

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la recherche Scientifique

**UNIVERSITE SAAD DAHLEB DE BLIDA**

**Faculté des Sciences de l'ingénieur**

**Département d'Electronique**

## **MEMOIRE DE MAGISTER**

Spécialité : Signaux et systèmes

**ETUDE ET DÉVELOPPEMENT DES TECHNIQUES D'INSERTION  
DE DONNÉES CACHÉES DANS LE FLUX JPEG2000 :  
APPLICATION AUX IMAGES MÉDICALES**

**Par**

**Fadila NAIT ABDESSELAM**

Devant le jury composé de :

Z. Benslama	Maître de conférences, Université de Blida	Président
N. Benblidia	Maître de conférences, Université de Blida	Examineur
L. Hamami	Professeur, Ecole nationale polytechnique	Examineur
A. Guessoum	Professeur, Université de Blida	Encadreur
Z. Brahim	Directrice des Etudes, MPTIC, Alger	Coencadreur

## RÉSUMÉ

L'insertion de données cachées (IDC) est une technique qui permet d'insérer une capacité très élevée de données dans une image numérique d'une manière imperceptible. En plus de son application initiale qui est la protection des droits d'auteur, l'IDC peut être utilisé dans d'autres domaines et en particulier dans le domaine de l'imagerie médicale.

L'objectif de ce travail est de dresser un état de l'art des différentes techniques d'insertion de données cachées dans les domaines spatial et fréquentiel. Il présente aussi les performances d'une technique d'insertion de données cachées dans le flux JPEG 2000 appliquée aux images médicales et optiques. Nous montrons à travers cette technique que d'autres paramètres inhérents à l'image rentrent en ligne de compte dans le choix de la technique. Une solution globale pour la sécurité des images médicales et les données du patient, a été proposée en combinant deux techniques. D'une part une nouvelle technique d'IDC irréversible, pour assurer une grande capacité d'insertion des données du patient. D'autre part une technique reposant sur l'expansion de différence des pixels adjacents. Elle fournit parmi les meilleures performances en termes de capacité- distorsion présentes dans la littérature, avec le bénéfice d'être complètement réversible pour assurer l'authentification de l'image. Pour augmenter sa capacité tout en évitant le problème de dépassement de capacité caractérisant la méthode originale, nous proposons une nouvelle manière de codage des coefficients d'expansion.

Mots-clés : Insertion de données cachées, JPEG2000, Capacité d'insertion, Réversibilité, Distorsion, Images.

## **ABSTRACT**

Data hiding is a technique that allows a large capacity data embedding with minimum distortion of the digital image. In addition to its initial application which is the copyright, data hiding can be used in other fields, particularly in the medical domain.

The objective of this work is to develop a state of the art of the data hiding techniques in the spatial and frequency domains. It also presents the performance of a data hiding technique in JPEG 2000 codestream applied to optical and medical images. We show through this technique that other parameters related to the image come into account in the choice of the technique. A global solution for the medical images and patient data security was proposed by combining two techniques. On the one hand, a new irreversible IDC technique, to ensure high capacity patient data embedding. On the other hand, a technique based on the expansion difference of adjacent pixels. It provides among the best performance in terms of capacity-distortion present in the literature, with the benefit of being completely reversible to provide image authentication. To increase capacity while avoiding the problem of the overflow characterizing the original method, we propose a new way of coding the expansion coefficients.

Key words: Data Hiding, JPEG2000, embedding capacity, Reversibility, Distortion, Images.

## الملخص

إن إدماج البيانات المخفية (IDC) هو التقنية التي تسمح بإدراج نسبة عالية جدا من بيانات في صورة رقمية، و ذلك بصفة غير محسوسة. بالإضافة إلى تطبيقاته الأولية و المتمثلة في حماية حقوق التأليف والنشر، يمكن استخدام إدماج البيانات المخفية في مجالات أخرى، لا سيما في مجال التصوير الطبي. الهدف من هذا العمل هو تطوير الحالة التقنية السابقة في مجال تقنيات إخفاء البيانات في الميادين الفضائية و الترددية، كما يعرض أداء تقنية إدماج البيانات المخفية ضمن التدفق JPEG 2000 المطبق على الصور الطبية والبصرية، كما نبين من خلال هذه التقنية أن معايير أخرى تتعلق بالصورة تؤخذ بعين الاعتبار في اختيار التقنية.

وقد تم اقتراح حل شامل للمحافظة على أمن الصور الطبية وبيانات المرضى، و ذلك من خلال الجمع بين تقنيتان. من جهة، تقنية إدماج بيانات مخفية جديدة غير قابلة للعكس، لضمان إدخال قدرة عالية من بيانات المريض. ومن جهة أخرى تقنية تقوم على توسيع فارق البكسل المجاورة، التي تقدم أداءً أفضل في حالة وجود تحريفات من الجانب الأدبي، مع الاستفادة من كونها قابلة للعكس تماما و هذا لتأمين التوثيق للصورة. ومن أجل زيادة قدراتها، وفي الوقت نفسه تجنب مشكلة (التدفق المفرط) الذي يميز الطريقة الأصلية، نقترح طريقة جديدة لترميز عاملات التوسع .

كلمات البحث: القدرة على إدماج البيانات، تقنية قابلة للعكس، التحريف، الصور،

JPEG 2000

## REMERCIEMENTS

Ce travail a été effectué au sein de l'équipe codage, sécurité et systèmes distribués, au centre de développement des technologies avancées (CDTA).

Je tiens en premier lieu à remercier A. Guessoum, professeur à l'université de Blida d'avoir bien voulu encadrer ce projet de Magister. Je vous remercie vivement Monsieur pour tous les conseils et toute l'aide que vous n'avez pas manqué à me prodiguer durant cet encadrement.

Toute ma gratitude va à M<sup>elle</sup> Zahia Brahimi, mon chef d'équipe au CDTA durant la réalisation de ce travail, et directrice des études au ministère de la Poste et des Technologies de la Communication et de l'Information (MPTIC) actuellement, qui a su diriger ce travail avec compétence et intérêt. J'ai pu au cours de ces années profiter de ses conseils avisés et apprécier ses qualités tant scientifiques qu'humaines.

Ensuite, je voudrai remercier les membres de jury d'avoir accepté d'évaluer mon travail. Je commence par remercier le président de mon jury, Monsieur Benslama zoubir, maître de conférences à l'université de Blida pour le grand honneur qu'il m'a fait en acceptant de présider le jury de ce mémoire.

Je remercie également Madame Benblidia nadjia, maître de conférences à l'université de Blida pour l'intérêt qu'elle a porté à mon travail en acceptant d'examiner mon mémoire.

Je remercie également Madame Hamami latifa, professeur à l'école nationale polytechnique pour l'intérêt qu'elle a porté à mon travail en acceptant d'examiner mon mémoire.

Je tiens également à faire part de toute ma gratitude et de ma sympathie à tous mes collègues. Bien au delà des considérations professionnelles, je les remercie vivement pour leur gentillesse et leurs grandes qualités humaines.

Bien évidemment, Je ne peux terminer sans remercier mon mari pour sa confiance et son soutien sans faille, ainsi que mes enfants pour leur patience aux cours de toutes ces années.

## TABLE DES MATIÈRES

<b>INTRODUCTION GÉNÉRALE</b> .....	11
<b>CHAPITRE 1: GÉNÉRALITÉS : CONCEPTS ET DÉFINITIONS</b> .....	14
1.1 Introduction.....	14
1.2 Etat de l'art.....	15
1.3 Règles en stéganographie.....	20
1.4 Architecture.....	21
1.5 Liens avec d'autres technologies.....	22
1.6 Conclusion.....	24
<b>CHAPITRE 2. TECHNIQUES D'INSERTION DE DONNÉES CACHÉES</b> .....	25
2.1 Introduction.....	25
2.2 Le principe de l'IDC.....	25
2.3 Les applications de l'IDC.....	27
2.4 Contraintes techniques de l'IDC.....	27
2.4.1 L'invisibilité/ Indétectabilité.....	27
2.4.2 La capacité.....	27
2.4.3 La robustesse.....	27
2.5 Techniques d'Insertion de données cachées.....	28
2.5.1 Les techniques de l'IDC dans le domaine Spatial.....	29
2.5.2 techniques de l'IDC dans le domaine fréquentiel et multirésolution.....	33
2.5.3 L'insertion de données cachées dans le flux compressé.....	35
2.6 Techniques d'évaluation des méthodes d'IDC.....	36
2.6.1 Evaluation de la robustesse.....	36
2.6.2 Evaluation de la qualité des images.....	37
2.7 Conclusion.....	40
<b>CHAPITRE 3. LE CODEUR JPEG2000</b> .....	41
3.1 Introduction.....	41
3.2 JPEG 2000 : une norme à douze parties.....	43
3.3 Fonctionnalités de JPEG 2000.....	47
3.3.1 Meilleure performance à faible débit.....	47
3.3.2 Compression avec et sans pertes dans un même algorithme.....	48
3.3.3 Transmission progressive par résolution ou qualité.....	48
3.3.4 Accès aléatoire aux données.....	50

3.3.5 Région d'intérêt : ROI .....	50
3.3.6 Détection d'erreurs en milieu bruité .....	50
3.3.7 Traitement d'une large variété d'images .....	51
3.3.8 Architecture ouverte.....	51
3.4 Processus de codage dans JPEG 2000 .....	53
3.4.1 Modèle d'image utilisé.....	55
3.4.2 La grille de référence .....	56
3.4.3 Prétraitement.....	58
3.4.4 La décomposition en ondelettes « DWT ».....	63
3.4.5 La quantification.....	64
3.4.6 Le codage entropique.....	65
3.4.7 Allocation de débit.....	68
3.4.8 Organisation du flux compressé.....	69
3.5 Conclusion.....	71
<b>CHAPITRE 4. TECHNIQUES D'IDC IMPLÉMENTÉES:TESTS ET</b>	
<b>RÉSULTATS</b> .....	72
4.1 Introduction.....	72
4.2 Présentation des approches développées.....	72
4.2.1 L'approche de Katsutoshi.....	73
4.2.2 L'approche par Expansion de différence.....	79
4.2.3 La DE modifiée.....	91
4.2.4 L'approche par BPCS.....	97
4.2.5 L'approche proposée.....	108
4.3 Conclusion.....	112
<b>CONCLUSION GÉNÉRALE</b> .....	113
<b>APPENDICE A. PRESENTATION DES CODEURS IBIG2 ET RLE</b> .....	115
<b>REFERENCES BIBLIOGRAPHIQUES</b> .....	116

## LISTE DES FIGURES

Figure 1.1 Problème des prisonniers _____	15
Figure 1.2 Jean Trithème et Steganographia _____	17
Figure 1.3 Schéma simplifié de dissimulation de données _____	21
Figure 1.4 Schéma complet : dissimulation et recouvrement de données _____	22
Figure 1.5 Liens avec d'autres technologies _____	23
Figure 2.1 Schéma général de l'insertion d'une marque _____	26
Figure 2.2 Schéma général de détection d'une marque _____	26
Figure 2.3 Compromis entre la capacité, l'indéteçtabilité et la robustesse _____	28
Figure 2.4 Insertion avec la méthode du LSB _____	30
Figure 2.5 Schéma bloc de l'algorithme de Wu et _____	31
Figure 2.6 Insertion dans le domaine multirésolution _____	34
Figure 2.7 Classification des attaques sur les schémas de tatouage selon Voloshinovskiy _____	37
Figure 3.1 Image reconstruite décompressée à 0.125 bpp par (a) JPEG (b) JPEG 2000 _____	48
Figure 3.2 Image reconstruite par résolution _____	49
Figure 3.3 Image reconstruite par composante _____	49
Figure 3.4 Image reconstruite par ordonnancement spatial _____	50
Figure 3.5 Chaîne de compression JPEG 2000 : (a) codage (b) décodage _____	54
Figure 3.6 Modèle d'image utilisé (a) image à N composantes (b) composante individuelle _____	56
Figure 3.7 Grille de référence _____	57
Figure 3.8 Image ski reconstruite (a) sans tiling (b) tiling 128x128 (c) tiling 64x64 _____	59
Figure 3.9 Découpage en tuiles dans la grille de référence _____	60
Figure 3.10 Codage des composantes couleurs _____	62
Figure 3.11 Décomposition en ondelettes à trois niveaux _____	64
Figure 3.12 La notion de precinct dans JPEG 2000 _____	66
Figure 3.13 (a) décomposition en code-blocks - (b) représentation en bit plane d'un code-block _____	66
Figure 3.14 Concept de stripe et ordre de balayage des codes-blocks _____	67
Figure 3.15 Concept de couches de qualités 'Layer' _____	69
Figure 3.16 Le flux compressé dans la norme JPEG 2000 _____	71
Figure 4.1 Attaque par compression JPEG2000 _____	73
Figure 4.2 Schéma d'insertion de données cachées proposée par Katsutoshi [32] : (a) Insertion (b) Extraction _____	74
Figure 4.3 (a) Image optique originale (b) Image après insertion (c) Image différence _____	75
Figure 4.4 (a) Image IRM originale (b) Image après insertion (c) Image différence _____	76
Figure 4.5 (a) Image IRM originale (b) Image après insertion (c) Image différence _____	76
Figure 4.6 (a) Image radiologique originale (b) Image après insertion (c) Image Différence _____	76
Figure 4.7 (a) Image radiologique originale (b) Image après insertion (c) Image différence _____	77
Figure 4.8 (a) Image IRM volumineuse (b) Image après insertion (c) Image différence _____	78
Figure 4.9 La valeur du flux de bits à insérer _____	84
Figure 4.10 Organigramme d'Insertion _____	86
Figure 4.11 Organigramme d'Extraction _____	89
Figure 4.12 Insertion par le schéma de Tian _____	91

Figure 4.13 Insertion par la DE modifiée	93
Figure 4.14 Extraction par la DE modifiée	93
Figure 4.15 (a) Image originale (b) DE: capacité d'insertion =26269 bits (0.4008 bpp), PSNR=37.24 dB (c) DE modifiée: capacité d'insertion =32768 bits (0.5 bpp), PSNR=39.01dB	94
Figure 4.16 (a) Image IRM originale (b) DE: capacité d'insertion =23864 bits (0.1308 bpp), PSNR=38,05 dB (c) DE modifiée : capacité d'insertion =32768 bits (0.5 bpp), PSNR=40,15 dB	94
Figure 4.17 (a) Image IRM originale (b) DE: capacité d'insertion =23864 bits (0.1308 bpp), PSNR=38,24 dB (c) DE modifiée : capacité d'insertion =32768 bits (0.5 bpp), PSNR=40,11 dB	95
Figure 4.18 (a) Image Radiologique originale (b) DE: capacité d'insertion =23864 bits (0.1308 bpp), PSNR=38,59 dB (c) DE modifiée : capacité d'insertion =32768 bits (0.5 bpp), PSNR=40,29 dB	95
Figure 4.19 (a) Image Radiologique originale (b) DE: capacité d'insertion =23864 bits (0.1308 bpp), PSNR=38,65 dB (c) DE modifiée : capacité d'insertion =32768 bits (0.5 bpp), PSNR=40,32 dB	95
Figure 4.20 (a) Image optique originale (b) Image marquée après compression jpeg2000 / décompression et après extraction de la donnée (c) Image de différence	96
Figure 4.21 (a) Image médicale originale (b) Image marquée après compression jpeg2000/ décompression et après extraction de la donnée (c) Image de différence	96
Figure 4.22 Valeurs $\alpha$ et $\beta$ pour un bloc 8x8	99
Figure 4.23 Représentation en bits plans et Codage Gray	101
Figure 4.24 Blocs ressources de deux séquences différentes de 8 caractères	102
Figure 4.25 Opération de conjugaison (exemple)	103
Figure 4.26 La technique BPCS	106
=32768 bits (0.5 bpp), PSNR=46,55dB	110
Figure 4.34 (a) Image Radiologique originale (b) BPCS+DE : capacité d'insertion =32768 bits (0.5 bpp), PSNR=46,52dB	110
Figure 4.35 (a) Image optique originale (b) Image marquée après compression jpeg2000 / décompression et après extraction de la donnée (c) Image de différence	110
Figure 4.36 (a) Image médicale originale (b) Image marquée après compression jpeg2000/ décompression et après extraction de la donnée (c) Image de différence	111

