inversible. Le processus inverse de traitement de la marque est calculé par :

$$H \cdot H^{-1} = n \cdot I \quad (\text{IV.3})$$

Où n est la taille de la matrice de Hadamard et I la matrice identité.

2. 1. 2 Insertion de la marque

L'insertion est effectuée au cours de la compression au niveau du module de transformation DCT 4×4 du codeur H.264. Le schéma synoptique du processus d'insertion est illustré sur la figure IV.3.

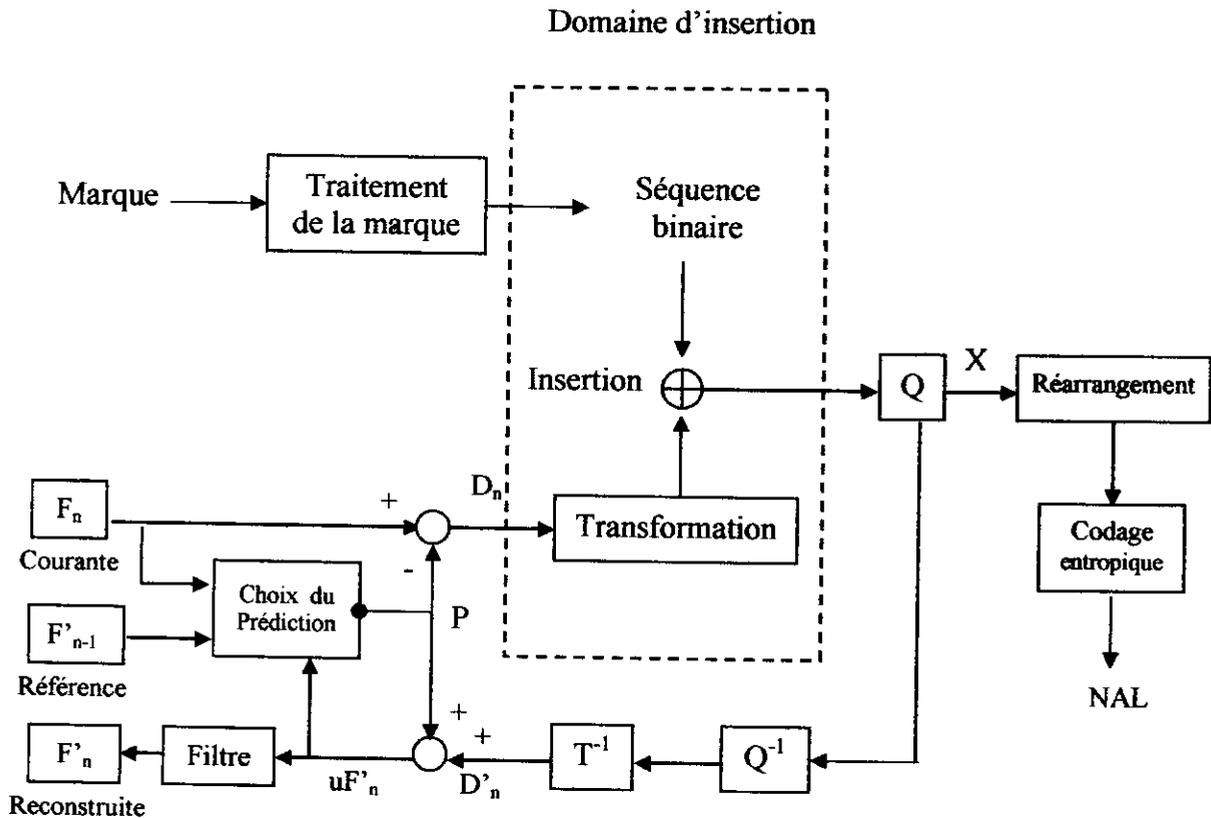


Figure IV.3 Schéma synoptique du processus d'insertion basé sur la DCT 4x4 du codeur H.264

L'insertion de la marque est effectuée sur les trames Intra du signal vidéo où chaque trame est décomposée en macroblocs. Ces derniers sont prédits pour former une prédiction P qui est soustraite au bloc courant, ce qui donne un bloc résiduel D_n comme illustré sur la figure ci-dessus. Les coefficients des blocs résiduels D_n sont transformés par la DCT 4x4, et à partir de là, le processus d'insertion commence. Les étapes d'insertion de cette approche sont comme suit :

- 1- Pour chaque macrobloc d'une trame I transformé, la sélection d'un bloc X est effectuée selon une fonction de générateur pseudo aléatoire. Soit $a(x, y)$, avec $0 \leq x, y \leq 3$, un coefficient du bloc X choisi.
- 2- À partir d'une table de binarisation, convertir le coefficient $a(x, y)$ en binaire.

- 3- Insérer la marque dans le $n^{\text{ième}}$ bit du coefficient $a(x, y)$ binarisé.
- 4- Convertir le coefficient binaire en décimal, on obtient le coefficient $a'(x, y)$.

Le schéma d'insertion est illustré sur la figure IV.4.

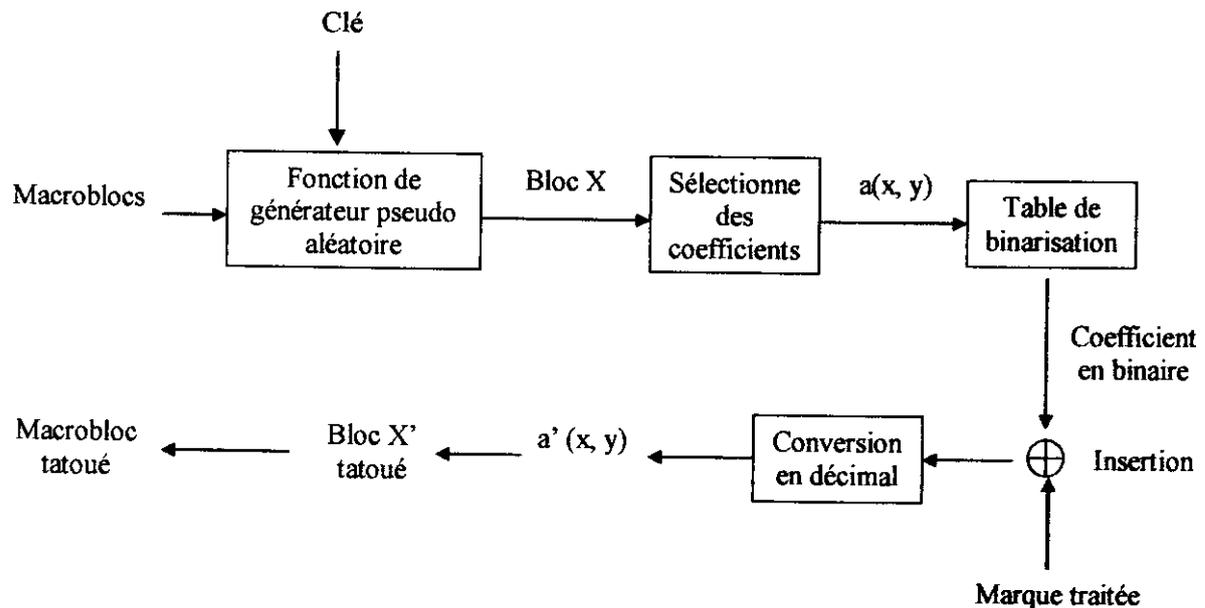


Figure IV.4 Schéma d'insertion de la marque

La fonction de générateur pseudo aléatoire

Le bloc X est sélectionné selon une séquence pseudo aléatoire $W(i)$ [33] tel que :

$$W(i) = \{G(i) / 0 \leq G(i) \leq 15\} \quad i \in \mathbb{N} \quad (\text{IV.4})$$

Où $G(i)$ est la fonction de générateur pseudo aléatoire qui sélectionne aléatoirement les positions des blocs dans un macrobloc. Cette fonction est donnée par la formule suivante :

$$G(i) = (a * G(i-1) + b) \% 16 \quad (\text{IV.5})$$

Avec a le multiplicateur, b l'incrément et $G(0)$ la racine. Ces paramètres sont des entiers naturels. $G(0)$ est une clé utilisée pour l'insertion et l'extraction de la marque. La figure IV.5 illustre un exemple de choix d'un bloc dans un macrobloc en utilisant $G(i)$. Pour $G(i) = 9$, l'insertion s'effectue dans le bloc numéro 9.

0	1	4	5
2	3	6	7
8	9	12	13
10	11	14	15

Figure IV.5 Exemple de sélection d'un bloc d'insertion dans un macrobloc

2. 1. 3 Extraction de la marque

L'extraction de la marque est exécutée au niveau du décodeur H.264/AVC, un décodage partiel est effectué sur la vidéo compressée : les coefficients fournis par le NAL sont décodés par le décodeur entropique et réarrangés pour obtenir les coefficients X. Ces derniers subissent une quantification inverse. À partir des coefficients déquantifiés, les informations insérées sont extraites (figure IV.6).

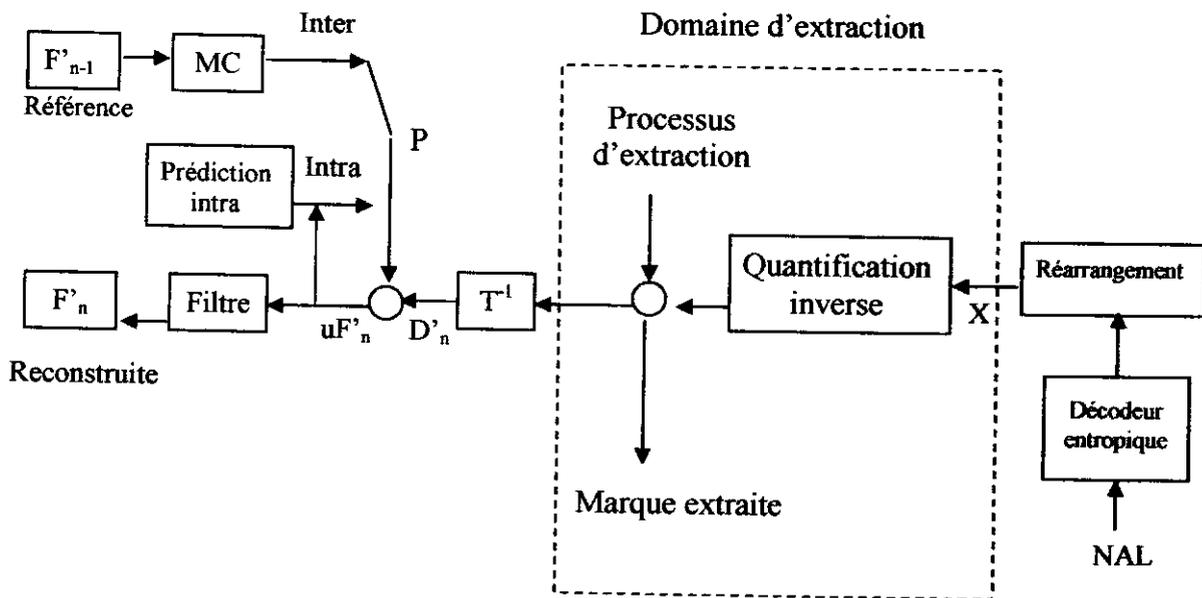


Figure IV.6 Schéma global d'extraction dans le décodeur H.264/AVC

Le processus d'extraction est effectué par l'application du traitement inverse d'insertion :

- 1- Application de la fonction du générateur pseudo aléatoire $G(i)$ pour sélectionner le bloc X tatoué.
- 2- À partir de la table de binarisation, convertir le coefficient déquantifié $a(x, y)$ en binaire.
- 3- Extraire la marque à partir du $n^{\text{ième}}$ bit du coefficient $a(x, y)$ binarisé.

2.1.4 Reconstruction de la marque

Cette étape est effectuée en appliquant le processus inverse de traitement de la marque :

- 1- À partir d'une table de binarisation, regrouper chaque cinq bits consécutifs de la séquence et les convertir en décimal. Les résultats obtenus sont des coefficients de la transformation de Hadamard.
- 2- Réarranger ces coefficients dans une matrice de taille égale à la taille de la marque.
- 3- Application de la transformation inverse de Hadamard 4×4 .

2.2 Choix expérimentaux et résultats

Le choix de chaque paramètre est justifié dans ce qui suit :

2.2.1 Choix de la taille de la marque

Notre procédure de tatouage consiste à marquer un ensemble de blocs déterminés par la fonction $G(i)$. La marque à insérer a une résolution de 16×16 pixels qui doit être ajusté avec les blocs déterminés. La marque a une capacité de 1280 bits qui nous permet de tatouer que cinq (05) blocs dans un macrobloc selon les positions générées par la séquence pseudo aléatoire.

Dans le codeur H.264/AVC, les trames d'une vidéo à coder sont de types I, P ou B. L'algorithme de tatouage propose d'insérer la marque que dans l'intra à cause des effets de dérive (Le fait de modifier les trames P ou B entraîne des perturbations dans les autres trames).

