REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

École Nationale Polytechnique

Département d'Électronique

Projet de fin d'études

en vue de l'obtention du diplôme
d'Ingénieur en Électronique

Thème

Watermarking Vidéo

Encadré par :
Mme L HAMAMI
Mr. H.M. KHELALFA

Étudié par :
Mr. Farid ABTROUN
Mr. Mohsen HADDAD

Promotion : juin 2001

Watermarking Vidéo

ملخص

إنّ تطوّر التكنولوجيا يسمح القيام بنسخ رقميّة ذو جودة مماثلة للأصل للمنتجات السينمائية و التلفزيّة. لذا ظهرت ضرورة حماية هذه المنتجات باستعمال علامات غير ظاهرة ("Watermarking") تسمح بمعرفة ذوي الحقوق لحماية الملكيّة الفكريّة. توجد ميادين استعمال أخرى ل "Watermarking" من بينها التلفزة غير المجّانيّة لمعرفة حقوق الزبون و كذلك إمكانيّة تتبّع المنتوج على شبكة .الانترنيت مثلا.

قمنا في هذا البحث بعرض بعض الطرق المستعملة في ميدان الحماية و التي ساعدتنا على إيجاد طريقة جديدة مقاومة لهجومات تهدف إلى إتلاف العلامة. الطريقة تستعمل تقنيّة نشر الطيف و التحويل التجيبي غير المستمر. العلامة في تدمج معاملات التردد السّفلى.

Résumé

Le développement de la technologie a permis l'apparition d'un nombre de plus en plus important de copies numériques illicites de vidéo de même qualité que l'original. Le besoin de protection s'est alors fait ressentir, d'où l'apparition du watermarking vidéo qui utilise des marques transparentes afin d'identifier les auteurs et ayants droit. D'autres applications sont envisageables, parmi elles on pourra citer la télévision payante et la possibilité de tracer une vidéo sur internet.

Nous avons exposé dans ce mémoire différentes méthodes de watermarking vidéo qui nous ont permis de développer une nouvelle méthode robuste aux attaques visant à détériorer la marque. La méthode développée utilise l'étalement du spectre et la DCT (Discrete Cosine Transform). L'insertion de la marque se fait au niveau des coefficients DCT de basses fréquences.

Abstract

Digital video can bring consumers supreme picture quality. But it also allows unauthorized copies to be made with the same high quality. To combat this, a method to check the origin and copyright status of the content is desired. Adding a digital signature, or a watermark, to the content is a possible answer, provided that it cannot be removed or modified. Other applications are pay per view video broadcast and the possibility of identifying and tracing video data.

In this work we expose some video watermarking methods that helped us to develop a new robust method used for fingerprinting. The developped method uses the spread spectrum technic and the DCT The label is embeded in low DCT coefficients.

Si peu fait ; tant à faire

Cecil J. RHODES.

Merci à tous

Mme HAMAMI, M KHELALFA, les membres du jury, nos familles et nos amis. Merci à tous ceux qui nous ont aidés, ceux qui nous ont soutenus et ceux qui pensent à nous.

SOMMAIRE

INTRODUCTION

Avec le développement de la diffusion des données numériques, et entre autre des vidéos, la protection du copyright est devenue plus importante car la copie est de même qualité que l'originale. Une méthode de protection du copyright consiste en l'insertion d'un watermark au niveau du signal vidéo. Le watermark est un code numérique dissimulé dans la vidéo qui indique les ayants droit. Si le watermark est appliqué à des copies individuelles, il pourra indiquer l'identité des détenteurs de chaque copie. Si des copies illégales apparaissent, le watermark dissimulé permettra de retrouver l'origine des copies.

Ce travail fait l'objet de notre projet de fin d'études, il s'insère dans un projet au niveau du laboratoire des logiciels de base du Centre de Recherche sur l'Information Scientifique et Technique (CERIST). Il consiste en une étude des différentes méthodes du watermarking vidéo et au design d'une nouvelle méthode dans le but d'insérer un fingerprint (empreinte numérique) Dans notre application nous nous intéresserons à tracer les distributeurs et/ou possesseurs agréés d'une séquence vidéo au format MPEG-1 en insérant différents labels constituant le watermark pour différents distributeurs ou possesseurs des séquences vidéos.

Sachant que le watermarking vidéo est un domaine nouveau dans le monde et qu'à notre connaissance il n'a pas encore été traité en Algérie, nous nous sommes attardés sur la partie théorique afin que le lecteur puisse avoir les notions de base pour une meilleure compréhension de la méthode développée. Notre but est aussi de constituer une référence pour des travaux futurs.

C'est ainsi que l'on a débuté avec un historique et des généralités au chapitre 1, suivis d'une étude des attaques possibles sur les watermarks au chapitre 2. Le chapitre 3 introduit le spread spectrum, principe de base sur lequel reposent beaucoup de méthodes (dont la notre). La partie traitant du système visuel humain, en § 3.2., donnera un descriptif des propriétés de ce dernier exploitées dans notre méthode. Le chapitre 4 décrit le format de compression MPEG, format de compression le plus utilisé dans la distribution de séquences vidéos. Au chapitre 5 nous présentons un bref exposé des méthodes nous ayant permis de développer notre méthode exposée au chapitre 7. Le chapitre 6 explique le choix de la DCT (Discrete Cosine Transform) comme transformation de domaine. Les résultats et tests de robustesse sont donnés dans le chapitre 8.

La conclusion clôturant ce mémoire propose des améliorations de la méthode.

Chapitre 1

GENERALITES

- **Historique**
- **Qu'est ce que le Watermarking ?**
- **Classification des systèmes de Watermarking**
- **Applications**
- **Modélisation de l'insertion d'un watermark**

GENERALITES

Ce chapitre traite des généralités sur le watermarking et le «data hiding » (dissimulation de données). Il retrace l'historique de l'évolution du « data hiding », suivit d'une définition du watermarking et des applications de ce denier. Il sera clôturé par une modélisation de l'insertion d'un watermark pour introduire un « fingerprint » ou numéro d'identification.

1.1. HISTORIQUE [1]

A partir du 16^{e} siècle, une littérature de plus en plus abondante a traité du data hiding en général, et la stéganographie en particulier. Dans son livre ***Scholasteganographica,*** **Schott (1608-1666)** expliquait comment dissimuler un message en utilisant comme couverture, ou stego-objet, des tablatures musicales. De plus, il a amélioré le code « ***Ave Maria »*** proposé par **Trithemius (1462-1516)** dans son livre ***Steganograhioe.*** Ce code contenait 40 tables, dont chacune comprenait 24 entrées en quatre langues : Latin, Français, Allemand et Italien. Chaque lettre du texte était remplacée par le mot ou la phrase qui apparaissait dans la table correspondante. Une autre méthode se basant sur le nombre d'occurrences des notes musicales était utilisée par **Bach. Wilkins (1614-1672),** dans le ***master of trinity college,*** a démontré comment deux musiciens pouvaient converser en utilisant leurs instruments musicaux. Il a aussi démontré que l'on peut dissimuler un message dans un dessin géométrique en utilisant les points, les lignes et les triangles.

Bien d'autres techniques existent, et aux fins de clarté une classification de ces techniques est nécessaire ; en effet, on distingue :

- **La sécurité par obscurité (cryptographie) :** Cette appellation est donnée aux anciennes méthodes de cryptographie qui considèrent que l'algorithme reste «obscure» à l'ennemi. **Kerchoff** en **1883**, a énoncé le premier principe de l'ingénierie cryptographique, qui consiste à prendre en compte le fait que la méthode de cryptage peut être connue par l'ennemi. La sécurité de la communication ne tient donc que grâce à l'existence d'une clé de cryptage. A partir de ce moment, cette technique (sécurité par obscurité) est devenue obsolète, car elle se base sur le fait que l'ennemi n'est pas sensé connaître la technique de stéganographie. On s'est donc orienté vers d'autres techniques plus efficaces.
- **Le camouflage :** Dès les premiers temps de l'architecture les artistes ont compris que les travaux de sculpture ou de peinture apparaissent différents selon l'angle de vue, et ont établi des règles pour la perspective. Pendant le 16ème et 17ème siècles les images amorphes fournissent un moyen idéal de camouflage d'informations politiques

dangereuses et d'idées hérétiques. Une grande œuvre de dissimulation d'images amorphes –The Vexierbild – a été créée vers 1530 par Sho, un graveur de Nurmberg élève de Durer (1471-1528), quand quelqu'un la regarde normalement, il aperçoit un paysage étrange, mais en regardant par le côté on voit le portrait de rois célèbres.

Dans ses mémoires, Herodotus (486-425 av. JC) montre comment vers 440 av. JC Histraeus avait rasé le crane de son esclave le plus fidèle pour y tatouer un message qui a disparu après que les cheveux aient repoussé (Le but était de soulever une révolte contre les Perses). Cette méthode a été utilisée au début du 20^{e} siècle par des espions allemands. Heradotus dit comment Demeratus, un grec à la cour de Perse, avait prévenu Sparta d'une invasion imminente par Xerxes : il avait enlevé la cire d'une tablette d'écriture, avait écrit son message sous le bois, et après, avait recouvert le message avec de la cire. Un grand nombre de techniques ont été inventées ou reportées par Aeneas le Tacticien, comme la dissimulation de messages dans les semelles des messagers ou dans les boucles d'oreilles de femmes, des textes écrits sur des tablettes de bois qui sont, ensuite, peintes à la chaux, et aussi des mots transportés par des pigeons.

L'équivalent numérique de ces techniques de camouflage est l'utilisation d'algorithmes de masquage. Comme la plupart des techniques de codage de source, ils se basent sur les propriétés du système de perception humain. La dissimulation du son par exemple, est un phénomène dans lequel un son interfère avec notre perception d'un autre. Le masquage fréquentiel a lieu quand deux sons de fréquences proches sont joués au même moment, le son le plus fort masquera le plus faible. Le masquage temporel a lieu quand un son de bas niveau est joué immédiatement après un son de niveau plus fort ; après l'arrêt du son fort il faut un petit moment avant que l'on puisse entendre un son plus faible de même fréquence. *Comme ces effets sont utilisés dans les standards de compression comme le MPEG, beaucoup de systèmes mettent les données à dissimuler dans les composantes les plus significatives (en terme de perception) de la donnée pour qu'elle survive à la compression.*

- **Dissimulation de l'endroit où l'information est tatouée :** Dans un protocole de sécurité développé en Chine Ancienne, l'expéditeur et le destinataire disposent d'un même masque de papier contenant des trous découpés dans différents endroits. L'expéditeur place ce masque sur un papier, écrit son message à travers les trous, enlève le masque et couvre le message secret par un autre message. Pour lire le message secret, le destinataire n'a qu'à placer le masque sur la feuille reçue.

  L'équivalent numérique est l'introduction de distorsions (modifications, décalages) à des endroits prédéterminés ou choisis pseudo-aléatoirement. Caméléon est une technique utilisée dans la diffusion de CD ou dans la télévision payante. Caméléon permet de diffuser un contenu ayant le même message chiffré tout en donnant à chaque utilisateur

une clé légèrement modifiée pour que chaque message déchiffré soit légèrement différent.

- **Étalement de l'information dissimulée :** Tirkel et al. ont été les premiers à noter que les techniques d'étalement du spectre étaient applicables au watermarking en utilisant les bandes passantes larges en comparaison avec la petite bande passante des données à dissimuler. Cox et al. présentent une méthode de watermarking image dans laquelle la marque est dissimulée dans les N composantes fréquentielles les plus significatives (pour l'œil) de la transformée en cosinus discrète d'une image. Le watermark est une séquence de nombres réels ayant une distribution gaussienne. Cette méthode résiste à différentes attaques mais son problème est qu'elle requiert l'image originale pour la vérification de la présence du watermark. Un deuxième problème est le faible taux d'informations. Le nombre de méthodes de dissimulation de données, travaillant dans des espaces transformés, augmente, car ceci améliore la robustesse contre la compression, le filtrage ou le bruit. Actuellement, on peut noter que l'utilisation d'une transformée donne de bons résultats contre les algorithmes de compression utilisant cette transformée. Quelques méthodes travaillent directement sur les objets compressés. Par exemple, des outils de stéganographie dissimulent des informations dans des fichiers GIF en inversant les couleurs des pixels avec celles qui leurs sont adjacentes dans la palette.

  Une nouvelle technique de codage par transformation est la dissimulation de l'écho qui repose sur le fait qu'on ne peut pas percevoir des échos courts (de l'ordre de la milliseconde). Pour dissimuler des données dans un signal audio on introduit deux types d'écho court ayant un temps différent pour coder les « uns » et les « zéros ». Ces échos sont codés à des endroits séparés par des espaces pseudo-aléatoires.

- **Techniques spécifiques à l'environnement :** La dissimulation par écho est une technique qui utilise les particularités d'un environnement. Une technologie émergente du monde militaire est « la communication par explosion de météorite », qui utilise le canal radio transitoire engendré par l'ionisation des traînées des météorites entrants dans l'atmosphère pour envoyer des paquets de données entre des stations mobiles et une base. La nature transitoire de ces canaux rend la localisation des stations mobiles très difficile.

  Un autre exemple est l'utilisation d'encres sensibles aux ultraviolets, ainsi, comme les lampes utilisées dans les photocopieuses dégagent des UV forts, la photocopie d'un papier contenant une écriture invisible « void » rendra cette inscription visible.

  Une technique utilisée dans la protection du copyright de logiciel est la transmission du numéro de série de la licence pour qu'un site vérifie s'il n'y a pas une autre copie utilisant ce même numéro (et donc illégale) sur un autre ordinateur.

1.2. QU'EST CE QUE LE WATERMARKING ? [2]

Le « digital watermarking » peut se définir comme l'ensemble des méthodes et techniques qui permettent de cacher l'information à transmettre par l'intermédiaire d'un media digital. Le tatouage (dissimulation) de l'information s'effectue en manipulant le contenu de la donnée numérique, de telle sorte que l'information fasse partie intégrante de la donnée après tatouage. Le processus de tatouage doit être conçu de telle manière que les modifications apportées au media doivent être imperceptibles à l'œil humain. Le plus souvent le watermark contient un message donnant des informations concernant le créateur ou le distributeur de la donnée.

Dans le but de sécuriser une communication on a souvent recours au cryptage des données, mais dans certains cas ce dernier s'avère inadéquat. C'est ainsi que Aenaes le tacticien ainsi que beaucoup d'autres écrivains tel Wilkins en 1641 se sont concentrés sur des méthodes qui consistent à cacher l'information plutôt que de la crypter, méthode qui éveille moins les soupçons.

Donc l'étude de la sécurité des communications ne se borne pas seulement à l'étude des techniques de cryptage mais traite aussi des techniques qui consistent à cacher l'information ; le watermarking et la stéganographie font partie de ces techniques comme le montre la figure suivante :

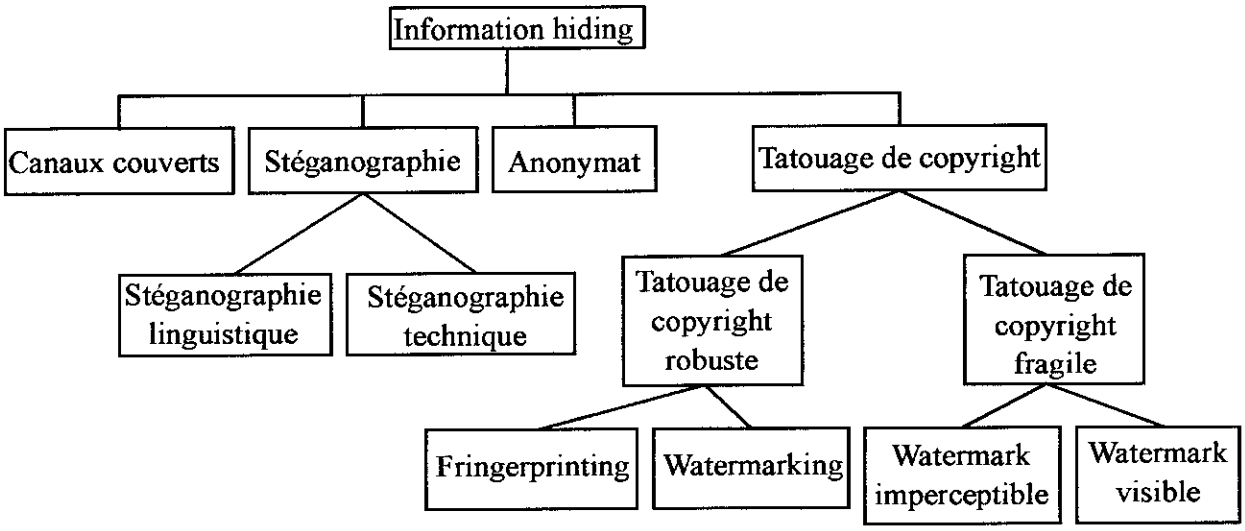

Figure-1 : Classification des techniques de data hiding

Alors que le but du cryptage est de protéger le contenu du message contre la lecture, la stéganographie, elle, cache une information dans une autre, le mot stéganographie est un mot grec qui veut dire littéralement « écriture couverte ».

Jusqu'à récemment, les techniques qui consistent à dissimuler l'information (data hiding) étaient sous-estimées par la communauté scientifique, mais les choses ont peu à peu changé

comme le montre la table 1.

Années	1992	1993	1994	1995	1996	1998	1999
Publications	2	2	4	13	29	64	103

Table-1 : Nombre de publications traitant du digital watermarking (source INSPEC janvier 1999)[1]

Il est à noter que les recherches menées sur le data hiding concernent en premier lieu les droits d'auteur (copyright). En effet, depuis que les données multimédia (vidéo, audio, images) sont disponibles sous une forme numérique, la duplication de ces dernières est devenue chose aisée, et de plus en plus de copies non- autorisées circulent. Les techniques de watermarking (message de copyright caché) et de fingerprinting (numéro de série ou label caché) permettent de protéger ces données, d'identifier les fraudeurs et pouvoir les poursuivre en justice si nécessaire. D'autres applications sont envisageables et seront exposées plus loin dans le document.

D'une manière générale, le modèle appliqué au data hiding consiste ainsi qu'explicité précédemment à dissimuler (à tatouer) une information dans une autre servant de couverture et produisant ce que l'on appelle le **stego-objet** ou objet marqué. La **stego-clé** quant à elle sert au contrôle du procédé de tatouage afin d'éviter la détection et donc de retrouver l'information tatouée, que se soit par des personnes la connaissant (l'information NDLR) ou par des personnes connaissant quelques valeurs dérivées de la stego-clé.

L'intérêt croissant que suscite les domaines de la stéganographie, le watermarking digital et le fingerprinting a conduit à une certaine confusion dans le domaine.

La stéganographie d'une part, le fingerprinting et le watermarking d'autre part se différencient par la condition de **robustesse**. La robustesse dépend en fait de l'application et concerne en général la résistance du watermark au traitement usuel de différents processus (compression, codage, conversion, filtrage, etc.,.).

Certains watermarks peuvent être visibles mais la plupart des techniques de watermarking traitées dans la littérature concernent les watermarks invisibles car ayant un champ d'application plus vaste. On distingue dans la littérature d'une part les watermarks dits fragiles qui ont la propriété de se détruire si une quelconque altération affecte le stego-objet (ce qui démontre que le stego-objet n'a pas été piraté) et d'autre part on distingue les watermarks dit robustes, qui ont la propriété inverse, donc de résister à toutes les tentatives d'altération du stego-objet et d'être inséparables de ce dernier, c'est à dire que le stego-objet est détruit si l'on tente de l'enlever.

Les critères sur lesquels sont jugés les *watermarks* sont les suivants [1]:

- **Imperceptibilité** : Le watermark digital tatoué au niveau de la vidéo hôte doit être imperceptible à l'œil humain.

- **Sécurité** : L'extraction du watermark doit être impossible une fois tatoué, même si le schéma de base de dissimulation du watermark est connu.
- **Robustesse** : Il doit être impossible de manipuler le watermark quelles que soient les opérations effectuées sur celui-ci, aussi bien sur la vidéo compressée que non-compressée, et ce sans dégrader la qualité de la vidéo hôte au point de la rendre commercialement inexploitable. Ces manipulations sont par exemple, l'ajout d'un signal, le filtrage, le codage, le « cropping » ...
- **Complexité** : Les procédés de watermarking et d'extraction doivent être assez simples donc de faible complexité, mais les degrés de complexité des divers procédés diffèrent en fonction de l'application et doivent tenir compte des différentes attaques possibles
- **Traitement au niveau du domaine compressé** : Il est évident que les distributeurs des vidéos vont les stocker au format compressé, par exemple au niveau d'un serveur World Wide Web. Il doit être possible de dissimuler le watermark au niveau du domaine compressé car :
  - Il est trop complexe de décompresser, dissimuler le watermark puis recompresser la vidéo.
  - La qualité des vidéos décompressées puis recompressées ne peut être garantie.
- **Bit rate constant** : La dissimulation du watermark ne doit pas augmenter le débit (bit rate), au moins en ce qui concerne les applications où un bit rate constant est nécessaire, par exemple pour des applications où la bande passante du canal doit être respectée.
- **Interopérabilité** : Même si de plus en plus le watermarking des vidéos compressées se généralise, il serait souhaitable que les vidéos non-compressées puissent être tatouées sans avoir à les compresser d'abord.
- **Coût :** La mise en place du système doit être d'un coût raisonnable sur les différents produits proposés.

1.3. CLASSIFICATION DES SYSTEMES DE WATERMARKING [2]

On distingue dans la littérature plusieurs types de watermarks dits robustes que l'on peut classifier de la manière suivante :

- **Systèmes privés de tatouage :** Ces systèmes extraient le watermark M de l'image potentiellement distordue I'' et utilisent l'image originale I pour identifier la position du watermark, ce sont les systèmes dits de type-1. D'autres systèmes qui requièrent aussi l'original I, produisent quant à eux une réponse du type « oui » ou « non » à la question : « est-ce que le watermark M est présent au niveau du document I'' ? », on a donc des systèmes de type-2, (I''*K*M-->{0,1}), ces systèmes sont évidemment plus

robustes car ils ne fournissent aucune information et requièrent l'accès au document secret original I.

- **Systèmes semi-privés de tatouage :** Ils ont les mêmes caractéristiques que les systèmes privés sauf qu'ils ne font pas appel au document original pour la détection. La plupart des méthodes proposées actuellement font parties de cette catégorie.
- **Systèmes publics de tatouage :** Ces systèmes ne requièrent ni l'original I ni le watermark M et extraient n bits d'information (la marque) du stego-objet (I''*K→M). Ces systèmes ont un champ d'application plus large que les autres systèmes et peuvent même servir pour des applications où des systèmes privés sont nécessaires, en augmentant la robustesse.
- **Systèmes asymétriques de tatouage (tatouage à clé publique) :** ces systèmes ont la propriété essentielle d'avoir un watermark qui peut être lu sans avoir la possibilité de pouvoir retirer ce dernier.

Les figures 2 et 3 introduisent les procédés généraux de tatouage et de détection de la marque.

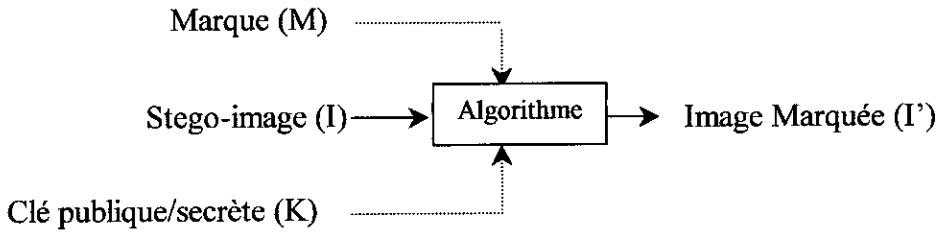

Figure-2 : Schéma de tatouage

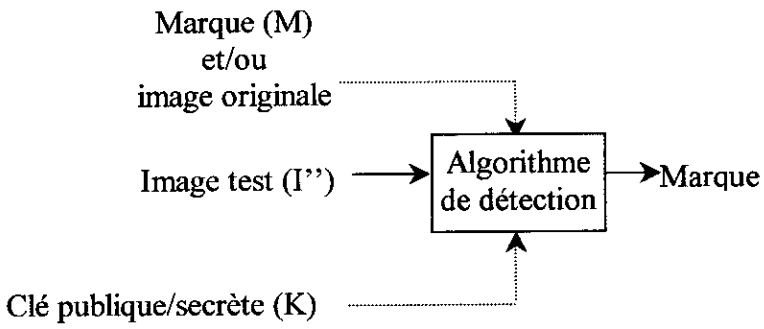

Figure-3 : Schéma de détection

1.4. APPLICATIONS

Les consommateurs ne vérifient pas le filigrane sur leurs billets de banque quand ils payent leurs achats, mais les caissiers et les employés de la banque peuvent utiliser le filigrane pour vérifier l'authenticité de votre argent. Les watermarks numériques fonctionnent de la même façon : Ils sont faits de telle façon qu'ils ne gênent pas le spectateur (ou l'auditeur dans le cas du son). Mais avec des détecteurs dédiés, le watermark peut être extrait et utilisé à des fins de vérification ou autres. Un watermark d'une vidéo numérique est une information invisible enregistrée sur cette vidéo. Ceci est fait en changeant légèrement le contenu de la vidéo suivant un modèle secret. Le watermark est invisible durant une visualisation normale mais un décodeur peut le détecter et l'extraire en utilisant la clé qui a servi pour le créer. Le watermark peut être inséré à différentes étapes du processus de création : dans le studio de production, par exemple, mais les stations d'émission ou les cinémas peuvent insérer leur identification plus loin dans la chaîne. Une fois appliqué, le watermark ne peut être ni enlevé ni changé sans connaître la clé secrète utilisée pour sa création. Il est lié au contenu, survivant aux différentes étapes de traitement du signal comme la copie, l'édition, la compression et même la transformation en analogique.

La vidéo numérique offre aux consommateurs une grande qualité d'image, mais permet aussi, des copies non autorisées avec la même qualité que celle de l'original. Pour faire face à ceci, une méthode de vérification de l'origine et du statut du copyright est souhaitable. Ajouter une signature numérique, ou un watermark, impossible à enlever ou à modifier est une solution possible. Avec le Watermarking la distribution d'émissions vidéos numériques peut être surveillée à travers le monde entier et en temps réel. Ceci aide les studios de production et les stations d'émission à vérifier l'utilisation de leurs programmes.

De nouveaux services apparaîtront comme le contrôle pour les publicités coûteuses diffusées durant les heures de grande écoute. Le passage sur les ondes d'un nouveau clip vidéo peut être mesuré pour mettre à jour automatiquement le hit parade. Le contrôle des copies et de la lecture d'enregistrements ainsi que le contrôle du nombre de copies pouvant être faites sera possible en intégrants des modules spéciaux dans le matériel vidéo.

1.4.1. Le contrôle du copyright

Dans quelques années les DVD enregistrables permettront aux consommateurs de composer leurs propres vidéos de haute qualité. Le développement du watermarking pour les vidéos numériques a été fortement motivé par le désir de protéger les produits DVD vidéo commerciaux contre les copies non autorisées.

Un watermark est une forme passive de protection mais très utile quand il est utilisé avec d'autres systèmes de protection. Le contenu des DVD est actuellement protégé contre l'utilisation non autorisée par le cryptage. Le problème rencontré avec cette procédure se situe lors de la transformation du signal décrypté en analogique car il peut être copié à ce moment (ce

qui est toujours le cas quelque part dans la chaîne). Le watermark par contre reste présent dans le signal analogique.

Le watermark peut être utilisé pour attacher des informations au contenu concernant les restrictions sur la copie (ne jamais copier, copier une seule fois, copie non limitée autorisée...). Dans les contrats de licence signés avec l'industrie des loisirs, les fabricants d'équipements vidéos d'enregistrement peuvent convenir d'ajouter la détection de watermark à leurs produits et ainsi respecter le copyright.

1.4.2. Surveillance de la diffusion

En plus de la vérification de copyright, les watermarks ont d'autres applications. L'idée de la surveillance de la diffusion consiste en l'ajout d'un watermark au contenu de la vidéo au moment de la production ou de la distribution à des stations de diffusion pour pouvoir suivre l'utilisation par un réseau de stations de surveillance qui cherchent le watermark dans les signaux diffusés.

De nos jours la couverture télévisée d'évènements sportifs ou des informations met en jeu de grandes sommes d'argent et les droits de diffusion sont souvent vendus à travers le monde entier. De plus, les stations de diffusion partagent de façon croissante le contenu à travers des réseaux. La distribution croissante du matériel vidéo rend plus difficile la vérification de l'utilisation légale ou non de ce dernier. Le watermarking utilisé pour la surveillance de la diffusion offre une solution attractive à ce problème.

La surveillance de l'utilisation de la vidéo a engendré une demande croissante quant à l'utilisation du watermark. Les clients ne veulent pas seulement savoir où leurs vidéos ont été utilisées, mais aussi, quelle partie, pour combien de temps et à quelle heure. Contrairement à la protection de copyright de media fixes, où un watermark constant d'une taille limitée est utilisé, la surveillance requiert un label unique contenu dans chaque seconde de la vidéo.

1.4.3. Nouvelles applications

Il est possible d'utiliser un watermark différent pour chaque utilisateur du matériel vidéo. C'est ce qu'on appelle un fingerprint. Cette méthode permet de « tracer » l'origine de copies illégales, car chaque watermark est unique. Avec les watermarks, toutes sortes d'informations utiles peuvent être insérées de façon invisible au contenu vidéo. Des informations concernant le titre, les acteurs, le type, etc., ajoutées au film peuvent être utilisées à des fins d'archivage ou pour la télévision.

Les discussions sur le MPEG-21, comme étape suivant l'intégration des formats de compression vidéo et audio numériques, viennent de commencer. Le MPEG-21 va couvrir la chaîne entière, de la production à l'utilisation du matériel, et va inclure la gestion des droits numériques.

1.4.4. La vidéo Mobile

Trouver la combinaison optimale de besoins contradictoires est le challenge principal dans le développement et le test des algorithmes de watermarking. Pendant que la robustesse, à travers

les différentes étapes de traitement du signal, demande un watermark fort, il doit être assez petit pour rester invisible durant une visualisation normale. De plus, plus on veut ajouter des informations au watermark, plus on aura du mal à le cacher. Les vidéos numériques utilisées dans les applications vidéos mobiles, l'accès itinérant à Internet, par exemple, sont très fortement compressées pour pouvoir utiliser efficacement la bande passante limitée.

1.5. MODELISATION DE L'INSERTION D'UN WATERMARK

Plusieurs applications de l'imbrication d'informations peuvent être décrites par la figure 4. On désire imbriquer une certaine information *m* dans un vecteur signal hôte $x \in \Re^N$, avec un taux de R bits par dimension.Une fonction d'imbrication relie x et *m* au signal composite $s \in \Re^N$.Cette fonction subit des contraintes comme la distorsion due à l'erreur quadratique donnée par :

$$D(s,x) = \frac{1}{N}\|s-x\|^2 \le D_{\max} \qquad (1)$$

Le signal composite passe à travers un canal où il subit différents traitements et distorsions. Les combinaisons de ces différents effets sont modélisées par le vecteur bruit $n \in \Re^N$, qui peut être déterministe ou aléatoire, dépendant ou indépendant du signal. On suppose que l'énergie du vecteur bruit est bornée, i.e. :

$$\|n\| \le N\sigma_n^2 \qquad (2)$$

Le décodeur extrait $\hat{m}$ l'estimé de m à partir de y. On quantifie la robustesse du système par la valeur maximale de σ^2_n garantissant $\hat{m} = m$.

Le problème auquel il faut faire face est de concevoir une fonction d'imbrication s (x , m) qui offre le meilleur compromis entre les trois paramètres : taux de bits ,distorsion et robustesse.

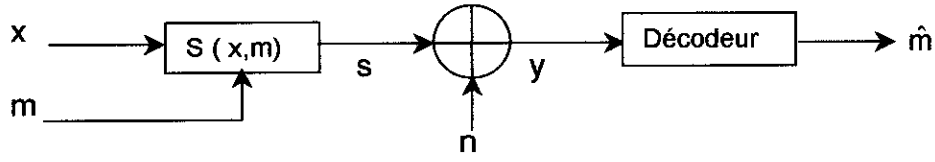

Figure-4 : Modèle général du problème d'imbrication d'information

Chapitre 2

ATTAQUE ET EVALUATION DES FILIGRANES ELECTRONIQUES

- **Exemple d'attaques et classification**
- **Techniques d'attaques**
- **Banc d'essais**
- **Conclusion**

ATTAQUE ET EVALUATION DES FILIGRANES ELECTRONIQUES

2.1. EXEMPLES D'ATTAQUES ET CLASSIFICATION

Au cours des dernières années, beaucoup de nouvelles techniques ont été proposées pour cacher des notices de copyright et des numéros de série personnalisés dans les documents multimédias de manière à empêcher ou du moins réduire, les copies illégales. Il est certain que des progrès utiles pourraient être réalisés en essayant d'attaquer toute cette première génération de méthodes de marquage [4]. Dans le domaine de la cryptographie, les progrès ont été itératifs : des algorithmes ont été proposés, des attaques ont été trouvées, de meilleurs algorithmes ont vu le jour, et ainsi de suite. Cette façon de procéder doit être appliquée aux filigranes numériques.

