Protection des contenus multimédias pour la certification des données

Pascal Lefèvre

THÈSE

POUR L'OBTENTION DU GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE POITIERS

FACULTÉ DES SCIENCES FONDAMENTALES ET APPLIQUÉES
DIPLÔME NATIONAL - ARRÊTÉ DU 25 MAI 2016

Ecole Doctorale : Sciences et Ingénierie pour l'Information, Mathématiques - S2IM
Secteur de Recherche : Traitement du Signal et des Images

Présentée par :
Pascal LEFÈVRE

Protection des contenus multimédias pour la certification des données

Directeurs de Thèse :
Philippe CARRÉ
Philippe GABORIT

Soutenue le 15 Juin 2018

Devant la Commission d'Examen

JURY

Caroline FONTAINE, Chargée de recherche, HDR, CNRS/Lab-STICC/CID ..... Rapporteur

William PUECH, Professeur des Universités, Université de Montpellier, LIRMM ..... Rapporteur

David ALLEYSSON, Chargé de recherche, LPNC, Université Grenoble-Alpes ..... Examinateur

Jean-Luc DUGELAY, Professeur des Universités, EURECOM Nice ..... Examinateur

Philippe CARRÉ, Professeur des Universités, Université de Poitiers ..... Directeur de Thèse

Philippe GABORIT, Professeur des Universités, Université de Limoges ..... Co-directeur

Remerciements

Pendant ces trois ans et demi de thèse au laboratoire XLIM de Poitiers, j'ai acquis une expérience exceptionnelle en produisant les travaux de recherche que je vous présente dans ce manuscrit. J'y ai tellement appris et évolué qu'il m'est impossible de ne pas exprimer aujourd'hui ma profonde gratitude à toutes les personnes que j'ai rencontré et qui m'ont permis d'avancer.

Je remercie tout d'abord mon directeur de thèse Philippe Carré, Professeur des universités à l'institut XLIM de l'Université de Poitiers de m'avoir accepté et de m'avoir guidé tout au long cette aventure sur le très intéressant sujet que représente le tatouage numérique. Sa patience, son soutien ainsi que ses précieux conseils m'ont permis de mener à bien ces travaux de thèse. Je remercie également mon co-encadrant de thèse Philippe Gaborit, Professeur des universités à l'institut XLIM de l'Université de Limoges pour son inspiration et toutes nos discussions sur les codes correcteurs.

Je voudrais ensuite remercier Caroline Fontaine, Chargée de recherche à l'IMT Atlantique et William Puech, Professeur des universités à l'Université de Montpellier d'avoir accepté de rapporter ce manuscrit. Je remercie également David Alleysson, Chargé de recherche au LPNC à l'Université Grenoble-Alpes et Jean-Luc Dugelay, Professeur des universités à EURECOM Nice d'avoir accepté d'examiner ce manuscrit.

Et aussi, je voudrais remercier la Délégation Générale de l'Armement ainsi que l'IUT de Poitiers d'avoir financé cette thèse.

Je remercie également les personnes avec lesquelles j'ai pu partagé mon bureau, Fan, Hermine, Lydie, Mathieu, Weiping et bien d'autres pour les nombreux moments passés ensemble, pour toutes les discussions et les échanges côté boulot mais aussi côté détente. J'exprime ma gratitude envers toutes les personnes avec qui j'ai pu partagé de bons moments au laboratoire notamment au «café social» et pendant les sorties de l'association ADIIS. Je voudrais remercier en particulier Charlie pour tous les moments que nous avons partagé.

Et finalement, c'est avec une attention particulière que je remercie ma femme Tingting de m'avoir accompagné et soutenu pendant ces quatres longues années passées au laboratoire. Un grand merci à ma famille qui m'a soutenu et qui m'a permis d'accomplir, de près ou de loin, ces travaux.

À ma femme,
À mes parents

Table des matières

Glossaire

Mesures

MSE : Mean Squared Error
PSNR : Peak Signal to Noise Ratio
ER : Embedding Rate
DWR : Document to Watermark Ratio
WNR : Watermark to Noise Ratio
BER : Binary Error Rate
SER : Symbol Error Rate
IER : Image Error Rate

Variables et notations

$\mathcal{X}$: Image hôte
$\mathcal{Y}$: Image marquée
$\mathcal{Z}$: Image modifiée par une attaque
$\mathcal{C}_{\mathcal{X}}$: Coefficients extraits de $\mathcal{X}$
$\mathcal{C}_{\mathcal{Y}}$: Coefficients modifiés correspondant à $\mathcal{Y}$
$\mathcal{C}_{\mathcal{Z}}$: Coefficients extraits de $\mathcal{Z}$
$\mathcal{S}_{\mathcal{X}}$: Image scalaire issue de $\mathcal{X}$
$\mathcal{S}_{\mathcal{Y}}$: Image scalaire issue de $\mathcal{Y}$
$\mathcal{S}_{\mathcal{Z}}$: Image scalaire issue de $\mathcal{Z}$

Codes correcteurs

d_H : Distance de Hamming
d_R : Distance rang
w_H : Poids de Hamming
w_R : Poids rang ou rang matriciel d'un mot de code en métrique rang

Méthode QIM

QIM : Quantization Index Modulation, Modulation d'Index
LQIM : Lattice QIM
Δ : Pas de quantification
$\mathbb{Z}^L$: Réseau euclidien de dimension L
Q_m : Quantificateur LQIM

Normes

$\|.\|_\infty$: Norme infinie
$\|.\|_1$: Norme L_1
$\|.\|_2$: Norme euclidienne

Paramètres d'attaques

q : facteur de qualité d'une compression JPEG
α : gain multiplicatif d'une modification de contraste
β : constante additive correspondant à une modification de luminance
σ : Paramètre de déviation standart σ d'un bruit additif blanc gaussien

Approches et méthodes de tatouage couleur

SVH : Système Visuel Humain
lms : Espace de transduction
LMS : Espace d'excitation
P_{RGB} : Pixel couleur P représenté dans l'espace RGB
P_{lms} : Pixel couleur P représenté dans l'espace lms
P_{LMS} : Pixel couleur P représenté dans l'espace LMS
u_P Axe de direction associé au pixel couleur P
Approche GA : Approche de base utilisant un axe de direction fixe
Approche AA : Approche psychovisuelle utilisant un axe de direction adaptatif

Table des figures

Liste des tableaux

Introduction générale

Motivations

La révolution du numérique se caractérise par une forte démocratisation des ordinateurs et des objets connectés, ainsi que par un nombre toujours plus importants d'utilisateurs (le nombre de foyers ayant accès à internet est passé de 12% en 2000 à 64% en 2010, selon Vincent Gombault, Insee, mars 2011). Ces internautes disposent aujourd'hui de nombreux moyens et outils, pour lesquels il n'y a plus obligatoirement besoin de formation pour visualiser et modifier des contenus digitaux.

Avec les progrès en matière de stockage en ligne, la quantité de contenus digitaux tels que le son, l'image ou la vidéo sur internet a explosé et continue d'augmenter. La mise à disposition gratuite de stockage d'images (banque d'images telle que *Getty Images*, réseaux sociaux tels que *Instagram*) et de plateformes mettant à disposition du contenu vidéos (*Youtube*, *Dailymotion*, etc) et de streaming (*Twitch*, etc) ne fait qu'accélerer le phénomène.

Ainsi, une image qui « fait le buzz » peut être copiée des millions de fois et subir un nombre de modifications non contrôlables, souvent intraçables et parfois malicieuses. La protection des données numériques contre les altérations, volontaires ou non, la copie ou le vol revêt des enjeux importants pour de nombreuses activités telles que le commerce, la communication (publique ou privée, politique ou militaire, etc) quand il ne s'agit pas tout simplement d'identifier et de protéger la propriété intellectuelle et industrielle.

Savoir identifier la source d'une image et/ou certifier si celle-ci a été modifiée sont des informations nécessaires pour authentifier une image. Plusieurs approches permettent de répondre à ces besoins d'authentification. Dans ce travail, nous proposons de nous intéresser au tatouage numérique, consistant à insérer une marque dans une image, et d'analyser différents aspects de cette problématique. Comme nous allons le voir, nous étudions le tatouage numérique dans le but de le rendre plus robuste aux modifications d'image grâce à l'utilisation de codes correcteurs. Ensuite, nous proposons d'améliorer l'invisibilité des méthodes de tatouage en étudiant la perception de la vision en couleur du système visuel humain.

Contexte de recherche

Le tatouage robuste des images est une solution fiable permettant de protéger des contenus multimédias. Dans ce cadre, nous proposons d'étudier deux aspects fondamentraux liés au tatouage d'image. Tout d'abord, nous proposons une étude de la robustesse d'une méthode de tatouage et notamment en intégrant de manière originale un puissant outil issu de la théorie de l'information qui est le code correcteur. Cette approche n'est pas nouvelle, par exemple des travaux datant des années 2000 ([1, 2], etc) ont été proposé dans la litérature ainsi que des travaux menés au sein du laboratoire XLIM ([3]) ont soutenu l'idée qu'un code correcteur améliorait la robustesse du tatouage et pouvait être plus efficace qu'un autre selon la structure de l'erreur à laquelle on est confronté, en fonction de l'attaque subie par l'image.

Dans ce cadre, nos travaux se sont portés sur la mise en évidence, la compréhension de la forme de l'erreur et la manière dont nous pouvons gérer au mieux le décodage de ces erreurs grâce à un choix adapté de codes correcteurs classiques de Hamming. Cette discussion de la forme de l'erreur nous a permis de nous interroger sur la stratégie de détection et notamment l'influence de la norme utilisée. Par la suite, nous nous intéressons à des codes en *métrique rang*, codes déjà utilisés en cryptographie et en télécommunications, mais inconnus dans le domaine du tatouage numérique. Cette métrique fonctionne différement de la distance classique de Hamming. Bien qu'elle soit *moins précise* ou plus *grossière* pour évaluer une distance, la métrique rang admet des propriétés originales offrant un nouveau type de robustesse. Afin d'expliciter ces propriétés, nous décrirons comment utiliser les codes en métrique rang pour le tatouage numérique.

En second temps, nous nous sommes intéressés à la manière dont nous pouvons améliorer l'invisiblité d'une marque, notamment dans le contexte d'une image couleur. À l'insertion d'une marque, l'image hôte subit des distorsions qu'il est nécessaire de maîtriser afin d'obtenir une invisibilité statisfaisante et un contenu hôte de qualité adéquate. Pour cela, nous avons choisi une approche perceptuelle qui consiste à contrôler les distorsions d'insertion en fonction de la zone de l'image. Quand le système visuel humain observe une image, certaines zones seront plus sensibles à la perception de modifications que d'autres. C'est pourquoi nous proposons d'étudier la perception des couleurs par le système visuel humain grâce à un modèle biologique. En déterminant le niveau de sensibilité du système visuel humain pour chaque couleur, nous avons pu proposer une méthode de tatouage permettant de réduire psychovisuellement le bruit d'insertion.

Organisation du manuscrit

Après une introduction générale, ce manuscrit délivre quatre chapitres. Le premier contient une introduction au tatouage numérique et aux codes correcteurs d'erreur ainsi qu'un état de l'art des contributions mélangeant les deux domaines. Nous proposons un premier exemple d'amélioration de la robustesse d'une méthode de tatouage en étudiant la forme de l'erreur à laquelle nous sommes confrontés en utilisant les codes classiques de Hamming.

Le second chapitre décrit l'étude et la mise en évidence de la forme de l'erreur selon les attaques puis décrit l'impact de la norme sur le détecteur de la méthode Lattice QIM définie dans le premier chapitre. Nous détaillerons notamment l'interaction possible entre les codes en norme infinie et la méthode QIM.

Le troisième chapitre introduit de codes correcteurs appelés *code en métrique rang* (des explications et des exemples détaillées peuvent être consultés en annexe). Nous proposons l'application de ces codes dans une stratégie de tatouage numérique. Nous montrons alors que les codes de Gabidulin (qui sont une famille de codes en métrique rang) associés à un détecteur Lattice QIM amélioré sont théoriquement invariants aux modifications de luminance. Puis, nous montrons que cette nouvelle méthode de tatouage est résistante au découpage d'image (ou *image cropping*) en utilisant une décomposition par blocs de l'image.

Enfin, le dernier chapitre aborde le second aspect du tatouage à savoir l'invisibilité. Après un état de l'art du tatouage couleur, ce chapitre propose l'étude et la conception d'une méthode de tatouage psychovisuelle robuste basée sur une modélisation du système visuel humain. Une méthode de quantification vectorielle dans le domaine RGB selon une direction est associée avec notre modèle psychovisuel. Celui-ci permet de minimiser les distorsions couleur provoquées par l'insertion d'une marque dans une image, c'est à dire que nous pouvons améliorer le compromis robustesse/invisibilité par rapport à une quantification vectorielle suivant l'axe des niveaux de gris.

Ce manuscrit s'achève sur une conclusion générale que nous complétons par une ouverture vers des perspectives.

CHAPITRE 1

État de l'art pour le tatouage numérique et les codes correcteurs

Contenu

1.1 Introduction

Le tatouage numérique a fait l'objet de plus de trente ans de recherche et un très grand nombre de contributions scientifiques sont disponibles à ce jour. Le tatouage numérique est défini comme une modification imperceptible d'un contenu pour y insérer de l'information en relation ou non avec ce contenu [4]. Cependant, cette propriété d'imperceptibilité n'est pas toujours un prérequis selon les applications. Le tatouage visible constitue un domaine de recherche bien différent du sujet auquel nous nous intéressons [5, 6].

Selon les applications, le tatouage numérique peut avoir des propriétés très différentes. Il constitue à lui seul un domaine de la sécurité informatique qui se concentre principalement sur les contenus multimédias. Il se trouve très proche de la stéganographie sur bien des aspects bien que leur objectif soit différent.

En effet, la stéganographie est l'art de transmettre un message de manière furtive dans un contenu hôte alors que le tatouage sert à protéger un contenu de valeur tout en étant invisible. D'un point de vue pratique, nous pouvons dire que le tatouage numérique et la stéganographie respecte un schéma générique (figure 1.1).

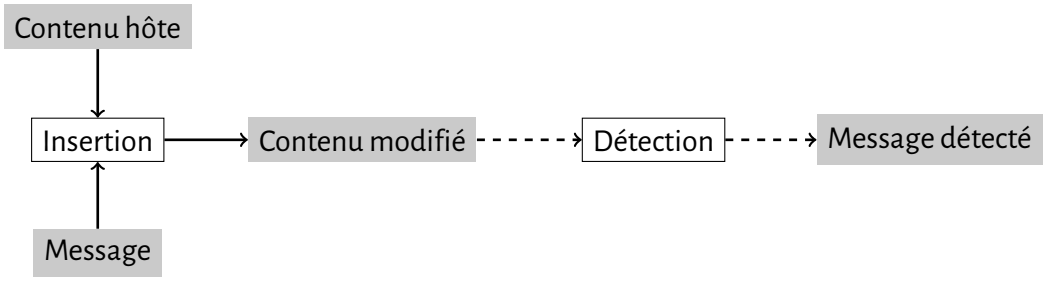


Figure 1.1 — Schéma générique d'insertion de données cachées. La flèche en pointillé indique l'étape de transmission du contenu modifié au destinataire.

Dans la première phase, un contenu hôte (un son, une image, une vidéo, ...) est modifié pour contenir un message grâce à la méthode d'insertion. Le contenu résultant est le contenu marqué. Lorsque celui-ci est transmis à son destinataire, la méthode de détection associée est utilisée sur le contenu reçu pour finalement en extraire une estimation du message originel.

Depuis les années 1990, il y a un intérêt de plus en plus fort pour protéger les contenus digitaux, surtout dans le domaine de la photographie, de la vidéo mais aussi dans le texte et les modèles 3D (les maillages 3D par exemple [7, 8, 9, 10]).

Certaines applications vont encore plus loin en allant jusqu'à marquer du code executable

([11]) et même du matériel ([12]). Quelques exemples pratiques d'application au tatouage numérique [13, 4] sont la protection des droits d'auteur [14, 15], la détection de modification de contenu [16, 17, 18] ou encore la surveillance de diffusion de contenus [19].

Dans ce chapitre, nous proposons tout d'abord une introduction au tatouage numérique des images (section 1.2) en nous concentrant sur le tatouage robuste. Ensuite, nous introduisons notre approche générale (section 1.6) permettant d'améliorer la robustesse d'une méthode de tatouage qui est la recherche et l'étude de la structure des erreurs qui endommagent le tatouage d'une image marquée transmise sur un canal (i.e. modifiée par une attaque). Les codes correcteurs de Hamming classiques ainsi que les codes BCH et les codes de Reed-Solomon sont présentés également. Notons que la discussion concernant l'invisibilité et notamment l'intégration de la modélisation du système visuel humain sera faite dans le dernier chapitre.

1.2 Tatouage numérique

Les besoins en protection des contenus multimédias ne cessent de grandir et de plus en plus rapidement à cause de la rapidité d'évolution des nouvelles technologies. Au cours des années, plusieurs paradigmes de tatouage ont émergé du aux exigences de protection de nombreuses applications. Dans cette section, nous décrivons brièvement trois paradigmes de tatouage numérique.

1.2.1 Tatouage fragile, semi-fragile et tatouage robuste

Il est très facile de modifier ou falsifier des images de nos jours grâce à l'accès tout public aux logiciels de traitement d'image. Vérifier l'intégrité et authentifier une image sont donc des problématiques importantes.

L'idée du tatouage fragile est d'insérer un motif indépendant du contenu de l'image de telle sorte que toute modification de l'image modifie le motif. Les régions modifiées peuvent donc être détectées en analysant le motif. Le désavantage de ce paradigme de tatouage est qu'il n'est pas possible de distinguer entre un contenu modifié par inadvertance ou sans mauvaise intention et un contenu malicieusement modifié. Par exemple, les plus anciennes méthodes classent une image compressée dans la catégorie des images falsifiées alors que la sémantique

de celle-ci n'a pas changé.

L'approche utilisée par ces méthodes est de protéger les bits de poids forts de l'image en les insérant sur les bits de poids faibles sur les coefficients DCT. Des exemples de travaux peuvent être trouvés dans [20, 21, 22, 23, 24].

Un paradigme similaire est le tatouage semi-fragile. Il se caractérise comme étant plus robuste que le tatouage fragile et permet de traiter les mêmes problématiques liées aux images. Les travaux sur le tatouage semi-fragile [16, 17, 18] ont été développés pour pouvoir distinger les modifications malicieuses de celles qui ne le sont pas. Des exemples de tentatives de falsifications d'images sont l'ajout et le retrait (changement local de la sémantique de l'image) de certaines régions de l'image.

Ces deux derniers paradigmes sont en opposition avec le tatouage robuste qui, lui, a pour objectif de résister aux modifications d'images. Les objectifs du tatouage fragile et semi-fragile diffèrent de ceux du tatouage robuste. Et même si les techniques de tatouage semi-fragile détectent les modifications malicieuses c'est-à-direqu'elles résistent à la compression JPEG, leur construction est différente de celle des techniques de tatouage robuste. Ces dernières sont, par définition, idéalement faites pour résister à tout type de modifications d'images afin d'extraire une information sans erreur.

Malgré le fait que le tatouage fragile et semi-fragile permettent de préserver une qualité de l'image hôte plus que suffisante, certaines applications nécessitent que l'image hôte ne soit pas modifiée.

1.2.2 Tatouage réversible et tatouage non-réversible

Ce paradigme de tatouage permet d'insérer une marque qui peut être effacée, i.e., on peut retrouver l'image d'origine intacte. Comme dans tous les paradigmes présentés, l'image à marquer subit des distorsions. Cependant, l'étape de détection permet non seulement d'extraire la marque mais aussi de retrouver les valeurs des pixels d'origine. Des applications sensibles telles que la médecine et les communications militaires nécessitent l'extraction complète de la marque afin de restaurer la qualité originelle de l'image. Une approche basée sur la compression sans perte est utilisée dans [25, 26]. D'autres méthodes appelées *difference expansion* [27, 28, 29] et *histogram shifting* [30, 31] permettent aussi faire du tatouage réversible (tatouage effaçable). Dans nos travaux de thèse, nous n'étudions pas de méthodes de tatouage réversible.

Nous proposons maintenant d'introduire le tatouage robuste qui est le paradigme sur lequel nos travaux se basent. En utilisant les codes correcteurs, l'objectif sera d'améliorer la robustesse face à diverses attaques.

1.3 Tatouage aveugle et non-aveugle

Pour certaines applications, le contenu hôte original non marqué est disponible à l'étape de détection d'une marque. Ces méthodes de tatouage sont alors dites *non-aveugle* (ou *non-blind*). En l'occurence, elles ont de meilleures performances puisque la marque seule peut être extraite de l'image marquée. Par exemple, un utilisateur pourra plus facilement identifier les distorsions subies par l'image et ainsi faciliter la détection de la marque.

Par ailleurs, il existe des situations où le contenu hôte n'est pas accessible à l'utilisateur qui souhaite détecter la marque. Les méthodes permettant d'extraire le message d'une marque sans le contenu original sont dites *aveugle* (ou *blind*). Dans la littérature, le terme *méthode de tatouage publique* est aussi employé par opposition aux *méthodes de tatouage privée*, qui, quant à elles, sont réservées à un groupe restreint d'utilisateur (typiquement des utilisateurs qui possède le contenu hôte légalement).

1.4 Tatouage robuste en détails

Dans cette partie, nous expliquons en détails en quoi consiste le tatouage robuste. De nombreuses applications telles que la protection des droits d'auteur nécessitent l'extraction parfaite d'une marque insérée dans une image qui a été modifiée. Lorsque la marque extraite est identique à celle d'origine, elle est dite robuste. En pratique, une méthode de tatouage numérique est résistante à une attaque dans une certaine mesure et on considère que lorsqu'une image est suffisament dégradée, i.e., n'est plus de valeur satisfaisante, l'extraction de la marque perd son intérêt.

Par exemple, la compression JPEG est une attaque très courante dans la vie d'une image. Pour des besoins de transmission efficace, la quantité de données nécessaire pour représenter une image est réduite en sous échantillonnant l'information de chrominance mais ce processus dégrade celle-ci et la marque qu'elle contient. Concrètement, nous pouvons considérer

qu'une image compressée à 50% n'a plus de valeur méritant une protection. Dans cette section, nous proposons une introduction au tatouage robuste.

1.4.1 Schémas d'insertion et scénario

Le paradigme du tatouage robuste admet un scénario en trois étapes : l'étape d'insertion de la marque dans une image hôte, la transmission de l'image marquée sur un canal bruité (bruité dans le sens image modifiée par une attaque) et enfin l'étape de détection. Nous détaillons chaque étape dans cette section.

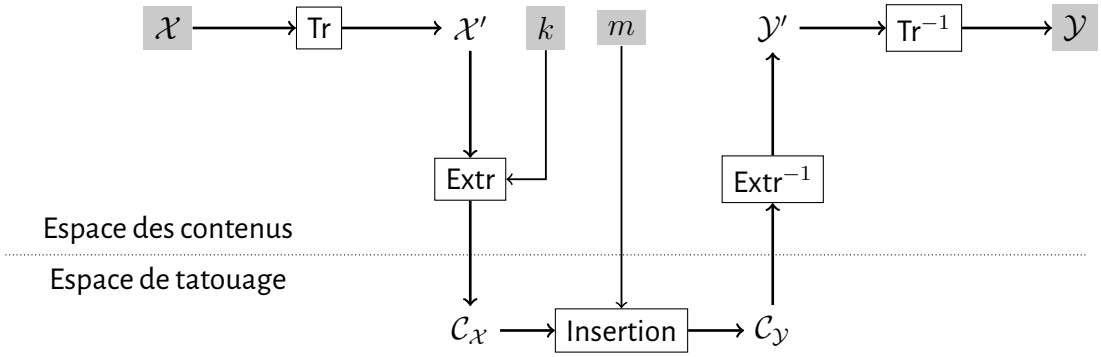


FIGURE 1.2 – Schéma d'insertion de tatouage classique.

Nous avons premièrement représenté le schéma d'insertion classique d'une marque dans la figure 1.2. Une image $\mathcal{X}$ choisie comme hôte de la marque peut être de différents types, i.e., en niveaux de gris ou en couleur (définie dans l'espace RGB ou autres espaces couleur). Selon les méthodes, $\mathcal{X}$ peut changer de représentation (coefficients DCT, coefficients ondelettes, etc). De l'image transformée $\mathcal{X}'$, des coefficients notés $\mathcal{C}_{\mathcal{X}}$ sont extraits de $\mathcal{X}'$ avec une méthode d'extraction Extr (sélection des sites d'insertion de manière aléatoire, décomposition de l'image en bloc) grâce à une clé secrète k.

Ensuite, le message m est inséré en modifiant $\mathcal{C}_{\mathcal{X}}$ dans l'espace de tatouage qui n'est accessible qu'avec la clé k pour sécuriser l'accès à cet espace. Les coefficients $\mathcal{C}_{\mathcal{Y}}$ sont alors réintégrés dans l'image transformée notée $\mathcal{Y}$ et enfin nous obtenons l'image marquée $\mathcal{Y}$ avec la transformée inverse Tr^{-1}.

Dans la deuxième étape de ce scénario, l'image marquée $\mathcal{Y}$ est transmise sur un canal bruité $\mathcal{C}$ (figure 1.3). Le bruit n représente les diverses attaques possibles sur un canal. Quelques exemples les plus connus sont la compression JPEG et le bruit gaussien.

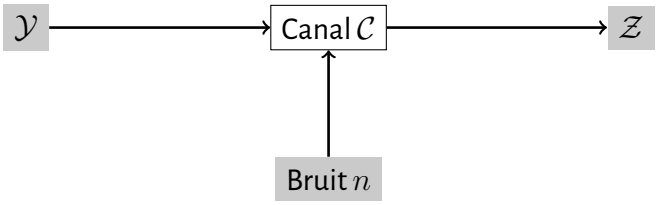


FIGURE 1.3 – Schéma de transmission d'une image sur un canal affecté par un bruit n.

Nous venons de décrire les différentes étapes pour insérer de l'information dans une image en niveaux de gris, associées à un scénario de transmission avec canal bruité.

La robustesse de cette information face à une attaque donnée dépend de plusieurs critères :

— la représentation de l'image

— la stratégie d'extraction des coefficients à modifier

— la méthode de tatouage choisie

— les paramètres (pas de quantification, nombre d'échantillons)

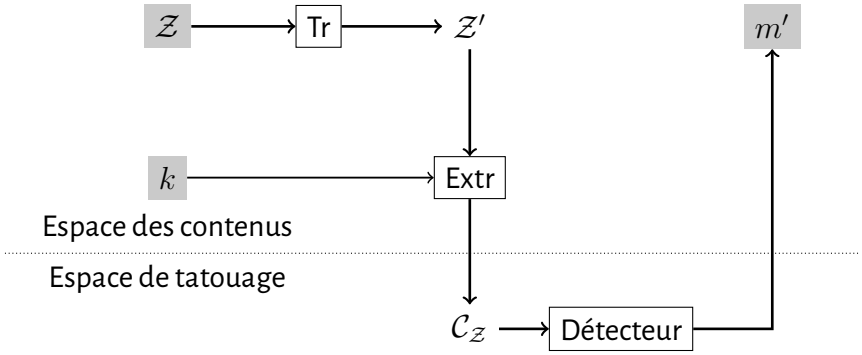


FIGURE 1.4 – Schéma classique de détection d'une marque.

À l'étape de détection (figure 1.4), nous accédons au canal du tatouage grâce à la fonction d'extraction Extr et à la clé k de l'image $\mathcal{Z}$. Puis, nous appliquons aux coefficents $\mathcal{C}_{\mathcal{Z}}$ la méthode de détection associée. Si la puissance de l'attaque est raisonnable, l'estimation m' est identique à m.

L'espace d'insertion est à choisir en fonction de l'attaque à laquelle la marque doit résister. Changer la représentation de l'image permet de profiter de certaines propriétés de résistances à certaines attaques. Par exemple, une résistance à un filtre passe-bas peut être obtenue en

modifiant des coefficients DCT correspondant aux basses fréquences. Inversement pour un filtre passe-haut en modifiant des coefficients DCT hautes fréquences.

Un autre point essentiel à considérer est la synchronisation de la marque c'est-à-dire le choix des sites d'insertion de celle-ci dans une image. Si les attaques auxquelles nous sommes confrontés ont une influence sur la position des coefficients, il est nécessaire d'en tenir compte à l'étape d'insertion. Par exemple, nous verrons dans le chapitre sur les codes en métrique rang qu'une décomposition par blocs de l'image permet à la méthode proposée d'être résistante au découpage d'image.

En fait, les notions d'espace transformé et de synchronisation ne sont pas si différentes. Par exemple, dans le domaine spatial, insérer de l'information sur des niveaux de gris représentant des zones de contours (choix des coefficients selon une propriété fixée) revient à insérer de l'information dans les hautes fréquences. Un résultat similaire peut être obtenu en modifiant les coefficients DCT hautes fréquences même nous perdons la notion spatiale. L'information de haute fréquence est donc résumée sur certains coefficients mais nous ne pouvons plus savoir spatialement quels sont les pixels modifiés. Cependant, il existe d'autres exemples où espace transformé et synchronisation se distinguent plus clairement.

À l'origine de ces travaux, nous avons choisi de concentrer nos études sur une méthode de tatouage par quantification appelée Modulation d'Index [32] (ou QIM) pour ses performances et ses perspectives de recherche prometteuses une fois combinée avec des codes correcteurs. En effet, la Modulation d'Index possède de meilleures performances que les méthodes linéaires d'étalement de spectre précédemment proposées [33, 34] en terme de capacité et de robustesse.

1.4.2 Modulation d'index

En 1999, Chen et Wornell ont proposé la Modulation d'Index (ou *méthode QIM*) pour faire du tatouage par quantification [32]. Cet algorithme est connu pour sa facilité d'implémentation et son faible coût en calculs mais aussi pour sa fragilité face aux attaques valumétriques de changement d'échelle. Dans [35], Pérez González et al. ont proposé une amélioration appelée *Rational Dither Modulation* pour mieux résister à cette dernière attaque.

Chen et Wornell ont également proposé des variantes telles que *Dither Modulation (DM) QIM* et *Distortion Compensated (DC) QIM* ainsi qu'une étude théorique associée. La méthode DM-

QIM est une implémentation de la méthode QIM qui décale les cellules de quantification avec un vecteur aléatoire [36, 37]. Les valeurs modifiées sont plus difficiles à identifier et rend l'accès au canal de tatouage plus difficile. La méthode DCQIM est composée d'une étape de traitement ajoutée après la quantification QIM afin d'améliorer le compromis invisibilité/robustesse.

Les auteurs proposent également une variante vectorielle appelée *Spread Transform Dither Modulation* qui consiste à quantifier la projection des vecteurs échantillons sur un axe de direction. Chen et Wornell ont proposé une seconde extension de la méthode QIM appelée *Lattice QIM* [38], méthode que nous avons choisi d'utiliser dans nos travaux de thèse pour sa structure régulière (réseau euclidien) et son potentiel de robustesse une fois associée aux codes correcteurs. Les contributions utilisant la méthode QIM et ses variantes sont nombreuses (par exemple [39, 40, 41, 42, 43, 3]).

Nous proposons maintenant dans les sous-sections suivantes d'introduire la méthode *Lattice QIM (LQIM)*.

1.4.3 Insertion

L'espace de quantification de cette méthode est un réseau euclidien ($\mathbb{Z}^L$) de dimension L où un échantillon hôte $x \in \mathbb{R}^L$ est transformé en un élément y de ce réseau. La méthode QIM scalaire est le cas particulier $L = 1$.

L'échantillon x est construit en sélectionnant des coefficients de l'image. Pour insérer un bit d'information, L valeurs de pixel seront sélectionnées dans l'image. Par exemple, une stratégie de sélection aléatoire de ces valeurs peut être adoptée. Ces sites d'insertion sont secrets et accessibles uniquement de l'émetteur et du récepteur grâce à une clé secrète.

Pour insérer un bit d'information $m \in \{0, 1\}$, x est quantifié en y appartenant à l'un des sous-ensembles suivants appelés *cosets* notés Λ_0 et Λ_1 définis tels que :

$$\begin{cases} \Lambda_0 = \Delta\mathbb{Z}^L - \dfrac{\Delta}{4} \\ \Lambda_1 = \Delta\mathbb{Z}^L + \dfrac{\Delta}{4} \end{cases} \tag{1.1}$$

grâce à la fonction de quantification Q_m suivante :

$$y = Q_m(x, \Delta) = \left\lfloor \frac{x}{\Delta} \right\rfloor \Delta + (-1)^{m+1} \frac{\Delta}{4} \tag{1.2}$$

avec Δ le pas de quantification de la méthode. La figure 1.5 est un exemple de quantification en dimension $L = 2$ d'un vecteur x. Pour toute croix ou cercle de centre y_m, les losanges en pointillés délimitent les frontières de chaque cellule de quantification. À l'insertion, x est transformé en un y_m le plus proche en distance euclidienne.

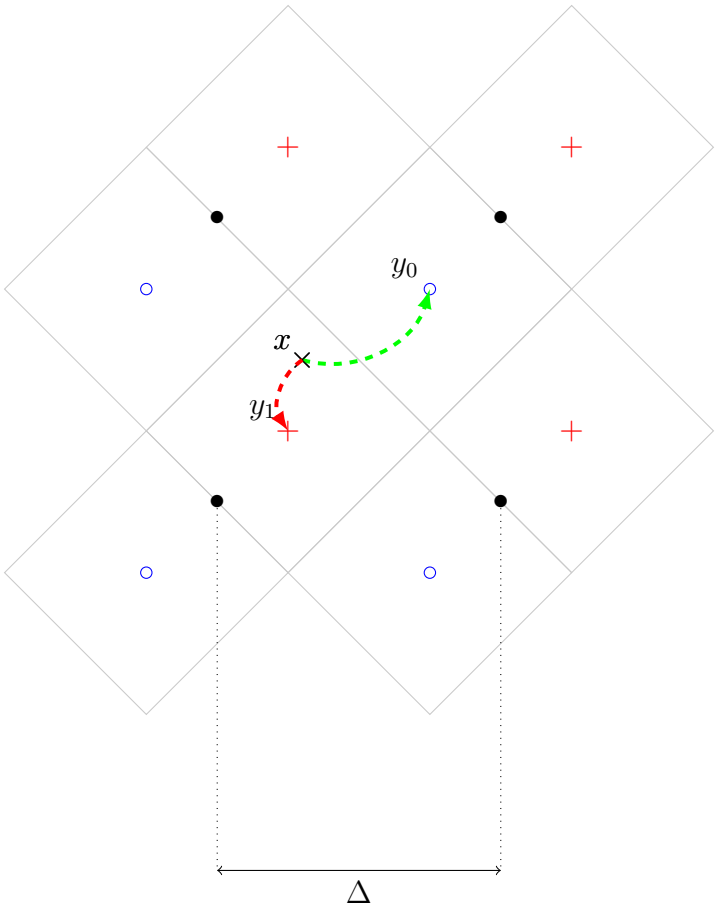


FIGURE 1.5 – Représentation de l'espace de quantification (ou réseau euclidien) en dimension $L = 2$. Le symbole $+$ représente le bit 1 (coset Λ_1) et les $\circ$ le bit 0 (coset Λ_0).

1.4.4 Détection

À la réception d'une image qui a été marquée puis modifiée, le message peut être détecté en faisant une extraction des coefficients utilisés à l'insertion grâce à la clé secrète. Pour ex-

traire le bit d'information m, on calcule le coset le plus proche du vecteur z associé pour déterminer l'estimation $\hat{m}$ telle que :

$$\begin{aligned} \hat{m} &= \underset{m\in\{0,1\}}{\arg\min}\, dist(z, \Lambda_m), \\ dist(z, \Lambda) &= \min_{y\in\Lambda} \|z - y\|_2 \end{aligned} \tag{1.3}$$

avec $\|.\|_2$ la norme euclidienne.

L'équation de détection 1.3 s'illustre avec la figure 1.5. Une estimation $\hat{m}$ du bit m est calculée en cherchant le coset $\Lambda_{m'}$ le plus proche de z (plus précisément le vecteur y le plus proche de z).

Nous avons décrit dans cette sous-section la partie Insertion et Détection des figures 1.2 et 1.4. Dans le cas d'une application réelle, la transformation Tr et la méthode d'extraction des coefficients ou de synchronisation doivent aussi être choisies. Par exemple, un exemple simple est l'insertion dans le domaine spatial avec extraction aléatoire des pixels.

1.4.5 Objectifs contradictoires

L'objectif du tatouage robuste est d'optimiser trois propriétés : la quantité d'information maximale qu'une image peut contenir, l'invisibilité de la marque (et préservation de la qualité du contenu hôte) et la robustesse de la marque aux modifications d'image voire dans certains cas la sécurité. Cependant, lorsque l'une de ces propriétés est améliorée, les autres sont dégradées.

1.4.5.1 Invisibilité

Le tatouage exige qu'une marque soit invisible à l'oeil nu et qu'elle ne dégrade pas le contenu hôte. Pour atteindre cet objectif, les distorsions entre le contenu hôte et le contenu marqué doivent être suffisament faibles. Idéalement, une marque devrait être imperceptible pour l'oeil humain. Celle-ci peut également être invisible pour une machine c'est-à-direune invisibilité statistique mais c'est une propriété qui relève davantage de la stéganographie que du tatouage numérique.