Comme la marque insérée est de taille $N = 16 \times 16$ pixels, la vidéo à marquer doit avoir 3 trames de type I pour que la marque soit complètement insérée. Une trame d'une vidéo (de type QCIF) de taille 144×176 contient 99 macroblocs, donc 99×5 bits sont insérés par trame.

Une marque de taille plus grande peut être insérée si et seulement si nous augmentons le nombre de trames I, par exemple avec une marque 32×32 , le nombre de trames I est supérieur à 10. Dans ce cas, le processus compression-tatouage devient plus long.

2.2.2 Localisation de la position d'insertion

Le coefficient à tatouer dans un bloc X sélectionné est choisi expérimentalement selon un parcours en zigzag des coefficients de basses fréquences (BF) aux hautes fréquences (HF) (Figure IV.7).

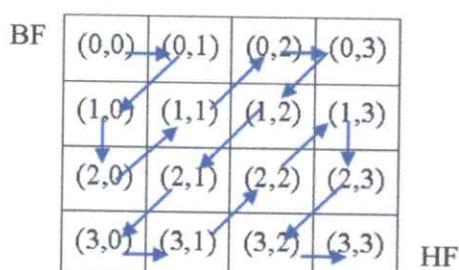
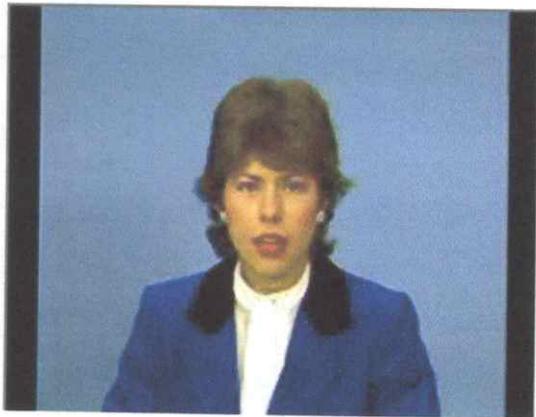


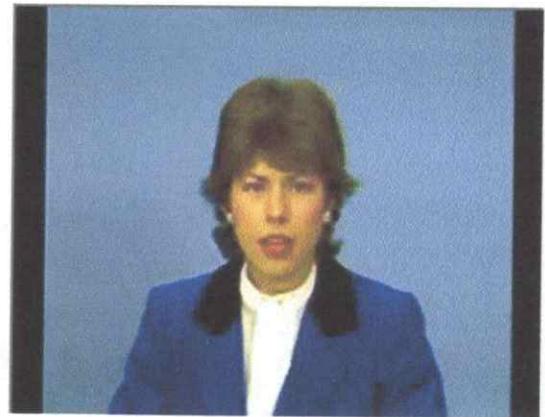
Figure IV.7 Parcours en zigzag des coefficients transformés dans un bloc

La meilleure position d'insertion trouvée empiriquement du point de vue qualité de la vidéo (PSNR) et intégrité de la marque est localisée au coefficient $a(x, y) = (3, 3)$.

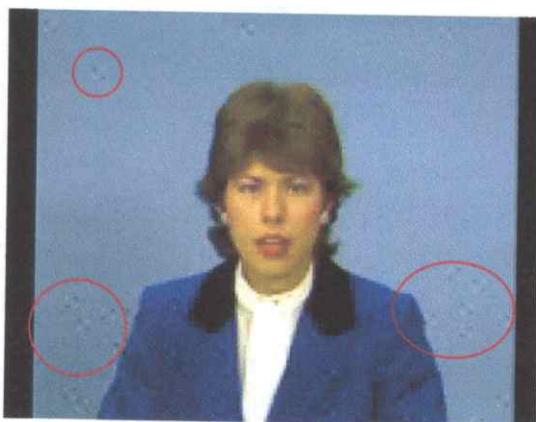
Les tests d'insertion effectués sur différentes positions sont illustrés sur la figure IV.8 pour la vidéo « claire » et la figure IV.9 pour « container » (les dégradations sont entourées en rouge sur les images). Le tableau IV.2 représente le PSNR et le taux d'extraction de la marque dans différentes positions.



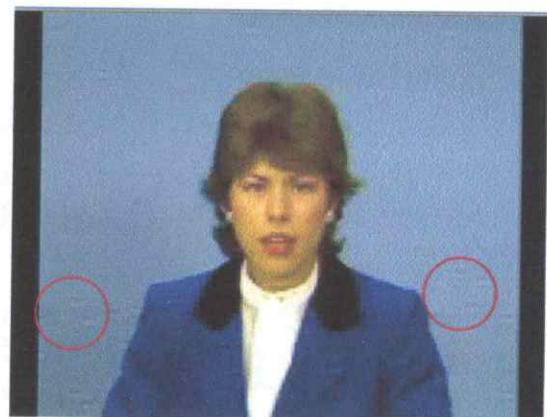
Vidéo compressée
PSNR = 41,29 dB



Vidéo tatouée en position (3,3)
PSNR = 40,38 dB



Vidéo tatouée en position (1,1)
PSNR = 40,36 dB



Vidéo tatouée en position (0,2)
PSNR = 40,55 dB

Figure IV.8 Séquence « claire » compressée et tatouée à différentes positions

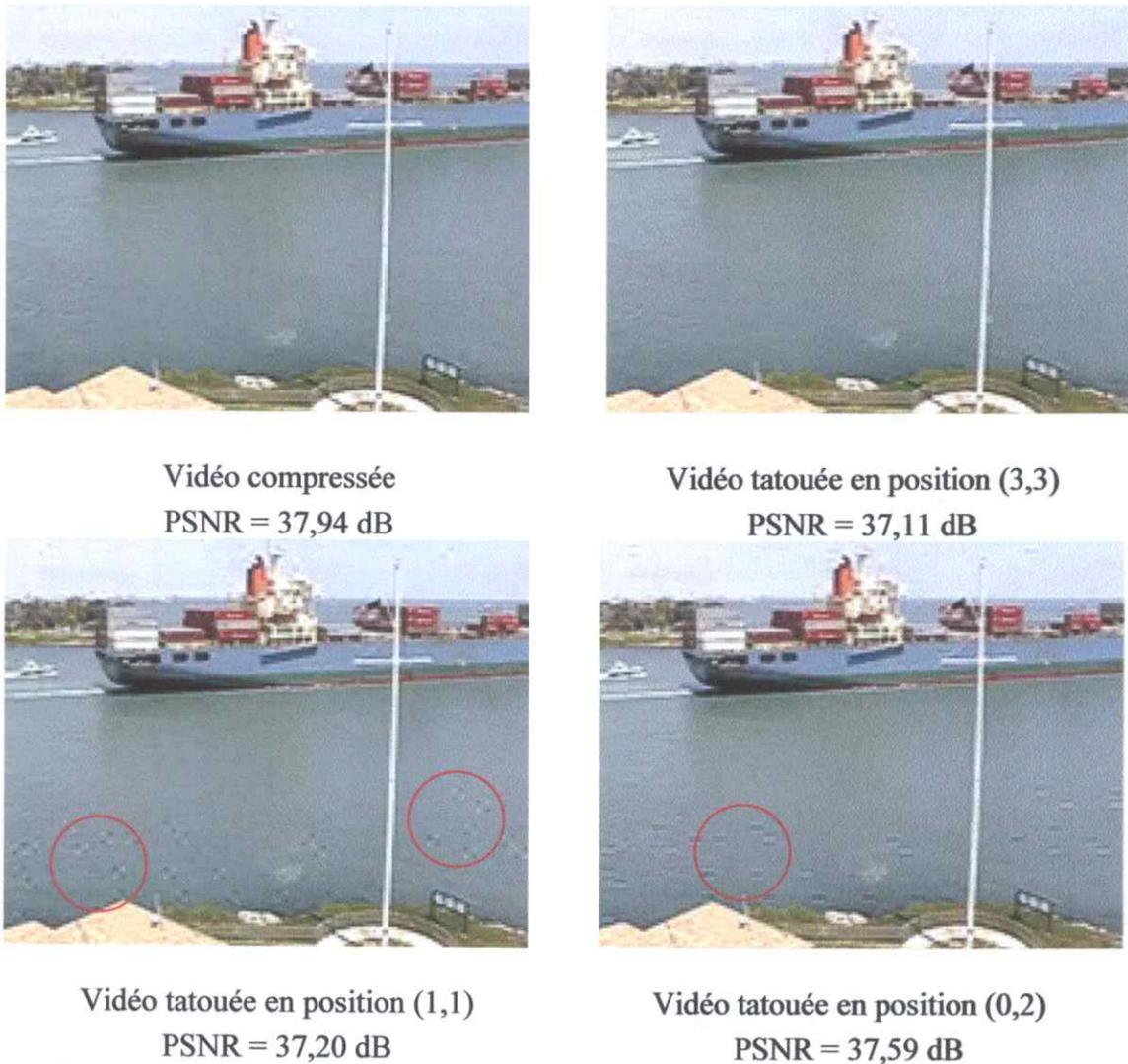


Figure IV.9 Séquence « container » compressée et tatouée à différentes positions

Tableau IV.2 Qualité et taux d'extraction pour différentes positions d'insertion

Séquences	Claire			Container		
	(3,3)	(1,1)	(0,2)	(3,3)	(1,1)	(0,2)
PSNR (dB)	40.38	40.36	40.55	37.11	37.20	37.59
Taux d'extraction (%)	100	73	96.87	100	62.5	39.06

À partir des résultats obtenus, on conclut que la meilleure position est localisée au point (3,3) du point de vue robustesse, qualité de la vidéo (PSNR) et intégrité.

Remarque

Dans la position $(x, y) = (2,2)$, la marque est complètement extraite mais la qualité de la vidéo est moindre par rapport à la position (3,3).

2.2.3 Choix du $n^{\text{ième}}$ bit à substituer

À l'aide d'une table de binarisation, le coefficient $a(3,3)$ est converti en binaire sur 16 bits (Figure IV.10). L'insertion est faite en substituant le $5^{\text{ième}}$ bit MSB (More Significant Bit) par un bit de la séquence à insérer. La substitution est donnée par :

$$a_{\text{binaire}}[5] = W[i] \quad 0 \leq i \leq 1280 \quad (\text{IV.6})$$

Où a_{binaire} est le coefficient $a(3,3)$ binarisé, $W[i]$ est la marque à insérer et i désigne la capacité de la marque.

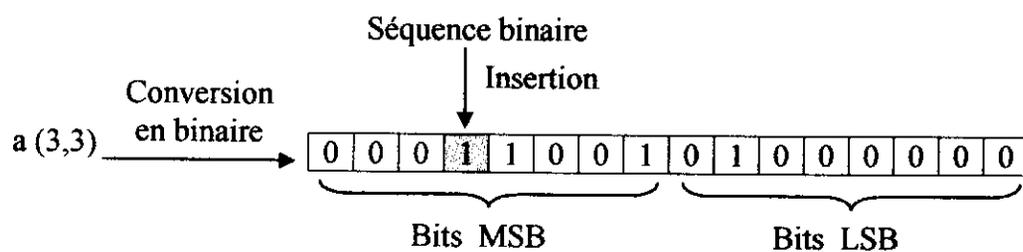
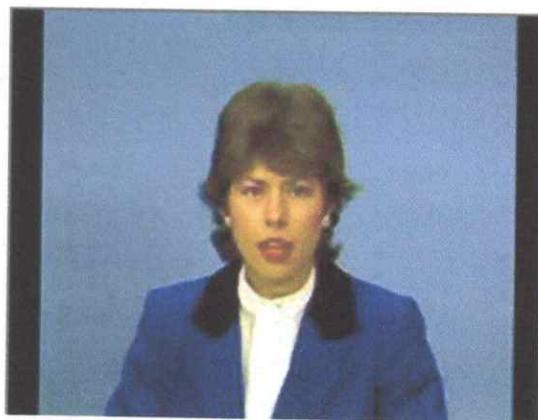


Figure IV.10 Insertion dans le $5^{\text{ième}}$ bit MSB du coefficient choisi $a(3,3)$

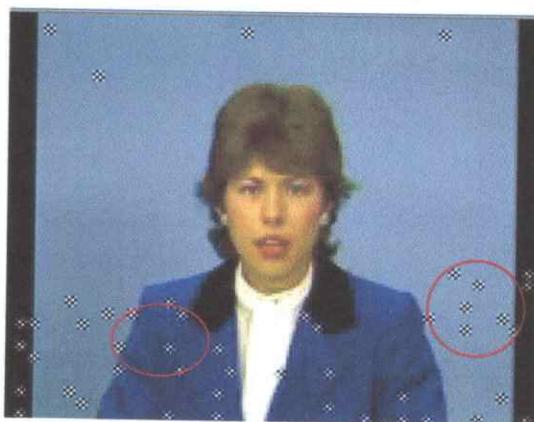
Cette décision a été adoptée après plusieurs tests commençant par les bits LSB (Less Significant Bit) aux bits MSB. Les deux figures (figure IV.11 et figure IV.12) montrent les tests effectués pour choisir le coefficient à substituer tout en conservant la qualité visuelle, Le tableau IV.3 illustre les variations des PSNR et le taux d'extraction de la marque de quelques positions du coefficient binaire à tatouer.