## LISTE DES TABLEAUX

Tableau 2.1 Table des différences	32
Tableau 4.1 Paramètres de copression	75
Tableau 4.2 Résultats d'inertion	77
Tableau 4.3 Résultats d'insertion pour une image IRM volumineuse	78
Tableau 4.4 les valeurs de différence pour l'insertion	83
Tableau 4.5 DE modifiée pour Lena	96
Tableau 4.6 DE modifiée pour l'image IRM	96
Tableau 4.7 Code binaire avant et après insertion	100
Tableau 4.8 Code Gray avant et après insertion	100
Tableau 4.9 BPCS+DE modifiée pour Lena	111
Tableau 4.10 BPCS+DE modifiée pour l'image IRM	111
Tableau 4.11 Tian pour Lena 512×512×8 bits	111

## LISTE DES SYMBOLES ET DES ABREVIATIONS

<b>IDC</b>	: Insertion de données cachées.
<b>DWT</b>	: Discrete Wavelet Transform
<b>ICT</b>	: Irreversible Color Transform
<b>RCT</b>	: Reversible color transform
<b>IRM</b>	: Imagerie par Résonance Magnétique
<b>JPEG</b>	: Joint Photographic Experts Group
<b>LSB</b>	: Least significant bit
<b>MSB</b>	: Most Significant Bit
<b>PSNR</b>	: Peak Signal to Noise Ratio
<b>wPSNR</b>	: Weighted Peak Signal to Noise Ratio
<b>SSIM</b>	: Structural Similarity
<b>SVH</b>	: Système Visual Humain
<b>codestream</b>	: flux compressé
<b>BPCS</b>	: Bit Plan Complexity Segmentation
<b>DCT</b>	: Discrete Cosine Transform
<b>DE</b>	: Expansion de différence
<b>JBIG</b>	: Joint Bi-level Image experts Group
<b>JPEG-LS</b>	: Lossless JPEG
<b>Lossless</b>	: Sans pertes
<b>MPEG</b>	: Moving Picture Experts Group
<b>VTC</b>	: Visual Texture Coding
<b>SQ</b>	: simple quantization
<b>MQ</b>	: Multiple quantization
<b>BQ</b>	: Bit-level quantization
<b>EBCOT</b>	: Embedded Block Coding with Optimised Truncation
<b>bpp</b>	: Bit par pixel
<b>Tile</b>	: Tuile

## INTRODUCTION GÉNÉRALE

Les histoires des communications secrètes ont été transmises de génération en génération, principalement utilisées par les militaires et les agences de renseignement. L'envie de résoudre certains problèmes importants comme l'authentification de documents et la gestion des droits d'auteurs, et de plusieurs nouvelles applications telles que la transmission sécurisée des images, ont fait en sorte que l'intérêt dans ce domaine ne cesse de croître. C'est jusqu'à la dernière décennie que cacher l'information a commencé à recevoir une large attention de la communauté de recherche et de l'industrie des technologies de l'information, avec des centaines de publications et des dizaines de brevets.

Cela est évident du fait que la littérature scientifique s'intéresse de plus en plus aux domaines de l'insertion de données cachées, tatouage numérique, la sécurité de l'information, la biométrie, et de la médecine légale. Comme le domaine d'application de l'insertion de données dans les sources multimédias numériques devient très large, plusieurs termes sont utilisés par différents groupes de chercheurs, entre autres : la stéganographie, tatouage numérique ou "watermarking", insertion de données cachées ou "data hiding".

Les techniques de dissimulation de données (data hiding) permettent de transmettre des informations supplémentaires au travers de données hôtes telles que les images numériques. L'insertion de données cachées est l'une des branches de la stéganographie. Son objectif est de dissimuler une quantité très importante de données dans l'image hôte avec le minimum de distorsion possible. Les données cachées doivent pouvoir être correctement récupérées durant l'étape d'extraction.

Nous allons étudier deux types de méthodes d'IDC, les méthodes irréversibles qui présentent l'inconvénient de la distorsion permanente de l'image originale et les méthodes réversibles, utilisées pour assurer l'intégrité de l'image, mais surtout permettre de retrouver l'image originale. Cette réversibilité est importante pour tous les secteurs d'imagerie sensible, surtout pour le secteur médical. En effet, le médecin reconstitue l'image originale (intacte sans aucune dégradation) et l'utilise dans son diagnostic évitant tout risque de modification, puisque la moindre altération de l'image médical peut conduire à un diagnostic faux.

La compression représente, non seulement un passage obligé du stockage ou du transfert d'images, mais aussi l'une des attaques les plus destructrices vis-à-vis de l'IDC. La compression jointe à l'IDC a suscité un intérêt récent. Associer l'IDC et la compression permet d'améliorer la capacité d'insertion et d'assurer une meilleure détection tout en maintenant une bonne qualité d'image. En effet, cette approche offre de nombreux avantages qui sont très recherchés dans certaines applications telles que le contrôle d'intégrité, l'authentification ou la production de documents enrichis. La norme de compression JPEG2000 a été développée par le groupe JPEG (Joint Photographic Experts Group). Ce standard offre de nombreuses fonctionnalités et se caractérise par une grande diversité des options d'encodage menant à de bons compromis compression/qualité.

Puisque les images sont habituellement disponibles sous une forme compressée (JPEG 2000 dans le cadre de ce projet), la mise en œuvre de ces techniques nécessite alors des étapes de codage et décodage supplémentaires, respectivement avant et après l'opération en question. Ceci pose un problème de complexité mais aussi d'accumulation de distorsion due à la répétition des opérations de codage et décodage entraînant, à chaque fois, un peu de perte. L'objectif de ce projet consiste donc à adapter et créer un algorithme d'IDC spécifique pour le domaine compressé de JPEG 2000. Certains algorithmes, tels que ceux d'authentification et de contrôle d'accès, existent déjà mais ont un intérêt limité dans le cas d'images JPEG 2000, du fait qu'ils ne tiennent pas compte des propriétés de la norme. Une mise en œuvre directe de ceux-ci, se traduit généralement par une réduction globale des performances de codage mais empêche aussi de bénéficier de certaines propriétés de la représentation codée (décodage progressif, accès aléatoire, ...). L'objectif de ce mémoire est donc d'adapter, créer et combiner les algorithmes d'insertion de données cachées avec ceux de codage et de décodage JPEG 2000 tout en restant compatible avec la norme.

Notre but aussi dans cette étude, est de présenter une méthodologie d'insertion de données cachées dans des images numériques sous la contrainte de réversibilité, avec comme objectif l'application aux images médicales. Ainsi, nous proposons une technique basée sur la technique du LSB, qui permet d'insérer une donnée cachée par substitution dans les LSBs du flux compressé JPG2000 de l'image.

Nous étudions aussi trois techniques développées dans le domaine spatial. Une technique d'IDC réversible basée sur l'expansion de différence, et une version améliorée de cette dernière. Une technique irréversible nommée Bit plan complexity segmentation(BPCS), basée sur l'insertion dans les bits plans de l'image, combinée avec la version améliorée de la technique d'expansion de différence pour obtenir une technique d'IDC réversible qui

permet une très grande capacité d'insertion avec une meilleure qualité visuelle de l'image support.

Dans le premier chapitre, nous présentons un état de l'art du thème relatif à notre sujet de recherche. Nous revenons sur les origines de l'IDC à savoir la stéganographie, pour ensuite expliciter le principe de l'IDC et ses spécificités. Dans le chapitre 2, nous présentons une revue d'algorithmes d'IDC existants et leurs contraintes techniques, ainsi que leurs domaines d'application. Une présentation du codeur JPEG2000 est introduite dans le chapitre 3. Le chapitre 4 porte sur la présentation des techniques IDCs implémentées dans notre travail, ainsi que les tests d'évaluation effectués et les résultats obtenus. Dans la conclusion, nous résumons notre travail en dégagant les apports des techniques développées, tout en précisant leurs limites ainsi que les perspectives des travaux réalisés dans le cadre de ce mémoire.

## CHAPITRE 1. GÉNÉRALITÉS : CONCEPTS ET DÉFINITIONS

### 1.1. Introduction

La stéganographie est l'art de la dissimulation de communications. Contrairement à la cryptographie, la stéganographie n'a pas pour objectif de sécuriser une communication, mais d'en cacher l'existence même. Les deux disciplines ont donc chacune leur propre domaine de compétence. Dans certaines situations, le fait de vouloir transmettre des données de manière chiffrée est jugé comme suspect. De la même manière, de plus en plus de pays mettent en place de fortes restrictions concernant la longueur des clés cryptographiques, ainsi que la cryptographie en elle-même.

Le regain d'intérêt actuel pour la stéganographie provient de ces restrictions imposées à la cryptographie. D'une manière générale, la stéganographie arrive en renforcement au chiffrement de données. Pour permettre de garantir une confidentialité maximum, les données sont tout d'abord chiffrées avant d'être dissimulées à l'aide d'un processus stéganographique.

Pour illustrer le domaine d'application de la stéganographie, cette dernière est souvent introduite par le problème des prisonniers [1].

Deux prisonniers souhaitent établir un plan d'évasion. Pour ce faire, ils ont la possibilité de se transmettre des messages. Cependant, ces messages transitent à travers le gardien, qui a donc accès au contenu. Tout contenu jugé inapproprié sera détruit, et pourra engendrer une lourde sanction. Dans cette optique, le contenu doit donc paraître anodin aux yeux du gardien. Cette situation rend inutilisable la cryptographie, car un contenu indéchiffrable attirera l'attention du géolier.

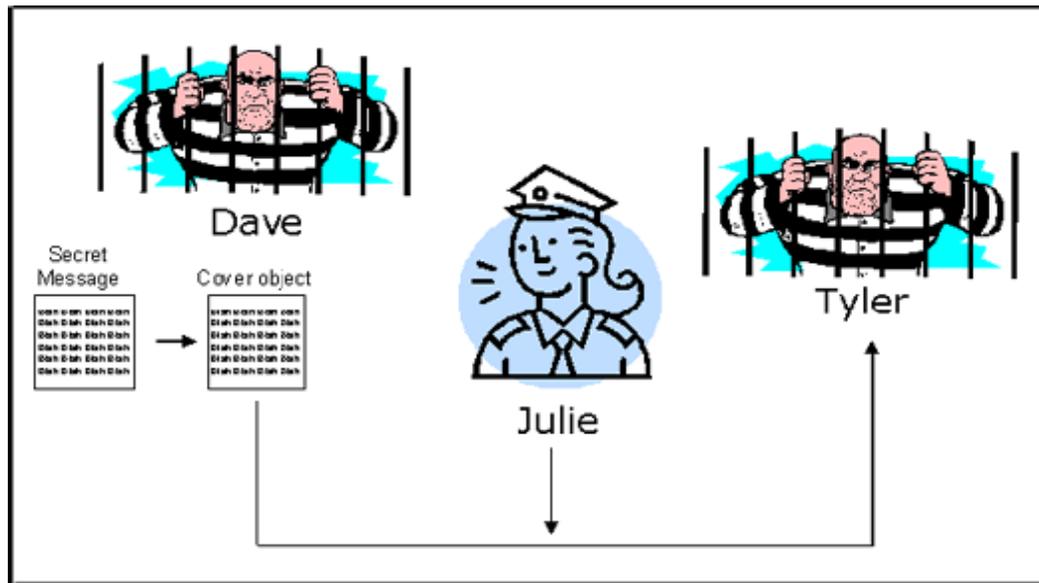


Figure 1.1 Problème des prisonniers

La stéganographie a donc comme objectif de pouvoir entretenir des communications sécurisées, sans pour autant attirer l'attention d'autrui.

De nos jours, la stéganographie a pris de l'ampleur dans des supports bien particuliers, les supports numériques. Que ce soit des fichiers audio, vidéo ou image, ils représentent des supports privilégiés pour la transmission d'information. Avec la généralisation d'Internet, le volume de données qu'il représente est énorme. Cela représente autant de supports possibles à la stéganographie. Parallèlement au développement des outils de la stéganographie, la communauté scientifique s'est organisée depuis le milieu des années 90 et propose depuis, des schémas de plus en plus évolués d'une part, mais aussi des détecteurs tout aussi efficaces.

## 1.2. Etat de l'art

Peut-être plus ancienne et souvent amalgamée à la cryptographie, la stéganographie vit dans l'ombre des « codes secrets », dissimulée derrière un objectif et un formalisme à la fois proches et différents de ceux de la cryptographie.

### 1.2.1. Définition

La stéganographie est un art utilisé depuis des siècles pour faire passer un message de manière inaperçue. Ce mot est issu du grec « Stéganô », qui signifie Je couvre et « Graphô » qui veut dire J'écris [2]. Ainsi, on dissimule les informations que l'on souhaite

transmettre confidentiellement dans un ensemble de données d'apparence anodine afin que leur présence reste imperceptible.

La stéganographie et la cryptographie ont tous deux le même but principal; c'est la protection de l'information. La différence intervient dans la manière avec laquelle chaque technique procède: La cryptographie brouille le message par conséquent il serait difficile de le lire si on ignore le protocole de brouillage, cependant la stéganographie plutôt que de chiffrer des messages avec des méthodes compliquées, elle cache ce message, afin que sa transmission n'éveille pas les soupçons.

