UNIVERSITÉ DE GENÈVE
Département d'informatique

FACULTÉ DES SCIENCES
Professeur T. Pun

Robust Digital Image Watermarking

THÈSE

présentée à la Faculté des sciences de Université de Genève
pour obtenir le grade de Docteur ès sciences, mention informatique

par

Shelby PEREIRA

de
Montréal (Canada)

Thèse N° 3191

GENÈVE
2000

La Faculté des sciences, sur le préavis de Messieurs T. PUN, professeur ordinaire et directeur de thèse (Département d'informatique), M. KUNT, professeur (Ecole Polytechnique Fédérale de Lausanne), C. PELLEGRINI, professeur ordinaire (Département d'informatique), A. HERRIGEL, docteur (DCT, Digital Copyright Technologies-Zürich) et S. VOLOSHYNOVSKIY, docteur (Département d'informatique), autorise l'impression de la présente thèse, sans exprimer d'opinion sur les propositions qui y sont énoncées.

Genève, le 22 août, 2000

Thèse -3191-

Le Doyen, Jacques WEBER

Table of Contents

Table of Contents . . . . . **i**

List of Tables . . . . . **iv**

List of Figures . . . . . **v**

Acknowledgements . . . . . **vii**

Abstract . . . . . **viii**

Résumé . . . . . **x**
0.1 Introduction x
0.1.1 Robustesse x
0.1.2 Visibilité xi
0.1.3 Capacité xi
0.1.4 Contributions principales xi
0.2 Stratégies d'insertion de Filigrane xii
0.2.1 Filigranage linéaire additif xii
0.2.2 Filigranage non-linéaire xiv
0.3 Filigranage dans le domaine DFT xiv
0.3.1 Insertion du Filigrane xv
0.3.2 Insertion du *Template* xv
0.3.3 Décodage xv
0.4 Filigranage dans le Domaine DCT xvi
0.5 Évaluation de qualité xvii
0.6 Résultats xix

1 Introduction **1**

2 Watermarking for Copyright Protection **4**
2.1 Watermarking Framework 4
2.2 Watermarking Requirements 6
2.2.1 Capacity 6

2.2.2 Robustness 7
2.2.3 Visibility 9

3 Watermarking as Communications 16
3.1 Watermarking as a Communications Problem 16
3.2 Coding 17
3.2.1 Spread Spectrum Encoding 17
3.2.2 Capacity 22
3.2.3 M-ary Modulation 23
3.2.4 Error Correction Coding 24

4 Algorithm Review 27
4.1 Linear Watermarking 27
4.1.1 Message embedding 27
4.1.2 Attacking channel 28
4.1.3 Message extraction 28
4.2 Category I watermarks 30
4.3 Category II Watermarks 32
4.3.1 Spatial Domain Category II Watermarking 32
4.3.2 Transform Domain Category II Watermarking 33
4.4 Category III Watermarking 36
4.5 Resisting Geometric Transformations 37
4.5.1 Invariant Watermarks 37
4.5.2 Template Based Schemes 37
4.5.3 Autocorrelation Techniques 38
4.6 Evaluation and Benchmarking 39
4.7 Analysis and Research Directions 41

5 Embedding in the Fourier Domain 43
5.1 The DFT and its Properties 44
5.1.1 Definition 44
5.1.2 General Properties of the Fourier Transform 44
5.1.3 DFT: Translation 45
5.2 Embedding 45
5.2.1 Embedding the Watermark 45
5.2.2 Embedding the Template 47
5.3 Decoding 47
5.3.1 Template Detection 47
5.3.2 Decoding the Watermark 51

6 Optimized Transform Domain Embedding **53**
6.1 Overview 53
6.2 Spatial Domain Masking 54
6.3 Problem Formulation 55
6.4 Effective Channel Coding 57
6.5 Wavelet Domain Embedding 58
6.6 Results 59
6.7 Summary and Open Issues 60

7 Attacks **63**
7.1 Problem formulation 64
7.2 Watermark attacks based on the weak points of linear methods 65
7.3 Watermark removal based on denoising 66
7.3.1 ML solution of image denoising problem 67
7.3.2 MAP solution of image denoising problem 67
7.4 Lossy wavelet Compression attack 69
7.5 Denoising/Compression watermark removal and prediction followed by perceptual remodulation 70
7.6 Watermark copy attack 71
7.7 Template/ACF removal attack 72
7.8 Denoising with Random Bending 73

8 Towards a Second Generation Benchmark **74**
8.1 Perceptual Quality Estimation 74
8.1.1 The Watson model 74
8.1.2 Comparison of the Watson metric and the PSNR 75
8.2 Second Generation Benchmarking 77

9 Results **80**
9.1 Perceptual Quality Evaluation 80
9.2 Benchmarking results 83

10 Conclusion and further research directions **86**

Bibliography **88**

List of Tables

1 Codage par Magnitude xvii

6.1 Magnitude Coding 57
6.2 JPEG thresholds at quality factor 10 58

8.1 PSNR and Watson measures for the images benz and mandrill 77

9.1 Watson measures for images bear, boat, girl, lena,watch and mandrill. 84
9.2 Results relative to Petitcolas' benchmark 84
9.3 Benchmark Results 85

List of Figures

1 Modèle d'un système de filigranage adaptatif xii
2 Modèle d'un système de filigranage non-linéaire xiv
3 a)Original Barbara b)PSNR=24.60 c)PSNR=24.59 d)PSNR=24.61 . xviii

1.1 Robustness, Visibility and Capacity 2

2.1 Model of Watermarking Process 5
2.2 Sensitivity of the HVS to changes in luminance 10
2.3 Original images of Barbara (a) and Fish (b) along with their NVF as determined by a generalized gaussian model (c) and (d). 14

3.1 Watermarking as a Communications Problem 16
3.2 Linear Feedback Shift Register 19
3.3 Maximal Length Sequence Generation 20
3.4 Error Probability for Msequences with a 1 bit message 21
3.5 Probability of correct decoding for a 60 bit message 21
3.6 Gaussian Channel Capacity 22
3.7 M-ary encoding 23
3.8 M-ary Decoding 23
3.9 M-ary System Performance 24
3.10 Turbo Encoding 25
3.11 Decoding of Turbo codes 26

4.1 Category I Watermark 31
4.2 Category II Watermark 32
4.3 Category III Watermark 36
4.4 Peaks in DFT before and after rotation 39

5.1 ORIGINAL LENA IMAGE AND LOG OF MAGNITUDE OF FFT . 46

6.1 Original image Lena(a) along with watermarked image (b) and watermark in (c)=(a)-(b). 62

7.1 Communication formulation of a watermarking system 64
7.2 Classification of image denoising methods 67

7.3 Scaling/shrinkage functions of the image denoising algorithms 69
7.4 Approximation of the soft-shrinkage function by quantization with zero-zone. 70
7.5 DFT peaks associated with a template based scheme(a) and auto-correlation based scheme(b) 73

8.1 76
8.2 a) The original Barbara image b)PSNR=24.60, TPE=7.73, NB1=119, NB2=0 c) PSN =24.59, TPE=7.87, NB1=128, NB2=0 d)PSNR=24.61, TPE=9.27, NB1=146, NB2=3 78
8.3 a) The original Lena image b) initials of a name added 79

9.1 Original test images. 81
9.2 Examples of images marked by the 4 algorithms. 82
9.3 a) The bear marked image b) total perceptual errors for blocks 16×16 83

Acknowledgements

I would first like to thank my professor Thierry Pun for accepting me as a student in the Vision Group and closely supervising this thesis over the past two years and offering many helpful suggestions during the course of this time.

I also thank the members of the watermarking group. Firstly I thank Sviatoslav Voloshynovskiy for many instructive technical discussions and also for offering helpful suggestions while editing this document. I also thank him for agreeing to be a jury member. Thanks also to Frederic Deguillaume, Gabriela Csurka and Joe Oruanaidh with whom I was able to share numerous enlightening discussions. I also thank Alexander Herrigel from Digital Copyright Technologies for funding this project in part and also for agreeing to be a member of the jury. Thanks also to Murat Kunt and Christian Pelligrini for agreeing to be members of the jury and for taking time to read my thesis.

I thank Lori Petrucci, Patrick Roth, Alex Dupuis, Stephane Marchand-Maillat, Wolfgang Muller, Henning Muller, David Squire, Alexandre Masselot, Paul Albuquerque, Sergui Startchi, Christian Rauber who all helped make my stay in Geneva more enjoyable. I thank them for the many lunches, coffee breaks, dinner partiess, barbeques, and outings we shared together which all helped me to integrate into Geneva. I thank also Germaine Gusthiot for her help with the many administrative tasks and Dorothy Hauser for her promptness in obtaining many references which greatly facilitated the research task.

Finally I thank the many friends in Geneva who all made my stay more pleasant.

Abstract

Invisible Digital watermarks have been proposed as a method for discouraging illicit copying and distribution of copyright material. While a myriad of algorithms have appeared in recent years, a number of problems remain. In this thesis we develop methods for robustly embedding and extracting 60 to 100 bits of information from an image. These bits can be used to link a buyer and a seller to a given image and consequently be used for copyright protection. When embedding the watermark, a fundamental tradeoff must be made between robustness, visibility and capacity. Robustness refers to the fact that the watermark must survive against attacks from potential pirates. Visibility refers to the requirement that the watermark be imperceptible to the eye. Finally, capacity refers to the amount of information that the watermark must carry.

We present 2 fundamentally different approaches. The first approach is based on the Discrete Fourier Transform (DFT), the magnitude of which is used for embedding bits. Affine transformations on an image lead to corresponding affine transformations in the DFT which suggests that the DFT can be used to recover watermarks which have undergone such transformations. We propose the use of a template consisting of peaks in the DFT domain. If the image is transformed, the peaks are detected and the transformation is detected by solving a point matching problem. In order to ensure that the watermark is invisible, a noise visibility function is calculated in the spatial domain and after the embedding is performed in the DFT domain, the pixels are modulated in the spatial domain to ensure invisibility. Our results indicate that the proposed method successfully recovers watermarks from transformed images, but is relatively weak against compression and cropping.

In recent years it has been recognized that embedding information in a transform domain leads to more robust watermarks. A major difficulty in watermarking in a transform domain lies in the fact that constraints on the allowable distortion at any pixel are usually specified in the spatial domain. Consequently the second approach consists of a general framework for optimizing the watermark strength in the transform domain when the visibility constraints are specified in the spatial domain. The main idea is to structure the watermark embedding as a linear programming problem in which we wish to maximize the strength of the watermark subject to a set of linear constraints on the pixel distortions as determined by a masking function. We consider the special cases of embedding in the DCT domain and wavelet domain

using the Haar wavelet and Daubechies 4-tap filter in conjunction with a masking function based on a non-stationary Gaussian model, but the algorithm is applicable to any combination of transform and masking functions. Unfortunately the algorithm is not applicable to the DFT since the computational complexity of global transformations is overwhelming. Our results indicate that the proposed approach performs well against lossy compression such as JPEG and other types of filtering which do not change the geometry of the image, however at this time the watermark cannot be recovered from images which have undergone an affine transformation.

As a second aspect of the thesis we also develop robust evaluation methods. This consists of two parts. Firstly we develop a new approach for evaluating watermark visibility based on the Watson metric. The results indicate that the perceptual quality as measured by the Watson metric is consistently more accurate than that provided by the typically used PSNR criterion. We also define a new benchmark consisting of attacks which take into account prior information about the watermark and the image. These attacks are much more powerful than the ones available in other benchmarking tools which do not use prior information, but rather perform general image processing operations. Our results relative to the proposed benchmark indicate that the optimized non-linear embedding approach we developed performs markedly better than existing commercial software which suggests that future research should consist of pursuing these lines rather than the linear additive watermarking paradigm dealt with in the bulk of the literature.

Résumé

0.1 Introduction

Le travail décrit dans cette thèse se situe dans le domaine du filigranage d'images digitales. Il s'insère dans un projet global dont le but est de développer un système pour la protection des droits d'auteur sur le Web. Nous regardons ici le cas particulier des droits d'auteurs appliqués aux images et nous nous limitons à l'aspect traitement d'image du problème. Les aspects cryptographiques sont traités par Herrigel [33].

L'idée principale du filigranage consiste à encoder une information dans une image en effectuant des légères modifications sur les pixels. Pour qu'un filigrane soit utile, trois critères principaux doivent être satisfaits: le filigrane doit être robuste, invisible, et doit contenir une certaine quantité d'information que l'on désigne par le terme de "capacité". Nous considérons ces trois points importants en détail.

0.1.1 Robustesse

Un filigrane doit être capable à résister plusieurs type d'attaques. Ces attaques peuvent être divisées en quatre catégories:

La premiere catégorie concerne les attaques simples qui ne changent pas la géométrie de l'image, et qui n'utilise pas d'information *a priori* sur l'image. Les attaques de cette catégorie sont : la compression JPEG et ondeletes, le filtrage, l'addition de bruit, la correction gamma, et l'impression suivie par une rénumérisation à l'aide d'un scanner.

La deuxième catégorie comprend les attaques qui modifient la géométrie de l'image. On y trouve la rotation, le changement d'échelle, et les transformation affines. Des attaques plus sophistiqués incluses dans cette même catégorie sont l'enlèvement de lignes ou de colonnes, ainsi que les distortions locales non-linéaires qui sont mises en oeuvre dans le programme Stirmark [66].

La troisième catégorie d'attaques comprend les attaques dites "d'ambiguité". Ici l'idée est de créer une situation où l'on ne peut pas décider de la véritable origine d'un filigrane. Craver [13] démontre que dans certaines conditions, on peut créer un faux original en soustraiant un filigrane d'une image déjà marquée. Récemment Kutter [43] a aussi montré qu'il est possible de copier un filigrane d'une image à une autre: la

marque peut être estimée statistiquement sur l'image protégée, puis ajoutée à l'image de destination.

La dernière catégorie comprend les attaques consistant à enlever le filigrane de l'image. Ici on utilise des informations *a priori* sur le filigrane et sur l'image pour effectuer un débruitage optimal de l'image, où le filigrane est traité comme un bruit. Voloshynovsky [95] démontre qu'il est même possible d'augmenter la qualité de l'image en enlevant le filigrane avec des algorithmes de débruitage bien adaptés.

0.1.2 Visibilité

Le filigrane doit aussi être invisible. En pratique ceci implique que l'image originale et l'image marquée doivent être indifférenciables à l'oeil. L'évaluation objective de la visibilité est un sujet difficile. Dans un premie temps, Petitcolas a proposé un PSNR de 38dB pour indiquer si deux images sont indifférenciable. Nous trouvons en pratique que cette mesure est insuffisante et dans cette thèse nous proposons une mesure basée sur la métrique de Watson.

0.1.3 Capacité

Le filigrane doit aussi contenir au moins 60 bits d'information. Ceci est important selon les dernières standardisations puisque l'on aimerait insérer un identificateur qui associe un vendeur, un acheteur et une image. Malheureusement la plupart des publications actuelles sur le sujet décrivent des filigranes ne portant qu'un seul bit d'information.

0.1.4 Contributions principales

Les contributions principales de cette thèse sont:

1. Le développement de techniques basées sur la transformée discrète de Fourrier (DFT) afin de rendre les filigranes résistants aux transformations géométriques.

2. Le développement d'un nouvel algorithme non-linéaire, basé sur la transformée DCT, qui dépend de l'image, et qui résiste à l'attaque consistant à copier un filigrane d'une image et l'insérer dans une autre.

3. Le développement d'une nouvelle méthode pour évaluer la qualité d'une image signée.

Nous détaillons ces contributions dans les sections qui suivent.

0.2 Stratégies d'insertion de Filigrane

Il existe plusieurs stratégies pour insérer un filigrane satisfaisant aux trois critères présentés. Cox [12] utilise le schéma général de la figure 0.2. Tout d'abord on calcule

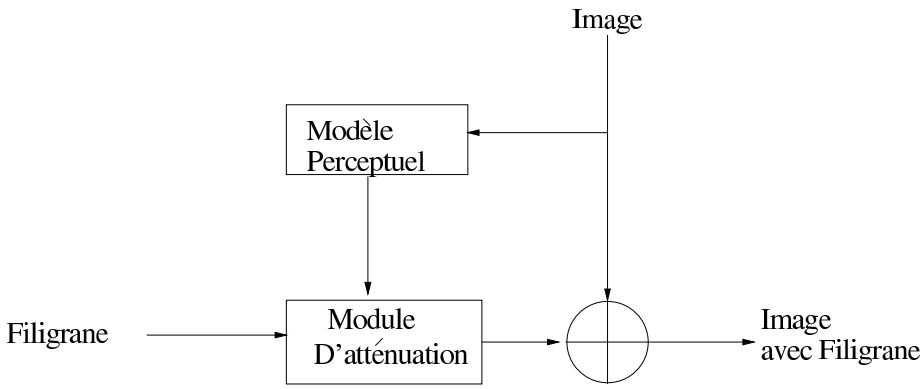


Figure 1: Modèle d'un système de filigranage adaptatif

des contraintes imposées à partir de l'image à marquer afin de préserver l'invisibilité du filigrane. On calcule ensuite le filigrane, on applique nos contraintes, et le résultat est ajouté à l'image originale, nous donnant ainsi l'image signée. Ceci est un système de filigranes additif et linéaire. La plus grande partie de la recherche dans le domaine de filigranage s'est effectué dans ce cadre, et nous en détaillons les principes.

0.2.1 Filigranage linéaire additif

Nous avons l'information binaire $b = (b_1, ..., b_L)$ à insérer dans l'image $x = (x_1, ..., x_N)^T$ de taille $M_1 \times M_2$, où $N = M_1 \cdot M_2$. Typiquement b est converti dans un forme plus robuste en utilisant les codes de corrections d'erreurs tel que les codes BCH, LDPC ou turbo [62, 93]. En général cette conversion peut être décrite par la fonction $c = Enc(b, Key)$ où Key est la clé secrète de l'utilisateur. Le filigrane s'exprime alors comme une fonction de c par:

$$w(j) = \sum_{k=1}^{K} c_k p_k(j) M(j) \tag{1}$$

où M est un masque local qui atténue le filigrane pour le rendre invisible, et p est un ensemble de fonction orthogonales en deux dimensions. L'image avec le filigrane additif s'écrit alors :

$$y = h(x, w) = x + w \tag{2}$$

Pour décoder le filigrane on considère que l'on reçoit l'image marquée y' après avoir subit diverses distortions. Afin d'extraire le filigrane nous calculons d'abord une estimation du filigrane:

$$\widehat{w} = Extr(y', Key) \tag{3}$$

Pour calculer cette estimation, on modélise le bruit introduit par l'image ainsi que le bruit apporté par des attaques par une distribution de probabilité $p_X(.)$, et l'on obtient alors le maximum de vraisemblance par:

$$\widehat{w} = \arg\max_{\widetilde{w} \in \Re^N} p_X(y' \mid \widetilde{w}) \tag{4}$$

Dans le cas d'un bruit Gaussien nous obtenons la moyenne. Dans le cas d'un bruit Laplacien, nous obtenons la médiane. Il est aussi possible d'utiliser l'estimateur MAP (*Maximum a Posteriori*).

$$\widehat{w} = \arg\max_{\widetilde{w} \in \Re^N} \{ p_X(y' \mid \widetilde{w}) \cdot p_W(\widetilde{w}) \} \tag{5}$$

où $p_W(.)$ est la distribution du filigrane. Si l'on fait l'hypothèse que l'image est locallement Gaussienne avec $x \sim N(\overline{x}, R_x)$ et $w \sim N(0, R_w)$ et avec matrices de covariance R_x et R_w, nous avons:

$$\widehat{w} = \frac{R_w}{R_w + R_x} (y' - \overline{y}') \tag{6}$$

où $\overline{y}' \approx \overline{x}$, et $\widehat{R}_x = \max\left(0, \widehat{R}_y - R_w\right)$ est l'estimation de la variance locale de l'image $\left(\widehat{R}_x = \sigma_x^2 I\right)$.

Pour décoder le filigrane nous effectuons d'abord les corrélations:

$$r = \langle \widehat{w}, p \rangle . \tag{7}$$

Le décodeur optimal s'exprime comme:

$$\widehat{b} = \arg\max_{\widetilde{b}} p\left(r \mid \widetilde{b}, x\right). \tag{8}$$

En pratique ceci revient à décoder des codes de correction d'erreurs. Il est important de noter que souvent on effectue une quantification des valeurs avant le décodage des codes de correction d'erreurs, ce qui conduit à une perte de 3-6dB. Les codes les plus puissants (turbo et LDPC) utilisent toute l'information afin d'obtenir une meilleure performance.

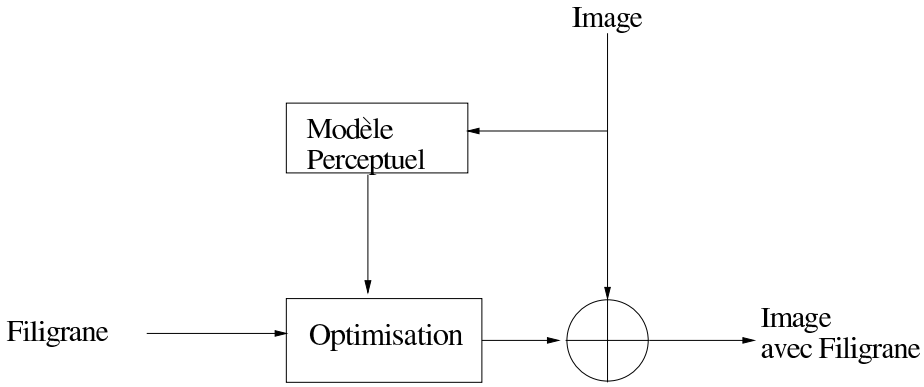


Figure 2: Modèle d'un système de filigranage non-linéaire

0.2.2 Filigranage non-linéaire

Le modèle de filigranage linéaire est pratique parce qu'il est simple. Malheureusement, l'utilisation de ce modèle entraîne des pertes à l'insertion de la signature qui résultent du fait qu'on traite l'image comme un bruit. Cependant, l'image est connue, et cette information devrait être exploitée pour l'insertion. Cox propose le modèle général en figure 0.2.2 pour un système de filigranage plus puissant. Dans ce système, au lieu d'une simple atténuation, nous optimisons le filigrane par rapport à l'image lors de l'insertion de la signature. Typiquement nous pouvons utiliser les tables de quantisation JPEG ou des ondelettes pour optimiser le filigrane par rapport à la compression.

0.3 Filigranage dans le domaine DFT

La DFT a plusieurs propriétés intéressantes qui font que cette transformée se prête bien au filigranage. Le DFT est défini en 2D par:

$$F(k_1, k_2) = \sum_{x_1=0}^{N_1-1} \sum_{x_2=0}^{N_2-1} f(x_1, x_2) e^{-j2\pi x_1 k_1/N_1 - j2\pi x_2 k_2/N_2} \tag{9}$$

et son inverse par:

$$f(x_1, x_2) = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} F(k_1, k_2) e^{j2\pi k_1 x_1/N_1 + j2\pi k_2 x_2/N_2} \tag{10}$$

où N_1 et N_2 sont les dimensions de l'image. En pratique nous travaillons avec la magnitude. Si nous appliquons une transformation $\boldsymbol{T}$ dans le domaine spatial nous

trouvons que la magnitude de la DFT subit la transformation suivante :

$$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \rightarrow \left(\boldsymbol{T}^{-1}\right)^T \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \tag{11}$$

Cette propriété est importante car si nous pouvons déterminer la transformation subit par l'image, alors nous pouvons la compenser dans le domaine DFT avant de décoder le filigrane. Nous allons voir comment la transformation peut être détectée à l'aide d'une mire de recalage (*template*). On note aussi que la magnitude de la DFT est invariante à la translation.

0.3.1 Insertion du Filigrane

Pour insérer un filigrane dans le domaine DFT nous commençons par compléter l'image avec des zéros afin de l'amener à une taille fixe de 1024×1024. Ensuite la séquence de bits est traduite par la correspondance $0 \rightarrow -1$ et $1 \rightarrow 1$. Pour insérer le message on choisit une bande de fréquences entre f_{w1} et f_{w2}, qui se situe au milieu du spectre. Dans cette bande, nous choisissons des paires de points aléatoirement et nous changeons les valeurs de sorte que $k_w \tilde{m}_{c_i} = (x_i, y_i) - (y_i, -x_i)$, où (x_i, y_i) sont les points choisis, $\tilde{m}_{c_i}$ sont les bits du message, et k_w est la force du filigrane. La force est choisie de façon à limiter les distorsions dans le domaine spatial lorsque l'on maximise la force dans le domaine spectral.

0.3.2 Insertion du *Template*

Pour insérer le *template* on choisit 8 points aléatoirement dans le DFT et on insère des pics à ces points. Le *template* ne contient aucune information mais il sert comme un outil pour détecter les transformations subies par l'image.

0.3.3 Décodage

Le décodage se divise en deux parties: détection du *template* et le décodage du filigrane. Pour détecter le *template* nous utilisons l'algorithme suivant:

1. Calcul de la magnitude du DFT de l'image.
2. Extraction des maxima locaux.
3. Estimation de la matrice de correspondance entre les positions connues du *template* et les positions des maxima détectés. Pour limiter l'exhaustivité de la recherche, on contraint le type de matrice à estimer aux transformations raisonables.

Si le *template* est détecté nous pouvons dire avec une probabilité $P_{false} = (\frac{9}{2000})^8 \times 3000 \approx 5.0 \times 10^{-16}$ qu'un filigrane était inséré. Une fois que le *template* est détecté, la matrice calculée nous permet de compenser pour la transformation géométrique et de décoder le filigrane correctement.

0.4 Filigranage dans le Domaine DCT

La contribution la plus importante de cette thèse est le développement d'une méthode non-additive tenant compte de toute l'information de l'image lors de l'insertion du filigrane. Le problème avec l'insertion d'un filigrane dans le domaine spectral est de limiter les distortions dans le domaine spatial tout en maximisant la force dans le domaine spectral. Voloshynovskiy propose l'utilisation d'une fonction NVF (Noise Visibility Fonction) pour mesurer les distortions visuelles, qui indique le niveau maximal de modification acceptable à chaque pixel sans qu'elle devienne visible. Cette fonction se définit comme suit :

$$NVF(i,j) = \frac{w(i,j)}{w(i,j) + \sigma_x^2}, \tag{12}$$

où $w(i,j) = \gamma[\eta(\gamma)]^\gamma \frac{1}{||r(i,j)||^{2-\gamma}}$ et $r(i,j) = \frac{x(i,j) - \bar{x}(i,j)}{\sigma_x}$. Le NVF nous dit surtout comment marquer les régions texturées. Pour marquer les régions uniformes, nous pouvons exploiter la luminance, en observant que l'oeil est moins sensible aux variations dans les régions claires que dans les régions foncées. Ceci conduit à:

$$\Delta_{pi,j} = (1 + k \cdot CST(x_{i,j}) \cdot ((1 - NVF(i,j)) \cdot S + NVF(i,j) \cdot S_1) \tag{13}$$

où CST est la fonction de sensibilité à la luminance, et k, S, et S_1 sont des constantes qui pondèrent le poid des composantes.

Pour encoder dans le domaine DCT nous pouvons alors formuler un problème de minimisation. Pour encoder un "1" on maximisera une valeur DCT et pour encoder un "0" on minimisera une valeur DCT. Pour optimiser la force du filigrane on formule le problème comme suit :

$$\min_{\boldsymbol{x}} \boldsymbol{f}'\boldsymbol{x} \qquad ; \qquad \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \tag{14}$$

où $\boldsymbol{x} = [x_{11} \ldots x_{81} x_{12} \ldots x_{82} \ldots x_{18} \ldots x_{88}]^t$ est le vecteur de valeurs DCT, $\boldsymbol{f}$ est un vecteur de zéros sauf aux positions où l'on aimerait insérer "1" ou "0". A ces positions on insère "1" et "-1" respectivement dans $\boldsymbol{f}$. $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$ contient les contraintes:

$$\boldsymbol{A} = \begin{bmatrix} IDCT \\ ---- \\ -IDCT \end{bmatrix} ; \quad \boldsymbol{b} = \begin{bmatrix} \Delta_p \\ ---- \\ \Delta_n \end{bmatrix} \tag{15}$$

où IDCT est la matrice d'inversion de la DCT. On résout la minimisation par la méthode Simplex.

Pour mettre ceci en pratique, nous divisons l'image en blocs de taille 8×8 et dans chaque bloc on choisit 2 endroits pour insérer l'information. Au décodage on lit la séquence des endroits choisis et l'on décode le code de correction d'erreurs. Dans cette contexte il se trouve que les turbo codes nous donnent la meilleure performance.

Une amélioration importante de la méthode consiste à utiliser le codage décrit dans le tableau 1. Au décodage on prend la magnitude de la valeur et on la compare

Table 1: Codage par Magnitude

signe(c_i)	bit	Codage
+	0	baisse c_i
−	0	augmente c_i
+	1	baisse c_i
−	1	augmente c_i

à un seuil T qu'on fixé à 15. L'avantage principal de ce type de codage est que le filigrane varie selon l'image. En particulier l'insertion du filigrane dépend de la valeur déjà présente dans le DCT, et par conséquent, ce type de filigrane ne peut pas être copié d'une image à une autre.