Vidéo tatouée en 5^{ième} bit MSB
PSNR = 40,38 dB



Vidéo tatouée en 6^{ième} bit MSB
PSNR = 38,24 dB



Vidéo tatouée en 8^{ième} bit MSB ; PSNR = 34,93 dB

Figure IV.11 Séquence « claire » tatouée en différentes positions du coefficient binaire à tatouer



Vidéo tatouée en 5^{ème} bit MSB
PSNR = 37,03 dB



Vidéo tatouée en 6^{ème} bit MSB
PSNR = 35,85 dB



Vidéo tatouée en 8^{ème} bit MSB ; PSNR = 32,72 dB

Figure IV.12 Séquence « container » tatouée en différentes positions du coefficient binaire à tatouer

Tableau IV.3 Résultats du choix du $n^{\text{ième}}$ bit MSB

Vidéo	Insertion	PSNR (dB)	Taux d'extraction (%)
Claire	8 ^{ième} bit LSB	41,29	0
	4 ^{ième} bit MSB	41,29	0
	5 ^{ième} bit MSB	40,38	100
	6 ^{ième} bit MSB	38,24	100
	8 ^{ième} bit MSB	34,93	100
Container	8 ^{ième} bit MSB	37,94	0
	4 ^{ième} bit MSB	37,94	0
	5 ^{ième} bit MSB	37,03	100
	6 ^{ième} bit MSB	35,85	100
	8 ^{ième} bit MSB	32,72	100

3. Méthode 2 : Tatouage appliqué aux coefficients transformés quantifiés

Le deuxième algorithme de marquage, qui été implémenté et adapté au codeur H.264 est basé sur la quantification. Il appartient à la classe de schéma additive aveugle où la vidéo d'origine n'est pas nécessaire pour l'extraction de la marque.

Le principe de cet algorithme consiste à traiter la marque en quantifiant les données. L'insertion est effectuée en calculant l'entropie de chaque bloc dans le macrobloc, afin d'ajuster les coefficients de la marque au degré de perception psychovisuelle, pour mieux exploiter les propriétés du système visuel humain (SVH) et avoir un tatouage invisible.

3.1 Algorithme proposé

L'algorithme proposé consiste à insérer une maque dans le module de quantification du standard H.264. La marque est tout d'abord traitée, pour être ensuite insérée dans les coefficients DCT quantifiés de moyennes fréquences, de chaque trame I de la séquence vidéo.

Les étapes de l'algorithme se résume comme suit (figure IV.13):

- 1) Le traitement de la marque : quantification de la marque.
- 2) L'insertion de la marque :
 - La recherche des blocs d'insertion par le calcul d'entropie,
 - La sélection des blocs d'insertion selon une clé,
 - La sélection des coefficients à tatouer,
 - le marquage des coefficients sélectionnés.
- 3) l'extraction de la marque :
 - le calcul d'entropie pour la sélection des blocs de forte entropie,
 - la récupération des coefficients marqués selon la clé,
 - l'extraction des bits de la marque.
- 4) la reconstruction de la marque.

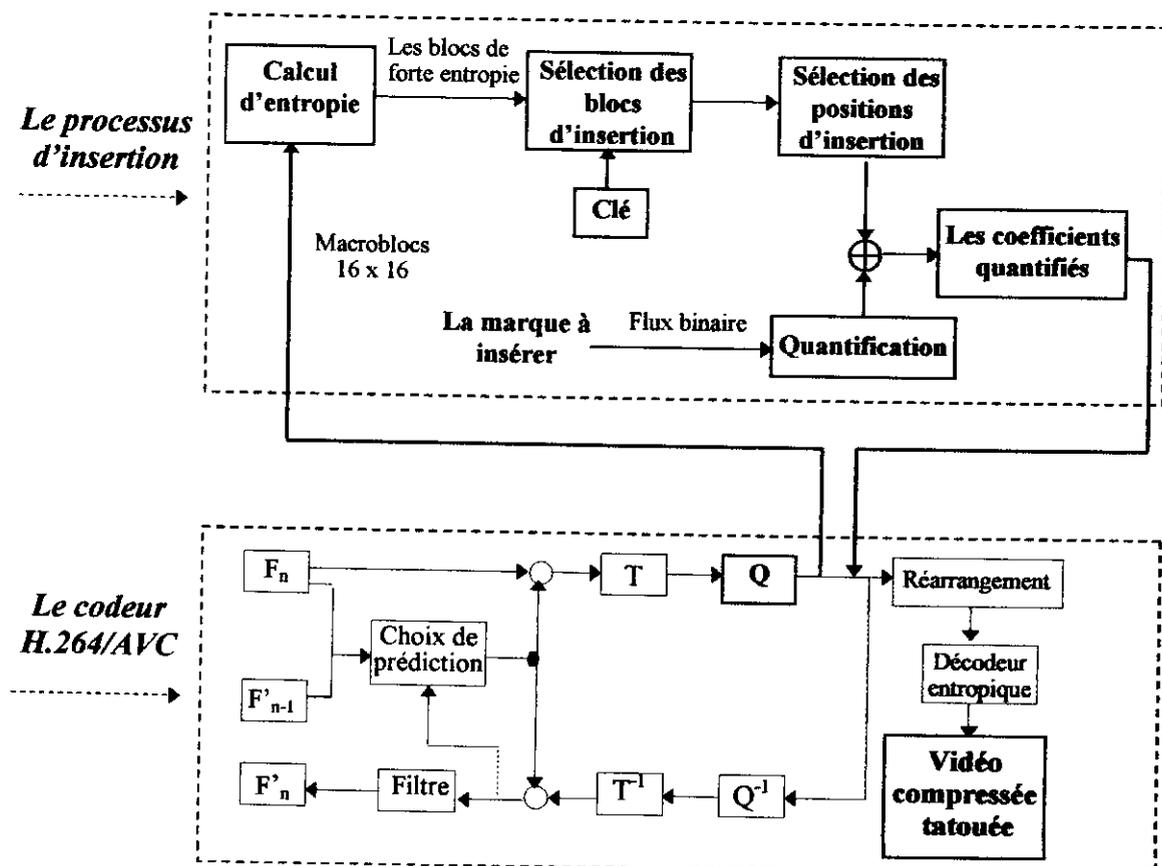


Figure IV .13 Schéma globale d'insertion dans le module de quantification du standard H.264/AVC

3. 1. 1 Traitement de la marque

La marque utilisée pour le tatouage est représentée par un train binaire B_i de taille N . le train B_i subit un traitement de quantification pour être inséré dans le flux vidéo. Ce traitement est choisi pour deux raisons : la première, l'intervalle entre les valeurs de la marque et celle de la vidéo est très large. Afin de palier à cet inconvénient, la marque doit être quantifiée pour être insérée.

La deuxième raison, peut se justifiée par l'utilisation de la fonction de quantification. C'est à dire, que dans le traitement de la marque plusieurs choix de QP sont possibles pour coder le même bit, le QP donnant une distance minimale entre les valeurs de la marque quantifiées et les coefficients quantifiés de la vidéo est choisi (parmi tous les y possibles, on sélectionne celui qui est le plus proche de x).

L'utilisation de quantification dans notre schéma de tatouage permet de préserver un bon compromis robustesse-invisibilité.

Le même processus de quantification de la norme H.264 est appliqué pour le traitement de la marque. Ce processus est détaillé au chapitre II. Dans la norme H.264/AVC, un total de cinquante-deux (52) valeurs de QP est utilisé. Pour la quantification de la marque, seulement quatre (04) QP sont choisis (tableau IV.4).

Tableau IV.4 Les QP utilisés pour la quantification de la marque

QP	28	32	36	40
Q_{step}	16	26	40	64

3. 1. 2 Insertion de la marque

Les étapes d'insertion de la marque sont les suivantes:

3.1.2.1 La recherche des blocs d'insertion

Pour chaque macroblocs 16x16 d'une trame I, un calcul d'entropie est effectué. Le bloc présentant une forte entropie est sélectionné pour l'insertion. Nous avons introduits le calcul d'entropie pour assurer plus d'invisibilité.

– Le calcul d'entropie

Une des propriétés fondamentales du tatouage est le caractère invisible de la marque dans l'image. Pour cela, les chercheurs dans ce domaine proposent d'insérer la signature intelligemment, en d'autre terme en étudiant le contenu de la séquence vidéo.

L'idée est d'utiliser les propriétés du SVH pour cacher au mieux la marque dans des régions de l'image pour lesquelles l'œil humain est insensible.

L'entropie permet de déterminer les zones de textures et des contours. Une fois ces régions connues, il suffit au tatoueur d'insérer l'information dans les zones adéquates. Pour adapter une marque à la vidéo à tatouer, et la rendre la moins visible à l'oeil, il est préférable d'insérer plus d'informations dans les zones texturées et moins dans les zones uniformes, car la modification des pixels due au tatouage dans les endroits uniformes à faibles variance est facilement détectable [34]. Sur la figure V.14 nous illustrons un exemple d'insertion de la marque dans deux zones respectivement texturée (a) et homogène (b).



Figure IV.14 Deux images extraites d'une vidéo, (a) zone texturée, (b) zone homogène

Sur l'image (b) de la figure IV.14, on constate aisément que la dégradation est plus évidente dans les zones homogènes, alors que sur (a) la dégradation est faible.

En effet, l'entropie est une grandeur qui caractérise la quantité d'information que contient une image, par exemple une image dont tous les pixels ont la même valeur, contient très peu d'information car elle est extrêmement redondante, son entropie est donc faible. En revanche, une image dont tous les pixels ont une valeur aléatoire contient beaucoup d'information, donc son entropie est forte.

En pratique, l'entropie d'une image numérique est inversement liée à la probabilité d'apparition des niveaux de gris dans l'image. Plus une valeur de gris k est rare, plus sa probabilité d'apparition $p(k)$ est faible, et cela contribue à une entropie globale plus grande.

Par définition, l'entropie H_0 est donnée par [35]:

$$H_0 = - \sum_{k=0}^n (p(k) \cdot \log_2 p(k)) \quad (\text{IV.7})$$

Une fois les entropies des blocs de chaque macrobloc sont calculées, celles-ci sont classées par ordre décroissant afin d'en choisir les blocs ayant de grande valeur d'entropie et d'y insérer la marque.

3.1.2.2 Le choix du bloc d'insertion

Pour l'amélioration des performances de l'algorithme point de vue robustesse et qualité visuelle. Nous avons introduit une séquence pseudo aléatoire sur les blocs de forte entropie. Le bloc de forte entropie sélectionné pour insérer un bit de la marque est choisi aléatoirement selon un générateur d'une séquence pseudo aléatoire. L'utilisation d'un tel générateur permet de mieux cacher l'information insérée et offre plus de robustesse aux attaques. La fonction utilisée pour la génération de la séquence est la même que celle utilisée pour la première méthode, mais avec une clé différente (la clé est représentée par le paramètre $G(0)$) utilisée pour l'insertion et l'extraction de la marque.

3.1.2.3 La sélection des coefficients à tatouer

L'insertion de la marque ne s'opère pas sur la totalité des coefficients du bloc, mais sur quelques uns soigneusement choisis. La fonction zigzag permet de trier dans l'ordre décroissant les amplitudes des coefficients DCT quantifiés d'un bloc, en commençant

par la composante continue DC qui représente la valeur moyenne des niveaux de gris des pixels d'un bloc. Suivie des composantes variables AC représentant les amplitudes des coefficients quantifiés dans les trois bandes de fréquence respectivement basses, moyennes et hautes (Figure IV.15).

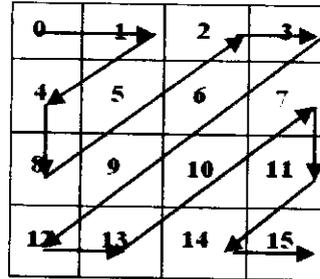


Figure IV.15 Le parcours en zigzag dans un bloc 4x4

Les basses fréquences correspondent aux zones homogènes de l'image, donc une insertion de la marque dans ces zones rend le tatouage robuste mais visible, alors qu'une insertion en hautes fréquences rend le tatouage invisible mais fragile.

Pour réaliser le compromis robustesse-imperceptibilité, il est préférable de choisir la zone des moyennes fréquences (Figure IV.16).