### 1.2.2. Historique

Tout au long de l'histoire, la stéganographie tient au même titre que la cryptographie, une place importante dans des événements marquants. Ainsi, Hérodote relate [4], dans son œuvre l'Enquête, comment en 440 av. J.-C, Histiee rase la tête d'un esclave fidèle afin d'y tatouer un message et attendit la repousse des cheveux, rendant ainsi le message invisible. Suite aux instructions de l'esclave, Aristagoras pouvait lire le message qui lui donnait le signal de la révolte contre la Perse. Et comment en 480 av. J.-C, Démarate réussit à prévenir les Grecques d'une invasion imminente du roi de Perse Xerxès 1er en envoyant un message gravé dans le bois d'une tablette d'écriture recouverte de cire, d'apparence vierge. En 300 av. J.-C., Enée le tacticien dans ses Mémoires sur la stratégie, décrit le premier système stéganographique qui consiste à marquer d'un trou les lettres d'un texte constitutives d'un message. En Chine, la coutume veut que le signal de la révolte des chinois contre la dynastie mongole Yuan lors de la fête de la lune, a été donné par des messages cachés dans des gâteaux de lune. Plus technique, l'invention de l'encre sympathique est attribuée au naturaliste Pline l'Ancien, romain du 1er siècle av. J.-C. et est encore utilisée de nos jours. Les techniques de stéganographie devenant de plus en plus savantes, très tôt les premiers ouvrages traitant du sujet voient le jour à partir du XVIIe siècle. En 1499, l'abbé Jean Trithème (1462-1516) publie le premier traité de stéganographie, intitulé Steganographia et composé de trois livres qui ne livrèrent tous leurs secrets que récemment. Le troisième livre n'a été finalement décodé par Thomas Ernst qu'en 1996 et indépendamment par Jim Reeds en 1998. Plus tard, dans les années 80, Margaret Thatcher réussit à identifier la source de nombreuses fuites de documents en traçant ceux-ci à l'aide de techniques de dissimulation d'informations. Enfin, plus récemment, de nombreux spécialistes relayés par les média avancent l'hypothèse selon

laquelle Bin Laden aurait coordonné les attentats du 11 septembre 2001 en utilisant des messages cachés dans des images de sites peu moraux [3].

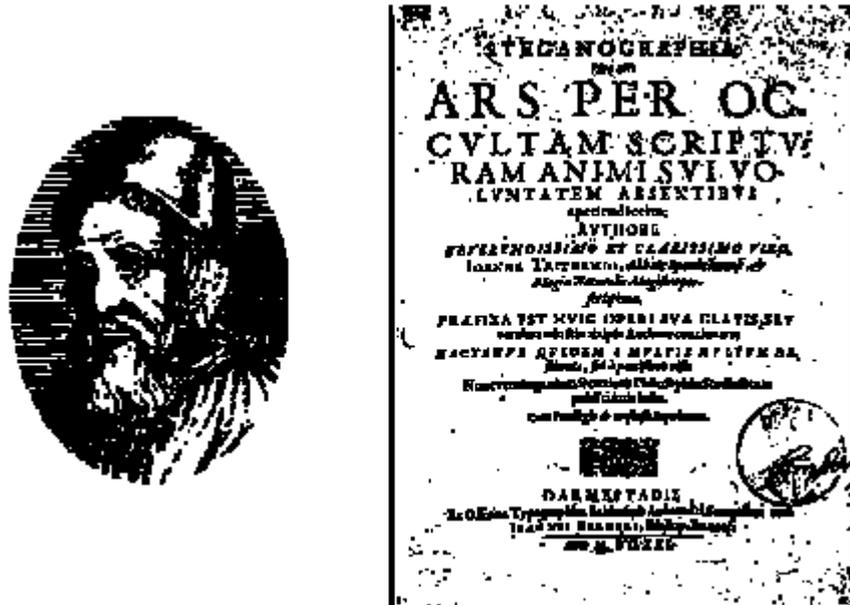


Figure 1.2. Jean Trithème et Steganographia

Paradoxalement, la stéganographie dite moderne, c'est-à-dire adaptée aux données numériques, est relativement jeune. En pleine expansion, elle suit depuis le milieu des années 90 un essor corrélé à celui d'Internet ; le nombre de conférences scientifiques proposant des sessions dédiées à la dissimulation d'information augmentant chaque année. De ce fait, on peut situer avec bonne approximation la naissance de la communauté des stéganographes en 1996, lors de la première édition de l'information cachée (Information Hiding) et l'adoption d'un corpus relatif à la dissimulation d'information. C'est d'ailleurs en 1997 qu'est soutenue une des premières thèses dans le domaine [3].

### 1.2.3. Formes possibles

La stéganographie peut être classée en deux catégories : la stéganographie linguistique et la stéganographie technique. La première comprend toutes les formes de styles possibles, jeux de langue ou utilisation de repères au niveau des caractères. La seconde regroupe les moyens de transmissions purement physiques [4].

### ➤ La stéganographie linguistique

#### – Sémagramme

La forme la plus connue en stéganographie linguistique est le sémagramme. De cette manière, le système stéganographique échappe totalement à l'observateur. Alfred de Musset est l'utilisateur le plus connu de ce procédé puisqu'il a entretenu une communication secrète avec Georges Sand (entre 1833 et 1834) [3] au travers de poèmes qu'il lui envoyait.

#### – Acrostiche

Ce procédé permet de transmettre des données au travers de lettres initiales dans chaque vers de poème et qui, lues de haut en bas, forment un mot ou une expression. Elle a de nombreuses variantes (mot placé dans des vers ou des chapitres,...).

#### – Ponctuation

L'utilisation de points, hauteur de lettres et virgules par les prisonniers de guerre a également permis de transmettre des messages à leur famille.

#### – Nulles

Les codes camouflés, aussi appelés les nulles, consistent à marquer d'un signe particulier certaines lettres d'un texte (par des piqûres d'aiguilles sur ou sous les lettres). Il suffit alors de rassembler les lettres marquées pour former un mot.

#### – Insertion d'erreurs

Mise en valeur de l'information au travers d'erreurs ou de formes de style dans un texte.

#### – Tricot

Pendant la seconde guerre mondiale, un procédé a été réalisé par les agents allemands en Angleterre qui envoyaient des pulls en Allemagne contenant des nœuds dans la laine. Les tricots étaient démaillés à leur arrivée. Sur un mur sur lequel se trouvait l'alphabet sous la forme d'une règle, ils posaient l'extrémité du fil à un point et regardait la position du nœud. Grâce à l'emplacement du nœud sur la règle, ils retrouvaient petit à petit le message caché.

➤ La stéganographie technique

– Encres invisibles

Les encres invisibles peuvent être faites à base de sucres de fruits ou de lait, le message réapparaissant grâce à la chaleur d'une bougie. Des produits chimiques comme le chlorate de soude peuvent également être utilisés ; l'écriture apparaît en passant sur l'encre sèche une petite éponge trempée dans une solution de vitriol de cuivre. Cette technique a également permis de remplacer les trous faits à l'aiguille pour marquer les lettres par des points invisibles. Elle reste encore très utilisée, puisque des encres ultraviolettes sont déposées notamment sur les billets, les certificats. Ce système crée une réaction en cas de photocopies en inscrivant le mot « Void » contribuant à la lutte contre la contrefaçon.

– Camouflage physique

Cette catégorie comprend tout objet ou parties du corps. Les ourlets d'habits contenant un message caché à l'intérieur furent couramment utilisés comme lieu sûr au fil du temps.

– Cardan

Utilisé aujourd'hui comme outils de sécurité par certaines banques, le cardan permet de décoder un message grâce à un masque composé de trous que l'on applique sur un ensemble de données.

– Bandes magnétiques

Cette technique se base sur le fait que les données ne sont visibles qu'au travers des machines et non pas par l'homme. C'est donc le cas pour les cartes de crédit et de téléphone, où les bandes magnétiques ainsi que les numéros de séries sont difficiles à lire et de ce fait, à reproduire. Ces technologies ne servent pas uniquement à la sécurité, mais également aux logiciels, pièces de rechanges mécaniques, produits pharmaceutiques, habits de marques...

– Micropoints

Cette méthode consiste à réduire un texte ou une image en un point d'un millimètre ou moins. Celui-ci est ensuite disposé dans un texte ou une image normale. Ce procédé a été utilisé dans les billets de banque suisses.

#### – Audio

Afin de transmettre de l'information de manière cachée dans du son, différentes techniques existent et se basent sur le fait qu'un son affecte la perception d'un autre :

- Un son plus fort peut en cacher un autre.
- Un son peut être caché temporairement lorsqu'il est moins fort et qu'il est placé avant ou après un son plus fort mais peut être plus facilement détectable.

Il est également possible de cacher des données en utilisant la représentation des notes.

#### – Formes géométriques

Les formes géométriques peuvent elles aussi transmettre de l'information : les points, les fins de lignes et les angles de la figure peuvent permettre de déterminer une lettre.

#### – Anamorphisme

L'anamorphisme peut être considéré comme un système stéganographique : suivant la position et les angles où l'on se situe, on peut voir les œuvres d'art ou les images différemment. Les illusions optiques en sont un exemple très connu.

#### – Autres voies

Il est également possible de communiquer une information en dissimulant un message (ou un fichier) dans les parties inutilisées d'un disque ou d'une disquette.

#### – Images, vidéo

Enfin, une image ou une vidéo peuvent également contenir un message. Une image est constituée de pixels. Il est possible d'insérer des lettres ou des chiffres à l'intérieur sans que ces modifications soient perceptibles à l'œil humain.

### 1.3. Règles en stéganographie

Avant tout, chaque technique doit suivre des règles très importantes et souvent difficiles à concilier :

- L'algorithme doit avoir une complexité minimum afin de pouvoir effectuer le marquage et/ou la détection de la présence de marque en un temps minimal.
- Le support dans lequel le message sera inscrit doit être le moins possible endommagé (l'œil humain ne doit pas s'en apercevoir).
- La capacité du fichier source doit être supérieure à celle du fichier à cacher.

– Le marquage doit pouvoir supporter différentes transformations possibles sans être trop dégradé.

#### 1.4. Architecture

La structure de la communication se base sur la dissimulation, c'est-à-dire l'insertion du message dans la source et le fait de récupérer cette information. Il est également possible de détecter les données sans pour autant les extraire.

Ce processus peut être schématisé comme dans les figures présentées ci-dessous [5].

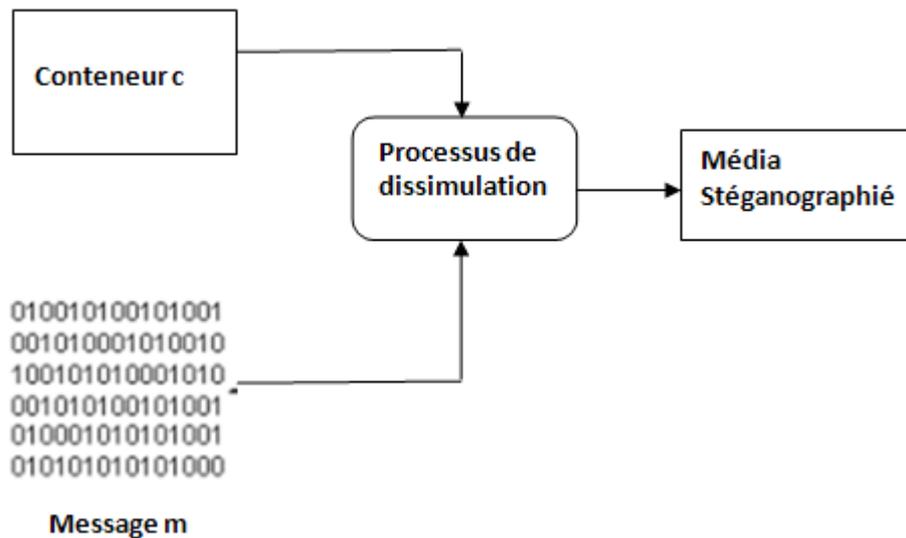


Figure 1.3 Schéma simplifié de dissimulation de données

Il existe trois types de protocoles en stéganographie, correspondant de près à ce qui existe en cryptographie.

La stéganographie pure est un système dans lequel le secret de dissimulation des données ne réside que dans l'algorithme utilisé à cet effet. La découverte de cet algorithme rompt la dissimulation de la communication.

La stéganographie à clé secrète est similaire à la cryptographie symétrique, l'échange de données confidentielles nécessite, au préalable, l'échange d'une clé secrète que l'on ne partagera qu'avec l'interlocuteur. Il est donc nécessaire d'avoir un canal sécurisé, ou de rencontrer en personne notre interlocuteur, afin d'être certain que la clé ne soit pas compromise. Cette clé aura une influence sur la manière de "cacher" l'information.

La stéganographie à clé publique, quant à elle, est similaire à la cryptographie asymétrique. La personne voulant envoyer des données à un autre interlocuteur, sans

éveiller de soupçons, utilisera la clé publique de ce dernier. La clé publique étant à priori connue de tout le monde, il n’y aura pas besoin d’échange préalable “sécurisé”. La personne recevant ce message sera la seule à pouvoir en extraire son contenu à l’aide de sa clé privée.

Voici un schéma plus complet du processus stéganographique :

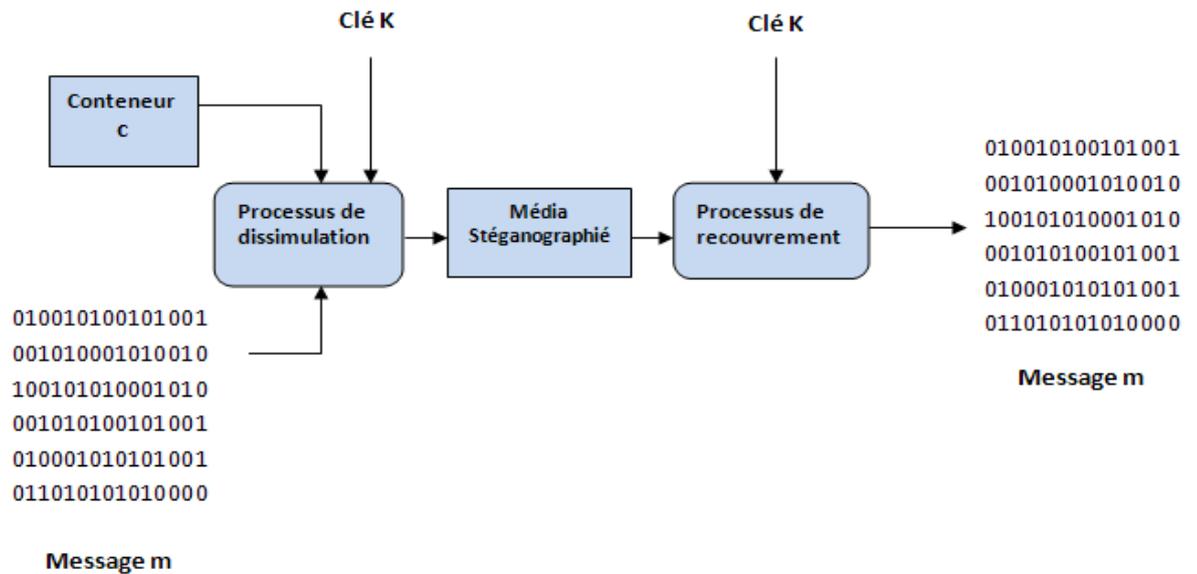


Figure 1.4 Schéma complet : dissimulation et recouvrement de données

### 1.5. Liens avec d’autres technologies

La stéganographie est un art proche de multiples technologies. C’est pourquoi il est essentiel de bien les distinguer les unes des autres [6].