0.5 Évaluation de qualité

Pour effectuer des comparaisons objectives de la qualité des images marquées, il faut définir un critère objectif. Le PSNR, qui est en général utilisé, est insuffisant. La figure 3 montre une comparaison entre 4 images: l'originale, le filigrane adaptatif utilisant un modèle stationnaire Gaussien généralisé, le filigrane adaptatif utilisant un modèle non-stationnaire Gaussien, et le filigrane non-adaptatif. Toutes les images sont marquées avec le même PSNR mais on remarque que les filigranes adaptatifs restent invisibles et alors que le filigrane non-adaptatif est visible. Ceci résulte du fait que l'oeil est plus sensible aux modifications dans les régions plates. Le PSNR est un critère global qui ne pondère pas les distorsions en fonction de la région.

Pour remédier à ce problème, nous pouvons utiliser la métrique de Watson, qui évalue l'impact des distorsions en fonction de la région de l'image. L'idée principale est de calculer la DCT bloc par bloc sur toute l'image. Les blocs sont de taille 8×8. Pour chaque coefficient DCT, nous fixons un seuil de visibilité t_{ij} qui a été déterminé expérimentalement. Ce seuil est ensuite ajusté en fonction de la luminance et de la texture par les équations :

$$t_{ijk} = t_{ij}(\frac{c_{00k}}{\overline{c}_{00}})_t^a \tag{16}$$

(a) original

(b) adaptatif sGG

(c) adaptatif nG

(d) non-adaptatif

Figure 3: a)Original Barbara b)PSNR=24.60 c)PSNR=24.59 d)PSNR=24.61

et

$$m_{ijk} = Max[t_{ijk}, |c_{ikj}|^{w_{ij}} {t_{ijk}}^{1-w_{ij}}] \tag{17}$$

où c_{00k} est le coefficient DC du bloc k, $\overline{c}_{00}$, et a_t sont des constantes, m_{ijk} est un seuil de visibilité, et w_{ij} est le degré de masquage dans les régions texturées. Pour mesurer la dégradation subie par une image, nous utilisons alors :

$$d_{ijk} = \frac{e_{ijk}}{m_{ijk}} \tag{18}$$

où e_{ijk} est l'erreur pour une composante DCT. En effet nous pondérons l'erreur par sa visibilité. Pour obtenir un critère global nous effectuons une sommation de Minkowski des erreurs aux carrées sur tous les blocs. En utilisant cette métrique, nous obtenons pour le filigrane non-adaptatif une erreur de 9.27, alors que les filigranes adaptatifs ont des erreurs comprises entre 7.73 et 7.87. Nous observons que la métrique arrive à bien différencier la qualité des images bien que le PSNR ne nous donne aucun renseignement utile.

0.6 Résultats

Les résultats indiquent que l'algorithme DFT à une bonne performance globale. En particulier, l'algorithme est robuste contre les distortions géométriques ainsi que contre les filtres passe-haut et passe-bas. Malheureusement l'algorithme ne résiste pas à une compression JPEG avec un facteur de qualité inférieur à 60%, et ne résiste pas aux distortions locales mises en oeuvres dans le program Stirmark. De plus, l'algorithme ne résiste pas à l'attaque de copie du filigrane. L'algorithme DCT, par contre, n'a aucune résistance aux distortions géométriques. En revanche, il résiste à l'attaque de copie du filigrane ainsi qu'à une compression JPEG à un facteur de qualité de 10%. Malheureusement, le filigrane ne résiste pas à des distorsions locales.

Par conséquent, le travail qui reste à faire consiste à ajouter un *template* à l'algorithme DCT pour le rendre résistant aux transformations géométriques. Il est aussi important de travailler sur de nouvelles méthodes qui résistent aux distortions locales.

Chapter 1

Introduction

The World Wide Web, digital networks and multimedia afford virtually unprecedented opportunities to pirate copyrighted material. Digital storage and transmission make it trivial to quickly and inexpensively construct exact copies. Consequently the need for methods to identify, authenticate, and protect multimedia has spurred active research in watermarking.

Watermarking is a special case of the general information hiding problem. The central idea is to robustly embed information in a medium known as the "cover" object in order to produce the "stego" object. The embedding is done in such a way that the cover and stego objects are indistinguishable. Cover objects include images, 3D graphics, video, music, text documents and html documents. A survey of methods used to hide information in various media is given in [65]. In this thesis we will be interested in the problem of image watermarking for the purpose of copyright protection. With respect to this application, we would like to embed information into an image which associates a buyer with a seller. In order to be useful this information must be easily and reliably recovered at a later time if the need arises.

The rapidly expanding world wide web provides a plethora of applications for watermarking technology. Recently, art galleries and museums as well as private photographers and artists have showed interest in selling their work on the web. The availablility of high resolution color scanners and monitors as well as image processing software makes the process of transforming traditional color images into high resolution digital images a relatively easy task. Once in digital form, the growing infrastructures for e-commerce can be exploited to reach a potentially enourmous clientele over the web. However, the drawback of this approach arises from the fact that with current technology, digital images can be reproduced easily, quickly and without loss. Consequently some mechanism for the protection of owner's rights must be developed to discourage theft within this context. This need has spurred the recent research efforts in image watermarking.

With respect to the general information hiding problem, a tradeoff is involved between robustness, visibility and capacity as illustrated in figure 1.1. Robustness

refers to the ability of the inserted information to withstand image modifications (intentional or unintentional). In information hiding we require that the modified object be indistinguishable from the original. In the image watermarking context we use the term "visibility" to designate this concept. In some publications the word "transparency" is also used. Finally capacity refers to the amount of information we are able to insert into the image. Designing and optimizing informaiton hiding algorithms involves the delicate process of juidiciously trading off between these three conflicting requirements.

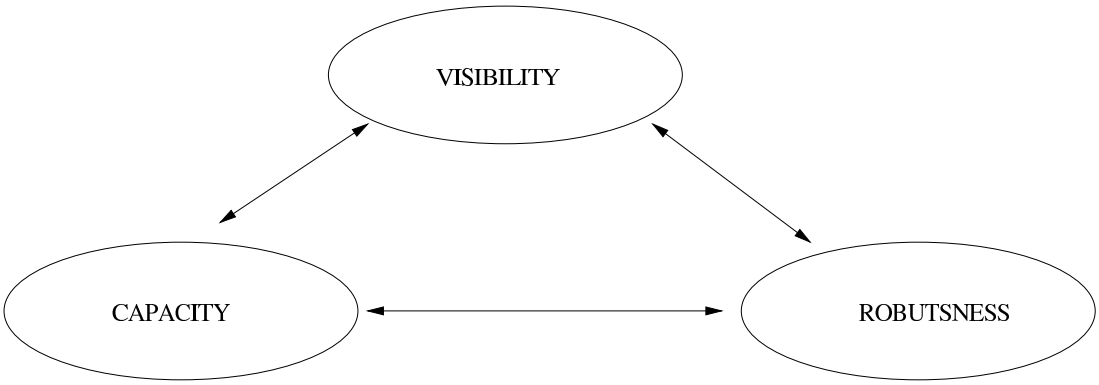


Figure 1.1: Robustness, Visibility and Capacity

In the multimedia context work in information hiding has been undertaken in the following 3 subgroups: steganography, tamper-proofing, and watermarking. In the case of steganography, we are interested in sending large quantities of information (large capacity), however we are less concerned with robustness. Our visibility requirement is that the cover object does not appear to contain any hidden information. Many applications can be found in military settings. For example secrets might be exchanged over the web by invisibly embedding information in images which are then transmitted. A wide array of possibilities including embedding information in music, videos and even in text.

Tamper-proofing [39] involves embedding information into the cover object which is then used at detection to determine if and how the object has been modified. In this case the robustness criteria is somewhat different since we would like the watermark to change as a function of the distortions introduced into the medium. Our capacity requirement is relatively low and our visibility requirements may be relatively weak since we may be able to tolerate some degradation in the image. One major application is in the authentication of digital evidence in a court case. In such a scenario a different approach employing a fragile watermark could be used to indicate that indeed a digital image has not undergone processing intended to sabotage evidence. Fragile watermarking refers to the process of inserting a watermark which will be destroyed if the medium is changed. This problem is considerably easier than copyright protection since only one bit of information is being transmitted. However

the general tamper-proofing problem is quite interesting since algorithms must be designed so that it is possible to determine the location of modifications in an image.

Watermarking involves embedding information which can be used to identify the buyer and seller of a given cover object. This information can later be detected to identify the true buyer and seller of a given cover object. In this context robustness is of prime importance since a malicious third party may intentionally attempt to remove the watermark and later illegally resell the object. Furthermore we also require that the watermark be invisible since the cover object is of value. This is in sharp contrast to steganography where the cover object has no value in itself. The capacity requirement is also different since unlike steganography we only have a small amount of information to communicate which includes buyer and seller identifiers. This usually amounts to roughly 100 bits.

This thesis covered several aspects of the watermarking problem. The most important result consists of a formalization of the embedding problem. In particular we demonstrate how to maximize the strength of the watermark subject to constraints on the visibility. Another important result is the development of methods for recovering watermarks which have undergone a general affine transformation. Finally, we also propose a new method for evaluating image quality and a new benchmark for evaluating the robustness of watermaking schemes. Both of these problems have been addressed in the literature, but the initial evaluation criteria for visibility and robustness prove to be inadequate in practice. Our results indicate that the visibility criteria developed yields a substantial improvement over the existing methods. Furthermore our new benchmarks contains attacks which correctly pinpoint weaknesses of existing methods.

The thesis is structured as follows. In chapter 2 we present the watermarking problem and the requirements that a successful watermarking algorithm must fulfill. In chapter 3 we formulate watermarking as a communications problem. In chapter 4 we review algorithms which have appeared in the recent literature. In chapter 5 we begin the presentation of the central contributions of this thesis by describing DFT domain algorithms. In chapter 6 we present the key contribution of this thesis which is a mathematical formalization of the embedding process. In chapter 7 we turn our attention to attacking watermarkings. We then define a new measure of perceptual quality of an image in chapter 8 and incroporate the quality measure and the attacks in a new benchmark which we use to evaluate our algorithms as well as two commercial software packages. Finally, we present our conclusions in chapter 9.

Chapter 2

Watermarking for Copyright Protection

While digital watermarking for copyright protection is a relatively new idea, the idea of data hiding dates back to the ancient Greeks and has progressively evolved over the ages. An excellent survey of the evolution of data hiding technologies can be found in [38]. The inspiration of current watermarking technology can be traced to paper watermarks which were used some 700 years ago for the purpose of dating and authenticating paper [28]. The legal power of such watermarks was demonstrated in a court case known as "Des Decorations" [21]. Watermarks in the context of digital images first appeared in 1990 [87] and has since received considerable attention with a particularly rapid proliferation of research into watermarking algorithms in the last six years.

In this literature survey we highlight some of the key developments in the digital watermarking community. We begin in section 2.1 by presenting the general watermarking framework. We continue in section 2.2 by stating the requirements a watermarking system must fulfill. In sections 3 we formulate the watermarking problem as a communications problem. Then in sections 4.2 to 4.4, we present three Categories of watermarking and present watermarking algorithms which fall under each category. Here we consider only robustness against attacks that do not modify the geometry of the image. Section 4.5 describes strategies for making watermarks robust to geometric transformations of the image. In section 4.6 we turn our attention to the problem of evaluating and comparing algorithms. Finally in 4.7 we point out the limitations of current strategies and indicate the directions taken in this thesis in order to improve the performance of current watermarking algorithms.

2.1 Watermarking Framework

A simple yet complete model of the watermarking framework appears in figure 2.1. The embedding process takes the cover data, a secret key and the copyright

message to be embedded and produces the stego data. The decoding process is completely analagous where we take the stego data, a secret key, and attempt to detect if the image contains a watermark, and if it does, we attempt to decode the message. With respect to the watermark decoding process, the notion of "oblivious"

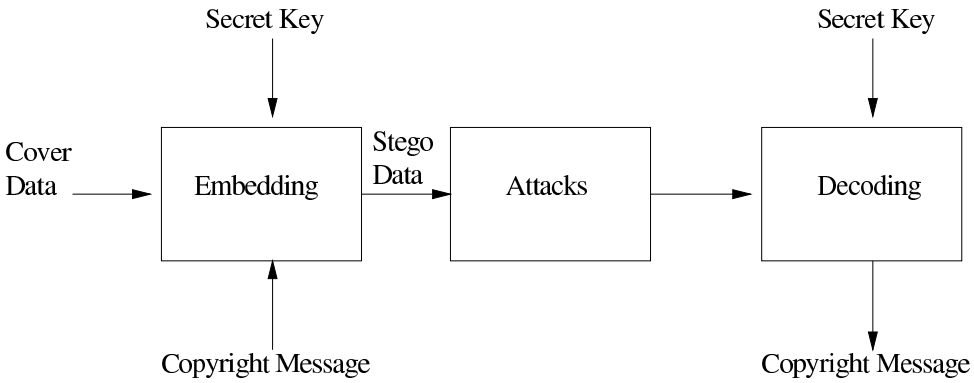
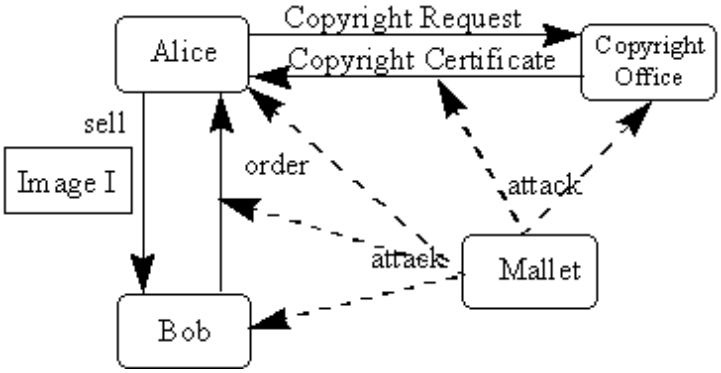


Figure 2.1: Model of Watermarking Process

recovery is important. By "oblivious" we mean that the watermark can be recovered from an image without the original. Clearly in this case the problem is more difficult since if the original is available, a simple subtraction can be performed to separate the image from the watermark. In practice oblivious recovery is an important requirement as otherwise a large search in a database of images may be required to find the original.

The notion of key based embedding and detection is important in order to ensure cryptographic security. Two types of systems exist: private and public. In a private key system only the copyright holder is able to decode the watermark with his secret key. In a public system, the copyright holder embeds the watermark with a private key, but the watermark can be read with a public key. This is useful in situations where we have a public detector. In particular, in certain contexts a public detector may check all images leaving or entering a local network. While appealing at first glance, Linnartz showed that in situations where a public watermark was available, the watermark could be removed without any degradations in image quality [47]. Consequently, we will only consider the case of private keys in which decoding of the watermark is only possible with a private key.

In a typical scenario, a number of transactions must take place in order for an image to be watermarked and securely transferred to the buyer. Figure 2.1 depicts the exchanges that take place between the owner Alice and the buyer Bob during the purchase of an image. Also depicted are the possible points of attack by the malicious Mallet. A formal set of transactions involving cryptographic protocols has been proposed in [78]. A large number of threats are linked with cryptographic security issues. In this thesis we will only be concerned with attacks directly related to the image processing aspects of the image watermarking problem and we will



attempt to develop algorithms which are robust to image processing attacks. We will elaborate on this subset of image processing attacks in section 2.2.2.2.

2.2 Watermarking Requirements

We return to figure 1.1 which summarizes the requirements and tradeoffs involved in image watermarking. The requirements fall into the three categories: robustness, visibility and capacity and in the case of image watermarking we must make compromises between these conflicting requirements.

2.2.1 Capacity

We first consider the capacity requirement since it is the simplest. There are two approaches considered in the current watermarking literature. The bulk of the literature contains 1 bit watermarks where at decoding, hypothesis testing [97] is used to determine if a watermark is present or not in the image. This is sufficient for some copyright applications however many more possibilities are available when the watermark contains 60-100 bits of information. This second class of approaches allows the watermark to contain information such as the buyer and the seller of the image Furthermore additional flag bits may be used to describe the image content. This can be useful in tracking illegal pornographical images over the Web. It is important to distinguish these two classes because algorithms using 1-bit watermarks are not easily extended to higher capacities. The brute force approach of embedding 1 watermark from a set of N possible watermarks and then using hypothesis testing to determine which watermark was embedded is a possible approach which conveys $log_2(N)$ bits of information. However this scheme breaks down when N becomes large since the number of tests which must be performed grows exponentially. We will return to this idea when presenting M-ary modulation [41] and a hybrid scheme in section 4.3 .

2.2.2 Robustness

A second important requirement of watermarking schemes is robustness. Clearly a watermark is only useful if it is resistant to typical image processing operations as well as to malicious attacks. However, it is important to note that the level of robustness required varies with respect to the application at hand. We first consider the required level of robustness of various applications and then categorize various types attacks against which watermarking algorithms must be robust.

2.2.2.1 Watermarking Applications

In [28] Hartung distinguishes the level of robustness required for four categories of applications: authentication, data monitoring, fingerprinting, and copyright protection. In authentication applications, only certain types of attacks and in particular mild compression must be considered. Since ultimately we wish to determine if an image is authentic, we would like the watermark to be destroyed when the data is manipulated. Data monitoring and tracking need higher levels of robustness. These applications required the detection of transmitted or stored media typically for the purposes of billing. Here the watermark should be resistant to compression as well as format conversions. In these first two categories we note that only a 1-bit watermark is needed since we need only make a decision about the presence or absence of a watermark. Consequently the capacity requirement is low. The bulk of the literature contains algorithms which address these two categories of algorithms and many viable solutions based on hypothesis testing [97] exist.

Fingerprinting applications associate receivers of watermarked media to the media itself by inserting a unique user ID and a unique document ID into the media. This provides a mechanism for tracing pirated copies to the person who pirated them. Since in a typical example one image may be sold to many people, we immediately see the need for higher capacity in this application. We need enough bits so that we can have a unique identifier for each image sold. The robustness requirements are also high since the watermark must survive data processing as well as malicious attacks by pirates who attempt to remove the watermark.

Similarly, copyright protection also requires high capacity (60-100 bits in typical commercial systems) [75, 78, 34, 9] and also require a high level of robustness. In addition to data processing and malicious attacks, watermarks for copyright protection must be able to withstand multiple watermarks which may be inserted by pirates in order to create a deadlock situation where the true owner of an image becomes ambiguous [14, 43].

In this thesis we will be concerned with the problem of copyright protection adn oblivious recovery which is the most challenging of the aforementioned problems.

2.2.2.2 Characterization of Attacks

As we have seen, depending on the type application, we require a certain level of robustness against intentional or unintentional attempts to remove the watermark. These "attacks" can be broken down into 4 categories as proposed by Hartung in [29].

The first class of attacks are simple attacks that do not change the geometry of the image and do not make any use of prior information about the watermark. For example these methods do not treat the watermark as noise, but assume the watermark and the host data are inseparable. Attacks in this category include filtering, JPEG and wavelet domain compression, addition of noise, quantization, digital to analog conversion, enhancement, histogram equalization, gamma correction, and printing followed by re-scanning. These attacks attempt to weaken detector response by increasing the noise relative to the watermark.

The second class of attacks are those that disable the synchronization of the watermark detector. This class of attacks includes geometric transformation such as cropping, rotations, scalings, and shearing or general linear transformations. More sophisticated attacks in this category include the removal of pixels, or lines and columns as done in the program UnZign [91]. Even more subtle attacks are performed in the program Stirmark 3.1 [67] where the image is unnoticeably distorted locally by bending and resampling. The main goal of these attacks are to render the watermark unreadable even though it is still present in the modified image.

The third class of attacks are "ambiguity attacks". Here the aim is to create a deadlock where it is unclear which image is original. One example is the insertion of a second watermark by a pirate [35]. Craver [14] introduces the concept of non-invertible watermarking schemes and demonstrates that under certain circumstances a "fake original" can be created. This creates a deadlock situation in which it is impossible to determine the true owner of an image. Another attack which can be placed in this category is the "copy attack" [43]. Here the attacker estimates the watermark from one image and adds it to another image to produce a watermarked image.

The final class of attacks are the "Removal Attacks". In many ways these are the most sophisticated attacks since they take into account prior knowledge of the watermarking process. These attacks attempt to estimate the watermark and then remove the watermark without visible degradation to the host media. Examples are collusion attacks which attempt to get a good estimate the watermark from several watermarked images [84]. Another possibility is denoising where the watermark is modeled as noise [44]. Recently, Voloshynovsky [95] showed that it is possible in some cases to improve the quality of the image while removing the watermark. This is an important result since it demonstrates the power of denoising schemes in performing an accurate separation of watermark and host data.

2.2.3 Visibility

We now turn our attention to the question of visibility. Since watermarking algorithms embed information by modifying the original data, in order to create invisible watermarks, we must have a precise understanding as to the amount pixels can be modified without visible changes in the host image.

Most of the ideas currently used in watermarking with respect to evaluating the visibility of watermarks have been inspired by perceptual coders designed for compression. The key idea is that both watermarking and compression introduce noise in the image and that in both cases we would like to minimize the perceived distortion.

Perceptual coders minimize the error perceived by the HVS. These were introduced since it was found that working with the peak signal to noise ratio (PSNR) criterion and the mean square error (MSE) criteria was inadequate in reducing perceived distortions introduced by compression. Unlike PSNR and MSE which are purely mathematical and global criteria perceptual coders use criteria which attempt to take into account the behavior of the HVS. These coders have been shown to greatly improve the performance of compression algorithms while minimizing visible distortion to the image. For example the JPEG compression scheme quantizes DCT coefficients based on tables which reflect the perceptual impact of each DCT coefficient.

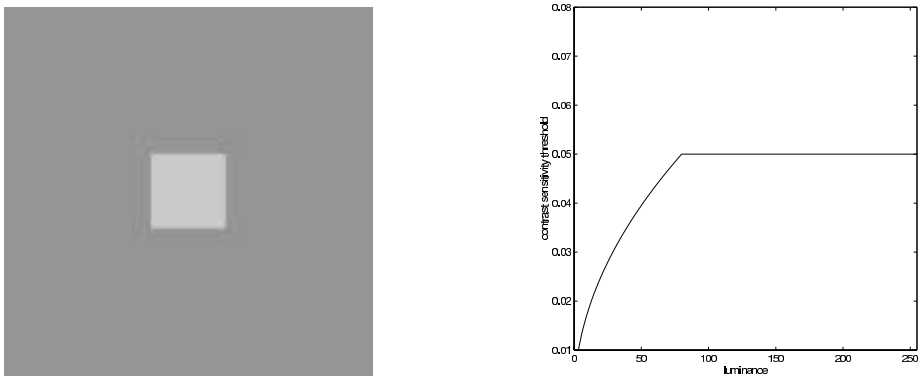
The most recent research on perceptual models incorporate low level vision factors as well as higher level factors. Until the early 1990's, most of the research concerned low level factors. We begin by examining the low level factors in detail and then briefly consider higher level factors which have not yet been seen in watermarking algorithms.

2.2.3.1 Low Level Vision Factors

Jayant [37] gives an overview of perceptual coders and identifies three important low level factors which influence whether noise will be visible in a given part of an image: luminance, frequency, and texture.

- **Luminance:** Jayant states that the HVS is more sensitive in detecting noise for mid-level gray values than for bright and dark regions respectively. However conflicting results have appeared in the literature. A typical experimental setup is given in figure 2.2a where the subject is asked at which point the central square becomes visible against a constant gray level background. All authors agree that at high luminance levels the sensitivity of the HVS follows Weber's law which states that $\frac{\delta l}{l} = k_{Weber}$ where δl is the change in luminance of the central square and l is the luminance of the background. However at lower luminance levels, while Jayant suggests that the HVS is less sensitive to noise, most recent publications suggest that the HVS is more sensitive to noise. Recent publications [57, 58, 77] use the DeVries-Rose law at low luminance level

(typically $< l_{th} = 10cd/m^2$) which states that $\frac{\delta l}{l} = \sqrt{\frac{l}{l_{th}}} * k_{Weber}$. The complete curve is shown in 2.2b. The x-axis contains the luminance between 0 and 255 while the y-axis indicates the contrast sensitivity threshold which is the ratio $\frac{\delta l}{l}$ at which the central square becomes visible.



(a) Experimental setup

(b) Sensitivity function

Figure 2.2: Sensitivity of the HVS to changes in luminance

- **Frequency:** the sensitivity of the eye to various frequencies can be determined by using the same experimental setup where a mid-level grey value is used and the subject is asked to determine the visibility threshold for the central square which contains given spatial frequencies. It has been determined that the eye is most sensitive at the lower frequencies and much less sensitive for high frequencies. Rather than using pure frequencies (sine waves) it is also possible to use the various subbands of a transformation, for example DCT or wavelet domain subbands. The DCT 8×8 block transform has been extensively studied since it is used in the JPEG compression standard [64, 1, 98]. Consequently the "masking" properties are well known. Masking refers to the fact that the presence of a pattern is undetectable to the eye if a similar pattern exists at the same spatial location. In the case of the DCT the patterns we consider are the DCT basis functions. For a given DCT component (i, j) we have the visibility threshold given by [98]:

$$log_{10} t_{ij} = log_{10} \frac{T_{min}}{r_{ij}} + k(log_{10} f_{ij} - log_{10} f_{min})^2 \quad (2.1)$$

$$\text{with:} r_{ij} = r + (1 - r) cos^2 \theta_{ij} \quad (2.2)$$

where

$$T_{min} = \begin{cases} \frac{L}{S_0} & if L > L_T \\ \frac{L}{S_0}(\frac{L_T}{L})^{1-a_t} & if L \leq L_T \end{cases} \tag{2.3}$$

$$k = \begin{cases} k_0 & if L > L_k \\ k_0(\frac{L}{L_k})^{a_k} & if L \leq L_k \end{cases} \tag{2.4}$$

$$f_{min} = \begin{cases} f_0 & if L > L_f \\ f_0(\frac{L}{L_f})^{a_f} & if L \leq L_f \end{cases} \tag{2.5}$$

where $L_T = 13.45 cd/m^2, S_0 = 94.7, a_t = 0.649, L_k = 300 cd/m^2, k_0 = 3.125, a_k = 0.0706, L_f = 300 cd/m^2, f_0 = 6.78 cycles/deg, r = 0.7$ and $a_f = 0.182$. Also,

$$f_{ij} = \frac{1}{16}\sqrt{(i/W_x)^2 + (j/W_y)^2} \tag{2.6}$$

where W_x and W_y are the horizontal and vertical size of a pixel in degrees of visual angle. The angular parameter is given by:

$$\theta_{ij} = arcsin\frac{2f_{i0}f_{0j}}{{f_{ij}}^2} \tag{2.7}$$

The parameters were determined by extensive subjective tests and are now widely adopted.

It has now been established that there is an important interaction between luminance and frequency which Watson incorporates in the model by setting

$$t_{ijk} = t_{ij}(\frac{c_{00k}}{\overline{c}_{00}})_t^a \tag{2.8}$$

where c_{00k} id the DC coefficient of block k, $\overline{c}_{00}$, and a_t determines the degree of masking (set to 0.65 typically).

- **Texture :** Texture masking refers to the fact that the visibility of a pattern is reduced by the presence of another in the image. The masking is strongest when both components are of the same spatial frequency, orientation and location. Watson extends the results of luminance and frequency masking presented above to include texture masking. This is done by setting:

$$m_{ijk} = Max[t_{ijk}, |c_{ikj}|^{w_{ij}} {t_{ijk}}^{1-w_{ij}}] \tag{2.9}$$

where m_{ijk} is the masked threshold and w_{ij} determines the degree of texture masking. Typically $w_{00} = 0$ and $w_{ij} = 0.7$ for all other coefficients.

From these three factors a perceptual error metric can be derived which yields an indicator of image quality after modifications to the image. This can be computed by first setting

$$d_{ijk} = \frac{e_{ijk}}{m_{ijk}} \tag{2.10}$$

where e_{ijk} is the error in a given DCT coefficient in a given block k. We then use Minkowski summation to obtain the quality metric:

$$d(I, \tilde{I}) = \frac{1}{N^2}[\sum_{ij}(\sum_{k} {d_{ijk}}^{\beta_s})^{\frac{\beta_f}{\beta_s}}]^{\frac{1}{\beta_s}} \tag{2.11}$$

with $\beta_s = \beta_f = 4$.

The model presented above has been developed by extensive testing and has been adapted for use in conjunction with the DCT 8×8 transform. Recently Podilchuk [71] used such a model in a DCT domain watermarking scheme which demonstrated the applicability of such models to the problem of watermarking. By contrast, recently a spatial domain mask has been derived based on a stochastic modelling of the image [94]. Since this mask will be used with the algorithms developed in this thesis, we consider the model in detail.