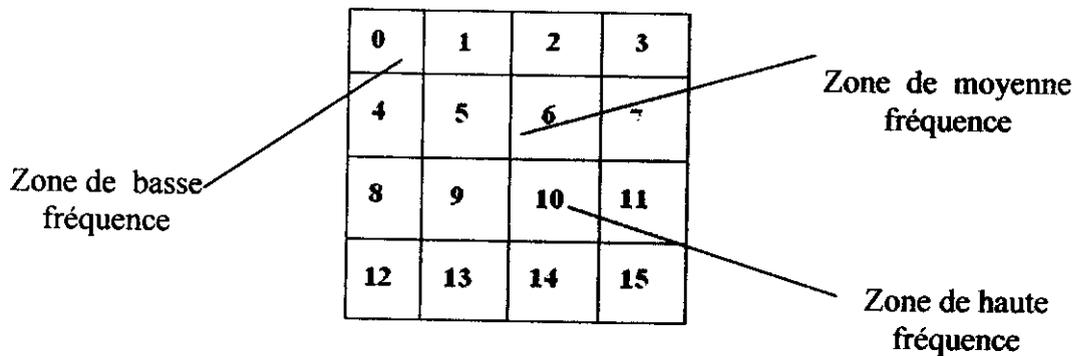


Figure IV.16 Sélection des coefficients de moyennes fréquences dans un bloc 4x4

3.1.2.4 Le processus d'insertion

La quantification de la marque se fait par quatre (04) QP différents (chaque QP est associé à un Q_{step} , donc $Q_{stepK} = \{Q_{step1}, Q_{step2}, Q_{step3}, Q_{step4}\}$, $k=1..4$), se qui permet, au cours d'insertion, de chercher le coefficient $MQ_{iK \min}$ ayant une distance minimale (DES_k) avec le coefficient quantifié du flux vidéo ($XQ(u, v)$). L'insertion se fait par le choix du maximum entre le coefficient quantifié du flux vidéo ($XQ(u, v)$) et $MQ_{iK \min}$.

L'insertion se fait selon la manière suivante:

Soit $Q_{stepK} = \{Q_{step1}, Q_{step2}, Q_{step3}, Q_{step4}\}$

Pour i allons de 1 jusqu'à N (taille de MQ_{iK}) faire

Si $MQ_{iK} = 0$ alors $XQ(u, v) = 0$

Si non

Calculer

1 - Pour K allons de 1 jusqu'à 4

$$DES_k = MQ_{iK} - XQ(u, v)$$

2 - $MIN = \min(DES_k) / k=1..4.$

3 - $XQ(u, v) = \max(MQ_{iK \min}, XQ(u, v))$

Où :

MQ_{iK} : Le $i^{\text{ième}}$ bit quantifié par le $k^{\text{ième}}$ QP.

$XQ()$: Le coefficient DCT quantifié.

(u, v) : Les indices de positionnement de $XQ()$ dans la matrice 4×4 .

i, k : conteurs.

DES_k : la distance entre $XQ()$ et MQ_{iK} .

MIN : la distance minimale entre $XQ()$ et MQ_{iK} .

K_{\min} : correspond au DES minimale.

3. 1. 3 Extraction et reconstruction de la marque

L'extraction de la marque est effectuée au cours du décodage partiel du flux compressé par la norme H.264. Elle se fait en aveugle, c'est-à-dire qu'elle ne nécessite pas la vidéo d'origine.

Pour l'extraction, le décodeur applique le même masque d'entropie, cette fois, sur la vidéo tatouée afin de reproduire les blocs de forte entropie. L'information est extraite de chaque bloc de forte entropie sectionné par une séquence pseudo aléatoire. La figure IV.17 illustre la phase d'extraction.

Le processus d'extraction de la marque peut se résumer comme suit :

Pour i allons de 1 jusqu'à N (N est la taille de MQ'_i) faire
 Si $XQ'(u, v) = 0$ alors $MQ'_i = 0$
 Si non $MQ'_i = 1$

Où MQ'_i la marque extraite et $XQ'()$ le coefficient tatoué dans le bloc de forte entropie sélectionné selon la fonction pseudo aléatoire.

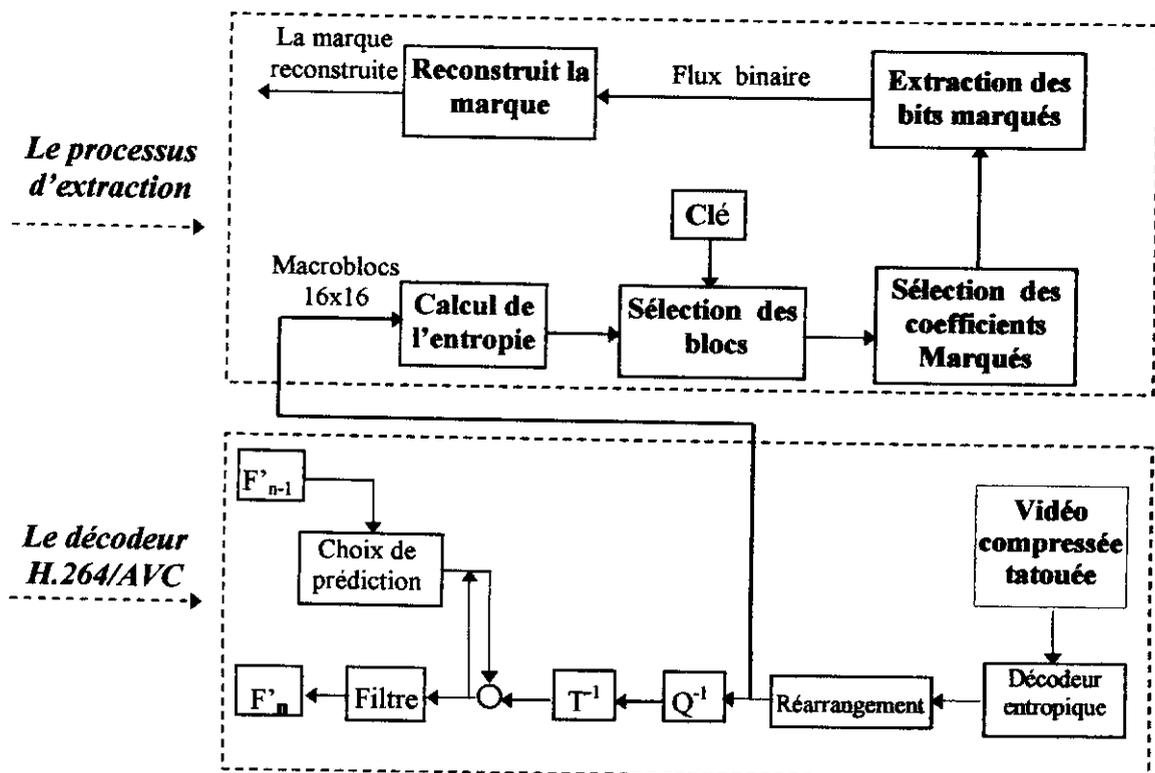


Figure IV.17 Schéma d'extraction de la marque dans le décodeur H.264/AVC

3.2 Choix expérimentaux et résultats

En faisant référence au schéma global de l'algorithme de tatouage implémenté (figure IV.13), il apparaît deux blocs importants pour assurer la robustesse : la sélection des positions d'insertion et le choix des valeurs de QP pour la quantification. La modification de ces paramètres influe sur les résultats obtenus.

3.2.1 Le choix des paramètres de quantification

La modification de la valeur de QP influe sur la qualité de la vidéo tatouée. L'expérience nous a montré que seulement quatre QP [28, 32, 36, 40] sont adéquats pour que l'algorithme converge vers la bonne qualité. La figure IV.18 illustre l'impact de la variation du paramètre de quantification sur la qualité de la marque reconstruite.

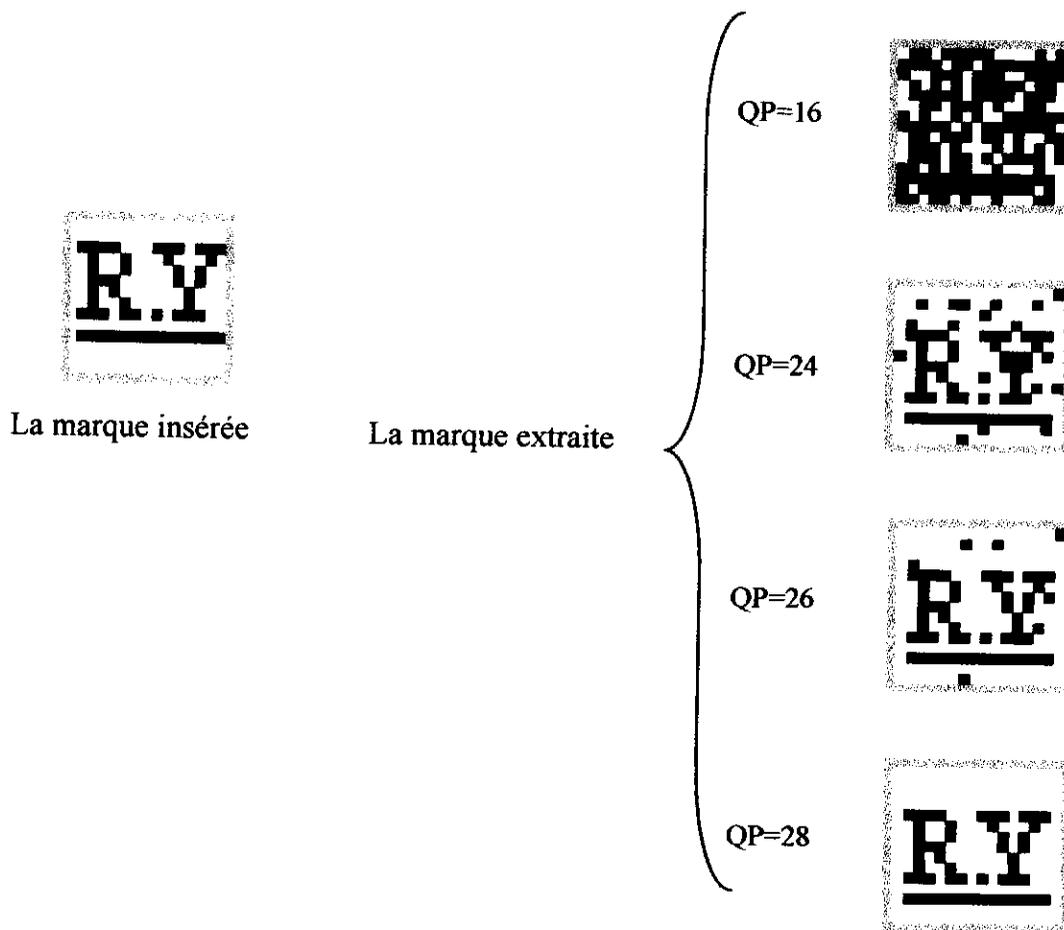
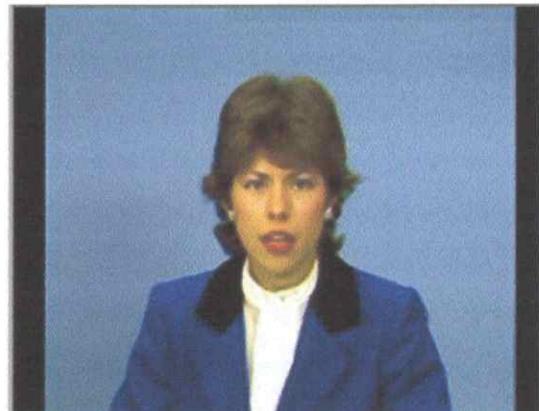


Figure IV .18 La marque extraite pour différentes valeurs de QP

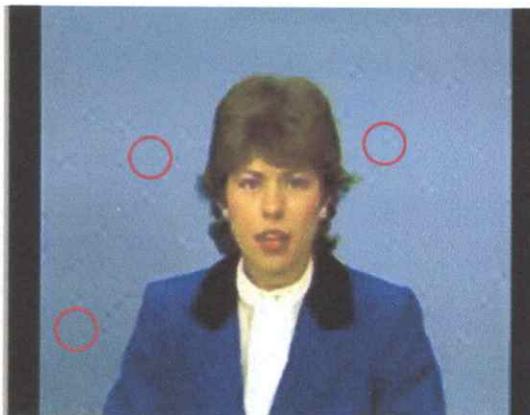
Pour QP inférieur à 28, la marque est bruitée et n'est pas complètement extraite. Au delà de QP =28 la marque est parfaitement extraite, mais si on augmente le QP jusqu'à 45 ou plus, la qualité de la vidéo se dégrade, et le tatouage devient visible (figure IV.19).