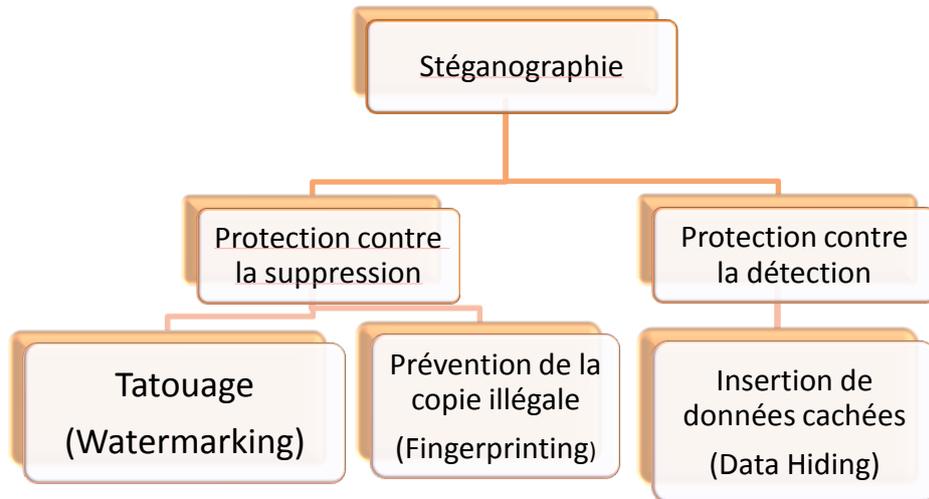


Figure 1.5 Liens avec d'autres technologies

➤ Tatouage (Watermarking)

Le Tatouage a comme objectif d'insérer une marque identificatrice de l'auteur ou du propriétaire légal du document image. Ainsi, en cas de conflit, la preuve de propriété peut être démontrée en détectant la marque. La robustesse de la marque est dans ce cas requise afin de protéger la marque contre toutes les tentatives visant à l'effacer ou la modifier.

➤ Prévention de la copie illégale (Fingerprinting)

Dans ce type d'application, la marque insérée dans le document image est identificatrice de l'entité à laquelle cet objet est légalement destiné (destinataire). Sachant qu'un document image est ainsi marqué, toute personne ayant l'intention de faire des copies (illégales) et de les redistribuer sera dissuadée de peur d'être identifiée. Ce type d'application requiert une marque robuste à toutes les tentatives visant à détruire la marque, à faire échouer sa détection ou à créer une ambiguïté de décision.

➤ Insertion de données cachées (Data Hiding)

L'insertion de données cachées IDC (en anglais data hiding) désigne l'insertion dans un support numérique d'une certaine quantité d'information binaire secrète de manière imperceptible et plus au moins robuste, suivant l'application visée. Dans ce cas, il doit être impossible de distinguer si le message de couverture contient un message utile ou non. La contrainte la plus importante est alors l'imperceptibilité.

## 1.6. Conclusion

Ce chapitre a fait l'objet d'une vue d'ensemble sur le domaine de la stéganographie et identifie une partie des nombreuses approches possibles. Malgré l'ancienneté de sa création, son évolution et son utilisation n'a fait qu'augmenter. Internet lui a permis une ouverture beaucoup plus vaste dans sa pratique et ses techniques. A cette date, la stéganographie est une discipline qui intéresse de plus en plus de personnes et des séminaires sur ce thème ne cessent d'être organisés.

Les travaux de ce mémoire sont dirigés vers une technologie particulière, à savoir l'insertion de données cachées. Dans le chapitre suivant, nous exposerons un état de l'art sur les techniques d'IDC. Bien que le procédé d'IDC soit applicable à tout type de données (texte, image, son), on ne s'intéressera dans ce travail qu'au seul cas des images numériques.

## CHAPITRE 2. TECHNIQUES D'INSERTION DE DONNÉES CACHÉES

### 2.1 Introduction

L'homme a toujours ressenti le besoin de dissimuler des informations, bien avant même l'apparition des premiers ordinateurs et de machines à calculer. Avec la création du réseau internet, qui s'est rapidement imposé comme outil essentiel de communication, ce besoin s'est encore accru. En effet, la communication sur internet pose de plus en plus des problèmes stratégiques et sécuritaires pour beaucoup d'entités communicantes. Les transactions faites à travers le réseau peuvent être interceptées, modifiées ou volées. Avec les nouveaux formats de fichiers multimédias, les outils de sécurité traditionnels tels que la cryptographie s'avèrent souvent inefficaces. Les outils de dissimulations de données sont des solutions potentielles et les dernières années ont vu beaucoup de chercheurs se pencher sur cette nouvelle approche de sécurité.

Le but de ce chapitre, est de présenter les différentes techniques d'insertion de données cachées (IDC) et de décrire les services de sécurité dans lesquels ils peuvent être implémentés, de même que les attaques [7] auxquelles ils peuvent être confrontés.

Bien que le procédé l'insertion de données cachées soit applicable à tout type de document numérique (texte, image ou son), nous nous intéresserons dans ce travail qu'au seul cas des images numériques.

### 2.2 Le principe de l'IDC

L'insertion de la marque peut être vue comme une fonction  $E$  qui produit une image marquée et qui prend comme paramètres d'entrée l'image hôte  $I$ , la marque  $W$  préalablement générée et éventuellement une clé privée  $K$  (voir la Figure 2.1). L'insertion de la marque peut se faire dans  $I_w$  le domaine spatial de l'image (directement sur les pixels) ou bien dans un domaine transformé de l'image (FFT, DCT, ondelettes, etc.) sur les coefficients [8].

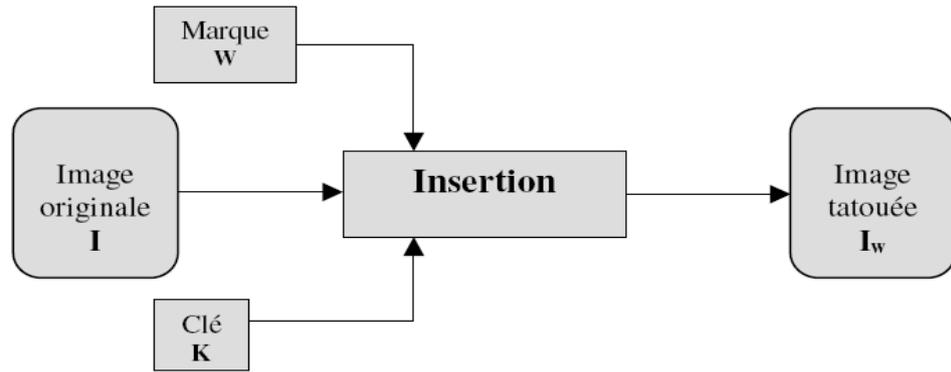


Figure 2.1 Schéma général de l'insertion d'une marque.

La détection de la marque consiste à analyser l'image marquée pour détecter la marque et décider sur un ou plusieurs objectifs de sécurité visés. Ces objectifs peuvent différer d'une application à une autre. Cependant, cette étape peut se dérouler de deux façons typiques :

- Mesure de la présence ou de l'absence d'une marque donnée dans une image donnée. Dans ce cas, le résultat de la détection est une décision binaire  $\{0,1\}$ .
- Extraction de la marque (supposée insérée) et comparaison éventuelle à une marque originale ou générée pour évaluer le degré de distorsion que cette marque a subi (voir la Figure 2.2). Des décisions sont prises en conséquence.

Une méthode de marquage est dite aveugle si la connaissance de l'image originale (non marquée) n'est pas requise à la détection. Nous nous intéressons en particulier aux méthodes aveugles.

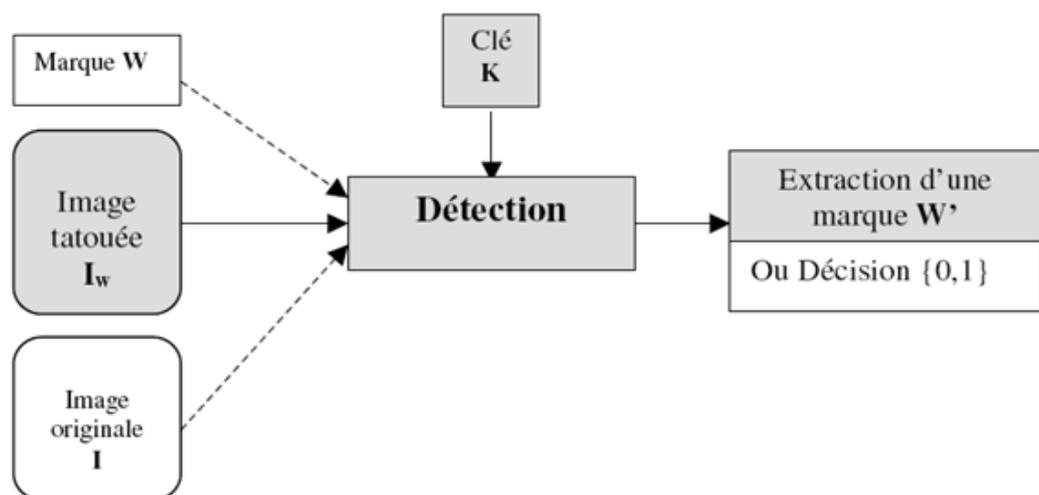


Figure 2.2 Schéma général de détection d'une marque

### 2.3 Les applications de l'IDC

De nombreux usages peuvent exister pour l'IDC dans des domaines très variés mais tous sensibles. Nous pouvons citer à titre d'exemple :

- Elle peut servir à dissimuler des données hautement confidentielles ou interdites au public, par exemple dans le domaine militaire ou médical.
- Elle peut être utilisée en espionnage industriel : toute entreprise a des secrets à protéger (information stratégique, formule chimique d'un nouveau produit, code source d'un logiciel propriétaire...). L'IDC peut rendre le vol de ce genre d'information improbable voir impossible.

### 2.4 Contraintes techniques de l'IDC

#### 2.4.1 L'invisibilité/ Indétectabilité

Dans le domaine d'insertion de données cachées des images, l'invisibilité de la donnée insérée représente un critère important. Il s'agit de faire en sorte que l'impact visuel de l'insertion soit le plus faible possible afin que l'image marquée soit perçue comme fidèle à l'image originale.

#### 2.4.2 La capacité

La capacité est la quantité d'information que l'image hôte peut dissimuler. Elle est généralement mesurée en bits. Dans le contexte de marquage pour la protection des droits d'auteurs, la capacité n'est pas primordiale. L'insertion d'un numéro d'identification codé sur 64 bits suffit dans la plupart des applications. Néanmoins pour l'IDC, cette propriété est très importante. L'augmentation de la capacité de la donnée à insérer se fait au détriment de l'invisibilité d'où la nécessité de prendre en compte les limites de la méthode d'insertion.

#### 2.4.3 La robustesse

Une image marquée peut subir différentes formes de transformations ou de manipulations qui peuvent affecter la donnée insérée. Nous pouvons penser notamment à la compression (JPEG ou JPEG2000 pour les images), à l'ajout du bruit, au filtrage, aux changements de formats, à la rotation et à d'autres formes de manipulations malicieuses comme l'ajout ou la suppression d'objets de l'image. Une insertion robuste garantit une protection de la donnée en résistant à ces manipulations qu'elles soient malicieuses ou

innocentes. Par robustesse on entend la capacité à détecter ou extraire la donnée même si l'image modifiée a été manipulée. Par conséquent, si la marque robuste a été détruite par les manipulations, il faut que cette opération entraîne une détérioration significative de l'image en question. Dans le domaine de la dissimulation d'information, il faut composer avec un compromis entre la capacité, l'indéfectabilité, et la robustesse. Ce compromis est traditionnellement représenté par un triangle comme illustré sur la figure 2.3. Dans les algorithmes d'IDC, le compromis qui nous intéresse est celui entre la capacité et l'indéfectabilité. En effet, l'objectif de l'IDC est bien d'envoyer le maximum d'information sans qu'un attaquant puisse le détecter. La notion de robustesse est plutôt importante pour le tatouage, ceux-ci ne rentrant pas dans le cadre de notre étude. Une bonne introduction sur le tatouage d'image peut être trouvée dans [9].

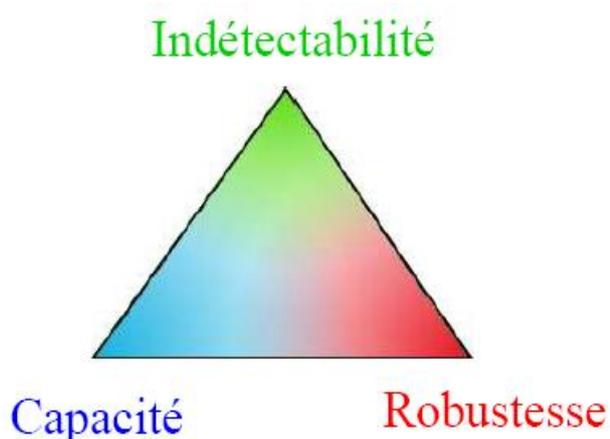


Figure 2.3 Compromis entre la capacité, l'indéfectabilité et la robustesse

### 2.5 Techniques d'Insertion de données cachées

Les techniques d'IDC appliquées aux images sont liées aux espaces de représentation appelés domaines, et chaque domaine d'insertion dispose de divers schémas d'insertion.

La technique du LSB, est de loin la technique la plus répandue. Son succès provient d'une grande facilité de mise en œuvre, ce qui permet d'en trouver de nombreuses implémentations. Sous l'appellation LSB est regroupé tout ce qui a trait à la dissimulation de données par la modification du bit de poids faible d'un élément. Cela va de la valeur d'un pixel, jusqu'à la modification de la valeur d'un coefficient DCT dans le cas de la norme JPEG. Tout se base sur l'insensibilité du système visuel humain à un faible changement de couleurs.

### 2.5.1 Les techniques de l'IDC dans le domaine Spatial

Les techniques d'IDC dans le domaine spatial regroupent les techniques basées sur les modifications des LSBs, les techniques basées sur l'expansion de différence, ainsi qu'une autre méthode connue sous le nom de BPCS (Bit Plane complexity Segmentation).

#### ➤ Les techniques basées LSB

Dans ce domaine, les méthodes modifient directement la valeur de la couleur du pixel. Ce sont des méthodes simples et peu coûteuses en temps de calcul. Certaines techniques se basent sur la représentation de l'image sous forme de matrices de pixels. Elles offrent souvent une bonne résistance aux opérations géométriques. Le schéma du « patchwork » se situe dans ce domaine. Cette technique additive a été proposée par Bender et .col. [10]. A l'aide d'une clé secrète, les auteurs choisissent de façon aléatoire N couples de pixels  $a_i$  et  $b_i$  dans le domaine de la luminance. Si on appelle  $a_i$  et  $b_i$  les valeurs de luminance, l'insertion consiste à modifier les valeurs suivant la formule :

$$\begin{aligned} a'_i &= a_i + 1 \\ b'_i &= b_i - 1 \end{aligned} \quad (2.1)$$

Pour la détection, on déterminera les N couples de pixels à partir de la clé secrète et on calculera :

$$s = \sum_{i=0}^{N-1} (a'_i - b'_i) = \sum_{i=0}^{N-1} (a_i - b_i) + 2 \times N \quad (2.2)$$

Puisque les couples sont choisis aléatoirement et, si on considère N assez grand, la somme des différences  $(a'_i - b'_i)$  est nulle en moyenne et négligeable devant N. Donc, si en détection S est proche de la valeur 2N, on pourra affirmer que la marque est présente, dans le cas contraire on conclura que l'image n'est pas marquée. Cette méthode nous permet d'insérer un seul bit dans l'image. Cette méthode de base n'est bien-sûr pas très robuste. Cependant, différentes extensions de cet algorithme ont vu le jour [11, 12, 13]. Elles permettent par exemple d'accroître la résistance du système à des opérations de filtrage sur l'image en considérant non plus des couples de pixels mais des couples de blocs.

