Université de Nice - Sophia Antipolis
Licence d'Informatique
Travail d'études

LE TATOUAGE D'IMAGES OÙ "WATERMARKING"

Julien PUGLIESI - Cedric PIOVANO
Encadrement: Pierre Crescenzo

Juin 2004

Table des matières

1 Introduction **3**
1.1 Watermarking, stéganographie, cryptographie? 3
1.1.1 Qu'est-ce que le tatouage d'images? 3
1.1.2 Différences avec la cryptographie 3
1.1.3 A quoi ça sert? 3
1.2 Une Part d'Histoire !!!! 4
1.3 Notion de Base 5
1.3.1 Differentes definitions 5
1.3.2 Transformée de Fourrier 6
1.3.3 Transformée et domaine DCT 6
Difference entre le FFT et le DCT 7
1.4 Un peu de terminologie 7
1.5 Les degrés de tatouages et différentes formes 7
1.5.1 Visibilité 8
1.5.2 Robustesse et fragilité 9
Un cas intermediaire : Les Semi-fragiles 10
1.5.3 Un troisième critére : Le Ratio 11

2 Algorithme **12**
2.1 Domaine spatial 12
2.1.1 Bits de poids faibles 12
Probleme du gif 13
2.1.2 Algorithme "Patchwork" 13
2.1.3 Autre approches 14
2.2 Domaine Frequentiel 15
2.2.1 Algorithme de Koch et Zhao 15
2.2.2 Etalement de Spectre (Spread-spectrum) 16
2.2.3 Les Fractales 17
Compression Fractale 18
Algorithme 18
2.3 Des images à la vidéo 20

3 Les Attaques **21**
3.1 Qu est ce qu une attaques? 21
3.2 Comment ça fonctionneă? 21
3.3 Les différentes attaques 21
3.3.1 Les attaques Basique Involontaire 22
Les transformations géométriques 22

Les transformations frequentielle 23
Bruitage et Filtrage 23
Les Compressions 25
3.3.2 Les attaques volontaires 26
3.3.3 Les attaques de nature cryptologique 28
Exemple d'attaque sur le copyright 28
Exemple d'attaque sur la protection de copie 28
3.4 Les Contre-attaques 29
3.4.1 La transformée de Fourier-Mellin 29
3.4.2 Les contres-attaque "Non-Blind" 30
3.4.3 Les contres-attaque "Blind" 30
3.4.4 Les autres outils a l'étude 31

4 Conclusion **33**

Chapitre 1

Introduction

1.1 Watermarking, stéganographie, cryptographie?

1.1.1 Qu'est-ce que le tatouage d'images?

Le tatouage d'images est une technique qui est en fait issue directement d'un art appelé la stéganographie. Cet art n'a pour ainsi dire qu'un but précis, qui est de cacher au sein d'un message primaire, un message secondaire. Bien entendu il faut que le message primaire soit lisible par tout un chacun, et qu'il reste visuellement inchangé par rapport à ce qu'il était avant introduction du message secondaire. Le message secondaire se doit d'être parfaitement invisible, mais uniquement accessible par des personnes propriétaires d'une information secrète, une *"clef"* par exemple qui permettrait son extraction.

1.1.2 Différences avec la cryptographie

En cryptographie, l'objectif n'est pas de dissimuler des informations dans d'autres, mais plus simplement de rendre l'information que l'ont désire transmettre complètement illisible à toute personne ne possédant pas la donnée nécessaire a son décodage. De plus en cryptographie si le message primaire est modifié, il devrait être impossible de le recouvrer, tandis qu'en stéganographie, le message secondaire est supposé rester accessible et ce même après de multiples recopies et manipulations diverses du message primaire.

1.1.3 A quoi ça sert?

Et bien comme tout le monde le sait, de nos jours presque tout les types de médias (images, sons, vidéos, etc.) sont stockés sous forme de données numériques, et leur libre accès pose de nombreux problèmes de droits d'auteur. Cela vient principalement du fait de la banalisation des connexions internet haut débit, et des graveurs de CD/DVD qui représente un manque à gagner et des préjudices importants pour les grandes industries de médias.
Le *"watermarking"* est certainement un moyen efficace de résoudre ces problèmes.
C'est la raison pour laquelle beaucoup se tournent vers cette technologie récente et sophistiquée.

Par exemple, la marque ajoutée pourrait être alors un simple copyright ou un numéro de licence. L'avantage ici se situe dans le fait qu'il serait non seulement **invisible** pour l'observateur, mais de plus **indélébile** et **robuste** face aux traitements classiques appliqués aux images comme le fenêtrage, le lissage, les transformations géométriques ou la compression avec pertes.

1.2 Une Part d'Histoire !!!!

L'apparition de la stéganographie est trés ancienne, elle remonte à l'antiquité. En effet, les premiers exemples connus nous viennent directement des Grecs. Ils rasaient les cheveux d'un esclave, puis tatouaient sur son crâne un message. Une fois les cheveux repoussés, l'esclave pouvait traverser les territoires ennemis sans éveiller les soupçons. Une fois à destination, il suffisait de raser à nouveau le crâne pour récupérer le message. Bien sûr, il ne fallait pas être pressé...
Au cours de l'histoire, les techniques ont évoluées sans cesse, et on a vu au fur et à mesure du temps la naissance de nouveau procédés plus efficaces. Par exemple les encres sympatiques, qui fut la méthode la plus utilisée au cours des siècles. On écrit, au milieu des textes écrits à l'encre, un message à l'aide de jus de citron, de lait ou de certains produits chimiques. Il est invisible à l'oeil, mais une simple flamme, ou un bain dans un réactif chimique, révèle le message (figure 1.1).

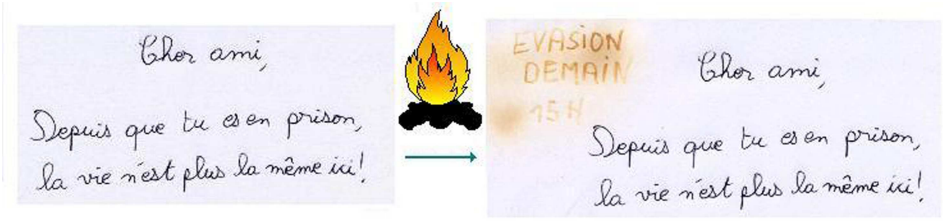

FIG. 1.1 – *Exemple de steganographie realisé à l'aide de lait*

Il a donc fallu attendre jusqu'en 1992 pour voir les premières apparitions commerciales du Watermarking, ou du moins ce fut la première année ou des articles commençaient à paraître sur le sujet (voir tableau 1.1) . En effet, il semblerait que l'ont eu recours aux tatouages de certains documents bien avant cette date. On raconte qu'en 1986, Margaret Thatcher, ne supportant plus que certains de ses ministres vendent des informations à la presse, exigea que tous les traitements de textes de son cabinet soient programmés afin que l'identité des utilisateurs soit encodée dans les espaces de leurs textes. Si une fuite advenait, on pouvait alors identifier le coupable.

Année	1992	1993	1994	1995	1996	1997	1998
Publications	2	2	4	13	29	64	103

TAB. 1.1 – *Nombre de publications de 1992 a 1998 (source INSPEC, janvier 1999).*

Bien entendu, la véritable explosion du watermarking s'est produite avec l'explosion des connexions Internet, et du libre échange des fichiers. De ce fait, certaines majors américaines l'utilisent, non pas pour les DVD grands publics, mais pour les screeners qui sont diffusés aux journalistes avant la sortie d'un film. S'ils revendent le DVD, ou sŠils le diffusent en DIVX, ils courront alors le risque que leurs "empreintes digitales" inscrites en filigrane ne les démasque.
Le principal problème dans le développement du "watermarking" est donc sa robustesse face aux attaques (compression du média, filtrage, etc.), or il y a à peine cinq ans, il était loin de satisfaire ces dures exigences, mais les progrès récents effectués utilisant des mathématiques de haut niveau permet une lutte équilibrée avec les pirates.
Maintenant, de nombreux industriels et sociétés de droits d'auteur se montrent confiants avec ce procédé. La JASRAC et la RIAJ ont par exemple reproduit des tests en condition réels utilisant différentes technologies de tatouage. Pour cela, ils ont incrusté en watermark le code standard international d'une oeuvre musicale (ISWC) dans un fichier audio qu'ils ont ensuite convertie au format MP3, et envoyé sur des serveurs Internet. En combinant ceci avec un système de surveillance, conçu pour dépister les fichiers musicaux illicites et équipés d'un programme de détection de watermark, les tests ont été concluants. La société Verance, de son côté, a su convaincre Universal Picture d'équiper dès 2004, la majorité de ses DVD, HSH, etc., de son nouveau système de watermarking vidéo.
En bref, cette science si on peut la nommer ainsi est en pleine explosion actuellement, et cela risque de continuer longtemps, du moins tant que des données pourront être copiées et échangées impunément.

1.3 Notion de Base

Ce travail d'étude étant particulièrement orienté vers l'imagerie, il nous semble nécessaire d'introduire certaines notions basiques que nous emploierons dans ce rapport.

1.3.1 Differentes definitions

Notion de pixel, valeur : un pixel, (picture element) est l élément indivisible permettant de coder l information relative a la luminosité en une certaine position pour les images en teinte de gris. sa valeur correspond alors a un nombre binaire codé généralement sur 8 bits, de 0 a 255 (du plus foncé au plus clair). pour les images colorées, le cas le plus courant est un codage de 8 bits pour chaque intensité lumineuse R G B (rouge, vert, bleu) ce qui correspond à un codage sûr 24 bits. La valeur de chaque pixel correspond, par conséquent, à un nombre entier, souvent representé en hexadécimal pour les images colorer (de 000000 a FFFFFF). Une représentation isomorphe est la représentation (Y,U,V) où Y désigne la luminance du pixel et U et V définissent sa chrominance. Plus la luminance est forte, plus le pixel est clair.