Beaucoup de systèmes récemment proposés sont qualifiés de « robustes » par leur inventeur. Malheureusement les critères utilisés pour démontrer cette robustesse varient d'un système à l'autre, et les attaques récentes [4], [5], [6], [7], [8] montrent que les critères utilisés pour démontrer cette robustesse sont parfois inadéquats. Compression J.P.E.G., bruit additif, filtrage passe-bas, changement de taille ou émargement sont pris en compte par la plupart des systèmes mais les transformations géométriques, même très simples, sont rarement évoquées [9, 10]. Dans certains cas, le système est dit seulement « résistant aux procédés usuels de traitement du signal et aux déformations géométriques sur certaines images standard ».

Différentes attaques [4] mettent en évidence des limitations sérieuses de plusieurs outils de marquage dont PictureMarc 1.51 [11], SysCoP [12], SureSign [13], JK_PGS, EIKONAmark [14], Giovanni et la méthode NEC [15], Echo Hiding [16]. Il va sans dire que des systèmes utilisant les mêmes techniques sont susceptibles d'être attaqués de la même manière.

L'attaque de base partait du constat que beaucoup de systèmes de marquage utilisent de façon plus ou moins déguisée la technique d'étalement de spectre. Cette dernière est très robuste à l'ajout de bruit ou aux distorsions de l'amplitude du signal mais supporte très mal les erreurs de synchronisation. Une méthode très simple pour briser cette synchronisation consiste simplement à supprimer quelques échantillons. Dans le cas des images, quelques colonnes de pixels suffisent. Bien qu'extrêmement simple cette attaque fonctionne sur les prototypes naïfs qui ne prennent en compte que le bruit additif.

2.1.1. Attaques générales

Certains systèmes de marquage d'images supportent également des manipulations simples que quiconque peut faire avec des outils de traitement d'image disponibles dans le commerce : rotation, redimensionnement, émargement, retournement horizontal et compression J.P.E.G. (Ceci est confirmé par les résultats de tests résumés dans la table 1). Malheureusement, des combinaisons de celles-ci suffisent généralement à mettre en défaut le système de marquage.

C'est ce qui a motivé la mise en œuvre de StirMark, initialement programmé par Markus G. Kuhn. Nombre d'améliorations ont été ajoutées depuis, et notamment la possibilité d'utiliser cet outil comme base d'un banc d'essai.

La version originale de StirMark applique de simples déformations bilinéaires aléatoires. Si A, B, C et D sont les sommets de l'image, un point M de ladite image peut être exprimé de la façon suivante : $M = \alpha(\beta A + (1-\beta)D) + (1-\alpha)(\beta B + (1-\beta)C)$ où $0 \le \alpha,\ \beta \le 1$ sont les coordonnées de M par rapport aux sommets de l'image. La déformation est appliquée en déplaçant légèrement et aléatoirement les sommets dans les deux directions. Les nouvelles coordonnées de M sont recalculées grâce à la formule précédente en gardant (α, β) constantes. L'avant dernière ligne de la table 1 montre que certains systèmes de marquage supportent les déformations.

Davantage de déformations—toujours invisibles—peuvent être appliquées à une image. En plus de la transformation bilinéaire précédente, les nouvelles versions de StirMark dévient légèrement chaque pixel de façon non uniforme : quelques 0,1% des dimensions de l'image au centre et quasiment rien sur les bords. Dans la version actuelle, la forme de cette « bosse » est simplement une fonction sinus : si (x, y), avec $0 \le x \le X$ et $0 \le y \le Y$, sont les coordonnées d'un pixel dans l'image après la déformation bilinéaire, alors ses nouvelles coordonnées sont : $x' = x + \lambda \sin(\pi y / Y)$ et $y' = y + \lambda \sin(\pi x / X)$. À cela est ajouté un déplacement de plus grande fréquence de la forme $\delta = \lambda \sin(\omega_x x)\sin(\omega_y y)(1 + n(x, y))$ où n est un nombre aléatoire : $x'' = x' + \delta_1$ et $y'' = y' + \delta_2$. Pour une bonne rapidité de traitement, le rééchantillonage utilise un algorithme d'approximation quadratique par B-spline [17] et, pour de meilleurs résultats une légère compression J.P.E.G. est appliquée à la fin du processus. Un exemple d'image « attaquée » est donné dans la figure 1.

Il existe aussi des méthodes générales pour attaquer les outils de marquage du son. Par exemple, les techniques de restauration de signaux sonores ont été étudiées en détail depuis de nombreuses années, et se sont montrées efficaces pour localiser et enlever les dégradations qui apparaissent dans les anciens enregistrements [18].

Ces mêmes méthodes peuvent être utilisées contre les outils de marquage du son. Ces attaques « reconstruisent » simplement le signal bloc par bloc en utilisant une partie du signal marqué pour prédire chaque bloc. Un simple modèle auto-régressif est utilisé pour la prédiction, dont l'algorithme est détaillé dans [19].

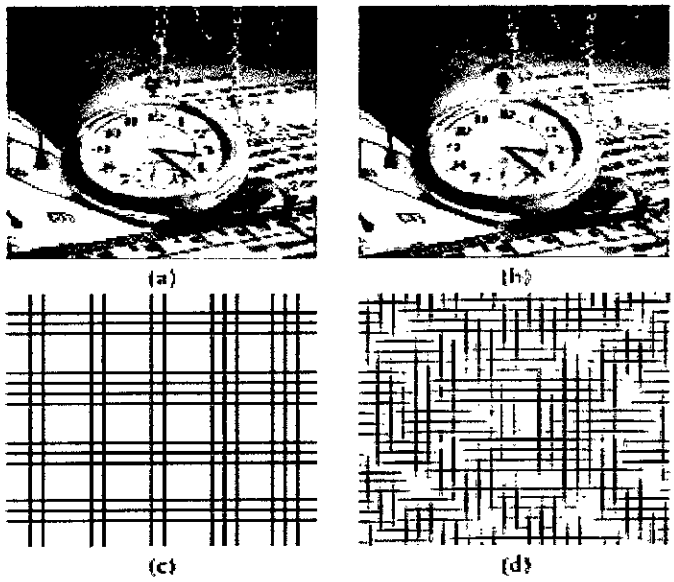

Figure-1 : Lorsque StirMark est appliqué sur des images relativement complexes ou des photographies, les déformations introduites sont quasiment invisibles: watch avant (a) et après (b) StirMark (paramètres par défaut). Pour comparaison, les mêmes déformations ont été appliquées à une grille (c et d). Image de synthèse : Pocket Watch on a Gold Chain. Copyright Kevin Odhner.

	Digimarc 1.51	SureSign 3.0 Demo	EikonaMark 3.01	Giovanni 1.1.0.2	SysCoP 1.0R1
Conversion GIF	100	100	100	60	80
Échelle (0,5, 0,75, 0,9, 1,1, 1,5, 2)	70	100	0	63	0
Émargement (1, 2, 5, 10, 15, 20, 25, 50 %)	100	100	0	15	0
Rotation (-2, -1, -0,5, 0,5, 1, 2 °)	82	58	0	10	0
J.P.E.G. (90, 85, 80, 75, 60, 50, 25, 10, 5)	56	72	90	12	58
Filtrage (médian 3 x3, Gauss)	100	100	100	60	80
Retournement horizontal	100	100	0	0	0
StirMark 1.0	80	80	0	0	0
StirMark 2.2	0	0	0	0	0

Table-1 : Test de robustesse pour cinq outils de marquage. Les valeurs sont des pourcentages de réussite. Pour chaque outil, cinq images de test *(lena, lunettes, fille, mercedes* et *babouin)* ont été utilisées. Chaque image a été tatouée avec les meilleurs paramètres n'introduisant aucun effet désagréable (à l'oeil). Bien que toute comparaison doive être effectuée avec le plus grand soin (tous ces outils de marquage n'ayant pas la même application), cette table confirme l'état de l'art dans le domaine.

2.1.2. Attaques spécifiques

Lorsque les méthodes générales ne permettent pas d'attaquer un système stéganographique, rien n'empêche un adversaire d'utiliser des méthodes spécifiques. C'est ce qui a été fait pour la méthode *echo hiding* qui dissimule de l'information en introduisant des échos de très court délai, de l'ordre de la milliseconde, imperceptibles à l'oreille [20]. L'attaque évidente (détaillée dans [4]) consiste simplement à détecter les paramètres de l'écho en utilisant la même méthode que les inventeurs d'*echo hiding*, c'est-à-dire « l'alanyse spectrale » de Bogert et al. [21]. Des essais sur des signaux aléatoires et sur de la musique montrent que la méthode est relativement précise pour des échos entre 0,5 et 3 ms et permet d'extraire le signal caché, montrant ainsi une faille

dans la méthode originale d'*echo hiding*.

Les faiblesses inhérentes au marquage en général peuvent aussi être utilisées. Ceci est mis en évidence par une attaque générale contre les robots traqueurs, attaque à la propriété initiale remarquable que l'image marquée et l'image attaquée sont les mêmes.

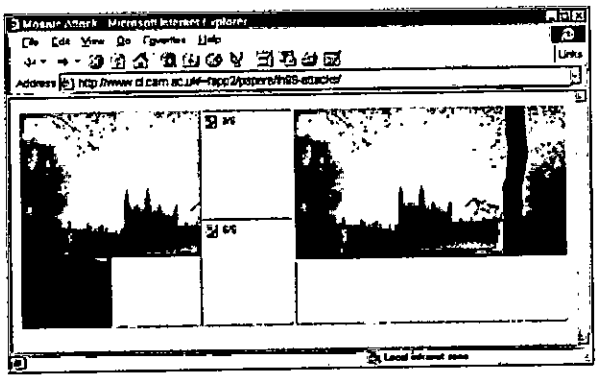

Figure-2 : Copie de la fenêtre d'un browser en train de télécharger une image après une attaque par mosaïque. Cette attaque découpe simplement l'image en morceaux qui sont automatiquement « recollés » par le browser lors de la présentation du résultat. Un logiciel permettant d'automatiser le découpage et la génération du code H.T.M.L. correspondant est disponible. Dans certains cas le chargement de la mosaïque est même plus rapide que l'image entière ! Pour cet exemple l'image (350×280 pixels) a été marquée avec PictureMarc 1.51 de Digimarc. Photographie : *Chapelle de King's College, Cambridge*. Copyright John Thompson, JetPhotographic, Cambridge.

Ces robots, qui font partie d'un système de détection automatique basé sur le *Web*, téléchargent image après image et vérifient si elles contiennent une marque. L'attaque consiste simplement à découper l'image en plusieurs parties, telle une mosaïque, et à générer du code H.T.M.L. de telle façon qu'un navigateur « recolle » les morceaux (Figure 2). Cette attaque est relativement générale puisque toute méthode de marquage d'image requiert une taille minimale. Par conséquent, cette division de l'image empêche le détecteur de retrouver la marque.

À cette attaque très simple, il convient d'en ajouter d'autres, basées sur des applets *Java* ou des contrôles *ActiveX*, qui peuvent être utilisés pour télécharger et afficher l'image à l'intérieur du fureteur ; ces objets peuvent éventuellement dé-brouiller l'image en temps réel si nécessaire. Déjouer de telles techniques, impliquerait de détecter les images dans la représentation de la page *Web* en mémoire et d'essayer d'en extraire une marque.

2.1.3. Attaques sur le protocole d'identification du propriétaire

La plupart des attaques « évidentes » tentent de filtrer, ou déformer l'image, pour atténuer le signal, ou empêcher la synchronisation du détecteur. Cet aspect ne représente qu'une petite fraction des possibilités offertes. Chaque élément de la chaîne de gestion des copyrights peut être mis en défaut.

Par exemple, Craver et al. ont montré comment empêcher un photographe de prouver qu'il est bien le détenteur du copyright d'une image même après l'avoir marquée [22]. L'idée de base

utilise le fait que beaucoup d'outils de marquage ne permettent pas de savoir quelle marque parmi plusieurs a été ajoutée la première : le processus de marquage est souvent additif, ou tout au moins commutatif et inversible. Par conséquent, si un ayant droit possède le document d, y dissimule un filigrane f, publie la version marquée, c'est-à-dire $d+f$, et n'a pas d'autre preuve de propriété, un pirate qui a enregistré un filigrane f', peut très bien prétendre que le document est le sien et que la version originale est $d+f-f'$.

Craver et al. concluent qu'il faut utiliser des systèmes de marquage dont l'inverse ne peut être calculé facilement ; autrement dit, des systèmes dont la marque dépend de l'original. Cependant il serai préférable d'utiliser les filigranes et les empreintes dans des systèmes plus larges, mettant en œuvre des mécanismes de datage sécurisé et des « notaires » électroniques.

Les mécanismes d'enregistrement ont reçu peu d'attention et les publications qui abordent le problème [23], [24] s'intéressent seulement à la protection du propriétaire et très peu aux droits des consommateurs qui peuvent très bien être abusés.

2.1.4. Problème de mise en œuvre

La robustesse des mécanismes de marquage et d'extraction n'est pas le seul problème à considérer. La plupart des attaques sur des outils de prestations cryptographiques proviennent de l'exploitation de failles découvertes accidentellement. La cryptanalyse s'est rarement révélée indispensable [25].

Il en est de même pour les outils de marquage. Une méthode pour modifier directement le programme PictureMarc de Digimarc était déjà disponible en août 1997 [26]. Elle utilise un outil de décompilation pour modifier directement le binaire du logiciel afin de pouvoir ajouter une marque quelconque même sur une image déjà marquée.

Plus grave, un utilisateur peut marquer très simplement une image à la place d'un autre, ouvrant la porte à un grand nombre d'abus. Une attaque similaire, due à Perrig, consiste à utiliser un détecteur/extracteur public comme « Devin » : des modifications infimes peuvent être appliquées au signal marqué jusqu'à ce que le décodeur échoue. Chaque détection apporte théoriquement un bit d'information : y a-t-il ou pas de marque [24].

2.2. TECHNIQUES D'ATTAQUES

Une *'attaque'* est une des manipulations que peut subir une image, et qui va plus ou moins l'endommager. Il est important de signaler que certaines de ces manipulations peuvent être appliquées en toute légitimité, tandis que d'autres ne concernent que des tentatives d'altérations intentionnelles de la marque dans un but malhonnête.

2.2.1. Manipulations géométriques

Les manipulations les plus simples sont de nature géométrique. Elles consistent en une des

transformations affines de l'image ou de parties de cette image telles que :

- *Rotation :* c'est une transformation qui est très utilisée après avoir scanné une image. Elle sert à réaligner des images (avec de petits angles), et peut être fatale à certains types de marquages.
- *Symétrie axiale :* elle n'est pas forcément décelable à priori si l'image présente naturellement cette symétrie.
- *Symétrie horizontale* : certaines images peuvent être « flipper » sans perdre de leur sens (par exemple un paysage).
- *Changement d'échelle* : cette manipulation peut être envisagée par un utilisateur tout à fait honnête, qui publie par exemple un livre et doit ajuster les images à la bonne échelle. Les transformations de ce genre peuvent être séparées en deux groupes :
  - Les transformations uniformes, pour lesquelles on conserve les proportions. L'échelle en X varie comme l'échelle en Y.
  - Les transformations non uniformes, où l'échelle en X ne varie pas comme l'échelle en Y

2.2.2. Extraction de morceaux de l'image au montage

- *Émargement (*cropping*) :* il s'agit de la suppression de quelques lignes ou colonnes sur les bords de l'image, ce qui peut détruire le marquage.
- *Extraction d'un détail d'une image :* on extrait un détail, par exemple un visage. Cette opération est du même type que l'émargement, mais beaucoup plus destructive; on ne se contente pas d'enlever quelques lignes.
- *Montage :* des morceaux d'images sont collés sur une autre image, par exemple remplacer le visage d'un personnage par un autre.

2.2.3. Changement de format et compression

Une particularité des images vidéos numériques est qu'elles peuvent être stockées sous divers formats (H 263, MPEG,....). Certains consistent en une compression avec éventuellement une perte d'informations, comme par exemple le format MPEG. Cette perte n'est pas forcément une baisse de qualité décelable à l'œil humain. Il faut noter que la compression MPEG est un mode très en vogue (surtout depuis l'avènement de l'Internet). L'avantage de cette méthode se situe dans les taux de compression importants que l'on peut obtenir, mais son désavantage se situe dans le fait qu'il s'agit d'une compression destructive. En effet plus on compresse la vidéo plus des défauts apparaissent. Notons que ces changements de formats ne résultent pas forcément d'une intention malhonnête : ils sont souvent simplement utilisés pour optimiser les capacités de stockage.

2.2.4. Filtrage

- *Filtre passe bas* : son application permet de supprimer ou d'amoindrir les fréquences élevées, il peut consister en :
  - Une méthode spatiale, où on applique un masque à l'image, consistant par exemple en un calcul de moyenne ou un filtre médian.
  - Une méthode fréquentielle, où on supprime les hautes fréquences dans la transformée de Fourier de l'image.
  - Un filtre fondé sur les ondelettes, comme les filtres miroir en quadrature(QMF).
- *Estompe,* flou : On peut introduire un flou plus ou moins important dans l'image en remplaçant, par exemple, chaque point de cette dernière par la moyenne médiane des points avoisinants. La quantité de flou introduite dépend du nombre de points considérés dans le calcul de la moyenne ou de la médiane.

2.2.5. Introduction du bruit dans l'image

On ajoute à l'image une deuxième image qui consiste en un bruit blanc, ce qui a pour effet de modifier légèrement des pixels uniformément répartis dans l'image.

2.2.6. Impression et renumérisation

Ce procédé consiste en l'impression du document puis en sa renumérisation à l'aide d'un scanner. Il peut aussi simplement signifier la numérisation d'une image imprimée présentée dans un livre ou un magazine.

2.2.7. Modification de couleurs

- *Passage de couleurs au gris : si* l'on considère la représentation (Y,U,V) alors le passage d'une image couleur à une image en nuance de gris, correspond à l'annulation des composantes U et V de tous les pixels. Leurs luminances restent inchangées.
- *Modification d'histogramme : cette* opération consiste en la modification d'histogrammes représentant les composantes (R G B) de l'image. Ce sont :
  - Une normalisation
  - Une égalisation
  - Une transformation gamma

2.3. BANC D'ESSAIS

Toutes ces attaques débouchent sur la même question : comment évaluer et comparer différents outils de marquage. Très peu d'auteurs ont publié des résultats de tests intensifs sur leurs outils de marquage [27], [28]. Un banc d'essai est donc nécessaire pour mettre en évidence les domaines de recherche et pour comparer rapidement les nouveaux algorithmes qui

apparaissent régulièrement.

Aujourd'hui encore, chaque chercheur utilise sa propre batterie de tests, ses propres images et sa propre méthodologie. Par conséquent, toute comparaison est impossible sans reprogrammer la méthode, dans les cas où il n'existe pas de logiciel d'évaluation. Avec un banc d'essai commun - même imparfait- les avantages sont évidents : un tableau d'évaluation type pourrait être fourni avec chaque nouvel algorithme, permettant ainsi d'avoir une idée de sa robustesse sans perdre de temps pour comprendre et évaluer la méthode.

Une première tentative, basée sur StirMark est proposée [29]. Elle prend uniquement en compte les processus de marquage et d'extraction qui sont considérés comme des boîtes noires (cf. Table 2 pour des résultats). La procédure est très simple :

- Marquer avec les meilleurs paramètres les images fournies avec StirMark de telle façon à ce que le P.S.N.R.(Peak Signal to Noise Ratio) (ou une autre mesure à définir) soit inférieur à 38 dB.
- Utiliser StirMark pour appliquer une série automatique de tests en une ligne de commande.
- Pour chaque image attaquée, tenter de détecter/extraire la marque (1 point en cas de succès ; 0 en cas d'échec).

Notons que si l'outil de marquage offre une interface sous forme de ligne de commande, cette procédure peut être entièrement automatisée en utilisant des scriptes Unix, Perl ou DOS.

Ce schéma général contient encore quelques inconnues, et notamment le nombre de bits cachés par le marqueur et la mesure de qualité. Pour le premier, il semble que 70 bits soit raisonnable [30]. Pour le second, il reste à prouver que la mesure utilisée a une influence significative sur les résultats des tests. La plupart des outils de marquage récents utilisent des modèles basés sur le système humain de perception.

2.4. CONCLUSION

La plupart des outils de marquage sont vulnérables à différentes attaques relativement simples et notamment aux déformations géométriques aléatoires utilisées par StirMark ou aux méthodes de restauration du signal dans le cas de signaux sonores. il est donc nécessaire d'instaurer une méthode d'évaluation et un banc d'essai pour évaluer les performances des outils de marquage de copyright.

Afin d'augmenter la résistance d'un système de marquage à différentes attaques, on peut essayer de prévoir les déformations possibles qu'un pirate peut utiliser : La marque pourrait alors être cachée dans l'espace de transformation inverse, ou dans un espace invariant à l'attaque. Ó Ruanaidh et Pun, par exemple, proposent d'utiliser la transformée de Mellin afin de résister aux rotations d'angle quelconques et aux changements de taille [7].

Dans le cas d'attaques plus générales comme StirMark, on peut remarquer que les

déformations, bien que globalement aléatoires, sont quasiment linéaires sur une petite surface de l'image. Ainsi, en décomposant l'image en petits blocs, il devrait être possible d'augmenter, par exemple, la valeur de corrélation entre le signal reçu et le signal d'étalement.

L'étude de notre perception des déformations géométriques devrait également permettre de modéliser encore mieux les images et d'améliorer la résistance des marques à des attaques comme StirMark.

	Digimarc	Unige	SureSign	SCMark
Filtrage				
Gauss	100	100	100	100
Médian	100	100	100	100
Rendre plus net	100	100	100	100
F.M.L.R.	100	67	100	100
Compression				
J.P.E.G.	65	63	87	100
GIF/Quantification des couleurs	100	1	100	20
Échelle				
Sans J.P.E.G. 90	81	86	97	0
Avec J.P.E.G. 90	72	83	83	0
Émargement				
Sans J.P.E.G. 90	100	83	94	2
Avec J.P.E.G. 90	98	83	91	2
Cisaillement				
X	50	38	42	0
Y	50	21	42	0
Rotation				
Auto-émargement	98	98	37	2
Auto-redimensionnement	97	98	51	26
Autres transformations géométriques				
Effacement de lignes et colonnes	100	83	89	7
Retournement horizontal	100	100	100	0
Déformations aléatoires (StirMark)	17	0	0	0

Table-2 : Résumé des résultats d'un banc d'essai basé sur StirMark 3.0. Un tableau détaillé est disponible sur www.cl.cam.ac.uk/~fapp2/watermarking/benchmark/. Outils de marquage testés : Batch Embedding Tool 1.00.13 et ReadMarc 1.5.8 de Digimarc, outils de l'Université de Genève (version du 15 janvier 1999), SureSign Server 1.94 de Signum Technologies et un outil de marquage de l'Université de Californie du sud (version du 29 mars 1999). Les séparations verticales indiquent d'une part que les conditions expérimentales étaient légèrement différentes pour Signum, et d'autre part, que le type de marquage est différent pour SCMark puisque celui-ci est privé, en ce sens qu'il utilise l'image originale.

Chapitre 3

TATOUAGE NUMERIQUE ET SYSTEME VISUEL HUMAIN

- **Tatouage numérique**
- **Modèle du système visuel**
- **Application au tatouage**

TATOUAGE NUMERIQUE ET SYSTEME VISUEL HUMAIN [31]

3.1. TATOUAGE NUMERIQUE

L'approche choisie pour le processus de tatouage numérique est basée sur la modulation à spectre étalé.

3.1.1. Insertion du tatouage numérique

Notre but est d'insérer une signature binaire $B=\{b_0,\ldots,b_N\}$ d'une longueur de N-bits dans une image I. Pour le moment nous ne considérons que les images comportant 256 niveaux de gris différents. Le tatouage w est défini par une superposition linéaire de fonctions bi-dimensionnelles $p_i(x,y)$, chacune représentant un bit :

$$w(x,y)=\sum_{i=1}^{N} p_i(x,y) \qquad (1)$$

L'image tatouée $\hat{I}$ est générée en ajoutant le tatouage w à l'image :

$$\hat{I}(x,y)=I(x,y)+w(x,y) \qquad (2)$$

Les fonctions bi-dimensionnelles p_i sont définies par :

$$p_i(x,y)=b_i\times\alpha(x,y)\times\phi_i(x,y) \qquad (3)$$

où b_i définit la valeur du bit i projeté de {0,1} à{-1,1}, et $\phi_i(x,y)$ est une fonction de modulation bi-dimensionnelle pseudo-aléatoire.

$\alpha(x,y)$ est une fonction de pondération ayant pour but d'adapter le tatouage numérique à l'image de telle sorte que l'énergie du tatouage soit maximisée sous la contrainte que la qualité de l'image tatouée ne soit pas inférieure à un certain seuil.

Les fonctions ϕ_i de modulation sont orthogonales, c'est à dire :

$$<\phi_i,\phi_j>=\delta_{ij}\|\phi_i\|^2 \qquad (4)$$

où $\langle\cdot,\cdot\rangle$ est le produit scalaire, δ_{ij} la fonction *Delta* de *Kronecker*, et $\|\cdot\|^2$ représente l'énergie de la fonction de modulation.

Il y a plusieurs moyens de générer ces fonctions de modulation. En principe, les fonctions de modulation sont définies de sorte que l'intersection des ensembles de positions (x,y) ayant une valeur $\phi_i(x,y)\neq 0$ soit vide. Cette technique assure que chaque valeur de l'image originale est modifiée par une fonction seulement. Les fonctions sont générées en utilisant des ensembles de positions S_i dépendant d'une clé k. Comme mentionné précédemment, l'intersection des ensembles doit être nulle, c'est à dire $S_i \cap S_j = 0, \forall i \neq j$. Les fonctions de modulation peuvent donc être définies ainsi :

$$\phi_i(x,y) = \begin{cases} S_i(x,y) & \text{si } (x,y)\in S_i \\ 0 & \text{ailleurs} \end{cases} \qquad (5)$$

Dans notre cas, les fonctions sont des fonctions pseudo-aléatoires avec une distribution bi-modale de {-1,1}, mais d'autres distributions sont possibles.

Dans les applications de tatouages numériques, il est important d'avoir un contrôle maximal sur les artéfacts. Pour cela, la densité D est introduite. Elle va définir la fraction des pixels de l'image qui sont modifiés par le processus de tatouage numérique. Cette densité D est donnée par:

$$D = \frac{\left|\langle \cup_{i=1}^{N} S_i \rangle\right|}{\left|\langle S \rangle\right|} \qquad (6)$$

où $|\langle\cdot\rangle|$ est le cardinal de l'ensemble et {S} l'ensemble universel.

La probabilité de distribution des positions dans l'image est uniforme, ce qui veut dire que la probabilité d'une position de faire partie de l'ensemble S_i est D/N.

3.1.2. Détection du tatouage numérique

Pour démoduler l'information insérée dans l'image, un corrélateur linéaire est utilisé. Ce corrélateur calcule la corrélation entre l'image tatouée et la fonction de modulation. Du fait que les propriétés statistiques de l'image ne sont pas stationnaires, et que l'espérance n'est pas égale à zéro, un processus de traitement préalable de l'image est introduit. Ce processus a pour but de diminuer la variance de l'image, ce qui augmente les performances du système. La statistique du détecteur et donnée par :

$$r_i = \langle \varepsilon(\hat{I}), \phi_i \rangle \qquad (7)$$

où ε est la fonction de traitement préalable de l'image, et $\hat{I}$ l'image tatouée.

3.2. MODELE DU SYSTEME VISUEL

Pour bien cacher le tatouage dans l'image, il est utile d'exploiter les faiblesses du système visuel humain. Cependant, le système visuel humain est extrêmement complexe, et plusieurs de ses propriétés ne sont pas bien comprises aujourd'hui encore. Nous nous concentrons ici sur deux aspects importants, qui sont la variation de la sensibilité au contraste
et le phénomène de masquage.

3.2.1. Sensibilité au contraste

Le contraste est une mesure de la variation relative de la luminance. Malheureusement, il n'y a pas de définition du contraste qui soit appropriée pour tous les genres de stimuli visuels. Dans les expériences psycho-visuelles, on utilise souvent des motifs périodiques, par exemple des sinusoïdes, dont la luminance varie entre L_{min} et L_{max}. Le contraste de ces grilles est donné par la définition de Michelson :

$$C_M = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} \qquad (8)$$

Par contre, le contraste d'un stimulus qui se compose d'un incrément ou décrément
sur un fond uniforme d'une luminance est mieux décrit par la définition de Weber :

$$C_W = \frac{\Delta L}{L} \qquad (9)$$

Ces deux définitions ne sont pas équivalentes : le contraste selon Michelson varie entre 0 et 1, tandis que le contraste selon Weber varie entre -1 et ∞. Il existe d'autres définitions similaires du contraste, mais aucune d'entres elles ne convient à la description du contraste d'images complexes, car pour la plupart de ces définitions, il suffit de quelques points très foncés ou très clairs pour déterminer le contraste de l'image entière.

En utilisant le fait que la sensibilité au contraste du système visuel humain varie avec l'adaptation à la luminance moyenne locale, Peli a proposé une mesure de contraste locale limitée en bande spectrale :

$$C_p(x,y) = \frac{b_p(x,y)}{I_p(x,y)} \qquad (10)$$

où $b_p(x,y)$ est la réponse passe-bande, et $I_p(x,y)$ est la réponse passe-bas qui contient l'énergie au-dessous de la bande passante. Des versions modifiées de cette définition du contraste local ont été utilisées avec succès dans certains modèles de vision.

La sensibilité au contraste est définie comme l'inverse du seuil de la visibilité, c'est à dire le contraste minimal nécessaire pour qu'on puisse percevoir le stimulus. Elle dépend non seulement

fortement de la fréquence spatiale des stimuli (voir figure 1), mais aussi de leur couleur et de leur intensité. En général, notre sensibilité est la plus haute pour des stimuli lumineux d'une fréquence basse à moyenne.

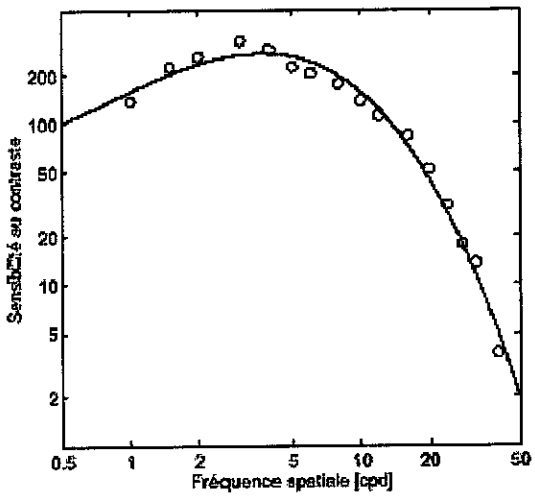

Figure-1 : La sensibilité au contraste en fonction de la fréquence spatiale

3.2.2. Masquage

Le masquage est un phénomène visuel très important, parce qu'il décrit les interactions entre stimuli.

On parle de masquage quand un stimulus, qui est visible seul, ne peut pas être perçu à cause de la présence d'un autre stimulus.

Dans le contexte du tatouage numérique, nous considérons le tatouage comme le stimulus qui est masqué par l'image originale, qui fonctionne donc comme fond. Ce masquage explique pourquoi le bruit du tatouage est désagréable dans certaines régions d'une image tandis qu'il n'est guère perceptible ailleurs.

En d'autres termes, le masquage augmente le seuil de la visibilité en fonction du contraste du masque C_M. Un modèle simple pour le masquage peut être formulé comme suit (voir figure 2):

$$k(C_M)=\begin{cases}1 & \text{si} \quad C_M<C_S \\ (C_M / C_S)^{\varepsilon} & \text{ailleurs}\end{cases} \qquad (12)$$

Le seuil de la visibilité du tatouage sans aucun masque C_0 est multiplié par ce facteur k. Normalement, C_S est proche du seuil du contraste du masque lui-même. ε dépend du type de stimuli et de leurs similarités; en général, $0{,}6 \le \varepsilon \le 1$

Le tatouage est modulé par la fonction $\alpha(x, y)$, qui tient compte du contraste local du bruit et

qui va représenter un facteur d'amplification appliqué à la marque pour l'ajuster en prenant en compte les caractéristiques du SVH (Système Visuel Humain) :

$$\alpha(x,y)=C_0 \times k(x,y) \times I_p(x,y) \qquad (13)$$

En raison de la structure multicanaux du système visuel humain, le masquage est le plus fort quand les stimuli ont des caractéristiques similaires, c'est à dire des fréquences, des orientations et des couleurs semblables. En général, le masquage entre des stimuli occupant des canaux différents est plus faible.

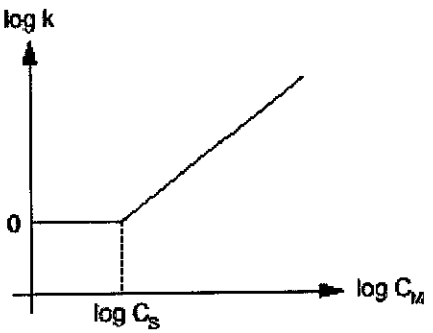

Figure-2 : Le modèle de masquage selon l'équation 12

3.3. APPLICATION AU TATOUAGE

Pour intégrer les deux phénomènes mentionnés ci-dessus dans un système de tatouage, il faut d'abord calculer le contraste local de l'image. La modélisation se complique parce que le tatouage en question représente un bruit blanc, et par conséquent comprend toutes les fréquences. Vu que l'œil humain est le plus sensible aux basses fréquences, nos efforts porteront sur cette région, car c'est là où l'on a le plus de marge pour cacher des informations dans l'image.

Parfois l'effet opposé se produit : un stimulus qui n'est pas visible seul peut être perçu à cause de la présence d'un autre.