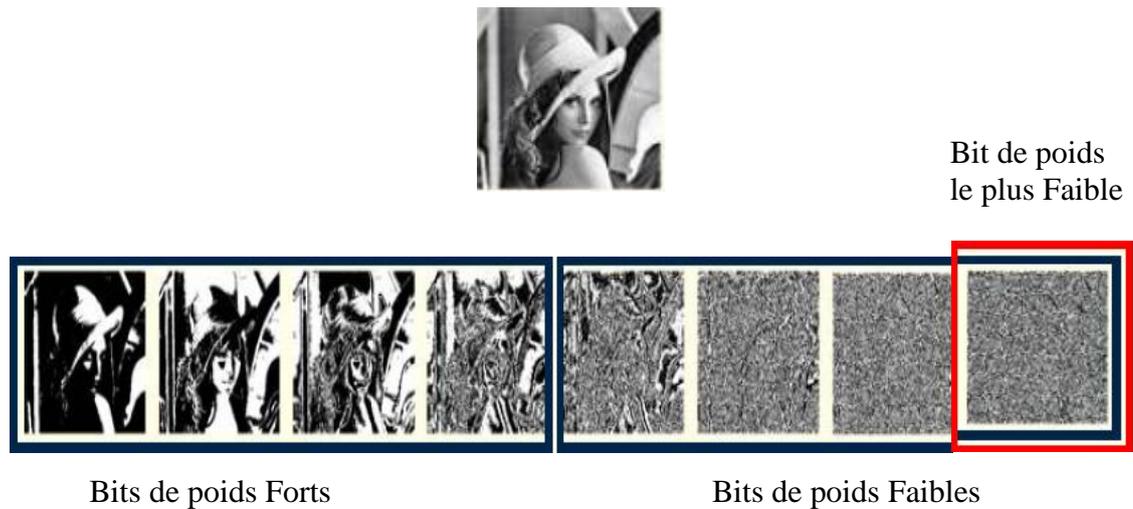


Figure 2.4 Insertion avec la méthode du LSB

➤ Les techniques basées expansion de différence

Les schémas basés expansion de différences utilisent l'expansion de différence entre deux pixels adjacents pour insérer l'information secrète. Nous introduisons ici les deux schémas les plus représentatifs de cette classe, à savoir celui de Tian [14] et Wu et Tsai [15].

— Schéma de Tian

Dans ce schéma, on définit d'abord une transformation entière qui opère sur deux pixels adjacents  $x$  et  $y$  en niveau de gris,  $0 \leq x, y \leq 255$  par :

$$l = \left\lfloor \frac{x+y}{2} \right\rfloor, \quad h = x - y \quad (2.3)$$

Où le symbole  $\lfloor \cdot \rfloor$  est la fonction qui signifie « le plus grand entier inférieur ou égal ». et la transformation inverse :

$$x = l + \left\lfloor \frac{h+1}{2} \right\rfloor, \quad y = l - \left\lfloor \frac{h}{2} \right\rfloor \quad (2.4)$$

De (2.3 ; 2.4), afin de prévenir les problèmes de débordement et dépassement, c'est à dire, de restreindre  $x, y$ , dans l'intervalle  $[0, 255]$ , ceci est équivalent à :

$$0 \leq l + \left\lfloor \frac{h+1}{2} \right\rfloor \leq 255 \quad ; \quad 0 \leq l - \left\lfloor \frac{h}{2} \right\rfloor \leq 255 \quad (2.5)$$

La technique de Tian a attiré l'attention de plusieurs chercheurs pour la simplicité de sa technique et sa facilité de programmation. Néanmoins, ces derniers ont apporté des améliorations quant au problème défini par (2.5) concernant le débordement [16][17][18], et dont nous proposerons dans nos travaux une autre manière de le résoudre.

Cette technique fait l'objet d'une implémentation dans nos travaux, nous la présenterons en détail dans la suite de notre mémoire.

— Schéma de Wu et Tsai

Ce schéma exploite le fait que le système visuel humain est moins sensible aux changements des pixels de fort contraste qu'aux changements des pixels de faible contraste. Il est donc intuitif que nous pouvons cacher plus de bits dans les régions non homogènes de l'image que dans les régions homogènes, à visibilité égale. Il se base sur le calcul de la différence entre les valeurs des pixels adjacents, et en fonction de la grandeur de cette différence, il est possible d'insérer une quantité plus au moins grande de bits d'information secrète. Pour ceci, une table de différence est construite pour déterminer le nombre de bits secrets pouvant être insérés dans chaque paire de pixels consécutifs. Le schéma bloc de l'algorithme est décrit dans la figure 2.5.

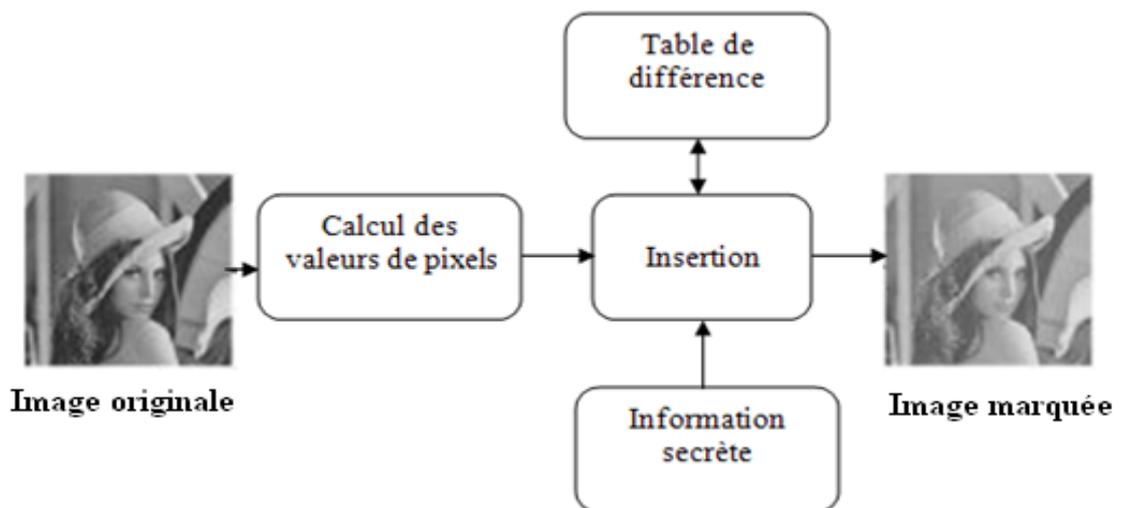


Figure 2.5 Schéma bloc de l'algorithme de Wu et Tsai [15]

Dans une image à niveaux de gris, les valeurs possibles de la différence entre pixels consécutifs sont situées dans l'intervalle  $[-255,255]$ . D'abord, toutes les valeurs de

différence négatives sont transformées en valeurs positives dans l'intervalle [0,255]. Ces valeurs sont ensuite partagées en six intervalles prédéterminés, la largeur de chacun d'entre eux étant en puissance de 2. Le nombre de bits secrets pouvant être insérés est alors donné par le logarithme base 2 de la largeur de l'intervalle correspondant.

Intervalle	0-7	8-15	16-31	32-63	64-127	128-255
Largeur	8	8	16	32	64	128
Nombre de bits	3	3	4	5	6	7

Tableau 2.1 Table des différences

Les grandes lignes de cet algorithme sont :

- L'image est divisée en un ensemble de blocs disjoints de deux pixels consécutifs  $p_j$  et  $p_{j+1}$ .
- Pour chaque bloc de deux pixels consécutifs  $p_j$  et  $p_{j+1}$ , calculer la valeur absolue de leur différence.

$$d_j = |p_j - p_{j+1}| \quad (2.6)$$

- Suivant la différence calculée et les intervalles prédéfinis dans la table des différences, le nombre de bits qui peuvent être insérés peut être déterminé. La valeur de  $d_j$  nous renseigne sur la nature de la région dans laquelle se trouvent  $p_j$  et  $p_{j+1}$  :  
 Si  $d_j \approx 0$  alors  $p_j$  et  $p_{j+1}$  sont dans une région homogène.  
 Si  $d_j \approx 255$  alors  $p_j$  et  $p_{j+1}$  sont dans une région de contours.
- Soit  $s_j$  une valeur secrète à insérer dans les deux pixels consécutifs  $p_j$  et  $p_{j+1}$ . Une nouvelle différence est calculée selon l'équation suivante

$$d'_j = b_j + s_j \quad (2.7)$$

où  $b_j$  appelé base est la borne inférieure de l'intervalle dans lequel le  $d_j$  correspondant se situe. Ceci permettra d'assurer que la valeur de la nouvelle différence appartient toujours à l'intervalle d'origine. La valeur temporaire  $m_j = d'_j - d_j$  est alors calculée et

les pixels originaux  $p_j$  et  $p_{j+1}$  sont transformés en  $p'_j$  et  $p'_{j+1}$  pour cacher les bits de  $s_j$  suivant les règles suivantes :

$$(p'_j, p'_{j+1}) = \begin{cases} \left( p_j - \frac{m_j+1}{2}, p_{j+1} + \frac{m_j-1}{2} \right) & , \text{ si } m_j \text{ est impair et } |d_j| \text{ est impair} \\ \left( p_j - \frac{m_j-1}{2}, p_{j+1} + \frac{m_j+1}{2} \right) & , \text{ si } m_j \text{ est impair et } |d_j| \text{ est pair} \\ \left( p_j - \frac{m_j}{2}, p_{j+1} + \frac{m_j}{2} \right) & , \text{ si } m_j \text{ est pair} \end{cases} \quad (2.8)$$

Il est évident qu'une petite valeur de  $m_j$  engendrera moins de distorsion de l'image qu'une plus grande valeur. Quand une paire de pixels est traitée avec succès, cet algorithme garantit que la valeur de la différence modifiée appartient toujours à l'intervalle original.

#### ➤ La technique BPCS

La méthode du BPCS [19] a été proposée avec comme objectif de fournir une grande capacité de dissimulation. Elle sectionne l'image en petits blocs, qui sont ensuite définis soit comme porteur d'information, soit comme bloc de bruit. La dissimulation se base sur le fait que l'œil humain ne discerne pas bien les modifications intervenues dans des zones fortement bruitées (ou complexes). Ces dernières peuvent être modifiées sans que la perception de l'image ne change. La grande capacité est apportée par la modification possible dans tous les plans de bit. Contrairement à la méthode du LSB, qui est aussi applicable au fichier audio, La BPCS n'est utilisable que sur des images. Malgré ses apparentes qualités, La BPCS n'est pratiquement pas utilisé dans les produits actuels. Il lui est souvent préféré la modification LSB, de par son implémentation très facile. Qtech-HV est le seul produit ayant été trouvé qui utilise cette technique. Il s'agit d'un logiciel développé à l'issue d'une étude menée sur la BPCS. Cette technique fait l'objet d'un développement dans nos travaux et elle sera présentée en détails dans la suite du mémoire.

#### 2.5.2 Les techniques de l'IDC dans le domaine fréquentiel et multirésolution

De nombreuses méthodes ont été développées dans le domaine fréquentiel et multirésolution. Les auteurs de ces méthodes visent ainsi en travaillant dans le domaine DCT [20, 21], anticiper et prévenir au moins les attaques liées à une compression JPEG. Le dernier point opérant en faveur d'une insertion dans le domaine DCT est qu'il est possible de bénéficier, au moins en partie, des études psychovisuelles déjà menées en codage de

source pour gérer les problèmes de qualité visuelle de l'image [22]. M. Barni et col proposent dans [23] un algorithme de tatouage aveugle opérant dans le domaine fréquentiel par utilisation de la DCT. Le principe de cette méthode se résume dans l'intégration d'une séquence pseudo-aléatoire dans un ensemble sélectionné de coefficients DCT. La signature est adaptée à l'image par exploitation des caractéristiques de masquage du système visuel humain, ce qui assure l'invisibilité de la marque. Des tests effectués sur cette méthode ont montré le manque de robustesse d'une telle approche face aux transformations géométriques vu que les coefficients DCT varient considérablement face à ce type de transformations. Les transformées en ondelettes qui tout comme la transformée DCT fait l'objet de nombreuses études dans le contexte du codage ont également trouvé un écho dans la communauté du tatouage d'images [24, 25, 26].

X. G. Xia et col., présentent dans [27] un schéma de tatouage non-aveugle opérant dans le domaine multirésolution par utilisation de la DWT. Le principe de cette méthode se résume dans l'addition d'un code pseudo-aléatoire sur certains coefficients des bandes à hautes et moyennes fréquence de la DWT pour une image donnée. Lors de la phase d'insertion, la marque est détectée par une inter-corrélation entre les bandes de l'image initiale et ceux de l'image tatouée. Cet intérêt repose d'une part sur les analyses en termes psychovisuels menées afin d'optimiser les tables de quantifications des codeurs, d'autre part sur l'aspect multi-échelle de telles transformées propices à une répartition plus robuste du tatouage. Ce gain en robustesse apporté par l'usage d'une transformée ondelette est particulièrement significatif si l'on considère les algorithmes basés sur la nouvelle norme de compression JPEG2000 [28].

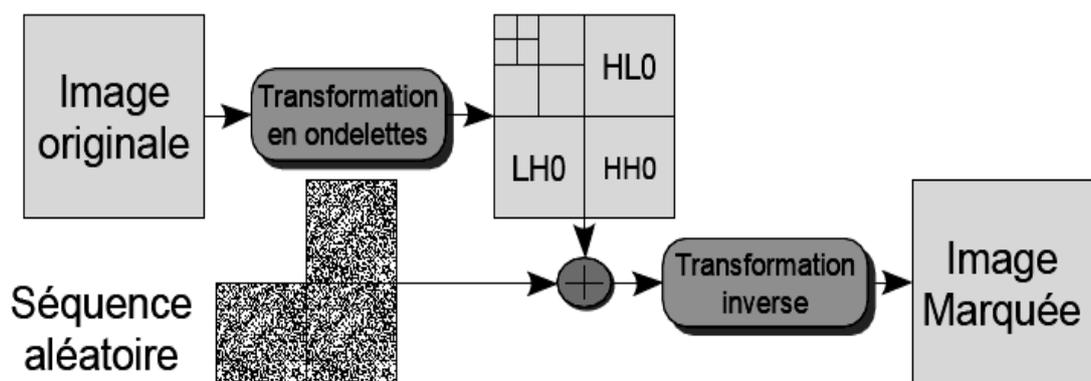


Figure 2.6 Insertion dans le domaine multirésolution

### 2.5.3 L'insertion de données cachées dans le flux compressé

La sphère des applications d'IDC ne cesse d'augmenter depuis quelques années, englobant la protection de données et l'authentification. La compression, quant à elle, est non seulement une opération incontournable pour les images destinées à être transmises, mais aussi une attaque majeure à laquelle doit faire face l'insertion de données cachées.