À ce jour, les méthodes permettant de créer des marques invisibles sont basées sur des algorithmes de modulation de bits de poids faibles. Cependant, celles-ci ne sont pas robustes et, statistiquement, il est possible de détecter l'existence d'une marque.

Pour l'oeil humain, l'approche perceptuelle du tatouage numérique des images permet de contraindre les distorsions d'insertion dans les zones de l'image les plus sensibles au changement visuel. Un modèle bien connu est le modèle de Watson [44, 45]. Il permet d'adapter la luminosité en fonction de la zone de l'image qui est modifiée. Par exemple, des modifications de luminosité dans des zones claires sont moins visibles que dans les zones sombres de l'image. Une autre composante importante est le mécanisme de masquage. Par analogie avec le son, il est très difficile d'écouter une personne parler lorsque plusieurs autres personnes parlent en même temps. Le même principe peut être appliqué à l'image. Il est moins visible pour l'oeil d'insérer du bruit de hautes fréquences dans des zones texturées de l'image que dans les zones homogènes. Il est alors possible de déterminer un seuil de perception pour chaque coefficient de l'image.

Dans le dernier chapitre de ce manuscrit, nous proposons une méthode de tatouage des image couleur qui prend en compte la sensibilité du système visuel humain pour minimiser les distorsions psychovisuelles à l'insertion.

Il existe plusieurs mesures pour évaluer la distorsion entre deux images, i.e., la qualité du contenu marqué. Nous avons *l'erreur moyenne au carré noté MSE* :

$$MSE(\mathcal{X}, \mathcal{Y}) = \frac{1}{hw} \sum_{1 \leq i \leq h, 1 \leq j \leq w} (\mathcal{X}[i,j] - \mathcal{Y}[i,j])^2 \tag{1.4}$$

Avec $\mathcal{X}$ l'image hôte et $\mathcal{Y}$ l'image marquée. MSE permet de calculer le *PSNR* :

$$PSNR(\mathcal{X}, \mathcal{Y}) = 10 \log_{10}(\frac{255^2}{MSE(\mathcal{X}, \mathcal{Y})}) \tag{1.5}$$

Nous pouvons aussi mesurer le *rapport signal sur bruit noté DWR* :

$$\text{DWR}(\mathcal{X}, \mathcal{Y}) = 10 \log_{10} \frac{\sigma_{\mathcal{X}}}{\sigma_{\mathcal{Y}} - \sigma_{\mathcal{X}}} \tag{1.6}$$

avec $\sigma_{\mathcal{X}}$ la variance de l'image $\mathcal{X}$.

Cependant, ces mesures ne permettent pas d'apprécier la qualité perceptuelle d'une image marquée. Par exemple, il n'est pas possible d'évaluer la qualité d'une image du point de vue du

système visuel humain.

1.4.5.2 Capacité

La capacité est la quantité maximale d'information utile que peut contenir l'image à marquer par rapport à une méthode d'insertion donnée. En général, on utilise toujours des symboles binaires mais il est possible de coder l'information sur plus que deux symboles. Pour un symbole, un certain nombre de coefficients de l'image est modifié. Plus nous insérons d'information, plus le nombre de coefficients modifiés augmente et plus il y a de distorsions. La qualité de l'image marquée diminue.

Pour mesurer cette quantité, on peut calculer le *taux d'insertion noté ER* (embedding rate) dans le domaine spatial tel que :

$$\mathrm{ER}(\mathcal{X},\mathcal{Y}) = \frac{l}{|\{(i,j)\,|\,\mathcal{Y}[i,j] \neq \mathcal{X}[i,j], 1 \leq i \leq h, 1 \leq j \leq w\}|} \tag{1.7}$$

avec l le nombre de bit du message et le dénominateur le nombre de pixels modifiés. L'unité de cette mesure est le bit par pixel. Nous pouvons aussi faire la même mesure dans un espace transformé (bit par coefficients modifiés).

Dans la pratique, la marque de l'image $\mathcal{Y}$ est considérée comme invisible lorsque qu'on a $\mathrm{DWR}(\mathcal{X},\mathcal{Y}) \geq 35$dB. En effet, avec un taux d'insertion fixé, plus l est grand, plus il y a de pixels modifiés et donc plus de dégradations de l'image marquée par rapport à l'image d'origine.

1.4.5.3 Robustesse

La robustesse d'une marque est sa résistance à une modification de l'image, c'est-à-direque le message originel peut être détecté sans erreur.

Il existe de nombreux types d'attaques. Les attaques de désynchronisation modifient l'emplacement des sites d'insertion de la marque. Par exemple, nous avons les modifications géométriques de l'image telles qu'une translation ou une rotation. Nous avons aussi le découpage et le recadrage d'image qui changent la taille de l'image. Dans cette thèse, nous étudions un type d'attaque qui ne désynchronise pas le tatouage tel que la compression JPEG et le bruit

gaussien, qui modifient uniquement les coefficients de l'image.

Pour mesurer la robustesse, nous calculons des taux d'erreur. Une des mesures les plus communes est le *taux d'erreur binaire (TEB ou BER)* défini tel que :

$$\text{BER} = \frac{|\{i \,|\, m_i \neq m'_i\}|}{l} \tag{1.8}$$

avec $m' = (m'_1, \ldots, m'_l)$ le message détecté.

Nous utiliserons aussi le *taux d'erreur image (TEI ou IER)* défini tel que :

$$\text{IER} = \begin{cases} 0 \text{ si BER} = 0 \\ 1 \text{ sinon} \end{cases} \tag{1.9}$$

qui permet de mesurer la performance d'un code correcteur par exemple. Dans le cas d'un code correcteur binaire, le nombre d'erreur détectées ne doit pas dépasser le taux de correction du code choisi. Un taux d'erreur image nul veut donc dire que le décodage s'est déroulé sans erreur et inversement pour un taux d'erreur image égal à 1. Si nous avons affaire à des symboles non binaires, nous utiliserons un taux d'erreur symbole noté SER qui est le ratio de symboles non binaires erronés.

En pratique, lorsqu'une attaque est suffisament puissante pour effacer une marque, le taux d'erreur binaire détecté est de 0.5 ce qui correspond à une séquence binaire aléatoire mais lorsque les distorsions engendrées par cette attaque sont moins conséquentes, le taux d'erreur binaire tend vers 0. Pourtant, il y a des situations ou le taux d'erreur binaire est proche de 1 ce qui veut dire que la quasi-totalité des bits ont été inversés. Dans ce cas, l'information détectée n'est pas détruite (car elle est 'inversée') dans le sens où l'entropie de Shannon ne varie pas. Nous verrons dans le troisième chapitre qu'il s'agit d'une structure d'erreur qu'il est possible de corriger à l'aide d'un nouveau type de codes correcteurs.

1.4.6 Sécurité

La sécurité d'un schéma de tatouage est un sujet aussi important que les trois propriétés précédemment discutées. Elle s'inspire de la sécurité en cryptographie pour protéger les contenus multimédias. Pourtant, il existe des différences importantes entre ces deux domaines de

recherche [46]. Pouvoir déterminer un niveau de sécurité requiert une étude spécifique de l'application à laquelle un schéma de tatouage se destine. Ainsi, si nous avons un même schéma pour deux applications distinctes, les caractéristiques de sécurité peuvent être différentes.

Dans le contexte du tatouage robuste, une méthode de tatouage a pour objectif d'insérer un signal invisible dans une image hôte tout en conservant une qualité d'image acceptable. Ici, l'analyse de sécurité consiste à évaluer la difficulté d'un attaquant (destinataire illégitime) d'accéder voire de modifier le canal du tatouage [47]. Une méthode de tatouage sécurisée produit des images marquées dont l'accès au canal de la marque est inaccessible sans une clé, qui est uniquement connue de l'émetteur et du destinataire légitime.

Dans ce manuscrit, l'analyse de sécurité ne fait pas partie de nos thématiques de recherche mais ne manque pas d'intérêt pour de futures perspectives de travaux. De nombreux travaux peuvent être consultés dans la littérature tels que [48, 7, 49, 50, 51, 52, 53, 54].

Nous proposons d'aborder dans la section suivante une de nos approches principales qui est l'intégration des codes correcteurs dans des stratégies de tatouage numérique.

1.5 État de l'art de l'utilisation des codes correcteurs pour le tatouage

Nous proposons dans cette section un état de l'art non exhaustif des codes correcteurs appliqué au tatouage numérique. Comme présenté précédemment, le message utile à insérer dans une marque est encodé par une matrice génératrice d'un code correcteur puis le mot de code obtenu est utilisé en tant que *payload* à l'étape d'insertion. À l'étape de détection, une estimation du mot de code est calculée puis décodée par le code correcteur. L'intégration des codes correcteurs pour le tatouage numérique se fait donc de manière ad-hoc.

Les premiers travaux sont apparus au début des années 2000. Kesal et al. [55] ont proposé les codes binaires produit de Reed Muller avec un décodage itératif [56, 57] pour faire face au canal gaussien. Darbon et al. [58] ont étudié des combinaisons de codes BCH et de codes à répétition afin d'optimiser les performances de robustesse contre des attaques telles que la compression JPEG et le filtre médian. Ils ont également déterminé les paramètres de chaque code afin que la concaténation de codes permette d'obtenir une capacité optimale. Pour plus de clarté, nous reprenons le schéma de concaténation des codes dans la figure 1.6. Nous avons

alors $\mathcal{C}_1$ un code BCH et $\mathcal{C}_2$ un code à répétition. La fonction d'encodage Enc_1 peut représenter l'utilisation d'un code à répétition ou d'un code BCH et la fonction Enc_2 représente un code à répétition.

$$m = (m_1, \ldots, m_k) \longmapsto \underbrace{c = mG_1 = (c_1, \ldots, c_{n_1})}_{\substack{\text{Encodage de } m \text{ par } \mathcal{C}_1 \\ n_1 - k \text{ bits de redondance}}} \longmapsto \underbrace{d = cG_2 = (d_1, \ldots, d_{n_2})}_{\substack{\text{Encodage de } c \text{ par } \mathcal{C}_2 \\ n_2 - n_1 \text{ bits de redondance}}} \quad (1.10)$$

FIGURE 1.6 – Schéma de concaténation (ou codage hybride) de deux codes correcteurs $\mathcal{C}_1(n_1, k)$ (inner coding) et $\mathcal{C}_2(n_2, n_1)$ (outer coding) de matrices génératrices respectives G_1 et G_2. Le mot m est un message utile de k bits. Le mot de code final d est inséré dans l'image en tant que marque.

La notion de forme de l'erreur n'est pas présente. En fait, face à ces attaques, la structure de l'erreur est aléatoire. De même pour les travaux de Baudry et al. [1], une étude similaire de ces codes a été proposé pour le canal gaussien. Même si Zinger et al. [2] ont proposé des travaux similaires (encodage *hybride*), ceux-ci s'intéressent à différentes variantes sur les codes BCH telles que les codes BCH *par soustraction*, *étendu* et *par morceau*.

L'objectif de ces variantes est gagner en flexibilité (longueur limitée à $n = 2^m - 1$) sur les paramètres des codes BCH afin d'optimiser la capacité. Utiliser les codes BCH par morceau consiste à utiliser plusieurs mots de code de longueur plus petite pour encoder le plus de bits possible par rapport à une taille de payload autorisée tout en améliorant également le nombre d'erreur maximal d'erreur à corriger. L'idée est la même avec les codes BCH par soustraction et les codes BCH étendus : les mots de code peuvent être racourcis ou allongés de quelques bits tout en conservant le même polynôme générateur (voir [59]).

Par la suite, de nombreux travaux sur les codes s'inpirant de ces travaux ont été publiés sur différents types de support de tatouage. Par exemple, Hsieh et al. et Chan et al. [60, 61] ont proposé des améliorations de la robustesse d'un tatouage image face à la compression JPEG, au filtre passe-bas, à certaines attaques géométriques et même face à l'attaque Stirmark.

En ce qui concerne la vidéo, Chan et al. [62] ont mis au point une méthode basée sur la décomposition en coefficients d'ondelette qui insère les bits de redondance d'un mot de code dans le canal audio et ont obtenu une meilleure robustesse face à diverses attaques vidéo telle que la réduction de qualité vidéo (*frame dropping*). C'est une idée qui n'est pas toujours possible d'utiliser lorsque l'on veut insérer une marque dans d'autres types de supports que la vidéo mais reste cependant très astucieuse. Des travaux existent aussi pour le traitement du son : Liu et al. [63] ont intégré des codes BCH afin d'améliorer la robustesse de leur marque face à des

attaques telles que les modifications d'amplitude, la compression mp3, etc.

D'autres exemples de travaux ont également proposé d'intégrer les codes de Reed-Solomon pour corriger les erreurs par paquet (Wadood et al. [3]). Les travaux développés implémentent aussi le décodage en liste pour les codes de Reed-Solomon et proposent de nombreux résultats d'expériences (compression, filtre, modifications de composantes couleur dans l'espace HSV, etc). Par ailleurs, les codes correcteurs ont permis d'améliorer les méthodes de tatouage pour authentifier des images [64, 65]. De même pour le traçage de données avec [66]. Des travaux plus récents que 2013 proposent de nouvelles méthodes intégrant les codes correcteurs mais leur utilisation demeure conceptuellement la même (encodage avant insertion et décodage après détection de la marque).

Pour finir, un grand nombre de publications ont appliqué des codes *convolutifs* (par exemple [67, 68, 69]) et des *Turbo codes* (par exemple [70, 71, 72, 73, 74, 75]) pour le tatouage. Dans le cas où le signal marqué est transmis sur un canal bruité (canal gaussien, compression JPEG, filtres passe-bas, etc), les Turbo codes permettent d'obtenir les meilleures peformances de détection tout en se rapprochant de la capacité (deuxième théorème de Shannon). Afin de résumer cette section, nous donnons le tableau 1.1 un récapitulif les travaux de la littérature.

Par rapport à la littérature présentée ici, nous explorons un point de vue différent sur l'application des codes correcteurs pour le tatouage. Face à certaines modification d'images, l'erreur produite peut avoir une structure d'erreur particulière. En tenant compte de cette dernière information, un choix adapté de codes correcteurs permettra d'optimiser les performances de détection. Par exemple, en considérant les codes BCH et les codes de Reed-Solomon, les codes BCH sont plus efficaces que les codes de Reed-Solomon lorsque l'erreur est aléatoire. Lorsque l'erreur a une structure par paquet, ce sont les codes de Reed-Solomon qui deviennent meilleurs.

Auteurs	Codes	Contributions
Kesal et al. [55]	Turbo, Reed-Muller, Hamming, décodage itératif	Quantification vectorielle, comparaisons codes, canal gaussien
Hernandez et al. [68]	Convolutif, Viterbi, BCH	Variante étalement de spectre, comparaison codes, image/vidéo, canal gaussien, compression JPEG

Eggers et al. [76]	Turbo, BCH	*random cookbook* (Costa) + Turbo, comparaison blind/non-blind, image, filtre et bruit gaussien
Darbon et al. [58]	Hybride[2]	*Substitution watermarking* [77]+ quantification, comparaisons de compromis, image, compression JPEG, bruit gaussien
Baudry et al. [1]	Hybride, convolutif, Viterbi	*Substitution watermarking* [77], analyse de redondance, image, compression JPEG, bruit gaussien
Zinger et al. [2]	Hybride, variantes BCH[3]	Étude de performances théorique, image, canal binaire symétrique
Hsieh et al. [60]	Hybride	Comparaisons de compromis codes, image, compression JPEG, filtres, attaques géométriques
Baldo et al. [71]	Turbo, décodage itératif	Variante étalement de spectre, comparaisons de codes image, bruit gaussien
Lancini et al. [69]	Convolutif, Turbo	Comparaisons de codes, image/vidéo, compression MPEG, attaques géométriques
Chan et al. [62, 61]	RS, Turbo	Insertion domaine audio, vidéo, attaques vidéos : suppression de trames, découpage
Rey et al. [72]	Turbo par bloc, BCH produit, répétition	Méthode Eurécom [78], comparaisons de compromis code, image, attaques : bruit gaussien, compression JPEG
Cvejic et al. [79]	Turbo	Étalement de spectre, améliorations par les codes, audio, attaques : filtres, compression MP3
Zhang et al. [64]	Convolutif	Authentification, reconstruction, améliorations par les codes image couleur, détection et localisation de modifications

Doerr et al. [75]	Turbo	Méthode Eurécom, resynchronisation, approche EGM [80], attaques géométriques locales et globales
Dugelay et al. [73]	Turbo	Méthode Eurécom, Optical Flow Regulation, image, attaques géométriques, Stirmark
Liu et al. [63]	BCH	Insertion domaine *Ceptstrum*, audio, attaques asynchrones
Chan et al. [65]	Hamming	Preuve de modifications malicieuses image avec les codes, *Torus automorphism*, détection et reconstruction des zones modifiées
Chemak et al. [74]	Turbo	Étalement de spectre, insertion champ multirésolution, image médicale, attaques : compression, bruits, filtres, etc
Schaathun et al. [66]	RS, décodage en liste	*Fingerprinting*, améliorations par les codes, attaques : bruit gaussien, *cut-and-paste*, etc
Wadood et al. [3]	Hybride, RS, décodage en liste	Quantification vectorielle+ondelettes, améliorations par les codes, image couleur, attaques : bruit gaussien, compression, filtres, HSV, etc
Al Maeeni et al. [81]	Turbo, BCH produit	Insertion de logo, concaténation de codes, image, attaques : compression, *cropping*, filtre médian
...[4]		

Table 1.1 – Tableau récapitulatif des principales contributions sur le tatouage et les codes correcteurs de 2000 à 2015.

2. Combinaison des codes à répétition et des codes BCH.
3. Codes BCH étendu, par soustraction et par morceau.
4. D'autres travaux plus récents existent après 2013.

1.6 Approche codes correcteurs

Les codes correcteurs sont des outils puissants de la théorie de l'information. Ils sont utilisés pour résoudre des problèmes de transmission de signaux sur des canaux bruités (non fiables). Le tatouage numérique robuste peut être vu comme un problème de transmission. Par exemple, le propriétaire d'une image peut garantir ses droits d'auteur en ajoutant une marque de manière invisible (ne dégradant pas la qualité de son œuvre) idéalement non effaçable tant que l'image conserve une qualité suffisante.

Dans ce manuscrit, nous proposons d'intégrer les codes correcteurs dans des stratégies de tatouage afin d'améliorer la robustesse à certaines attaques. Selon l'attaque rencontrée, les erreurs produites peuvent avoir une structure particulière et donc un code correcteur peut être plus efficace qu'un autre. En tant que première thématique, nous proposons de rechercher et d'étudier la forme de l'erreur. La suite de cette section est dédiée à un bref aperçu des codes BCH [82] et de Reed-Solomon (RS) [83] qui sont des codes de Hamming [84].

1.6.1 Généralités

Lorsque l'on transmet des données sur un canal, des erreurs de transmission peuvent se produire. Le rôle des codes correcteurs d'erreur est de corriger ces erreurs de transmission. Pour protéger cette information, un code correcteur ajoute de la redondance dans un message de façon à ce que les erreurs puissent être détectées et corrigées après transmission. Nous donnons dans la figure 1.7 un diagramme illustrant les différentes étapes de transformation et de transmission d'un message.

Une fonction d'encodage est une application injective ϕ définie telle que :

$$\phi\colon \{0,1\}^k \to \{0,1\}^n$$

avec k la dimension du code et n la longueur du code. On dit que ϕ est un code de paramètre (k,n). Ainsi, un mot (de source) $m \in \{0,1\}^k$ a pour image le mot de code $c = \phi(m) \in \{0,1\}^n$. La fonction d'encodage ϕ est aussi désignée par la notation Enc (figure 1.7). Un code correcteur est un ensemble $C = Im(\phi)$.

Lorsqu'un mot de code c' est transmis sur un canal, celui-ci peut contenir des erreurs. Pour

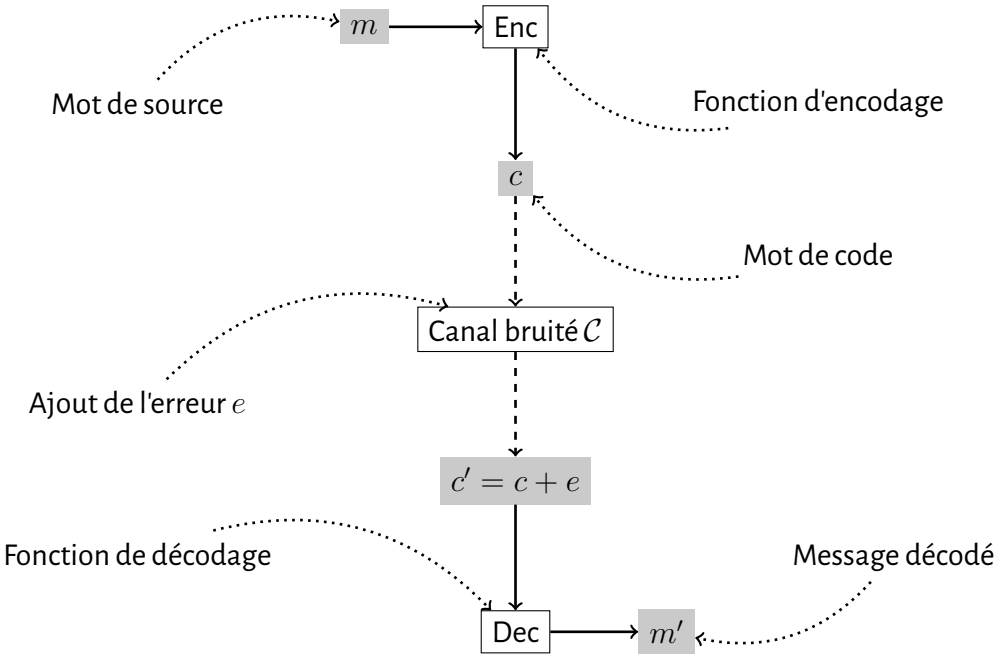


FIGURE 1.7 – Diagramme des différentes étapes permettant la transmission de manière fiable d'un message sur un canal.

corriger celles-ci, un algorithme de base consiste à comparer le mot reçu aux mots de code c'est-à-dire des éléments de C tel que :

$$C = Im(\phi) = \{\phi(m) \mid m \in \{0,1\}^k\}$$

pour trouver $c \in C$ qui est le plus proche de c' au sens de la distance de Hamming. On décode alors c' en c.

La *distance de Hamming* est le nombre de lettres distinctes entre deux mots c_1 et c_2. Elle est définie telle que :

$$d_H(c_1, c_2) = \#\{i \mid (c_1)_i \neq (c_2)_i\}$$

On définit aussi le *poids de Hamming* comme le nombre de lettres non nulles d'un mot c. Il est défini tel que :

$$w_H(c) = \#\{i \mid c_i \neq 0\}$$

Notons que cet algorithme de base prend beaucoup de temps avant de se terminer car il y a un nombre exponentiel de comparaisons à faire (complexité en $\mathcal{O}(n2^k)$). Si k est petit, le

temps de décodage est acceptable mais cela devient impraticable lorsque k est plus grand. Au décodage, nous supposons implicitement que lorsque le nombre d'erreurs est assez faible, le nombre de mots de code les plus proches de c' est égal à 1 (codes parfaits).

Une autre notion de base importante sur les codes correcteurs est la notion de *distance minimale* qui représente la plus petite distance entre deux mots de code d'un code C. Elle permet de caractériser la puissance de décodage. Un code de distance minimale d peut aussi se noter (n, k, d) et est capable de corriger t erreurs telles que :

$$t = \frac{d-1}{2}$$

Il existe des codes avec plus de structure, tels que les codes linéaires. Un code est dit *linéaire* s'il existe une matrice $G \in \mathcal{M}_{n,k}(GF(2))$ à k lignes et n colonnes à coefficients dans $\{0, 1\}$ de rang = k telle que :

$$\forall m \in \{0,1\}^k, \phi(m) = G \times m$$

La linéarité d'un code donne à l'image C une structure de sous-espace vectoriel de $\{0, 1\}$. La distance minimale d'un code linéaire d est donc égale au plus petit poids de Hamming non nul d'un mot de code de C. Pour un code linéaire donné, il peut être laborieux de déterminer sa distance minimale. Cependant, on a quand même une majoration de celle-ci appelée *borne de Singleton* :

$$d \leq n - k + 1$$

Lorsqu'il y a égalité, on dit que le code de paramètre (n, k, d) est *MDS (Maximum Distance Separable)*.

G est appelé la *matrice génératrice de* ϕ. Un code linéaire possède aussi la notion de *matrice de contrôle*. La *matrice de contrôle* $H \in \mathcal{M}_{n-k,n}(GF(2))$ du code C est définie telle que :

$$m \in C \iff Hm = 0$$

L'intérêt principal de la linéarité d'un code est de pouvoir construire des algorithmes de décodage plus rapides. Nous avons introduit l'essentiel des notions de base sur les codes correcteurs et proposons maintenant de donner un aperçu des codes BCH et RS.

1.6.2 Codes BCH (binaires) et codes de Reed-Solomon

Les codes BCH on été inventés par Hocquenghem, Bose et Ray-Chaudhuri en 1960 et constituent une généralisation des codes de Hamming. Ce sont des codes *cycliques*[1] très efficaces de longueur $n = 2^m - 1, m \geq 3$ dont la distance minimale $d \geq 2t + 1$ peut être choisie. Le nombre de bits de parité d'un code BCH binaire est majoré : $n-k \leq mt$. Ils sont optimaux face à des erreurs aléatoires, c'est-à-direque les erreurs sont positionnées de manière aléatoires sur les symboles d'un mot de code.

Les codes RS, quant à eux, sont un cas particulier des codes BCH cette fois-ci non binaires c'est-à-dire qu'ils sont définis sur un alphabet (ou corps fini) plus grand $GF(q^m)$, avec q premier. Ainsi, un symbole d'un mot est encodé sur plusieurs bits ce qui permet de mieux corriger les erreurs de transmission quand elle arrivent par paquet (*burst errors* en anglais). Les codes de RS sont MDS c'est-à-dire que leur distance minimale est égale $d = n - k + 1$ pour un code (n, k, d).

Selon la nature de l'erreur (aléatoire ou par paquet), un code BCH sera plus efficace qu'un code RS équivalent (en terme de longueur et de dimension) et inversement. Soit un code BCH de paramètres (n', k', t') de longueur n', de dimension k' corrigeant t' erreurs. Un code de RS défini sur $GF(2^8)$ de paramètres $(n, k, t = (n - k)/2)$ est dit *équivalent* au code BCH (n', k', t') si $n' = 8n$ et $k' = 8k$ (car un symbole est encodé sur huit bits). Nous avons alors $t = 4(n' - k')$. Et nous pouvons voir que si $n' > k' + 1$ alors :

$$8t = 4(n' - k') > n' - k' + 1 \geq t' \tag{1.11}$$

avec t le nombre de symboles erronés maximal autorisé par le code de RS. Si le code de RS était confronté à des erreurs par paquet, il pourrait alors couvrir dans le meilleur des cas $8t > t'$ erreurs binaires. Nous pouvons donc voir que dans le cas des erreurs par paquet, un code de RS sera bien plus efficace pour corriger les erreurs qu'un code BCH.

Nous donnons dans la table 1.2 quelques exemples de codes BCH arbitrairement choisis avec leur code de RS équivalent c'est-à-dire avec $n' = 8n$ et $k' = 8k$. Nous pouvons voir que pour chaque ligne, un code de RS pourra toujours corriger plus d'erreurs binaires lorsque les erreurs ont une structure par paquet. Par contre, dans le cas où les erreurs sont aléatoires, presque tous les symboles d'un mot de code de RS seront affectés par une erreur binaire et

1. Tout décalage circulaire d'un mot de code est un mot de code.

BCH (n', k', t')	RS_{2^8}	
(n', k', t')	(n, k, t)	$8t$
$255, 87, \mathbf{26}$	$32, 11, 10$	$\mathbf{80}$
$255, 131, \mathbf{18}$	$32, 16, 8$	$\mathbf{64}$
$255, 171, \mathbf{11}$	$32, 21, 5$	$\mathbf{40}$
$511, 175, \mathbf{46}$	$64, 22, 21$	$\mathbf{168}$
$511, 250, \mathbf{31}$	$64, 31, 16$	$\mathbf{128}$
$511, 340, \mathbf{20}$	$64, 43, 10$	$\mathbf{80}$

TABLE 1.2 – Table de paramètres de code BCH et de code de RS équivalents.

donc il sera inefficace d'utiliser un code de RS contre ce type d'erreur. L'alternative optimale est l'utilisation d'un code BCH.

L'intégration des codes correcteurs dans une stratégie de tatouage numérique que nous proposons consiste à insérer un mot de code dans une image. Face à une attaque donnée, nous pouvons choisir le code correcteur apportant les meilleures performances de correction en observant la structure des erreurs liées à cette attaque. Par exemple, nous proposons de combiner les codes en métrique rang à la méthode LQIM afin d'obtenir une méthode robuste à une structure d'erreur particulière liée aux changements de luminance et au découpage d'images dans le chapitre 3.

1.7 Conclusion

Dans ce chapitre, nous avons donné une introduction au tatouage numérique. Différents paradigmes de tatouage ont été présentés tels que le tatouage fragile, semi-fragile et réversible. Ils permettent de résoudre des problèmes d'intégrité et d'authentification d'image en insérant un motif de manière imperceptible. Tandis que le tatouage fragile permet de détecter des modifications d'image, le tatouage semi-fragile a pour objectif de détecter des modifications d'image malicieuses. Pourtant, ces deux paradigmes ne peuvent pas satisfaire certaines applications car même une faible dégradation de l'image est inacceptable. Pour cela, le concept de marque effaçable a été proposé et est connu sous le nom de tatouage réversible.

Puis, nous sous sommes concentrés sur le paradigme de tatouage robuste qui a pour objectif d'insérer une marque invisible et robuste aux modifications d'image. Nous avons présenté la méthode de quantification Lattice QIM que nous utiliserons dans nos stratégies de tatouage.

Nous avons proposé ensuite une introduction aux codes correcteurs BCH et de Reed-Solomon dans le but des les intégrer au tatouage robuste et ainsi améliorer la robustesse face aux attaques dont les erreurs ont une structure particulière. Dans le chapitre suivant, nous introduisons nos deux premières approches contribuant à améliorer la robustesse d'une marque en étudiant la méthode LQIM.

CHAPITRE 2

Tatouage par quantification et normes

Contenu

2.1 Introduction

Dans ce chapitre, nous proposons une étude sur l'amélioration de la robustesse d'une méthode de tatouage, à savoir la méthode LQIM présentée au début de ce chapitre. Ces deux pistes d'étude représentent les premiers travaux abordés au cours de cette thèse qui posent les bases des contributions présentées dans le chapitre suivant.

Premièrement, nous introduisons les codes en norme infinie qui sont des codes correcteurs applicables à une stratégie de tatouage par quantification similaire à la méthode LQIM. Après avoir défini ceux-ci, nous montrons une adaptation de ces codes pour le tatouage. Avec cette reformulation de la méthode QIM par les codes en norme infinie, nous proposons d'étudier un type d'erreur appelé *erreurs homogènes*. Ensuite, en ajoutant une étape de décodage supplémentaire à l'aide de codes BCH, nous montrons comment étendre les capacités de décodage aux erreurs homogènes ou presque.

Utiliser des codes correcteurs pour améliorer la robustesse n'est pas un concept nouveau dans la littérature (par exemple [1, 2, 3]). Un des objectifs de ces travaux de thèse est de trouver comment intégrer de manière originale des codes correcteurs au tatouage numérique.

Dans un second temps, nous nous intéressons au détecteur de la méthode Lattice QIM (Section 2.4.2) et proposons une version modifiée de ce détecteur nous permettant d'étudier l'impact du choix de la norme sur les performances de robustesse face à une attaque. L'idée qu'une norme puisse avoir un impact sur le détection s'inspire d'une visualisation de l'espace de quantification en deux dimensions ainsi que de la position des erreurs produites par un canal bruité.

Utiliser différentes normes permet de créer un recouvrement de l'espace de quantification défini par les boules associées à une norme. Par exemple, la représentation dans le plan d'une boule pour la norme euclidienne est le cercle. Pour finir, nous proposons différentes expériences afin d'étudier le comportement et les performances de détection associées à différentes normes face à plusieurs attaques.

2.2 Quantification vectorielle

À l'origine de ce travail, nous cherchons à mettre en lien les codes en norme infinie et la méthode de modulation d'index. Dans cette section, nous montrons d'abord l'équivalence entre

la méthode LQIM et les codes en norme infinie puis proposons une caractérisation des erreurs homogènes ou presque et non-homogènes.

2.2.1 Lattice QIM (LQIM) : rappel

Afin de rendre la lecture de ce chapitre plus fluide, nous avons choisi de rappeler la définition de la méthode LQIM dans cette section [32, 38]. L'espace de quantification de cette méthode est un réseau euclidien ($\mathbb{Z}^L$) de dimension L où un échantillon hôte $x \in \mathbb{R}^L$ est transformé en un élément y de ce réseau. La méthode QIM scalaire est un cas particulier obtenu en prennant $L = 1$.

Pour mémoire, pour tatouer une image en pratique, des étapes supplémentaires sont nécessaires comme nous avons pu le voir dans le chapitre précédent. Un échantillon x est obtenu en sélectionnant par exemple des valeurs de pixel dans le domaine spatial de l'image. Pour insérer un bit d'information, L valeurs de pixel seront sélectionnées dans l'image. Par exemple, une stratégie de sélection aléatoire de ces valeurs peut être adoptée. Ces sites d'insertion sont secrets et accessibles uniquement de l'émetteur et du récepteur grâce à une clé secrète.

Pour insérer un bit d'information $m \in \{0, 1\}$, nous quantifions l'échantillon x en y appartenant à l'un des sous-ensembles suivants appelés *cosets* notés Λ_0 et Λ_1 définis tels que :

$$\begin{aligned} \Lambda_0 &= \Delta\mathbb{Z}^L - \frac{\Delta}{4} \\ \Lambda_1 &= \Delta\mathbb{Z}^L + \frac{\Delta}{4} \end{aligned} \tag{2.1}$$

grâce à la fonction de quantification suivante :

$$y = Q_m(x, \Delta) = \left\lfloor \frac{x}{\Delta} \right\rfloor \Delta + (-1)^{m+1} \frac{\Delta}{4} \tag{2.2}$$

À la réception d'une image marquée puis modifiée, le message peut être détecté en faisant une extraction des coefficients utilisés à l'insertion grâce à la clé secrète. Pour extraire le bit d'information m, on calcule le coset le plus proche du vecteur z associé pour déterminer

l'estimation $\hat{m}$ tel que :

$$\begin{aligned} \hat{m} &= \underset{m\in\{0,1\}}{\arg\min}\, dist(z, \Lambda_m), \\ dist(z, \Lambda) &= \min_{y\in\Lambda} \|z - y\|_2 \end{aligned} \tag{2.3}$$

avec $\|.\|_2$ la norme euclidienne.

En fonction de l'attaque étudiée, comme nous l'avons dit, l'erreur possède non seulement une structure différente mais aussi une nature différente.

À ce stade de l'étude, dans notre stratégie d'insertion, nous ne prennons pas en compte les attaques géométriques c'est-à-direqu'il n'y a pas de désynchronisation. Nous nous concentrons sur des attaques qui agissent uniquement sur les coefficients de l'image telles que la compression JPEG et le bruit gaussien.

Nous proposons maintenant de décrire les recherches explorant les codes en norme infinie et les possibilités qu'ils peuvent offrir pour la problématique du tatouage.

2.2.2 Codes en norme infinie

L'idée générale à la base de la création de ces codes est de pouvoir corriger des erreurs homogènes grâce à la norme infinie définie telle que :

$$\|x\|_\infty = \max_{1\leq i\leq n} |x_i| \tag{2.4}$$

avec $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$. Comme pour les nombres réels, il est possible d'adapter cette définition sur les entiers modulo $q \geq 2$. Pour $x \in (\mathbb{Z}/q\mathbb{Z})^n$, nous avons la définition formelle suivante :

$$||x||_\infty = \max_i(\min(|x_i|, |q - x_i|)), 0 \leq x_i < q. \tag{2.5}$$

Soient q et r deux entiers premiers entre eux tel que $0 < r < q$. Un message à encoder m est un entier tel que $0 \leq m < r$.

Nous avons la fonction d'encodage e qui encode un message m en un mot de code $c = e(m)$:

$$e\colon \mathcal{M} = \mathbb{Z}/r\mathbb{Z} \to \mathcal{C} \subset \mathbb{Z}/q\mathbb{Z}$$
$$m \mapsto mr^{-1} \bmod q.$$

et nous avons aussi la fonction de décodage d :

$$d\colon \mathbb{Z}/q\mathbb{Z} \to (\mathbb{Z}/q\mathbb{Z})/r\mathbb{Z}$$
$$y \mapsto yr \bmod q \bmod r.$$

Par exemple, posons $(q, r) = (257, 7)$ et $m = 5$. L'encodage et le décodage respectifs donnent $c = mr^{-1} \bmod q = 221$ et $m = cr \bmod q \bmod r = 5$.