One of the most popular stochastic image models, which has found wide application in image processing, is the *Markov Random Field (MRF)* model [24]. The distribution of MRF's is written using a Gibbs distribution:

$$p(x) = \frac{1}{Z} e^{-\sum_{c \in A} V_c(x)}, \tag{2.12}$$

where Z is a normalization constant called the *partition function*, $V_c(\cdot)$ is a function of a local neighboring group c of points and A denotes the set of all possible such groups or cliques. An important special case of this model is the Generalized Gaussian (GG) model. Assume that the cover image is a random process with non-stationary mean. Then, using autoregressive (AR) model notation, one can write the cover image as:

$$x = A \cdot x + \varepsilon = \bar{x} + \varepsilon, \tag{2.13}$$

where $\bar{x}$ is the non-stationary local mean and ε denotes the residual term due to the error of estimation. The particularities of the above model depend on the assumed stochastic properties of the residual term:

$$\varepsilon = x - \bar{x} = x - A \cdot x = (I - A) \cdot x = C \cdot x, \tag{2.14}$$

where $C = I - A$ and I is the unitary matrix. If A is a low-pass filter, then C represents a high-pass filter. We note that the same distribution models the error term as the cover image with the only difference being the local mean $\bar{x}$.

This type of model has found widespread applications in image restoration and denoising [52, 8] as well as in some recent wavelet compression algorithms [48, 50]. Here we use the stationary Generalized Gaussian (GG) model for the residual term ε. The advantage of this model is that it takes local features of the image into account. This is accomplished by using an energy function, which preserves the image discontinuities under stationary variance.

The auto-covariance function for the stationary model is equal to:

$$R_x = \begin{pmatrix} \sigma_x^2 & 0 & \cdots & 0 \\ 0 & \sigma_x^2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sigma_x^2 \end{pmatrix}, \tag{2.15}$$

where σ_x^2 is the global image variance. The model can be written as:

$$p_x(x) = \left(\frac{\gamma\eta(\gamma)}{2\Gamma(\frac{1}{\gamma})}\right)^{\frac{N}{2}} \cdot \frac{1}{|\det R_x|^{\frac{1}{2}}} \cdot \exp\{-\eta(\gamma)(|Cx|^{\frac{\gamma}{2}})^T R_x^{-\frac{\gamma}{2}} |Cx|^{\frac{\gamma}{2}}\}, \tag{2.16}$$

where $\eta(\gamma) = \sqrt{\frac{\Gamma(\frac{3}{\gamma})}{\Gamma(\frac{1}{\gamma})}}$ and $\Gamma(t) = \int_0^\infty e^{-u}u^{t-1}du$ is the gamma function, R_x is determined according to (2.15), and the parameter γ is called the *shape parameter*. Equation (2.16) includes the Gaussian ($\gamma = 2$) and the Laplacian ($\gamma = 1$) models as special cases. For real images the shape parameter is in the range $0.3 \leq \gamma \leq 1$.
It has been shown [94] that from the generalized Gaussian model, we can derive a noise visibility (NVF) at each pixel position as:

$$NVF(i,j) = \frac{w(i,j)}{w(i,j) + \sigma_x^2}, \tag{2.17}$$

where $w(i,j) = \gamma[\eta(\gamma)]^\gamma \frac{1}{||r(i,j)||^{2-\gamma}}$ and $r(i,j) = \frac{x(i,j) - \bar{x}(i,j)}{\sigma_x}$.

The particularities of this model are determined by the choice of two parameters of the model, e.g. the shape parameter γ and the global image variance σ_x^2. To estimate the shape parameter, we use the *moment matching* method in [50]. The shape parameter for most of real images is in the range $0.3 \leq \gamma \leq 1$. Once we have computed the noise visibility function we can obtain the allowable distortions by computing:

$$\Delta_{pi,j} = (1 - NVF(i,j)) \cdot S + NVF(i,j) \cdot S_1 \tag{2.18}$$

where S and S_1 are the maximum allowable pixel distortions in textured and flat regions respectively Typically S may be as high as 30 while S_1 is usually about 3. We note that in flat regions the NVF tends to 1 so that the first term tends to 0 and consequently the allowable pixel distortion is at most S_1 which is small. Intuitively

this makes sense since we expect that the watermark distortions will be visible in flat regions and less visible in textured regions. Examples of NVFs for two images are given in figure 2.3. We note that the model correctly identifies textured and flat regions. In particular the NVF is close to 0 in textured regions and close to 1 in flat regions.

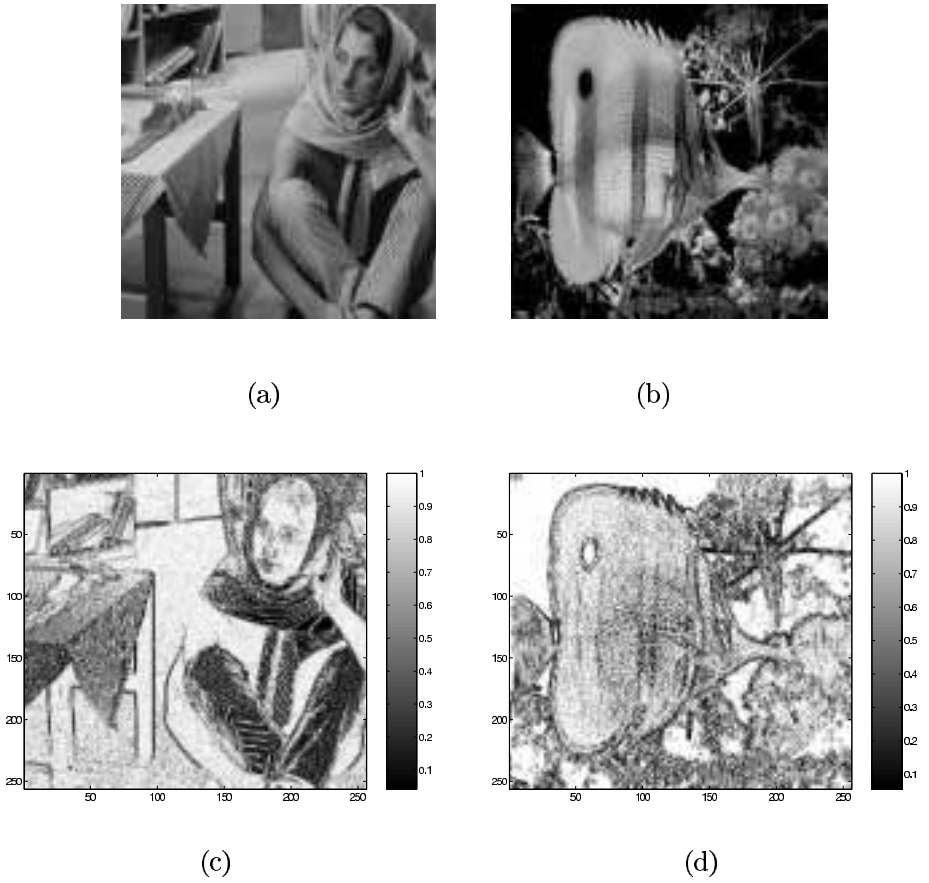


Figure 2.3: Original images of Barbara (a) and Fish (b) along with their NVF as determined by a generalized gaussian model (c) and (d).

2.2.3.2 High Level Vision Factors

We now briefly consider higher level factors which will not be exploited further in this thesis, but may well be the subject of future research. Unlike low level vision factors, high level vision factors require feedback from the brain. In particular, recently much attention has been given to the problem of determining the importance subjects assign to regions in an image. To this end, 8 factors have been identified [57]:

- **Contrast:** Regions which have a high contrast with their surrounds attract our attention and are likely to be of greater visual importance [100].
- **Size:** Larger regions attract our attention more than smaller ones however a saturation point exists after which the importance due to size levels off [22].
- **Shape:** Regions which are long and thin (such as edges) attract more attention than round flat regions [80].
- **Colour:** Some particular colours (red) attract more attention. Further more the effect is more pronounced when the colour of a region is distinct from that of the background [54].
- **Location:** Humans typically attach more importance to the center of an image [20]. This may or may not be important in watermarking applications since we must account for the situation where the image is cropped. When this occurs, the center changes.
- **Foreground/Background:** Viewers are more interested in objects in the foreground than those in the background [6].
- **People:** Many studies have shown that we are drawn to focus on people in a scene and in particular their faces, eyes, mouth and hands [100].
- **Context:** Viewers eye movements can be dramatically changed depending on the instructions they are given prior to or during the observation of an image [100].

Chapter 3

Watermarking as Communications

Having looked at the requirements of a watermarking algorithm, we are now in position to adopt a standard formulation which will serve as a paradigm for analyzing and developing watermarking schemes.

3.1 Watermarking as a Communications Problem

We formulate watermarking as a communications problem as depicted in figure 3.1 [95]. Here the original binary message $\boldsymbol{b}$ is encoded typically by error correction codes

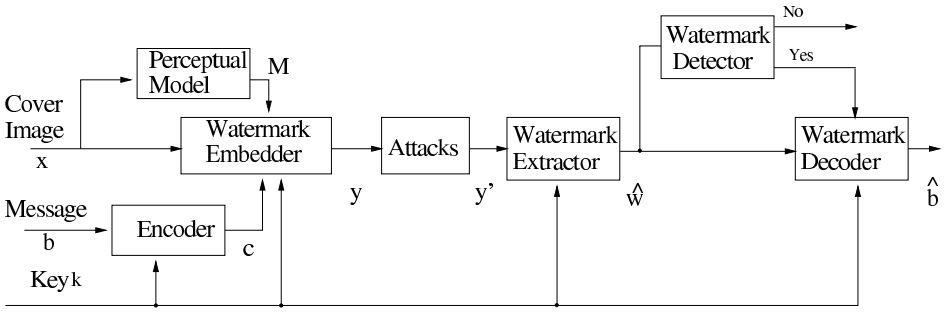


Figure 3.1: Watermarking as a Communications Problem

[30] or M-sequences [89] to produce the coded message $\boldsymbol{c}$. This coded message is then embedded into the cover image $\boldsymbol{x}$ by some embedding function $\boldsymbol{y} = E(\boldsymbol{c}, \boldsymbol{x}, k)$ where k is a cryptographic key. Typically the embedder first takes the coded message $\boldsymbol{c}$ and the perceptual mask $\boldsymbol{M}$ to create a watermark $\boldsymbol{w}$ which is then inserted (usually added or multiplied) into the image either in the spatial domain or in a transform domain. The resulting image $\boldsymbol{y}$ may then undergo attacks which produce the $\boldsymbol{y'}$ from which we would like to recover the watermark. The watermark recovery process in its most general form consists of three steps: watermark extraction (or estimation), detection and decoding. In the first step, we assume that we have $\boldsymbol{y'} = f(\boldsymbol{y}, \boldsymbol{n})$ where

$\boldsymbol{n}$ is the noise and f is a function which models the distortions introduced by the attacker. Our problem is to obtain $\hat{\boldsymbol{w}}$ which is an estimate of the watermark. This estimate is then fed to the detector which yields a decision $H1$ or $H0$ corresponding to watermark detected or watermarked not detected respectively. If the watermark has been detected we proceed to the decoder which attempts to recover the bit sequence from the estimated watermark. As we will see in the rest of the literature survey, not all components are included in every scheme. In fact, early watermarking schemes used very simplified models of the watermarking process. Furthermore most of the schemes in the literature describe one bit watermarks and as such at the watermark recovery stage only the detector (and not the decoder) is needed.

3.2 Coding

In this section, we will describe various types of encoding strategies in order to motivate our choices with respect to the watermarking algorithms we develop. It is natural to start with the encoding process since if we return to our communications paradigm in figure 3.1 we note that prior to embedding the watermark, the binary message we wish to embed can be encoded. This chapter will highlight the fact that important gains can be made by a judicious application of encoding strategies that already exist in the communications literature. Unfortunately to a large extent, the best encoding schemes have been for the most part ignored in favor of schemes known to be suboptimal. We will present three encoding techniques: direct sequence spread spectrum (DSSS) via M-sequences, M-ary modulation, and error correction coding using BCH, convolution and turbo codes.

3.2.1 Spread Spectrum Encoding

Our central problem concerns converting the binary message $\boldsymbol{b}_m$ into a robust form. This can be accomplished by error correction coding, modulation or both. Error correction coding referes to the conversion of the original bit sequence into a longer sequence where the added bits can be used for error detection and correction. Modulation refers to the process of converting each bit into a waveform which is sent across a channel. We begin our description by considering spread spectrum (SS) communications. SS communications were originally developed for military applications. Pickholtz [68] identifies several important properties that are typically sought after:

1. Antijamming.
2. Antiinterference.
3. Low probability of intercept.
4. Multiple user random access communications with selective addressing capability.

In the context of watermarking, these first are directly applicable and have therefore led this community to adopt SS techniques. We consider these four properties in more detail. Typically in communications a band of frequencies may be used to transmit information. If an attacker knows these frequencies, he may attempt to insert a large amount of noise within these frequencies, a process which is known as jamming. The second property refers to the fact that we would like the transmitted signals to be immune to other signals which may be transmitted simultaneously. The third property addresses the fact that it is important that a malicious third party not be able to decode the signal. The fourth property relates to the fact that in watermarking several bits must be transmitted simultaneously. We will see how it is possible to use SS communications in such a way that each bit represents a user.

The key idea of SS communications is to transform a low dimensional signal into high dimensional signal which satisfies a set of desired properties. Here our low dimensional signal is our bit sequence $\boldsymbol{b}_m$ which typically will have $N = 60$ bits. This can be transformed into a signal $\boldsymbol{S}$ of length $L >> N$ where $L = 2^k - 1$ for some k as follows. The binary form of the message $\boldsymbol{b}_m$ is first transformed to obtain the vector $\boldsymbol{b} = (b_1, b_2, ...b_N)^\top$, with $b_i \in \{1, -1\}$ by the simple transformation $(0, 1) \rightarrow (-1, 1)$. For each bit b_i of $\boldsymbol{b}$ we associate a bipolar sequence $\boldsymbol{v}_i$. the encoded message can be obtained by:

$$\boldsymbol{w} = \sum_{i=1}^{N} b_i \boldsymbol{v}_i \tag{3.1}$$

where $\boldsymbol{w}$ is the resulting encoded signal.

The choice of sequences is critical since these sequences determine the properties of the spread spectrum. Many types of binary sequences have been proposed in the literature [27]. One of the most interesting set of sequences, however, is the set of M-sequences [26]. This sequence set has optimal auto-correlation properties. In particular for the signal set of M-sequences composed of $\boldsymbol{v}$ (after applying the mapping $(0, 1) \rightarrow (-1, 1)$) and all its cyclic shifts, we have:

$$< \boldsymbol{v}_i, \boldsymbol{v}_j >= \begin{cases} L & i = j \\ -1 & i \neq j \end{cases} \tag{3.2}$$

From this property we immediately see the decoding strategy which consists of calculating the inner products $< \boldsymbol{w}, \boldsymbol{v}_i >$, taking the sign and performing the inverse mapping $(-1, 1) \rightarrow (0, 1)$ in order to obtain the original sequence.

One possible way of generating such sequences consists of sampling the outputs of a linear feedback shift register (LFSR). For M-sequences, the code sequence $\boldsymbol{v}$ satisfies the recurrence relation:

$$\boldsymbol{v}_n = \sum_{k=1}^{r} a_k \boldsymbol{v}_{n-k} \mod 2; \qquad a_r = 1 \tag{3.3}$$

where a_k are a set of binary coefficients and r is the number of registers. The choice of a_k determines the properties of the resulting sequences. The structure for the LFSR corresponding to the recurrence relation in equation 3.3 is given in figure 3.2.1. In

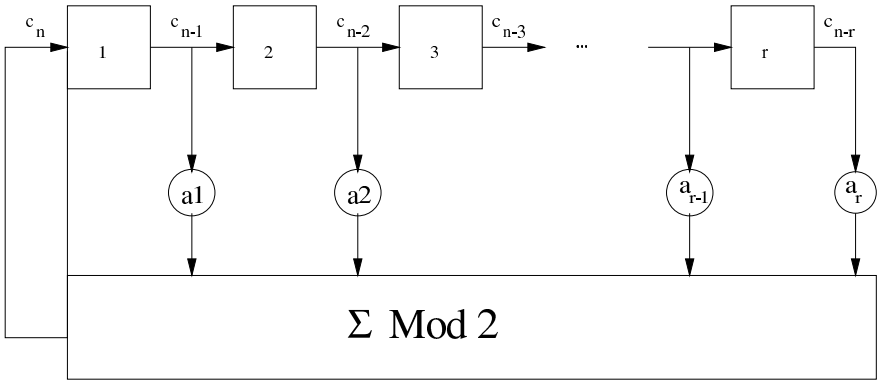


Figure 3.2: Linear Feedback Shift Register

practice the most useful sequences are the maximal length sequences. Maximal length refers to the fact that the period of the LFSR is equal to $2^r - 1$. An example for the case with $r = 4$ is given in figure 3.2.1. Here we have that $a_1 = 1$ and $a_4 = 1$ while a_2 and a_3 are 0. The state of each register over a whole period is given at the right of the figure. The code words $\boldsymbol{v}$ can be any column containing 15 elements. It is easily verified that the inner products between cyclic shift of a codewords satisfy equation 3.2. Methods of obtaining the coefficients a_k are given in [68] and are based on finding minimal poynomials over GF(2) where GF is a Galois Field. One important point is that the sequence of vectors obtains depends on the initial state of the registers. This state can be key-based which provides cryptographic security In particular, we note that an attacker cannot decode or detect the signal since he cannot generate the underlying sequences.

Having considered the properties and generation of M-sequences, we are now ready to consider the resistance in the presence of noise. In order to obtain a closed form solutions, we will assume additive white Gaussian noise (AWGN). We also note that in watermarking applications sequences of length about $L = 2^{13} - 1$ are used [15] and about $N = 60$ bits are encoded. For this case we obtain our encoded vector $\boldsymbol{w}$ from equation 3.1. We assume now that at the receiver we obtain

$$\boldsymbol{w}' = \boldsymbol{w} + \boldsymbol{n} \tag{3.4}$$

where $\boldsymbol{n}$ is an AWGN vector of independently and identically distributed samples (i.i.d.) with variance σ^2. In order to decode we calculate the inner products $< \boldsymbol{w}', \boldsymbol{v}_i >$. We first note that since we take M large (for watermarking L is about 8191), we can neglect the interference between $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ which results from the fact

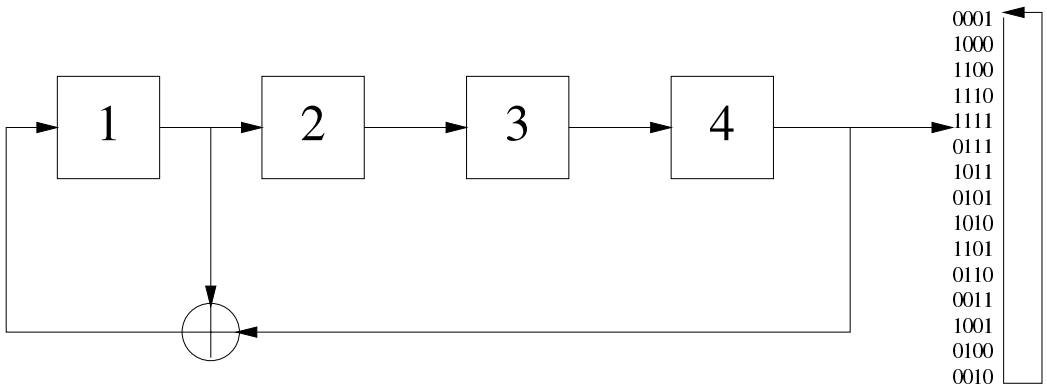


Figure 3.3: Maximal Length Sequence Generation

that the inner product is -1 and not 0 for the case where $i \neq j$. Consequently we conclude that a bit will be falsely decoded if $< \boldsymbol{n}, \boldsymbol{v}_i >$ is greater than L. By the central limit theorom, we obtain a random variable whose mean is 0 and whose variance is $\sigma'^2 = L\sigma^2$. The probability of error P_e is obtained immediately as:

$$\int_L^\infty \frac{1}{\sqrt{2\pi\sigma'^2}} \exp(\frac{-x^2}{2\sigma'^2})dx \tag{3.5}$$

By defining,

$$erfc(x) = 2\pi^{\frac{-1}{2}} \int_x^\infty \exp(-t^2)dt \tag{3.6}$$

we obtain

$$P_e = \frac{1}{2} erfc(\frac{L}{\sigma'\sqrt{2}}) \tag{3.7}$$

In practice we are interested in knowing the probability of error for a given SNR. Since we have considered a bipolar normalized signal with variance=1 we immediately obtain the SNR as $10log_{10}\frac{1}{\sigma^2}$. In figure 3.2.1 we plot the probability of error versus the SNR for a 1 bit message. We note that as the SNR goes up the error probability goes to 0, while as the SNR goes down, the error probability tends to 0.5. This latter case corresponds in fact to a random result obtained at the decoder. The extension to the multiple bit case is straightforward. An important observation is that in watermarking applications we require the decoded sequence to be recovered perfectly (i.e. BER=0). Consequently, for this modulation scheme to be applicable, we would require an SNR of no less than -30dB. Since a useful watermark contains between 60 and 100 bits of data, we must also consider the probability of error in this case. The extension to the multiple bit case is straightforward. In the 60 bit case the

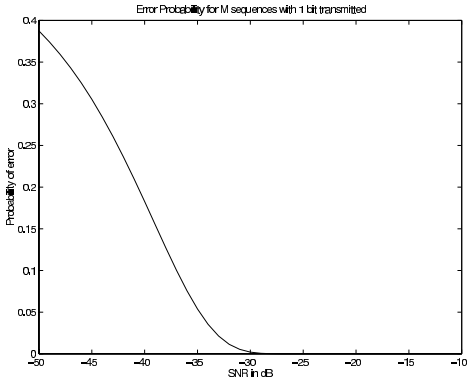


Figure 3.4: Error Probability for Msequences with a 1 bit message

probability of correctly decoding all the bits is:

$$P_e = (1 - \frac{1}{2} erfc(\frac{L}{60\sigma'\sqrt{2}}))^{60} \tag{3.8}$$

The factor 60 multiplying σ' arises from the fact that in order to fairly compare the 2 situations (1 bit versus 60 bits), we must normalize the noise variance. A plot of this error probability as a function of the SNR is given in figure 3.2.1. We see from the graph that in this case we require an SNR of at least -9dB to be sure of correctly decoding all the bits. Intuitively this makes sense since we require more energy to send more bits reliably. In this section we have presented M-sequences as a means of

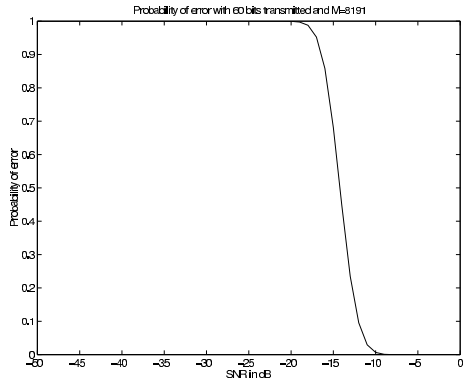


Figure 3.5: Probability of correct decoding for a 60 bit message

modulating a set of bits. While this method is an extremely popular one because of the properties discussed above, other choices are possible. These include: sine waves, Hadamard basis functions or other orthogonal signal sets. The primary advantage of M-sequences is the fact that they provide cryptographic security. This is not the case for typical sets of orthogonal functions such as DCT or DFT basis functions.

3.2.2 Capacity

Having considered the problem of modulation, one natural question to ask is if the scheme described in the previous section to send 60 bits is optimal. To answer this question we must first consider the problem of channel capacity. One of the most celebrated results in information theory is the capacity of the Gaussian Channel first calculated by Shannon as [82]:

$$C = \frac{1}{2}\log_2(1 + SNR) \tag{3.9}$$

where C is the maximum number of information bits per transmission that can be sent with an arbitrarily small probability of error if signals of infinite length and arbitrarily complex coding schemes are used. Since we use relatively long signals, the capacity from equation 3.9 provides a good starting point for evaluating the performance of various coding strategies.

In our case the length of our resulting encoded vector $\boldsymbol{w}$ which we denoted as M represented the number of available transmissions so that we obtain:

$$C_{wm} = \frac{M}{2}\log_2(1 + SNR_{wm}) \tag{3.10}$$

where C_{wm} is the capacity of the image. A graph of this function appears in figure 3.2.2. From the graph we see that it is possible to transmit 60 bits of information at an SNR of -20dB. This suggests that some optimization is possible since by using the method presented in the previous section we required -9dB for reliable transmission.

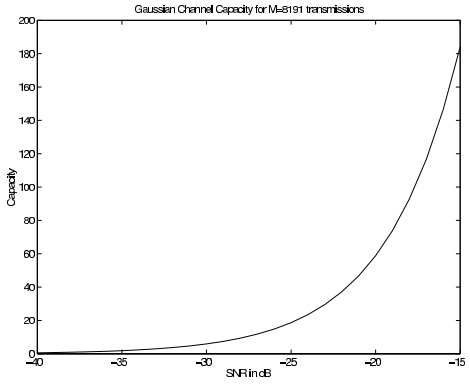


Figure 3.6: Gaussian Channel Capacity

The question of how to design effective coding schemes which yields results which are closer to capacity has been extensively studied in the communications literature. Here we will consider two possibilities which have found applications in watermarking: M-ary modulation and error correctoin coding.

3.2.3 M-ary Modulation

Since we cannot reach capacity by adopting the scheme in equation 3.1, we must attempt to adopt more sophisticated coding or modulation strategies. One possibility is the use of M-ary modulation which has been proposed by Kutter [41] in the context of image watermarking. The key idea is to increase the number of possible waveforms being transmitted, but to send only a small subset of the waveforms with more energy transmitted in each waveform. This can be done by grouping $log_2(M)$ bits of the original message and taking the corresponding decimal value as an index into a set of possible basis functions. Here M is the number of basis functions used. The encoding process is depicted in figure 3.2.3.

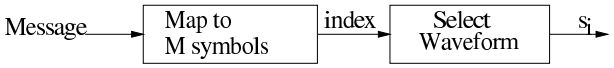


Figure 3.7: M-ary encoding

The detection consists of calculating the correlation between each basis functions and the received waveform to yield r_i and choosing the decimal value which corresponds to the highest correlation. This is depicted in figure 3.2.3. In practice bi-orthogonal signalling is used. Bi-orthogonal signalling consists of modulating a set of waveforms so that for each waveform s_i we send either s_i or $-s_i$. By doing this, we immediately reduce the number of correlations by a factor of two. In equation

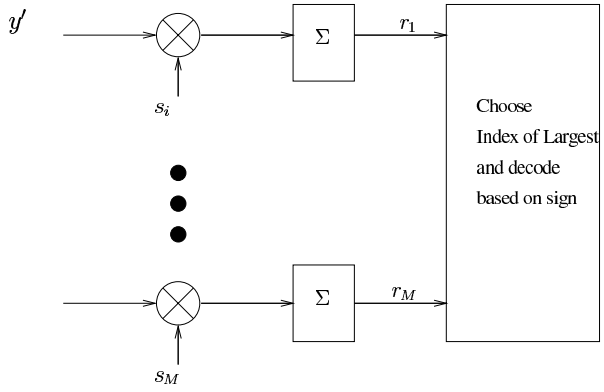


Figure 3.8: M-ary Decoding

3.1, we used $M = 2$ basis functions. By increasing M, a superior performance is obtained. The performance of M-ary modulation schemes with bi-orthogonal signalling has been investigated by Kutter [41]. Kutter shows that the probability of correctly

decoding the signal is given by:

$$P_c = \frac{1}{2\pi\sigma'^2}\int_0^{\infty} \exp(-\frac{x^2}{2\sigma'^2})[1 - 2erfc(x/\sigma')]^{M/2-1}dx \tag{3.11}$$

If we assume that signal s_1 was sent then the first term in the integral corresponds to the probability that r_1 was greater than 0 while the second term corresponds to the probability that the magnitude of all other r_i were less than the magnitude of r_1. In figure 3.2.3 we plot the probability of false detection versus SNR for various values of M. We note in particular that as M increases, the probability of error goes to 0 for a lower value of the SNR which means that increasing the number of symbols leads to better performance of the system.

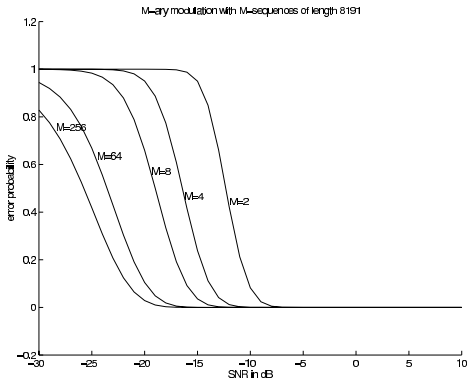


Figure 3.9: M-ary System Performance

It has been shown [72] that as M goes to infinity the system performance achieves capacity. However, the decoding becomes extremely complex since M correlations must be performed. In the case of 64 bits, in order to achieve capacity, we would require $M = 2^64$ signals which corresponds to $1.8 * 10^{19}$ correlations which is clearly unacceptable. In practice roughly $M = 64$ signals are used.