Chapitre 4

LE FORMAT MPEG

(Moving Picture Expert Group)

- **Les fondements des algorithmes MPEG**
- **MPEG-1 : Un standard générique pour le codage d'images animées et du son associé pour les média de stockage numérique jusqu'à un taux de 1.5 Mbits/s**

LE FORMAT MPEG (MOVING PICTURE EXPERT GROUP)

4.1. LES FONDEMENTS DES ALGORITHMES MPEG [32]

Généralement, les séquences vidéos contiennent une quantité significative de redondances statistiques et subjectives. L'ultime but du codage source en vidéo est la réduction du bit-rate en exploitant ces redondances. La performance des techniques de compression vidéo dépend de la quantité de redondances dans les données aussi bien que dans les techniques de compression utilisées pour le codage.

Dans la pratique, le codage est un compromis entre les performances et la complexité de l'implémentation. Pour le développement des algorithmes de compression MPEG, il est important de prendre en compte les capacités des technologies actuelles et la durée de vie des standards.

En ce qui concerne le watermarking, on peut envisager une compression avec ou sans pertes des données vidéos. Le but de la compression sans pertes est de réduire la quantité de données vidéos, de les sauvegarder et de les transmettre en gardant la même qualité que l'original (La qualité de l'image décodée doit être la même que celle de l'image avant codage). A l'opposé, le but de la compression avec pertes (c'est ce que l'on vise avec les standards vidéos MPEG-1 et MPEG-2) est d'avoir un certain bit-rate lors du stockage et de la transmission.

Parmi les applications on trouve la transmission de la vidéo à travers des canaux de communication à bande passante faible ou limitée et le stockage. Plus le « bit-rate » demandé par le canal est petit, plus on devra compresser les données vidéos et plus la différence avec l'originale sera visible.

Dans ces cas le fort taux de compression est obtenu en dégradant la qualité de la vidéo. La qualité « objective » de l'image décodée est inférieur à celle de l'image avant codage (on utilise comme critère de qualité objectif l'erreur moyenne quadratique entre les deux images).

Le but principal des techniques de codage avec pertes est d'optimiser la qualité de l'image pour un bit-rate demandé sujet à un critère d'optimisation objectif ou subjectif. On doit noter que le degré de dégradation de l'image dépend de sa complexité autant que du perfectionnement de la technique de compression.

Des techniques simples de compression permettent une bonne reconstitution de l'image sans défauts visibles si les textures de l'image sont simples et que l'activité vidéo est basse. Dans ce qui va suivre nous introduirons les fondements du principe général de la compression MPEG

4.1.1. Le modèle du codeur source vidéo du MPEG

Les techniques de codage vidéo numérique MPEG sont de nature statistique. Les séquences

vidéos contiennent généralement des redondances statistiques dans les deux domaines temporel et spatial. La propriété statistique de base, sur laquelle reposent les techniques de compression du MPEG, est la corrélation inter-pixels. Elle inclut l'hypothèse d'un mouvement de translation simple corrélé entre des images consécutives. Ainsi, il est supposé que la valeur d'un pixel peut être prédite à partir des pixels proches dans la même image (en utilisant des techniques de codage intra-images) ou à partir des pixels d'une image proche (en utilisant des techniques inter-images). Il faut noter que durant certaines séquences (i.e. durant les changements de scènes dans une séquence vidéo) la corrélation temporelle entre les pixels des images proches est petite ou quasi nulle ; la scène vidéo assemble alors une collection d'images non corrélées. Dans ce cas, les techniques de codage intra-image sont appropriées pour explorer la corrélation spatiale afin d'avoir une bonne compression de données.

Les algorithmes de compression MPEG emploient les techniques de codage de la transformée en cosinus discrète (DCT) sur des blocs d'images de 8x8 pixels pour explorer efficacement la corrélation spatiale entre des pixels proches dans une même image. Cependant, si la corrélation entre les pixels dans des images proches, i.e. dans les cas où deux images consécutives ont un contenu similaire, il est préférable d'utiliser les techniques de codage DPCM (Differential Pulse Code Modulation) inter-images en employant la prédiction temporelle (prédiction de compensation de mouvement entre images).

Dans les schémas de codage vidéo MPEG, une combinaison adaptative de la compensation de mouvement temporelle est suivie par le codage par transformée des informations spatiales restantes pour obtenir une grande compression des données (Codage DPCM/DCT hybride de vidéo)

La figure 1 montre un exemple des propriétés de corrélation pixel à pixel intra image, modélisée ici par un modèle statistique simple. La supposition d'un modèle simple hérite des propriétés de corrélation de base de plusieurs images typiques. Les algorithmes MPEG reposent sur cette corrélation. On remarque une forte corrélation entre les pixels adjacents et une décroissance uniforme de la corrélation avec la croissance de la distance entre pixels. Nous allons utiliser le modèle supposé pour démontrer plus tard quelques propriétés du codage par transformation.

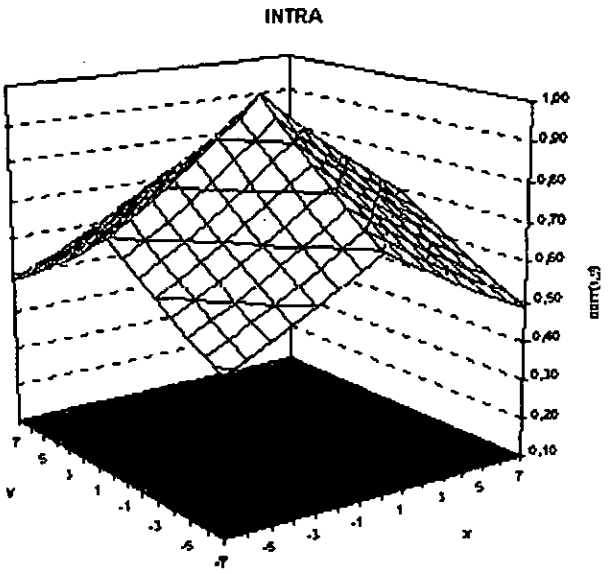

Figure-1 : Corrélation spatiale entre éléments d'image « typiques » calculée en utilisant un modèle d'image Gauss Markov AR avec une haute corrélation pixel-pixel. Les variables x et y décrivent la distance entre les pixels horizontalement et verticalement.

4.1.2. Sous échantillonnage et interpolation

Le principe de base du sous-échantillonnage est de réduire la dimension de la vidéo entrante (Dimension horizontale et/ou verticale) et aussi le nombre de pixels à coder avant le processus de codage. Dans certaines applications, la vidéo est sous-échantillonnée dans le domaine temporel pour réduire le taux d'image avant codage. Au récepteur les images décodées sont interpolées avant d'être affichées. Cette technique peut être considérée comme l'une des techniques les plus élémentaires de la compression qui fait aussi usage des caractéristiques physiologiques de l'œil humain et enlève ainsi la redondance subjective contenue dans les données vidéos (i.e. l'œil humain est plus sensible au changement de luminance qu'à ceux de chrominance). La méthode de codage MPEG divise, en premier, les images en composantes YUV (une de luminance et deux de chrominance). Après, la composante chrominance est sous échantillonnée relativement à celle de luminance avec un taux Y :U :V spécifique à l'application (i.e. avec le standard MPEG le taux 4 :1 :1 ou 4 :2 :2 est utilisé)

4.1.3. Prédiction de compensation de mouvement

La prédiction de compensation de mouvement est un outil puissant pour réduire la redondance temporelle entre les images et est beaucoup utilisée dans les standards de codage MPEG-1 et MPEG-2 comme technique de prédiction pour codage temporel DPCM. Le concept de compensation de mouvement est basé sur l'estimation du mouvement entre les images vidéos, i.e. si tous les éléments dans une scène vidéo sont spatialement déplacés, le mouvement entre les images peut être décrit par un nombre limité de paramètres de déplacement tels des vecteurs de

déplacement. Dans cet exemple simple la meilleure prédiction de la valeur du pixel courant est donnée par une prédiction compensée du mouvement des pixels d'une image précédemment codée. Habituellement, l'erreur de prédiction et les vecteurs de mouvement, sont transmis au récepteur. Cependant, coder une information de mouvement de chaque pixel de l'image codée est généralement ni voulu ni nécessaire. Comme la corrélation spatiale entre des vecteurs de mouvement est souvent élevée il est supposé qu'un seul vecteur de mouvement est représentatif pour le mouvement d'un bloc de pixels adjacents. A cette fin, les images sont séparées en blocs disjoints de pixels (i.e. 16x16 pixels dans le MPEG-1 et le MPEG-2) et seulement un vecteur de mouvement est estimé, codé et transmis pour chacun de ces blocs (figure 2).

Dans les algorithmes de compression MPEG les techniques de prédiction de compensation de mouvement sont utilisées pour réduire la redondance temporelle entre les images ; et seules les images d'erreur de prédiction sont codées (la différence entre les images originales et les images de prédiction de compensation de mouvement). En général, la corrélation entre les pixels (dans les images d'erreurs) à coder est réduite, en comparaison avec les propriétés de corrélation intra-image de la figure 1 due à la prédiction basée sur l'image codée précédemment.

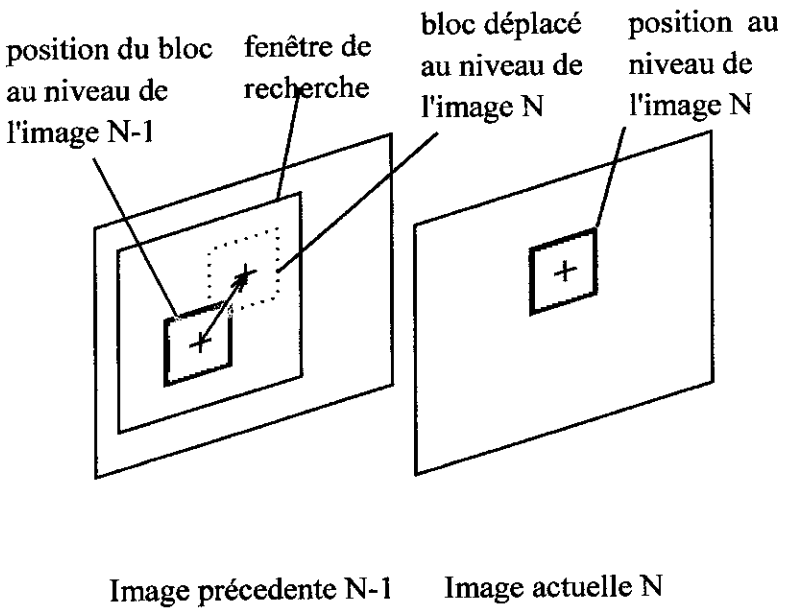

Figure-2 : Un vecteur de compensation (mv) est estimé pour chaque bloc de l'image N actuelle à coder. Le vecteur de mouvement pointe vers un bloc de référence de même taille dans l'image N-1 précédemment codées. L'erreur de prédiction de mouvement compensé est calculée en soustrayant chaque pixel dans un bloc de ses équivalents dont le mouvement a été décalé dans le bloc de référence de l'image précédente.

4.1.4. Codage par transformation

Le codage par transformation à été beaucoup étudié durant les deux dernières décennies et est devenu une méthode de compression très populaire pour le codage d'image et le codage de vidéo. L'objet du codage par transformation est de décorréler le contenu des images d'erreur inter- ou intra-image et de coder les coefficients de la transformée au lieu des pixels originaux des images. Pour ceci, l'image en entrée est découpée en blocs **b** de NxN pixels (en général N=8). La transformation peut être représentée par une opération matricielle utilisant une matrice A de dimension NxN pour obtenir les NxN coefficients de transformation -c- basée sur une transformation directe linéaire, séparable et unitaire.

$$c = A\, b\, A^T \qquad (1)$$

Ici, A^T représente la transposée de la matrice A. Il est à noter que la transformation est réversible, donc le bloc b original de NxN pixels peut être reconstruit en utilisant une transformation inverse linéaire et séparable.

$$b = A^T c\, A \qquad (2)$$

Parmi plusieurs alternatives, la transformée en cosinus discrète (DCT : Discrete Cosine Transform) est appliquée à des blocs images de 8x8 pixels, elle est devenue la plus utilisée pour le codage image et vidéo. En fait, les implémentations basées sur la DCT sont utilisées dans la plupart des standards de codage image et vidéo. Ceci est du à leurs performances de décorrélation très élevées et à la disponibilité des algorithmes de DCT rapides qui s'accordent bien avec l'implémentation en temps réel : les implémentations VLSI (Very Large Scale Integration) opèrent à des taux convenables pour un grand éventail d'applications vidéos disponibles dans le commerce aujourd'hui.

Le but majeur du codage par transformation est de rendre les coefficients de transformation aussi petits que possible pour les rendre insignifiants (en terme de mesures subjectives et statistiques), et donc de ne pas avoir à les coder pour la transmission. On doit aussi minimiser la dépendance statistique entre les coefficients dans le but de réduire la quantité de bits demandée pour coder le reste des coefficients. La figure 3 représente la variance (énergie) d'un bloc 8x8 de coefficients DCT intra-image basés sur un modèle statistique simple , (supposition déjà étudiée dans la figure 1). Ici, la variance pour chaque coefficient représente la variabilité de ce coefficient comme moyenne à travers un grand nombre d'images. Les coefficients avec une petite variance sont moins significatifs pour la reconstruction des blocs image que les coefficients à grande variance. Comme montré en figure 3 et en moyenne seulement un petit nombre de coefficients DCT ont besoin d'être transmis au récepteur pour obtenir une reconstruction approximative valable des blocs image. Mieux encore, les coefficients DCT les

plus significatifs sont concentrés autour du coin gauche supérieur (les coefficients bas de la DCT) et l'importance des coefficients diminue avec l'augmentation des distances. Ceci implique que les coefficients de hautes fréquences de la DCT sont moins importants pour la reconstruction que les coefficients de basses fréquences. En employant aussi la prédiction compensée de mouvement, la transformation utilisant la DCT donne une représentation compacte du signal DPCM temporel dans le domaine de la DCT. Celle-ci hérite essentiellement de la même cohérence statistique que le signal dans le domaine de la DCT pour les signaux intra-image en figure 3 (même de faible énergie). C'est la raison pour laquelle les algorithmes MPEG emploient le codage DCT pour la compression inter-image avec succès.

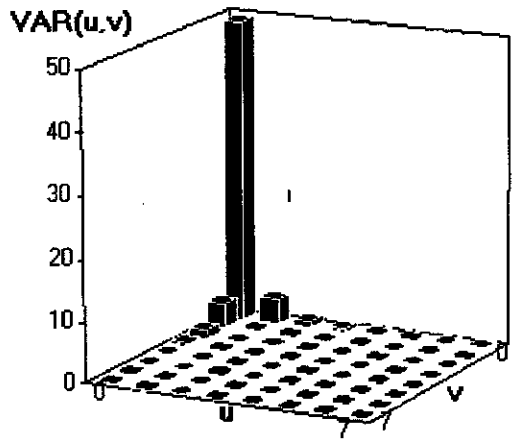

Figure-3 : Cette figure montre la distribution de la variance des coefficients DCT calculés « typiquement » comme la moyenne sur un grand nombre de blocs images. La variance des coefficients de la DCT est calculée en se basant sur le modèle statistique utilisé en figure 1. u et v décrivent les variables horizontale et verticale du domaine transformé dans le bloc 8x8. La variance est concentrée autour du coefficient DCT continu (u=0, v=0).

La DCT est liée de près à la Transformée de Fourier Discrète (TFD ou DFT en anglais) et il est important de réaliser qu'on peut donner aux coefficients de la DCT une interprétation fréquentielle proche de la DFT. Ainsi les coefficients bas de la DCT sont assimilés aux fréquences spatiales basses dans les blocs images et les hauts coefficients aux hautes fréquences. Cette propriété est utilisée dans le schéma de codage MPEG pour enlever la redondance subjective contenue dans les données de l'image basée sur les critères du système visuel humain. Comme le spectateur humain est plus sensible aux erreurs de reconstruction dans les basses fréquences que dans les hautes fréquences, une pondération (quantification) adaptative en fréquence des coefficients s'accordant avec la perception visuelle humaine est employée pour améliorer la qualité visuelle de l'image codée pour un bit-rate donné.

La combinaison des deux techniques décrites ci-dessus – prédiction de compensation de mouvement temporelle et le codage par transformation – est considérée comme l'élément clé des standards de codage MPEG. Un troisième élément caractéristique des algorithmes MPEG est que

ces deux techniques sont appliquées sur de petits blocs images (16x16 pixels pour la compensation de mouvement et 8x8 pixels pour le codage DCT). Pour cette raison les algorithmes de codage MPEG sont considérés comme des algorithmes hybrides DPCM/DCT basés sur des blocs.

4.2. MPEG-1 [32] : UN STANDARD GENERIQUE POUR LE CODAGE D'IMAGES ANIMEES ET DU SON ASSOCIE POUR LES MEDIA DE STOCKAGE NUMERIQUES JUSQU'A UN TAUX DE 1.5 MBITS/S

Les techniques de compression vidéo développées par le MPEG-1 couvrent beaucoup d'applications, des systèmes interactifs sur CD-ROM à la distribution de vidéo sur les réseaux de télécommunications. Le standard de codage vidéo MPEG-1 est dit générique. Pour supporter le large éventail des profils d'applications, le réglage des paramètres d'entrée, de la taille des images ainsi que de leurs taux de compression est laissé au soin de l'utilisateur. Cependant, le standard MPEG a recommandé un ensemble de paramètres : chaque décodeur compatible MPEG-1 doit être capable de supporter des paramètres de source vidéo correspondant à la taille TV : un minimum de 720 pixels par ligne et 576 lignes par image, un taux d'images de 30 images/seconde et un bit-rate de 1.86 Mbits/s au minimum. L'entrée vidéo standard consiste en un format d'images vidéos non entrelacé. On doit noter que le MPEG-1 n'est pas limité à ces paramètres.

Les algorithmes vidéos MPEG-1 ont été développés en respectant les activités du JPEG et du H.261.

4.2.1. Schéma de base du codage MPEG–1

La technique de compression du standard MPEG est basée sur la structure macro-blocs, le repliement conditionnel et la compensation de mouvement. Comme le montre la figure 4 -a, l'algorithme de codage du MPEG-1 code la première image d'une séquence vidéo avec un codage en mode dit intra-image, et obtient donc ce que l'on appelle des images I. Chaque image subséquente est codée en utilisant un codage dit prédiction inter-image (images P) - seuls les données de la plus proche image P ou I précédente sont utilisées pour la prédiction. Le procédé qu'utilise l'algorithme du format MPEG-1 est basé sur une décomposition de chaque image de la séquence vidéo en blocs. Chaque couleur d'une image d'une séquence vidéo est divisée en macro-blocs (sans recouvrement) comme le montre la figure 4 - b. Chaque macro-bloc contient les données concernant la luminance et les chrominances; il y a quatre blocs de luminance (Y1,Y2,Y3,Y4) et deux blocs de chrominance (U,V). Chacun d'eux (les blocs NDLR) a une

taille de 8×8 pixels. Ainsi, le taux d'échantillonnage des données de luminances et de chrominances Y :U :V est de 4 :1 :1.

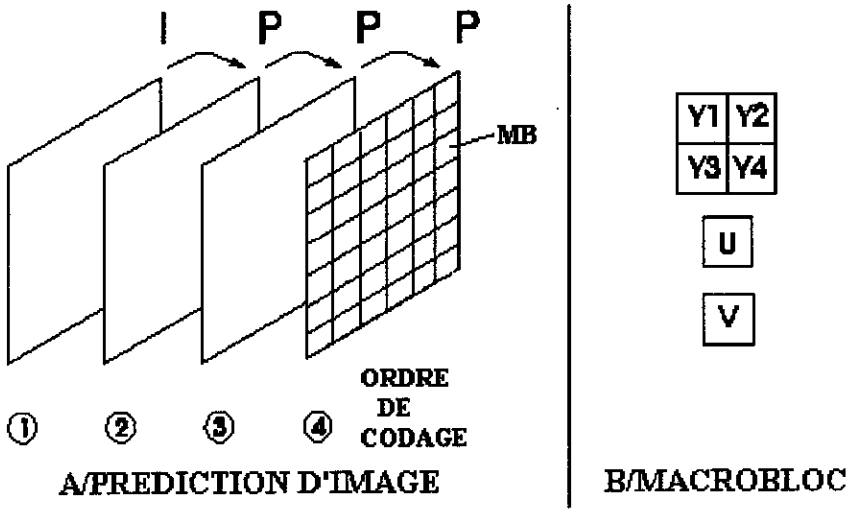

Figure-4 : Illustration d'images I et P dans une séquence vidéo.

Le schéma bloc de base d'un codeur MPEG-1 et d'une structure de décodage est explicité par la figure 5. La première image de la séquence vidéo est codée en mode intra- image, donc sans aucune image de référence passée ou future. Au niveau du codeur une DCT est appliquée à chaque bloc de 8×8 pixels, qu'il soit de luminance ou de chrominance ; en sortie de ce traitement chacun des 64 coefficients est uniformément quantifié (Q). Les tailles des niveaux de quantification (Step size SZ) utilisés pour quantifier les coefficients de DCT au niveau d'un macro-bloc sont transmises au récepteur. Après quantification, le coefficient de plus basse fréquence (coefficient DC) est traité différemment des autres coefficients (coefficients AC). Le coefficient DC correspond à l'intensité moyenne des composants du bloc, de plus, ce coefficient est codé en utilisant une méthode de prédiction différentielle DC. Les coefficients différents de zéro restants et leurs positions sont ensuite lus en « ZIG ZAG » et codés en utilisant un codage entropique dit ***Run-lenth entropy*** avec une table de codage de mot de longueur variable (***Variable Lenth Coding*** ou ***VLC)***.

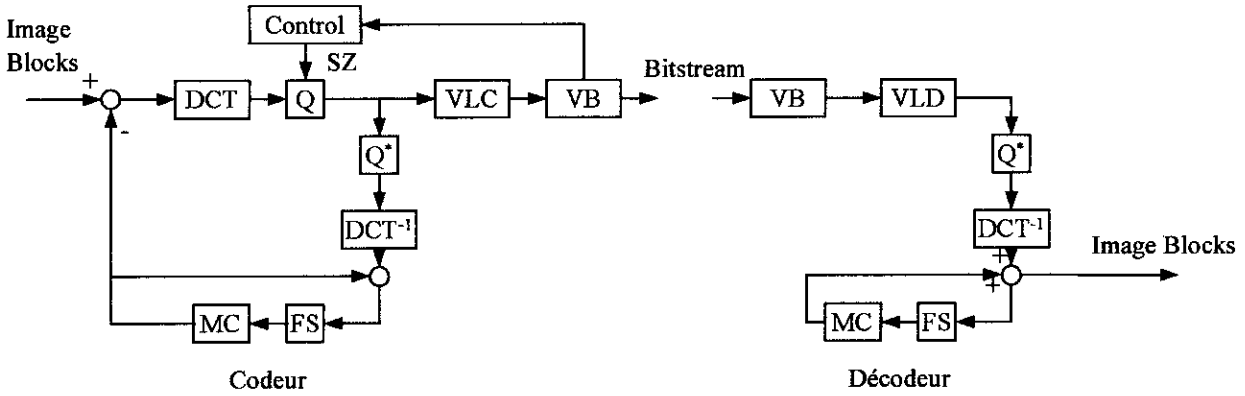

Figure-5 : Schéma bloc d'une structure de base d'un codeur /décodeur DCT/DPCM.

Le concept de lecture en « ZIG ZAG » des coefficients de DCT est explicité par la figure 6. Le fait de lire en « ZIG ZAG » ces coefficients (signal bi-dimensionel) puis d'appliquer un codage en longueur variable sert à la mise en forme du signal image bi-dimensionnel en un bitstream (flux de bits) mono-dimensionnel. Les valeurs des coefficients AC, différents de zéro, quantifiés (length) sont détectées lors de la lecture de la ligne en même temps que la distance (run) entre deux coefficients successifs. Chaque paire (run, length) consécutive est codée en transmettant seulement un mot de code VLC (code word). Le but d'appliquer une lecture en « ZIGZAG » est donc de tracer les coefficients de DCT de basses fréquences (contenant la plupart de l'énergie) avant ceux de haute fréquences.

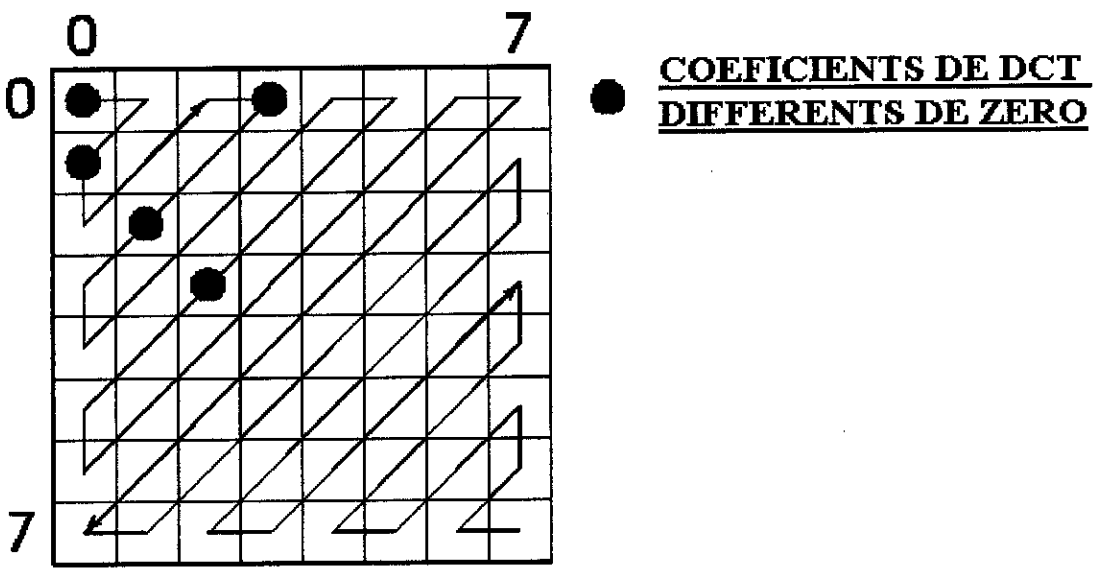

Figure-6 : Lecture en « ZIG ZAG »des coefficients de DCT dans un bloc de 8×8 pixels. Seuls les coefficients différents de zéro sont codés. Les positions possibles des coefficients différents de zéro sont indiquées dans la figure. En référence à la figure 3, le coefficient de DCT(0,0) contient la majeure partie de l'énergie de l'image, cette énergie est concentrée autour de ce coefficient.

Le décodeur fait l'opération inverse ; en effet, ce dernier extrait et décode le mot codé du bitstream (en utilisant un codage du mot à longueur variable), ceci afin d'obtenir la position et la valeur des coefficients différents de zéro pour chaque bloc. Les valeurs des pixels de chaque bloc sont obtenues à partir de la reconstruction (Q*) de tous les coefficients différents de zéro appartenant à ce bloc et en appliquant une DCT^{-1}. En utilisant un tel traitement sur tout le bitstream (flux de bits), tous les blocs codés de l'image seront reconstruits.

En ce qui concerne le codage P, l'image de type I ou P précédente est sauvegardée (frame store ou FS) au niveau du codeur et du décodeur, puis une compensation de mouvement (MC) est appliquée sur chaque macro-bloc de base - juste un vecteur de compensation est estimé entre l'image N et N-1 pour un macro-bloc particulier à coder. Ces vecteurs de mouvements sont codés et transmis au récepteur, l'erreur de prédiction de compensation de mouvement est calculée en soustrayant chaque pixel dans un macro-bloc avec son équivalent au niveau de l'image précédente dont le mouvement est décalé. Une DCT 8×8 pixels est alors appliquée à chacun des blocs contenus dans un macro-bloc, puis on quantifie et on code les coefficients en utilisant un codage entropique (VLC) avec un codage run-length. Un buffer vidéo (vidéo buffer VB) est utilisé pour s'assurer que le débit en sortie du codeur est constant ; la taille des niveaux de quantifications (SZ) peut être ajustée pour chaque macro-bloc afin d'atteindre un bit-rate (débit) déterminé et éviter l'overflow ou l'underflow du buffer.

Le décodeur utilise le procédé inverse pour reproduire les blocs de l'image N. Après décodage à longueur de mot variable (variable length word VLD) du mot contenu dans le buffer (VB), les valeurs de l'erreur de prédiction des pixels sont reconstruites (Q^{*-1} puis DCT^{-1}). La compensation de mouvement des pixels à partir de l'image précédente contenue dans la sauvegarde (FS) est ajoutée à l'erreur de prédiction de compensation de mouvement pour reconstruire un macro-bloc particulier de l'image N.

L'avantage d'un codage utilisant la prédiction de compensation de mouvement (à partir de l'image N-1 reconstruite dans un codeur MPEG) est illustré par les figures 7 - a à 7 - d, et ceci pour une séquence test typique. La figure 7 - a montre une image à un instant N à coder, la figure 7 - b l'image reconstruite à l'instant N-1 qui est sauvegardée au niveau du frame store (FS) du récepteur et de l'émetteur. Les vecteurs de mouvements (MV) (figure 2) utilisés en figure 7 - b ont été estimés par une procédure d'estimation du mouvement au niveau du codeur qui fournit une prédiction des mouvements des macro-blocs au niveau de l'image N par rapport à l'image N-1. La figure 7 - c donne le signal différence entre l'image N et N-1 (image N - image N-1) qui est obtenu si aucune prédiction de compensation de mouvement n'est utilisée par le procédé de codage ; dans ce cas tous les vecteurs de mouvements sont pris égaux à zéro. La figure 7 - d donne la différence des signaux images compensés en mouvement quand les vecteurs de mouvement sont utilisés pour la prédiction en figure 7 - b. Il apparaît clairement que les signaux résiduels à coder sont grandement réduits en utilisant une compensation de mouvement en comparaison avec la différence des signaux images à coder en figure 7 - c.

Figure-7-a

Figure-7-b

Figure-7-c

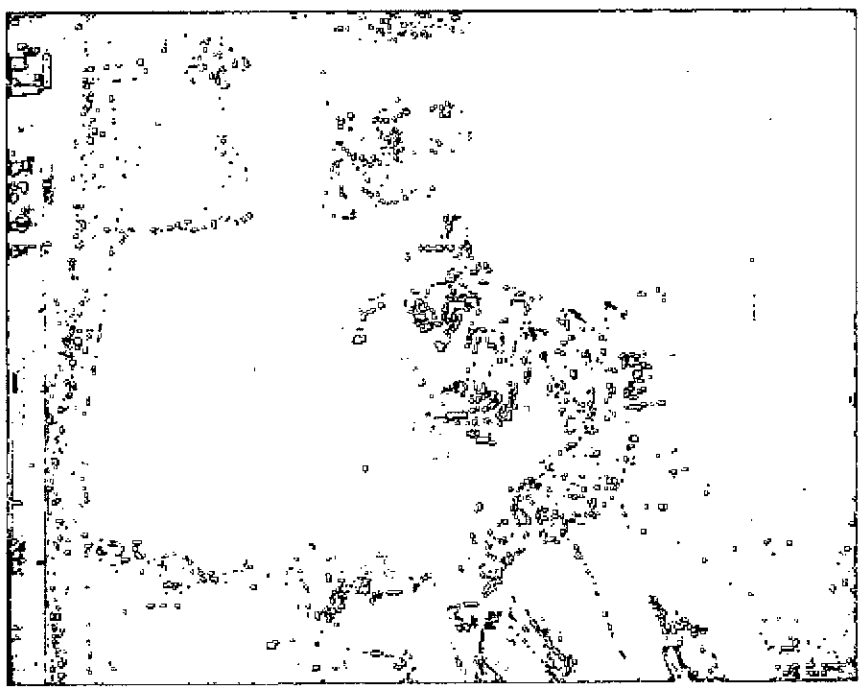

Figure-7-d

4.2.2. Mise à jour conditionnelle

Une caractéristique essentielle de l'algorithme de codage MPEG-1 est la possibilité de mettre à jour les informations du macro-bloc – au décodeur seulement si nécessaire - si le contenu du macro-bloc a changé par rapport au contenu du même macro-bloc de l'image précédente (mise à jour conditionnelle). La clé d'un codage efficace d'une séquence vidéo à des débits faibles est la

sélection du mode de prédiction approprié. Le standard MPEG distingue trois différents types de codage des macro-blocs (type MB) :

- **Type *skipped MB :*** une prédiction à partir de l'image précédente avec les vecteurs mouvements pris égaux à zéro. Aucune information concernant les macro-blocs n'est codée, ni transmise.
- **Type *Inter MB :*** une prédiction utilisant la compensation de mouvement à partir de l'image précédente est appliquée : le type MB, l'adresse MB et, si nécessaire, le vecteur mouvement, les différents coefficients de DCT et la taille des niveaux de quantification sont transmis.
- **Type *Intra MB :*** Aucune prédiction utilisant la compensation de mouvement à partir de l'image précédente n'est appliquée (prédiction basées sur les image I seulement) : dans ce cas seuls le type MB, l'adresse MB, les différents coefficients de DCT et la taille des niveaux de quantification sont transmis au récepteur.

4.2.3. Fonctionnalités spécifiques de sauvegarde du média

Pour accéder à la vidéo à partir d'une sauvegarde du media, l'algorithme de compression MPEG-1 doit satisfaire à certaines fonctionnalités comme l'accès aléatoire, l'avance et le retour rapide (FF et RF). Le concept d'image B (image bidirectionnelle prédite / interpolée) a été introduit par le standard MPEG-1 afin d'incorporer les caractéristiques requises pour une sauvegarde du media et pour utiliser au mieux les avantages de la compensation de mouvement.