Domaine spatial : Le domaine spatial est le domaine classique où chaque valeur en (x,y) correspond à la valeur des pixels ; nous pouvons alors la visualiser dans un espace a 3 dimensions ou les axes X et Y representent

les deux dimensions de l'image, et l'axe Z represente la valeur des pixels (figure 1.2).

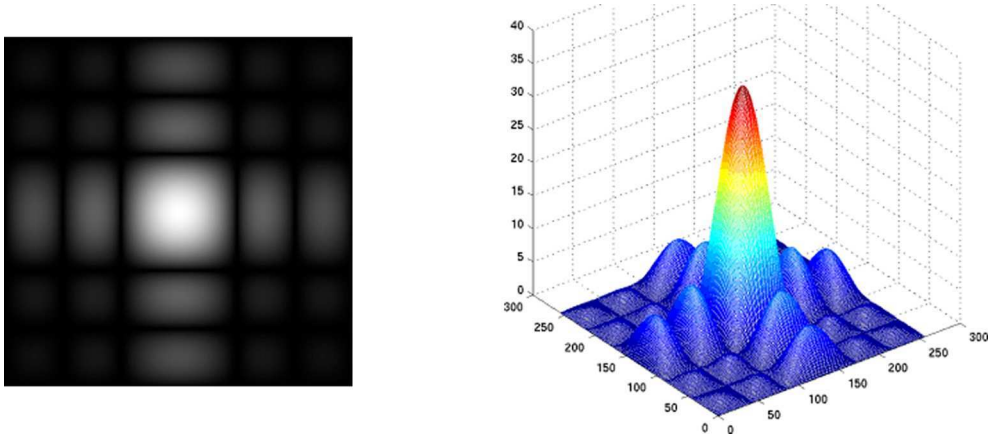

FIG. 1.2 – *Exemple de representation tridimensionelle dans le domaine spatial*

Domaine fréquentiel : Le domaine fréquentiel est un espace dans lequel l'image sera considere comme une somme de fréquences de differentes amplitudes (voir domaine DCT).

Filtre de Convolution : Sans entrer dans les details, une convolution sur des blocs de 3*3 permet de modifier le pixel courant par differentes operations sur les valeurs des pixels du voisinnage (par exemple une matrice rempli de 1 changera le pixel courant par la moyenne des 8 autre pixels du bloc 3*3).
Ces filtres de convolution s'appliquent dans le domaine spatial. En fréquentiel, cela se resume a multiplier 2 fonctions.

1.3.2 Transformée de Fourrier

Elle permet simplement de passer du domaine spatial au domaine frequentiel. Cette transformée rend donc visible les composante en fréquence de l'image

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) * e^{-1\omega * t}.dt \tag{1.1}$$

Il est interessant de noter que nous pouvons revenir au domaine spatial via la transformée de fourrier inverse. Une application brute de cette formule etant extrement longue, Une autre façon d'effectuer ce calcul permet de limiter considerablement la duree de cette transformation. C'est ce que l'on appelle la FFT (Fast Fourier Transform).

1.3.3 Transformée et domaine DCT

La DCT (Discret Cosinus Transform) est une transformée fort semblable à la FFT, travaillant sur un signal discret. Elle prend un ensemble de points d'un domaine spatial et les transforme en une représentation équivalente dans le

domaine fréquentiel. La DCT transforme un signal d'amplitude (chaque valeur du signal représente l' "amplitude" (voir domaine fréquentiel) d'un phénomène) discret bidimensionnel en une information bidimensionnelle de "fréquences".
la formule de la DCT est detaillée ci-dessous.

$$F(u,v) = \frac{2}{N}c(u).c(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} Img(x,y).\cos\left[\frac{\pi}{N}u(x+\frac{1}{2})\right].\cos\left[\frac{\pi}{N}v(y+\frac{1}{2})\right] \quad (1.2)$$

Voici son inverse (connue aussi sous le nom de IDCT).

$$Img(x,y) = \frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} c(u).c(v).F(u,v).\cos\left[\frac{\pi}{N}u(x+\frac{1}{2})\right].\cos\left[\frac{\pi}{N}v(y+\frac{1}{2})\right] \quad (1.3)$$

$$\text{ou} \begin{cases} c(0) = (2)^{-\frac{1}{2}} \\ c(w) = 1 \qquad \text{pour } w = 1,2,\ldots,N-1 \end{cases} \quad (1.4)$$

Cette transformation étant très lourde, elle s'applique généralement en bloc de 8x8 (compression Jpeg). Concrètement, et en terme simple, cette transformation va essaye de faire correspondre des blocs de 8x8 de l'image en une somme de fonction basique qui sont donnée dans la matrice 8x8 de la DCT (DCT matrix). Un exemple est donné dans la figure 1.3. Les valeurs de la matrice de la transformée corresponde par conséquent à l'intensité lumineuse pour chaque fonction de la matrice.

Difference entre le FFT et le DCT le DCT est actuellement une version simplifier de la FFT.

- Seule la partie reelle de la FFT est conservé
- Beaucoup plus simple en terme de coup de programmation
- la DCT est efficace dans la compression de multimedia (Jpeg)
- DCT *Beaucoup* plus utilisée.

1.4 Un peu de terminologie

Pour nommer le tatouage d'images dans le monde, on utilise généralement le mot anglais *"Watermarking"*, ou plus exactement *"digital watermarking"*. Ce terme anglo-saxon, signifiant *"filigramme"* à la base, correspond dorenavant au fait même de masquer des données sur un autre support. Cela regroupe par conséquent des notions plus précises et restrictives de marques, invisibles et robustes, appliquées au service de protection des droits d'auteurs. Ce qui nous amènes donc à employer une expression associant davantage cette idée d'enfouissement de données, le *"data embedding"*.

1.5 Les degrés de tatouages et différentes formes

Plusieurs formes et degrés de tatouages éxistent. Ils sont généralement répertorié par leurs degrés de priorités :

- visibles ou non visibles

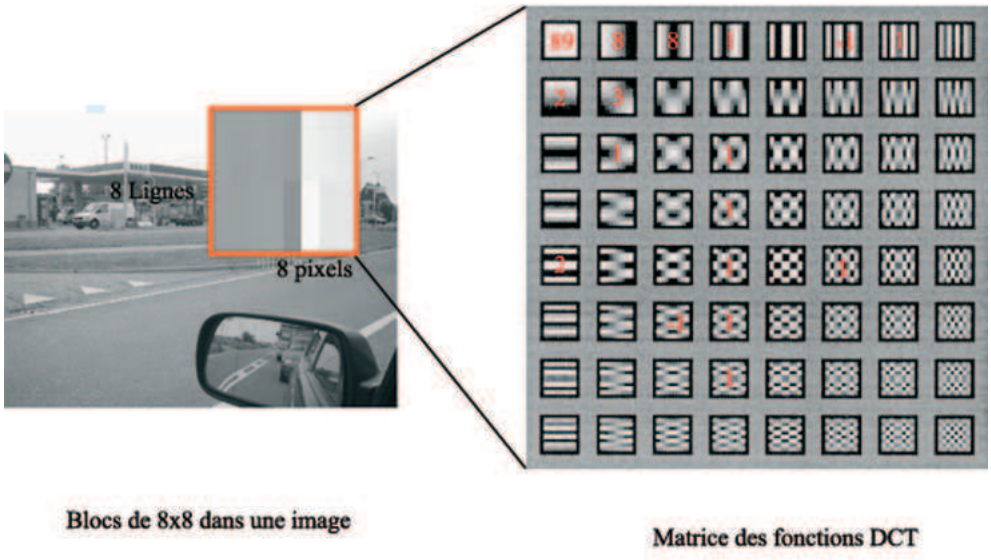

FIG. 1.3 – *Exemple concret de transformation en domaine DCT*

– robustes ou fragiles

1.5.1 Visibilité

Le principe fondamental du tatouage visible (figure 1.4) consiste à dissimuler partiellement une image. Pour ce faire il faudra utiliser un nombre indéterminé de marques visibles, qui ne pourront être efficacement effacées que si l'on possède une *"clef secrète"* adéquate. Ce type de tatouage est étudié actuellement pour gérer tout ce qui concerne les contrôles d'accès d'un unique document, correspondant en quelque sorte a une distribution de permission. On entend par contrôles d'accès la possibilité de restreindre la divulgation d'un document en fonction de l'appartenance d'un utilisateur ou non la classe des *"ayants droit"* à la lecture de ce document.

En pratique, l'interet d'un watermark éfficace reside dans son invisibilité (figure 1.5). Et c'est d'ailleurs l'un des trois principaux criteres d'un algorithme de marquage : faire en sorte que la difference avec l'original soit la plus minime possible.
Plusieur methodes tentent de renforcer cette invisibilité (voir section contre-attaque).

Image utilisé en tant que watermark

Image originale

Image watermarkée (coin superieur gauche)

FIG. 1.4 – *Exemple de tatouage visible*

1.5.2 Robustesse et fragilité

Le deuxieme principal critere de qualite d'un algorithme de tatouage concerne sa robustesse face a des manipulations de l'image : celui-ci doit pouvoir conserver l'information stoquer dans le marquage en depis de diverse transformation.
Or tres peux d algorithme resiste a une simple compression ou un changement de format, mais ils sont sensés resister tout de même à des attaques basiques telle que des translations ou des rotations de l'image (souvent utilisé pour la mise en page).