3.2.4 Error Correction Coding

Since capacity can only be achieved at the expense of exponentially complex decoding in the case of M-ary modulation, it is natural to ask if other roads to capacity are available. In fact another possibility exists in the form of error correction coding (ECC). Here the aim is to take the original k bit message and transform it into an n bit message where $k < n$. The extra bits add redundancy and will be used at decoding to correct errors. In order for the scheme to be effective the energy invested in the extra bits must be more than compensated by the number of errors being corrected. With respect to watermarking, the well known BCH codes have been the

ones that have been adopted most often [32]. These codes can in some cases yield a 6dB improvement over methods without coding.

One problem with BCH codes is that they employ hard decoding. This refers to the fact that the received signal is first quantized to a binary signal before decoding. It is now known that this inevitably incurs at least a 3dB loss in performance. Consequently a more attractive solution consists of using convolution codes. Convolution codes allow for the use of the well known Viterbi maximum likelihood algorithm [76] which allows for *soft* decisions at decoding. Here *soft* refers to the fact that we do not quantize before decoding. While the best convolution codes offer up to a 3dB gain over BCH codes, capacity is still not achieved. It is only recently, with the advent of turbo codes developed by Berrou [5] that a performance close to capacity was achieved. Turbo codes consists of two convolution coders and a random bit interleaver as depicted in figure 3.2.4. In order to perform the decoding, two Viterbi decoders are

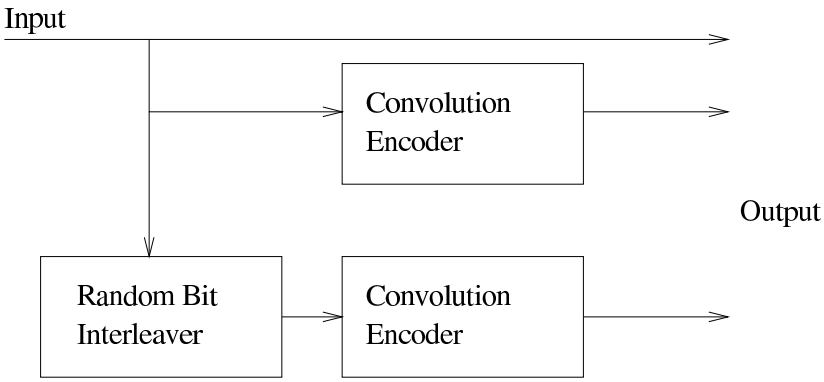


Figure 3.10: Turbo Encoding

used iteratively such that the *a posteriori* outputs are sent to the other decoder while the corresponding inputs are used as *a priori* estimates. This yields a suboptimal yet excellent performance. Unfortunately turbo codes are still not well understood theoretically and consequently this domain remains an active research area. The decoding process is depicted in figure 3.2.4. The main advantage over M-ary modulation is that turbo codes offer a reasonable decoding complexity. In practice the codes are decoded in seconds for a 64 bit sequence. Since turbo codes offer the best performance over Gaussian channels we will use them in the algorithms presented in this thesis even though the channels will not always be Gaussian. This will necessarily imply losses however very little work has been done in coding for Gaussian channels, and a study of the subject is a thesis topic in its own right.

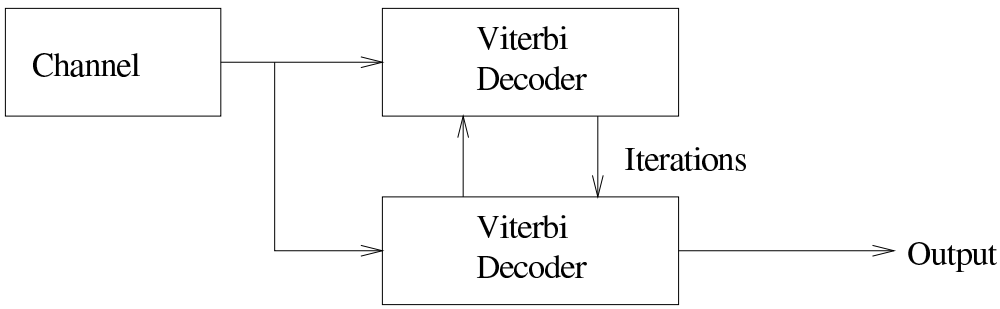


Figure 3.11: Decoding of Turbo codes

Chapter 4

Algorithm Review

Having presented the basic communications formulation and having developed several coding strategies we now turn our attention to a review of watermarking algorithms that have appeared in the literature. Due to the variety of algorithms, no unique way to classify the various approaches exists. We choose to detail the important class of linear watermarks first. We then follow the approach adopted by Cox [12] who describes three watermarking categories which can be used to classify all algorithms. Within this framework, algorithms are categorized relative to the type of *embedding* strategy adopted (linear or otherwise). We note that the historical evolution of algorithms has more or less followed these categories starting from the oldest Category I algorithms to recent Category III.

4.1 Linear Watermarking

4.1.1 Message embedding

A message $b = (b_1, ..., b_L)$ is to be embedded in the cover image $x = (x_1, ..., x_N)^T$ of size $M_1 \times M_2$, where $N = M_1 \cdot M_2$. The message b contains information about the owner and can be used for authentication purposes. To convert the message into a form efficient for communication, it is encoded using either error correction codes (ECC) or modulated using binary antipodal signaling [32] or M-ary modulation [41]. With respect to ECC, mostly Bose Chaudhuri (BCH) or convolutional codes are used [55, 31]. Recent publications [62, 93] report the successful results using novel Turbo codes and low-dencity parity-check (LDPC) codes in the DCT and wavelet domains. In the general case, the type of ECC and the set of basis functions for M-ary modulation can be key-dependent. The above conversion is performed in the encoder that produces the codewords $c = Enc(b, Key)$, $c = (c_1, ..., c_K)^T$ which are mapped from {0,1} to {-1,1}.

A watermark w is created by some key-dependent function $w = \varepsilon(c, p, M, Key)$ that ensures the necessary spatial allocation of the watermark based on a key-dependent

projection function p, and according to HVS features as expressed by a perceptual mask M in order to improve the watermark. The typical choice for the projection function p is a set of two dimensional orthogonal functions used for every codeword bit $\{c_k\}$ such that the empty set is formed by the intersection $P_k \cap P_l, \forall k \neq l$ [41, 32][2]. The projection function performs a "spreading" of the data over the image area. Moreover, the projection function can have a particular spatial structure with given correlation properties that can be used for the recovery of affine geometrical transformations [40, 93]. The resulting watermark is obtained as the superposition

$$w(j) = \sum_{k=1}^{K} c_k p_k(j) M(j) \tag{4.1}$$

where $j \in Z$. The watermark embedder performs the insertion of the watermark into the cover image in some transform or coordinate domain, yielding the stego image:

$$y = T^{-1}[h(T[x], w)] \tag{4.2}$$

where T is any orthogonal transform like block DCT, full-frame FFT and DCT, wavelet or Radon transforms ($T = I$ for the coordinate domain), and $h(.,.)$ denotes the embedding function. The most used class of embedding functions conforms to the linear additive model

$$y = h(x, w) = x + w \tag{4.3}$$

that is considered in this paper.

4.1.2 Attacking channel

An attacking channel produces the distorted version y' of the stego image y. The attacking channel can be modelled in the framework of stochastic formulation using a probabilty mass function (p.m.f) $Q(y'|y)$ to describe random distortions in the stego image. The successful attack should damage or destroy watermark while preserving its commercial quality. Therefore, an attacker should introduce distortions that are bounded by some upper allowable bound according to the chosen distrotion criterion. Althougth, the MMSE is not perfectly matched with the subjective human assestment of image quality, it is commonly used due to the obtained tractable results and the wide usage of this criteria in the comminication community due to the known results for the additive Guassian channels. Therefore, the aim of the attacker consists in the decrease of the rate of reliable communication subject to the allowable distortion.

4.1.3 Message extraction

The recovery processconsists of the watermark extractor and decoder which are described below.

4.1.3.1 Watermark extractor

The watermark extractor performs an estimate $\widehat{w}$ of the watermark based on the attacked version $\widehat{y}$ of the stego-image:

$$\widehat{w} = Extr\left(T\left[y'\right], Key\right) \tag{4.4}$$

In the general case, the extraction should be key-dependent. However, the desire to recover data after affine transformation based on the above mentioned self-reference principle, and the opportunity to enhance the decoding performance by reducing the variance of the image considered as noise [41, 30], have motivated the development of key-independent watermark extraction methods. They could represent the main danger to linear additive watermarking technologies, as will be shown below.

Different methods are used for watermark estimation, such as the cross-shaped filter [40], or MMSE estimates [32]. In the most general case, the problem of watermark estimation can be solved based on a stochastic framework by using Maximum Likelihood (ML) or MAP estimates [94]. Assuming that both the noise due to the cover image and the noise introduced by an attack can be considered additive with some target distribution $p_X(.)$, one can determine the ML-estimate:

$$\widehat{w} = \arg\max_{\widetilde{w}\in\Re^N} p_X\left(y' \mid \widetilde{w}\right) \tag{4.5}$$

which results either in a local average predictor/estimator in the case of a locally stationary independent identically distributed (i.i.d.) Gaussian model of $p_X(.)$, or a median predictor in case of a corresponding Laplacian p.d.f.. The MAP estimate is given by:

$$\widehat{w} = \arg\max_{\widetilde{w}\in\Re^N}\left\{ p_X\left(y' \mid \widetilde{w}\right) \cdot p_W\left(\widetilde{w}\right)\right\} \tag{4.6}$$

where $p_W(.)$ is the p.d.f. of the watermark. Assuming that the image and watermark are conditionally i.i.d. locally Gaussian, i.e. $x \sim N\left(\overline{x}, R_x\right)$ and $w \sim N\left(0, R_w\right)$ with covariance matrices R_x and R_w, where R_w also includes the effect of perceptual watermark modulation, one can determine:

$$\widehat{w} = \frac{R_w}{R_w + R_x}\left(y' - \overline{y}'\right) \tag{4.7}$$

where it is assumed $\overline{y}' \approx \overline{x}$, and where $\widehat{R}_x = \max\left(0, \widehat{R}_y - R_w\right)$ is the ML estimate of the local image variance $\left(\widehat{R}_x = \sigma_x^2 I\right)$.

4.1.3.2 Watermark decoding

In the general case the decoder/demodulator design is based on ML or MAP approaches. Since the appearance of b is assumed to be equiprobable and due to the

high complexity of the MAP decoders, ML decoders are mostly used in practice. The watermark decoder can be considered to consist of two main parts: a matched filter (detector) that performs a despreading of the data in the way of "coherent accumulation" of the sequence c spread in the watermark w, and the decoder itself that produces the estimate of the message. In most cases the results of attacks and of prediction/extraction errors are assumed to be additive Gaussian. The detector is therefore designed using an ML formulation for the detection of a known signal (projection sets are known due to the key) in Gaussian noise, that results in a correlator detector with reduced dimensionality:

$$r = \langle \widehat{w}, p \rangle . \tag{4.8}$$

Therefore, given an observation vector r, the optimum decoder that minimizes the conditional probability of error assuming that all codewords b are equiprobable is given by the ML decoder:

$$\widehat{b} = \arg\max_{\widetilde{b}} p\left(r \mid \widetilde{b}, x\right). \tag{4.9}$$

Based on the central limit theorem (CLT) most researchers assume that the observed vector r can be accurately approximated as the output of an additive Gaussian channel noise [41, 30].

4.2 Category I watermarks

The scheme for Category I watermarks is presented in figure 4.1. The watermarking process consists of generating a watermarking, setting a global strength and then adding the result to the image which produces the stego image. It is important to note than here the strength is set globally and independently of the image.

Several early schemes fall into this framework. In [89] Tirkel proposes adding an M-sequence to the least significant bit of the each image pixel. M-sequence are bipolar sequences which have excellent autocorrelation properties [49]. The decoding exploits these well properties. In an improved version, the watermark was embedded by inserting 2-D M-sequences [90].

Another idea that appeared with several variants was the "Patchwork" algorithm developed by Bender [4]. The algorithm consists of selecting random pairs of pixels (a_i, b_i) and increasing the a_i's by one and decreasing the b_i's by one. The watermark is detected by comparing the sum of the differences between the a_i's and b_i's to a threshold which is chosen so that the probability of false detection is below a certain level. This is in fact a hypothesis testing problem which was formalized within the watermarking community by Pitas [69]. The watermarking scheme Pitas describes consists of embedding a watermark which is a binary pattern $S = s_{m,n}$ the same size of the original image. The image is divided into two sets A and B and a positive

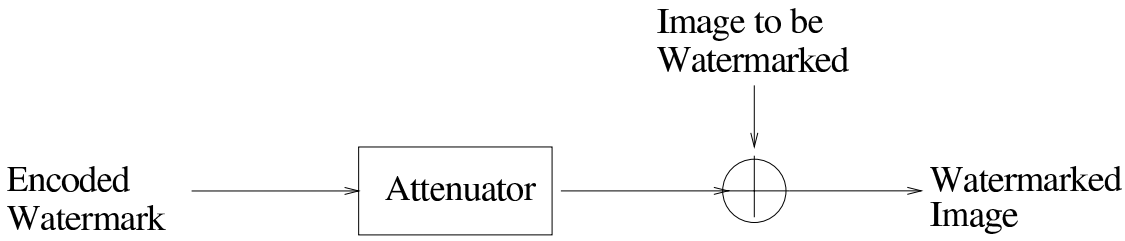


Figure 4.1: Category I Watermark

factor k is added to all elements of A. To detect the watermark, the test statistic q defined below is compared against a threshold.

$$q = \frac{\overline{b} - \overline{a'}}{\sigma^2_{A'} + \sigma^2_B} \tag{4.10}$$

where $\overline{b}$ and $\overline{a'}$ are the means of the two set (the prime is used since the A set has been modified) and $\sigma^2_{A'}$ and σ^2_B are the variances. We note that in comparison to Tirkel's scheme, the schemes of Bender and Pitas are one bit watermarks since we are only interested in determining if a watermark is present or not. In Tirkel's scheme, several bits can be encoded via the M-sequences.

The methods of Pitas and Bender have been refined so that multiple bits can be encoded. The main idea, proposed by Langelaar [46, 45] consists of subdividing the image into blocks and for each block and then adding or subtracting kP depending on the bit value from each block. Here k is a scaling factor and P is a random binary matrix. The bits are decoded by calculating the differences between the means in the sets I_1 and I_0 where I_1 and I_0 are the sets corresponding to locations in the matrix P containing 1 or 0 respectively.

The initial watermarking algorithm developed by Digimarc is also an example of a category I algorithm. The idea described in the patent [75] consists of assigning a pseudo-random noise sequence of the same size as the image to each of the bits being embedded. The noise patterns are then added to the image. Decoding is done by crosscorelating each pseudo-random noise pattern with the image to obtain each bit.

4.3 Category II Watermarks

The scheme for Category II watermarks is presented in figure 4.2. The main improvement over Category I watermarks lies in the fact that the image is now used to generate a perceptual mask. The watermark is then generated in accordance with this mask so as to embed most strongly in regions where the watermark will be visible and less strongly in regions where the watermark will be readily seen. The bulk of the current literature describes watermarking methods that fall into this category. In what follows we consider separately spatial domain schemes and transform domain schemes.

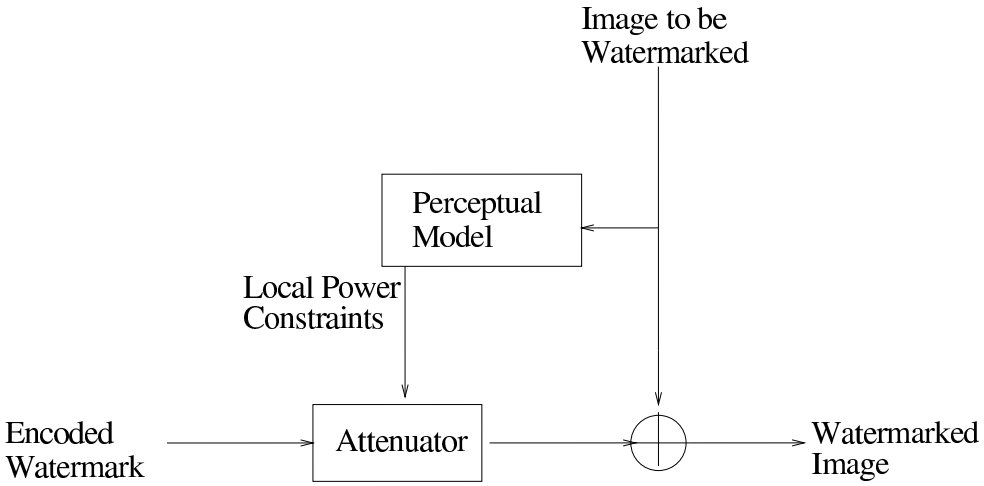


Figure 4.2: Category II Watermark

4.3.1 Spatial Domain Category II Watermarking

One of the first schemes spatial domain watermarks to fall into category II was developed by Goffin [25]. The watermark consisted of a spatial domain binary pattern which is low-pass filtered, frequency modulated, masked and then added to the host data. The masking process uses phenomena based on gratings (typically sine wave gratings). This is similar to the frequency masking described in section 2.2.3. Watermark detection is accomplished by demodulation followed by correlation and comparison with a threshold.

In a fundamentally different approach, Kutter proposes the use of luminance masking in the blue channel [41]. It has been shown that the HVS is less sensitive to the blue channel than the red and green components. Kutter embeds a binary number through amplitude modulation in the spatial domain. A single bit is embedded at a pseudo-randomly selected location (i, j) by either adding or subtracting, depending

on the bit, a value which is proportional to the luminance at the same location:

$$B_{i,j} \leftarrow B_{i,j} + \alpha(-1)^b L_{i,j} \tag{4.11}$$

where $B_{i,j}$ describes the blue value at location (i,j), $L_{i,j}$, the luminance at the same location and α , the embedding strength. To recover an embedded bit, an estimate of the original, non-watermarked, value is computed using a linear combination of neighboring pixels in a cross shape

$$\hat{B}_{i,j} = \frac{1}{4c}(\sum_{k=-c}^{c} B_{i+k,j} + \sum_{k=-c}^{c} B_{i,j+k} - 2B_{j,k}) \tag{4.12}$$

where c defines the size of the cross-shaped neighborhood. The bit value is determined by looking at the sign of the difference $\delta_{i,j}$ between the pixel under inspection and the estimated original. The idea of estimating the watermark prior to decoding is a novel idea which considerably improves watermark robustness. Returning to figure 3.1, we note that the cross-shaped filtering described by Kutter is an example of watermark extraction. While Kutter adopts a heuristic approach with the use of cross-shaped filters, the problem of watermark estimation has recently been more formally dealt with as a MAP estimation problem in image denoising where probability density function are used to model the image and watermark in order to improve the estimation [95, 93].

Another idea which Kutter proposes to improve the robustness of watermarks is the use of M-ary modulation. We note that this is only applicable where multiple bits are being embedded. The idea is to associate a signal $\boldsymbol{s}_i$ for each possible message $\boldsymbol{m}_i$. For the optimal performance, bi-orthogonal signals should be used. The decoding consists of calculating all inner products $< \boldsymbol{s}_i, \hat{\boldsymbol{w}} >$ and selecting the message $\boldsymbol{m}_i$ which corresponds to the maximum inner product. It is well known within the communications community that this approach yields the optimal performance of the system [73]. In practice, the complexity of the decoder is too great since the number of messages is 2^{n_b} where n_b is the number of bits in the message. Consequently, in practice the message is broken into small subsets (typically 8 bits) and M-ary modulation is applied to each subset.

4.3.2 Transform Domain Category II Watermarking

To this point, only spatial domain methods have been investigated. Adaptive Watermarking may also be carried out in the frequency domain. The most popular transforms are the DCT, DFT, and Wavelet transforms. There are several motivations for watermarking in a transform domain. Firstly certain transforms are intrinsically robust to certain transformations. For example it is easy to construct watermarks in the DFT domain which are robust against cropping as will be seen in chapter 5. Secondly, the most popular compression schemes operate in transform domains, for

example JPEG in the DCT domain and EZW in the wavelet domain. By matching the domain of the watermark with the domain of the compression it is possible to optimize an embedding algorithm so that it is optimal with respect to a given compression technique. This idea will be further investigated in chapter 6. Finally, masking functions may be specified in the transform domain. For example the Watson masking function described in section 2.2.3 is specified in the DCT domain. In order to most easily mask a watermark it is best that the mask be specified in the same domain as the watermark is being inserted in. However in chapter 6 we will show how to overcome difficulties associated with watermarking in one domain while masking in another.

In [11] Cox inserts a watermark consisting of a sequence of random numbers $\boldsymbol{x} = x_1 \dots x_n$ with a normal distribution. The watermark is inserted in the DCT domain of the image by one of three methods:

$$v_i' = v_i + \alpha x_i \tag{4.13}$$
$$v_i' = v_i(1 + \alpha x_i) \tag{4.14}$$
$$v_i' = v_i e^{\alpha x_i} \tag{4.15}$$

where α determines the watermark strength and the v_is are perceptually significant spectral components. The most interesting approach is the second approach. We note that the image adaptivity is implicit since the watermark in this case multiplied with given DCT coefficients. In other words, strong coefficients are changed more than weak coefficients which corresponds to frequency masking. Further masking can be achieved by allowing $\alpha = \alpha_i$ that is the strength is changed for different spectral components. To verify the presence of the watermark, the normalized correlation coefficient is used:

$$sim(\boldsymbol{X}, \boldsymbol{X}^*) = \frac{\boldsymbol{X}\boldsymbol{X}^*}{\sqrt{\boldsymbol{X}^*\boldsymbol{X}^*}} \tag{4.16}$$

where $\sim$ is a measure of similarity, $\boldsymbol{X}^*$ is the recovered watermark obtained by taking the difference between the recovered image and the original image, and $\boldsymbol{X}$ is the original watermark.

Podilchuk [71] also watermarks in the DCT domain, however the image is first divided into 8×8 blocks and the DCT of each block is computed. For each block the watermark is modulated onto selected coefficients as follows:

$$I_{u,v}^* = \begin{cases} I_{u,v} + JND_{u,v} \times w_{u,v}, & \text{if} I_{u,v} > JND_{u,v} \\ I_{u,v} & \text{otherwise} \end{cases} \tag{4.17}$$

where $I_{u,v}$ are the transform coefficients of the original image, $w_{u,v}$ are the watermark values and $JND_{u,v}$ is the computed JND based on visual models. For the case of the DCT domain, Podilchuk calculates the JND's from Watson's model. Watermark detection is based on the correlation between the difference of the original image and

the image under inspection and the watermark sequence. The same strategy has also been applied to the Wavelet domain where the JND's in the Wavelet domain are calculated based on the model presented in [70]. Voloshynovsky also describes a wavelet domain watermarking scheme however in his scheme the JND's are calculated in each subband based on NVFs [93].

In a different approach, Barni [3] uses equation 4.14 to embed a watermark in the magnitude of the DFT of the image. However, since the magnitude of the DFT is employed, the condition

$$|\alpha x_i| < 1 \tag{4.18}$$

must be met. x_i take on values in the interval $[-1, 1]$ and $\alpha \leq 1$. After marking the image in the DFT domain, the image is further masked in the spatial domain to ensure invisibility. The mask is calculated by computing the local variance in a 9×9 window and normalizing the result with respect to the maximum. The resulting watermarked image is then given by:

$$\boldsymbol{I}'' = \boldsymbol{I} + \boldsymbol{M}(\boldsymbol{I}' - \boldsymbol{I}) = \boldsymbol{I} + \boldsymbol{M}\boldsymbol{W} \tag{4.19}$$

where $\boldsymbol{I}$ is the original image, $\boldsymbol{M}$ is the mask which takes on values between 0 and 1, $\boldsymbol{I}'$ is the watermarked image before masking, $\boldsymbol{W}$ is the watermark, and $\boldsymbol{I}''$ is the watermarked image after spatial domain masking. The use of local variance is a simple way of identifying textured regions however we note that it fails at edges which separate two flat regions. Unfortunately such regions have a high variance, but cannot in general be strongly watermarked since the eye is sensitive to the presence of edges particularly in the horizontal and vertical directions.

One challenging aspect of marking in this way in the DFT domain is the design of the decoder. Here we cannot assume assume a Gaussian distribution for the coefficients since the coefficients take on only positive values. Consequently methods based on correlation which implicitly assume Gaussian statistics perform poorly. Barni derives the optimal detector by fitting a Weibull distribution to the coefficients. The likelihood ratio is then compared to a threshold to determine if a watermark is present. The test takes the form:

$$l(\boldsymbol{y}) = \frac{f_y(\boldsymbol{y}|M_1)}{f_y(\boldsymbol{y}|M_0)} > threshold \tag{4.20}$$

where $\boldsymbol{y}$ is the recovered sequence, M_1 is the message being tested for and the distribution $f_y(\boldsymbol{y}|M_0)$ is the probability that $\boldsymbol{y}$ is observed when no message has been embedded. The optimal detector is shown to recover the watermark even after JPEG quality factor 10. Furthermore, the original image is not required at decoding. Unfortunately Barni only uses 1000 possible messages which corresponds to roughly 10 bits which is insufficient in typical applications. The problem with increasing the number of messages is that the complexity goes up linearly with the number of messages. Consequently if we wish to encode 64 bits 2^{64} likelihood ratios must be calculated which is prohibitive.

4.4 Category III Watermarking

The scheme for Category III watermarks is presented in figure 4.3. The improvement over Category II watermarks lies in the fact that all information about the image is now used to generate a watermark of maximal robustness. Although schematically simple, the method represents a significant improvement over all other watermarking techniques. The main distinction is that the image is no longer treated as additive noise. Since this idea is relatively new, little work has been done here. One of the

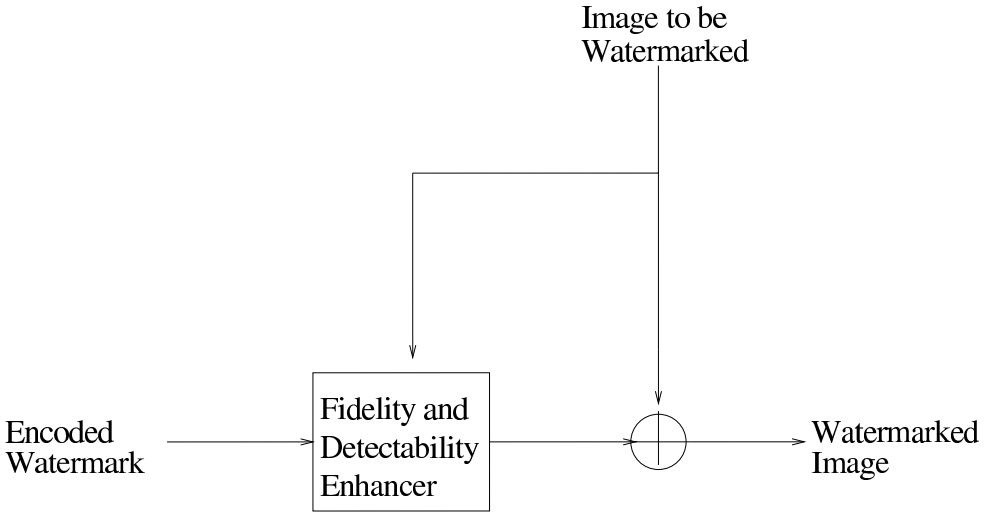


Figure 4.3: Category III Watermark

main contributions of this thesis is the development of a category III watermark in the transform domain. This will be presented in chapter 6.

Cox presents a scheme for Category III watermarking which works as follows. The image and watermark are both treated as vectors. The embedder uses the knowledge of the image vector $\boldsymbol{r}_0$ to compute a region $S(\boldsymbol{r}_0)$ within which visibility constraints on the image are satisfied. The watermarking algorithm consists of choosing a vector from within this region so that for a fixed detection strategy, the probability of detection is maximized. This is fundamentally different from other approaches which use a brute force addition or multiplication of the watermark without taking into account the detector. Cox only shows how this strategy can be applied for a 1 bit watermark in the spatial domain. In chapter 6 we will show how it is possible to embed 64 bits in the transform domain within the Category III framework. Furthermore, it will be shown how to take into account prior knowledge of JPEG quantization tables in order to improve robustness against compression, a problem which is not addressed by Cox.