Vidéo tatouée avec QP=20
PSNR= 40.38 dB



Vidéo tatouée avec QP=28
PSNR= 40.00 dB



Vidéo tatouée avec QP=35
PSNR= 39.02 dB



Vidéo tatouée avec QP=48
PSNR= 36.28 dB

Figure V.19 Séquence « claire » tatouée par différentes valeurs de QP



Vidéo tatouée avec QP=20
PSNR= 38.29 dB



Vidéo tatouée avec QP=28
PSNR= 37.87 dB



Vidéo tatouée avec QP=35
PSNR= 35.02 dB



Vidéo tatouée avec QP=48
PSNR= 31.57 dB

Figure IV.20 Séquence « container » tatouée par différentes valeurs de QP

3.2.2 Le choix de la position d'insertion

La meilleure position d'insertion dans un bloc se trouve dans les coefficients de moyennes fréquences. Pour définir cette position, nous avons établi des tests pour chaque position dans le bloc en le parcourant en zigzag, des coefficients basses fréquences aux hautes fréquences. Quelques résultats sont représentés sur le tableau suivant :

Tableau IV.5. Résultats d'insertion suivant le parcours en zigzag

Position		Qualité de la vidéo	PSNR (dB)	Taux d'extraction (%)
Basse fréquence	(0,1)	Mauvaise	36,48	100
	(1,0)	Mauvaise	35,26	99.60
Moyenne fréquence	(0,2)	moyenne	41,11	100
	(1,1)	Bonne	41,00	100
	(2,1)	Moyenne	41,03	100
	(3,0)	Moyenne	41,10	99.60
Haute fréquence	(2,3)	Moyenne	41,45	20
	(3,2)	bonne	43,96	98

D'après les résultats présentés dans le tableau ci-dessus on remarque que l'insertion dans les coefficients hautes fréquences est à écarter, car la qualité de la vidéo est mauvaise c'est-à-dire le tatouage est visible. De même pour les basses fréquences où l'extraction de la marque n'est pas à 100%. L'insertion dans les coefficients de moyennes fréquences est adoptée, car l'extraction est toujours proche ou égale à 100%. Donc la meilleure position qui garantit l'invisibilité et l'extraction complète de la marque est la position (1,1).

4. Conclusion

Les deux algorithmes de tatouage vidéo ont été développés pour la protection de copyright dans la récente norme de compression H.264/AVC. Le premier algorithme est dérivé d'une méthode de tatouage des images fixes, alors que le second présente une technique appliquée à la vidéo. Ces deux algorithmes s'appliquent au cours de la compression, l'extraction de la marque s'effectue lors du décodage partiel du flux compressé. Ils appartiennent à la classe des schémas de tatouage aveugles, c'est-à-dire que la vidéo d'origine n'est pas nécessaire. Afin de sécuriser l'information, une séquence pseudo aléatoire est utilisée dans les deux algorithmes.

Les points de différence entre ces deux algorithmes développés sont résumés dans le tableau IV.6 :

- Le premier algorithme consiste à insérer l'information dans les coefficients DCT 4×4. Tandis que dans le deuxième, la marque est insérée dans les coefficients DCT quantifiés.
- La capacité d'insertion pour la première méthode est de 1280 bits, tandis que pour la deuxième seulement 256 bits sont insérés.
- Dans la deuxième méthode en plus de la séquence pseudo aléatoire le calcul d'entropie est introduit pour assurer l'invisibilité.

Tableau IV.6 Tableau comparative des algorithmes développés

Les points de différence	Tatouage basé sur la DCT 4×4	Tatouage basé sur la quantification
Domaine d'insertion de la marque	Domaine de transformer (DCT)	Domaine de quantification
Nombre de bits insérés	1280 bits	256 bits
Type de schéma	Schéma substitutif	Schéma additif
Nombre de bloc marqué	Cinq blocs par macrobloc	un seul bloc par macrobloc
Traitement effectué sur la marque	Hadamard	Quantification
Caractéristique de Marquage	séquence pseudo aléatoire	l'entropie + séquence pseudo aléatoire



Chapitre V
Robustesse face aux attaques

1. Introduction

Afin d'évaluer les performances des deux méthodes de tatouage vidéo développées pour la protection de copyright dans la norme de compression H.264\AVC, trois différentes attaques bienveillantes ont été appliquées :

- **Attaque par compression** : deux normes de compression ont été utilisées pour cette attaque, le codeur H.264/AVC et le codeur H.263.
- **Attaque par transcodage** : le transcodage vidéo convertit un format vidéo en un autre. Pour notre application, nous avons utilisé un convertisseur YUV/RGB comme attaque pour tester la robustesse des deux algorithmes. Les tests par cette attaque sont réalisés grâce au logiciel YUVTools.2.0.40. C'est un ensemble d'outils destiné pour la lecture, la conversion et l'analyse des signaux vidéo de YUV dans d'autres formats tel que RGB.
- **Attaque géométrique** : cette attaque redimensionne la vidéo de taille 176×144 pixels en une dimension 352×288 pixels, cela revient à prendre un pixel et le multiplié par deux. Cette même vidéo est ensuite redimensionnée à sa taille initiale (176×144 pixels) pour pouvoir extraire les informations insérées.

2. Résultats expérimentaux

2.1 Attaque par compression H.264

Les paramètres relatifs à la robustesse sont la qualité de la vidéo attaquée et l'intégrité de la marque extraite. Ces deux paramètres sont expérimentés comme suit :

La compression H.264 a été appliquée sur les séquences « claire » et « container » tatouées par la méthode de DCT 4×4 , puis par la méthode de quantification. Les résultats obtenus lors des tests, montrent un bon comportement de deux méthodes face à cette attaque. La figure V.1 présente la marque extraite de la vidéo « claire » attaquée.

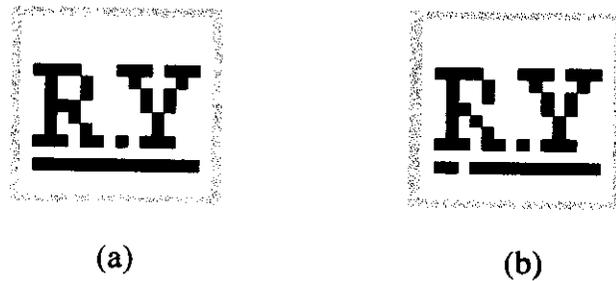


Figure V.1 Marques extraites de la vidéo « claire » : (a) méthode basée sur la DCT 4×4 et (b) méthode basée sur quantification

On a estimé le taux d'extraction de la marque lors du décodage. On rappelle que dans la première méthode, l'insertion est faite dans le 5^{ème} bit MSB du coefficient de position (3,3). D'où la marque est extraite de la même position par un taux égal à 100%. Pour le deuxième algorithme le taux est à 99.21%.

Pour la séquence « container », l'intégrité est à 100% dans la première méthode et à 96.09% dans la deuxième (Figure V.2).

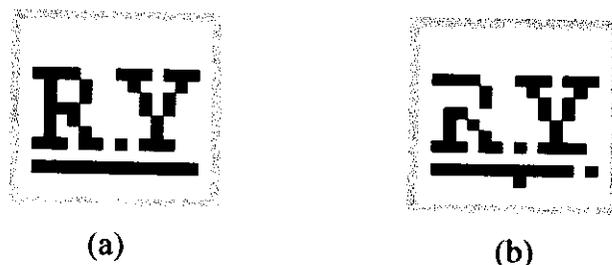
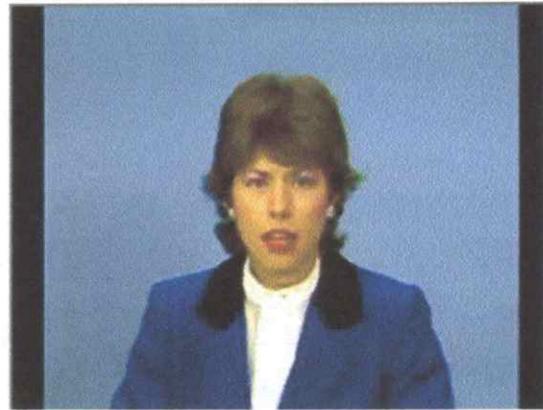


Figure V.2 Marques extraites de la vidéo « container » : (a) méthode basée sur la DCT 4×4 et (b) méthode basée sur la quantification

D'après les résultats présentés, nous pouvons conclure que les deux algorithmes sont robustes face à l'attaque par la norme de compression H.264 avec une bonne qualité visuelle des vidéos attaquées. Les figures V.3 et V.4 présentent respectivement les résultats de la qualité subjective des séquences « claire » et « container ».



Vidéo tatouée basée sur la méthode DCT 4×4



Vidéo tatouée basée sur la méthode DCT 4×4 et attaquée par le codeur H.264



Vidéo tatouée basée sur la méthode de quantification



Vidéo tatouée basée sur la méthode de quantification et attaquée par le codeur H.264

Figure V.3 Qualité visuelle de la séquence « claire » attaquée par le codeur H.264



Vidéo tatouée basée sur la méthode DCT 4×4



Vidéo tatouée basée sur la méthode DCT 4×4 et attaquée par le codeur H.264



Vidéo tatouée basée sur la méthode de quantification



Vidéo tatouée basée sur la méthode de quantification et attaquée par le codeur H.264

Figure V.4 Qualité visuelle de la séquence « container » attaquée par le codeur H.264

2.1 Attaque par compression H.263

D'après les expériences effectuées, on remarque que l'attaque par le codeur H.263 dégrade la vidéo tatouée. Et le phénomène d'artefact (effet de blocs) apparaît. La question qui se pose : est ce que la marque est extraite complètement ou non ? La réponse à cette question est démontrée dans ce qui suit :

Méthode basée sur la DCT 4×4 : après l'extraction de la séquence binaire insérée, le processus de reconstruction de la marque regroupe chaque cinq bits successifs pour appliquer la transformation inverse de Hadamard 4×4. Si un seul bit manque, un bloc 4×4 de la marque est dégradé. Pour la vidéo « claire » attaquée, La marque est extraite par un taux de 21.87% et pour container à 28.90% (Figure V.5), donc la marque est complètement détruite. La qualité visuelle est illustrée par la figure V.6 pour la séquence « claire » et la figure V.7 pour la séquence « container ».

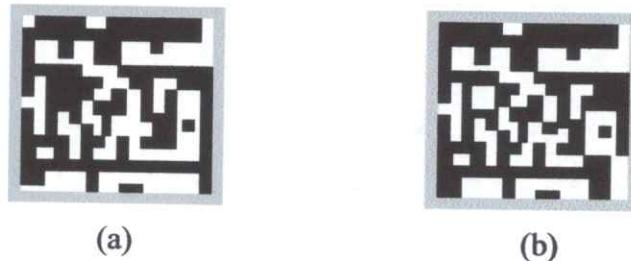


Figure V.5 Marques extraites des vidéos basée sur la DCT 4×4 et attaquée par le codeur H.263 : (a) de la séquence « claire » et (b) de la séquence « container »



Séquence « claire » tatouée

Séquence « claire » attaquée

Figure V.6 Qualité visuelle de la séquence « claire » attaquée par le codeur H.263

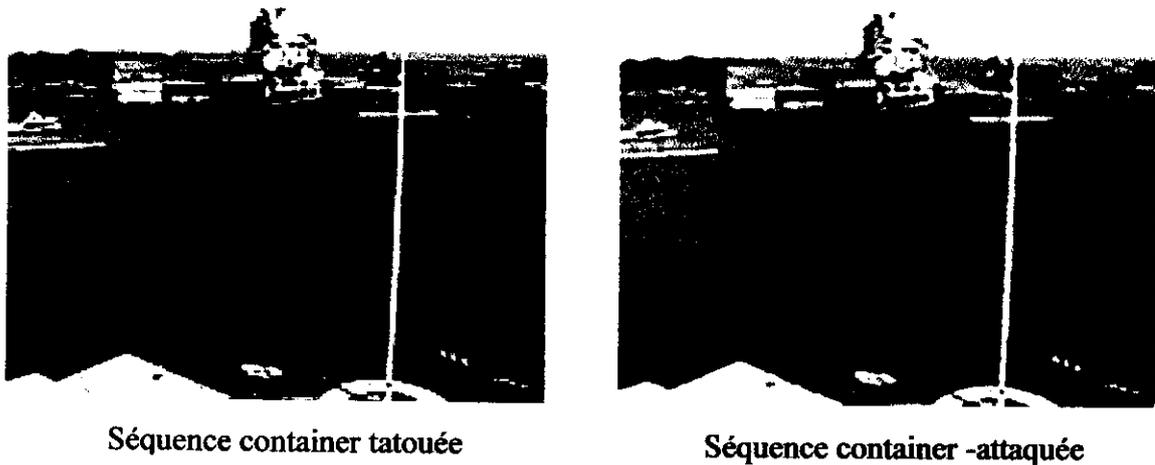


Figure V.7 Qualité visuelle de la séquence « container » attaquée par le codeur H.263

Méthode de quantification : La forte dégradation engendrée par la compression H.263 sur les vidéos, influe sur l'intégrité de la marque. La marque extraite des vidéos tatouées par la méthode de quantification est aussi détruite : 35.15 % pour « claire » et 44.53 % pour « container » (Figure V.8). La qualité visuelle est illustrée par la figure V.9 et figure V.10.

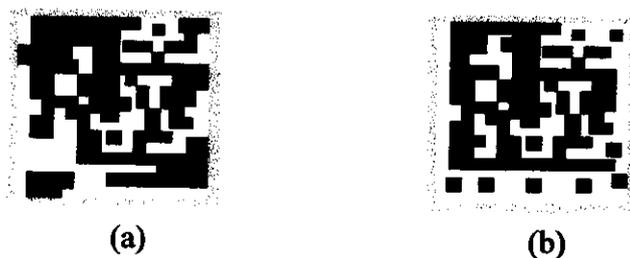


Figure V.8 Marque extraite après l'attaque par le codeur H.263 : (a) pour « claire » et (b) pour « container »



Séquence « claire » tatouée



Séquence « claire » attaquée

Figure V.9 Qualité visuelle après l'attaque par H.263 de la séquence « claire » tatouée par la deuxième méthode



Séquence « container » tatouée



Séquence « container » attaquée

Figure V.10 Qualité visuelle après l'attaque par H.263 de la séquence « container » tatouée par la deuxième méthode

D'après les résultats obtenus, on peut conclure que les deux algorithmes ne sont pas robustes face à l'attaque par la compression vidéo H.263. Même si l'algorithme de quantification présente une résistance par rapport à celui de DCT 4×4, ils restent faibles face à cette attaque qui perd une partie de la valeur commerciale du signal vidéo.