Il est neanmoins interessant de remarquer qu'il peux être utile, dans certain cas, de favoriser une fragilité plutot qu une robustesse(figure 1.6).
Pour s'assurer par exemple de l'integrité d'un document, le fait de le watermarquer avec un algorithme fragile permettra, par la suite de verifier si l'information marquée est toujours presente, ce qui sous entends donc qu'elle n'a subit aucune modification malveillante (par exemple, une modification brutale de certaine partie textuelle). Cela permet donc une certaine falsification de l'image.

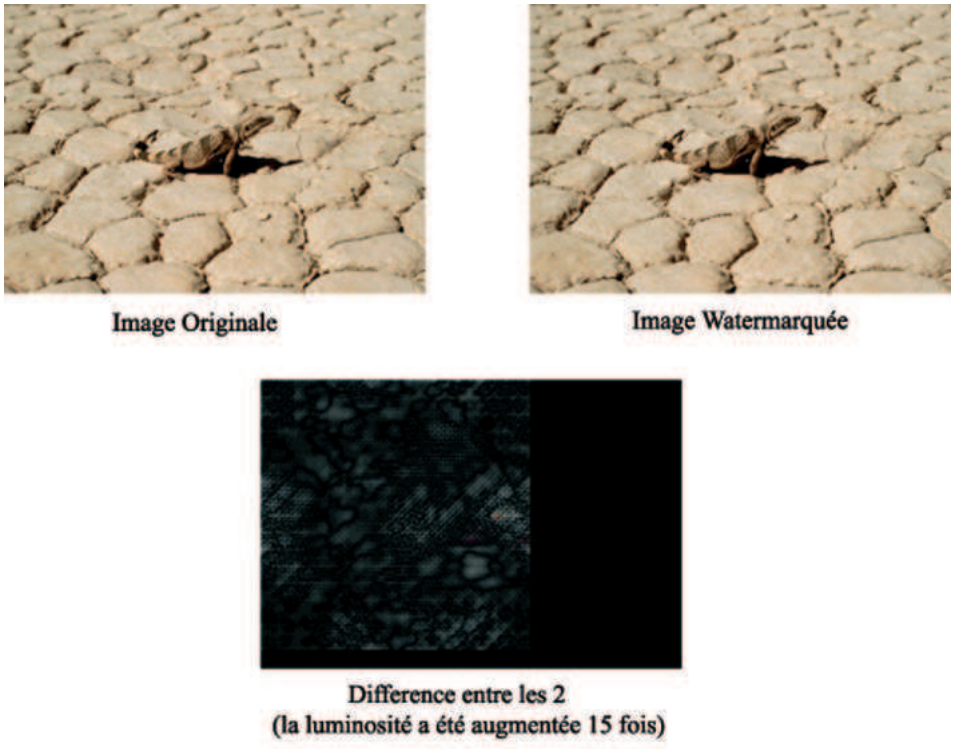

FIG. 1.5 – *Exemple de tatouage (pseudo) invisible*

Ce critère de robustesse et de fragilité s'applique surtout à un marquage invisible (il n'y a aucun interet a se poser ce genre de question pour un tatouage visible).

Un cas intermediaire: Les Semi-fragiles

Les watermarks semi-fragiles combinent à la fois les propriétés des marquage robuste et fragile. Comme les robuste, ils toleres certains changements de l'image, comme des rotations, translations ou addition de bruit. Et comme les watermark fragile, ils sont capable de determiner les regions ou l'image a été brutalement modifié et celles ou elle reste authentique.
Par consequent les watermarks semi fragile arrivent a differencier les changement "leger" comme l'ajoue d'un bruit a des changement "destructeur".
Ces algorithme sont surtout tres utile par exemple dans le cas ou une image marquées doit etre diffuser sur le net, ou des types de compression comme le JPEG vont etre employer. Un algorithme fragile ne supporterait pas ce type de transformation, et un robuste permettrait a quiconque recupererait cette image sur le net d'en modifier des parties sans que le marquage (donc le pseudo copyright servant a la falsification d'un document) soit altéré : tout le monde peux se presenter comme ayant un document original en sa possession. L'exemple d'un tatouage fragile (figure 1.6) permet d'illustrer aussi ce cas.

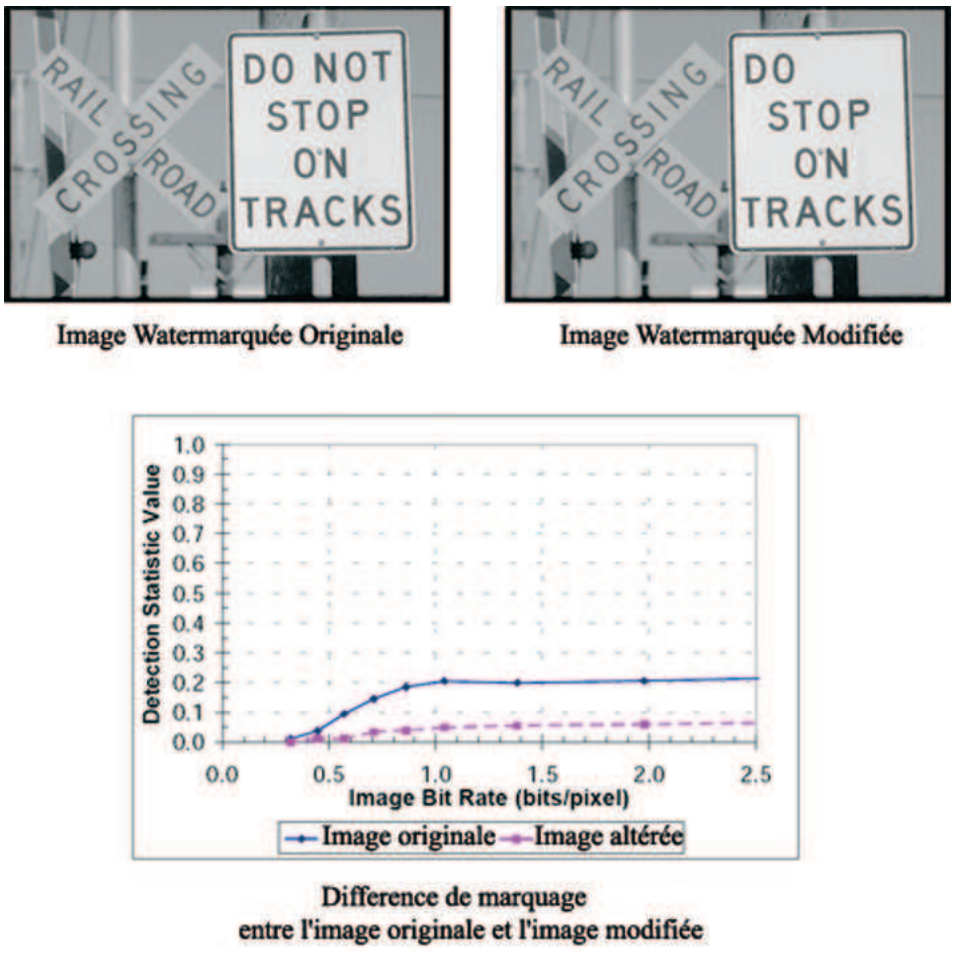

FIG. 1.6 – *Exemple de tatouage fragile*

1.5.3 Un troisième critére : Le Ratio

Celui-ci constitue la quantité d'information que l'on peux rentrer dans une image.
En pratique, de 16 a 64 bit suffisent pour assurer le systeme de copyright. Des donnée brute volumineuse sont rarement caché en tant que watermark (signifiant "filigramme" comme cité plus haut, donc quelque chose de relativement discret), mais plutot des information suffisament detaillées pour permettre recuperer une donnée via un autre moyen (tel des pointeur vers une donnée exterieur a l'image). Il existe neanmoins certaine image qui s'auto-suffise, ayant l'information brute stoquée en tant que marquage.
Ce genre de marquage s'applique surtout plus a une video qu'une image fixe.

Chapitre 2

Algorithme

Ce chapitre n'a pas pour but de proceder a une simple liste des differents algorithme de tatouage, mais plutot de presenter successivement plusieur methode permettant d'avoir une idée concrete sur les notion de base d'un algorithme de tatouage.
Nous traiterons par consequent en priorité des algorithmes relativement basique plutot que les principaux utilisé actuellement.
Ceux-ci nous permettront de mettre en avant les differentes contradiction et limite rencontré par chacun d'eux, et d'introduire des methode pouvant potentiellement combler ces defaut inherant au traitement de l'image.
Ces algorithme se distingue essentiellement par quatre points :

- La facon de selectionner les differents points de l'image originale (ou blocs) qui contiendront les données du watermark.
- la manière de faire correspondre l'image hote avec l'information a enfouir (relation binaire entre les bits par exemple). C'est ce que l'on appele la modulation.
- Le pre-traitement de l'information avant son enfouissement : pre-formatage (voir section contre-attaque) ou encore redondance de l'information.
- Le choix du domaine de travail : spatial ou frequentiel (DCT).

Concretement, les algorithme appartiennent a deux grande famille : ceux operant sur le domaine spatial, et ceux sur le domaine DCT.
Nous analyseront d'une part des exemple de type d'algorithme travaillant sur le domaine spatial, en notant nottament leur principaux avantages et defauts, puis nous detailleront des algorithme operant sur le domaine DCT.