Lorsqu'un mot de code c est transmis, il est possible que celui-ci subisse des erreurs. Soit (q, r) un couple de paramètres définissant un code en norme infinie et $u \in \mathcal{C}^n$ tel que $u = (u_1, ..., u_n)$ avec $u_i = e(m_i), m_i \in \mathcal{M}$.

Si u est modifié en $u' = u + h$ avec $h = (h_1, ..., h_n), h_i > 0$, il est possible de retrouver les m_i à partir de u' si et seulement si :

$$r||h||_\infty < q \tag{2.6}$$

De plus, il est aussi possible de gérer les erreurs négatives en ajoutant un décalage de $q/(2r)$ à chaque mot de code. Nous avons donc la condition de décodage sur les $h_i \in \mathbb{Z}$ qui est : $|h_i| < q/(2r)$.

En se basant sur l'exemple précédent, si $c' = 250$, nous pouvons décoder le bon message $c'r \bmod q \bmod r = 250 \times 7 \bmod 257 \bmod 7 = 5$, car $0 < c' - c = 29 < q/r$.

Dans la sous-section suivante, nous allons montrer qu'une application de ces codes pour le tatouage consiste en une reformulation de la méthode QIM scalaire.

2.2.3 Insertion et détection

Les codes en norme infinie peuvent être adaptés à la quantification. L'entier r^{-1} représente le pas de quantification Δ de la méthode QIM mais pour les codes en norme infinie.

Pour chaque mot de code d'un code $\mathcal{C}$ trié par ordre croissant en valeur absolue, nous associons un bit d'information 0 ou 1 alternativement. Pour insérer un bit d'information m dans un échantillon $x = (x_1, \ldots, x_n)$ avec $n > 1$, nous associons à chaque x_i le mot de code de $\mathcal{C}$ associé au bit m le plus proche.

Au décodage, nous recevons $z = (z_1, \ldots, z_n)$ puis appliquons la fonction de décodage d sur les z_i. Nous retrouvons alors le bit associé à chaque mot de code $d(z_i)$. Lorsqu'il n'y a pas d'erreur, nous décodons le même bit sur chaque coordonnée. Par contre, si un bit est différent d'un autre, alors nous ne pouvons pas, a priori, prendre une décision sur le résultat du décodage. Une manière de parvenir à une estimation de m est d'utiliser un vote à la majorité des bits (code à répétition).

L'adaptation des codes en norme infinie pour le tatouage décrite précédemment est une reformulation de la méthode QIM scalaire. Par exemple, le paramètre r^{-1} représente le pas de quantification Δ de la méthode QIM. Nous avons vu qu'une extension au cas vectoriel est possible en ajoutant un code à répétition. En fait, les codes en norme infinie nous permettent de voir la problématique d'étude de la forme de l'erreur sous un autre point de vue. Nous proposons dans la suite d'introduire un type d'erreur appelé *erreurs homogènes* et expliquons comment il peut être corrigé efficacement par les codes en norme infinie.

2.3 Erreurs homogènes

Nous proposons dans un premier temps de définir les erreurs homogènes et de montrer comment les codes en norme infinie permettent de corriger ces erreurs. Dans un second temps, nous montrons également comment les erreurs presque homogènes peuvent aussi être gérées en combinant les codes en norme infinie avec des codes BCH.

2.3.1 Définition

Soit $e = z - y$ l'erreur calculée à partir du vecteur z transmis et du vecteur marqué y. Nous définissons une *erreur homogène* si :

$$\exists\, b_1, b_2 \,|\, b_1 \leq \|e\|_\infty \leq b_2 \tag{2.7}$$

avec b_1 et b_2 des entiers proches. Visuellement, nous pouvons voir que les composantes e_i de e sont bornées dans un tube de petit diamètre relativement à l'ensemble des valeurs prises par les e_i.

Nous pouvons alors dire qu'une erreur homogène est corrigée par un code en norme infinie de paramètre (q, r) si et seulement si $b_2 \leq q/(2r)$.

La définition d'une erreur *presque homogène* est plus générale que celle d'une erreur homogène. L'idée consiste à dire qu'une erreur presque homogène est une erreur homogène sauf sur un petit nombre de composantes. Soit E_1 l'ensemble des indices des composantes qui ne sont pas dans le tube. Nous avons :

$$E_1 = \{i \,|\, b_1 \leq e_i \text{ou } e_i \geq b_2\} \tag{2.8}$$

avec le cardinal de E_1 noté $|E_1|$, petit.

Lorsqu'une erreur n'est ni homogène, ni presque homogène alors elle est dite *non homogène*. Si une erreur homogène est bornée par $q/(2r$, alors le code en norme infinie associé permet de la corriger. Cependant, cela n'est plus suffisant face à une erreur presque homogène. Dans la suite, nous proposons d'introduire les codes BCH afin de résoudre ce problème.

2.3.2 Application des codes BCH

L'ajout de codes correcteurs, typiquement les codes BCH (optimaux pour les erreurs arrivant aléatoirement), permettent de corriger des erreurs presque homogènes. Pour un vecteur erreur e donné, les coordonnées non bornées par $q/(2r)$ peuvent être visualisées comme des "piques d'erreur" et ceux-ci peuvent être corrigés par un code BCH.

Soit un code BCH de paramètre (n, k, t) corrigeant t erreurs. Un message $m = (m_1, \ldots, m_k)$

est encodé en un mot de code $c = (c_1, \ldots, c_n)$. En sélectionnant n valeurs de pixel d'une image en tant qu'échantillon hôte $x = (x_1, \ldots, x_n)$, nous pouvons insérer le bit c_i à l'aide des x_i pour tout i. Au décodage, nous avons une erreur $e = z - y$. En associant les codes en norme infinie aux codes BCH, nous pouvons corriger des erreurs homogènes ou presque homogènes si et seulement si :

$$|e_i| \leq \frac{q}{2r}$$

et

$$|E_1| \leq t$$

Cette dernière équation nous permet de montrer que la correction des erreur presque homogène dépend directement de la capacité de correction du code BCH utilisé.

Après un bref rappel de la méthode LQIM, nous avons introduit les codes en norme infinie puis avons montré une application de ces codes pour le tatouage par quantification. Nous avons ensuite montré comment les erreurs presque homogènes peuvent être corrigées en associant les codes en norme infinie et les codes BCH.

Dans la section suivante, nous proposons une nouvelle piste d'étude de la structure de l'erreur. En étudiant une variante du détecteur de la méthode Lattice QIM, l'impact de la norme utilisée à la détection face aux erreurs est étudié.

2.4 Détecteur Lattice QIM modifié

Dans le cas du détecteur LQIM classique, nous avons un pavage de l'espace de quantification. Ainsi, un vecteur traité à l'étape de détection est toujours garanti pour tout échantillon traité. À cette étape, changer la norme n'influence pas les performances de robustesse. En effet, les mesures de taux d'erreurs binaires moyens calculées à partir d'images marquées et modifiées pour chaque norme sont les mêmes quelque soit l'attaque considérée.

Inspiré par une visualisation en deux dimensions de l'espace de quantification, nous avons modifié le détecteur classique de façon à pouvoir observer un impact sur les performances de détection entre différentes normes, ce qui n'est pas possible d'obtenir avec le détecteur classique par construction (recherche du plus proche voisin). Ainsi, une certaine norme permettrait d'obtenir une détection plus efficace qu'une autre selon la structure de l'erreur. Par rapport

au détecteur classique, l'objectif est également de savoir si le détecteur modifié proposé avec une certaine norme permettrait d'obtenir de meilleurs résultats.

2.4.1 Concept

Nous reprennons les mêmes notations tout au long de cette section, à savoir x un échantillon hôte, y le résultat de la quantification de x et $z = y + e$ le vecteur reçu avec e une erreur. Inspiré par la méthode LQIM, nous visualisons de manière intuitive et informelle une étape de détection (Figure 2.1). L'idée est de dire qu'une boule de centre y issue d'une certaine norme contient des éléments z qui se décodent en y.

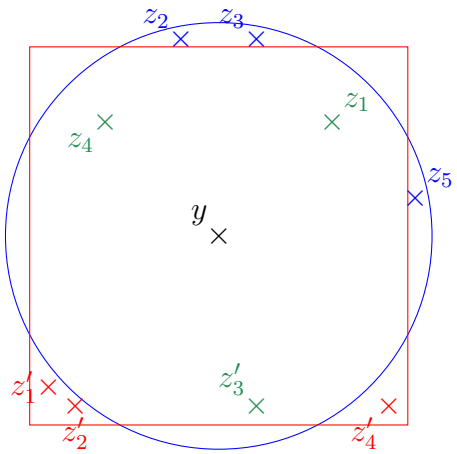


FIGURE 2.1 – Décodage des vecteurs z_i et z'_i pour la norme euclidienne ($\mathcal{B}_{\|.\|_2}$) et la norme infinie ($\mathcal{B}_{\|.\|_\infty}$).

Par exemple, lorsque la norme utilisée par le détecteur est $\|.\|_2$, nous avons la boule $\mathcal{B}_{\|.\|_2}(y, r)$ de centre y et de rayon r :

$$\mathcal{B}_{\|.\|_2}(y, r) = \mathcal{B}_2(y, r) = \{z \mid \|z - y\|_2 \leq r\} \tag{2.9}$$

Parmi les éléments z_i et z'_i, ceux qui peuvent être décodés par la norme euclidienne sont z_1, z_2, z_3, z_4 et z'_3 (représentés en vert et bleu) car ils appartiennent à $\mathcal{B}_2(y, r)$. Par contre, ce n'est pas le cas pour les éléments z'_1, z'_2 et z'_4 (représentés en rouge) car ils n'appartiennent pas à $\mathcal{B}_2(y, r)$.

Nous pouvons faire le même raisonnement avec la norme infinie $\|.\|_\infty$ avec la boule $\mathcal{B}_{\|.\|_\infty}(y, r)$:

$$\mathcal{B}_{\|.\|_\infty}(y, r) = \mathcal{B}_\infty(y, r) = \{z \mid \|z - y\|_\infty \leq r\} \tag{2.10}$$

Nous avons alors $z'_1, z'_2, z'_3, z'_4, z_1, z_4 \in \mathcal{B}_\infty(y, r)$ (éléments représentés en rouge et vert) et $z_2, z_3, z_5 \notin \mathcal{B}_\infty(y, r)$ (éléments représentés en bleu).

Face à une attaque dont les erreurs sont de même type que les z_i, la norme euclidienne est plus efficace pour corriger les erreurs. De même, la norme infinie sera plus efficace que la norme euclidienne lorsque les erreurs se comportent de la même manière que les z'_i.

Cependant, la description d'un tel détecteur ne correspond pas à celle du détecteur Lattice QIM classique. En effet, c'est la notion du plus proche voisin (fonction $\min$) qui est utilisée et permet de définir un pavage de l'espace de quantification. Dans la sous-section suivante, nous proposons une définition formelle de ce nouveau détecteur. Formaliser le détecteur que nous venons de décrire nous permettrait d'observer l'impact du choix de la norme face aux erreurs d'un certain type et même d'améliorer les performances de robustesse de la méthode.

2.4.2 Définition du détecteur modifié

Pour tout élément $y \in \Lambda = \Lambda_0 \cup \Lambda_1$, nous associons une boule $\mathcal{B}$ de centre y et de rayon $r > 0$ pour une norme donnée $\|.\|$ tel que :

$$\mathcal{B}_{\|.\|} = \mathcal{B}(y, r) = \{z \mid \|z - y\| \leq r\} \tag{2.11}$$

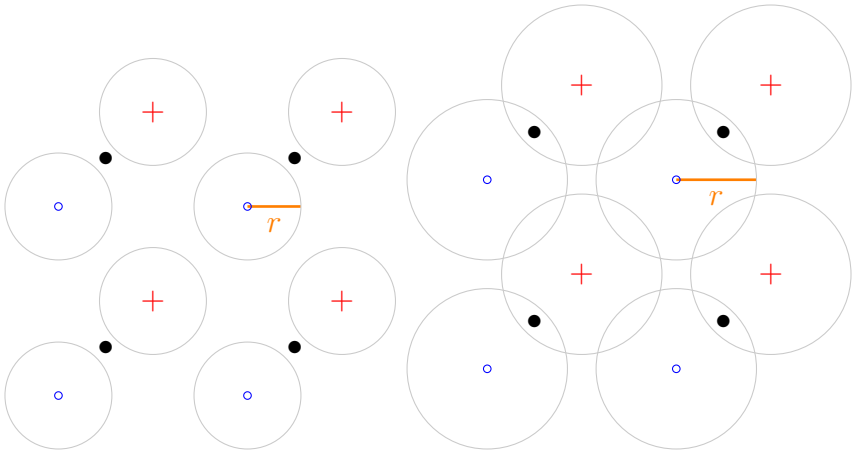
Soit z un vecteur issu d'une transmission. Nous définissons le détecteur proposé à l'aide des trois règles suivantes :

1. Si $z \in \mathcal{B}_y$ et $\forall y' \neq y, z \notin \mathcal{B}_{y'}$ alors z se décode en y.
2. Si $\exists y, y'$ tel que $z \in \mathcal{B}_y \cap \mathcal{B}_{y'}$, alors nous ne pouvons pas corriger l'erreur associée à z (sur-recouvrement).
3. Si $\forall y, z \notin \mathcal{B}_y$, alors nous ne pouvons pas corriger l'erreur associée à z non plus (sous-recouvrement).

Lorsque nous nous trouvons dans le premier cas, il est possible d'extraire le bit m associé à $y \in \Lambda_m$. Dans les autres cas, il n'est pas possible d'associer $y \in \Lambda$ à z.

Nous proposons maintenant de commenter des exemples de ce détecteur en introduisant la norme L_∞, la norme L_2 et la norme L_1. Comme le montrent les figures 2.2, 2.4 et 2.3, nous observons des boules de formes différentes selon la norme. Respectivement, les boules sont associées aux formes de cercle, carré et losange.

Par rapport à la méthode LQIM, nous n'avons pas de pavage de l'espace de quantification dans ces trois exemples. Dans certains cas, il y a effacement du symbôle détecté car il n'est pas possible d'associer une boule à z ce qui n'arrive pas avec le détecteur LQIM (recherche du plus proche voisin pour la norme euclidienne).



(a) Sous-recouvrement (b) Sur-recouvrement

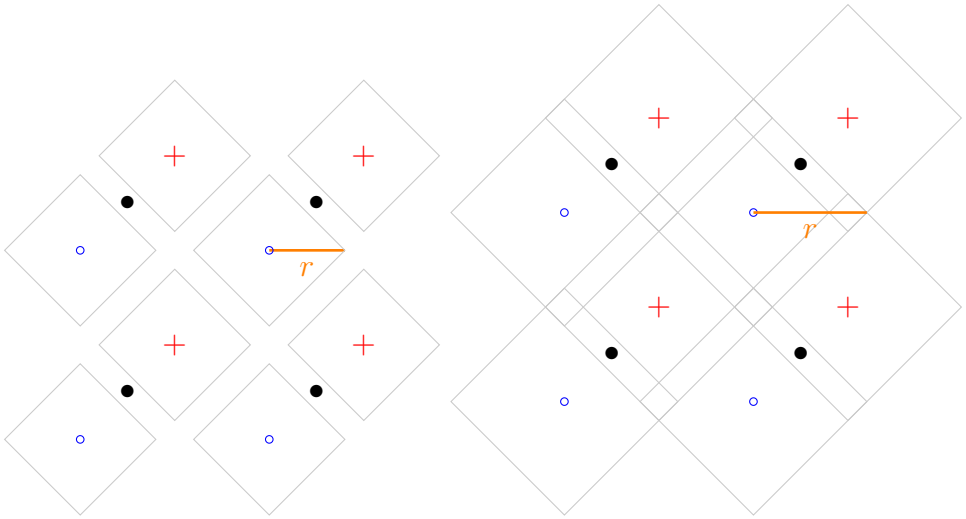
FIGURE 2.2 – Zones de détection pour la norme euclidienne dans un espace de quantification $L = 2$.

La figure 2.2 nous permet de visualiser en deux dimensions les zones de détection du détecteur proposé avec la norme euclidienne. À gauche (sous-figure 2.2a), les zones de détection sont constituées de cercles de même rayon r centrés en $y \in \Lambda$. Le rayon r est suffisament petit pour ne pas avoir d'intersection entre les cercles. Le détecteur proposé est donc uniquement soumis à la première et la dernière règle de détection (cas du sous-recouvrement de l'espace).

Sur la partie droite (sous-figure 2.2b), nous avons représenté le même espace mais avec un rayon r plus grand. Le deuxième cas de détection (sur-recouvrement) apparait car les cercles sont en intersection avec les cercles voisins. En observant les figures 2.3 et 2.4, nous pouvons faire les mêmes remarques.

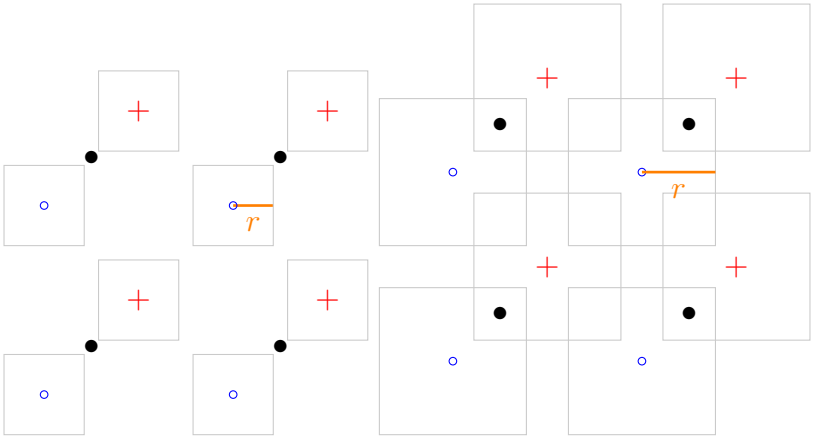
Nous pouvons donc dire que la valeur du rayon r a une influence sur les performances de détection pour un pas de quantification Δ fixé. Un rayon trop petit est visuellement susceptible d'affaiblir les performances à cause d'un sous-recouvrement. Il en est de même avec un rayon trop grand à cause du problème de sur-recouvrement.

Comme nous l'avons vu, les performances de détection varient selon le choix de la norme pour une structure d'erreur donnée. En obversant de plus près le recouvrement des boules en norme L_1 (figure 2.3), il est possible de choisir un rayon tel que le détecteur LQIM modifié ne



(a) Sous-recouvrement (b) Sur-recouvrement

FIGURE 2.3 – Zones de détection pour la norme L_1 dans un espace de quantification $L = 2$.



(a) Sous-recouvrement (b) Sur-recouvrement

FIGURE 2.4 – Zones de détection pour la norme infinie dans un espace de quantification $L = 2$.

soit plus confronté aux deux dernières règles de détection. Il n'y a ni sous-recouvrement, ni sur-recouvrement. Un élément de cet espace appartient forcément à une et une seule boule $\mathcal{B}(y, r)$. Nous remarquons alors qu'il s'agit d'un cas particulier qui n'est autre que le détecteur LQIM classique. Ce cas particulier est obtenu en choisissant $r = \Delta/2$ pour $\mathcal{B}_{\|.\|_1}(y, r)$ [1]. Visuellement, nous pouvons voir que lorsque r tend vers $\Delta/2$, la frontière d'une boule de la figure 2.3 se rapproche des frontières voisines sans créer de sur-recouvrement. C'est n'est qu'une fois que

1. La distance entre deux éléments de Λ_m est $r = \Delta/2$ d'après la construction de l'espace euclidien.

$r > \Delta/2$ que le sur-recouvrement apparait (partie droite de la figure).

Dans la section suivante, nous mesurons les performances de décodage avec notre décodeur modifié avec les trois normes face à plusieurs attaques.

2.4.3 Comparaisons des normes avec le détecteur modifié

L'objectif de ces expériences est de mettre en évidence l'impact de la norme utilisée dans le détecteur que nous proposons face à plusieurs attaques. Le second objectif est d'identifier les améliorations de robustesse possibles. Nous avons mesuré des taux d'erreur image moyens en faisant varier l'impact d'une attaque.

Afin de comparer le détecteur modifié avec différentes normes de manière équitable, nous avons choisi de paramétrer les rayons des boules en fonction de l'espace qu'elles occupent dans l'espace de quantification. Ainsi, l'aire occupée par les sous-recouvrements associés à chaque norme est la même.

L'expression de l'aire d'une boule de centre y et de centre r s'exprime différemment selon la norme. Nous avons :

$$\begin{cases} \mathcal{A}_2 = \mathcal{A}_{\|.\|_2} = \pi r_2^2 \\ \mathcal{A}_\infty = \mathcal{A}_{\|.\|_\infty} = (2r_\infty)^2 \\ \mathcal{A}_1 = \mathcal{A}_{\|.\|_1} = 2r_1^2 \end{cases} \tag{2.12}$$

En exprimant le rayon en fonction de l'aire associée, nous avons :

$$\begin{cases} r_2 = \sqrt{\dfrac{\mathcal{A}_2}{\pi}} \\ r_\infty = \dfrac{\sqrt{\mathcal{A}_\infty}}{2} \\ r_1 = \sqrt{\dfrac{\mathcal{A}_1}{2}} \end{cases} \tag{2.13}$$

Par exemple, en visualisant l'espace de quantification en deux dimensions, nous pouvons

choisir $\mathcal{A} = \Delta^2/8$ et nous avons :

$$\begin{cases} r_2 = \sqrt{\dfrac{\Delta}{4\pi}} \\ r_\infty = \dfrac{\Delta}{4\sqrt{2}} \\ r_1 = \dfrac{\Delta}{4} \end{cases} \tag{2.14}$$

Remarquons que si $\mathcal{A} = \Delta^2/2$, on a $r_1 = \Delta/2$ ce qui permet d'obtenir un pavage de l'espace de quantification (détecteur classique de la méthode LQIM).

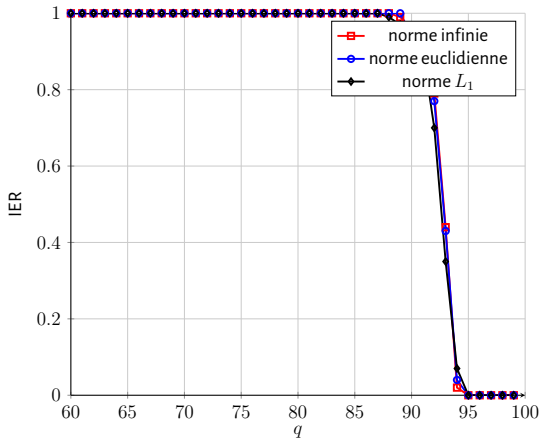


FIGURE 2.5 – Taux d'erreur du détecteur modifié avec les trois normes en fonction du facteur de qualité JPEG q.

Les figures 2.5, 2.6, 2.7 et 2.8 représentent les courbes des taux d'erreur moyens pour différentes images et pour différents paramétrages des attaques. Chaque figure contient une courbe de taux d'erreur par norme avec leur rayon de boule correspondant pour les attaques suivantes :

— compression JPEG

— modification de contraste

— modification de luminance

— bruit additif gaussien blanc

Chaque point d'une courbe représente la moyenne calculée sur une sélection aléatoire de 1000 images de la base Corel. Pour chaque image, nous insérons un message aléatoire de 128 bits. D'autres paramètres choisis arbitrairement sont $\Delta = 60$ et $L = 2$. Les résultats de taux d'erreur sont similaires pour $L = 3, 4, 5$.

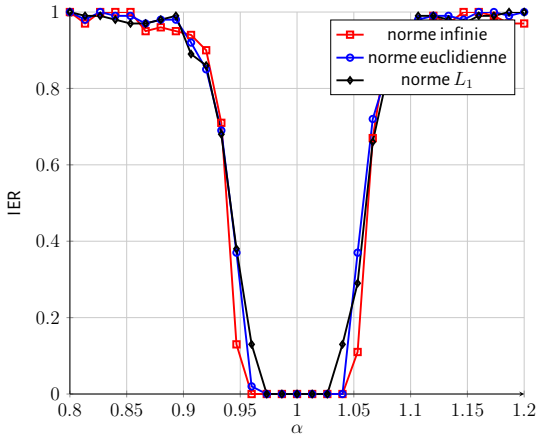


FIGURE 2.6 – Taux d'erreur du détecteur modifié des trois normes en fonction d'un gain α.

Pour une attaque donnée, nous observons que les trois courbes se superposent dans les figures 2.5, 2.6 et 2.8). Nous en déduisons que la norme utilisée n'a pas d'impact sur la robustesse du détecteur modifié face à la compression JPEG, la modification de contraste et le bruit gaussien. Par contre, nous pouvons observer que dans le cas de la modification de luminance (figure 2.7), les taux d'erreur image pour la norme infinie restent nuls sur un intervalle ($10 \leq \beta \leq 10$) plus grand que les intervalles des autres courbes.

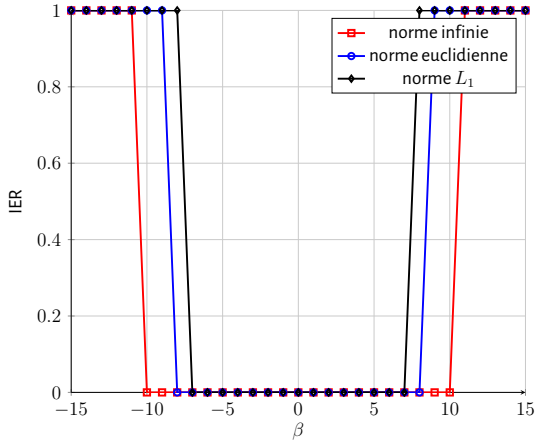


FIGURE 2.7 – Taux d'erreur du détecteur modifié des trois normes en fonction du paramètre de luminance β.

Même s'il semble y avoir une meilleure performance de détection contre la modification de luminance pour la norme infinie, l'écart de performance n'est pas suffisament significatif pour valider l'amélioration de robustesse en utilisant la norme infinie. Cette étude ne nous permet donc pas de mettre en évidence l'impact d'une norme sur les performances de détection face à une attaque donnée. Nous ne pouvons pas non plus en déduire une amélioration des perfor-

mances de détection par rapport à la méthode LQIM classique.

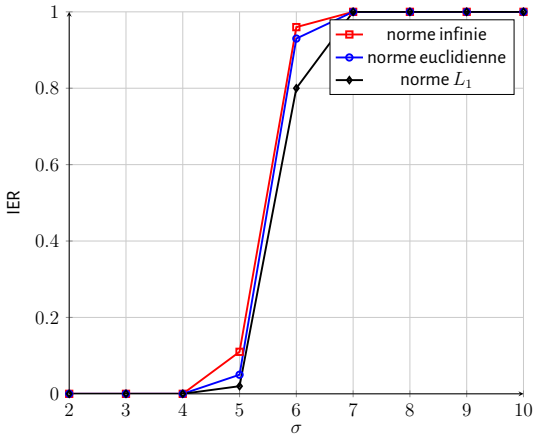


FIGURE 2.8 – Taux d'erreur du détecteur modifié des trois normes en fonction de la déviation standart σ d'un bruit additif gaussien blanc.

Nous avons comparé de manière équitable les performances de robustesse du détecteur modifié. En construisant un recouvrement de l'espace de quantification pour chaque norme, nous avons calculé les rayons correspondant à chaque norme afin que l'aire occupée associée soit la même. Puis, nous avons calculés des taux d'erreur moyens afin de comparer les résultats associés à chaque norme. Nous avons pu montré que le détecteur modifié est une version plus générale du détecteur LQIM classique.

En choisissant le rayon approprié, le détecteur modifié avec la norme L_1 est une définition équivalente du détecteur LQIM classique.

Cependant de ces expériences nous n'avons pas pu montrer qu'une des trois normes est significativement plus efficace qu'une autre face aux quatre attaques. Nous n'avons pas non plus pu en déduire une amélioration de la robustesse du détecteur modifié.

2.5 Conclusion

Nous avons rappelé la méthode LQIM et introduit les codes en norme infinie puis avons montré comment adapter ceux-ci dans une stratégie de tatouage par quantification. Nous avons ensuite utilisé cette reformulation d'un point de vue code correcteur pour étudier un certain type d'erreur. En associant les codes en norme infinie et les codes BCH, nous avons montré que

cette combinaison de codes correcteurs permet par construction d'être robuste aux erreurs homogènes ou presque.

En deuxième partie, nous avons étudié l'impact du changement de la norme lors de l'étape de détection de la méthode LQIM en proposant un détecteur modifié qui est en fait une version plus générale de celui-ci. En choisissant la norme L_1 et le rayon approprié, nous obtenons un cas particulier qui est le détecteur classique.

Ce détecteur modifié a été proposé dans le but de mettre en évidence l'impact du choix de la norme sur les performances de détection face à une attaque donnée. Cependant, nous n'avons pas obtenu de résultats montrant l'influence de la norme (norme infinie, norme L_1 et norme euclidienne) à l'étape de détection. Nous n'avons donc pas non plus d'amélioration des performandes de robustesse. Ceci nous incite donc à clôturer cette stratégie d'amélioration de la robustesse car numériquement l'apport n'est pas probant.

Toutefois, ce chapitre nous permet de développer le concept de structure de l'erreur pour améliorer la robustesse d'une marque pour la méthode LQIM, concept qui sera utilisé dans le chapitre suivant.

CHAPITRE 3

Tatouage numérique et codes correcteurs en métrique rang

Contenu

3.1 Introduction

Comme nous l'avons vu, un outil efficace pour augmenter la robustesse d'une marque est d'utiliser des codes correcteurs d'erreur. Ils vont permettre de corriger les erreurs produites par une attaque donnée. En fonction de l'attaque et de la structure des erreurs induite par celle-ci, le type de codes utilisé sera plus ou moins efficace. Par exemple, si l'erreur induite sur la marque est aléatoire, les meilleurs résultats sont obtenus en utilisant des codes binaires comme par exemple les codes BCH.

Les codes basés sur la distance de Hamming existent depuis plus de 40 ans et ont fait l'objet de nombreuses études [59] et d'applications dans de divers domaines tel que le traitement du signal et des images (par exemple Wadood et al. ont proposé une méthode de tatouage dans le domaine des ondelettes combinée avec différents codes correcteurs [3]).

Comme nous l'avons vu, il arrive que les erreurs surviennent par paquet. Dans ce cas, il est préférable d'utiliser des codes plus structurés c'est-à-dire des codes définis sur un alphabet plus grand (par exemple $GF(2^m)$), tels que les codes de Reed-Solomon qui sont capables de décoder les erreurs par paquet. Alors, les erreurs ne se décodent plus indépendemment sur chaque bit mais sur des paquets de m bits de façon à ce que plusieurs erreurs dans le même paquet binaire ne comptent que pour une seule erreur (une erreur de symbôle) dans un mot de code. Des explications plus détaillées sur le fonctionnement de ces codes peuvent être consultées dans le chapitre d'introduction de ce manuscrit.

Ainsi, en fonction de l'attaque i.e. des erreurs produites, nous pouvons choisir un code correcteur adapté. C'est un concept déjà bien connu qui a permis de nombreuses applications industrielles de ces codes utilisant la distance de Hamming. Par exemple, les codes de Reed-Solomon sont utilisés dans les technologies de stockage [85, 86] tel que le DVD [87, 88].

Dans ce chapitre, nous considérons l'utilisation d'une nouvelle métrique appelée la *métrique rang*. Ces codes sont déjà très utilisés en télécommunication dans le cadre du codage de réseaux [89] et en cryptographie [90, 91]. Ils sont capables de corriger des erreurs d'une structure spécifique. Si on considère un code sur $GF(2^m)$ de taille m, chaque coordonnée d'un mot de code sur $GF(2^m)$ est encodée sur m bits et puisque le code est de longueur m, tout mot de code peut être vu comme une matrice de taille $m \times m$.

Comme la métrique utilisée est le rang d'une matrice binaire, la condition de décodage s'exprime avec cette métrique c'est-à-direque seuls les mots de code reçus avec des erreurs et

ayant un rang faible peuvent être corrigés correctement. Par exemple, une attaque qui inverse tous les bits d'une marque (c'est-à-dire d'un mot de code) ne pourra pas être corrigée par un code de Hamming. Par contre, avec des codes en métrique rang, l'erreur sera de rang 1 car l'erreur est une matrice remplie de symbôles binaires 1 et ainsi, on retrouve sans difficulté le mot de code qui a été transmis.

Contribution : Nous proposons d'introduire ce type de correction dans un schéma de tatouage. Les codes en métrique rang sont tout d'abord définis dans la section 3.2. Ensuite, nous proposons une méthode de tatouage combinant la méthode Lattice QIM et les codes en métrique rang. Cette méthode est capable de gérer des erreurs structurées produites par des attaques particulières.

Dans un premier temps, nous montrons que la méthode proposée est théoriquement insensible aux modifications de luminosité en améliorant le détecteur LQIM avec une stratégie de multi-détection sur des versions volontairement modifiées de l'image reçue. Puis, grâce à une décomposition par bloc de l'image, nous montrons comment la méthode proposée peut également être résistante au découpage d'image. Dans le cas de ces deux attaques, les codes en métrique de Hamming sont loins d'être aussi efficace que les codes en métrique rang.

3.2 Codes correcteurs en métrique rang

Dans un premier temps, nous donnons quelques généralités sur les codes en métrique rang puis montrons des exemples en pratique d'utilisation de ces codes pour le tatouage.

3.2.1 Définitions et propriétés

Considérons un code linéaire $\mathcal{C}$ de longueur n défini sur un alphabet $\mathcal{A} = GF(q^m)$. Les mots de code de $\mathcal{C}$ sont des vecteurs lignes de l'espace vectoriel $GF(q^m)^n$. Chaque composante d'un mot de code de $\mathcal{C}$ peut être exprimé comme un vecteur appartenant à $GF(q)^m$. Il est possible d'écrire ces composantes sous forme d'un vecteur colonne et donc de représenter un mot de code par une matrice définie sur $GF(q)_{m\times n}$.

En considérant une base $\mathcal{B}$ de $GF(q^m)$ sur $GF(q)$ et un mot de code $x = (x_1, ..., x_n) \in$

$GF(q^m)^n$, nous obtenons une représentation matricielle de x notée $Mat(x) = (x_{ij})_{i,j}$ ou X lorsqu'il n'y a pas d'ambiguité définie :

$$X = \begin{pmatrix} x_{11} & \dots & x_{1n} \\ \vdots & & \vdots \\ x_{m1} & \dots & x_{mn} \end{pmatrix}$$

tel que pour tout $1 \leq j \leq n$:

$$x_j = \sum_{i=1}^{m} x_{ij}\beta_i$$

C'est grâce à cette représentation matricielle qu'il est possible de définir une nouvelle métrique sur $GF(q^m)^n$ en utilisant le rang d'une matrice.

3.2.2 Distance rang

Soit $x = (x_1, \dots, x_n) \in GF(q^m)^n$. Le *rang de* x est noté $w_R(x)$ et est égal à :

$$w_R(x) = rk(X) \tag{3.1}$$

Soit $y = (y_1, \dots, y_n) \in GF(q^m)^n$. La *distance rang* de x à y, notée $d_R(x, y)$ est égale à :

$$d_R(x) = rk(X - Y) \tag{3.2}$$

d_R possède les propriétés d'une distance mathématique. Comparé à la distance de Hamming, nous avons la propriété suivante :

$$w_R(x) \leq w_h(x) \tag{3.3}$$

ave x un mot de code et w_h la fonction poids de Hamming. Nous pouvons aussi en déduire

que :

$$d_R(x) \leq d_h(x) \tag{3.4}$$

Cette propriété est vraie car le nombre de lignes ou colonnes linéairement indépendante (c'est-à-dire le rang) d'un mot de code est toujours plus petit que le nombre de symboles différents deux-à-deux (poids de Hamming). En d'autres termes, puisque le rang d'un vecteur est indépendant de la base utilisée, la métrique rang est moins précise que la métrique de Hamming car deux vecteurs ayant le même poids peuvent avoir le même rang.

3.2.3 Codes en métrique rang

Delsarte [92] a été le premier à avoir étudier la métrique rang en 1978. De nombreuses propriétés des codes de Hamming ont été adaptées aux codes en métrique rang. Un code linéaire $\mathcal{C}$ défini sur un corps fini $GF(q^m)$ peut être vu comme un sous-espace de $GF(q^m)^n$ mais aussi comme un espace métrique muni de la distance rang.

De plus, comme $\mathcal{C}$ est un code linéaire, $\mathcal{C}$ est un sous-espace de $GF(q^m)^n$. La linéarité d'un code est une propriété intéressante car elle permet de manipuler les mots de code plus facilement. Comme pour les codes de Hamming, on peut définir la distance minimale d_{min} d'un code en métrique rang par :

$$d_{min} = \min_{x \neq y \in \mathcal{C}} d_R(x, y) \tag{3.5}$$

Avec la propriété de linéarité du code, on a donc aussi :

$$d_{min} = \min_{x \in \mathcal{C}^*} w_R(x) \tag{3.6}$$

Nous noterons un code en métrique rang linéaire $\mathcal{C}$ de longueur n, de dimension k et de distance minimale d par ses paramètres $[n, k, d]_r$ ou $[n, k, (d-1)/2]_r$ ou encore $[n, k]_r$ s'il n'est pas nécessaire de préciser la distance minimale.