#### ➤ La compression JPEG

Avec le développement des réseaux, la compression est devenue un point essentiel pour garantir des transmissions rapides. Les méthodes d'insertion de données se doivent d'être robustes aux attaques de compression dont celle de JPEG largement utilisé comme standard. De nombreuses stratégies d'insertion de données cachées ont été développées pour ce format d'images. Des méthodes proposent de modifier des valeurs de la table de quantification standard du JPEG [29]. Alors que certaines suscitent l'altération des coefficients AC et DC de la DCT [30]. D'autres méthodes insèrent dans le codestream JPEG [31].

#### ➤ La compression JPEG2000

Aujourd'hui, grâce à la norme JPEG2000 la DWT est devenue la vedette des transformations de domaines pour les images. Ceci a fait surgir diverses techniques qui modifient les coefficients d'ondelettes pour dissimuler l'information [32, 33]. D'autres approches d'insertion de données cachées directement dans le schéma de fonctionnement de JPEG2000 sont apparues ces dernières années [34, 35,36, 37,38].

La méthode proposée par Katsutoshi [34], consiste en l'insertion de données dans le second layer du flux compressé d'une image donnée. Elle a pour but de préserver la structure et la taille du flux compressé après insertion, et donc n'importe quel décodeur peut facilement reconstruire l'image en question. Cette technique peut tout aussi bien être appliquée dans les deux modes de compression JPEG 2000, avec et sans pertes. Cette technique sera développée dans la suite du rapport car elle a été adoptée dans nos travaux comme une technique d'initiation dans ce domaine.

Chen et al. [35] ont proposé de réaliser l'opération de dissimulation de données directement au niveau du bitstream (flux de codage binaire) JPEG2000 en simulant une nouvelle phase de contrôle de taux. Les couches de qualité sont réorganisées de manière à libérer un espace suffisamment grand pour cacher des données. Su et al. [36] insèrent

l'information cachée dans le bitstream après la phase de contrôle de taux en exploitant l'option lazy mode de JPEG2000. Thomos et al. [37] ont présenté un algorithme de décodage de codes correcteurs pour la dissimulation de données dans des images au format JPEG2000.

## 2.6 Techniques d'évaluation des méthodes d'IDC

Le contrôle de qualité est un problème essentiel dans l'implémentation, la conception et les tests de performance de tout système.

### 2.6.1 Evaluation de la robustesse

Les premières attaques sur les schémas de tatouage ont été proposées par plusieurs chercheurs en 1995 [39]. Ils ont discuté des propriétés du tatouage de documents, et étudié leur robustesse aux attaques. Chacun d'eux proposa sa propre attaque : des attaques de collusion contre les schémas d'étalement de spectre, des attaques d'inversion, des attaques géométriques ou encore des attaques cryptographiques.

Pour essayer d'unir ces différentes approches, petitcolas et al [40] ont créé le banc de tests Stirmark. De nombreuses attaques forment sa structure qui vise à évaluer la performance des algorithmes de tatouage d'images.

Les deux principales classifications dans la littérature scientifiques ont été présentées par Hurtung et Voloshinovskiy [41]. Hurtung propose de classer les attaques en cinq groupes :

1. Les simples attaques qui ne sont pas intentionnelles comme les filtrages, les compressions, le bruit dû à l'utilisation des données, la quantification ou encore les conversions analogiques-numériques
2. Les attaques de désynchronisation qui empêchent la détection telles que les transformations géométriques, le fenêtrage, les permutations de pixels, l'insertion ou la suppression de pixels.
3. Les attaques d'ambigüités qui visent à rendre confuse la détection telle que l'attaque de copie.
4. Les attaques de suppression dont le but est d'estimer la marque dans un premier temps pour ensuite la soustraire à la donnée numérique. Dans cette catégorie, on peut inclure les attaques de collusion, de débruitage, ou les attaques constituées à partir de filtrage non linéaire.

Voloshinovskiy a amélioré cette classification en ajoutant de nouvelles techniques d'attaques. Il a proposé le schéma suivant :

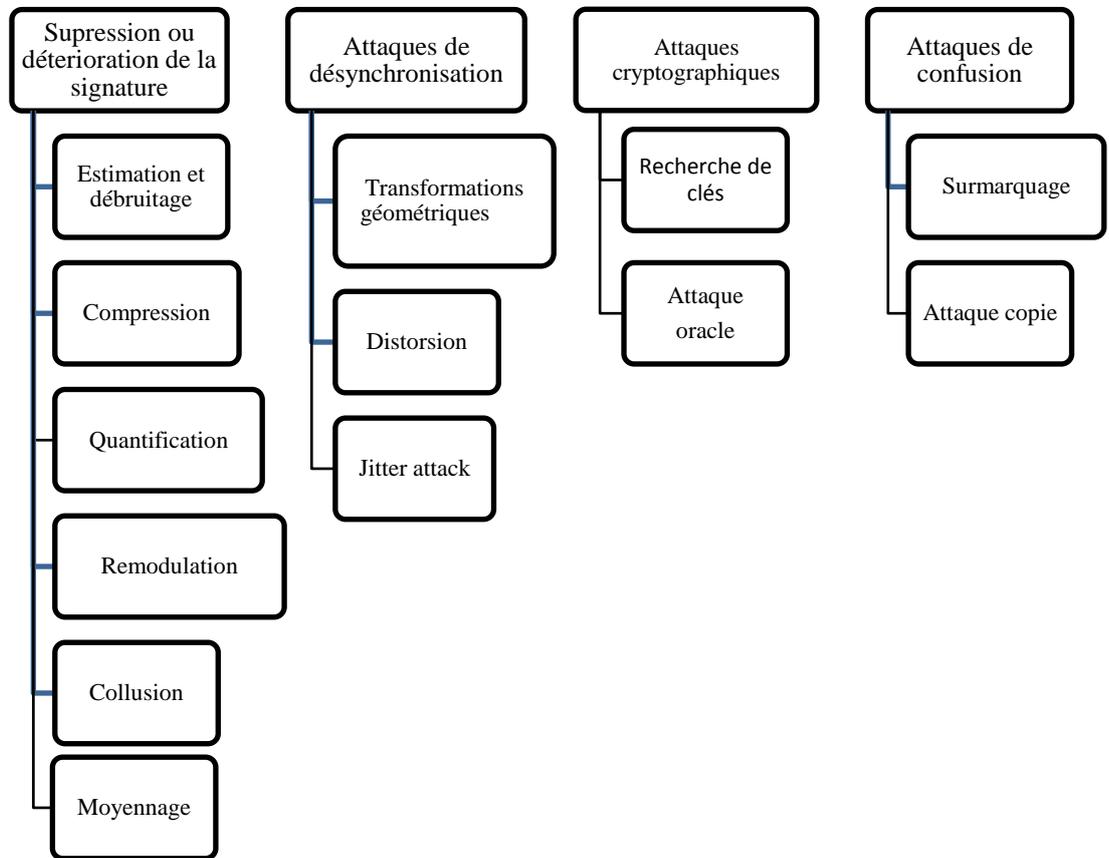


Figure 2.7 Classification des attaques sur les schémas de tatouage selon Voloshinovskiy [41].

### 2.6.2 Evaluation de la qualité des images

Afin de réduire la distorsion visuelle introduite par l'insertion d'une marque dans une image, les algorithmes de tatouage tirent généralement partie des caractéristiques du système visuel humain (HVS) en cachant la marque dans les régions les moins sensibles de l'image (e.g. les contours et les zones de textures). Une bonne métrique de qualité d'image se doit également de prendre en compte les caractéristiques HVS. Malheureusement, le PSNR, qui est la mesure la plus couramment utilisée (comme l'ensemble des métriques basées pixel) ne tient absolument pas compte du contenu fréquentiel de l'image. Bien que servant toujours de référence, le PSNR n'est pas la métrique la plus appropriée au contexte

du tatouage d'image. Certains auteurs préconisent alors l'utilisation d'autres métriques plus adaptées, telles que le wPSNR (weighted PSNR ou PSNR pondéré), le MPSNR (masked PSNR), la mesure de Watson [42].

Les contraintes du domaine médical nécessitent d'être prudent dans l'utilisation des techniques de compression, ou du Data Hiding. Ainsi, une valeur élevée de PSNR peut sembler suffisante pour accepter une technique de codage, alors que la dégradation moyennement faible peut être concentrée juste sur une petite zone importante de l'image, et par là empêcher d'établir un diagnostic précis. La question primordiale est la fidélité du diagnostic; pour mesurer cela, les évaluations subjectives utilisées ne sont pas non plus appropriées. C'est pour cette raison que des approches objectives sont nécessaires pour l'évaluation de la qualité d l'image médicale.

Un nouveau cadre est proposé pour mesurer la qualité d'image fondée sur l'hypothèse que le système visuel est fortement adapté à l'extraction des informations à partir du champ visuel, telles que l'index de similarité (SSIM), Index de qualité universel (UQI), Komparator, et la Référence Réduite de qualité d'image (RRIQA) [43].

#### — Pondération perceptuelle

L'approche la plus pratique est l'introduction d'une pondération perceptuelle  $w$  au sein de la mesure classique d'erreur quadratique moyenne. Le wPSNR - (weighted PSNR) est défini par

$$wPSNR = 10 \log_{10} \left( \frac{\max^2}{wEQM} \right) \quad wEQM = \frac{1}{n} \sum_{i=1}^n \varphi^2 (x_i - y_i)^2 \quad (2.9)$$

où  $\varphi_i$  est une pondération représentant l'importance du  $i^{\text{eme}}$  échantillon. Plusieurs pondérations ont été proposées. La pondération la plus connue est celle de Watson [41].

$$\varphi_i^2 = \frac{1}{\sigma_{bi}^2 + V_i^2} \quad V_i^2 = \frac{1}{\|\phi_i\|} \sum_{j \in \phi_i} |x_j|^\rho \quad (2.10)$$

Dans ces formules, pour le  $i^{\text{eme}}$  coefficient,  $V_i$  est une mesure d'activité de voisinage,  $\phi_i$  (de taille  $\|\phi_i\|$ ) est l'ensemble qui représente les indices des voisins et enfin la variable  $\sigma_{bi}$  est un seuil de visibilité qui dépend de la distance d'observation. Ce seuil est fixé à  $10^{-2}$

pour JPEG2000. Les meilleurs résultats sont obtenus pour  $\rho = 1/2$ . Le wPSNR de Watson a été conçu pour les images dans le domaine DCT notamment JPEG. Il utilise une table de niveau de sensibilité pour les 64 coefficients d'un bloc DCT. Ceci permet de prendre en compte la sensibilité fréquentielle et les phénomènes de masquage dûs à la luminance et au contraste. Une version plus simplifiée du wPSNR de Watson est utilisée dans JPEG2000.

— SSIM (Structural SIMilarity)

C'est l'une des mesures utilisées actuellement, c'est une mesure de similarité entre deux images numériques. Elle a été développée pour mesurer la qualité visuelle d'une image compressée, par rapport à l'image originale. L'idée de SSIM est de mesurer la similarité de structure entre les deux images, plutôt qu'une différence pixel à pixel comme le fait par exemple le PSNR. L'hypothèse sous-jacente est que l'œil humain est plus sensible aux changements dans la structure de l'image. La métrique SSIM est calculée sur plusieurs fenêtres d'une image. La mesure entre deux fenêtres  $x$  et  $y$  de taille  $N \times N$  est :

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2cov_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (2.11)$$

$\mu_x$ ;  $\mu_y$  les moyennes,  $\sigma_x^2$ ,  $\sigma_y^2$ , les variances de  $x$  et  $y$ , respectivement.  $COV_{xy}$  covariance de  $y$ .

$c_1 = (k_1L)^2$ ,  $c_2 = (k_2L)^2$  deux variables destinées à stabiliser la division quand le dénominateur est très faible.  $L$  est l'intensité des valeurs des pixels.

Pour l'évaluation de la qualité d'une image, la formule précédente est appliquée sur la luminance uniquement. Typiquement, les grandeurs sont calculées sur des fenêtres de taille  $8 \times 8$ . La fenêtre courante peut se déplacer pixel par pixel sur l'ensemble de l'image. Cependant, les auteurs proposent de ne considérer qu'un sous-ensemble de ces fenêtres, par exemple en réduisant leur nombre d'un facteur deux dans les deux dimensions. Ceci permet de diminuer la complexité du calcul [44, 45].

## Conclusion

La dissimulation de données est un domaine jeune qui progresse rapidement. Si la robustesse aux attaques reste le talon d'Achille des techniques de tatouage, la propriété la plus importante dans un système d'IDC est l'indéfectabilité. Même si de grands progrès ont été fait ces dernières années, beaucoup reste à faire du coté de la sécurité des algorithmes d'IDC, pour pouvoir rivaliser avec la maturité des outils purement cryptographiques.

Les travaux de ce mémoire sont dirigés vert une application particulière, à savoir la dissimulation de données dans un flux JPEG2000 avec une insertion de données importantes. Ainsi, il est nécessaire de présenter un état de l'art de la norme JPEG2000 dans le prochain chapitre, afin de mieux cerner la possibilité d'intégration d'un schéma d'insertion de données cachées dans un flux JPEG2000.

## CHAPITRE 3. LE CODEUR JPEG2000

### 3.1 Introduction

Aujourd'hui, les applications multimédia nécessitent des capacités de stockage et des débits de transmission élevés. Et comme la bande de transmission et l'espace de stockage sont généralement limités, il s'avère, en conséquence, fondamental de les compresser. La compression vise à réduire la quantité d'information utilisée dans les supports multimédias, tels que l'image et le son, en exploitant les redondances intrinsèques des informations qu'ils contiennent.

L'élaboration de normes de compression par l'ISO « the International Organization for Standardization », l'ITU « the International Telecommunications Union » et l'IEC « the International Electrotechnical Commission » pour la transmission et stockage de l'audio, images et vidéos a conduit au développement de plusieurs systèmes hardware et software applicables à diverses disciplines. Parmi les techniques de compression des images on trouve JPEG, JBIG, JPEG-LS, MPEG-4 VTC et JPEG 2000.

JPEG est le standard international ISO/IEC : 10928-1, ITU-T : T.81 pour la compression numérique et le codage des images fixes [46], le plus utilisé sur internet. JPEG utilise les principaux modes: baseline, sans pertes (lossless), progressive et hiérarchique.

Le mode baseline, le plus utilisé, ne supporte que le codage avec pertes [47] [48] [49]. Dans le mode baseline, l'image est divisée en blocs 8x8 et chacun de ces blocs est transformé avec une DCT (Discret Cosine Transform). Les blocs transformés sont quantifiés avec une quantification scalaire uniforme, scannés en zigzag et codés avec un codeur entropique, le codeur de Huffmann. Le pas de quantification d'un bloc est spécifié dans un tableau de référence, identique pour tous les blocs. Les coefficients DCT des différents blocs sont codés séparément en utilisant un schéma prédictif.

Le second mode, sans pertes (lossless), est basé sur un algorithme complètement différent qui utilise un schéma prédictif. La prédiction est basée sur les trois voisins les plus proches (nearest three causal neighbors). L'erreur de prédiction est codée avec le codeur de Huffmann.

Le mode progressif, diffère du mode hiérarchique dans la partie codage et calcul des coefficients DCT. Le mode progressif, code les coefficients quantifiés avec un mélange de sélection spectrale et d'approximation successive. Le mode hiérarchique, utilise une approche pyramidale multi résolution pour le calcul des coefficients DCT.

