Détection statistique d'information cachée dans des images naturelles

Cathel Zitzmann

Thèse
de doctorat
de l'UTT

Cathel ZITZMANN

Détection statistique d'information cachée dans des images naturelles

Spécialité :
Optimisation et Sûreté des Systèmes

2013TROY0012

Année 2013

utt
université de technologie
Troyes

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Cathel ZITZMANN

le 24 juin 2013

Détection statistique d'information cachée dans des images naturelles

JURY

M. J.-L. DUGELAY	PROFESSEUR EURECOM	Président
M. P. BAS	CHARGE DE RECHERCHE CNRS	Examinateur
M. C. DELPHA	MAITRE DE CONFERENCES - HDR	Rapporteur
M. L. FILLATRE	PROFESSEUR DES UNIVERSITES	Directeur de thèse
M. I. NIKIFOROV	PROFESSEUR DES UNIVERSITES	Directeur de thèse
M. W. PUECH	PROFESSEUR DES UNIVERSITES	Rapporteur

Personnalité invitée

M. F. DELOST	MINISTERE DE L'INTERIEUR

Remerciements

Cette thèse a été menée à son terme grâce à l'aide de nombreuses personnes : il me sera difficile de remercier tout le monde.

Je souhaite tout d'abord remercier mes directeurs de thèse, M. Igor Nikiforov, Professeur des Universités de l'Université de Technologie de Troyes et M. Lionel Fillatre, Professeur des Universités de l'Université de Nice Sophia-Antipolis. Je leur suis reconnaissante pour le temps conséquent qu'ils m'ont accordé ainsi que pour leurs qualités pédagogiques et scientifiques. J'ai beaucoup appris à leurs côtés et leur adresse toute ma gratitude.

Claude Delpha, Maître de Conférences au L2S, et William Puech, Professeur des Universités au LIRMM, m'ont fait l'honneur d'être rapporteurs de ma thèse. Je les remercie pour avoir pris le temps de lire mes travaux, ainsi que pour leurs remarques judicieuses.

Je tiens également à remercier Jean-Luc Dugelay, Professeur à EURECOM et président du jury de thèse, ainsi que Patrick Bas, chargé de recherche CNRS au LAGIS, d'avoir accepté d'examiner ma thèse.

Merci à François Delost, responsable du pôle technique de la DGPN, membre du projet RIC, pour sa présence dans le jury lors de ma soutenance de thèse.

Ce travail a été réalisé à l'UTT, au sein de l'équipe de Modélisation et Sûreté des Systèmes (LM2S) de l'Institut Charles Delaunay (ICD). Je remercie tous les membres du LM2S pour leur accueil, et plus particulièrement Mari-José Rousselet et Véronique Banse, secrétaires du pôle ROSAS, qui ont contribué grandement au bon déroulement de cette thèse, merci à elles.

Différentes personnes ont participé au projet RIC : je tiens à remercier Rémi Cogranne, Florent Retraint et Philippe Cornu, Léopold et Stéphane, nos stagiaires, ainsi que Sabrina et Laurent, nos ingénieurs.

Mes remerciements vont également toutes les personnes que j'ai rencontrées au cours de ces années de thèse, pour les nombreuses discussions et bons moments passés ensemble, que ce soit autour d'un café ou d'un thé, en conférence, lors des soirées Ellidoc, au water-polo...

Merci à Laurent, Léa, Mahdi, Sabrina et Céline pour leur soutien, leur écoute, et leur présence pendant ces années.

Merci à Léa, Mahdi et Slim pour leur temps, leurs remarques et leurs conseils pour que ma soutenance se déroule au mieux.

Un grand merci à mère pour toutes les heures qu'elle a consacrées à la relecture de cette thèse, et à la préparation de l'excellent pot de thèse (on m'en parle encore !).

Enfin, j'adresse mille mercis à mes parents, mes frères, mes amis et mon compagnon, qui ont toujours cru en moi, qui m'ont soutenue et supportée durant cette thèse... et qui le font depuis bien longtemps !

Résumé

Résumé

La nécessité de communiquer de façon sécurisée n'est pas chose nouvelle : depuis l'antiquité des méthodes existent afin de dissimuler une communication. La cryptographie a permis de rendre un message inintelligible en le chiffrant, la stéganographie quant à elle permet de dissimuler le fait même qu'un message est échangé. Cette thèse s'inscrit dans le cadre du projet "Recherche d'Informations Cachées" financé par l'Agence Nationale de la Recherche, l'Université de Technologie de Troyes a travaillé sur la modélisation mathématique d'une image naturelle et à la mise en place de détecteurs d'informations cachées dans les images. Ce mémoire propose d'étudier la stéganalyse dans les images naturelles du point de vue de la décision statistique paramétrique. Dans les images JPEG, un détecteur basé sur la modélisation des coefficients DCT quantifiés est proposé et les calculs des probabilités du détecteur sont établis théoriquement. De plus, une étude du nombre moyen d'effondrements apparaissant lors de l'insertion avec les algorithmes F3 et F4 est proposée. Enfin, dans le cadre des images non compressées, les tests proposés sont optimaux sous certaines contraintes, une des difficultés surmontées étant le caractère quantifié des données.

Abstract

The need of secure communication is not something new : from ancient, methods exist to conceal communication. Cryptography helped make unintelligible message using encryption, steganography can hide the fact that a message is exchanged. This thesis is part of the project "Hidden Information Research" funded by the National Research Agency, Troyes University of Technology worked on the mathematical modeling of a natural image and creating detectors of hidden information in digital pictures. This thesis proposes to study the steganalysis in natural images in terms of parametric statistical decision. In JPEG images, a detector based on the modeling of quantized DCT coefficients is proposed and calculations of probabilities of the detector are established theoretically. In addition, a study of the number of shrinkage occurring during embedding by F3 and F4 algorithms is proposed. Finally, for the uncompressed images, the proposed tests are optimal under certain constraints, a difficulty overcome is the data quantization.

Table des matières

Notations et abréviations

Notation	Signification
JPEG	Joint Photographic Expert Group,
DCT	Discrete Cosine Transform,
LSB	Least Significant Bit (bit de poids faible)
PP	Plus Puissant,
UPP	Uniformément le Plus Puissant,
AUPP	Asymptotiquement Uniformément le Plus Puissant,
LAUPP	Localement Asymptotiquement Uniformément le Plus Puissant,
RV	Rapport de Vraisemblance,
RVG	Rapport de Vraisemblance Généralisé,
MV	Maximum de Vraisemblance,
TIFF	Tag(ged) Image File Format,
BMP	BitMaP,
NEF	Nikon Electronic Format,
CCD	Charge-Coupled Device,
CMOS	Complementary Metal Oxide Semiconductor,
CFA	Color Filter Array,
ISO	International Organization for Standardization,
CCITT	Consultative Committee for International Telegraph and Telephone,
ITU	International Telecom Union,
$\mathbb{N}$	Ensemble des entiers naturels,
$\mathbb{R}$	Ensemble des réels,
$\mathbb{R}^n$	Espace vectoriel réel de dimension n,
$\mathbb{E}(\cdot)$	Espérance mathématique,
$\mathrm{V}ar(\cdot)$	Variance,
$\mathcal{H}_0$	Hypothèse nulle (medium sain),
$\mathcal{H}_1$	Hypothèse alternative (medium stéganographié),
δ	Test statistique,
$\beta(\delta)$	Puissance du test δ,
α	Probabilité de fausse alarme,
$\mathcal{K}_\alpha$	Classe des tests dont la probabilité de fausse alarme est bornée par α,
h	Seuil de décision,
$\Lambda(.)$	Rapport de vraisemblance,

P_θ	Distribution statistique de paramètre θ,
f_θ	Densité de probabilité de la distribution P_θ,
q_θ	Densité de probabilité de la distribution P_θ après quantification,
$\mathcal{B}(1,p)$	Distribution de Bernoulli de paramètre p,
$\mathcal{B}(n,p)$	Distribution binomiale de paramètres n et p,
$\mathcal{N}(\mu,\sigma^2)$	distribution gaussienne de moyenne μ et d'écart-type σ,
$\Phi(\cdot)$	Fonction de répartition de la loi de distribution normale centrée réduite $\mathcal{N}(0,1)$,
b	Paramètre d'échelle de la distribution laplacienne,
$\mathrm{Lap}(b,\Delta)$	Distribution laplacienne de paramètre b et quantifiée par Δ,
$\xrightarrow{\mathcal{L}}$	Convergence en loi,
$C_n = (c_1,\ldots,c_n)^T$	Vecteur représentant un medium de couverture de longueur n,
$Z_n = (z_1,\ldots,z_n)^T$	Vecteur représentant le medium analysé de longueur n,
n	Entier naturel représentant la longueur d'un vecteur,
$V = \{V_1,\ldots,V_{64}\}$	Ensemble formé des 64 vecteurs de coefficients DCT,
n_k	nombre de coefficients utilisables (i.e. différents de 0 et 1),
$V_k = (v_{k,1},\ldots,v_{k,n_k})^T$	Vecteur composé des n_k coefficients DCT utilisables de la k-ième fréquence,
$\bar{v}_{k,i}$	Coefficient $v_{k,i}$ avec le LSB inversé,
R	Taux d'insertion, $R \in [0,1]$
$\tilde{R}$	Taux d'insertion réel,
$Q_1[.]$	Opération de quantification uniforme (partie entière) à 2^q niveaux de reconstruction,
$Q_2[.]$	Opération de quantification uniforme à 2^{q-1} niveaux de reconstruction,
$\mathrm{signe}(x)$	Fonction qui vaut -1 si x est négatif, 1 si x est positif,
v. a.	Variable aléatoire,
M	v. a. discrète représentant le bit du message inséré, $M \in \{0,1\}$,
X	v. a. discrète représentant la valeur du coefficient DCT utilisé, $X \in [-1023,1024]$,
Y	v. a. discrète représentant la valeur du coefficient DCT après insertion, $Y \in [-1023,1024]$,
τ_1	v. a. discrète qui décrit le nombre de bits insérés successivement avant l'apparition du premier effondrement, $\tau_1 \in \mathbb{N}$,
τ_2	v. a. discrète qui décrit le nombre d'effondrements successifs qui peuvent apparaître, $\tau_2 \in \mathbb{N}$,
$\mathrm{Nb}_{\mathrm{eff}}$	v. a. qui représente le nombre total d'effondrements qui peuvent apparaître lors de l'insertion,
$S = \tau_1 + \tau_2$	v. a. discrète qui représente une séquence, i.e. décrit la durée au bout de laquelle un effondrement est résolu, $S \in \mathbb{N}$.

Chapitre 1

Introduction générale

1.1 Contexte général

La nécessité de communiquer de façon sécurisée n'est pas chose nouvelle : depuis l'antiquité des méthodes existent afin de dissimuler une communication. La cryptographie a permis de rendre inintelligible un message en le chiffrant, la stéganographie quant à elle permet de dissimuler le fait même qu'un message est échangé.

Le problème des prisonniers introduit par G. Simmons [76], représenté sur le schéma 1.1, permet d'introduire les notions de stéganographie et de stéganalyse. Alice et bob sont deux prisonniers enfermés dans des cellules éloignées, qui souhaitent mettre en place un plan d'évasion. Ils sont autorisés à communiquer et Wendy, la gardienne, peut intercepter les communications. Wendy fera cesser toute communication si elle décèle quelque chose de suspect. Pour communiquer de manière discrète, Alice insère un message, qui peut être préalablement chiffré, dans un conteneur (image, son, etc.). Lorsqu'il récupère le media, Bob peut extraire le message caché à l'aide d'une clé connue par eux seuls. Alice et Bob ont recours à la stéganographie. Le rôle de Wendy (également appelée Eve) est de détecter si un message est contenu dans le media échangé par Alice et Bob. Il s'agit là de stéganalyse.

Dans le cadre du projet "Recherche d'Informations Cachées" (RIC) financé par l'Agence Nationale de la Recherche (ANR), et regroupant divers partenaires tels que l'Institut de Recherche Criminelle de la Gendarmerie Nationale, le Centre Technique d'Assistance et Thalès Communications, l'Université de Technologie de Troyes a travaillé sur la modélisation mathématique d'une image naturelle [7] et à la mise en place de détecteurs d'informations cachées dans les images. Cette thèse fait l'objet du second point.

L'objet de cette thèse est donc d'étudier les méthodes d'insertion et les media susceptibles de contenir des informations cachées. Cela afin de mettre en place des détecteurs statistiques fiables d'informations cachées dans les images, notamment dans les images compressées au format JPEG, en se basant sur un modèle statistique paramétrique des données quantifiées.

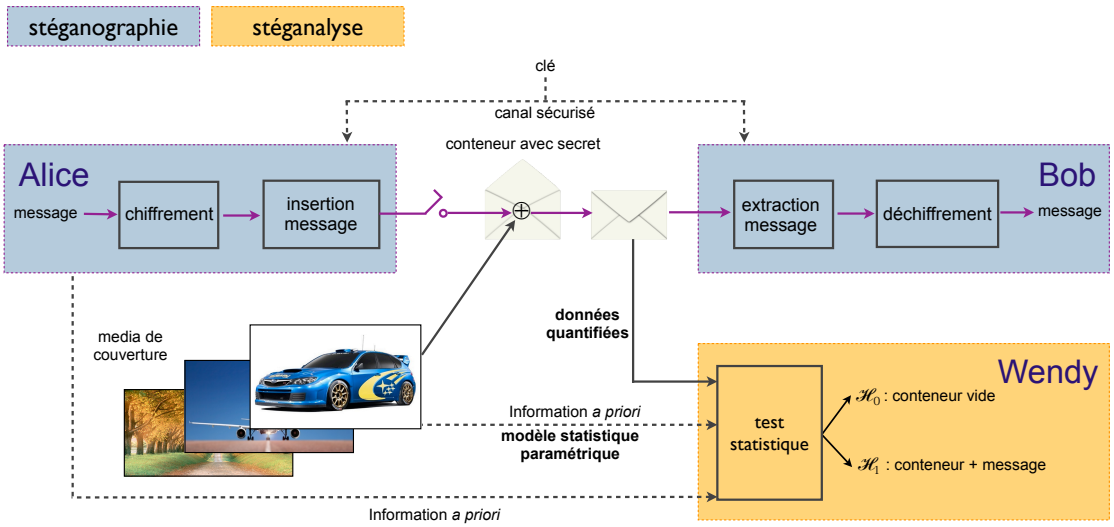


FIGURE *1.1* – *Problème des prisonniers.*

1.2 Organisation du manuscrit

Le chapitre 2 présente une introduction aux méthodes d'insertion dans des images ainsi qu'aux méthodes de détection existantes. Ainsi la notion de stéganographie est introduite dans la section 2.1 en exposant les principales méthodes d'insertion dans les bits de poids faible des images naturelles et en particulier dans les images JPEG. Ensuite la stéganalyse dont le but est de déceler la présence d'un message caché dans un medium est abordée dans la section 2.2, où sont présentés des algorithmes de détection. Enfin, la section 2.3 présente les outils statistiques nécessaires à la construction de détecteurs d'informations cachées et discute du choix des outils en fonction des contraintes fixées.

Après avoir introduit les outils de décision statistique, il est nécessaire de modéliser le medium dans lequel on souhaite détecter la présence d'informations cachées. Le chapitre 3 propose donc la modélisation de différents types d'images : les images *brutes*, directement issues d'un capteur, les images *non compressées* qui restituent fidèlement la scène, et les images *compressées* qui facilitent la transmission.

L'objectif du chapitre 4 est de présenter les détecteurs mis en place à l'aide des outils de décision statistique introduits dans le chapitre 2 et des modèles d'images présentés dans le chapitre 3. Dans une première section 4.1, un détecteur basé sur la modélisation des coefficients DCT quantifiés est proposé. Un test statistique basé sur la modélisation laplacienne est introduit et les calculs de ses probabilités sont établis puis une généralisation de ce test à une distribution quelconque est proposé. Ensuite, la section 4.2 présente une étude originale liée au phénomène d'effondrement qui se produit lors de l'utilisation des algorithmes F3, F4 et F5 présentés dans le chapitre 2. Une approche asymptotique permettant de calculer le nombre de 0 introduits par ce phénomène est présentée. Enfin, différents tests statistiques sont construits dans le domaine spatial en s'appuyant sur les outils présentés dans le chapitre 2 et leurs propriétés statistiques étudiées dans la section 4.3.

Dans le chapitre 5, consacré aux expérimentations numériques, les résultats théoriques sont validés par le biais de simulations Monte-Carlo, des tests sur données réelles sont également proposés. Tout cela met en évidence la pertinence de la méthodologie proposée dans les chapitres précédents.

Enfin ce manuscrit s'achève par une conclusion générale et des perspectives de travail possible sont proposées.

Chapitre 2

Stéganographie et Stéganalyse

Sommaire

Introduction

Ce chapitre présente une introduction aux méthodes d'insertion dans des images ainsi qu'aux méthodes de détection existantes. Ainsi la notion de stéganographie est introduite dans la section 2.1 en exposant les principales méthodes d'insertion dans les bits de poids faible des images naturelles et en particulier dans les images JPEG. Ensuite la stéganalyse, dont le but est de déceler la présence d'un message caché dans un medium, est abordée dans la section 2.2, où sont présentés des algorithmes de détection. Enfin, la section 2.3 présente les outils statistiques nécessaires à la construction de détecteurs d'informations cachées et discute du choix des outils en fonction des contraintes fixées.

2.1 Stéganographie - Méthodes d'insertion dans les LSBs

La *stéganographie* [1] est l'art de la communication invisible ou de la dissimulation. On l'utilise afin de dissimuler qu'un message caché est échangé, c'est la communication qui est dissimulée. C'est en ce sens que cela est très différent de la cryptographie où seul le contenu du message est incompréhensible.

Les origines de la stéganographie remontent à l'antiquité. Son utilisation est décrite par deux fois dans l'*Enquête* d'Hérodote [2]. Un premier passage relate qu'Aristagoras fit raser la tête de son plus fidèle esclave et y fit tatouer son message. Une fois les cheveux repoussés, l'esclave pouvait s'en aller transmettre le message. Le destinataire n'avait plus qu'à raser le crâne de l'esclave pour y voir apparaître le message. Un autre passage fait référence à Demarate, ancien roi de Sparte exilé en Perse, qui informa Sparte que les Perses préparaient une invasion de la Grèce en écrivant le message sur une tablette en bois puis en recouvrant celle-ci de cire. Cela permit de déjouer une attaque qui survint quatre ans plus tard.

Au cours des siècles, l'utilisation de la stéganographie a évolué. Dans la littérature, les acrostiches tels que la correspondance supposée entre George Sand et Alfred de Musset en sont un très bel exemple. Le message peut être décodé en lisant verticalement les initiales ou les premiers mots d'une suite de vers (voir annexe B).

De nos jours, les techniques de *stéganographie moderne* utilisent des fichiers numériques anodins tels que des images, des sons ou des vidéos en opérant de légers changements sur les fichiers.

De nombreuses techniques de stéganographie existent, cependant, nous étudierons principalement les méthodes d'insertion dans les bits de poids faibles. En effet, bien que simples, elles

1. Ce mot d'origine grecque est composé de "steganos", qui signifie *couvrir*, et de "graphō", *j'écris*.
2. Historien grec (484-420 av. J.-C.)

n'en restent pas moins les plus usitées. En effet, en décembre 2011, WetStone Technologies répertoriaient 836 outils de stéganographie dont 70% utilisaient l'insertion dans les LSBs [31]. Nous présenterons tout d'abord les méthodes associées au domaine spatial, puis les algorithmes dédiés au domaine fréquentiel.

2.1.1 Domaine spatial

L'insertion dans le domaine spatial concerne les formats d'image tels que TIFF, BMP, NEF, etc. Une image fixe non compressée est représentée par un tableau ou une suite de *pixels*. Notons $C_n = (c_1, \ldots, c_n)^T$ le vecteur représentant la suite des n pixels d'une image. Cette image peut être en noir et blanc, en niveaux de gris ou en couleurs. Une image couleur peut être codée sur 24 bits, chaque pixel c sera alors représenté par un triplet d'octets $c = (c_R, c_G, c_B)$ avec $c \in \{0, \ldots, 255\}^3$ (voir schéma Fig 2.1). Le premier octet codant la couleur rouge, le second le vert et le dernier le bleu.

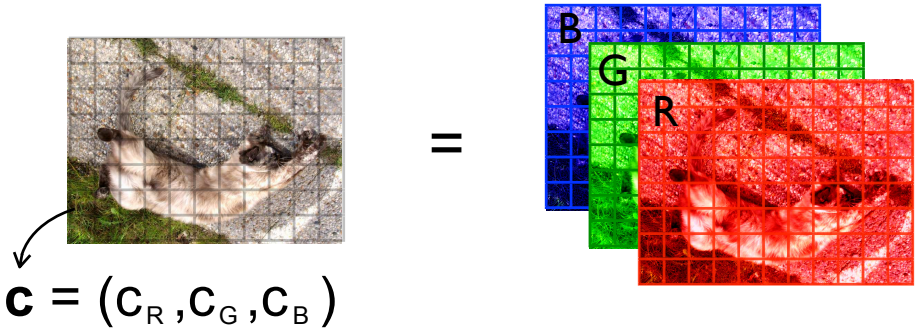


FIGURE 2.1 – *Image couleur composée des canaux rouge, vert et bleu.*

Dans ce manuscrit nous considérons des images en niveaux de gris, c'est à dire que nous considérons un seul canal.

2.1.1.1 Substitution des LSBs

La substitution des bits de poids faible (LSB pour Least Significant Bit) est la méthode d'insertion la plus simple : les LSBs des pixels sont remplacés par les bits du message. Grâce à sa simplicité, son implémentation facile et sa grande capacité d'insertion, de nombreux programmes utilisent cette méthode.

Pour insérer le message de longueur l noté $M = (m_1, \ldots, m_l)^T$, le bit de poids faible de chaque pixel est remplacé par un bit du message à insérer. La figure 2.2 représente la modification des pixels en fonction du bit inséré. Rappelons que les pixels pairs ont un LSB qui vaut 0 et les pixels impairs, un LSB égal à 1. Lors de l'insertion d'un 0, le pixel contaminé devient (ou

reste) pair. L'insertion fonctionne par paires de valeurs : un pixel valant 2, ne pourra prendre que les valeurs 2 ou 3 après insertion.

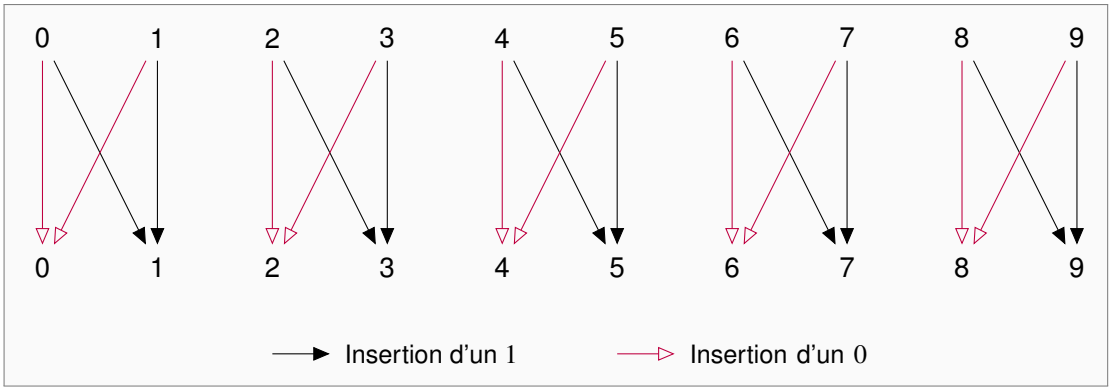


FIGURE 2.2 – *Modification des LSB des pixels par substitution.*

Dans le chapitre 4 sont présentés des tests statistiques permettant de détecter l'insertion par substitution des LSBs dans le domaine spatial.

2.1.1.2 Insertion par correspondance des LSBs

La méthode d'insertion par correspondance des LSBs, encore appelée *LSB matching* ou *±1 embedding* est une modification de la méthode par substitution des LSBs. L'insertion se fait également dans les LSBs des pixels, mais en incrémentant ou décrémentant aléatoirement la valeur du pixel.

2.1.2 Domaine fréquentiel

Nous nous intéressons à présent aux algorithmes utilisés pour des images compressées au format JPEG, c'est-à-dire à des algorithmes qui opèrent dans le domaine fréquentiel. Dans cette sous-section, nous introduirons brièvement le format JPEG, puis présenterons des algorithmes utilisés pour de telles images.

2.1.2.1 Introduction au format JPEG

Le schéma 2.3 décrit la chaîne de compression au format JPEG. Après des étapes facultatives de changement d'espace colorimétrique et de sous-échantillonage, les tableaux de pixels de chaque canal de couleur sont découpés en blocs de 8×8 pixels. La transformée en cosinus discrète est ensuite appliquée à chaque bloc. On obtient ainsi des blocs de 8×8 coefficients DCT. Chaque coefficient est ensuite quantifié en utilisant la table de quantification associée (présente dans l'en-tête du fichier). Le reste de la compression consiste à coder les tableaux de coefficients.

La perte d'information liée à la compression apparaît lors de l'étape de quantification. C'est pourquoi les algorithmes insèrent le message dans les coefficients DCT quantifiés (comme in-

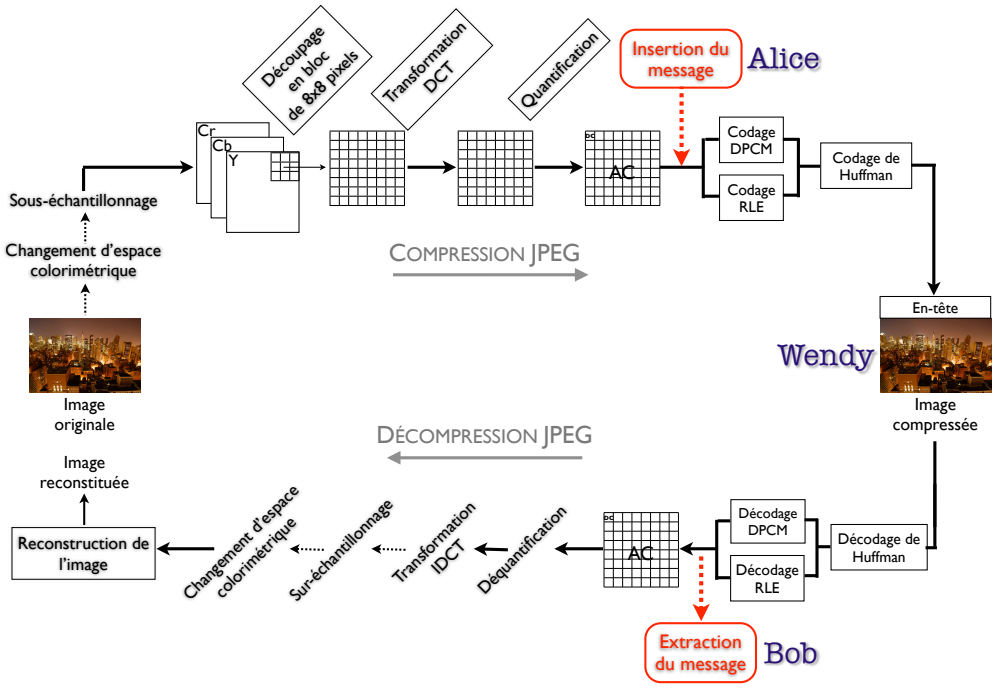


FIGURE 2.3 – *Schéma compression/décompression JPEG*

diqué sur le schéma 2.3). Ainsi, lors de l'insertion, tout se passe comme si une compression d'image était réalisée, mais après l'étape de quantification, les bits du message sont insérés dans les coefficients DCT quantifiés.

2.1.2.2 Algorithme Jsteg

Jsteg, créé par Derek Upham [79], est le premier algorithme qui insère des messages dans les fichiers JPEG compressés.

Le principe utilisé est la substitution des bits de poids faible des coefficients DCT quantifiés par les bits du message. Il est à noter que les coefficients valant 0 ou 1 ne sont pas utilisés car des coefficients non nuls apparaîtraient dans les hautes fréquences, ce qui est contraire au principe même de la compression JPEG (qui code efficacement les 0 présents dans les hautes fréquences), cela mènerait donc à des artefacts perceptibles et statistiquement détectables.

Méthode d'insertion

La méthode d'insertion consiste à remplacer le bit de poids faible de chaque coefficient DCT (différent de 0 ou 1) par un bit du message à insérer. Le format de la chaîne à insérer dans les LSBs avec l'outil Jsteg (Fig. 2.4) est défini comme suit :

A est codé sur 5 bits, et renseigne la longueur (en bits) du champ B,

B représente une suite de n bits $\in \{0, 31\}$ qui exprime la taille (en octets) du fichier à insérer,

C représente les bits du message à insérer.

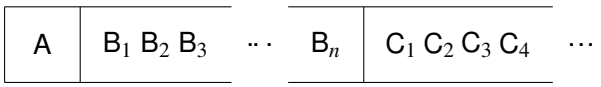


FIGURE 2.4 – *Format de la chaîne à insérer avec Jsteg.*

Il faut en effet pouvoir récupérer des informations sur la taille du message pour que l'extraction du message puisse être réalisée correctement.

MÉTHODE D'INSERTION DE L'ALGORITHME JSTEG :

1. Chaîne à insérer mise au bon format.
2. **Début de la compression JPEG** de l'image de couverture.
   Arrêt après l'étape de quantification.
3. **Substitution des LSB** des coefficients par les bits de la chaîne à insérer dans les coefficients DCT $\neq \{0, 1\}$.
4. Fin de la compression JPEG.

Nous pouvons observer sur la figure 2.5 que les modifications sont effectuées par paires de valeurs lors de la substitution des LSBs. Par exemple, pour la paire (2,3), si un 0 doit être inséré dans un coefficient dont la valeur est 2, il n'y a pas de modification car le LSB de 2 est 0. En revanche, si un 0 doit être inséré dans un 3, le bit de poids faible étant 1, le coefficient est modifié en substituant le LSB valant 1 par 0. La nouvelle valeur du coefficient est donc 2. De même, si un 1 doit être inséré dans un 2, le coefficient devient 3.

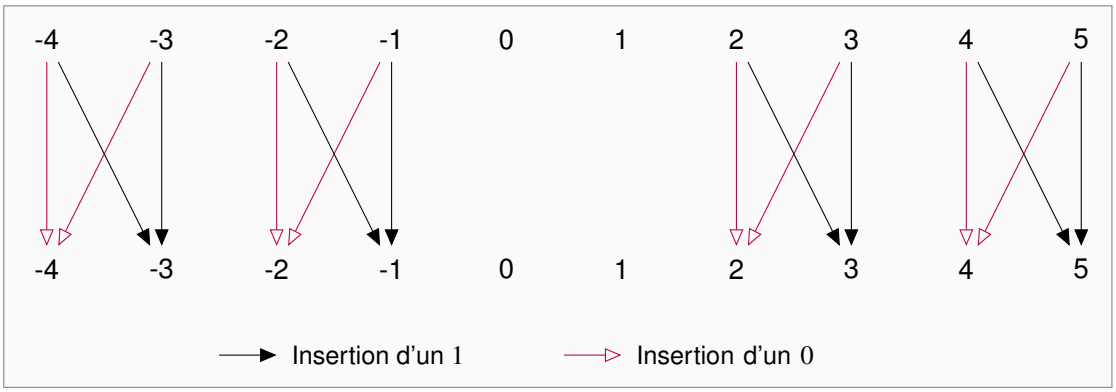


FIGURE 2.5 – *Modification des coefficients DCT par Jsteg*

Impact de Jsteg sur l'histogramme des coefficients DCT d'une image

Jsteg influence la fréquence d'occurrence des paires de coefficients ou paires de valeurs en uniformisant la distribution des paires de valeurs. Ceci est observable sur la figure 2.6. Les coefficients 0 et 1 sont grisés car n'étant pas modifiés, leur fréquence d'ocurrence ne varie pas.

La première version de Jsteg insère les bits du message séquentiellement, c'est à dire dans l'ordre d'apparition des coefficients DCT dans le vecteur, ce qui permet d'être détecté de manière

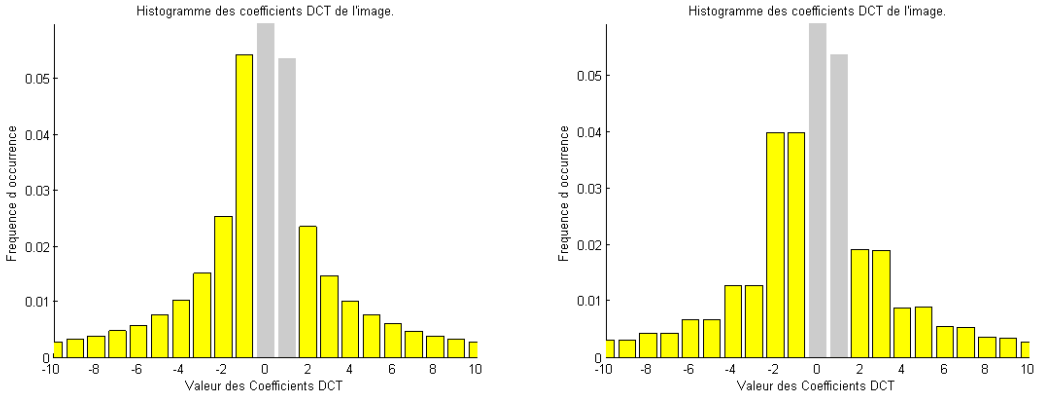


FIGURE 2.6 – *Histogramme des coefficients DCT d'une image saine (à gauche) et stéganographiée par Jsteg (à droite)*

efficace par l'attaque par histogramme [87]. Il résiste aux attaques visuelles, mais les attaques statistiques repèrent les changements opérés. Une version améliorée de cet algorithme permet l'insertion selon un chemin pseudo-aléatoire. C'est cette version que nous considérerons par la suite. Un algorithme de détection reposant sur la distribution des coefficients DCT est présenté dans le chapitre 4.

2.1.2.3 Algorithme F3

Pour remédier aux inconvénients de Jsteg, Andreas Westfeld [85] a donc apporté une amélioration avec l'algorithme F3.

Différence avec Jsteg

L'insertion d'un bit du message ne s'effectue plus par une substitution de LSB, mais par une décrémentation de la valeur absolue du coefficient DCT si le LSB ne correspond pas au bit à insérer (sauf si le coefficient vaut 0). Cela implique que, lorsqu'un 1 ou un -1 est décrémenté, un 0 est produit. Ce phénomène est appelé *effondrement* (*shrinkage* en anglais) et sera analysé plus précisément dans le chapitre 4. Le problème est que le destinataire de la stégo-image ne peut faire la différence entre un 0 non utilisé pour l'insertion et un 0 produit suite à l'insertion. Le bit du message concerné doit donc être réinséré dans le prochain coefficient disponible.

Sur la figure 2.7 est représentée l'insertion des bits 0 et 1 dans les coefficients DCT de l'image. Les traits en pointillés correspondent à la réinsertion suite à un effondrement.

Nous pouvons observer sur le graphique 2.8 la répartition des coefficients DCT après insertion d'un message. Un surplus de coefficients pairs apparaît du fait de la réinsertion d'un 0 à chaque fois qu'un effondrement se produit, et cela apparaît seulement pour des coefficients impairs. Or, un fichier JPEG est contient davantage de coefficients impairs utilisables (voir Fig. 2.6) que de coefficients pairs.

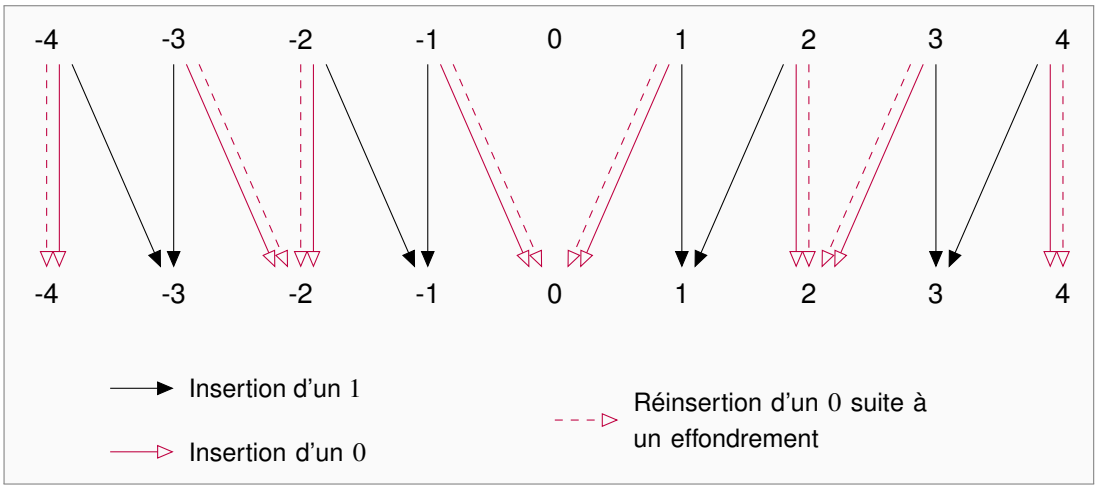


FIGURE 2.7 – *Modification des coefficients DCT par F3*

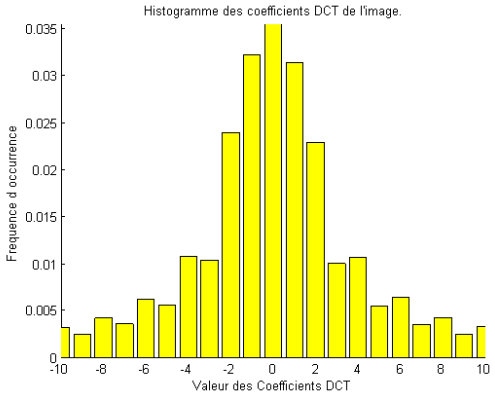


FIGURE 2.8 – *Histogramme des coefficients DCT après l'utilisation de F3.*

Avec F3, l'histogramme après stéganographie se rapproche davantage de l'histogramme initial, en ne produisant plus d'égalité des paires de valeurs, mais ceci a pu être amélioré par F4.

2.1.2.4 Algorithme F4

Différence avec F3

L'algorithme F4 corrige les faiblesses de F3 en redéfinissant le LSB pour les coefficients négatifs. Sur le tableau 2.1, nous avons représenté les bits de poids faible tels qu'ils sont définis.

Ainsi, un coefficient impair négatif aura un LSB qui vaut 0 pour F4. Cela permet d'avoir un transfert des valeurs mieux réparti entre les coefficients pairs et impairs (voir Fig. 2.9).

Nous pouvons voir sur la figure 2.10 que le fait de considérer différemment le LSB des coefficients DCT négatifs a permis de corriger l'histogramme stéganographié. Pour une même

Coefficient	Jsteg, F3		F4	
	pair	impair	pair	impair
positif	0	1	0	1
négatif	0	1	**1**	**0**

TABLE 2.1 – *Correspondance entre les coefficients et leur LSB.*

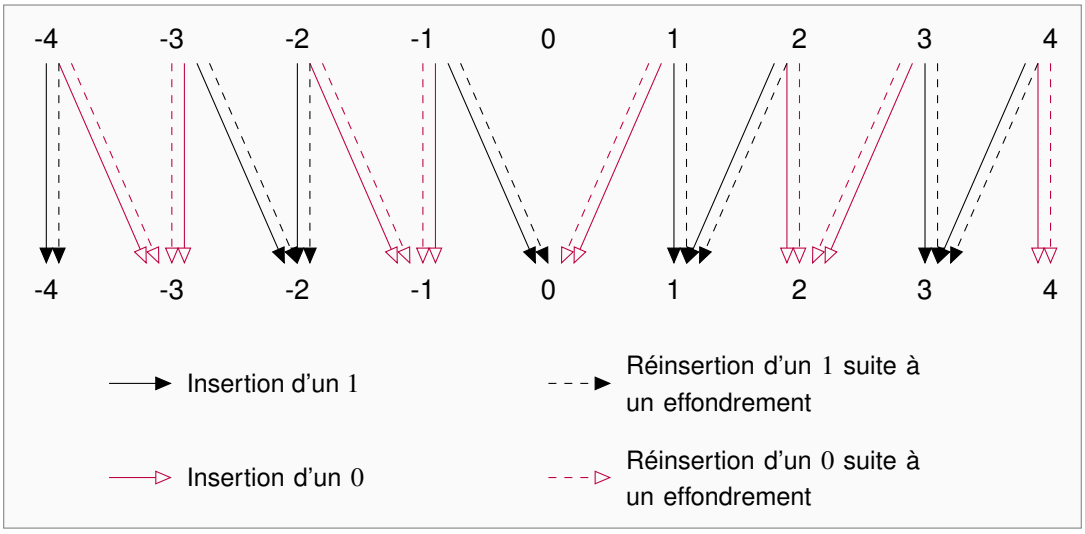


FIGURE 2.9 – *Codage des coefficients DCT pour F4.*

quantité d'information insérée, l'algorithme F4 est donc plus discret que F3.

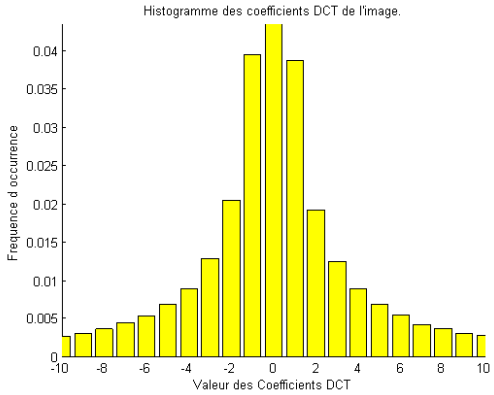


FIGURE 2.10 – *Histogramme des coefficients DCT après l'utilisation de F4.*

2.1.2.5 Algorithme F5

L'algorithme F5 a été inventé par Andreas Westfeld et a joué un rôle prépondérant dans le développement des schémas actuels de la stéganographie moderne. Un nouvel élément, le codage matriciel, est introduit dans la conception de l'algorithme. Il s'agit d'une technique qui

permet de réduire considérablement le nombre de modifications lors de l'insertion, notamment pour les taux d'insertion faibles. F5 conserve la méthode d'insertion de F4 (décrémentation de la valeur absolue et codage des LSB négatifs), mais utilise les codages matriciels afin de minimiser le nombre de changements à effectuer. Un même message peut donc être transmis en créant moins de distorsion qu'avec F4.

L'implémentation de l'algorithme est la suivante :

ALGORITHME F5 :

1. **Début de la compression JPEG** de l'image de couverture.
   Arrêt après l'étape de quantification.
2. **Initialisation du générateur de nombre pseudo-aléatoire** avec une clé.
3. **Calcul de la permutation** qui désignera le chemin à suivre pour l'insertion du message, en fonction d'un nombre aléatoire calculé précédemment et du nombre de coefficients DCT, à l'aide du générateur.
4. **Calcul des paramètres du codage de Hamming** en fonction de la capacité du medium de couverture et du message à insérer, afin de répartir au mieux les modifications.
5. **Insertion du message** avec le codage de Hamming.
6. Fin de la compression JPEG.

Calcul des paramètres du codage matriciel de Hamming [34]

Le codage linéaire utilisé a pour paramètres $[n, k, d]$ où :
- n est la longueur du mot de code,
- k sa dimension,
- d sa distance.

Dans l'algorithme F5, $d = 1$ et la longueur des mots de code n est définie par $n = 2^k - 1$. Le paramètre k du code est déterminé en fonction de la capacité C d'insertion du conteneur et de la longueur du message à insérer (voir Tableau Fig. 2.2).

Fridrich [28] propose une estimation de la capacité C :

$$C = Nb - \frac{Nb}{64} - Nb_{X=0} - Nb_{|X|=1} + 0.49 \times Nb_{|X|=1} \tag{2.1}$$

Le terme $-\frac{Nb}{64}$ vient du fait que les coefficients DCT DC ne sont pas utilisés pour l'insertion puisque leur modification provoquerait des changements trop visibles dans l'image. Le terme $-Nb_{X=0}$ correspond au fait que l'algorithme n'insère pas d'information dans les coefficients nuls. Enfin, $-Nb_{|X|=1} + 0.49 \times Nb_{|X|=1}$ correspond à l'estimation du nombre d'effondrements.

L'insertion par codage matriciel permet d'améliorer la discrétion de l'insertion en diminuant le nombre de changements nécessaires.

k	n	densité de changement	taux d'insertion	efficacité d'insertion
1	1	50.00%	100.00%	2
2	3	25.00%	66.67%	2.67
3	7	12.50%	42.86%	3.43
4	15	6.25%	26.67%	4.27
5	31	3.12%	16.13%	5.16
6	63	1.56%	9.52%	6.09
7	127	0.78%	5.51%	7.06
8	255	0.39%	3.14%	8.03
9	511	0.20%	1.76%	9.02

TABLE 2.2 – *Relation entre la densité de changement et le taux d'insertion.*

Les paramètres du code de Hamming sont donc $[n,k,d] = [2^k - 1, k, 1]$ puisque le but est de faire au plus un changement par groupe de coefficients. Le cas *sans codage matriciel* correspond au cas où $n = k = 1$.

Exemple :

Pour choisir les paramètres à utiliser, il suffit de faire comme suit ; supposons que nous voulions insérer un message de 5000 bits dans une image dont la capacité est 100000 bits. Le taux d'insertion est donc de $\frac{5000}{100000} = 5\%$, ce qui se situe entre $k = 7$ et $k = 8$. Si nous prenons des mots de code de taille 127, on peut utiliser $\lfloor \frac{100000}{127} \rfloor = 787$ mots de codes pour insérer $787 \times 7 = 5509$ bits. En revanche, en prenant $n = 255$, nous pouvons utiliser 392 mots de codes, mais nous pouvons seulement insérer 3136 bits.

Il faut donc choisir le k tel que le taux d'insertion est immédiatement supérieur au taux d'insertion calculé.

Insertion du message

Le message à insérer est découpé en paquets de k bits. L'insertion du message se fait de la manière suivante :

MÉTHODE D'INSERTION POUR F5 :

1. Remplir un tampon avec n coefficients DCT non nuls.
2. Calculer le syndrome s du tampon.
3. Sommer k bits du message avec s.
4. Si la somme est nulle, le tampon n'est pas modifié. Sinon la somme correspond à l'index du coefficient du tampon dont il faut décrémenter la valeur absolue.
5. Si un effondrement apparaît, retour à l'étape 3 en éliminant le 0 produit et en prenant un coefficient non nul supplémentaire dans le tampon.
   Sinon avancer aux prochains coefficients non nuls situés après le tampon.
   S'il y a encore des données à insérer, recommencer depuis l'étape 1.

On se place dans le cas d'un code $[n,k,d] = [3,2,1]$. Supposons que l'on veut insérer 2 bits x_1 et x_2 dans les 3 bits de poids faible a_1, a_2 et a_3.

$H = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ est la matrice de parité du code

L'étape 2. revient à faire le calcul suivant où s est le syndrome calculé :

$$s = H \times \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}, \quad s \in \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$$

L'étape 3. correspond au calcul suivant :

$$e = s + \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Pour savoir quel coefficient modifier, il suffit de regarder e .

– Si $e = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, alors il n'y a pas de modification à faire.
– Sinon, e est égal à la colonne de H dont il faut décrémenter la valeur absolue du coefficient correspondant.

Exemple

Voici un exemple du codage matriciel, afin d'observer comment il fonctionne. Après l'étape de quantification des coefficients DCT, les coefficients sont permutés, le code choisi est [3,2,1]. Nous allons donc insérer le message `Msg` dans la chaîne de coefficients `Coeff`.

```
Msg = [0 1 1 1 0 0 1 1]
Coeff = [5 0 0 2 3 -1 0 -3 0 1 -3 0 1 -3 1 -2 -2 1 0 1 3]
```

Avant d'insérer le message, nous allons :
– repérer les coefficients dans lesquels nous pourront coder de l'information,

– récupérer les LSBs des coefficients non-nuls.

Coefficients DCT	5	0	0	2	3	-1	0	-3	0	1	-3	1	-2	-2	1	0	1	3
LSB	1	×	×	0	1	0	×	0	×	1	0	1	1	1	1	×	1	1
Après insertion	5	0	0	1	3	-1	0	-3	0	1	-2	1	-2	-2	1	0	0	2

La matrice de parité H correspondant à notre code est : $H = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$.

Le message est découpé en segments de $k = 2$ bits : `Msg = 0 1 | 1 1 | 0 0 | 1 1`

Pour commencer, nous voulons insérer les bits `[0 1]` dans les LSBs `[1 0 1]` :

$$e_1 = H \times \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Les coefficients ne sont donc pas modifiés. Puis, nous voulons insérer les bits `[1 1]` dans les LSBs `[0 0 1]` et les coefficients ne sont toujours pas modifiés.

Ensuite, nous voulons insérer les bits `[0 0]` dans les LSBs `[0 1 1]` :

$$e_2 = H \times \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Le premier coefficient est donc modifié et sa valeur absolue décrémentée.

Ensuite, nous voulons insérer les bits `[1 1]` dans les LSBs `[1 1 1]` :

$$e_2 = H \times \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Le troisième coefficient est donc modifié et sa valeur absolue décrémentée. Un 0 est produit, il faut donc recommencer l'insertion : `[1 1]` insérés dans les LSBs `[1 1 1]`. C'est donc le troisième coefficient qui va être modifié.