3.2.4 Décodage des codes de Gabidulin

Les bornes de décodage des codes en métrique rang (bornes de Singleton et de Gilber-Varshanov) sont similaires à celles des codes de Hamming. Elles sont très utiles pour construire

des algorithmes de décodage. Contrairement au codes classiques en métrique de Hamming, il n'y a qu'un petit nombre de familles de codes pour lesquels on connaît un algorithme de décodage.

Les codes de Gabidulin [93] constituent une de ces familles et a pour paramètres $[n, k, n-k+1]_r$ sur $GF(q^n)$ avec $d = n-k+1$ la distance minimale. Ces codes sont dits MRD (Maximum Rank Distance) et peuvent donc corriger jusqu'à $(n-k)/2$ erreurs. Il peuvent être vus comme une famille analogue en métrique rang des célèbres codes de Reed-Solomon (qui sont MDS). Depuis 1985, de nombreux algorithmes de décodage ont été proposés dans la littérature tels que [94, 95]. En fait, l'algorithme de décodage des codes de Reed-Solomon peut être adapté pour les codes de Gabidulin. En annexe de ce manuscrit, nous donnons plus de détails sur les codes de Gabidulin accompagnés d'exemples numériques.

3.2.5 Introduction de la métrique rang dans une stratégie de tatouage

En pratique, nous utilisons ces codes dans une extension $GF(q^m)$ de $GF(2)$ et nous pouvons associer un vecteur binaire de taille m à chaque coordonnée d'un mot de code de telle sorte qu'un mot de code c peut être représenté par une matrice binaire de taille $m \times m$.

Le mot de code inséré c modifié par une erreur e peut aussi être perçu comme une matrice binaire de taille $m \times m$.

Pour évaluer dans quel cas la métrique rang est plus efficace que la métrique de Hamming classique, nous comparons les comportements de ces dernières en observant quelques exemples d'erreurs. Soient $y = c + e$ un mot de code reçu après extraction d'une marque.

Exemple : $m = 4$ et un mot de code c :

$$c = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

Avec y tel que :

$$y = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

On a l'erreur suivante :

$$e = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

On voit que la matrice erreur e est de rang 2. Si un code en métrique rang est capable de corriger jusqu'à deux erreurs alors on décode de manière unique y en c. Avec un code de Hamming de longueur 16, on aurait une erreur de poids égal à 4. Dans ce cas là, il est possible de trouver des codes performants avec les deux métriques pour une dimension raisonnable k.

Indépendemment de y et c, considérons une erreur maintenant :

$$e = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

Ici, le poids de Hamming de e est de 9 alors que le rang de e est de 4. Il est alors impossible de trouver des codes performants avec les deux métriques. Si e était la matrice identité, on aurait une erreur de rang plein et seul les codes de Hamming auraient été utiles.

De cet exemple on constate que les codes en métrique rang deviennent plus intéressants lorsque l'erreur possède une structure particulière. Par exemple, pour l'erreur :

$$e = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

avec la métrique de Hamming on a 9 erreurs sur 16 bits transmis et il n'existe pas de code capable de décoder correctement, alors qu'avec la métrique rang, e est de rang 1 seulement

et nous pouvons donc facilement la corriger (par exemple, avec un code de Gabidulin de paramètres $[4, 2, 3]$).

De la même façon que certaines attaques vont être “liées” aux codes RS ou BCH, ce type d'erreur se retrouve dans le cadre pour des attaques particulières. Les codes en métrique rang seraient, dans ce cas, une alternative dans le contexte du tatouage plus performante que les codes classiques de Hamming.

Nous allons maintenant proposer une méthode de tatouage intégrant les codes en métrique rang afin de se protéger contre certains types d'attaques.

3.3 Lattice QIM (LQIM) et métrique rang

Nous choisissons de poursuivre notre travail avec la méthode LQIM [32, 38] déjà présentée au chapitre précédent. Pour plus de détails sur la mise en place et la manipulation d'un code en métrique rang, le lecteur pourra se référer à l'annexe située en fin de manuscrit.

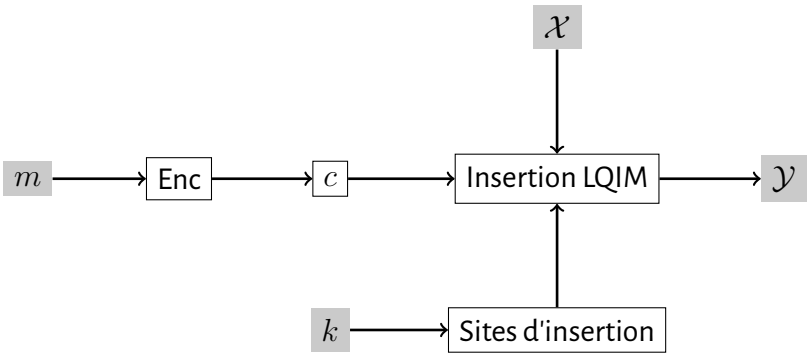


FIGURE 3.1 – Schéma d'insertion LQIM avec un code en métrique rang. $\mathcal{X}$ et $\mathcal{Y}$ représentent les images hôtes et marquées respectivement.

La méthode que nous proposons permet de combiner les codes en métrique rang et la méthode LQIM grâce aux étapes suivantes. Pour l'insertion, voici les étapes (illustrées dans la figure 3.1) :

1. Encoder le message m avec la fonction d'encodage de Gabidulin Enc.
2. Sélectionner les sites d'insertion aléatoirement grâce à une clé secrète k. Par exemple, les positions des pixels en niveaux de gris ou couleur, ou encore les positions des coefficients de l'image dans un domaine transformé.

3. Pour chaque pixel ou vecteur hôte, modifier les valeurs associées aux sites d'insertion correspondants, grâce à la méthode d'insertion LQIM.

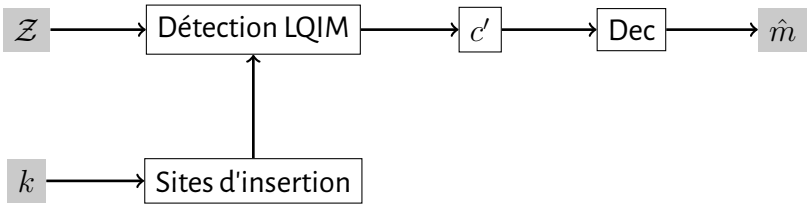


FIGURE 3.2 – Schéma de détection LQIM avec un code en métrique rang. $\mathcal{Z}$ représente l'image attaquée.

À l'étape de détection, nous avons les étapes suivantes (illustrées dans la figure 3.2) :

1. Extraire les sites d'insertion de la marque à l'aide de la clé k.
2. Pour chaque pixel ou vecteur, estimer le bit d'information avec la méthode de détection LQIM pour reconstuire le mot de code c'.
3. Enfin, transformer c' sous forme de matrice et décoder en métrique rang pour obtenir le message $\hat{m}$.

Dans la section suivante, nous proposons d'analyser l'impact de la perturbation provoquée par une modification de luminance sur notre méthode afin de mettre en évidence sa structure d'erreur particulière.

3.4 Contexte : attaque par modification de luminance

3.4.1 Définition de l'attaque

Nous proposons de réintroduire une attaque déjà présentée précédemment. Un changement de luminance est paramétré par une constante réelle β. Supposons que l'on ait récupéré une image marquée endommagée par une modification de luminance. À l'étape de détection, nous avons :

$$z = y + \beta \times u \tag{3.7}$$

avec $u = (1, \ldots, 1) \in \mathbb{R}^L$ et z la version modifiée d'un vecteur quantifié y. Tous les y subissent la même distorsion. Quelques exemples d'images ayant subi une modification de luminance sont proposés dans la figure 3.3. Lorsque $\beta < 0$, l'image devient plus sombre et inversement, lorsque $\beta > 0$, l'image blanchit (ou sature). Comme les niveaux de gris varient

entre 0 et 255, une image marquée peut perdre de l'information par effet de bords si $|\beta|$ est suffisament grand.

(a) $\beta = -60$ (b) $\beta = -40$ (c) $\beta = -20$ (d) $\beta = 0$

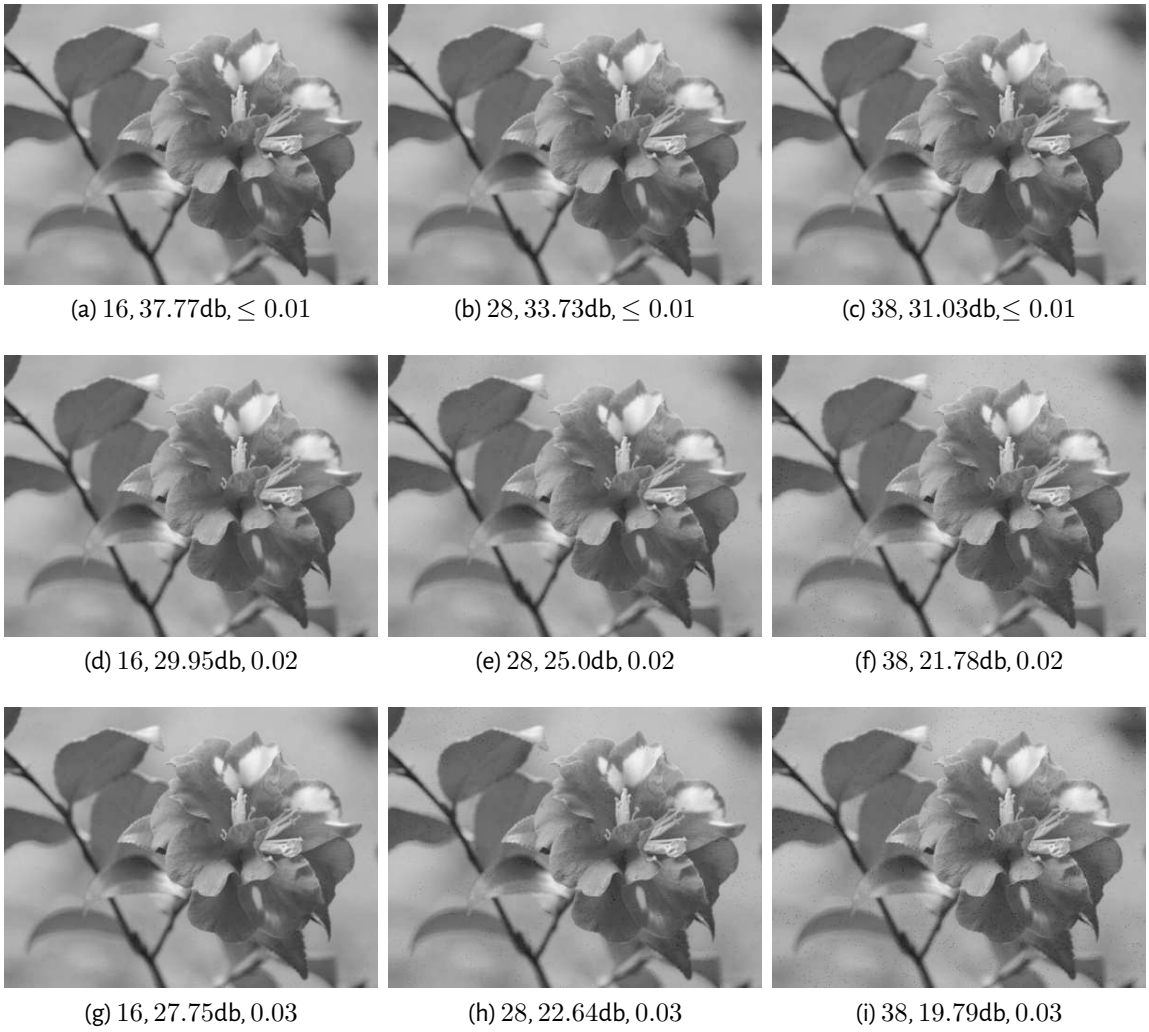
(e) $\beta = 20$ (f) $\beta = 40$ (g) $\beta = 60$

FIGURE 3.3 – Exemples d'images ayant subies une modification de luminance.

3.4.2 Analyse de la structure de l'erreur

Afin d'étudier empiriquement la structure d'erreur associée à cette attaque, nous montrons les taux d'erreur binaire (figure 3.5) entre le message original (qui représente un mot de code en métrique rang sur $GF(2^7)$ et de paramètres $[7, 3, 5]$) et le message décodé en fonction du paramètre β de l'attaque de luminance. Nous utilisons la base d'image Corel où 1000 images ont été choisies équiprobablement dans les 10000 images disponibles. Chaque marque contient un mot de code en tant que message de longueur 49 bits. Dans le cas d'une application de cette méthode de tatouage, le message avant encodage possède $km = 3 \times 7 = 21$ bits. La dimension de l'espace de quantification est $L = 6$ et le pas de quantification est $\Delta = 16$. Ces paramètres ont été fixés ainsi pour maintenir une qualité d'image fixe (DWR $\geq 30db$, voir figure 3.4 pour des exemples d'image).

Dans la figure 3.5, nous pouvons observer une courbe de taux d'erreur en escalier périodique oscillant entre 0 et 1. Celle-ci se caractérise par trois états. Dans le premier cas, BER $= 0$ (aucune erreur à la détection) périodiquement sur un intervalle donné (par exemple lorsque $0 \leq \beta \leq 6$, ou encore $22 \leq \beta \leq 34$). Dans le second cas, nous avons BER $= 0.5$ (détection



(a) $16, 37.77$db, ≤ 0.01 (b) $28, 33.73$db, ≤ 0.01 (c) $38, 31.03$db,≤ 0.01

(d) $16, 29.95$db, 0.02 (e) $28, 25.0$db, 0.02 (f) $38, 21.78$db, 0.02

(g) $16, 27.75$db, 0.03 (h) $28, 22.64$db, 0.03 (i) $38, 19.79$db, 0.03

FIGURE 3.4 – Exemples d'images marquées avec différents paramètres (Δ, DWR, ER). Plus le taux d'insertion (noté ER) et le pas de quantification Δ sont élevés, plus la qualité de l'image diminue (DWR décroit) et plus le bruit de quantification est visible à l'oeil nu. Les sites d'insertion ont été choisi aléatoirement ce qui explique l'apparition d'un bruit de type poivre et sel.

d'une séquence aléatoire) mais cette fois-ci de manière pontuelle ($\beta = 7, 21, \ldots$). Le dernier cas est similaire au premier mais avec BER $= 1$ c'est-à-dire que tous les bits du message sont inversés (par exemple $8 \leq \beta \leq 20, 36 \leq \beta \leq 49, \ldots$).

Cette courbe montre clairement l'existence d'une structure partielle de l'erreur et que les codes en métrique rang sont adaptés pour gérer ce type d'erreur. Tous les vecteurs quantifiés y subissent la même distorsion.

Quand β augmente, la valeur du pixel modifié z sature. D'un point de vue géométrique (en

2D pour la figure 3.6), tous les vecteurs z d'une cellule de quantification migrent vers une autre inversant le bit de la cellule courante à chaque changement de cellule.

Pour être plus précis, les trois cas précédemment décrits correspondent aux différents états décrits dans la figure 3.6. Pour $0 \leq \beta \leq 6$, nous avons BER $= 0$ et la distorsion correspondante est illustré dans la sous-figure 3.6a.

Puis, lorsque $\beta = 7$, nous avons vu que BER $= 0.5$ ce qui correspond à l'état illustré dans la sous-figure 3.6b car nous pouvons voir que les vecteurs modifiés par la distorsion de luminance sont positionnés sur la frontière de deux cellules de quantification voisines. Le détecteur LQIM estime alors avec une probabilité de 0.5 le bit originel.

Ensuite, nous observons que BER $= 1$ quand $8 \leq \beta \leq 20$, c'est-à-direque la séquence détectée est inversée par rapport à la séquence originelle car chaque vecteur z ont subi la même distorsion (état de la sous-figure 3.6c). À $\beta = 21$, nous avons un nouvel état transitoire avec BER $= 0.5$ et pour finir, la séquence détectée est de nouveau inversée (BER $= 0$) pour $21 \leq \beta \leq 30$ (sous-figuree 3.6d).

Nous allons montrer dans la sous-section suivante que l'utilisation d'un code en métrique rang combiné à la méthode LQIM permet de supprimer presque toutes les erreurs au décodage.

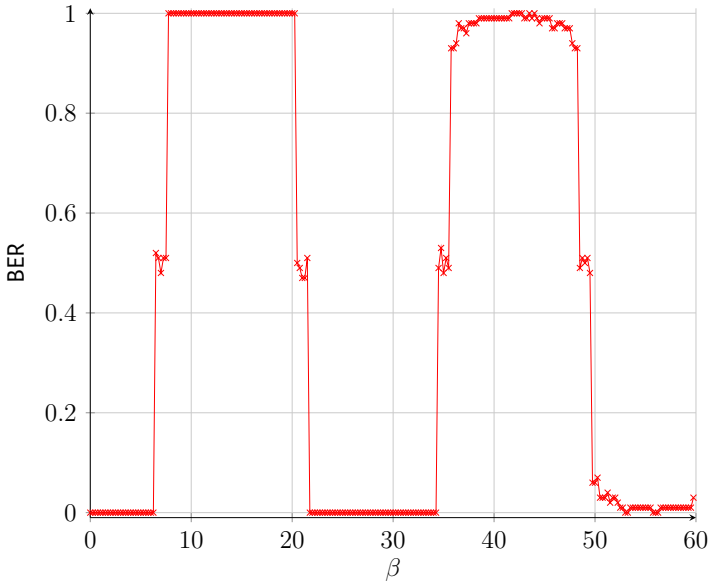
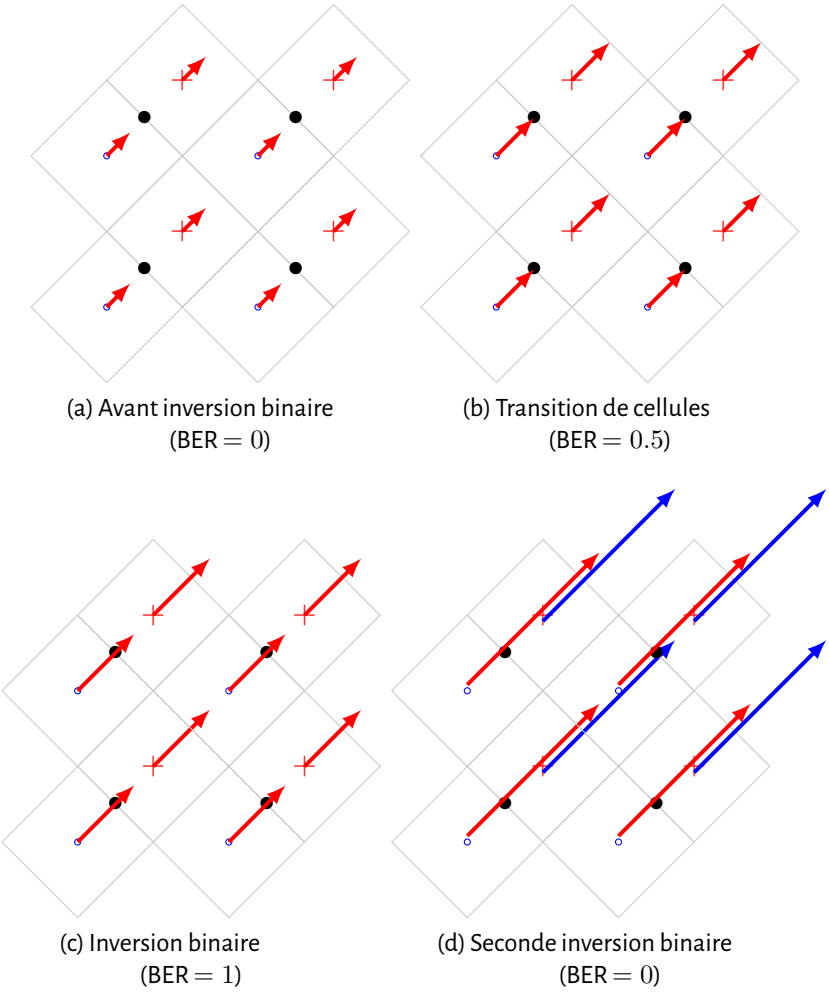


FIGURE 3.5 – Taux d'erreur binaire de la méthode LQIM dans le domaine spatial en fonction d'une modification additive de luminance de paramètre β. Ici, $\beta \geq 0$ mais la courbe se comporte de la même manière pour $\beta < 0$.



(a) Avant inversion binaire (BER $= 0$)

(b) Transition de cellules (BER $= 0.5$)

(c) Inversion binaire (BER $= 1$)

(d) Seconde inversion binaire (BER $= 0$)

FIGURE 3.6 – Représentation de l'espace de quantification en dimension $L = 2$ avec les trois cas observables après une modification de luminance.

3.4.3 Application des codes en métriques rang

Dans cette seconde expérience, nous avons combiné la méthode LQIM avec un code correcteur en métrique rang en insérant un mot de code en tant que message de la marque. Les paramètres du code sont $[7, 3, 5]$ et celui-ci peut corriger des erreurs de rang 2. Cette fois, nous mesurons des taux d'erreur image c'est-à-dire le ratio d'images pour lesquelles le message décodé n'est pas correct (i.e. le rang de l'erreur est strictement supérieur à 2).

Dans la figure 3.7, nous pouvons voir que le code en métrique rang est très efficace car le taux d'erreur image est nul presque partout et nous obtenons donc une correction quasiment

parfaite pour tout β. Le seul cas pour lequel le taux d'erreur image est égal à 1 est celui correspondant à un taux d'erreur binaire à 0.5, ce qui se traduit par le décodage d'une séquence binaire aléatoire. Il est admis que le rang d'une matrice à coefficients aléatoires possède une très forte probabilité d'avoir un rang maximal. Ladite structure d'erreur gérée par le code en

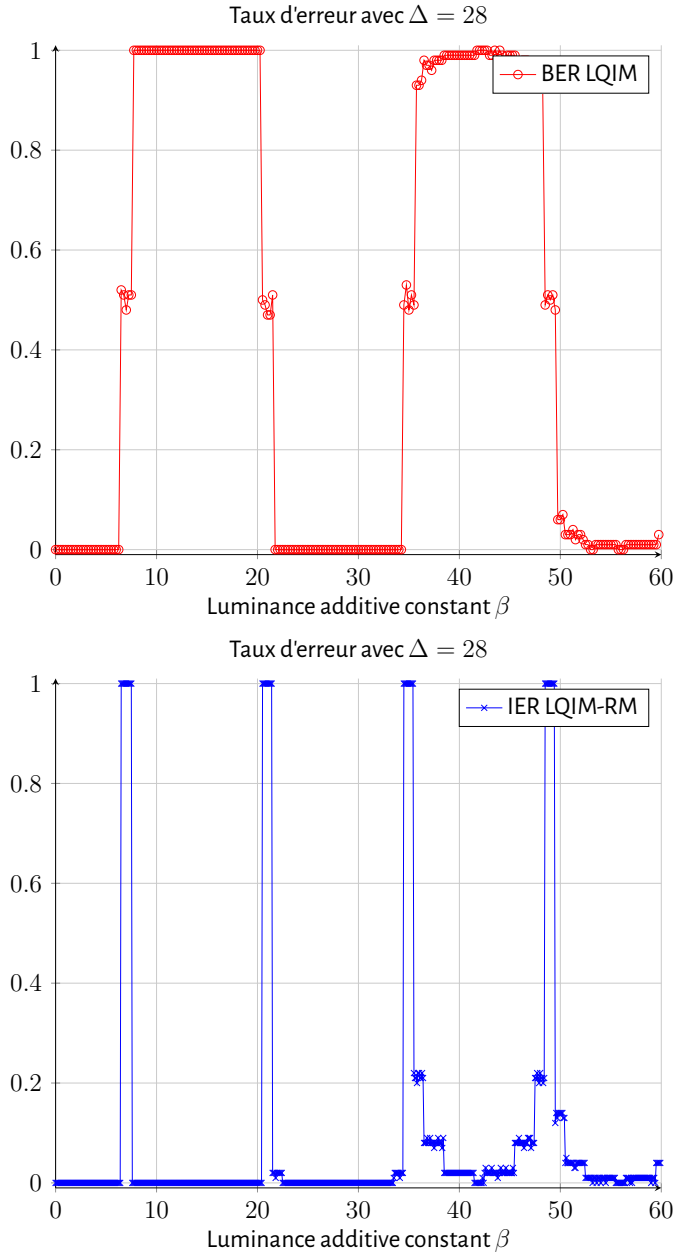


FIGURE 3.7 – En bleu, taux d'erreur binaire de la méthode LQIM en fonction de β. Pour des valeurs négatives de β, cette courbe possède la même allure. En rouge, taux d'erreur image de la méthode LQIM en fonction de β. Chaque point de cette courbe représente le ratio d'images où le rang de l'erreur $rk(e) \geq 2$ (décodage échoué).

métrique rang n'est plus présente, ce qui ne permet pas de corriger la séquence reçue vers le message originel.

À la détection, la probabilité de tomber sur une valeur de β pour laquelle l'erreur n'est pas structurée pour la métrique rang dépend du pas de quantification choisi Δ : plus il est petit, plus il est probable de ne pas pouvoir décoder car les cellules de quantification sont plus petites.

De plus, on a pu voir que la courbe de taux d'erreur binaire n'admet pas toujours un motif périodique (variations rapides du taux d'erreur de 0 à 1 ou inversement répétées périodiquement). Par exemple pour $55 \leq \beta \leq 50$, on voit que les taux d'erreur se comportent de manière instable au dessus de 0. Ceci est dû à la nature aléatoire des valeurs des pixels des images, c'est-à-direque certaines images saturent plus vite pour de petites valeurs de β (exemples d'image illustrés dans le figure 3.11). Un code capable de corriger des erreurs de rang 2 a été choisi mais, en théorie, un code corrigeant des erreurs de rang 1 suffit.

Dans le cas d'une modification de luminance d'une image, les codes en métrique rang permettent de corriger presque parfaitement les erreurs produites par cette attaque. Dans la sous-section suivante, nous proposons une amélioration du décodeur LQIM en métrique rang pour éliminer les erreurs restantes en introduisant une stratégie de détection multiple associée à un vote majoritaire.

3.4.4 Optimisation du décodeur LQIM + métrique rang

En partant de l'équation 3.7, il est possible d'optimiser les performances de décodage en modifiant de manière contrôlée la luminance de l'image.

Tout d'abord, nous pouvons remarquer que les valeurs de β pour lesquelles nous ne pouvons pas décoder se trouvent à intervalles réguliers. Ces valeurs de β représentent les transitions des vecteurs z d'une cellule à une autre. D'après la section 3.3, la construction des cosets permet d'en déduire que ces valeurs sont en fait des multiples de $\sqrt{2}\Delta/4$ (moitié de la distance entre un cercle et une croix les plus proches).

Soient $\delta_1, \ldots, \delta_n$ des entiers positifs plus petit que $\sqrt{2}\Delta/4$. D'après l'observation des différentes courbes de taux d'erreur image (figure 3.8) et les taux d'erreur binaire associés(figure 3.9), on voit qu'en dégradant volontairement l'image, les courbes d'erreur sont décalées et les varia-

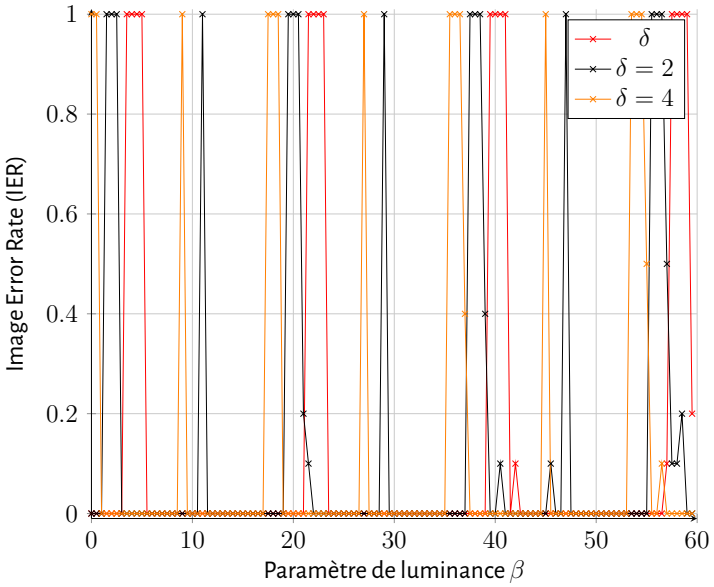


FIGURE 3.8 – Taux d'erreur image de la méthode LQIM combinée avec une code correcteur en métrique rang en fonction de β avec $\delta = 0, 2, 4$..

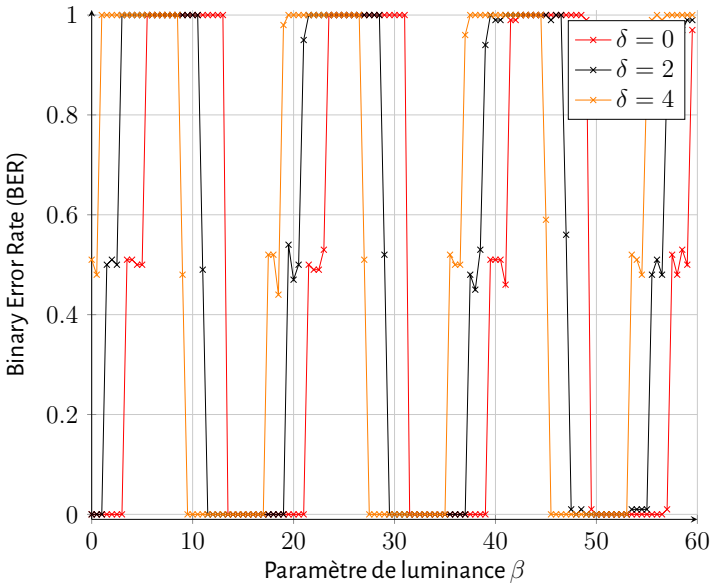


FIGURE 3.9 – Taux d'erreur binaire du décodeur LQIM en fonction du paramètre de luminance β avec $\delta = 0, 2, 4$.

tions brutales de taux d'erreur (pics) se produisent à intervalles réguliers.

Nous pouvons donc en déduire la propriété suivante : pour $1 \leq i \leq n$ il existe un unique entier $1 \leq i_0 \leq n$ tel que l'image modifiée $z + \delta_{i_0}$ ne peut être décodée avec le décodeur LQIM + métrique rang. De plus, pour tout $1 \leq j \leq n, j \neq i_0$, l'image modifiée $z + \delta_j$ est

parfaitement décodée avec le décodeur LQIM + métrique rang.

En modifiant de manière contrôlée la luminance de l'image reçue avec de petites valeurs δ, nous pouvons garantir que pour une valeur de β fixée et un vecteur z fixé, la majorité des $z+\delta_j$ auront un décodage correct c'est-à-dire que la valeur issue de la majorité des bits décodés sera identique à celui inséré au départ. Avec cette stratégie, nous pouvons donc décoder sans erreur au prix d'un temps de décodage n fois plus long. Une valeur de $n = 3$ suffit pour décoder sans erreur avec :

$$\begin{cases} d = \frac{\sqrt{2}}{4}\Delta \\ \delta_1 = 0 \\ \delta_2 = \frac{1}{3}d \\ \delta_3 = \frac{2}{3}d \end{cases}$$

Exemple de décodage : On a les valeurs suivantes :

$$\begin{cases} d \simeq 6 \\ \delta_1 = 0 \\ \delta_2 \simeq 2 \\ \delta_3 \simeq 4 \end{cases}$$

qui représentent les versions modifiées de l'image transmise z. On extrait alors trois estimations du message original m_1, m_2 et m_3. D'après la propriété, deux des trois messages extraits sont corrects pour β fixé. Nous faisons donc un vote à la majorité.

Le résultat du décodage amélioré est illustré par les résultats d'expérience présentés dans la figure 3.10. Nous observons que les taux d'erreur sont nuls quasiment partout.

Cependant, pour $\beta = 55, \ldots, 60$ par exemple, nous pouvons observer des taux d'erreur non nuls (variation brutale des taux d'erreur, i.e. apparition de pics en fin de courbe). À cause de la nature aléatoire des images choisies, il arrive que la distorsion infligée par la modification de luminance soit tellement grande que l'information insérée est perdue à cause des effets de saturation (valeur des pixels comprises entre 0 et 255). Des exemples d'image illustrant ce cas de figure sont présentés dans la figure 3.11.

Dans le cas de nos expériences, nous avons choisi $\beta > 0$ ce qui explique que les déco-

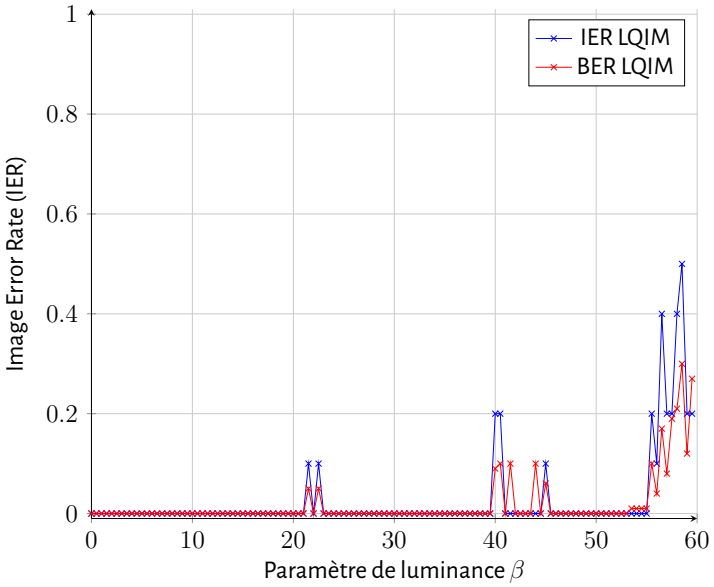


FIGURE 3.10 – Taux d'erreur binaire et taux d'erreur image associé du décodeur LQIM + métrique rang amélioré dans le domaine spatial en fonction du paramètre de luminance β.

dages qui ont échoués se sont produits avec des images très claires. Si nous avions considéré le cas $\beta < 0$ (similaire au cas précédent), nous aurions obtenus également des décodages défaillants avec des images très sombres. Dans ce cas, nous pouvons considérer que l'information initiale est elle-même très dégradée.

Pour résumer, nous avons proposé une stratégie pour améliorer le décodeur LQIM couplé avec le décodage d'un mot de code en métrique rang. Le cas où le taux d'erreur binaire est égal à 0.5 peut être évité en prenant une estimation à la majorité du décodage de plusieurs images dont la luminance a été volontairement modifiées à partir de l'image transmise (les piques observables sur les courbes de taux d'erreur image ont été supprimées). Nous proposons maintenant d'étudier d'autres attaques intervenant sur la valeur du pixel.

3.5 Attaques entrainant des modifications de valeurs “ponctuelles”

La section précédente a été dédiée à l'étude de la modification de luminance dont les erreurs produites étaient structurées et donc adaptées pour des codes correcteurs en métrique

(a) 706.jpg, $\beta = 59$

(b) 5371.jpg, $\beta = 58$

(c) 6211.jpg, $\beta = 57$

FIGURE 3.11 – Exemples de couples d'images marquées/attaquées représentant les cas où l'information insérée est effacée après une modification de luminance.

rang. Dans cette section, nous nous intéressons à d'autres attaques telles que la compression JPEG, la modification de contraste et l'ajout de bruit additif blanc gaussien (AWGN) dont les taux d'erreur sont donnés dans la figure 3.12. Pour chacune de ces attaques, nous ré-utilisons le même procédé d'analyse que pour le chapitre précédent : évolution du taux d'erreur en fonction du paramètre de l'attaque.