Ce concept est explicité en figure 8 par un groupe d'images d'une séquence vidéo. Les trois types d'images considérées sont :

- Image I : Les images Intra sont codées sans références comme introduit précédemment par la figure 4. Elles permettent d'avoir un accès aléatoire et des fonctionnalités de FF/RF mais ne permettent pas d'avoir des taux de compression élevés.
- Image P : Les images prédites par procédés inter-image sont codées en utilisant comme référence la plus proche image I ou P précédemment codée. En général, on incorpore dans le procède de codage la compensation de mouvement pour augmenter l'efficacité de ce dernier. Malgré le fait que les images P sont en général utilisées comme référence pour la prédiction d'images futures ou passées ,elles ne peuvent fournir des points d'accès pour d'éventuelles fonctionnalités d'accès aléatoire ou de FF/RF ou d'édition.
- Images B : Les images bidirectionnelles et interpolées emploient la technique de compensation de mouvement. La prédiction des ces images requiert quant à elle des images de références passées et futures. L'utilisation d'images B permet d'obtenir de haut taux de compression. Il est a noter que les images B ne sont jamais prises comme référence.

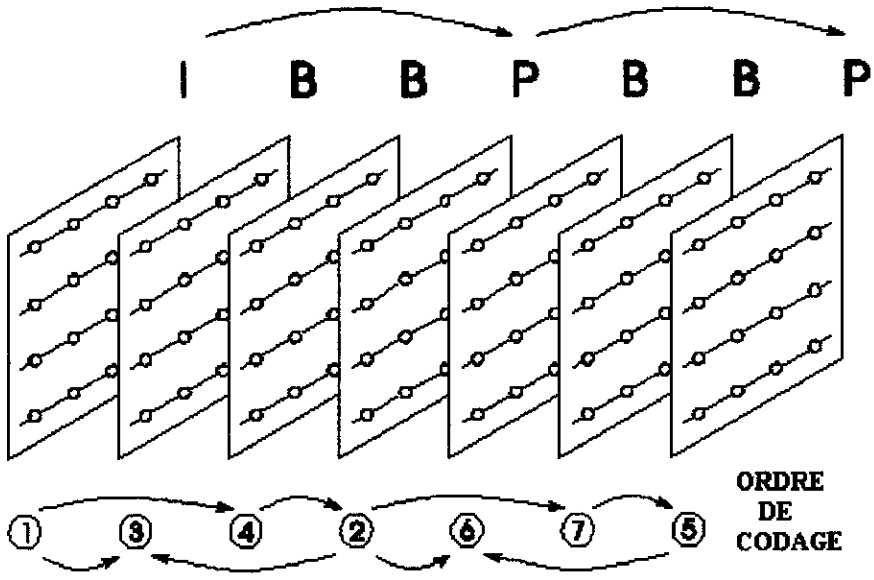

Figure-8 : Images I, P,B utilisées dans une séquence MPEG-1. Les images B sont codées en utilisant la prédiction de compensation de mouvement à partir des deux images codées les plus proches (images I ou P). L'ordre de codage des images est flexible et la direction de prédiction est indiquée dans la figure.

L'utilisateur peut arranger les types d'images dans une séquence vidéo avec un grand degré de flexibilité. En règle générale, une séquence vidéo utilisant des images I uniquement (IIIIII...) permet un plus grand degré d'accès aléatoire, de fonctionnalités de FF/RF et d'éditabilité, mais au détriment du taux de compression. Une séquence codée avec une mise à jour régulière en image I et sans images B (i.e. IPPPPPPIPPPP...) permet d'atteindre des taux de compression modérés mais en gardant un certain degré d'accès aléatoire, de fonctionnalités de FF/RF et d'édition. L'incorporation des trois types d'images, ainsi qu'explicité en figure 8 (IBBPBBPBBIBBP...), permet une compression plus efficace et un degré d'accès aléatoire, de fonctionnalités de FF/RF et d'édition acceptable, mais en augmentant les délais de décodages significativement, ce qui est inacceptable pour des applications en temps réel tel la vidéoconférence.

4.2.4. Contrôle de débits

Une caractéristique importante de l'algorithme de codage MPEG-1 est la possibilité de modifier le bit-rate (débit de bit) et donc la qualité de la vidéo reconstruite pour certaines applications. Ceci se fait en ajustant la taille des niveaux de quantification des coefficients de DCT permettant une sauvegarde ou une transmission de la vidéo avec de hauts taux de compression.

Le standard MPEG-1 permet de sélectionner différentes valeurs de quantification pour chaque macro-bloc codé au niveau du codeur - ceci permet un grand degré de flexibilité pour allouer des bits au niveau des images, si nécessaire, pour améliorer la qualité de l'image ; de

plus, cela permet la génération de débits de bits constants ou variables pour la sauvegarde ou la transmission en temps réel de la vidéo compressée.

Les informations décrivant le contenu de la vidéo compressée sont variables de nature, ceci est dû de manière générale par le contenu variable des images vidéos successives. Pour sauvegarder ou transmettre la vidéo avec un bit-rate constant, il est nécessaire d'utiliser un buffer vidéo (VB) comme le montre la figure 5. A l'entrée du buffer (au niveau du codeur) le flux de bits est variable alors qu'en sortie il sera constant. A l'inverse, au niveau décodeur, le flux de bits est constant en entrée et variable en sortie. En général, le codeur et le décodeur MPEG utilisent des buffers de même taille pour éviter les erreurs de reconstruction.

L'algorithme de contrôle au niveau du codeur ajuste la taille des niveaux de quantification (SZ) en fonction du contenu des séquences vidéos et de son activité pour s'assurer que les buffers ne seront par surchargés (overflow) - ceci en essayant de garder le buffer le plus rempli possible, pour augmenter la qualité de l'image et de la vidéo en même temps.

En théorie, on peut éviter la surcharge du buffer en utilisant un buffer de grande taille. Cependant, en plus du prix coûteux de ces dispositifs, d'autres désavantages apparaissent, comme le temps de traitement accru entre le codeur et le décodeur. Le standard MPEG a défini la taille minimale des buffers vidéos du décodeur. Cette valeur est en général identique à la valeur maximale de la taille du buffer vidéo que le codeur utilise pour générer le bitstream vidéo. Afin de réduire les délais et la complexité du codeur il est possible de choisir un buffer de taille virtuelle au niveau du codeur. Ce buffer sera de taille plus petite que celle minimale au décodeur. Cette valeur virtuelle sera transmise au décodeur avant l'envoi du flux de bits vidéo.

4.2.5. Codage des sources vidéos entrelacées

Le format vidéo standard de l'MPEG-1 est non entrelacé. Cependant, le codage vidéo couleur entrelacé (cas de la télévision) avec 525 et 625 lignes à 29.97 et 25 images par seconde respectivement est une application importante du standard MPEG-1. Une suggestion dans ce sens a été faite pour le codage des signaux télévisuels couleurs numériques Rec.601, basée sur la conversion de la source entrelacée en un format intermédiaire progressivement. D'une manière générale, seule une trame de chaque image vidéo entrelacée est codée. Les étapes de la procédure sont décrites en détail dans le document « informative Annex of the MPEG-1 International Standard ».

Chapitre 5

ETAT DE L'ART

- **Quelques travaux réalisés dans le domaine du watermarking**
- **Une approche basée sur le codage source et canal, appliquée au Watermarking vidéo**
- **Data hiding d'un média vidéo**
- **Une méthode robuste de Watermarking vidéo**
- **Data hiding d'un média vidéo basé sur une approche multi-niveaux**
- **Watermarking des vidéos pré-compressées en utilisant l'étalement de spectre**

ETAT DE L'ART

5.1. QUELQUES TRAVAUX REALISES DANS LE DOMAINE DU WATERMARKING

Brassil et al [33] ont travaillé sur différentes méthodes de watermarking dans le but de marquer du texte dans un document avec un mot code binaire unique. Ce mot code identifie les utilisateurs légitimes du document. Le mot code est tatoué en modifiant la structure du document, cette modification porte sur la modulation des lignes, les espacements des inter-mots ou le style des caractères. Ainsi, la présence du mot code peut être détectée en comparant le document avec l'original. De plus, les opérations standards effectuées sur ce genre de document (photocopie, scan,...) n'affectent pas le mot code.

Kurak et Mc hugh [34] utilisent les données redondantes des images numériques pour dissimuler des informations. Leur but est de transmettre de dangereux virus (ou chevaux de Troie) au niveau des bits les moins significatifs du « data stream ». Kurak et Mc hugh ont pu remarquer que le fait de visionner l'image ne permet pas toujours de détecter la présence des données corrompues. Il est donc possible d'exploiter la faible sensibilité de l'œil humain (et la faible qualité des moniteurs vidéo) pour dissimuler une image de moins bonne qualité dans l'image hôte. Walton [35] a développé pour sa part une technique de watermarking dite fragile, en ce sens qu'il introduit un « cheksum » au niveau des bits les moins significatifs pour se prémunir contre d'éventuelles altérations de l'image. Dautzebzeg et Boland [36] ont considéré l'utilisation des bits les moins significatifs dans le but d'insérer une marque dans l'image ; cette approche a donné de mauvais résultats car la marque ne pouvait survivre aux traitements de compression comme le JPEG par exemple, qui affecte le bit le moins significatif durant l'étape de quantification.

Zhao et koch [37] ont concentré leurs efforts sur une approche radicalement différente des précédentes, qui se base sur l'algorithme de compression JPEG et sur la segmentation de l'image en blocs de 8×8 pixels. Seulement huit coefficients occupant des positions déterminées du bloc DCT sont utilisés pour dissimuler la marque. Ceux-ci comprennent trois composantes de basses fréquences à l'exclusion du coefficient qui contient la valeur moyenne du bloc (coefficient de coordonnée (0,0)) et des deux coefficients de basses fréquences de coordonnées (0,1) et (1,0), trois autres coefficients DCT sont choisis d'une manière aléatoire.

Matsui et Tanaka [38] ont utilisé la prédiction linéaire pour marquer la vidéo et les images. Leur approche consiste à confondre le watermark avec le bruit. D'une certaine manière on peut considérer cette méthode comme étant perceptiblement adaptative car le bruit de quantification se concentre sur les bords et aux endroits fortement texturés. Cox et al [39] pensent que cette

méthode ne serait pas robuste au « cropping ». O Ruanldh et al [40] et Cox et al ont développé une méthode qui utilise une transformation perceptiblement adaptative. Contrairement aux approches précédentes, cette méthode est fondée sur une dissimulation de la marque au niveau des coefficients les plus significatifs. Cette approche consiste donc à diviser l'image en blocs par une transformation de domaine basée sur la DCT, la transformée HAR ou la transformée en ondelettes (wavelet transform)

L'utilisation de la transformation de domaine possède beaucoup d'avantages. Elle permet de dissimuler la marque adaptativement aux endroits où ils seront les moins visibles. et donc d'avoir une image qui ressemblera fortement à l'originale. La marque peut aussi être irrégulièrement distribuée au niveau des sous-blocs de l'image ce qui rendra le décodage et la lecture de la marque difficile à tout contrevenant en possession de copies de l'image indépendantes.

La méthode proposée par Cox et al diffère de celle de O Ruanldh et al. La différence principale entre ces méthodes est le codage et le décodage de la marque. Cox et al. dissimulent une séquence gaussienne unique au niveau des coefficients. La distribution gaussienne est utilisée pour se prémunir contre d'éventuelles attaques par comparaison de l'image marquée avec d'autres copies indépendantes. O Ruanldh et al ont une approche qui consiste a dissimuler la marque directement au niveau des coefficients de l'image. Un des avantage de cette méthode est de ne pas avoir besoin d'une base de données des marques utilisées.

D'autres approches ont été proposées dans le domaine du watermarking vidéo. Parmi celles-ci on remarquera celle Hartung [41], dont le but est de palier à certaines faiblesses d'autres méthodes. Cette méthode propose de dissimuler la marque directement au niveau du bitstream MPEG 2, ce qui représente une continuité par rapport aux autres méthodes qui proposaient de dissimuler la marque en utilisant les algorithmes de compression existant c'est à dire le MPEG1, le H261 et H263. Cependant, cette méthode requiert le calcul de la DCT inverse et de la compensation de mouvement.

Bien d'autres méthodes existent encore : la méthode de Langelaar qui consiste à dissimuler un label au niveau des intra-images dans le domaine DCT [42]. Linnarz [43] a aussi proposé une méthode dont le but est de dissimuler des données au niveau de la séquence d'images constituant la vidéo, en suivant un ordre prédéfini de codage du type des images. Les méthodes proposées ci-dessus ont toutes démontré des faiblesses aux attaques par changement du GOP (Group Of Picture) de la vidéo ainsi marquée. Pour résoudre ces problèmes, Dung Ryung Kim et Gung Han Park ont proposé de détecter le label à partir du domaine DCT de chaque image I et au niveau du bit stream. Pour cela, une table de correspondance entre le « pixel pattern » d'un bloc et la séquence de signe des coefficients DCT de basses fréquences est exploité. Un des avantages de cette méthode est le faible temps de traitement du processus de watermarking car ni la DCT inverse ni la compensation de mouvement n'est requise pour la dissimulation.

Une autre approche dans un but de contrôle d'accès a été proposée par Min Wu et Hong

Heather yu [44]. Une des caractéristiques importantes d'une séquence vidéo est que les images la composant sont fortement similaires (exception faite des scènes changeantes et des mouvements rapides) ; il est donc possible d'ajouter, d'enlever quelques images de la séquence ou même de changer l'ordre des images sans pour autant l'altérer irrémédiablement. Ces manipulations sont autant d'attaques possibles que l'on doit prendre en compte lors de la conception de systèmes robustes de watermarking.

Min Wu et Hong Heather yu se sont donc concentrés sur une approche dite de redondance pour sa simplicité de traitement.

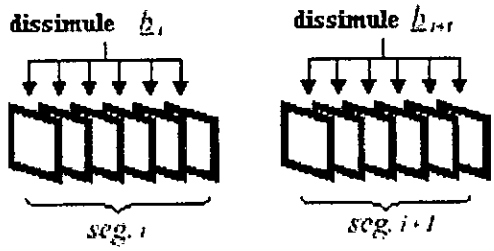

Figure-1 : Illustration du domaine de dissimulation

Comme illustré en figure 1 la vidéo est partitionnée en segments, chacun d'eux est constitué d'images similaires consécutives. Il sera donc possible de dissimuler la même donnée dans chaque image du segment, ce qui permet de combattre les différentes attaques citées. De plus, cela va permettre une meilleure robustesse par rapport aux distorsions et une meilleure détection de la marque. L'extraction peut être faite via un vote où le poids le plus fort sera attribué à l'image la moins distordue. Afin de combattre une éventuelle attaque qui consisterait en une inversion des images dans la séquence vidéo, un index est dissimulé au niveau de chaque image. Cette information est appelée FRAME SYNCH et sera une partie de la donnée de contrôle. Elle servira lors de la détection de la marque et de la détection du « jittering » de l'image. De plus, cette méthode utilise l'approche classique qui consiste à utiliser le data hiding, et ainsi dissimule la même donnée d'utilisateur et l'index dans chaque image pour combattre le frame jittering.

La vidéo pouvant être vue comme étant une séquence d'images consécutives, le watermarking vidéo peut donc se baser sur la plupart des méthodes de watermarking d'images. Cependant, des techniques propres à la vidéo existent, une des techniques les plus anciennes est la méthode proposée par Cox et al qui est une technique de spread spectrum (étalement du spectre) [45].

L'idée principale sur laquelle repose cette technique est la distribution du message sur une large bande de fréquences du stego-objet. Beaucoup de chercheurs dans le domaine se sont basés sur l'utilisation des coefficients de DCT (Discrete Cosine Transform) ou de la transformée en ondelettes (wavelet transform) pour dissimuler la marque.

Dans ce domaine, plusieurs méthodes sont proposées dans le but de dissimuler une marque

audio ou vidéo dans une séquence vidéo. Par exemple, Swanson et al [46] ont proposé un algorithme de data hiding pour dissimuler une vidéo compressée et des données audio dans une séquence vidéo. La marque est dissimulée au niveau des coefficients DCT. Ces auteurs ont démontré la robustesse de cette méthode par rapport à un bruit gaussien additif et une compression MPEG. Plus récemment, Mukhenjee et al [47] ont présenté une technique dont le but est de dissimuler du son dans une séquence video. Ils utilisent un treillis multidimensionnel pour dissimuler un signal parole de 8Khz.

Jij Chae et DS Majunath utilisent une technique de data hiding robuste au codage MPEG. La composante clé de cette technique est l'utilisation d'un treillis multidimensionnel [48,49]. La marque et la séquence vidéo sont transformées en utilisant des blocs DCT 8×8 pixels. Les coefficients de la marque sont quantifiés et codés en utilisant le treillis multidimensionnel puis insérés au niveau des coefficients DCT de l'hôte. Cette insertion est adaptative à la texture du contenu des blocs DCT. Enfin, la vidéo est compressée (au standard MPEG) ; ensuite, la marque est détectée à partir de la vidéo compressée.

5.2. UNE APPROCHE BASEE SUR LE CODAGE SOURCE ET CANAL APPLIQUEE AU WATERMARKING VIDEO [50]

La figure2 présente un schéma classique de watermarking. Le résultat de la dissimulation de la marque est l'apparition d'une distorsion par rapport à la vidéo originale, ce qui se traduit par une erreur quadratique moyenne (Mean squared error MSE). Afin d'assurer une transparence parfaite, le MSE doit être le plus faible possible. Lors de la distribution de la vidéo, cette dernière va subir des traitements (compression et autres transformations standards). On désire donc qu'après extraction le MSE_S « erreur moyenne de la signature » extraite, soit le plus faible possible pour que la marque extraite soit semblable à la signature dissimulée. Ce qui nous ramène à un problème de codage source et canal.

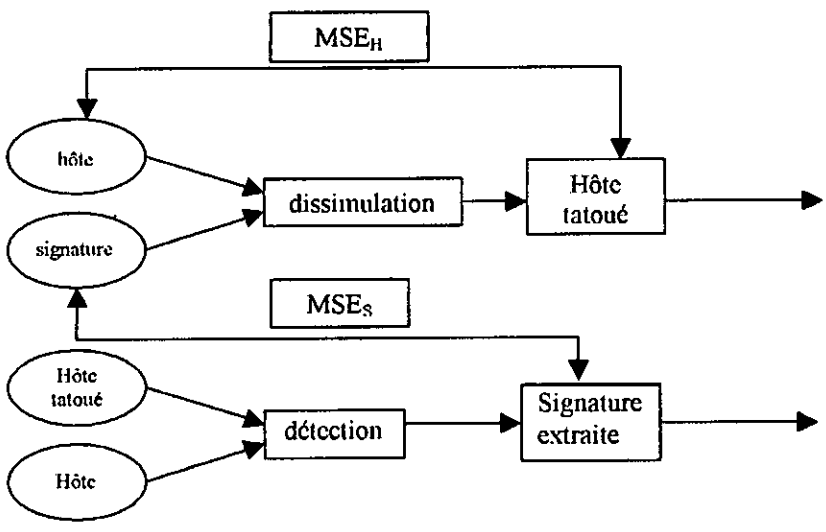

Figure-2 : Data Hiding

Dans la méthode décrite, les données hôtes sont transformées orthogonalement avant dissimulation de la donnée, on notera que cette transformation dépend du standard utilisé. Considérons une donnée hôte ($x_1,\ldots,x_N$) transformée en un ensemble de coefficients ($c_1,\ldots,c_N$). La transformation de données ainsi effectuée, et la dissimulation de la donnée à tatouer perturbent les coefficients ($c_1\ldots\ldots c_N$), ce qui se traduit par un nouvel ensemble de coefficients hôtes ($\overline{c_1},\ldots,\overline{c_N}$). La transformation de domaine inverse donnera une donnée tatouée ($\overline{x_1},\ldots,\overline{x_N}$). Etant donné que la transformation est orthogonale, l'erreur moyenne quadratique de la donnée hôte est :

$$MSE_H = \frac{1}{N}\sum_{i=1}^{N}\left|x_i - \overline{x_i}\right|^2 = \frac{1}{N}\sum_{i=1}^{N}\left|c_i - \overline{c_i}\right|^2 \qquad (1)$$

Un seuil de transparence ou contrainte de transparence P est fixé pour donner une information sur l'efficacité de la dissimulation, plus ce seuil est bas, plus la dissimulation est efficace :

$$\frac{1}{N}\sum_{i=1}^{N}\left|x_i - \overline{x_i}\right|^2 < P \Rightarrow \frac{1}{N}\sum_{i=1}^{N}\left|c_i - \overline{c_i}\right|^2 < P \qquad (2)$$

De plus, sachant que le volume de données N est assez grand dans une vidéo, les auteurs ont jugé nécessaire de grouper les N coefficients en vecteurs de dimension K avec K<<N, en satisfaisant pour chaque vecteur la contrainte de transparence. Ainsi, il sera nécessaire de dissimuler ou de perturber seulement un nombre M de coefficients sur les N coefficients, par exemple les coefficients d'une bande ou d'une sous-bande de fréquences de l'image de la séquence vidéo. Les M coefficients sont donc groupés en M/K vecteurs notés v_j ; j=1..M/K et les vecteurs perturbés sont notés $\overline{v_j}$; j=1..M/K. La condition de transparence va donc s'écrire :

$$\frac{1}{K}\left\|v_j - \overline{v_j}\right\|^2 < P_c \text{ avec } \quad P_c = \frac{N}{4} \times P \quad \text{ avec } \quad j=1 \ldots M/K \qquad (3)$$

Le principe général de dissimulation est explicité en figure 3.

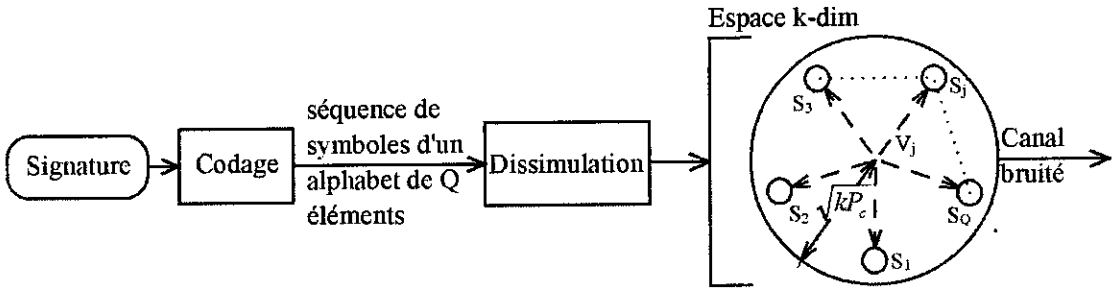

Figure-3 : Principe de dissimulation

La signature (donnée à dissimuler) est codée, ce qui génère une séquence de symboles d'un alphabet de Q éléments ($s_1,\ldots,s_Q$). Il est à noter que les valeurs de s_i sont toutes à une distance euclidienne n'excédant pas $\sqrt{K \times P_c}$ de v_j afin de répondre à la condition de transparence.

Les perturbations possibles constituent ce qui est appelé le code Book du canal. Ce code Book est reéchelonné par un facteur α. Ainsi, les vecteurs perturbés sont donnés par :

$$\overline{v_j} = v_j + \alpha \times C(s_i) \qquad (4)$$

où $C(s_i)$ est l'ensemble des vecteurs qui constituent le « code Book » de taille Q.

Le principe d'extraction est explicité en figure 4. Considérons que le $j^{éme}$ vecteur distribué $\overline{v_j}$ correspondant au symbole s_i a été reçu comme étant w_j. Si w_j est estimé comme faisant toujours partie de la zone de décision de s_i alors le symbole s_i sera extrait sans erreurs. Les zones de décision prises dépendent d'un modèle statistique choisi pour un type de bruit additif.

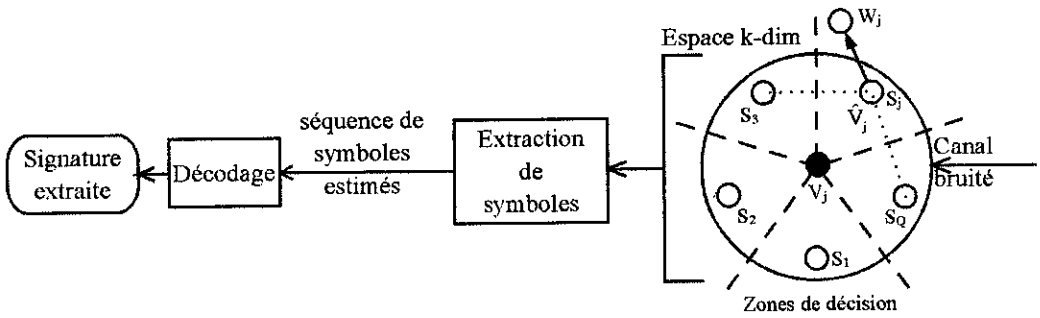

Figure-4 : Principe d'extraction

La séquence est extraite puis décodée pour reconstruire la signature. Cependant, ici la vidéo originale est nécessaire ce qui constitue un désavantage. En général, les codes choisis sont des structures de codage multidimensionnel.

Le principe général de dissimulation est de transformer chaque image de la vidéo en utilisant

par exemple une transformation par ondelettes. Les coefficients transformés sont groupés en vecteurs, la signature est quantifiée, puis les indices sont dissimulés au niveau des coefficients des vecteurs dans une bande ou sous-bande en utilisant un codage approprié. La même donnée signature peut être répétée au niveau de quelques images successives de la séquence vidéo pour optimiser la robustesse du système par rapport aux opérations de compression à haut taux de compression.

On choisit la bande des basses fréquences pour la dissimulation de la signature, ce qui réduit la probabilité de destruction de la donnée et n'affecte pas l'efficacité du codage, cependant la distorsion introduite est plus importante.

Dans le cas où l'extraction doit se faire sans faire appel à l'original de la vidéo, le choix des basses fréquences est inadéquat. Le choix de cette option d'extraction a amené les auteurs à s'orienter vers une autre manière de procéder au lieu d'utiliser la vidéo originale pour dissimuler la signature, c'est une vidéo convertie qui est utilisée comme base pour l'extraction. En effet partant du constat que les images naturelles sont d'énergie faible au niveau des hautes fréquences, le simple fait de forcer tous ces coefficients à zéro n'introduira qu'un faible MSE et n'affectera que quelques détails de l'image. Le processus d'extraction n'a plus qu'à utiliser les vecteurs nuls de ces sous bandes afin d'estimer les vecteurs « perturbés » reçus. Ceci contredisant le fait que les basses fréquences soient les plus appropriées pour la dissimulation , un compromis à été fait en utilisant les coefficients de la sous-bande LL-HH ainsi que le montre la figure 5.

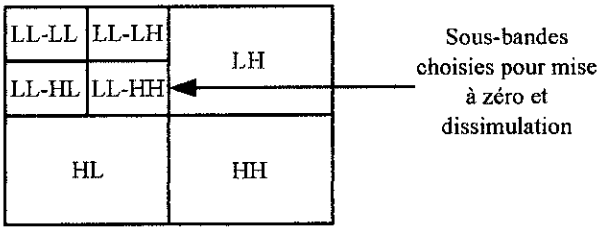

Figure-5 : Sous-bandes choisies pour mise à zéro

La figure 6 explicite le mécanisme de dissimulation et d'extraction

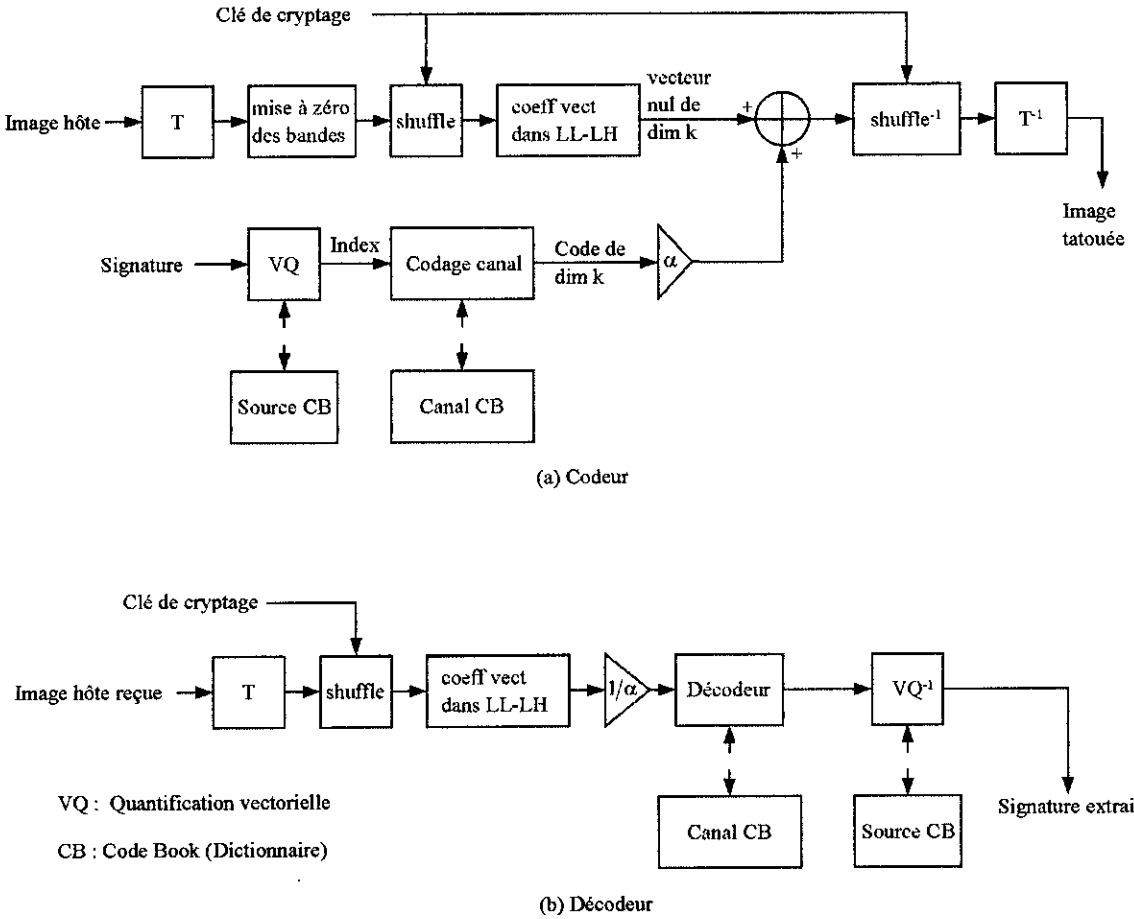

Figure-6 : Schéma de dissimulation et d'extraction en utilisant les bandes LL-HH

La vidéo hôte est en premier lieu transformée en utilisant la transformation par ondelette. Une clé de cryptage est ensuite utilisée pour éparpiller d'une manière pseudo-aléatoire les coefficients au niveau de la sous-bande choisie et ceci avant même de les grouper en vecteurs de dimension K. La donnée dissimulée est quantifiée vectoriellement d'une manière appropriée. Les indices obtenus sont dissimulés au niveau des vecteurs de dimension K du domaine transformé.

La clé de cryptage ajoute une sécurité additionnelle autre que celle introduite par le code Book choisi. Il est donc impossible pour des personnes non autorisées de pirater l'information dissimulée sans connaître le code Book et la clé de cryptage.

5.3. DATA HIDING D'UN MEDIA VIDEO [51]

L'approche exposée ici propose de dissimuler le watermark via une technique appelée « texture masking » (figure 7). L'image signature (watermark) et l'image hôte de la séquence vidéo sont toutes deux transformées en utilisant des blocs DCT de 8×8 pixels. Les coefficients de l'image signature sont quantifiés et codés en utilisant une structure en treillis multi-dimensionnel puis insérés au niveau des coefficients DCT de l'image hôte. Cette insertion est adaptative à la texture locale du contenu des images vidéos hôtes. Les images vidéos ainsi

tatouées seront ensuite compressées en utilisant le standard MPEG.

A cause des limitations physiologiques de l'œil humain, l'insertion de données additionnelles est moins visible au niveau des régions les plus texturées (hautes fréquences). Le « texture masking » est donc une technique utilisée pour rendre le tatouage adaptatif en déterminant la quantité de données du watermark à tatouer pour chaque bloc et ceci en déterminant un facteur d'échelle γ qui contrôle la quantité de données watermark insérée.