2.1 Domaine spatial

2.1.1 Bits de poids faibles

Il s'agit certainement de la méthode la plus basique du "data embedding". En reprenant la définition de la valeur d'un pixel nous savons donc que pour les images en teinte de gris cette valeur varie de 1 à 255 correspondants a différents niveaux de gris (0 étant le Noir 255 le Blanc). Chaque pixel est donc codé

sur 8 bits. Si nous considérons le fait qu il est imperceptible pour l oeil humain un changement une variation d une unité de gris, nous pouvons raisonnablement considérer que le dernier bit (bit de poids faible) n'est pas important, donc que nous pouvons le changer à notre guise.
C est ce que nous faisons pour cacher par exemple une image binaire (noir et blanc) dans une image en nuance de gris, en ne reprenant simplement que le dernier bit de chaque pixel. Pour les images en couleurs, il suffit de travailler sur la luminance.
Cette méthode ne presente néanmoins aucun des critères abordés précédemmentă:

robustesse : Il est très simple d'enlever ce marquage en mettant par exemple à 0 tous les bits de poids faible. De plus, tous les types de transformations fréquentielles, tels des filtres, sont radicaux pour ce marquage. Entre autres la compression JPEG ne lui laisse quasiment aucune chance.

Visibilité : Contrairement, a ce que l on peut penser, l oeil humain est très sensible aux contrastes dans les gris de faibles intensités et beaucoup moins dans les teintes proche du blanc. Ainsi, certaines méthodes profitent de cela en adaptant le nombre de bits de poids faible à coder en fonction de la teinte en cours et de la teinte adjacente (tout en se référant à des données physiologiques sur les couleurs).

Probleme du gif De plus, cette méthode dépend réellement du format de l'image. Par exemple pour les GIF, ou les valeurs des pixels correspondent non pas a des intensités de couleur, mais a des références dans une palette de 256 couleurs, incrémenter de 1 cette valeur peux entraîner certains changements aberrant dans l'image. Un exemple concret serait le fait de modifier la valeur 1111 1110 (254) correspondant au bleu foncé, en 1111 1111 (255) correspondant au rouge vif.
Pour remédier a ce problème une solution basique consisterait se limiter sur une palette de 128 et à créer une palette ou les couleurs marchent par pairsă:

Couleur	Rouge	Vert	Bleu
00	150	12	59
01	152	10	50
10	50	200	98
11	50	212	95

Ainsi, en ne changeant que le bit de poids faible des pixels, on reste quasiment sur la même couleur.

2.1.2 Algorithme "Patchwork"

Pour renforcer un peu plus la robustesse de la méthode précédente, une idée basique, proposée par Bender & al en 1995, consiste à répéter le même bit un grand nombre de fois pour qu'une étude statistique nous donne le bit marqué.
Toujours dans le domaine spatial, cette amélioration reste néanmoins relativement faibleă: il est très facile de vérifier qu'une image est marquée. En effet, bien que faisant partie des marquages "invisiblesă", une étude statistique des

bits de poids faible nous renseigne sur l'existence du watermark.
Voyons à présent les étapes constituant cet algorithmeă:

1. Selectionner grace a une clé generé aleatoirement des sequences de n paires de *pixel*.
2. Modifier la luminance de chaque paire (p_i, q_i) en (p'_i, q'_i) de cette facon

$$\begin{cases} p'_i = p_i + 1 \\ q'_i = q_i + 1 \end{cases} \tag{2.1}$$

TAB. 2.1 – *Algorithme d'insertion "Patchwork"*

1. Recuperer d'une part toute les n paires grace à la clé secrete.
2. Calculer S,

$$S = \sum_{i=1}^{n} (p'_i - q'_i) \tag{2.2}$$

TAB. 2.2 – *Algorithme d'extraction "Patchwork"*

Pour n suffisament grand, l'equation suivante est verifié :

$$\sum_{i=1}^{n} (p_i - q_i) = 0, \tag{2.3}$$

Seul un utilisateur possedant la clé secrete obtiendra un score S different de 0. La clé permet ici par consequent la localisation de zones secrètes ou la donnée sera cachée.

2.1.3 Autre approches

Il existe bien d'autre approche dans le domaine spatial, entre autres les travaux de Kutter, Jordan et Bosser, qui consiste moduler l'information dans la composante bleue à différents endroits de l'image, créant une pseudo modification proportionnelle à la luminance, ou encore le fait de découper l'image en différents blocs de taille variable, où l'information est stoquée dans la moyenne des valeurs des pixels de ces blocs.
Néanmoins, les marquages dans le domaine spatial résistent très mal à tout type d attaque, géométrique ou fréquentiel. Le simple fait d'appliquer une rotation corrompt la plupart du temps le watermark.
Voyons à présent des algorithmes dans le domaine fréquentiel.

2.2 Domaine Frequentiel

Constatant la mauvaise performance des algorithmes de marquage dans le domaine spatial vis-à-vis de certaine modification, la plupart du temps étant involontaire comme entre autres de la compression JPEG, de nombreuses méthodes ont été développées à partir de connaissance acquise en traitement de signal.
Une bonne partie de ces méthodes travaille sur le domaine DCT, espérant surtout renforcer la robustesse du marquage sur les compressions utilisant ce type de domaine. Le plus connu étant le jpeg pour ce qui concerne les images fixes, mais également le Mpeg pour la vidéo.

2.2.1 Algorithme de Koch et Zhao

Une approche consisterait a extraire un certain nombre de carré de 8x8 pixels de l'image, de calculer la transformée DCT de ces blocs et d'aller marquer un bit sur les moyennes fréquences correspondantes, sachant que la modification des basses fréquences de l'image la changerait trop, les basses fréquences correspondant aux zones homogènes les plus grandes sur l'image, par exemple un noir uniforme dans les zones sombres, et que les hautes fréquences sont enlevées par la compression JPEG, correspondant aux zones homogènes les plus petites d'une image, à savoir les details au niveau de chaque pixel. Voici une description formelle des algorithmes d'insertion et d'extraction de cette méthode.

1. Soit une séquence de k bits $(b_1, \ldots, b_k)$ à cacher dans l'image
2. Sélectionner dans l'image (selon une clé secrète) k blocs B $(B_1, \ldots, B_k)$ de taille 8x8.
3. Calculer les coefficients DCT $(a_{11}, \ldots a_{88})$ de chaque bloc sélectionné, si nécessaire.
4. Pour i allant de 1 à k :
   Soient (a_{kl}) et (a_{mn}) deux des coefficients DCT du bloc B_i, et b_i le bit à cacher
   - Si $\{(b_i = 1)$ et $(a_{kl})_i > (a_{mn})_i\}$ ou $\{(b_i = 0)$ et $(a_{kl})_i < (a_{mn})_i\}$, alors ne rien faire.
   - Sinon modifier les valeurs de $(a_{kl})_i$ et $(a_{mn})_i$ pour que la relation précédente soit vérifiée.
5. Calculer la DCT inverse à partir des valeurs ainsi modifiées afin d'obtenir l'image marquée, et revenir dans le domaine spatial, si besoin est.

TAB. 2.3 – *Algorithme d'insertion Koch & Zhao*

Bien qu'étant nettement plus robuste a des attaques involontaires de type fréquentiel, cet algorithme presente néanmoins quelques inconvénientsă:

- Le fait de cacher l'information dans des blocs permet au mieux de stoquer un bit dans ces blocs, donc limite le ratio.

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

TAB. 2.4 – *Algorithme d'extraction Koch & Zhao*

- Le fait même d'utilisé des blocs a toujours cet inconvénient d'être vite mis en difficulté face a des attaques géométrique. Le simple fait d'appliquer une rotation modifie le maillage d'origine de l'image, donc la segmentation en bloc de 8*8 ne correspond plus du tout a l'originale.
- Enfin le terme de *"moyennes fréquences"* est particulièrement dur a définir, ce qui entraîne assez vite un conflit visibilité/robustesseă: si l'on choisit des fréquence relativement basse, le tatouage est certes plus robuste, mais devient visible, parallèlement plus nous prenons des fréquences hautes plus le marquage se fond dans l'image, mais perd en robustesse (même face à une compression jpeg).

Cet algorithme a subi de nombreuses modifications pour essayer de palier à ces problèmes. Entre autres le compromis robustesse vs visibilité a été particulièrement affiné.
A noter de plus que cette approche est la base même du watermark dans le domaine DCT, et à engendrer de nombreux algorithme bien plus performant.