4.5 Resisting Geometric Transformations

Having surveyed a wide variety of algorithms which do not address resistance to geometrical transformations, we now turn our attention to some strategies which can be used to overcome these problems. We note that the ideas presented here can in some cases be used in conjunction with other algorithms to render an algorithm robust against geometrical transformations. Three types of strategies have been proposed in the literature. The first to appear in the literature consists of designing invariant watermarks. The second consists of embedding synchronization information which we call a template. The third approach is to embed the watermark repetitively in a known way and then use the autocorrelation function to extract synchronization information. We examine these approaches in more detail.

4.5.1 Invariant Watermarks

Ruanaidh [55] develops a scheme which is invariant to rotation, scales and translations (RST). The process consists of first computing the DFT of the image. From this DFT, we work with the magnitude which is invariant to translations in the spatial domain. The log polar map (LPM) of the magnitude is then calculated. To calculate the LPM consider a point $(x, y) \in \Re^2$ and define: $x = e^{\mu} \cos\theta; y = e^{\mu} \sin\theta$ where $\mu \in \Re$ and $0 \leq \theta < 2\pi$. One can readily see that for every point (x, y) there is a point (μ, θ) that uniquely corresponds to it. The new coordinate system has the properties that rotations and scales are converted to translations as follows: $(\rho x, \rho y) \leftrightarrow (\mu + \log\rho, \theta)$ and $(x\cos(\delta) - y\sin(\delta), x\sin(\delta) + y\cos(\delta)) \leftrightarrow (\mu, \theta + \delta)$. computing again the DFT of the LPM and taking the magnitude results RST invariance. We note that taking the DFT of the LPM is equivalent to computing the Fourier-Mellin transform. A variation on the method based on the Radon transform has been proposed by Wu [99].

In video applications it is more desirable to have invariance to changes in aspect ratio than to rotation and scale changes. The above method can be modified by using log-log maps (LLM) in place of LPMs. The LLM is computed as follows. Consider a point $(x, y) \in \Re^2$ and define: $x = e^{\mu_1}; y = e^{\mu_2}$ where $\mu_1, \mu_2 \in \Re$. One can readily see that for every point (x, y) there is a point (μ_1, μ_2) that uniquely corresponds to it. The new coordinate system has the property that changes in aspect ratio are converted to translations. We note however that it is impossible to have both aspect ratio invariance and RST invariance within this paradigm since the transformations do not commute.

4.5.2 Template Based Schemes

The approach using a template is fundamentally different. The template contains no information but is merely a tool used to recover possible transformations in the

image. Ultimately, the recovery of the watermark is a two stage process. First we attempt to determine the transformation (if any) undergone by the image, then we invert or compensate for the transformation when decoding the watermark. Pereira proposes using templates of approximately 25 points [59]. The points of the template are uniformly distributed in the DFT domain with the low frequencies being excluded. The low frequencies are excluded since they contain the much of the image energy and result in visible artifacts. The points are chosen pseudo-randomly as determined by a secret key. The strength of the template is determined adaptively. Inserting points at a strength equal to the local average value of DFT points plus one standard deviation yields a good compromise between visibility and robustness during decoding. We note in particular that points in the high frequencies are inserted less strongly since in these regions the average value of the high frequencies is usually lower than the average value of the low frequencies.

At detection, in order to recover rotation or scale changes the LPM or LLM can be used as follows

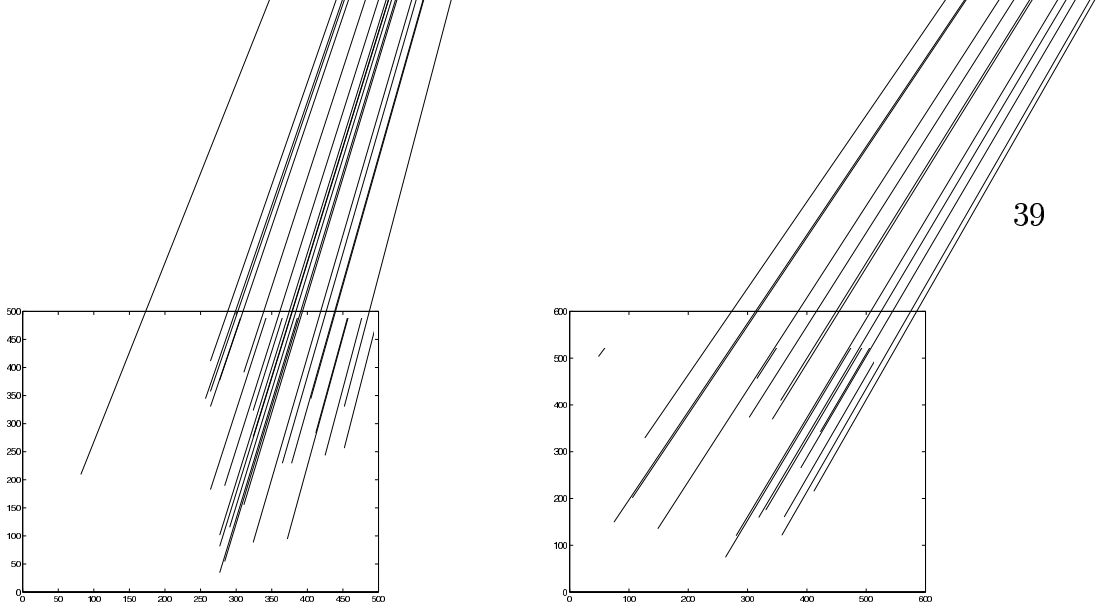
1. If the image is rectangular, extract the largest available square from the image.
2. Compute the magnitude of the FFT of the image.
3. Calculate the positions of the local peaks in the DFT using a small window (10 to 14 works well) and store them in a sparse matrix.
4. Compute the corresponding points in log polar space.
5. Compute the positions of the points in log polar space or log log space of the known template whose points are generated pseudo-randomly based on a key.
6. Compute the translation offset by exhaustive search which maximizes the numbers of points matched between the known template and the image.

While this approach is theoretically sound, the sampling problems posed by the non-uniform sampling of the LPM and LLM lead to problems with the *accurate* recovery of RST or aspect ratio changes.

In chapter 5 algorithms will be developed based on the use of a DFT domain template for the recovery of general affine transformations.

4.5.3 Autocorrelation Techniques

The third method proposed in the literature for the recovery of geometrical transformations is the use of the auto-correlation function first proposed by Kutter [40]. Kutter uses a watermark which is repeated four times in an overlapping fashion. At detection, cross-shaped filters are used to estimate the watermark as described in section 4.2 and then the auto-correlation function is calculated. The peaks in the auto-correlation are obtained due to the repetitive insertion of the watermark. Since



(a) original DFT

(b) DFT after rotation

Figure 4.4: Peaks in DFT before and after rotation

the auto-correlation of the inserted watermark is known, this is compared with the auto-correlation function of the recovered watermark. A transformation matrix is calculated based on the two sets of peaks. this transformation is then inverted and the watermark is decoded.

While this method is intuitively appealing since no additional "template" is required, the method proves to be inaccurate in recovering some rotations and scale changes. Recently Voloshynovsky [93] proposed a more robust approach where the watermark is inserted in 32×32 blocks repetitively. The resulting structure yields a DFT with many more peaks which results in better estimation of the transformation matrix. An example of the DFT before and after a geometrical transformation appears in figure 4.4.

4.6 Evaluation and Benchmarking

In order to evaluate the robustness of a given watermarking algorithm to attacks, Petitcolas and Kutter propose a benchmarking tool known a Stirmark [67, 42]. The Stirmark benchmark divides attacks into the following 9 categories: signal enhancement, compression, scaling, cropping, shearing, rotation, linear transformations, other geometric transformations, and random geometric distortions. We consider each subsection separately.

- **Signal Enhancement:** this section includes Gaussian filtering, median filtering, sharpening, and the frequency mode laplacian removal (FMLR).
- **Compression:** Two types of compression are considered here, JPEG and GIF. In the case of JPEG compression, samples are taken at quality factors between 10% and 90%.

- **Scaling:** The image is rescaled for various values between 0.5 of the original size and 2 times the original size.
- **Cropping:** A percentage of the image is cropped. This percentage varies between 1% and 75% of the original image area.
- **Shearing:** The images are sheard in the x, y and both directions for values of 1%, 5% and 10% shearing.
- **Rotation:** Two types of attacks are considered in this section. In the first case the rotation is followed by a cropping which removes the black portions which arise frm zero padding after rotation. In the second case, the image is first cropped to remove the zero padded portions and then resized to the original image size.
- **Linear:** A small linear transformation is applied to the image.
- **Other geometric transformations:** In this section, two types of attacks are considered: flipping and row and column removal. With respect to the latter, between 1 and 17 rows and/or columns are removed at *regularly* spaced intervals.
- **Random Geometric Distortions:** Only one attacked image is generated in this section. This attack consists of local bending the image in such a way that the resulting image is perceptually indistinguishable from the original. After the bending, a small random diaplacement incurred by each pixel and the resulting image is then compressed by JPEG at a quality factor of 90%.

We note that in the case of signal scaling, cropping, shearing, rotation, linear transformations, and other geometric transformations, the attacked images are obtained with and without JPEG 90% quality factor compression. In order to produce a score relative to the benchmark, we assign a score of 1 when the watermark is decoded and 0 when the watermark is not decoded. The average is then computed for each of the 9 sections, and the average of the results is computed to obtain an overall score. The benchmark should also be averaged over several images. In order to ensure a fair comparison, Petitcolas suggests imposing a minimum PSNR of 38dB for the watermarked image. Furthermore the number of bits in the watermark should be the same for all algorithms.

While Peticolas' Stirmark presents an important step towards the fair evaluation of watermarking algorithms, several problems exits.

1. The tests make no use of prior information on the watermark or image, in other words, the tests proposed are a generic set aimed at covering general image processing operations, however they do not address the issue of intelligent "attacks" on watermarks.

2. Many of the operations proposed in the benchmark substantially degrade image quality in a way that renders the result of little commercial value.

3. The quality metric of PSNR proposed is inadequate in determining if a watermark is visible. We will see in chapter 8 that the PSNR is insufficient particularly with respect to the latest perceptual coding method which can produce images with a low PSNR in which the watermark is completely invisible.

Given these limitations, in chapter 5 we propose a second generation benchmark based on more dedicated attacks. Furthermore we propose the use of Watson's metric for the purpose of evaluating the visual quality of a watermarked image. Both benchmarks will be used in evaluating algorithms presented in this thesis.

4.7 Analysis and Research Directions

While at first glance it may appear that a wide variety of viable solutions to the watermarking problem exist, a more detailed look reveals otherwise. In fact there are 9 major weaknesses with the bulk of current methods:

1. Much of the literature deals with watermarking schemes which require the original image at decoding. This is a severe limitation on watermark recovery since a search for the original must be undertaken each time we wish to detect a watermark. In many cases such a search may be extremely costly if the image is stored in a large databse.

2. Most of the methods described in the literature deal with one bit watermarks. In practice watermarks of length 60-100 allow many more commercial applications. Unfortunately results are not easily extended from the 1 bit case to the 60 bit case since the one bit case requires only detection while in the multiple bit case we need to perform detection as well as decoding. In general a fundamentally different approach must be adopted.

3. Most current algorithms with the best performances fall under the class of Category II algorithms. Consequently there is much to be gained by further investigation into the class of Category III algorithms which can only be more robust.

4. Very few algorithms deal with the problems associated with geometric changes. For the most part, early work was relegated to the recovery of rotation, scale changes and aspect ratio changes via the use of LPMs and LLMs even though quite recently results were extended to affine transformations.

5. Robustness of algorithms against JPEG compression has been limited for the most part to roughly 50% Quality factor. Unfortunately, the most recent compression schemes based on wavelet compression perform much better with fewer

visible distortions. Consequently further work needs to be done to improve the robustness of algorithms.

6. *No* algorithms proposed in the literature consistently survive the random bending attack proposed in the Stirmark benchmark. This is due to the fact that no algorithm has been able to locally compensate for the distortions introduced by this attack.

7. Few methods use optimal coding schemes. Unfortunately, only Kutter, who uses M-ary modulation, adopts a scheme which approaches the optimal performance of the system. However, even here, due to the problems associated with a practical implementation of M-ary modulation, a suboptimal compromise must be adopted. Furthermore, the bulk of the publications which deal with error-correction codes use BCH codes [32] which are known to be suboptimal.

8. Many transform domain algorithms adopt a suboptimal masking procedure where the effects of a modulation as determined by a spatial domain mask are not accounted for at embedding and decoding.

9. Since the bulk of the literature contains linear additive watermarks, few algorithms resist the watermark copy attack and ambiguity attack.

In the rest of this thesis, algorithms will be presented where we attempt to overcome the limitations discussed above.

Chapter 5

Embedding in the Fourier Domain

Many of the current techniques for embedding marks in digital images have been inspired by methods of image coding and compression. Information has been embedded using the Discrete Cosine Transform (DCT) [36, 11], Wavelets [10], Linear Predictive Coding [51], and Fractals [74] as well as in the spatial domain [69, 88]. While these methods perform well against compression, they lack robustness to geometric transformations. Consequently methods have emerged which exploit the properties of the Discrete Fourier Transform (DFT) to achieve robustness against rotation and scaling. The DFT methods can be divided into two classes, those based on invariance [55, 33] and those which embed a template into the image which is searched for during the detection of the watermark and yields information about the transformation undergone by the image [59, 75]. However both these methods exploit the properties of log-polar-maps (LPM) and can only be used to detect changes of rotation and scale. Similarly the log-log-map (LLM) [16] has also been proposed as a means of detecting changes in aspect ratio. However, once again general transformations cannot be recovered. Furthermore, results indicate that the LPMs and LLMs encounter sampling problems which lead to inaccurate compensation of geometrical transformations.

The method we propose in this chapter consists of embedding a watermark in the DFT domain. The watermark is composed of two parts, a template and a spread spectrum message containing the information or payload. The template contains no information in itself, but is used to detect transformations undergone by the image. Once detected, these transformations are inverted and then the spread spectrum signal is decoded.

The main contribution of this chapter lies in the development of a method for recovering a watermark from an image which has undergone a general affine transformation. Unlike algorithms which use log-polar or log-log-maps, we propose searching the space of possible affine transformations. Since an exhaustive search leads to an intractable problem, we demonstrate how a careful pruning of the search space leads to robust detection of transformations reasonable quickly. We will see in chapter 9 that the proposed method performs very well relative to the extensive series of tests

implemented in Petitcolas' benchmark.

The rest of this chapter is structured as follows. In section 5.1 we review the properties of the DFT which make it attractive for watermarking. In section 5.2 we describe the embedding approach. Finally in section 5.3, we describe the extraction algorithm.

5.1 The DFT and its Properties

5.1.1 Definition

Let the image be a real valued function $f(x_1, x_2)$ defined on an integer-valued Cartesian grid $0 \leq x_1 < N_1, 0 \leq x_2 < N_2$.

The Discrete Fourier Transform (DFT) is defined as follows:

$$F(k_1, k_2) = \sum_{x_1=0}^{N_1-1} \sum_{x_2=0}^{N_2-1} f(x_1, x_2) e^{-j2\pi x_1 k_1/N_1 - j2\pi x_2 k_2/N_2} \tag{5.1}$$

The inverse transform is

$$f(x_1, x_2) = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} F(k_1, k_2) e^{j2\pi k_1 x_1/N_1 + j2\pi k_2 x_2/N_2} \tag{5.2}$$

The DFT of a real image is generally complex valued. This leads to magnitude and phase representation for the image:

$$A(k_1, k_2) = |F(k_1, k_2)| \tag{5.3}$$

$$\Phi(k_1, k_2) = \angle F(k_1, k_2) \tag{5.4}$$

5.1.2 General Properties of the Fourier Transform

It is instructive to study the effect of an arbitrary linear transform on the spectrum of an image.

Once $N_1 = N_2$ (i.e. square blocks) the kernel of the DFT contains a term of the form:

$$x_1 k_1 + x_2 k_2 = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \tag{5.5}$$

If we compute a linear transform on the spatial coordinates:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \boldsymbol{T} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{5.6}$$

then one can see that the value of the DFT will not change if:

$$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \rightarrow \left(\boldsymbol{T}^{-1}\right)^T \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \tag{5.7}$$

Since our watermarks are embedded in the DFT domain, if we can determine the transformation $\boldsymbol{T}$ undergone by the image in the spatial domain, it will be possible to compensate for this transformation in the DFT domain and thereby recover the watermark. The matrix $\boldsymbol{T}$ is an arbitrary matrix which can be a composition of scale changes, rotations, and/or skews. In section 5.3.1 we will discuss how to recover watermarks when an arbitrary matrix $\boldsymbol{T}$ is applied to the image.

5.1.3 DFT: Translation

Another important property of the DFT is its translation invariance. In fact, shifts in the spatial domain cause a linear shift in the phase component.

$$F(k_1, k_2) \exp\left[-j(ak_1 + bk_2)\right] \leftrightarrow f(x_1 + a, x_2 + b) \tag{5.8}$$

From equation 5.8 of the Fourier transform it is clear that spatial shifts affect only the phase representation of an image. This leads to the well known result that the magnitude of the Fourier transform is invariant to translations in the spatial domain. This property leads directly to the fact that the watermark is robust against cropping.

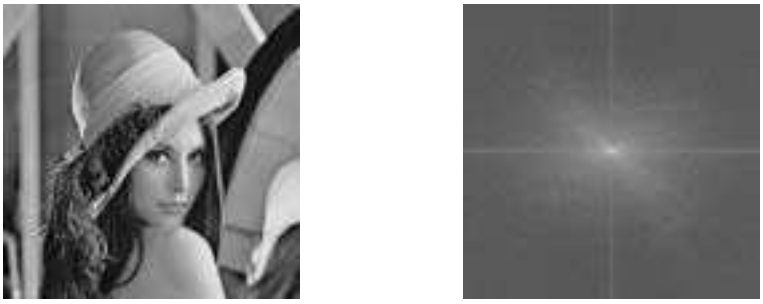
5.2 Embedding

The embedding process is divided into two steps: embedding of the watermark and embedding of the template. We consider these two steps separately.

5.2.1 Embedding the Watermark

In image watermarking, we are given a message to be embedded which can be represented in binary form as $\boldsymbol{m} = (m_1, m_2 ... m_M)$ where $m_i \in \{0, 1\}$ and M is the number of bits in the message. In realistic applications M is roughly 60 bits which contain the necessary copyright information as well as flags which can be used to indicate the type of content in the image. In our scheme, the binary message is first coded using turbo codes to produce the message $\boldsymbol{m}_c$ of length $M_c = 128$. We then apply the mapping $0 \rightarrow -1$ and $1 \rightarrow 1$ to produce the bipolar signal $\tilde{\boldsymbol{m}}_c = (\tilde{m}_{c1} ... \tilde{m}_{cM_c})$.

When working with color images, we first extract the luminance component and then rescale the RGB components accordingly. In order to embed the watermark for an image of size (m, n), we first pad the image with zeros so that the resulting size is 1024×1024. If the image is larger than 1024×1024 then the image is divided into



(a) LENA (b) LOG OF FFT

Figure 5.1: ORIGINAL LENA IMAGE AND LOG OF MAGNITUDE OF FFT

1024×1024 blocks and the watermark is calculated for each block. The watermark is embedded into the DFT domain between radii f_{w1} and f_{w2} where f_{w1} and f_{w2} are chosen to occupy a mid-frequency range. We note that the strongest components of the DFT are in the center which contains the low frequencies as illustrated in figure 5.1. Since during the recovery phase the image represents noise, these low frequencies must be avoided. We also avoid the high frequencies since these are the ones most significantly modified during lossy compression such as JPEG.

To embed the mark between the chosen radii, we first generate a sequence of points $(x_1, y_1)...(x_{M_c}, y_{M_c})$ pseudo-randomly as determined by a secret key. Here, x_i, y_i are integers such that $f_{w1} < \sqrt{x_i^2 + y_i^2} < f_{w2}$. We note that only half the available points in the annulus $\{f_{w1}, f_{w2}\}$ can be marked since the DFT must be symmetric in order to yield a real image upon inversion. In what follows we work in the upper half plane and assume that the corresponding modifications are made in the lower half plane $(\pm x_i, y_i)$ to fulfill the symmetry constraints.

Since the magnitude of the DFT is positive valued, in order to encode the bipolar message $\tilde{M}_c$, we adopt the following differential encoding scheme. For each message bit $\tilde{m}_{c_i}$ we modify the points (x_i, y_i) and $(y_i, -x_i)$ such that $k_w \tilde{m}_{c_i} = (x_i, y_i) - (y_i, -x_i)$. In other words 2 points 90° apart are modified such that the difference is equal to the desired message value. The parameter k_w is the strength of the watermark. The watermark strength can be set interactively or can be set adaptively as function of the average value and standard deviation of the DFT components of the image lying between f_{w1} and f_{w2}. If the strength is set interactively, the user can examine the artifacts introduced in the image as the strength is increased and finally settle on a strength which is as high as possible while at the same time leaving the watermark relatively invisible.

5.2.2 Embedding the Template

The template contains no information but is merely a tool used to recover possible transformations in the image. In previous work the template consisted of a random arrangement of peaks in the FFT domain [59]. We have found experimentally that using templates of approximately 8 points works best. The points of the template are distributed uniformly in the DFT with radii varying between $f_{t1} = 0.3$ and $f_{t2} = 0.35$. The angles (θ_i) and radii (r_{ij}) are chosen pseudo-randomly as determined by a secret key. The strength of the template is determined adaptively as well. We find that inserting points at a strength equal to the local average value of DFT points plus three standard deviations yields a good compromise between visibility and robustness during decoding. We note in particular that points in the high frequencies are inserted less strongly since in these regions the average value of the high frequencies is usually lower than the average value of the low frequencies. We also note that it is critical that the points be inserted strongly enough so that they remain peaks after interpolation errors from possible transformations. The reason for this will be clear in the next section.

5.3 Decoding

The watermark extraction process is divided into two phases. First we have the template detection phase and then we decode the watermark if the template has been detected.

5.3.1 Template Detection

In what follows we present two fundamentally different approaches for recovering templates. The first approach is based on the LPM/LLM paradigm, but uses an iteration based on the Chirp-Z transform to solve the problems of inaccuracy encountered with a simple application of the LPM. unfortunately, the approach based on the LPM/LLM is limited to the recovery of rotation and scale changes or aspect ratio changes, but not general linear transformations. Consequently, in the second approach we present a method for the fast recovery of general affine transformations.

5.3.1.1 Detection based on the LPM

As discussed in section 4.5.2, one possible way of recovering watermarks from rotated and scaled images, consists of applying an LPM and then using a correlation between the known template and the recovered peaks. However this approach proves to be inaccurate in practice. Consequently the basic algorithm was perfected by Pereira [61]. Since the algorithm contains novel contributions we consider it in more detail. The basic idea consists of performing a second iteration and locally using the

Chirp-Z transform defined in equation 5.9 in order to improve the resolution of the sampling in frequency space.

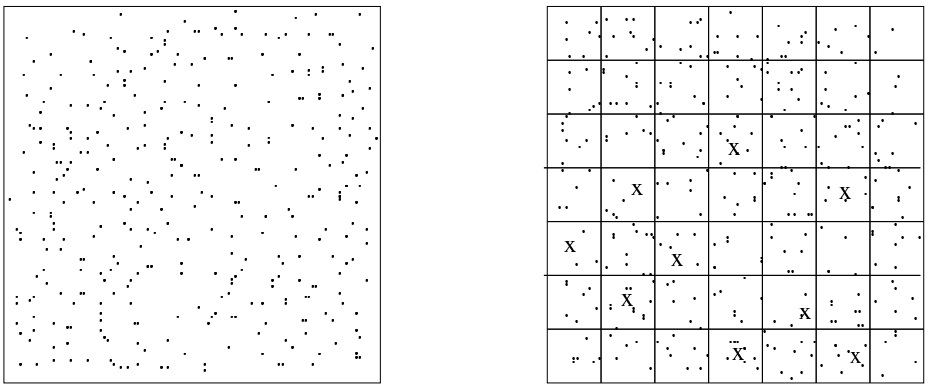
$$X(z_{1k}, z_{2l}) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x(n,m) z_{1k}^{-n} z_{2l}^{-m} \tag{5.9}$$

Where k and l are integers which go from 0 to the number of points in the transform domain. In our case z_{1k} and z_{2l} are equally spaced points in frequency space around the peak which we seek and x(n,m) is the image being transformed. The values N and M are the number of points in the image. By using the chirp-Z transfromation we can zoom in on frequencies of interest.

The new algorithm consists of the following steps:

1. If the image is rectangular, extract the largest available square from the image.
2. Compute the magnitude of the DFT of the windowed image.
3. Calculate the positions of the local peaks in the filtered DFT using a small window (10 to 14 works well) and store them in a sparse matrix.
4. Compute the corresponding points in log polar space.
5. Compute the positions of the points in log polar space of the known template.
6. Exhaustively search the space of possible matches between the known template and the extracted peaks and thereby deduce the optimal offset.
7. Use the offset to compute the scale and rotation change.
8. Invert the rotation and scale.
9. Perform the Chirp-Z transform [7] of size 50×50 on the image with the frequencies centered at a known template point.
10. Detect the location of the peak.
11. If the peak is not in the center of the template, we must further compensate the x and y scales of the image. If necessary iterate steps 9-11 at other peak locations and average the result. In practice averaging over 5 peaks yields sufficient precision.

The main problem with steps 1-8 lies in the fact that the resolution of the DFT is limited by the number of samples in the image. Steps 9-11 therefore represent the key point of the algorithm. This iteration increases the accuracy by an important factor. The Chirp-Z transform defined in equation 5.9 allows us to tightly sample the frequency domain centered at a point of interest with a resolution independent of the number of samples in the image.



(a) Peaks (b) Grid

Simply replacing the DFT in step 2 with a Chirp-Z transform is not feasible since we must then tightly sample the whole frequency domain which leads to a problem which is intractable. By performing the iteration in steps (9)-(11), we can limit ourselves to points of interest which are the known template points. Another key point is that this iteration allows us to recover small changes in aspect ratio (and not solely in uniform scaling), since when we perform the Chirp-Z transform, the x and y offsets provide the scale changes along each axis. Such small changes in aspect ratio are often incurred as a result of rounding errors, so these changes must be detected and compensated for.

We also note that step 6 can be rendered more efficient by sorting the points in the search space and dividing this space into a grid. Then, only points lying in the square of the template points need to be considered during the matching process. This is illustrated in figures (a) and (b) which contain the search space of recovered peaks which is divided into a grid. The template points are represented by "X" and only the peaks in the associated square need to be considered during the point matching. The results indicate that the proposed approach accurately recovers rotations to within 0.2 degrees and the scale of an image with less that 1% error.

5.3.1.2 Recovery of general linear transformations

We now turn our attention to the much more general problem of recovering affine transformations. In this case the problem is much more complex since the search space is much larger. Consequently, steps must be taken to optimize the algorithms to render them useful in practical situations.

The template detection process involves several steps which are enumerated below. We first present the basic algorithm and then indicate how to effectively prune the search space by applying a few heuristics.

1. Calculate the FFT of the image zero padded to 1024×1024.

2. Extract the positions of all the local peaks (p_{xi}, p_{yi}) in the image. We denote this set of peaks as $\boldsymbol{P}$.

3. Choose two points in the template (x'_1,y'_1) and (x'_2,y'_2).

4. For all pairs of points $(p_{xi}, p_{yi}); (p_{xj}, p_{yj})$ perform the following steps

   (a) Compute the transformation matrix $\boldsymbol{A}$ which maps the two template points to the point pair $(p_{xi}, p_{yi}); (p_{xj}, p_{yj})$

   (b) Apply the transformation to the other template points

   (c) Count the number of transformed template points lie within a small radius r_{min} of any peak in the set $\boldsymbol{P}$. These are the matched template points

   (d) If we have at least N_m matches we conclude that the template is found and terminate the search, otherwise we proceed to the next point pair.

5. If all point pairs have been tested and we never obtain N_m matches, we conclude that no watermark has been found.

6. If we have at least N_m matches, recalculate the linear transformation $\boldsymbol{A}$ using all the matched points such that the mean square estimation error in equation 5.10 is minimized.

$$mse = \frac{1}{nummatches} \| \boldsymbol{A} \begin{bmatrix} x'_1 & y'_1 \\ \vdots & \vdots \\ x'_l & y'_l \end{bmatrix}^T - \begin{bmatrix} p_{x1} & p_{y1} \\ \vdots & \vdots \\ p_{xl} & p_{yl} \end{bmatrix}^T \|^2 \tag{5.10}$$

   We note that $\boldsymbol{A}$ is a 2×2 linear transformation matrix so that we understand the notation $\|.\|$ to mean the sum of the magnitude of the two rows of the error matrix. The rows contain the errors in estimating the $\boldsymbol{x}$ and $\boldsymbol{y}$ from the known template positions $\boldsymbol{x'}$ and $\boldsymbol{y'}$ after applying the transformation $\boldsymbol{A}$. If we have less than N_m matches we conclude that no watermark is found.