Soit une transformation de domaine de HAR utilisée sur des blocs de 8×8 pixels, l'utilisation de cette transformation permet de constituer l'ensemble B= {LH, HL, HH}, qui est un ensemble de sous-bandes de l'image hôte tatouée. Pour $b \in B$ soit $\mu_w(b)$ l'énergie moyenne en bande -b- après transformation de l'image hôte, soit $\mu_D(b)$ l'énergie moyenne en bande -b- du bloc considéré, ainsi l'énergie de la texture du bloc est donnée par :

$$\mu_T(b)=\frac{\mu_D(b)}{\mu_w(b)} \qquad (5)$$

si $\mu_T(b)$ est supérieure à un seuil $T_H(b)$, le bloc est considéré comme ayant une texture significative en bande b. Si l'énergie en texture du bloc dépasse le seuil en deux ou trois sous-bandes, alors le bloc est considéré comme étant hautement texturé, et si l'énergie en texture du bloc est inférieure au seuil $T_L(b)$ pour trois sous-bandes, alors le bloc est faiblement texturé. Les blocs DCT de l'image hôte sont classés en trois catégories : hautement, normalement et faiblement texturé. A chaque classe correspondra un facteur γ. D'une manière générale : $T_L(b)=\frac{3}{4}$; $\forall b \in B$ et $T_H(b)=\frac{4}{3}$; $\forall b \in B$. De plus les valeurs suivantes sont attribuées à γ :

$$\gamma(\text{high})=2 \; ; \; \gamma(\text{normal})=0 \; ; \; \gamma(\text{low})=-2.$$

Les coefficients de basses fréquences requièrent plus de niveaux de quantification que ceux de hautes fréquences. Sachant cela, considérerons plusieurs structures en treillis multidimensionnel. Ainsi pour mille deux cent trente deux niveaux de quantification sur lesquels sont quantifiés les blocs DCT, une structure E_8 est utilisée. Bien que celle ci puisse coder deux mille quatre cent mots seuls mille deux cent trente deux mots sont utilisés ; cette structure à huit composantes requiert huit coefficients hôtes pour les dissimuler. De même les trois cent quarante deux et quarante huit niveaux de quantifications requièrent six coefficients et quatre coefficients hôtes pour des structures E_6 et D_4 respectivement. Étant donné que six composantes à mille deux

cent trente deux niveaux de quantifications, que seize composantes à trois cent quarante deux niveaux de qualifications et que douze composantes à quarante huit niveaux de quantifications sont utilisées pour un bloc DCT de l'image signature (watermark), il faudra cent quatre vingt douze composantes hôtes pour la dissimuler. Ces composants seront pris de plusieurs blocs DCT de l'image hôte, et les divers choix offerts constituent autant de clés que de choix possibles.

L'approche décrite utilise les blocs DCT car il a été prouvé que la robustesse est accrue par rapport à une compression JPEG ou MPEG.

Les coefficients de l'image signature sont quantifiés et dissimulés en trois étapes :

1. en utilisant la quantification standard MPEG ;
2. en utilisant une quantification spécifique qui détermine la taille relative des données à dissimuler, suivie d'un codage par treillis multidimensionnel ;
3. En insérant la donnée codée au niveau des coefficients DCT.

En résumé, les diverses étapes qui constituent cette approche sont (figure 8):

1. l'image hôte et signature (watermark) sont transformées en utilisant la DCT ;
2. chaque bloc 8x8 pixels de l'image hôte est analysé pour déterminer la texture du contenu et pour calculer le facteur γ ;
3. les coefficients signature (du watermark) sont quantifiés puis codés en utilisant une structure en treillis multidimensionnel ;
4. les codes signature sont échelonnés en utilisant un facteur d'échelle $\delta = \gamma + \alpha$ et la quantification MPEG ;
5. les coefficients hôtes sont remplacés par les codes signature quantifiés et combinés avec les coefficients DCT originaux ;
6. les coefficients ainsi traités subissent une transformation inverse pour produire l'image de la séquence vidéo tatouée.

La quantification MPEG va servir à renormaliser le vecteur codé pour que sa portée dynamique soit similaire à celle d'un bloc DCT typique (pour que les écarts entre les valeurs des coefficients les constituant ne soient pas trop importants).

La détection utilisée est une détection non-cohérente donc sans l'aide de l'originale de la vidéo.

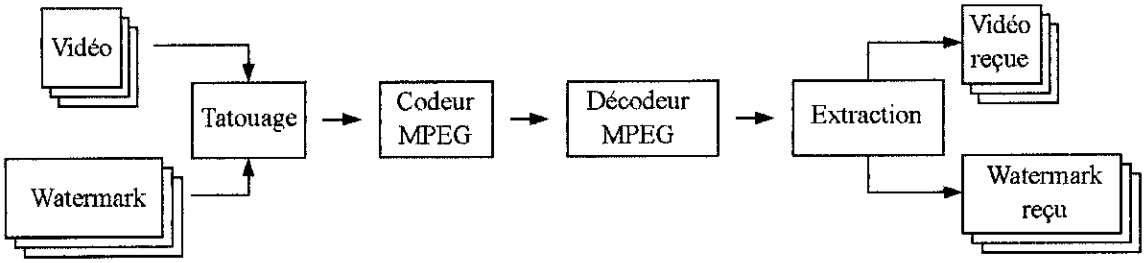

Figure-7 : Schéma du principe de watermarking

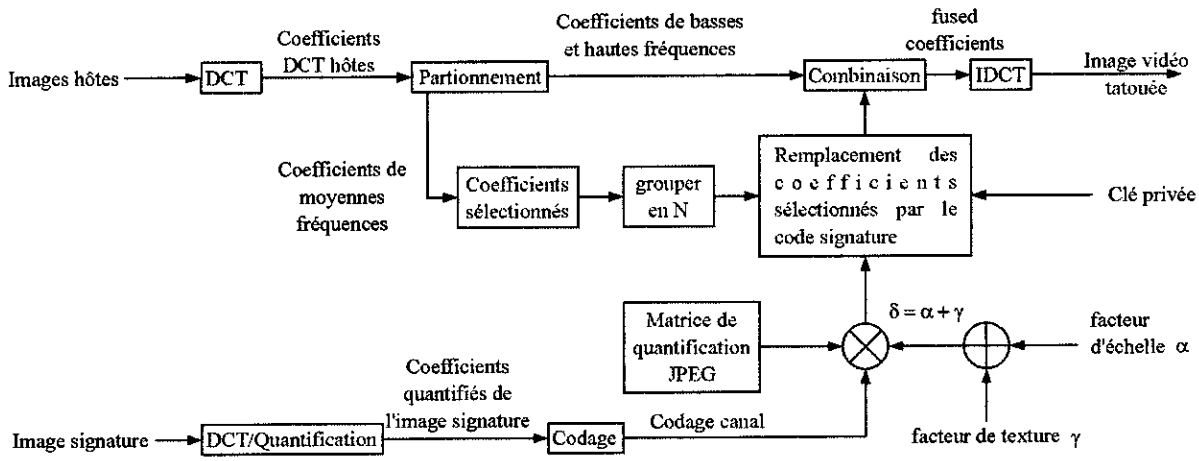

Figure-8 : Schéma du codeur

5.4. UNE MÉTHODE ROBUSTE DE WATERMARKING VIDÉO [52]

Cette méthode propose de dissimuler un label au niveau des pixels de l'image et de les détecter à partir du domaine DCT de chaque intra-image.

Afin de dissimuler un label, l'image est divisée en autant de blocs label que de bits à dissimuler. Chaque bloc label est en fait constitué de plusieurs blocs de 8×8 pixels. Les séquences du signe de 8 coefficients DCT de basses fréquences d'un bloc élémentaire sont utilisées pour dissimuler le watermark. La figure 9(a) explicite l'échantillon watermark d'un bloc élémentaire où les « 1 » et les « 0 » représentent les pixels du watermark dont la valeur est supérieure ou inférieure respectivement à la valeur moyenne des pixels d'un bloc élémentaire du watermark. En ce qui concerne l'échantillon pixel de la figure 9 (a), la séquence du signe de 8 coefficients de basses fréquences est « ++-+++---» ainsi que le montre la figure 9 (b).

1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0
1	1	1	0	0	0	0	1
1	1	1	0	0	0	1	1
1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	0	1	1	1	1	1

(a)Watermark pixel pattern

395.8	9.9	10	0.20	0.12	0	-0.2	0.3
-10.2	9.81	9.8	0	0.1	0	0.6	-0.1
-10.1	-9.5	-9.8	0	0.6	0.2	-0.2	-0.1
-0.1	-0.4	0	0.4	0	0.3	-0.4	-0.3
-0.1	-0.4	0	-0.4	0.1	0.2	0.1	-0.3
0	-0.2	-0.3	0	0.3	-0.2	0	0.5
0	0	0	0	0.6	0	-0.1	0
-0.5	0.5	-0.1	-0.3	-0.3	0.1	-0.3	-0.1

(b) Coefficients DCT

Figure-9 : Exemple d'un watermark pixel pattern

De cette manière, on peut constituer 256 séquences de signes et de leurs échantillons pixels correspondants. La table 1 est une table récapitulative qui montre la relation entre l'échantillon pixel et les séquences de signes des coefficients DCT. Chaque ligne de la table 1 représente un nombre hexadécimal à 2 octets. De plus, les signes « + » et «-» sont remplacés respectivement par « 1 » et « 0 ».

PIXEL PATTERN	SEQUENCE DU SIGNE DES COEFFICIENTS DCT
1C,1E,1E,FF,FF,FF,7F,1F	00
07,0E,3C,3C,7C,FE,C7,83	01
1C,1C,1F,1F,1F,FF,FC,F8	02
•	•
•	•
7F,3F,1F,1E,1C,00,C0,E0	27
•	•
•	•
E3,E3,C7,07,07,07,07,07	7F
1C,1C,38,F8,F8,F8,F8,F8	80
•	•
•	•
80,C0,E0,E1,E3,FF,3F,1F	D8
•	•
•	•
E3,E1,E1,00,00,00,80,E0	FF

Table-1 : Table récapitulative du pixel pattern et du signe des coefficients DCT

Pour dissimuler un bit, on divise les 256 séquences de signes des coefficients DCT en deux groupes. La table 2 en est un exemple. Un groupe est utilisé pour dissimuler un « 1 » et l'autre pour dissimuler un « 0 ». Ainsi, le nombre de tables possibles est de $2^{128}*128!$. Un des 128 groupes ID est assigné à chaque bloc élémentaire en utilisant une séquence aléatoire (pseudo-noise sequence). La valeur du pixel d'une image donnée correspondant à la position du 1 et 0 au niveau de l'échantillon watermark est incrémentée et décrémentée respectivement par la valeur

(quantité) du watermark. La valeur du watermark pour chaque pixel est déterminée en considérant le SVH (système visuel humain) basé sur le modèle de Petterson. Ce modèle détermine un seuil pour chaque pixel en considérant les valeurs des pixels voisins.

Groupe ID	Séquence du signe des coefficients DCT	
	Groupe A	Groupe B
0	00	FF
1	01	FE
2	02	FD
• •	• •	• •
39	27	D8
• •	• •	• •
126	7^E	81
127	7F	80

Table-2 : Exemple de table utilisée pour la dissimulation

La figure 10 donne le schéma bloc des procédures de dissimulation des labels proposées. La figure 11 explicite l'algorithme de dissimulation du label, où L_m est le $m^{éme}$ bit à dissimuler au niveau du $m^{éme}$ label bloc. P_{ij} est la valeur du pixel de position (i,j) dans un bloc élémentaire, p'_{ij} est la valeur du pixel tatoué, et t_{ij} est la valeur du watermark.

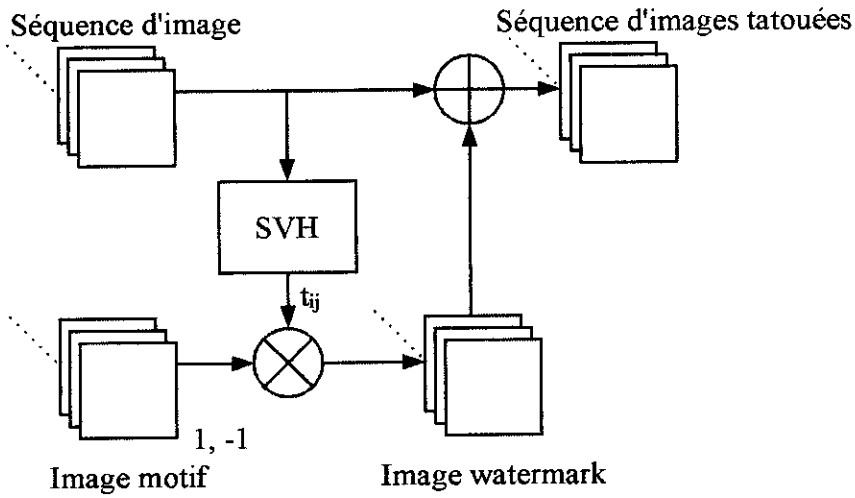

Figure-10 : Diagramme de l'algorithme de dissimulation

```
Procedure Embedding_Label_of_Moving_Picture()
{
  for each frame {
       Getting_Y_Plane(frame)
       for (m=1 to N) {  /*pour chaque bloc du label */
              Group_ID=Using_PNseq(PN1)
              if (L_m=1)
                     PixelPattern=Agroup(Group_ID)
              else
                     PixelPattern=Bgroup(Group_ID)
              for (i=1 to 8)  /* pour chaque bloc de base */
                     for (j=1 to 8) {
                            If (PixelPattern(i,j)=1)
                                   p̂_ij =p_ij + t_ij
                            else
                                   p̂_ij =p_ij - t_ij
                     }
       }  /* bloc du label*/
  }  /*image*/
}
```

Figure-11 : Algorithme de dissimulation

Pour ce qui est de la détection du label, les coefficients DCT sont extraits à partir du bitstream et seulement les intra-images sont prises en compte. Après décodage à longueur variable et quantification des coefficients DCT des intra-images, un échantillon de signes est assigné à chaque bloc élémentaire de coefficients de DCT du watermark en utilisant une séquence aléatoire. Afin d'améliorer la détection du watermark, chaque coefficient est traité de manière à rendre la valeur des coefficients inférieure au seuil T :

$$W_{ij}=C_{ij} \quad \text{si } (|Cij| < T)$$

$$W_{ij}=\text{sign}(C_{ij})T \quad \text{si } (|Cij| > T) \qquad (6)$$

Où les C_{ij} sont les coefficients DCT du watermark de fréquence (i,j) et W_{ij} sont les coefficients traités et utilisés pour la détection du label.

Soit l'ensemble des coefficients DCT tatoués correspondant à « 1 » défini par X, et l'ensemble des coefficients DCT tatoués correspondant à « 0 » défini par Y : Si un bit de valeur « 1 » est dissimulé au niveau d'un bloc label, la valeur moyenne des coefficients traités de l'ensemble X est supérieure à celle de l'ensemble Y. Et inversement, si un bit de valeur « 0 » est dissimulé au niveau d'un bloc label, la valeur moyenne des coefficients traités de l'ensemble Y est supérieure à celle de l'ensemble X. Donc, seul un bit dissimulé dans un bloc label sera détecté par l'équation 7 :

$$I_M=0 \text{ si } \overline{X_M} \geq \overline{Y_M}$$
$$I_M=1 \text{ si } \overline{X_M} \leq \overline{Y_M} \qquad (7)$$

Où I_m est la valeur du bit détecté au niveau du $m^{éme}$ bloc e l'image, $\overline{X_M}$ et $\overline{Y_M}$ sont les valeurs moyennes des coefficients traités au niveau des ensembles X et Y.

L'équation 8 est utilisée est pour réduire le BER (Bit Rate Error) qui est basé sur le théorème central limite :

$$BER = \frac{1}{2} \times erfc\left(\frac{-K}{\sqrt{2 \times \frac{(\sigma_K^2 + \sigma^2)}{N}}} \right) \qquad (8)$$

Où K et σ_K^2 sont la moyenne et la variance de la taille du watermark à ajouter au niveau du domaine DCT, N est le nombre de coefficients DCT pour la dissimulation et σ^2 la variance des coefficients DCT traités.

L'équation 8 montre bien que plus K et N sont élevés et plus σ^2 et σ_K^2 sont faibles plus le BER est bas. D'où le traitement effectué au niveau de l'équation 1.

5.5. DATA HIDING D'UN MEDIA VIDEO BASE SUR UNE APPROCHE MULTI-NIVEAUX [53]

Les systèmes de watermarking peuvent être assimilés à un système simple de communications. Le canal représentera la vidéo ou l'image hôte, les attaques et les différents traitements constitueront le bruit du canal. Le but de l'approche introduite est de convoyer des données à de hauts débits lorsque le bruit est faible et en même temps de pouvoir atteindre un degré de robustesse assez élevé dans des conditions où le bruit est plus intense.

Cette approche va utiliser une transformation de domaine et une détection dite non- cohérente donc sans avoir recours à l'originale lors de la phase de détection, ceci afin d'atteindre un bon compromis entre imperceptibilité et robustesse de la marque, et ce malgré l'utilisation d'une détection non-cohérente qui introduit un bruit supplémentaire (possibilité d'interférence) dû au media hôte.

Dans le cas où l'application considérée est la protection des droits d'auteurs, l'utilisation d'un label peut se révéler amplement suffisante. Ceci se fera en insérant des bits au niveau des blocs DCT.

Considérons N composantes du media hôte $\{X_i\}$; i=1,...,N prisent dans le domaine spectral. L'approche considérée peut être formulée comme étant un test d'hypothèse, où le vecteur de watermarking W_i module le bit b et où la valeur α_i est utilisée pour ajuster le watermark en

considérant les caractéristiques du SVH (Système Visuel Humain) :

$$H_0 : Y_i = X_i - \alpha_i \times W_i + N_i \qquad \text{(b=-1) i=1,...,N}$$

$$H_1 : Y_i = X_i + \alpha_i \times W_i + N_i \quad \text{(b=1)} \qquad (9)$$

Le bruit dû au traitement est généralement modélisé comme étant identiquement distribué (id) et gaussien de moyenne nulle et de variance σ_i^2, α_i et W_i sont considérés comme étant des signaux connus. De plus, les hypothèses H_0 et H_1 sont considérées équiprobables.

Étant donné que un modèle de détection non-cohérent est considéré, les deux coefficients X_i et N_i sont rassemblés et considérés identiquement distribués, mutuellement indépendants, de moyenne nulle, de variance σ_{iX}^2 et σ_{iN}^2 respectivement et gaussiens (la combinaison linéaire de deux distributions gaussiennes est aussi gaussienne). Soit M_i une variable gaussienne de moyenne nulle et de variance $\sigma_{iM}^2 = \sigma_{iX}^2 + \sigma_{iN}^2$, (le modèle de bruit choisi considère les composantes M_i indépendantes de moyenne nulle), de distribution gaussienne et de variance σ_{iM}^2 .On aura donc :

$$H_0 : Y_i = -S_i + N_i \quad \text{(b=-1) i=1,...,N} \quad \Rightarrow \quad H_0 : \underline{Y} \approx N(-S, Diag(\sigma_{M1}^2, ..., \sigma_{Mn}^2))$$

$$H_0 : Y_i = S_i + N_i \quad \text{(b=1)} \qquad\qquad H_1 : \underline{Y} \approx N(+S, Diag(\sigma_{M1}^2, ..., \sigma_{Mn}^2)) \qquad (10)$$

Le détecteur optimal choisi en considérant la théorie de détection et d'estimation [43] est le suivant :

$$T = \frac{1}{N} \sum_{i=1}^{N} \frac{Y_i \times S_i}{\sigma_{iM}^2} \geq \text{ou} < 0 \qquad (11)$$

Nous allons à présent comparer le test de corrélation T à zéro. On remarquera que les coefficients Y_i sont normalisés en fonction de leur standard de déviation ce qui a pour but de rendre leur distribution identiquement distribuée. Ce détecteur peut être vu comme étant un corrélateur pondéré qui donne les poids les plus élevés aux coefficients les moins bruités. Ce corrélateur est assez similaire au détecteur Euclidien de distance minimum[43] :

$$T = \frac{1}{N} \sum_{i=1}^{N} Y_i \times S_i \geq \text{ou} < 0 \qquad (12)$$

Ce détecteur conviendrait dans le cas où nous aurions pris comme hypothèse des composantes M_i identiquement distribuées et gaussiennes, de moyenne nulle et de variance $\sigma_M^2 = \sigma_X^2 + \sigma_N^2$; dans ce cas les hypothèses de test seraient les suivantes :

$$H_0 : \underline{Y} \approx N(-S, \sigma_M^2 I)$$
$$H_1 : \underline{Y} \approx N(+S, \sigma_M^2 I) \qquad (13)$$

Dans le cas considéré la variance de la statistique unité normalisée T_N et sa moyenne vont donner une mesure des performances de détections :

$$T_N = \frac{T}{\sqrt{Var(T)}} = \frac{\sum_{i=1}^{N} \frac{Y_i \times S_i}{\sigma_{iM}^2}}{\sqrt{\sum_{i=1}^{N} \frac{S_i^2}{\sigma_{iM}^2}}} \quad \text{et} \quad E(T_N) = b \cdot \times \sqrt{\sum_{i=1}^{N} \frac{S_i^2}{\sigma_{iM}^2}} \qquad (14)$$

Ces équations ont deux implications : D'une part, laisser une bande non tatouée réduira la valeur de $E(T_N)$ et donc accroître la probabilité d'erreur, d'autre part, la contribution des bandes les plus bruitées va être réduite car multipliée par l'inverse de la variance du bruit. Ces bandes bruitées comprennent les basses fréquences qui ont un bruit d'interférence avec le media hôte assez élevé. Le fait de ne pas utiliser ces basses fréquences pour dissimuler la marque ne causera pas de dégradations significatives au processus de détection. Si un modèle de bruit plus général où les composantes $\{X_i\}$et/ou$\{N_i\}$sont dépendants est considéré, le blanchissement et la normalisation doivent intervenir avant d'appliquer le détecteur Euclidien.

Dans le but de réduire les interférences avec le media hôte lors de la détection, les coefficients de moyennes fréquences sont utilisés plutôt que les coefficients de basses fréquences car ayant une énergie plus élevée. Ce raisonnement repose sur l'hypothèse que $\sigma_M^2 = \sigma_X^2 + \sigma_N^2$. Ainsi, les coefficients de basses fréquences contribuent à élever la valeur de σ_X^2 et ceux de hautes fréquences, sujets aux distorsions, contribuent à élever σ_N^2.

De plus, l'énergie totale du watermark influe sur la statistique de détection. Donc, laisser un grand nombre de coefficients basses fréquences non tatoués va réduire l'énergie du watermark et accroître la probabilité d'erreur.

A partir de la q-statistique de détection proposée par Zeng et al et de la q-statistique pondérée (pondération en fonction du bruit) donnée par les équations 15 et 16 respectivement :

$$q = \frac{M_Z}{\sqrt{V_Z / N}} \qquad \text{où } Z_i = Y_i \times W_i,\ M_Z = \frac{1}{N}\sum_{i=1}^{N} Z_i,\ V_Z = \frac{1}{N-1}\sum_{i=1}^{N} (Z_i - M_Z)^2 \qquad (15)$$

$$q' = \frac{M_Z}{\sqrt{V_Z / N}} \quad \text{où } Z_i = Y_i \times W_i / C_i,\ M_Z = \frac{1}{N}\sum_{i=1}^{N} Z_i,\ V_Z = \frac{1}{N-1}\sum_{i=1}^{N} (Z_i - M_Z)^2 \qquad (16)$$

Avec les coefficients C_i reflètant l'impact de la variance du bruit dans l'équation précédente.

La figure 13 montre que la valeur de la statistique q est maximale quand on utilise les coefficients DCT à partir des bandes 6 à 10 (troisième ligne de coefficients). Cette valeur décroît lorsque l'on considère moins de bandes pour dissimuler la marque ou signature. La valeur de q' est généralement supérieure à celle de q et décroît aussi lorsque moins de bandes sont prises en compte pour dissimuler la marque ou signature. Mais cette décroissance n'est pas significative si les bandes de 1 à 5 ne sont pas utilisées (figures 12 et 13). Les coefficients de basses fréquences seront utilisés pour dissimuler une autre donnée via une technique appelée « odd-even », autrement dit « pair-impair » où on arrondira la valeur du coefficient DCT à la valeur supérieure si on veut dissimuler un « 1 » et à la valeur inférieure si on veut dissimuler un « 0 ».

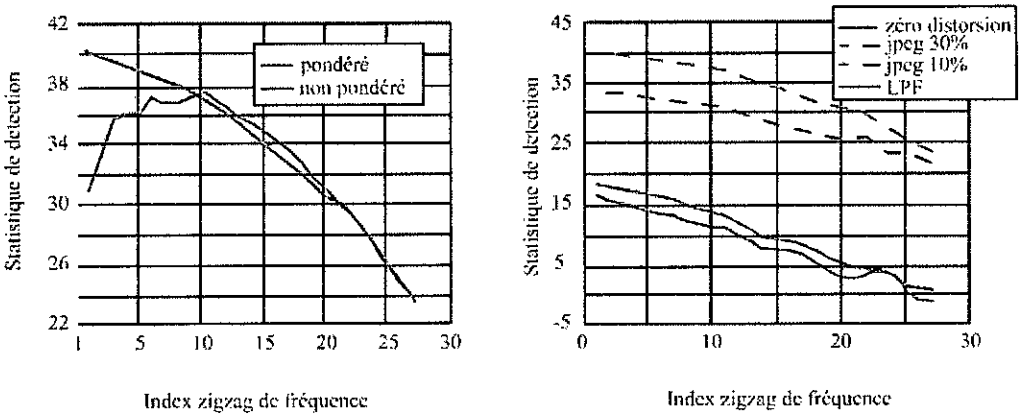

Figure-12 : Statistique de détection moyenne d'un corrélateur et d'un corrélateur pondéré à partir de 114 images naturelles. a/statistique de détection du corrélateur non-pondéré et pondéré (q et q') dans des conditions où il n'y a pas de distorsions. b/statistique de détection du corrélateur pondéré q' sous plusieurs conditions de distorsions.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Figure-13 : Ordre de ZigZag des coefficients

L'approche décrite ici peut être aussi bien appliquée à des images qu'à des vidéos. Dans ce contexte, les redondances de la séquence vidéo sont utilisées pour accroître la robustesse de la marque aux attaques d'inversion d'images et de « dropping ». Ainsi, la vidéo est divisée en segments constitués d'images successives assez similaires. La même donnée est dissimulée au niveau d'un même segment en utilisant l'approche multi-niveaux. La figure 14 donne un schéma récapitulatif de la méthode proposée.

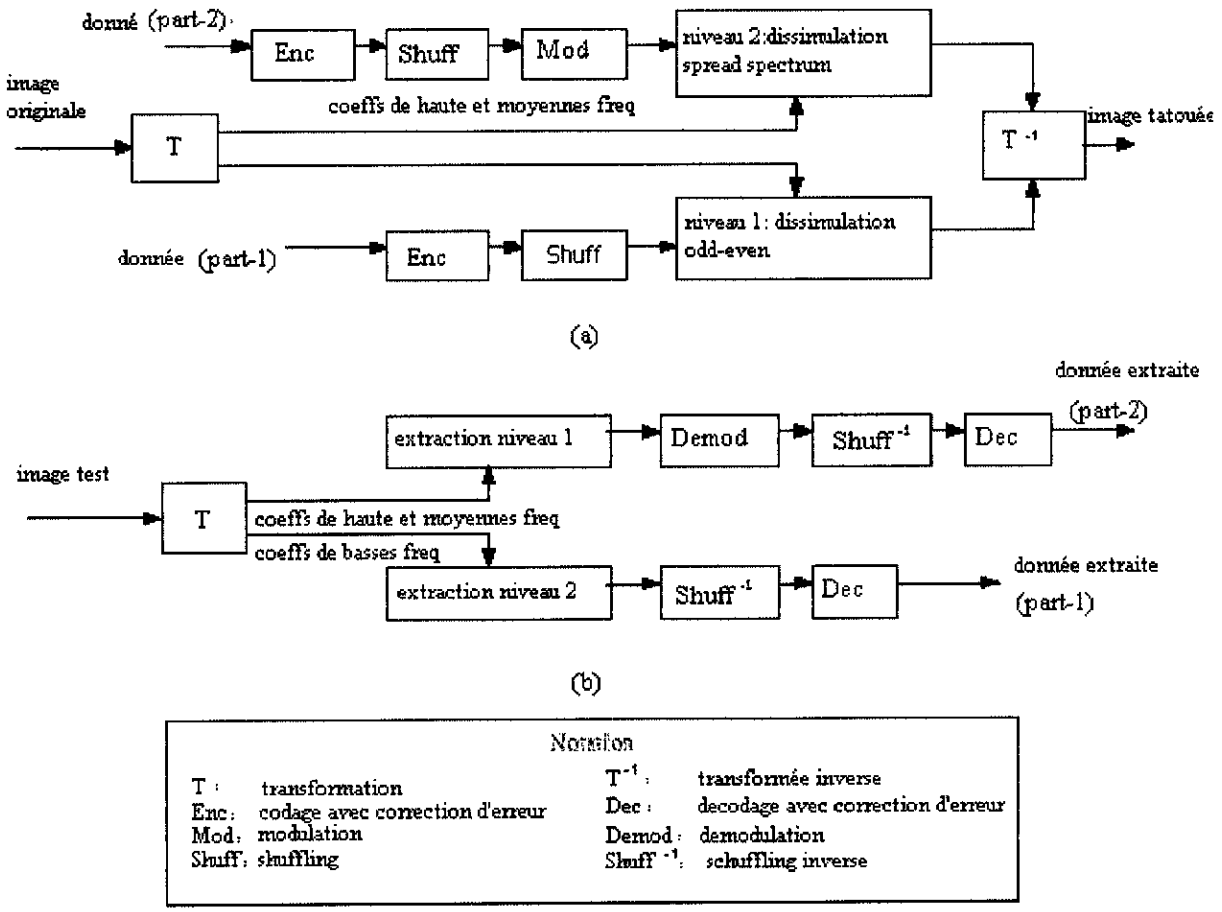

Figure-14 : Schéma bloc du principe de dissimulation et d'extraction multi-niveaux

5.6. WATERMARKING DES VIDEOS PRE-COMPRESSEES EN UTILISANT L'ETALEMENT DE SPECTRE [54]

5.6.1. Watermarking au niveau du domaine pixels

La technique de spread spectrum (étalement de spectre) consistant en la transmission d'un signal de bande relativement étroite via une bande de fréquences plus large, l'approche proposée ici en est donc dérivée.

En général, une séquence vidéo est un signal tridimensionnel. Cependant, dans l'approche

proposée ce signal est pris comme étant monodimensionnel. Ceci est acquis par une lecture des coefficients (pixels) de la manière explicitée en figure 15 :

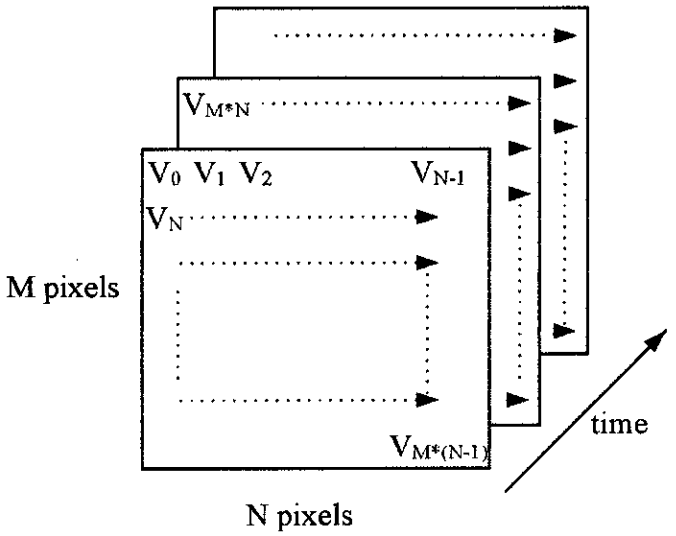

Figure-15 : Lecture en ligne des pixels d'une vidéo

Soit a_j, a_j {-1,1}, j∈N une séquence de bits constituant le watermark. Ce signal va être dissimulé en utilisant un facteur d'étalement cr appelé chip-rate. Ceci permettra d'ajouter une redondance en dissimulant un bit au niveau de cr pixels de la séquence vidéo, ainsi :

$$b_i=a_j \qquad j\times cr \le i \le (j+1)\times cr,\ i\in N \qquad (17)$$

La séquence b_i est amplifiée grâce à un facteur d'ajustement local α pour exploiter les phénomènes de masquage spatial et temporel qui caractérisent le SVH (Système Visuel Humain). L'amplitude de ce facteur peut varier en fonction des propriétés locales du signal vidéo,. La séquence b_i est ensuite modulée en utilisant une séquence pseudo-aléatoire binaire p_i :

$$p_i,\ p_i\{-1,1\},\ j\in N$$

Le résultat de ces traitements donnera le signal watermark modulé :

$$w_i=\alpha_i\times b_i\times p_i,\ i\in N \qquad (18)$$

Ce signal est additionné au signal vidéo scanné en ligne v_i ce qui nous donnera le signal vidéo tatoué :

$$\bar{v}_i = v_i + \alpha_i \times b_i \times p_i, \quad i \in N \qquad (19)$$

Ce denier doit être réarrangé en une matrice pour qu'il puisse être affiché.

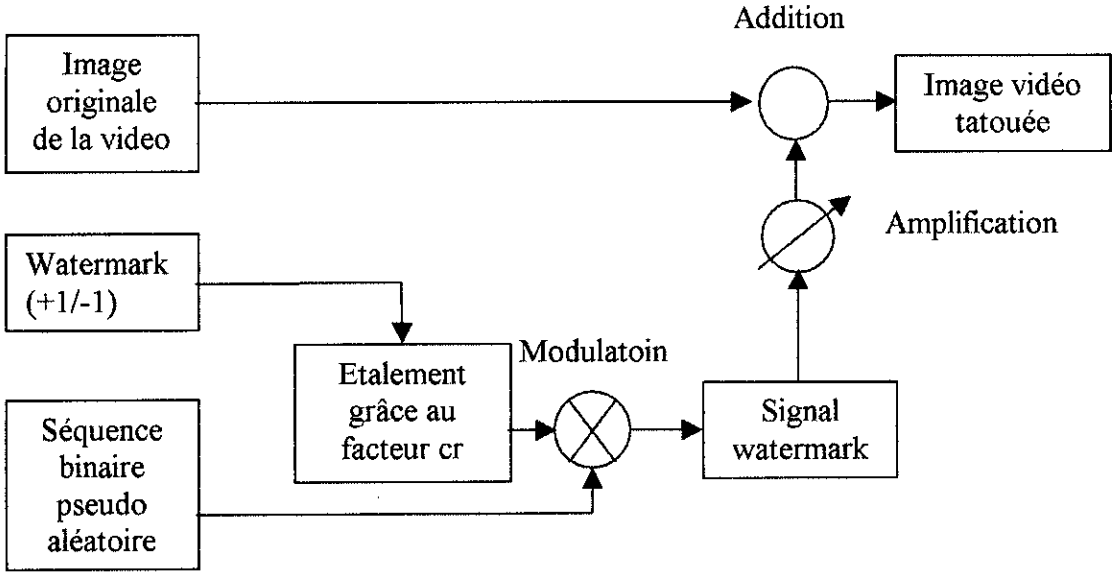

Figure-16 : Processus de dissimulation

L'utilisation de la séquence pseudo-aléatoire permet d'atteindre une meilleure robustesse de l'ensemble car le watermark sera difficile à détecter, à localiser et à manipuler sans la connaissance exacte de cette séquence.