2.1.2.6 nsF5

L'algorithme nsF5 (non-shrinkage F5) [32] proposé par Fridrich *et al.* est une amélioration de F5, dans lequel le phénomène d'effondrement est éliminé en utilisant un codage plus efficace (tel que les codes à papier mouillé par exemple) plutôt que de réinsérer le bit. De cette manière, le même message peut être inséré en faisant moins de modifications, la sécurité est ainsi meilleure. Comme les coefficients -1 et 1 sont les coefficients DCT non nuls les plus fréquent dans les images JPEG, nsF5 est une amélioration de F5 non négligeable en terme de capacité d'insertion.

2.1.2.7 OutGuess

OutGuess a été développé par Niels Provos [65, 64, 66] en réponse à l'attaque statistique du χ^2 [87] de Westfeld. Cet algorithme préserve l'histogramme des coefficients DCT de l'image et ne permet pas d'attaque utilisant les statistiques du premier ordre. En effet, il opère en deux temps afin de restaurer statistiquement l'image : tout d'abord, certains coefficients sont utilisés pour l'insertion, puis les coefficients restants sont modifiés afin de restaurer l'histogramme initial. Comme Jsteg, Outguess n'utilise pas les coefficients 0 et 1 et l'insertion se fait par substitution des LSBs. Bien que Outguess ne puisse être détecté par des statistiques de premier ordre, il a été stéganalysé avec succès à plusieurs reprises [29, 6].

2.1.2.8 JPHide&Seek

JPHS [50] est un algorithme développé par Allan Latham en 1998. Bien qu'il n'existe pas de publication sur son fonctionnement, en inspectant les images saines et les images stéganographiées, on peut déduire quelques mécanismes utilisés lors de l'insertion. Tout d'abord, l'insertion dans les coefficients positifs se fait par substitution des LSBs, alors que pour les coefficients négatifs, il s'agit de substitution du LSB inversé. L'insertion est donc symétrique par rapport à zéro et l'histogramme d'origine préservé. Ensuite, l'algorithme utilise également les coefficients DC. Pour des taux d'insertion petits, ce sont même les seuls coefficients à être modifiés. De plus, l'en-tête du message est de taille importante, ce qui crée une distorsion, même pour des taux d'insertion faibles, ce qui permet à certains détecteurs de les détecter.

2.1.2.9 StegHide

StegHide, algorithme créé par Stefan Hetzl en 2003 [36, 37], représente une approche par la théorie des graphes de la stéganographie. Tout comme JPHS, StegHide utilise les coefficients DC pour l'insertion et une en-tête de message importante, ce qui fait que l'image est considérée stégo même s'il s'agit d'un message de taille 0 bit qui est inséré.

2.1.2.10 MBS

Model-based steganography (MBS) est un algorithme créé par Phil Sallee en 2004 [71, 70]. Il essaie de préserver le modèle des coefficients DCT pour chaque fréquence. En particulier, pour chaque fréquence AC, les paramètres de la distribution de Cauchy généralisée sont estimés par la méthode du maximum de vraisemblance par rapport à l'histogramme des paires de coefficients qui sont invariantes par insertion. Une fois le modèle construit, les parties du message uniformément distribuées sont biaisées avec un décompresseur entropique de sorte à correspondre aux probabilités extraites du modèle.

Les algorithmes présentés ici ne sont pas une liste exhaustive des méthodes d'insertion existantes, le lecteur intéressé pourra compléter sa lecture par les livres [13, 26, 3].

2.2 Détection d'informations cachées

Cette section présente la stéganalyse ainsi que différentes méthodes de détection, en particulier celles auxquelles les détecteurs proposés dans le chapitre 4 seront comparés.

2.2.1 Introduction à la stéganalyse

Les notions de stéganographie et de stéganalyse sont très bien illustrées par le problème des prisonniers, formalisé par G. J. Simmons [76] en 1983. Alice et Bob sont deux prisonniers, enfermés dans des cellules éloignées, qui tentent de mettre au point un plan d'évasion. Leur seul moyen de communication est l'envoi de messages, dont le contenu doit être intelligible, par l'intermédiaire de la gardienne Wendy. Les prisonniers doivent alors dissimuler leur plan dans les messages échangés. L'évasion sera alors compromise si Wendy détecte une anomalie dans un des messages. La stéganographie est utilisée par Alice et Bob dans le but d'élaborer un plan d'évasion en toute discrétion. La stéganalyse quant à elle représente les moyens mis en oeuvre par Eve pour déceler la présence d'une communication secrète.

2.2.2 Méthodes de stéganalyse existantes

2.2.2.1 Zhang et Ping

En 2003, T. Zhang and X. Ping [90] ont proposé une méthode afin de détecter l'utilisation de Jsteg (version séquentielle et aléatoire).

Deux ensembles de valeurs L et R sont définis de la manière suivante, pour i un coefficient DCT :

$$L = \{i > 0 | i \text{ pair}\} \cup \{i < 0 | i \text{ impair}\},$$

$$R = \{i > 0 | i \text{ impair}\} \cup \{i < 0 | i \text{ pair}\}.$$

La proportion p de la capacité utilisé par Jsteg randomisé est :

$$p = \frac{1}{h(1)} \cdot \left(\sum_{r \in R} h(r) - \sum_{l \in L} h(l) \right), \tag{2.2}$$

avec $h(k)$ qui représente le nombre de coefficients DCT ayant pour valeur k.

2.2.2.2 Category Attack

L'*attaque par catégorie* est présentée en 2006 par Lee et al. [54, 53] Les classes sont numérotés et une version ajustée de l'histogramme qui exclue les valeurs 0 et 1 (qui ne sont pas utilisées par Jsteg) est créée :

$$h'(k) = \begin{cases} h(k) & \text{pour } k < 0, \\ h(k-2) & \text{pour } k > 1. \end{cases}$$

La proportion p de la capacité utilisée par Jsteg randomisé est estimée en utilisant la statistique s pour les catégories décalées (*shifted* en anglais) qui ne sont pas égalisées et induisent des catégories qui sont égalisées par l'insertion avec Jsteg :

$$s(k) = \frac{(h'(k) - h'(k+1))^2}{h'(k) - h'(k+1)}, \tag{2.3}$$

$$p = \frac{\sum_i s(2i) - \sum_i s(2i-1)}{\sum_i s(2i) + \sum_i s(2i-1)}. \tag{2.4}$$

2.2.2.3 Méthodes provenant du domaine spatial

Les méthodes suivantes proviennent du domaine spatial, c'est-à-dire qu'elles ont été adaptées de sorte à fonctionner dans le domaine spatial [86]. Afin d'utiliser les algorithmes du domaine spatial pour le domaine fréquentiel, un prétraitement des coefficients DCT est nécessaire. Soit x un vecteur comportant les coefficients DCT, il s'agit d'éliminer tous les coefficients 0 et 1 (car Jsteg ne les utilise pas), on obtient ainsi x' puis les valeurs sont modifiées de sorte à ce qu'elles soient toutes positives. Ainsi, soit d le vecteur contenant les nouvelles valeurs :

$$d_i = \begin{cases} x'_i - x_{\min} & \text{si } x'_i < 0, \\ x'_i - x_{\min} - 2 & \text{si } x'_i > 1, \end{cases} \tag{2.5}$$

$$\text{avec } x_{\min} = \begin{cases} \min x & \text{si min } x\text{est pair}, \\ \min x - 1 & \text{sinon.} \end{cases}$$

De même, notons d_{AC} le vecteur contenant seulement les coefficients AC.

JWS

L'attaque par stégo-image pondérée (Weighted Stego-image) de Fridrich et Goljan est également appliquée directement à la séquence d de l coefficients DCT normalisés. La proportion p de la capacité utilisée par Jsteg randomisé est estimée par :

$$p = \frac{2\sum_{i=3}^{l-2} q_i}{\sum_{i=3}^{l-2} \frac{1}{1+v_i}} \tag{2.6}$$

avec :

$$v_i = \frac{1}{3}((d_{i-2} - \mu_i)^2 + (d_{i-1} - \mu_i)^2 + (d_{i+!} - \mu_i)^2 + (d_{i+2} - \mu_i)^2),$$

$$\mu_i = \frac{1}{4}(d_{i-2} + d_{i-1} + d_{i+1} + d_{i+2}),$$

et

$$q_i = \begin{cases} \dfrac{1}{1+v_i}(\mu_i - d_i) & \text{si } d_i \text{ est pair,} \\ \dfrac{1}{1+v_i}(\mu_i - d_i) & \text{sinon.} \end{cases}$$

L'attaque JWS estime la valeur originale du coefficient q_i comme la moyenne locale de quatre valeurs voisines. Les valeurs originales pourraient être estimées plus précisément à partir des voisins qui n'ont pas été modifiés stéganographiquement. Cependant, toutes ces valeurs inchangées ont été supprimées lors du prétraitement.

JSPA

L'attaque par paires d'échantillons, développée par Dumitrescu et al. [15], est directement applicable à une séquence de coefficients DCT normalisée. Soit dmax le coefficient pair maximal incrémenté par 1. Les deux ensembles suivants classifient les paires d'échantillons (u,v) d'éléments consécutifs (sans chevauchement) de la séquence d :

$$A = \{(u,v)|u \geq v, u \text{ pair}\} \cup \{(u,v)|u < v, u \text{ impair}\},$$

$$B = \{(u,v)|u < v, u \text{ pair}\} \cup \{(u,v)|u \geq v, u \text{ impair}\}.$$

Nous définissons deux histogrammes

$$h_0(|u_0 - v_0|) \text{ pour } (u_0, v_0) \in A \text{ et}$$

$$h_1(|u_1 - v_1|) \text{ pour } (u_1, v_1) \in B$$

La proportion p estimée de la capacité utilisée par Jsteg randomisé est la plus grande des deux racines des équations quadratiques suivantes :

$$0 = ap^2 + bp + c,$$

$$\text{où } c = \sum_{i=0}^{j-1} h_0(2i) - h_1(2i+1),$$

$$b = h_0(0) + h_1(0) - \frac{h_0(2j+2) + h_1(2j+2)}{2} + \frac{c}{2}, \text{ et}$$

$$a = \sum_{i=0}^{l} \frac{h_0(i) + h_1(i)}{2} - \frac{h_0(2j+2+i) + h_1(2j+2+i)}{4}.$$

JPairs

L'analyse par paires proposée par Fridrich et al. [30] évalue le nombre de paires homogènes (00 ou 11) et inhomogènes (01 ou 10) dans une séquence binaire. Ces séquences binaires peuvent être obtenues suite à un prétraitement appliqué à une séquence de coefficients DCT. Soit $d_{\max}$ le coefficient pair maximal incrémenté de 1 :

$$d_{\max} = \begin{cases} \max d + 1 & \text{si max } d\text{est pair,} \\ \max d & \text{sinon.} \end{cases}$$

L'attaque JPairs crée une séquence binaire z pour toutes les paires de valeurs de coefficients DCT normalisés (0, 1), (2, 3), $\cdots$, $(d_{\max} - 1, d_{\max})$. Pour construire ces séquences, la séquence d est scannée $\frac{dmax}{2}$ fois. Lors du premier passage, 0 est accolé à la première séquence initialement vide dès qu'un 0 est rencontré et un 1 est accolé dès qu'un 1 est trouvé. Lors du second passage, un 0 (resp. 1) est ajouté dès qu'un 2 (resp. 3) est rencontré, et ainsi de suite. Cela est réalisé pour chaque paire. De même, une séquence z' est créée pour les paires décalées (1,2), (3, 4), . . . , (dmax, 0). Soit l la longueur de d. La proportion p de la capacité utilisée par Jsteg randomisé est estimée par :

$$p = 1 - \sqrt{1 - \frac{q - q'}{\frac{l-1}{2} - \bar{q}}}, \tag{2.7}$$

$$\text{où } q = \sum_{i=1}^{l-1} |z_i - z_{i+1}|,$$

$$q' = \sum_{i=1}^{l-1} |z'_i - z'_{i+1}|,$$

$$\text{et } \bar{q} = \sum_{j=1}^{l} \frac{1}{2^j} \sum_{i=1}^{l-j} |z'_i - z'_{i+j}|.$$

2.2.3 Méthodes basées sur la détection par apprentissage

La stéganalyse est un problème reposant fondamentalement sur la notion de tests d'hypothèses, étant donné qu'un stéganalyste souhaite déterminer si le medium analysé contient des informations cachées. On peut alors considérer deux classes (images saines et images stéganographiées). Lorsqu'un modèle précis du medium n'est pas disponible, le problème peut être vu comme un problème de classification et être résolu en utilisant des méthodes d'apprentissage supervisé. Ces méthodes consistent à apprendre au système à classer les données. Lors d'une première phase, un oracle étiquette les données issues de la base d'apprentissage $\mathcal{B}_a$, et détermine un modèle des données étiquetées. Ensuite, la seconde phase consiste à prédire l'étiquette de nouvelles données issues de la base de test $\mathcal{B}_t$, en utilisant le modèle préalablement appris.

De nombreuses publications proposent des méthodes basées sur de l'apprentissage supervisé (ou analyse discriminante) :

- un *classifieur binaire* permet de distinguer deux classes : la classe des images saines et celle des images stéganograhiées [60, 46],
- un *classifieur à k classes* permet de décider quel est l'algorithme utilisé pour stéganographié l'image parmi k, par exemple [62]. Parmi toutes les techniques utilisées, on trouve les séparations linéaires, les méthodes à noyaux (SVM par exemple) [60] ou encore les méthodes par ensemble [46].

Le principe de base des machines à vecteurs de support est présenté dans cette sous-section de manière brève. Pour une introduction plus complète, le lecteur peut se référer à [80].

Considérons le problème de classification binaire. Soit $\mathcal{X}$ un ensemble arbitraire non vide et $\mathcal{Y} = \{-1, +1\}$ l'ensemble des étiquettes. Soit $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ une paire de variables aléatoires distribuée selon la loi P inconnue. L'ensemble d'apprentissage $\mathcal{B}_a = \{(x_i, y_i) | (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}, i \in \{1, \ldots, l\}\}$ consiste en l réalisations indépendantes et identiquement distribuées selon P. Le but est de construire une fonction $f : \mathcal{X} \to \mathcal{Y}$ qui prédit Y à partir de X. Pour construire cette fonction f, le critère d'une faible probabilité d'erreur est choisi. Le risque de f est défini par :

$$R(f) = P(f(X) \neq Y) = \mathbb{E}\left[\mathbb{1}_{f(X) \neq Y}\right].$$

Résoudre ce problème revient à trouver la fonction t qui minimise le risque fonctionnel :

$$R(t) = \inf_f R(f).$$

Comme P est inconnu, le risque ne peut pas être calculé directement et la valeur de t en chaque point n'est pas connue. En revanche, le risque empirique

$$R_l(f) = \frac{1}{l} \sum_{i=1}^{l} \mathbb{1}_{f(X_i) \neq Y_i}$$

est un critère de sélection pour t.

Ensemble linéairement séparable

Supposons que $\mathcal{X} = \mathbb{R}^n$

Définition 2.1. *On dit qu'un ensemble test est linéairement séparable s'il existe* $\mathbf{w} \in \mathbb{R}^n$ *et* $b \in \mathbb{R}$ *tels que la fonction de décision*

$$f(\mathbf{x}) = signe((\mathbf{x} \cdot \mathbf{w}) + b), \tag{2.8}$$

possède un risque empirique nul sur l'ensemble d'apprentissage.

La fonction $f(\mathbf{x})$ classe le point $\mathbf{x} \in \mathbb{R}^n$ selon le côté de l'hyperplan $(\mathbf{x} \cdot \mathbf{w}) + b$ où il se situe. Si la base d'apprentissage est linéairement séparable, il existe une infinité de fonctions qui classent parfaitement l'ensemble avec un risque empirique nul.

Classifieur SVM à marge maximale

Les SVM cherchent l'hyperplan séparateur qui maximise la distance entre les deux classes de sorte à minimiser la probabilité de mauvaise classification d'un élément qui ne serait pas dans l'ensemble. Cet hyperplan séparateur optimal, noté f^*, est unique. Il peut être trouvé en résolvant le problème d'optimisation suivant :

$$[\mathbf{w}^*, b^*] = \arg \max_{\mathbf{w}\in\mathbb{R}^n, b\in\mathbb{R}} \{\min\{\|\mathbf{x} - \mathbf{x}_i\|, \mathbf{x} \in \mathbb{R}^n, (\mathbf{x}\cdot\mathbf{w}) + b = 0, i \in \{1,\ldots,l\}\}\}$$

sous la contrainte

$$y_i((\mathbf{x}\cdot\mathbf{w}) + b) > 0, \forall i \in \{1,\ldots,l\}.$$

Ce problème d'optimisation peut être reformulé par :

$$[\mathbf{w}^*, b^*] = \arg \min_{\mathbf{w}\in\mathbb{R}^n, b\in\mathbb{R}} \frac{1}{2}\|\mathbf{w}\|^2 \quad s.t. \quad y_i((\mathbf{x}\cdot\mathbf{w}) + b) \geq 1, \forall i \in \{1,\ldots,l\}.$$

La figure 2.11 représente un classifieur à marge maximale où les vecteurs supports sont entourés et l'hyperplan séparateur optimal est représenté par la droite d'équation $\mathbf{x}\cdot\mathbf{w} + b = 0$.

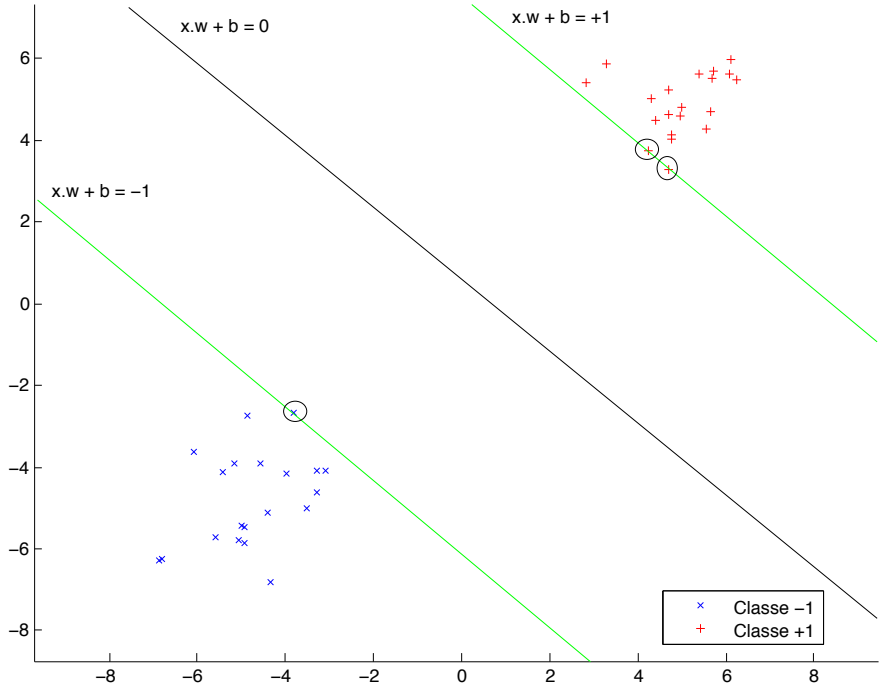


FIGURE 2.11 – *Classifieur linéaire à marge maximale*

Classifieur SVM à marge douce

En pratique, les données réelles ne sont pas linéairement séparables, cela est dû notamment à la présence de bruit dans les observations. Si le risque empirique ne peut être nul, on souhaite

tout de même minimiser le nombre d'erreurs de classifications commises. La fonction f^* doit donc minimiser le risque empirique et avoir une distance maximale entre les données à classifier. Le problème d'optimisation à résoudre peut être ramené à :

$$[\mathbf{w}^*, b^*] = \arg\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i \tag{2.9}$$

$$s.t. \quad y_i((\mathbf{x}\cdot\mathbf{w}) + b) \geq 1 - \xi_i, \forall i \in \{1,\ldots,l\} \text{ et } \xi_i \geq 0, \forall i \in \{1,\ldots,l\}$$

où C est une constante de régularisation qui pondère le coût des erreurs sur les contraintes et ξ_i une variable qui mesure la distance des données mal classifiées de l'hyperplan séparateur. Nous parlons alors de classifieur SVM à marge douce.

Ce problème est résolu en utilisant la méthode du Lagrangien. La solution est donnée par :

$$\hat{\alpha} = \arg\max_{\alpha}\left(\sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)\right) \tag{2.10}$$

où les α_i sont les multiplicateurs de Lagrange satisfaisant les contraintes $\sum_{j=1}^{l}\alpha_i y_i = 0$ et $C \geq \alpha_i \geq 0$.

Classification non linéaire

De nombreux ensembles de données ne sont pas linéairement séparables. Ce problème est donc contourné en trouvant une transformation Φ de l'espace d'entrée vers un espace de plus grande dimension dans lequel les données sont linéairement séparables [73, 80].

La fonction de décision dans ce nouvel espace est :

$$f(\mathbf{x}) = \text{signe}((\Phi(\mathbf{x})\cdot\mathbf{w}) + b), \text{ avec } \Phi : \mathbb{R}^n \mapsto \mathbb{R}^{n'}, n' > n \tag{2.11}$$

Les méthodes vues précédemment (cas linéaire) restent applicables après projection des données d'apprentissage dans le nouvel espace :

$$\hat{\alpha} = \arg\max_{\alpha}\left(\sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j(\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j))\right) \tag{2.12}$$

Une fonction appelée *noyau* et exprimant le produit scalaire dans le nouvel espace est définie : $\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j) = (\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j))$

L'utilisation des SVM est très répandue en stéganalyse. Cependant, quelques limites s'imposent : il est difficile de sélectionner les bonnes caractéristiques, et leur nombre important peut générer des difficultés liées à la *malédiction de la dimension*, fléau de l'apprentissage supervisé. De plus, la complexité des images naturelles implique que la construction d'une base d'apprentissage suffisamment exhaustive pour être représentative de la réalité est compliquée. Enfin, les

probabilités d'erreurs sont empiriques, il n'est donc pas possible de respecter une contrainte sur la probabilité de fausse alarme.

Dans le cadre du projet RIC, la principale contrainte est le respect d'une probabilité de fausse alarme. Cela impose donc de construire des fonctions de décision dont les probabilités d'erreur sont maîtrisables. La théorie de la décision statistique paramétrique se prête donc parfaitement au problème posé initialement.

2.3 La stéganalyse du point de vue de la théorie de la décision - Outils de la décision statistique paramétrique

Comme le montre le schéma des prisonniers, un problème de décision se présente à Eve : "l'image analysée est-elle saine ou stéganographiée ?". Afin de répondre à cette question, il est nécessaire de définir les deux ensembles d'images (saine ou stéganographiée), et de manière plus générale, définir la notion d'hypothèse statistique, de test statistique, etc. Cette section présente tous les outils et résultats de la théorie de la décision statistique sur lesquels reposent les travaux présentés dans ce manuscrit, notamment dans le chapitre 4.

Lorsque Eve intercepte une image, elle doit décider si l'image analysée, que l'on note $Z_n = (z_1, \ldots, z_n)^T$, est saine ou stéganographiée. La décision qu'elle doit prendre consiste à faire un choix entre les hypothèses $\mathcal{H}_0$ (*image saine*) et $\mathcal{H}_1$ (*image stéganographiée*).

Définition 2.2 (Hypothèse simple)**.** *Une hypothèse simple $\mathcal{H}_j$ est une hypothèse définissant de manière unique la distribution de l'échantillon Z, on la note :*

$$\mathcal{H}_j = \{z_1, \ldots, z_n \sim P_{\theta_j}\}, j = 0, 1. \tag{2.13}$$

Ainsi, Eve doit choisir entre les hypothèses suivantes :

$$\begin{cases} \mathcal{H}_0 & : \{Z \sim P_{\theta_0}\} \quad \text{l'image } Z \text{ est saine,} \\ \mathcal{H}_1 & : \{Z \sim P_{\theta_1}\} \quad \text{l'image } Z \text{ est stéganographiée.} \end{cases} \tag{2.14}$$

Une fois les hypothèses définies, le "choix" est fait à travers un test statistique.

Définition 2.3 (Test statistique)**.** *On appelle test statistique (ou règle de décision) entre deux hypothèses $\mathcal{H}_0$ et $\mathcal{H}_1$ une application δ surjective et mesurable de l'espace des observations $\mathbb{R}^n$ sur l'ensemble des hypothèses envisagées $\delta : \mathbb{R}^n \to \{\mathcal{H}_0, \mathcal{H}_1\}$.*

Définir un test statistique binaire revient donc à partitionner l'espace des observations $\mathbb{R}^n$ en deux régions d'acceptation Ω_0 et Ω_1 telles que $Z \in \Omega_j$ si $\delta(Z) = \mathcal{H}_j$.

Définition 2.4 (Fonction de décision)**.** *Une fonction de décision π associée au test δ est l'application $\pi : \mathbb{R}^n \to [0; 1]$ définie par :*

$$\pi(Z) = \begin{cases} 1 & si \quad Z \in \Omega_1 \\ 0 & si \quad Z \in \Omega_0 \end{cases} \tag{2.15}$$

On peut à présent s'intéresser à la qualité du test qui peut se définir par le nombre d'erreurs commises, pour cela, on utilise des probabilités d'erreur.

Définition 2.5 (Probabilité d'erreur)**.** *On appelle probabilité d'erreur de j-ième espèce du test δ la probabilité de rejeter l'hypothèse $\mathcal{H}_j$ lorsque celle-ci est vraie :*

$$\alpha_j(\delta) = \mathbb{P}_{\theta_j}(\delta(z_1, \ldots, z_n) \neq \mathcal{H}_j) \tag{2.16}$$

Dans ce manuscrit, nous considérons des tests binaires entre les hypothèses $\mathcal{H}_0$ (hypothèse de base) et $\mathcal{H}_1$ (hypothèse alternative). Les quatre événements suivants sont possibles :

	$\mathcal{H}_1$ est vraie	$\mathcal{H}_0$ est vraie
$\mathcal{H}_0$ rejetée $\mathcal{H}_1$ acceptée	Décision correcte $\beta(\delta) = 1 - \alpha_1(\delta)$	Erreur de 1$^{\text{ère}}$ espèce $\alpha_0(\delta)$
$\mathcal{H}_0$ acceptée $\mathcal{H}_1$ rejetée	Erreur de 2$^{\text{ème}}$ espèce $\alpha_1(\delta)$	Décision correcte $1 - \alpha_0(\delta)$

Lorsque le medium analysé est déclaré stéganographié par le détecteur δ, alors que ce n'est pas le cas, on parle de *fausse-alarme*. La probabilité associée à cet événement est la probabilité d'erreur α_0 et est appelée *probabilité de fausse alarme*. Dans le cas contraire, si une image stéganographiée n'est pas détectée, on parle de *non-détection* et α_1 représente la *probabilité de non-détection*. Enfin, la probabilité de détection notée $\beta(\delta)$ est appelée puissance du test.

Lorsque l'on construit un test, il est nécessaire de se fixer un critère d'optimalité. Dans le cas de deux hypothèses simples, il existe trois approches possibles :

- test le plus puissant (approche bi-critère) ;
- test bayésien (approche mono-critère) ;
- test minimax (approche mono-critère).

Les approches mono-critère sont basées sur la minimisation d'un unique critère, alors que les approches bi-critères recherchent des tests optimaux selon deux critères.

2.3.1 Test le plus puissant entre deux hypothèses simples

Il s'agit d'un test proposé par Neyman et Pearson basé sur les critères $\alpha_0(\delta)$ et $\beta(\delta)$. L'idée de ce test est de fixer la probabilité maximale de fausse alarme α que l'on considère comme acceptable, puis de maximiser la puissance du test.

Définition 2.6 (Classe $\mathcal{K}_\alpha$)**.** *On définit la classe $\mathcal{K}_\alpha$ comme l'ensemble des tests dont la probabilité de fausse alarme est bornée supérieurement par α :*

$$\mathcal{K}_\alpha = \{\delta : \mathbb{P}_{\theta_0}(\delta(z_1, \ldots, z_n) = \mathcal{H}_1) \leq \alpha\}. \tag{2.17}$$

Définition 2.7 (Test le plus puissant)**.** *On appelle test le plus puissant dans la classe $\mathcal{K}_\alpha$ le test $\tilde{\delta}$ pour lequel l'inégalité $\alpha_1(\tilde{\delta}) \leq \alpha_1(\delta)$ est vraie pour tout test δ de cette classe.*

On cherche donc le test le plus puissant parmi tous les tests de la classe $\mathcal{K}_\alpha$, où α est le niveau maximal de fausse alarme que l'on se fixe. Dans la pratique, la construction d'un tel test est donnée par le lemme de Neyman-Pearson :

Théorème 2.1 (Lemme de Neyman-Pearson)**.** *Soient les distributions P_{θ_0} et P_{θ_1} de densités de probabilité f_{θ_0} et f_{θ_1}, supposons que $\Xi = (z_1, \ldots, z_n)^T$ est issu de l'une d'elles. Considérons les deux hypothèses :*

$$\mathcal{H}_0 : \{Z \sim P_{\theta_0}\} \quad et \quad \mathcal{H}_1 : \{Z \sim P_{\theta_1}\}.$$

Le test $\tilde{\delta}$ du rapport de vraisemblance

$$\tilde{\delta}(z_1, \ldots, z_n) = \begin{cases} \mathcal{H}_0 & si \quad \Lambda(z_1, \ldots, z_n) = \dfrac{f_{\theta_1}(z_1, \ldots, z_n)}{f_{\theta_0}(z_1, \ldots, z_n)} < h \\ \mathcal{H}_1 & si \quad \Lambda(z_1, \ldots, z_n) \geq h \end{cases} \tag{2.18}$$

est le plus puissant (PP) dans la classe $\mathcal{K}_\alpha$. Le paramètre h ou seuil est la solution de l'équation

$$\mathbb{P}_0(\Lambda(z_1, \ldots, z_n) \geq h) = \alpha. \tag{2.19}$$

La statistique utilisée comme critère dans cette approche est le rapport de vraisemblance :

$$\Lambda(z_1, \ldots, z_n) = \frac{f_{\theta_1}(z_1, \ldots, z_n)}{f_{\theta_0}(z_1, \ldots, z_n)}, \tag{2.20}$$

où f_{θ_j} est la densité de probabilité de P_{θ_j}, $j = 0, 1$.

2.3.2 Test bayésien entre deux hypothèses simples

L'approche de ce test diffère du précédent dans le sens où les hypothèse simples $\mathcal{H}_j$, et donc le paramètre θ de la distribution testée, ne sont plus fixes mais aléatoires. Ainsi, nous supposons que les probabilités $\mathbb{P}(\mathcal{H}_j) = q_j$ telles que $q_0 + q_1 = 1$ sont connues. $Q = (q_0, q_1)$ est une distribution *a priori* sur l'ensemble des hypothèses $\mathcal{H}_j$.

Le critère que l'on considère alors est le coût de Bayes, qui représente une somme pondérée des risques de j-ème espèce :

$$J_Q(\delta) = q_0\alpha_0 + q_1\alpha_1 \tag{2.21}$$

Définition 2.8 (Test bayésien)**.** *On appelle test bayésien associé à la distribution Q le test δ_Q qui minimise le coût de Bayes $J_Q(\delta)$.*

Proposition 2.1. *Dans le cas de deux hypothèses simples, la règle de décision qui minimise le coût de Bayes $J_Q(\delta) = q_0\alpha_0 + q_1\alpha_1$ est définie par le lemme de Neyman-Pearson :*

$$\delta_Q(z_1, \ldots, z_n) = \begin{cases} \mathcal{H}_0 & si \quad \Lambda(z_1, \ldots, z_n) < \dfrac{q_0}{q_1} \\ \mathcal{H}_1 & si \quad \Lambda(z_1, \ldots, z_n) \geq \dfrac{q_0}{q_1} \end{cases} \tag{2.22}$$

2.3.3 Test minimax entre deux hypothèses simples

Cette approche fait un compromis entre la fausse alarme et la non-détection en considérant comme critère le maximum des risques α_0, α_1 :

$$\bar{J}(\delta) = \max\{\alpha_0, \alpha_1\}. \tag{2.23}$$

Définition 2.9 (Test minimax)**.** *On appelle test minimax le test $\bar{\delta}$ qui minimise le maximum de risques $\bar{J}(\delta)$.*

Proposition 2.2. *S'il existe un test bayésien $\bar{\delta}$ tel que les risques de première et seconde espèces sont égaux : $\alpha_0(\bar{\delta}) = \alpha_1(\bar{\delta})$, alors ce test est minimax.*

2.3.4 Tests randomisés

Nous avons vu que le test le plus puissant du lemme de Neyman-Pearson est basé sur le rapport de vraisemblance de distributions continues. Supposons que la distribution de Z sous l'hypothèse $\mathcal{H}_j$ soit discrète, sous ces conditions, l'équation

$$\mathbb{P}_{\theta_0}(\Lambda(Z) \geq h) = \alpha \tag{2.24}$$

n'admet pas de solution exacte. Il est donc nécessaire de remplacer cette équation par l'inégalité

$$\mathbb{P}_{\theta_0}(\Lambda(Z) \geq h) < \alpha. \tag{2.25}$$

Ce remplacement a pour conséquence la réduction de la puissance du test, et un test optimal non-randomisé n'est plus garanti dans la classe $\mathcal{K}_\alpha$. Le test du rapport de vraisemblance est optimal seulement pour certaines valeurs α.

Dans le cadre de la stéganalyse, les observations représentent les pixels d'une image brute ou les coefficients DCT d'une image JPEG. Or, ces données ont été quantifiées (voir chapitre 3), et par conséquent la distribution de la variable aléatoire Z n'est pas continue mais discrète. Les tests randomisés peuvent donc être utilisés dans la détection d'informations cachées.

Pour remédier à cette situation, un test randomisé est utilisé. La construction de ce test est similaire à celle du test non-randomisé à une ligne près :

$$\widetilde{\pi}(Z) = \begin{cases} 1 & \text{si} \quad \Lambda(Z) = \dfrac{f_{\theta_1}(Z)}{f_{\theta_0}(Z)} \quad > \quad h \\ p & \text{si} \quad \Lambda(Z) \quad = \quad h \\ 0 & \text{si} \quad \Lambda(Z) \quad < \quad h \end{cases} \tag{2.26}$$

Si $\Lambda(Z) = h$, alors l'hypothèse alternative $\mathcal{H}_1$ est acceptée avec la probabilité p et elle est rejetée avec la probabilité $1 - p$, les paramètres h et p étant définis par :

$$\mathbb{P}_{\theta_0}(\Lambda(Z) > h) + p\mathbb{P}_{\theta_0}(\Lambda(Z) = h) = \alpha. \tag{2.27}$$

Dans le cas où la statistique T est discrète, la fonction critique $\tilde{\pi}$ du test randomisé $\tilde{\delta}$ UPP dans la classe $\mathcal{K}_\alpha$ est la suivante :

$$\tilde{\pi}(Z) = \begin{cases} 1 & \text{si} \quad T(Z) > h \\ p & \text{si} \quad T(Z) = h \\ 0 & \text{si} \quad T(Z) < h \end{cases} \tag{2.28}$$

où les paramètres h et p sont définis de l'équation

$$\mathbb{P}_{\theta_0}(T(Z) > h) + p\mathbb{P}_{\theta_0}(T(Z) = h) = \alpha. \tag{2.29}$$

2.3.5 Test entre deux hypothèses composites

Jusqu'à présent, on se plaçait dans le cadre d'hypothèses simples, ce qui est le cas lorsque l'on considère connu le taux d'insertion par exemple. Or, dans la pratique, le taux d'insertion R n'est pas connu, on peut donc considérer les hypothèses suivantes : $\mathcal{H}_0 = \{R \leq r^*\}$ et $\mathcal{H}_1 = \{R > r^*\}$.

Définition 2.10 (Hypothèse composite)**.** *Une hypothèse qui n'est pas simple est appelée composée (ou composite). Dans le cas paramétrique, on définit une telle hypothèse par :*

$$\mathcal{H}_j = \{z_1, \ldots, z_n \sim P_{\theta_j} | \theta_j \in \Theta_j\}, j = 0, 1. \tag{2.30}$$

où Θ_j est un domaine de l'espace paramétrique.

L'hypothèse $\mathcal{H}_j$ est vérifiée lorsque le vecteur aléatoire $Z \sim P_\theta$ de densité de probabilité f_θ avec $\theta \in \Theta_j$. Dans nos travaux, l'espace paramétrique Θ est partitionné en deux ensembles disjoints :

$$\Theta = \Theta_0 \cup \Theta_1, \quad \text{avec } \Theta_0 \cap \Theta_1 = \varnothing.$$

Sous ces conditions, le lemme de Neyman-Pearson n'est plus applicable, redéfinissons donc les notions de probabilité d'erreur et de puissance dans le cas d'hypothèses composées.

Définition 2.11 (Probabilité d'erreur et puissance)**.** *On appelle probabilité d'erreur première espèce du test δ la probabilité* maximale *de rejeter l'hypothèse composée $\mathcal{H}_j$ lorsque celle-ci est vraie :*

$$\alpha_0(\delta) = \sup_{\theta \in \Theta_0} \mathbb{P}_\theta(\delta(z_1, \ldots, z_n) = \mathcal{H}_1) \tag{2.31}$$

La puissance du test δ est la probabilité d'accepter l'hypothèse composée $\mathcal{H}_1$:

$$\beta_\delta(\theta) = \mathbb{P}_\theta(\delta(z_1, \ldots, z_n) = \mathcal{H}_1) \tag{2.32}$$

Définition 2.12 (Test uniformément le plus puissant)**.** *On appelle un test $\tilde{\delta}$ uniformément le plus puissant (UPP) dans la classe*

$$\mathcal{K}_\alpha = \left\{ \delta : \sup_{\theta \in \Theta_0} P_\theta(\delta(z_1, \ldots, z_n) = \mathcal{H}_1) \leq \alpha \right\} \tag{2.33}$$

si pour tout $\delta \in \mathcal{K}_\alpha$ l'inégalité suivante est vérifiée

$$\beta_{\tilde{\delta}}(\theta) \geq \beta_\delta(\theta) \quad \forall \theta \in \Theta_1. \tag{2.34}$$

En pratique, il est difficile d'obtenir un test qui soit uniformément le plus puissant. En revanche, on peut obtenir un test UPP si le paramètre est scalaire et que la famille $\mathcal{P} = \{P_\theta\}_{\theta\in\Theta}$ admet un rapport de vraisemblance monotone, i.e. :

$$\Lambda(z)\frac{f_{\theta_1}(z)}{f_{\theta_0}(z)} = g(T(z)) \text{ est une fonction croissante ou décroissante de } T(z).$$

Proposition 2.3. *Supposons que $\mathcal{P} = \{P_\theta\}_{\theta\in\Theta}$ est une famille de distributions dépendant d'un paramètre scalaire et que cette famille possède un rapport de vraisemblance monotone. On souhaite faire un choix entre les hypothèses suivantes :*

$$\mathcal{H}_0 = \{\theta \leq \theta^*\} \quad \textit{contre} \quad [\mathcal{H}_1 = \{\theta > \theta^*\}. \tag{2.35}$$

La règle de décision du test $\tilde{\delta}$ UPP dans la classe $\mathcal{K}_\alpha$ est la suivante :

$$\tilde{\delta}(z_1,\ldots,z_n) = \begin{cases} \mathcal{H}_0 & si \quad T(z_1,\ldots,z_n) < h \\ \mathcal{H}_1 & si \quad T(z_1,\ldots,z_n) \geq h \end{cases}, \tag{2.36}$$

où le seuil h est la solution de l'équation

$$\mathbb{P}_{\theta^*}(T(z_1,\ldots,z_n) \geq h) = \alpha.$$

La fonction $\theta \mapsto \beta_{\tilde{\delta}}(\theta)$ (puissance de $\tilde{\delta}$) est strictement croissante.

Nous venons de voir que, sous certaines conditions, nous pouvons obtenir un test optimal avec des hypothèses composées. Cependant, il n'est pas possible de pouvoir prétendre construire un test optimal dans n'importe quelle situation. Les résultats du lemme de Neyman-Pearson ne peuvent être utilisés en l'état puisque les fonctions f_{θ_0} et f_{θ_1} étant inconnues, il est impossible de calculer le rapport de vraisemblance $\Lambda(Z)$. L'idée est donc d'estimer les paramètres inconnus θ_0 et θ_1, et d'utiliser ces estimations pour le calcul du rapport de vraisemblance généralisé :

$$\hat{\Lambda}(Z) = \frac{\sup_{\theta\in\Theta_1} f_{\theta_1}(Z)}{\sup_{\theta\in\Theta_0} f_{\theta_0}(Z)} \tag{2.37}$$

Définition 2.13 (Test du rapport de vraisemblance généralisé)**.** *Soient deux hypothèses composées $\mathcal{H}_0$ et $\mathcal{H}_1$, on appelle test du rapport de vraisemblance généralisé entre ces deux hypothèses, le test $\hat{\delta}$ défini par la règle de décision :*

$$\hat{\delta}(Z) = \begin{cases} \mathcal{H}_0 & si \quad \hat{\Lambda}(Z) < h \\ \mathcal{H}_1 & si \quad \hat{\Lambda}(Z) \geq h \end{cases},$$

où le seuil h est la solution de l'équation $\sup_{\theta_0\in\Theta_0} \mathbb{P}(\hat{\delta}(Z) \geq h) = \alpha.$

2.3.6 Test invariant

Soit la variable aléatoire Z distribuée suivant la loi $P_{\theta,\eta}$, où $\theta \in \Theta$ sont des *paramètres informatifs* et $\eta \in \mathcal{Y}$ sont des *paramètres de nuisance*. La prise en compte des paramètres de nuisance est fondamentale puisqu'ils interviennent dans la distribution de la variable aléatoire, en revanche, ils n'interviennent pas dans l'estimation du paramètre informatif ou dans la prise de décision entre les hypothèses $\mathcal{H}_j$.

En stéganalyse des images, les informations sont cachées dans le bruit, le contenu de l'image est donc considéré comme un paramètre de nuisance.

Dans le cadre de la détection statistique en présence de paramètres de nuisance, la théorie de l'invariance peut être particulièrement utile pour rejeter les paramètres de nuisance. Dans la section 4.3.6 seront présentés les résultats obtenus en utilisant cette approche.

Dans cette section nous introduisons donc la notion d'invariance d'une famille de distributions et étudions les tests invariants. Supposons que $\xi \in P_\theta$, où $\mathcal{P} = \{P_\theta\}_{\theta\in\Theta}$ est une famille de distributions dépendant d'un paramètre θ et vérifiant la condition suivante : $P_{\theta_1} \neq P_{\theta_2}$ pour $\theta_1 \neq \theta_2$. On considère un groupe G des applications mesurables et bijectives g de l'espace $\mathcal{Y}$ dans lui-même [3]. Donc, toute application g applique $\mathcal{Y}$ sur $\mathcal{Y}$: pour tout $y \in \mathcal{Y}$ il existe $x \in \mathcal{Y}$ tel que $y = g(x)$.

Propriétés 2.1 (Groupe)**.** *1.* $\exists\, e \in G \ : \forall g \in G, ge = eg = g$,

2. *Si* $g_1 \in G, g_2 \in G$ *alors il existe* $g = g_1g_2 \in G$ *et* $\forall g_1 \in G, \forall g_2 \in G,\ \forall g_3 \in G, (g_1g_2)g_3 = g_1(g_2g_3)$
3. *Si* $g \in G$ *alors il existe l'application réciproque* $g^{-1} \in G \ : g^{-1}g = gg^{-1} = e$.

Définition 2.14. *On dit que la* famille de distributions $\mathcal{P} = \{P_\theta\}_{\theta\in\Theta}$ est invariante *par le groupe d'applications G si pour tout paramètre* $\theta \in \Theta$*, toute transformation* $g \in G$*, et tout événement A il existe un paramètre unique* $\theta_g = \bar{g}(\theta) \in \Theta$ *tel que l'égalité suivante est vérifiée :*

$$\mathbb{P}_\theta(g(\xi) \in A) = \mathbb{P}_{\bar{g}(\theta)}(\xi \in A).$$

Dans cette situation les applications $\bar{g}$ forment un groupe $\bar{G}$.

Définition 2.15. *On dit que le problème de choix entre deux hypothèses :* $\mathcal{H}_1 = \{\Xi_n \sim P_{\theta_1} | \theta_1 \in \Theta_1\}$ *et* $\mathcal{H}_2 = \{\Xi_n \sim P_{\theta_2} | \theta_2 \in \Theta_2\}$*, au vu d'un échantillon* $\Xi_n \sim P_\theta$ *est* invariant *si,*

1. *la famille* $\mathcal{P} = \{P_\theta\}_{\theta\in\Theta}$ *est invariante par un groupe G ;*
2. *les domaines* Θ_1 *et* Θ_2 *sont invariants par* $\bar{g}$*, c'est-à-dire que* $\bar{g}(\Theta_j) = \Theta_j,\ j = 1, 2$.

Si nous avons un problème invariant, alors il est bien naturel d'utiliser un test invariant pour résoudre ce problème.

Définition 2.16. *On dit que une* statistique $T(X)$ est invariante *par le groupe G si :*

$$\forall X \in \mathbb{R}^n, \forall g \in G, T(g(X)) = T(X) :$$

Définition 2.17. *On dit que un* test est invariant *si sa fonction critique* $\pi(\xi)$ *est une statistique invariante.*

3. C'est-à-dire que si ξ est une variable aléatoire qui prend ces valeurs dans $\mathcal{Y}$, alors $g(\xi)$ est également une variable aléatoire qui prend ces valeurs dans le même espace.

Définition 2.18. *On dit que une statistique $T(X)$ est un invariant maximal par le groupe G si :*

1. *T est une statistique invariante ;*
2. *$\forall X_1$ et $\forall X_2$, $T(X_1) = T(X_2) \Rightarrow \exists g \in G : X_2 = g(X_1)$.*

Lorsqu'un problème de choix entre deux hypothèses est invariant par un groupe G, une démarche classique pour rechercher un test optimal consiste à calculer un invariant maximal puis à chercher un test optimal qui s'appuie sur cette statistique. En effet, d'après la proposition suivante, tout test invariant dépend nécessairement de l'invariant maximal.

Proposition 2.4. *Soit $T(X)$ un invariant maximal. Une statistique S est invariante si et seulement si elle dépend de l'observation X par l'intermédiaire de T, c'est-à-dire s'il existe une fonction φ telle que $S(X) = \varphi(T(X))$.*

2.3.7 Tests asymptotiques

En stéganographie, dans le cas d'insertion dans des images, le medium de couverture doit être suffisamment grand pour contenir une quantité d'informations cachées non négligeable. Ainsi, en stéganalyse, lors de la construction de tests statistiques, il est concevable de se placer dans le cas asymptotique où l'image de couverture (le support) est supposée de taille infinie ($n \to \infty$).

La section 2.3.7 concernant les hypothèses contiguës présente notamment les outils utilisés dans le cas où le taux d'insertion est inconnu (hypothèses composites) traité dans la section 4.3.5.

Cette section présente les principaux résultats concernant la théorie des tests asymptotiques.

Approche asymptotique de Wald

La théorie développée par Wald [81] est applicable dans les situations bien plus complexes qu'un simple modèle gaussien $z \sim \mathcal{N}(\theta, \Sigma)$. Mais, dans le cas général, cette théorie exige une supposition très forte - le caractère asymptotique du test. Soit d_n un test statistique entre deux (ou plusieurs) hypothèses basé sur l'échantillon $Z_n = (z_1, \ldots, z_n)^T$ de taille n. L'approche asymptotique repose sur l'idée que l'on peut remplacer un test statistique entre deux (ou plusieurs) hypothèses d par une suite des tests $d_1, \ldots, d_n = \{d_n\}_{n \in \mathbb{N}}$ et analyser cette suite de tests lorsque le nombre d'observations n tend vers l'infini. Il est entendu qu'à partir de maintenant toutes les définitions et propriétés ont un caractère asymptotique. Leur application est rendue délicate car, en pratique, la taille d'échantillon est toujours finie. Pour cette même raison, l'usage des résultats de la théorie asymptotique n'est pas aisé, mais le grand intérêt de cette approche est sa globalité du point de vue théorique et son universalité. Cette théorie est assez complexe et nous proposons ici un très bref aperçu des principaux résultats [81, 67, 4].

Considérons le cas général où l'ensemble des observations $Z_n = (z_1, \ldots, z_n)^T$ est issu d'une distribution de probabilité P_θ quelconque, $\theta \in \Theta \subseteq \mathbb{R}^m$. On note f_θ la densité de probabilité de P_θ. On teste l'hypothèse $\mathcal{H}_0 = \{P_{\theta_0}\}$ contre l'hypothèse $\mathcal{H}_1 = \{P_\theta \mid \theta \neq \theta_0\}$.