Some observations are necessary. In step 1 we pad to 1024×1024 in order to obtain a good resolution in the FFT domain. This is important in order to obtain accurate estimates of the transformation matrix. We also note that the re-estimation in step 6 is important since it increases the accuracy of the matrix $\boldsymbol{A}$. Finally we note that step 4d contains a criterion for asserting the presence of a watermark. Consequently the template serves the dual purpose of recovering geometrical transformations and asserting the presence of a watermark even if the watermark itself may be falsely decoded. Some alternate schemes consist of using cross-correlation as in [85, 96, 17] or Bayesian models as in [56]. The advantage of using the template is that it is much more robust than the watermark itself since we concentrate a significant amount of energy into a few points in the FFT. We note however that the cost of the template ($<$1dB) remains small compared to the total watermark energy.

Step 4 contains the most costly part of the algorithm. As it stands the cost of the algorithm is $O(N^2M)$ where N is the number of points in $\boldsymbol{P}$ and M is the number of points in the template. N can be as large as 500 points so that we find that the algorithm may take up to 5 minutes in some cases on a Pentium 400MHz machine to find a search.

However the search space can be drastically pruned by exploiting two heuristics. Firstly we observe that if we choose in step 5 the points (x'_1,y'_1) and (x'_2,y'_2) which are (r'_1,θ'_1) and (r'_2,θ'_2) in polar coordinates then if $r'_1 > r'_2$ we need only consider points in the $\boldsymbol{P}$ where $r_1 > r_2$. Similarly if $r'_1 < r'_2$ we need only consider points in the $\boldsymbol{P}$ where $r_1 < r_2$. Here we are in fact exploiting the fact that for realistic transformations small and large frequencies will never be inverted. This immediately reduces the search space by a factor of 2.

The second important observation is that for transformations which are likely to occur in practice the difference in θ between two template points will change only slightly. In practice we can limit the change in θ to roughly $\pm 20°$. Exploiting this observation further reduces the search space to $\frac{40°}{360°}$ of the original size. When we apply these heuristics we obtain an algorithm which finds the affine transformation in roughly 15 seconds on a Pentium 400MHz.

Since we are using the template to assert the presence of a watermark, it is important to evaluate the probability of a false detection in order to justify the approach. The evaluation of this probability is relatively straightforward. We first note that on a 1024×1024 grid with 500 points (which for simplicity we assume are uniformly distributed) the probability of finding a point in a given location is $500/1024^2 \approx \frac{1}{2000}$. Since we work on a discrete grid, when we look for a match we also look in the 8 surrounding neighbors (i.e. r_{min} equals one pixel in step 5c), we multiply by 9 to obtain $\frac{9}{2000}$. We must now count the number of transformations which will be calculated. If we ignore the heuristics used to prune the search space, an exhaustive search involves $2N^2 = 2 \times 500^2$ transformations. The factor of 2 comes from the fact that the ordering is essential. By pruning the search space, we reduce the number of calculated transformations by a factor of 2×9 so that roughly 3000 transformations are evaluated. Now if we embed a template with 8 points and insist that all 8 points be matched at detection, we obtain that the probability of a false match given by equation 5.11. This probability is extremely small and in practice no false detections have been encountered.

$$P_{false} = (\frac{9}{2000})^8 \times 3000 \approx 5.0 \times 10^{-16} \tag{5.11}$$

5.3.2 Decoding the Watermark

Once the transformation matrix $\boldsymbol{A}$ has been detected the decoding of the watermark is straightforward and proceeds as follows.

1. Calculate the FFT of the windowed image $\boldsymbol{I}_w$ of size (I_m, I_n).

2. Generate the sequence of points $(x_1, y_1)...(x_{M_c}, y_{M_c})$ pseudo-randomly as determined by the secret key used during embedding.

3. Calculate the normalized coordinates in Fourier domain of the points as follows $(x_{ni}, y_{ni}) = (x_i/1024, y_i/1024)$.

4. Apply the transformation matrix $\boldsymbol{A}$ to the normalized coordinates to yield $(\tilde{x}_1, \tilde{y}_1)...(\tilde{x}_{M_c}, \tilde{y}_{M_c})$

5. Extract the watermark from the transformed coordinates, taking into account the 90° coding scheme used during embedding and using bilinear interpolation to obtain values for samples which do not correspond directly to samples directly on the calculated FFT. This yields the bipolar signal $\tilde{\boldsymbol{m}}'$.

6. We then take the sign of $\tilde{\boldsymbol{m}}'$ and apply the transformation $-1 \rightarrow 0$ and $1 \rightarrow 1$ to yield the recovered binary bit sequence $\boldsymbol{b}$.

7. The bit sequence $\boldsymbol{b}$ represents the recovered message encoded by the BCH error correcting codes. This sequence is now decoded to yield the recovered message $\boldsymbol{m}_r$. For a message of length 72, if there are fewer than 5 errors, the 60 bit recovered message $\boldsymbol{m}_r$ will be identical to the embedded message $\boldsymbol{m}$ since these errors will be corrected by the BCH codes.

We note that in the first step we do not pad the image with zeros since this leads to artifacts in the DFT domain. Rather, we perform directly the FFT on the windowed image and then work in normalized frequencies. We note that when performing the FFT, it is not necessary for the image size to be a power of 2 in order to transform from the spatial domain to DFT and vice-versa since we adopt the FFTW package [23] to calculate FFTs of arbitrary size efficiently. We will see in chapter 9 that the proposed algorithm performs extremely well relative to the watermarking benchmark proposed by Petitcolas.

Chapter 6

Optimized Transform Domain Embedding

Having presented the state of the watermarking literature, we now turn our attention to deriving concrete solutions to the problems with current watermarking technologies. In this chapter we mathematically formulate the problem of watermarking in the transform domain when the visibility constraints are specified in the spatial domain. By accurately modelling the embedding process we will overcome many of the limitations discussed in section 4.7.

6.1 Overview

We recall from section 4.3.2 that the advantages of transform domain watermarking are threefold: intrinsic properties of the transformation (i.e. resistance to translation or rotation), facility of adapting to compression algorithms, and ease of incorporating masking constraints. Here we will primarily be concerned with the first two points since we will consider the case where our masking constraints are specified by an NVF in the spatial domain.

While transform domain watermarking clearly offers benefits, the problem is more challenging since it is more difficult to generate watermarks which are adapted to the human visual system (HVS). The problem arises since constraints on the acceptable level of distortion for a given pixel may be specified in the spatial domain. In the bulk of the literature on adaptive transform domain watermarks, a watermark is generated in the transform domain and then the inverse transform is applied to generate the spatial domain counterpart. The watermark is then modulated as a function of a spatial domain mask in order to render it invisible. However this spatial domain modulation is suboptimal since it changes the original frequency domain watermark. In the case of a DFT domain watermark, multiplication by a mask in the spatial domain corresponds to convolution of the magnitude of the spectrum. Unfortunately, to correctly account for the effects of the mask at decoding a deconvolution problem

would have to be solved. This is known to be difficult and to our knowledge in the context of watermarking this problem has not been addressed. Methods proposed in the literature simply ignore the effects of the mask at decoding. One alternative which has recently appeared is the attempt at specifying the mask in the transform domain [71]. However other authors (e.g. Swanson [85]) have noted the importance of masking in the spatial domain even after a frequency domain mask has been applied.

Here we derive an optimized strategy for embedding a watermark in the wavelet and DCT domains when the masking constraints are specified in the spatial domain. This framework overcomes the problems with many proposed algorithms which adopt a suboptimal spatial domain truncation and modulation as determined by masking constraints. Furthermore we will develop an algorithm which is image dependent. Unlike most of the embedding strategies described in the literature which treat the image as noise possibly modelled by a probability distribution, the algorithm we describe uses all the information about the image at embedding. We consider only the problem of generating watermarks which are robust against attacks that do not change the geometry of the image. We will work with an 80 bit watermark which corresponds to a capacity sufficient for most watermarking applications. We begin in section 6.2 by presenting the spatial domain masking methods we adopt. In section 6.3 the embedding algorithm is described and applied to the case of DCT domain embedding. Then, in section 6.4 we derive a new channel coding strategy which greatly improves the performance of the underlying algorithm. In section 6.5 we show how the algorithm can be applied in the wavelet domain. In section 6.6 we present the results and a comparison of the DCT and wavelet domain algorithms followed by the conclusion in section 6.7.

6.2 Spatial Domain Masking

Here we will adopt a variation of the masking function described in section 2.2.3.1 where a noise visibility function (NVF) [94] at each pixel position is obtained as:

$$NVF(i,j) = \frac{w(i,j)}{w(i,j) + \sigma_x^2}, \tag{6.1}$$

where $w(i,j) = \gamma[\eta(\gamma)]^\gamma \frac{1}{||r(i,j)||^{2-\gamma}}$ and $r(i,j) = \frac{x(i,j)-\bar{x}(i,j)}{\sigma_x}$,$\eta(\gamma) = \sqrt{\frac{\Gamma(\frac{3}{\gamma})}{\Gamma(\frac{1}{\gamma})}}$ and $\Gamma(t) = \int_0^\infty e^{-u}u^{t-1}du$ is the gamma function. Once we have computed the noise visibility function we can obtain the allowable distortions by computing:

$$\Delta_{pi,j} = (1 - NVF(i,j)) \cdot S + NVF(i,j) \cdot S_1 \tag{6.2}$$

While this model accurately models textures, the importance of luminance masking has also been noted in the literature as discussed in section 2.2.3.1. In particular at high luminance levels the sensitivity of the HVS follows Weber's law which states that

$\frac{\delta l}{l} = k_{Weber}$ where δl is the local change in luminance and l is the luminance of the background. At lower luminance levels the HVS is more sensitive to noise. Osberger [57] uses the DeVries-Rose law at low luminance levels (typically $< l_{th} = 10cd/m^2$) which states that $\frac{\delta l}{l} = \sqrt{\frac{l}{l_{th}}} * k_{Weber}$. In order to incorporate luminance masking into the model we propose multiplying the texture component by $(1 + k * CST(x_{i,j})$ to obtain

$$\Delta_{pi,j} = (1 + k \cdot CST(x_{i,j}) \cdot ((1 - NVF(i,j)) \cdot S + NVF(i,j) \cdot S_1) \qquad (6.3)$$

Experiments indicate that choosing $k = 5$ yields good results. This corresponds to increasing the allowable distortion by 5 in textured areas where the luminance level is high.

6.3 Problem Formulation

Having derived the spatial domain masking methods, we now mathematically formulate the embedding process as a constrained optimization problem. We assume that we are given an image to be watermarked denoted $\boldsymbol{I}$. If it is an RGB image we work with the luminance component. We are also given a masking function $\boldsymbol{V}(\boldsymbol{I})$ which returns 2 matrices of the same size of $\boldsymbol{I}$ containing the values $\Delta_{pi,j}$ and $\Delta_{ni,j}$ corresponding to the amount by which pixel $I_{i,j}$ can be respectively increased and decreased without being noticed. We note that these are not necessarily the same since we also take into account truncation effects. That is pixels are integers in the range $0-255$ consequently it is possible to have a pixel whose value is 1 which can be increased by a large amount, but can be decreased by at most 1. In the general case, the function $\boldsymbol{V}$ can be a complex function of texture,luminance, contrast, frequency and patterns, however we choose to use the masking functions described in section 2. We wish to embed $\boldsymbol{m} = (m_1, m_2...m_M)$ where $m_i \in \{0, 1\}$ and M is the number of bits in the message. In general, the binary message may first be augmented by a checksum and/or coded using error correction codes to produce a message $\boldsymbol{m}_c$ of length $M_c = 512$.

Without loss of generality we assume the image $\boldsymbol{I}$ is of size 128×128 corresponding to a very small image. For larger images the same procedure is adopted for each 128×128 large block. To embed the message, we first divide the image into 8×8 blocks. In each 8×8 block we embed 2 bits from $\boldsymbol{m}_c$. In order to embed a 1 or 0 we respectively increase or decrease the value of a DCT coefficient. Once the DCT domain watermark has been calculated, we compute the inverse DCT transform and add it to the image in the spatial domain. At decoding, we take the sign of the DCT coefficient, apply the mappings $(+ \rightarrow 1)$,$(- \rightarrow 0)$ and then decode the BCH codes to correct possible errors.

The central problem with this scheme is that during embedding we would like to increase or decrease the DCT coefficients as much as possible for maximum robustness,

but we must satisfy the constraints imposed by $\boldsymbol{V}$ in the spatial domain. In order to accomplish this, we formulate the problem for each 8×8 block, as a standard constrained optimization problem as follows. For each block we select 2 mid-frequency coefficients in which we will embed the information bits. We then have:

$$\min_{\boldsymbol{x}} \boldsymbol{f}'\boldsymbol{x} \qquad ; \qquad \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \tag{6.4}$$

$\boldsymbol{x} = [x_{11} \ldots x_{81} x_{12} \ldots x_{82} \ldots x_{18} \ldots x_{88}]^t$ is the vector of DCT coefficients arranged column by column. $\boldsymbol{f}$ is a vector of zeros except in the positions of the 2 selected coefficients where we insert a -1 or 1 depending on whether we wish to respectively increase or decrease the value of the coefficients as determined by $\boldsymbol{m}_c$. $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$ contain the constraints which are partitioned as follows.

$$\boldsymbol{A} = \begin{bmatrix} IDCT \\ ---- \\ -IDCT \end{bmatrix} ; \quad \boldsymbol{b} = \begin{bmatrix} \Delta_p \\ ---- \\ \Delta_n \end{bmatrix} \tag{6.5}$$

where IDCT is the matrix which yields the 2D inverse DCT transform of $\boldsymbol{x}$ (with elements of the resulting image arranged column by column in the vector). If we let D_{ij} be the coefficients of the 1D DCT transform than it is easily shown that the the matrix IDCT is given by:

$$IDCT = \begin{bmatrix} D_{11}D_{11} & \ldots & D_{18}D_{11} & D_{11}D_{21} & \ldots & D_{18}D_{21} & \ldots & D_{11}D_{81} & \ldots & D_{18}D_{81} \\ D_{21}D_{11} & \ldots & D_{28}D_{11} & D_{21}D_{21} & \ldots & D_{28}D_{21} & \ldots & D_{21}D_{81} & \ldots & D_{28}D_{81} \\ \vdots & & & & & & & & & \\ D_{81}D_{11} & \ldots & D_{88}D_{11} & D_{81}D_{21} & \ldots & D_{88}D_{21} & \ldots & D_{81}D_{81} & \ldots & D_{88}D_{81} \\ D_{11}D_{12} & \ldots & D_{18}D_{12} & D_{11}D_{22} & \ldots & D_{18}D_{22} & \ldots & D_{11}D_{82} & \ldots & D_{18}D_{82} \\ \vdots & & & & & & & & & \\ D_{81}D_{12} & \ldots & D_{88}D_{12} & D_{81}D_{22} & \ldots & D_{88}D_{22} & \ldots & D_{81}D_{82} & \ldots & D_{88}D_{82} \\ \vdots & & & & & & & & & \\ D_{81}D_{18} & \ldots & D_{88}D_{18} & D_{81}D_{28} & \ldots & D_{88}D_{28} & \ldots & D_{81}D_{88} & \ldots & D_{88}D_{88} \end{bmatrix}$$

We also note that we take Δ_p and Δ_n to be column vectors where the elements are taken column wise from the matrices of allowable distortions. Stated in this form the problem is easily solved by the well known Simplex method. Stated as such the problem only allows for spatial domain masking, however many authors [86] suggest also using frequency domain masking. This is possible by adding the following constraints:

$$\boldsymbol{L} \leq \boldsymbol{x} \leq \boldsymbol{U} \tag{6.6}$$

Here $\boldsymbol{L}$ and $\boldsymbol{U}$ are the allowable lower and upper bounds on the amount we by which we can change a given frequency component. The Simplex method can also be used to solve the problem with added frequency domain constraints.

We note that by adopting this framework, we in fact allow *all* DCT coefficients to be modified (in a given 8×8 block) even though we are only interested in 2 coefficients at decoding. This is a novel approach which has not appeared in the literature. Other publications select a subset of coefficients to mark while leaving the rest unchanged. This is necessarily suboptimal relative to our approach. In words, we are "making space" for the watermark in an optimal fashion by modifying elements from the orthogonal complement of the coefficients we are interested in, while satisfying spatial domain constraints.

6.4 Effective Channel Coding

Rather than coding based on the sign of a coefficient as in [63], we propose using the magnitude of the coefficient. To encode a 1 we will increase the *magnitude* of a coefficient and to encode a 0 we will decrease the *magnitude*. At decoding a threshold T will be chosen against which the magnitudes of coefficients will be compared. The coding strategy is summarized in table 6.1 where c_i is the selected DCT coefficient.

Table 6.1: Magnitude Coding

sign(c_i)	bit	Coding
+	0	decrease c_i (set $\boldsymbol{L}$ to stop at 0)
−	0	increase c_i (set $\boldsymbol{U}$ to stop at 0)
+	1	increase c_i
−	1	decrease c_i

The actual embedding is performed by setting $\boldsymbol{f}$ in equation 6.4 based on whether we want to increase or decrease a coefficient.

The major advantage of this scheme over encoding based on the sign is that the image is no longer treated as noise. As noted by Cox [12] this is an important characteristic of the potentially most robust schemes since all *a priori* information is used . Clearly the best schemes should not treat the image as noise since it is known at embedding. However most algorithms in the literature do not take advantage of this knowledge except in the extraction of perceptual information. In our case, based on the observed image DCT coefficient we encode as indicated in table 6.1. At decoding the image is once again not noise since it contributes to the watermark. Another important property of this scheme is that it is highly image dependent. This is an important property if we wish to resist against the watermark copy attack [43] in which a watermark is estimated from one image (typically by denoising) and added to another image to produce a fake watermark. If this is done, the watermark will be falsely decoded since at embedding and decoding the marked image is an integral part of the watermark. Consequently changing the image implies changing the watermark.

It is also possible to incorporate JPEG quantization tables into the model in order to increase the robustness of the algorithm. Assume for example that we would like to aim for resistance to JPEG compression at quality factor 10. Table 6.2 contains the threshold value below which a given DCT coefficient will be set to 0. In order to

Table 6.2: JPEG thresholds at quality factor 10

30	30	30	40	60	100	130	130
35	35	35	50	65	130	130	130
40	35	45	45	65	130	130	130
40	45	60	75	130	130	130	130
50	55	95	130	130	130	130	130
60	90	130	130	130	130	130	130
125	130	130	130	130	130	130	130
130	130	130	130	130	130	130	130

improve the performance of the algorithm we can add bounds based on the values in table 6.2 to the amount we increase a coefficient. In particular, if we wish to embed a 1 we need only increase the magnitude of a coefficient to the threshold given in table 6.2 in order for it to survive a JPEG compression at quality factor 10. This is accomplished by setting the bounds $\boldsymbol{L}$ and $\boldsymbol{U}$. Since 2 bits are embedded per block, the remaining energy may be used to embed the other bit. It is important to note that it may not be possible to achieve the threshold since our visibility constraints as determined by $\boldsymbol{V}$ in the spatial domain must not be violated, however the algorithm will embed as much as much energy as possibly via the minimization in equation 6.4. We note that we choose only to embed the watermark in randomly chosen coefficients where the value in table 6.2 is less than 70 since for larger values we will require more energy to be sure that the coefficient survives at low JPEG compression. We avoid the 4 lowest frequency components in the upper left hand part of the DCT block since these tend to be visible even with small modifications.

6.5 Wavelet Domain Embedding

In order to embed the message, in the wavelet domain, we perform a similar optimization as the one performed in the DCT domain. We first divide the image into 16×16 blocks and perform the 1-level wavelet transform. In order to embed a 1 or 0 we adopt a differential encoding strategy in the lowest subband (LL). In particular we choose four neighbouring coefficients and increase two coefficients while decreasing the other two. The choice of which two to increase or decrease is a function of whether we wish to encode a 1 or a 0 so that at decoding we take the difference between the sums of the two pairs of coefficients and apply the mappings $(+ \rightarrow 1)$,$(- \rightarrow 0)$. We note that

it is important to select a 2×2 block of *neighbouring* coefficients since the underlying assumption is that the difference on average is 0. In order to embed the largest possible values while satisfying masking constraints, the problem is formulated for each 16×16 block as a constrained optimization problem. In the case of the Haar wavelet, for a 16×16 block, we have 64 coefficients available in the LL subband. In each block we encode 8 bits by selecting 32 coefficients grouped into 8 2×2 blocks. We then have equation 6.4 as before however $\boldsymbol{x} = [x_{1,1} \ldots x_{16,1} x_{12} \ldots x_{16,2} \ldots x_{1,16} \ldots x_{16,16}]^t$ is the vector of coefficients arranged column by column. Furthermore, $\boldsymbol{f}$ is a vector of zeros except in the positions of the selected coefficients where we insert a (-1) or (1) depending on whether we wish to respectively increase or decrease the value of a coefficient. The constraints are now partitioned as:

$$\boldsymbol{A} = \begin{bmatrix} IDWT \\ - - - - \\ -IDWT \end{bmatrix}; \quad \boldsymbol{b} = \begin{bmatrix} \Delta_p \\ - - - - \\ \Delta_n \end{bmatrix} \tag{6.7}$$

where IDWT is the matrix which yields the 2D inverse DWT transform of $\boldsymbol{x}$ (with elements of the resulting image arranged column by column in the vector). We also note that we take Δ_p and Δ_n to be column vectors where the elements are taken columnwise from the matrices of allowable distortions. If we let D_{ij} be the coefficients of the 1D inverse DWT (known as the synthesis matrix [92]) then it is easily shown that the matrix IDWT in our 2D case is given analagously to equation 6.3. The problem is once again solved by the Simplex method. Furthermore, extra constraints in the frequency domain can also be incorporated as before via equation 6.6. Unfortunately, in this case it is not possible to optimize the method relative to JPEG compression since the quantization matrice are specified in the DCT domain.

6.6 Results

Both algorithms were tested on several small images of size 128x128. Prior to embedding the 80 bit message, we first append a 20 bit checksum and then encode the message using turbo codes [5] to yield a binary message of length 512. Turbo codes provide near optimum performance for Gaussian channels and are consequently superior to other codes used currently in watermarking (mostly BCH and convolution). We note that even though this channel is not Gaussian, tests indicate that turbo codes outperform BCH codes. In fact, JPEG compression introduces quantization noise which is difficult to model. However, the development of optimal codes for quantization channels is well beyond the scope of this paper. The 20 bit checksum is essential in determining the presence of the watermark. At detection if the checksum is verified we can safely say (with probability $\frac{1}{2^{20}}$ of error) that a watermark was embedded and successfully decoded.

With respect to wavelet domain watermarking, both the Haar wavelet and the Daubechies 4-tap filter were tested. In the case of the Haar wavelet, the algorithm

was resistant down to a level of 70% quality factor. Better results were obtained for the 4-tap Daubechies filter where the algorithm is robust down to a level of 50% quality factor and is resistant as well to low and high pass filtering. By resistant, we understand that all the bits are correctly decoded and the checksum verified. We note that for the case of the Daubechies 4-tap filter, some minor modifications must be made to the embedding strategy. In particular, when taking the inverse DWT we obtain a block size which is bigger than the original block. These boundary problems are well known in the wavelet literature. The difficulties are easily overcome by imposing that the extra boundary pixels be constrained to be 0. This is done in practice by setting the appropriate values in Δ_p and Δ_n to 0.1 and -0.1 respectively. We do *not* set these all the way to zero since often this leads to an overly constrained problem. An example is given in figure 6.6 where the original image (128×128), watermarked image (Daubechies 4-tap filter) and watermark (difference between original and watermarked) are presented. We observe that the watermark is stronger in textured regions as expected. We note that the watermark is slightly visible along the long vertical edge to the left of the image. This is a limitation of the visibility model which does not take into account the high amount of structure to which the eye is particularly sensitive. In order to overcome this problem more sophisticated models are being developed which take into account the presence of lines in the image. In these regions, the allowable distortion must be reduced. Maximizing the strength of the watermark while minimizing visibility in an automatic way over a wide range of images is a delicate problem since each image is unique and presents its own difficulties

On a Pentium 233MhZ computer the algorithm takes 20 minutes to embed the watermark. This time is non-negligible. The problem arises from the fact that a formidable optimization problem must be solved at embedding. That is at each block we have $2 * 16 * 16 = 512$ constraints. On the other hand the optimization in each block is independent once the global mask has been calculated. Consequently the algorithm can be carried out in parallel.

In chapter 9 we evaluate the algorithm relative to the benchmark.

6.7 Summary and Open Issues

In this chapter we have described a new mathematical model which describes the process of embedding a watermark in a transform domain when the masking constraints are specified in the spatial domain. This model has 5 characteristics which make it extremely appealing:

1. The algorithm is extremely flexible in that constraints as determined by masking functions can be easily incorporated in the spatial domain and any linear transform domain may be used although here we considered the special cases of the Haar and Daubechies wavelets as well as DCT domain embedding. Also,

extra constraints may be added in the frequency domain.

2. We show how to handle problems with truncation in an optimal way and propose the novel approach of modifying all coefficients even though we are only interested in a subset.

3. The algorithms resist well against JPEG compression and we observe in particular that matching the embedding domain with the compresssion domain and incorporating JPEG quantization tables at the embedding stage leads to considerable gains.

4. The algorithm generates an image dependent watermark which consequently resists the watermark copy attack [43].

5. At the embedding stage the image is not treated as noise which is an important property of the most robust watermarking schemes as noted by Cox [12]. In fact the algorithm uses all available information about the image at the embedding stage to maximize the strength of the watermark unlike most of the embedding strategies described in the literature which treat the image as noise possibly modelled by a probability distribution.

While much has been accomplished by structuring the problem of watermarking within this framework, many new research directions arise. We note four possiblities in particular:

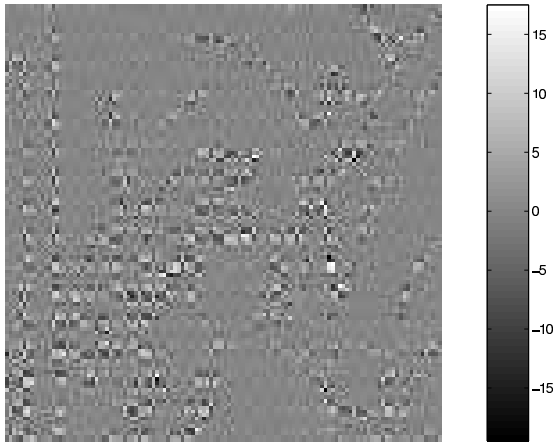
1. While the DCT domain algorithm resists well against JPEG compression further research is needed in order to adapt the wavelet domain approach so that it is resistant against EZW and SPIHT compression.

2. Work is currently also under way to apply the ideas of [60] so as to make the algorithm resistant to geometric changes as well.

3. Another topic of further research is the incorporation of more sophisticated spatial domain masks. Most of the masks proposed in the watermarking literature model texture, luminance and/or frequency. Osberger [57] however identifies several higher order factors which have been used to weight distortion metrics (typically the distortion produced by compression algorithms). As discussed in section 2.2.3.2, we note that these factors are specified in the spatial domain and not easily converted to the frequency domain. Further work could involve incorporating these elements in the attempt to generate more accurate spatial domain masks.

4. While some work has been done in capacity (e.g. [81] the bulk of the results concern additive watermark. An interesting topic of further research is the calculation of the capacity of the proposed non-additive scheme.

(c)



(d) (e)

Figure 6.1: Original image Lena(a) along with watermarked image (b) and watermark in (c)=(a)-(b).

Chapter 7

Attacks

Digital watermarking has emerged as an appropriate tool for the protection of author's rights [11]. It is now well accepted that an effective watermarking scheme must successfully deal with the triple requirement of *imperceptibility* (visibility) - *robustness* - *capacity* [70]. Given the relatively complex tradeoffs involved in designing a watermarking system, the question of how to perform fair comparisons between different algorithms naturally arises. A lack of systematic benchmarking of existing methods however creates confusion amongst content providers and watermarking technology suppliers. Existing benchmarking tools like Stirmark [66] or Unzign [91] integrate a number of image processing operations or geometrical transformations aimed at removing watermarks from a stego image. However, the quality of the processed image is often too degraded to permit further commercial exploitation. Moreover, the design of these tools does not take into account the statistical properties of the images and watermarks in the design of attacks. As a result, pirates can design more efficient attacks that are not currently included in the benchmarking tools. This could lead to a tremendous difference between what existing benchmarks test and real world attacks.

Within this context, the goal of this chapter is threefold. First, we derive new methods of removing watermarks and emphasize the cse of linear additive watermarks. Secondly, in the spirit of Fabien Petitcolas' Stirmark benchmarking tool, we propose a second generation benchmark which attacks watermarking schemes in a more effective manner. In particular the attacks contained in our benchmark take into account prior information about the image and watermark as well as the watermarking algorithm used. Our main conclusion is that while several algorithms perform well against the benchmark proposed by Petitcolas, the algorithms we evaluate all perform poorly relative to the proposed benchmark. This suggests that although claims about "robust" watermarks persist in the literature, the reality of the situation as demonstrated by systematic testing is otherwise. Third, we propose the use of Watson's metric as a fair criteria for comparing the visibility of different watermarking schemes. We show that PSNR as proposed by Petitcolas is inadequate, and that Watson's metric is quite

robust in yielding a fair comparison between algorithms.