Le recouvrement de la marque dans cette approche se fait grâce à un corrélateur au niveau du récepteur. Il est à noter que le décodage de la marque (figure 16) se fait sans avoir recours à la séquence vidéo originale (détection non- cohérente). Aussi, avant d'utiliser le corrélateur, la séquence vidéo tatouée est filtrée par un filtre passe-haut, ainsi on obtiendra la séquence vidéo tatouée filtrée $\bar{\bar{v}}_i$. Cette opération va permettre d'ôter la majeure partie des composantes de la séquence vidéo originale (par exemple, en utilisant un filtre passe-haut non adaptatif 3×3). Le filtrage n'est pas une opération nécessaire mais cette dernière optimise les performances générales du système en réduisant les interférences entre le signal vidéo original et le signal watermark.

L'étape suivante consiste en la démodulation de la vidéo tatouée : en multipliant cette vidéo par la séquence pseudo-aléatoire p_i, on obtiendra ainsi la somme de corrélation s_j :

$$s_j = \Sigma_1 + \Sigma_2 = \sum_{i=j\times cr}^{(j+1)\times cr-1} p_i \times v_i = \sum_{i=j\times cr}^{(j+1)\times cr-1} p_i \times \bar{v}_i + \sum_{i=j\times cr}^{(j+1)\times cr-1} \overline{p_i \times p_i \times \alpha_i \times b_i} \qquad (20)$$

où Σ_1 et Σ_2 désignent les contributions à la somme de corrélations du signal vidéo filtré et le signal watermark filtré respectivement.

Si on considère que Σ_1 est nul ce qui voudrait dire que le signal vidéo a été filtré et a donné $\bar{v}$ et si on considère aussi que $\overline{p_i \times \alpha_i \times b_i} = p_i \times \alpha_i \times b_i$ ce qui voudrait dire que le signal vidéo a une influence négligeable sur le bruit pseudo-aléatoire blanc du signal watermark, alors dans ces conditions s_j devient :

$$s_j = \Sigma_1 + \Sigma_2 = \sum_{i=j\times cr}^{(j+1)\times cr-1} p_i^2 \times \alpha_i \times b_i = a_j \times \sigma_p^2 \times moy(\alpha_i) \qquad (21)$$

où σ_p^2 est la variance de la séquence pseudo aléatoire. Ainsi :

$$sign(s_j) = sign(a_j \times \sigma_p^2 \times cr \times moy(\alpha_i)) = sign(a_j) = a_j \qquad (22)$$

Ce qui nous permet de retrouver le watermark.

5.6.2. Watermarking des vidéos codées au niveau du bitstream MPEG

L'idée sur laquelle repose cette approche est la dissimulation du watermark au niveau du bitstream MPEG. Ainsi, la syntaxe de celui-ci permet d'insérer des données d'utilisateurs (au niveau de n'importe laquelle des séquences d'images, de groupe d'image ou d'en-tête d'image).Mais on ne pourra pas utiliser cette caractéristique car :

1. le watermark doit persister au niveau de la séquence vidéo après décodage.
2. le bit rate de la séquence vidéo ne doit pas augmenter après l'opération de dissimulation.

Donc, l'idée proposée est de dissimuler le watermark au niveau de la donnée représentant le signal utile. Sachant cela, on extraira de chaque bloc 8×8 de la vidéo, le bloc correspondant du watermark. On va ainsi transformer le bloc de 8×8 pixels du watermark en utilisant la DCT , puis on l'additionnera au bloc DCT de la vidéo originale. La figure 17 montre le schéma bloc explicatif de l'approche décrite. On va donc suivre les étapes suivantes :

1. génération du signal watermark à dissimuler au niveau de chaque image vidéo de la séquence de la même manière que celle utilisée dans la technique précédente ;
2. réarrangement du signal watermark généré en un signal watermark bidimensionnel de même taille que l'image vidéo hôte ;

3. transformation des blocs 8×8 du signal watermark correspondant à chaque bloc 8×8 de l'image hôte codée au niveau du bitstream en utilisant la DCT ;
4. les blocs 8×8 pixels du signal vidéo et les blocs 8×8 du signal watermark sont additionnés au niveau du domaine DCT ce qui donne les blocs 8×8 tatoués qui seront codés et formeront le nouveau bitstream ;
5. toutes les autres parties du bitstream compressé sont recopiées au niveau du nouveau bitstream.

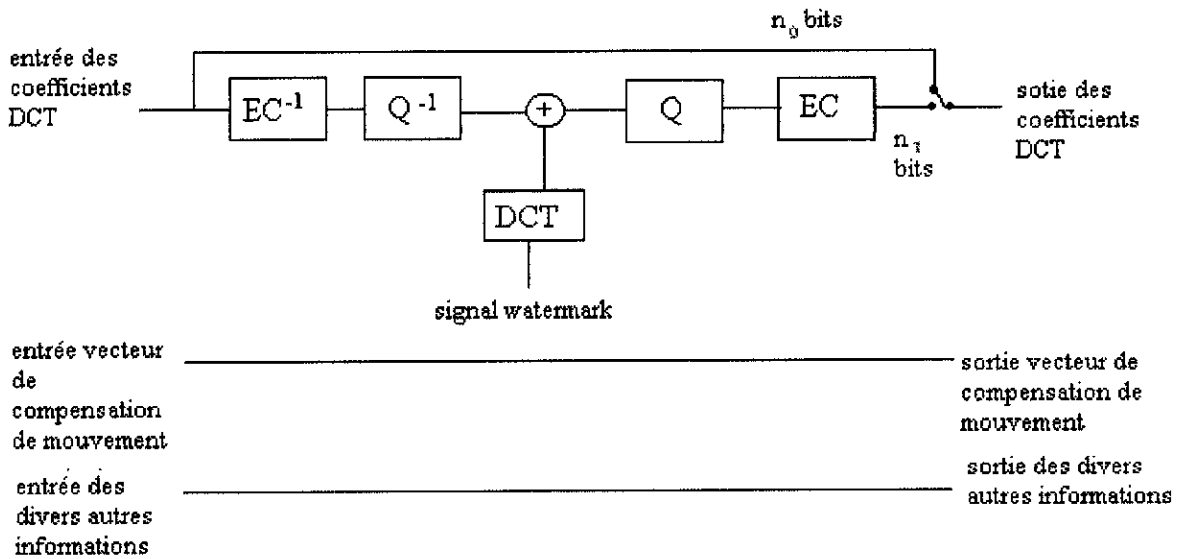

Figure-17 : Schéma général de dissimulation des vidéos compressées

A gauche, le bitsream entrant est séparé en un en-tête, une série d'informations secondaires, une série de données représentant les vecteurs de compensations de mouvement et en une donnée représentant les blocs du signal codé. Dans ce cas, seule cette partie sera altérée. Les blocs DCT sont représentés par une séquence de codes d'Huffman au niveau du bitstream, chacun représentant une paire (run et level coding) et ainsi un coefficient DCT non nul du bloc courant. Chaque code d'Huffman est décodé puis quantifié inversement (conversion de l'index de quantification à la valeur quantifiée représentative). On obtiendra donc un coefficient DCT du bloc courant, cette valeur sera additionnée à celle du coefficient DCT correspondant du bloc watermark dans le domaine de transformation. A présent il ne restera plus qu'à quantifier et coder le coefficient tatoué en utilisant les tables standardisées d'Huffman pour le standard MPEG.

Une fois le codage de ces coefficients tatoués terminé, on comparera le nombre n_1 de bits qui constitue le nouveau mot code d'Huffman avec le nombre de bits n_0 qui constitue l'ancien mot code d'Huffman. Ceci permet de ne pas augmenter le bit rate car on ne transmet le mot code du

coefficient tatoué que si $n_1 \leq n_0$. Etant donné que le watermark est dissimulé au niveau de plusieurs coefficients DCT, le fait d'en laisser quelques uns non tatoués n'affectera pas beaucoup les performances du système. De plus on pourra si nécessaire augmenter le chip rate cr (dissimuler un bit d'information du watermark au niveau d'un nombre plus élevé cr de coefficients DCT de la séquence vidéo). Ceci aura pour effet d'augmenter la robustesse mais aussi de réduire la quantité de données que l'on pourra dissimuler. En résume on notera que :

- Seulement les coefficients DCT différents de zéro seront utilisés pour la dissimulation
- Parmi les coefficients utilisés seuls ceux qui n'augmenteront pas le bit rate une fois tatoués seront retenus.

Le fait que le format MPEG utilise la compensation de mouvement comme base pour son fonctionnement va représenter un désavantage, en ce sens que le fait d'altérer une partie de la séquence vidéo va se répercuter et donc va introduire une distorsion suite à l'accumulation des dégradations introduites. Pour y remédier, il faut ajouter un signal de réduction de dérive (drift) pour compenser la dégradation introduite par le watermark. Ainsi, on doit ajouter exactement la différence de prédiction faite au codeur et au décodeur. La figure 18 explicite une extension du principe de watermarking proposé en utilisant la compensation de dérive, qui est la différence entre la compensation de mouvement (motion compensation ou MC) du bitstream non tatoué (bloc de gauche) et la compensation de mouvement du bitstream tatoué (bloc de droite).

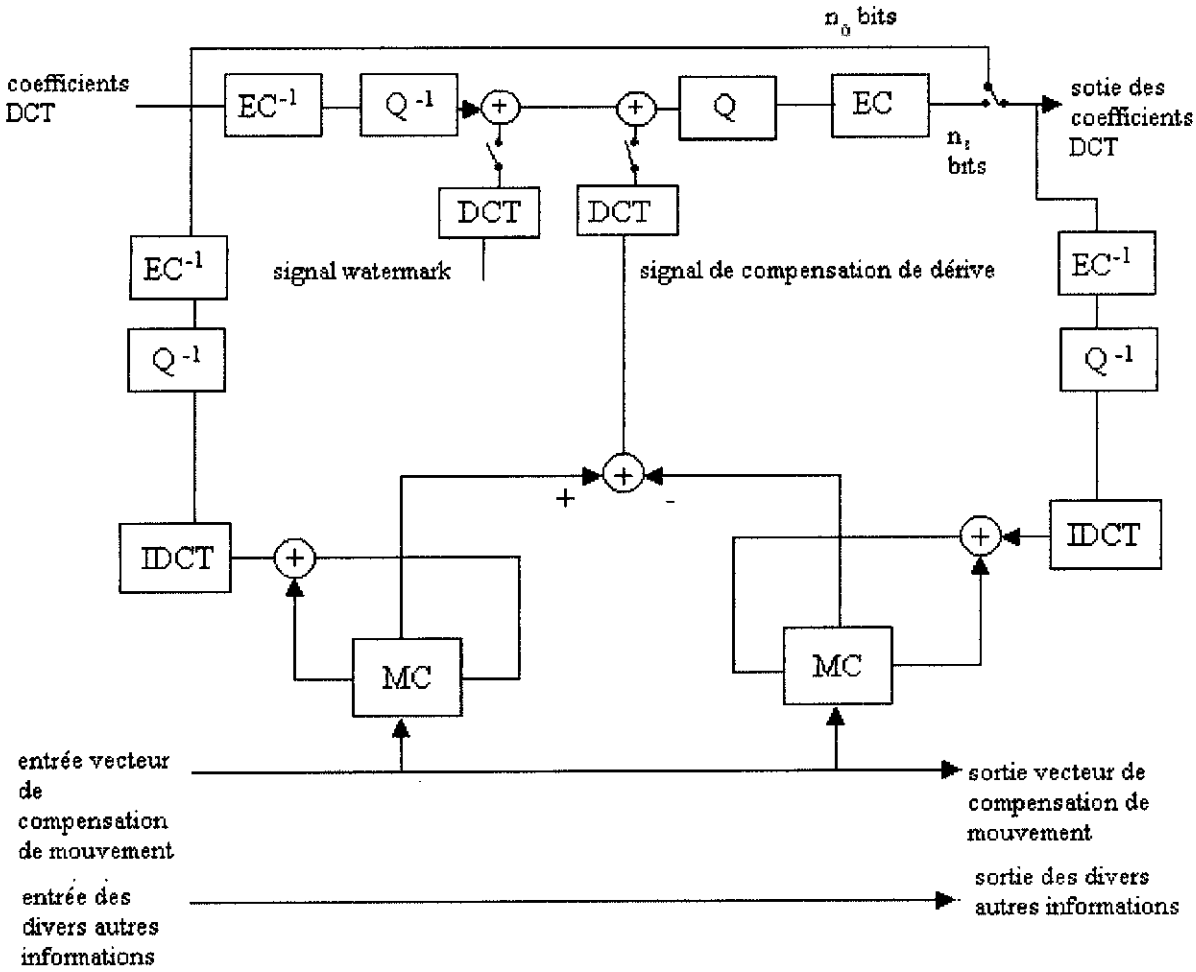

Figure-18 : Schéma général de dissimulation des vidéos compressées avec compensation de dérive

METHODE	DOMAINE DE DISSIMULATION	TYPE DE WATERMARKING	TYPE DE TRANSFORMEE	FACTEUR D'AMPLIFICATION
APPROCHE BASEE SUR LE CODAGE SOURCE ET CANAL, APPLIQUEE AU WATERMARKING VIDEO	spectral	additif	DCT (blocs 8×8 pixels)	Oui
DATA HIDING D'UN MEDIA VIDEO	spectral	additif	DCT (blocs 8×8 pixels)	Oui
UNE METHODE ROBUSTE DE WATERMARKING VIDEO	spectral	additif	DCT (blocs 8×8 pixels)	Non
DATA HIDING D'UN MEDIA VIDEO BASEE SUR UNE APPROCHE MULTI-NIVEAUX	spectral	additif	DCT (blocs 8×8 pixels)	Oui
Watermarking au niveau du domaine pixels	temporel	additif	DCT (blocs 8×8 pixels)	Oui
Watermarking des vidéos codées au niveau du bitstream MPEG	Bit stream MPEG	additif	DCT (blocs 8×8 pixels)	Oui

Table-3 : Table de comparaison des méthodes exposées

METHODE	BANDE DE DISSIMULATION	CLE DE CRYPTAGE	TYPE DE DONNEES DISSMULEES	ETALEMENT	QUANTIFICATION VECTORIELLE DU WATERMARK
APPROCHE BASEE SUR LE CODAGE SOURCE ET CANAL, APPLIQUEE AU WATERMARKING VIDEO	Moyennes fréquences	Oui	Tous type de Données numériques	Dans un GOP	Oui
DATA HIDING D'UN MEDIA VIDEO	Sélection aléatoire de fréquences	Oui	Images	Dans un GOP	Oui
UNE METHODE ROBUSTE DE WATERMARKING VIDEO	Sur toutes les fréquences	Non	Fingerprint	Intra images	Non
DATA HIDING D'UN MEDIA VIDEO BASEE SUR UNE APPROCHE MULTI-NIVEAUX	Basses et moyennes fréquences	Oui	Fingerprint	Dans un GOP	Non
Watermarking au niveau du domaine pixels	Basses et moyennes fréquences	Oui	Fingerprint	Etalement sur plusieur pixels	Non
Watermarking des vidéos codées au niveau du bitstream MPEG	Basses fréquences principalement	Oui	Fingerprint	Etalement sur plusieur coefficients	Non

Table-4 : Table de comparaison des méthodes exposées

METHODE	EXTRACTION	AVANTAGES
APPROCHE BASEE SUR LE CODAGE SOURCE ET CANAL, APPLIQUEE AU WATERMARKING VIDEO	Non-cohérente	Grande capacité de dissimulation
DATA HIDING D'UN MEDIA VIDEO	Non-cohérente	Grande capacité de dissimulation
UNE METHODE ROBUSTE DE WATERMARKING VIDEO	Non-cohérente	N'affecte pas la qualité de la vidéo originale
DATA HIDING D'UN MEDIA VIDEO BASEE SUR UNE APPROCHE MULTI-NIVEAUX	Non-cohérente	Grande robustesse
Watermarking au niveau du domaine pixels	Non-cohérente	Grande robustesse
Watermarking des vidéos codées au niveau du bitstream MPEG	Non-cohérente	Moindre compexité

Table-5 : Table de comparaison des méthodes exposées

CHAPITRE 6

CHOIX D'UNE TRANSFORMEE

- **Introduction**
- **Définitions**
- **Capacité du canal de tatouage dans le domaine spatial**
- **Capacité du canal de tatouage dans le domaine spectral**
- **Résultats et conclusions**

CHOIX D'UNE TRANSFORMEE (PERFORMANCES DE LA DCT)

6.1. INTRODUCTION

Dans ce chapitre nous allons expliquer le choix de l'utilisation de la DCT dans notre méthode, pour cela on estime le nombre de bits qui peuvent être tatoués dans une séquence vidéo compressée [55]. Les résultats obtenus à partir de la comparaison de divers types de transformées (DCT : Discrete Cosine Transform, HAR : Hartley, ID : Identité) vont être exposés pour étayer ce choix.

6.2. DEFINITIONS

6.2.1. Traitements préalables

Soit I_k une image originale de la séquence vidéo, à laquelle un signal message S_k (quelques bits d'informations) est ajouté :

$$F_k=I_k+S_k \qquad (1)$$

Où F_k est l'image modifiée d'une manière visuellement imperceptible.

Les images F_k subissent maintenant une compression MPEG avec $I'_k=C(F_k)$, où C() est l'opérateur de compression /décompression. Dans notre cas les bits d'informations seront extraits à partir de I'_k (image tatouée) sans que l'image originale I soit.

On peut distinguer deux sources de bruit qui sont : le bruit I dû à l'image originale et le bruit P dû au procédé de compression/décompression. S'est le message dit « corrompu ».

6.2.2. DCT (Discrete Cosine Transform)

De nombreuses méthodes ont été développées à partir des connaissances acquises auparavant en codage de source. Les auteurs des ces méthodes espèrent ainsi en travaillant dans le domaine DCT, anticiper et prévenir au moins les attaques liées à la compression MPEG. Ils espèrent également pouvoir travailler plus rapidement en couplant le tatouage vidéo avec le codage source. En d'autres termes le tatouage est réalisé directement sur le flux compressé

La DCT est un outil mathématique permettant d'utiliser le domaine spectral afin d'insérer le watermark. Une DCT bidimensionnelle sur des blocs de 8×8 pixels est utilisée. Les formules mathématiques décrivant ce traitement sont :

$$F(u,v)=\tfrac{1}{4}C(u)\times C(v)\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)\cos\left((2x+1)u\frac{\pi}{16}\right)\cos\left((2x+1)v\frac{\pi}{16}\right)$$

$$\begin{cases} C(j)=\dfrac{1}{\sqrt{2}} & \text{si } j=0 \quad \text{et u,v,x,y=1 ,...,7} \\ C(j)=1 & \text{si } j>0 \end{cases}$$

La DCT inverse est exprimée par :

$$f(x,y)=\tfrac{1}{4}\sum_{x=0}^{7}\sum_{y=0}^{7}C(u)\times C(v)F(u,v)\cos\left((2x+1)u\frac{\pi}{16}\right)\cos\left((2x+1)v\frac{\pi}{16}\right)$$

6.3. CAPACITE DU CANAL DE TATOUAGE DANS LE DOMAINE SPATIAL

Dans la figure 1, les deux sources de bruit dans le canal (I ~ [f_I(i) , σ_i^2] et P ~ [f_P(p) , σ_p^2]) peuvent être substituées par une source gaussienne. Ainsi, dans le but de connaître la capacité du canal, on calcule en premier lieu l'entropie différentielle du canal

Concernant l'optimisation de la capacité du canal de tatouage, le choix du domaine spectral est le plus adéquat. En effet, utiliser le domaine spectral permet de diviser le canal en sous-canaux, ce qui a pour effet d'atteindre le but fixé.

$$H=h(I)=-\int f_I(i)\log_2(f_I(i))di \qquad \text{bits} \qquad (2)$$

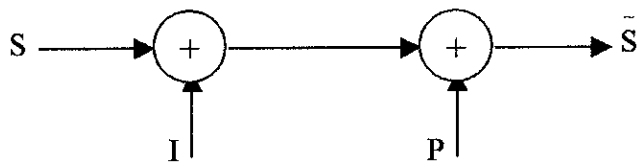

Figure-1 : Canal de tatouage

On calcul la variance σ_{ig}^2 du bruit gaussien équivalent, qui a la même entropie que l'image originale I. A présent, on remplace I ~ [f_I(i) , σ_i^2] par $I_g \sim N[0 , \sigma_{ig}^2]$. On peut supposer que le bruit P est de distribution gaussienne [56] car étant le résultat de plusieurs opérations indépendantes. Donc les deux bruits I et P peuvent être substitués par une source de bruit de variance $\sigma_{ig}^2+\sigma_p^2$. Si σ_s^2 est l'énergie du signal message, alors la capacité du canal est donnée par [55,57] :

$$C_h=\frac{1}{2}\log_2(1+\frac{\sigma_s^2}{\sigma_{ig}^2+\sigma_p^2}) \qquad (3)$$

D'une manière générale, l'entropie $h(I)$ et la variance gaussienne équivalente σ_{ig}^2 sont données par [56] :

$$h(I)= \frac{1}{2}\log_2(12\sigma_i^2) \text{ bits, et } \sigma_{ig}^2=\frac{12}{2\pi e}\sigma_i^2 \qquad (4)$$

où le bruit de l'image I (pixels de l'image) est donné par σ_i^2.

Les statistiques obtenues à partir de plusieurs images d'une séquence vidéo montrent que typiquement σ_i^2=55 donc σ_{ig}^2=46.1. En admettant une dégradation de l'image après addition du message avec un PSNR de 42 dB, alors σ_s^2 sera égale à 4.

De plus, si l'image est compressée en utilisant le procédé de compression MPEG (avec un taux de compression de 50), alors σ_p^2 sera égale à 6,7. Dans ce cas on aura une capacité du canal de l'ordre de 0.0013 bit/pixel. Cette valeur ne variera pas beaucoup même en dégradant la qualité de l'image compressée en fixant σ_p^2 à 20. Dans ce cas on a une capacité du canal de l'ordre de 0.0011 bit/pixel.

6.4. CAPACITE DU CANAL DE TATOUAGE DANS LE DOMAINE SPECTRAL

6.4.1. Besoin d'une décomposition

L'utilisation du domaine spectral est donc préférable à l'utilisation du domaine spatial, dans le but d'optimiser l'utilisation du canal en le divisant en sous-canaux. La figure 2 donne une distribution typique en fonction de la fréquence des bruits I et P dus à l'image originale et au procédé de compression/décompression respectivement.

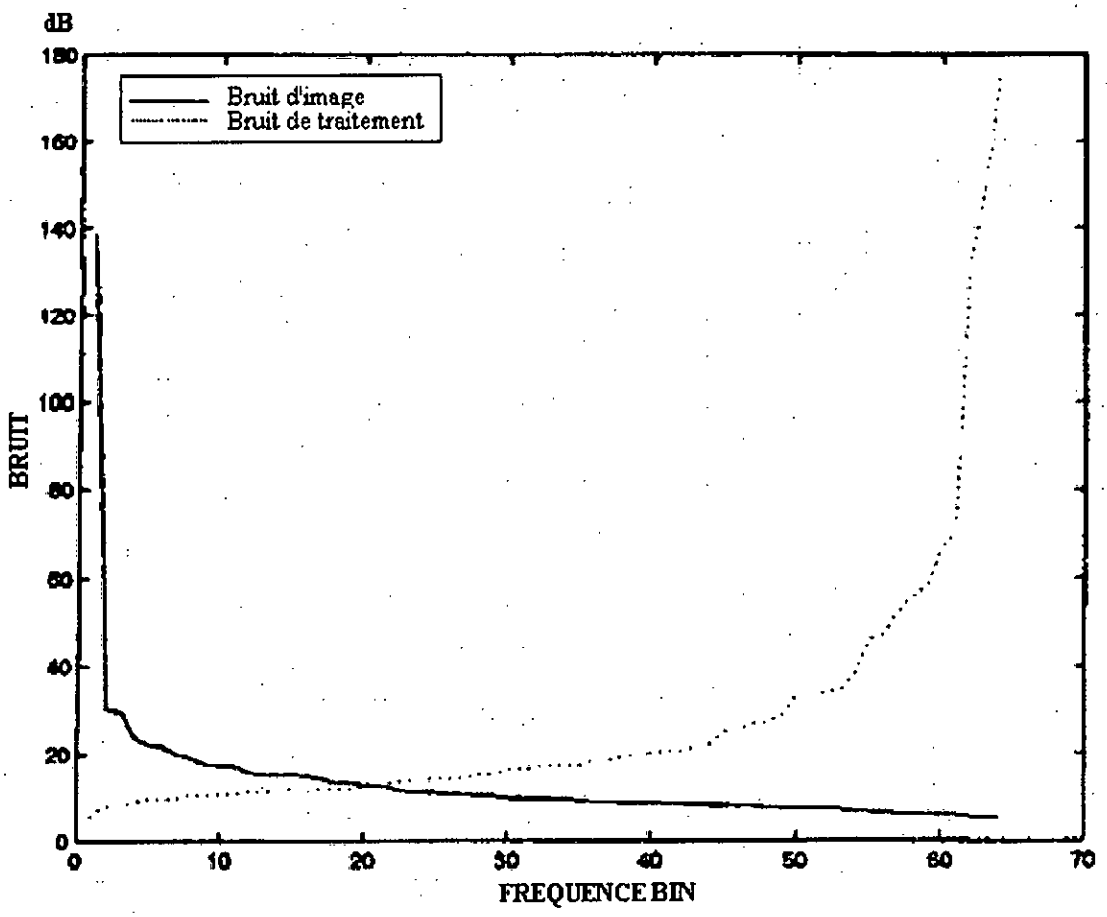

Figure-2 : Distribution fréquentielle typique du bruit d'image et du bruit dû au procédé de compression/décompression

On remarque qu'aux basses fréquences, le bruit I est élevé et le bruit P est faible, par contre, aux hautes fréquences on remarque le phénomène inverse. Ainsi, si le bruit dû au procède de compression/décompression est négligeable, alors une décomposition à l'aide d'une transformation à haut gain de codage [58] va concentrer le bruit dans un petit nombre de sous-canaux. A l'inverse, si le bruit dû au procédé de compression/décompression est élevé (vidéo MPEG de mauvaise qualité), les sous-canaux de hautes fréquences seront sévèrement affectés, ne laissant qu'un petit nombre de sous-canaux utilisables.

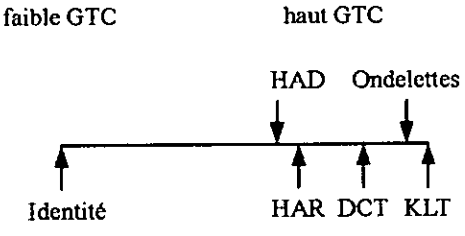

Figure-3 : Gain de codage de différentes transformation

A l'inverse, si on utilise une transformée à faible gain de codage, les fréquences de bandes moyennes ne seront relativement pas affectées par le procédé de compression/décompression. La figure 3 donne la position des différentes transformées au niveau de l'échelle en gain de codage.

6.4.2. Capacité de canaux multiples

La figure 4, donne le schéma bloc typique d'un procédé d'insertion/extraction de données à dissimuler. Les transformées et transformées inverses décomposent le canal en sous-canaux. La décomposition d'une image en L sous-bandes donne donc L sous-canaux.

Si $\sigma_{ij}, j=1..L$ est la variance des coefficients pour chaque sous-bande (la variance du bruit de l'image pour chaque sous-bande), et si $\sigma^2_{ig_j}$ et $\sigma^2_{p_j}$ sont les variances gaussiennes équivalentes des coefficients et du procédé de compression/décompression dans chaque sous-bande respectivement, alors la capacité combinée des L sous-bandes est donnée par : (pour une image de MxN pixels)

$$C_h=\frac{MN}{2L}\sum_{j=1}^{L}\log_2\left(1+\frac{v_j^2}{\sigma^2_{ig_j}+\sigma^2_{p_j}}\right) \text{ bits} \qquad (5)$$

où v_j est le seuil visuel donné par :

$$v_j^2=K_2\sigma^2_{q_j} \qquad (6)$$

et où $\sigma^2_{q_j}$ est la variance de l'erreur de quantification $e_{q_j}=\bar{i}_j-i_j$ avec i_{jk} les coefficients de l'image originale et $\bar{i}_{jk}$ les coefficients de l'image après compression.

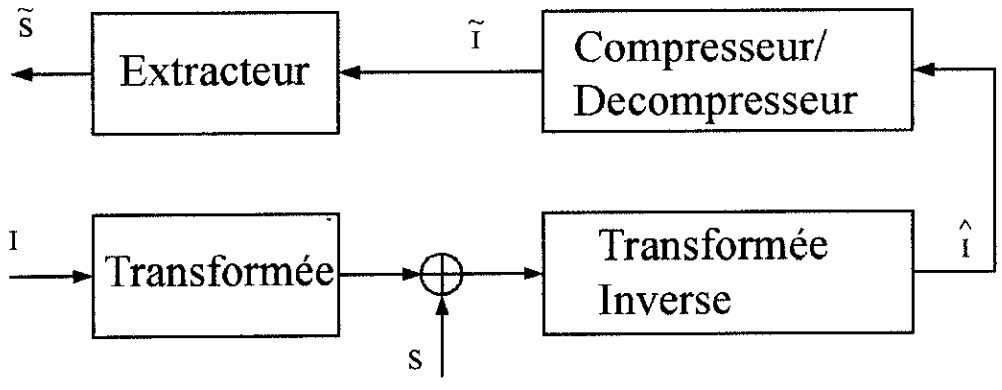

Figure-4 : Schéma d'insertion/extraction de données à dissimuler

6.5. RESULTATS ET CONCLUSIONS

Le calcul de la capacité de tatouage de 90 images de deux séquences vidéos monochromes (table tennis et football) donne les résultats donnés en figure 5 et 6. Les capacités ont été calculées pour trois types de transformées – DCT, Hartley (HAR) et identité (ID) – et ceci pour cinq différents scénarios et pour deux types d'images : image I (ligne du haut) et image P/B (ligne du bas). Les scénarios considérés sont une compression MPEG à 30 images/secondes, 15 images dans un groupe d'images (Goup Of Pictures ou GOP) et une distance d'image I/P de 3 à des taux de compression de 1,10,25,50 et 100 pour les scénarios 1,2,3 et 4 respectivement. Pour la simulation il a été choisi de donner la valeur 0.3 à K_2 dans l'équation 6.

A partir de la figure 5, on peut remarquer que le bit-rate chute au fur et à mesure que le bruit, dû au procédé de compression/décompression augmente.

La figure 6 donne les bruits d'images et le bruit dû au procédé de compression/décompression dans les 64 sous-canaux d'une décomposition DCT et Hartley (scénario 4 à un taux de compression de 50).

Alors que les sous-canaux de DCT de hautes fréquences sont affectés par un fort bruit dû au procédé de compression/décompression, les sous-canaux de Hartley de hautes fréquences eux ne sont pas autant affectés par ce bruit. Comme on s'y attendait la DCT donne de meilleurs résultats en ce qui concerne les scénarios où un faible bruit dû au procédé de compression/décompression est introduit (scénarios 1 et 2). La HAR donne de meilleurs résultats dans les autres cas (scénarios 3,4 et 5).

Finalement il n'y a pas de différence entre les images P, I et B, malgré le fait qu'elles subissent des traitements différents. Les images P et B donnent des capacités supérieures de 10% en comparaison avec les images I, ce qui n'est pas surprenant car le PSNR est supérieur de 2dB en faveur des images P et B par rapport aux images I. La différence entre les images P et B est négligeable et c'est pour cela qu'elles ont été groupées dans la figure 6.

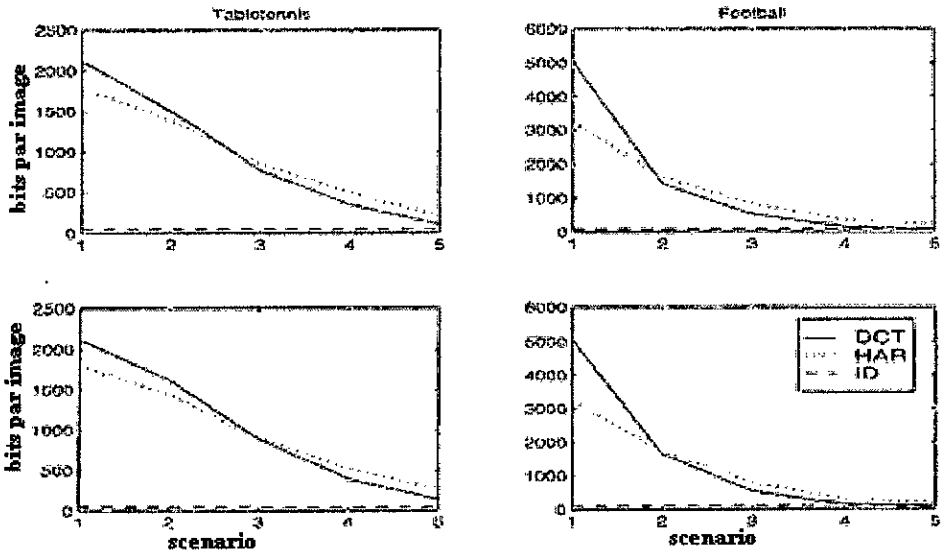

Figure-5 : Capacités des canaux de différentes décompositions. Ligne du haut : Images I, ligne du bas : Images P/B.