Une notion essentielle pour la théorie asymptotique est la *matrice d'information de Fisher*

$$\mathcal{F}(\theta) = \begin{pmatrix} \mathcal{F}_{1,1}(\theta) & \mathcal{F}_{1,2}(\theta) & \dots & \mathcal{F}_{1,m}(\theta) \\ \mathcal{F}_{2,1}(\theta) & \mathcal{F}_{2,2}(\theta) & \dots & \mathcal{F}_{2,m}(\theta) \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{F}_{m,1}(\theta) & \mathcal{F}_{m,2}(\theta) & \dots & \mathcal{F}_{m,m}(\theta) \end{pmatrix} \tag{2.38}$$

où

$$\mathcal{F}_{i,j}(\theta) = \mathbb{E}_\theta\left[\frac{\partial l(z(k),\theta)}{\partial \theta_i}\frac{\partial l(z(k),\theta)}{\partial \theta_j}\right], \quad l(z(k),\theta) = \log f_\theta(z(k)), \quad i,j = 1,\dots,m.$$

Wald a démontré [81] que le test

$$d(z_1, z_2, \dots, z_n) = \begin{cases} \mathcal{H}_1 & \text{si} \quad W(Z_n) < h \\ \mathcal{H}_2 & \text{si} \quad W(Z_n) \geq h \end{cases} \tag{2.39}$$

basé sur la statistique de Wald

$$W(z_1,\dots,z_n) = n(\widehat{\theta}(Z_n) - \theta_0)^T \mathcal{F}(\widehat{\theta}(Z_n))(\widehat{\theta}(Z_n) - \theta_0) \tag{2.40}$$

avec l'estimation du maximum de vraisemblance $\widehat{\theta}(Z_n)$ définie par

$$\widehat{\theta}(Z_n) = \underset{\theta}{\operatorname{argmax}}\, f_\theta(Z_n), \tag{2.41}$$

est asymptotiquement UPP (i.e. lorsque $n \to \infty$) et possède une fonction de puissance $\beta(\theta)$ *constante* sur chaque surface S_c de la famille

$$\mathcal{S} = (\theta - \theta_0)^T \mathcal{F}(\theta_0)(\theta - \theta_0). \tag{2.42}$$

Le test (2.39) est également *le plus rigoureux* asymptotiquement (asymptotically most stringent test) dans la classe $\mathcal{K}_\alpha$.

Schéma d'hypothèses contiguës - tests locaux

L'objet de cette section est de poursuivre l'étude asymptotique commencée dans la sous-section 2.3.7. Construire un test optimal de choix entre deux hypothèses consiste à trouver une règle statistique et à calculer la fonction de puissance de ce test. Souvent, ces deux problèmes sont difficiles à résoudre. Par exemple, nous avons vu dans la sous-section 2.3.7 que pour trouver un test UPP avec la fonction de puissance constante sur une famille de surfaces, il nous fallait réaliser préalablement l'estimation du maximum de vraisemblance $\widehat{\theta}(Z_n)$ du paramètre inconnu θ. Ce problème peut être difficile à résoudre, surtout pour un usage en temps réel. Une solution possible réside dans l'utilisation de l'approche *asymptotique locale* qui permet de résoudre approximativement ces problèmes pour de grandes valeurs de n. Cette méthode est connue également sous le nom de *schéma d'hypothèses contiguës*. L'idée de l'approche asymptotique locale consiste à réduire un problème de choix entre les hypothèses voisines $\mathcal{H}_0$ et $\mathcal{H}_1$ au vu de $Z_n \sim P_\theta$

au problème de choix entre deux hypothèses sur le vecteur moyenne d'une distribution normale [4, 52].

On considère que l'ensemble d'observations Z_n est issu d'une distribution de probabilité P_θ quelconque . On suppose que $n \to \infty$. Soit $\{\Theta(n)\}_{n\in\mathcal{N}}$ une suite d'ensembles $\Theta(n) = \theta^* + \frac{1}{\sqrt{n}}\boldsymbol{v}$, $\boldsymbol{v} \subset \vartheta \subset \Theta \subset \mathbb{R}^m$, où ϑ est un sous-ensemble de Θ, $\theta^* \in \Theta$ est un vecteur (scalaire) constant. Les hypothèses contiguës sont présentées sur la figure 2.12.

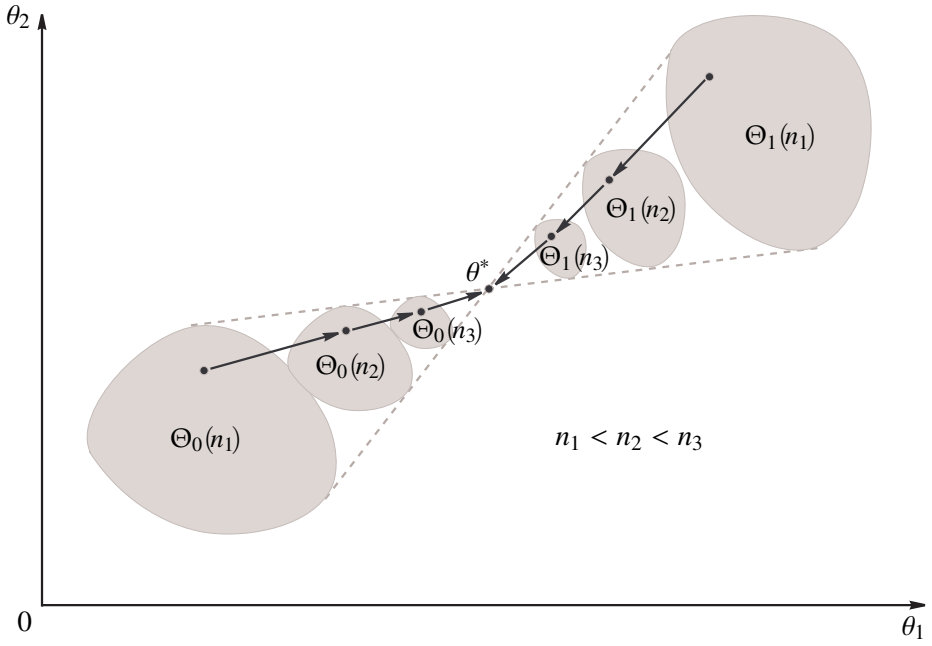


FIGURE 2.12 – *Schéma d'hypothèses contiguës.*

On considère les suites d'hypothèses $\mathcal{H}_j(n) = \{\theta \in \Theta_j(n)\}$ se rapprochant l'une de l'autre ($j = 0, 1$) (voir la figure 2.12). Les domaines $\Theta_j(n)$ sont de la forme $\Theta_j(n) = \theta^* + \frac{1}{\sqrt{n}}\boldsymbol{v}$, où $\boldsymbol{v} \subset \vartheta_j$. La vitesse de rapprochement est $\frac{1}{\sqrt{n}}$.

Pour simplifier l'écriture, soient $\mathcal{H}_0(n) = \{\theta^*\}$ et $\mathcal{H}_1(n) = \{\theta_n = \theta^* + \frac{1}{\sqrt{n}}\boldsymbol{v}\}$. Dans ce cas, le logarithme du rapport de vraisemblance

$$S(\boldsymbol{v}) = \log \Lambda\left(\frac{1}{\sqrt{n}}\boldsymbol{v}\right) = \log f_{\theta^*+\frac{1}{\sqrt{n}}\boldsymbol{v}}(Z_n) - \log f_{\theta^*}(Z_n),$$

possède des propriétés asymptotiques très importantes. Soit $\{\lambda_n\}_{n\in\mathbb{N}}$ une suite réelle positive convergente vers 0. Si $Z_n = (z_1, \ldots, z_n) \sim P_\theta$, alors pour les $\boldsymbol{v}$ tels que $\left\|\frac{1}{\sqrt{n}}\boldsymbol{v}\right\|_2 \leq \lambda_n$ on a

$$S(\boldsymbol{v}) = \log \Lambda\left(\frac{1}{\sqrt{n}}\boldsymbol{v}\right) = \boldsymbol{\zeta}_n^T(\theta^*; Z_n)\boldsymbol{v} - \frac{1}{2}\boldsymbol{v}^T\mathcal{F}(\theta^*)\boldsymbol{v}(1 + \epsilon_n(Z_n, \theta^*, \boldsymbol{v}))$$

où $\|\epsilon_n(Z_n;\theta^*,\boldsymbol{\upsilon})\|_2 \leq \epsilon_n(Z_n;\theta^*) \overset{p.s.}{\to} 0$,

$$\boldsymbol{\zeta}_n(\theta;Z_n) = \frac{1}{\sqrt{n}}\nabla_\theta\{\log f_\theta(Z_n)\} = \frac{1}{\sqrt{n}}\sum_{k=1}^{n}\nabla_\theta\{\log f_\theta(z(k))\} \tag{2.43}$$

est le *score efficace* et le vecteur gradient d'une fonction $\boldsymbol{x}\mapsto f(\boldsymbol{x})$ est défini par

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \frac{\partial f(\boldsymbol{x})}{\partial x_2}, \ldots, \frac{\partial f(\boldsymbol{x})}{\partial x_n}\right)^T \text{ avec } \boldsymbol{x} = (x_1,\ldots,x_n).$$

Soit la matrice d'information de Fisher $\mathcal{F}(\theta)$ de l'observation $z(k)$ bornée et définie positive pour tous les $\theta\in\Theta$. La partie principale du rapport de vraisemblance peut être représentée sous la forme

$$S(\boldsymbol{\upsilon}) \simeq \boldsymbol{\zeta}_n^T(\theta^*;Z_n)\boldsymbol{\upsilon} - \frac{1}{2}\boldsymbol{\upsilon}^T\mathcal{F}(\theta^*)\boldsymbol{\upsilon},$$

où

$$\zeta_n(\theta^*) \rightsquigarrow \begin{cases} \mathcal{N}(\mathbf{0},\mathcal{F}(\theta^*)) & \text{sous } z(k) \sim P_{\theta^*} \\ \mathcal{N}(\mathcal{F}(\theta^*)\boldsymbol{\upsilon},\mathcal{F}(\theta^*)) & \text{sous } z(k) \sim P_{\theta^*+\frac{\upsilon}{\sqrt{n}}} \end{cases}. \tag{2.44}$$

La valeur $\boldsymbol{\upsilon}_n^* = \sqrt{n}(\widehat{\theta}-\theta^*) = \operatorname{argmax} S(\boldsymbol{\upsilon})$ qui réalise le maximum de $S(\boldsymbol{\upsilon})$ se représente sous la forme

$$\boldsymbol{\upsilon}_n^* = (1+\epsilon_n(Z_n,\theta^*))\mathcal{F}^{-1}(\theta^*)\zeta_n(\theta^*), \quad \epsilon_n(Z_n,\theta^*) \overset{p.s.}{\to} 0$$

et $2S(\boldsymbol{\upsilon}_n^*) \rightsquigarrow \chi^2_m$, où χ^2_m est une loi du χ^2 centrée à m degrés de liberté.

En d'autres termes, un problème de choix entres les hypothèses voisines $\mathcal{H}_0(n) = \{P_\theta|\theta\in\Theta_1(n)\}$ et $\mathcal{H}_1(n) = \{P_\theta|\theta\in\Theta_K(n)\}$ au vu de $Z_n\sim P_\theta$ se réduit au problème de choix entres deux hypothèses sur le vecteur moyenne du score efficace (2.43) qui est asymptotiquement gaussien.

Conclusion

Dans ce chapitre, les notions de stéganographie et de stéganalyse ont été introduites. Les outils d'aide à la décision statistique présentés sont une introduction aux détecteurs mis en place dans le chapitre 4. Nous avons vu que la détection de la présence d'un message consiste à choisir entre deux hypothèses : l'hypothèse $\mathcal{H}_0$ qui suppose le medium seul, et l'hypothèse $\mathcal{H}_1$ qui suppose en plus la présence d'un message. Afin de caractériser chacune des hypothèses, il est nécessaire de pouvoir modéliser paramétriquement le medium. Ainsi, le chapitre 3 présente la modélisation des images naturelles non compressées et en particulier compressées au format JPEG.

Chapitre 3

Modélisation des images naturelles : du RAW au JPEG

Sommaire

Introduction

Après avoir introduit les outils décision statistique dans le chapitre précédent (voir chapitre 2), il est nécessaire de modéliser le medium dans lequel on souhaite détecter la présence d'informations cachées (voir chapitre 4).

Ce chapitre propose donc la modélisation de différents types d'images : les images *brutes*, directement issues d'un capteur, les images *non compressées* qui restituent fidèlement la scène et les images *compressées* qui facilitent la transmission. Cette étude permettra la mise en place des détecteurs maîtrisables présentés au chapitre 4 afin de détecter la présence d'informations cachées dans des images.

Le principe général de la plupart des mécanismes de stéganographie consiste à insérer un message secret dans la partie bruitée d'un signal. En effet, comme les modifications apportées au medium de couverture ne doivent pas être visibles, il faut que le support contienne des données redondantes qui peuvent être remplacées par des informations à cacher. Dans notre étude, nous nous consacrons aux images numériques dites *naturelles* [1], c'est-à-dire acquises par un système optique puis numérisées et/ou compressées.

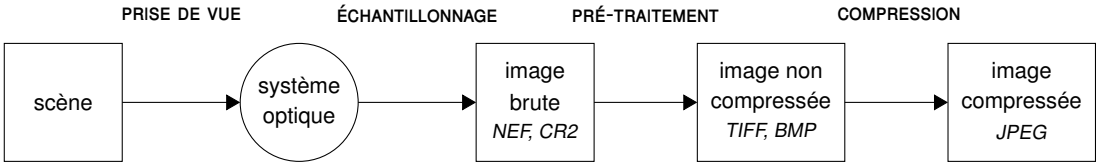


FIGURE 3.1 – *Chaîne d'acquisition des images naturelles.*

Comme le montre la figure 3.1, la chaîne d'acquisition d'une image comporte différentes étapes de la prise de vue à la compression du fichier numérique.

Prise de vue : lors d'une prise de vue, une scène est capturée par un système optique (appareil photographique, scanner, caméscope, etc.). La lumière émanant de cette scène passe à travers un jeu de lentilles et atteint le capteur photographique : on obtient une image optique.

Échantillonnage : la numérisation d'une image permet d'en obtenir une représentation informatique. Les deux étapes nécessaires sont :

- l'échantillonnage spatial selon l'axes des abscisses et des ordonnées : les pixels sont l'unité de base de cette représentation,
- la quantification permet d'échantillonner l'intensité lumineuse en valeurs pouvant être prises par les pixels.

On obtient ainsi une image brute. La modélisation des pixels d'une telle image est présentée dans la section 3.1.

Pré-traitement : différents traitements sont subits par l'image brute afin d'en améliorer le rendu visuel. Parmi ces traitements, figurent la balance des blancs, le dématriçage ou encore la correction gamma (voir section 3.2).

Compression : afin d'en faciliter la transmission, la compression permet de réduire la taille de l'image. Différents formats d'images compressées existent, notamment les formats JPEG et JPEG 2000. Nous nous intéressons au format JPEG qui est le plus populaire (section **??**) et à la modélisation des coefficients DCT (section 3.4).

1. Par opposition aux images *artificielles* qui auraient été créés informatiquement : les images vectorielles ou issues d'un logiciel de dessin par exemple ne contiennent pas suffisamment de données redondantes pour être utilisées en stéganographie.

3.1 Modèle d'image brute

Lors du processus d'acquisition de l'image numérique, le capteur photographique (CCD [2], CMOS [3]) convertit les photons incidents en électrons. Les charges créées dans chaque photosite sont ensuite stockées et les tensions électriques mesurées et quantifiées.

Avant numérisation, les bruits principaux que l'on peut considérer sont [38, 35] :

Bruit photonique ou *shot noise* : ce bruit caractérise les fluctuations du nombre de photons qui arrivent sur le capteur.

Non-uniformité de la réponse des photosites : des imperfections ou erreurs lors de la fabrication des capteurs engendrent des différences dans les réponses des photosites à une lumière uniforme.

Bruit thermique ou courant d'obscurité : en l'absence de lumière, des électrons libres sont générés dans le CCD par l'énergie thermique. Ces électrons étant indistinguables des photoélectrons, ils peuvent engendrer des erreurs dans le comptage des électrons.

Lors de la numérisation, de nouveaux processus engendrent du bruit :

Bruit d'amplification : l'amplificateur transforme la charge électrique collectée dans chaque pixel en tension et engendre un bruit gaussien.

Bruit de quantification : le convertisseur analogique-numérique quantifie et numérise le signal analogique. La quantification introduit un bruit de moyenne nulle et de variance $\frac{q^2}{12}$, où q est le pas de quantification.

Modélisons à présent la valeur d'un pixel d'une image brute. Au cours de la numérisation, différentes phénomènes aléatoires modifient la valeur z_i du pixel à la position i représentée par :

$$z_i = \theta_i + \xi_i \tag{3.1}$$

où $\theta_i = \mathbb{E}[z_i]$ est une valeur déterministe représentant la valeur moyenne du pixel, et ξ_i est la réalisation de la variable aléatoire Ξ_i qui représente l'ensemble des bruits perturbant l'image au cours de sa formation sur le capteur et lors de sa numérisation.

A. Foi et al. [24] considèrent un modèle de bruit simple mais précis pour les données brutes issues de capteurs d'imagerie numérique. Ce modèle de bruit est composé de deux parties mutuellement indépendantes ; la première, poissonnienne, modélise la sensibilité aux photons et dépend donc du signal, la seconde, gaussienne représente les perturbations stationnaires.

Ainsi, le bruit ξ_i associé au pixel z_i peut être représenté par :

$$\xi_i = \eta_{\mathrm{p}}(\theta_i) + \eta_{\mathrm{g}}(i) \tag{3.2}$$

où $\eta_{\mathrm{p}}(\theta_i)$ représente la partie poissonnienne du bruit, dépendante du signal (bruit quantique par exemple) et $\eta_{\mathrm{g}}(i)$ la partie gaussienne du bruit liée aux phénomènes stationnaires (bruit thermique, etc.).

2. Charge-Coupled Divice
3. Complementary Metal Oxide Semi-conductor

Un processus poissonnien pouvant être traité comme un processus gaussien hétéroscédastique, la variable aléatoire représentant l'ensemble des bruits peut être modélisée par une distribution gaussienne de variance σ^2. Il a été montré [24] qu'il existe une relation affine entre la moyenne du pixel et sa variance $\sigma^2 = a\theta_i + b$.

La modélisation d'un pixel brut est donc la suivante :

$$z_i = \theta_i + \xi_i, \quad \xi_i \sim \mathcal{N}(0, a\theta_i + b). \tag{3.3}$$

3.2 Pré-traitements - modèle d'image non compressée

Afin d'en améliorer le rendu visuel, une image brute subit souvent des pré-traitements. On retrouve principalement le dématriçage, la balance des blancs et la correction gamma. Ces traitements sont présentés dans cette section.

3.2.1 Dématriçage

Dans les appareils photographies numériques, les capteurs CCD et CMOS sont monochromatiques. En effet, les photosites les composant ne distinguent pas les différentes longueurs d'onde du signal lumineux. Ils mesurent seulement la quantité de lumière atteignant chaque photosite.

Les couleurs sont introduites par la superposition d'un filtre coloré sur le capteur. On trouve différentes mosaïques de couleurs ou CFA (*Color Filter Array*), mais la plus populaire est celle de Bayer. Ce filtre possède la particularité d'utiliser deux fois plus d'éléments verts que d'éléments rouges ou bleus afin de se rapprocher du système de vision humain.

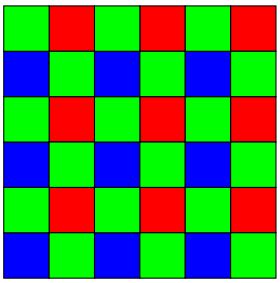


FIGURE 3.2 – *Matrice de Bayer.*

Une étape d'interpolation des deux composantes de couleurs manquantes, appelée dématriçage, est nécessaire pour chaque pixel afin de produire une image en couleur. Différentes méthodes d'interpolation existent, notamment les interpolations bilinéaire et bicubique.

Suite à ces interpolations, la propriété de normalité du pixel et la relation moyenne-variance sont conservées.

Cependant, il faut noter que le dématriçage crée une corrélation entre les différents pixels de couleur.

3.2.2 Balance des blancs

La balance des couleurs, ou la balance des blancs plus précisément, est un traitement d'image qui permet d'ajuster les intensités des différentes couleurs primaires (R, G, et B). L'idée est d'améliorer le rendu visuel du blanc qui a pu être altéré par l'éclairage ambiant. En effet, le système visuel humain a la capacité de percevoir la couleur d'un objet dans différentes conditions d'éclairage, alors que l'appareil photographique numérique capturera l'image sans faire d'ajustement.

Le traitement consiste donc à compenser les différences de couleurs causées par des sources de lumière diverses. Mathématiquement, cela correspond à la multiplication de l'intensité des pixels par un scalaire. La constante multiplicative est propre à chaque canal de couleur.

Ainsi, un pixel, après application de la balance des blancs, suit toujours une loi normale et la relation moyenne-variance persiste.

3.2.3 Correction gamma

La correction de gamma est une opération non linéaire qui compense les propriétés du système visuel humain. En effet, l'oeil est plus sensible à des changements dans les tons sombres que dans les tons clairs ; alors qu'un appareil photographique réagit de manière linéaire. La correction effectuée correspond à l'équation suivante :

$$V_{\text{oeil}} = V_{\text{appareil}}^{\gamma} \tag{3.4}$$

où V_{appareil} la luminance initiale et V_{oeil} représente la luminance après correction. Cela revient à coder les tons sombres avec davantage de bits que les teintes claires, afin de créer un signal *visuellement* uniforme.

Nous pouvons considérer cette courbe linéaire localement car la variance du pixel reste faible. Ainsi après correction gamma, la modélisation de la valeur d'un pixel suit toujours une loi normale localement et la nature hétéroscédastique est amplifiée.

3.3 Modèle d'image compressée - JPEG

Le format de compression d'image JPEG est une norme définie dans [39, 40, 41] en 1991 par les comités ISO et CCITT, qui ont travaillé conjointement dans le but d'établir une norme pour les images naturelles.

Jusqu'à lors, les techniques de compression [4] permettaient de réduire la taille d'un fichier de 10% à 50% sans affecter la qualité de l'image. La création d'une norme est devenue une nécessité afin de permettre l'interopérabilité entre des applications de différentes sociétés. Il existe différents algorithmes de compression JPEG (Baseline, Extended Sequential, etc.), mais celui qui va être étudié est l'algorithme JPEG Baseline puisqu'il s'agit du plus répandu.

4. Codage des répétitions ou Lempel-Ziv-Welch.

La compression d'une image couleur peut être vue comme la compression de plusieurs images en dégradé de gris. Il existe deux modes de compressions pour une image couleur :

- composante par composante [5] (la compression de la deuxième débute lorsque celle de la première est terminée.)
- par bloc de 8×8 pixels en alternant les composantes [6] image (premier bloc de la première composante, puis premier bloc de la deuxième composante ,etc.).

Les étapes principales de la compression JPEG sont : le découpage en blocs, la transposition amplitude-fréquence, la quantification et enfin le codage. L'algorithme de compression JPEG peut être résumé par le schéma (Fig. 3.3).

3.3.1 Pré-traitement de l'image

La phase de pré-traitement de l'image est une phase optionnelle ne faisant pas partie de la norme JPEG et qui est effectuée seulement pour les images couleurs. Elle permet la conversion d'une image dans le modèle colorimétrique adéquat et une première compression sur les composantes les moins sensibles (celles dont l'oeil ne verra pas forcément les modifications).

3.3.1.1 Changement de modèle colorimétrique

Classiquement les images sont codées en RGB (Red-Green-Blue). Selon la compression et la précision choisies, on peut définir chaque composante d'un pixel comme un entier S de précision P bits $(2 \leq q \leq 16)$ où $S \in \{0, \ldots, 2^q - 1\}$, la seule restriction est que la précision doit être la même pour tous les pixels. Dans le cas de l'algorithme *JPEG Baseline*, chaque pixel est défini par n octets (où n est le nombre de composantes, et $q = 8$). Par exemple, si l'espace colorimétrique initial est RGB, chaque pixel de l'image est représenté par 3 octets qui indiquent respectivement l'intensité des couleurs rouge, vert et bleu.

La luminance est une notion importante dans les domaines de compression et de traitement d'image numérique. Il s'agit d'une quantité proportionnelle à la puissance de la source lumineuse et est définie comme une somme pondérée de rouge, vert et bleu (respectivement avec les poids $77/256$, $150/256$ et $29/256$).

L'oeil étant particulièrement sensible aux faibles variations de luminance, le modèle YCbCr est particulièrement adapté à la compression. Une manière simplifiée d'obtenir un tel espace est de soustraire la luminance Y aux composantes B et R de l'espace RGB, afin d'obtenir Y, Cb = B - Y et Cr = R - Y comme nouvel espace colorimétrique.

D'après ce que l'on vient de voir, il est possible de dégrader la chrominance d'une image tout en gardant une bonne qualité ; c'est pourquoi le modèle colorimétrique YCbCr est utilisé pour les images JPEG, en effet, il permet de réduire la taille d'une image.

On peut passer d'un modèle colorimétrique à un autre en utilisant une transformation. Par

5. Non-interleaved encoding.
6. Interleaved encoding.

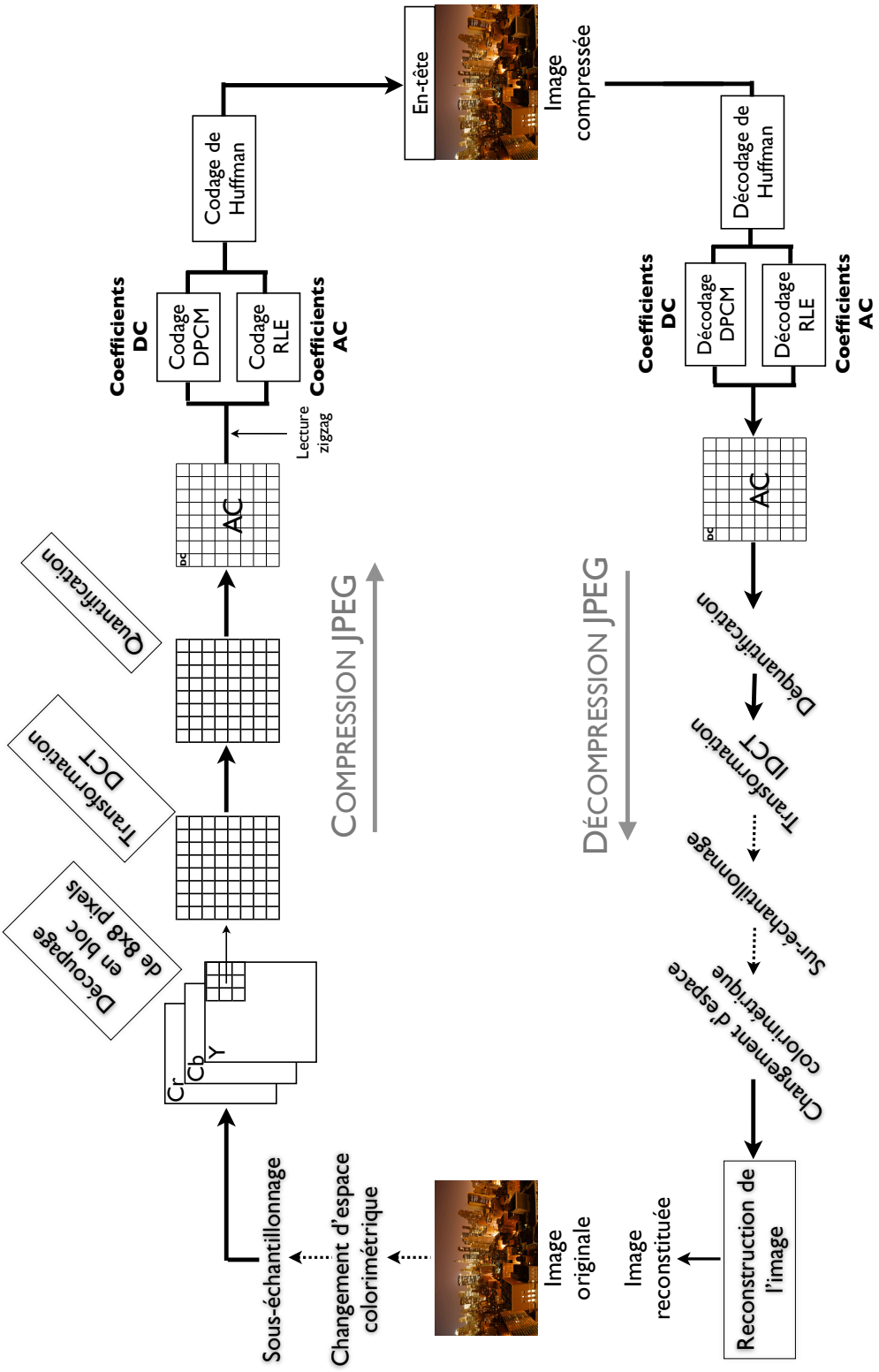


FIGURE 3.3 – *Compression et décompression JPEG.*

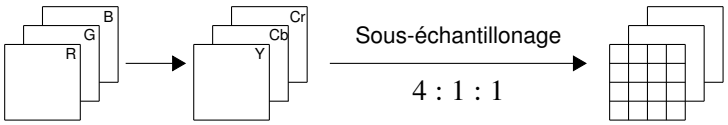


FIGURE 3.4 – *Pré-traitement de l'image.*

exemple, pour convertir un vecteur RGB en un vecteur YCbCr [7], on utilise la transformation suivante :

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} \frac{77}{256} & \frac{150}{256} & \frac{29}{256} \\ -\frac{44}{256} & -\frac{87}{256} & \frac{131}{256} \\ \frac{131}{256} & -\frac{110}{256} & -\frac{21}{256} \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix}, \tag{3.5}$$

$$\text{où} \begin{cases} 0 \leq R,G,B \leq 255 \\ 0 \leq Y \leq 255 \\ -128 \leq Cb,Cr \leq 127 \end{cases}$$

Dans le processus de compression JPEG, les valeurs d'intensité des pixels sont centrées autour de 0. Le changement de codage des couleurs préalable à la compression sera donc :

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} \frac{77}{256} & \frac{150}{256} & \frac{29}{256} \\ -\frac{44}{256} & -\frac{87}{256} & \frac{131}{256} \\ \frac{131}{256} & -\frac{110}{256} & -\frac{21}{256} \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} -128 \\ 0 \\ 0 \end{pmatrix} \tag{3.6}$$

où $-128 \leq Y,Cb,Cr \leq 127$

Lors du passage d'un modèle colorimétrique à un autre, les seules pertes possibles sont donc celles relatives aux arrondis. Différents espaces colorimétriques peuvent être utilisés (YCbCr, YUV, CIELUV, CIELAB...) ; celui qui est utilisé est spécifié dans l'en-tête.

3.3.1.2 Sous-échantillonnage

Les images couleur peuvent être sous-échantillonnées ; c'est ce qui permet de faire une première compression. L'oeil étant moins sensible aux informations colorées qu'à l'intensité lumineuse, on peut ne coder les composantes couleurs qu'un pixel sur deux ou un pixel sur 4. Il est à noter que la composante de luminance n'est jamais sous-échantillonnée.

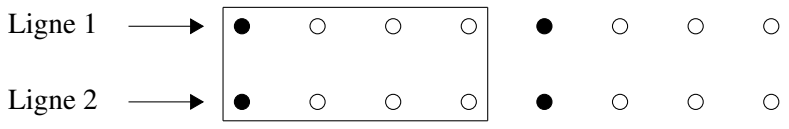
7. L'espace colorimétrique YCbCr a été développé dans Recommendation ITU-R BT.601. Dans la recommandation, $16 \leq Y \leq 235$ et $16 \leq Cb,Cr \leq 240$.

La seconde étape du pré-traitement est donc le sous-échantillonnage des composantes de chrominance suivant les lignes et/ou les colonnes.

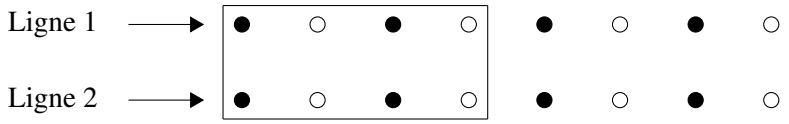
Le sous-échantillonnage est défini par la notation suivante : $J : a : b$ où J représente la largeur du bloc de référence considéré de hauteur 2, a le nombre de pixel de chrominance présents sur la première ligne et b le nombre de pixel de chrominance présents sur la seconde ligne.
Le sous-échantillonnage peut se faire de différentes manières :

- *sous-échantillonnage "2h2v" ou "*$4:1:1$*"* qui correspond à un ratio 2 : 1 horizontalement et verticalement, ce qui réduit la taille à $1/4 \times 2/3 + 1/3 = 1/2$ de la taille originale de l'image.
- *sous-échantillonnage "2h1v" ou "*$4:2:2$*"* qui correspond à un ratio 2 : 1 horizontalement et 1 : 1 verticalement, ce qui réduit la taille à $1/2 \times 2/3 + 1/3 = 2/3$ de la taille originale de l'image.

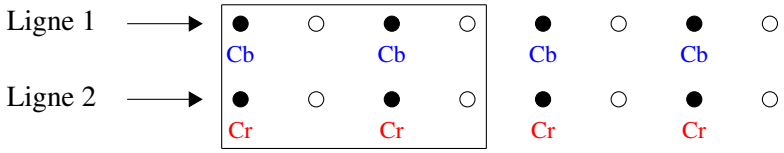
Dans la plupart des logiciels, l'échantillonnage des chrominances est 4 : 1 : 1, ce qui correspond à un gain de 50% de place.



(a) Sous-échantillonnage 4 : 1 : 1.



(b) Sous-échantillonnage 4 : 2 : 2.



(c) Sous-échantillonnage 4 : 2 : 0 (pour Cb) et 4 : 0 : 2 (pour Cr).

FIGURE 3.5 – *Sous-échantillonnage.*

Exemples :

- Microsoft® Paint® compresse toujours les chrominances en 4 : 1 : 1.
- Adobe® Photoshop® prend un facteur de qualité en paramètre (de 0 à 12) et compresse les chrominances de la qualité 0 à 6 et ne les compresse plus de 7 à 12.

Le problème avec la compression des chrominances est que dans certain cas, le résultat est médiocre. En effet, des traits de 1 pixel d'épaisseur sont très sensibles à cette compression. En revanche, ce phénomène n'arrive presque jamais pour la compression des photos car elles n'ont que très rarement des variations importantes de chrominance. Dans la plupart des cas, cette compression est totalement imperceptible.

En ce qui concerne le choix de l'échantillonnage et de la modification de la valeur des pixels, ils sont laissés au codeur.

3.3.2 Découpage en blocs

Les pixels de chaque composante sont regroupés par bloc de 64 (soit des blocs de 8 lignes et 8 colonnes de pixels). Si le nombre de lignes ou de colonnes n'est pas un multiple de 8, la dernière ligne et la dernière colonne sont dupliquées autant de fois que nécessaire (fig. 3.6). Les blocs seront traités selon la méthode choisie (interleaving [8] ou non-interleaving [9]) de gauche à droite et de haut en bas.

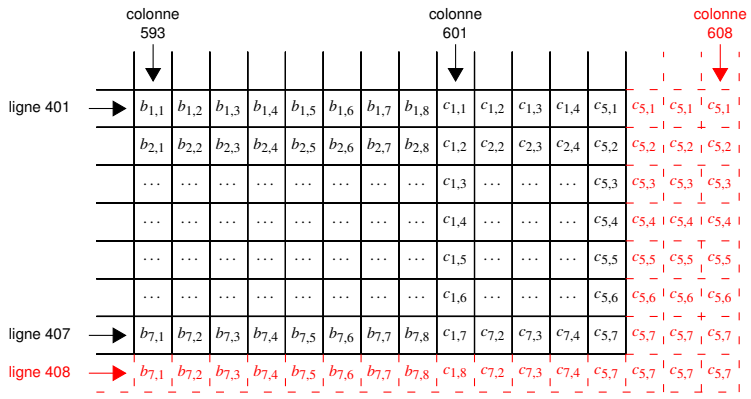


FIGURE 3.6 – *Exemple de complétion des pixels manquants pour une image de taille* 605×407 *pixels.*

Par la suite, toutes les opérations seront effectuées sur un bloc. Le codeur prendra en entrée des blocs de 8×8 pixels dont la valeur des composantes est un entier signé de l'intervalle $[-2^{q-1}, 2^{-1} - 1]$, où $q \in \{8, 12\}$ (le codage sur 12 bits étant utilisé principalement pour l'imagerie médicale). Par conséquent, si une image a été codée sur un nombre de bits différent, il faut se ramener à l'une des deux possibilités, mais cela doit être fait lors du pré-traitement. Dans le cas de JPEG Baseline, chaque pixel est codé sur 8 bits et a donc une valeur comprise entre -128 et 127 à l'issue du pré-traitement.

Le choix d'un découpage 8×8 provient d'un compromis entre la performance et la qualité de la compression.

3.3.3 Transformée en cosinus discrète

L'étape qui suit est la transposition amplitude/fréquence. Cela consiste à passer du domaine spatial (avec les intensités lumineuses des pixels) au domaine fréquentiel (avec les amplitudes des fréquences). L'opération mathématique qui fait cette transposition est la transformée en cosinus discrète (Discrete Cosine Transform ou DCT) [1, 5].

8. Supposons que les composantes de l'image sont A (de dimension $x \times y$), B (de dimension $x/2 \times y/2$) et C (de dimension $x/2 \times y/2$), l'ordre de traitement des blocs sera : A1, A2, B1, C1, ..., A_{n-1}, A_n, $B_{n/2}$ et $C_{n/2}$.

9. Supposons que les composantes de l'image sont A (de dimension $x \times y$), B (de dimension $x/2 \times y/2$) et C (de dimension $x/2 \times y/2$), l'ordre de traitement des blocs sera : A1, ..., A_n, B_1, ..., $B_{n/2}$, C_1, ..., C_{n-1} et $C_{n/2}$.

3.3.3.1 Définition et utilisation dans la compression JPEG

La DCT est une application linéaire bijective de $\mathbb{R}^N$ dans $\mathbb{R}^N$.

Comme les pixels d'une image sont corrélés dans 2 dimensions (chaque pixel est corrélé avec ses voisins horizontaux et ses voisins verticaux), les mécanismes de compression d'image utilisent la DCT bi-dimensionnelle ; elle peut être vue comme une matrice carrée $N \times N$ inversible. Elle permet de transformer des blocs de $N \times N$ pixels en un tableau de N^2 fréquences.

Soit la formule de la DCT (pour des blocs de $N \times N$) :

$$\mathrm{DCT}(i,j) = \frac{2}{N} C_i C_j \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \mathrm{pixel}(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \tag{3.7}$$

où $$C_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{pour } k = 0 \\ 1 & \text{sinon} \end{cases}$$

Dans notre cas, nous utiliserons seulement des blocs de 8×8 pixels, voici donc la formule de la DCT bi-dimensionnelle :

$$\mathrm{DCT}(i,j) = \frac{1}{4} C_i C_j \sum_{x=0}^{7} \sum_{y=0}^{7} \mathrm{pixel}(x,y) \cos\left[\frac{(2x+1)i\pi}{16}\right] \cos\left[\frac{(2y+1)j\pi}{16}\right] \tag{3.8}$$

Soit pour $N = 8$ selon Matlab,

$$DCT = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{bmatrix}$$

La transformée de cosinus discrète va être appliquée à chaque bloc pour créer un bloc de composantes fréquentielles. Les valeurs du bloc représentent la valeur moyenne de tous les pixels ainsi que les changements de fréquence à l'intérieur du bloc.

3.3.3.2 Coefficients DCT DC et AC

Le coefficient DCT(0,0) est noté DC (*direct current*) et représente la moyenne des intensités des points du bloc (composante continue (de fréquence nulle) de la décomposition), et les $N^2 - 1$ autres coefficients sont notés AC (*alternating current*). Les coefficients comportant l'information globale de l'image sont localisés dans les basses fréquences (en haut à gauche du bloc) et les coefficients situés en bas à droite représentent les hautes fréquences, qui peuvent être assimilées à du bruit.

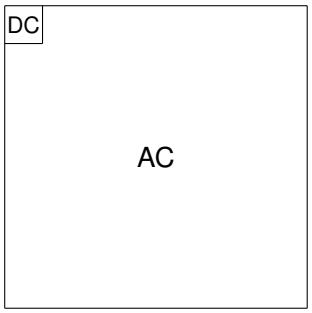


FIGURE 3.7 – *Coefficients DCT DC et AC.*

Dans le cadre de la stéganographie et de la stéganalyse, ce sont principalement les coefficients AC qui portent l'information cachée. En effet, modifier le coefficient DC reviendrait à changer la valeur moyenne du bloc, c'est à dire l'intensité moyenne des pixels, et impacterait donc davantage l'aspect visuel de l'image.

L'utilisation dans la compression JPEG de la DCT provoque des pertes d'information dues aux arrondis du calcul de la DCT ; en effet, même sans l'étape de quantification, l'utilisation de fonction trigonométrique provoque une perte d'information due à la précision limitée d'un ordinateur. La norme JPEG ne spécifie pas quel algorithme doit être utilisé pour calculer la DCT et la DCT inverse, un test est fourni (figurant parmi les tests de conformité de [40]), ce qui permet de vérifier que la précision obtenue correspond aux attentes de la norme.

3.3.4 Quantification

Le calcul des coefficients DCT de chaque bloc est suivi de l'étape de quantification. Il s'agit d'une opération destructrice : c'est elle qui génère les pertes de la compression.

Chacun des 64 coefficients DCT est divisé par un pas de quantification et est arrondi à l'entier le plus proche. La table de quantification contient les 64 pas de quantification correspondant aux 64 coefficients DCT. Les pas de quantification sont des entiers compris entre 1 et 255. La table de quantification est un des paramètres de l'algorithme de compression et peut donc être spécifiée par l'utilisateur (ou par l'application utilisée). Cependant, dans la pratique, ces coefficients utilisés sont ceux définis dans des tables de quantification fournies par le standard JPEG. Il y a une table définie pour la composante de luminance (fig. 3.8) et une autre pour les chrominances (fig. 3.9).

Le processus de quantification introduit une altération qui peut être perçue en décodant l'image. L'amplitude de la distorsion peut varier selon les pas de quantification de la table de quantification. Cela est dû à une erreur de l'ordre d'un demi pas de quantification introduite au cours du processus de décompression. Plus le pas de quantification est grand, plus la distorsion sera grande.

Chaque coefficient DCT est divisé par un quantificateur ou pas de quantification. Pour une matrice de quantification donnée, où $\Delta_{i,j}$, $i, j = 0, 1, \ldots, 7$ sont les composantes, le coefficient

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

FIGURE 3.8 – *Matrice de quantification recommandée pour la luminance.*

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

FIGURE 3.9 – *Matrice de quantification recommandée pour les chrominances.*

quantifié est défini par l'équation suivante :

$$Q[DCT(i,j)] = \left\lfloor \frac{DCT(i,j)}{\Delta_{i,j}} \right\rceil \tag{3.9}$$

où la fonction $\lfloor \cdot \rceil$ arrondit à l'entier le plus proche.

Les tables de quantification ne font pas partie de la norme JPEG, et par conséquent, chaque éditeur de logiciel, créateur d'encodeur ou utilisateur peut choisir la table de quantification de son choix. Par exemple, une matrice de quantification Δ peut être calculée en fonction d'un paramètre R (vu comme un facteur de qualité), choisi par l'utilisateur :

$$\Delta_{i,j} = 1 + (i+j) \times R \tag{3.10}$$

Cela garantit que les pas de quantification augmentent lorsque les indices augmentent.

Si l'étape de quantification est effectuée correctement, très peu de coefficients non-nuls resteront dans le bloc, et ils seront concentrés dans la partie supérieure gauche.

Après ces étapes, la forme matricielle n'est plus judicieuse pour la transmission de données ou le stockage ; il faut donc trier les valeurs dans un certain ordre et les coder afin de terminer le processus de compression JPEG.

Ce sont les coefficients DCT quantifiés qui seront utilisés en stéganographie et stéganalyse puisque les étapes suivantes dans la compression ne sont pas destructrices. C'est pourquoi, dans la mise en place de nos détecteurs (voir chapitre 4), la connaissance, et donc l'étude, de la distribution des coefficients DCT est nécessaire.

3.4 Modélisation de la distribution des coefficients DCT

La construction de détecteur peut nécessiter la connaissance de la distribution des coefficients DCT (voir chapitre 4). La modélisation des coefficients DCT est un problème intéressant qui reste ouvert.

3.4.1 État de l'art

Dans la littérature, différentes modélisations ont été proposées. En effet, Reininger et Gibson [68] proposent en 1983 une distribution gaussienne pour les coefficients DC et une distribution laplacienne pour les coefficients AC. Eggerton et Srinath[16] en 1986 proposent une distribution de Cauchy en précisant que la distribution dépendrait du type d'image et que de manière générale, la distribution de Cauchy serait un bon candidat. Müller [57] quant à lui, suggère en 1993 de modéliser les coefficients DCT par une distribution gaussienne généralisée (dont les distributions laplaciennes et gaussiennes sont des cas particuliers). Smoot et Rowe [77] confirment cela en utilisant à nouveau la distribution laplacienne. Toutes ces suggestions sont appuyées par des tests d'adéquation tels que le test du χ^2 ou le test de Kolmogorov-Smirnov.

En 2000, Lam et Goodman [49] proposent une explication sur la modélisation mathématique de la distribution gaussienne généralisée que suivraient les coefficients DCT. Ils présentent une analyse mathématique basée sur un double modèle stochastique des images. Ce modèle est présenté plus en détails dans la sous-section 3.4.2.

Dans le cadre de la stéganographie et de la stéganalyse, lorsque les coefficients DCT quantifiés sont modélisés par des distributions usuelles, il s'agit soit de la distribution de Laplace [89], de Gauss généralisée [42], ou encore de Cauchy généralisée [71].

3.4.2 Modèle proposé par Lam et Goodman

Lam et Goodman [49] ont donc proposé une justification mathématique à la distribution suivie par les coefficients DCT AC.

Tout d'abord, ils supposent que les pixels, notés $c_{p,q}$ sont identiquement distribués. Rappelons la formule 3.8 de la DCT :

$$d_{m,n} = \frac{2}{N} C_m C_n \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} c_{p,q} \cos\left[\frac{(2p+1)m\pi}{2N}\right] \cos\left[\frac{(2q+1)n\pi}{2N}\right] \tag{3.11}$$

où $d_{m,n}$ représente le coefficient DCT (m,n) et $c_{p,q}$ le pixel à la position (p,q).

Le théorème central limite leur permet d'affirmer que la distribution suivie par un coefficient DCT (somme pondérée de variables aléatoires i.i.d.) peut être approchée par une loi normale. Ainsi, ils parviennent à :

$$d_{m,n} \sim \mathcal{N}(0,\sigma^2), \qquad f(d_{m,n}|\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{d_{m,n}^2}{2\sigma^2}\right\} \tag{3.12}$$

où σ^2 est proportionnel à la variance des pixels du bloc.

La nature des images naturelles implique que la variance n'est pas constante par bloc, elle est donc également une variable aléatoire et possède une distribution propre. Un coefficient DCT, de densité de probabilité $f(d_{m,n})$, peut alors être modélisé par :

$$f(d_{m,n}) = \int_0^\infty f(d_{m,n}|\sigma^2) f(\sigma^2) d(\sigma^2). \tag{3.13}$$

Pour obtenir la distribution des coefficients AC, il suffirait donc de connaître la distribution de la variance d'un bloc, ce qui n'est pas chose facile et dépasse le cadre de cette thèse.

Cependant, Lam et Goodman montre que si la variance suit une loi exponentielle ou demi-gaussienne, alors on peut se ramener à une distribution laplacienne et donc à une gaussienne généralisée (puisque la distribution laplacienne est un cas particulier).

En effet, si nous considérons une distribution exponentielle, i.e. $f(\sigma^2) = \lambda \exp\{-\lambda\sigma^2\}$, pour une moyenne non nulle μ, la distribution des coefficients DCT devient :

$$\begin{aligned}
f(d_{m,n}) &= \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(d_{m,n}-\mu)^2}{2\sigma^2}\right\} \lambda \exp\{-\lambda\sigma^2\} d(\sigma^2) \\
&= \sqrt{\frac{2}{\pi}}\lambda \int_0^\infty \exp\left\{-\lambda\sigma^2 - \left(\frac{(d_{m,n}-\mu)^2}{2}\right)\frac{1}{\sigma^2}\right\} d\sigma \\
&= \left(\sqrt{\frac{2}{\pi}}\lambda\right)\left(\frac{1}{2}\sqrt{\frac{\pi}{\lambda}}\right) \exp\left\{-2\sqrt{\lambda\frac{(d_{m,n}-\mu)^2}{2}}\right\} \\
&= \frac{\sqrt{2\lambda}}{2} \exp\left\{-\sqrt{2\lambda}|d_{m,n}-\mu|\right\}
\end{aligned} \tag{3.14}$$

Nous obtenons une distribution laplacienne de paramètre d'échelle $b = \frac{1}{\sqrt{2\lambda}}$ et de moyenne μ. Un calcul analogue montrerait que pour une distribution demi-gaussienne de la variance, la distribution des coefficients DCT AC peut être approchée par une distribution laplacienne.

Remarque 3.1. *L'approche proposée par Lam et Goodman a été complétée par S. Nadarajah qui modélise la distribution suivie par les coefficients DCT en fonction de différentes lois que pourrait suivre la variance d'un bloc [59, 58]. Les estimations des paramètres par la méthode des moments et du maximum de vraisemblance sont également présentées.*

Par exemple, si la variance d'un bloc est modélisée par la distribution gamma de paramètres β et μ :

$$f(\sigma^2) = \frac{(\sigma^2)^{\beta-1} \exp(-\sigma^2/\mu)}{\mu^\beta \Gamma(\beta)} \tag{3.15}$$

La distribution obtenue pour les coefficients DCT est :

$$f(d_{m,n}) = \frac{d_{m,n}^{\beta-\frac{1}{2}}}{\sqrt{\pi}\,\Gamma(\beta)\mu^{\frac{\beta}{2}+\frac{1}{4}}2^{\frac{\beta}{2}-\frac{3}{4}}\mathbf{K}_{\beta-\frac{1}{2}}\left(2\sqrt{\frac{2d_{m,n}^2}{\mu}}\right)}, \qquad \text{avec } \beta > 0 \text{ et } \mu > 0 \tag{3.16}$$

β et μ sont respectivement les paramètres de forme et d'échelle et $\mathbf{K}$ représente la fonction de Bessel.