2.2.2 Etalement de Spectre (Spread-spectrum)

L'étalement de spectre est une technique utilisée dans les télécommunications radio, notamment par les militaires, pour disperser un signal sur une large bande de fréquence, de façon à le rendre discret et résistant aux interférences. On comprend donc que ce modèle est d'application immédiate au watermarking.
Au cours des paragraphes précédents, seules les idées de base imposant une relation entre d'une part le message et d'autre part l'image ont été présentées. Nous allons décrire a present une nouvelle approche, proposé par F. Hartung et al qui se base sur un pseudo preformatage de la donnée a enfouir en l'*"étalant"* au niveau de la taille de l'image. Il génère ensuite une cle aléatoire de la taille de la donnée préformâtes, puis applique, en terme simpliste, un opérateur binaire*"XOR"* de cette clé et de la donnée étalée (figure 2.1).
Il suffit d'ajouter le résultat obtenu à notre image pour obtenir une image marquée.
Cette méthode est particulièrement intéressante, car elle permet d'exposer plusieurs notions fondamentales des algorithmes de tatouage d'images à savoir l'étalement de spectre, le fait d'utiliser une clé secrète, la modulation d'amplitude, etc.Les étapes d'insertion et d'extraction peuvent se résumer ainsiă:

A noter le fait que les auteurs proposent de plus un système détection permettant de vérifier la présence d'une donnée masquée par tous, mais d'une robustesse relativement médiocre.
Le principe en lui même de spread-spectrum est utilisé sur un domaine DCT

- Etant donné un signal original v_i
- Etant donnée une sequence binaire $a_j \in \{-1, +1\}$ à cacher

1. Etaler ou plus exactement **sur-échantillonner** la sequence a_j d'un facteur "cr" afin d'obtenir une sequence b_i (que nous supposerons ici de la meme longueur que v_i pour des raisons de simplicite).
2. **Amplifier** la sequence bi d'un facteur α; puis la **moduler** avec un **bruit** pseudo aleatoire (ce bruit sert de **clé** secrete) $p_i \in \{-1, +1\}$ afin d'obtenir la marque suivante,

$$w_i = \alpha.b_i.p_i, \tag{2.4}$$

3. La video tatouee est obtenue par addition des deux signaux: video originale et marque precedemment mise en forme,

$$v'_i = v_i + w_i \tag{2.5}$$

TAB. 2.5 – *Algorithme d'insertion Bender & al*

1. Calculer la séquence s, en démodulant la vidéo tatouée à l'aide du bruit,

$$s_j = \Sigma_{cr} p_i.v'_i = \Sigma_{cr} p_i.(v_i + w_i) \tag{2.6}$$
$$\approx \Sigma_{cr} p_i.w_i = \Sigma_{cr} p_i^2.\alpha.b_i \tag{2.7}$$
$$\approx cr.\alpha.b_i = cr.\alpha.a_j \tag{2.8}$$

   Note : Afin que l'hypothèse $\Sigma_{cr} p_i.v_i = 0$ soit vérifiée au mieux, l'auteur propose d'extraire la marque à partir d'une **version filtrée** $v"_i$ de v'_i.
2. Chaque a'_j est donné ensuite par le signe de s_j .

TAB. 2.6 – *Algorithme d'extraction Bender & al*

par l'algorithme de Hartung, mais il peut très bien être appliqué dans différents domaines, tels le spatial (dont l'intérêt est plutôt limité), ou encore des domaines compressés.

2.2.3 Les Fractales

Une autre approche, beaucoup plus technique que les précedentes, consiste à modifier la compression fractale décrite par Fisher ou Jacquin. Une technique initiale est présentée par Puate et Jordan en 96.

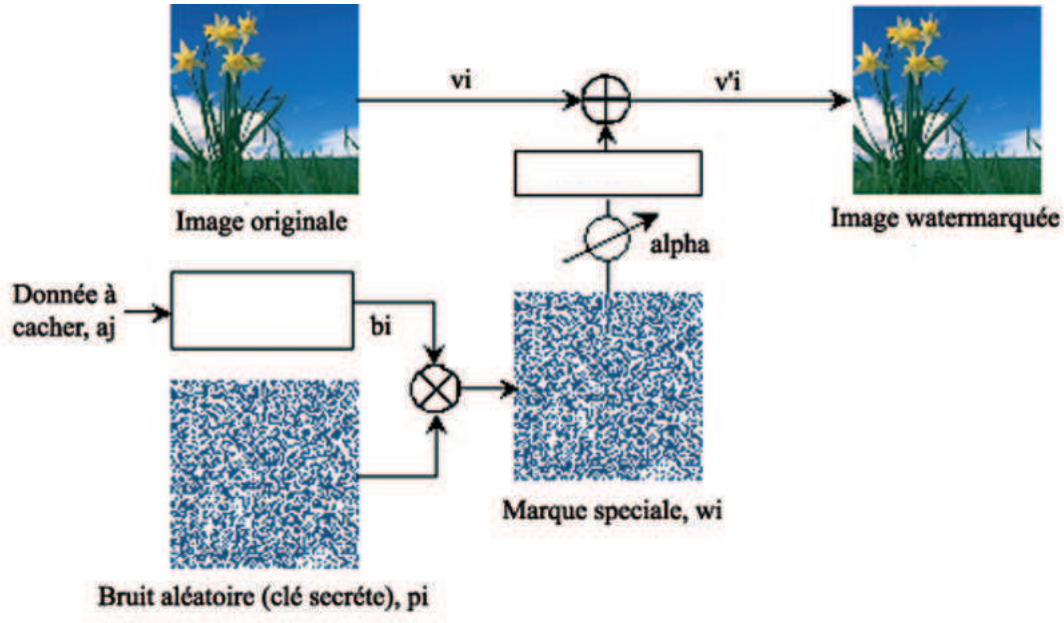

FIG. 2.1 – *Exemple de Spread-spectrum*

Compression Fractale

Tout d'abord, un bref resumé sur cette compression (figure 2.2):
Il consiste à chercher un système de fonctions itérées (IFS) qui permet de repésenter l'image comme un *attracteur*. En terme topologique, un IFS est en quelque sorte un ensemble de fonctions wi : K ->K, contractantes et définies sur un compact (K, d). L'application W qui, à une partie A de K, définie par $W(A) = \bigcup_{i=1}^{n} w_i(A)$ est contractante pour la métrique de Hausdorff et admet un unique point fixe, l'attracteur de l'IFS.
Le principe de la compression fractale est de déterminer les w_i dont l'*attracteur* est l'image qu'on souhaite compresser et qui, pour simplifier le problème, sont des fonctions affines.
On partitionne alors l'image en carrés R_i de taille n par n, appelés *range blocks*, et on recherche des carrés de taille égale ou différente, appelés *domain blocks*, transformables par les w_i en *range blocks* via une transformation spatiale et en niveaux de gris.
En compression, les *domain blocks* sont en général plus gros que les *range blocks*, mais cette contrainte n'intervient plus dans le cadre du watermarking.

Algorithme

Voyons à present en detail les differentes étapes de cet algorithme de watermarking :

Comme l'algorithme precedent, il peux etre utilisé sur un domaine DCT ou spatial, les blocs pouvant etre des donnée brute ou des frequences.

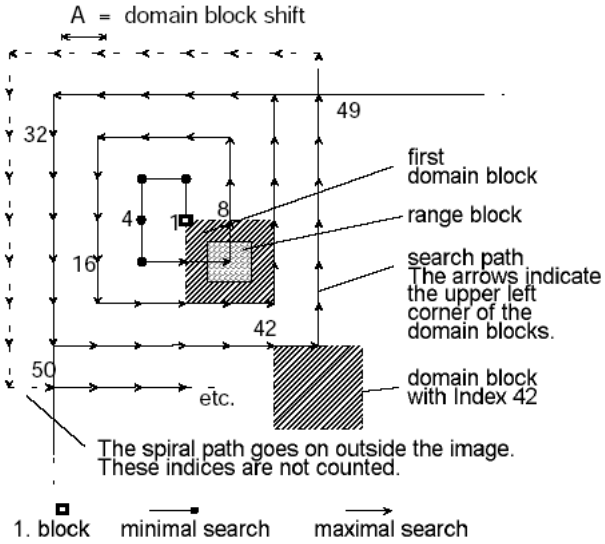

FIG. 2.2 – *Schema de compression en fractale (spiral path)*

- Etant donnée une sequence binaire $S = \{s_1, \ldots, s_k\}$ à cacher, avec une redondance U.

1. Pour chaque bit s_i, nous selectionons U *range block* $\{Rb\}_i$ (choisi aléatoirement par une clé secrete, connu uniquement de l'utilisateur) ;
   - si $s_i = 1$, $\{Rb\}_i$ est codé en cherchant son *domain block* associé $\{Mb\}_i$ par une recherche des regions locale(Local Search Region) $\{A\}_i$;
   - si $s_i = 1$, $\{Rb\}_i$ est codé en cherchant $\{Mb\}_i$ dans $\{B\}_i$;
2. Le reste des *range block* $\{Rb\}_j$ est code en cherchant $\{Mb\}_j$ dans $\{C\}_i = \{A\}_i U \{B\}_i$, comme dans le cas d'un codage de fractale "classique".
3. Créer l'*attracteur*.

TAB. 2.7 – *Algorithme d'insertion Puate & Jordan*

1. A partir de l'attracteur, et pour tout les blocks signé (indiqué grace à la clé secrete ; trouver son domain block associé Vj; si Vj appartient a la region Aj alors, un "1" a ete caché, sinon un "0".
2. Pour chaque bit s_i, la decision finale est prise en considerant pour le groupe de U reponses la majorité de "0" ou de "1".

TAB. 2.8 – *Algorithme d'extraction Puate & Jordan*

2.3 Des images à la vidéo

Le compromis général (ratio, visibilité et robustesse) du tatouage est bien évidemment fortement modifié si l'on considère de la vidéo et non pas des images fixes. Pour une marque à cacher donnée, nous disposons de beaucoup plus de place.
Cependant, à cause de la dimension temporelle, la distorsion visuelle introduite par le marquage est plus difficile à contrôler. La liste des attaques possibles sur une vidéo augmente : par exemple, un schéma de tatouage vidéo doit considérer les conversions de format en termes de nombre d'images par seconde. Enfin, plus que jamais, les problèmes de complexités algorithmiques et de temps de calcul sont à considérer si le temps réel est recherché.
Jusqu'à présent, la majorité des études ont concerné les images fixes, et de nombreuses questions de base se rapportant à la vidéo sont encore ouvertes. On peutpar exemple se demander s'il est préférable de tatouer une vidéo image par image?
Ou bien au contraire, doit-on opter pour une stratégie variant dans le temps?
Malgré tout, la plupart des idées définies en images fixes peuvent être utilisées en vidéo, avec plus ou moins de succès, et vice versa.