JBIG est utilisé pour la compression sans pertes des images binaires, des images en niveaux de gris ou en couleur avec un nombre limité de bits par pixel. Il est utilisé essentiellement pour les fax.

JPEG-LS est l'un des derniers standards de compression sans pertes d'images fixes, il prend en charge aussi une compression « Quasi-sans perte ». Dans la Partie 1, le système baseline, utilise la prédiction adaptative, la modélisation du contexte, et le codage de Golomb. La compression « Quasi-sans perte » est faite en fixant l'erreur maximale permise pour les échantillons. La Partie 2, présente des extensions telles que le codeur arithmétique. L'algorithme JPEG-LS, se caractérise par une faible complexité pour des taux de compression élevés. Toutefois, il n'offre aucune autre fonctionnalité telle la protection face aux erreurs.

Dans le MPEG-4, le codage visuel de texture (VTC) est l'algorithme utilisé pour la compression de la texture visuelle et les images fixes. Il est basé sur la DWT avec la paire (9,3) de Daubechies, la quantification scalaire, le codage zero-tree et le codage arithmétique. La quantification appliquée est de trois types : simple (SQ), multiple (MQ) ou binaire (BQ). VTC ne supporte pas le codage sans pertes.

Après avoir exposé les différents standards qui existent, le nombre d'applications multimédia restent très limitées quand au choix de la technique de compression à adopter pour satisfaire des besoins critiques, en matière de :

- Qualité de compression quelque soit le taux de compression, chose difficile avec JPEG.
- La protection face aux erreurs est une caractéristique qui prend de plus en plus d'importance actuellement vu l'augmentation des transmissions à travers des réseaux sans fil (Wireless) entre autres. JPEG et JPEG-LS offrent des mécanismes de protection basiques, alors que le format PNG (Portable Network Graphics) ne dispose pas du tout de mécanismes de détection d'erreur.
- Besoin d'architecture flexible utilisant à la fois la compression avec et sans perte.

Pour remédier à ces inconvénients et pour servir les exigences des applications d'aujourd'hui demandant à la fois flexibilité et performance, le groupe JPEG a élaboré une nouvelle norme pour le codage des images fixes, le format JPEG 2000.

La norme JPEG 2000 est la dernière norme internationale pour la compression des images fixes, développée en 1997 conjointement par l'ISO et l'IEC. La norme est également recommandée par l'ITU-T [50]. JPEG 2000 a été conçue, pour surmonter le manque d'efficacité de la norme JPEG, et pour répondre aux besoins de nouvelles applications requises par les communications mobiles et Internet. L'avantage principal de cette nouvelle norme est sa capacité d'adaptation. Quelle que soit l'application visée, quelles que soient les ressources ou la bande passante disponible, JPEG 2000 s'adapte de manière optimale.

La partie 1, qui définit le cœur du système de la norme JPEG 2000, est basée sur la DWT, la quantification scalaire, la modélisation du contexte, le codage arithmétique et l'allocation du débit. La DWT supporte des filtres réversibles et non réversibles, respectivement pour une compression sans pertes ou avec pertes. Le codeur arithmétique est basé sur le schéma EBCOT (Embedded Block Coding with Optimised Truncation) proposé à l'origine par TAUBMAN [51]. JPEG2000 tout comme l'EBCOT opère sur des blocs indépendants et non chevauchants codés en plusieurs couches de bits afin de créer un flux scalable.

Il est à noter qu'en plus de fournir une meilleure performance en terme de taux de compression à qualité équivalente, JPEG2000 offre plusieurs fonctionnalités : accès aléatoire aux données, possibilité de définir des régions d'intérêt ROI, protection face aux erreurs, etc., le tout dans un même algorithme.

### 3.2 JPEG 2000 : une norme à douze parties

La norme JPEG 2000 (ISO/IEC 15444-1 et ITU-T T.800) définit douze parties, dont la première appelée le cœur du système. Cette première partie ne définit que l'algorithme de décodage et le format des données compressées. Ceci laisse une assez grande liberté dans le choix du système de codage et ouvre la porte à la compétition pour l'élaboration de codeurs optimaux, à condition que ceux-ci produisent des flux compressés conformes aux spécifications de la norme. Les onze autres parties sont relativement indépendantes.

### 3.2.1 Partie 1

C'est un standard international 'ISO/IEC 15444-1' depuis Décembre 2000, elle définit l'algorithme de décodage, le format du flux en sortie du codeur : le flux de bits (codestream) ou le fichier JP2. Elle a pour but de couvrir une grande majorité des applications touchant au domaine de l'imagerie numérique (archivage, diffusion sur Internet...). L'algorithme de codage dans cette première partie est résumé comme suit : Une image est représentée par une ou plusieurs composantes. Chaque composante est subdivisée en imagerie rectangulaires dites tile(s) de même taille. Chaque tile subit une décomposition en ondelettes dyadique par deux filtres : Daubechies (9,7) pour une compression avec pertes et LEGALL (5,3) pour une compression sans pertes. Le codage de la région d'intérêt (ROI) est assuré par une technique dite « Max Shift ».

### 3.2.2 Partie 2

Une partie standardisée (ISO/IEC 15444-2) en Novembre 2001. Elle définit les extensions de l'algorithme de base 'du format JP2 en format JPX', afin d'améliorer les performances ou augmenter le nombre de fonctionnalités.

En termes de décompositions en ondelettes, l'utilisateur peut choisir n'importe quel filtre et différentes décompositions, et ne se limite pas à une décomposition dyadique. La transformée en couleur définie dans la partie 1, (RGB/YC<sub>B</sub>C<sub>R</sub>) est étendue pour les images de plus de trois composantes. La région d'intérêt est prise en charge dans cette partie par la technique GSBM (General Scaling Based Method). le format JPX permet par exemple [52] d'insérer des liens HTML ou des objets XML. Il permet aussi de gérer plus soupagement l'espace de couleurs utilisé, ou encore de fractionner un flux de bits sur plusieurs fichiers.

### 3.2.3 Partie 3

Motion JPEG2000 est la troisième partie standardisée (ISO/IEC 15444-3) en Novembre 2001. Le format MJ2 [53] en sortie de cette partie est conçu pour contenir une ou plusieurs séquences d'images JPEG 2000. Motion JPEG2000, définit un format de codage de séquences audiovisuelles, en tant que succession d'images compressées par un système respectant la partie 1 de la norme. Cette partie peut être utilisée dans une large variété d'applications, en particulier dans celles qui nécessitent le codage de chaque séquence (frame) indépendamment.

### 3.2.4 Partie 4

Définit les règles de compatibilité avec la norme JPEG 2000. Elle englobe une série de tests de conformité afin de vérifier le comportement correct de la partie 1 du codeur JPEG2000. C'est un standard international ISO/IEC 15444-4 en Mai 2002.

Les profils et les classes de conformité établis ont été créés afin de définir des restrictions de mise en œuvre. Les profils correspondent à une limitation en termes d'options et paramètres de codage. Les classes assurent un niveau minimal de qualité d'images pour un décodeur donné.

Il ya deux profils et trois classes de conformités pour la partie 1 de JPEG 2000. Avec le respect et la définition de la procédure de test, la partie 4, fournit la conformité des flux de bits, pour les deux profils.

### 3.2.5 Partie 5

Aussi standardisée en Novembre 2001 (ISO/IEC 15444-5) [64]. Afin de vérifier la syntaxe définie dans la partie1, et faciliter l'utilisation du format JPEG 2000, deux logiciels libres (open source) ont été développés. Chacun d'eux fournit un codeur et un décodeur compatible avec la Partie 1 de JPEG2000. Les deux implémentations sont :

- JJ2000 réalisée en Java par une équipe de Canon, de l'EPFL et d'Ericsson,
- JasPer écrit en C réalisée par Michael David Adams et soutenu par 'the University of British Columbia' et par 'Image Power'.

### 3.2.6 Partie 6

Définit un format de fichier normatif, mais optionnel (JPM) pour stocker des images composées à la fois de textes, d'images et de graphiques. Le format JPM est une extension du format de fichier JP2 défini dans la partie 1, et utilise également certains éléments du format JPX. Le format JPM [54], utilise une technique permettant de séparer dans un document le texte des images. Cette partie est standardisée (ISO/IEC 15444-6) en mois d'avril 2003.

### 3.2.7 Partie 7

Cette partie n'a pas pour but de réaliser un codeur hardware JPEG 2000, mais de fournir certaines consignes d'implémentation en vue de garantir la compatibilité des réalisations. Ne donnant pas de résultats pertinents, cette partie a été abandonnée.

### 3.2.8 Partie 8

JPSEC (Secure JPEG 2000) propose des solutions pour permettre à des applications de générer, décoder et échanger des flux compressés JPEG 2000 sécurisés [55]. De nombreux aspects touchant à la sécurité, tels l'authentification, l'intégrité des données, la protection des copyrights et des propriétés intellectuelles, la confidentialité et l'accès conditionnel sont du ressort de JPSEC. JPSEC est un Standard International (ISO/IEC 15444-8) depuis Juillet 2006.

### 3.2.9 Partie 9

L'idée de base du standard JPIP est la spécification d'un protocole interactif pour l'échange des images sur internet [56]. Dans un scénario typique de cas d'utilisation [57], un serveur pourrait livrer une portion du fichier JPEG2000 pour satisfaire certaine demande d'un client. Cette demande peut être formulée en termes de région d'intérêt dans l'image, le niveau de résolution et la qualité désirée. Afin de satisfaire ce scénario, il est important que le client et le serveur aient une "conversation intelligente" en vue d'interagir pleinement.

Cette partie vise à définir un protocole efficace, lorsqu'une série structurée d'interactions entre un client et un serveur, par laquelle une partie ou la totalité du flux de bits JPEG2000, des métadonnées et des structures, peuvent être échangées de manière efficace. Ces structures comprennent des définitions de la sémantique et les valeurs qui doivent être échangées, et suggère la façon dont ceux-ci peuvent être échangés en utilisant une variété de moyens de communication réseau. La partie 9, devient un Standard International en Octobre 2004 (ISO/IEC 15444-9).

### 3.2.10 Partie 10

JPEG2000 prévoit d'encoder des images 3D et volumétriques grâce à JP3D [58]. JP3D, est une extension des parties 1 et 2, ce qui signifie que toute partie conforme à la première ou la deuxième partie sera reconnu comme un flux de bits JP3D conforme. JP3D peut servir pour les applications de calcul scientifique tel que l'imagerie volumétrique (images médicales, l'imagerie biologique) et la télédétection (imagerie hyper spectrale, géophysique / astrophysique).

### 3.2.11 Partie 11

JPWL traite le transfert des images et correction d'erreurs dans les réseaux sans fil (Wireless). Cette partie, a pour objectif de proposer des mécanismes additionnels de protection face aux erreurs [59]. Ces mécanismes consistent par exemple à accorder une importance différente à une erreur en fonction du paquet de données affecté.

### 3.2.12 Partie 12

Base Media File Format est le format commun entre JPEG 2000 et MPEG4. Ce format est conçu pour contenir une succession d'images synchronisées à du son et possède une structure orientée objet ; un fichier peut ainsi être décomposé en objets de base très simples. La partie 12, est standardisée (ISO/IEC 15444-12) en Juillet 2003.

## 3.3 Fonctionnalités de JPEG 2000

Dans le domaine de la compression d'images, JPEG s'avère encore très compétitif lorsqu'il est utilisé à des taux de compression faibles ou moyens, mais la piètre qualité des images obtenues à des taux de compression plus élevés ainsi que son manque de flexibilité et de fonctionnalités témoignent de la nécessité que la nouvelle norme JPEG 2000 doit disposer d'une plus grande flexibilité que JPEG. Une flexibilité en terme de qualité des images, du type d'images traitées, des applications visées (application militaire, commerciale, médicale, ...), ainsi que l'environnement utilisé (réseau sans fil, réseau à large bande passante, PDA, ...).

Les principales caractéristiques de JPEG 2000 qui font sa spécificité et son intérêt sont:

### 3.3.1 Meilleure performance à faible débit

JPEG 2000 offre une meilleure qualité d'images à des taux de compression élevés, comparés aux standards qui existent. Alors que JPEG, offre un excellent rapport débit-distorsion à de faibles taux de compression, la distorsion devient inacceptable à des taux plus élevés (en dessous de 0.25 bpp, par exemple pour des images très détaillées en niveaux de gris tel que le montre la figure 3.1).



Figure 3.1 Image reconstruite décompressée à 0.125 bpp par (a) JPEG (b) JPEG 2000 [58]

### 3.3.2 Compression avec et sans pertes dans un même algorithme

Un avantage fondamental par rapport aux standards qui existent, JPEG 2000 permet la compression avec pertes et sans pertes dans un même algorithme. Dans [60][61][46], des exemples d'applications utilisant cette fonctionnalité incluent les images médicales. L'archivage où la plus haute qualité est essentielle pour la préservation des images mais pas nécessaire pour la fonctionnalité d'affichage. Aussi, les applications réseaux connectées à des dispositifs de différentes capacités. Dans ce cas, il suffit qu'une application ne transmette que le début du fichier contenant une image haute résolution pour qu'elle offre déjà une version de moins bonne qualité.

### 3.3.3 Transmission progressive par résolution ou qualité

La transmission progressive dans JPEG 2000 est assurée par résolution ou par qualité pour différentes applications : navigation sur internet, l'archivage et l'impression des images.

Il est possible de commencer par une version de basse qualité, ou de faible résolution de l'image (figure 3.2), l'affinant ensuite en précision. On peut également placer les composantes des images couleur les unes à la suite des autres (Figure 3.3). On peut enfin exiger un ordonnancement spatial, qui commence par placer toute l'information concernant une zone de l'image avant de passer à une autre (figure 3.4).

Il est à signaler que le passage d'un mode de transmission à un autre ne nécessite pas la décompression de l'image mais seulement un réordonnancement des paquets de données dans le flux compressé JPEG 2000 généré.