3.4.3 Modèle laplacien quantifié

La distribution laplacienne étant un bon compromis entre la simplicité du modèle et la fidélité aux données empiriques, nous proposons un détecteur basé sur cette distribution dans le chapitre 4.

La distribution laplacienne continue de moyenne a et de paramètre d'échelle b est donnée par :

$$f_{a,b}(x) = \frac{1}{2b}\exp\left(\frac{-|x-a|}{b}\right), \forall x \in \mathbb{R}. \tag{3.17}$$

Il a été montré [49] que la distribution des coefficients DCT est symétrique et de moyenne nulle. Ainsi, seul le paramètre b est inconnu.

Dans le cas des images numériques, et en particulier pour les images JPEG, les données sont quantifiées, nous considérons donc des distributions discrètes et non continues.

Soit b le paramètre d'échelle, la densité de probabilité $q_{d_{m,n}}(b,\Delta)$ du coefficient DCT $d_{m,n}$ modélisé par une distribution laplacienne discrète quantifiée avec le pas de quantification Δ est obtenue par :

$$q_{d_{m,n}}(b,\Delta) = \begin{cases} \exp\left(\dfrac{-\Delta|d_{m,n}|}{b}\right)\sinh\left(\dfrac{\Delta}{2b}\right) & \text{si } x \neq 0, \\ 1-\exp\left(-\dfrac{\Delta}{2b}\right) & \text{si } x = 0. \end{cases} \tag{3.18}$$

Conclusion

Dans ce chapitre, les images brutes mais également les images compressées ont été modélisées dans le but de pouvoir construire des détecteurs d'informations cachées fiables. En particulier, les modélisations des coefficients DCT des images JPEG présentes dans la littérature ont été présentées. Ainsi, dans le chapitre 4 sont présentés différents détecteurs dans le domaine spatial et dans le domaine fréquentiel.

Chapitre 4

Détection statistique de stéganographie

Sommaire

Introduction

L'objectif du chapitre 4 est de présenter les détecteurs mis en place à l'aide des outils de décision statistique introduits dans le chapitre 2 et des modèles d'images présentés dans le chapitre 3. Dans une première section 4.1, un détecteur basé sur la modélisation des coefficients DCT quantifiés est proposé. Un test basé sur la modélisation laplacienne est introduit puis une généralisation de ce test à une distribution quelconque est proposé. Ensuite, la section 4.2 présente une étude liée au phénomène d'effondrement qui se produit lors de l'utilisation des algorithmes F3, F4 et F5 présentés dans le chapitre 2. Une approche asymptotique permettant de calculer le nombre de 0 introduits par ce phénomène est présentée. Enfin, différents tests statistiques sont construits dans le domaine spatial en s'appuyant sur les outils présentés dans le chapitre 2 et leurs propriétés statistiques étudiées dans la section 4.3.

4.1 Test basé sur la modélisation des coefficients DCT quantifiés

Dans cette section est proposé un test basé sur l'hypothèse que les coefficients DCT quantifiés suivent une certaine distribution. Pour commencer, nous considérerons la distribution laplacienne quantifiée, puis nous généraliserons l'approche à d'autres lois.

Présentation du test

L'objet du problème de détection auquel nous tâchons de trouver une solution, est la construction d'un test statistique qui détecte des informations cachées en utilisant l'algorithme Jsteg dans un ensemble de vecteurs de coefficients DCT supposés indépendants.

Lorsqu'une image est analysée, deux scenarii sont possibles :

$\mathcal{H}_0$: l'image est un medium de couverture,
$\mathcal{H}_1$: l'image est stéganographiée avec un taux d'insertion R.

Afin de construire un détecteur, nous devons connaître la distribution des coefficients DCT d'une image naturelle. Comme nous avons pu le voir dans le chapitre précédent, il s'agit d'un problème intéressant qui reste ouvert. Soit $V_k = \{v_{k,1}, \dots, v_{k,n}\}$ un vecteur de longueur n composé de la valeur du $k^{\text{ième}}$ coefficient de chaque bloc (voir Fig. 4.1), n_k représente le nombre de coefficients utilisables (i.e. différents de 0 et 1), le vecteur peut être permuté de sorte à ce que les n_k premiers coefficients soient les coefficients utilisables. Notons P_{θ_k,Δ_k} la loi de paramètre θ_k quantifiée avec Δ_k le pas de quantification associé à la k-ième fréquence.

Le test d'hypothèses considéré s'écrit comme suit :

$$\begin{aligned} \mathcal{H}_0 &: \{v_{k,i} \sim P_{\theta_k,\Delta_k}, \forall k = 2,\dots,64, \forall i = 1,\dots,n_k\} \\ \mathcal{H}_1 &: \{v_{k,i} \sim P_{\theta_k,\Delta_k,R}, \forall k = 2,\dots,64, \forall i = 1,\dots,n_k\}. \end{aligned} \tag{4.1}$$

Nous recherchons le test le plus puissant dans la classe $\mathcal{K}_{\alpha_0}$ définie par :

$$\mathcal{K}_{\alpha_0} = \left\{\delta^\star : P_{\mathcal{H}_0}(\delta^\star(V) = \mathcal{H}_1) < \alpha_0\right\},$$

où $V = \{V_2, \ldots, V_{64}\}$ pour résoudre le problème (4.1). Si nous supposons les paramètres θ_k connus, le test est donné par le lemme de Neyman-Pearson et consiste en la règle de décision suivante :

$$\delta^\star(V) = \begin{cases} \mathcal{H}_0 & \text{if} \quad \Lambda^\star(V) = \sum_{k=2}^{64} \log \Lambda_k(V_k) < h, \\ \mathcal{H}_1 & \text{if} \quad \Lambda^\star(V) = \sum_{k=2}^{64} \log \Lambda_k(V_k) \geq h. \end{cases} \tag{4.2}$$

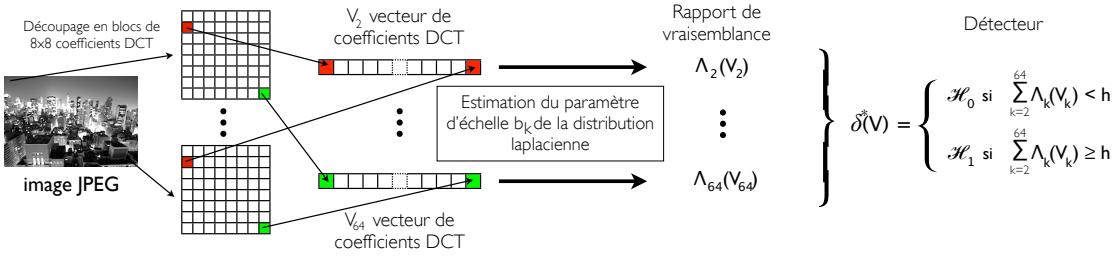


FIGURE *4.1* – *Schéma du détecteur.*

4.1.1 Test basé sur la modélisation laplacienne des coefficients DCT

Comme cela a été présenté dans la section 3.4.3, la distribution laplacienne continue de paramètre d'échelle b est donnée par :

$$f_b(x) = \frac{1}{2b} \exp\left(\frac{-|x|}{b}\right), \forall x \in \mathbb{R}. \tag{4.3}$$

Soit b_k le paramètre d'échelle, la densité de probabilité Lap (b_k, Δ_k) du coefficient DCT $v_{k,i}$ modélisé par une distribution laplacienne discrète quantifiée avec le pas de quantification Δ_k est obtenue par :

$$\mathbb{P}\left(y = v_{k,i} | y \notin \{0,1\}\right) = q_{v_{k,i}}(b_k, \Delta_k) = \exp\left(\frac{-\Delta_k |v_{k,i}|}{b_k}\right) \sinh\left(\frac{\Delta_k}{2b_k}\right). \tag{4.4}$$

Comme la loi est symétrique, $q_{v_{k,i}}(b_k, \Delta_k) = q_{-v_{k,i}}(b_k, \Delta_k)$.

Nous proposons d'étudier le test dans le cas où la distribution suivie par les coefficients DCT est une distribution laplacienne quantifiée.

Lors de l'insertion d'un message, le coefficient DC n'est jamais utilisé ; pour les autres, seules les valeurs différentes de 0 et 1 sont utilisables. Par conséquent, le nombre de coefficients utiles de chaque vecteur est représenté par une variable aléatoire $n_k \leq n$, qui dépend de l'image analysée (principalement de son contenu) et de la matrice de quantification. On peut montrer que n_k est une variable aléatoire positive à valeurs entières qui suit une loi binomiale $\mathcal{B}(n, p_k^*)$,

où p_k^* est la probabilité qu'un coefficient $v_{k,i}$ du vecteur V_k soit utilisable. Par convention, les n_k premiers éléments du vecteur sont des valeurs utilisables et les $n - n_k$ derniers sont les valeurs interdites.

Le test d'hypothèses 4.1 devient :

$$\begin{aligned} &\mathcal{H}_0 : \{v_{k,i} \sim \text{Lap}\left(b_k, \Delta_k\right), \forall k = 2, \ldots, 64, \forall i = 1, \ldots, n_k\}, \\ &\mathcal{H}_1 : \{v_{k,i} \sim \text{Lap}\left(b_k, \Delta_k, R\right), \forall k = 2, \ldots, 64, \forall i = 1, \ldots, n_k\}. \end{aligned} \tag{4.5}$$

Après insertion par écrasement des LSBs, la densité de probabilité d'un coefficient est notée $\text{Lap}\left(b_k, \Delta_k, R\right)$ où R est le taux d'insertion :

$$\begin{aligned} \mathbb{P}^R\left(D = v_{k,i} | D \notin \{0,1\}\right) &= q^R_{b_k,\Delta_k}(v_{k,i}) \\ &= \left(1 - \frac{R}{2}\right) q_{b_k,\Delta_k}(v_{k,i}) + \frac{R}{2} q_{b_k,\Delta_k}(\bar{v}_{k,i}), \end{aligned} \tag{4.6}$$

où $\bar{v}_{k,i} = v_{k,i} + (-1)^{v_{k,i}}$ est le coefficient $v_{k,i}$ avec le LSB inversé. Le taux d'insertion est la probabilité qu'un coefficient soit utilisé pour l'insertion.

4.1.1.1 Calcul du rapport de vraisemblance

$$\begin{aligned} \Lambda_k(V_k) &= \Lambda_k(v_{k,1}, \ldots, v_{k,n_k}) \\ &= \prod_{i=1}^{n_k} \frac{q^R_{b_k,\Delta_k}(v_{k,i})}{q_{b_k,\Delta_k}(v_{k,i})} \\ &= \prod_{i=1}^{n_k} \frac{\left(1 - \frac{R}{2}\right) q_{b_k,\Delta_k}(v_{k,i}) + \frac{R}{2} q_{b_k,\Delta_k}(\bar{v}_{k,i})}{q_{b_k,\Delta_k}(v_{k,i})} \\ &= \prod_{i=1}^{n_k} \left(1 - \frac{R}{2} + \frac{R}{2} \frac{q_{b_k,\Delta_k}(\bar{v}_{k,i})}{q_{b_k,\Delta_k}(v_{k,i})}\right) \\ &= \prod_{i=1}^{n_k} \left(1 - \frac{R}{2} + \frac{R}{2} \frac{\exp\left(\frac{-\Delta_k|\bar{v}_{k,i}|}{b_k}\right) \sinh\left(\frac{\Delta_k}{2b_k}\right)}{\exp\left(\frac{-\Delta_k|v_{k,i}|}{b_k}\right) \sinh\left(\frac{\Delta_k}{2b_k}\right)}\right) \\ &= \prod_{i=1}^{n_k} \left(1 - \frac{R}{2} + \frac{R}{2} \exp\left(\frac{\Delta_k|v_{k,i}|}{b_k} - \frac{\Delta_k|\bar{v}_{k,i}|}{b_k}\right)\right) \end{aligned} \tag{4.7}$$

$$\log \Lambda_k(V_k) = \sum_{i=1}^{n_k} \log\left(1 - \frac{R}{2} + \frac{R}{2} \exp\left(\frac{\Delta_k}{b_k}\Big(\underbrace{\text{signe}(v_{k,i})(v_{k,i} - \bar{v}_{k,i})}_{=\zeta_{k,i}}\Big)\right)\right). \tag{4.8}$$

La variable aléatoire $\zeta_{k,i}$ représente l'impact de l'insertion et vaut 1 ou -1 selon la parité et le signe de $v_{k,i}$. Cela vient de la nature asymétrique de l'insertion avec Jsteg (voir Fig. 2.5) : les coefficients pairs peuvent seulement être incrémentés et les coefficients impairs décrémentés lorsque le LSB est écrasé, de plus, la valeur du LSB est inversée selon le signe du coefficient. Par exemple, si $v_{k,i}$ est positif et impair, le LSB vaut 0 alors que le LSB vaut 1 si $v_{k,i}$ est impair négatif.

4.1.1.2 Test de Neyman-Pearson

Le théorème suivant établit les probabilités statistiques du test de Neyman-Pearson.

Théorème . *Le test (4.2) atteint la puissance $\beta_{\delta^\star}$ lorsque n tend vers l'infini, pour un seuil de décision h défini par α_0 :*

$$\beta_{\delta^\star} \simeq 1 - \Phi\left(\frac{\sigma_0^\star}{\sigma_1^\star}\Phi^{-1}(1-\alpha_0) + \frac{\mu_0^\star - \mu_1^\star}{\sigma_1^\star}\right) \tag{4.9}$$

où $h \simeq \sigma_0^\star\,\Phi^{-1}(1-\alpha_0) + \mu_0^\star$, *et où* $\mu_j^\star = \sum_{k=2}^{64} \mu_{k,j}$ *et* $\sigma_j^{\star 2} = \sum_{k=2}^{64} \sigma_{k,j}^2$.

$$\mu_{k,j} = n p_k^* \left[a_k(1-p_{k,j}) + c_k p_{k,j}\right], \tag{4.10}$$

$$et\ \sigma_{k,j}^2 = n p_k^* \left[a_k^2(1-p_k^*) + (c_k - a_k)^2 p_k^* p_{k,j}(1 - p_k^* p_{k,j})\right]. \tag{4.11}$$

Le seuil h est la solution de l'équation $P_0(\Lambda^\star(V) \geq h) = \alpha_0$. Le rapport $\frac{\mu_0^\star - \mu_1^\star}{\sigma_1^\star}$ mesure la séparabilité entre les deux hypothèses $\mathcal{H}_0$ et $\mathcal{H}_1$. Il dépend de R, b_k et Δ_k. Ce théorème est asymptotique car n est proportionnel au nombre de pixels de l'image, et pour une grande image, on peut considérer que n tend vers l'infini.

Démonstration. Pour tenir compte de la présence de valeurs interdites dans le vecteur de coefficients DCT, il est nécessaire de considérer la variable aléatoire n_k dans les calculs de performance du détecteur.

Pour une fréquence k donnée, le log-rapport de vraisemblance est :

$$\log \Lambda_k(v_{k,1}, \ldots, v_{k,n_k}) = \sum_{i=1}^{n_k} X_{k,i} = \sum_{i=1}^{n_k} \log\left(1 - \frac{R}{2} + \frac{R}{2}\exp\left(\frac{\Delta_k}{b_k}\zeta_{k,i}\right)\right) \tag{4.12}$$

avec $\zeta_{k,i}$ une variable aléatoire qui vaut 1 avec une probabilité p_k et qui vaut -1 avec une probabilité $1 - p_k$. On note p_k la probabilité que $v_{k,i}$ est impair positif ou pair négatif :

$$p_k = \mathbb{P}(v_{k,i} \equiv 0 \bmod 2, v_{k,i} < 0) + \mathbb{P}(v_{k,i} \bmod 2, v_{k,i} > 0) \tag{4.13}$$

Le log-rapport de vraisemblance peut s'écrire :

$$\begin{aligned} \log \Lambda_k(v_{k,1}, \ldots, v_{k,n_k}) &= n_k a_k + (c_k - a_k)\sum_{i=1}^{n_k} Y_{k,i} \\ &= n_k a_k + (c_k - a_k) S_{n_k}, \end{aligned} \tag{4.14}$$

avec $a_k = \log\left(1 - \frac{R}{2} + \frac{R}{2}\exp\left(-\frac{\Delta_k}{b_k}\right)\right)$, et $c_k = \log\left(1 - \frac{R}{2} + \frac{R}{2}\exp\left(\frac{\Delta_k}{b_k}\right)\right)$.

Les variables aléatoires $Y_{k,i} = \frac{X_{k,i} - a_k}{c_k - a_k}$ sont indépendantes, identiquement distribuées et suivent une distribution de Bernoulli de moyenne p_k et de variance $p_k(1-p_k)$.

Soit $S_{n_k} = \sum_{i=1}^{n_k} Y_{k,i}$. Comme $Y_{k,i}$ suit la loi de Bernoulli $\mathcal{B}(1, p_k)$, S_{n_k} suit la loi binomiale $\mathcal{B}(n_k, p_k)$. Notons que la variable aléatoire n_k suit la distribution binomiale $\mathcal{B}(n, p_k^*)$, où p_k^* est la probabilité que le coefficient $v_{k,i}$ du vecteur V_k soit utilisable, i.e. $v_{k,i} \notin \{0, 1\}$. Il résulte de cela que S_{n_k} suit la distribution binomiale $\mathcal{B}(n, p_k^* p_k)$. En effet, une loi binomiale de paramètres (m, α) dont le paramètre m résulte d'une loi binomiale de paramètres (n, p) est assimilable à une loi binomiale de paramètres $(n, p\alpha)$.

Comme n est grand et que $p_k^* p_k \in]0, 1[$, selon le théorème de De Moivre-Laplace, S_{n_k} converge en loi vers une loi normale. Ainsi, pour chaque fréquence, nous obtenons :

$$\frac{S_{n_k} - n p_k^* p_k}{\sqrt{n p_k^* p_k (1 - p_k^* p_k)}} \xrightarrow[n\to\infty]{\mathcal{L}} \mathcal{N}(0, 1) \tag{4.15}$$

L'erreur d'approximation gaussienne est majorée par :

$$\frac{C}{\sqrt{n p_k^* p_k (1 - p_k^* p_k)}}, \text{ avec } C < 0,4784\,^{1}. \tag{4.16}$$

D'où

$$\frac{\log \Lambda_k - \mu_k}{\sigma_k} \xrightarrow[n\to\infty]{\mathcal{L}} \mathcal{N}(0, 1), \tag{4.17}$$

où $\xrightarrow[n\to\infty]{\mathcal{L}}$ représente la convergence en loi et avec :

$$\mu_{k,j} = n p_k^* \left[a_k (1 - p_k) + c_k p_k \right], \tag{4.18}$$

$$\text{et } \sigma_{k,j}^2 = n p_k^* \left[a_k^2 (1 - p_k^*) + (c_k - a_k)^2 p_k^* p_k (1 - p_k^* p_k) \right]. \tag{4.19}$$

En appliquant le théorème central limite à la somme $\Lambda^\star$ des log-rapports de vraisemblance, sous l'hypothèse $\mathcal{H}_j$, le détecteur optimal est donné par :

$$\frac{\Lambda^\star(V) - \mu_j^\star}{\sigma_j^\star} \xrightarrow[n\to\infty]{\mathcal{L}} \mathcal{N}(0, 1), \tag{4.20}$$

où $\mu_j^\star = \sum_{k=2}^{64} \mu_{k,j}$ et $\sigma_j^{\star 2} = \sum_{k=2}^{64} \sigma_{k,j}^2$.

□

4.1.2 Généralisation à une distribution quelconque

Soit P la distribution de paramètre Θ quantifiée avec le pas de quantification Δ et P^R la distribution après insertion par la méthode d'écrasement des LSB pour un taux d'insertion R.

1. Valeur de la constante C obtenue par Asof en 2011.

$$
\begin{aligned}
P\left(y = v_i | y \notin \{0,1\}\right) &= q_{\Theta,\Delta}(v_i) && (4.21)\\
P^R\left(y = v_i | y \notin \{0,1\}\right) &= q^R_{\Theta,\Delta}(v_i) && (4.22)\\
&= \left(1 - \frac{R}{2}\right) q_{\Theta,\Delta}(v_i) + \frac{R}{2} q_{\Theta,\Delta}(\bar{v}_i) && (4.23)
\end{aligned}
$$

Le test d'hypothèses considéré pour une fréquence est :

$$
\begin{aligned}
&\mathcal{H}_0 : \{v_{k,i} \sim q_{\Theta_k,\Delta_k}, \forall k = 2,\ldots,64, \forall i = 1,\ldots,n_k\}\\
&\mathcal{H}_1 : \{v_{k,i} \sim q^R_{\Theta_k,\Delta_k}, \forall k = 2,\ldots,64, \forall i = 1,\ldots,n_k\}.
\end{aligned} \tag{4.24}
$$

Calcul pour un vecteur de coefficients AC

Le rapport de vraisemblance obtenu pour le k-ième coefficient AC est :

$$
\begin{aligned}
\Lambda_k(V_k) &= \Lambda_k(v_{k,1},\ldots,v_{k,n_k})\\
&= \prod_{i=1}^{n_k} \frac{q^R_{\Theta_k,\Delta_k}(v_{k,i})}{q_{\Theta_k,\Delta_k}(v_{k,i})}\\
&= \prod_{i=1}^{n_k} \frac{\left(1 - \frac{R}{2}\right) q_{\Theta_k,\Delta_k}(v_{k,i}) + \frac{R}{2} q_{\Theta_k,\Delta_k}(\bar{v}_{k,i})}{q_{\Theta_k,\Delta_k}(v_{k,i})} && (4.25)\\
nonumber &= \prod_{i=1}^{n_k} \left(1 - \frac{R}{2} + \frac{R}{2} \frac{q_{\Theta_k,\Delta_k}(\bar{v}_{k,i})}{q_{\Theta_k,\Delta_k}(v_{k,i})}\right)\\
S_{n_k} = \log \Lambda_k(V_k) &= \sum_{i=1}^{n_k} \log\left(1 - \frac{R}{2} + \frac{R}{2} \frac{q_{\Theta_k,\Delta_k}(\bar{v}_{k,i})}{q_{\Theta_k,\Delta_k}(v_{k,i})}\right) && (4.26)\\
&= \sum_{i=1}^{n_k} Y_{k,i}
\end{aligned}
$$

Nous pouvons calculer, l'espérance et la variance de la variable aléatoire Y_k sous l'hypothèse $\mathcal{H}_0$:

$$
\begin{aligned}
\mu_{k,0} &= \mathbb{E}_0\left[Y_k\right] = \sum_{v_{k,i} \in \mathbb{Z}} Y_{k,i} \cdot q_{\Theta_k,\Delta_k}(v_{k,i}) && (4.27)\\
\sigma^2_{k,0} &= \mathrm{Var}_0\left[Y_k\right] = \sum_{v_{k,i} \in \mathbb{Z}} \left(Y_{k,i} - \mu_{k,0}\right)^2 \cdot q_{\Theta_k,\Delta_k}(v_{k,i}) && (4.28)
\end{aligned}
$$

où $\mathbb{E}_j$ et Var_j représentent l'espérance et la variance sous l'hypothèse $\mathcal{H}_j$, $j = \{0,1\}$.

De la même manière, sous l'hypothèse $\mathcal{H}_1$, l'espérance et la variance de la variable aléatoire Y_k sont données par :

$$\mu_{k,1} = \mathbb{E}_1[Y_k] = \sum_{v_{k,i}\in\mathbb{Z}} Y_{k,i}\cdot q^R_{\Theta_k,\Delta_k}(v_{k,i}), \tag{4.29}$$

$$\sigma^2_{k,1} = \mathrm{Var}_1[Y_k] = \sum_{v_{k,i}\in\mathbb{Z}} (Y_{k,i}-\mu_{k,1})^2\cdot q^R_{\Theta_k,\Delta_k}(v_{k,i}). \tag{4.30}$$

Prise en compte de tous les vecteurs de coefficients AC

Le test optimal au sens de Neyman-Pearson, c'est-à-dire le test le plus puissant dans la classe $\mathcal{K}_\alpha$ (où α est la probabilité de fausse alarme fixée), consiste en la règle de décision suivante :

$$\delta^\star(V) = \begin{cases} \mathcal{H}_0 \quad \text{si} \quad \Lambda^\star(V) = \sum_{k=2}^{64} S_{n_k} < h, \\ \mathcal{H}_1 \quad \text{si} \quad \Lambda^\star(V) = \sum_{k=2}^{64} S_{n_k} \geq h, \end{cases} \tag{4.31}$$

où le seuil h est solution de l'équation $\mathbb{P}_0(\Lambda^\star(V)\geq h) = \alpha$.

L'étude théorique des probabilités du détecteur est plus compliquée dans le cas d'une distribution quelconque que dans le cas d'une distribution laplacienne et fera l'objet de futurs travaux.

4.2 Étude du phénomène d'effondrement

Lorsqu'un détecteur est mis en place en considérant que tout est connu, nous supposons connue la distribution des coefficients DCT sous l'hypothèse $\mathcal{H}_0$, c'est à dire lorsque l'image est saine. Connaissant l'algorithme d'insertion, nous pouvons connaître la distribution des coefficients DCT de l'image lorsque l'image est stéganographiée. Dans cette section, nous nous proposons d'étudier la distribution des coefficients DCT lorsque le phénomène d'effondrement apparaît, c'est le cas pour les algorithmes F3 et F4 (voir sections 2.1.2.3 et 2.1.2.4). Pour l'algorithme F5, la difficulté supplémentaire introduite est l'utilisation du codage de Hamming dans le but de diminuer l'impact de l'insertion.

Afin de connaître la distribution des coefficients DCT après insertion, nous allons calculer la probabilité d'apparition de chaque coefficient. Pour cela, nous devons connaître le nombre d'effondrements qui peuvent apparaître lors de l'insertion, et par conséquent, commencer par étudier mathématiquement les algorithmes d'insertion.

4.2.1 Représentation mathématique des algorithmes F3 et F4

Contrairement à Jsteg, présenté dans la section 2.1.2.2, la méthode utilisée par les algorithmes F3 et F4 n'est pas une substitution, mais une décrémentation : si le bit à insérer ne correspond pas au bit de poids faible de la valeur du coefficient DCT, la valeur absolue du coefficient est décrémentée.

Soit $\mathcal{A}$ une fonction qui représente l'insertion d'un bit du message par l'algorithme dans un coefficient. Elle prend en entrée x et m qui correspondent respectivement à la valeur d'un coefficient DCT quantifié et au bit à insérer. Cette fonction renvoie le coefficient DCT obtenu correspondant aux entrées passées dans l'algorithme.

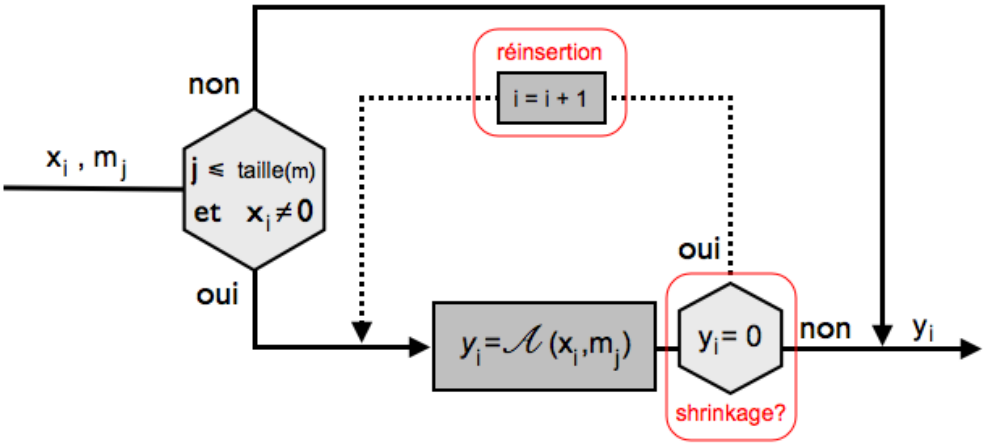


FIGURE 4.2 – *Schéma des algorithmes F3 et F4*

Le schéma 4.2 illustre l'insertion d'un bit du message dans un coefficient DCT avec F3 ou F4 : si le coefficient x_i est non nul, on insère le bit m_j dans le coefficient. Si le bit de poids faible du coefficient est différent du bit inséré, on décrémente la valeur absolue du coefficient. Si la nouvelle valeur y_i du coefficient est 0 (il y a effondrement), on réinsère le bit du message dans le prochain coefficient. La différence entre F3 et F4 réside dans la fonction $\mathcal{A}$, où le codage des LSB est modifié.

4.2.1.1 Algorithme F3

L'algorithme d'**insertion** de F3 peut être représenté par la fonction suivante :

$$\begin{array}{rccl} & K \times J & \longrightarrow & K \\ \mathcal{A} \ : & (x,m) & \longmapsto & y = \mathcal{A}(x,m) \end{array}$$

$$(x,m) \longmapsto \begin{cases} x & \text{si } m = \mathrm{mod}(x,2) \\ x - \dfrac{x}{|x|} & \text{sinon.} \end{cases}$$

où $K = [-1024, 1023]\backslash\{0\}$ et $J = \{0,1\}$

Tous les coefficients peuvent être utilisés pour l'insertion, à l'exception du 0, puisque cela modifierait la compression de l'image. Si au cours de l'insertion, un bit valant 0 dans un coefficient dont la valeur est 1 ou -1, un effondrement apparaît : le coefficient devient nul puisque la valeur absolue est décrémentée. Or, les 0 n'étant pas utilisés pour l'insertion, ils ne sont pas utilisés pour retrouver le message inséré (voir chapitre 2). Ainsi, le bit précédemment inséré ne sera

pas récupéré ; il faut donc le réinsérer dans le prochain coefficient disponible. Ce phénomène est mis en évidence dans le schéma 4.2.

4.2.1.2 Algorithme F4

La différence entre les algorithmes F3 et F4 réside dans l'écriture binaire des nombres négatifs. En effet, les effondrements apparaissent avec F3 lors de l'insertion d'un 0 dans un 1 ou dans un -1 comme le montre le schéma d'insertion 2.7 page 24.

Pour F4, on considère que le bit de poids faible d'un nombre négatif impair est 0 et que celui d'un nombre pair est 1. Ainsi, en modifiant la représentation des nombres négatifs, les effondrements apparaissent lors de l'insertion d'un 0 dans un 1 et d'un 1 dans un -1 (voir schéma 2.9).

4.2.2 Motivation de l'étude

Lors de la construction d'un détecteur où tout est connu, nous supposons connus la distribution suivie par les coefficients DCT, l'algorithme utilisé et le taux d'insertion. Pour réaliser ce détecteur, la connaissance de la distribution des coefficients DCT après insertion est nécessaire. Pour cela, nous avons besoin de connaître l'algorithme utilisé et de modéliser l'impact de l'insertion sur la distribution des coefficients DCT. Après avoir vu plus précisément quand apparaissait un effondrement lors de l'utilisation des algorithmes F3 et F4, nous souhaitons calculer les probabilités d'apparition des différents coefficients après insertion. Pour cela, commençons par étudier le nombre moyen d'effondrements qui apparaissent lors de l'insertion.

4.2.3 Étude du nombre moyen d'effondrements

Nous considérons que le message est inséré dans toute l'image et que la distribution du support est connue pour l'étude. Pour calculer le nombre moyen d'effondrements, nous utilisons une approche asymptotique : nous considérons que l'image dans laquelle est insérée le message est très grande.

Sur les figures 4.3 et 4.4, les notations M, X, Y, τ_1 et τ_2 représentent respectivement :

- M : variable aléatoire discrète représentant le bit du message inséré, $M \in \{0, 1\}$, $M \sim B(1, p)$ avec p la probabilité que le bit soit 0.
- X : variable aléatoire discrète représentant la valeur du coefficient DCT utilisé, $X \in [-1023, 1024]$,
- Y : variable aléatoire discrète représentant la valeur du coefficient DCT après insertion, $Y \in [-1023, 1024]$,
- τ_1 : variable aléatoire discrète qui décrit le nombre de bits insérés successivement avant l'apparition du premier effondrement, $\tau_1 \in \mathbb{N}$,
- τ_2 : variable aléatoire discrète qui décrit le nombre d'effondrements successifs qui peuvent apparaître, $\tau_2 \in \mathbb{N}$.

De plus, dans toute la section 4.2 nous utiliserons les notations suivantes :

- $\mathrm{Nb}_{\mathrm{eff}}$: variable aléatoire qui représente le nombre total d'effondrements qui peuvent apparaître lors de l'insertion,
- $S = \tau_1 + \tau_2$: variable aléatoire discrète qui représente une séquence, i.e. décrit la durée au bout de laquelle un effondrement est résolu, $S \in \mathbb{N}$,
- N : variable aléatoire qui représente le nombre de séquences,
- $\mathrm{Nb}_{\mathrm{coef}}$: variable aléatoire qui représente le nombre de coefficients utilisables.

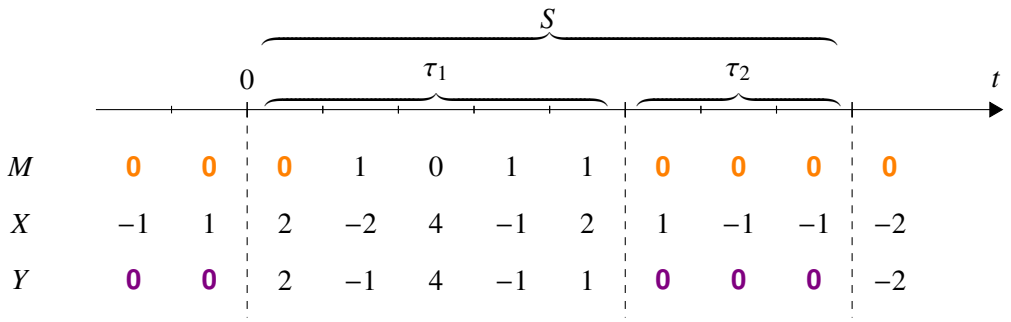


FIGURE 4.3 – *Séquence d'insertion pour F3.*

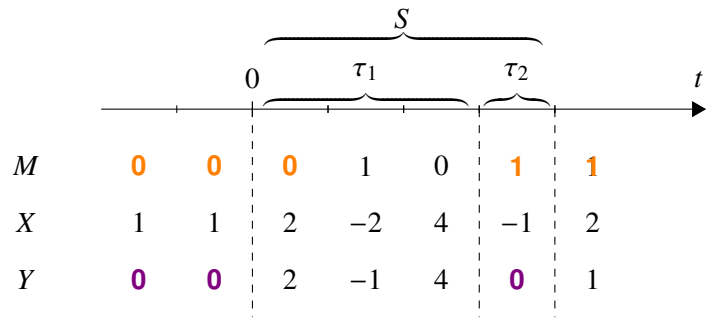


FIGURE 4.4 – *Séquence d'insertion pour F4.*

Une séquence d'insertion est une succession d'insertions que l'on peut scinder en deux phases : la première correspond à une insertion *normale* (aucun effondrement n'est apparu) et la deuxième, qui est une succession d'effondrements. τ_1 est le temps au bout duquel apparaît le premier effondrement, et τ_2 est la durée au bout de laquelle l'effondrement est résolu [2].

Pour calculer la probabilité qu'il y ait un effondrement en un coefficient donné, nous allons estimer $\mathbb{E}(\tau_1)$, le nombre de bits moyen insérés avant le premier effondrement, puis $\mathbb{E}(\tau_2)$, le nombre d'effondrements successifs.

2. On dit que l'effondrement est résolu si le 0 qui doit être inséré, est finalement inséré dans un coefficient sans produire de nouvel effondrement.

4.2.3.1 Calcul de $\mathbb{E}(\tau_1)$ et de $\mathbb{E}(\tau_2)$

Nous souhaitons connaître la durée moyenne d'une insertion sans effondrement $\mathbb{E}(\tau_1)$ ainsi que la durée moyenne d'un effondrement $\mathbb{E}(\tau_2)$. Pour cela, nous avons besoin de la probabilité $\mathbb{P}(\tau_1 = n)$ qui correspond au fait d'avoir n bits insérés exactement dans des coefficients DCT avant d'avoir un effondrement. De même, $\mathbb{P}(\tau_2 = n)$ est la probabilité d'avoir exactement n effondrements successifs sachant qu'il y en a eu un premier.

Pour l'algorithme F3

Comme le présente la figure 2.7, un effondrement apparaît lorsqu'un 0 est inséré dans un coefficient valant -1 ou 1, c'est à dire dans un coefficient de valeur absolue égale à 1. De plus, pour rappel, les coefficients dont la valeur est 0 ne sont pas utilisés pour l'insertion, les probabilités calculées sont donc conditionnées par $X \neq 0$.

Soit p_e la probabilité d'apparition d'un premier effondrement :

$$p_e = \mathbb{P}(M = 0) \cdot \mathbb{P}(|X| = 1 \,|\, X \neq 0). \tag{4.32}$$

Lorsque l'insertion se déroule bien, c'est-à-dire qu'un bit est inséré sans produire d'effondrement, deux situations se présentent :

- soit le bit à insérer est un 1,
- soit le bit à insérer est un 0 et le coefficient DCT est différent de -1 et 1, i.e. la valeur absolue du coefficient est strictement supérieure à 1.

La probabilité $p_{\bar{e}}$ d'insertion sans effondrement est donnée par

$$\begin{aligned} p_{\bar{e}} &= \mathbb{P}(M = 1) + \mathbb{P}(M = 0) \cdot \mathbb{P}(|X| > 1 \,|\, X \neq 0) && (4.33) \\ &= 1 - p_e. && (4.34) \end{aligned}$$

τ_1 est la variable aléatoire représentant le nombre de bits insérés successivement avant apparition d'un effondrement. La probabilité $\mathbb{P}(\tau_1 = n)$ d'avoir eu n bits insérés avant qu'il y ait un effondrement est donc :

$$\begin{aligned} \mathbb{P}(\tau_1 = n) &= {p_{\bar{e}}}^n \cdot p_e \\ &= (1 - p_e)^n \cdot p_e. && (4.35) \end{aligned}$$

Afin de calculer la valeur moyenne de τ_1, nous calculons son espérance mathématique. La loi de τ_1 est très proche d'une loi géométrique, son espérance est donc :

$$\begin{aligned}\mathbb{E}(\tau_1) &= \sum_{k=0}^{n} k \cdot \mathbb{P}(\tau_1 = k) \\ &= \sum_{k=0}^{n} k \cdot (1-p_e)^k \cdot p_e \\ &= \frac{1-p_e}{p_e}. \end{aligned} \tag{4.36}$$

Soit $\mathbb{P}(\tau_2 = n)$, la probabilité d'avoir exactement n effondrements successifs.

Lorsqu'une série d'effondrements a commencé, la probabilité $p_{e,go}$ que l'effondrement se poursuive est donnée par

$$p_{e,\text{go}} = \mathbb{P}(|X| = 1 \,|\, X \neq 0), \tag{4.37}$$

et la probabilité $p_{e,\text{stop}}$ que l'effondrement s'arrête est donnée par

$$\begin{aligned} p_{e,\text{stop}} &= \mathbb{P}(|X| > 1 \,|\, X \neq 0) \\ &= 1 - p_{e,\text{go}}. \end{aligned} \tag{4.38}$$

On en déduit que

$$\mathbb{P}(\tau_2 = 0) = p_{\bar{e}} = 1 - p_e, \tag{4.39}$$
$$\mathbb{P}(\tau_2 = 1) = p_e \cdot p_{e,\text{stop}}, \tag{4.40}$$
$$\mathbb{P}(\tau_2 = 2) = p_e \cdot p_{e,\text{go}} \cdot p_{e,\text{stop}}, \tag{4.41}$$
$$\mathbb{P}(\tau_2 = 3) = p_e \cdot p_{e,\text{go}}^2 \cdot p_{e,\text{stop}}. \tag{4.42}$$

donc plus généralement, la probabilité d'avoir exactement n effondrements successifs est :

$$\mathbb{P}(\tau_2 = n) = p_e \cdot p_{e,\text{go}}^{n-1} \cdot p_{e,\text{stop}}, \forall n \geq 1. \tag{4.43}$$

Comme pour τ_1, la loi de τ_2 est très proche d'une loi géométrique, son espérance mathématique est donc facilement calculable et conduit à :

$$\mathbb{E}(\tau_2) = \frac{p_e}{p_{e,\text{stop}}}. \tag{4.44}$$

Pour l'algorithme F4

Dans le cas de l'algorithme F4 (voir schéma 2.9), un effondrement se produit lorsqu'un 0 est inséré dans un 1, ou lorsqu'un 1 est inséré dans un coefficient valant -1. Notons $p_{e,F4}$ la probabilité d'apparition d'un premier effondrement lors d'une insertion :

$$p_{e,F4} = \mathbb{P}(M = 0) \cdot \mathbb{P}(X = 1 \,|\, X \neq 0) + \mathbb{P}(M = 1) \cdot \mathbb{P}(X = -1 \,|\, X \neq 0). \tag{4.45}$$

En revanche, l'insertion se passe sans encombre dans les conditions suivantes :

– le coefficient a une valeur absolue supérieure à 1,
– un 1 est inséré dans un coefficient valant 1,
– ou un 0 est inséré dans un coefficient DCT de valeur −1.

La probabilité d'insertion sans effondrement est donnée par

$$\begin{aligned} p_{\bar{e},F4} &= \mathbb{P}(|X|>1\,|\,X\neq 0) && (4.46)\\ &\quad +\mathbb{P}(M=0)\cdot\mathbb{P}(X=-1\,|\,X\neq 0)+\mathbb{P}(M=1)\cdot\mathbb{P}(X=1\,|\,X\neq 0) && (4.47)\\ &= \mathbb{P}(M=1)\cdot\mathbb{P}(|X|=1\,|\,X\neq 0)+\mathbb{P}(|X|>1\,|\,X\neq 0) && (4.48)\\ &= 1-p_{e,F4}. && (4.49)\end{aligned}$$

Ainsi, la probabilité $\mathbb{P}(\tau_1 = n)$ d'avoir eu n bits insérés successivement avant qu'il y ait un effondrement est :

$$\begin{aligned}\mathbb{P}(\tau_1=n) &= {p_{\bar{e},F4}}^n\cdot p_{e,F4}\\ &= (1-p_{e,F4})^n\cdot p_{e,F4}. && (4.50)\end{aligned}$$

Comme précédemment, (voir equation 4.36), l'espérance de τ_1 pour l'algorithme F4 est

$$\mathbb{E}(\tau_1)=\frac{1-p_{e,F4}}{p_{e,F4}}. \qquad (4.51)$$

Comme cela a été évoqué, dans l'algorithme F4, un effondrement peut se produire lors de l'insertion d'un 0 ou d'un 1. Il est donc nécessaire de distinguer le cas où l'effondrement provient de l'insertion d'un 0 et celui où l'insertion provient d'un 1. Nous définissons les $p_{e,\text{go},0}$ et $p_{e,\text{stop},0}$ (resp. $p_{e,\text{go},1}$ et $p_{e,\text{stop},1}$) correspondant respectivement à la poursuite et à l'arrêt de l'effondrement lors de l'insertion d'un 0 (resp. 1).

$$\begin{aligned} p_{e,\text{go},0} &= \mathbb{P}(X=1\,|\,X\neq 0) && (4.52)\\ p_{e,\text{stop},0} &= \mathbb{P}(|X|>1\,|\,X\neq 0)+\mathbb{P}(X=-1\,|\,X\neq 0)\\ &= 1-p_{e,\text{go},0} && (4.53)\end{aligned}$$

$$\begin{aligned} p_{e,\text{go},1} &= \mathbb{P}(X=-1\,|\,X\neq 0) && (4.54)\\ p_{e,\text{stop},1} &= \mathbb{P}(|X|>1\,|\,X\neq 0)+\mathbb{P}(X=1\,|\,X\neq 0)\\ &= 1-p_{e,\text{go},1} && (4.55)\end{aligned}$$

De même, nous pouvons définir les probabilités d'apparition d'un effondrement $p_{e,0}$ et $p_{e,1}$ de la manière suivante :

$$\begin{aligned} p_{e,0} &= \mathbb{P}(M=0)\cdot\mathbb{P}(X=1\,|\,X\neq 0) && (4.56)\\ p_{e,1} &= \mathbb{P}(M=1)\cdot\mathbb{P}(X=-1\,|\,X\neq 0) && (4.57)\\ p_{e,F4} &= p_{e,0}+p_{e,1}. && (4.58)\end{aligned}$$

Lorsqu'un effondrement se poursuit, le bit que l'on cherche à insérer est le même que celui qui a initié l'effondrement, il faut donc conditionner la probabilité $\mathbb{P}(\tau_2 = n)$ par le bit inséré. Ainsi, la probabilité $\mathbb{P}(\tau_2 = n \,|\, M = m)$ d'avoir exactement n effondrements successifs suite à l'insertion d'un bit valant m, $m \in \{0,1\}$ est obtenue par :

$$\begin{aligned} \mathbb{P}(\tau_2 = 0 \,|\, M = m) &= 1 - p_{e,m}, & (4.59)\\ \mathbb{P}(\tau_2 = n \,|\, M = m) &= p_{e,m} \cdot p_{e,\text{go},m}^{n-1} \cdot p_{e,\text{stop},m}, \forall n \geq 1. & (4.60) \end{aligned}$$

La probabilité totale $\mathbb{P}(\tau_2 = n)$ d'avoir exactement n effondrements successifs est :

$$\mathbb{P}(\tau_2 = n) = \mathbb{P}(M = 0) \cdot \mathbb{P}(\tau_2 = n \,|\, M = 0) + \mathbb{P}(M = 1) \cdot \mathbb{P}(\tau_2 = n \,|\, M = 1). \quad (4.61)$$

L'espérance totale de τ_2 est :

$$\begin{aligned} \mathbb{E}(\tau_2) &= \mathbb{P}(M = 0) \cdot \mathbb{E}(\tau_2 \,|\, M = 0) + \mathbb{P}(M = 1) \cdot \mathbb{E}(\tau_2 \,|\, M = 1) & (4.62)\\ &= \mathbb{P}(M = 0) \cdot \frac{p_{e,0}}{p_{e,\text{stop},0}} + \mathbb{P}(M = 1) \cdot \frac{p_{e,1}}{p_{e,\text{stop},1}}. & (4.63) \end{aligned}$$

Remarque 4.1. *En pratique, les messages sont cryptés avant insertion dans un medium. Un message est donc distribué selon une loi de Bernoulli de paramètre $\frac{1}{2}$, i.e. $M \sim \mathcal{B}(1, \frac{1}{2})$. Ainsi il est raisonnable de considérer que*

$$\mathbb{P}(M = 0) = \mathbb{P}(M = 1) = \tfrac{1}{2}.$$

De plus, il est admis que la distribution des coefficients DCT est symétrique, donc

$$\mathbb{P}(X = 1 \,|\, X \neq 0) = \mathbb{P}(X = -1 \,|\, X \neq 0).$$

Sous ces hypothèses,

$$\begin{aligned} \mathbb{E}(\tau_1)|_{F3} &= \mathbb{E}(\tau_1)|_{F4}, & (4.64)\\ \mathbb{E}(\tau_2)|_{F3} &= \mathbb{E}(\tau_2)|_{F4}. & (4.65) \end{aligned}$$

C'est à dire que les deux algorithmes produisent autant d'effondrements l'un que l'autre lors de l'insertion.

4.2.3.2 Calcul du nombre moyen d'effondrements

Après avoir calculé $\mathbb{E}(\tau_1)$ et $\mathbb{E}(\tau_2)$ qui correspondent à la durée moyenne d'une insertion sans effondrement et au nombre moyen d'effondrements successifs, nous nous proposons de calculer le nombre moyen d'effondrements, c'est-à-dire le nombre de 0 qui sont apparus après insertion.

Identité de Wald Soit $X_1, X_2, \ldots$ une suite de variables aléatoires de somme partielle $T_n = X_1 + \cdots + X_n$. Soit N un temps d'arrêt de la filtration générée par $\{X_n\}$. Le premier lemme de Wald [82] énonce que si $\{X_n\}$ est une suite i.i.d. telle que $\mathbb{E}(X_1) < \infty$ et $\mathbb{E}(N) < \infty$, alors :

$$\mathbb{E}(T_N) = \mathbb{E}(X_1)\mathbb{E}(N) \quad (4.66)$$

Appliquons l'identité de Wald afin de connaître le nombre moyen de séquences qui apparaissent au cours de l'insertion. Soit $(S_n)_{n\in\mathbb{N}^*}$ une suite de séquences et N le temps d'arrêt correspondant. Ainsi :

$$\mathbb{E}(S_1 + \cdots + S_N) = \mathbb{E}(S_1)\cdot\mathbb{E}(N) \tag{4.67}$$

$\mathbb{E}(\mathrm{Nb}_{\mathrm{coef}}) = \mathbb{E}(S_1 + \cdots + S_k)$ correspond au nombre total moyen de coefficients utilisables pour l'insertion.

$$\mathbb{E}(\mathrm{Nb}_{\mathrm{coef}}) = \mathbb{E}(\tau_1 + \tau_2)\cdot\mathbb{E}(N) \tag{4.68}$$
$$\mathbb{E}(\mathrm{Nb}_{\mathrm{coef}}) = [\mathbb{E}(\tau_1) + \mathbb{E}(\tau_2)]\cdot\mathbb{E}(N) \tag{4.69}$$
$$\mathbb{E}(N) = \frac{\mathbb{E}(\mathrm{Nb}_{\mathrm{coef}})}{\mathbb{E}(\tau_1) + \mathbb{E}(\tau_2)} \tag{4.70}$$

En appliquant à nouveau le lemme de Wald à la suite $(\tau_{2,n})_{n\in\mathbb{N}^*}$ et au temps d'arrêt N. Connaissant $\mathbb{E}(N)$ le nombre moyen de séquences qui apparaissent lors de l'insertion, nous obtenons le nombre moyen d'effondrements $\mathbb{E}(\mathrm{Nb}_{\mathrm{eff}})$ apparus dans l'image, ce qui correspond à la quantité de coefficients nuls supplémentaires dans la stégo-image.

$$\mathbb{E}(\tau_{2,1} + \cdots + \tau_{2,N}) = \mathbb{E}(\tau_2)\cdot\mathbb{E}(N) \tag{4.71}$$
$$\mathbb{E}(\mathrm{Nb}_{\mathrm{eff}}) = \mathbb{E}(\mathrm{Nb}_{\mathrm{coef}})\cdot\frac{\mathbb{E}(\tau_2)}{\mathbb{E}(\tau_1) + \mathbb{E}(\tau_2)} \tag{4.72}$$

Remarque 4.2. *Comme on pouvait s'y attendre, le nombre moyen d'effondrements correspond a une proportion des coefficients utilisables. En l'occurrence, il s'agit de ceux qui ont pris la valeur* 0 *après insertion.*

4.2.3.3 Impact des effondrements d'un point de vue statistique

Le phénomène d'effondrement a pour effet d'augmenter le nombre de 0 qui sont insérés dans l'image. Dans le cas de l'algorithme F3, cela peut être vu comme l'insertion d'un message qui ne suivrait plus une loi de Bernoulli de paramètre $\mathbb{P}(M = 0)$.