2.3 Attaque par transcodage YUV/RGB

Nous avons utilisé le logiciel de transcodage YUVTools.2.0.40 pour la séquence vidéo « claire » tatouée par les deux méthodes. Ce logiciel convertit ces vidéos de format YUV (4 :2 :0) vers un format RGB (4 :4 :4). Il les convertit ensuite au format d'origine (YUV) pour que l'information insérée soit extraite. Les deux marques extraites respectivement par les deux méthodes appliquées sur la séquence « claire » sont montrées sur la figure V.11.

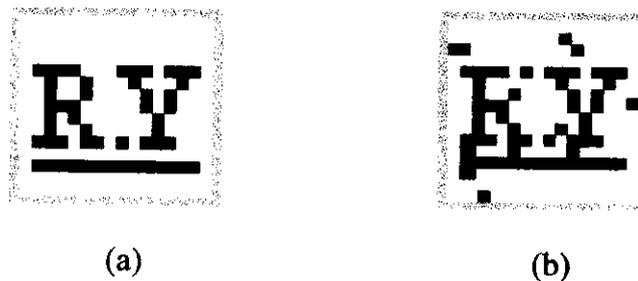


Figure V.11 Marques extraites de la vidéo « claire » après le transcodage: (a) basée sur la méthode DCT 4×4 et (b) basée sur la méthode de quantification

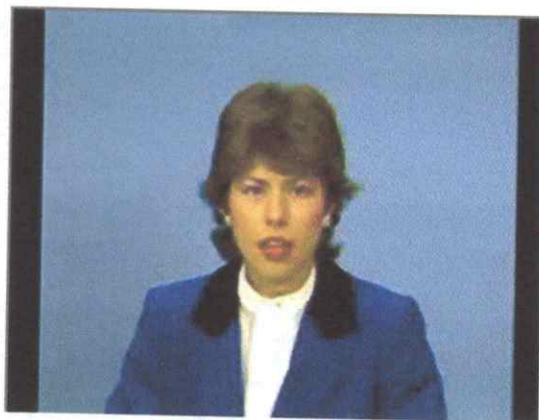
D'après ces résultats, les taux d'intégrité de la marque dans les deux vidéos « claire » attaquées sont estimés à 100% pour le premier algorithme et 92.96% pour le deuxième.

Les mêmes tests ont été appliqués sur les deux vidéos tatouées « container », les marques extraites sont présentées sur la figure V.12 suivante.



Figure V.12 Marques extraites de la vidéo « container » après le transcodage : (a) basée sur la méthode de DCT 4×4 et (b) basée sur la méthode de quantification

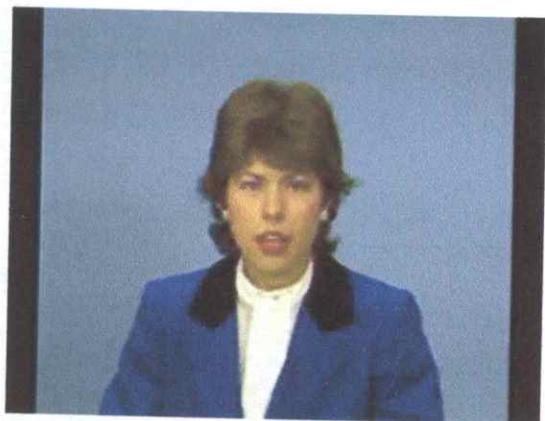
Des résultats obtenus, on remarque que la marque n'est pas complètement extraite dans les deux vidéos. Ils restent quelques bits non récupérés. Donc le taux d'extraction de la vidéo attaquée basée sur la première méthode est de 92.96 % tandis que pour la deuxième méthode, il est de 96.87%. Les figures V.13 et V.14 illustrent respectivement la qualité subjective des deux vidéos « claire » et « container » après le transcodage.



Vidéo tatouée au niveau de
DCT 4×4



Vidéo attaquée basée sur la méthode
de DCT 4×4



Vidéo tatouée au niveau des coefficients
transformés quantifiés



Vidéo attaquée basée sur la méthode
de quantification

Figure V.13 Qualité visuelle des vidéos « claire » face à l'attaque de transcodage



Vidéo tatouée au niveau de DCT
4×4



Vidéo attaquée basée sur la méthode
de DCT 4×4



Vidéo tatouée au niveau des coefficients
transformés quantifiés

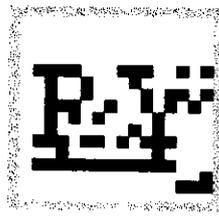


Vidéo attaquée basée sur la méthode
de quantification

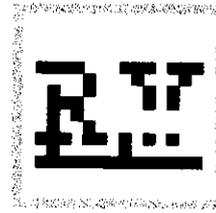
Figure V.14 Qualité visuelle des vidéos « container » face à l'attaque de transcodage

2.4 Attaque par redimensionnement

Nous avons utilisé le logiciel YUVTools.2.0.40 pour redimensionner les vidéos tatouées par les deux méthodes, de format QCIF (176×144 pixels) en un format CIF (352×288 pixels). Puis nous les redimensionnons au format d'origine pour extraire la marque. Les taux d'extraction obtenus de la première méthode sont : 85.54% pour « claire » et 93.35% pour « container ». La figure V.15 montre les marques extraites.



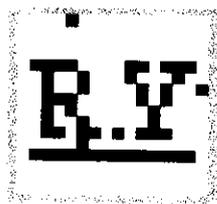
Marque extraite de
« claire »



Marque extraite de
« container »

Figure V.15 Marque extraite après l'attaque de redimensionnement de la vidéo tatouée par la première méthode

Pour la deuxième méthode, l'intégrité de la marque extraite est estimée à 96.87 % pour la vidéo « claire », et à 94.14% pour la vidéo « container ». Les deux marques sont sur la figure V.16.



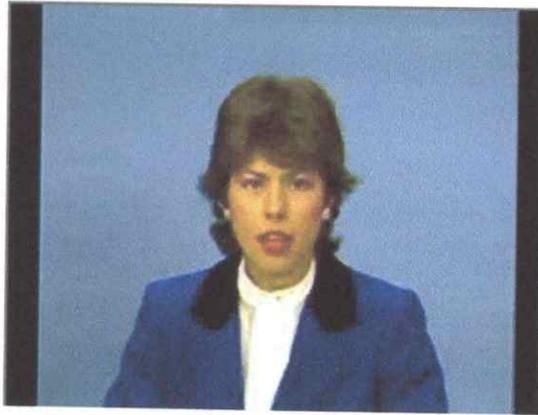
Marque extraite de
« claire »



Marque extraite de
« container »

Figure V.16 Marque extraite après l'attaque de redimensionnement de la vidéo tatouée par la deuxième méthode

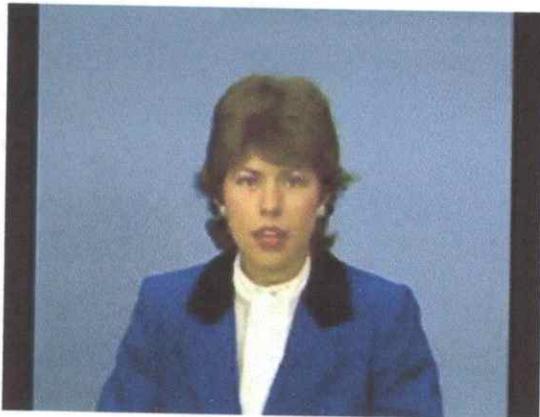
L'attaque par le redimensionnement ne dégrade pas la qualité visuelle, (Figures V.17 et V.18). La totalité de la marque insérée est récupérée sauf quelques bits sont détruits dans les deux méthodes.



Vidéo tatouée au niveau de
DCT 4×4



Vidéo attaquée basée sur la méthode
de DCT 4×4



Vidéo tatouée au niveau des coefficients
transformés quantifiés



Vidéo attaquée basée sur la méthode
de quantification

Figure V.17 Qualité visuelle des vidéos « claire » face à l'attaque de redimensionnement



Vidéo tatouée au niveau de DCT
4×4



Vidéo attaquée basée sur la méthode
de DCT 4×4



Vidéo tatouée au niveau des coefficients
transformés quantifiés



Vidéo attaquée basée sur la méthode
de quantification

Figure V.18 Qualité visuelle des vidéos « container » face à l'attaque de redimensionnement

3. Étude comparative des deux algorithmes

Les algorithmes de tatouage vidéo développés pour la protection de copyright dans la norme de compression H.264/AVC sont robustes face à l'attaque par codeur H.264 et l'attaque par transcodage. Par contre ils ne le sont pas face à la compression H.263. Pour l'attaque par redimensionnement, la deuxième méthode est robuste, alors que pour la première la robustesse a diminué.

Les résultats obtenus par les deux méthodes sont regroupés sur le tableau V.1 pour la séquence « claire » et le tableau V.2 pour « container ».

Tableau V.1 Résultats de tatouage de la séquence « claire » par les deux algorithmes

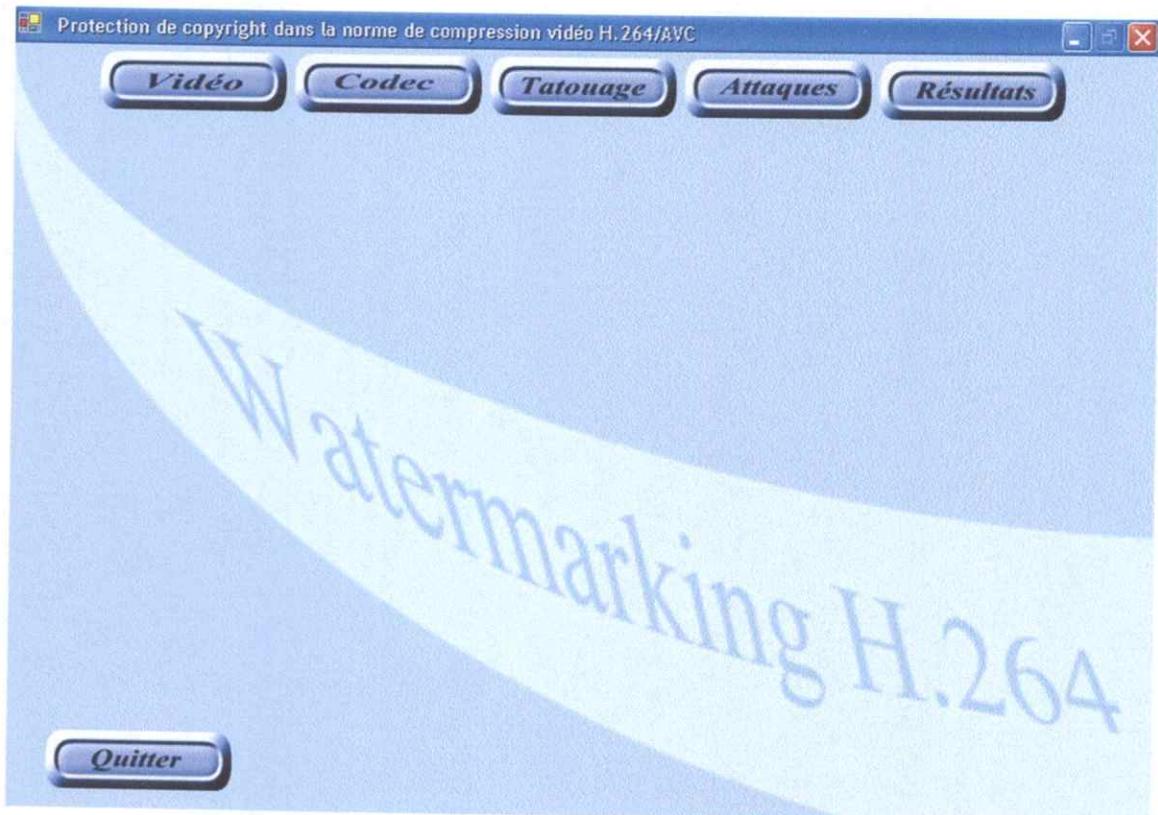
Paramètres	Algorithme basé sur la DCT 4×4						Algorithme basé sur la quantification											
	Compression	Tatouage	Attaques			Compression	Tatouage	Attaques			Compression	Tatouage	Attaques					
			H.264	H.263	Trans			Redim	H.264	H.263			Trans	Redim	H.264	H.263	Trans	Redim
PSNR (dB)	41,29	40.38	39.94	35.44	38.53	37.84	41.29	41.00	39.47	35.69	38.06	38.62						
Taux d'extraction (%)	-	100	100	21.87	100	85.54	-	100	99.21	35.15	92.96	96.87						
Nombre de bits récupérés	-	256	256	56	256	219	-	256	255	90	238	248						

Tableau V.2 Résultats de tatouage de la séquence « container » par les deux algorithmes

Paramètres	Algorithme basé sur la DCT 4×4						Algorithme basé sur la quantification											
	Compression	Tatouage	Attaques			Compression	Tatouage	Attaques			Compression	Tatouage	Attaques					
			H.264	H.263	Trans			Redim	H.264	H.263			Trans	Redim	H.264	H.263	Trans	Redim
PSNR (dB)	37.94	37.38	37.33	32.03	36.45	37.06	37.94	37.88	37.65	32.09	37.58	37.36						
Taux d'extraction (%)	-	100	100	28.90	92.96	93.35	-	100	96.87	44.53	96.87	94.14						
Nombre de bits récupérés	-	256	256	74	231	239	-	256	248	114	248	241						

4. Présentation du logiciel

Nous allons présenter la fenêtre principale de l'application et les différents outils permettant la communication avec le logiciel. La figure (a) montre la structure de la fenêtre principale.