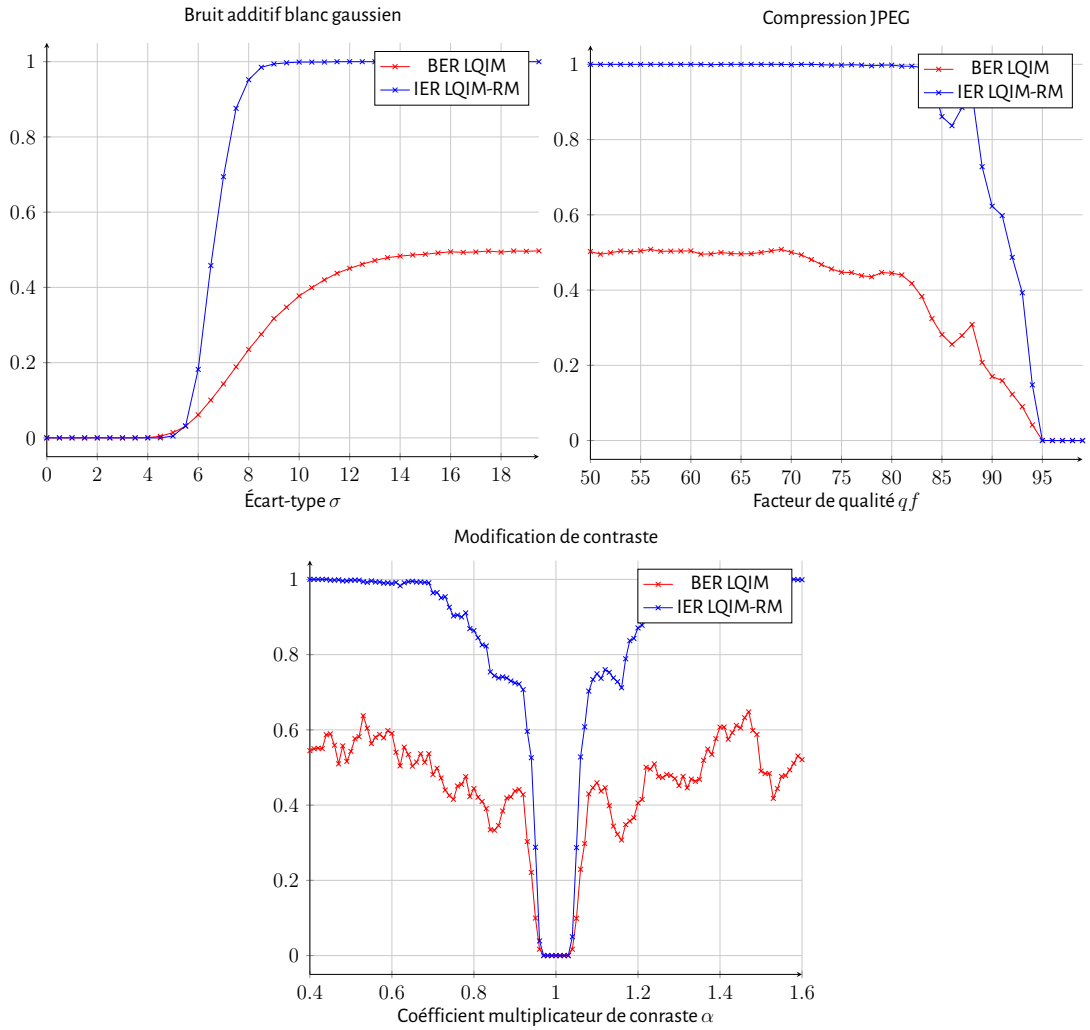


FIGURE 3.12 – Taux d'erreur binaire et image de la méthode LQIM et LQIM-RM respectivement en fonction de différentes attaques dans le domaine spatial dans la même configuration donnée dans la section précédente.

Quand le bruit produit par chaque attaque est trop important, les taux d'erreur binaire moyens (courbe en rouge) tendent vers 0.5. De même pour les taux d'erreur image moyens (avec l'application d'un code en métrique rang), ceux-ci tendent vers 1. Dans le cas du bruit gaussien, c'est à partir de la valeur $\sigma = 5$ que les taux d'erreur ne sont plus nuls. Pour la compression JPEG, nous avons des valeurs non nulles pour des indices de qualité en dessous de 95. De même pour la modification de contraste avec $\alpha \notin [0.95, 1.05]$.

Pour ces trois attaques, aucune structure d'erreur adaptée pour une correction en métrique rang n'est apparente. En effet, dans l'évolution des taux d'erreur binaire, nous avons pu observer un premier cas où ceux-ci sont égaux à 0 puis un état transitoire où les valeurs tendant vers

un taux d'erreur égal à 0.5 et enfin des taux d'erreur binaire égaux à 0.5. Ce dernier cas correspond au décodage d'un message représentant une séquence aléatoire et n'est pas une structure adaptée au décodage en métrique rang.

En effet, le rang d'une erreur (c'est-à-direle rang d'une matrice binaire aléatoire) possède une très forte probabilité d'avoir un rang maximal et ne nous permet donc pas une correction avec des codes en métrique rang. Comme nous l'avons mentionné précédemment, les codes optimaux face aux erreurs aléatoires sont les codes BCH.

Une application efficace des codes en métrique rang au tatouage numérique face aux attaques précédemment étudiées n'est pas possible car aucune structure d'erreur n'est visible, au moins, dans notre configuration de tatouage.

Même si la distance rang ne permet de corriger uniquement que des erreurs d'une structure bien particulière, l'espoir d'une application plus large de ces codes réside non seulement dans le type d'attaque auquel on fait face mais aussi dans la manière d'insérer de l'information à protéger. Ceci renforce donc l'idée précédemment développée dans [3] que pour un type d'attaque, il existe un code correcteur plus efficace qu'un autre.

Nous proposons maintenant de nous intéresser à une autre famille d'attaque : la perte d'information par cropping.

3.6 Une méthode de tatouage robuste au découpage d'image

3.6.1 Bref état de l'art

Le découpage d'image (ou *image cropping*) efface une région d'une image ce qui endommage sérieusement celle-ci. L'information qui a localement été insérée est donc détruite. D'après [96], une des plus vieilles contributions sur le découpage d'image a été proposée par Swanson et al. [97, 98]. Leur méthode de tatouage utilise un algorithme de modulation des bits de poids faibles sur les coéfficients DCT pour insérer leur marque. Le découpage d'image peut aussi être associé avec une attaque similaire appelé *collage attack* qui consiste à remplacer une région d'une image par une autre. Cette attaque est aussi connue comme étant une variation de l'attaque de contrefaçon d'image étudiée par Holliman-Memon [99] dans un contexte d'authentification d'image numérique étudié par Fridrich et al. [100].

Ce type d'attaque est surtout étudié dans des problématiques d'authentification d'image ou de détection de modifications (malicieuses ou non). Plus tard, d'autres travaux tel que [101] ont proposé un mécanisme d'insertion d'une image dans elle-même (ou *self-embedding*) pour résister à plusieurs attaques telles que la falsification, le découpage (ou recadrage) ou encore le remplacement de régions dans image. Leur idée consiste à insérer une version compressée de l'image hôte dans elle-même à l'aide d'une modulation des bits de poids faibles sur les coefficients DCT.

Des travaux plus récents ont été publiés sur le même sujet (voir [102, 103, 104, 105]). Leur approche est originale et s'inspire des mathématiques du Sudoku [106, 107]. Leur stratégie d'insertion se base sur le même principe que le mécanisme de self-embedding (réduction de la taille de l'image hôte pour obtenir une marque de taille plus petite) précédemment décrit pour résoudre le problème du découpage et du recadrage d'image. Dans leur étape d'insertion, l'image hôte est divisée en N cellules identifiées par les entiers de 1 à N (comme dans une grille de Sudoku). Une nouvelle image est alors générée avec chaque cellule replacée dans une grille solution dont la taille est réduite puis insérée dans l'image hôte grâce à une modulation des bits de poids faibles sur les coefficients DCT.

Cependant, ces contributions n'abordent que des problèmes d'authentification d'image et de détection de modifications par des méthodes de tatouage fragile. L'idée de ce paradigme de dissimulation de données est d'insérer un motif de telle sorte que celui-ci soit modifié si et seulement si l'image hôte a été modifié, falsifié, etc. Comparé au tatouage robuste, les objectifs sont complètement différents. Par exemple, le contenu d'un tatouage robuste doit être résistante aux attaques alors que dans une stratégie de tatouage fragile, nous pouvons uniquement dire si l'image marquée est modifiée ou non.

Le problème du découpage d'image devient plus compliqué à étudier lorsque la taille de l'image attaquée est de taille inférieure à celle de l'image marquée. À l'étape de détection, la marque doit être synchronisée avant de pouvoir l'extraire. Une contribution pour traiter ce problème a été proposée par Kutter [108] en 1999. Pour insérer une marque robuste, celle-ci est répétée à plusieurs endroits de l'image. C'est cette méthode qui est utilisée dans tous les travaux précédemment présentés.

Nous proposons d'utiliser la particularité du code à métrique rang pour lutter contre ce type d'attaque.

3.6.2 Description de la méthode proposée et discusions

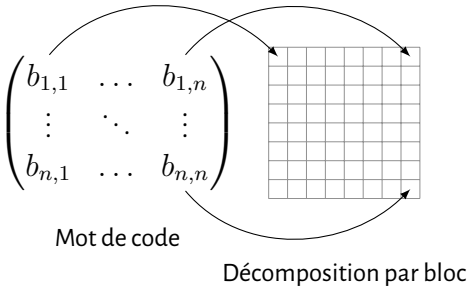


FIGURE 3.13 – Stratégie d'insertion à l'aide d'un code en métrique rang et d'une décomposition par bloc de image. Chaque bit $b_{i,j}$ est associé à un bloc.

Comme nous l'avons vu, il semble plutôt difficile de concevoir une méthode résistante à ce type d'attaque. Fondamentalement, toutes les contributions se basent sur l'utilisation de la répétition d'information, i.e. sur l'utilisation d'une forme de redondance d'information. Par conséquent, chaque information localement insérée doit dépendre d'autres informations insérées dans d'autres régions de l'image pour détecter la marque sans erreur.

La seule solution pour reconstituer le message initial sans erreur est de répartir la marque dans toute l'image. Si une région est découpée et est suffisament petite, alors le message peut être extrait correctement. Dans la littérature, les approches utilisées sont équivalentes à utiliser un code à répétition.

Nous proposons une approche différente pour traiter le problème du découpage d'image à l'aide des codes en métrique rang. Pour cela, une image hôte est décomposée en n^2 blocs avec n la longueur d'un code en métrique rang (schéma de décomposition illustré dans la figure 3.13). Puis, chaque bit du mot de code rang est inséré dans un bloc en utilisant $L = 2$ coefficients. Nous obtenons une image qui ressemble exactement à la matrice d'un mot de code rang. Par conséquent, les distortions produites par l'image attaquée se reproduisent directement sur la matrice erreur e.

Par exemple, une région de forme carrée (de taille l) est découpée. Les blocs affectés par cette attaque sont directement transposés sur la matrice d'erreur e. Considérons maintenant une ligne ou une colonne de largeur l découpée. Dans la matrice e, une ligne ou une colonne de bit 1 apparait. Comme nous l'avons vu précédemment, e possède une structure d'erreur particulière qui est parfaitement gérée par des codes en métrique rang. En effet, $rk(e) = r'$ avec r' le nombre de blocs (en largeur) affectée par le découpage d'image.

Figure 3.14 – Images découpées dont les erreurs sont de même rang.

Les deux exemples précédemment décrits produisent des erreurs de même rang. Dans le second cas, la distorsion est maximisée comparé au premier cas (carré découpé) pour des erreurs de même rang. Dans la figure 3.14, nous montrons des exemples d'images attaquées qui ont tous une matrice d'erreur de même rang.

Un autre exemple intéressant à propos de la métrique rang est le suivant : permuter deux lignes ou colonnes de même largeur ne modifie pas le rang de la matrice d'erreur si les coefficients choisis dans chaque bloc ont les mêmes positions et que les bandes coupées chevauchent exactement la décomposition en bloc de l'image. Cette opération de permutations peut être répétées indéfiniment sans faire varier le rang de l'erreur. Ce cas d'étude n'est pas considéré dans ce chapitre par manque de réalisme.

Dans la sous-section suivante, nous allons décrire et analyser la robustesse de la méthode proposée contre le découpage d'image.

3.6.3 Application de la métrique rang contre le cropping

Dans notre étude, nous distinguons deux types de découpage : le premier type regroupe des images dont des bandes verticales ou horizontales sont entièrement découpées (subfigure 3.15a) et le second type regroupe des images où seulement des formes rectangulaires sont découpées (subfigure 3.15b). Notons également que les bandes découpées ne chevauchent pas forcément la décomposition par bloc de l'image. Pour des erreurs de même rang, la distorsion est maximisée avec le premier type comparée au deuxième.

Dans nos mesures expérimentales, nous considérons le premier type de découpage pour des raisons pratiques et de clarté. Avec le second type, le rang moyen de l'erreur est exactement le même avec un pourcentage de découpage plus faible.

Nous mesurons les distorsions des images à l'aide du pourcentage de régions de l'image découpée noté cr. Pour le premier type, nous définissons alors cr tel que :

$$cr = 100.\frac{l}{h} \tag{3.8}$$

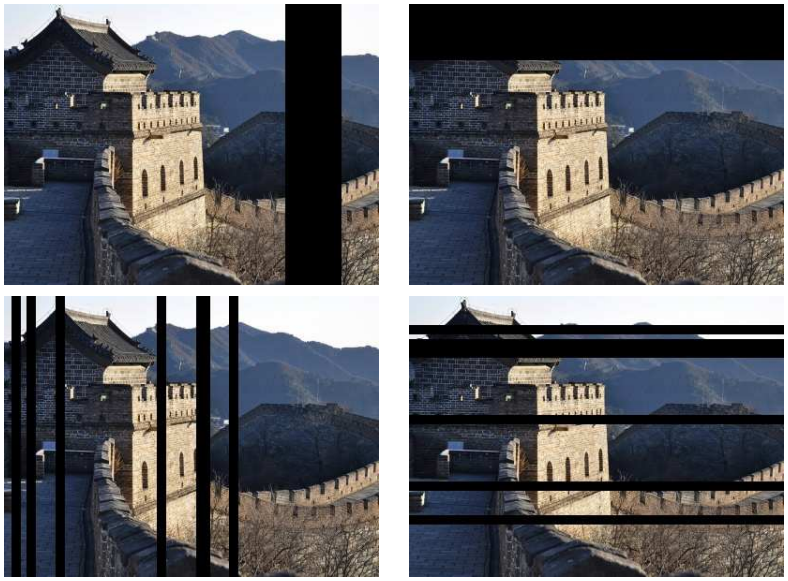
avec l le nombre de pixel en largeur de la colonne découpée et h la hauteur de l'image. En d'autres termes, nous considérons des découpages de largeur l sur le côté gauche de l'image.

Nous avons aléatoirement choisi 1000 images de la base d'image Corel dont la taille est 300×400 ou 400×300 et calculons les moyennes respectives des taux d'erreur binaire, des taux d'erreur image et des rangs d'erreur. Nous avons choisi differents codes de Gabidulin pour mesurer la robustesse de la méthode proposée face au découpage d'image.

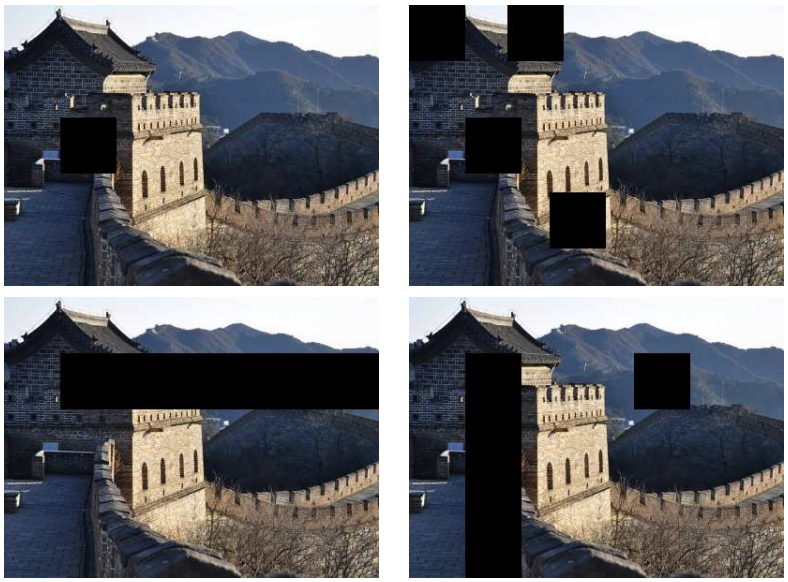
Premièrement, la méthode proposée peut être résistante au découpage d'image par construction. Dans la figure 3.17, nous observons que les courbes de taux d'erreur binaire et de rangs de l'erreur sont linéairement croissantes quand cr augmente pour tout n.

Quand le rang de l'erreur moyen (± 1.6) devient supérieur au nombre maximum d'erreur, nous pouvons voir que les courbes IER varient rapidement de 0 à 1. Il devient donc impossible de détecter le message originel. Nous notons alors cr_{max} le pourcentage le plus haut permettant une détection sans erreur.

Cette première observation montre expérimentalement l'applicabilité des codes en métrique rang dans une stratégie de tatouage numérique face au découpage d'image. Bien sûr, il arrive que les performances de détection soient très mauvaises. L'idée principale de la mé-



(a) Type 1



(b) Type 2

FIGURE 3.15 – Types de découpage d'image

thode proposée est de profiter des propriétés mathématiques du rang d'une matrice. Il est possible de fabriquer un exemple d'image marquée puis découpée où une détection est impossible (voir figure 3.16). Puisque la décomposition en blocs de l'image est directement liée avec la position des bits du mot de code rang inséré, il suffit de découper dans l'image des zones rectangulaires sur un nombre suffisant de lignes et de colonnes pour augmenter le rang de l'er-

$cr' = 7.5\%$ $cr' = 6.75\%$

FIGURE 3.16 – Exemples d'images attaquées obtenant les plus mauvaises performances de détection. Le rang de l'erreur est grand ($\geq (n-k)/2$) par rapport au pourcentage de région découpée (cr').

reur et que celui-ci soit supérieur à $(n-k)/2$.

En discutant des paramètres plus en détails, quand le taux d'encodage k/n baisse, la puissance de correction t augmente ce qui permet d'obtenir une valeur de cr_{max} plus élevée, i.e., la marque est plus robuste. De même, quand n augmente avec k fixé.

Cependant, la qualité de l'image doit être évaluée avec attention à l'insertion de la marque car si n augmente cela dégrade de manière importante l'image hôte. En effet, une valeur plus grande de n implique la quantification de $n^2 \times L$ coefficients (les blocs sont plus petits).

Comparés aux codes BCH, les codes en métrique rang sont plus efficaces lorsque nous sommes confrontés au premier type de découpage d'image. Nous pouvons voir que les valeurs de cr_{max} atteintes par les codes en métrique rang correspondent à des taux d'erreur binaire plus élevés que les taux de correction donnés par des codes BCH équivalents. Par exemple, avec $(n,k) = (23,8)$ et $cr_{max} = 37\%$, nous avons BER $= 0.12 > t'/n' = 0.09$ (voir table 3.18 pour d'autres exemples).

Néanmoins, les codes en métrique rang sont moins efficaces que les codes BCH quand nous faisons face au découpage d'image du second type. Ceci s'explique par le fait que le taux d'erreur binaire obtenu à la valeur cr_{max} est plus petit que le taux de correction t'/n' associé au code BCH équivalent. De plus, les codes BCH sont capables de corriger des erreurs aléatoires et par conséquent, sont plus robustes contre tout type de découpage d'image si le taux d'erreur binaire est plus petit que t'/n'.

Pourtant, nous sommes convaincus que les codes en métrique rang restent un meilleur choix de codes correcteurs surtout dans le cas où les codes de Gabidulin sont MRD (Maximum Rank Distance). Même si les codes BCH sont optimaux pour des erreurs aléatoires, nous sommes

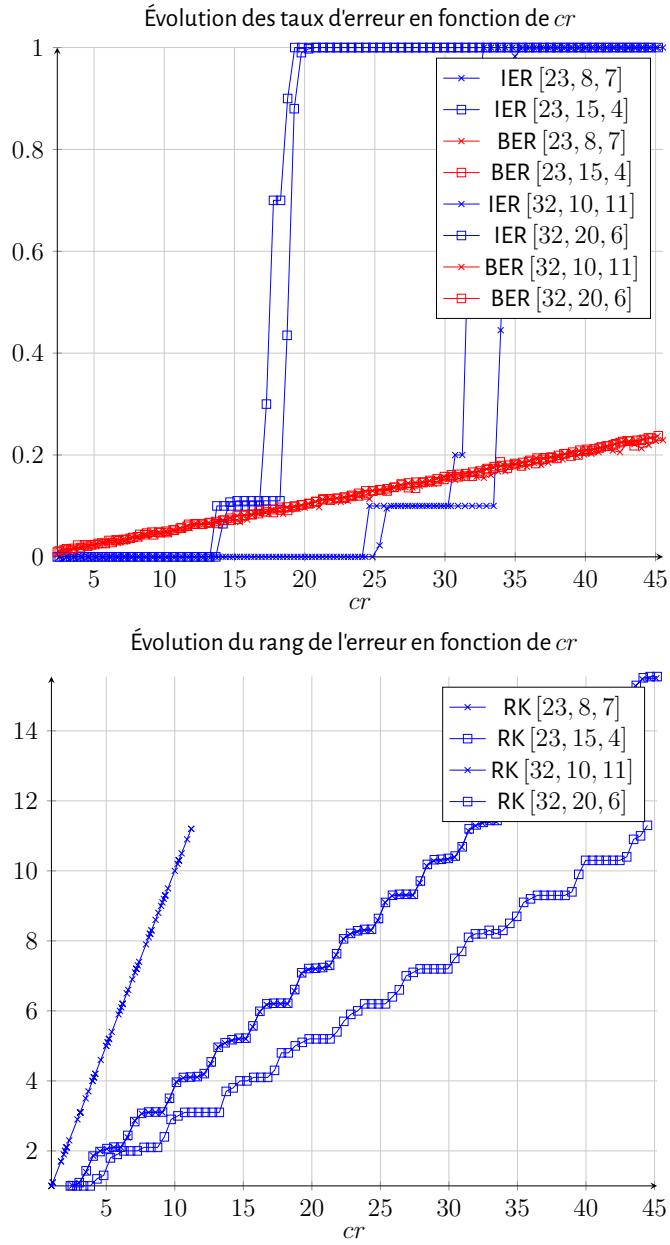


FIGURE 3.17 – Taux d'erreur et rang moyens en fonction du pourcentage de découpage cr.

contraints par le choix des paramètres. Par exemple, on ne pourrait choisir uniquement des longueurs de code n tel que $n = 2^m - 1$.

Dans cette section, nous avons montré que les codes en métrique rang permettent d'obtenir une résistance au découpage d'image. Combiné avec la méthode LQIM et une stratégie de décomposition par bloc dans le domaine spatial, nous avons montré que la méthode proposée

Gabidulin			BCH	
$[n, k, t]$	cr_{max}	BER à cr_{max}	$[n', k', t']$	t'/n'
$16, 5, 5$	41%	0.11	$255, 87, 26$	0.10
$16, 8, 4$	42%	0.1	$255, 131, 18$	0.07
$16, 11, 2$	41%	0.04	$255, 171, 11$	0.04
$23, 8, 7$	37%	0.12	$511, 175, 46$	0.09
$23, 12, 5$	36%	0.10	$511, 250, 31$	0.06
$23, 15, 4$	39%	0.07	$511, 340, 20$	0.04
$32, 10, 11$	35%	0.13	$1023, 348, 87$	0.09
$32, 16, 8$	35%	0.10	$1023, 513, 57$	0.06
$32, 21, 5$	35%	0.07	$1023, 688, 36$	0.04

FIGURE 3.18 – Table de paramètres de codes correcteurs. Pour chaque ligne de la table, nous avons les paramètres d'un code en métrique rang à gauche et à droite les paramètres d'un code BCH équivalent ($t/n \simeq t'/n'$).

peut être robuste à différents types de découpage d'image en prennant en compte certaines contraintes de paramètres et de compromis. De plus, ces codes nouvellement introduits au tatouage numérique permettent d'obtenir de meilleures performances de détection dans certains cas (premier type de découpage d'image) Cependant, il reste un problème important associé à cette modification d'image : la synchronisation de la marque lorsque la taille de l'image attaquée est différent de celle de l'image marquée. Une approche pour résoudre ce problème est de travailler avec une représentation de l'image qui est insensible aux modifications géométriques. Par exemple, le recadrage d'image peut s'inscrire comme une forme de translation de l'image. Ainsi, la recherche de points d'intérêt [109] dans une image serait une piste d'étude fiable pour résoudre ce problème.

3.7 Conclusion

Nous avons présenté un nouveau type de codes correcteurs d'erreur utilisant la métrique rang au lieu de la métrique de Hamming qui est habituellement utilisée. Techniquement, les codes code Gabidulin ont été combinés à la méthode LQIM pour obtenir un nouveau type de résistance à des attaques particulières.

Tout d'abord, notre analyse de la modification de luminance nous a permis de montrer que les codes en métrique rang peuvent avoir un grand pouvoir de correction si la structure de l'erreur est adaptée. Nous avons proposé une stratégie de multi-détection (répétition de la

détection sur des versions volontairement modifiées de l'image attaquée) pour obtenir une résistance théoriquement parfaite à cette attaque.

Dans un second temps, nous avons ajouté à la méthode précédente une stratégie de décomposition par bloc de l'image au lieu d'insérer de l'information à des positions aléatoires. L'image est divisée en blocs et chaque bit d'un mot de code rang est associé à un bloc. Cette stratégie d'insertion nous permet de profiter des avantages de la structure de la métrique rang face à différentes configurations de découpage d'image. Après avoir étudié la robustesse de la méthode proposée avec les codes en métrique rang contre cette attaque, nous montrons également que les codes de Gabidulin sont plus efficaces que les codes BCH quand les distorsions sont maximisées pour un rang d'erreur fixé.

Dans ce chapitre, nous avons montré le potentiel des codes en métrique rang pour le tatouage numérique. Ces codes permettent de corriger des erreur d'une structure particulière que les codes de Hamming classiques ne peuvent pas gérer.

Bien que l'étude de la structure des erreurs est une approche bien connue pour améliorer la robustesse d'une marque, nous sommes convaincus qu'il reste encore bien des perspectives à étudier pour ces codes. Théoriquement, les erreurs produites par un canal dit bruité doivent être soigneusement étudiées pour pouvoir les corriger.

Néanmoins, nous pensons que la méthode d'insertion ainsi que le choix des sites d'insertion doivent aussi être pris en compte comme nous avons pu le voir avec le choix de la méthode d'insertion et la décomposition par bloc de l'image.

Une perspective d'étude est l'étude d'autres types de méthodes d'insertion tels que les méthodes d'insertion basées sur les treillis. Pour finir, il serait également intéressant d'étudier le codage par syndrôme (déjà utilisé en stéganographie) avec les codes en métrique rang.

Ce chapitre termine l'étude correspondant à l'optimisation de la robustesse à travers l'utilisation de codes. Nous proposons maintenant d'analyser l'une des caractéristiques complémentaires à savoir l'invisibilité.

CHAPITRE 4

État de l'art et approche psychovisuelle pour le tatouage des images couleur

Contenu

4.1 Introduction et état de l'art

Dans ce chapitre, nous allons nous intéresser à la prise en compte du Système Visuel Humain (SVH) pour le tatouage numérique afin d'optimiser son invisibilité. De nombreuses méthodes de tatouage existent pour les images en couleur. Les premières solutions ([110, 111]) ont proposé de tatouer sur la composante bleue pour minimiser les changements perceptuels de l'image tatouée, ce qui malheureusement fragilise la marque. D'autres méthodes ont consisté à modifier la composante de luminance [112] ou celle de la saturation [113] ce qui leur a permis d'être plus robustes mais impliquent une forte distorsion lors de l'insertion. Ces méthodes traitent les composantes des vecteurs couleur de manière indépendante. La notion de couleur n'est donc pas prise en compte et ne permet donc pas d'obtenir une invisibilité optimale pour le SVH.

Plus tard, d'autres travaux utilisant des approches vectorielles du traitement des images couleur sont apparus. Abadpour et al. [114] ont par exemple exploité des informations issues des projections sur les composantes principales d'une analyse en composantes principales, mais n'ont donc pas pris en compte le SVH.

Les travaux proposés par Chareyron et al. [115] permettent d'insérer une marque 2D dans le plan chromatique xy. Leur approche se base sur une manipulation d'histogramme proposée par Coltuc et Bolon [116]. Pour améliorer l'invisibilité de la marque, les distances couleur correspondant aux distorsions sont calculées dans l'espace couleur L*a*b*. Plus tard, ils ont proposé d'étendre leur méthode dans l'espace L*a*b* dans [117], ce qui a permis d'améliorer l'invsibilité.

Des contributions plus récentes ont proposées de travailler dans des domaines transformés, telles [118] qui se base sur des décompositions d'ondelettes particulières (*curvelets*) et [119] qui se base sur la transformée en cosinus discrète quaternionique. Il existe aussi des méthodes vectorielles sur coefficients d'ondelettes calculés à partir des trois plans RGB ([120] par exemple). Cependant, celles-ci ne prennent pas en compte la dimension perceptuelle de la couleur.

À notre connaissance, la perception des différences de couleur n'a pas complètement été explorée dans le cadre du tatouage. Selon nous, une compréhension plus fine du traitement de la couleur par le SVH permettrait d'améliorer l'invisibilité d'une marque.

Ce chapitre détaille l'étude et la conception d'une méthode de quantification vectorielle (section 4.2) associée à un modèle biologique des photorécepteurs du SVH pour pouvoir insérer de l'information dans des images en couleur. Ce modèle se base sur une approche psy-

chovisuelle (section section 4.3) permettant de comprendre la perception des différences de couleur du SVH (travaux introduits par Alleysson [121]). Nous l'adaptons ensuite au tatouage numérique dans la section 4.4 afin de minimiser les distorsions psychovisuelles. Puis, nous détaillons les différentes étapes d'un scénario de tatouage robuste pour les images en couleur ainsi que les algorithmes utilisés afin d'y adapter la méthode Lattice QIM (LQIM).

Afin de valider notre approche psychovisuelle pour le tatouage numérique des images couleur, nous proposons des expériences psychovisuelles et de robustesse en fin de chapitre.

4.2 Quantification couleur

Comme nous l'avons dit, différentes approches ont été proposées avec notamment l'utilisation d'espace couleur adapté à l'insertion d'un tatouage invisible à l'oeil nu. L'une des première approche a consisté à insérer une marque dans la composante bleu de l'espace RGB car l'oeil humain est moins sensible aux distorsions dans ce canal [110, 111]. La marque est alors moins visible mais résiste moins aux modifications d'image. Le compromis inverse se produit en insérant dans la composante L de l'espace couleur L*a*b* [112]. D'autres travaux proposent aussi de réduire ces distorsions d'insertion en travaillant dans l'espace L*a*b* tels que [115, 117] et parviennent à améliorer l'invisibilité en profitant de la faible sensibilité du SVH aux différences de couleur. Cependant, une compréhension plus fine du SVH n'est pas présente.

Puis, nous avons d'une part des travaux qui traitent la couleur de manière vectorielle et non plus indépendemment sur chaque compostante telles que [114] ce qui représente une évolution dans l'approche du tatouage numérique couleur. En ajoutant un traitement perceptuel de la couleur au sens du SVH, il est possible de minimiser les distortions couleur à l'insertion.

Dans notre étude, nous proposons un modèle psychovisuel permettant comprendre la perception des différences de couleur. Nous nous positionnons ensuite dans l'espace RGB et connectons celui-ci à l'espace du modèle des photorécepteurs du SVH afin de minimiser les distorsions d'une marque à l'insertion.

Ce concept perceptuel existe déjà pour les images en niveaux de gris dans la littérature du tatouage. Par exemple, le modèle perceptuel de Watson [44] proposé en 1993 permet d'optimiser visuellement les matrices DCT de quantification pour une image donnée lors de sa compression grâce à des adaptations de contraste et de luminance. Ces travaux ont été intégré au tatouage numérique par Li et al. [122] et aussi par Hu et al. [45] afin de minimiser le bruit d'in-

sertion.

Nous proposons de modifier chaque pixel en fonction de sa couleur et de la perception associée (ou de la dégradation acceptable). Pour cela, tout d'abord une méthode de quantification vectorielle est décrite.

4.2.1 Quantification vectorielle dans l'espace RGB

La quantification vectorielle nous permet d'insérer de l'information sur les valeurs d'un pixel couleur (trois dimensions). Pour les méthodes de tatouage en niveaux de gris, l'espace de quantification est celui des valeurs de niveaux de gris c'est-à-dire de dimension 1. La quantification vectorielle que nous étudions peut être considérée comme une quantification sur un segment de droite engendrée par un vecteur direction (Figure 4.1).

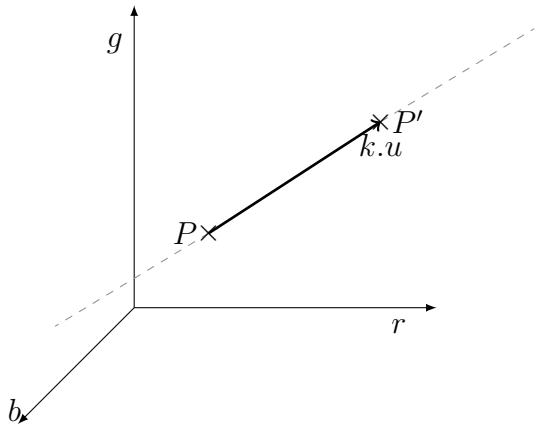


FIGURE 4.1 – Quantification dans l'espace couleur RGB sur une droite (en pointillée) orientée par un vecteur direction u.

Soit P la valeur d'un pixel couleur. Le résultat de la quantification noté P' est défini par :

$$P' = P + \frac{s' - s}{\|u_P\|^2}.u_P \tag{4.1}$$

avec u_P un vecteur direction et $s =< P, u_P >$ le produit scalaire canonique de P par u_P. Cette équation modifie la couleur P en fonction de la modification de s selon l'axe de direction u_P. Selon la couleur P sur laquelle la quantification vectorielle est appliquée, la distorsion couleur perçue par le SVH est différente.

À l'étape de détection, la valeur à estimer est s'. Si la couleur est modifiée après l'insertion du tatouage, on a :

$$s' =< P'', u_{P''} > \tag{4.2}$$

avec P'' une couleur modifiée après insertion d'information et $u_{P''}$ son vecteur direction associé. Nous avons alors une condition de décodage sur les vecteurs directions : il est nécessaire que les vecteurs direction u_P et $u_{P''}$ soient suffisament proches pour avoir une bonne estimation de s'. Nous noterons le vecteur différence u_e tel que $u_{P''} = u_P + u_e$.

L'erreur maximale autorisée sur s' dépends de la méthode d'insertion qui modifie cette valeur. Cette erreur s'exprime avec l'expression suivante :

$$|s'' - s'| = | < P'', u_{P''} > - < P, u_{P'} > | = | < P'', u_P > + < P'', u_e > -s'| \tag{4.3}$$

En supposant que nous sommes dans un cas où la détection est possible c'est-à-dire que la différence de couleur $P'' - P'$ est suffisament petite alors on a :

$$< P'', u_{P''} > \simeq < P', u_P > = s'$$

et finalement, nous voyons que la distance $|s'' - s'|$ dépends du vecteur différence u_e avec :

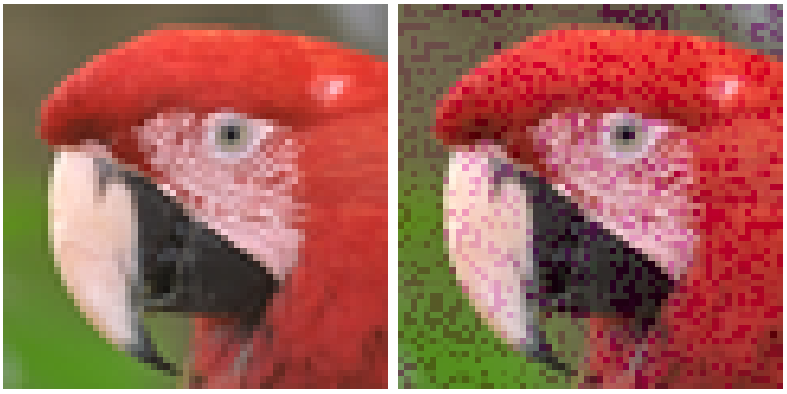
$$|s'' - s'| \simeq | < P', u_P > + < P'', u_e > -s'| = | < P'', u_e > | \tag{4.4}$$

D'après l'équation 4.4, nous pouvons alors voir que l'erreur entre l'estimation s'' et s' dépend bien sûr de l'erreur sur la direction u_e et de la couleur modifié P''.

4.2.2 Choix d'un vecteur direction

L'objectif est de trouver une direction qui minimise le bruit de quantification couleur pour le SVH. Pour commencer, nous avons étudié une approche élémentaire qui consiste à choisir une direction fixée pour toute l'image selon laquelle l'information est insérée.

Si nous choisissons aléatoirement un vecteur direction constant pour toute couleur de l'image, nous percevons des distorsions couleur très facilement à l'oeil nu (figure 4.2b) car ce choix de direction introduit des couleurs qui ne sont pas pertinentes au contenu de l'image d'une couleur à une autre. L'image est donc modifiée par un bruit en couleur visible par l'oeil humain.



(a) Image source (b) Direction aléatoire constante

(c) Direction u_g constante (d) Direction optimale adaptée pour chaque couleur

FIGURE 4.2 – Exemple d'insertion d'une marque avec différentes approaches et vecteurs direction à distorsion numérique équivalente. L'image utilisée est une version recoupée de l'image kodim23.png *de la base d'image Kodak.*

Cependant, nous pouvons réduire le bruit de quantification perçu. En utilisant une direction fixée pour toute l'image, nous avons noté expérimentalement que le vecteur direction engendrant le moins de distorsion couleur est le vecteur $u_g = (1, 1, 1)$ qui représente l'axe des niveaux de gris dans l'espace RGB. Un exemple est illustré dans la sous-figure 4.2c. En effet, comparée à la sous-figure 4.2b, pour un même niveau de modification, nous ne voyons plus de fausses couleurs apparaître. Si maintenant nous sommes capable de choisir un vecteur direction différent et adapté à chaque couleur pour réduire la distorsion d'insertion, nous obtenons un résultat bien meilleur en terme d'invisibilité (sous-figure 4.2d).