Nous allons donc déterminer la loi suivie par le message. Considérons que l'insertion se fait dans tous les coefficients disponibles.

S'il n'y avait aucun effondrement, il y aurait $\mathbb{P}(M = 0)\mathbb{E}(\mathrm{Nb}_{\mathrm{coef}})$ bits valant 0 insérés et $\mathbb{P}(M = 1)\mathbb{E}(\mathrm{Nb}_{\mathrm{coef}})$ bits valant 1. Connaissant le nombre d'effondrements apparus, nous pouvons calculer le nombre moyen de 0 (resp. 1) insérés $Nb_{0\ ins}$ (resp. $Nb_{1\ ins}$) :

$$\mathbb{E}(Nb_{0\ ins}) = \mathbb{P}(M = 0)\mathbb{E}(\mathrm{Nb}_{\mathrm{coef}}) + \frac{\mathbb{E}(\mathrm{Nb}_{\mathrm{eff}})}{2}, \tag{4.73}$$
$$\mathbb{E}(Nb_{1\ ins}) = \mathbb{P}(M = 1)\mathbb{E}(\mathrm{Nb}_{\mathrm{coef}}) - \frac{\mathbb{E}(\mathrm{Nb}_{\mathrm{eff}})}{2}. \tag{4.74}$$

Nous introduisons la variable aléatoire discrète M' à valeur dans $[0\,;1]$ qui décrit la valeur d'un bit qui a été inséré. Nous calculons donc la probabilité d'avoir inséré un 0 (resp. 1) s'il y a eu une insertion faite avec F3 :

$$\mathbb{P}(M'=0) = \frac{\mathbb{E}(Nb_{0\ ins})}{\mathbb{E}(\mathrm{Nb_{coef}})} = \mathbb{P}(M=0) + \frac{1}{2}\frac{\mathbb{E}(\mathrm{Nb_{eff}})}{\mathbb{E}(\mathrm{Nb_{coef}})}, \tag{4.75}$$

$$\mathbb{P}(M'=1) = \frac{\mathbb{E}(Nb_{1\ ins})}{\mathbb{E}(\mathrm{Nb_{coef}})} = \mathbb{P}(M=1) - \frac{1}{2}\frac{\mathbb{E}(\mathrm{Nb_{eff}})}{\mathbb{E}(\mathrm{Nb_{coef}})}. \tag{4.76}$$

Pour l'algorithme F4, comme les effondrements peuvent provenir de l'insertion d'un 0 ou d'un 1, on définit l'espérance conditionnelle $\mathbb{E}(\mathrm{Nb_{eff}} \,|\, M=m)$ le nombre moyen d'effondrements provenant de l'insertion d'un bit valant m. Ainsi :

$$\mathbb{P}(M'=0) = \frac{\mathbb{E}(Nb_{0\ ins})}{\mathbb{E}(\mathrm{Nb_{coef}})} = \mathbb{P}(M=0) + \frac{1}{2}\frac{\mathbb{E}(\mathrm{Nb_{eff}} \,|\, M=0)}{\mathbb{E}(\mathrm{Nb_{coef}})} - \frac{1}{2}\frac{\mathbb{E}(\mathrm{Nb_{eff}} \,|\, M=1)}{\mathbb{E}(\mathrm{Nb_{coef}})}, \tag{4.77}$$

$$\mathbb{P}(M'=1) = \frac{\mathbb{E}(Nb_{1\ ins})}{\mathbb{E}(\mathrm{Nb_{coef}})} = \mathbb{P}(M=1) - \frac{1}{2}\frac{\mathbb{E}(\mathrm{Nb_{eff}} \,|\, M=0)}{\mathbb{E}(\mathrm{Nb_{coef}})} + \frac{1}{2}\frac{\mathbb{E}(\mathrm{Nb_{eff}} \,|\, M=1)}{\mathbb{E}(\mathrm{Nb_{coef}})}, \tag{4.78}$$

Dans le cas où nous considérons que le message a été crypté préalablement à l'insertion, pour l'algorithme F4, nous aurions :

$$\mathbb{P}(M'=0) = \mathbb{P}(M=0) = \mathbb{P}(M'=1) = \mathbb{P}(M=1) \tag{4.79}$$

Nous pouvons également calculer la probabilité P_{sh} qu'un coefficient valant 0 provienne d'un effondrement.

$$P_{sh} = \mathbb{P}(Y=0 \,|\, X \neq 0) = \frac{\mathbb{E}(\mathrm{Nb_{eff}})}{\mathbb{E}(\mathrm{Nb_{coef}})} \tag{4.80}$$

Calculons à présent les probabilités d'apparition des différents coefficients.

4.2.4 Calcul des probabilités d'apparition des coefficients DCT

Afin d'observer de quelle manière les coefficients sont modifiés par l'algorithme, nous allons calculer les probabilités $\mathbb{P}(Y=0)$, $\mathbb{P}(Y=2k)$ et $\mathbb{P}(Y=2k+1)$, $k>0$.

Pour l'algorithme F3

Calcul de la probabilité $\mathbb{P}(Y=0)$

Comme il l'a été indiqué auparavant, aucun bit n'est inséré dans un coefficient nul. Cependant, des coefficients peuvent se voir transformés en 0 suite à un effondrement. Ainsi, dans

l'image de sortie, les coefficients nuls supplémentaires proviendront tous d'un effondrement (0 inséré dans un 1 ou un −1).

Le coefficient $Y = 0$ peut donc être obtenu de deux manières :
- le coefficient de l'image vaut 0 ($X = 0$).
- le coefficient utilisé vaut 1 ou −1 et le bit inséré est un 0 ($|X| = 1$ et $M = 0$).

La probabilité d'apparition d'un coefficient nul dans la stégo-image est la probabilité d'apparition d'un coefficient nul dans l'image à laquelle s'ajoute la probabilité qu'une insertion ait conduit à un effondrement :

$$\mathbb{P}(Y = 0) \quad = \quad \mathbb{P}(X = 0) + \mathbb{P}(M = 0) \cdot P_{sh} \tag{4.81}$$

Calcul de la probabilité $\mathbb{P}(Y = 2k)$, $k > 0$

La probabilité d'apparition d'un coefficient pair dans la stégo-image est la probabilité que l'insertion d'un bit ait conduit à ce coefficient pair. F3 décrémente la valeur absolue d'un coefficient DCT si le bit de poids faible du coefficient est différent du bit à insérer.

Le coefficient $Y = 2k$ peut être obtenu de deux manières :
- le coefficient utilisé vaut $2k$ et le bit inséré est un 0 ($X = 2k$ et $M = 0$).
- le coefficient utilisé vaut $2k + 1$ et le bit inséré est un 0 ($X = 2k + 1$ et $M = 0$).

Nous avons vu que la probabilité d'insérer un bit 0 est modifié après insertion, et vaut $\mathbb{P}(M' = 0)$. La probabilité d'apparition du coefficient ayant pour valeur $2k$ est donc :

$$\mathbb{P}(Y = 2k) \quad = \quad \mathbb{P}(M' = 0) \cdot [\mathbb{P}(X = 2k) + \mathbb{P}(X = 2k + 1)] \tag{4.82}$$

Calcul de la probabilité $\mathbb{P}(Y = 2k + 1)$, $k > 0$

La probabilité d'apparition d'un coefficient impair dans la stégo-image est la probabilité que l'insertion d'un bit ait conduit à ce coefficient impair.

Le coefficient $Y = 2k + 1$ peut être obtenu de deux manières :
- le coefficient utilisé vaut $2k + 1$ et le bit inséré est un 1 ($X = 2k + 1$ et $M = 1$),
- le coefficient utilisé vaut $2k + 2$ et le bit inséré est un 1 ($X = 2k + 2$ et $M = 1$).

La probabilité d'apparition du coefficient ayant pour valeur $2k + 1$ est donc :

$$\mathbb{P}(Y = 2k + 1) \quad = \quad \mathbb{P}(M' = 1) \cdot [\mathbb{P}(X = 2k + 1) + \mathbb{P}(X = 2k + 2)]. \tag{4.83}$$

Probabilité d'apparition pour les coefficients négatifs

De manière similaire, pour les coefficients négatifs, nous aurons pour $k < 0$:

$$\mathbb{P}(Y = 2k) = \mathbb{P}(M' = 0) \cdot [\mathbb{P}(X = 2k) + \mathbb{P}(X = 2k-1)], \quad (4.84)$$
$$\mathbb{P}(Y = 2k+1) = \mathbb{P}(M' = 1) \cdot [\mathbb{P}(X = 2k+1) + \mathbb{P}(X = 2k)]. \quad (4.85)$$

Pour l'algorithme F4

Les calculs de probabilité sont similaires à ceux effectués pour l'algorithme F3, en revanche, ils diffèrent dans le cas des coefficients négatifs puisque les LSB sont codés différemment.

Calcul des probabilités d'apparition

Comme pour l'algorithme F3, le coefficient $Y = 0$ peut être obtenu de deux manières :
– le coefficient de l'image vaut 0 ($X = 0$),
– le coefficient utilisé est un 1 ou un -1 et a subi un effondrement.
La probabilité d'apparition d'un coefficient nul dans l'image stéganographiée est :

$$\mathbb{P}(Y = 0) = \mathbb{P}(X = 0) + P_{sh} \quad (4.86)$$

Pour les coefficients positifs, les calculs sont les mêmes que pour F3, $k > 0$:

$$\mathbb{P}(Y = 2k) = \mathbb{P}(M' = 0) \cdot [\mathbb{P}(X = 2k) + \mathbb{P}(X = 2k+1)], \quad (4.87)$$
$$\mathbb{P}(Y = 2k+1) = \mathbb{P}(M' = 1) \cdot [\mathbb{P}(X = 2k+1) + \mathbb{P}(X = 2k+2)]. \quad (4.88)$$

En revanche, les LSB des coefficients négatifs étant codés différemment, le bit de poids faible est modifié si un 1 est inséré dans un coefficient pair, et un 0 inséré dans un coefficient impair. Ainsi, pour $k < 0$, nous obtenons :

$$\mathbb{P}(Y = 2k) = \mathbb{P}(M' = 1) \cdot [\mathbb{P}(X = 2k) + \mathbb{P}(X = 2k-1)], \quad (4.89)$$
$$\mathbb{P}(Y = 2k+1) = \mathbb{P}(M' = 0) \cdot [\mathbb{P}(X = 2k+1) + \mathbb{P}(X = 2k)]. \quad (4.90)$$

Les probabilités d'apparition des différents coefficients permettent la construction d'un détecteur reposant sur la distribution d'un coefficient DCT avant et après insertion avec F3 et F4. Les calculs théoriques présentés dans cette section ont été validés numériquement par des simulations Monte-Carlo présentées dans le chapitre 5.

4.3 Détecteur dans le domaine spatial

Nous souhaitons construire des algorithmes de détection dont nous pouvons prédire et borner analytiquement les probabilités de fausse alarme. Les verrous théoriques d'un tel problème

concernent notamment la quantification des données. En effet, on peut se demander comment construire un test basé sur des données quantifiées et quel est l'impact de la quantification sur les probabilités de fausse alarme et de non détection. De plus, nous introduisons un modèle statistique paramétrique d'un medium de couverture, nous verrons donc quel bénéfice nous pouvons tirer d'un tel modèle.

4.3.1 Modélisation d'un medium de couverture quantifié

Soit un vecteur d'observations $C_n = (c_1, \ldots, c_n)^T$ représentant un medium de couverture, il est défini de la manière suivante :

$$C_n \quad = \quad Q_1[Y_n], \quad Y_n \sim P_\theta \tag{4.91}$$

où $Q_1[y_i] = \lfloor y_i \rfloor$ est l'opération de quantification uniforme (partie entière de y_i) et le vecteur $Y_n = (y_1, \ldots, y_n)^T$ suit la distribution P_θ paramétrée par le vecteur paramétrique θ.

La représentation binaire de c codé sur q bits est :

$$c = Q_1[y] \quad = \quad \sum_{i=0}^{q-1} b_i 2^i, \text{ où } b_i \in \{0,1\}, c \in \{0,1,2,\ldots,2^q-1\}. \tag{4.92}$$

Par la suite, nous considérons le modèle quantifié du medium de couverture (4.91). Le phénomène de saturation n'est pas pris en compte, c'est à dire que la probabilité que l'observation y soit hors des frontières 0 et $2^q - 1$ est négligeable.

4.3.2 Test entre deux hypothèses

Définissons les deux hypothèses alternatives pour une observation quantifiée z :

$$\mathcal{H}_0 : z = c = Q_1[y] \sim Q_{Q_1} = [q_0, \ldots, q_{2^q-1}] \tag{4.93}$$

et

$$\mathcal{H}_1 : z = \begin{cases} Q_2[y] + z_s & \text{avec une probabilité } R \\ c = Q_1[y] & \text{avec une probabilité } 1-R, \end{cases} \tag{4.94}$$

où R est le taux d'insertion, $Q_2[y] = \sum_{i=1}^{q-1} b_i 2^i$ est la quantification uniforme utilisant 2^{q-1} niveaux de reconstruction, $Q_2[y] \sim Q_{Q_2}$, $z_s \sim Q_s = \mathcal{B}(1,p)$ est la distribution de Bernoulli qui définit l'information cachée (en général, $p = 0.5$).

En d'autres termes, la double quantification $Q_2[y]$ de $z = Q_1[y]$ revient à effacer le LSB (i.e. $b_0 = 0$). Ainsi, sous l'hypothèse $\mathcal{H}_1$, le LSB est utilisé comme support de l'information cachée. Nous considérons dans la suite $Q_2[z] = Q_2[y]$.

4.3.3 Taux d'insertion connu : test du rapport de vraisemblance

Supposons que les distributions $Q_s(z_s) = \frac{1}{2}$, $z_s \in \{0,1\}$, Q_{Q_1}, Q_{Q_2} et le taux d'insertion R sont connus. Dans ce cas, le rapport de vraisemblance (RV) pour une observation s'écrit comme suit :

$$\Lambda_R(z) = R\Lambda_1(z) + (1-R), \quad \Lambda_1(z) = \frac{Q_s(b_0)Q_{Q_2}(Q_2[z])}{Q_{Q_1}(z)} = \frac{Q_{Q_2}(Q_2[z])}{2Q_{Q_1}(z)}. \tag{4.95}$$

où b_0 est le LSB de z.

Le test le plus puissant (PP) de Neyman-Pearson dans la classe

$$\mathcal{K}_{\alpha_0} = \{\delta : \mathbb{P}_0(\delta(Z_n) = \mathcal{H}_1) \leq \alpha_0\} \tag{4.96}$$

où $\mathbb{P}_i$ représente la probabilité sous l'hypothèse $\mathcal{H}_i, i = 0,1$, est donné par la règle de décision suivante :

$$\delta_R(Z_n) = \begin{cases} \mathcal{H}_0 & \text{si } \Lambda_R(Z_n) = \prod_{i=1}^{n} \Lambda_R(z_i) < h \\ \mathcal{H}_1 & \text{si } \Lambda_R(Z_n) = \prod_{i=1}^{n} \Lambda_R(z_i) \geq h \end{cases}, \tag{4.97}$$

où le seuil h est la solution de l'équation $\mathbb{P}_0(\Lambda_R(Z_n) \geq h) = \alpha_0$. Le test le plus puissant maximise la puissance

$$\beta_{\delta_R} = 1 - \mathbb{P}_1(\delta_R(Z_n) = \mathcal{H}_0) = 1 - \alpha_1 \tag{4.98}$$

dans la classe $\mathcal{K}_{\alpha_0}$.

Modèle simplifié du medium de couverture

Comme cela a été évoqué dans le chapitre précédent, la valeur y_n du n-ième pixel du média de couverture peut être décomposée, avant quantification, sous la forme :

$$y_n = \theta_n + \xi_n$$

où $\theta_n = \mathbb{E}[y_n]$ est une valeur déterministe représentant l'énergie lumineuse mesurée par le photo-détecteur, et ξ_n est la réalisation d'une variable aléatoire Ξ_n due aux différents phénomènes stochastiques intervenant lors de l'acquisition. Ξ_n peut être modélisé par une variable aléatoire gaussienne de moyenne nulle et de variance σ_n^2. Nous considérons donc par la suite que $\Xi_n \sim \mathcal{N}(0, \sigma_n^2)$.

Rapport de vraisemblance exact et approché

Soit une suite aléatoire indépendante $y_1, \ldots, y_n, y_i \sim \mathcal{N}(\theta, \sigma^2)$. La variable aléatoire quantifiée z_i suit une distribution normale discrète :

$$z_i = Q_1[y_i] \sim Q_{Q_1} = [q_0, \ldots, q_{2^q-1}], \;\; z \in [0,1,2,\ldots,2^q-1], \tag{4.99}$$

où les coefficients q_i sont calculés de la façon suivante :

$$q_i = \int_i^{i+1} \varphi(x)dx = \Phi(i+1) - \Phi(i), \;\; \varphi(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\theta)^2}{2\sigma^2}\right\}, \tag{4.100}$$

où $\Phi(x) = \int_{-\infty}^{x} \varphi(u)du$. On peut voir que pour n'importe quel taux d'insertion R, le rapport de vraisemblance donné par l'équation (4.95) dépend des observations à travers le rapport de vraisemblance $\Lambda_1(z)$ calculé pour $R = 1$. L'équation exacte du log-rapport de vraisemblance est donnée par :

$$\begin{aligned} \log \Lambda_1(Z_n) &= \frac{n}{2} + \sum_{i=1}^{n} \log Q_{Q_2}\left(Q_2[z_i]\right) - \sum_{i=1}^{n} \log Q_{Q_1}\left(z_i\right) \\ &= \sum_{i=1}^{n} \frac{1}{2\sigma^2}\Big[-\left(Q_2[z_i]+1+\delta_{2,i}-\theta\right)^2 + \left(z_i+0.5+\delta_{1,i}-\theta\right)^2\Big]. \end{aligned} \tag{4.101}$$

L'équation approchée du log-RV est :

$$\log \Lambda_1(Z_n) \simeq \log \widetilde{\Lambda}_1(Z_n) = \sum_{i=1}^{n} \frac{1}{2\sigma^2}\left[-\left(Q_2[z_i]+1-\theta\right)^2 + \left(z_i+0.5-\theta\right)^2\right]. \tag{4.102}$$

Les termes correctifs provenant de la quantification $\delta_{1,i}$ and $\delta_{2,i}$ sont omis dans la dernière équation.

Moments du log-rapport de vraisemblance approché

Il fait suite au théorème central limite [75] que la fraction

$$\frac{\log \widetilde{\Lambda}_1(Z_n) - n\mathbb{E}\left(\log \widetilde{\Lambda}_1(z)\right)}{\sigma\sqrt{n}} \underset{n\to\infty}{\rightsquigarrow} \mathcal{N}(0,1), \tag{4.103}$$

où $\sigma^2 = \mathrm{V}ar\left(\log \widetilde{\Lambda}_1(z)\right)$, $\rightsquigarrow$ est la convergence faible et $\log \widetilde{\Lambda}_1(Z_n)$ le log-RV approché donné par (4.102), converge en distribution vers la loi normale lorsque n tend ver l'infini. L'espérance et la variance sont notées respectivement $\mathbb{E}(\cdot)$ et $\mathrm{V}ar(\cdot)$. Afin de calculer les probabilités d'erreur, il est nécessaire d'obtenir l'espérance et la variance lu log-RV approché.

Sous l'hypothèse $\mathcal{H}_0$, le log-RV approché peut être réécrit comme suit :

$$\log \widetilde{\Lambda}_1(Z_n) = \sum_{i=1}^{n}\left[\frac{\zeta_i(b_{0,i}-0.5)}{\sigma^2} - \frac{(b_{0,i}-0.5)^2}{2\sigma^2}\right] = \sum_{i=1}^{n}\left[\frac{\zeta_i(b_{0,i}-0.5)}{\sigma^2} - \frac{1}{8\sigma^2}\right], \tag{4.104}$$

où $\zeta_i = z_i + 0.5 - \theta$, $b_{0,i} = \mathrm{LSB}(z_i)$ et sous l'hypothèse $\mathcal{H}_1$ est

$$\log \widetilde{\Lambda}_1(Z_n) = \sum_{i=1}^{n}\left[\frac{\xi_i(b_{0,i}-0.5)}{\sigma^2} + \frac{1}{8\sigma^2}\right], \tag{4.105}$$

où $\xi_i = Q_2[z_i] + 1 - \theta$ and $b_{0,i} = z_{s,i}$.

Sous l'hypothèse $\mathcal{H}_0$, l'espérance du log-RV approché est donné par l'expression suivante :

$$m_0 = \mathbb{E}_0\left[\log \widetilde{\Lambda}_1(z)\right] = -\frac{1}{8\sigma^2} + \frac{\varepsilon}{\sigma^2}, \tag{4.106}$$

où le coefficient ε définit l'impact de la quantification. Ce coefficient est donné par :

$$\begin{aligned}\varepsilon &= \mathbb{E}_0\left[\zeta(b_0 - 0.5)\right] = \sum_{m=-\infty}^{\infty}\left[\Phi\left(\frac{2m+2-\theta}{\sigma}\right) - \Phi\left(\frac{2m+1-\theta}{\sigma}\right)\right]\frac{(2m+1.5-\theta)}{2} \\ &\quad - \sum_{m=-\infty}^{\infty}\left[\Phi\left(\frac{2m+1-\theta}{\sigma}\right) - \Phi\left(\frac{2m-\theta}{\sigma}\right)\right]\frac{(2m+0.5-\theta)}{2}.\end{aligned} \tag{4.107}$$

La variance est donnée par :

$$\sigma_0^2 = \mathrm{V}ar_0\left[\log \widetilde{\Lambda}_1(z)\right] = \frac{1}{\sigma^4}\left\{\mathbb{E}_0\left[\zeta^2(b_0-0.5)^2\right] - \left[\mathbb{E}_0\left(\zeta(b_0-0.5)\right)\right]^2\right\} = \frac{\mathbb{E}_0\left[\zeta^2\right] - 4\varepsilon^2}{4\sigma^4}, \tag{4.108}$$

où

$$\mathbb{E}_0\left[\zeta^2\right] = \sum_{m=-\infty}^{\infty}\left[\Phi\left(\frac{m+1-\theta}{\sigma}\right) - \Phi\left(\frac{m-\theta}{\sigma}\right)\right](m+0.5-\theta)^2. \tag{4.109}$$

Sous l'hypothèse $\mathcal{H}_1$, l'espérance et la variance du log-RV approché sont données par les expressions suivantes :

$$m_1 = \mathbb{E}_1\left[\log \widetilde{\Lambda}_1(z)\right] = \frac{1}{8\sigma^2}, \tag{4.110}$$

$$\sigma_1^2 = \mathrm{V}ar_1\left[\log \widetilde{\Lambda}_1(z)\right] = \mathrm{V}ar_1\left[\frac{\xi(b_0 - 0.5)}{\sigma^2}\right] = \frac{1}{4\sigma^4}\mathbb{E}_1\left[\xi^2\right], \tag{4.111}$$

où

$$\mathbb{E}_1\left[\xi^2\right] = \sum_{m=-\infty}^{\infty}\left[\Phi\left(\frac{2m+2-\theta}{\sigma}\right) - \Phi\left(\frac{2m-\theta}{\sigma}\right)\right](m+1-\theta)^2. \tag{4.112}$$

Les équations simplifiées suivantes peuvent être proposées pour l'espérance et la variance du log-RV approché donné par (4.102) sans tenir compte de l'impact de la quantification sous l'hypothèse $\mathcal{H}_i$, $i = 0, 1$.

$$m_i = (-1)^{i+1}\frac{1}{8\sigma^2}, \tag{4.113}$$

$$\sigma_i^2 = \frac{1}{4\sigma^2}. \tag{4.114}$$

Proposition 4.1. *Supposons que le taux d'insertion réel prenne une valeur arbitraire* $\widetilde{R} : 0 < \widetilde{R} \leq 1$. *La puissance* β_{δ_1} *du test PP (4.97) avec le log-RV* $\log \widetilde{\Lambda}_1(Z_n)$ *donné par (4.102) peut être approximée par*

$$\beta_{\delta_1} \simeq 1 - \Phi\left(\Phi^{-1}(1-\alpha_0)\frac{\sigma_0}{\sigma_{\widetilde{R}}} - \frac{(m_1 - m_0)\widetilde{R}\sqrt{n}}{\sigma_{\widetilde{R}}}\right) \tag{4.115}$$

pour n grand.

Les espérances m_i et la variance σ_0^2 sont calculées en utilisant les équations (4.106) - (4.112) (resp. (4.113)) en tenant compte (resp. sans tenir compte) de l'impact de la quantification.

La variance $\sigma_{\widetilde{R}}^2$ est également calculée en tenant compte de l'impact de la quantification :

$$\sigma_{\widetilde{R}}^2 = \frac{1}{4\sigma^2}\left[\left(\mathbb{E}_1\left[\xi^2\right] + \frac{1}{16}\right)\widetilde{R} + \left(\mathbb{E}_0\left[\zeta^2\right] + \frac{1}{16} - \varepsilon\right)(1-\widetilde{R})\right] - \left[m_1\widetilde{R} + m_0(1-\widetilde{R})\right]^2 \quad (4.116)$$

ou sans prendre en compte l'impact de la quantification :

$$\sigma_{\widetilde{R}}^2 = \frac{1+\widetilde{R}-\widetilde{R}^2}{4\sigma^2}. \quad (4.117)$$

La démonstration de cette proposition est faite dans [7].

Remarque 4.3. *La forme explicite de la fonction de puissance β_{δ_1} donnée dans la Proposition 4.1 est conforme au fait établi dans [45], qui assure que la capacité stéganographique "sûre" est proportionnelle à la racine carrée du nombre d'observations n.*

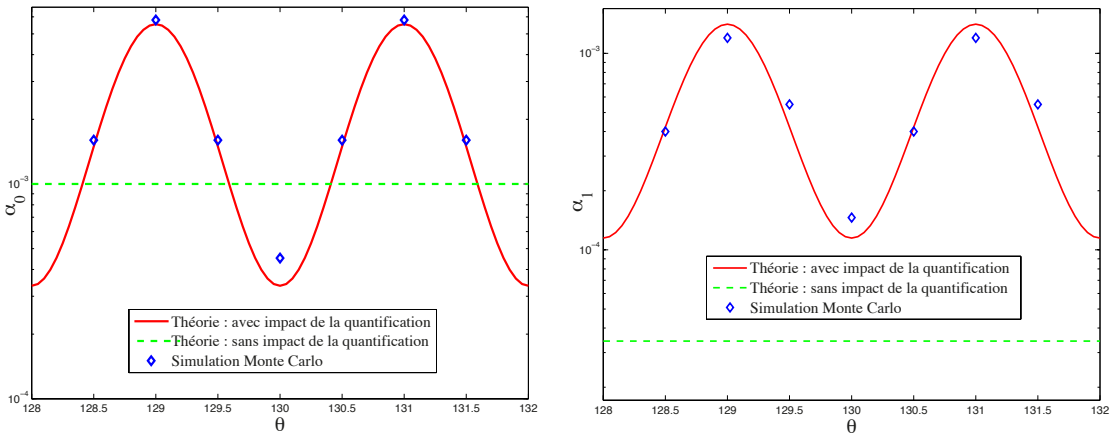


FIGURE 4.5 – *Impact de la quantification sur les probabilités de fausse alarme α_0 (à gauche) et de non détection α_1 (à droite).*

Pour illustrer l'impact de la quantification, supposons les paramètres du modèle gaussien du medium de couverture : $\widetilde{R} = 1, \theta \in [128; 132], \sigma = 1$ et $n = 200$. La comparaison des équations théoriques pour α_0 and α_1 avec une simulation de Monte Carlo (10^6 répétitions) est présentée sur la Figure 4.5.

La figure de gauche présente la probabilité de fausse alarme α_0 calculée avec (trait plein) et sans (pointillés) la prise en compte de l'impact de la quantification. Ici, la probabilité de fausse alarme fixée est $\alpha_0 = 10^{-3}$. D'abord, le seuil h pour le test le plus puissant $\delta_1(Z_n)$ donné par (4.97) est calculé en utilisant les équations (4.113) et (4.115). Ensuite, la probabilité de fausse alarme $\alpha_0 = \alpha_0(h)$ est calculée comme fonction du seuil h en utilisant les équations corrigées pour les espérances et les variances du log-RV, i.e. (4.106) - (4.111) et (4.115) (en tenant compte de l'impact de la quantification).

La figure de droite montre la probabilité de non détection α_1 calculée avec (trait plein) et sans (pointillés) la prise en compte de l'impact de la quantification pour le taux de fausse alarme prescrit $\alpha_0 = 10^{-3}$.

Comme il suit de la Figure 4.5, l'impact de la quantification sur les probabilités de fausse alarme α_0 et non détection α_1 est significatif.

4.3.4 Taux d'insertion inconnu : test d'hypothèses composites

Supposons à présent que les distributions Q_s, Q_{Q_1}, Q_{Q_2} sont connues mais que le taux d'insertion R demeure inconnu. Les hypothèses composites suivantes sont testées en utilisant n observations, Z_n représentant le medium de couverture :

$$\mathcal{H}_0 = \{R \leq r^*\} \text{ contre } \mathcal{H}_1 = \{R > r^*\}. \tag{4.118}$$

où r^* est un taux d'insertion "frontière" séparant les hypothèses $\mathcal{H}_0$ et $\mathcal{H}_1$.

Ainsi, le rapport de vraisemblance (4.95) devient

$$\Lambda_{R_0,R_1}(Z_n) = \prod_{i=1}^{n} \frac{R_1\Lambda_1(z_i) + (1-R_1)}{R_0\Lambda_1(z_i) + (1-R_0)}, \quad \Lambda_1(z_i) = \frac{Q_{Q_2}\left(Q_2[z_i]\right)}{2\,Q_{Q_1}\left(z_i\right)}. \tag{4.119}$$

où $R_0 \leq r^* < R_1$.

La principale difficulté réside dans le fait que les valeurs d'un taux d'insertion acceptable R_0 ou inacceptable R_1 sont inconnues. Dans le cas de deux hypothèses composites, le challenge est d'obtenir un test uniformément le plus puissant (UPP) qui maximise la fonction de puissance

$$\beta(R) = 1 - \mathbb{P}_R(\delta(Z_n) = \mathcal{H}_0) \tag{4.120}$$

pour tout $R > r^*$ dans la classe

$$\mathcal{K}_{\alpha_0} = \left\{\delta : \sup_{R \leq r^*} \mathbb{P}_R(\delta(Z_n) = \mathcal{H}_1) \leq \alpha_0\right\} \tag{4.121}$$

Le test d'hypothèses (4.118) peut être résolu efficacement par un test UPP seulement si le rapport de vraisemblance de la distribution est monotone [4, 55]. Dans notre contexte, cela signifie que pour n'importe quel $R_0 < R_1$ le rapport de vraisemblance donné par (4.119) est une fonction monotone d'une certaine statistique $T = T(Z_n)$. Malheureusement, ce n'est pas le cas pour le rapport de vraisemblance (4.119) et, l'existence d'un test UPP est donc compromise.

4.3.5 Approche asymptotique locale

Une solution permettant de résoudre ce problème consiste à considérer l'approche asymptotique locale proposée par Le Cam [4, 52, 51, 69]. L'idée de cette approche est que la distance entre les hypothèses alternatives dépend de la taille de l'échantillon n dans le sens où les hypothèses convergent lorsque n tend vers l'infini. En utilisant un développement asymptotique du log-RV, un test d'hypothèse particulier peut être ramené localement à un simple test UMP entre deux hypothèses gaussiennes simples différant par leur moyenne [4, 52, 51, 69].

Cette approche est appliquée au modèle :

$$Z_n \sim Q_R = \prod_{i=1}^{n} R\frac{1}{2}Q_{Q_2}\left(Q_2[z_i]\right) + (1-R)Q_{Q_1}\left(z_i\right) \tag{4.122}$$

Considérons les deux suites d'hypothèses convergentes $\mathcal{H}_j(n) = \{R \in \mathbb{R}_j(n)\}$ $(j = 0, 1)$. Les ensembles $\mathbb{R}_j(n)$ sont de la forme $\mathbb{R}_j(n) = r^* + \frac{1}{\sqrt{n}}\mu_r$. La vitesse de convergence est $\frac{1}{\sqrt{n}}$.

Le problème consiste alors à choisir entre les hypothèses suivantes :

$$\mathcal{H}_0(n) = \{R = r^*\} \text{ et } \mathcal{H}_1(n) = \left\{R = r^* + \frac{1}{\sqrt{n}}\mu_r\right\}. \tag{4.123}$$

Le log-rapport de vraisemblance

$$\log\Lambda\left(Z_n; \frac{1}{\sqrt{n}}\mu_r\right) = \log Q_{r^*+\frac{1}{\sqrt{n}}\mu_r}\left(Z_n\right) - \log Q_{r^*}\left(Z_n\right), \tag{4.124}$$

peut être réécrit en utilisant le développement asymptotique :

$$\log\Lambda\left(Z_n; \frac{1}{\sqrt{n}}\mu_r\right) \simeq \frac{1}{\sqrt{n}}\mu_r\zeta_n(Z_n; r^*) - \frac{1}{2}\mu_r^2\mathcal{F}(r^*) \tag{4.125}$$

où $\mathcal{F}(R)$ est l'information de Fisher et

$$\zeta_n(Z_n; r^*) = \sum_{i=1}^{n} \left.\frac{\partial \log Q_R(z_i)}{\partial R}\right|_{R=r^*} \tag{4.126}$$

est la fonction de score efficace qui est asymptotiquement gaussienne

$$\zeta_n(Z_n; r^*) \rightsquigarrow \begin{cases} \mathcal{N}(0, \mathcal{F}(r^*)) & \text{sous } \mathcal{H}_0 \quad \text{i.e. } z_i \sim Q_{r^*} \\ \mathcal{N}(\mathcal{F}(r^*)\mu_r, \mathcal{F}(r^*)) & \text{sous } \mathcal{H}_1 \quad \text{i.e. } z_i \sim Q_{r^*+\frac{\mu_r}{\sqrt{n}}} \end{cases} \tag{4.127}$$

Il peut être montré que le score efficace est donné par

$$\zeta_n(Z_n; r^*) = \sum_{i=1}^{n} \zeta_n(z_i; r^*) = \sum_{i=1}^{n} \frac{\Lambda_1(z_i) - 1}{r^*\Lambda_1(z_i) + (1 - r^*)} \tag{4.128}$$

et l'information de Fischer $\mathcal{F}(R)$ est

$$\mathcal{F}(R) = \mathbb{E}_R\left[\frac{\Lambda_1(z) - 1}{R\Lambda_1(z) + (1-R)}\right]^2. \tag{4.129}$$

Ainsi, le test UPP local entre deux hypothèses alternatives (4.118) est donné par la règle de décision :

$$\delta_{r^*}(Z_n) = \begin{cases} \mathcal{H}_0 & \text{if} \quad \zeta_n(Z_n; r^*) < h \\ \mathcal{H}_1 & \text{if} \quad \zeta_n(Z_n; r^*) \geq h \end{cases}. \tag{4.130}$$

4.3.6 Modélisation du medium de couverture plus réaliste

D'après l'équation (4.115), la puissance β d'un détecteur optimal dépend de l'écart-type σ du medium de couverture pour un taux de fausse alarme donné α_0. Ainsi, pour augmenter la puissance β, il faut réduire l'écart-type σ, ce qui peut être fait en utilisant un modèle paramétrique du medium de couverture. Un vecteur d'observation extrait du medium de couverture est donc caractérisé par un modèle régressif par segment. Soit C le vecteur d'observation découpé en M sous-vecteurs statistiquement indépendants C_j de longueur n, i.e. $C^T = (C_1^T, \ldots, C_M^T)$. On suppose que chaque segment C_j est approximé par :

$$C_j = Q_1[Y_j], \quad Y_j = Hx_j + \xi \sim \mathcal{N}(Hx_j, \sigma_j^2 I_n), \quad j = 1, \ldots, M, \tag{4.131}$$

où H est une matrice $[n \times l]$ connue de rang plein, $n > l$, $x_j \in \mathbb{R}^l$ est un paramètre de nuisance (représentant le contenu de l'image), I_n est une matrice identité $(n \times n)$ et σ_j^2 est la variance résiduelle. Les l colonnes de H engendrent un sous-espace $R(H)$ de l'espace des observations $Y_j \in \mathbb{R}^n$. Il est supposé qu'une des colonnes de H est nécessairement formée de 1. Un tel modèle paramétrique est une méthode efficace pour réduire l'écart-type σ [9].

Le nouveau test d'hypothèses avec un modèle paramétrique du medium de couverture consiste à choisir entre

$$\mathcal{H}_0 \ : \ Z = C = Q_1[Y], \tag{4.132}$$

et

$$\mathcal{H}_1 : z_i = \begin{cases} Q_2[y_i] + z_{s,i} & \text{avec probabilité} \quad R \\ c_i = Q_1[y_i] & \text{avec probabilité} \quad 1-R \end{cases}, \ i = 1, \ldots, Mn, \tag{4.133}$$

où $Y^T = (Y_1^T, \ldots, Y_M^T)$, $Y_j \sim \mathcal{N}(Hx_j, \sigma_j^2 I_n)$.

Le log-LR $\log \Lambda_1(Z_j)$ dans le cas du modèle paramétrique, peut être réécrit comme suit :

$$\log \Lambda_1(Z_j) = -\frac{1}{2\sigma_j^2}\left\|Q_2[Z_j] - Hx_j + \mathbf{1}_n + \Delta_2\right\|_2^2 + \frac{1}{2\sigma_j^2}\left\|Z_j - Hx_j + 0.5 \cdot \mathbf{1}_n + \Delta_1\right\|_2^2, \tag{4.134}$$

où $\mathbf{1}_n$ est un vecteur de dimension n composé de 1, Δ_j est un vecteur de dimension n composé des termes correctifs dus à la quantification $\delta_{j,i}$, $j = 1, 2$.

Le log-RV approché est donné par

$$\log \Lambda_1(Z_j) \simeq -\frac{1}{2\sigma_j^2}\left\|Q_2[Z_j] - Hx_j + \mathbf{1}_n\right\|_2^2 + \frac{1}{2\sigma_j^2}\left\|Z_j - Hx_j + 0.5 \cdot \mathbf{1}_n\right\|_2^2. \tag{4.135}$$

En pratique, x_j et σ_j^2 sont inconnus. Les aspects théoriques qui traitent des paramètres de nuisance dans le cadre de la théorie de la décision statistique sont discutés dans [4, 55]. Une approche efficace à ce problème est basée sur la théorie de l'invariance en statistiques. Les tests invariants optimaux et leurs propriétés dans le cadre du traitement d'image ont été proposés et étudiés dans [20, 21, 22, 72].

Supposons d'abord que σ_j^2 est connu. Le paramètre de nuisance x_j peut être estimé (ou plus précisément rejeté) en utilisant $Q_2[Z_j] = Q_2[Y_j]$ qui ne contient pas d'information cachée. Pour

rejeter les paramètres de nuisance, la théorie de l'invariance est souvent utilisée dans le cas de données non quantifiées. La description détaillée des aspects théoriques et pratiques ainsi que l'utilisation du principe d'invariance dans le cas d'un modèle régressif peuvent être trouvées dans [20, 21, 22, 72]. L'idée de l'approche par les tests d'hypothèses invariants est basée sur l'existence d'une invariance naturelle du problème de détection par rapport à un certain groupe de transformations.

Remarquons que le problème de décision donné par (4.132) - (4.133) reste *presque* invariant pour le groupe des translations $G = \{g : g(Y) = Y + Hx\}$, $x \in \mathbb{R}^l$. Le terme *presque* est dû à la quantification $Q_j[y]$, $j = 1, 2$. Sans quantification, l'invariance serait exacte. Dans un tel cas, la décision statistique devrait être basée sur un invariant maximal du groupe de translations G, i.e. tous les tests invariants par rapport à G sont des fonctions d'un invariant statistique maximal (voir la définition dans [18]). Il est montré que la projection $\varepsilon = W^T Y$ de Y sur le noyau $R(H)^\perp$ de la matrice H est un invariant maximal. La matrice $W = (w_1, \ldots, w_{n-l})$ de taille $n \times (n-l)$ est composée des vecteurs propres $w_1, \ldots, w_{n-l}$ de la matrice de projection $P_H^\perp = I_n - H(H^T H)^{-1} H^T$ correspondant à la valeur propre 1. La matrice W satisfait les conditions suivantes : $W^T H = 0$, $WW^T = P_H^\perp$ et $W^T W = I_{n-l}$. En pratique, la réjection du paramètre de nuisance est fait en utilisant la matrice $P_H^\perp$, car $P_H^\perp H = 0$. Cependant, si la matrice H est de rang plein, alors le test invariant est équivalent au test du rapport de vraisemblance généralisé (ou RVG).

Le log-RVG approché ou *presque* invariant est donné par

$$\begin{aligned} \log \widehat{\Lambda}_1(Z_j) &\simeq -\frac{1}{2\sigma_j^2}\left\|Q_2[Z_j] - H\widehat{x} + \mathbf{1}_n\right\|_2^2 + \frac{1}{2\sigma_j^2}\left\|Z_j - H\widehat{x} + 0.5 \cdot \mathbf{1}_n\right\|_2^2 \\ &= \frac{1}{\sigma_j^2}\left[P_H^\perp Q_2[Z_j]\right]^T \left[B_0 - 0.5 \cdot \mathbf{1}_n\right] + \frac{n}{8\sigma_j^2}, \end{aligned} \tag{4.136}$$

où $B_0 = (b_{0,1}, \ldots, b_{0,n})^T$ et $\widehat{x} = (H^T H)^{-1} H^T Q_2[Z_j]$ est l'estimation du paramètre de nuisance x par la méthode du maximum de vraisemblance.

Sous l'hypothèse $\mathcal{H}_0$, l'espérance et la variance du log-RVG approché pour le vecteur d'observation Y sont donnés par les expressions suivantes :

$$m_0 = \mathbb{E}_0\left[\sum_{j=1}^{M} \log \widehat{\Lambda}_1(Z_j)\right] \simeq \frac{M(2l-n)}{8\overline{\sigma}^2} \quad \text{avec} \quad \frac{1}{\overline{\sigma}^2} = \frac{1}{M}\sum_{j=1}^{M} \frac{1}{\sigma_j^2} \tag{4.137}$$

et

$$\sigma_0^2 = \mathrm{Var}_0\left[\sum_{j=1}^{M} \log \widehat{\Lambda}_1(Z_j)\right] \simeq M(n-l)\left[\frac{1}{4\overline{\sigma}^2} + \frac{1}{16\overline{\sigma}^4}\right] \text{ avec } \frac{1}{\overline{\sigma}^4} = \frac{1}{M}\sum_{j=1}^{M} \frac{1}{\sigma_j^4}. \tag{4.138}$$

Supposons que le taux d'insertion réel prend une valeur arbitraire $\widetilde{R}$: $0 < \widetilde{R} \leq 1$.

Sous l'hypothèse $\mathcal{H}_1$ avec le taux d'insertion réel $\widetilde{R}$, l'espérance et la variance du log-RVG approché pour le vecteur d'observation Y sont données par les expressions suivantes :

$$m_{\widetilde{R}} = \mathbb{E}_{\widetilde{R}}\left[\sum_{j=1}^{M} \log \widehat{\Lambda}_1(Z_j)\right] \simeq \frac{M(2l - n + 2\widetilde{R}(n-l))}{8\overline{\sigma}^2} \tag{4.139}$$

et

$$\begin{aligned}\sigma^2_{\widetilde{R}} &= Var_{\widetilde{R}}\left[\sum_{j=1}^{M}\log\widehat{\Lambda}_1(Z_j)\right] \simeq \frac{M(n-l)}{4\overline{\sigma}^2} \\ &+ \frac{M(n^2\widetilde{R}+4(1-\widetilde{R})(n-l)+(2l-n)^2(1-\widetilde{R})-(2l-n+2\widetilde{R}(n-l))^2)}{64\overline{\sigma}^4}\end{aligned} \tag{4.140}$$

Proposition 4.2. *Supposons que la condition de Lindeberg imposée au log-RV* $\log\widehat{\Lambda}_1(Z_j)$ *est satisfaite. Il suit du théorème central limite que la fraction suivante*

$$\frac{\sum_{j=1}^{M}\log\widehat{\Lambda}_1(Z_j)-\mathbb{E}_{\widetilde{R}}\left[\sum_{j=1}^{M}\log\widehat{\Lambda}_1(Z_j)\right]}{\sqrt{Var_{\widetilde{R}}\left[\sum_{j=1}^{M}\log\widehat{\Lambda}_1(Z_j)\right]}} \underset{M\to\infty}{\rightsquigarrow} \mathcal{N}(0,1) \tag{4.141}$$

converge faiblement vers la loi normale [75]. La puissance β_{δ_1} *du test (4.97) avec le log-RV* $\sum_{j=1}^{M}\log\widehat{\Lambda}_1(Z_j)$ *donné par (4.136) peut être approchée par*

$$\beta_{\delta_1} \simeq 1-\Phi\left(\Phi^{-1}(1-\alpha_0)\frac{\sigma_0}{\sigma_{\widetilde{R}}}-\frac{(m_{\widetilde{R}}-m_0)\sqrt{n}}{\sigma_{\widetilde{R}}}\right) \tag{4.142}$$

pou M grand.

Dans [7] figurent les éléments nécessaires à la démonstration de cette proposition.

Si la variance résiduelle σ_j^2 est inconnue, alors le RVG suivant est utilisé

$$\log\widehat{\Lambda}_1(Z_j) \simeq \frac{1}{\widehat{\sigma}_j^2}\left[P_H^{\perp}Q_2[Z_j]\right]^T\left[B_0-0.5\cdot\mathbf{1}_n\right]+\frac{n}{8\widehat{\sigma}_j^2}, \tag{4.143}$$

où $\widehat{\sigma}_j^2 = \frac{1}{n-l}\left\|P_H^{\perp}Q_2[Z_j]\right\|_2^2$.

Le premier terme du membre de droite de l'équation (4.136) définit la sensibilité du test car le second terme $\frac{n}{8\widehat{\sigma}^2}$ ne dépend pas du message secret inséré. Néanmoins, le second terme du membre de droite $\frac{n}{8\widehat{\sigma}^2}$ de (4.136) est nécessaire afin de calculer correctement le seuil h dans (4.97) en utilisant l'équation $\mathbb{P}_0(\sum_{j=1}^{M}\log\widehat{\Lambda}_1(Z_j)\geq h)=\alpha_0$. Le premier terme du membre de droite de l'équation (4.136) représente un produit scalaire du vecteur des résidus $\varepsilon = P_H^{\perp}Q_2[Z_j]$, i.e. le vecteur de projection de $Q_2[Z_j]$ sur le complément orthogonal $R(H)^{\perp}$ du sous-espace $R(H)$, et le vecteur $[B_0-0.5\cdot\mathbf{1}_n]$ composé de $\mathrm{LSB}(z_i)-0.5$:

$$\frac{1}{\widehat{\sigma}^2}\left[P_H^{\perp}Q_2[Z_j]\right]^T[B_0-0.5\cdot\mathbf{1}_n]=\sum_{i=1}^{n}\overbrace{\widehat{\sigma}^{-2}}^{=\text{"poids"}}\cdot\overbrace{(Q_2[z_i]-(H\widehat{x}_j)_i+1)}^{=\text{"résidus" }\varepsilon_i}\overbrace{(b_{0,i}-0.5)}^{=\mathrm{LSB}(z_i)-0.5} \tag{4.144}$$

où $(H\widehat{x}_j)_i$ est la i-ème composante du vecteur $H\widehat{x}_j$.

Comparons à présent la dernière équation avec des détecteurs existants [27, 43, 44]. Ces détecteurs sont basés sur la statistique suivante [44] :

$$\sum_{i=1}^{n} \overbrace{w_i}^{=\text{"poids"}} \cdot \overbrace{(z_i - \mathcal{F}(z)_i)}^{=\text{"résidus" } \varepsilon_i} \cdot \overbrace{(z_i - \bar{z}_i)}^{=2\cdot(\mathrm{LSB}(z_i)-0.5)} , \tag{4.145}$$

où $\mathcal{F}(s)$ représente un filtre destiné à estimer l'image de couverture en filtrant l'image stéganographiée, le poids choisi est $w_i = \dfrac{1}{1+\sigma_i^2}$, σ_i^2 est la variance *locale* et $\bar{z}_i$ représente l'entier positif z_i avec le LSB inversé.