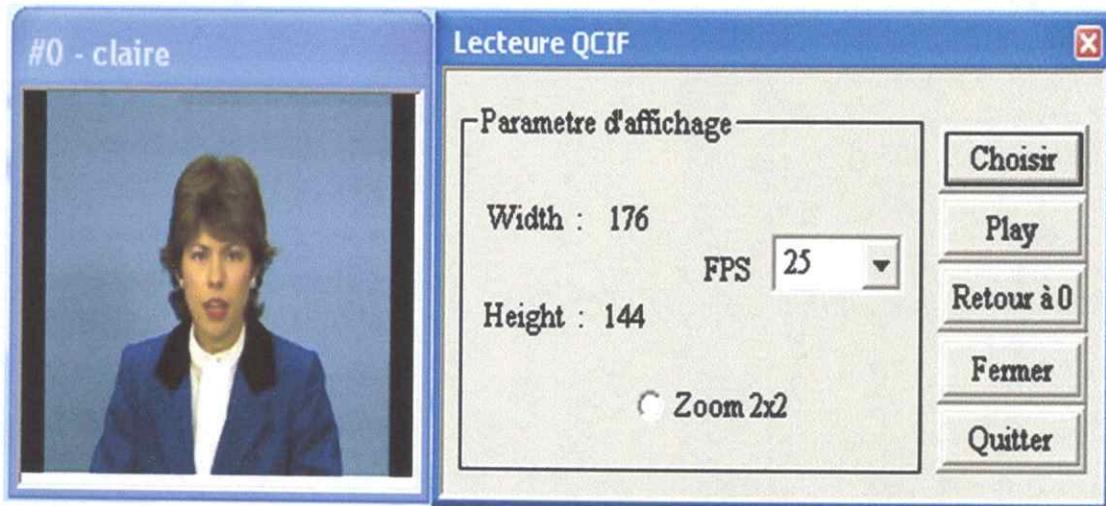


(a)

La barre des boutons apparaît sous forme de barre horizontale, elle est constituée de :

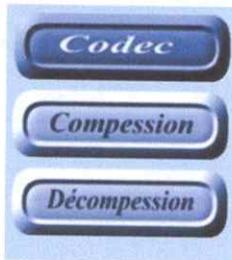
- **Le bouton « Vidéo »**

Permet de visualiser une vidéo en format QCIF comme le montre la figure (b).



(b)

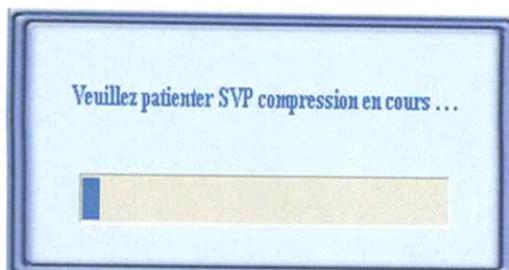
- Le bouton « Codec »



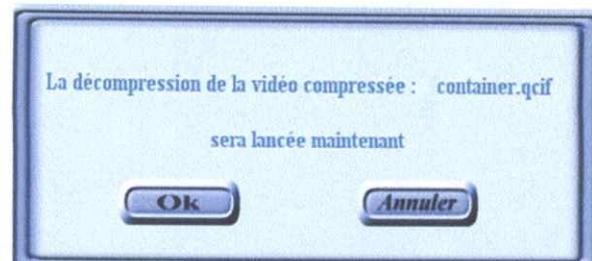
Permet :

- Une compression par la norme H.264.
- Une décompression par la norme H.264.

Pour la compression une fenêtre apparaît permet de choisir la vidéo à compresser. Lorsque le choix est fait, la compression H.264 s'exécute. La figure (c) représente une 'Wait bar' indiquant que la compression est en cours. La décompression est effectuée sur la vidéo compressée, elle est indiquée par la fenêtre (d).



(c)



(d)

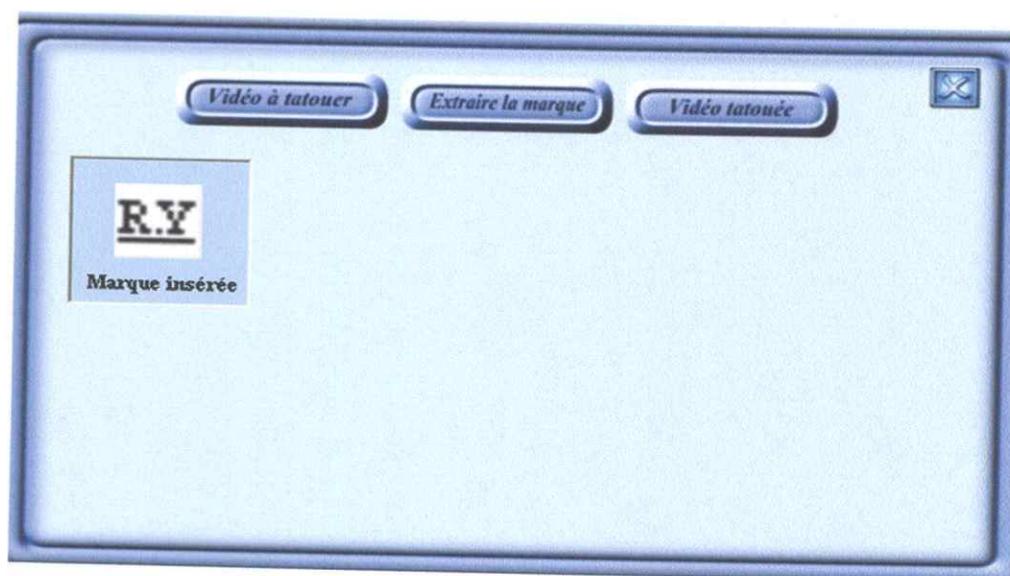
- Le bouton « Tatouage »



Il permet :

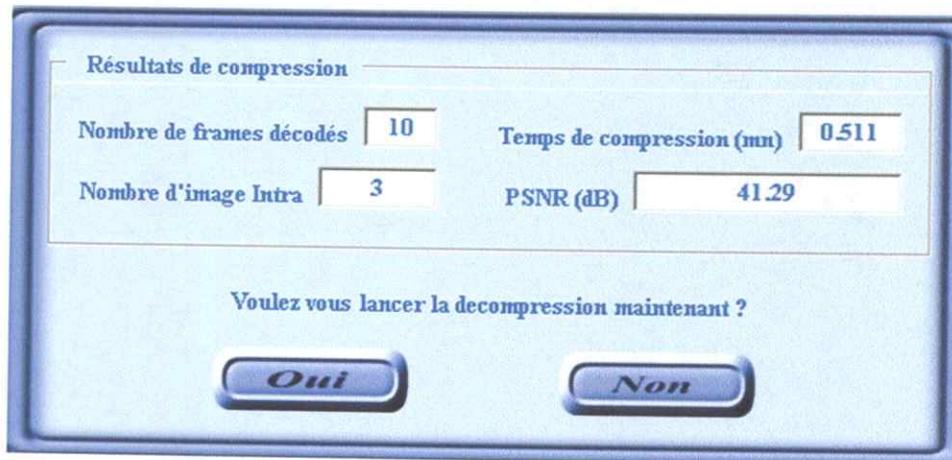
- Un tatouage basé sur la méthode de DCT 4×4.
- Un tatouage basé sur la méthode de coefficients quantifiés.

Pour chaque méthode de tatouage, la marque à insérer et la vidéo à tatouer sont des choix de l'utilisateur (fenêtre (e)).



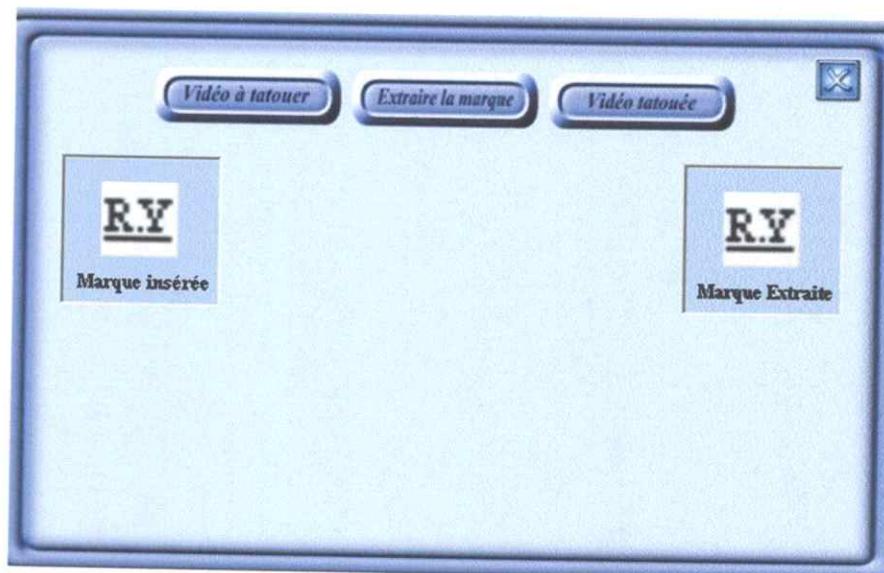
(e)

Lorsque la compression avec tatouage commence, une fenêtre 'wait bar' apparaît. Les résultats de compression apparaissent sur la fenêtre (f) à la fin de compression.



(f)

Pour l'extraction de la marque, la décompression de la vidéo tatouée est nécessaire. L'extraction de la signature est sur la figure (g).

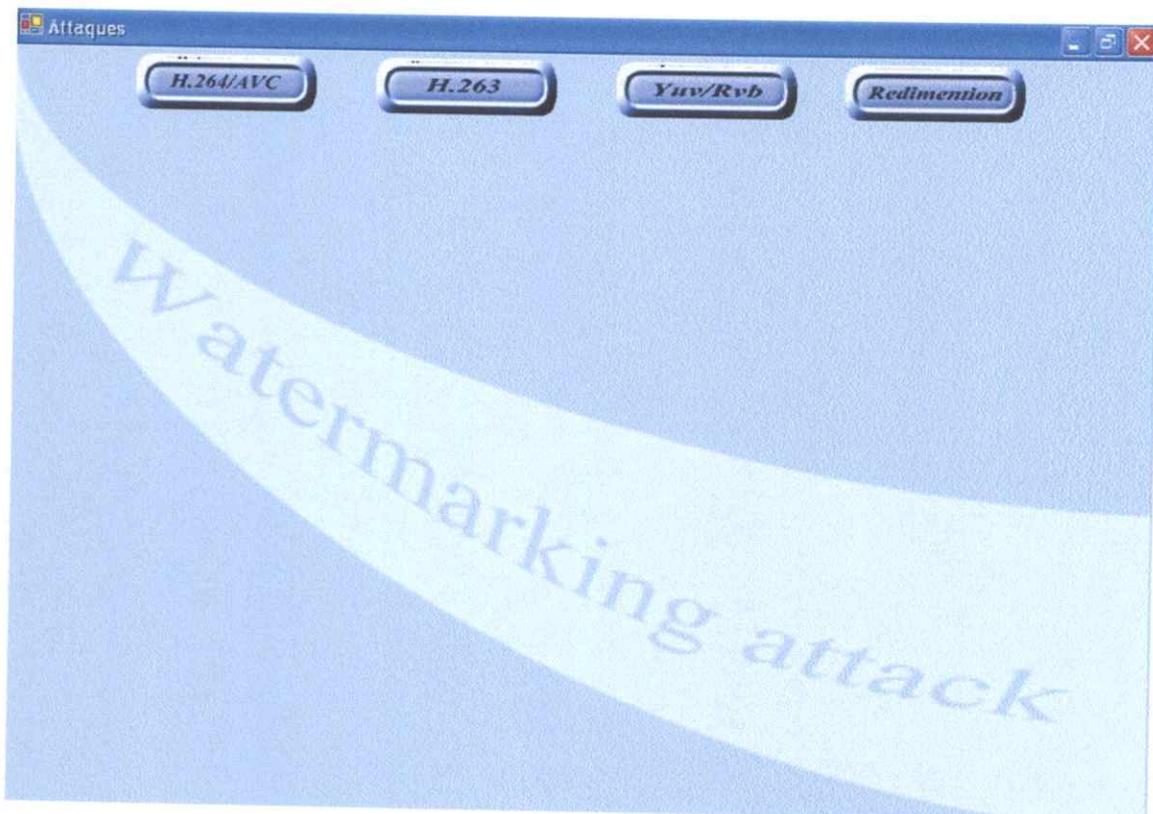


(g)

- Le bouton « Attaque »

Permet de faire :

- Une attaque par Le codeur H.264.
- Une attaque par le codeur H.263.
- Une attaque par le transcodage.
- Une attaque par le redimensionnement.