Chapitre 3

Les Attaques

3.1 Qu est ce qu une attaques?

Comme nous l'avons vu précédemment, un des points forts d'un tatouage efficace réside dans sa robustesse. Néanmoins, certaines transformations basiques peuvent effacer le marquage, ou du moins potentiellement l'altérer. Toutes ces transformations, volontaires ou involontaires, ayant une influence directe sur le tatouage de l'image, sont appelées des attaques.

3.2 Comment ça fonctionneă?

Dorénavant nous savons comment fonctionne généralement le système de marquage. Nous avons vu que la plupart de ces marques sont en rapport direct avec l'image, soit en modifiant les caractéristiques de certains pixels, soit en modifiant les coefficients DCT etc. Imaginez maintenant que vous voudriez supprimer ces marques, ou du moins les altérer, comment vous y prendriez vous? Et bien évidemment, il vient assez rapidement à l'esprit plusieurs moyens simples, efficaces et rapides. Un exemple simple, la marque est effectuée en modifiant la luminance de certains pixels, il suffit alors d'effectuer un filtre passe-bas sur l'image, et on a alors la quasi certitude d'avoir détruit complètement le tatouage. Autre exemple, un tatouage pris sur le modèle DCT, sera mis en grande difficulté face à une attaque de type géométrique. Malheureusement, il est pour le moment plus simple de lessiver une marque, que d'en graver une solide, à cause du grand nombre d'attaques et de leur efficacité.

3.3 Les différentes attaques

Il existe deux grands types d'attaques sur les images watermarkéesă:

– Les attaques liées a l'image (ou au signal de watermark), dite "aveugleă", dont le but est clairement une suppression simple d'une potentielle donnée masqué dans l'image, en ignorant son contenu. Cela se résume à des transformations plus ou moins violentes. Ces transformations ont pour but de rendre illisible le marquage. Il est intéressant de remarquer néanmoins que ces attaques ne sont pas forcément volontaires. En effet, sans

le savoir l'image peut être dégradée suffisamment pour que le tatouage soit effacé. Un algorithme de marquage robuste est sensé résister de manière efficace a ces types de transformation, ou du moins tant que l image reste utilisable

– Les attaques plus "malicieusesă" dont le but et de retrouver le marquage. Pour cela il suffit de récupérer par différent moyen la "clefă" utilise au marquage de l image originale. Nous pourrons alors modifier comme bon nous semble modifier le tatouage de cette image, le lire, le supprimer...

3.3.1 Les attaques Basique Involontaire

Commençons par une liste, non exhaustive, des types de transformations pouvant potentiellement altérer le tatouage :

Les transformations géométriques

Symétrie horizontale : Le fait simplement d'inverser horizontalement une image est très souvent fatal a une grande partie de watermark. Cette transformation peux sembler aux premier abord bien trop brutale pour conserver le sens d'une image, mais il peux passer inaperçu pour un paysage, ou même pour un film ou aucune scène d'écriture n'intervient (sous-titrage à proscrire). Un exemple est illustré sur la figure 3.1.

FIG. 3.1 – *Exemple de symétrie horizontale*

Recadrement : Ces transformations concernent surtout la mise en page de diverses images scannées. Cela peut être une simple rotation de quelque degrés, ou bien un decoupage brutal d'une partie de l'image. Ces types de recadrements peuvent être des attaques très efficace (voir figure 3.2).

Mise a l échelle : Le fait d'étirer horizontalement ou verticalement une image. Souvent utilisé dans la mise en page également (figure 3.3).

Composition d'images, mosaïque : Il s'agit ici d'utiliser un decoupage d'une image d'une façon beaucoup plus violente et qui se prête assez bien aux pages HTML. Il suffit de découper l'image en autant de morceaux que l'on désire (plus il y a de morceaux plus l'attaque à des chances d'aboutir), puis de recoller cette image au moment de l'affichage en créant par exemple en HTML un tableau dont chacune des cellules contiendra un

FIG. 3.2 – *Decoupage simple d'une image*

FIG. 3.3 – *Exemple de mise en page : Rotation (7°) & Mise à échelle (120%) et Découpage*

morceau de l'image. Cette attaque est très peu applicable en pratique, et heureusement car elle est d'une rare efficacité si l'on se donne les moyens de bien découper l'image (figure 3.4).

Les transformations frequentielle

Ces transformations modifient essentiellement les coefficients de la DCT.

Bruitage et Filtrage Le bruit est une alteration de l'image : toute l'information pertinente dans l'image n'est pas simplement accessible.
Des exemple de bruit artificiel peuvent être :

- le bruit gaussien qui consiste à un ajout successif de valeurs générées aléatoirement à chaque pixel d'une image (figure 3.5).
- ou encore le bruit *"sel et poivre"* qui transforme aléatoirement des pixels de l'image en pixel noir ou blanc (figure 3.6).

Le bruitage d'une image ayant utilisation particulièrement limité, voyons à present les different type de filtre servant justement à récupérer une certaine comprehension de l'image en y filtrant les bruits.
Filtres passe-bas : Faisant partie de la catégorie des filtrages linéaires. On uti-

FIG. 3.4 – *Exemple de mosaïque d'image : Il suffit simplement de la découper en plusieurs parties*

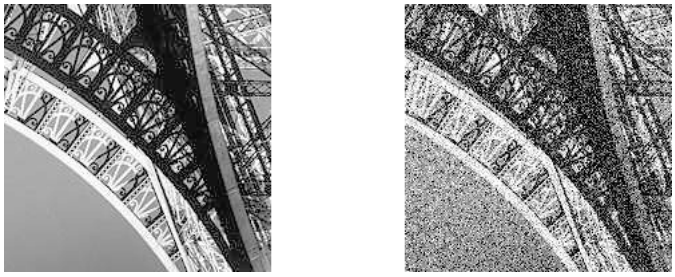

FIG. 3.5 – *Exemple de bruitage gaussien*

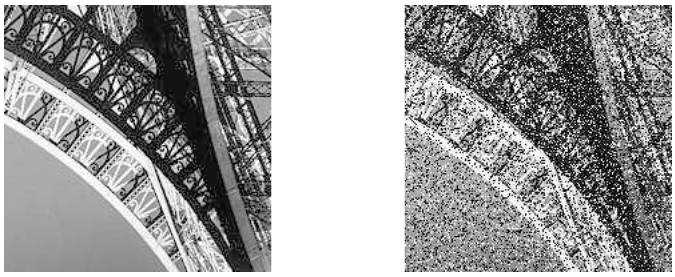

FIG. 3.6 – *Exemple de bruitage Sel & poivre a 30% (15% blanc, 15% noir)*

lise ici la transformée de Fourier pour travailler dans l'espace des fréquences de l'image et dans lequel on ne laisse alors passer que les basses fréquences. En fait, il ne s'agit ni plus ni moins que d'un produit de convolution du signal avec une fonction passe bas (figure 3.7).

Filtre passe-haut : Toujours dans les filtrages linéaires, et souvent appeler "*Sharpen*" du au fait qu il a pour but d'accentuer des contours. Il s'agit simplement de l'inverse du filtre passe-bas, car il ne conserve que les hautes

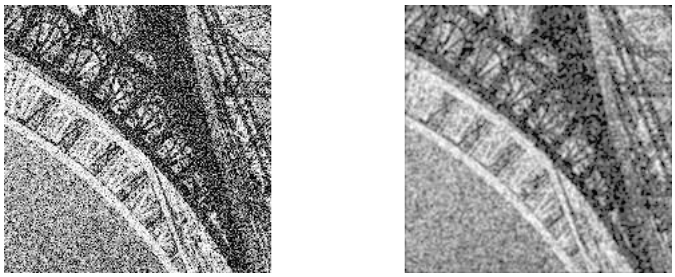

FIG. 3.7 – *Exemple d'application de filtre linéaire*

fréquences. Cette attaque est certainement la moins efficace des transformations car elle conserve le bruit, et que c'est souvent à ce niveau la que se situe le tatouage.

Filtre median : Ce filtre, non linéaire, remplace la valeur d'un pixel par la médiane des valeur de ces voisin. Il est plus robuste que le precedent pour differents types de bruits artificiels, donc plus éfficace en tant qu'attaque (figure 3.8).

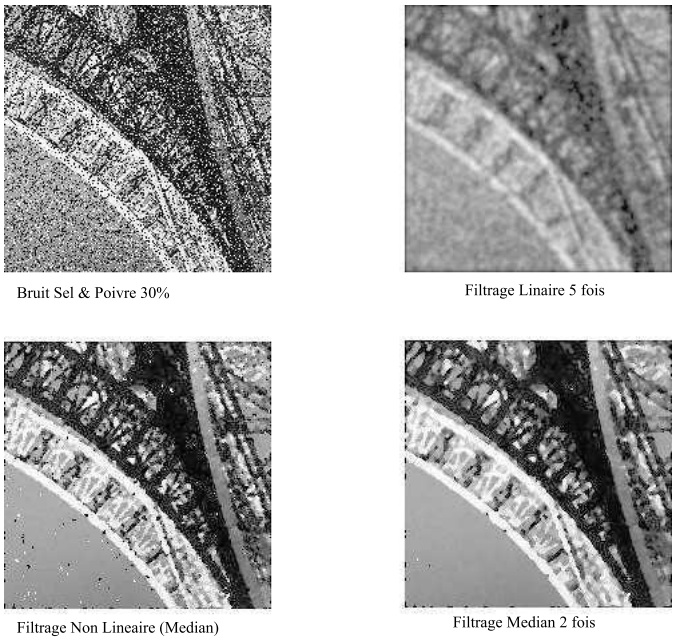

FIG. 3.8 – *Difference entre le filtre median et linéaire pour un bruit sel & poivre*

Les Compressions Les compressions à pertes sont souvent une succession des différentes transformation vu précédemment, ce qui en font des attaques involontaires et souvent très efficace.