Ces premiers résultats ne sont pas satisfaisants en terme d'invisibilité de tatouage. Puisque selon la couleur que nous modifions, la perception du bruit de quantification est différent, nous devons introduire une seconde approche dite *adaptative*. Pour chaque couleur P, nous

devons déterminer quel choix de vecteur direction minimise le bruit de quantification au sens du SVH.

La figure 4.2d montre justement un exemple d'image tatouée avec cette approche adaptative que nous allons introduire par la suite. Nous pouvons observer que le bruit de quantification est beaucoup moins visible pour le SVH. En regardant de plus près, nous observons un léger bruit colorée dans les zones homogènes (ou non texturée) de l'image tel que l'arrière-plan vert et marron.

Dans la section suivante, nous proposons de discuter un modèle psychovisuel qui rend cette approche adaptative possible, i.e., nous montrons comment déterminer les vecteurs direction optimaux pour toute couleur RGB.

4.3 Approche psychovisuelle du SVH

4.3.1 Neurogéométrie et perception

Dans le contexte d'une application au tatouage numérique, il s'agit de comprendre la perception des différences de couleur pour minimiser les distorsions psychovisuelles provoquées par l'insertion d'une marque dans une image couleur. Pour cela, nous avons besoin d'un modèle de la vision trichromatique humaine c'est-à-dire un modèle prennant en compte l'information de la couleur en fonction des trois photorécepteurs du SVH.

Dans sa forme la plus ancienne, la neurogéométrie utilise l'information achromatique pour évaluer l'estimation des formes. Ce traitement de l'information s'effectue dans le cortex visuel primaire V1. Nous pourrons nous référer à [123, 124] pour des revues détaillées sur la neurogéométrie. Quelques années auparavant, d'autres travaux ont été proposé sur le comportement des photorécepteurs de la rétine humaine tels que [125, 126].

Dans ce chapitre, nous nous limitons à la modélisation et à l'application de ce modèle pour le tatouage numérique des images en couleur en utilisant les travaux de Alleysson et Hérault sur la perception de la vision en couleur [127, 128].

Plus précisément, nous utilisons un modèle d'ellipsoïde basé sur la loi de Naka-Rushton [121] qui est la loi de la cinétique des photorécepteurs. À notre connaissance, ce modèle est le plus

récent de la littérature.

4.3.2 Modèle des photorécepteurs et modèle trichromatique

À partir de la loi de la cinétique des photorécepteurs, nous présentons une construction en trois dimensions d'un modèle trichromatique du SVH présenté dans [121]. Le SVH est un système complexe qui s'adapte en fonction de ses conditions environnementales ce qui justifie en partie que la vision en couleur soit un phénomène non-linéaire.

4.3.2.1 Photorécepteurs

Dans la rétine, les réponses électriques à différents flashs lumineux de chaque cône ont été mesuré (chapitre 4 de [129]). À partir de ces mesures, l'état d'adaptation peut être modélisé par la loi de Naka-Rushton (cinétique des photorécepteurs). D'après [128], nous avons :

$$x = \alpha_X \frac{X}{X + X_0} \tag{4.5}$$

avec x le niveau de transduction c'est-à-dire x représente la réponse électriques du cône en fonction de X le niveau d'excitation du cône produit par la lumière et X_0 l'état d'adaptation. X_0 est modifié en fonction du niveau d'excitation moyen du photorécepteur.

L'équation 4.5 représente le comportement d'un photorécepteur en fonction de deux constantes α_X et X_0 fixées par des conditions environnementales et par l'état du SVH par exemple. En fait, le processus qui permet au photorécepteur de s'adapter n'est à ce jour que partiellement connu.

La rétine humaine possède trois types de photorécepteurs qui sont les cônes L, M et S chacun paramétrés par deux constantes tel que :

$$\begin{cases} l = \alpha_L \dfrac{L}{L + L_0} \\ m = \alpha_M \dfrac{M}{M + M_0} \\ s = \alpha_S \dfrac{S}{S + S_0} \end{cases} \tag{4.6}$$

où les paramètres α_L, α_M et α_S sont des gains sur les composantes L, M et S respectivement. Les constantes L_0, M_0 et S_0 sont les états d'adaptation des photorécepteurs.

Grâce aux cônes L, M et S, la perception des couleurs est possible. L'ensemble des triplets (l, m, s) appartiennent à un espace couleur psychophysique appelé espace de transduction. De même pour les triplets (L, M, S), ils vivent dans un espace couleur appelé espace d'excitation (des photorécepteurs).

4.3.2.2 Vision trichromatique

Le modèle de la vision trichromatique proposée par Alleysson permet de construire une représentation en trois dimensions de la perception de la couleur à l'aide de la modélisation des cônes L, M et S. Dans l'espace de transduction lms, deux paires de points de cet espace séparées par la même distance possède un même niveau de perception de différence de couleur.

Lorsque ces paires de points sont converties dans l'espace d'excitation LMS, les distances euclidiennes pour chaque paire de point sont différentes même si le même niveau de perception soit conservé.

Ce phénomène de distorsion peut être mieux observé en construisant une sphère $\mathcal{S}$ de centre $P_{lms} = (l_c, m_c, s_c)$ dans l'espace lms d'équation :

$$\begin{cases} l = r\cos(u)\cos(v) + l_c \\ m = r\sin(u)\cos(v) + m_c \\ s = r\sin(v) + s_c \end{cases} \tag{4.7}$$

avec r le rayon, $-\pi \leq u \leq \pi$ et $-\pi/2 \leq v \leq \pi/2$. En convertissant $\mathcal{S}$ dans l'espace LMS, nous avons le volume $\mathcal{V}$ d'équation :

$$\begin{cases} L = \dfrac{(r\cos(u)\cos(v) + l_c)L_0}{\alpha_L - r\cos(u)\cos(v) - l_c} \\ M = \dfrac{(r\sin(u)\cos(v) + m_c)M_0}{\alpha_M - r\sin(u)\cos(v) - m_c} \\ S = \dfrac{(r\sin(v) + s_c)S_0}{\alpha_S - r\sin(v) - s_c} \end{cases} \tag{4.8}$$

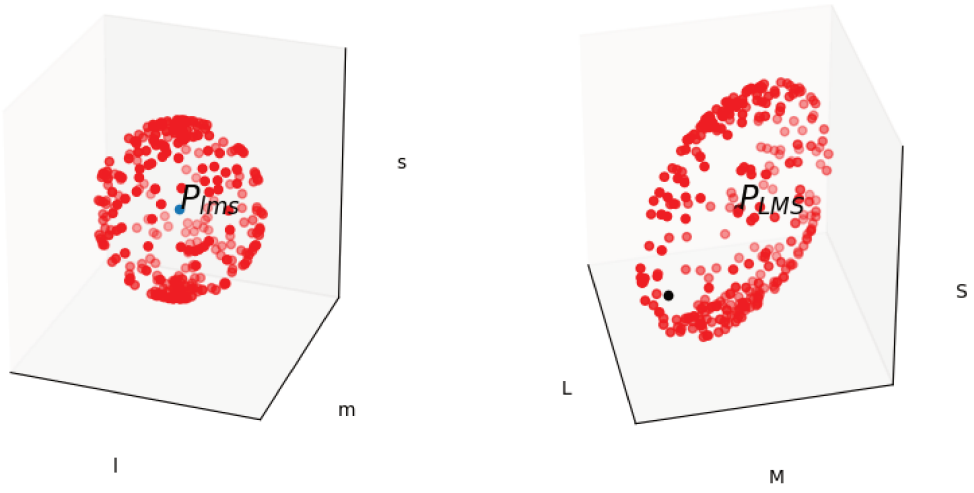


FIGURE 4.3 – Exemple d'une sphère dans l'espace lms à gauche et le volume résultat de la conversion dans l'espace LMS.

Nous pouvons visualiser en trois dimensions la sphère $\mathcal{S}$ (figure 4.3 à gauche) grâce à l'équation 4.7 en générant aléatoirement des points sur celle-ci. Puis, en convertissant les points obtenus avec l'équation 4.8, nous pouvons aussi visualiser le volume $\mathcal{V}$ dans la figure 4.3 à droite. Si nous étudions la distance entre le centre du volume et d'un point de sa surface, alors pour la sphère $\mathcal{S}$, nous avons bien sûr une distance euclidienne constante pour chaque paire de point c'est-à-dire :

$$\|P - P_{lms}\|_2 = r, \forall P \in \mathcal{S} \tag{4.9}$$

avec P_{lms} le centre du volume, pour un même niveau de perception alors que pour le volume $\mathcal{V}$ à droite, ce n'est plus le cas car nous pouvons observer que le volume $\mathcal{V}$ ressemble à un ellipsoïde dont le centre est P_{LMS}[1].

Dans un contexte d'application au tatouage numérique (que l'on souhaite invisible à l'oeil nu), nous parlerons de distorsion psychovisuelle pour désigner le niveau de perception de différence de couleur que nous mettons en opposition avec la distorsion numérique (euclidienne) associée avec le bruit d'insertion d'une marque dans une image. L'intérêt de cette modélisation est de pouvoir augmenter les distorsions numériques c'est-à-dire ajouter de la robustesse à une marque tout en préservant le même niveau de distorsion psychovisuelle. Nous allons alors chercher la paire de point du volume $\mathcal{V}$ qui possède la plus grande distance euclidienne.

Cependant, il reste une dernière étape avant de pouvoir utiliser cette modélisation de la vi-

1. conversion du point P_{lms} dans l'espace LMS

sion en couleur. Les constantes et les gains doivent être paramétrés afin de pouvoir représenter l'état d'un SVH moyen.

4.3.3 Calibration du modèle

Afin d'appliquer le modèle décrit précédemment au tatouage numérique, les constantes L_0, M_0 et S_0 ainsi que les gains α_L, α_M et α_S peuvent être paramétrés grâce aux ellipses de Macadam (figure 4.4).

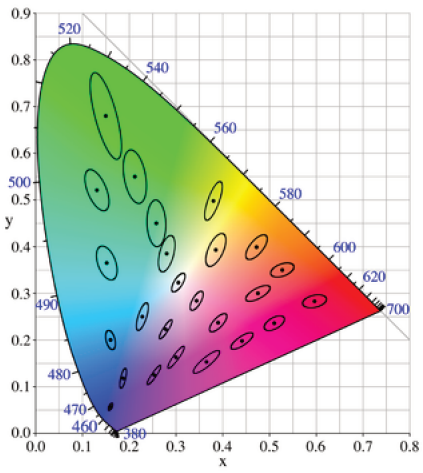


FIGURE 4.4 – Ellipses de MacAdam dans un plan de luminance de l'espace couleur xyY de 1931. Les ellipses sont agrandies 10 *fois.*

À l'origine celles-ci ont été calculés pour estimer les erreurs de mesures produites lors d'expériences d'égalisation de couleur. Pour une couleur donnée, MacAdam a mesuré la couleur proche dont la différence est juste discernable (JND en anglais pour Just Noticeable Difference).

Pour que notre modèle de la vision en couleur puisse avoir une sensibilité proche de celle d'un humain, 25 ellipses ont été reproduites grâce à la conversion d'ellipsoïdes de notre modèle dans le même plan de luminance que les ellipses de MacAdam. Puis, les paramètres des ellipsoïdes ont été ajusté pour minimiser les différences avec les ellipses de MacAdam. Ce procédé

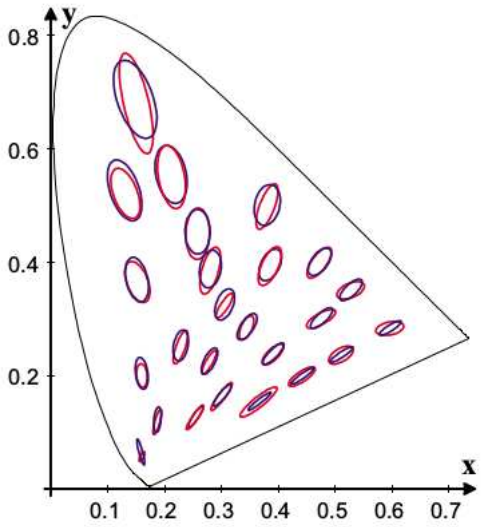


FIGURE 4.5 – Ellipses du modèle psychovisuel quasiment confondues avec celles de MacAdam après un choix optimal des gains et des constantes calculées par Alleysson.

est décrit dans le chapitre 4 de [129] et permet d'obtenir les valeurs numériques suivantes :

$$\begin{cases} \alpha_L = 1665 \\ \alpha_M = 1665 \\ \alpha_S = 226 \end{cases} \tag{4.10}$$

et

$$\begin{cases} L_0 = 66 \\ M_0 = 33 \\ S_0 = 0.16 \end{cases} \tag{4.11}$$

Les travaux sur les ellipses de MacAdam [130] sont étendus d'un plan de luminance fixé vers un espace couleur perceptuel en trois dimensions. À notre connaissance, il n'existe pas d'autres extensions 3D de ces travaux. Pour chaque couleur de l'espace RGB, il devient possible d'évaluer le niveau de perception de différence de couleur dans toutes les directions de l'espace. Pour un pixel couleur donné, nous pouvons donc maintenant choisir une direction qui maximise la distortion d'insertion d'une marque avec une distorsion psychovisuelle fixée. Dans la suite, nous expliquons comment ce modèle est appliqué au tatouage numérique des images en couleur.

4.4 Application du modèle au tatouage numérique

Dans la section précédente, nous avons rappelé comment modéliser la vision en couleur du SVH dans le but de développer une méthode de tatouage prennant en compte la perception des différences de couleur du SVH. Dans l'espace couleur LMS, nous avons construit des ellipsoïdes représentant la notion de distorsion psychovisuelle. Avec l'objectif d'améliorer la robustesse d'une marque, il est possible de maximiser les distorsions numériques tout en conservant un même niveau de distorsion psychovisuelle.

En d'autres termes, nous proposons dans cette section d'extraire pour chaque pixel couleur un vecteur direction dont la norme est la plus grande dans une ellipsoïde donnée donnant ainsi la direction dans la cube RGB de modification maximale pour un niveau de dégradation fixé.

4.4.1 Conversions et calcul des ellipsoïdes

Pour travailler dans l'espace couleur RGB. Des changements d'espaces couleur sont nécessaires pour convertir des ellipsoïdes de l'espace LMS vers l'espace RGB. Nous avons choisis ces transformations linéaires suivant la norme standart CIE de 1931 [2]. Les conversions entre espaces couleur sont représentés par les matrices suivantes :

$$M_1 = M_{RGB\to XYZ} = \begin{pmatrix} 0.4887180 & 0.3106803 & 0.2006017 \\ 0.1762044 & 0.8129847 & 0.0108109 \\ 0.0000000 & 0.0102048 & 0.9897952 \end{pmatrix}$$

$$M_2 = M_{XYZ\to LMS} = \begin{pmatrix} 0.38971 & 0.68898 & -0.07868 \\ -0.22981 & 1.18340 & 0.04641 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}$$

Soient P_{RGB}, P_{XYZ} et P_{LMS} les représentations d'un pixel couleur dans les espaces RGB,

2. Le blanc de référence choisi est le point blanc d'énergie égal à E.

XYZ et LMS respectivement. On a les égalités de conversions :

$$\begin{aligned} P_{LMS} &= M_2 P_{XYZ} \\ P_{XYZ} &= M_1 P_{RGB} \\ P_{LMS} &= M_2 M_1 P_{RGB} \end{aligned} \tag{4.12}$$

Les propriétés de distorsions discutées plus tôt dans ce chapitre peuvent donc être transférées à travers les espaces couleur grâce aux formules de conversion. Dans la partie suivante, nous détaillons le procédé d'extraction d'un vecteur direction pour un pixel couleur donné.

4.4.2 Extraction des vecteurs direction

Dans la section 4.2, nous avons vu que le choix d'une direction avait un impact non négligeable sur l'invisibilité d'une marque. Lorsque l'on choisi une direction fixée pour toutes les couleurs à quantifier (avec distorsions numériques fixées), nous pouvions observer des distorsions (psychovisuelle) sur l'image marquée sans alors qu'en choisissant une direction adaptée pour chaque pixel couleur à modifier, l'invisibilité de la marque est améliorée de manière significative.

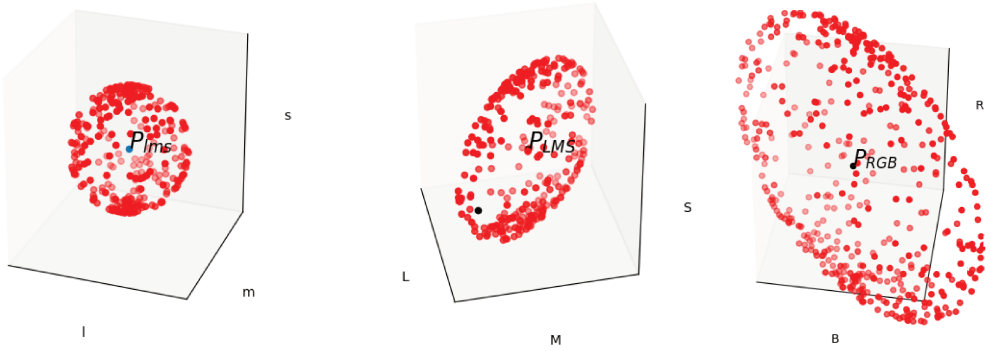


FIGURE 4.6 – De la gauche vers la droite. Représentations d'une sphère dans l'espace lms, d'un ellipsoïde issu de la conversion de la sphère lms dans l'espace LMS et de l'ellipsoïde correspondante dans l'espace couleur RGB.

Pour extraire un vecteur direction optimal, nous devons construire l'ellipsoïde associée à la couleur P considérée. Après avoir converti P dans l'espace lms, la sphère de centre P_{lms} est convertie dans l'espace RGB (exemple de volumes illustré dans la figure 4.6). Nous avons choisi le rayon r de la sphère lms assez petit ($r = 10^{-2}$) pour éviter que les asymptôtes ne

déforment de manière incohérente les volumes calculés auquel cas nous n'aurions plus de volume de forme proche de celle d'un ellipsoïde.

Puisque chaque point de la surface de l'ellipsoïde possède le même niveau de distorsion psychovisuelle avec P, nous pouvons choisir le point le plus éloigné de P noté P_f afin d'assurer un maximum de robustesse à l'insertion d'une marque. Le vecteur direction u_P associé à P se caractérise par :

$$u_P = \overrightarrow{PP_f} \tag{4.13}$$

Notons $\mathcal{E}$ l'ensemble des points d'un volume de perception associé à P. Le point le plus éloigné de P s'exprime grâce à l'équation suivante :

$$P_f = \max_{P' \in \mathcal{E}} \|P - P'\|_2 \tag{4.14}$$

Nous n'avons pas de formulation analytique du volume, cependant nous pouvons proposer une méthode numérique pour le calcul du vecteur direction optimal associé à P par recherche exhaustive. Cependant, il est possible d'être plus efficace

En appliquant une analyse en composante principale sur les éléments de $\mathcal{E}$, nous pouvons extraire des vecteurs *d'inertie*. Plus précisément, nous pouvons calculer la matrice de covariance associé aux points de $\mathcal{E}$ puis déterminons les vecteurs propres qui lui sont associés. Le vecteur possédant la plus grande norme est le vecteur direction u_P recherché.

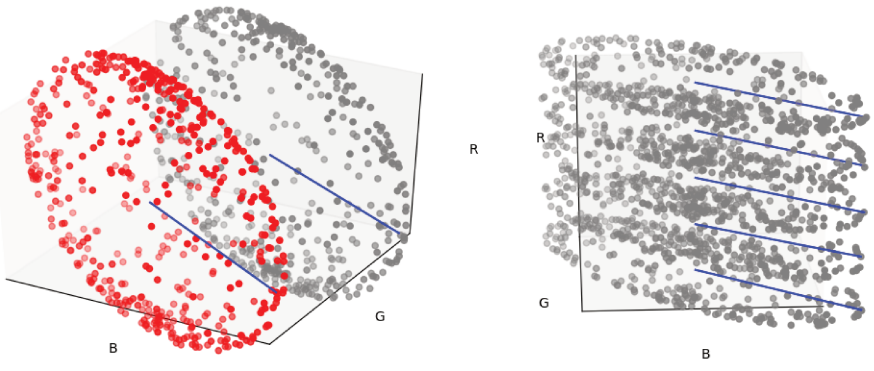


FIGURE 4.7—Illustrations de la stabilité des vecteurs direction dans l'espace RGB. Plusieurs ellipsoïdes de perception sont représentées côte à côte avec leur vecteur direction respectif. Nous pouvons observer que ceux-ci ne varient que très peu en terme de direction.

Enfin, pour mémoire, nous devons nous assurer qu'après modification d'une image marquée, la détection de la marque soit possible si les dommages subies par l'image sont raisonnables. Autrement dit, les vecteurs direction à l'insertion et à la détection d'une marque doivent être suffisament proches pour assurer l'intégrité du message inséré. Nous en fournissons une première "preuve" sur la figure 4.7 avec l'illustration des directions pour des couleurs "proches".

4.5 Algorithme psychovisuel pour les images couleur

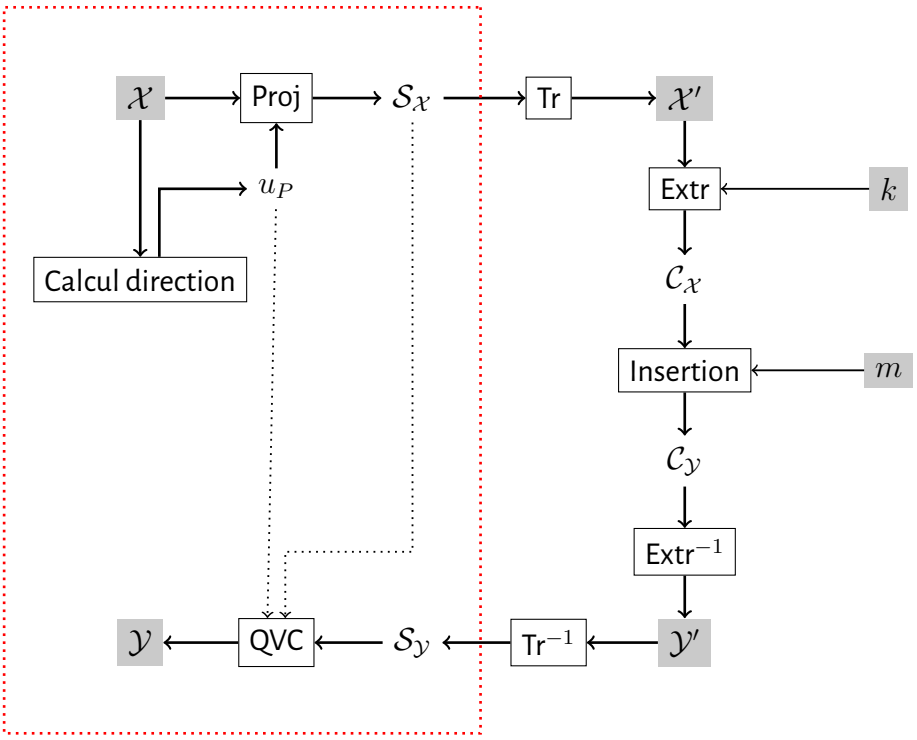


FIGURE 4.8 – Schéma d'insertion classique combiné avec une quantification vectorielle (QVC) couleur basée sur un modèle psychovisuel. Les éléments encadrés en rouge représente les étapes de la quantification vectorielle. Tr et Extr sont les fonctions de tranformation d'espace et d'extraction des coefficients, k est la clé secrète et m le message binaire.

L'algorithme que nous présentons dans cette section permet d'insérer une marque dans une image couleur dont la qualité est améliorée d'un point de vue psychovisuel. Par rapport à un scénario classique d'insertion illustré en début de ce manuscrit, nous proposons d'ajouter une étape à ce scénario pour adapter une méthode de tatouage des images en niveau de gris vers des images en couleur (illustré dans la figure 4.8).

Avant de transformer l'image hôte dans l'espace de quantification choisi, pour chaque pixel couleur de l'image $\mathcal{X}$, nous calculons un vecteur direction. Ce vecteur direction va être défini à partir de l'axe optimal tel qu'il est défini dans la section précédente. Nous introduisons alors le produit scalaire s (étape Proj de la figure 4.8) :

$$s =< P, u_P > \tag{4.15}$$

Nous obtenons alors une image en niveau de gris $\mathcal{S}_\mathcal{X}$. En considérant la connaissance de l'ensemble des vecteurs direction u_P calculés, nous avons une correspondance bijective entre $\mathcal{X}$ et $\mathcal{S}_\mathcal{X}$. Sur la figure 4.9, nous pouvons voir que l'image $\mathcal{S}_\mathcal{X}$ est une version en niveau de gris sombre de l'image couleur $\mathcal{X}$.

L'étape suivante consiste alors à changer la représentation de $\mathcal{S}_\mathcal{X}$. Une première représentation pour insérer une marque est le domaine spatial. Il est aussi possible de représenter une image autrement. Par exemple, il existe différentes transformées réversibles telles que la transformée en cosinus discret (DCT) et la transformée en coefficients d'ondelette (DWT). Ces différentes représentations (noté Tr et Tr^{-1} pour l'opération inverse) de l'image permettent d'obtenir des propriétés utiles selon les besoins telles que la décomposition en bande de fréquences indépendantes de l'image.

Ensuite, une fonction d'extraction Extr doit être choisie pour sélectionner les coefficients à modifier. Dans ce chapitre, nous avons choisi une fonction qui détermine aléatoirement les sites d'insertion de l'image $\mathcal{S}_\mathcal{X}$. Une autre méthode de sélection des coefficients a été présenté dans le chapitre précédent qui consiste à décomposer une image en blocs puis de sélectionner aléatoirement des coefficients dans chaque bloc. Le résultat de cette extraction se passe dans l'espace de tatouage qui est un espace secret (connu uniquement par l'auteur de la marque et son destinataire). L'accès à cet espace peut être sécurisé grâce à une clé secrète k.

Une fois que les coefficients $\mathcal{C}_\mathcal{X}$ sont sélectionnés, ils sont modifiés par la méthode d'insertion (méthode LQIM mais il est aussi possible d'utiliser d'autres méthodes de tatouage) en fonction du message m puis intégré dans la représentation de l'image utilisée à l'extraction et nous obtenons l'image marquée $\mathcal{Y}'$. Ainsi, pour toute couleur modifiée P' (étape QVC), nous avons :

$$P' = P + \frac{(s' - s)}{\|u_P\|^2} u_P \tag{4.16}$$

avec s' le scalaire modifié par la méthode d'insertion choisie. L'ensemble des s' forme l'image en niveau de gris $\mathcal{S}_\mathcal{Y}$. Notons bien sûr que l'utilisation de différents codes tel que nous l'avons

FIGURE 4.9 – Paires d'images (image hôte $\mathcal{X}$, image scalaire associée $\mathcal{S}_{\mathcal{X}}$). Images aléatoires de la base Corel.

discuté dans les chapitres précédents est tout à fait possible et direct. Nous nous concentrons içi uniquement sur l'aspect "contrôle de l'invisbilité" et son optimisation par rapport à la robus-

tesse.

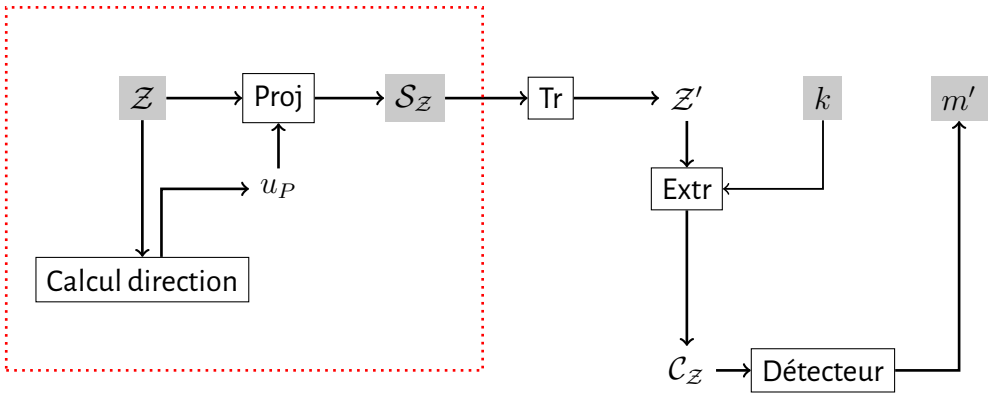


FIGURE 4.10 – Schéma classique de détection d'une marque. Comme à l'insertion, nous retrouvons la partie d'extraction de l'image scalaire $\mathcal{S}_{\mathcal{Z}}$ encadrée en rouge.

À l'étape de détection (figure 4.10), nous déterminons l'image en niveaux de gris $\mathcal{S}_{\mathcal{Z}}$ en recalculant les vecteurs direction associés à chaque couleur de l'image reçue $\mathcal{Z}$ (équation 4.15) puis accédons au canal du tatouage grâce à la fonction d'extraction Extr et la clé k.

Ensuite, nous appliquons aux coefficents $\mathcal{C}_{\mathcal{Z}}$ le détecteur associée à la méthode d'insertion choisie. Si la puissance de l'attaque est raisonnable et les paramètres de robustesse de la marque sont bien choisis, l'estimation m' doit être identique à m.

Comme nous l'avons dit, pour la détection le calcul des vecteurs direction ets une étape importante. En admettant que les couleurs soient raisonnablement modifiés, la capacité à détecter correctement le message réside dans la variation du vecteur direction c'est-à-dire que le vecteur d'erreur u_e doit être proche du vecteur nul. Nous verrons plus tard dans nos expériences que les variations des vecteurs direction sont acceptables et permettent d'assurer une bonne détection.

Nous avons détaillé un algorithme de tatouage classique pour les images couleur utilisant une méthode de quantification vectorielle couleur basée sur un modèle psychovisuel du SVH. La méthode LQIM a été choisie pour une adaptation au tatouage des images couleur (introduite dans le premier chapitre). Pour pouvoir l'utiliser, nous adaptons le schéma classique d'insertion avec notre modèle de quantification psychovisuelle 4.8 puis intégrons dans la fonction Insertion le quantificateur Q_m de la méthode LQIM. Les sites d'insertion sont choisis aléatoirement parmis les coefficients disponibles à l'insertion. Un des avantages de cette stratégie d'insertion est d'améliorer l'invisibilité de la marque surtout dans les zones de l'image qui sont

texturées. À l'étape de détection, nous combinons également l'étape d'extraction des coefficients reçus $\mathcal{C}_{\mathcal{Z}}$(figure 4.10) avec le schéma classique de décodage puis utilisons le détecteur LQIM.

La section suivante donne une évaluation des performances de robustesse de la méthode CLQIM face à diverses modifications d'image.

4.6 Validation expérimentale

Les images utilisées pour les expériences appartiennent à la base Corel (1000 images aléatoires sélectionnées parmis les 10000 disponibles). Les sites d'insertion sont déterminés aléatoirement. Pour l'évaluation de l'invisibilité, des paramètres tels que la taille du message ou la dimension du réseau euclidien L de la méthode Lattice QIM sont calculés pour obtenir une qualité d'image et un taux d'insertion adéquats selon l'expérience d'invisibilité ou de robustesse.

4.6.1 Invisibilité psychovisuelle

Dans cette sous-section, nous proposons différents ensembles d'images marquées afin d'apprécier l'amélioration en invisibilité psychovisuelle par rapport à des images marquées issues une quantification vectorielle de vecteur direction constant.

Nous désignons alors par GA et AA les deux approches suivantes :

— la version couleur avec un axe de direction constant $u = (1, 1, 1)$ (GA). Nous avons choisi ce vecteur direction car choisir l'axe de luminance est selon nous un compromis satisfaisant pour garantir un bon niveau d'invisibilité d'une marque.

— la version couleur avec un axe de direction adaptatif u_P (AA)

Ces deux méthodes de tatouage sont les adaptations couleur de la méthode LQIM présentée précédemment dans ce chapitre. Pour chaque approche, les vecteurs direction sont de même norme (arbitrairement fixée à 0.5). Pour un même niveau de distorsion numérique, il s'agit de valider expérimentalement que l'approche AA insert bien une marque psychovisuellement plus invisible que l'approche GA.

Dans ces expériences, nous évaluons la distorsion numérique grâce au rapport signal sur bruit caractérisant le bruit de quantification de la méthode LQIM noté DWR.

FIGURE 4.11 – Images couleur recadrées (Lenna et base Kodak de taille 60×60) marquées avec la méthode LQIM (approche GA), DWR $\simeq -5.5$dB en moyenne et ER $= 0.5$.

Dans la figure 4.11, nous montrons des exemples d'images en couleur marquées avec l'approche GA de la méthode LQIM. Pour chaque image de cette figure, nous voyons facilement que les couleurs saturent vers le gris. Par rapport au pixels voisins, il devient donc plus facile de percevoir des différences de couleur à l'oeil nu. En effet, une piste d'explication possible est que l'aspect visuel du bruit de quantification est celui d'un bruit poivre et sel et ajoute donc de la texture à l'image. L'attention du SVH va donc se concentrer sur celle-ci par effet de saillance visuelle.

Avec l'approche AA (figure 4.12), le bruit de quantification a pour effet de saturer les couleurs vers le bleu et le vert. Celui-ci s'adapte en fonction de la couleur modifiée. Il est donc plus difficile à percevoir. Par rapport aux pixels voisins dans des zones de couleurs homogènes, nous constatons que détecter une différence de couleur est difficile.

À distorsion numérique égale, une image marquée avec les approches GA et AA ne sont pas affectées avec le même bruit de quantification d'un point de vue psychovisuel. La perception du bruit de quantification avec l'approche AA est bien plus faible pour le SVH comparée à celle de l'approche AA. Nous pouvons le voir sur les images des figures 4.11 et 4.12 mais ces résultats se sont aussi confirmés sur d'autres tests.

Pour pouvoir mieux observer le bruit d'insertion, nous avons choisi des images de taille

FIGURE 4.12 – Images couleur recadrées (Lenna et base Kodak de taille 60×60) marquées avec la méthode LQIM (approche AA), DWR $\simeq -5.5$dB en moyenne et ER $= 0.5$.

60×60 et avons choisi un fort taux d'insertion (ER $= 0.5$). En pratique, les images sont de taille bien plus grande (par exemple 1080×1920 pour de la haute définition) et le taux d'insertion est plus faible ce qui rend le bruit d'insertion beaucoup moins visible à l'oeil nu.

Approches	Approche GA	Approche AA
Votes moyens	$4\% \pm 3\%$	$96\% \pm 3\%$

TABLE 4.1 – Expériences psychovisuelles de comparaisons d'images marquées. Chaque personne devait décider quelle image était plus dégradée que l'autre (une image tatouée avec l'approche constante et l'autre avec l'approche adaptative). Les paramètres sont DWR = 20dB, ER = $1/2$ pour chaque image. Ce tableau montre le pourcentage d'image noté comme moins dégradée pour chaque approche.

Dans un second temps, nous avons répété l'expérience de comparaison des approches GA et AA décrite précédemment avec différents observateurs (au nombre de 15) et avec la base d'image Kodak composée de 24 éléments de taille 768×512. Notons que ces observateurs ont été pris au sein du laboratoire. Pour chaque image de cette base, nous avons proposé une paire d'images marquée avec nos deux approches couleur GA et AA (toujours avec la méthode LQIM). Pour les 24 paires d'images, chaque observateur vote pour l'image la 'moins bruitée'.

Les résultats de cette expérience sont donnés dans le tableau 4.1. Parmis ces sujets, seulement 4% des images ont été décrites comme 'moins bruitées' en moyenne avec l'approche GA. Basé sur ces résultats, nous concluons que l'approche AA permet d'obtenir une bien meilleure invisibilité psychovisuelle comparée à l'approche GA.

L'amélioration en invisibilité psychovisuelle de l'approche AA nous permet maintenant de

comparer les deux approches en terme de performances de robustesse face à plusieurs modification d'image.

4.6.2 Robustesse

4.6.2.1 Protocole

Nous proposons deux domaines d'insertion pour un message de taille $n = 128$ bits et $L = 2$. Premièrement, le domaine spatial et puis celui des coefficients d'ondelettes (base d'ondelettes Debauchies 4). Nous avons choisi une décomposition à l'échelle 2 dans le plan diagonal HH pour obtenir de meilleurs performances de détection. Bien que notre manière de modifier les coefficients de l'image soit la même, l'insertion d'information dans cette représentation de l'image est différente car il ne s'agit plus de modifier des coefficients entiers mais réels.

Avant de présenter les performances de robustesse, nous souhaitons expliquer comment comparer les performances des méthodes GA et AA sur un pied d'égalité.

Afin de garantir une qualité d'image et une invisibilité de marque satisfaisante, nous avons déterminé le pas de quantification maximal moyen avant qu'une marque ne se soit discernable pour chaque méthode et dans chaque domaine (voir tableau 4.2).