The chapter is structured as follows. In section 7.1 we review the formulation for linear additive watermarks which will be the basis for developing attacks. Then in section 7.2 we discuss the weak points of linear additive watermarks. We then consider denoising, denoising and perceptual remodulation, lossy compression, template removal, watermark copy attack and the denoising followed by random bending attacks in sections 7.3-7.8. We then turn our attention to the subject of perceptual quality estimation and propose a new metric in section 8.1 and finally present the benchmark in section 8.2.

7.1 Problem formulation

We first review aspects of linear embedding algorithms since this paradigm will be the base for deriving new dedicated attacks which will be included in a second generation of benchmarking tools. We return to the general model of a watermarking system according to a communications formulation. Its block diagram is shown in Figure 7.1. The watermarking system consists of three main parts, i.e. message

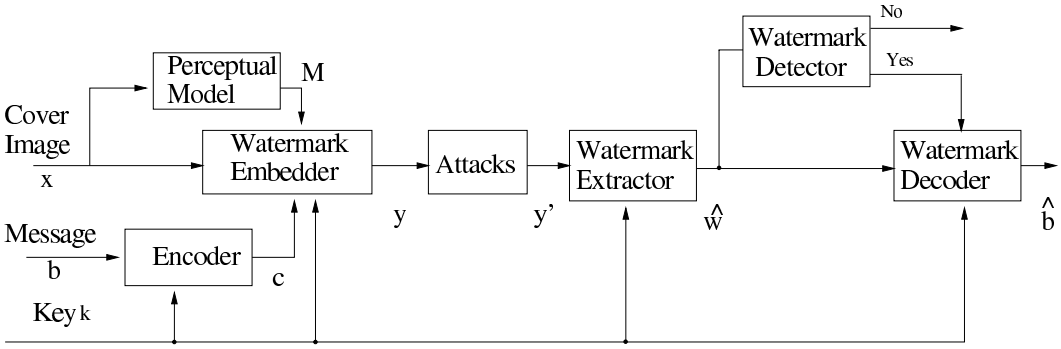


Figure 7.1: Communication formulation of a watermarking system

embedding, attack channel and message extraction. We briefly review the message extraction since this is the base of our attacks. A detailed description has been given in 4.1.3.

The recovery process consists of the watermark extractor and decoder which are reviewed below. The watermark extractor performs an estimate $\widehat{w}$ of the watermark based on the attacked version $\widehat{y}$ of the stego-image:

$$\widehat{w} = Extr\left(T\left[y'\right], Key\right) \tag{7.1}$$

In the general case, the extraction should be key-dependent. However, the desire to recover data after affine transformation based on the above mentioned self-reference principle, and the opportunity to enhance the decoding performance by reducing the

variance of the image considered as noise [41, 30], have motivated the development of key-independent watermark extraction methods. They could represent the main danger to linear additive watermarking technologies, as will be shown below.

The MAP estimate of the watermark is given by:

$$\widehat{w} = \arg\max_{\widetilde{w}\in\Re^N}\{\, p_X\left(y' \mid \widetilde{w}\right) \cdot p_W\left(\widetilde{w}\right)\} \tag{7.2}$$

where $p_W(.)$ is the p.d.f. of the watermark. Assuming that the image and watermark are conditionally i.i.d. locally Gaussian, i.e. $x \sim N(\overline{x}, R_x)$ and $w \sim N(0, R_w)$ with covariance matrices R_x and R_w, where R_w also includes the effect of perceptual watermark modulation, one can determine:

$$\widehat{w} = \frac{R_w}{R_w + R_x}\left(y' - \overline{y}'\right) \tag{7.3}$$

where it is assumed $\overline{y}' \approx \overline{x}$, and where $\widehat{R}_x = \max\left(0, \widehat{R}_y - R_w\right)$ is the ML estimate of the local image variance $\left(\widehat{R}_x = \sigma_x^2 I\right)$.

In most cases the results of attacks and of prediction/extraction errors are assumed to be additive Gaussian. The detector is therefore designed using an ML formulation for the detection of a known signal in Gaussian noise, that results in a correlator detector with reduced dimensionality:

$$r = \langle \widehat{w}, p \rangle . \tag{7.4}$$

Therefore, given an observation vector r, the optimum decoder that minimizes the conditional probability of error assuming that all codewords b are equiprobable is given by the ML decoder:

$$\widehat{b} = \arg\max_{\widetilde{b}}\, p\left(r \mid \widetilde{b}, x\right). \tag{7.5}$$

Based on the central limit theorem (CLT) most researchers assume that the observed vector r can be accurately approximated as the output of an additive Gaussian channel noise [41, 30].

7.2 Watermark attacks based on the weak points of linear methods

A key-independent watermark prediction according to (7.3) presents several problems. The first problem is connected with the assumption that the stego image is not significantly altered after attack which allows the perceptual mask used at embedding to be estimated from the attacked stego image [41, 30, 18]. However this assumption does not hold for attacks connected with histogram modification that could have a

significant influence on models based on luminance masking, and lossy JPEG compression attack whose strong blocking artifacts could alter models based on texture masking.

Another series of problems are tied to the general security-robustness issue. Since the watermark can be predicted based on (7.3) without knowledge of the key, the following problems appear:

1. The redundancy in the watermark and global watermark energy can be considerably reduced as a result of denoising and compression, this especially in flat image regions.

2. Special types of distortions could be introduced in the watermark, aiming to create the least favorable conditions for the decoder. In particular, perceptual remodulation of the watermark aimed at creating the least favorable statistics for the AWGN decoder designed based on (7.4,7.5) will be shown to be an extremely effective attack.

3. The synchronization can be destroyed by estimating template or the parameters of periodical watermarks and then removing the synchronization mechanism.

4. If we ignore perceptual masking, most algorithms generate watermarks independently from the image. This leads to vulnerability with respect to the watermark copy attack in which the watermark is estimated from one image and added to another one in order to generate a falsely watermarked image.

We will consider each of these points in detail in the text that follows.

7.3 Watermark removal based on denoising

The watermark can be removed from the stego image in some cases or its energy can be considerably decreased using denoising/compression attack. Consider the MAP estimation of the cover image as image denoising according to the additive model (6.3)

$$\widehat{x} = \arg\max_{\widetilde{x}\in\Re^N} \left\{\ln p_W\left(y \mid \widetilde{x}\right) + \ln p_X\left(\widetilde{x}\right)\right\}. \tag{7.6}$$

With the assumption of uniform prior on the statistics of the image, one receive the ML-estimate:

$$\widehat{x} = \arg\max_{\widetilde{x}\in\Re^N} \left\{p_W\left(y \mid \widetilde{x}\right)\right\}. \tag{7.7}$$

To solve problems (7.6) and (7.7) it is necessary to use accurate stochastic models for the cover image $p_X(x)$ and the watermark $p_W(w)$. Here we consider the sGG and nG models derived in section 2.2.3.1.

We can now classify the possible image denoising methods into ML (no prior on image) and MAP (with image prior) estimates. An overview of the denoising methods depending on the image and watermark statistics is shown in figure 7.2.

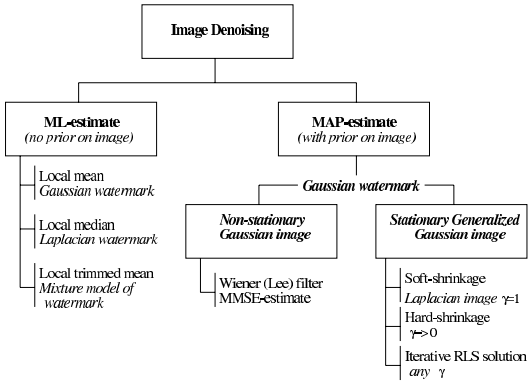


Figure 7.2: Classification of image denoising methods

7.3.1 ML solution of image denoising problem

The ML-estimate (7.7) has a closed form solution for several cases when the watermark has either the Gaussian, or the Laplacian, or the mixture of the Gaussian and Laplacian distributions. If the watermark has a Gaussian distribution the ML-estimate is given by the local mean of y: $\widehat{x} = localmean\,(y)$.

On the other hand, if the watermark can be modeled by the Laplacian distribution the solution of the ML-estimate is given by the local median: $\widehat{x} = localmedian\,(y)$. In the theory of robust statistics, the mixture model of the Gaussian and Laplacian distributions (so called $\epsilon-$contaminated model) is used. The closed solution in this case is the local trimmed mean filter that uses order statistics such as the median filter but produces the trimmed version of the mean centered about the median point. The size of the window used for the mean computation is determined by the percentage of the impulse outliers given by parameter ϵ hence the name "$\epsilon-$contaminated".

In practice, a sliding square window is used in which either the local mean or median is computed. However, in the case of natural images one can compute more accurate estimates of the local mean or median by considering only pixels in a cross-shaped neighborhood. This is due to the fact that natural images feature a higher correlation in the horizontal and vertical directions.

7.3.2 MAP solution of image denoising problem

Assuming $w \sim i.i.d.\ N(0, R_W)$, $R_W = \sigma_w^2 I$ the MAP problem (7.6) is reduced to [94]

$$\widehat{x} = \arg\min_{\widetilde{x}\in\Re^N} \left\{ \frac{1}{2\sigma_w^2} \|y - \widetilde{x}\|^2 + \rho\,(res) \right\} \tag{7.8}$$

where $\rho\,(res) = [\eta\,(\gamma)\,|res|]^\gamma$, $res = \frac{x-\overline{x}}{\sigma_x}$, $||.||$ denotes the matrix norm, and $\rho\,(res)$ is the energy function for the sGG model.

To generalize the iterative approaches to the minimization of the non-convex function (7.8) we propose to reformulate it as a *reweighted least squares (RLS)* problem. Then equation (7.8) is reduced to the following minimization problem [94]:

$$x^{k+1} = argmin_{\tilde{x}\in\Re^N}\{\frac{1}{2\sigma_n^2}\|y-\tilde{x}^k\|^2 + \phi^{k+1}\|r^k\|^2\}, \tag{7.9}$$

where

$$\phi^{k+1} = \frac{1}{r^k}\rho'(r^k), \tag{7.10}$$

$$r^k = \frac{x^k - \bar{x}^k}{\sigma_x^k}, \tag{7.11}$$

$$\rho'(r) = \gamma[\eta(\gamma)]^\gamma \frac{r}{\|r\|^{2-\gamma}}, \tag{7.12}$$

and k is the number of iterations. In this case, the penalty function is quadratic for a fixed weighting function ϕ. We can also receive a closed form solution for several important image priors.

First, consider the model: $x \sim N\left(\overline{x}_j, \sigma_{x_j}^2\right)$, $w \sim N\left(0, \sigma_w^2 I\right)$. The solution to this problem is the well known adaptive Wiener or Lee filter:

$$\widehat{x} = \overline{y} + \frac{\sigma_x^2}{\sigma_x^2 + \sigma_w^2}\left(y - \overline{y}\right). \tag{7.13}$$

Second, we assume that $x \sim sGG\left(\overline{x}, 1, \sigma_x^2 I\right)$, i.e. Laplacian, and $w \sim N\left(0, \sigma_w^2 I\right)$. The solution to this problem is soft-shrinkage [94] which is well known in the wavelet domain [19]

$$\widehat{x} = \overline{y} + \max\left(0, |y - \overline{y}| - T\right) sign\left(y - \overline{y}\right) \tag{7.14}$$

where $T = \sqrt{2}\frac{\sigma_w^2}{\sigma_x}$. It was shown recently [53] that the hard-shrinkage denoiser can be determined under the same priors in the limiting case $\gamma \to 0$:

$$\widehat{x} = \overline{y} + \psi\left(|y - \overline{y}| > T\right)\left(y - \overline{y}\right) \tag{7.15}$$

where $\psi\left(.\right)$ denotes a thresholding function that keeps the input if it is larger than T and otherwise sets it to zero. The main idea of all the above denoisers (7.13-7.15) is to decompose the image into a low frequency part $\overline{y}$ and a high frequency part $(y - \overline{y})$. Each part is then treated separately. The scaling part of the Wiener solution is depicted in (7.3)a, and shrinkage functions for soft- and hard-thresholds are shown

in (7.3b) and 7.3c, respectively. Relatively small values of $(y - \overline{y})$ represent the flat regions (the same statement is true for wavelet coefficients), while the high amplitude coefficients belong to the edges and textures. Therefore, denoising is mostly due to the "suppression" of noise in the flat regions where the resulting amplitude of the filtered image is either decreased by a local factor $\frac{\sigma_x^2}{\sigma_x^2+\sigma_w^2}$ as in the Wiener filter or just simply equalized to zero as in the case of shrinkage methods. The obvious conclusion is that the shrinkage methods behave in a more aggressive way with respect to the removal of watermark coefficients from the flat image regions, in comparison to the Wiener filter which only decreases their strength. Therefore, it is possible either to remove the watermark in the flat regions completely or to decrease considerably its energy. It is also necessary to note that since the watermark is removed or its strength is decreased the MMSE is decreased while the perceptual quality is enhanced after attack.

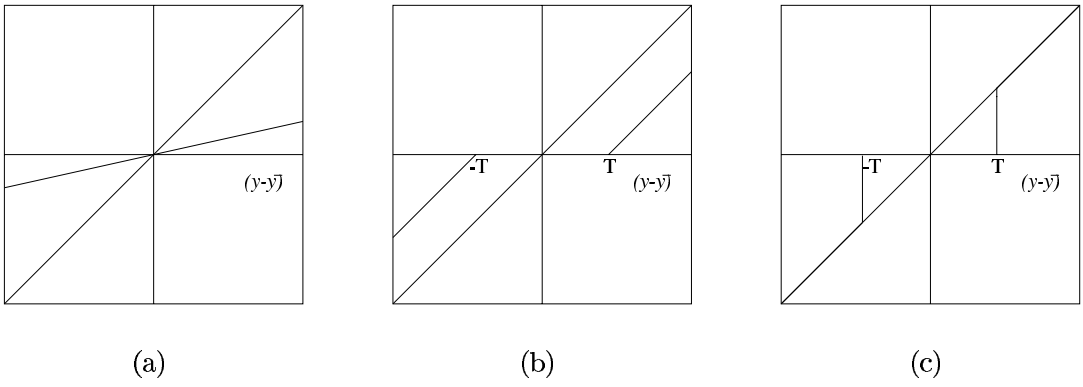


Figure 7.3: Scaling/shrinkage functions of the image denoising algorithms

7.4 Lossy wavelet Compression attack

The modern wavelet lossy compression algorithms exploit both intra- and inter-scale redundancy of real images. A well-known example of intrascale model based methods that exploits the zero-correlation across the subbands is the embedded zerotree wavelet (EZW) algorithm proposed in [83] and its extended implementation in SPHIT [79]. The example of interscale coder is EQ-coder proposed by Lopresto et al [48] that utilizes the above stochastic image models for quantization scheme design. 7.4. The main difference between denoising and lossy compression is that compression includes some type of quantization. This is depicted in figure 7.4 where we see the result of soft-shrinkage and then the effect of soft-shrinkage followed by compression.

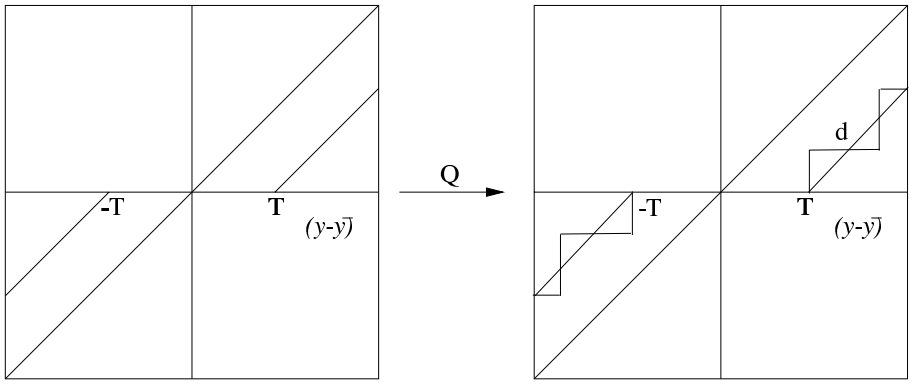


Figure 7.4: Approximation of the soft-shrinkage function by quantization with zero-zone.

7.5 Denoising/Compression watermark removal and prediction followed by perceptual remodulation

Another way to prevent satisfactory estimation of the watermark sign is to attempt to flip its sign. This however has to be done for fraction of the pixels, otherwise one would get a flipping of the watermark which could easily be retrieved. It is thus necessary to change the signs randomly (or periodically if some information about the ACF is available) so as to create the least favorable situation for the decoder. There are two different ways to attain this goal.

The first possibility is to estimate the watermark and then to perform remodulation in such a way that the projection of the watermark on the space p in 7.4 will be on average a zero-mean vector. The particular cases of this generalized attack were studied in [44, 29] assuming that the watermark is extracted from the stego image with some strength factor. These attacks have several drawbacks in the case of content-adaptive watermarking, where the strength of the watermark is differs as a function of image regions. In these cases the assumption that the watermark as well as the image are zero-mean, wide-sense stationary Gaussian processes is satisfied neither for the content adaptive watermark nor for the images. As a consequence, the extraction of the watermark is applied with the same strength for flat regions and for edges and textures. Therefore, the watermark could just be inverted and non-visibility is not guaranteed here. This indicates that watermark remodulation should be content adaptive.

The second possibility consists of creating outliers with a sign opposite to the local estimated watermark sign, taking into account visibility constraints [95]. Considering the prior reduction of sampling space in the flat regions due to denoising/compression, this will lead to an unsatisfactory solution when the CLT assumption is made. The resulting distribution of errors due to outliers will no longer be strictly Gaussian. In

this case, the decoder designed for the AWGN will be not optimal and the general performance of the watermarking system will be decreased. Additionally, if the attacker can discover some periodicity in the watermark structure, this could be effectively used for remodulation to reach the above goal.

We now present practical aspects of remodulation. One method consists of changing the amplitude relationship among the pixels in a given neighborhood set. In the most general case, one has to solve a local optimization problem of watermark sign change under constraint of minimal visible distortions for every pixel in the set. We call this approach *perceptual remodulation*. Based on practically driven motivations one can assume that only some pixels in a neighborhood set should be changed during the optimization. This will certainly constrain the level of variability but has the benefit of leading to very simple closed form solutions.

Assume one can have the estimate of the watermark sign based on the predictor (7.3) as

$$s = sign\,(y - \overline{y})\,. \tag{7.16}$$

The idea is to remodulate the watermark by a sign opposite to s, according to a perceptual mask that will assign stronger weights for the textures and edges and smaller ones for the flat regions (if the Wiener filter is used for the denoising/compression attack). We have used here the texture masking property of the HVS for this perceptual remodulation based on the noise visibility function (NVF) [94]. Other reasonable models could be used here as well. In the case of NVF the resulting attacked image can be written as:

$$y = x + [(1 - NVF) \cdot S_e + NVF \cdot S_f] \cdot (-s) \cdot p' \tag{7.17}$$

where S_e and S_f are the strengths of the embedded watermark for edges and textures and for flat regions, respectively, $p' \in \{0, 1\}$ is a spreading function for non-periodical watermark with probability of appearance "0" equal to ω, and "1" - $(1 - \omega)$. The performance of the attack will be demonstrated in chapter 9.

7.6 Watermark copy attack

The idea of the this attack is to copy a watermark from one image to another image without knowledge of the key used for the watermark embedding [43]. The attack consists of two basic stages, i.e. watermark prediction and addition to another image with the adaptation of the predicted watermark features to the target images. As the watermark prediction scheme one can use either the above considered ML or MAP estimates considering the stego image as noisy image with the additive noise to be the watermark. Then the watermark can be computed from the denoised image by taking the difference between the estimate $\widehat{x}$ of the cover image and the stego image: $\widehat{w} = y - \widehat{x}$.

In the next stage the predicted watermark is adapted to the target image to keep it imperceptible while maximizing the energy. There are many practical ways to adapt the watermark to the target image based on the methods exploiting the contrast sensitivity and texture masking phenomena of the HVS. To model texture masking we use the NVF based on the stationary Generalized Gaussian model [94]. The NVF characterizes the local texture of the image and varies between 0 and 1, where it takes 1 for flat areas and 0 for highly textured regions. In addition, it is also proposed to take into account [43] the contrast sensitivity to combine it with the NVF. The contrast sensitivity is described by the Weber-Fecher law, which states that the detection threshold of noise is approximately proportional to the local luminance. The final weight is then given by: $M = ((1 - NVF)\,\alpha + NVF\,(1 - \alpha))\,Lum$, where α describes the relation between the watermark strength in the textured areas and flat areas, and Lum is the local luminance. If we set $\alpha = 1$, the watermark will be concentrated in the texture areas, while taking $\alpha = 0$, the watermark will be mainly embedded in the flat areas.

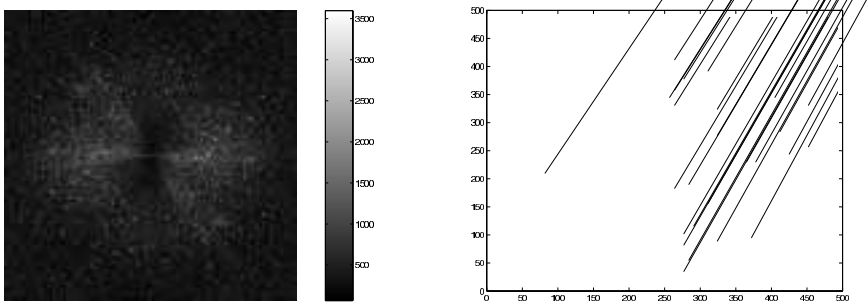
The fake watermarked image is then generated by scaling the weighted function M, multiplying it by the sign of the predicted watermark, and then adding the result to the target image: $y' = t + \beta M sign\,(\widehat{w})$ where t is the target image and β is the overall watermark strength.

7.7 Template/ACF removal attack

Synchronization is a key issue of digital watermarking and the synchronization attacks can be considered as a separate important class of attacks. We concentrate on two main methods of watermark synchronization based on the template in the magnitude image spectrum and the ACF of periodically extended watermark. The main idea of our approach is detect synchronization mechanisms by analysis of the magnitude spectrum of the predicted watermark $|\Im\,(\widehat{w})|$. The main assumption is that with state of the art technologies, synchronization is largely based on generating periodic structures. Two possibilities exist. The first consists of inserting peaks in the DFT which is the so called "template" approach used recently by Pereira to recover affine transformations [60]. The second approach consists of directly embedding the watermark periodically as done by Kutter [41] and more recently Voloshynovskiy [93]. In both cases peaks are generated in the DFT which can be exploited by an attacker.

It is obvious that the template peaks will be easily detected since the spectrum of periodically repeated watermark has a discrete structure with the period inversely proportion to the period of watermark in the coordinate domain. Figure 7.5 contains an example of detected synchronization in the magnitude image spectrum used in the template approach (a) and the periodic watermark (b).

As an example of this idea, we extracted peaks from the watermarked images based on the template principle used by commercial software A which is shown in 7.5. Once the peaks have been detected, the next step of desynchronization is to interpolate the



(a) template (b) auto-correlation

Figure 7.5: DFT peaks associated with a template based scheme(a) and auto-correlation based scheme(b)

spectrum of the stego image or previously attacked image in the locations of spatial frequencies determined by a local peak detector. We use a simple neighborhood interpolation scheme. As a consequence, all affine geometrical transforms will destroy the watermark synchronization and leave the watermark undetectable.

7.8 Denoising with Random Bending

Since one the most effective attack to date is the Stirmark random bending attack, we propose an improved version of this attack. Instead of immediately applying random geometric distortions, we choose to first apply soft thresholding. This is done in order to effectively suppress the watermark in flat areas with little or no impact on visual quality. The Stirmark random bending attack is then applied to the denoised image.

Chapter 8

Towards a Second Generation Benchmark

The aim of this chapter is twofold. We first present a new method for determining the visibility of a watermark in an image. We then define a new benchmark based on the attacks in the previous section and on the perceptual quality metric.

8.1 Perceptual Quality Estimation

In order to reduce the visibility effects of the insertion of a watermark, algorithms can take advantage of the HVS characteristics by inserting watermarks in the less sensitive regions of the images, such as the textured regions. A good image quality metric should take into account the HVS characteristics to provide accurate measurements and therefore objectively state whether or not a given watermark is visible. Unfortunately the widely used PSNR metric does not take into account such characteristics and it cannot be used as a reference metric for measuring image quality. In fact the PSNR metric does not take into account image properties. The attractiveness of PSNR in applications such as image restoration and segmentation arises from the fact that it is directly related to the squared error. Since typically algorithms attempt to minimize square error, the PSNR accurately measures to what extent this goal was attained. However, in watermarking applications, the goal is to produce a watermark which is as robust as possible while still being invisible. Within this context, PSNR is inadequate as we will see.

In what follows we propose an image quality metric based on the Watson model, but adapted for the application of watermarking.

8.1.1 The Watson model

The central aim of the Watson metric [98] is to weight the errors for each DCT coefficient in each block by its corresponding sensitivity threshold which is a function

of the contrast sensitivity, luminance masking and contrast masking. A detailed derivation of this model was given in section 2.2.3.1. However in order to render the result applicable to the watermarking problem, we choose to define a total perceptual error (TPE) by using the formula:

$$TPE = \frac{1}{N^2} \sum_k \sum_{i,j} |d_{ijk}| \tag{8.1}$$

where the d_{ijk} are defined in section 2.2.3.1. We note that this pooling differs from the Minkowski summation proposed by Watson, however our tests indicate that with respect to the watermarking application better results are obtained.

8.1.2 Comparison of the Watson metric and the PSNR

In this section we present some examples that demonstrate the accuracy of the Watson metric in cases where the PSNR is inadequate.

For the first example we add the same additive white noise to the images Benz and mandrill. The PSNR for both images is the same but the visual quality for the mandrill image is much better than for the Benz image as we can see in Figure (8.1). The Watson metric states that the quality for the mandrill image is much better than for the benz image, which is in accordance with our perception. In order to objectively specify if an image is acceptable or not, we must specify some thresholds beyond which the image is declared unacceptable relative to the proposed objective measure. We determine the following thresholds by performing subjective tests for different types of images and then taking the average values.

1. A global perceptual error threshold $GT = 4.1$, so that images with a total perceptual error less than this threshold are considered to be globally of good quality. We find however that in some cases, even though the global threshold is satisfied, the image is too distorted to be of commercial value. This may arise in cases where the watermark has been inserted too strongly at a few locations. While this may only slightly influence the global criteria of a large image, the watermark will still be visible locally. Consequently we propose also using local measures.

2. A first local perceptual error threshold $LT1 = 7.6$ for blocks of size 16×16. Blocks with greater total perceptual error than this threshold may be locally visible but not enough to systematically reject the image. The variable NB1 contains the number of these potentially visible blocks. This reflects the fact that the metric is not pefect and that in some cases human judgment is necessary.

3. A second local perceptual error threshold $LT2 = 30$ for blocks of size 16×16, so that the error in blocks with greater total perceptual error than this threshold

Figure 8.1:

(a) original benz

(b) original mandrill

(c) PSNR=31, TPE=9

(d) PSNR=31, TPE=4

Table 8.1: PSNR and Watson measures for the images benz and mandrill

	Same additive white noise			
	PSNR	TPE	NB1	NB2
benz	31.71	9.07	522	0
mandrill	31.69	4.15	28	0

are in all cases visible so that the image cannot be accepted. The number of these blocks is reported in the variable NB2 and the image is rejected if NB2 is equal or greater than one.

With respect to the tests, the images were displayed on 24 bit screens from an Ultra Sparc 10. The application used to display the images was XV version 3.10a. It is important to notice that the errors visible to the human eye will depend on the luminance and contrast parameters from the screen and the application used to display the images.

Table (8.1) reports the PSNR and the Watson measures for the images benz and mandrill from Figure (8.1). We note that the Watson metric correctly indicates that the errors are much more visible in the Benz image than in the Mandrill image even though the PSNR is the same.

For our second example we consider 3 different watermarked versions of the Barbara image with different image quality but the same PSNR, see Figure (8.2). Once again the Watson metric with provides measurements according to the perceptual reality, thus proving to be more precise than the PSNR. For our last example we consider the Lena image to which initials of a name have being added and are locally very visible, see Figure (8.3). The PSNR metric does not say anything about this local degradation, and the total perceptual error is not so useful in this case, but we obtain TPE=0.26, NB1=6, and NB2=2 from the Watson metric. The number of blocks NB2 with greater perceptual error than the second local threshold is 2. Consequently, the image is rejected.