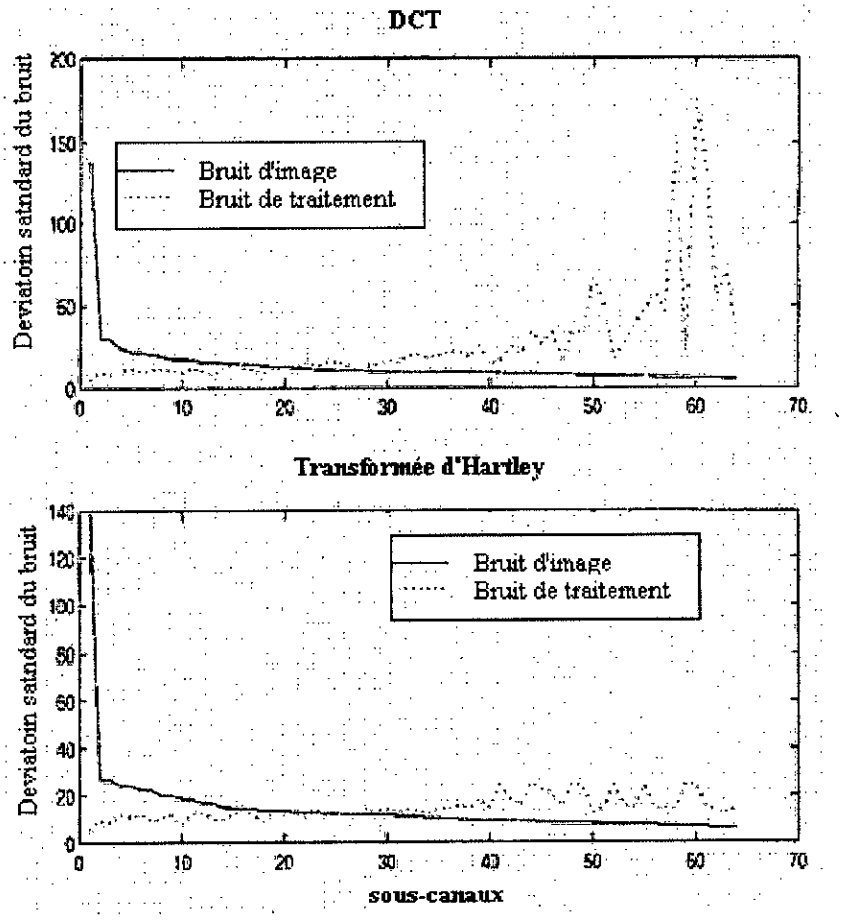

Figure-6 : Bruit d'image et de traitement de différents canaux pour des décomposition DCT et HAR

Il est à noter que l'analyse présentée considère que le bruit introduit comme conséquence aux

divers traitements que va subir le stego-objet est gaussien. Ceci n'est certainement pas une hypothèse qui convient dans tous les cas. Cependant, cette hypothèse est prise sachant que le but est de connaître les règles et heuristiques à appliquer d'une manière générale en considérant différents scénarios de traitements. Théoriquement le modèle gaussien peut être considéré comme étant un modèle de bruit général de par sa nature conservatrice. On justifie son adoption par un argument (en l'absence d'informations concernant les statistique du bruit), cette justification est le théorème central limite [59]. De plus, un bruit gaussien blanc additif représente la condition la plus difficile pour établir une communication [60].

Chapitre 7

METHODE DEVELOPPEE

- Introduction
- Estimation du mouvement
- Principe de dissimulation
- Détection
- Performances
- Mise en oeuvre de la méthode

METHODE DEVELOPPEE

7.1. INTRODUCTION

La méthode que nous avons développée est basée sur le spread spectrum (§ 3.1.) (Étalement de spectre) et utilise comme transformation de domaine la DCT (définition en § 6.2.2.) sur des blocs de 8×8 pixels. Les raisons du choix de la DCT sont la possibilité d'insérer plus de bits dans le canal qui est une vidéo compressée au format MPEG-1 avec un taux de compression de 10 (cf. § 6.4.2.) et le fait que le MPEG utilise cette transformée sur des blocs de même taille parce que le MPEG n'altérera pas la marque en procédant à la transformation de domaine(c.f. § 2.4., [1]).

Dans le but d'augmenter les performances de détection nous avons utilisé un codage correcteur d'erreur et un éparpillement (interleaving) ce qui permet de corriger des erreurs apparaissant sur des bits successifs, ce qui est souvent le cas en pratique.

Nous avons également utilisé le principe de superposition des watermarks (cf. § 2.1.3.) pour dissimuler deux labels de 20 bits chacun dans chaque image, avec deux clé pseudo-aléatoires différentes dans des vidéos de 240×320 pixels. C'est ce qui nous permet d'avoir un fingerprint de 40 bits.

Notre méthode utilise le domaine temporel ainsi qu'introduit par Min Wu et Hong Heather yu en [44] (voir chapitre 5 figure 1), afin d'accroître le nombre de labels dissimulés et la robustesse par rapport aux divers types de compressions

Afin d'améliorer la robustesse par rapport aux attaques dites *collusion attacks* et aux attaques statistiques, nous faisons une estimation du mouvement (§ 7.2.) au niveau de zones prédéterminées de la vidéo, pour insérer des bits de bourrage ou de compensation fixes ou changeants selon le cas. Ceci nous permet d'utiliser un watermark fixe ou un watermark indépendant selon les caractéristiques de la zone (fixe ou en mouvement).

7.2. ESTIMATION DU MOUVEMENT

Comme mentionné ci-dessus nous utilisons l'estimation de mouvement pour nous prémunir contre les attaques statistiques et les *collusion attacks* :

Pour les watermarks indépendants les attaques statistiques exploitent les changements dus au watermark au niveau des zones non mouvementées en faisant une comparaison et une « moyenne » entre les images pour ôter le watermark.

Pour ôter le label, les *collusion attacks*, quant à elles, exploitent les similarités dans les zones mouvementées et également dans des scènes complètement différentes.

L'estimation du mouvement se fait sur des blocs dont la taille peut être ajustée, la taille des blocs devant être toujours supérieure à celle d'un bloc de base de 8x8 pixels.

Nous distinguons donc deux types de blocs : des blocs dits mouvementés (1) et des blocs non mouvementés (0). Nous avons choisi un rapport de 70% de blocs non mouvementés contre 30% de blocs mouvementés, sur la base d'une étude statistique faite sur des échantillons d'images de divers vidéos.

Pour simplifier le processus de traitement, nous avons gardé la même estimation sur un ensemble d'images successives. La taille des zones, sur lesquelles la décision est prise, est ajustable en fonction du type de la vidéo à traiter. Ainsi, une fois la taille des zones fixée, le même traitement est effectué sur chacune d'elles, puis, une fois la décision prise (mouvementé ou pas) elle concernera les images suivantes.

Nous définissons des blocs 3D de taille NxNxM avec :

N : taille des zones (ou macroblocs) de l'image sur lesquelles la décision sera prise (mouvementées ou pas)

M : profondeur (ou nombre d'images successives) sur laquelle on aura la même décision pour les mêmes zones.

Sur ces blocs 3D nous calculons le « taux de changement » des pixels qui donnera une mesure du mouvement le long de l'axe temporel pixel par pixel de la façon suivante :

Si $X_p^k(m_j,n_j)$ est la valeur du pixel de la j-ème image dans le k-ème bloc 3D et $V_t^k(n_j,m_j)$ le taux de changement du pixel le long de l'axe des temps t sera.

$$V_t^k(n_j,m_j)=\frac{X_p^k(m_{j+1},n_{j+1})-X_p^k(m_j,n_j)}{t_{j+1}-t_j} \qquad (1)$$

où $1\le n_j\le blk\_size$, $1\le m_j\le blk\_size$, $1\le j\le depth$.

et où k est le nombre de blocs 3D de dimensions blk_size× blk_size ×depth, depth et blk_size étant paramétrables.

Pour augmenter la vitesse de calcul, on sélectionne les images à des intervalles de quatre images dans le bloc 3D le long de l'axe temporel. Le bloc 3D original de taille blk_size x blk_size x depth est reéchelonné en un bloc 3D de taille blk_size x blk_size x (depth/4).

Aprés le calcul du taux de changement d'un pixel à l'autre le long de l'axe des temps dans le k-ème bloc 3D, la valeur absolue maximale du taux sera donnée par :

$$V_{max}^{k}(m,n) = \max_{j}\left\{V_{t}^{k}(nj, m_j)\right\} \qquad (2)$$

La sommation des blk_size x blk_size valeurs maximales du taux de changement du k-ème bloc 3D est :

$$V_{som-max}^{k} = \sum_{m=1}^{blk\_size}\sum_{n=1}^{blk\_size} V_{max}^{k}(m,n) \qquad (2\text{-a})$$

Soit B_{max} la valeur maximale des valeurs calculées précédemment donnée par :

$$B_{max} = \max_{k}\left\{V_{som-max}^{k}\right\} \qquad (a\text{-b})$$

B_{max} sera la valeur de référence à partir de laquelle, sur un total de k blocs, 70% seront dits non mouvementés et 30% seront dits mouvementés selon la valeur $V_{som-max}^{k}$ de ces blocs.

7.3. PRINCIPE DE DISSIMULATION

Le principe de dissimulation (figure 2) réside dans le fait de tatouer chaque image de la vidéo séparément et de répéter le traitement effectué sur chaque image afin d'accroître la robustesse de l'ensemble, principalement par rapport aux attaques de changement de l'ordre des images, de suppression d'images, de compression, de filtrage, de redimensionnement...

Donc la première étape du processus de tatouage consiste en l'évaluation du mouvement au niveau des zones déterminées selon le principe précédemment explicité.

Puis, au niveau de chaque image, un changement de domaine est effectué en appliquant une DCT (Discrete Cosine Transform) sur des blocs de base de 8x8 pixels.

Le watermark à tatouer se compose de deux labels (Label1 et Label 2), chacun d'eux sera dissimulé à l'aide d'une clé (clé1 et clé 2 respectivement).

L'étape suivante consistera en l'insertion du watermark : Soit a_j, $a_j \in \{1,-1\}$, la séquence de bits constituant les labels du watermark. On insérera chaque bit du label au niveau de –cr– coefficients DCT afin d'ajouter une redondance, –cr– est appelé chip-rate ou facteur d'étalement, ainsi : $b_i=a_j$ $j \in N$ et $i \in N$ $j \times cr \le i < (j+1) \times cr$.

La séquence b_i est amplifiée en utilisant un facteur d'ajustement α pour exploiter les phénomènes de masquage spatial et temporel qui caractérisent le SVH (Systéme Visuel Humain) et augmenter l'énergie de la marque. La séquence b_i est ensuite modulée en utilisant une séquence pseudo-aléatoire p_i, $pi \in \{1,-1\}$, $i \in N$. Le résultat de ces traitements donnera le signal watermark modulé :

$$w_i = \alpha_i \times p_i \times b_i \qquad (3)$$

L'utilisation de la séquence pseudo-aléatoire améliore la robustesse de l'ensemble en rendant difficile la détection, la localisation et la manipulation de la marque sans la connaissance exacte de cette séquence.

Nous allons utiliser les coefficients DCT de 2 à 13 de chaque bloc DCT (lecture des coefficients en zig zag) pour insérer le watermark et les bits de compensations au niveau de chaque bloc DCT de 8×8. Notre choix s'est porté sur ces coefficients car ceux-ci contiennent la majeure partie de l'énergie de l'image, ce qui rendra d'éventuelles tentatives d'annulation du watermark impossibles, de plus, ces coefficients resteront inchangés lors d'une compression MPEG.

La lecture des coefficients DCT se fait de la manière suivante (figure 1): Les macroblocs, sur lesquels l'estimation du mouvement est effectuée, sont lus de gauche à droite et de haut en bas. De la même manière, les blocs DCT de 8×8 constituant les macroblocs sont lus de gauche à droite et de haut en bas. Les coefficients DCT au niveau de chaque bloc DCT de 8×8 coefficients sont lus en zigzag de la même façon que la lecture faite par le format MPEG (§ 4.2.1 figure 6).

Ainsi lus, les 8 premiers coefficients sont tatoués à l'aide du watermark w_i et les 2 coefficients suivants sont tatoués à l'aide de 2 bits de compensation fixes ou variables selon que les coefficients tatoués se trouvent dans un macrobloc mouvementé ou pas.

A la fin de ce processus nous appliquons une transformation DCT inverse pour reconstruire la vidéo.

Afin de tatouer les deux labels au niveau d'une même image nous utiliserons la propriété d'addition de label tatoués : en effet, si on tatoue les deux labels à l'aide de deux clés différentes, il sera possible de retrouver les deux labels (indépendamment) lors de la détection.

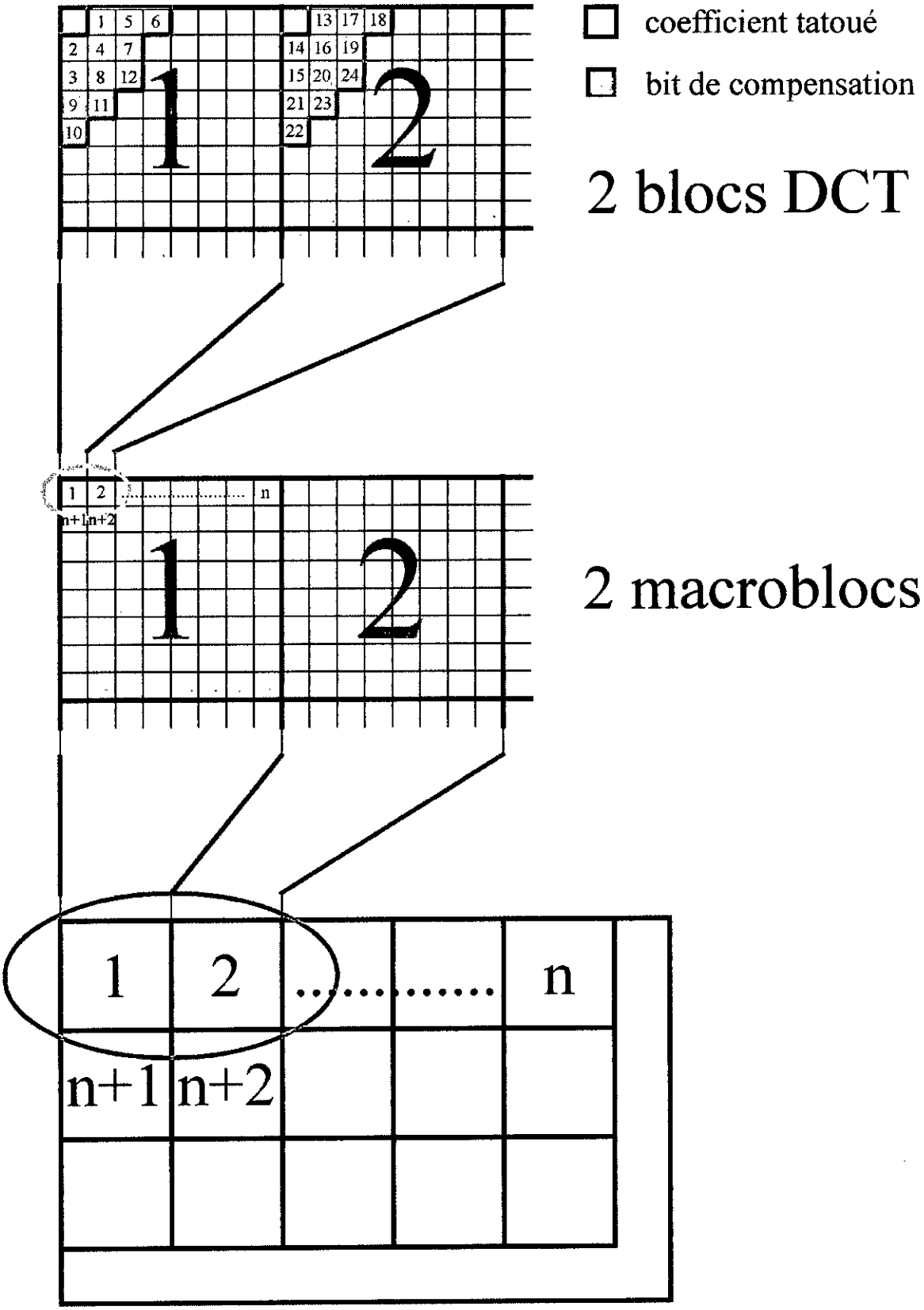

Figure-1 : Ordre d'écriture et de lecture dans notre procédé de watermarking

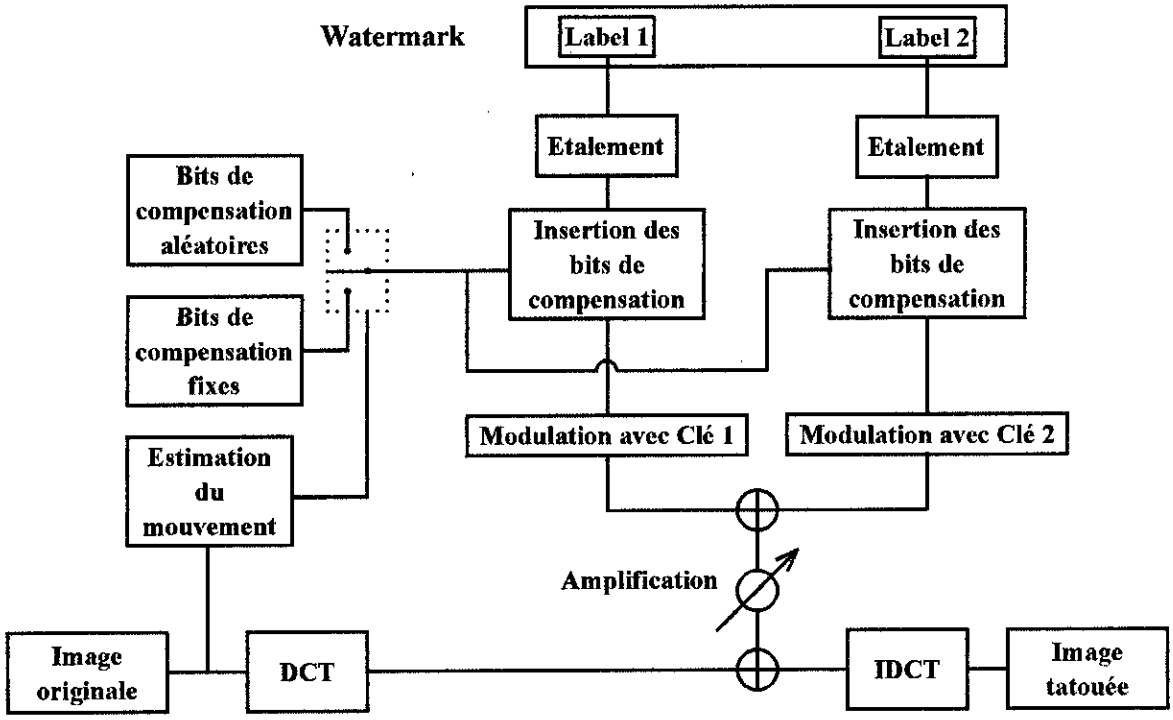

Figure-2 : Schéma bloc du processus de watermarking

7.4. DETECTION

La procédure de détection (figure 3) utilisée ne requiert pas la vidéo originale, c'est une détection dite non cohérente.

Ainsi, nous utilisons les images de la vidéo tatouée en appliquant une transformation de domaine (DCT) à n'importe quelle image de cette séquence vidéo. Par la suite nous démodulons le watermark.

L'ordre de lecture des coefficients DCT des blocs de base est semblable à celui de l'écriture, ce qui nous permettra de reconstituer la séquence :

$$\overline{v_i} = v_i + w_i. \qquad (4)$$

Nous utilisons une corrélation afin de démoduler la vidéo tatouée et de retrouver la marque a_i. Ceci va se faire de la même manière que dans [54] en multipliant la séquence $\overline{v_i}$ par la séquence pseudo-aléatoire p_i (qui suit une loi normale, $\mu_p = 0$, σ_p^2).

Ainsi nous obtiendrons la somme de corrélations s_j :

$$s_j = \Sigma_1 + \Sigma_2 = \sum_{i=j\times cr}^{(j+1)\times cr-1} p_i \times \overline{v_i} = \sum_{i=j\times cr}^{(j+1)\times cr-1} p_i \times v_i + \sum_{i=j\times cr}^{(j+1)\times cr-1} p_i \times \alpha_i \times b_i \times p_i \qquad (5)$$

Où Σ_1 et Σ_2 désignent les contributions à la somme de corrélations du signal vidéo et du signal watermark.

Avant d'examiner l'équation précédente d'une manière plus détaillée, considérons la somme Σ_1 sensiblement nulle. On obtiendra ainsi :

$$s_j = \Sigma_1 + \Sigma_2 \approx \sum_{i=j\times cr}^{(j+1)\times cr-1} p_i^2 \times \alpha_i \times b_i \qquad (6)$$

Où σ_p^2 est la variance de la séquence pseudo-aléatoire p_i.

En considérant l'équation 6 et étant donné que : $b_i=a_j$ pour $j\times cr \le i < (j+1)\times cr$ et $\mu_p = 0$, on obtient :

$$s_j = \sum_{i=j\times cr}^{(j+1)\times cr-1} p_i^2 \times \alpha_i \times b_i = a_j \times \sum_{i=j\times cr}^{(j+1)\times cr-1} p_i^2 \times \alpha_i$$

$$= a_j \times \alpha_j \times \sum_{i=j\times cr}^{(j+1)\times cr-1} (p_i - \mu_p)^2$$

$$= a_j \times \alpha_j \times \sigma_p^2 \times cr \qquad (7)$$

Le signe de la somme de corrélation sera donc égal au bit d'information dissimulé :

$$sign(s_j) = sign(a_j \times \sigma_p^2 \times cr \times \alpha_j) = sign(a_j) = a_j \qquad (8)$$

L'équation (8) donnera donc un (+1) si la corrélation entre le signal vidéo et le signal pseudo-aléatoire est positive, et un (-1) si la corrélation est négative.

L'extraction des labels constituant le watermark a lieu sur plusieurs images de la séquence vidéo. Ceci nous permettra d'éviter de commettre des erreurs en prenant en compte le nombre de fois où la valeur de a_i est à (+1) ou (-1) sur les images démodulées.

Si la séquence pseudo-aléatoire utilisée à la détection ne correspond pas à celle utilisée lors de l'insertion du watermark, ou si cette séquence n'est pas en synchronisation parfaite avec celle utilisée à l'insertion, la détection ne se fera pas correctement et les bits retrouvés seront aléatoires.

Le recouvrement de l'information dissimulée, ainsi que décrit précédemment ne requiert pas la vidéo originale non tatouée. Cependant, le recouvrement sera plus robuste si la vidéo originale

est connue, et permettra de la soustraire avant la démodulation (Σ_1 sera nul dans l'équation 5). Cette soustraction va éliminer toute interférence entre le signal vidéo et watermark tatoué.

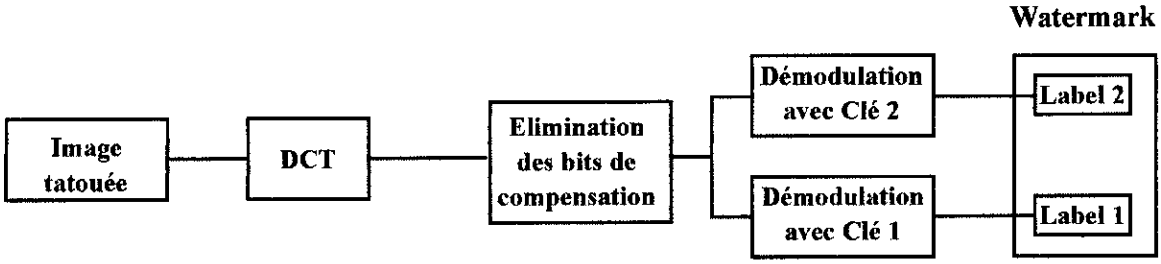

Figure-3 : Schéma bloc du processus de détection

7.5. PERFORMANCES

La supposition que $\Sigma_1=0$ n'est pas valable car il faut prendre en compte l'énergie de la vidéo originale. Les erreurs de détection auront lieu si $sign(\Sigma_1+\Sigma_2)\neq sign(\Sigma_2)$ et ce cas se présente lorsque $sign(\Sigma_1)\neq sign(\Sigma_2)$ et $|\Sigma_1|>|\Sigma_2|$.

Dans ce qui suit, la probabilité que ceci ait lieu, est calculée en fonction des paramètres utilisés (chip rate cr, facteur d'amplification α, variance de la fonction pseudo-aléatoire σ_p^2) et des propriétés des signaux vidéo en général. Les valeurs typiques de la moyenne et de la variance d'une image vidéo dans le domaine spatial et transformé sont données par la table 1.

Domaine	Moyenne (μ)	Variance (σ^2)
Spatial	μ_s=127,5	σ_s^2=5461,2
Fréquentiel	μ_f=20,94	σ_f^2=548,84

Table-1 : Valeurs typiques de la moyenne et de la variance d'une image vidéo dans les domaines spatial et transformé.

Dans la somme Σ_1 (qui décrit le terme de distorsion) le signal vidéo sera multiplié par la séquence p (de moyenne $\mu_p=0$ et de variance σ_p^2). La moyenne du produit p×v sera $\mu_{pv}=0$ et la variance $\sigma_{pv}^2=\sigma_p^2\times(\sigma_v^2+\mu_v^2)$.

Dans Σ_1, le produit p v est sommé cr fois. Ainsi, en se referant au théorème central-limit, la fonction de densité de probabilité de la somme est assimilée à une distribution normale de moyenne $\mu_{\Sigma 1}=cr\times\mu_{pv}=0$ et de variance $\sigma_{\Sigma 1}^2=cr\times\sigma_{pv}^2=cr\times\sigma_p^2\times(\sigma_v^2+\mu_v^2)$.

Une erreur a lieu si le bit d'information courant est un +1 et $\Sigma_1 < -\sigma_p^2 \times cr \times \alpha$ ou si le bit est un −1 et $\Sigma_1 > \sigma_p^2 \times cr \times \alpha$. Comme Σ_1 est assimilé à une loi normale, le BER (Bit Error Rate) est donné par :

$$BER = P(|\Sigma_1| > \sigma_p^2 \times cr \times \alpha) \qquad (9)$$

$$= \frac{1}{\sqrt{2 \times \pi \times \sigma_{\Sigma 1}^2}} \times \int_{\sigma_p^2 \times cr \times \alpha}^{\infty} \exp\left(-\frac{t^2}{2 \times \sigma_{\Sigma 1}^2}\right) dt$$

$$= \frac{1}{\sqrt{2 \times \pi \times \sigma_{\Sigma 1}^2}} \times \sqrt{\frac{\pi}{2}} \times \sigma_{\Sigma 1} \times erfc\left(\frac{\sigma_p^2 \times cr \times \alpha}{\sqrt{2} \times \sigma_{\Sigma 1}}\right)$$

$$BER = \frac{1}{2} \times erfc\left(\frac{\sigma_p \times \sqrt{cr \times \alpha}}{\sqrt{2} \times \sqrt{\sigma_v^2 + \mu_v^2}}\right) \qquad (10)$$

Si on augmente le chip rate –cr-, le facteur d'amplification α ou la variance du signal pseudo-aléatoire, le BER diminue. La table 2 donne quelques exemples de paramètres utilisés et de BER résultants.

Cr	α	BER
100	5	0,0376
	10	$1,86.10^{-4}$
	15	$4,72.10^{-8}$
	20	$5,54.10^{-13}$
200	5	0,0059
	10	$2,42.10^{-7}$
	15	$2,21.10^{-14}$
	20	$3,99.10^{-24}$
300	5	0,001
	10	$3,57.10^{-10}$
	15	$1,18.10^{-20}$
	20	$3,29.10^{-35}$

Table-2 : Exemple d'estimations et de mesures de bit error

Pour améliorer les performances de notre algorithme nous avons eu recours au codage correcteur d'erreur (7,4) de Hamming dont le principe est le suivant :

Pour quatre bits d'information b_0 b_1 b_2 b_3 nous ajoutons trois bits de contrôle (bits superviseurs) r_0 r_1 r_2 comme suit :

$r_0 = b_3 \oplus b_1 \oplus b_0$

$r_1 = b_3 \oplus b_2 \oplus b_0$

$r_2 = b_3 \oplus b_2 \oplus b_1$

Pour le décodage nous utilisons les bits de contrôle suivants :

$s_1 = b_3 \oplus b_2 \oplus b_1 \oplus r_2$

$s_2 = b_3 \oplus b_2 \oplus b_0 \oplus r_1$

$s_3 = b_3 \oplus b_1 \oplus b_0 \oplus r_0$

Suivant la valeurs de ces bits nous pourrons détecter le bit erroné (1 bit sur les 7 ainsi formés peut être corrigé). Le bit erroné est donné par la table 3 :

s_1 s_2 s_3	bit erroné	s_1 s_2 s_3	bit erroné
0 0 1	r_0	1 0 1	b_1
0 1 0	r_1	1 1 0	b_2
1 0 0	r_2	1 1 1	b_3
0 1 1	b_0	0 0 0	Pas d'erreur

Table-3 : Correction des bits erronés

Ce codage est appliqué au label avant de l'insérer donc les bits superviseurs en feront partie et seront tatoués eux aussi. Comme nous ne pouvons corriger qu'un bit sur sept et qu'en général les erreurs sont groupées, nous avons éparpillé (interleaving) les bits à tatouer. L'éparpillement nous permet de corriger jusqu'à quatre erreurs de suite.

7.6. MISE EN ŒUVRE DE LA METHODE

L'application a été développée sous environnement MATLAB v5.3 en raison de l'existence de fonctions intégrées qui nous ont permis de lire les images vidéos consécutives constituant la séquence vidéo à tatouer et aussi en raison de l'existence de la fonction permettant d'appliquer la transformation DCT bidimensionnelle. Ces deux fonctions sont disponibles au niveau de la librairie IMAGE PROCESSING TOOLBOX.

Lors du développement de l'algorithme nous avons minimisé l'appel aux fonctions intégrées fournies par MATLAB (pour des opérations telles que le modulo par exemple) pour optimiser le temps de calcul que notre traitement prend. En effet, la complexité de l'algorithme développé est une fonction en $O(n^4)$, du fait que nous avons imbriqué au plus quatre boucles et que nous n'avons aucun appel récursif. Aussi et afin de réduire le temps de calcul nous avons compilé le

programme écrit (en ce qui concerne la plupart des fonction écrites) en utilisant le compilateur fourni par MATLAB (compilé en C++) ceci nous a permis de réduire ce temps de calcul, mais en raison de l'utilisation de fonctions intégrées de la librairie IMAGE PROCESSING TOOLBOX nous n'avons pas pu générer un exécutable (*.exe) en raison de l'utilisation et la génération par ces fonction de variables du type UINT8 (unsigned integer values codées sur 8 bits), type que ne supporte pas le compilateur C++. Nous avons donc généré pour ces fonctions des fichiers dits 'MEX files' qui, au contraire des fichier exécutables, sont toujours dépendants de MATLAB pour les exécuter.

Les fonctions écrites sont :

watermarking1 : script principal dont les paramètres d'entrées sont :

function watermarking1

n1=1; (première image vidéo à tatouer)

n2=16; (dernière image à tatouer)

depth=16; (profondeur ou nombres d'images successives utilisées pour l'estimation du mouvement)

blk_size=64; (taille des sous-blocs utilisés pour l'estimation du mouvement)

label1=[1 1 1 1 1 -1 1 -1 1 1 -1 1 1 -1 -1 1 -1 1 -1 1 1 -1 1 -1 1 1 1 1 1 -1 -1 1]; (Exemple de Label constituant le watermark)

label2=[1 1 -1 1 1 -1 1 1 1 1 -1 1 1 1 -1 -1 1 1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 1 1]; (Exemple de Label constituant le watermark)

alpha=20; (Amplification de l'énergie du watermark)

cr=300; (facteur d'étalement d'un bit des Labels)

lire : lecture des images constituant la séquence vidéo originale.

pn_mtrx : génération de clés de cryptage.

motio_est : estimation du mouvement.

Dctmtrx : applique une DCT bidimensionnelle sur des blocs e 8×8 pixels des images de la video originale.

embeded_mtx : étale les bits des différents Labels.

idctmtrx : applique une DCT bidimensionnelle inverse sur des blocs e 8×8 coefficients DCT des images de la vidéo tatouée.

ecrire : forme une séquence vidéo à partir des images vidéos tatouées.

demod : extrait les Labels constituant le watermark.

compil : compile à l'aide d'un compilateur C++ les fonctions précédentes.