Un exemple peut être la compression Jpeg (figure 3.9).
L'avantage de cette méthode reside dans les taux de compression important que l'on peut obtenir, mais son désavantage se situe lui dans le fait qu'il s'agit d'une compression destructive.
En effet plus l'on compresse l'image plus des défauts apparaissent.
Voyons en detail les différentes étapes de cette compression :

1. On découpe d'abord l'image en blocs carrés de 8 pixels sur 8 pixels.
2. On effectue ensuite une Transformée de Fourier (en pratique, une transformée DCT) en 2 dimensions du bloc
3. Et pour finir, on applique un filtre passe bas et c'est la que l'on choisit le taux de compression (en pratique, avec des matrice de quantification).

Plus celui ci va être élevé, plus l'on va supprimer une gamme de fréquences importantes et plus l'image va être dégradée.
Ce type d'attaque s'applique aussi à tout ce qui est conversion de format, par exemple du jpeg vers du gif.

FIG. 3.9 – *Exemple de perte lors de compression jpeg*

3.3.2 Les attaques volontaires

Plusieurs outils commerciaux permettent de *"lessiver"* le marquage d'une image en essayant d'altérer au minimum l'image originaleă; Ces programmes sont appelés*"crackers"*.
Ceux-ci perturbent l'image de telle sorte que, même si la marque reste présente dans l'image tatouée, pour la plupart des algorithmes de marquage, celle-ci est difficile, voire très difficile à extraire, sans recourir à l'image originale, afin de se recaler. Ces perturbations sont souvent des transformations géométriques (voir section précédente) de manière aléatoire. Parmi les outils actuellement disponibles qui réalisent une telle perturbation, les plus référencer sont actuellement**Unzign** et surtout**Stirmark.** UnZign modifie sensiblement la taille de l'image et Stirmark crée des déformations locales et simule un processus d'impression suivie d'une digitalisation de l'image à l'aide d'un scanner. Un exemple d'image obtenu après plusieurs itérations d'unzign peut être observé sur la figure **??**.

Plusieurs itérations mettent en évidence les modifications du format de l'image de cet algorithme.
Un exemple avec stirmark à présent, sur la figure 3.11, met clairement en évidence les modifications géométriques apportées par ce programme.

Concrètement, une illustration de ce que fait stirmark sur un maillage de pixel peut être observée sur la figure 3.12.

FIG. 3.10 – *L'image de gauche correspond a l'image originale, celle du milieux a une iteration de unZign, et celle de droite a cinq iteration*

FIG. 3.11 – *Image originale a gauche, 1 iteration au milieu et 5 iteration a droite de Stirmark*

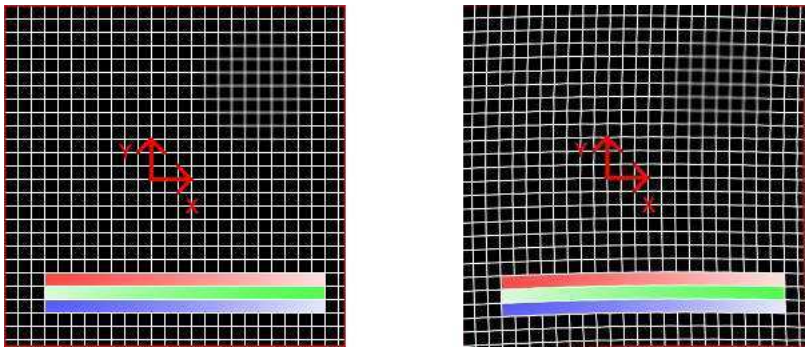

FIG. 3.12 – *Image d exemple mettant en evidence les distortion physique de l algorithme StirMark*

3.3.3 Les attaques de nature cryptologique

Ce type d'attaques est beaucoup plus intéressant, car il demande des connaissances en traitement du signal ainsi qu'une analyse sérieuse du marquage. Les attaques malicieuses sont différentes des attaques aveugles car le pirate va s'attacher à trouver la faiblesse du système qui utilise le watermark. Selon cette faiblesse, il ciblera son attaque.
Par bien des aspects, cela se rapproche beaucoup de la *cryptanalyse* (l'art de briser les systèmes de chiffrement en cryptologie).

Exemple d'attaque sur le copyright

Une des utilisations du watermarking peut être la protection des droits d'auteur (*"copyright"*). Par exemple, le document va être tatoué avec en paramètres le nom de l'auteur, l'identification du contenu, un secret etc. Seul l'auteur connaît ces paramètres. Cette version marquée sera mise à disposition sur Internet. La version originale ne sera pas divulguée. L'auteur est le seul à pouvoir détecter le watermark pour prouver que ce document lui appartient.
Dans ce cas précis, le pirate va chercher à semer le trouble sur l'origine de l'image.
En effet, il ne sert à rien d'ajouter une autre marque au contenu divulgué sur Internet. L'auteur a toujours à sa disposition la version originale.
Le pirate essaie plutôt de recréer une image originale (c'est à dire sans marquage) en soustrayant un faux watermark. Ainsi, il existe deux personnes prétendant avoir la copie originale du contenu divulgué sur Internet. Il est impossible de confondre l'usurpateur.

Exemple d'attaque sur la protection de copie

Ici, tous les contenus (films, fichiers musicaux) vont être tatoués avec la même clef. Sachant cela, le pirate va chercher à estimer cette clef, ce qui lui permettra de laver tous les contenus.
Voici une description très simplifiée de *"l'average attack"*. Le pirate veut trouver la clef utilisée pour marquer un film. Pour simplifier, supposons que la clef soit aussi le signal de watermarkă: W. Pour tout contenu original I^k, la version tatouée est obtenue ainsi :

$$I_w^k = I^k + W \tag{3.1}$$

Le pirate recherche W. Il a accès à toutes les images tatouées I_w^k, mais il ne connaît pas les images originales I^k. Or, sans entrer dans les details, sur un grand nombre d'images, la moyenne des images non tatouées tend vers un gris uniforme de valeur G. On peut écrireă:

$$W' = \frac{1}{N}\sum_{i=1}^{n} I_w^i - G \tag{3.2}$$

Ainsi, il pourra estimer la clef et la soustraire à chaque image pour pirater les contenus. Une implementation de cette attaque à été réalisée en tenant compte d'hypothèses plus réalistes. L'image originale est estimée par un filtrage passe-bas F(.) de l'image tatouée I_wă:

$$I' = F(I_w) \tag{3.3}$$

Le watermark est estimé par une simple différenceă:

$$W' = I_w - I' \tag{3.4}$$

Ceci est fait sur un grand nombre d'images tatouées. Les différents signaux W'_k sont moyennés dans un buffer pour améliorer la qualité de l'estimation.

3.4 Les Contre-attaques

Nous avons vu dans les paragraphes précédents que l'algorithme d'extraction est généralement le duale de l'algorithme d'insertion. Autrement dit, les opérations réalisées lors de l'extraction dérivent directement de celles définies lors de l'insertion du tatouage. Cependant, face à certaines attaques qui laissent la marque dans l'image, mais empêchent le détecteur de l'extraire correctement, des publications récentes proposent d'accroître la robustesse des tatouages en rajoutant de nouvelles étapes à l'extraction, sans remettre en cause le schéma utilisé à l'insertion.
De plus, le problème difficile des attaques géométriques nous amène aussi a suggère que la détection, plutôt que le marquage, est le problème principal du watermarking.
Pour ce faire, plusieurs solutions sont proposées.
Celles-ci servent a augmenter la robustesse du watermark contre les attaques*"aveugle"*(en particulier tout ce qui est rotation ou translation). L'idée basique consiste a utiliser des *"Codes correcteurs"* (ou CC) a la place d une simple duplication-répétition des bits qui va en quelque sorte*"preformater"* le watermark, part exemple en changeant le domaine (spatial ou fréquentiel).

3.4.1 La transformée de Fourier-Mellin

Connue aussi sous le nom de FMT (Fourier Mellin Transform). Celle-ci permet une robustesse face à n'importe quelle transformation à base de rotation ou de translation.
Tout d'abord, faisons la différence entre les 2 transformée, Mellin et Fourier :

Transformée de Fourier : Cette transformée s'applique sur le principe suivantă: toutes les fonctions sont décomposables en une somme de sinusoïdes à des fréquences différentes. Ainsi, lorsque l'on représente une fonction dans un repère Amplitude/Temps, la transformation de Fourier permet de la voir dans un repère Amplitude/Fréquence. On voit donc les composantes en fréquence d'un signal. La transformée étant une sinusoïde, une translation de l'image originale n'a aucune influence dessus.

Transformée de Mellin : Cette transformée va en fait en quelque sorte analyser d'une part notre image en tant que repère orthogonal, ayant l'extrémité inférieure gauche comme origine. Elle fera correspondre ensuite à chaque pixel de l image un vecteur le pointant à partir de cette origine, caractérise par une norme, et un angle. Ce vecteur est alors stoquer sur un 2eme repère correspondant a l image de la transformée, dont l'axe X représente la norme de ce vecteur, et l'axe Y son angle.
Par conséquent le fait d appliquer une rotation a notre image originale

ne changera pas lŠabscisse de l'image de la transformé, mais seulement l'ordonnée.
Une application de cette transformée a celle de Fourier nous permet de garder une ordonnée fixe.