Les paramètres constants de cette expérience sont $L = 2$ et $n = 128$.

Δ/DWR	GA	AA
SP	$16/29.4$	$24/38.8$
DWT	$45/24.6$	$40/33.4$

TABLE 4.2 – Tableau contenant les niveaux de distorsion d'insertion et les pas de quantification Δ maximaux moyens correspondants à des images dont les marques sont invisibles.

Nous pouvons remarquer que la méthode AA nous permet d'utiliser un pas de quantification plus élévé et donc aussi une distorsion numérique plus élévée par rapport à la méthode GA sans mettre en évidence la marque. Dans le domaine spatial et celui des ondelettes, nous avons un gain en distorsion numérique de $5dB$ et de $5.4dB$ respectivement. La méthode AA permet donc un niveau de robustesse plus élévé par rapport à la méthode GA. Grâce à ces valeurs, nous ajustons le pas de quantification de chaque méthode afin de régler leur compromis

invisibilité/robustesse. Nous notons Δ_{GA} et Δ_{AA} les pas de quantification respectifs des méthodes GA et AA.

Quant aux mesures de performances de robustesse, nous avons mesurés des taux d'erreur binaire moyens (100 répétitions pour le même pas de quantification) en fonction de la force de la modification image appliquée. Les mesures possèdent une précision à ± 0.01 près. Nous proposons maintenant d'analyser la robustesse face à des attaques classiques.

4.6.2.2 Compression JPEG

La compression JPEG est une méthode de compression avec perte de données. Elle permet de réduire la quantité de donnée nécessaire pour encoder une image en fonction d'un facteur qualité. Plus celui-ci est faible, plus la qualité de l'image en question diminue et inversement. Pour les images en couleur, la compression JPEG commence par supprimer l'information de chrominance.

Dans le domaine spatial, la courbe d'erreur de la méthode LQIM AA reste proche de la valeur 0.5 alors que la courbe LQIM GA commence à tendre vers 0 à partir de $q = 80$ et atteint 0 quand $q = 95$. Notons que le comportement de la courbe LQIM AA est surprenant et reste inexpliqué. En effet, les distorsions provoquées par la compresson JPEG sont très faibles pour $q > 95$ et devrait permettre à la courbe LQIM AA de tendre vers 0.

Le fait que notre méthode psychovisuelle LQIM AA soit basée sur une modification légère de la valeur des pixels par construction, elle ne peut pas être résistante à la compression JPEG dans le domaine spatial. Cette affirmation se confirme par l'observation de la courbe LQIM AA (figure 4.13).

Par contre, dans le domaine des ondelettes, nous observons une chute brutale des taux d'erreur (nul pour $q \geq 55$) pour LQIM GA et LQIM AA par rapport au domaine spatial. En effet, ce résultat n'est pas surprenant car insérer de l'information dans les coefficients d'ondelettes permet à nos méthodes couleur de résister à la compression JPEG car ceux-ci ne sont pas modifiés par cette modification d'image. La courbe LQIM AA est légèrement plus proche de 0 que LQIM GA. Dans le cas de la compression JPEG, nous avons une amélioration de la robustesse avec la méthode AA grâce au compromis invisibilité/robustesse par rapport à la méthode GA dans le domaine des ondelettes.

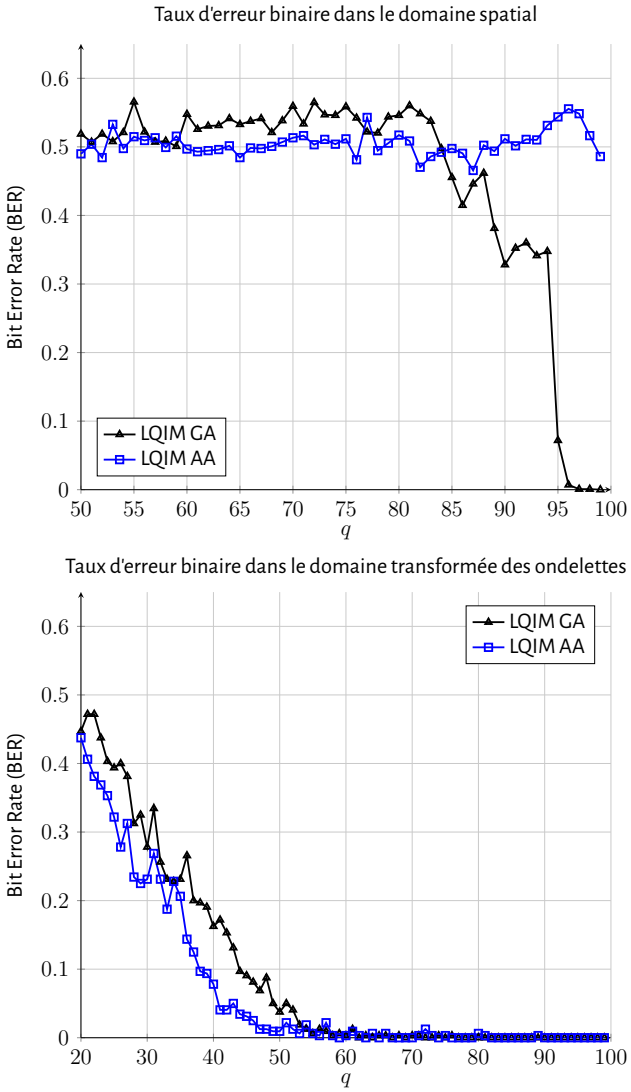


FIGURE 4.13 – Variations des taux d'erreur binaires des méthodes GA et AA en fonction du facteur de qualité q.

4.6.2.3 Modification de contraste

Nous définissons la modification de constraste par un gain multiplicatif de paramètre α. Soit x la valeur d'un pixel. Alors, on a :

$$y = \alpha x \tag{4.17}$$

avec y la version modifiée du pixel x. Plus la distance entre α et 1 est grande, plus les taux d'erreur augmentent. Cette modification d'image dépend du contenu de l'image modifiée.

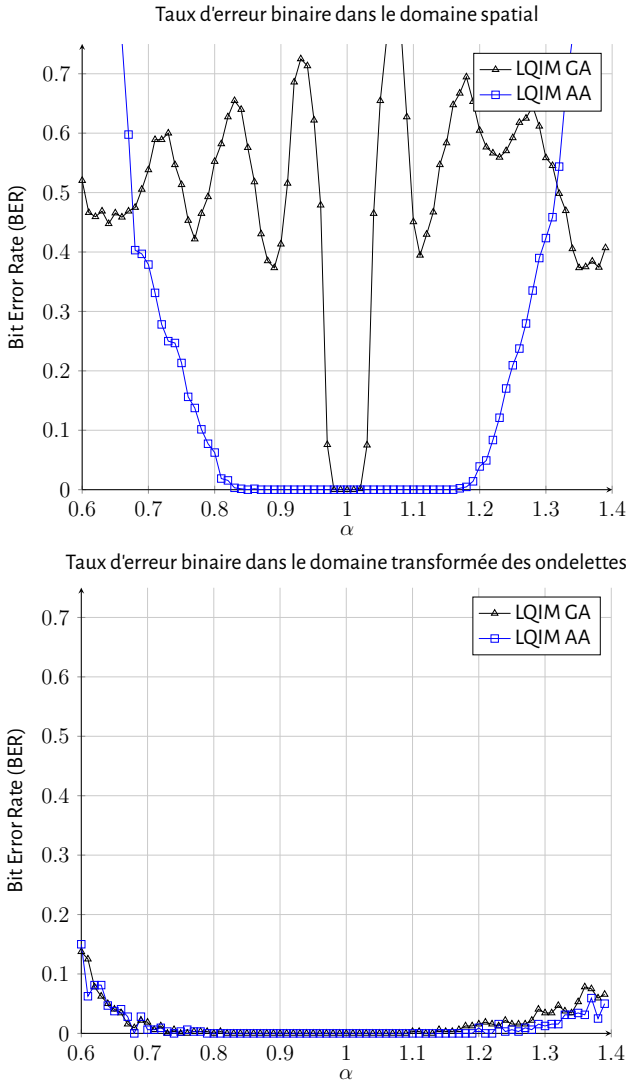


FIGURE 4.14 – Variations des taux d'erreur binaires des méthodes GA et AA en fonction d'un gain multiplicatif α.

Les résultats de robustesse contre cette modification d'image sont présentés dans la figure 4.14. Dans le domaine spatial, les taux d'erreur se comportent de manière attendue. Plus la distance entre α et 1 est grande, plus les taux d'erreur augmentent et oscillent autour de 0.5. Nous remarquons que pour la courbe LQIM AA possède un intervalle à 0 plus grand que celui de la courbe LQIM GA ($\alpha \in \{0.81, \ldots, 1.19\}$ contre $\alpha \in \{0.98, \ldots, 1.02\}$) car $\Delta_{AA} > \Delta_{GA}$.

Dans le domaine transformée des ondelettes, les taux d'erreur LQIM GA et LQIM AA sont quasiment nuls sur l'intervalle $\{0.8, \ldots, 1.2\}$ puis ceux-ci remontent légèrement en dehors de cet intervalle. Les deux méthodes sont plus robustes que dans le domaine spatial.

4.6.2.4 Modification de luminance

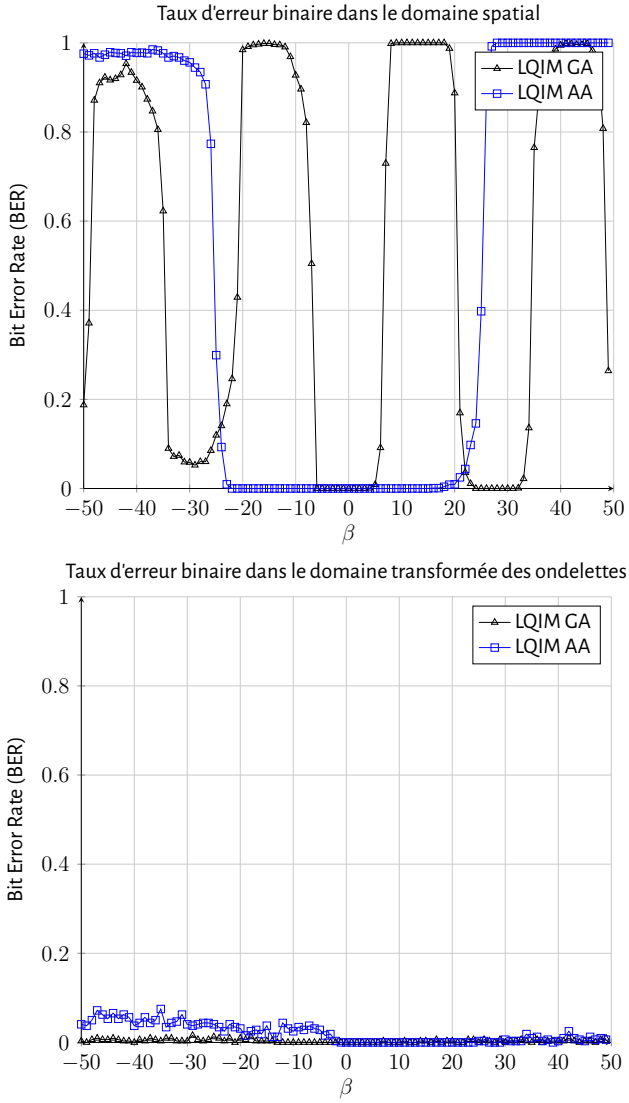


FIGURE 4.15 – Variations des taux d'erreur binaire des méthode GA et AA en fonction d'une constante additive β.

Nous modélisons la modification de luminance par l'équation suivante :

$$y = x + \beta \times (1, \ldots, 1) \tag{4.18}$$

avec y la version modifiée de x la valeur d'un pixel.

Les résultats de robustesse sont présentés dans la figure 4.15. Dans le domaine spatial, les courbes d'erreur oscillent à la manière d'une fonction en escalier entre 0 et 1 de manière pério-

dique. Cette modification est particulière car elle applique des modifications qui ne dépendent pas de l'image. Ce comportement se justifie à la fois par l'utilisation de la méthode LQIM et la structure de l'erreur produite par la modification de luminance (etudié en détails dans le chapitre décrivant le tatouage combiné avec des codes en métrique rang).

Pour la courbe LQIM AA, la période d'oscillation est plus grande que celle de la courbe LQIM GA encore une fois car $\Delta_{AA} > \Delta_{GA}$, ce qui donne un interval sans erreur autour de 0 plus important. Dans le domaine des ondelettes, les courbes d'erreur sont confondues. Les taux d'erreur sont proches de 0 pour $\beta < 0$ puis nuls pour $\beta \geq 0$.

En effet, le domaine transformée des ondelettes représente une image en terme de variations, i.e., en bande de fréquence ce qui explique la résistance des deux méthodes à la modification de luminance. Comme dans la modification d'image précédente, insérer une marque dans le domaine des ondelettes nous permet d'obtenir une meilleure robustesse par rapport au domaine spatial.

4.6.2.5 Bruit additif blanc gaussien

Dans cette partie, Nous nous intéressons au comportement des taux d'erreur face à l'ajout d'un bruit additif blanc gaussien de paramètres $\mu = 0$ et σ variable. Ajouté à une image, le bruit affecte de manière indépendante celle-ci sur chacune des composantes couleur. Dans le domaine spatial comme celui des ondelettes, nous remarquons que les taux d'erreur sont plus faibles pour LQIM AA c'est-à-dire une amélioration de la robustesse de la marque avec la méthode AA par rapport à la méthode GA.

En reprennant les pas de quantification de la méthode AA dans les domaines spatial et des ondelettes, nous voyons de manière surprenante que $\Delta_{AA,SP} > \Delta_{AA,DWR}$, ce qui explique que la méthode AA soit plus robuste dans le domaine spatial. Cette inégalité des pas de quantification est peut être due au fait que le nombre de coefficients dans le domaine spatial est plus grand que le nombre de coefficients ondelette modifiés ce qui produit un bruit de quantification plus visible.

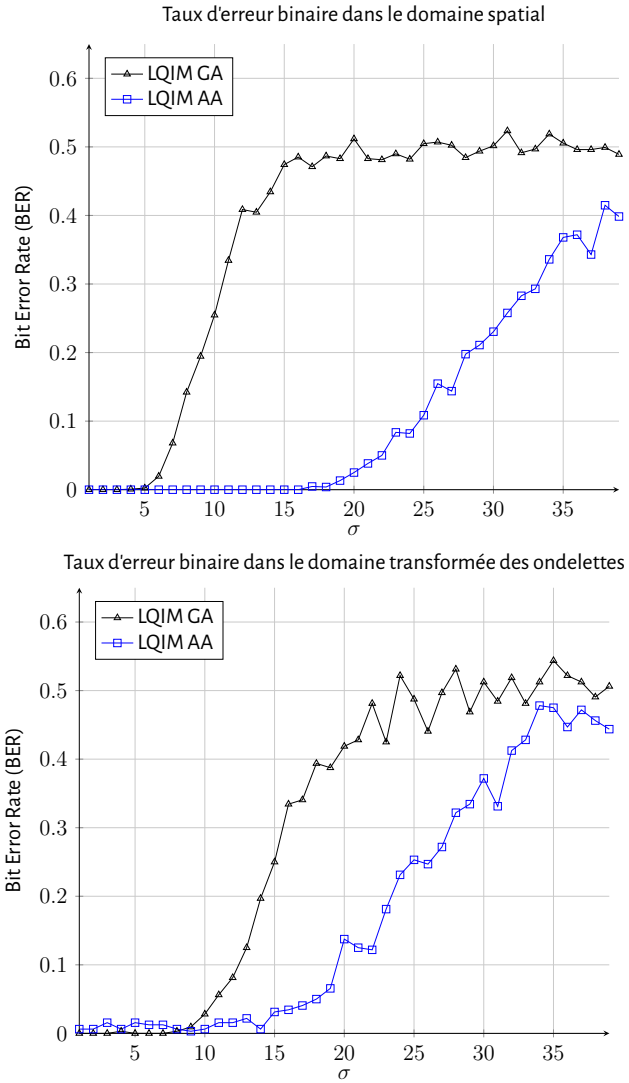


FIGURE 4.16 – Variations des taux d'erreur binaire des méthodes GA et AA en fonction d'un bruit additif blanc gaussien centré de paramètre σ.

4.6.2.6 Modifications de teinte, saturation et luminance

Dans cette sous-section, nous considérons des exemples de modifications d'image couleur. En représentant une image dans l'espace couleur HSV, nous modifions chaque composante couleur h, s et v. Les valeurs utilisées pour les abscisses des différents graphiques de la figure 4.17 dépendent de l'implémentation de la bibliothèque d'imagerie d'OpenCV. C'est pourquoi les composantes couleur d'image représentée dans l'espace HSV ont pour valeur des entiers entre 0 et 255 au lieu d'un paramètre d'angle dans nos expériences.

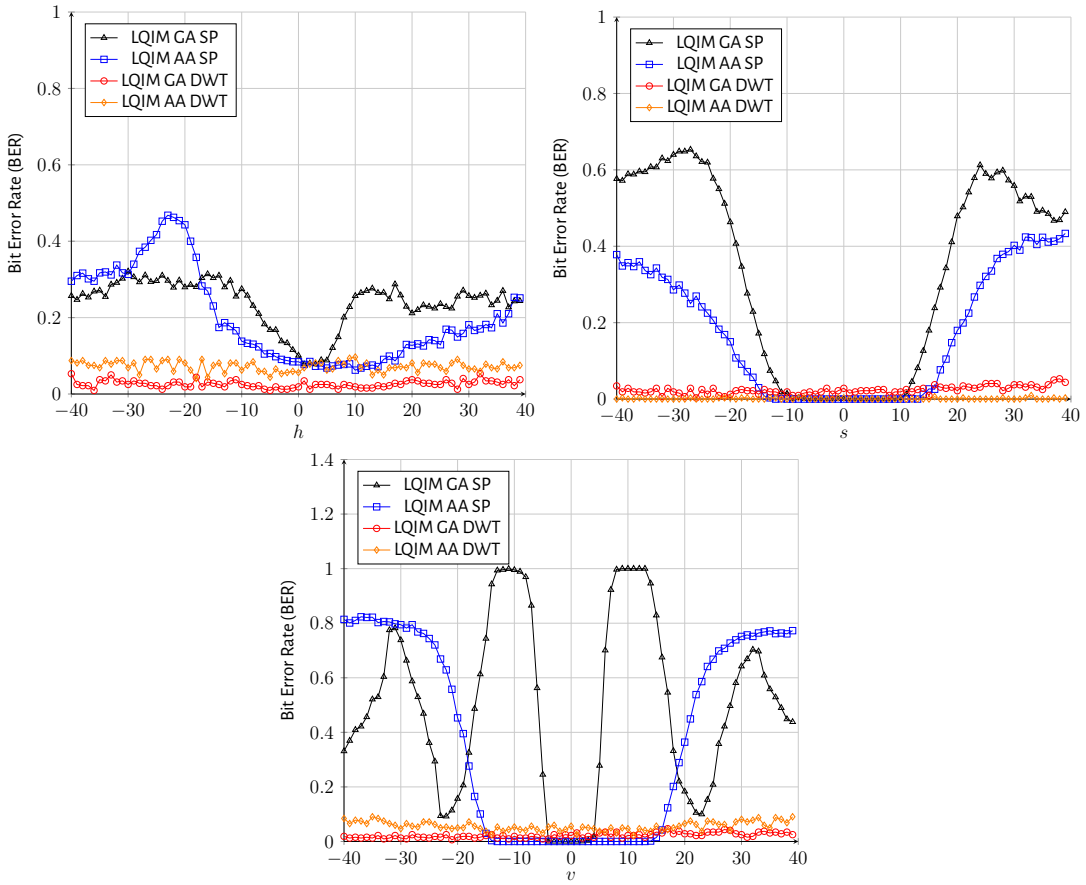


FIGURE 4.17 – Taux d'erreur binaire des méthodes GA et AA dans le domaine spatial et des ondelettes en fonction des variations sur les composantes couleur de l'espace HSV h, s et v.

Dans cette figure, nous observons sur chaque composante que les courbes LQIM AA SP sont plus proches de 0 que les courbes LQIM GA SP. Pour la modification de la composante de luminance v, la courbe LQIM GA SP oscille de manière similaire à l'attaque de modification de luminance étudiée dans le chapitre précédent. Pour les deux composantes, les courbes LQIM SP sont proches de 0 aux alentours de $h = s = 0$. D'autre part, les taux d'erreur des courbes DWT sont proches de 0 sur tout l'intervalle présenté. L'insertion d'information dans le domaine des ondelettes nous permet d'obtenir une robustesse presque parfaite face à ces modifications couleur.

4.6.2.7 Attaques géométriques

Pour finir, nous donnons quelques commentaires sur certaines attaques géométriques. Les méthodes de tatouage couleur discutées dans ce chapitre ne sont pas résistantes aux modifications géométriques d'image telles que les translations et les rotations car ce type d'attaque désynchronise les sites d'insertion d'une marque. Il en est de même lorsqu'une image est recadrée car la taille de celle-ci est modifiée. Dans le cas du découpage d'image (*image cropping*), nous pouvons obtenir une résistance partielle. En effet, comme la stratégie de synchronisation de nos méthodes sélectionne les sites d'insertion de manière aléatoire et uniforme, le pourcentage de régions découpées de l'image marquée est lié au taux d'erreur. Cependant, il est tout à fait possible d'adapter des solutions proposées dans la littérature dans le cadre des images en niveaux de gris.

4.6.2.8 Conclusion des expérimentations

Ces expériences psychovisuelles permettent de valider une amélioration significative de l'invisibilité d'une marque de la méthode AA qui sélectionne un axe de direction adaptatif psychovisuel en fonction du pixel couleur

Dans ces expériences psychovisuelles, nous avons comparé l'invisibilité de la méthode GA qui se base sur une quantification couleur avec un axe de direction fixe et la méthode AA basée sur une quantification couleur qui sélectionne un axe de direction adaptatif psychovisuel en fonction du pixel couleur. En moyenne, les marques insérées avec la méthode AA possèdent une bien meilleure insivilibilité psychovisuelle par rapport à celles insérées avec la méthode GA à distorsion numérique égale. Ce résultat peut être visuellement apprécié par le lecteur avec des images qui ont été agrandies et recadrées mais ce résultat a été confirmé par des expériences avec observateurs humains.

Face à plusieurs attaques, nous avons comparé la robustesse des méthodes GA et AA à distorsion psychovisuelle égale. Plus précisément, des images ont été marquées avec différents niveaux de distorsions numériques puis nous avons déterminés à l'oeil nu le pas de quantification moyens à partir duquel la marque insérée devenait visible. La méthode AA devient donc plus robuste que la méthode GA car $\Delta_{AA} > \Delta_{GA}$ sauf dans le cas d'attaques par filtrage où l'étape de détection échoue (taux d'erreur égaux ou proches de 0.5).

4.7 Conclusion

Dans ce chapitre, nous avons proposé une méthode de quantification vectorielle pour le tatouage numérique des images couleur. Celle-ci permet d'appliquer des techniques de quantification selon une direction fixée dans l'espace RGB. Dans un premier temps, nous avons étudié et constaté que fixer la même direction pour tous les pixels ne permettait pas d'obtenir un tatouage suffisament invisible.

Nous avons ensuite proposé une approche psychovisuelle du SVH nous permettant de modéliser le comportement des photorécepteurs de la rétine humaine. Grâce à la calibration des constantes basée sur les mesures d'ellipses de MacAdam, le modèle des photorécepteurs a permis de simuler le comportement d'un SVH moyen au niveau de la perception des différences de couleur.

Nous l'avons ensuite appliqué au tatouage numérique des images en couleur afin d'y extraire des vecteurs direction nécessaires à la quantification vectorielle et avons choisi d'adapter la méthode LQIM pour pouvoir tatouer des images en couleur.

En terme d'invisibilité, la méthode AA permet d'obtenir une bien meilleure invisibilité psychovisuelle que la méthode GA à distorsion numérique égale. En réajustant le compromis invisibilité/robustesse, nous avons une amélioration en robustesse avec la méthode AA par rapport à la méthode GA pour toutes les modifications d'image sauf avec la compression JPEG dans le domaine spatial. Les meilleures performances étant obtenues dans le domaines des ondelettes.

Conclusion et perspectives

Durant ce travail, nous avons étudié et apporté des contributions quant à la problématique de protection des contenus multimédias. L'approche choisie pour répondre à ce problème est le tatouage numérique robuste des images. En premier lieu, nous avons étudié le problème de la robustesse. Nous nous sommes positionné dans le cadre de l'intégration de codes correcteurs bien choisis pour renforcer la robustesse d'une marque. Dans un second temps, nous avons étudié l'invisibilité de la marque à travers l'utilisation de modèle psychovisuel basé sur la modélisation du système visuel humain.

Plus précisemment, nous nous sommes intéressés à l'amélioration de la robustesse de méthodes de tatouage. Une image marquée puis transmise sur un canal subit des modifications qui vont dégrader la marque. Nous avons déployé une piste théorique visant à chercher et étudier la structure des erreurs produites par une attaque.En analysant la structure de ces erreurs induites, nous pouvons a-priori les corriger efficacement avec un code correcteur adéquat. Nous avons étudié en premier lieu les codes en norme infinie et avons proposé une adaptation au tatouage par quantification. Nous avons montré que cela correspond à une reformulation de la méthode QIM sous le point de vue des codes correcteurs. En poursuivant ce volet, nous avons étudié une structure d'erreur particulière appelée erreurs homogènes ou presque. Ce type d'erreur peut être corrigé par l'association des codes en norme infinie et des codes BCH.

Ensuite, notre attention s'est portée sur l'étude d'un détecteur modifié de la méthode LQIM et plus particulièrement sur l'influence de la norme utilisée avec ce détecteur face aux erreurs provoquées par une attaque. Nous avons mis en évidence certaines propriétés sur les différences entre détecteurs. Cependant, nous n'obtenons pas de résultats expérimentaux permettant de montrer l'impact de la norme utilisée dans une utilisation «appliquée».

En reprenant le concept de l'étude de la structure de l'erreur, nous nous sommes alors intéressés aux codes en métrique rang, qui selon nos travaux, possèdent un fort potentiel d'application pour le tatouage numérique. Nous avons alors combiné ces codes avec la méthode LQIM et décrit la chaine numérique insertion-détection associée. Nous avons alors discuté d'une invariance théorique aux modifications de luminance avec un coût en distorsion d'insertion faible

de l'image hôte, tout en assurant une robustesse standard face aux autres attaques classiques selon le domaine d'insertion. Ensuite, en intégrant une stratégie par bloc et associée à un multidécodage nous avons ouvert la voix à une robustesse au découpage et au collage d'images, assurant un spectre large à la fois à des attaques «ponctuelles» mais aussi géométriques.

Dans la seconde partie de cette thèse, nous nous sommes intéressés au problème de l'invisibilité d'une marque pour le SVH. Pour cela, nous avons construit une méthode de quantification vectorielle pour le tatouage numérique des images couleur qui insère de l'information tout simplement selon une direction fixée dans l'espace RGB. Cependant, nous avons montré que fixer la direction a un impact très important sur l'invisibilité et indirectement la robustesse.

Pour résoudre ce problème, nous avons proposé une approche psychovisuelle reposant sur l'utilisation d'un modèle du SVH permettant de traduire le comportement des photorécepteurs de la rétine humaine. Grâce à la calibration des constantes basée sur les mesures d'ellipses de MacAdam, le modèle des photorécepteurs a permis de simuler le comportement d'un SVH moyen au niveau de la perception des différences de couleur. Nous l'avons ensuite appliqué au tatouage numérique des images en couleur afin d'y extraire des vecteurs direction nécessaires à la quantification vectorielle et avons choisi d'adapter la méthode LQIM pour pouvoir tatouer des images en couleur. Nous avons alors pu montrer que cette méthode adaptative permet d'obtenir une bien meilleure invisibilité qu'une méthode à direction constante (à distorsion numérique égale). De plus, en réajustant le compromis invisibilité/robustesse, nous avons obtenu une amélioration en robustesse avec cette méthodepour la majorité des attaques, puisque nous acceptons une dégradation numérique supérieure.

Perspectives de recherche

Tatouage et codes correcteurs

Le concept utilisé dans la première partie de cette thèse est l'étude de la structure des erreurs afin de mieux les corriger. Pour y parvenir, nous avons proposé d'utiliser des codes correcteurs dont les codes en métrique rang. Nous sommes convaincus qu'il existe encore de nombreux moyens d'intégrer ces codes dans une stratégie de tatouage pour résister à d'autres attaques. Par ailleurs, la robustesse d'une méthode face à un certain type d'erreur ne dépend pas uniquement du code correcteur utilisé mais aussi de la méthode d'insertion. Par exemple, le codage par treillis et le codage par syndrome peuvent être étudiés dans la suite de ces travaux. De même, la méthode de synchronisation des sites d'insertion joue un rôle important comme nous avons pu le voir avec la décomposition par bloc de l'image dans le chapitre 3. D'autre part, ces travaux peuvent s'étendre directement au tatouage des images couleur et ouvre un champ d'étude quant à la possible structure d'erreur liée à la notion de couleur.

Tatouage psychovisuel

L'oeil humain n'est pas le seul endroit où la perception de la couleur est prise en compte. Une partie du cerveau humain s'occupe de traiter et d'interpréter les informations perçues par la rétine. L'étude et la modélisation de celle-ci permettraient d'obtenir une compréhension plus fine du SVH et donc un modèle psychovisuel appliqué au tatouage des images couleur plus performant.

Dans ces travaux, nous avons traité le problème de la minimisation de la perception des différences de couleur entre deux pixels couleur. D'ailleurs, une perspective mène à la création d'une métrique psychovisuelle permettant de mesurer les distorsions entre deux images telle que le ferait un SVH. Cependant, la notion de tatouage couleur perceptuel inclut également le concept de saillance visuelle par exemple. Une autre perspective de travaux peut alors se porter sur une étude locale liée au contenu des images couleur. En effet, certaines zones d'une image couleur sont plus attractives que d'autres pour le SVH. Par conséquent, ajouter un bruit de quantification plus faible dans les régions d'intérêt d'une image couleur améliorerait encore l'invisibilité d'une marque.

Annexe Codes de Gabidulin

Nous proposons dans cette annexe des détails sur les codes en métrique rang pour le traitement d'image et le tatouage numérique.

Pour pouvoir assimiler les notions abordées, le lecteur doit être familié avec l'algèbre linéaire et l'arithmétique des corps finis en premier lieu. De nombreuses ressources pédagogiques sont disponibles en ligne. Il est de même pour les codes de Hamming, les codes BCH et les codes de Reed-Solomon. Plus de détails sur les codes en métrique rang peuvent être consultés dans [131, 132, 133].

Outils préliminaires

Automorphisme de Frobenius

Tout d'abord, nous avons besoin de définir l'automorphisme de Frobenius θ sur $GF(q^m)$ ($GF(q)$-espace vectoriel de dimension m) :

$$\theta : x \mapsto x^q \tag{4.19}$$

et nous notons θ^i la i-ème puissance (au sens de la composition) de l'automorphisme de Frobenius tel que :

$$\theta^i : x \mapsto x^{q^i} = x^{[i]} \tag{4.20}$$

Par exemple, nous avons pour $i = 2$:

$$\theta^2(x) = \theta \circ \theta(x) = (x^q)^q = x^{q^2}$$

Anneaux des q-polynômes

L'automorphisme de Frobenius est utilisé dans l'anneau des q-polynômes (ou polynômes linéarisés) de variable X que nous notons $GF(q^m)[X, \theta]$. Nous avons :

$$GF(q^m)[X] = \{A(X) = \sum_{i=0}^{n} a_i X^{[i]}, a_i \in GF(q^m), n \geq 0\} \tag{4.21}$$

L'ensemble $GF(q^m)[X, \theta]$ est l'anneau des polynômes à coefficients dans $GF(q^m)$ muni de l'addition usuelle et de la multiplication (c'est la composition des polynômes, l'opération est donc non commutative). Elle est définie pour tout $A, B \in GF(q^m)[X]$ de q-degré respectifs d_A et d_B telle que :

$$A(X) \times B(X) = A(B(X)) = \sum_{i=0}^{d_A} a_i \theta^i \left(\sum_{j=0}^{d_B} b_j \theta^j(X) \right) = \sum_{i=0}^{d_A} \sum_{j=0}^{d_B} a_i b_j^{[i]} X^{[i+j]} \tag{4.22}$$

Le produit de deux q-polynomes est un polynome de q-degré $d_A + d_B$.

Par exemple, en se plaçant dans le corps $GF(2^8) \cong GF(2)/(X^3 + X + 1)$, les produits de deux 2-polynômes $A(X) = X^4 + \alpha X^2 + X$ et $B(X) = \alpha^2 X^2 + X$ sont :

$A(X)B(X) = A(B(X)) = (\alpha^2 X^2 + X)^4 + \alpha(\alpha^2 X^2 + X)^2 + \alpha^2 X^2 + X = \alpha^8 X^8 + X^4 + \alpha(\alpha^4 X^4 + X^2) + \alpha^2 X^2 + X = \alpha X + X^4 + \alpha^5 X^4 + \alpha X^2 + \alpha^2 X^2 + X = (1 + \alpha^5) X^4 + (\alpha + \alpha^2) X^2 + (1 + \alpha) X = \alpha^4 X^4 + \alpha^4 X^2 + \alpha^3 X$

$B(X)A(X) = B(A(X)) = \alpha^2(X^4 + \alpha X^2 + X)^2 + X^4 + \alpha X^2 + X = \alpha^2(X^8 + \alpha^2 X^4 + X^2) + X^4 + \alpha X^2 + X = (1 + \alpha^4) X^4 + (\alpha + \alpha^2) X^2 + (1 + \alpha^2) X = \alpha^5 X^4 + \alpha^4 X^2 + \alpha^6 X$

Nous pouvons voir que l'opération de multiplication ne commute pas car $A(X)B(X) \neq B(X)A(X)$. L'anneau des q-polynômes est un cas particulier des polynômes de Ore [134, 135, 136]. Cet anneau est euclidien à droite c'est à dire que pour deux q-polynômes A et B ($d_B < d_A$), il existe deux q-polynômes uniques Q et R avec $d_R < d_Q$. Si $R = 0$, B divise A à droite. Il est alors possible de définir les notions de PGCD et de PPCM à droite, etc.

Construction

Les codes de Gabidulin [93] sont une famille de codes en métrique rang analogue aux codes de Reed-Solomon pour la métrique de Hamming. La construction d'un code de Gabidulin se fait en choisissant un vecteur $g = (g_1, \ldots, g_n) \in GF(q^m)^n$. Avec chaque composante g_i qui peuvent être représentée comme un élément de $GF(q)^m$ (structure d'espace vectoriel sur le corps de base). La famille de vecteur $\{g_1, \ldots, g_n\}$ doit être libre dans $GF(q)^m$.

En posant la matrice G suivante :

$$G = \begin{pmatrix} g_1 & \ldots & g_n \\ \vdots & & \vdots \\ g_1^{[k-1]} & \ldots & g_n^{[k-1]} \end{pmatrix} \tag{4.23}$$

nous avons un code rang linéaire de matrice génératrice G. C'est un code de Gabidulin de dimension k et de longueur n sur le corps $GF(q)^m$ de vecteur générateur g noté $Gab_k(g)$. Sa distance minimale est $d = n - k + 1$.

L'ensemble des mots de code de $Gab_k(g)$ est donné par :

$$Gab_k(g) = \left\{ (A(g_1), \ldots, A(g_n)); A(X) = \sum_{i=0}^{k-1} a_i X^{[i]}, a_i \in GF(q^m) \right\} \tag{4.24}$$

Nous pouvons voir que le code rang $Gab_k(g)$ est un code d'évaluation de q-polynômes de q-degré strictement supérieur à k sur le vecteur g. Notons que les codes de Reed-Solomon sont aussi des codes d'évaluation de polynômes.

Un exemple important de matrice génératrice est la matrice génératrice G_s sous forme systématique de $Gab_k(g)$:

$$G_s = \begin{pmatrix} 1 & 0 & \ldots & 0 & P_1(g_{k+1}) & \ldots & P_1(g_n) \\ 0 & 1 & \ddots & 0 & P_2(g_{k+1}) & \ldots & P_2(g_n) \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & P_k(g_{k+1}) & \ldots & P_k(g_n) \end{pmatrix} \tag{4.25}$$

avec P_i ($i \in \{1, \ldots, k\}$) l'unique polynôme de q-degré k vérifiant pour tout $j \leq k, P_i(g_j) = \delta_{i,j}$ (δ est le symbole de Kronecker).

Nous pouvons aussi donner une matrice de parité $H \in \mathcal{M}_{n-k,n}(GF(q^m))$ de $Gab_k(g)$:

$$H = \begin{pmatrix} h_1 & \dots & h_n \\ \vdots & \ddots & \vdots \\ h_1^{[d-2]} & \dots & h_n^{[d-2]} \end{pmatrix} \tag{4.26}$$

avec $h = (\lambda_1^{[d]}, \dots, \lambda_n^{[d]})$ et les λ_i solutions de l'équation

$$\sum_{i=1}^{n} \lambda_i g_i^{[j]} = 0 \tag{4.27}$$

Grâce à cette matrice de parité H, nous pouvons vérifier qu'un vecteur $y \in GF(q^m)^n$ appartient ou non à $Gab_k(g)$ en calculant le syndrome $s(y)$ de y :

$$s(y) = Hy^T \tag{4.28}$$

car $y \in Gab_k(g)$ si et seulement si $s(y) = 0$. D'ailleurs, nous avons $HG^T = 0$.