8.2 Second Generation Benchmarking

Having described various new attacks and having proposed an accurate and objective measure of image quality, we are now in position to define a second generation benchmark. We note that the benchmark we propose is not intended to replace the benchmark proposed by Kutter and Petitcolas [42], but rather to complement it. While their benchmark heavily weights geometric transformations and contains non-adaptive attacks, the benchmark we propose includes models of the image and watermark in order to produce more effective attacks.

The benchmark consists of six categories of attacks where for each attacked image

(e) original

(f) sGG model

(g) nG model

(h) non-adaptive

Figure 8.2: a) The original Barbara image b)PSNR=24.60, TPE=7.73, NB1=119, NB2=0 c) PSN =24.59, TPE=7.87, NB1=128, NB2=0 d)PSNR=24.61, TPE=9.27, NB1=146, NB2=3

(a) (b)

Figure 8.3: a) The original Lena image b) initials of a name added

a 1 is assigned if the watermark is decoded and 0 if not. The categories are the following where we note in parentheses the abbreviations we use later for reporting results:

1. Denoising (DEN): We perform three types of denoising, Wiener filtering, soft thresholding and hard thresholding. We take the average of the three scores.

2. Denoising followed by perceptual remodulation (DPR).

3. Denoising followed by Stirmark random bending (DRB).

4. Copy Attack (CA): We estimate the watermark using Wiener filtering and copy it onto another image. If the watermark is successfully detected in the new image, 0 is assigned otherwise a score of 1 is obtained.

5. Template removal followed by small rotation (TR).

6. Wavelet Compression (WC): In this section we compress the image using bitrates [7,6,5,4,3,2,1,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2]. We weight the samples between 7 and 1 by 75% while the rest count for 25%. The finer sampling at smaller bitrates is important since most current algorithms survive until a bitrate of 1 or 2 and then start to break down. The finer sampling allows us to better localize at which point the algorithms break down. In some applications such as video, bitrates in the range of 0.2 are frequently encountered. We note that this corresponds roughly to a JPEG quality factor of 10% however the artifacts are much less problematic since the blocking effects do not occur with wavelet compression.

Chapter 9

Results

In this chapter we present the results of the proposed algorithms and compare them to existing commercial software packages. The chapter is structured as follows in section 9.1 we present the results of the algorithms relative to Watson's metric in order to evaluate the perceptual quality of watermarked images. In section 9.2 we present results relative to Fabien Petitcolas' benchmark and our benchmark. Section 10 contains our conclusions and proposals for future research.

9.1 Perceptual Quality Evaluation

In this section we report the results relative to the proposed benchmark for two commercial software packages which we denote A and B as well as the algorithm C which is the DCT algorithm presented in chapter 3 and algorithm D which is the FFT algorithm presented in chapter 4 which includes the recovery of affine transformations. We note that the results for the wavelet domain approach presented in chapter 3 are the same as those for the DCT domain approach except in the case of JPEG compression where the DCT algorithm performs better. Due to the redundancy and for clarity of presentation we omit the presentation of the wavelet domain results. For our tests we use the six images bear, boat, girl, lena, watch, and Mandrill. The original images are presented in figure 9.1

(a) bear

(b) boat

(c) watch

(d) lena

(e) mandrill

(f) girl

Figure 9.1: Original test images.

(a) watch:software A

(b) lena:software B

(c) mandrill:software C

(d) girl:software D

Figure 9.2: Examples of images marked by the 4 algorithms.

Examples of marked images appear in figure 9.1. In all cases after printing, the marked images are indistinguishable from the originals. Table (9.1) shows Watson measures for 6 images for the watermarks generated by the four approaches. In all cases the value NB2 was 0 except for the bear image where we obtained the values 13,4,4, and 8 for softwares A-D respectively. This indicates that watermark produced by the four watermarking algorithms is locally visible for the bear image to the point that the image is rejected. One example is given in Figure (9.3) which shows the marked version of the bear image for software C and the total perceptual errors for blocks of size 16×16. We notice that the errors on the marked images are not visible when printed but they are clearly visible on the screen. It is important to note that the errors were not visible under all viewing conditions, but in practice image watermarks must be invisible under all conditions that might be encountered in practice. The Watson metric identified that for all four approaches the watermark was visible in the dark flat areas of the bear image. For the rest of the images the

(a) (b)

Figure 9.3: a) The bear marked image b) total perceptual errors for blocks 16×16

Watson metric reports a good quality which is in accordance with our observations on the screen.

9.2 Benchmarking results

In this section we evaluate the proposed approach relative to a standard series of tests detailed by Petitcolas and Kutter [67, 42].

We first use the stirmark 3.1 [66] program to evaluate the algorithm. The tests are divided into the following 9 sections: signal enhancement, compression, scaling, cropping, shearing, linear transformations, rotation, row/column removal, and random geometric distortions. For each attack we consider the attack by itself and where

Table 9.1: Watson measures for images bear, boat, girl, lena,watch and mandrill.

	A		B		C		D	
	TPE	NB1	TPE	NB1	TPE	NB1	TPE	NB1
bear	7.10	29	2.48	14	1.65	13	6.05	25
boat	2.61	1	0.98	0	0.31	0	2.35	0
girl	3.44	5	1.31	0	0.59	0	1.78	0
lena	2.63	0	1.02	0	0.34	0	2.35	0
watch	3.69	11	1.35	0	0.49	0	1.46	0
mand	3.09	1	1.05	0	0.39	0	1.65	0

Table 9.2: Results relative to Petitcolas' benchmark

	A	B	C	D
Enhancement	1	1	1	1
Compression	0.81	0.95	1	0.74
Scaling	0.72	0.95	0	0.78
Cropping	1	1	0	0.89
Shearing	0.5	0.5	0	1
Linear	0	0	0	1
Rotation	0.94	0.5	0	1
Row/column removal+flip	1	1	0	1
Random Geometrical Distortions	0.33	0	0	0
Average	0.7	0.65	0.22	0.82

applicable after JPEG compression at a quality factor of 90. For each image we assign a score of 1 if for that case, the watermark is correctly decoded. If the watermark is incorrectly decoded, we assign a value of 0. We then compute an average for each section and summarize the results.

Relative to the benchmark series of tests, The FFT approach (algorithm D) performs well and better than commercially available algorithms. The algorithm fails for random geometric distortions since the FFT is severely distorted. However, to our knowledge, at this time no algorithm systematically decodes watermarks successfully after being attacked by the random geometric distortions implemented in the stirmark package.

The major improvement of the DFT over the other algorithms lies in recovery general affine transforms. We note in particular that the algorithm is successful 100% of the time in cases of shearing and linear transformations whereas the other algorithms only recover the watermark in cases where the shearing is small and never in the case of linear transformations. Unfortunately, the DCT algorithm (C) performs markedly less well. This is due to the fact that it is unable to recover geometric transforma-

tions. Since the benchmark is heavily weighted towards geometrical attacks, the DCT scores poorly.

Table 9.3 reports the scores of the four algorithms relative to our proposed benchmark. The results were averaged over 5 images. We note that the maximum possible score is 6. The results indicate that the DCT algorithm C based performs markedly

Table 9.3: Benchmark Results

	DEN	DPR	DRB	CA	TR	WC	total
A	0	0	0	0	0	0.79	0.79
B	0	0	0	1	0	0.75	1.75
C	0.93	0.8	0	1	0	0.79	3.53
D	0	0	0	0	0	0.79	0.79

better than the other commercial softwares tested and the DFT. Clearly a far different picture is obtained from this benchmark than from Petitcolas' benchmark. This results from the fact that algorithm C uses non-linear and non-adaptive techniques. Such watermarks will be inherently more resistant to the attacks proposed. While it is true that the attacks proposed in the benchmark target linear additive schemes, it is important to note that developing effective denoising approaches for non-additive and non-linear watermarks is much more difficult which suggests in itself that effective watermarking algorithms should not be based on the linear additive paradigm. We also note that all algorithms fail against the template removal attack and the denoising followed by random bending attacks which indicates that technologies are still not mature relative to the problem of synchronization.

We note that to obtain a complete picture, we must take into account the results of both benchmarks. The results suggest that future research should aim at rendering the DCT algorithm resistant to geometrical changes. We note that rendering the commercial software or the DFT approach resistant against the denoising based attacks is a much more difficult task since fundamental changes must be made with respect to the embedding strategy. On the other hand it may be possible to render the DCT approach resistant to geometrical attacks by using a template.

Chapter 10

Conclusion and further research directions

In this thesis, we have developed two approaches based on fundamentally different technologies for addressing the problem of watermarking. The FFT approach described is now a mature technology which has more or less reached it's limit in the sense that it is unclear how to improve the algorithm. The algorithm performs well relative to Petitcolas' benchmark, but extremely poorly relative to the benchmark we propose. On the other hand, the technology based on the DCT algorithm is relatively new. Preliminary results demonstrate the potential of the algorithm. In particular, relative to our second generation benchmark, the algorithm outperforms by a large margin existing commercial software as well as the DFT algorithm. On the other hand, the algorithm at this time does not resist against geometrical transformations.

The other main contribution of this thesis is the proposal of a second generation benchmark which includes much more sophisticated attacks. Better understanding of the mechanisms of possible attacks will lead to the development of more efficient and robust watermarking techniques and as such our results present an important step in this direction. Furthermore we have proposed a new quality metric which provides a much better objective measure of image quality in the context of watermarking. The results clearly show that the metric we propose presents an important improvement over the PSNR which breaks down in many practical situations.

While much has been accomplished in this thesis, a number of research directions arise naturally as extensions of the current work particularly with respect to the DCT algorithm. These are the following:

1. While the DCT domain algorithm resists well against JPEG, work remains to be done to adapt the method for resistance against wavelet compression.

2. The DCT algorithm must be made resistant to geometrical changes either by use of a template or techniques based on auto-correlation.

3. More sophisticated masks should be tested which incorporate higher level vision factors as discussed in the literature survey.

4. Another important theoretical aspect which remains to be developed is the calculation of capacity. This will probably prove to be quite difficult in the case of DCT embedding due to the non-linear nature of the embedding.

5. Techniques must be developed which render the algorithm resistant to the local random bending attack which remains a big problem.

6. Yet more sophisticated attacks should be developed particularly targeting non-linear systems such as the DCT domain algorithm proposed. Once again, this problem will be challenging, but nevertheless important since it will inevitable lead to more robust embedding techniques.

While the above mentioned points deal mainly with the technical aspects of the embedding process, other research directions include the search for new applications for watermarking. One interesting application which has arisen recently is the notion of an Internet bridge. In such an application, an image containing information in the form of a watermark which would be presented in front of a camera typically on top of the monitor. The watermark would then be decoded and the information could be used to reference a web-site.

Bibliography

[1] A. J. Ahumada and H. A. Peterson. Luminance model-based dct quantization for color image compression. In *Proc. SPIE:Human vision, Visual Processing and Digital Display III*, volume 1666, pages 365–374. SPIE, 1992.

[2] A. Alattar. "Smart Images" using digimarc's watermarking technology. In Ping Wah Wong and Edward J. Delp, editors, *IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II*, volume 3971 of *SPIE Proceedings*, San Jose, California USA, 23–28 jan 2000.

[3] M. Barni, F. Bartolini, A. De Rosa, and A. Piva. A new decoder for the optimum recovery of non-additive watermarks. *IEEE Transactions on Image Processing*, submitted 2000.

[4] W. Bender, D. Gruhl, and N. Morimoto. Method and apparatus for data hiding in images. *U.S. Patent # 5689587*, 1996.

[5] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: turbo-codes. *IEEE Trans. Comm.*, pages 1261–1271, October 1996.

[6] T. Boersema and H. J. G. Zwaga. Searching for routing signs in public buildings: the distracting effects of advertisements. *Visual Search*, pages 151–157, 1990.

[7] G. Bonmassar and E. Schwartz. Space-variant fourier analysis: The exponential chirp transform. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, October 1997.

[8] S. Chang, B. Yu, and M. Vetterli. Spatially adaptive wavelet thresholding with content modeling for image denoising. In *Proc. of 5th IEEE International Conference on Image Processing ICIP98*, Chicago, USA, October 1998.

[9] Digimark Corporation. http://www.digimark.com/. January 1997.

[10] M. Corvi and G. Nicchiotti. Wavelet-based image watermarking for copyright protection. In *The 10th Scandinavian Conference on Image Analysis*, June 1997.

[11] I. Cox, J. Killian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for images, audio and video. In *Proceedings of the IEEE Int. Conf. on Image Processing ICIP-96*, pages 243–246, Lausanne, Switzerland, 1996.

[12] I. J. Cox, M. L. Miller, and A. L. McKellips. Watermarking as communications with side information. *Proceedings of the IEEE*, 87(7):1127–1141, July 1999.

[13] S. Craver, N. Memon, B. L. Yeo, and M. M. Yeung. Can invisible watermark resolve rightful ownerships? In *Fifth Conference on Storage and Retrieval for Image and Video Database*, volume 3022, pages 310–321, San Jose, CA, USA, February 1997.

[14] S. Craver, N. Memon, B. L. Yeo, and M. M. Yeung. Resolving rightful ownership with invisible watermarking techniques. *IEEE J. Selec. Areas Communic. (Special Issue on Copyright and Piracy Protection)*, 16:573–586, May 1998.

[15] G. Csurka, F. Deguillaume, J. K. Ó Ruanaidh, and T. Pun. A bayesian approach to affine transformation resistant image and video watermarking. In *Third International Workshop on Information Hiding*, Dresden, Germany, September 29 - October 1st 1999.

[16] F. Deguillaume, G. Csurka, J. J. K. Ó Ruanaidh, and T. Pun. Robust 3D dft video watermarking. In *IS&T/SPIE Electronic Imaging99, Session: Security and Watermarking of Multimedia Contents*, San Jose, CA, USA, January 1999.

[17] J. F. Delaigle, C. De Vleeschouwer, and B. Macq. Watermarking algorithm based on a human visual model. *Signal Processing*, 66:319–335, 1998.

[18] G. Depovere, T. Kalker, and J. P. Linnartz. Improved watermark detection reliability using filtering before correlation. In *IEEE Int. Conference on Image Processing 98 Proceedings*, Chicago, Illinois, USA, October 1998. Focus Interactive Technology Inc.

[19] D. Donoho and I. Johnstone. Ideal spatial adaptation via wavelet shrinkage. In *Biometrika*, volume 81, pages 425–455, 1994.

[20] G. S. Elias, G. W. Sherman, and J. A. Wise. Eye movements while viewing NTSC format television. Technical report, SMPTE Psychophysics Subcommittee, March 1984.

[21] O. Emery. Des filigranes de papier. *A. T. I. P. Bull.:Bull de l'association Technique de l'Industrie Papetiere*, 12(6):185–188, 1958.

[22] J. M. Findlay. The visual stimulus for saccadic eye movement in human observers. *Perception*, 9(7), 1980.

[23] M. Frigo and S. Johnson. fftw-1.3. In *software*, MIT, Boston, Massachusetts, 1997-98.

[24] D. Geman and S. Geman. Stochastic relaxation, gibbs distributions and the bayesian restorations of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(6):367–383, 1984.

[25] F. Goffin, J. F. Delaigle, C. De Vleeschouwer, B. Macq, and J. J. Quisquater. A low cost perceptive digital picture watermarking method. In I. K. Sethin and R. C. Jain, editors, *Storage and Retrieval for Image and Video Database V*, volume 3022, pages 264–277, San Jose, California, U.S.A., February 1997. The Society for Imaging Science and Technology (IS&T) and the International Society for Optical Engineering (SPIE), SPIE.

[26] R. Gold. Optimal binary sequences for spread spectrum multiplexing. *IEEE Transactions on Information Theory*, IT-14:619–621, 1967.

[27] S. W. Golomb. *Shift Register Sequences*. Holden Day, 1967.

[28] F. Hartung and M. Kutter. Multimedia watermarking techniques. *Proceedings of the IEEE*, 87(7):1079–1107, July 1999.

[29] F. Hartung, J. K. Su, and B. Girod. Spread spectrum watermarking: Malicious attacks and counterattacks. In *Proc. SPIE Security and Watermarking of Multimedia Contents 99*, San Jose, CA., January 1999.

[30] J. R. Hernández and F. Pérez-González. Statistical analysis of watermarking schemes for copyright protection of images. *Proceedings of the IEEE*, 87(7):1142–1166, July 1999.

[31] J. R. Hernández, F. Pérez-González, J. M. Rodrígez, and G. Nieto. The impact of channel coding on the performance of spatial watermarking for copyright protection. *in Proc. ICASSP'98*, 5:2973–2976, May 1998.

[32] J. R. Hernández, F. Pérez-González, J. M. Rodrígez, and G. Nieto. Performance analysis of a 2-D-multipulse amplitude modulation scheme for data hiding and watermarking of still images. *IEEE Journal on Selected Areas in Communications*, 16(4):510–523, May 1998.

[33] A. Herrigel, J. Oruanaidh, H. Peterson, Pereira S., and Pun T. Secure copyright protection techniques for digital images. In *2nd International Information Hiding Workshop*, Portland, Oregon, April 1998.

[34] D. Hilton. Method of and apparatus for manipulating digital data works. International Publication number WO 96/27259, september 1996.

[35] M. Holliman and N. Memon. Counterfeiting attacks on linear watermarking systems. In *Proc. IEEE Multimedia Systems 98, Workshop on Security Issues in Multimedia Systems*, Austin, Texas, June 1998.

[36] C.-T. Hsu and J.-L. Wu. Hidden digital watermarks in images. *IEEE Transactions on Image Processing*, 8(1):58–68, January 1999.

[37] N. Jayant, J. Johnston, and R. Safranek. Signal compression based on human perception. *Proceedings of the IEEE*, 81(10):1385–1422, october 1998.

[38] M. Kobayashi. Digital watermarking: Historical roots. Technical report, IBM Research, Tokyo Res. Lab, 1997.

[39] D. Kundur and D. Hatzinakos. Digital watermarking for telltale tamper proofing and authentication. *Proceedings of the IEEE (USA)*, 87(7):1167–1180, July 1999.

[40] M. Kutter. Watermarking resistent to translation, rotation and scaling. In *Proc. SPIE Int. Symp. on voice, Video, and Data Communication*, November 1998.

[41] M. Kutter. *Digital image watermarking: hiding information in images*. PhD thesis, EPFL, Lausanne, Switzerland, August 1999.

[42] M. Kutter and F. A. P. Petitcolas. A fair benchmark for image watermarking systems. In *Electronic Imaging '99, Security and Watermarking of Multimedia Contents*, volume 3657, pages 219–239, San Jose, CA, USA, January 1999.

[43] M. Kutter, S. Voloshynovskiy, and A. Herrigel. Watermark copy attack. In Ping Wah Wong and Edward J. Delp, editors, *IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II*, volume 3971 of *SPIE Proceedings*, San Jose, California USA, 23–28 jan 2000.

[44] G. C. Langelaar, R. L. Lagendijk, and J. Biemond. Removing spatial spread spectrum watermarks by non-linear filtering. In *Proc. Europ. Signal Processing Conf. (EUSIPCO 98)*, Rhodes, Greece, Sept. 1998.

[45] G. C. Langelaar, J. C. A. van der Lubbe, and J. Biemond. Copy protection for multimedia data based on labeling techniques. In *Proc. 7th Sympo. Information Theory in Benelux*, Enschede, The Netherlands, May 1996.

[46] G. C. Langelaar, J. C. A. van der Lubbe, and R. L. Lagendijk. Robust labeling methods for copy protection of images. In *Storage and Retrieval for Image and Video Databases V.*, IS&T/SPIE Proceedings, pages 298–309, San Jose, CA, USA, February 1997.

[47] J. P. Linnartz and M. van DIjk. Analysis of the sensitivity attack against electronic watermarks in images. In *International Information Hiding Workshop*, April 1998.

[48] S. LoPresto, K. Ramchandran, and M. Orhard. Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework. In *Data Compression Conference 97*, pages 221–230, Snowbird, Utah, USA, 1997.

[49] H. D. L ukr. *Korrelationssignale*. Springer, Berlin, Germany, 1992.

[50] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. PAMI*, 11:674–693, 1989.

[51] K. Matsui and K. Tanaka. Video-Steganography: How to secretly embed a signature in a picture. In *IMA Intellectual Property Project Proceedings*, pages 187–206, January 1994.

[52] P. Moulin and J. Liu. Analysis of multiresolution image denoising schemes using generalized-gaussian priors. In *Proc. IEEE Sig. Proc. Symp. on Time-Frequency and Time-Scale Analysis*, Pittsburg, USA, October 1998.

[53] P. Moulin and J. Liu. Analysis of multiresolution image denoising schemes using generalized-gaussian and complexity priors. In *Proc. IEEE Trans. Info. Theory*, volume 45, pages 909–919, April 1999.

[54] E. Niebur and C. Koch. *Computational Architectures for Attention*. MIT press, 1997.

[55] J. Oruanaidh and T. Pun. Rotation, scale and translation invariant spread spectrum digital image watermarking. *Signal Processing*, 66(3):303–317, 1998.

[56] J. J. K. Ó Ruanaidh and G. Csurka. A bayesian approach to spread spectrum watermark detection and secure copyright protection for digital image libraries. In *IEEE Conf. on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, USA, June 1999.

[57] W. Osberger, N. Bergmann, and A. J. Maeder. An automatic image quality assessment technique incorporating higher level perceptual factors. In *IEEE ICIP-98*, Chicago,USA, October 1998.

[58] E. Peli. In search of a contrast metric: Matching the perceived contrast of gabor patches at different phases and bandwidths. *Vision Research*, 37(23):3217–3224, 1997.

[59] S. Pereira, J. J. K. Ó Ruanaidh, F. Deguillaume, G. Csurka, and T. Pun. Template based recovery of Fourier-based watermarks using Log-polar and Log-log maps. In *Int. Conference on Multimedia Computing and Systems, Special Session on Multimedia Data Security and Watermarking*, Juin 1999.

[60] S. Pereira and T. Pun. Fast robust template matching for affine resistant watermarks. In *3rd International Information Hiding Workshop*, Dreseden, Germany, September 1999.

[61] S. Pereira and T. Pun. An iterative template matching algorithm using the chirp-z transform for digital image watermarking. *Pattern Recognition*, 33(1), January 2000.

[62] S. Pereira, S. Voloshynovskiy, and T. Pun. Effective channel coding for DCT watermarks. In *ICIP 2000*, Vancouver, Canada, submitted.

[63] Shelby Pereira and Thierry Pun. A framework for optimal adaptive dct watermarks using linear programming. In *Tenth European Signal Processing Conference (EUSIPCO'2000)*, Tampere, Finland, sep 5–8 2000.

[64] H. A. Peterson, H. Peng, and W. B. Pennebaker. Quantization of color image components in the dct domain. In *Proc. SPIE:Human vision, Visual Processing and Digital Display II*, volume 1453, pages 210–222. SPIE, 1991.

[65] F. Petitcolas, R. Anderson, and M. Kuhn. Information hiding: A survey. *Proceedings of the IEEE:Special Issue on Identification and Protection of Multimedia Information*, 87(7), July 1999.

[66] F. A. P. Petitcolas. http://www.cl.cam.ac.uk/ fapp2/watermarking/stirmark/. In *Stirmark3.1(79)*, 1999.

[67] F. A. P. Petitcolas and R. J. Anderson. Attacks on copyright marking systems. In *2nd International Information Hiding Workshop*, pages 219–239, Portland, Oregon, USA, April 1998.

[68] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein. Theory of spread spectrum communications - A tutorial. In *IEEE Transactions on Communications*, volume COM-30(5), pages 855–884, May 1982.

[69] I. Pitas. A method for signature casting on digital images. In *Proceedings of the IEEE Int. Conf. on Image Processing ICIP-96*, pages 215–218, Lausanne, Switzerland, September16-19 1996.

[70] C. I. Podilchuk and W. Zeng. Perceptual watermarking of still images. In *Proc. Electronic Imaging*, volume 3016, San Jose, CA, USA, February 1996.

[71] C. I. Podilchuk and W. Zeng. Image-adaptive watermarking using visual models. *IEEE Journal on Selected Areas in Communications*, 16(4):525–539, May 1998.

[72] J. G. Proakis. *Digital Communications*. McGraw-Hill, 1995.

[73] J. G. Proakis and D. G. Manolakis. *Introduction to Digital Signal Processing*. Maxwell Macmillan Publishing Company, New York, 1989.

[74] J. Puate and F. Jordan. Using fractal compression scheme to embed a digital signature into an image. In *Proceedings of SPIE Photonics East'96 Symposium*, November 1996.

[75] G. B. Rhoads. Steganography systems. In *International Patent WO 96/36163 PCT/US96/06618*, November 1996.

[76] C. B. Rorabaugh. *Error Coding Cookbook*. The McGraw-Hill Companies, 1996.

[77] J. Rovamo, J. Mustonen, and R. Nasanen. Modelling contrast sensitivity as a function of retinal illuminance and grating area. *Vision Research*, 34(10):1301–1314, 1994.

[78] J. J. K. Ó Ruanaidh, H. Petersen, A. Herrigel, S. Pereira, and T. Pun. Cryptographic copyright protection for digital images based on watermarking techniques. *Theoretical Computer Science*, 226(1–2):117–142, 17 September 1999. (Special Issue: Cryptography, C. Ding, Ed.).

[79] A. Said and W. A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:243–250, June 1996.

[80] J. W. Senders. Distribution of attention in static and dynamic scenes. In *Proceedings SPIE 2016*, pages 186–194, San Jose, February 1997.

[81] S. Servetto, C. Podilchuk, and K. Ramchandran. Capacity issues in digital image watermarking. In *IEEE Int. Conference on Image Processing '98 Proceedings*, Chicago, Illinois, USA, October 1998. Focus Interactive Technology Inc.

[82] C. E. Shannon. A mathemtaical theory of communications. *Bell Syst. Tech. J*, 27(3):379–423, October 1948.

[83] J. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41:3445–3462, December 1993.

[84] H. S. Stone. Analysis of attacks on image watermarks with randomized coefficients. Technical report, NEC Res. Inst., Princeton, New Jersey, May 1996.

[85] M. D. Swanson, B. Zhu, and A. H. Tewfik. Multiresolution scene-based video watermarking using perceptual models. *IEEE Journal on Selected Areas in Communications*, 16(4):540–550, May 1998.

[86] M. D. Swanson, B. Zhu, and A.H. Tewfik. Robust data hiding for images. In *7th IEEE Digital Signal Processing Workshop*, pages 37–40, Loen, Norway, September 1996. G:WM1-A23.

[87] K. Tanaka, Y. Nakamura, and K. Matsui. Embedding secret information into a dithered multilevel image. In *IEEE Military Commun. Conf.*, pages 216–220, September 1990.

[88] A. Z. Tirkel, C.F. Osborne, and T.E. Hall. Image and watermark registration. *Signal processing*, 66:373–383, 1998.

[89] A. Z. Tirkel, G. A. Rankin, R. G. van Schyndel, W. J. Ho, N. R. A. Mee, and C. F. Osborne. Electronic watermark. In *Dicta-93*, pages 666–672, Macquarie University, Sydney, December 1993.

[90] A. Z. Tirkel, R. G. van Schyndel, and C. F. Osborne. A two-dimensional digital watermark. In *Dicta-95*, pages 378–383, University of Queensland, Brisbane, December6-8 1995.

[91] Unzign watermark removal software. Technical report, http://altern.org/watermark/, July 1997.

[92] M. Vetterli and J. Kovacević. *Wavelets and Subband Coding*. Prentice Hall, 1995.

[93] S. Voloshynovskiy, F. Deguillaume, and T. Pun. Content adaptive watermarking based on a stochastic multiresolution image modeling. In *EUSIPCO 2000*, Tampere, Finland, submitted 2000.

[94] S. Voloshynovskiy, A. Herrigel, N. Baumgaertner, and T. Pun. A stochastic approach to content adaptive digital image watermarking. In *Third International Workshop on Information Hiding*, Dresden, Germany, September 29 - October 1st 1999.

[95] S. Voloshynovskiy, S. Pereira, A. Herrigel, N. Baumgärtner, and T. Pun. A generalized watermark attack based on stochastic watermark estimation and perceptual remodulation. In Ping Wah Wong and Edward J. Delp, editors, *IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II*, volume 3971 of *SPIE Proceedings*, San Jose, California USA, 23–28 January 2000. (Paper EI 3971-34).

[96] G. Voyatzis and I. Pitas. Protecting digital image copyrights: A framework. *IEEE Computer Graphics and Applications*, 19(1):18–23, January 1999.

[97] G. Voyatzis and I. Pitas. The use of watermarks in the protection of digital multimedia products. *Proceedings of the IEEE*, 87(7), July 1999.

[98] A. B. Watson. Dct quantization matrices visually optimized for individual images. In *Proc. SPIE:Human vision, Visual Processing and Digital Display IV*, volume 1913, pages 202–216. SPIE, 1993.

[99] M. Wu, M. L. Miller, J. A. Bloom, and I. J. Cox. A rotation, scale and translation resilient public watermark. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Phoenix, Arizona, August 1999.

[100] A. L. Yarbus. *Eye movements and vision.* Plenum Press, New York, NY, 1967.