Il suit des équations (4.144) - (4.145) que les détecteurs développés dans [27, 43, 44] coïncident avec le premier terme du log-RVG (4.136).

Conclusion

Dans ce chapitre, le problème de détection d'informations cachées a été traité dans le domaine fréquentiel, à travers des tests basés sur la modélisation des coefficients DCT, mais également dans le domaine spatial.

Dans le cadre d'une insertion faite avec Jsteg, un test basé sur la modélisation laplacienne des coefficients DCT a été proposé et les calculs des probabilités du détecteur ont été établis sous des conditions asymptotiques (on considère une image grande). Une généralisation à une distribution quelconque a été proposée dans le cas où tout est connu.

Ensuite, certains algorithmes d'insertion (F3, F4 et F5) introduisent des effondrements, leur impact sur les probabilités d'apparition des coefficients DCT a été étudié d'un point de vue statistique. Cela permet la construction d'un détecteur basé sur le nombre d'effondrements apparus lors de l'insertion. En effet, les probabilités calculées permettent d'obtenir la distribution des coefficients avant et après insertion, un test de Neyman-Pearson est donc réalisable.

Enfin, la détection d'insertion par substitution des LSB dans le cadre du domaine spatial, a fait l'objet d'une étude statistique rigoureuse notamment en présentant l'impact de la quantification sur les performances du test le plus puissant lorsque tout est connu (les paramètres de distribution des pixels et le taux d'insertion). En négligeant l'impact de la quantification, le test uniformément le plus puissant a été construit en utilisant une approche asymptotique locale. Enfin un modèle paramétrique du medium de couverture a permis de montrer théoriquement que notre test coïncide avec les détecteurs WS existants.

Chapitre 5

Expérimentations numériques

Sommaire

Introduction

Ce chapitre, consacré aux expérimentations numériques, a pour but de valider la pertinence de la méthodologie proposée dans cette thèse, pour répondre aux contraintes imposées par le projet RIC (maîtrise du taux de fausse alarme, taux de détection le meilleur pour une fausse alarme fixée). Ainsi, les détecteurs conçus et présentés dans le chapitre 4 font l'objet de résultats numériques issus de données réelles et de simulations.

Tout d'abord, les bases d'images utilisées pour les expérimentations sont exposées dans la section 5.1. La section 5.2 présente le schéma fonctionnel du détecteur conçu dans le cadre de la détection dans les images JPEG pour l'algorithme d'insertion Jsteg. La pertinence du modèle laplacien est discutée et les performances du détecteur en fonction de différents paramètres sont présentées. Pour l'algorithme F3, la section 5.3 expose les résultats théoriques obtenus dans le chapitre précédent, qui sont validés en les confrontant aux résultats obtenus pour des simulations. Pour finir, la section 5.4 présente le schéma fonctionnel et compare les détecteurs 1D et 2D construits dans le cadre d'images non compressées.

5.1 Introduction aux expérimentations

5.1.1 Bases d'images utilisées

Pour tester nos détecteurs, présentés dans le chapitre 4, de nombreuses bases d'images s'offraient à nous. Notre choix s'est porté sur deux bases d'images de référence : la base d'image USC-SIPI de l'université de Californie du Sud, dont la première version date de 1977, et la base d'image BOSS, datant de 2010. Ces bases ont été créées dans le but de soutenir la recherche en permettant aux chercheurs de comparer leurs algorithmes sur les mêmes images, dans des domaines tels que le traitement d'image ou la stéganalyse. La base USC-SIPI a été utilisée à de nombreuses reprises dans la littérature, notamment pour modéliser les coefficients DCT, d'autre part, la base BOSS contient un nombre d'images important, ce qui est idéal pour tester nos détecteurs et obtenir des courbes COR représentatives. De plus, ces images sont très variées : elles représentent des paysages, des portraits, certaines sont très texturées, d'autres plus uniformes. Ces bases sont donc représentatives des images pouvant être testées par nos algorithmes.

Base d'images USC-SIPI

La base de données USC-SIPI [1] est une base d'images fréquemment utilisée dans les domaines de recherche relatifs aux images. Il s'agit d'images au format TIFF, de différentes tailles (256×256, 512×512 et 1024×1024), en noir et blanc ou en couleur, et dont la plupart sont des images scannées. Les images sont réparties en différentes catégories : des images texturées, des vues aériennes, des portraits ou encore des images provenant de vidéos.

Base d'images BOSS

La base d'images BOSS a été créée à l'occasion du challenge BOSS (Break Our Steganographic System) [2]. Elle contient 10 000 images provenant de 8 appareils photographiques. Ces images couleurs initialement au format RAW ont été redimensionnées et recadrées au format 512×512 en noir et blanc.

1. University of Southern California - Signal and Image Processing Institute
http ://sipi.usc.edu/database/

5.1.2 Utilisation des images pour les expérimentations

Les détecteurs ont été comparés sur des images simulées ainsi que sur des images réelles (base d'image BOSS). Les images réelles au format PGM ont été converties par `imagemagick` au format JPEG avec les facteurs de qualité 50 et 70. Pour les images simulées, les paramètres estimés $\hat{b}_k$ des images réelles ont été utilisés pour générer les vecteurs V_k à l'aide un générateur pseudo-aléatoire.

Dans les expérimentations, nous considérons seulement des images en dégradés de gris, ou, lorsqu'il s'agit d'images couleurs, nous utilisons le canal de luminance.

5.2 Détection de Jsteg dans les images JPEG

Cette section présente les résultats numériques obtenus par le détecteur basé sur la modélisation laplacienne des coefficients DCT. Le schéma fonctionnel de l'algorithme est exposé tout d'abord, puis la pertinence du modèle laplacien ainsi que l'estimation du paramètre de la distribution sont présentés. Ensuite, les performances du détecteur sont analysées, afin de mettre en évidence l'impact de chaque paramètre sur le détecteur. Enfin, nous comparons notre détecteur avec des détecteurs existants.

5.2.1 Schéma fonctionnel de l'algorithme

En pratique, lorsqu'une image est analysée, nous souhaitons savoir si un message caché a été inséré dans le medium avec un taux d'insertion donné. Le détecteur proposé dans la section 4.1 repose sur différentes étapes. Ces étapes sont représentées sur la figure 5.1 à travers le schéma fonctionnel de l'algorithme de détection.

La première étape consiste à extraire de l'image, la structure $V = [V_1 \cdots V_{64}]$ qui contient les 64 vecteurs V_k de coefficients DCT par fréquence, ainsi que la matrice de quantification Δ.

Le vecteur V_1, qui représente la composante DC, n'est pas considéré par la suite puisqu'il n'est pas utilisé lors de l'insertion. Ainsi, pour chaque fréquence des coefficients AC (i.e. $V_2, \cdots, V_64$), le paramètre b_k de la distribution laplacienne est estimé. Cela permet de calculer le rapport de vraisemblance (voir équation (4.8)). De plus, la probabilité qu'un coefficient soit utilisable, la probabilité que la variable $\zeta_{k,i}$ soit égale à 1 et la moyenne et la variance du rapport de vraisemblance sont calculées de manière théorique.

Enfin, l'algorithme renvoie la décision prise en comparant le rapport de vraisemblance au seuil calculé théoriquement pour une probabilité de fausse alarme α_0 fixée.

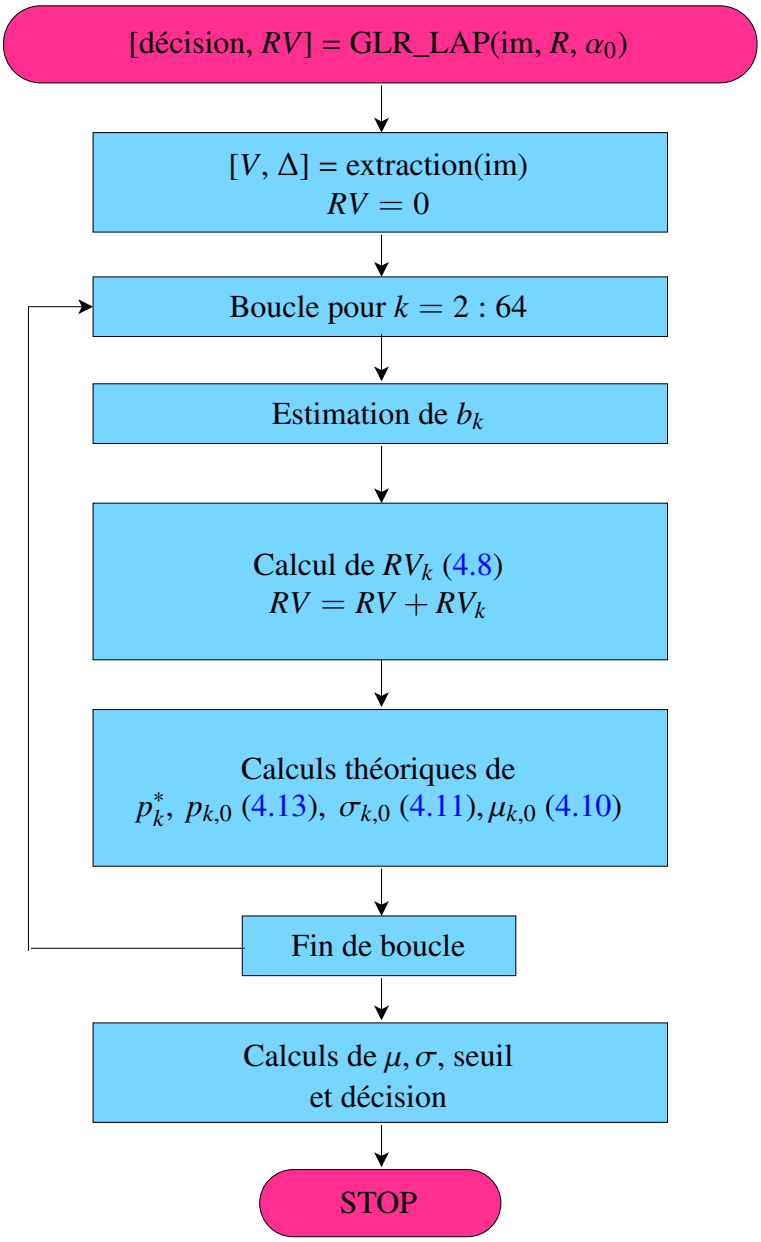


FIGURE 5.1 – *Schéma fonctionnel de l'algorithme.*

5.2.2 Pertinence du modèle laplacien

Dans cette thèse, il a été choisi d'utiliser une modélisation laplacienne des coefficients DCT quantifiés afin de mettre en place un détecteur paramétrique présenté dans la section 4.1. Nous avons pu voir, dans la littérature, différents modèles suggérés (voir section 3.4) pour modéliser ces coefficients, notre choix s'est porté sur le modèle laplacien qui reste simple mais pertinent.

Pour un modèle plus complexe, la méthodologie utilisée pour la création du détecteur serait la même que celle qui a été présentée (voir section 4.1.2).

Base USC-SIPI

Rappelons d'abord que dans un bloc DCT 8×8 contenant 64 coefficients, l'ordre de lecture considéré est l'ordre zigzag (Fig 5.2) qui consiste à balayer les coefficients depuis les basses fréquences jusqu'aux hautes fréquences.

1	3	4	10	11	21	22	36
2	5	9	12	20	23	35	37
6	8	13	19	24	34	38	49
7	14	18	25	33	39	48	50
15	17	26	32	40	47	51	58
16	27	31	41	46	52	57	59
28	30	42	45	53	56	60	63
29	43	44	54	55	61	62	64

FIGURE 5.2 – *Ordre zigzag*

Le premier coefficient ($k = 1$), appelé coefficient DC, n'est pas utilisé pour l'insertion d'informations cachées ou pour la détection. La loi suivie par ce coefficient n'est pas laplacienne mais serait plutôt gaussienne. Pour la détection, nous nous intéressons à la distribution des coefficients AC ($k = 2, \ldots, 64$).

L'histogramme des 64 coefficients DCT est présenté sur les figures 5.3 (compression 70) et 5.4 (sans compression) pour l'image Mandrill. Nous pouvons voir que le paramètre d'échelle de la distribution laplacienne des coefficients AC varie selon la fréquence, mais la forme générale demeure.

La figure 5.5 présente la distribution du deuxième coefficient DCT de cinq images provenant de la base d'images USC-SIPI. Ces images de taille 512×512 ont été compressées au format JPEG avec un facteur de qualité 70. On peut remarquer que l'estimation du paramètre b de la distribution laplacienne varie assez peu : elle prend des valeurs allant de 5.83 à 8.55 en fonction des images.

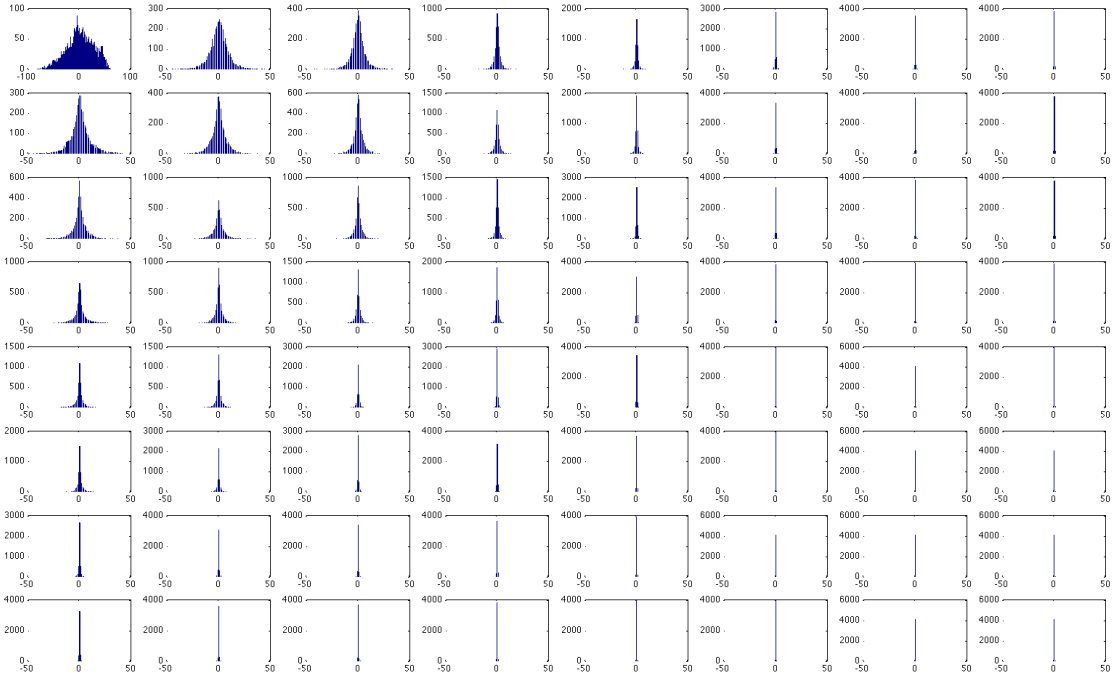


FIGURE 5.3 – *Histogramme des 64 coefficients DCT de l'image Mandrill compressée avec un facteur de qualité 70.*

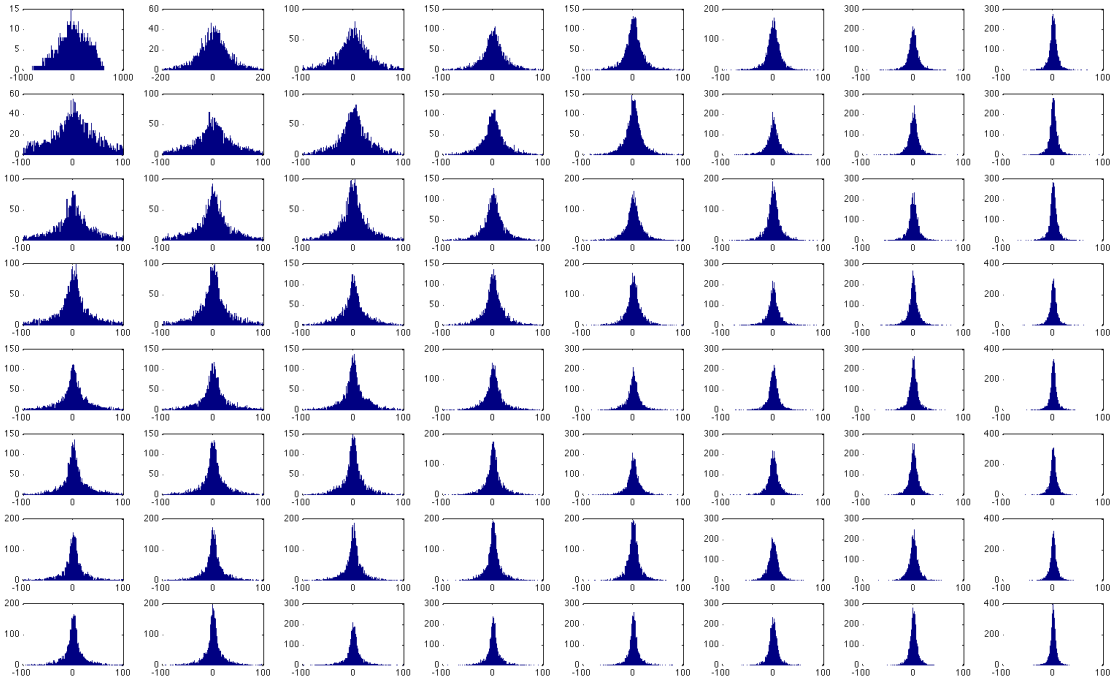


FIGURE 5.4 – *Histogramme des 64 coefficients DCT de l'image Mandrill au format JPEG non compressée (convertie avec un facteur de qualité 100).*

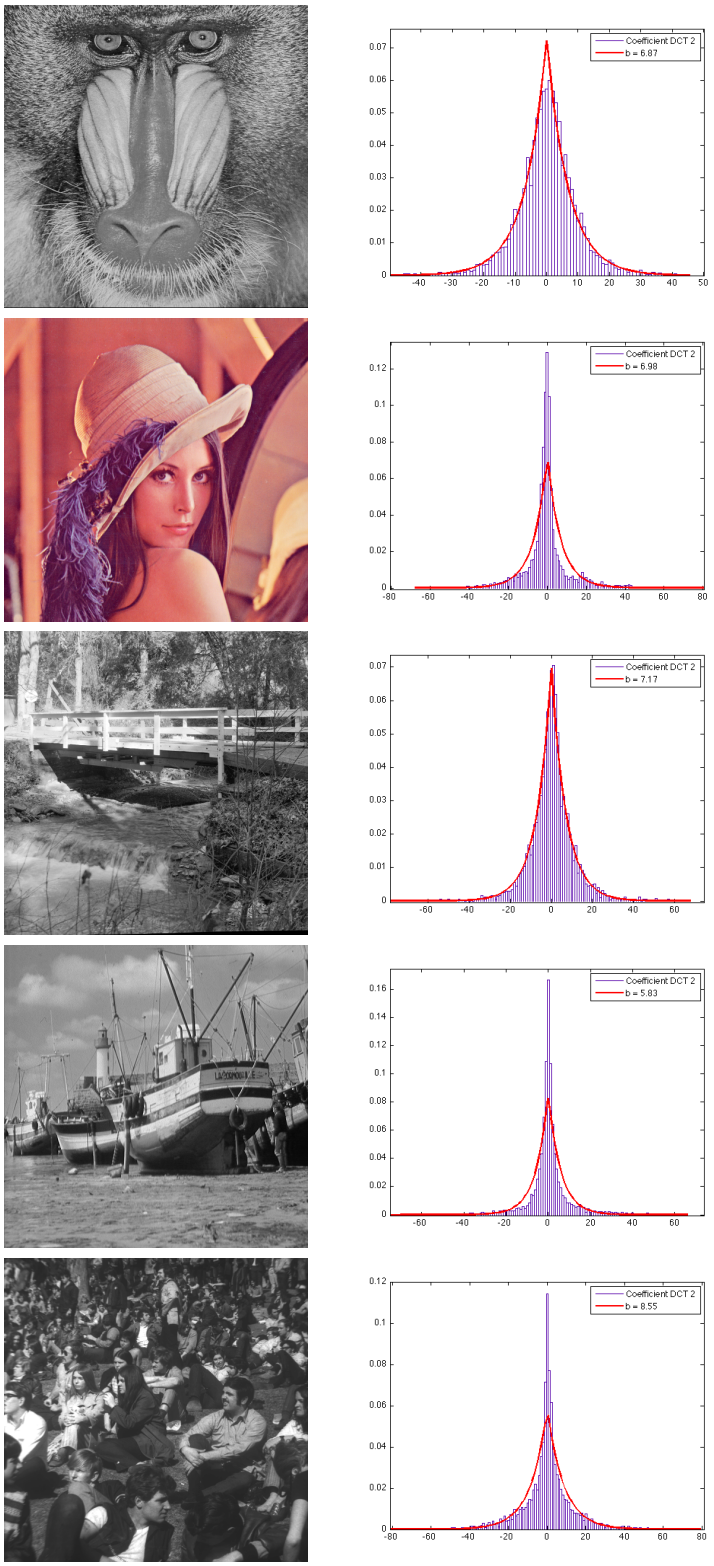
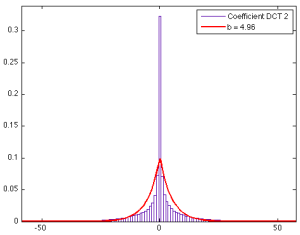
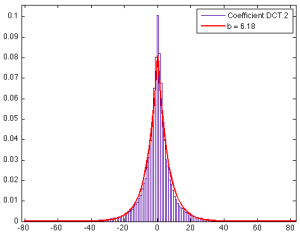


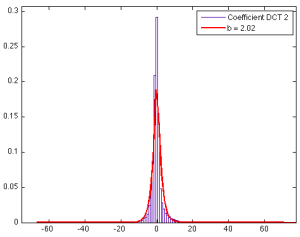
FIGURE 5.5 – *Images de la base USC-SIPI avec la distribution empirique du 2^e coefficient DCT et l'estimation du paramètre d'échelle de la distribution laplacienne.*

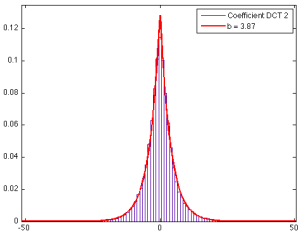
UTT

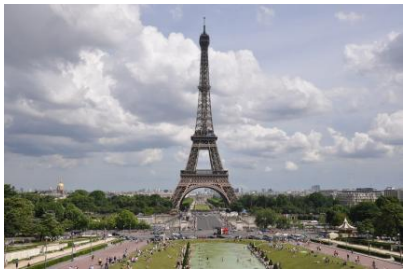
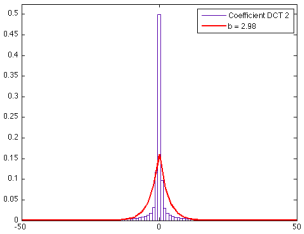
Feuille

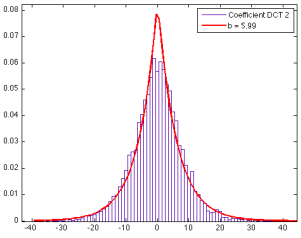
Bouteilles



Right

Tour Eiffel

Pomme

FIGURE 5.6 – *Images personnelles avec la distribution empirique du 2^e coefficient DCT et l'estimation du paramètre d'échelle de la distribution laplacienne.*

Images personnelles

La figure 5.6 présente la distribution du deuxième coefficient DCT d'images JPEG provenant d'un Nikon D90 de dimension 4288×2848 ainsi qu'une image de taille 800×600 compressée par un téléphone. On peut remarquer que l'estimation du paramètre b de la distribution laplacienne varie : elle prend des valeurs allant de 2.02 à 6.18 en fonction des images.

Les images utilisées dans cette section couvrent un éventail représentatif des différentes scènes qui peuvent être rencontrées. En effet, Mandrill, Feuille et Pomme sont des images très texturées, alors que Boat, UTT et Tour Eiffel comportent des zones uniformes (ciel). De plus ces images ont des tailles différentes et ne proviennent pas d'un même appareil photographique.

Nous avons donc pu voir sur les figures 5.5 et 5.6 que la distribution laplacienne permet de modéliser raisonnablement la distribution des coefficients DCT quantifiés.

5.2.3 Estimation des paramètres

Soit $X = \{x_1, \ldots, x_n\}$ le vecteur contenant les coefficients DCT d'une fréquence. Une des premières étapes est d'estimer le paramètre b de la distribution laplacienne modélisant la distribution du vecteur X.

Pour une distribution laplacienne continue de moyenne nulle, la vraisemblance s'écrit :

$$L(x_1, \ldots, x_n; b) = \prod_{i=1}^{n} \frac{1}{2b} \exp\left(-\frac{|x_i|}{b}\right) = \frac{1}{(2b)^n} \prod_{i=1}^{n} \exp\left(-\frac{|x_i|}{b}\right). \tag{5.1}$$

Comme la vraisemblance est positive, on considère son logarithme :

$$\log L(x_1, \ldots, x_n; b) = -n \log(2b) - \frac{1}{b} \sum_{i=1}^{n} |x_i|. \tag{5.2}$$

La dérivée s'annule lorsque :

$$\frac{\partial \log L(x_1, \ldots, x_n; b)}{\partial b} = 0 \tag{5.3}$$

$$\Leftrightarrow \quad \frac{1}{b^2}\left(\sum_{i=1}^{n} |x_i| - nb\right) = 0 \tag{5.4}$$

$$\Leftrightarrow \quad \hat{b} = \frac{1}{n} \sum_{i=1}^{n} |x_i|. \tag{5.5}$$

La dérivée seconde s'écrit :

$$\frac{\partial^2 \log L(x_1, \ldots, x_n; b)}{\partial b^2} = \frac{n}{b^2} - \frac{2}{b^3} \sum_{i=1}^{n} |x_i| = \frac{1}{b^2}\left(n - \frac{2}{b} \sum_{i=1}^{n} |x_i|\right). \tag{5.6}$$

Pour que $\hat{b}$ soit un maximum, il faut que la dérivée première s'annule en $b = \hat{b}$ et que la dérivée seconde soit négative au point critique $b = \hat{b}$:

$$\frac{\partial^2 \log L(x_1, \ldots, x_n; b)}{\partial b^2} \leq 0 \tag{5.7}$$

$$\Leftrightarrow \quad \frac{1}{b^2}\left(n - \frac{2}{b}\sum_{i=1}^{n}|x_i|\right) \leq 0 \tag{5.8}$$

$$\Leftrightarrow \quad b \leq \frac{2}{n}\sum_{i=1}^{n}|x_i|. \tag{5.9}$$

Or $\hat{b} = \frac{1}{n}\sum_{i=1}^{n}|x_i|$, donc la dérivée seconde est négative en $\hat{b}$, il s'agit donc d'un maximum.

Par conséquent, l'estimateur du maximum de vraisemblance pour la distribution laplacienne continue est : $\hat{b} = \frac{1}{n}\sum_{i=1}^{n}|v_{k,i}|$.

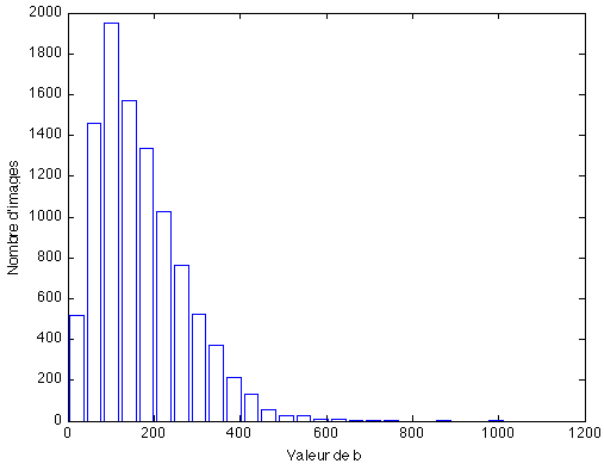
Pour une distribution laplacienne quantifiée avec un pas de quantification Δ et de moyenne nulle, Price et Rabbani [63] proposent l'estimateur suivant :

$$\hat{b}_\Delta = -\frac{\Delta}{2\ln(\gamma)} \quad \text{avec } \gamma = \frac{-n_0\Delta}{2n\Delta + 4S} + \frac{\sqrt{n_0^2\Delta^2 - (2n_1\Delta - 4S)(2n\Delta + 4S)}}{2n\Delta + 4S}, \tag{5.10}$$

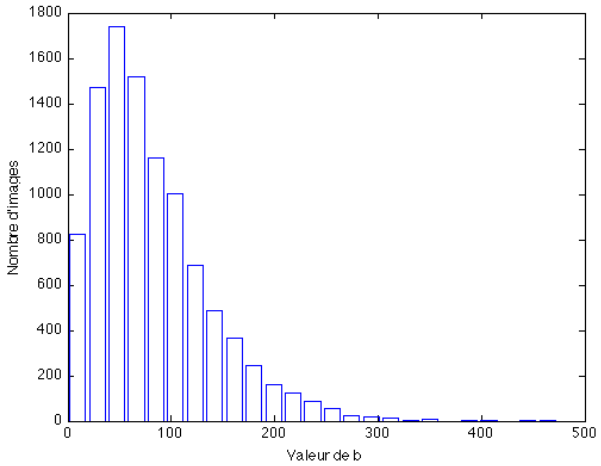
où n_0 est le nombre de coefficients nuls, n_1 le nombre d'observations non nulles, n le nombre d'observations totales ($n = n_0 + n_1$) et $S = \sum_{i=1}^{n}|x_i|$. Si $S = 0$, l'estimateur n'est pas valide, il s'agit du cas où le vecteur de coefficients de contient que des zéros, il ne sera donc pas utilisé pour la détection d'informations cachées.

Il est à noter que pour une image de couverture ou une image stéganographiée par Jsteg, les estimations de b sont très proches.

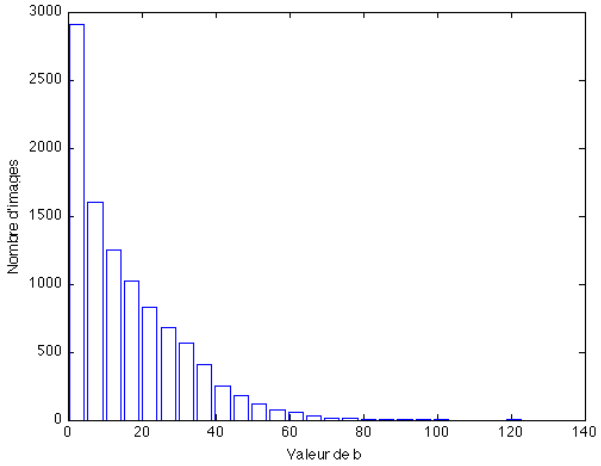
La figure 5.7 présente des histogrammes réalisés à partir de la base d'images BOSS compressée au format JPEG. Les histogrammes de la valeur moyenne du paramètre de la distribution laplacienne par fréquence mènent à plusieurs constatations. Tout d'abord, l'histogramme de la fréquence $k = 2$ s'apparente à une distribution asymétrique de valeur médiane 150, où la valeur estimée du paramètre varie de 0 à 1000. L'asymétrie demeure présente, quelle que soit la fréquence analysée. Cela met en évidence que pour une même fréquence et un même pas de quantification, le contenu d'une image (i.e. la scène photographiée) impacte la valeur du paramètre de la distribution laplacienne.



(a) k = 2



(b) k = 5



(c) k = 15

FIGURE 5.7 – *Histogramme de la valeur de $\hat{b}$ sur la base BOSS compressée avec facteur de qualité 50 pour les coefficients 2, 5 et 15 (dans l'ordre zigzag).*

5.2.4 Étude des performances théoriques du test pour une fréquence

Les figures 5.8, 5.9 et 5.10 présentent les performances théoriques du test pour une seule fréquence. Les courbes de la figure 5.8 sont donc obtenues à partir du calcul théorique de la puissance (voir équation 4.9) pour un taux d'insertion $R = 0.2$, un nombre de coefficients DCT fixé à $n = 10^4$ et en faisant varier le paramètre b de la distribution laplacienne. Pour les figures 5.9 et 5.10, les paramètres qui varient sont respectivement le nombre de coefficients DCT et le taux d'insertion.

La figure 5.8, qui compare la puissance du test pour différentes valeurs du paramètre laplacien, nous permet de constater que la détection est meilleure pour une valeur plutôt petite. Les fréquences possédant cette caractéristique, à savoir un paramètre estimé $\hat{b}$ petit, correspondent soit à de hautes fréquences (voir Fig. 5.3), soit à une forte compression (qualité de compression faible : Q50 par exemple).

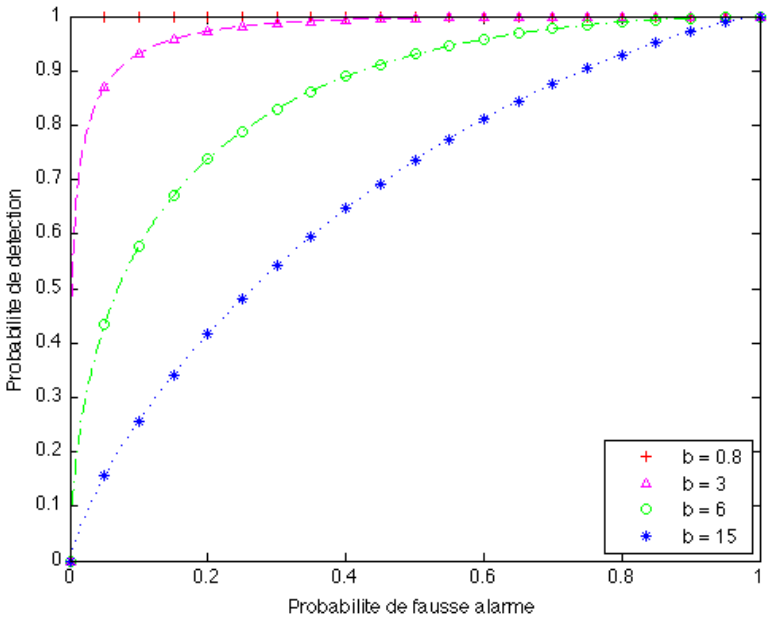


FIGURE 5.8 – *Comparaison de la puissance théorique du test proposé en fonction de la probabilité de fausse alarme pour différentes valeur du* ***paramètre*** *b. Le taux d'insertion est* $R = 0.20$*, et le nombre de coefficients DCT est* 10^4.

La figure 5.9 compare les performances théoriques du détecteur en faisant varier le nombre de coefficients DCT. Nous constatons que le détecteur est meilleur lorsque le nombre de coefficients DCT est élevé, c'est le cas des grandes images.

Pour ce qui est du taux d'insertion, plus il est élevé, meilleurs sont les résultats. Cela était prévisible, et la figure 5.10, qui représente respectivement la puissance du test en fonction du taux d'insertion, confirme cela. En effet, augmenter le taux d'insertion revient à modifier une proportion plus importante de l'image, les modifications sont donc plus facilement décelables.

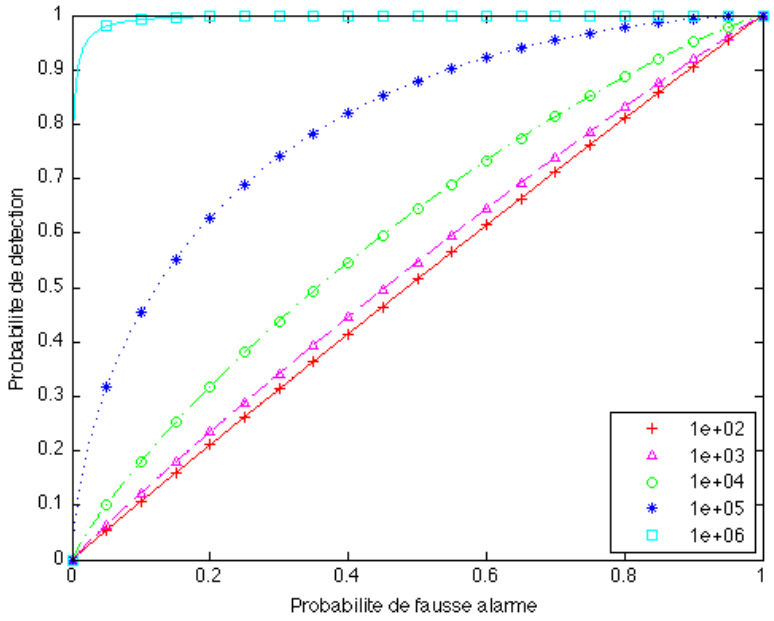


FIGURE *5.9 –* *Comparaison de la puissance théorique du test proposé en fonction de la probabilité de fausse alarme pour différents* ***nombres de coeffcients DCT****,* $R = 0.05$*,* $b = 6$*.*

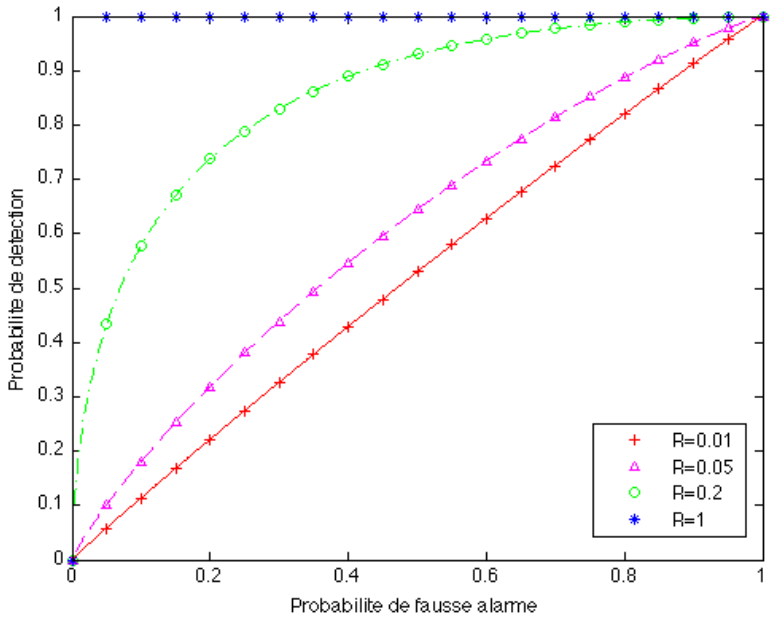


FIGURE *5.10 –* *Comparaison de la puissance théorique du test proposé en fonction de la probabilité de fausse alarme pour différents* ***taux d'insertion****,* $b = 6$ *et* 10^4 *échantillons.*

5.2.5 Nombre de coefficients utilisables n_k

Après avoir visualisé les performances théoriques obtenues pour une fréquence, nous allons observer et analyser le nombre de coefficients utilisables par fréquence. Rappelons que dans le cas de Jsteg, les coefficients utilisables sont les coefficients dont la valeur diffère de 0 et 1.

La figure 5.11 représente le nombre moyen de coefficients utilisables par fréquence (dans l'ordre zigzag). Les données ont été calculées à partir des images de la base BOSS compressées avec un facteur de qualité 50. Nous pouvons remarquer qu'au delà du coefficient 20, il n'y a que très peu d'observations disponibles. Comme dans la sous-section précédente, nous avons constaté que les performances du détecteur étaient meilleures pour les hautes fréquences. Or, dans les hautes fréquences, très peu de coefficients sont utilisables, il pourrait donc être envisagé de ne prendre en considération qu'un certain nombre de fréquences afin d'optimiser le détecteur.

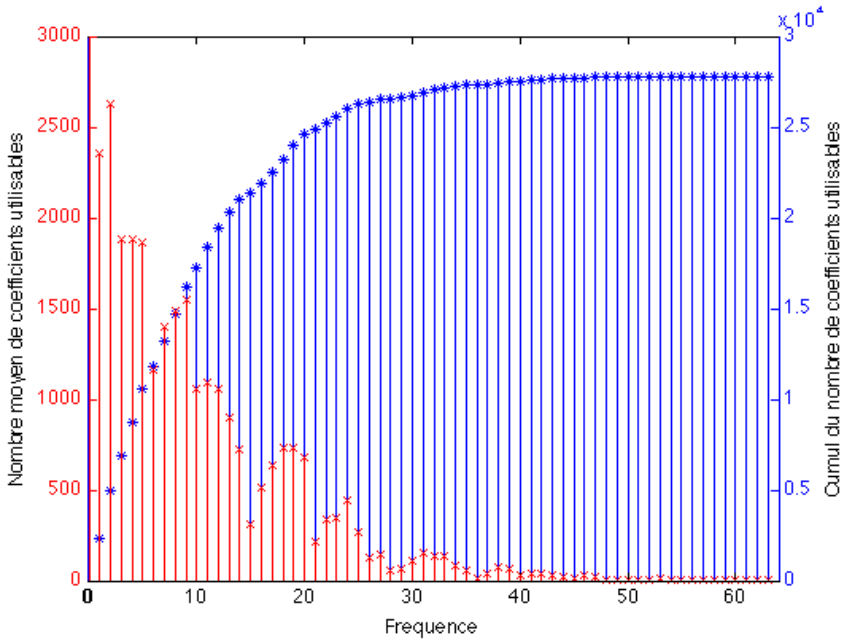
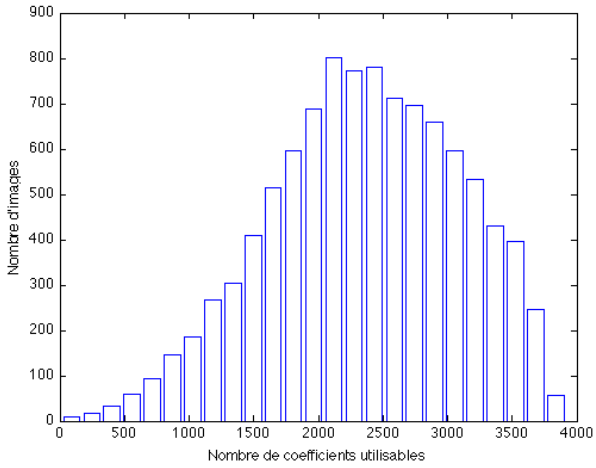
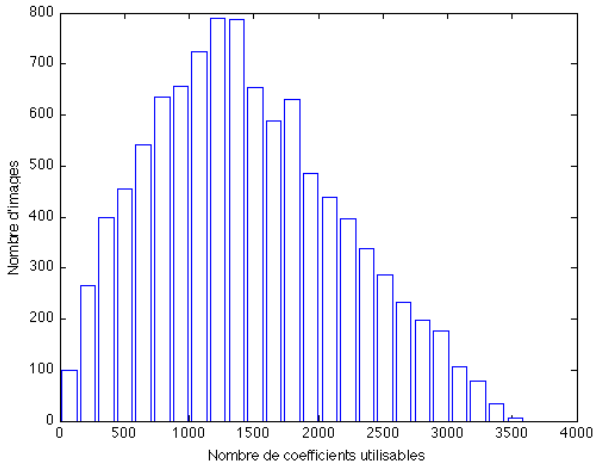


FIGURE 5.11 – *Nombre moyen de coefficients utilisables des coefficients DCT AC par fréquence (x) et cumul (*) sur la base BOSS compressée avec facteur de qualité 50.*

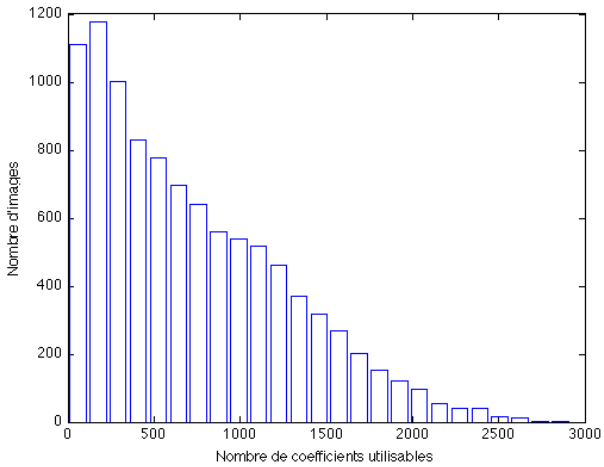
Les histogrammes du nombre moyen de coefficients utilisables par fréquence sont présentés sur la figure 5.12. Pour chaque fréquence, $k = 2$, $k = 5$ et $k = 15$, on observe que les images ont un nombre très disparate de coefficients utilisables. Cela provient du fait que la nature de la scène d'une image a un impact non négligeable sur le nombre de coefficients utilisables. Cette particularité, liée au format JPEG, a un impact sur les performances du détecteur puisque nous comparerons une décision prise à partir d'un nombre très différent de coefficients. Ce n'est en effet pas le cas dans le domaine spatial, où pour une image comportant n pixels, les n pixels sont utilisables pour la détection.



(a) k = 2



(b) k = 5



(c) k = 15

FIGURE 5.12 – *Histogramme du nombre de coefficients utilisables sur la base BOSS compressée avec facteur de qualité 50 pour les coefficients 2, 5 et 15 (dans l'ordre zigzag). En abscisse est représenté le nombre de coefficients DCT utilisables, en ordonnée le nombre d'images.*

5.2.6 Impact du taux d'insertion

Après avoir étudié les performances théoriques du détecteur par fréquence et étudié les histogrammes des coefficients utilisables, nous souhaitons étudier les performances de notre détecteur sur des images réelles et des images simulées pour différents taux d'insertion.

Les images simulées ont été réalisées à partir des images de la base BOSS, où le paramètre de la distribution laplacienne a été estimé pour chaque fréquence. Ensuite, des vecteurs de coefficients DCT laplaciens simulés ont été générés de manière pseudo-aléatoire à partir de ces estimations.

Dans le cadre de l'utilisation de Jsteg, plus le taux d'insertion est important, plus il y a de modifications dans une image. Ainsi, comme on peut s'y attendre, le détecteur obtient de meilleures performances pour un taux d'insertion élevé (R=0.2 par exemple). En effet, le détecteur possède une puissance supérieure pour une même probabilité de fausse alarme. Cela peut être observé sur la figure 5.13 qui présente les courbes COR pour différents taux d'insertion pour des images réelles et des images simulées.

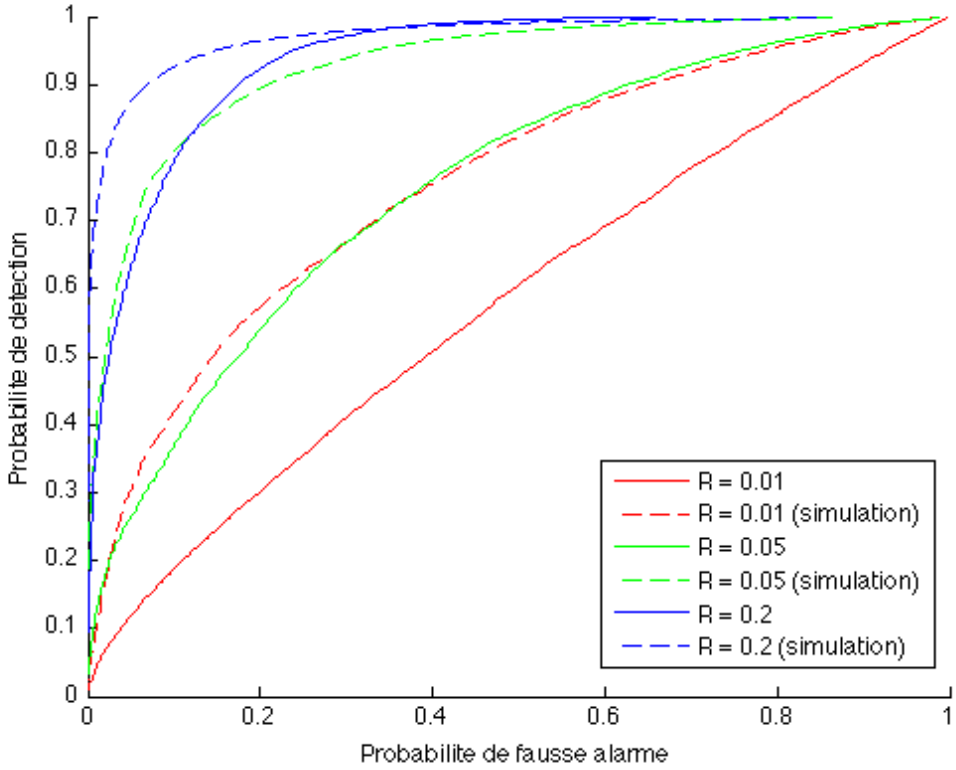


FIGURE 5.13 – *Comparaison de la puissance du test proposé en fonction de la probabilité de fausse alarme pour différents taux d'insertion sur la base BOSS quantifiée avec le facteur de qualité 50. En trait plein sont représentées les courbes COR correspondant aux images réelles, les images simulées étant représentées en pointillés.*

5.2.7 Impact du pas de quantification sur les performances du test

Nous souhaitons à présent étudier les performances de notre détecteur sur des images réelles et des images simulées pour différents taux de compression.

Une compression Q 100 correspond à une compression sans perte, c'est-à-dire que tous les pas de quantification sont égaux à 1. La compression avec perte correspond par exemple aux compressions Q50 et Q100. Plus la compression est importante, plus les pas de quantification utilisés sont grands.

La figure 5.14 présente donc les performances du détecteur pour des images simulées et des images réelles de la base BOSS pour différentes compressions.