Par consequent, une application successive de ces 2 transformée renforce la robustesse aux attaques géométriques. Un exemple de transformation FMT poussé a l'extrême peut être observé sur la figure 3.13. Il sagit concretement d'une simple transformation en bloc frequentiel dans un repere polaires.

FIG. 3.13 – *Exemple de transformation Fourier-Melin*

3.4.2 Les contres-attaque "Non-Blind"

Cette méthode permet surtout de limiter les différences entre l'image marquée, et l'image originale.
la solution proposée par Davoine & al (figure 3.14) a été inspiré par les travaux précédents effectués sur la vidéo en utilisant les pièces triangulaires pour compenser le mouvement.
L'idée basique est de découper l'image originale en un ensemble de pièces triangulaire.
Cette maille servira alors de maille de référence et sera gardée dans la mémoire pour une étape de prétraitement de récupération du watermark.
Un découpage identique sera alors effectué sur l'image marquée. En utilisant des procédures de compensation, ce maillage modifie sera alors compenser avec le précédent, ce qui limitera les différences dans les possibles transformations affines dû au marquage.
Il suffit par la suite de récompenser l'image marquée avec son maillage original pour retrouver la donnée masquée. Cette contre-attaque nécessite néanmoins l'image originale pour retrouver l'information.

3.4.3 Les contres-attaque "Blind"

Afin de préserver la possibilité de retrouver l'information sans avoir recours à l'image originale, une alternative, proposée par Kutter, consiste à inclure dans certaines parties du watermark des valeurs connue permettant une

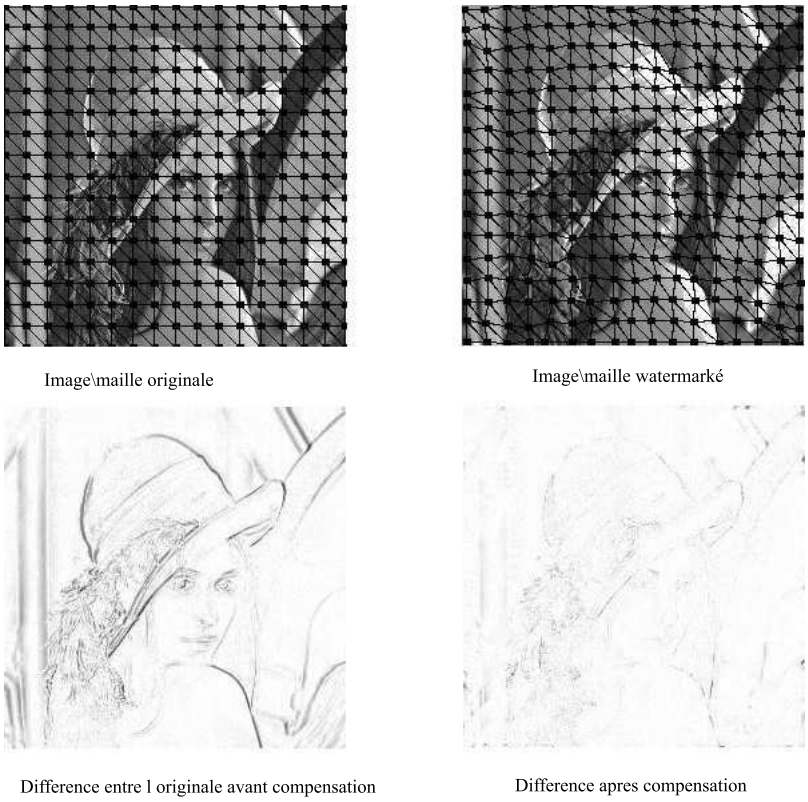

FIG. 3.14 – *Difference apres un traitement de Davoine*

synchronisation spatiale
Cette approche a néanmoins le désavantage de diminuer le ratio avec l information caché, dans le sens ou elle utilise une partie de la place possible pour son propre décryptage.
De plus, sŠil est certain que n'importe quelle transformation affine peux être compensé en testant toute les potentielles transformation inverse, et en sélectionnant le meilleur résultat, cette approche est très lourde lorsque l'on considère une image complète
Donc a la place d utiliser des points précis qui seront comparés avant et apprête le marquage (voir *"Non-Blind"*), Kutter a introduit l'idée de système qui inclue dans le watermark a plusieurs endroits ces points précise.
Cette idée est appelée aussi *"Self-Resynchronisation"*. Le marquage devient alors sa propre référence, rendant la synchronisation possible en nŠayant recours à aucune autre information originale.

3.4.4 Les autres outils a l'étude

Plusieurs études sont menées actuellement via différente méthode, plus ou moins complexe.
Nous en citerons essentiellement 2 relativement prometteur, bien qu'étant très

techniqueă:

Les Ondelettes : Contrairement a la transformée de Fourier ou DCT (utilisé dans Jpeg) qui est une transformée dite fréquentielle (décomposition de l image en un ensemble de sinusoïdes 2D de fréquence différentes) , la transformée en odelette est une transformée spatio fréquentielle qui décompose l image en un emsemlble de sinusoïdes "localesă" de fréquence différent en utilisant une famille de fonctions (fonctions à différentes échelles) qui donne donc en plus l'aspect multi résolution. Sans entrer dans les détails, l idée de base est pour des ondelettes de fréquence basse on a une forme très lissée de l'image et plus on monte en fréquence plus on gagne en résolution et en détails.
Cette méthode renforce un peu plus la robustesse de la transformée de fourrier.

Les Fractales : Les fractales sont également étudiées pour résister entre autres a des attaques telles que le zoom ou l'indexation.

Chapitre 4

Conclusion

Malgré la récente apparition du tatouage d'images, on peut aisément se rendre compte de la pleine expansion de ce domaine. Et ce aussi bien au niveau des techniques de codages qui sont de plus en plus variées, mais aussi au niveau de l'efficacité des algorithmes utilisés en termes de complexité (temps et espace) et de robustesse.
En effet, les chercheurs ont encore une immense tâche à accomplir dans ce domaine, car même si actuellement on peut poser une marque sur un document quelconque sans trop de difficulté, il est encore plus simple de la lessiver. Eh oui comme vous pourrez le constater dans le II, il existe un nombre important d'attaques efficaces et puissantes face à tout type de marques connues. C'est pour cela que les pirates potentiels ont encore un léger avantage face au tatouage d'images.
Or on sait que l'application première du "watermarking" est la défense des droits d'auteurs, et donc si n'importe qui peut détruire n'importe quelle marque, il est alors logique de se demander si le tatouage est utile. Celui-ci se verrait remis en question. En effet, le but premier d'un copyright par exemple est de ne pouvoir être supprimé, de même pour n'importe quelle autre information d'appartenance d'un document. C'est pour cela que les recherches continuent dans cette branche, et que récemment nous avons vu apparaître des techniques bien plus fiables que les antérieures, et de plus les algorithmes se spécialisant (on entend par là le fait que les algorithmes vont dorénavant viser un domaine plutôt qu'un autre, par exemple viser la visibilité plutôt que la robustesse, ou bien la fiabilité plutôt que la robustesse) permettent de répondre au mieux à des applications données.
Même si nous parvenons à marquer efficacement les images, repérer les fraudeurs, ou mettre en place un service de tatouage est encore un autre problème. En effet, pour ce qui est de la mise en place d'un service de tatouage, il faudrait alors par exemple que tous les utilitaires permettant le traitement d'images apposent une marque sur celle-ci, et on voit alors apparaître encore un grand nombre d'autres problèmes. Pour ce qui est de la détection de fraudeurs, il est quasiment nécessaire d'utiliser des mouchards ou autre système équivalent. Des produits permettant de détecter automatiquement les marques commencent a voir le jour, par exemple MarcSpider de la société DigiMarc.
Enfin, outre le "watermarking", les techniques de tatouage peuvent se diversifier. Il est possible de tatouer plusieurs supports de médias différents, tels que

les vidéos, les sons, etc.
Bien sur, on peut s'imaginer que les tatouages ne sont plus forcément le rôle de protection de droits, mais il pourrait simplement véhiculer des informations utiles sur le média en question. Par exemple, pour un DVD marquer le zonage, afin de restreindre l'utilisation du disque à une zone définie.

Table des figures

1.1 Exemple de steganographie realisé à l'aide de lait 4
1.2 Exemple de representation tridimensionelle dans le domaine spatial 6
1.3 Exemple concret de transformation en domaine DCT 8
1.4 Exemple de tatouage visible 9
1.5 Exemple de tatouage (pseudo) invisible 10
1.6 Exemple de tatouage fragile 11

2.1 Exemple de Spread-spectrum 18
2.2 Schema de compression en fractale (spiral path) 19

3.1 Exemple de symétrie horizontale 22
3.2 Decoupage simple d'une image 23
3.3 Exemple de mise en page : Rotation (7°) & Mise à échelle (120%) et Découpage 23
3.4 Exemple de mosaïque d'image : Il suffit simplement de la découper en plusieurs parties 24
3.5 Exemple de bruitage gaussien 24
3.6 Exemple de bruitage Sel & poivre a 30% (15% blanc, 15% noir) . 24
3.7 Exemple d'application de filtre linéaire 25
3.8 Difference entre le filtre median et linéaire pour un bruit sel & poivre 25
3.9 Exemple de perte lors de compression jpeg 26
3.10 L'image de gauche correspond a l'image originale, celle du milieux a une iteration de unZign, et celle de droite a cinq iteration 27
3.11 Image originale a gauche, 1 iteration au milieu et 5 iteration a droite de Stirmark 27
3.12 Image d exemple mettant en evidence les distortion physique de l algorithme StirMark 27
3.13 Exemple de transformation Fourier-Melin 30
3.14 Difference apres un traitement de Davoine 31