Afin, de simplifier l'entrée des paramètres de tatouage et d'extraction nous avons développé un interface graphique plus convivial qui se présente comme suit :

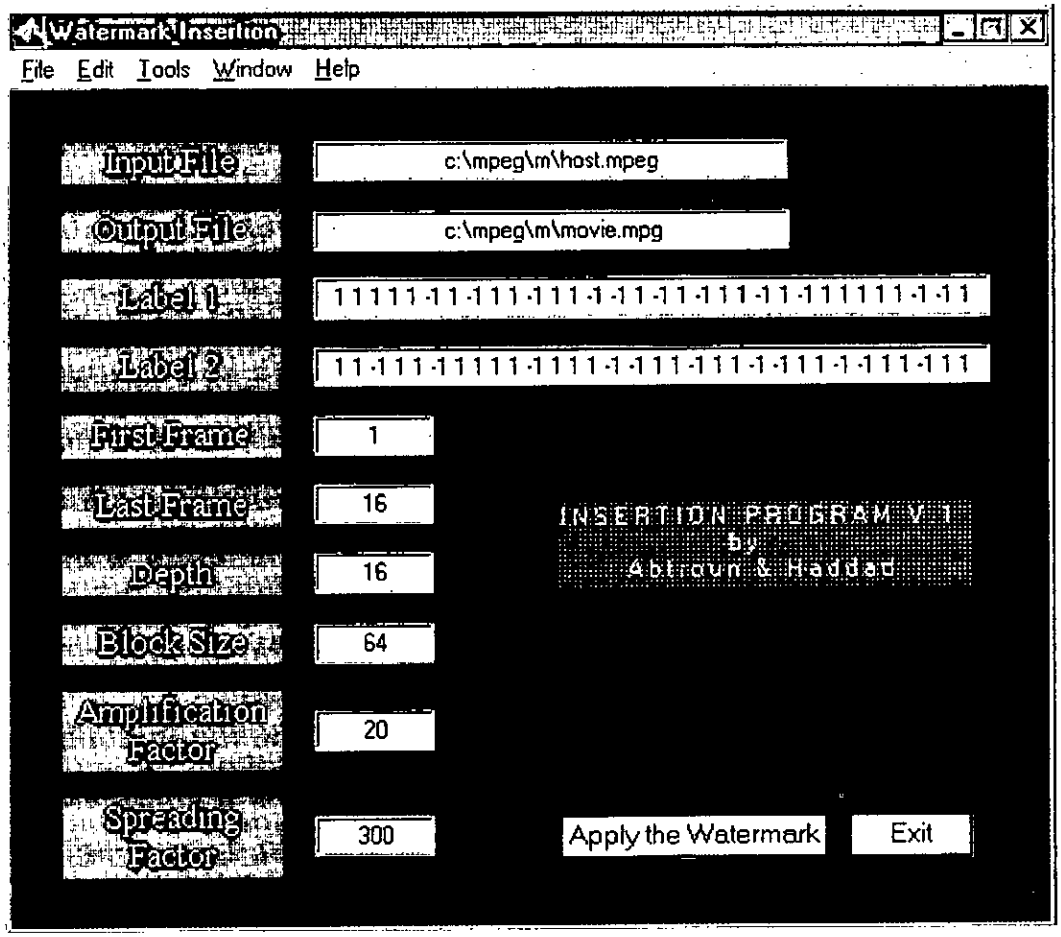

Figure-4 : GUI (Graphic User Interface) d'insertion du watermark

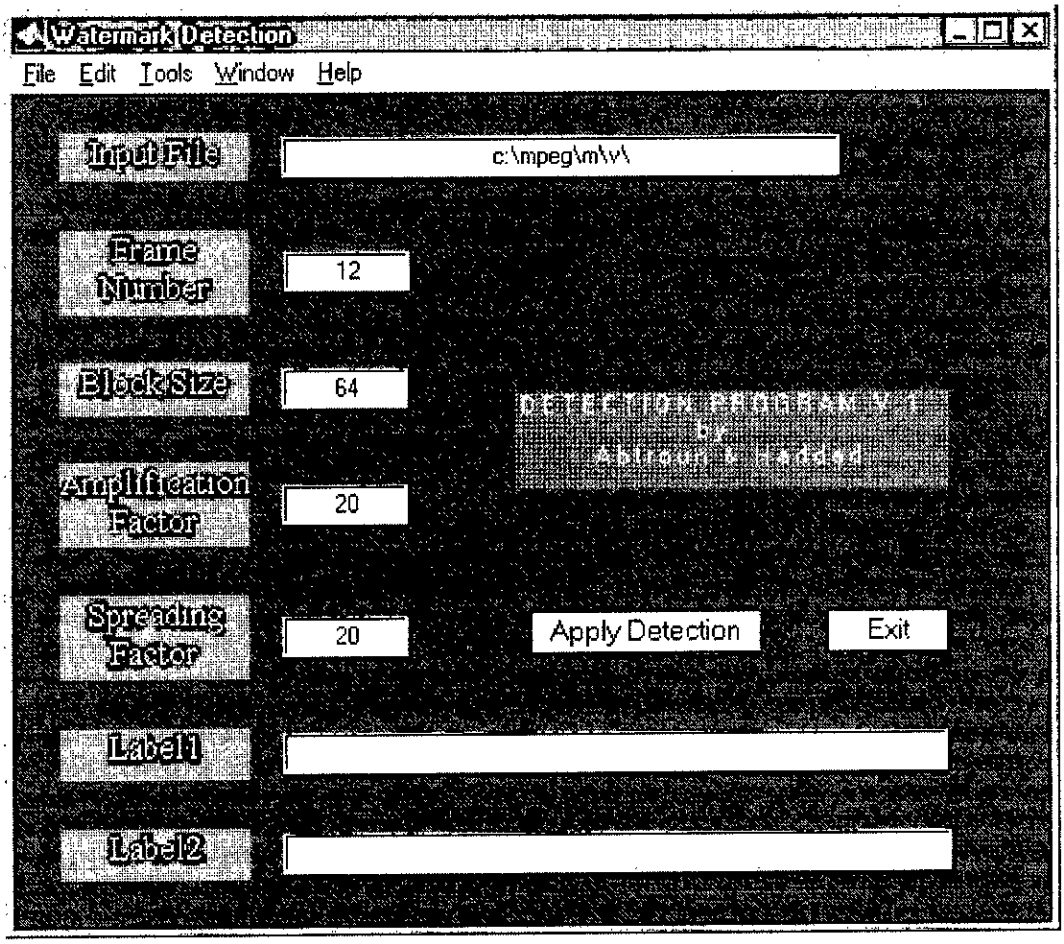

Figure-5 : GUI (Graphic User Interface) d'extraction du watermark

Chapitre 8

EXPERIMENTATIONS ET RESULTATS

- **Introduction**
- **Expérimentations**
- **Résultats et perspectives**

EXPERIMENTATIONS ET RESULTATS

8.1. INTRODUCTION

Afin d'évaluer les performances (imperceptibilité et robustesse) de l'algorithme nous avons effectué une batterie de tests consistant en l'évaluation du PSNR entre la vidéo tatouée et la vidéo originale. Nous avons aussi attaqué la vidéo tatouée à l'aide des différents traitements usuels qu'elle pourrait subir sans altérer sa qualité d'une manière significative (estimation basée sur le PSNR).

La vidéo tatouée est de taille de 240*320 pixels, constituée de 248 images, et codée au standard MPEG-1 (30 images par seconde).

Le watermark est constitué de deux Label (Label1 et 2) qui sont insérés à l'aide de deux clés différentes. Les Labels sont insérés indépendamment l'un de l'autre au niveau de la même série de coefficients DCT sélectionnée pour la dissimulation de chaque Label.

La table 1 donne le rapport signal sur bruit entre la vidéo originale et la vidéo tatouée (PEAK SIGNAL –to-NOISE RATIO ou PSNR)

Les tables 2,3,4 et 5 donnent le PSNR entre l'image vidéo tatouée attaquée et l'image vidéo originale.

IMAGES VIDEO	PSNR									
Image vidéo de 1 à 10	51,18	51,09	51,06	50,89	50,80	50,71	50,69	50,53	50,31	50,31
Image vidéo de 11 à 20	50,09	50,00	49,88	49,79	49,68	49,57	49,48	49,35	49,26	49,12
Image vidéo de 21 à 30	49,01	48,91	48,81	48,68	48,53	48,45	48,36	48,30	48,19	48,09
Image vidéo de 31 à 40	47,98	47,88	47,77	47,67	47,57	47,49	47,41	47,38	47,29	47,24
Image vidéo de 41 à 50	47,20	47,19	47,20	47,17	47,14	47,08	47,07	47,11	47,05	46,98
Image vidéo de 51 à 60	46,94	46,84	46,83	46,82	46,84	46,86	46,91	46,91	49,26	46,94
Image vidéo de 61 à 63	46,95	46,98	46,99							

Table-1 : PSNR entre les images de la vidéo originale et celles de la vidéo tatouée

IMAGES VIDEO	PSNR									
Image vidéo de 1 à 10	35,51	34,84	34,82	34,38	34,11	33,77	33,86	33,43	33,14	33,14
Image vidéo de 11 à 20	32,76	32,64	32,56	32,52	32,33	32,16	32,05	31,83	37,61	31,61
Image vidéo de 21 à 30	31,48	31,32	31,19	31,09	30,87	30,78	30,63	30,49	30,36	30,24
Image vidéo de 31 à 40	30,15	30,8	30,04	30	30	30,01	29,97	29,93	28,56	2,991
Image vidéo de 41 à 50	29,8	29,6	29,55	29,4	29,31	29,29	29,1	29,05	28,97	28,86
Image vidéo de 51 à 60	28,88	28,89	28,94	29,02	29,28	29,52	29,59	29,62	29,6	29,68
Image vidéo de 61 à 63	29,71	29,63	29,63							

Table-2 : PSNR entre les images de la vidéo originale et celles de la vidéo attaquée par compression

IMAGES VIDEO	PSNR									
Image vidéo de 1 à 10	30,86	31,06	31,19	30,82	30,83	30,98	31,04	30,68	30,8	30,78
Image vidéo de 11 à 20	30,35	30,32	30,58	30,52	30,29	30,35	30,22	29,87	29,75	29,84
Image vidéo de 21 à 30	29,71	29,49	29,47	29,3	29,01	28,95	28,96	28,88	28,69	28,63
Image vidéo de 31 à 40	28,59	28,4	28,35	28,39	28,42	28,39	28,4	28,37	28,25	28,28
Image vidéo de 41 à 50	28,22	28,08	28,02	27,9	27,77	27,71	27,6	27,57	27,52	27,41
Image vidéo de 51 à 60	27,52	27,54	27,58	27,72	28,03	28,15	28,19	28,36	28,35	28,38
Image vidéo de 61 à 63	28,38	28,42	28,4							

Table-3 : PSNR entre les images de la vidéo originale et celles de la vidéo attaquée par un filtre 3×3

IMAGES VIDEO	PSNR									
Image vidéo de 1 à 10	32,41	32,53	32,68	32,26	32,25	32,32	32,42	32,03	32,1	32,11
Image vidéo de 11 à 20	31,63	31,56	31,77	31,69	31,43	31,41	31,25	30,88	30,78	30,8
Image vidéo de 21 à 30	30,65	30,43	30,39	30,21	29,93	29,85	29,83	29,73	29,54	29,46
Image vidéo de 31 à 40	29,4	29,24	29,2	29,23	29,25	29,22	29,23	29,19	29,07	29,12
Image vidéo de 41 à 50	29,04	28,87	28,83	28,7	28,58	28,51	28,4	28,37	28,33	28,22
Image vidéo de 51 à 60	28,32	28,34	28,38	28,52	28,85	29,02	29,06	29,23	29,19	29,21
Image vidéo de 61 à 63	29,2	29,24	29,23							

Table-4 : PSNR entre les images de la vidéo originale et celles de la vidéo attaquée par flou

IMAGES VIDEO	PSNR									
Image vidéo de 1 à 10	31,73	31,74	31,75	31,42	33,73	31,42	31,49	31,13	31,22	31,22
Image vidéo de 11 à 20	30,82	30,75	30,94	30,9	30,71	30,69	30,59	30,26	30,18	30,21
Image vidéo de 21 à 30	30,09	29,88	29,83	29,69	29,42	29,35	29,34	29,26	29,08	29,02
Image vidéo de 31 à 40	28,96	28,82	28,78	28,79	28,83	28,83	28,83	28,8	28,7	28,76
Image vidéo de 41 à 50	26,95	28,55	28,5	28,38	28,25	28,2	28,07	28,03	27,98	27,89
Image vidéo de 51 à 60	27,96	27,98	28,03	28,16	28,45	28,61	28,66	28,74	28,71	28,75
Image vidéo de 61 à 63	28,77	28,76	28,74							

Table-5 : PSNR entre les images de la vidéo originale et celles de la vidéo attaquée par redimensionnement

8.2. EXPERIMENTATIONS

La vidéo tatouée a été attaquée à l'aide d'un certain nombre de traitements disponibles au niveau du logiciel de traitement de vidéo VirtualDub V1.4c, ces attaques sont :

8.2.1. Compression

Attaque non ciblée : ce traitement n'est pas considéré comme une attaque en soit, en effet, le but de la méthode avancée est de pouvoir compresser la vidéo tatouée au format MPEG-1

Attaque ciblée (traitement usuel) : le type de compression utilisé ici est une compression au format MPEG-4. Le logiciel utilisé pour compresser la video tatouée au format MPEG-4 est Adobe Premiere V5.5 en utilisant le codec MPEG-4 (Table 2).

8.2.2. Filtres 3x3 (3x3 Average)

Remplace chaque pixel par la valeur moyenne de ce dernier avec les pixels voisins.

Ce filtre applique une opération de flou rapide.

En réalité ce n'est pas exactement une moyenne : la pondération est de 32/256 pour le pixel central et 28/256 pour les pixels voisins.

Ce filtre donne de bons résultats pour les grandes images (>320*240)qui contiennent du bruit ou tout autre parasite (table 3).

8.2.3. Flou (blur)

Cela consiste en une diminution de la netteté des images vidéos au moyen de filtres. Ces filtres produisent un effet de changement de mise au point des images vidéos, en éliminant le bruit aux endroits où des transitions significatives de couleurs se produisent. L'utilisation de ces filtres est très répandue surtout pour les clips de haute résolution

Certains bits constituant les labels peuvent être modifiés après compression (ceci est dû a la compensation de mouvement introduite par le format de compression MPEG) et après avoir effectué quelques attaques. Pour résoudre ce problème nous avons dû faire une détection au niveau de plusieurs images vidéos tatouées (Table 4).

8.2.4. Redimensionnement (Resize)

Réechantillonne une image afin de la rendre plus petite ou plus grande que sa taille originale en utilisant le redimensionnement bilinéaire. Ce dernier utilise une approximation linéaire dans chaque direction.

Ce filtre donne de bons résultats pour une diminution de 66% ou un agrandissement x8.

La plupart des cartes vidéos utilisent un filtrage bilinéaire au niveau du hardware : Le filtrage bilinéaire tend à produire une image légèrement floue et a comme conséquence l'apparition de formes trapézoïdales quand le facteur d'agrandissement est trop important (Table 5).

8.3. RESULTATS ET PERSPECTIVES

Une fois les attaques effectuées, nous avons remarqué que le watermark inséré est retrouvé dans la plupart des images à partir desquelles la détection a été faite. Cependant, nous avons remarqué que dans certaines images, au plus trois bits en particulier, sont erronés. Ces erreurs de détection peuvent être expliquées par le fait que lors de l'insertion des labels, certaines zones sont totalement blanches, donc la valeur de leurs coefficients était de 255 (codage sur 8 bits).

Pour résoudre ce problème, on peut envisager trois solutions possibles :

La première, simple de mise en œuvre, consiste à détecter le label au niveau de plusieurs images. Ces images peuvent être successives, mais il est préférable de les choisir espacées les

unes des autres. Une fois détecté, il ne restera plus qu'à comparer les bits du label détecté sur ces images et de prendre la décision en fonction du nombre de fois où « 1 » ou « -1 » est apparu.

La deuxième solution consiste à insérer plus de labels dans une image (au moins trois) à l'aide de clés différentes, ce qui nous permet d'augmenter le nombre de bits que l'on peut dissimuler. Ainsi, on pourra répéter un bit plusieurs fois, ce qui nous permettra de prendre une décision même si un ou plusieurs bits sont erronés. Par exemple on peut répéter le bit –1 trois fois et le coder par –1 –1 –1 ce qui nous permettra, lors de la détection, de prendre la décision en fonction du nombre de fois ou –1 est apparu (dans notre cas au moins deux fois pour constater que le bit est –1). Cependant, on ne pourra pas ajouter plusieurs labels en utilisant la propriété d'addition des watermarks car cela causera une dégradation de l'image.

La troisième solution consiste à utiliser un code correcteur d'erreur. Compte tenu du faible nombre d'erreurs nous utilisons un code correcteur d'erreurs de HAMMING (7 :4 :Hamming) qui corrige un bit sur quatre en ajoutant trois bits superviseurs. On notera que ce code correcteur est capable de détecter une erreur et de la corriger même si l'erreur se produit sur l'un des bits superviseurs. L'adoption de cette solution limitera le nombre de bits que l'on pourra insérer dans chaque image à 40 bits pour des vidéos de 240*320 pixels. Cette solution a été mise en œuvre et a donné de bons résultats ; le watermark a été retrouvé dans tous les cas.

Pour de futurs travaux, nous recommandons d'améliorer le temps d'exécution de l'algorithme. Ceci pourra se faire en intégrant des fonctions de lecture/écriture de fichiers MPEG proposées par MATLAB dans sa version 6. De plus le compilateur disponible dans cette version du logiciel permettra de compiler le programme écrit et de générer un fichier.exe indépendant de MATLAB et beaucoup plus rapide d'exécution.

Une autre voie à explorer serait de traduire le programme en C++ en utilisant des librairies MPEG si celles ci sont disponibles ou les programmer.

En ce qui concerne les expérimentations à mener, plusieurs axes sont à explorer, tels, la recherche des paramètres optimaux concernant le facteur d'amplification et le facteur d'étalement, le nombre de labels que l'on peut superposer, le nombre d'images dans lesquelles il faut insérer le même label pour que les labels ne se mélangent pas. On pourra aussi utiliser un autre code correcteur d'erreurs.

CONCLUSION

Le travail qui nous a été proposé consistait à insérer un fingerprint dans le but d'identifier des distributeurs ou des possesseurs d'une séquence vidéo. De ce fait, nous avons, dans la méthode conçue, inséré un watermark constitué de deux labels (label 1 et label 2). L'association de ces deux labels permet d'avoir un fingerprint de 40 bits utiles (Le reste des bits étant attribué au code correcteur) dans des images vidéos de 240×320 pixels.

L'insertion de la marque se fait au niveau du domaine DCT (Discrete Cosine Transform) de chaque image vidéo. Cette transformation a été choisie car elle permet d'insérer plus de bits et d'éviter une dégradation due à la compression MPEG. Les coefficients DCT retenus pour l'insertion sont ceux de basses fréquences. Aussi, si quelqu'un tente de manipuler le watermark il déformera l'image vidéo. Chaque label est inséré à l'aide d'une clé pseudo-aléatoire. L'une des particularités de ce procédé de tatouage réside dans l'estimation du mouvement au niveau des images, ce qui permet de s'assurer que les labels ne peuvent être détectés ou altérés par las attaques dites statistiques et/ou les *collusion attacks*.

La vidéo obtenue après tatouage est de qualité proche de l'original (PSNR minimum de 46 dB). Les attaques utilisées (compression MPEG-4, filtre 3×3, le flou, changement d'échelle et addition de bruit blanc) nous ont permis de tester la robustesse du procédé. L'apparition de trois bits erronés au plus nous a insité à utiliser un code correcteur d'erreurs (7 : 4 Hamming) qui corrige jusqu'à quatre bits erronés successifs grâce à un « interleaving ».

Pour de futurs travaux, nous recommandons l'amélioration du temps d'exécution de l'algorithme en le programmant en C++. Pour cela, il faudra d'abord programmer des librairies en C++ pour la lecture de fichiers vidéos. On pourra aussi trouver les paramètres optimaux pour les compressions MPEG-4, MPEG-7 et MPEG-21, c'est à dire le nombre d'images sur lesquelles il faudra répéter le watermark pour qu'il n'y ait pas mélange entre les watermarks. Pour pouvoir corriger plus de bits on pourra utiliser d'autres codes correcteurs d'erreurs en faisant attention à la complexité de l'algorithme.

Bibliographie :

[1] J. J. K. Ó Ruanaidh, W. J. Dowling, F. M. Boland, "*Watermarking digital images for copyright protection*", IPA95 Special Section in IEE Proc.-Vis. Image Process., vol. 143, No. 4, August 1996, pp. 250-256.

[2] F. A. P. Petitcolas, R. J. Anderson, M. G. Kuhn, "*Information Hiding-A Survey*", in Proceedings of the IEEE vol.87 no.7, July 1999, pp. 1062-1078.

[3] A. A. C. Kalker, "*Omnipresent and invisible digital watermarking*", in Philips Research Password 5, October 2000, pp. 10-13.

[4] F. A. P. Petitcolas, R. J. Anderson et M. G. Kuhn, "*Attacks on copyright marking systems*", in Aucsmith [5], pp. 218–238, ISBN 3-540-65386-4.

[5] D. Aucsmith, éditeur. "*Information hiding: second international workshop*", vol. 1525 de Lecture Notes in Computer Science, Portland, Oregon, U.S.A., avril 1998. Springer Verlag, Berlin, Allemagne. ISBN 3-540-65386-4.

[6] J-P. M. G. Linnartz et M. van Dijk, "*Analysis of the sensitivity attack against electronic watermarks in images*", in Aucsmith [5], pp. 258–272. ISBN 3-540-65386-4.

[7] M. Maes, "*Twin peaks: the histogram attack on fixed depth image watermarks*", in Aucsmith [5], pp. 290–305. ISBN 3-*540*-65386-4.

[8] G. C. Langelaar, R. L. Lagendijk et J. Biemond, "*Removing spatial spread spectrum watermarks by non-linear filtering*", in 9th European Signal Processing Conference (EUSIPCO'98), pp. 2281—2284, Île de Rhodes, Grèce, 8–11 septembre 1998. ISBN 960-7620-05-4.

[9] J. J. K. Ó Ruanaidh et T. Pun "*Rotation, scale and translation invariant spread spectrum digital image watermarking*", Signal Processing, vol. 66, no 3, pp. 303–317, mai 1998. ISSN 0165-1684. European Association for Signal Processing (EURASIP).

[10] M. Kutter, "*Watermarking resisting to translation, rotation, and scaling*", in Proceedings of S.P.I.E. International Symposium on Voice, Video, and Data Communications, vol. 3528, pp. 423–431, Boston, U.S.A., novembre 1998.

[11] G. B. Rhoads, "*Steganography methods employing embedded calibration data*", Digimarc Corporation. Brevet U.S.A. 5.636.292, 3 juin 1997.

[12] E. Koch et J. Zhao, "*Towards robust and hidden image copyright labeling*", in Workshop on Nonlinear Signal and Image Processing, pp. 452–455, Neos Marmaras, Grèce, 20–22 juin 1995. I.E.E.E.

[13] Signum Technologies – SureSign digital fingerprinting. www.signumtech.com, octobre 1997.

[14] L. Pitas, "*A method for signature casting on digital images*", in International Conference on Image Processing, vol. 3, pp. 215–218, septembre 1996.

[15] I. J. Cox, Joe Kilian, T. Leighton et T. Shamoon, "*A secure, robust watermark for multimedia*", in Anderson [16], pp. 183–206. ISBN 3-540-61996-8.

[16] R. J. Anderson, éditeur, "*Information hiding: first international workshop*", vol. 1174 de Lecture notes in Computer Science, Newton Institute, Cambridge, Grande Bretagne. Springer Verlag, Berlin, Allemagne, mai 1996. ISBN 3-540-61996-8.

[17] N. A. Dodgson, "*Quadratic interpolation for image resamplin*", I.E.E.E. Transactions on Image Processing, vol. 6, no 9, pp. 1322–1326, septembre 1997. ISSN 1057-7149.

[18] S. J. Godsill, P. J.W. Rayner et O. Capp.é, "*Digital audio restoration*", in Mark Kahrs et Karlheinz Brandenburg, éditeurs, App.lications of Digital Signal Processing to Audio and Electroacoustics. Kluwer Academic Publishers, 1998.

[19] F. A. P. Petitcolas et R. J. Anderson, "*Evaluation of copyright marking systems*", présenté à I.E.E.E. International Conference on Multimedia Computing & Systems, Florence, Italie, 7–11 juin 1999.

[20] D. Gruhl, W. Bender et A. Lu, "*Echo hiding*", in Anderson [16], pp. 295–315. ISBN 3-540-61996-8.

[21] B. P. Bogert, M.J.R. H. et J. W. Tukey, "*The quefrency alanysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-ceptstrum and saphe cracking*", in M. Rosenblatt, éditeur, Symposium on Time Series Analysis, pp. 209–243, New-York, U.S.A., 1963. John Wiley & Sons, Inc.

[22] S. Craver, N. Memon, B-L. Yeo et M. M. Yeung, "*Resolving rightful ownerships with invisible watermarking techniques: limitations, attacks, and implications*", I.E.E.E. Journal of Selected Areas in Communications (J-SAC) – Numéro spécial sur la protection des copyright et de la vie privée, vol. 16, no 4, pp. 573–586, mai 1998. ISSN 0733-8716.

[23] M. Cooperman et S. A. Moskowitz, "*Steganographic method and device*", Société DICE. Brevet U.S.A. 5.613.004, 18 mars 1995.

[24] A. Perrig, "*A copyright protection environment for digital images*", rapp.ort de Diploma, École Polytechnique Fédérale de Lausanne, Lausanne, Suisse, février 1997.

[25] Ross J. Anderson. *"Why cryptosystems fail.* Communications of the A.C.M., vol. 37, no 11, pp. 32–40, novembre 1994.

[26] Anonyme (zguan.bbs@bbs.ntu.edu.tw). *Learn cracking IV – another weakness of PictureMarc.* news://tw.bbs.comp.hacker. Copie disponible sur www.cl.cam.ac.uk/~fapp.2/watermarking/image_watermarking/digimarc_crack.html, août 1997.

[27] G. W. Braudaway, "*Results of attacks on a claimed robust digital image watermark*", in van Renesse [28]. ISBN 0-8194-2556-7.

[28] R. L. van Renesse , éditeur, "*Optical Security and Counterfeit Deterrence Techniques II*", vol. 3314, San Jose, Californie, U.S.A., 28–30 janvier 1998. La Société pour les sciences et techniques de l'image (I.S.&T.) et la Société internationale d'ingénierie optique (S.P.I.E.). ISSN 0277-786X. ISBN 0-8194-2556-7.

[29] M. Kutter et F. A. P. Petitcolas, "*A fair benchmark for image watermarking systems*", in proceedings of Electronic Imaging '99, Security and Watermarking of Multimedia Contents, vol. 3657, pp. 226–239, San Jose, Californie, U.S.A., 25–27 janvier 1999. La Société internationale d'ingénierie optique (S.P.I.E.).

[30] J-F. Delaigle, "*Common functional model*", compte rendu AC019-UCL-TEL-DR-P-D12-b1, CEC, 29 mars 1996. Projet Tracing author's rights by labelling image services and monitoring access.

[31] S. Winkler, M. KUTTER, "*Vers un tatouage à étalement de spectre optimal utilisant le système visuel humain*", Laboratoire de traitement des signaux, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Suisse.

[32] T. Sikora, "*MPEG-1 and MPEG-2 Digital Video Coding Standards", http://wwwam.hhi.de/mpeg-video/papers/sikora/mpeg1_2/mpeg1_2.htm* , Heinrich-Hertz-Intitut Berlin - Image Processing Department.

[33] J. Brassil, S. Low, N. Maxemchuk, L. O'Gorman, "*Electronic marking and identifiationtechniques to discourage document copying*", Proceedings of INFOCOM 94, 1994.

[34] C. Kurak, J. Mac Hugh, "*A cautionary note on image downgrading*", Proceedings 8th Annual Computer Security App.lication Conference, San Antonio, 1992.

[35] S. Walton, "*Image authentication for slippery new age*", Dr J. Dobb's; 1995, pp. 18, 26,82,87.

[36] C. Dautzenberg, F. M. Boland, "*Watermarking Images*", Technical Report, Department of Electronic and Electrical engineering, trinity College, 1994.

[37] J. Zhao, E. Koch,"*Embadding robust label into images for copyright protection*", Technical report, Fraunhofer Institute for Computer Graphics, Darmsdadt, Allemagne, 1994.

[38] K. Matsui, K. Tanaka, "*Video steganography : how to secretly embed a signature in a picture*", IMA intellectual Property Project Proceedings, Janvier 1994, pp. 187-206.

[39] L. Cox, J. Killian, T. Leighton, T. Shamoon, "*Secure Spread spectrum Communication for Multimedia*", technical report, NEC Research Institute, 1995.

[40] J. J. K. O Ruanaldh, W. J. Dowling, F. M. Boland, "*Phase watermarking of images*", IEEE International Conference on Image Processing, Lausanne, Suisse, Septembre, 1997.

[41] F. Hartung, B. Girod,"*Digital watermarking of MPEG 2 coded video in the bit stream domain*", Proceeding of Int Con. On Acoustic, Speech and signal processing, vol. 4, pp. 2621-2624, avril 1997.

[42] R. L. Pickholtz, D. L. Schilling, L. B. Milstein, "*Theory of Spread spectrum commuications- a tutorial*", IEEE trans, 1982, COM-30, pp. 855-884.

[43] J.P.M.G. Linaarz, J. C. Talstar, "*MPEG PTY-Marks: Cheap detection of embedded copyright dat in DVD-video*", Proceedings of ESORICS'98 5th European symposium on Research In Computer Security, Belgique, septembre 1998.

[44] Min Wu, Hong Heather Yu, Alex Gelman, "*Multi-level Data Hiding for Digital Image and Video*", Electrical Engineering Dept., Princeton Univ., Princeton, NJ 08544, Panasonic Information & Networking Technologies Laboratory, Princeton, NJ 08540.

[45] I. J. Cox, J. Killian, T. Leighton, and T. Shamoon, "*A secure Robust watermark for Multimedia*", IEEE Trans. Image Processing, vol. 6. no. 12, pp. 1673-1687, decembre 1997.

[46] M. D. Swanson, B. Zhu and A. H. Tewfik, "*Data Hiding for Video-in-Video*", Proceedings of IEEE International Conference Image Processing (ICIP '97), vol. 2, pp. 676-679, Santa Barbara, California, octobre, 1997.

[47] D. Mukherjee, J. J. Chae and S. K. Mitra, "*A Source and Channel Coding App.roach to Data Hiding with App.lication to Hid-ing Speech in Video*", Proceeding of IEEE ICIP 98, vol. 1, pp.348-352, Chicago, octobre, 1998.

[48] J. J. Chae and B. S. Manjunath, "*A Technique for Image Data Hiding and Reconstruction without Host Image*", to app.ear in the Proceeding of SPIEEI '99, Security and Watermarking of Multimedia Contents, San Jose, California, janvier, 1999.

[49] J. J. Chae, D. Mukherjee and B. S. Manjunath, "*Color Image Embedding using Multidimensional Lattice Structures*", Proceedings of IEEE International Conference of Image Processing (ICIP'98), vol. 1, pp. 460-464, Chicago, Illinois, Octobre, 1998.

[50] D. Mukherjee, J. J. Chae, S. K. Mitra, "*A Source and Channel Coding Approach to Data Hiding with Application to Hiding Speech in Video*", Proceeding of IEEE ICIP `98, Vol. 1, pp. 348-352, Chicago, October, 1998.

[51] J. J. Chae, B. S. Manjuntah, "*Data Hiding in Video*", Proceedings of 6th IEEE International Conference on Image Processing (ICIP'99), Kobe, Japan, 24-28 Oct. 1999. p.311-15 vol.1.

[52] D. Kim, S. Park, "*A Robust Video Watermarking Method*", IEEE Trans. Image Processing, vol. 2, pp. 763-766 , 2000.

[53] M. Wu, H. H. Yu, A. Gelman, "*Multi-level Hiding for Digital image and Video*", Photonics East' 99- Multimedia Sys. and Application, SPIE vol. 3845, 1999.

[54] F.Hartung, B. Girod, "*Copyright Protection in Video Delivery Networks by Watermarking of Pre-Compressed Video*", Multimedia Applications, Services and Techniques- ECMAST'97, Springer Lecture Notes in Computer Science, vol. 1242, pp. 423-436, Springer, Heildelberg, 1997.

[55] M. Ramkumar, A. N. Akansu, "*Information theoretic Bounds for Data Hiding in Compressed Images*", presented in the 1998 Workshop on Multimedia Signal Processing (MMSP-98), Los Angeles, CA, USA, december 7-9 1998.

[56] A. Papoulis, "*Probability, Random Variables, and Stochastic Process*", 3rd Edition, McGraw Hill Inc. 1991.

[57] T. M. Cover, J. A Thomas, "*Elements of information theory*", Second edition, Jhon-Wiley and Sons Inc, 1991

[58] A. N. Akansu, J.A. Haddad, "*Multiresolution signal Decomposition: Transform, Subbands and wavelets*", Academic Press Inc. 1992.

[59] Papoulis, "*probability, random variables and stochastic processes*", Mc Graw Hill, 1984, 2nd Edition.

Maritime Research Laboratory, Cambidge, England, 1989.

[60] R. G. Gallager, "*Information Theory and reliable communication*", Wiley, 1968.