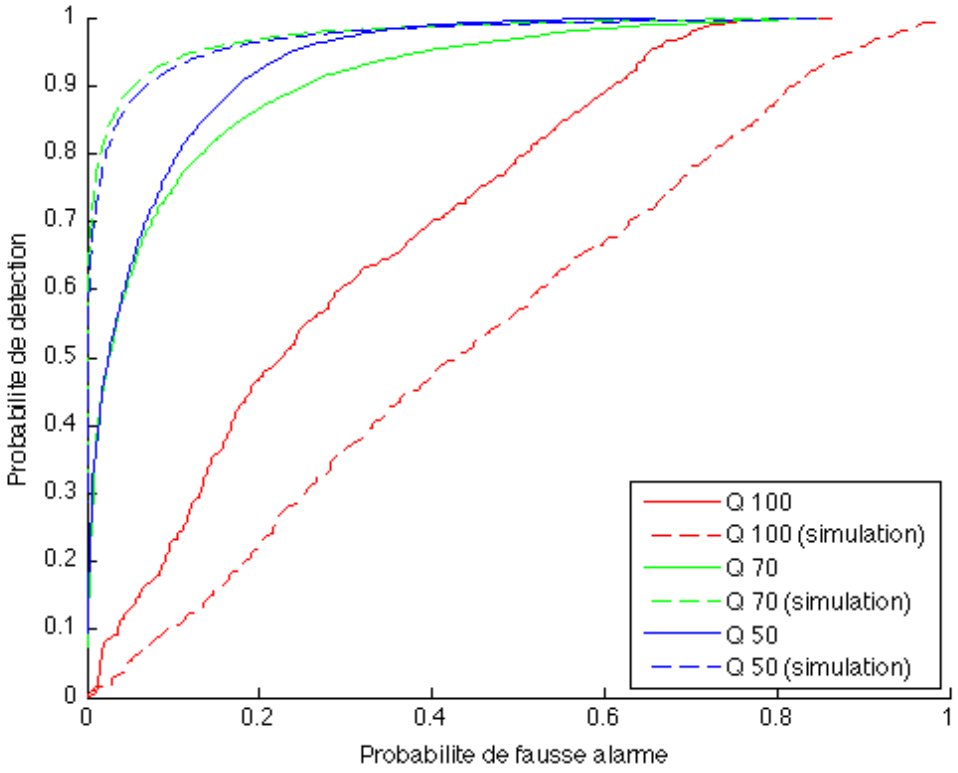


FIGURE 5.14 – *Comparaison de la puissance du test proposé en fonction de la probabilité de fausse alarme pour différents taux de compression sur la base BOSS. En trait plein sont représentées les courbes COR correspondant aux images réelles, les images simulées étant représentées en pointillés.*

Nous pouvons déduire de la figure 5.14 que, plus la compression est forte, meilleurs sont les résultats. En effet, en comparant les courbes des images compressées sans perte (Q 100) avec les courbes compressées avec perte (Q 50 et Q 70), nous observons une perte de puissance importante. Cela résulte du fait que pour les images compressées sans perte, le paramètre estimé de la distribution laplacienne est relativement grand par rapport à celui des images compressées, le détecteur étant meilleur pour un paramètre petit, la perte de puissance est expliquée (voir

section 5.2.4). D'autre part, on constate, pour les compressions Q 70 et Q 50, que les performances sont similaires, cela résulte du compromis entre l'ordre de grandeur du paramètre de la laplacienne et le nombre de coefficients utilisables. En effet, plus le nombre de coefficients utilisables est important, meilleure est la détection pour une même valeur du paramètre laplacien (voir section 5.2.4).

5.2.8 Comparaison des performances du test avec des détecteurs existants

Dans cette sous-section, notre détecteur est comparé au détecteur proposé par Zhang et Ping (ZP) [90] ainsi qu'à l'attaque par catégorie (CA) de Lee et al. [54, 53]. La base de données utilisée est toujours la base BOSS contenant 10 000 images de taille 512×512 compressées au format JPEG avec différents facteurs de qualité. Pour les images simulées, les paramètres estimés $\hat{b}_k$ ont été utilisés pour générer les vecteurs V_k en utilisant un générateur pseudo-aléatoire.

La figure 5.15 présente la puissance du test en fonction de la probabilité de fausse alarme pour des images simulées à partir de l'estimation de $\hat{b}$ des images de la base BOSS. Les simulations sont faites pour le facteur de qualité 50 et pour 1000 coefficients DCT par fréquence.

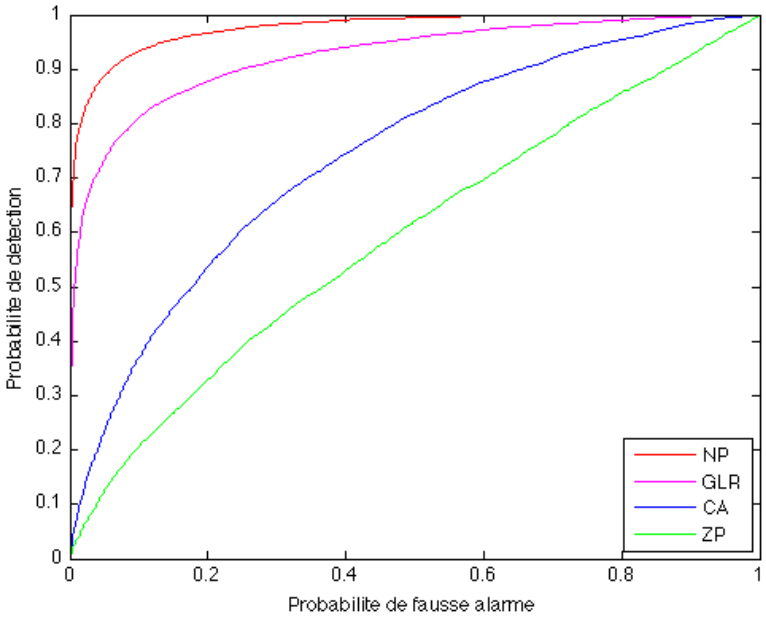
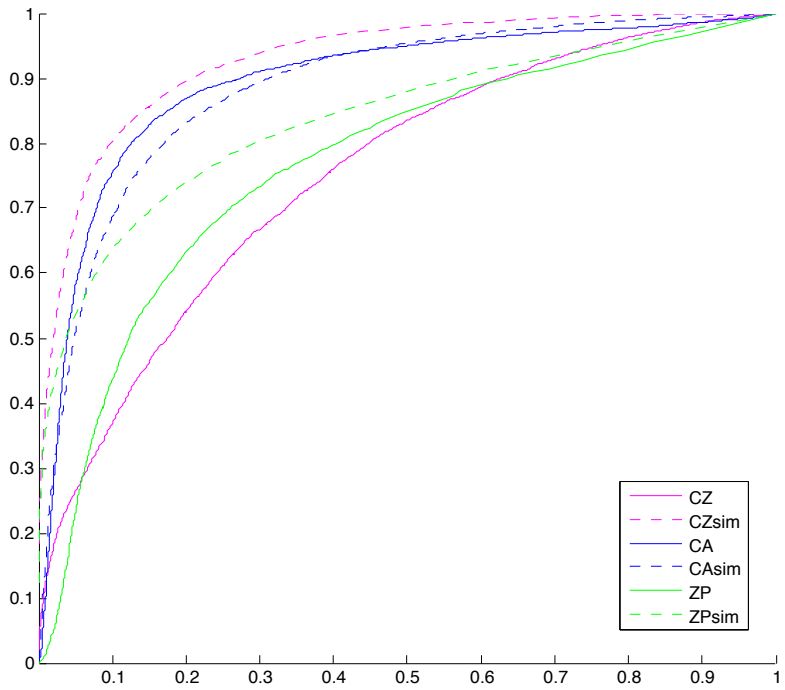
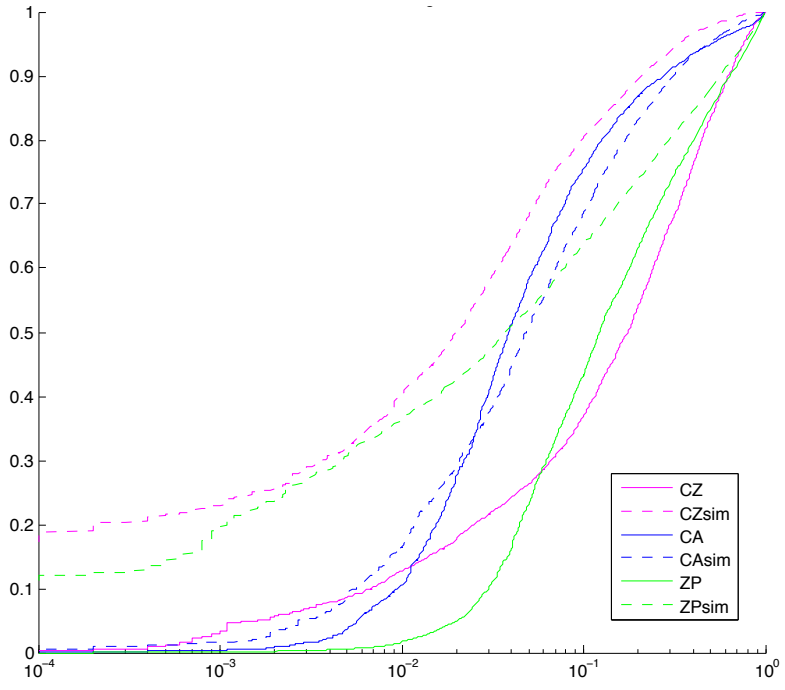


FIGURE 5.15 – *Comparaison de la puissance des tests proposés en fonction de la probabilité de fausse alarme pour la base BOSS* ***simulée*** *avec un facteur de qualité 50 et un taux d'insertion* $R = 0.05$.

Ainsi, pour des images simulées, nous pouvons voir que les tests que nous proposons : le test de Neyman-Person pour lequel le paramètre b est connu (NP), et notre détecteur où le paramètre de la laplacienne est estimé (GLR), ont des performances supérieures aux algorithmes ZP et CA. Il s'agit là d'un cas qui nous est favorable, étant donné que notre test est conçu pour des données laplaciennes.



(a) Échelle linéaire



(b) Échelle semi-logarithmique

FIGURE *5.16* – *Comparaison des courbes de puissance sur des images réelles et simulées. Le taux d'insertion considéré est* $\mathbf{R} = \mathbf{0.05}$*. Les images sont compressées avec un* ***facteur de qualité 50****.*

Comme nous pouvons le voir sur la figure 5.16, le détecteur proposé a de meilleures performances que les algorithmes ZP et CA pour les données simulées (pointillés). Cela est expliqué par le fait que les données expérimentales correspondent au modèle théorique. Cependant, ce n'est pas totalement le cas pour les images réelles (traits pleins). Soulignons l'écart de performance important du détecteur que nous proposons entre les images simulées et les images réelles. Cela signifie que la modélisation des vecteurs V_k pourrait être améliorée afin de réduire la perte de puissance obtenue pour des données réelles par rapport aux données simulées.

Le comportement de notre détecteur sur les images réelles nous permet donc de dire que la distribution laplacienne ne modélise pas parfaitement les vecteurs de coefficients DCT, et par conséquent, la perte de puissance est très importante. Cependant, en utilisant une distribution plus adéquate (distribution gaussienne généralisée ou Gamma par exemple), et en construisant un détecteur sur le modèle proposé dans la section 4.1.2, de meilleurs résultats auraient certainement été obtenus.

5.3 Détection basée sur les effondrements pour F3

Dans cette section sont présentées les simulations validant les résultats obtenus dans le chapitre précédent, section 4.2. Nous rappelons que les calculs sont présentés pour un taux d'insertion $R = 1$. Il en est donc de même pour les expérimentations numériques qui sont présentées dans cette section.

Après insertion d'un message caché, la distribution des coefficients DCT est modifiée. Comme cela a été vu, lors de l'insertion les algorithmes F3, F4 et F5 créent des effondrements. L'étude de ces effondrements a permis de calculer la distribution théorique des coefficients DCT après insertion.

5.3.1 Distribution de τ_1 et τ_2

Afin de calculer la probabilité qu'un effondrement apparaisse, il a été nécessaire de calculer le nombre moyen de coefficients insérés successivement ($\mathbb{E}(\tau_1)$) sans qu'il n'y ait de problème, ainsi que le nombre moyen d'effondrements successifs pouvant apparaître ($\mathbb{E}(\tau_2)$). Les simulations ont été faites pour une distribution gaussienne discrétisée de moyenne nulle et de variance σ^2 comportant 10^6 échantillons.

La figure 5.17 représente la distribution de τ_1 pour différents écart-types σ. En rouge (resp. en bleu) est représentée la distribution pour $\sigma = 3$ (resp. $\sigma = 1$) pour 10^6 échantillons. Les distributions théoriques sont tracées en pointillés. Nous pouvons remarquer que la distribution empirique coïncide parfaitement avec la distribution théorique calculée dans la section 4.2.3.1.

De même, la figure 5.18 représente la distribution de τ_2 pour les écart-types $\sigma = 1$ et $\sigma = 3$. Les distributions théoriques et empiriques de τ_2 coïncident très bien. Ces deux figures nous permettent de valider la pertinence de la méthodologie utilisée dans la section 4.2.3.1.

Nous pouvons également remarquer qu'il y a d'autant plus d'effondrements successifs que σ est petit. C'est ce qui est mis en évidence sur la figure 5.18. Cela provient de la proportion

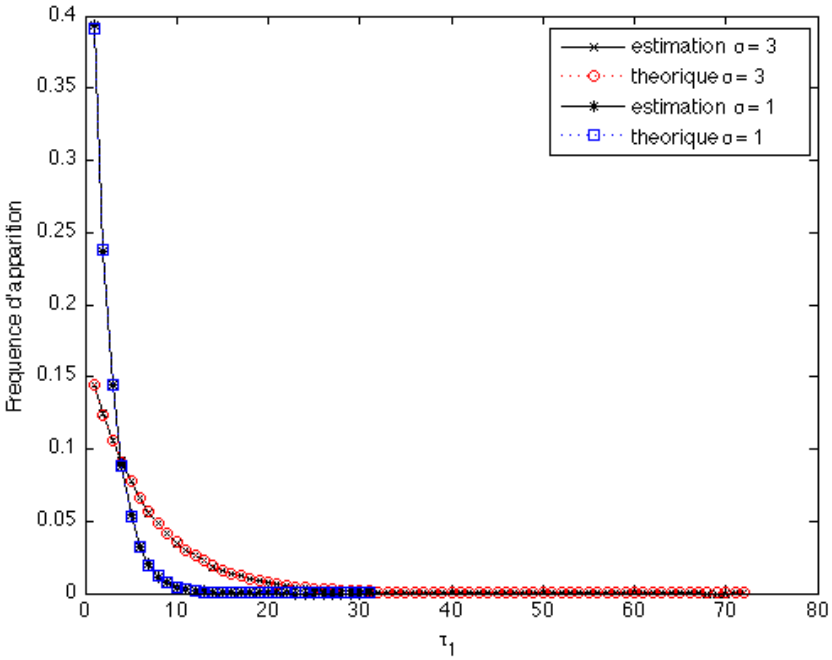


FIGURE 5.17 – *Distribution de τ_1 pour 10^6 échantillons.*

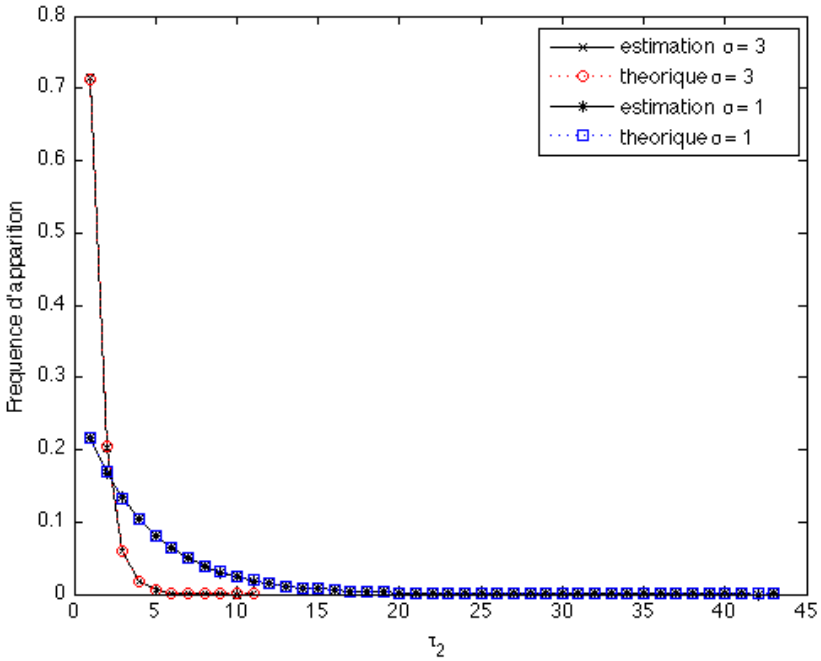


FIGURE 5.18 – *Distribution de τ_2 pour 10^6 échantillons.*

plus importante de coefficients pouvant provoquer un effondrement, i.e. il y a davantage de 1 et de -1 pour un faible σ.

5.3.2 Distribution des coefficients DCT sous $\mathcal{H}_1$

Après avoir vérifié que la distribution empirique de τ_2 reflétait les calculs théoriques, nous allons vérifier que la distribution empirique des coefficients DCT après insertion coïncide toujours avec la théorie.

La figure 5.19 représente les distributions théoriques et empiriques sous $\mathcal{H}_0$ et $\mathcal{H}_1$ des coefficients DCT. Les simulations sont faites en considérant 10^6 échantillons.

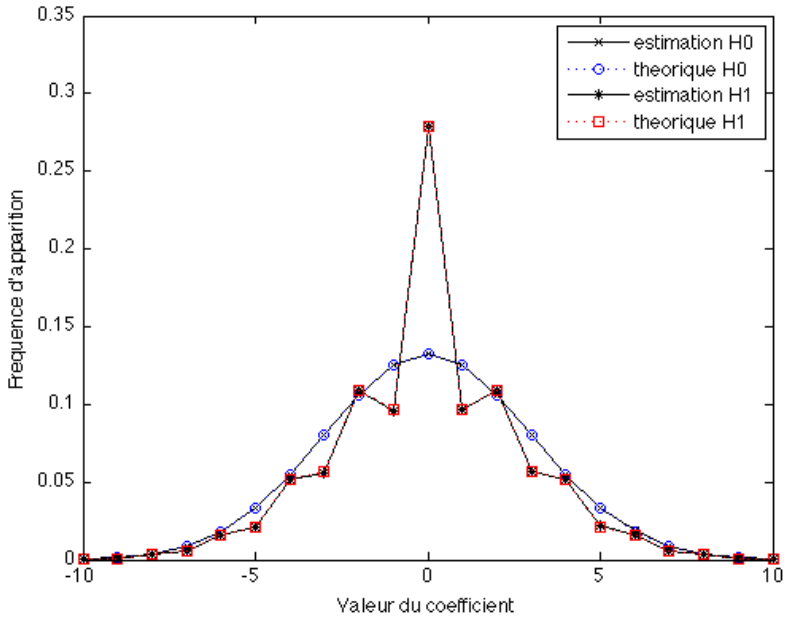


FIGURE 5.19 – *Histogramme théorique et empirique des coefficients sous $\mathcal{H}_0$ et $\mathcal{H}_1$ pour l'algorithme F3 pour 10^6 échantillons et $\sigma = 3$.*

À nouveau, les distributions théoriques et empiriques coïncident parfaitement. La figure 5.18 montrait que plus σ est petit, plus il y a d'effondrements, on remarque sur la figure 5.19, que cela se traduit par l'augmentation importante du nombre de 0 dans l'histogramme des coefficients DCT.

5.3.3 Courbe COR

Regardons à présent comment se comporte le détecteur basé sur le test du rapport de vraisemblance.

La figure 5.20 présente la puissance du test en fonction de la probabilité de fausse alarme. Les données sont simulées pour 100 échantillons et 10^4 tests ont été effectués. La courbe COR est proposée pour $\sigma = 3$ et $\sigma = 8$.

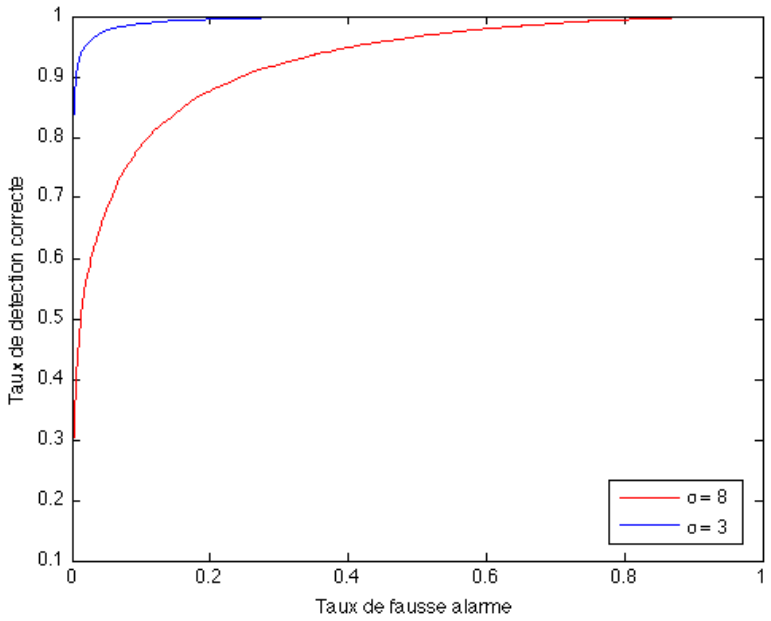


FIGURE 5.20 – *Courbe COR pour la détection de F3 pour* 10^4 *simulations de 100 échantillons et différentes valeurs de* σ.

Le test est donc d'autant plus performant que σ est petit, ce qui confirme ce que l'on pouvait supposer, à savoir que pour σ petit, il y a davantage de coefficients pouvant subir un effondrement, par conséquent, il y a davantage d'effondrements lors de l'insertion et cela se traduit par une meilleure détection.

De la même manière, à partir des résultats obtenus dans le chapitre 4, nous aurions pu montrer que les résultats empiriques coïncidaient avec les calculs théoriques pour l'algorithme F4.

5.4 Détection dans les images non compressées

Dans cette section sont présentés les résultats obtenus pour la détection dans les images non compressées (voir section 4.3).

5.4.1 Comparaison entre les tests PP et localement PP

Nous souhaitons observer la perte d'optimalité du test le plus puissant basé sur le rapport de vraisemblance $\log \Lambda_1(Z_n)$ (voir équation 4.97), construit pour $R = 1$ par rapport au test localement le plus puissant donné par (4.130) avec $r^* = 0.05$ et le test le plus puissant basé sur $\log \Lambda_{\widetilde{R}}(Z_n)$ lorsque le taux d'insertion réel est $\widetilde{R} = 0.1$.

Considérons le modèle d'image gaussien suivant : $\theta = 129, \sigma = 1$ et $n = 10^4$. La comparaison de la puissance théorique $\beta = \beta(\alpha_0)$ comme fonction du taux de fausse alarme α_0 pour ces tests, est présentée sur la figure 5.21. 10^5 simulations de Monte Carlo ont été réalisées.

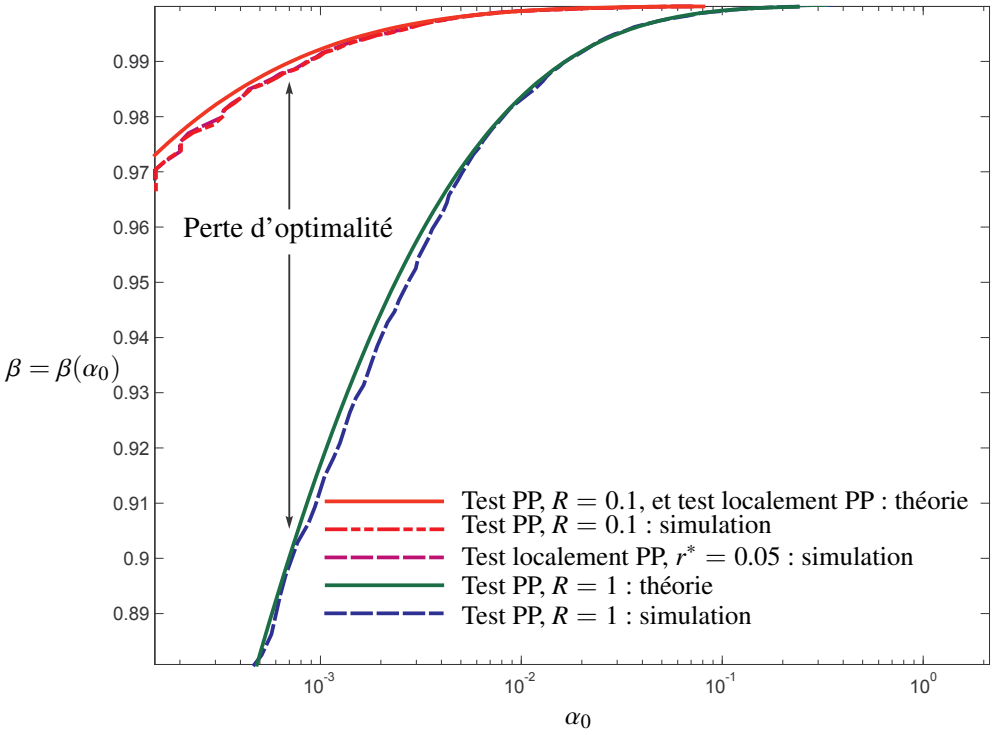


FIGURE 5.21 – *Courbe COR du test le plus puissant construit pour les taux d'insertion $R = 1$ et $R = 0.1$, ainsi que le test localement le plus puissant pour $r^* = 0.05$. Le taux réel d'insertion est $\widetilde{R} = 0.1$ et $\sigma = 1$.*

Sur cette figure, le taux d'insertion réel est $\widetilde{R} = 0.1$, et pour le test le plus puissant (PP), les taux d'insertion testés sont $R = 0.1$ (cas idéal) et $R = 1$ (le pire cas). Il est donc normal d'observer une perte d'optimalité dans le cas où $R = 1$. Une augmentation de σ ou du taux réel d'insertion $\widetilde{R}$ réduirait l'écart entre les tests comparés. Il est donc important de noter que malgré

la perte de puissance observée, les résultats des tests restent très bons. Pour ce qui est du test localement le plus puissant, il est lui aussi testé dans un cas favorable, puisque $r^* = 0.05$, nous pouvons souligner les bonnes performances de ce test, puisqu'elles sont similaires au test le plus puissant testé pour $R = 0.1$.

5.4.2 Modèle régressif de l'image

Dans la section 4.3.6, nous avons modélisé l'image de couverture de manière plus réaliste par une approximation polynomiale, bien que cela ne permette pas la parfaite prise en compte de la structure de l'image. Ainsi, la figure 5.22 représente un vecteur de pixels (i.e. une ligne de l'image) et son approximation en utilisant le modèle régressif présenté dans la section 4.3.6. Ici, H possède $l = 5$ colonnes, les segments sont de longueur $k = 16$ pixels et M est le nombre de segments, soit $n = M \times k$ pixels au total. Le j-ième segment, noté Z_j est approximé par $H\widehat{x}_j$.

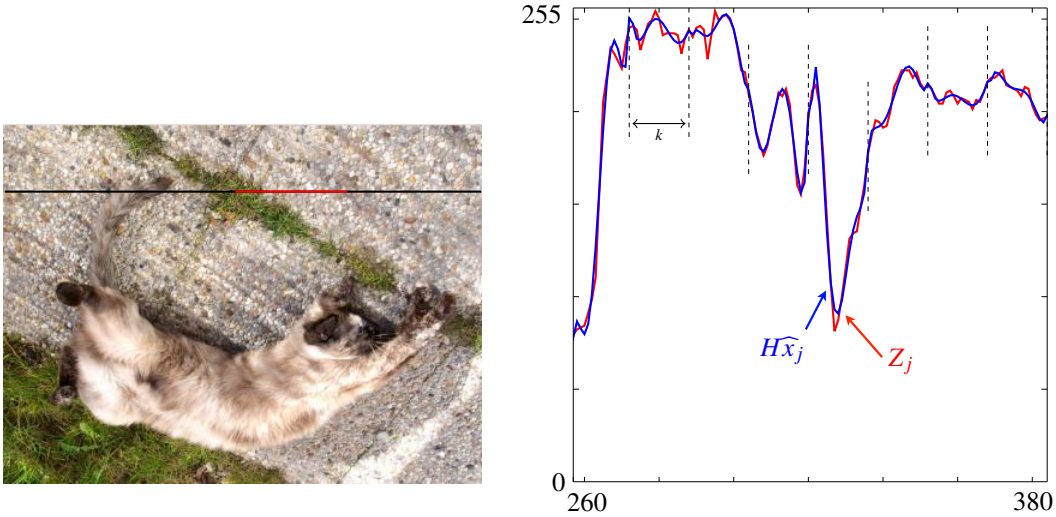


FIGURE 5.22 – *Modèle régressif de l'image.*

Différents découpages de l'image peuvent être faits : nous pouvons en effet considérer des découpages par segment dans toutes les directions (horizontale, verticale, diagonale, etc.), on parlera alors de 1D ou encore des découpages par blocs, i.e. 2D.

Schéma fonctionnel de l'algorithme 2D

La figure 5.23 présente le schéma fonctionnel de l'algorithme de détection dans le domaine spatial en considérant non pas des vecteurs (1D) comme on peut le voir sur la figure 5.22, mais des blocs (2D) de dimension $b_s \times b_s$. Pour construire le vecteur Y à partir des blocs de l'image, les pixels peuvent être récupérés selon l'ordre voulu (lexicographique, zigzag, etc.).

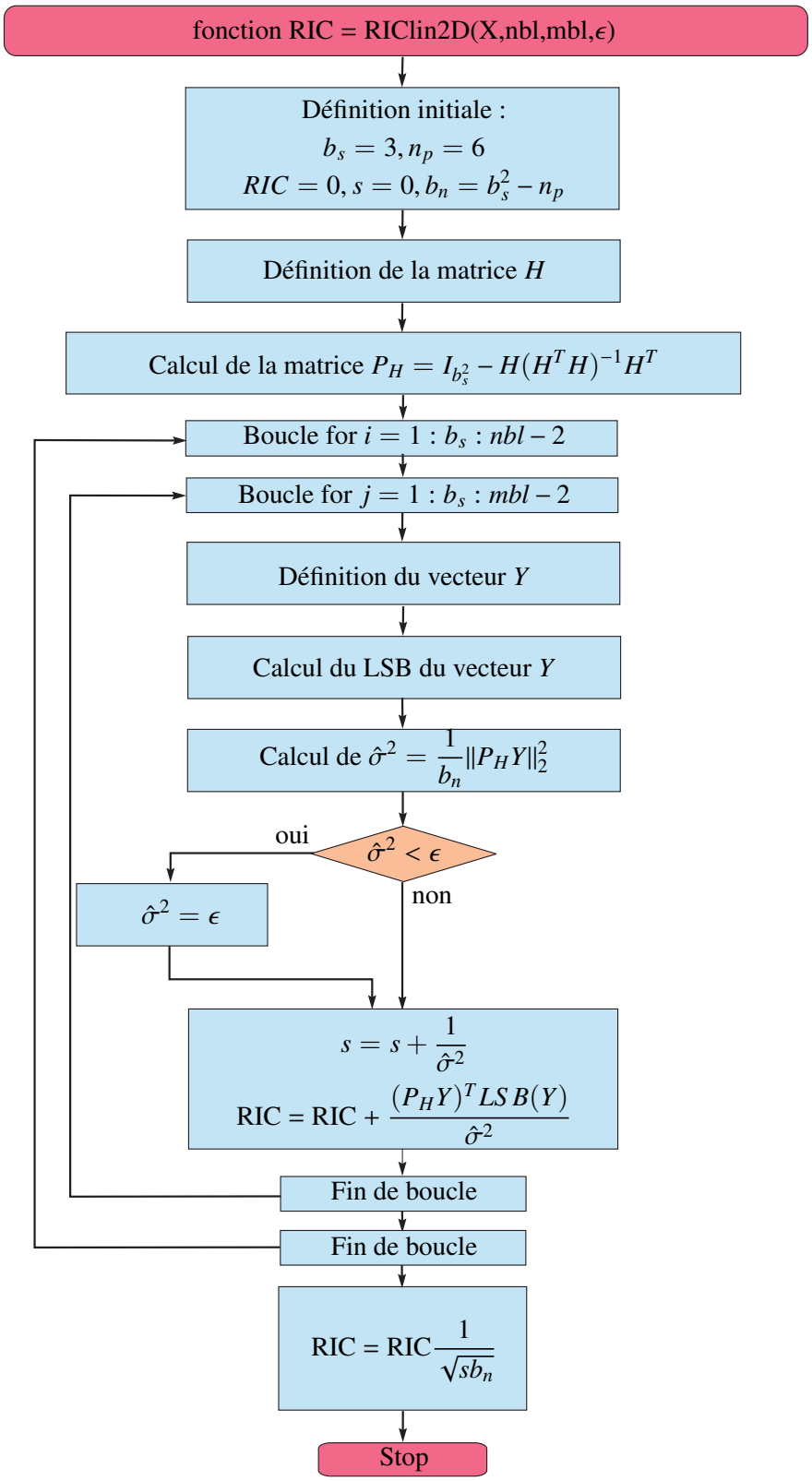


FIGURE 5.23 – *Schéma bloc de l'algorithme RIC linéaire 2D.*

La variable *RIC* représente la statistique de décision, b_s la taille des blocs et n_p le degré du polynôme. Pour chaque bloc, le bloc est mis sous la forme du vecteur Y, puis le plan des LSB est récupéré. Ensuite, la variance du bloc est calculée ; si elle est inférieure à un seuil fixé, on substitue la variance par la valeur de ce seuil. La statistique de décision totale est la somme normalisée de la statistique de chaque bloc. Pour finir, il faut comparer cette statistique *RIC* au seuil de décision obtenu pour une probabilité de fausse alarme fixée.

Comparaison des performances des algorithmes 1D et 2D

Nous souhaitons à présent comparer les performances des algorithmes 1D et 2D en fonction du degré de l'approximation.

La figure 5.24 représente les courbes COR du détecteur 1D pour un taux d'insertion $R = 0.05$, pour 1000 images segmentés par vecteurs de 9 pixels et différents degrés d'approximation. La puissance du test est tracée en fonction de la probabilité de fausse alarme : pour une probabilité de fausse alarme inférieure à 0.02, l'approximation la moins puissance est celle où $n_p = 1$, alors qu'il s'agit de l'approximation où $n_p = 6$ pour un taux de fausse alarme supérieur à 0.02. Cette figure montre donc que l'on obtient globalement de meilleures performances lorsque $n_p = 4$, i.e. pour une approximation de degré 3.

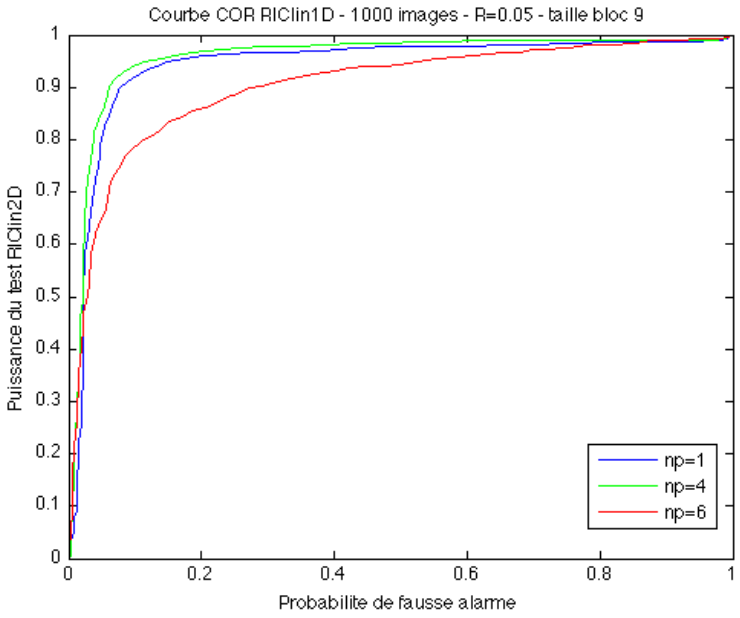


FIGURE 5.24 – *Courbe COR du test RIClin 1D pour un taux d'insertion R=0.05 et une approximation polynomiale de degré $n_p - 1$ pour 1000 images.*

La figure 5.25 représente les courbes COR du détecteur 2D pour un taux d'insertion $R = 0.05$, pour 50 images segmentées par des carrés de taille 3×3 pixels et des approximations polynomiales de différents degrés. Pour un taux de fausse alarme faible, la puissance est croît avec le degré, en revanche, à partir d'un taux de fausse alarme de 0.02, les courbes se superposent

pour certains degrés $n_p = 5, 8$. Ainsi, pour ce qui est du modèle 2D, un algorithme basé sur une approximation polynomiale de degré 4 (i.e. $n_p = 5$) semble convenir.

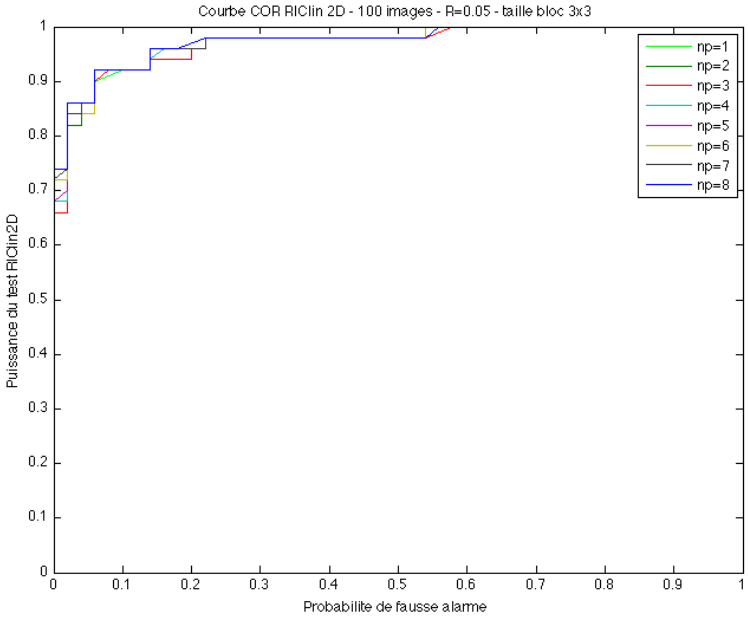


FIGURE 5.25 – *Courbe COR du test RIClin 2D pour un taux d'insertion R=0.05 et une approximation polynomiale de degré $n_p - 1$ pour 100 images.*

Les performances des algorithmes 1D et 2D sont ensuite comparées sur la figure 5.26. Nous pouvons conclure que la modélisation 2D par une approximation polynomiale de degré 4 semble être celle dont les résultats sont les meilleurs, notamment pour un taux de fausse alarme faible, et comparables aux résultats pour $n_p = 4$ en 1D, pour un taux plus important. Ces résultats pourraient être affinés, en utilisant davantage d'images pour faire ces comparaisons.

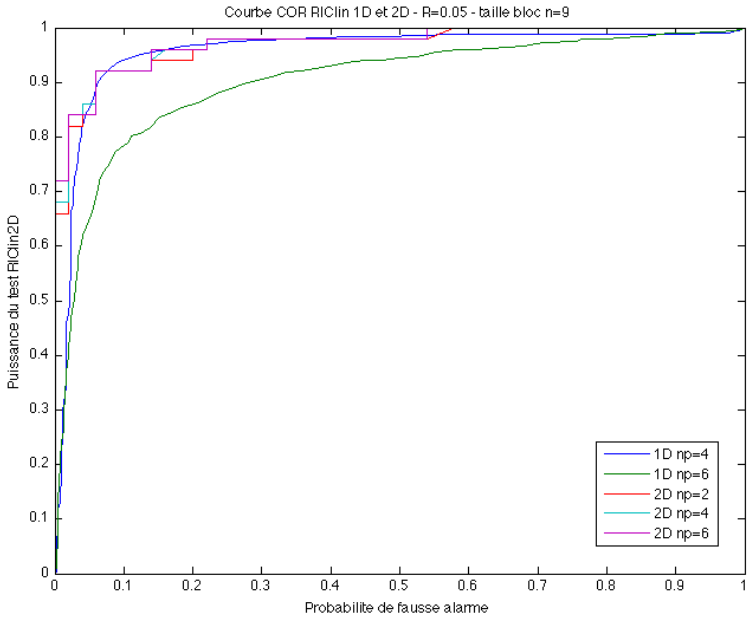


FIGURE *5.26 –* *Courbe COR du test RIClin 1D et 2D pour un taux d'insertion R=0.05 et une approximation polynomiale de degré* $n_p - 1$.

Conclusion

Ce chapitre a fait l'objet d'expérimentations numériques visant à confirmer les résultats théoriques obtenus dans le chapitre 4.

Dans le cadre de la détection de Jsteg, la robustesse du test proposé basé sur la distribution laplacienne des coefficients DCT dans les images JPEG a été vérifiée par rapport à tous les paramètres dont il dépend. En effet, nous avons pu observer les bons résultats obtenus en faisant varier différents paramètres (taux d'insertion, nombre de coefficients utilisables, valeur du paramètre de la distribution, pas de quantification, etc.) dans le cas de données simulées. La comparaison avec d'autres algorithmes a permis de confirmer que les performances de notre test étaient meilleures pour des données simulées. En revanche, dans le cadre des images réelles, le comportement de notre détecteur nous permet de dire que la distribution laplacienne ne modélise pas parfaitement la distribution des coefficients DCT, il est donc envisageable d'obtenir de très bons résultats sur données réelles en considérant une modélisation des coefficients DCT plus réaliste.

L'étude statistique menée dans le chapitre précédent sur l'apparition des effondrements dans le cas de l'utilisation des algorithmes F3, F4 et F5 a été validée en confirmant l'adéquation des résultats empiriques avec les résultats théoriques. En effet, afin de calculer les probabilités d'apparition des coefficients DCT, les variables aléatoires τ_1 et τ_2 représentant respectivement le nombre de coefficients insérés avant l'apparition d'un effondrement et le nombre d'effondrements successifs pouvant apparaître ont été introduites et leur distribution établie de manière théorique. À l'aide de simulations de Monte-Carlo, nous avons pu confirmer les résultats établis. De plus, les histogrammes empiriques et théoriques des coefficients DCT après insertion coïncident parfaitement et la courbe COR obtenue pour le détecteur de Neyman-Pearson est très encourageante pour de futurs travaux.

Enfin, en ce qui concerne les détecteurs dans le domaine spatial, nous avons présenté des résultats comparant nos détecteurs, pour différents taux d'insertion. Ainsi nous avons pu voir que lorsque le test localement le plus puissant est construit pour un taux d'insertion R proche du taux d'insertion réel $\tilde{R}$, les résultats obtenus pour les simulations sont très bons. Des détecteurs utilisant un modèle régressif 1D ou 2D ont ensuite été présentés et nous avons pu comparer leurs bonnes performances en fonction du degré du polynôme, l'algorithme 2D présentant des résultats légèrement meilleurs que l'algorithme 1D.

Chapitre 6

Conclusions et perspectives

Dans ce manuscrit, les problématiques de détection d'information cachée ont été abordées du point de vue de la décision statistique, dans le cadre des images compressées et non compressées. Dans le chapitre 2, après avoir présenté les notions de stéganographie et de stéganalyse, la stéganalyse a été abordée du point de vue de la décision statistique, comme un problème de choix entre deux hypothèses, simples ou composites, basé sur des données quantifiées.

Dans le chapitre 3, les images brutes mais également les images compressées ont été modélisées dans le but de pouvoir construire des détecteurs d'informations cachées fiables. Les données quantifiées qui font l'objet de notre attention, à savoir, les pixels des images non compressées et les coefficients DCT des images JPEG, ont été étudiées plus particulièrement. Les modélisations des coefficients DCT utilisées dans la littérature ont été présentées. La modélisation de ces données permet par la suite de mettre en place les détecteurs paramétriques qui ont été proposés.

Ainsi, dans le chapitre 4, un détecteur basé sur la modélisation laplacienne des coefficients DCT a été mis en place afin de détecter la présence d'information cachée avec Jsteg dans les images compressées au format JPEG. Les calculs des probabilités théoriques du détecteur ont été établies sous des conditions asymptotiques (on considère une image grande). Une étude originale reposant sur les effondrements provoqués par F3, F4 et F5, a permis d'établir théoriquement les probabilités d'apparition des coefficients DCT après insertion. Cela permet la construction d'un détecteur basé sur le nombre d'effondrements apparus lors de l'insertion. En effet, les probabilités calculées permettent d'obtenir la distribution des coefficients avant et après insertion, un test de Neyman-Pearson est donc réalisable. Dans le domaine spatial, le test le plus puissant est établi lorsque les hypothèses sont simples et le taux d'insertion connu. De plus, l'impact de la quantification sur les performances du test a été mis en évidence. Dans le cas d'hypothèses composites, le test localement asymptotiquement le plus puissant proposé permet de constater la perte de puissance due à un taux d'insertion testé différent du taux réel. Enfin, en se basant sur une modélisation paramétrique linéaire locale, l'application pratique des tests proposés a permis de mettre en évidence un lien avec le détecteur WS en calculant théoriquement des paramètres qui jusqu'alors n'étaient déterminés qu'empiriquement.

Les résultats numériques présentés dans le chapitre 5 ont mis en évidence l'adéquation des résultats théoriques avec les simulations. Dans le cadre de la détection de Jsteg, la robustesse

du test proposé basé sur la distribution laplacienne des coefficients DCT dans les images JPEG a été vérifiée par rapport à tous les paramètres dont il dépend. La comparaison avec d'autres algorithmes a permis de confirmer que les performances de notre test étaient meilleures pour des données simulées. L'étude statistique menée dans le chapitre précédent sur l'apparition des effondrements dans le cas de l'utilisation des algorithmes F3, F4 et F5 a été validée en confirmant l'adéquation des résultats empiriques avec les résultats théoriques. À l'aide de simulations de Monte-Carlo, nous avons pu valider les résultats établis, les histogrammes empiriques et théoriques des coefficients DCT après insertion coïncidant parfaitement. Enfin, en ce qui concerne les détecteurs dans le domaine spatial, nous avons présenté des résultats comparant nos détecteurs les plus puissants, et localement les plus puissants pour différents taux d'insertion. Des détecteurs utilisant un modèle régressif 1D ou 2D ont ensuite été comparés en fonction du degré du polynôme, l'algorithme 2D présentant des performances sensiblement meilleures que l'algorithme 1D.

L'approche proposée dans ces travaux, à savoir l'utilisation de tests statistiques paramétriques dans le but de fournir des détecteurs fiables dans le respect de la contrainte de fausse alarme fixée, s'est donc avérée être pertinente dans le sens où les résultats numériques obtenus sont satisfaisants et reposent sur des bases théoriques solides. Les idées et des résultats figurant dans ce manuscrit ont été exposés lors de différentes conférences (voir Annexe B).

Suite aux travaux présentés dans le cadre de cette thèse, certaines améliorations possibles que nous avons identifiées feront l'objet de perspectives à court terme.

- En effet, une étude approfondie du test basé sur une distribution quelconque dans le cadre de la détection de Jsteg (et présenté dans le chapitre 4), en établissant les calculs théoriques des performances du test lorsqu'une modélisation plus réaliste des coefficients DCT est considérée, semble être une bonne approche.
- Après avoir étudié l'impact des différents paramètres sur les performances du détecteur, nous pourrions envisager de sélectionner efficacement les fréquences à utiliser, en pondérant par exemple la statistique de décision obtenue pour chaque fréquence en fonction du nombre de coefficients utilisables, du paramètre de la distribution ou encore du pas de quantification.
- Il peut également être envisagé de faire de la stéganalyse quantitative en créant un estimateur du taux d'insertion à partir de notre statistique de décision.
- Tout comme cela a été fait dans le domaine spatial, l'étude de l'impact de la quantification sur les probabilités d'erreur pourrait faire l'objet d'une étude dans le domaine fréquentiel.
- Aux vues des résultats que nous avons obtenus dans le cadre de l'étude originale de F3 et F4, une étude de F5, tenant compte du codage matriciel est envisagée.

Parmi les perspectives à plus long terme de cette thèse figurent :

- L'adaptation des travaux présentés dans le cadre de la détection dans les fichiers JPEG à d'autres types de fichiers tels que les images JPEG 2000 (utilisant les transformées en ondelettes).
- De manière plus large, la méthodologie présentée pourrait s'avérée efficace pour la stéganalyse de fichiers sons de type MP3 qui utilisent également la transformée en cosinus discrète, ou encore les vidéos (MPEG, H.264).
- L'étude originale proposée dans le cadre de F3 et F4 pourrait être étendue à l'algorithme d'insertion LSB matching ou insertion ± 1.

– Enfin, l'intégration de tous les canaux de couleur dans les détecteurs, qui nécessite de tenir compte de la corrélation entre les différents canaux, sera une étape nécessairement profitable aux détecteurs que nous concevons puisque l'insertion d'informations cachées détruit en partie cette corrélation inter-canaux.

Les travaux réalisés dans le cadre de cette thèse, et plus généralement au sein du projet RIC, ont permis de renforcer les apports de théorie de la décision statistique pour la stéganalyse. La méthodologie proposée s'est avérée être pertinente et les nombreuses perspectives possibles restent un réel challenge pour de futurs travaux.

Annexe A

Format d'image JPEG

Dans cette annexe figurent les étapes de codage de la compression JPEG ainsi que le processus de décompression d'une image et un exemple.