Décodage

Les codes de Gabidulin sont MRD (Maximum Rank Distance) au même titre que les codes de Reed-Solomon qui sont MDS (Maximum Distance Separable). En posant $y = c + e, c \in Gab_k(g)$, on peut décoder de manière uniquement y en c si et seulement si :

$$rk(e) \leq \lfloor (n-k)/2 \rfloor \tag{4.29}$$

Le décodage par syndrome est un algorithme non trivial qui peut être utilisé pour le décodage des codes de Gabidulin mais reste assez lent.

Comme énoncé précédemment, ce sont des codes d'évaluation de polynômes (linéaires) comme pour les codes de Reed-Solomon. Ainsi, l'algorithme de Welch-Berlekamp peut être adapté pour le décodage des codes de Gabidulin. Cette algorithme a été proposé par Loidreau [95] et est bien plus rapide que l'algorithme de décodage par syndrome précédemment présenté. D'autres algorithmes de décodage plus rapide ont été proposé dans la littérature ([137, 138]).

Exemple numérique

Énumération des éléments d'un corps fini de petit cardinal

Nous proposons maintenant un exemple concret de code de Gabidulin sur le corps $GF(16) \cong GF(2)[X]/(X^4+X+1) \cong GF(2)(\alpha)$. C'est une extension de corps de degré 4 sur le corps de base $GF(2)$. Nous donnons dans la table **??** la liste des éléments de ce corps ainsi que leur puissance correspondante par rapport à l'élément primitif α qui vérifie l'expression $\alpha^4+\alpha+1=0$. Il est bien primitif car $\alpha^3 \neq 1$ et $\alpha^5 = \alpha^2+\alpha \neq 1$, α est d'ordre 15.

$x \in GF(16)$	$\in (GF(2))^4$		$x \in GF(16)$	$\in (GF(2))^4$
0	$(0,0,0,0)$		α^7	$(1,1,0,1)$
1	$(1,0,0,0)$		α^8	$(1,0,1,0)$
α	$(0,1,0,0)$		α^9	$(0,1,0,1)$
α^2	$(0,0,1,0)$		α^{10}	$(1,1,1,0)$
α^3	$(0,0,0,1)$		α^{11}	$(0,1,1,1)$
α^4	$(1,1,0,0)$		α^{12}	$(1,1,1,1)$
α^5	$(0,1,1,0)$		α^{13}	$(1,0,1,1)$
α^6	$(0,0,1,1)$		α^{14}	$(1,0,0,1)$

Table des éléments du corps fini $GF(16)$ avec représentation dans le $GF(2)$-espace vectoriel de base $\{1,\alpha,\alpha^2,\alpha^3\}$. Cette table facilite les calculs à la main.

D'après la table, on a $\alpha^4 = 1.1+1.\alpha+0.\alpha^2+0.\alpha^3$, $\alpha^5 = 0.1+1.\alpha+1.\alpha^2+0.\alpha^3$, $\alpha^6 = 0.1+0.\alpha+1.\alpha^2+1.\alpha^3$, $\alpha^7 = 1.1+1.\alpha+0.\alpha^2+1.\alpha^3$, etc.

Matrice génératrice et matrice de parité

Nous désirons maintenant construire un code de Gabidulin de longueur $n=4$ et de dimension $k=2$ sur $GF(16)$. Ce code corriger des erreurs de rang 1 au plus d'après 4.29.

Choisissons maintenant 4 éléments de $GF(16)$ noté $g=(g_1,\ldots,g_4)$ tel quel que $\{g_1,\ldots,g_4\}$ est une famille libre sur $GF(2)^4$. Par exemple, nous pouvons prendre $g_1=1, g_2=\alpha, g_3=\alpha^2$ et $g_4=\alpha^3$. Les calculs peuvent être vérifiés avec le logiciel SageMath [139].

La matrice génératrice G de vecteur générateur g est :

$$G = \begin{pmatrix} 1^{[0]} & \alpha^{[0]} & (\alpha^2)^{[0]} & (\alpha^3)^{[0]} \\ 1^{[1]} & \alpha^{[1]} & (\alpha^2)^{[1]} & (\alpha^3)^{[1]} \end{pmatrix} = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 \\ 1 & \alpha^2 & \alpha+1 & \alpha^2+\alpha^3 \end{pmatrix} \tag{4.30}$$

Nous pouvons également obtenir la matrice génératrice sous forme systématique G_s soit en calculant les polynômes décrit dans l'équation 4.25, soit appliquant la méthode du pivot de Gauss pour obtenir la matrice identité sur le bloc de gauche de G dont les étapes sont les suivantes :

$$\begin{cases} L_2 \leftarrow L_2 + L_1 \\ L_2 \leftarrow \alpha^{(2^4-1)-2} \times L_2 \\ L_1 \leftarrow L_1 + \alpha \times L_2 \end{cases} \tag{4.31}$$

Nous obtenons :

$$G_s = \left(\begin{array}{cc|cc} 1 & 0 & \alpha^3 & \alpha^8 \\ 0 & 1 & \alpha^5 & \alpha^{12} \end{array}\right) \tag{4.32}$$

Grâce à G_s, le calcul de la matrice de parité associée est directe. Notons A le bloc de droite de G_s. Nous avons :

$$H = \left(\begin{array}{cc|cc} \alpha^3 & \alpha^5 & 1 & 0 \\ \alpha^8 & \alpha^{12} & 0 & 1 \end{array}\right) \tag{4.33}$$

Nous pouvons voir que le bloc de gauche de H est $-A^T = A^T$ car nous sommes en caractéristique $q = 2$. En calculant le produit $H \times (G_s)^T$, le lecteur pourra vérifier que le résultat est la matrice nulle.

Encodage et syndrome

Soit $m = (1, \alpha) \in GF(2)^2$ un message à encoder. Le mot de code associé est :

$$c = mG = (1, \alpha) \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 \\ 1 & \alpha^2 & \alpha+1 & \alpha^2+\alpha^3 \end{pmatrix} = (1+\alpha, \alpha+\alpha^3, \alpha, \alpha+1) \tag{4.34}$$

L'expression de c sous forme matricielle est :

$$Mat(c) = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{4.35}$$

car chaque colonne (donnée dans la table **??**) exprime chaque composante de c dans le corps de base $GF(2)$. Le code $Gab_2(g)$ corriger des erreurs de rang 1 au plus.

Nous pouvons aussi encoder m avec G_s :

$$\begin{aligned} c_1 = mG_s =& (1, \alpha) \begin{pmatrix} 1 & 0 & \alpha^3 & \alpha^8 \\ 0 & 1 & \alpha^5 & \alpha^{12} \end{pmatrix} \\ =& (1, \alpha, \alpha^3 + \alpha^6, \alpha^8 + \alpha^{13}) \\ =& (1, \alpha, \alpha^2, \alpha^3) \end{aligned} \tag{4.36}$$

puis vérifier que le syndrome est nul :

$$\begin{aligned} s = Hc_1^T =& \begin{pmatrix} \alpha^3 + \alpha^6 + \alpha^2 + 0 \\ \alpha^8 + \alpha^{13} + 0 + \alpha^3 \end{pmatrix} \\ =& \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{aligned} \tag{4.37}$$

Décodage

Nous pouvons retrouver le mot de code c_1 à partir du mot reçu $y = c_1 + e$ si et seulement si le rang de l'erreur est inférieur ou égal à 1 car la distance minimale du code est $d = n - k + 1 = 3$. En utilisant l'algorithme de décodage naïf, nous avons $(2^4)^2$ calculs pour déterminer quelle est l'erreur de poids le plus faible tel que :

$$\mathcal{D} = \{y - c, c \in Gab_2(g)\} \tag{4.38}$$

Nous pouvons observer la distribution des rangs d'erreur avec :

$$\mathcal{D}_r = \{e \in \mathcal{D}, rk(e) = r\} \tag{4.39}$$

et

$$\mathcal{R} = \{|\mathcal{D}_1|, \ldots, |\mathcal{D}_n|\} \tag{4.40}$$

Ainsi, pour un erreur e de rang 1, nous avons la distribution suivante :

$$\mathcal{R} = \{1, 28, 149, 78\} \tag{4.41}$$

Le poids rang minimal est donc 1 et l'ensemble $\mathcal{D}_1$ ne possède qu'un seul élément, nous pouvons donc identifier notre mot de code.

Insertion du mot de code dans l'image

Dans une stratégie de tatouage de base, l'insertion d'un mot de code rang se fait en utilisant l'expression sous forme de matrice binaire. Par exemple, nous pouvons reprendre la matrice $Mat(c)$ en 4.35 pour insérer le mot de code rang c. Puis nous concaténons les lignes de $Mat(c)$ en ligne telle que :

$$Mat(c) = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \longmapsto w = (\underbrace{1,0,0,1}_{\text{Ligne1}}, \underbrace{1,1,1,1}_{\text{Ligne2}}, \underbrace{0,0,0,0}_{\text{Ligne3}}, \underbrace{0,1,0,0}_{\text{Ligne4}}) \tag{4.42}$$

avec w est le message binaire à insérer. Puis, chaque bit de w est associé à un ou plusieurs pixels (choisis aléatoirement par exemple) de l'image hôte.

Cependant, nous proposons dans le chapitre 3 une stratégie d'insertion différente afin de résister au cropping. Pour un mot de code rang de taille $m \times n$ (sous forme matricielle), nous décomposons l'image hôte en mn blocs puis associons à chaque bloc un bit d'information.

Dans notre exemple, nous avons $m = n = 4$. Nous obtenons :

$$C = (c_{i,j})_{i,j} = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} \\ c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} \end{pmatrix} \longmapsto \underbrace{\begin{array}{|c|c|c|c|} \hline c_{1,1} & c_{1,1} & c_{1,3} & c_{1,4} \\ \hline c_{2,1} & c_{2,1} & c_{2,3} & c_{2,4} \\ \hline c_{3,1} & c_{3,1} & c_{3,3} & c_{3,4} \\ \hline c_{4,1} & c_{4,1} & c_{4,3} & c_{4,4} \\ \hline \end{array}}_{\text{Image hôte découpée en bloc } 4 \times 4} \tag{4.43}$$

Dans chaque bloc, L coefficients (dans le cas de la méthode LQIM) sont choisis aléatoirement pour être associé au bit correspondant. Pour une image de taille $h \times w$, la taille d'un bloc est de $h/m \times w/n$.

Bibliographie

[1] S. Baudry, J.-F. Delaigle, B. Sankur, B. Macq, and H. Maitre, "Analyses of error correction strategies for typical communication channels in watermarking," *Signal Processing*, vol. 81, no. 6, pp. 1239–1250, 2001.

[2] S. Zinger, Z. Jin, H. Maitre, and B. Sankur, "Optimization of watermarking performances using error correcting codes and repetition," in *Communications and Multimedia Security Issues of the New Century*, pp. 229–240, Springer, 2001.

[3] W. Abdul, P. Carré, and P. Gaborit, "Error correcting codes for robust color wavelet watermarking," *EURASIP Journal on Information Security*, vol. 2013, p. 1, Feb 2013.

[4] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*. Morgan Kaufmann, 2007.

[5] U. Kohl, J. Lotspiech, and M. A. Kaplan, "Safeguarding digital library contents and users," *D-lib Magazine*, vol. 3, no. 9, 1997.

[6] J. Meng and S.-F. Chang, "Embedding visible video watermarks in the compressed domain," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 1, pp. 474–477, IEEE, 1998.

[7] F. Cayre, P. Rondao-Alface, F. Schmitt, B. Macq, and H. Maıtre, "Application of spectral decomposition to compression and watermarking of 3d triangle mesh geometry," *Signal Processing : Image Communication*, vol. 18, no. 4, pp. 309–319, 2003.

[8] F. Cayre, O. Devillers, F. Schmitt, and H. Maître, *Watermarking 3D triangle meshes for authentication and integrity*. PhD thesis, INRIA, 2004.

[9] P. R. Alface, B. Macq, and F. Cayre, "Blind and robust watermarking of 3d models : How to withstand the cropping attack ?," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 5, pp. V–465, IEEE, 2007.

[10] W. Yuan, H. Li, L. Li, X. Feng, J. Lu, and C.-C. C. C.-C. Chang, "A watermarking mechanism with high capacity for 3d mesh objects using integer planning," *IEEE MultiMedia*, 2018.

[11] J. P. Stern, G. Hachez, F. Koeune, and J.-J. Quisquater, "Robust object watermarking : Application to code," in *International Workshop on Information Hiding*, pp. 368–378, Springer, 1999.

[12] A. B. Kahng, S. Mantik, I. L. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe, "Robust ip watermarking methodologies for physical design," in *Proceedings of the 35th annual Design Automation Conference*, pp. 782–787, ACM, 1998.

[13] M. L. Miller, I. J. Cox, J.-P. M. Linnartz, and T. Kalker, "A review of watermarking principles and practices," *Digital signal processing in multimedia systems*, pp. 461–485, 1999.

[14] R. Liu and T. Tan, "An svd-based watermarking scheme for protecting rightful ownership," *IEEE transactions on multimedia*, vol. 4, no. 1, pp. 121–128, 2002.

[15] Z. Chen, L. Li, H. Peng, Y. Liu, and Y. X. Yang, "A novel digital watermarking based on general non-negative matrix factorization," *IEEE Transactions on Multimedia*, vol. PP, no. 99, pp. 1–1, 2018.

[16] H. Lu, R. Shen, and F.-L. Chung, "Fragile watermarking scheme for image authentication," *Electronics Letters*, vol. 39, pp. 898–900, Jun 2003.

[17] C. K. Ho and C.-T. Li, "Semi-fragile watermarking scheme for authentication of jpeg images," in *Information Technology : Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, vol. 1, pp. 7–11, IEEE, 2004.

[18] K. Maeno, Q. Sun, S.-F. Chang, and M. Suto, "New semi-fragile image authentication watermarking techniques using random bias and nonuniform quantization," *IEEE Transactions on Multimedia*, vol. 8, no. 1, pp. 32–45, 2006.

[19] G. Depovere, T. Kalker, J. Haitsma, M. Maes, L. De Strycker, P. Termont, J. Vandewege, A. Langell, C. Alm, P. Norman, *et al.*, "The viva project : digital watermarking for broadcast monitoring," in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, vol. 2, pp. 202–205, IEEE, 1999.

[20] A. Khan, A. Siddiqa, S. Munib, and S. A. Malik, "A recent survey of reversible watermarking techniques," *Information Sciences*, vol. 279, pp. 251 – 272, 2014.

[21] M. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Image Processing, 1997. Proceedings., International Conference on*, vol. 2, pp. 680–683, IEEE, 1997.

[22] R. B. Wolfgang and E. J. Delp, "Fragile watermarking using the vw2d watermark," in *Security and Watermarking of Multimedia Contents*, vol. 3657, pp. 204–214, International Society for Optics and Photonics, 1999.

[23] A. Shehab, M. Elhoseny, K. Muhammad, A. K. Sangaiah, P. Yang, H. Huang, and G. Hou, "Secure and robust fragile watermarking scheme for medical images," *IEEE Access*, vol. 6, pp. 10269–10278, 2018.

[24] S. Bravo-Solorio, F. Calderon, C.-T. Li, and A. K. Nandi, "Fast fragile watermark embedding and iterative mechanism with high self-restoration performance," *Digital Signal Processing*, vol. 73, pp. 83–92, 2018.

[25] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-lsb data embedding," *IEEE transactions on image processing*, vol. 14, no. 2, pp. 253–266, 2005.

[26] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," in *Security and Watermarking of Multimedia contents III*, vol. 4314, pp. 197–209, International Society for Optics and Photonics, 2001.

[27] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE transactions on image processing*, vol. 13, no. 8, pp. 1147–1156, 2004.

[28] H. Heijmans and L. Kamstra, "Reversible data embedding based on the haar wavelet decomposition," *Proceeding of VIIth Digital Image Computing : Techniques and Applications*, 2003.

[29] J. Stach and A. M. Alattar, "A high-capacity invertible data-hiding algorithm using a generalized reversible integer transform," in *Security, Steganography, and Watermarking of Multimedia Contents VI*, vol. 5306, pp. 386–397, International Society for Optics and Photonics, 2004.

[30] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on circuits and systems for video technology*, vol. 16, no. 3, pp. 354–362, 2006.

[31] M. van der Veen, F. Bruekers, A. van Leest, and S. Cavin, "High capacity reversible watermarking for audio," in *Security and Watermarking of Multimedia Contents V*, vol. 5020, pp. 1–12, International Society for Optics and Photonics, 2003.

[32] B. Chen and G. W. Wornell, "Quantization index modulation : A class of provably good methods for digital watermarking and information embedding," *IEEE TRANS. ON INFORMATION THEORY*, vol. 47, no. 4, pp. 1423–1443, 1999.

[33] R. Pickholtz, D. Schilling, and L. Milstein, "Theory of spread-spectrum communications–a tutorial," *IEEE transactions on Communications*, vol. 30, no. 5, pp. 855–884, 1982.

[34] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE transactions on image processing*, vol. 6, no. 12, pp. 1673–1687, 1997.

[35] F. Pérez-González, C. Mosquera, M. Barni, and A. Abrardo, "Rational dither modulation : A high-rate data-hiding method invariant to gain attacks," *IEEE transactions on signal processing*, vol. 53, no. 10, pp. 3960–3975, 2005.

[36] R. Zamir and M. Feder, "On lattice quantization noise," in *Data Compression Conference, 1994. DCC'94. Proceedings*, pp. 380–389, IEEE, 1994.

[37] G. W. W. Brian Chen, "Dither modulation : a new approach to digital watermarking and information embedding," 1999.

[38] P. Moulin and R. Koetter, "Data-hiding codes," *Proceedings of the IEEE*, vol. 93, pp. 2083–2126, Dec 2005.

[39] J. J. Eggers, J. Su, and B. Girod, "A blind watermarking scheme based on structured codebooks," 2000.

[40] P. Bas, N. Le Bihan, and J.-M. Chassery, "Color image watermarking using quaternion fourier transform," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 3, pp. III–521, IEEE, 2003.

[41] S.-H. Wang and Y.-P. Lin, "Wavelet tree quantization for copyright protection watermarking," *IEEE transactions on image processing*, vol. 13, no. 2, pp. 154–165, 2004.

[42] J. C. Oostveen, T. Kalker, and M. Staring, "Adaptive quantization watermarking," in *Security, Steganography, and Watermarking of Multimedia Contents VI*, vol. 5306, pp. 296–304, International Society for Optics and Photonics, 2004.

[43] Q. Li and I. J. Cox, "Improved spread transform dither modulation using a perceptual model : robustness to amplitude scaling and jpeg compression," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 2, pp. II–185, IEEE, 2007.

[44] A. B. Watson, "Dct quantization matrices visually optimized for individual images," in *proc. SPIE*, vol. 1913, 1993.

[45] R. Hu, F. Chen, and H. Yu, "Incorporating watson's perceptual model into patchwork watermarking for digital images," in *2010 IEEE International Conference on Image Processing*, pp. 3705–3708, Sept 2010.

[46] I. J. Cox, G. Doërr, and T. Furon, "Watermarking is not cryptography," in *Digital Watermarking* (Y. Q. Shi and B. Jeon, eds.), (Berlin, Heidelberg), pp. 1–15, Springer Berlin Heidelberg, 2006.

[47] M. Barni, F. Bartolini, and T. Furon, "A general framework for robust watermarking security," *Signal Processing*, vol. 83, no. 10, pp. 2069–2084, 2003.

[48] T. Kalker, "Considerations on watermarking security," in *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*, pp. 201–206, IEEE, 2001.

[49] P. Bas and J. Hurri, "Security of dm quantization watermarking schemes : a practical study for digital images," in *International Workshop on Digital Watermarking*, pp. 186–200, Springer, 2005.

[50] L. Pérez-Freire, P. Comesana, J. R. Troncoso-Pastoriza, and F. Pérez-González, "Watermarking security : a survey," in *Transactions on Data Hiding and Multimedia Security I*, pp. 41–72, Springer, 2006.

[51] P. Bas and F. Cayre, "Achieving subspace or key security for woa using natural or circular watermarking," in *Proceedings of the 8th workshop on Multimedia and security*, pp. 80–88, ACM, 2006.

[52] P. Bas and G. Doërr, "Practical security analysis of dirty paper trellis watermarking," in *International Workshop on Information Hiding*, pp. 174–188, Springer, 2007.

[53] P. Bas and T. Furon, "A new measure of watermarking security : the effective key length," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1306–1317, 2013.

[54] P. Bas, T. Furon, F. Cayre, G. Doërr, and B. Mathon, *Watermarking Security : Fundamentals, Secure Designs and Attacks*. Springer, 2016.

[55] M. Kesal, M. K. Mihcak, R. Koetter, and P. Moulin, "Iteratively decodable codes for watermarking applications," in *Proc. 2nd Int. Symp. on Turbo Codes and Related Topics*, 2000.

[56] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, pp. 498–519, Feb 2001.

[57] O. Al-Askary, *Iterative decoding of product codes*. PhD thesis, Signaler, sensorer och system, 2003.

[58] J. Darbon, B. Sankur, and H. Maitre, "Error correcting code performance for watermark protection," in *Security and Watermarking of Multimedia Contents III*, vol. 4314, pp. 663–673, International Society for Optics and Photonics, 2001.

[59] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*. Elsevier, 1977.

[60] C.-T. Hsieh and Y.-K. Wu, "Digital image multiresolution watermark based on human visual system using error correcting code," vol. 4, no. 3, pp. 201–209, 2001.

[61] P. W. Chan, M. R. Lyu, and R. T. Chin, "A novel scheme for hybrid digital video watermarking : approach, evaluation and experimentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 1638–1649, Dec 2005.

[62] P.-W. Chan and M. R. Lyu, "A dwt-based digital video watermarking scheme with error correcting code," in *International Conference on Information and Communications Security*, pp. 202–213, Springer, 2003.

[63] S.-C. Liu and S. D. Lin, "Bch code-based robust audio watermarking in the cepstrum domain," *Journal of Information Science and Engineering*, vol. 22, no. 3, pp. 535–543, 2006.

[64] F. Zhang, X. Zhang, and Z. Chen, "Digital image authentication based on error-correction codes," in *International Conference on Computational and Information Science*, pp. 433–438, Springer, 2005.

[65] C.-S. Chan and C.-C. Chang, "An efficient image authentication method based on hamming code," *Pattern Recognition*, vol. 40, no. 2, pp. 681–690, 2007.

[66] H. G. Schaathun, "On error-correcting fingerprinting codes for use with watermarking," *Multimedia Systems*, vol. 13, no. 5-6, pp. 331–344, 2008.

[67] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, April 1967.

[68] J. R. Hernandez, J.-F. Delaigle, and B. M. Macq, "Improving data hiding by using convolutional codes and soft-decision decoding," in *Security and Watermarking of Multimedia Contents II*, vol. 3971, pp. 24–48, International Society for Optics and Photonics, 2000.

[69] R. Lancini, F. Mapelli, and S. Tubaro, "A robust video watermarking technique in the spatial domain," in *International Symposium on VIPromCom Video/Image Processing and Multimedia Communications*, pp. 251–256, 2002.

[70] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding : Turbo-codes. 1," in *Communications, 1993. ICC'93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, vol. 2, pp. 1064–1070, IEEE, 1993.

[71] F. Baldo, F. P. González, and S. Scalise, "Turbo coding for sample-level watermarking in the dct domain.," in *ICIP (3)*, pp. 1003–1006, 2001.

[72] C. Rey, K. Amis, J.-L. Dugelay, R. Pyndiah, and A. Picart, "Enhanced robustness in image watermarking using turbo codes," in *Security and Watermarking of Multimedia Contents V*, vol. 5020, pp. 330–337, International Society for Optics and Photonics, 2003.

[73] J.-L. Dugelay, S. Roche, C. Rey, and G. Doërr, "Still-image watermarking robust to local geometric distortions," *IEEE transactions on image processing*, vol. 15, no. 9, pp. 2831–2842, 2006.

[74] C. Chemak, J. C. Lapayre, and M. S. Bouhlel, "A new scheme of image watermarking based on 5/3 wavelet decomposition and turbo-code," *WSEAS Transactions on Biology and Biomedicine*, vol. 4, no. 4, pp. 45–52, 2007.

[75] G. Doërr, C. Rey, and J.-L. Dugelay, "Watermark resynchronization based on elastic graph matching," in *Proc. Int. Conf. Sciences of Electronic, Technologies of Information and Telecommunications*, Citeseer, 2005.

[76] J. J. Eggers, J. K. Su, and B. Girod, "Robustness of a blind image watermarking scheme," in *Image Processing, 2000. Proceedings. 2000 International Conference on*, vol. 3, pp. 17–20, IEEE, 2000.

[77] S. Burgett, E. Koch, and J. Zhao, "Copyright labeling of digitized image data," *IEEE Communications Magazine*, vol. 36, no. 3, pp. 94–100, 1998.

[78] J. Dugelay and C. Rey, "Method of marking a multimedia document having improved robustness," *Pending Patent EP*, vol. 99480075, 2001.

[79] N. Cvejic, D. Tujkovic, and T. Seppanen, "Increasing robustness of an audio watermark using turbo codes," in *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, vol. 1, pp. I–217, IEEE, 2003.

[80] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. Von Der Malsburg, R. P. Wurtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Transactions on computers*, no. 3, pp. 300–311, 1993.

[81] S. Al Maeeni, F. Kalbat, and H. Al-Ahmad, "Logo embedding in grayscale images using turbo product codes," in *Information and Communication Technology Research (ICTRC), 2015 International Conference on*, pp. 96–99, IEEE, 2015.

[82] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and control*, vol. 3, no. 1, pp. 68–79, 1960.

[83] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the society for industrial and applied mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[84] R. W. Hamming, "Error detecting and error correcting codes," *Bell Labs Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[85] D. V. Sarwate and N. R. Shanbhag, "High-speed architectures for reed-solomon decoders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 5, pp. 641–655, 2001.

[86] J. S. Plank and Y. Ding, "Note : Correction to the 1997 tutorial on reed-solomon coding," *Software : Practice and Experience*, vol. 35, no. 2, pp. 189–194, 2005.

[87] H. Chang and C. Shung, "A reed-solomon product-code (rs-pc) decoder for dvd applications," in *Solid-State Circuits Conference, 1998. Digest of Technical Papers. 1998 IEEE International*, pp. 390–391, IEEE, 1998.

[88] H.-C. Chang, C. B. Shung, and C.-Y. Lee, "A reed-solomon product-code (rs-pc) decoder chip for dvd applications," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 2, pp. 229–238, 2001.

[89] D. Silva and F. R. Kschischang, "On metrics for error correction in network coding," *IEEE Transactions on Information Theory*, vol. 55, no. 12, pp. 5479–5490, 2009.

[90] E. M. Gabidulin, A. Paramonov, and O. Tretjakov, "Ideals over a non-commutative ring and their application in cryptology," in *Workshop on the Theory and Application of Cryptographic Techniques*, pp. 482–489, Springer, 1991.

[91] P. Gaborit, O. Ruatta, J. Schrek, and G. Zémor, "New results for rank-based cryptography," in *International Conference on Cryptology in Africa*, pp. 1–12, Springer, 2014.

[92] P. Delsarte, "Bilinear forms over a finite field, with applications to coding theory," *Journal of Combinatorial Theory, Series A*, vol. 25, no. 3, pp. 226–241, 1978.

[93] E. M. Gabidulin, "Theory of codes with maximum rank distance," *Problemy Peredachi Informatsii*, vol. 21, no. 1, pp. 3–16, 1985.

[94] E. M. Gabidulin, *A fast matrix decoding algorithm for rank-error-correcting codes*, pp. 126–133. Berlin, Heidelberg : Springer Berlin Heidelberg, 1992.

[95] P. Loidreau, *A Welch–Berlekamp Like Algorithm for Decoding Gabidulin Codes*, pp. 36–45. Berlin, Heidelberg : Springer Berlin Heidelberg, 2006.

[96] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079–1107, 1999.

[97] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Transparent robust image watermarking," in *Image Processing, 1996. Proceedings., International Conference on*, vol. 3, pp. 211–214, IEEE, 1996.

[98] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Robust data hiding for images," in *Digital Signal Processing Workshop Proceedings, 1996., IEEE*, pp. 37–40, IEEE, 1996.

[99] M. Holliman and N. Memon, "Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes," *IEEE Transactions on image processing*, vol. 9, no. 3, pp. 432–441, 2000.

[100] N. D. M. Jessica Fridrich, Miroslav Goljan, "Further attacks on yeung-mintzer fragile watermarking scheme," 2000.

[101] B. Fridrich and M. Goljan, "Protection of digital images using self embedding," 05 2001.

[102] A. Aggarwal and M. Singla, "Robust watermarking of color images under noise and cropping attacks in spatial domain," *image*, vol. 6, no. 9, p. 11, 2011.

[103] S. K. A. Khalid, M. M. Deris, and K. M. Mohamad, "Anti-cropping digital image watermarking using sudoku," *International Journal of Grid and Utility Computing*, vol. 4, no. 2-3, pp. 169–177, 2013.

[104] S. Saneie and A. Naghsh, "Introducing a new method of robust digital image watermarking against cropping and salt & pepper noise using sudoku," *Majlesi Journal of Multimedia Processing*, vol. 4, no. 4, 2016.

[105] M. S. Goli and A. Naghsh, "Introducing a new method robust against crop attack in digital image watermarking using two-step sudoku," in *Pattern Recognition and Image Analysis (IPRIA), 2017 3rd International Conference on*, pp. 237–242, IEEE, 2017.

[106] B. Felgenhauer and F. Jarvis, "Mathematics of sudoku i," *Mathematical Spectrum*, vol. 39, no. 1, pp. 15–22, 2006.

[107] E. Russell and F. Jarvis, "Mathematics of sudoku ii," *Mathematical Spectrum*, vol. 39, no. 2, pp. 54–58, 2006.

[108] M. Kutter, "Watermarking resistance to translation, rotation, and scaling," in *Multimedia Systems and Applications*, vol. 3528, pp. 423–432, International Society for Optics and Photonics, 1999.

[109] P. Bas, J.-M. Chassery, and B. Macq, "Geometrically invariant watermarking using feature points," *IEEE transactions on image Processing*, vol. 11, no. 9, pp. 1014–1028, 2002.

[110] M. Kutter, F. D. Jordan, and F. Bossen, "Digital signature of color images using amplitude modulation," vol. 3022, pp. 518–526, 1997.

[111] P.-T. Yu, H.-H. Tsai, and J.-S. Lin, "Digital watermarking based on neural networks for color images," *Signal Processing*, vol. 81, no. 3, pp. 663 – 671, 2001. Special section on Digital Signal Processing for Multimedia.

[112] G. Voyatzis and I. Pitas, "Digital image watermarking using mixing systems," *Computer & Graphics*, vol. 22, pp. 405–416, 1998.

[113] H.-S. Kim, H.-K. Lee, H.-Y. Lee, and Y.-H. Ha, "Digital watermarking based on color differences," vol. 4314, pp. 10–17, 2001.

[114] A. Abadpour and S. Kasaei, "Color pca eigenimages and their application to compression and watermarking," *Image and Vision Computing*, vol. 26, no. 7, pp. 878–890, 2008.

[115] G. Chareyron, B. Macq, and A. Tremeau, "Watermarking of color images based on segmentation of the xyz color space," *Conference on Colour in Graphics, Imaging, and Vision*, vol. 2004, no. 1, pp. 178–182, 2004.

[116] D. Coltuc and P. Bolon, "Robust watermarking by histogram specification," in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, vol. 2, pp. 236–239, IEEE, 1999.

[117] G. Chareyron and A. Trémeau, "Color images watermarking based on minimization of color differences," in *Int. Workshop on Multimedia Content, Representation, Classification and Security, MRCS'2006*, 2006.

[118] C. R. S. Rao and M. V. N. K. Prasad, *Color Image Watermarking Techniques Based on Magic Square and Ridgelets*, pp. 59–74. Cham : Springer International Publishing, 2016.

[119] J. Li, Q. Lin, C. Yu, X. Ren, and P. Li, "A qdct- and svd-based color image watermarking scheme using an optimized encrypted binary computer-generated hologram," *Soft Computing*, pp. 1–19, 2016.

[120] A. Parisis, P. Carre, C. Fernandez-Maloigne, and N. Laurent, "Color image watermarking with adaptive strength of insertion," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, vol. 3, pp. iii–85, IEEE, 2004.

[121] D. Alleysson and D. Méary, "Neurogeometry of color vision," *J Physiol Paris*, vol. 106, pp. 284–96, Mar. 2012.

[122] Q. Li and I. J. Cox, "Using perceptual models to improve fidelity and provide resistance to valumetric scaling for quantization index modulation watermarking," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 2, pp. 127–139, 2007.

[123] J. Petitot, "The neurogeometry of pinwheels as a sub-riemannian contact structure," *Journal of Physiology-Paris*, vol. 97, no. 2, pp. 265–309, 2003.

[124] J. Petitot, *Neurogéométrie de la vision : modeles mathematiques et physiques des architectures fonctionnelles*. Editions Ecole Polytechnique, 2008.

[125] J. Koenderink, W. Van de Grind, and M. Bouman, "Models of retinal signal processing at high luminances," *Kybernetik*, vol. 6, no. 6, pp. 227–237, 1970.

[126] E. Zrenner, A. Stett, S. Weiss, *et al.*, "Can microphotodiode arrays replace degenerated retinal photoreceptors to restore vision," *Vision Res*, vol. 39, no. 15, pp. 2555–2567, 1999.

[127] D. Alleysson and J. Hérault, "Differential thresholds in colour perception : a consequence of retinal processing and photoreceptor nonlinearities," *Perception ECVP abstract*, vol. 27, pp. 0–0, 1998.

[128] D. Alleysson and J. Hérault, "Variability in color discrimination data explained by a generic model with nonlinear and adaptive processing," *Color Research and Application*, vol. 26, no. S1, pp. S225–S229, 2001.

[129] D. Alleysson, *Le traitement du signal chromatique dans la rétine : un mod ?le de base pour la perception humaine des couleurs*. PhD thesis, Université Joseph Fourier-Grenoble 1 Sciences et Géographie, March 1999.

[130] D. L. MacAdam, "Visual sensitivities to color differences in daylight∗," *J. Opt. Soc. Am.*, vol. 32, pp. 247–274, May 1942.

[131] P. Loidreau, *Metrique rang et cryptographie*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2007.

[132] C. Faure, *Etudes de systèmes cryptographiques construits à l'aide de codes correcteurs, en métrique de Hamming et en métrique rang*. PhD thesis, Ecole Polytechnique X, 2009.

[133] G. Murat, *Résultants de polynômes de Ore et Cryptosystèmes de McEliece sur des Codes Rang faiblement structurés*. PhD thesis, Université de Limoges, 2014.

[134] O. Ore, "Theory of non-commutative polynomials," *Annals of mathematics*, pp. 480–508, 1933.

[135] O. Ore, "On a special class of polynomials," *Transactions of the American Mathematical Society*, vol. 35, no. 3, pp. 559–584, 1933.

[136] O. Ore, "Contributions to the theory of finite fields," *Transactions of the American Mathematical Society*, vol. 36, no. 2, pp. 243–274, 1934.

[137] D. Silva and F. R. Kschischang, "Fast encoding and decoding of gabidulin codes," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pp. 2858–2862, IEEE, 2009.

[138] A. Wachter-Zeh, V. Afanassiev, and V. Sidorenko, "Fast decoding of gabidulin codes," *Designs, codes and cryptography*, vol. 66, no. 1-3, pp. 57–73, 2013.

[139] W. Stein *et al.*, *Sage Mathematics Software (Version 8.2)*. The Sage Development Team, 2018. `http://www.sagemath.org`.

Contributions

1. **Pascal Lefèvre**, Philippe Carré et Philippe Gaborit, *A new blind color watermarking using a psychovisual and quantization approaches* International Conference on Image Processing (ICIP) IEEE, Pékin 2017.
2. **Pascal Lefèvre**, Philippe Carré et Philippe Gaborit, *Une approche psychovisuelle pour le tatouage des images couleur* Groupement de Recherche en Traitement du Signal et de l'Image (GRETSI), Juan-les-pins 2017.
3. **Pascal Lefèvre**, Philippe Carré et Philippe Gaborit, *Tatouage et codes en métrique rang (best paper award)* COmpression et REprésentation des Signaux Audiovisuels (CORESA), Caen 2017.
4. **Pascal Lefèvre**, Philippe Carré et Philippe Gaborit, *Watermarking and rank metric codes* International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Calgary Canada 2018.
5. **Pascal Lefèvre**, Philippe Carré et Philippe Gaborit, *Application of rank metric codes in digital image watermarking (selected paper, under review)* Elsevier Signal Processing : Image Communication.