(h)

Chaque bouton de cette fenêtre (fenêtre (h)) exécute l'attaque qui l'indique sur la vidéo tatouée choisie par l'utilisateur. Pour l'attaque RVB/YUV et le redimensionnement, le logiciel YUVTools.2.0.40 est utilisé.

- Le bouton « Résultats »

Il regroupe tous les résultats (PSNR, taux d'extraction) obtenus par la compression, le tatouage et les attaques des vidéos.



**Conclusion générale et
perspectives**

Conclusion générale et perspectives

Le tatouage représente une alternative à la protection de la propriété intellectuelle des supports multimédia courant. Au début, le tatouage apparaissait comme étant la solution à tous les problèmes de protection de contenus face aux multiples attaques, la communauté de tatouage s'est aperçue progressivement que, créer un algorithme de tatouage assurant une protection absolue représentait une mission impossible.

En effet, nous savons aujourd'hui que le tatouage seul ne peut pas répondre à une protection suffisamment fiable dans un milieu grand public où les degrés de liberté en terme de manipulation des contenus sont trop élevés. C'est pourquoi beaucoup d'industriels s'orientent vers l'élaboration de solutions spécifiques dépendant de l'application visée, essentiellement destinées au domaine professionnel. Il semble de plus en plus évident que l'on ne peut jamais empêcher le piratage grand public, à moins de créer des systèmes exclusivement propriétaires, mais dans ce cas il n'est absolument pas assuré que la protection soit efficace.

La norme H.264/AVC, toute récente norme de compression vidéo nécessite la protection de son contenu pour différentes applications telles que : copyright, authentification, fingerprint... etc.

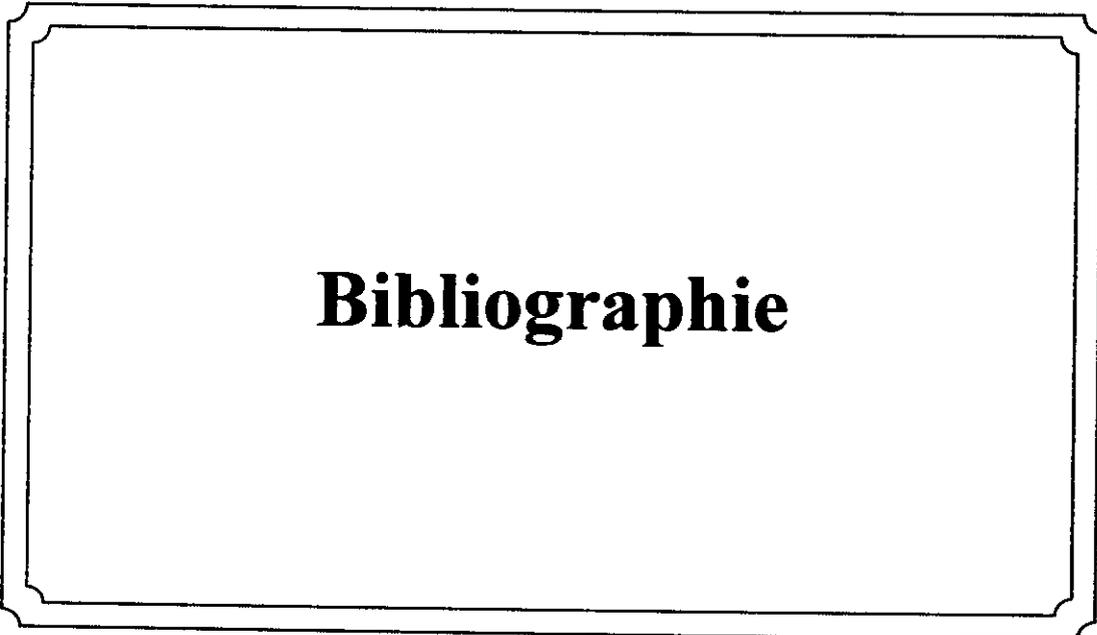
Nous nous sommes intéressées dans ce mémoire à la protection du contenu dans la norme H.264 pour la protection de copyright. Deux algorithmes ont été développés. L'un est basé sur la DCT 4x4 et le second sur les coefficients quantifiés. Pour les deux méthodes, la marque a été respectivement traitée par la transformation de Hadamard et la quantification. Nous souhaitons à travers ce modeste travail avoir contribué au développement d'un domaine aussi promoteur qui est la protection du contenu vidéo.

Toutefois, ce travail peut être amélioré en envisageant les perspectives suivantes :

- En premier lieu, il est intéressant d'approfondir l'étude de formation de la marque à insérer et notamment d'étudier les codes correcteurs d'erreur (BCH), qui consiste à introduire une redondance au niveau de la marque, tout en considérant

l'image comme un support de transmission et la marque l'information utile à transmettre donc à protéger.

- Il serait nécessaire d'évaluer les performances des algorithmes développés face à d'autre type d'attaques.



Bibliographie

Bibliographie

- [1]. P.Bas & J.Chassery, "Tatouage d'images résistant aux formation géométriques," Université de France, 1999.
- [2]. Yann BODO, "Elaboration d'une technique d'accès conditionnel par tatouage et embrouillage vidéo basée sur la perturbation des vecteurs de mouvement," thèse de doctorat de l'école national supérieur des télécommunications, Sept 2004.
- [3]. M. Kutter & F.A.P. Petitcolas. "A fair benchmark for image watermarking systems," Electronic Imaging 99, Security and Watermarking of Multimedia Contents, San Jose, CA, USA, vol. 3657, 1999.
- [4]. Alan C. Bovik, D. Jerry. Gibson & Al Bovik, "Handbook of Image and Video Processing," Academic Press, USA, 2000.
- [5]. P. Meuel, "Protection et enrichissement de régions d'intérêts dans des séquences vidéos," Thèse de master, Université Montpellier II, Juin 2006.
- [6]. M. Ghanbari, "Standard Codec: Image Compression to Advanced Video Coding," IAEFE Journal. Institution Electrical Engineers, 2003.
- [7]. F. Loras & J. Fournier, "H.264/MPEG-4 AVC, un nouveau standard de compression vidéo," Projet de Laboratoire de Service de visiophonie de Groupe et de Visioconférence, France, 2003.
- [8]. C. K. Armand, A. El maghnouji & Z. Y. Kane, " Le codage de MPEG," Document de EPITA Masters, version 1.0, pp. 11, Octobre 2004.
- [9]. Standard MPEG-4: "Information Technology – Coding of Audio-Visual Objects," ISO/IEC journal, vol. 14496-2, 2002.
- [10]. Stéphane Péchard, " Utilisation de modèles psychovisuels pour le contrôle de la prédiction spatiale de la norme de compression vidéo H.264/AVC," Thèse doctoral, Ecole polytechnique-Nantes, Avril 2004.

-
- [11]. R. Schäfer, T. Wiegand & H. Schwarz, "H.264 / AVC la norme qui monte," UER–Revue Technique–Sélection, Institut Heinrich Hertz, Berlin, 2003.
- [12]. M.G. Xavir ; "Optimisation des performances d'un encodeur suivant la norme Advanced Video Coding pour une machine vectorielle," Thèse magistère, Université libre de Bruxelles, 2006.
- [13]. A. Marie Andrée, " Etude du codage des vecteurs de mouvement pour les codeurs vidéo de nouvelle génération," Thèse de Master, Université Nice Sophia Antipolis, 2005.
- [14]. J. Sullivan, P. Topiwala & A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," H.264/AVC, August 2004.
- [15]. O. Rioul, " Codage entropique à longueur variable," ENST/COMELEC journal, 2003.
- [16]. D. Marpe, H. Schwarz, & T. Wiegand: "Context-Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," in IEEE Transactions on Circuits and Systems for Video Technology, this issue, July 2003.
- [17]. J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, & T. Wedi, "Video coding with H.264/AVC: Tools, Performance, and Complexity," IEEE Circuits and Systems Magazine, pp. 15-16. First quarter 2004.
- [18]. ISO/IEC & ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services," vol. 14496–10, 2003.
- [19]. C. T. Hsu, J.L. Wu, "Digital watermarking for video," DSP'97 Santoniri project, Juil 97.
- [20]. J. Dittman, M. Stabenau, & R. Steinmetz, "Robust mpeg video watermarking technologies," ACM Multimedia project, 1998.

-
- [21]. J. Fridrich. "Methods for data hiding," Center for Intelligent Systems & Department of Systems Science and Industrial Engineering, 1997.
- [22]. J. J. Chae, B. S. Manjunath, "Data hiding in video," IEEE International Conference on Image Processing Kobe- Japan, pp. 11-15 , vol.1, Oct 1999.
- [23]. Swanson, M. Zhu, B, Tewfik, "Multiresolution Scene-based Video Watermarking Using Perceptual Models," IEEE journal on selected areas in communications, vol. 16, no. 4 , 1998.
- [24]. N. Checcacci, M. Barni, F. Bartolini, & S. Basagni, "Robust video watermarking for wireless multimedia communications," Proceedings IEEE Wireless Communications and Networking Conference, pp 1530–1535, vol 3, 2000.
- [25]. G.C. Langelaar, J.C.A. vander Lub, & R. L. Lagendijk, "Robust labeling methods for copy protection of images," Electronic Imaging, San Jose, CA, vol. 3022, pp 298-309, Feb. 97.
- [26]. K. Matsui, K. Tanaka., "Video-steganography: how to secretly embed a signature in a picture," the interactive Multimedia Association Intellectual Property Project, pp. 187-206, 1994.
- [27]. F. Hartung, B. Girod, "Watermarking of uncompressed and compressed video; IEEE transaction on Signal processing," pp 66-283–301, 1998.
- [28]. E .Koch, J. Zhao, "Towards Robust and Hiden Image Copyright Labeling," IEEE Workshop on Nonlinear Signal and Image Processing , pp. 452-455, Grec, 1995.
- [29]. M. Kutter, F. Jordan, & T. Ebrahimi, "Proposal of a watermarking technique for hiding/retrieving data in compressed and decompressed video," JTC1/SC29/WG11 M2281-ISO/IEC journal, 1997.
- [30]. J. Zhang, J. Li, and L. Zhang, "Video watermarking technique in motion vector. Proceedings of XIV Brazilian Symposium on Computer Graphics and Image Processing," pp 179–182, 2001.

-
- [31]. L. Rajab, S. Sarhan & M. Habib, "Digital watermarking System in the DCT and Hadamard Domain for the Proof of Ownership," University of Jordan, pp. 1-4, 2005.
- [32]. J. Peter, "Hadamard matrices," The Encyclopaedia of Design Theory, Cameron, June 2006.
- [33]. L. Aurélien & L. Guillaume, "Génération de nombres aléatoires par numérisation d'émissions radiatives," Université de Limoges, pp 5-6, 2003.
- [34]. M. Miller, G. Doërr, and I. Cox., "Applying informed coding and informed embedding to design a robust, high capacity watermark," IEEE Image Processing, vol. 13, pp. 792–807, Jun 2004.
- [35]. V.Boris, "Technique multicouches pour le tatouage d'image et adaptation aux flux video MPEG-2 et MPEG-4, " thèse de doctorat Ecole National Polytechnique d'Alger, Nov 2003.



Sites Web

- [sw1]. [fr.wikipedia.org / wiki / H.264](http://fr.wikipedia.org/wiki/H.264)
- [sw2]. Iain E G Richardson, "H.264 / MPEG-4 Part 10 : Inter Prediction," www.vcodex.com
- [sw3]. Iain E G Richardson, "H.264 / MPEG-4 Part 10: Intra Prediction," www.vcodex.com
- [sw4]. Iain E G Richardson, "H.264 / MPEG-4 Part 10: Transform & Quantization," www.vcodex.com
- [sw5]. JVT (2004). Joint Video Team (JVT), reference software JM7.5c (codec de reference). <http://bs.hhi.de/~suehring/tml/download/jm75a.zip>.
- [sw6]. Iain E G Richardson, "H.264 / MPEG-4 Part 10: Variable Length Coding," www.vcodex.com.
- [sw7]. http://iphome.hhi.de/suehring/tml/download/old_jm/