Codage RLE et de Huffman

Après ces étapes, la forme matricielle n'est plus judicieuse pour la transmission de données ou le stockage ; il faut donc trier les valeurs dans un certain ordre et les coder.

Codage différentiel

La composante DC d'un bloc 8×8 étant fortement corrélée avec la composante DC du bloc précédent, ces coefficients vont être codés différemment des coefficients AC, en utilisant un codage différentiel de type DPCM (Differential Pulse Code Modulation). C'est la différence E_k qui sera codée et non le coefficient.

$$E_k = DC_k - DC_{k-1} \text{ , où } DC_{-1} = 0.$$

Lecture zigzag

Ensuite, les coefficients sont lus suivant l'ordre zigzag (Fig. A.1). C'est cette lecture qui a été choisie, car cette opération de mise en ordre a pour objet de classer les résultats dans l'ordre croissant des fréquences, ce qui classe les coefficients les plus significatifs (basses fréquences) en tête et les moins significatifs (hautes fréquences) en queue.

Cette lecture produit une chaîne de 64 nombres qui débute avec des valeurs non-nulles et termine avec de nombreux zéros.

Cette séquence sera codée avec une codage de type RLE puis un codage de Huffman.

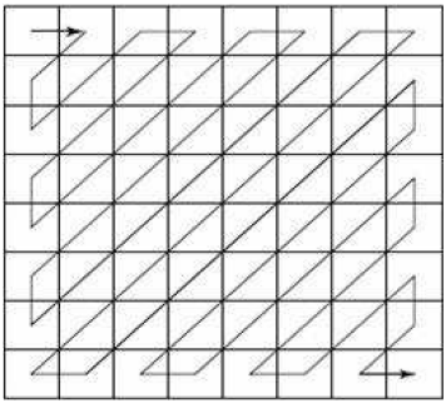


FIGURE A.1 – *Mise en ordre zigzag.*

Première étape : Codage RLE

Le principe du codage RLE (Run Length Encoding) est de coder les répétitions d'un même pixel. Pour ce faire, les coefficients DCT quantifiés d'un bloc sont transformés en une suite de symboles qui sera elle-même codée.

Chaque coefficient AC non nul est représenté par (`symbol-1`, `symbol-2`), une paire de symboles définis par la figure A.2.

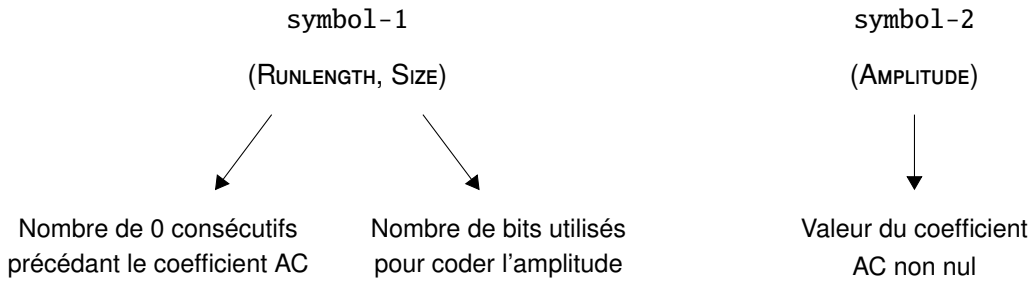


FIGURE A.2 – *Codage RLE pour les coefficients AC.*

RUNLENGTH code la chaîne de 0 précédant un coefficient AC non nul. La valeur de RUNLENGTH est comprise entre 0 et 15. Bien évidemment, la longueur d'une série de 0 peut être supérieure à 15 ; pour la coder, il suffira donc de coder une série de 16 coefficients nuls par `(15,0)`. On peut compter jusqu'à 3 `(15,0)` consécutifs avant d'avoir le `(symbol-1)` qui termine la chaîne de 0. Le dernier `(symbol-1)` est toujours suivi par un `(symbol-2)`, sauf si le dernier coefficient est un 0 et qu'il appartient à une série de 0. Dans ce cas, on utilise le symbole `(0,0)` qui fait office de marqueur de fin de bloc ou EOB (End Of Bloc).

En analysant la DCT sur des blocs 8×8, on remarque que si les pixels sont codés sur 8 bits, alors les coefficients DCT pourront être codés sur 11 bits. Par conséquent, si les pixels appartiennent à l'intervalle $[-128, 127]$, les coefficients DCT auront des valeurs comprises entre -1024 et $1023]$.

Size correspond au nombre de bits nécéssaires pour coder la valeur du coefficient DCT. (Size $\in [1, 10]$.

Amplitude correspond à la valeur du coefficient DCT.

En ce qui concerne les coefficients DC, comme il s'agit de la différence de deux coefficients DC consécutifs qui est codée, la valeur peut être deux fois plus grande, par conséquent, Size $\in [1, 11]$.

`symbol-1`	`symbol-2`
(Size)	(Amplitude)

Figure *A.3* – *Codage RLE pour les coefficients DC.*

Deuxième étape : Codage de Huffman

Le principe du codage de Huffman est de coder ce qui est fréquent avec un minimum de bits et de coder sur des séquences plus longues ce qui apparaît rarement. Dans la compression JPEG, le codage de Huffman dont on se sert, utilise des tables de codages qui figurent dans les en-têtes du fichier. Il existe jusqu'à deux tables pour la luminance et deux tables pour les chrominances.

Size	Amplitude
1	-1,1
2	-3, -2, 2, 3
3	-7..-4, 4..7
4	-15..-8, 8..15
5	-31..-16, 16..31
6	-63..-32, 32..63
7	-127..-64, 64..127
8	-255..-128, 128..255
9	-511..-256, 256..511
10	-1023..-512, 512..1023

Figure *A.4* – *Structure du codage du `symbol-2`.*

Une fois la suite de symboles intermédiaire obtenue, des codes à longueur variable sont associés aux différents symboles. Pour les coefficients AC et DC, les `(symbol-1)` sont codés avec un code VLC (Variable Length Code) provenant de la table de Huffman associée. Chaque `(symbol-2)` est encodé avec un code VLI (Variable Length Integer) dont la longueur des mots de code est indiquée dans le tableau A.5 p. 132.

Category	**Luminance**		**Chrominance**	
	Code Length	Code Word	Code Length	Code Word
0	2	00	2	00
1	3	010	2	01
2	3	011	2	10
3	3	100	3	110
4	3	101	4	1110
5	3	110	5	11110
6	4	1110	6	111110
7	5	11110	7	1111110
8	6	111110	8	11111110
9	7	1111110	9	111111110
10	8	11111110	10	1111111110
11	9	111111110	11	11111111110

FIGURE *A.5* – *Table pour coder les différences entre les coefficients DC pour la luminance et les chrominances.*

Structure des données dans un fichier JPEG

Les paramètres utiles à la décompression sont stockés dans l'en-tête du fichier JPEG. Tous les paramètres sont organisés selon une syntaxe basée sur des marqueurs (tableau A.7). La structure est décrite sur le schéma A.6 :

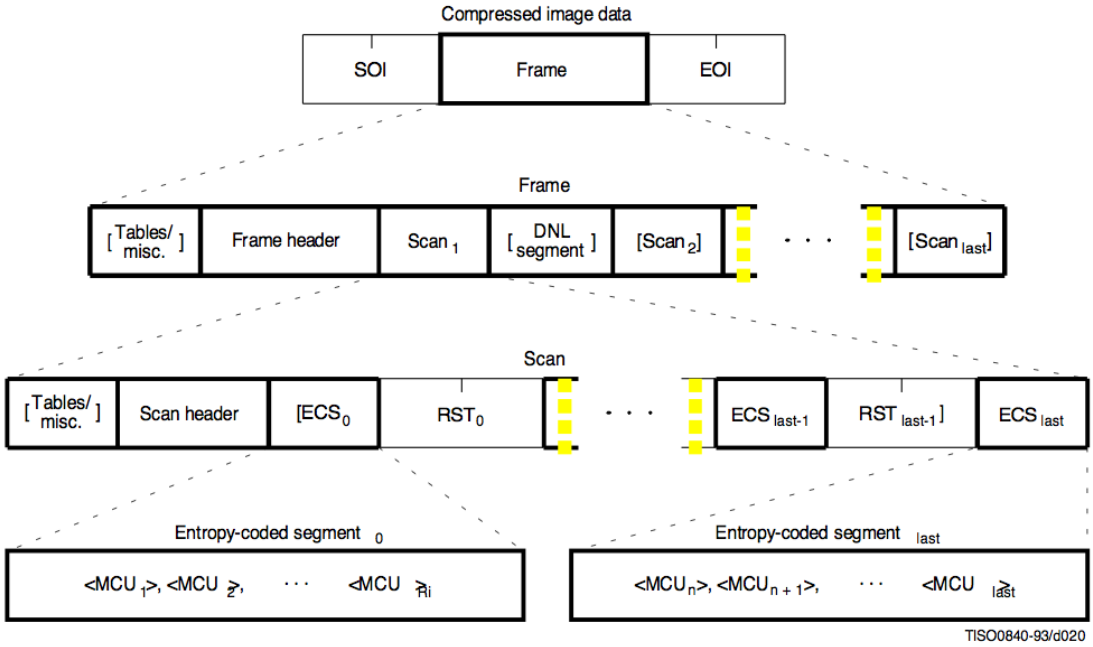


FIGURE *A.6* – *Structure d'un flux JPEG.*

CODE ASSIGNMENT	SYMBOL	DESCRIPTION
FF C4	DHT	Define Huffman Table
FF D8	SOI	Start of Image
FF D9	EOI	End of Image
FF DA	SOS	Start of Scan
FF DB	DQT	Define Quantization Table
FF C0	SOF_0	Start of frame - Baseline DCT
FF Ei	APPi	Reserved for Application Segments
FF FE	COM	Comment

FIGURE *A.7* – *Marqueurs d'un fichier JPEG*

Le fichier débute toujours par le marqueur de début d'image (SOI) et se termine par le marqueur de fin d'image (EOI). Les métadonnées et/ou commentaires sont placés immédiatement après le marqueur SOI.

Les détails complémentaires de la structure de l'en-tête apparaissent dans l'Annexe A.

Parmi les différents types de format JPEG existants, on peut citer les formats JFIF (JPEG File Interchange Format) et EXIF (EXchangeable Image file Format for digital still camera).

Pour connaître le format utilisé, on peut se référer au marqueur correspondant aux applications. Les informations spécifiques au format peuvent être organisées de deux manières : selon Intel (II) ou selon Motorola (MM).

Format JFIF Le format JFIF utilise un sous-échantillonnage 4 : 2 : 2 après une conversion des couleurs. Le début du flux JPEG apparaît comme suit :

```
SOI - APP0 - Taille de l'entête - Version JFIF - Tables de
quantification - SOF - Table de Huffman - SOS - etc.
```

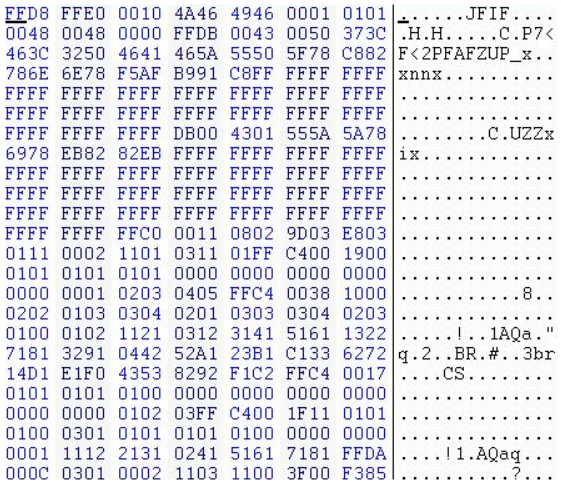


FIGURE A.8 – *Début d'un fichier au format JFIF en hexadécimal.*

Format SPIFF Ce format est repéré par le marqueur APP8, et l'en-tête a une taille fixe de 32 octets :

```
SOI - APP8 - Version - Nombre de composantes - Hauteur - Largeur
 - Espace colorimétrique - Précision - Mode de compression -
Unité de résolution - Ratio vertical - Ratio horizontal
```

Format EXIF Ce format a été proposé par un consortium de fabricants d'appareils photo numériques japonais. Il s'agit d'une évolution du format JFIF. Ce format est repéré par le marqueur APP1 et contient des informations précises sur :

- la structure de l'image
- les caractéristiques de l'image (colorimétrie, codage des couleurs, etc.)
- le fichier (titre, date/heure, identifiant de l'appareil, etc.)
- les données relatives au GPS (27 étiquettes)
- la prise de vue (27 étiquettes dont l'ouverture, le diaphragme, le flash, la distance, etc.).

```
FFD8 FFE0 0010 4A46 4946 0001 0101|......JFIF....
0048 0048 0000 FFE1 3527 4578 6966|.H.H....5'Exif
0000 4D4D 002A 0000 0008 000B 010F|..MM.*........
0002 0000 0012 0000 0092 0110 0002|..............
0000 000A 0000 00A4 0112 0003 0000|..............
0001 0001 0000 011A 0005 0000 0001|..............
0000 00AE 011B 0005 0000 0001 0000|..............
00B6 0128 0003 0000 0001 0002 0000|...(..........
0131 0002 0000 000B 0000 00BE 0132|.1...........2
0002 0000 0014 0000 00CA 0213 0003|..............
0000 0001 0002 0000 8769 0004 0000|.........i....
0001 0000 00DE 8825 0004 0000 0001|.......%......
0000 275E 0000 2770 4E49 4B4F 4E20|..'^..'pNIKON 
434F 5250 4F52 4154 494F 4E00 4E49|CORPORATION.NI
4B4F 4E20 4439 3000 0000 0048 0000|KON D90....H..
0001 0000 0048 0000 0001 4749 4D50|.....H....GIMP
2032 2E36 2E34 0000 3230 3039 3A30| 2.6.4..2009:0
343A 3330 2031 353A 3538 3A31 3500|4:30 15:58:15.
0028 829A 0005 0000 0001 0000 02C4|.(............
829D 0005 0000 0001 0000 02CC 8822|............."
0003 0000 0001 0002 0000 8827 0003|...........'..
0000 0001 00C8 0000 9000 0007 0000|..............
0004 3032 3231 9003 0002 0000 0014|..0221........
0000 02D4 9004 0002 0000 0014 0000|..............
```

FIGURE *A.9* – *Début d'un fichier au format EXIF en hexadécimal.*

Décompression JPEG

La décompression peut être visualisée sur la figure 3.3 p. 55 et se déroule de la manière suivante :

Décodage

Les coefficients DC et AC sont décodés à l'aide des tables situées dans l'en-tête du fichier JPEG.

Déquantification

L'étape suivante est la déquantification. Pour effectuer la déquantification, on effectue simplement une multiplication de la valeur du coefficient DCT, noté $S_{Q_{ij}}$, par le pas de quantification Q_{ij} correspondant :

$$\mathrm{DCT}(i,j) = S_{Q_{ij}} \times Q_{ij}.$$

où Q est la table de quantification.

DCT inverse

La DCT inverse est ensuite appliquée aux coefficients déquantifiés ainsi obtenus :

Formule de la DCT inverse (pour des blocs de $N \times N$) :

$$\mathrm{pixel}(x,y) = \frac{2}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C_i C_j \,\mathrm{DCT}(i,j) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

Formule de la DCT inverse (pour des blocs de 8×8) :

$$\mathrm{pixel}(x,y) = \frac{1}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} C_i C_j \,\mathrm{DCT}(i,j) \cos\left[\frac{(2x+1)i\pi}{16}\right] \cos\left[\frac{(2y+1)j\pi}{16}\right]$$

On obtient donc des valeurs qui correspondent à l'intensité de chaque pixel pour chaque composante Y, Cb et Cr.

Sur-échantillonnage Si un sous-échantillonnage a été appliqué lors de la compression, un sur-échantillonnage est effectué. Selon le décodeur utilisé, il peut s'agir d'une interpolation, d'une copie des pixels, etc. Rien n'est précisé dans la norme, tout dépend du décodeur.

Changement d'espace colorimétrique Enfin, on peut éventuellement effectuer un changement d'espace colorimétrique si l'on souhaite. Le changement d'espace colorimétrique pour passer de l'espace YCbCr à l'espace RGB se fait de la manière suivante :

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.371 \\ 1 & -0.336 & -0.698 \\ 1 & 1.732 & 0 \end{pmatrix} \times \left[\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} + \begin{pmatrix} 128 \\ 0 \\ 0 \end{pmatrix} \right]$$

Exemple

L'exemple que l'on prend est celui d'une image JPEG, l'espace colorimétrique est correct, il n'y a donc pas de changement d'espace à effectuer. L'échantillonnage étant une étape facultative, nous n'effectuerons pas cette étape dans notre exemple.

Tous les calculs suivants ont été faits sous Matlab. Tout d'abord, l'image choisie (NY.jpg) est lue, et nous récupérons sa taille, ainsi que le nombre de composantes :

```
>> NY = imread('NY.jpg');
>> taille = size(NY)
taille =
         669        1000           3
```

Nous choisissons un bloc, et récupérons les valeurs des pixels correspondants dans 3 tableaux (un par composante).

```
>> Y = NY(321:328,481:488,1)
>> Cb = NY(321:328,481:488,2)
>> Cr = NY(321:328,481:488,3)
```

$$\mathrm{Y} = \begin{bmatrix} 60 & 64 & 72 & 77 & 80 & 125 & 132 & 73 \\ 70 & 70 & 72 & 76 & 74 & 92 & 107 & 102 \\ 72 & 67 & 74 & 82 & 72 & 75 & 80 & 77 \\ 77 & 79 & 69 & 65 & 71 & 67 & 53 & 49 \\ 67 & 72 & 65 & 62 & 70 & 65 & 58 & 62 \\ 80 & 80 & 80 & 81 & 80 & 81 & 96 & 98 \\ 72 & 80 & 84 & 88 & 87 & 84 & 111 & 121 \\ 76 & 74 & 85 & 99 & 85 & 93 & 162 & 192 \end{bmatrix} \quad \mathrm{Cb} = \begin{bmatrix} 19 & 24 & 30 & 30 & 29 & 79 & 99 & 51 \\ 25 & 28 & 28 & 29 & 23 & 46 & 73 & 79 \\ 25 & 22 & 28 & 35 & 25 & 31 & 46 & 52 \\ 32 & 34 & 25 & 19 & 27 & 27 & 18 & 21 \\ 27 & 30 & 21 & 18 & 28 & 25 & 21 & 26 \\ 41 & 36 & 33 & 34 & 35 & 36 & 47 & 46 \\ 29 & 33 & 30 & 32 & 31 & 28 & 48 & 51 \\ 30 & 20 & 24 & 34 & 21 & 28 & 86 & 110 \end{bmatrix}$$

$$\mathrm{Cr} = \begin{bmatrix} 0 & 0 & 5 & 2 & 0 & 43 & 54 & 2 \\ 4 & 4 & 3 & 1 & 0 & 13 & 36 & 38 \\ 5 & 0 & 4 & 9 & 0 & 4 & 19 & 22 \\ 11 & 11 & 0 & 0 & 2 & 2 & 0 & 0 \\ 1 & 5 & 0 & 0 & 3 & 0 & 0 & 0 \\ 10 & 9 & 7 & 8 & 6 & 5 & 14 & 9 \\ 0 & 5 & 6 & 9 & 4 & 0 & 4 & 2 \\ 0 & 0 & 3 & 14 & 0 & 0 & 36 & 52 \end{bmatrix}$$

Nous devons ensuite centrer les valeurs des pixels en 0 pour qu'elles soient dans l'intervalle $[-128; 127]$:

```
>> Y = double(Y)-128
```

$$\mathrm{Y} = \begin{bmatrix} -68 & -64 & -56 & -51 & -48 & -3 & 4 & -55 \\ -58 & -58 & -56 & -52 & -54 & -36 & -21 & -26 \\ -56 & -61 & -54 & -46 & -56 & -53 & -48 & -51 \\ -51 & -49 & -59 & -63 & -57 & -61 & -75 & -79 \\ -61 & -56 & -63 & -66 & -58 & -63 & -70 & -66 \\ -48 & -48 & -48 & -47 & -48 & -47 & -32 & -30 \\ -56 & -48 & -44 & -40 & -41 & -44 & -17 & -7 \\ -52 & -54 & -43 & -29 & -43 & -35 & 34 & 64 \end{bmatrix}$$

Ensuite nous appliquons la transformée en cosinus discrète, ce qui revient à faire l'opération suivante :

$$\text{Transformée } = D \times A \times D'$$

où D est la matrice (définie ci-dessous) de la DCT pour une matrice carrée, et A la matrice des coefficients à transformer (Y dans notre exemple).

$$D = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{bmatrix}$$

```
>> DCTY = double(D)*double(Y)*double(D')
```

$$\mathrm{DCTY} = \begin{bmatrix} -365.9715 & -73.5677 & 23.0691 & -9.0896 & -8.8773 & 16.1137 & -9.9627 & 0.2140 \\ -48.3941 & 26.0847 & -34.1421 & 41.7157 & -28.3161 & 7.9366 & 6.5326 & -6.3897 \\ 89.1163 & -90.9363 & 24.2805 & -12.9876 & -11.0849 & 25.1775 & -7.5717 & 0.4044 \\ -12.2248 & 9.6345 & -17.7338 & 23.2652 & -19.9399 & -0.8515 & 0.2871 & 0.3848 \\ -8.1271 & -5.6262 & -0.5461 & 9.9169 & -8.6273 & 0.3061 & -0.9915 & -0.3486 \\ -8.2099 & 20.1232 & -19.9315 & 13.4733 & -9.8516 & 0.2455 & -0.2426 & 0.1494 \\ 14.9039 & -10.3698 & 0.6766 & -0.3689 & -0.9554 & 11.5462 & -0.2856 & -0.7090 \\ -12.9014 & -0.5641 & -9.5614 & -0.4525 & -0.1619 & 0.7078 & 0.4123 & -0.0972 \end{bmatrix}$$

L'étape suivante est l'étape de quantification. Nous allons utiliser les tables de quantification suggérées par la norme, à savoir les tables Quant_Luminance et Quant_Chrominance.

$$\text{Quant\_Luminance} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

$$\text{Quant\_Chrominance} = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

La quantification se fait de la manière suivante : chaque coefficient du bloc va être divisé par le pas de quantification correspondant puis arrondi à l'entier le plus proche : » `QY = round(DCTY ./ Quant_Luminance)`

$$\text{QY} = \begin{bmatrix} -23 & -7 & 2 & -1 & 0 & 0 & 0 & 0 \\ -4 & 2 & -2 & 2 & -1 & 0 & 0 & 0 \\ 6 & -7 & 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Supposons que le coefficient DC du bloc précédent soit -20, on codera le coefficient DC de notre bloc(-23) par -3.
Ensuite, il faut effectuer une lecture zigzag sur les coefficients et appliquer un codage RLE sur la séquence lue.

```
Lecture(QY) = -3 -7 -4 6 2 2 -1 -2 -7 -1 0 1 2 2 0 0 -1 -1 -1 0
0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
```

```
Codage_RLE(QY) = DC :(2,-3) AC :(0,3)(-7) (0,3)(-4) (0,3)(6)
(0,2)(2) (0,2)(2) (0,1)(-1) (0,2)(-2) (0,3)(-7) (0,1)(-1)
(1,1)(1) (0,2)(2) (0,2)(2) (2,1)(-1) (0,1)(-1) (0,1)(-1)
(3,1)(1) (1,1)(1) (0,0)
```

Lors de la décompression, on déquantifie les valeurs obtenues :

```
>> Dequant_QY = QY.*Quant_Luminance
```

$$\texttt{Dequant\_QY} = \begin{bmatrix} -368 & -77 & 20 & -16 & 0 & 0 & 0 & 0 \\ -48 & 24 & -28 & 38 & -26 & 0 & 0 & 0 \\ 84 & -91 & 32 & -24 & 0 & 0 & 0 & 0 \\ -14 & 17 & -22 & 29 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 35 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```
>> YY=round(double(D')*double(Dequant_Y)*double(D))
```

$$\mathrm{YY} = \begin{bmatrix} -62 & -57 & -56 & -55 & -39 & -20 & -20 & -33 \\ -74 & -65 & -60 & -58 & -46 & -28 & -20 & -24 \\ -61 & -54 & -52 & -58 & -59 & -52 & -48 & -49 \\ -49 & -49 & -51 & -57 & -61 & -65 & -71 & -77 \\ -58 & -62 & -62 & -56 & -51 & -54 & -63 & -69 \\ -52 & -57 & -56 & -48 & -44 & -47 & -47 & -43 \\ -47 & -46 & -40 & -37 & -43 & -43 & -21 & 8 \\ -64 & -54 & -39 & -34 & -42 & -31 & 18 & 71 \end{bmatrix}$$

Enfin, on renvoie les valeurs dans l'intervalle $[0;255]$:

```
>> YY = YY+128
```

$$\mathrm{YY} = \begin{bmatrix} 66 & 71 & 72 & 73 & 89 & 108 & 108 & 95 \\ 54 & 63 & 68 & 70 & 82 & 100 & 108 & 104 \\ 67 & 74 & 76 & 70 & 69 & 76 & 80 & 79 \\ 79 & 79 & 77 & 71 & 67 & 63 & 57 & 51 \\ 70 & 66 & 66 & 72 & 77 & 74 & 65 & 59 \\ 76 & 71 & 72 & 80 & 84 & 81 & 81 & 85 \\ 81 & 82 & 88 & 91 & 85 & 85 & 107 & 136 \\ 64 & 74 & 89 & 94 & 86 & 97 & 146 & 199 \end{bmatrix}$$

YY correspond donc aux nouvelles valeurs des pixels de l'image compressée.

Annexe B

Stéganographie littéraire

Voici les lettres de la correspondance supposée entre George Sand et Alfred de Musset. Pour faire apparaître le message de la première lettre, il suffit de lire une ligne sur deux. Afin de découvrir les réponses, il suffit de lire les premiers mots de chaque vers.

Lettre de George Sand à Alfred de Musset

Je suis très émue de vous dire que j'ai
bien compris l'autre soir que vous aviez
toujours une envie folle de me faire
danser. Je garde le souvenir de votre
baiser et je voudrais bien que ce soit
là une preuve que je puisse être aimée
par vous. Je suis prête à vous montrer mon
affection toute désintéressée et sans cal-
cul, et si vous voulez me voir aussi
vous dévoiler sans artifice mon âme
toute nue, venez me faire une visite.
Nous causerons en amis, franchement.
Je vous prouverai que je suis la femme
sincère, capable de vous offrir l'affection
la plus profonde comme la plus étroite
en amitié, en un mot la meilleure preuve
dont vous puissiez rêver, puisque votre
âme est libre. Pensez que la solitude où j'ha-
bite est bien longue, bien dure et souvent
difficile. Ainsi en y songeant j'ai l'âme
grosse. Accourrez donc vite et venez me la
faire oublier par l'amour où je veux me
mettre.

Réponse d'Alfred de Musset

Quand je mets à vos pieds un éternel hommage
Voulez-vous qu'un instant je change de visage ?
Vous avez capturé les sentiments d'un cœur
Que pour vous adorer forma le Créateur.
Je vous chéris, amour, et ma plume en délire
Couche sur le papier ce que je n'ose dire.
Avec soin, de mes vers lisez les premiers mots
Vous saurez quel remède apporter à mes maux.

Réponse de George Sand

Cette insigne faveur que votre cour réclame
Nuit à ma renommée et répugne mon âme.

Liste des illustrations

Liste des tableaux

Bibliographie

[1] N. Ahmed, T. Natarajan, and K. Rao. Discrete cosine transform. *Computers, IEEE Transactions on*, C-23(1) :90 –93, jan. 1974.

[2] P. Bas, T. Filler, and T. Pevný. Break Our Steganographic System — the ins and outs of organizing BOSS. In T. Filler, editor, *Information Hiding, 13th International Workshop*, Lecture Notes in Computer Science, Prague, Czech Republic, May 18–20, 2011. Springer-Verlag, New York.

[3] R. Böhme. *Advanced Statistical Steganalysis*. Information Security and Cryptography. Springer Berlin / Heidelberg, 1st edition, 2010.

[4] A. A. Borovkov. *Mathematical Statistics*. Gordon and Breach Sciences Publishers, Amsterdam, 1998.

[5] V. Britanak, K. R. Rao, and P. C. Yip. *Discrete Cosine and Sine Transforms : General Properties, Fast Algorithms and Integer Approximations*. Elsevier, San Diego, CA, 2006.

[6] C. Chen and Y. Shi. JPEG image steganalysis utilizing both intrablock and interblock correlations. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 3029 –3032, may 2008.

[7] R. Cogranne. *Détection statistique d'informations cachées dans une image naturelle à partir d'un modèle physique*. PhD thesis, Université de Technologie de Troyes, December 2011.

[8] R. Cogranne, C. Zitzmann, L. Fillatre, I. Nikiforov, F. Retraint, and P. Cornu. Reliable detection of hidden information based on a non-linear local model. In *Statistical Signal Processing Workshop (SSP), 2011 IEEE*, pages 493 –496, june 2011.

[9] R. Cogranne, C. Zitzmann, L. Fillatre, F. Retraint, I. Nikiforov, and P. Cornu. A cover image model for reliable steganalysis. In *Information Hiding*, volume 6958 of *Lecture Notes in Computer Science*, pages 178–192, 2011.

[10] R. Cogranne, C. Zitzmann, L. Fillatre, F. Retraint, I. Nikiforov, and P. Cornu. Détection quasi-optimale d'informations cachées basée sur un modèle local non-linéaire. In *Actes du XXIIIième colloque GRETSI*, page 4, Bordeaux, France, Sept. 2011.

[11] R. Cogranne, C. Zitzmann, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu. Statistical detection of LSB matching in the presence of nuisance parameters. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 912–915. IEEE, 2012.

[12] R. Cogranne, C. Zitzmann, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu. Statistical detection of LSB matching using hypothesis testing theory. In *Information Hiding*, pages 46–62. Springer, 2013.

[13] I. Cox, M. Miller, J. Bloom, and M. Miller. *Digital watermarking*. Morgan Kaufmann, 2001.

[14] O. Dabeer, K. Sullivan, U. Madhow, S. Chandrasekaran, and B. Manjunath. Detection of Hiding in the Least Significant Bit. *Signal Processing, IEEE Transactions on*, 52(10) :3046 – 3058, oct. 2004.

[15] S. Dumitrescu, X. Wu, and Z. Wang. Detection of LSB steganography via sample pair analysis. In *Revised Papers from the 5th International Workshop on Information Hiding*, IH '02, pages 355–372, London, UK, UK, 2003. Springer-Verlag.

[16] J. E. Eggerton and M. D. Srinath. Statistical distributions of image DCT coefficients. *Comput. Electr. Eng.*, 12(3-4) :137–145, Jan. 1986.

[17] H. Farid. Detecting Steganographic Messages in Digital Images. Technical Report TR2001-412, Department of Computer Science, Dartmouth College, 2001.

[18] T. Ferguson. *Mathematical Statistics : A Decision Theoretic Approach*. Academic Press, New York and London, 1967.

[19] L. Fillatre. Adaptive steganalysis of least significant bit replacement in grayscale natural images. *Signal Processing, IEEE Transactions on*, 60(2) :556 –569, feb. 2012.

[20] L. Fillatre and I. Nikiforov. A statistical detection of an anomaly from a few noisy tomographic projections. *Journal of Applied Signal Processing, Special issue on advances in intelligent vision systems : methods and applications-Part II*, 2005(14) :2215–2228, January 2005.

[21] L. Fillatre and I. Nikiforov. Non-Bayesian Detection and Detectability of Anomalies From a Few Noisy Tomographic Projections. *IEEE Trans. Signal Processing*, 55(2) :401–413, February 2007.

[22] L. Fillatre, I. Nikiforov, and F. Retraint. ε-Optimal Non-Bayesian Anomaly Detection for Parametric Tomography. *IEEE Trans. on Image Processing*, 17(11) :1985–1999, 2008.

[23] T. Filler, A. D. Ker, and J. J. Fridrich. The square root law of steganographic capacity for markov covers. In E. J. Delp, J. Dittmann, N. D. Memon, and P. W. Wong, editors, *Media Forensics and Security I, part of the I&ST-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 19, 2009, Proceedings*, volume 7254 of *SPIE Proceedings*, 2009.

[24] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical Poissonian-Gaussian Noise Modeling and Fitting for Single-Image Raw-Data. *IEEE Transactions on Image Processing*, 17 :1737–1754, Oct. 2008.

[25] M. Fouladirad and I. Nikiforov. Optimal statistical fault detection with nuisance parameters. *Automatica*, 41(7) :1157–1171, 2005.

[26] J. Fridrich. *Steganography in Digital Media : Principles, Algorithms, and Applications*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[27] J. Fridrich and M. Goljan. On estimation of secret message length in LSB steganography in spatial domain. In *SPIE proceedings series*, pages 23–34. Society of Photo-Optical Instrumentation Engineers, 2004.

[28] J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of JPEG images : Breaking the F5 algorithm. In F. Petitcolas, editor, *Information Hiding*, volume 2578 of *Lecture Notes in Computer Science*, pages 310–323. Springer Berlin / Heidelberg, 2003.

[29] J. Fridrich, M. Goljan, D. Hogea, and D. Soukal. Quantitative steganalysis of digital images : estimating the secret message length. *Multimedia Systems*, 9 :288–302, 2003. 10.1007/s00530-003-0100-9.

[30] J. Fridrich, M. Goljan, and D. Soukal. Higher-order statistical steganalysis of palette images. *Proceedings*, 5020(1) :178–190, 2003.

[31] J. Fridrich and J. Kodovskỳ. Steganalysis of LSB replacement using parity-aware features. In *Information Hiding*, pages 31–45. Springer, 2013.

[32] J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable JPEG steganography : dead ends challenges, and opportunities. In *Proceedings of the 9th workshop on Multimedia & Security*, MM&Sec '07, pages 3–14, New York, NY, USA, 2007. ACM.

[33] F. Gustafsson and R. Karlsson. Statistical results for system identification based on quantized observations. *Automatica*, 45(12) :2794–2801, Dec. 2009.

[34] R. W. Hamming. Error detecting and error correcting codes. *Syst. Tech. J.*, 29 :147–160, 1950.

[35] G. Healey and R. Kondepudy. Radiometric CCD camera calibration and noise estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(3) :267–276, 1994.

[36] S. Hetzl. Implémentation de StegHide. http ://steghide.sourceforge.net, 2003.

[37] S. Hetzl and P. Mutzel. A graph–theoretic approach to steganography. In *Communications and Multimedia Security*, pages 119–128. Springer, 2005.

[38] G. Holst and T. Lomheim. *CMOS/CCD Sensors and Camera Systems*. SPIE Press Book, 1st edition, June 2007.

[39] ISO/IEC 10918-1. Information technology – digital compression and coding of continuous-tone still images : Requirements and guidelines, 1994.

[40] ISO/IEC 10918-2. Information technology – digital compression and coding of continuous-tone still images : Compliance testing, 1995.

[41] ITU-T Rec. T.81. Information technology – digital compression and coding of continuous-tone still images : Requirements and guidelines, September 1992.

[42] H. Junhui, T. Shaohua, and L. Bin. Model-based steganalytic method towards color jpeg images. *Journal of Computational Information Systems*, 3(6) :2293–2302, 2007.

[43] A. D. Ker. Locating steganographic payload via WS residuals. In *Proceedings of the 10th ACM workshop on Multimedia & security*, MM&Sec '08, pages 27–32, New York, NY, USA, 2008. ACM.

[44] A. D. Ker and R. Böhme. Revisiting weighted stego-image steganalysis. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6819 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Mar. 2008.

[45] A. D. Ker, T. Pevný, J. Kodovský, and J. Fridrich. The square root law of steganographic capacity. In *Proceedings of the 10th ACM workshop on Multimedia & security*, MM&Sec '08, pages 107–116, New York, NY, USA, 2008. ACM.

[46] J. Kodovský. *Steganalysis of Digital Images Using Rich Image Representations and Ensemble Classifiers*. PhD thesis, Binghamton University, 2012.

[47] J. Kodovský and J. Fridrich. Quantitative steganalysis of lsb embedding in jpeg domain. In *Proceedings of the 12th ACM workshop on Multimedia and security*, MM&amp ;#38 ;Sec '10, pages 187–198, New York, NY, USA, 2010. ACM.

[48] J. Kodovský and J. Fridrich. Quantitative structural steganalysis of jsteg. *Information Forensics and Security, IEEE Transactions on*, 5(4) :681 –693, dec. 2010.

[49] E. Y. Lam and J. W. Goodman. A mathematical analysis of the DCT coefficient distributions for images. *Image Processing, IEEE Transactions on*, 9(10) :1661 –1666, oct 2000.

[50] A. Latham. Implémentation de JPHS. http ://linux01.gwdg.de/~alatham/stego.html, 1998.

[51] L. Le Cam. *Asymptotic Methods in Statistical Decision Theory*. Series in Statistics, Springer, New York, 1986.

[52] L. Le Cam and G. L. Yang. *Asymptotics in Statistics*. Springer-Verlag, New York, Berlin, Heidelber, 1990.

[53] K. Lee, A. Westfeld, and S. Lee. Generalised category attack- improving histogram-based attack on JPEG LSB embedding. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding*, volume 4567 of *Lecture Notes in Computer Science*, pages 378–391. Springer Berlin / Heidelberg, 2007.

[54] K. Lee, A. Westfeld, S. Lee, and T. U. Dresden. Category attack for LSB steganalysis of JPEG images. In *Digital Watermarking (5th International Workshop) IWDW 2006 Jeju Island, Korea, November 8-10, 2006, Revised Papers. Volume 4283 of LNCS*, pages 35–48. Springer-Verlag, 2006.

[55] E. L. Lehmann. *Testing statistical hypotheses*. Wiley, New York, 1986.

[56] Y. Miche, P. Bas, A. Lendasse, C. Utten, and O. Simula. Avantages de la sélection de caractéristiques pour la stéganalyse. In *Actes de Colloques du GRETSI 2007*. GRETSI, Groupe d'Études du Traitement du Signal et des Images, 2007.

[57] F. Muller. Distribution shape of two-dimensional DCT coefficients of natural images. *Electronics Letters*, 29(22) :1935 –1936, oct. 1993.

[58] S. Nadarajah. Gaussian DCT Coefficient Models. *Acta Applicandae Mathematicae*, 106 :455–472, 2009.

[59] S. Nadarajah and S. Kotz. On the dct coefficient distributions. *Signal Processing Letters, IEEE*, 13(10) :601 –603, oct. 2006.

[60] T. Pevný. *Kernel Methods in Steganalysis*. PhD thesis, Binghamton University, 2008.

[61] T. Pevny, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. *Information Forensics and Security, IEEE Transactions on*, 5(2) :215 –224, june 2010.

[62] T. Pevny and J. Fridrich. Determining the stego algorithm for JPEG images. *Information Security, IEEE Proceedings*, 153(3) :77 –86, sept. 2006.

[63] J. Price and M. Rabbani. Biased reconstruction for JPEG decoding. *Signal Processing Letters, IEEE*, 6(12) :297 –299, dec 1999.

[64] N. Provos. Defending against statistical steganography. In *Proc 10th USENIX Security Symposium*, 2001.

[65] N. Provos and P. Honeyman. Detecting steganographic content on the internet. Technical report, Center for Information Technology Integration, University of Michigan, November 2001.

[66] N. Provos and P. Honeyman. Hide and seek : an introduction to steganography. *Security Privacy, IEEE*, 1(3) :32 – 44, may-june 2003.

[67] C. R. Rao. *Linear statistical Inference and Its Applications*. Wiley, 1965.

[68] R. Reininger and J. Gibson. Distributions of the two-dimensional DCT coefficients for images. *Communications, IEEE Transactions on*, 31(6) :835 – 839, jun 1983.

[69] G. G. Roussas. *Contiguity of Probability Measures, Some Applications in Statistics*. Cambridge University Press, Mass., 1972.

[70] P. Sallee. Implémentation de MBS. http : //www.philsallee.com/mbsteg, 2004.

[71] P. Sallee. Model-based steganography. In *Digital Watermarking*, volume 2939 of *Lecture Notes in Computer Science*, pages 254–260, 2004.

[72] L. Scharf and B. Friedlander. Matched subspace detectors. *IEEE Trans. Signal Processing*, 42(8) :2146–2157, 1994.

[73] B. Schölkopf and A. J. Smola. *Learning With Kernels : Support Vector Machines, Regularization, Optimization and Beyond*. Adaptative computation and machine learning series. "The" MIT Press, 2002.

[74] W. F. Sheppard. On the calculation of the most probable values of frequency-constants, for data arranged according to equidistant division of a scale. *Proceedings of the London Mathematical Society*, s1-29(1) :353–380, 1897.

[75] A. N. Shiryaev. *Probability*. 2nd edn. Springer, New York, 1996.

[76] G. J. Simmons. The Prisoners' Problem and the Subliminal Channel. In *Advances in Cryptology - CRYPTO '83*, pages 51–67, New York, 1984. Lecture Notes in Computer Science.

[77] S. R. Smoot and L. A. Rowe. DCT coefficient distributions. In *Proceedings SPIE 2657*. Stephen R. Smoot and Lawrence A. Rowe, "DCT coefficient distributions", Proc. SPIE 2657, 403 (1996) ; doi :10.1117/12.238737, 1996.

[78] K. Sullivan, O. Dabeer, U. Madhow, B. Manjunath, and S. Chandrasekaran. Llrt based detection of lsb hiding. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I – 497–500 vol.1, sept. 2003.

[79] D. Upham. Jpeg-Jsteg, modification of the independent JPEG's group's JPEG software (release 4) for 1-bit steganography in JFIF output files, 1992-1997.

[80] V. Vapnik. *The nature of statistical learning theory*. springer, 1999.

[81] A. Wald. Tests of statistical hypotheses concerning several parameters when the number of observations is large. *Transactions of the American Mathematical Society*, 54 :426–482, 1943.

[82] A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2) :pp. 117–186, 1945.

[83] G. Wallace. The jpeg still picture compression standard. *Consumer Electronics, IEEE Transactions on*, 38(1) :xviii –xxxiv, feb 1992.

[84] Y. Wang and P. Moulin. Optimized feature extraction for learning-based image steganalysis. *Information Forensics and Security, IEEE Transactions on*, 2(1) :31 –45, march 2007.

[85] A. Westfeld. F5-a steganographic algorithm. In *Proceedings of the 4th International Workshop on Information Hiding*, IHW '01, pages 289–302, London, UK, UK, 2001. Springer-Verlag.

[86] A. Westfeld. Generic adoption of spatial steganalysis to transformed domain. In *Information Hiding*, volume 5284 of *Lecture Notes in Computer Science*, pages 161–177, 2008.

[87] A. Westfeld and A. Pfitzmann. Attacks on steganographic systems. In *Proceedings of the Third International Workshop on Information Hiding*, IH '99, pages 61–76, London, UK, 2000. Springer-Verlag.

[88] B. Widrow, I. Kollar, and M.-C. Liu. Statistical theory of quantization. *Instrumentation and Measurement, IEEE Transactions on*, 45(2) :353 –361, apr 1996.

[89] M. Wu, Z. Zhu, and S. Jin. Detection of jsteg hiding using image statistical model. *Chinese Journal of Electronics*, 15(1) :165–168, 2006.

[90] T. Zhang and X. Ping. A fast and effective steganalytic technique against JSteg-like algorithmslike algorithms. In *Proceedings of the 2003 ACM symposium on Applied computing*, SAC '03, pages 307–311, New York, NY, USA, 2003. ACM.

[91] C. Zitzmann, R. Cogranne, L. Fillatre, I. Nikiforov, F. Retraint, and P. Cornu. Détection optimale à base de distribution laplacienne quantifiée. In *Proceedings of the 23rd Symposium on Signal and Image Processing (GRETSI)*, Bordeaux, France, September 2011.

[92] C. Zitzmann, R. Cogranne, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu. Hypothesis testing by using quantized observations. In *Statistical Signal Processing Workshop (SSP), 2011 IEEE*, pages 501 –504, june 2011.

[93] C. Zitzmann, R. Cogranne, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu. Statistical decision methods in hidden information detection. In *Information Hiding*, volume 6958 of *Lecture Notes in Computer Science*, pages 163–177, 2011.

Liste des publications

Ce manuscrit contient notamment des idées et des résultats qui ont été présentés lors de différentes conférences.

Présentations lors de conférences internationales (avec actes et comité de lecture) :

- R. Cogranne, **C. Zitzmann**, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu, *Statistical detection of LSB matching using hypothesis testing theory*, in Information Hiding, Springer, 2013, pp. 46–62.

- R. Cogranne, **C. Zitzmann**, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu, *Statistical detection of LSB matching in the presence of nuisance parameters*, in Statistical Signal Processing Workshop (SSP), IEEE, 2012, pp. 912–915.

- **C. Zitzmann**, R. Cogranne, L. Fillatre, I. Nikiforov, F. Retraint, and P. Cornu, *Hidden information detection based on quantized laplacian distribution*, ICASSP 2012, 25-31 March 2012.

- R. Cogranne, **C. Zitzmann**, L. Fillatre, F. Retraint, I. Nikiforov, and P. Cornu, *Statistical decision by using quantized observations*, in Information Theory Proceedings (ISIT), IEEE International Symposium on, August 2011, pp. 1210–1214.

- **C. Zitzmann**, R. Cogranne, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu, *Hypothesis testing by using quantized observations*, in IEEE Statistical Signal Processing Workshop (SSP), June 2011, pp. 501–504.

- R. Cogranne, **C. Zitzmann**, L. Fillatre, I. Nikiforov, F. Retraint, and P. Cornu, emphReliable detection of hidden information based on a non-linear local model, in IEEE Statistical Signal Processing Workshop (SSP), June 2011, pp. 493–496.

- **C. Zitzmann**, R. Cogranne, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu, *Statistical decision methods in hidden information detection*, Information Hiding Conference, 18-20 May 2011.

- R. Cogranne, **C. Zitzmann**, L. Fillatre, F. Retraint, I. Nikiforov, and P. Cornu, *A cover image model for reliable steganalysis*, Information Hiding Conference, 18-20 May 2011.

Présentations lors de conférences nationales (avec actes et comité de lecture) :

- **C. Zitzmann**, R. Cogranne, F. Retraint, I. Nikiforov, L. Fillatre et P. Cornu, *Détection Optimale à base de laplacienne quantifiée*, XXIIIième colloque GRETSI, 5-8 septembre

2011, Bordeaux.

- R. Cogranne, **C. Zitzmann**, L. Fillatre, F. Retraint, I. Nikiforov et P. Cornu, *Détection Optimale d'information cachées indépendante du contenu des images*, XXIIIième colloque GRETSI, 5-8 septembre 2011, Bordeaux.

- R. Cogranne, L. Fillatre, F. Retraint et **C. Zitzmann**, *Détection Optimale d'information cachées indépendante du contenu des images*, JCS 11, janvier 2011, Saint-Cyr Coëtquidan.

- R. Cogranne, L. Fillatre, F. Retraint et **C. Zitzmann**, *Modélisation des images naturelles pour la criminalistique numérique*, workshop 3SGS, Reims, septembre 2010 (Prix de la meilleure présentation).

Cathel ZITZMANN

Doctorat : Optimisation et Sûreté des Systèmes

Année 2013

Détection statistique d'information cachée dans des images naturelles

La nécessité de communiquer de façon sécurisée n'est pas chose nouvelle : depuis l'antiquité des méthodes existent afin de dissimuler une communication. La cryptographie a permis de rendre un message inintelligible en le chiffrant, la stéganographie quant à elle permet de dissimuler le fait même qu'un message est échangé.
Cette thèse s'inscrit dans le cadre du projet "Recherche d'Informations Cachées" financé par l'Agence Nationale de la Recherche. L'Université de Technologie de Troyes a travaillé sur la modélisation mathématique d'une image naturelle et à la mise en place de détecteurs d'informations cachées dans les images.
Ce mémoire propose d'étudier la stéganalyse dans les images naturelles du point de vue de la décision statistique paramétrique. Pour les images JPEG, un détecteur basé sur la modélisation des coefficients DCT quantifiés est proposé et les calculs des probabilités du détecteur sont établis théoriquement. De plus, une étude du nombre moyen d'effondrements apparaissant lors de l'insertion avec les algorithmes F3 et F4 est proposée. Enfin, dans le cadre des images non compressées, les tests proposés sont optimaux sous certaines contraintes, une des difficultés surmontées étant le caractère quantifié des données.

Mots clés : test d'hypothèses (statistique) – cryptographie - JPEG (norme de codage d'images) – numérisation.

Statistical Detection of Hidden Information in Natural Images

The need of secure communication is not something new: from ancient, methods exist to conceal communication. Cryptography helped make unintelligible message using encryption, steganography can hide the fact that a message is exchanged.
This thesis is part of the project "Hidden Information Research" funded by the National Research Agency. Troyes University of Technology worked on the mathematical modeling of a natural image and built detectors of hidden information in digital pictures.
This thesis proposes to study the steganalysis in natural images in terms of parametric statistical decision. For JPEG images, a detector based on the modeling of quantized DCT coefficients is proposed and calculations of probabilities of the detector are established theoretically. In addition, a study of the number of shrinkage occurring during embedding by F3 and F4 algorithms is proposed. Finally, for the uncompressed images, the proposed tests are optimal under certain constraints, a difficulty overcome is the data quantization.

Keywords: statistical hypothesis testing – cryptography - JPEG (image coding standard) – digitization.P"°

Thèse réalisée en partenariat entre :

Agence Nationale de la Recherche ANR

utt université de technologie Troyes

Ecole Doctorale "Sciences et Technologies"