Figure 3.2 Image reconstruite par résolution

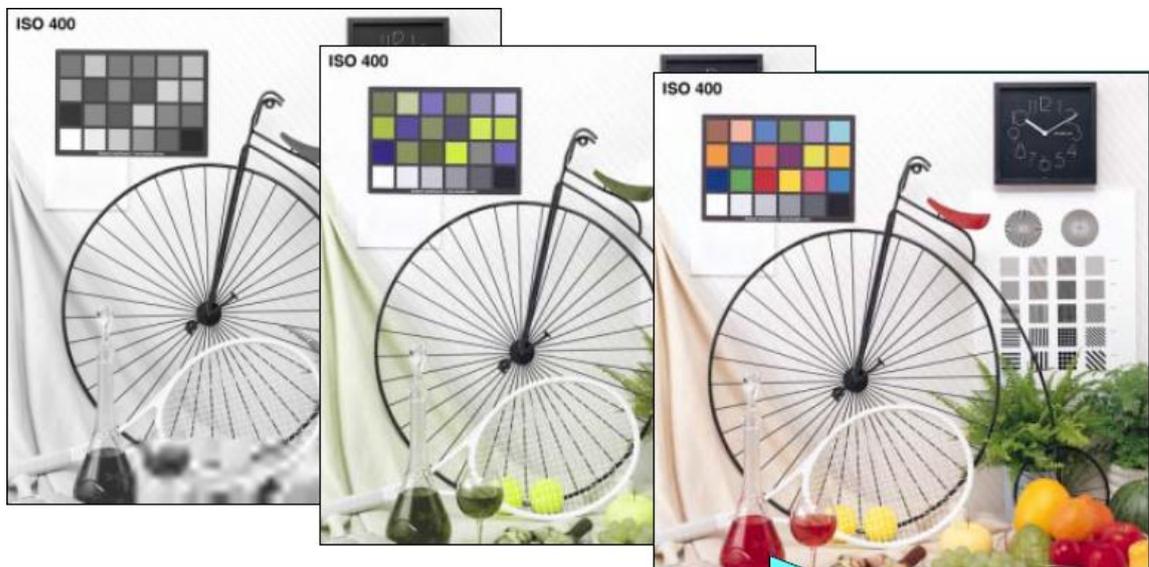


Figure 3.3 Image reconstruite par composante

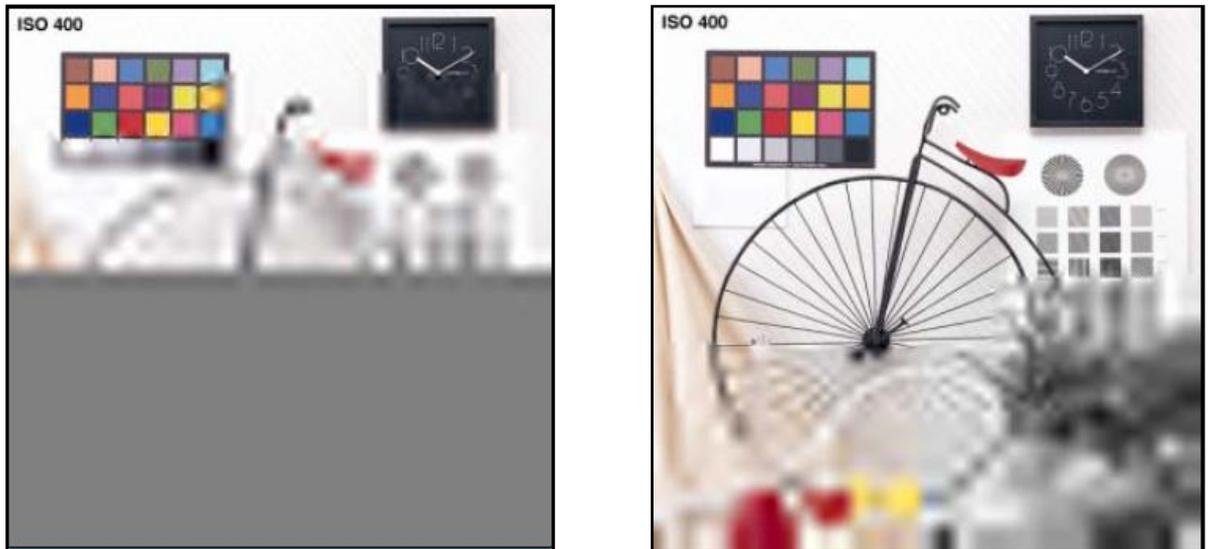


Figure 3.4 Image reconstruite par ordonnancement spatial.

### 3.3.4 Accès aléatoire aux données

Le flux compressé en sortie du codeur JPEG2000 contient un certain nombre d'informations sur les données compressées. Celles-ci permettent d'accéder à n'importe quelle zone de l'image, sans avoir à décompresser entièrement le fichier. Grâce à cette fonctionnalité, on peut extraire facilement les caractéristiques de l'image, effectuer des rotations, des translations, du filtrage, ou définir des régions d'intérêts.

### 3.3.5 Région d'intérêt : ROI

Souvent, une image fixe contient des parties de haute importance que le reste de l'image. Ces parties sont appelées régions d'intérêt. JPEG2000 utilise le principe de régions d'intérêt, pour transmettre ces zones avec une qualité supérieure [62]. Cette fonctionnalité permet aux utilisateurs de définir des Régions d'intérêts devant être codées et transmises sans pertes ou avec meilleure qualité et distorsion minimale au reste de l'image.

### 3.3.6 Détection d'erreurs en milieu bruité

Une fonctionnalité très importante lors des transmissions dans des environnements bruités, tels que les réseaux sans fil. Lors de transmission du flux compressé d'une image donnée, il peut arriver qu'un certain nombre d'erreurs soient introduites dans le fichier,

réduisant la qualité de l'image à la décompression, ou rendant même le fichier Inutilisable. JPEG2000 offre des mécanismes de détection et correction d'erreurs afin d'éviter un échec ou une décompression catastrophique [63].

### 3.3.7 Traitement d'une large variété d'images

La norme JPEG2000 s'adapte à l'encodage d'une très large variété d'images. Le système doit compresser et décompresser des images avec différentes plages dynamiques (par exemple, 1 bit à 16 bits) pour chaque composante de couleur. Des exemples d'applications qui peuvent utiliser cette fonctionnalité incluent les documents composés d'images et de texte, les images médicales avec annotation, les graphiques et les images générées par ordinateur et de télécopieur.

### 3.3.8 Architecture ouverte

L'objectif de JPEG2000 est d'offrir une architecture souple pour répondre à des besoins et des utilisations très diverses. Grâce à cette fonctionnalité, seuls un décodeur est nécessaire pour mettre en œuvre l'ensemble des outils de base et un analyseur qui comprend la syntaxe du flux compressé [60][46].

Chacune des fonctionnalités énoncées ci-dessus possède des applications dans des domaines très variés. Nous pouvons citer:

#### ➤ Imagerie satellite

Les images acquises par satellite offrent une perspective unique de la terre, de ses ressources et de l'impact humain sur celles-ci. La caractéristique principale des images traitées par un satellite est leur grande taille. Ainsi, il sera courant de devoir traiter des images de 30 000 pixels sur 30 000 pixels, avec une précision de 16 bits sur chaque pixel. On parle dans ce cas de méga-images.

La navigation au sein de ces méga-images est un problème complexe, pouvant être très gourmand en bande passante ou en ressources calculatoires. Plusieurs fonctionnalités de JPEG2000 s'avèrent très utiles pour mettre en place une navigation optimisée au sein de ces grandes images. C'est en particulier le cas de l'approche multi-résolution, du protocole JPIP ou de la gestion des régions d'intérêts.

### ➤ Imagerie médicale

L'imagerie médicale est certainement un des secteurs d'application clé de JPEG2000. La variété des images à traiter, le volume important de données à traiter et la qualité exigée par les médecins nécessitent une flexibilité et une qualité de compression que JPEG2000 assure largement.

Sans rentrer dans les détails des images médicales, on peut dégager une caractéristique qui est la grande quantité de données produites par un même examen médical. En effet, il s'agit soit d'une seule image de très grande résolution, soit d'une série d'images (frames), de résolution moyenne tel est le cas des images DICOM. L'imagerie médicale nécessite également des outils performants de navigation dans des méga-images.

A cet effet, l'imagerie médicale tirera donc profit des mêmes fonctionnalités que dans le cas des images satellites. Par ailleurs, le traitement des images volumétriques pris en charge par JP3D pourra être très utile. Motion JPEG peut servir quant à lui à encapsuler l'ensemble des images (frames) d'un même examen médical et même à lui associer des données audio comme les commentaires du médecin.

### ➤ Cinéma numérique

Le cinéma numérique est caractérisé principalement par le haut débit des données à garantir et par la très grande exigence de qualité des images à projeter.

Concernant la qualité des images, on requiert habituellement pour le cinéma que les images projetées soient visuellement sans pertes, ce qui ne permet pas d'utiliser des taux de compression très élevés.

Les phases clé du développement d'un film sont : le tournage, l'édition, l'archivage, la distribution et la projection du film. Au cours de chacune de ces étapes, l'utilisation de JPEG2000 présente un certain nombre d'avantages. Citons entre autres, la compression sans pertes (pour l'archivage), la possibilité d'obtenir à partir d'un même fichier des versions de différentes qualités du film (pour la distribution), l'accès aléatoire au flux de bits (pour l'édition). Notons que le tournage et la projection du film requièrent une réalisation hardware du (dé)codeur.

### ➤ Imagerie Mobile

Grâce aux réseaux sans fils que l'on trouve dans de plus en plus de lieux, de nombreuses applications en imagerie voient le jour. Par exemple, combiné à la technologie GPS, JPEG2000 pourrait être utilisé pour la localisation et le guidage de l'utilisateur dans

une ville, comme proposé dans [64]. Muni de son PDA, disposant d'une fonction GPS, l'utilisateur transmet ses coordonnées à un serveur d'images qui lui renvoie la portion d'images (un plan de ville) nécessaire pour se localiser. L'utilisateur peut alors zoomer sur une zone désirée ou encore demander des informations supplémentaires sur le trafic.

➤ Photographie numérique

Un appareil photo intégrant un codeur JPEG2000 présenterait de nombreux avantages comme celui de pouvoir choisir la qualité de la photo après l'avoir prise, permettant une meilleure gestion de l'espace mémoire, ressource limitée dans ce genre d'appareil. Les albums photo du particulier pourraient ensuite être déposés sur un serveur et partagés avec son groupe d'amis en profitant de tous les avantages de JPIP.

➤ Vidéosurveillance

La vidéosurveillance pourrait tirer profit des avantages offerts par JPEG2000, en particulier de la gestion des régions d'intérêt et de l'approche multi-résolution. Une camera de surveillance pourrait ainsi avoir un comportement "intelligent" en se focalisant sur les objets en mouvement et en ne transmettant aux responsables de la sécurité (munis par exemple d'ordinateurs de type PDA) que les zones potentiellement intéressantes de l'image.

### 3.4 Processus de codage dans JPEG2000

La nouvelle norme JPEG 2000 a été conçue en fonction des principes de la DWT. Le même algorithme peut être utilisé pour une compression avec ou sans perte. Pour le codage des données, le codeur utilise la technique de codage par plan de bit et un codage arithmétique plus spécifiquement le codeur MQ de la norme JBIG2.

Seul l'algorithme de décodage est décrit dans la partie une de JPEG2000. Toutefois, il semble plus aisé de comprendre les concepts sous-jacents à chaque bloc dans le sens du codage. Le processus de codage JPEG 2000 (codeur et décodeur) est illustré en figure 3.5.

Dans la partie codage, l'image source subit une décomposition discrète en ondelettes. Les coefficients DWT sont ensuite quantifiés et codés avec un codeur entropique avant de former le flux compressé en sortie « le codestream ».

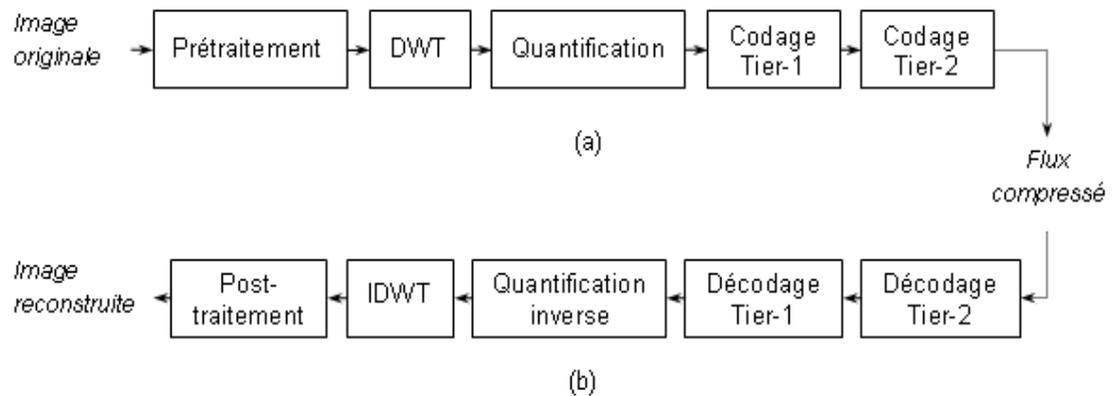


Figure 3.5 Chaîne de compression JPEG2000 : (a) codage (b) décodage

La chaîne de décodage se comprend facilement en parcourant les blocs dans l'ordre inverse. Le flux compressé est décodé avec un décodeur entropique, une quantification inverse est appliquée suivie d'une décomposition en ondelettes inverse pour générer l'image reconstruite en sortie.

Même si les blocs généraux du diagramme ressemblent à ceux du standard JPEG, il y a des différences radicales dans tous les processus de chaque bloc de la chaîne de codage. D'une manière simplifiée, le système est décrit de la façon suivante [46]:

- Si l'image source est multi composante, chaque composante est traitée de manière indépendante.
- Les images sont découpées (optionnel) en imagettes rectangulaires dites tuiles (tiles). La tuile d'une composante donnée (tile-component) est l'unité de base de l'image originale ou reconstruite.
- Chaque imagette d'une composante subit une transformation en ondelettes pour générer différents niveaux de résolution.
- Un niveau de décomposition est formé de sous-bandes des coefficients qui décrivent les caractéristiques des fréquences des zones locales des composantes imagettes.
- Les coefficients des sous-bandes sont quantifiés et regroupés dans des blocs appelés « code-blocks ».
- Les bit-planes des coefficients dans un code-block sont codés avec un codeur entropique.

- Le codage est réalisé de telle manière que certaines régions d'intérêt peuvent être codées avec une meilleure qualité au reste de l'image.
- Des marqueurs sont ajoutés au flux compressé pour permettre la détection d'erreurs.
- Le flux compressé possède une entête (main header) principale qui décrit l'image originale et les différents modèles de décomposition et codage utilisés pour localiser, extraire, décoder et reconstruire l'image avec la résolution désirée, la fidélité souhaitée, les régions d'intérêt et autres caractéristiques.

La structure générale du codeur JPEG2000 schématisée en figure 3.5, englobe les blocs principaux suivants : prétraitement / post-traitement, la transformée en ondelettes/transformation en ondelettes inverse, la quantification / quantification inverse, codage entropique / décodage entropique. Il convient ici de noter que, la structure du décodeur est le miroir du processus de la partie codage. Il existe une correspondance une à une entre les deux processus : chaque bloc fonctionnel dans le décodeur est exactement ou approximativement l'inverse du bloc correspondant dans la partie codage. Dans ce qui suit, chaque bloc est détaillé dans le sens du codage avec spécification des changements s'ils existent correspondants à la partie décodage.

#### 3.4.1 Modèle d'image utilisé

Dans le codeur JPEG2000, une image est composée de plusieurs composantes (limitée à  $2^{14}$ ) [65], tel que montré en figure 3.6 (a). Chaque composante est constituée d'une matrice rectangulaire d'échantillons (figure 3.6(b)). Les échantillons de chaque composante sont des valeurs entières, signées ou non signées avec une précision de 1 à 38 bits/échantillons. Le signe et la précision des échantillons sont spécifiés individuellement par composante.

Ces composantes sont utilisées pour représenter les informations spectrales de l'image originale. Une image en couleur RGB, par exemple, est composée de 3 composantes : une pour la couleur rouge, une pour la couleur verte, et la dernière représente la couleur bleue. Une image en niveaux de gris par contre ne compte qu'une seule composante correspondant à l'information de luminance de l'image. Notons que les composantes d'une image ne sont pas échantillonnées à une résolution unique, et donc peuvent avoir des tailles différentes [65].

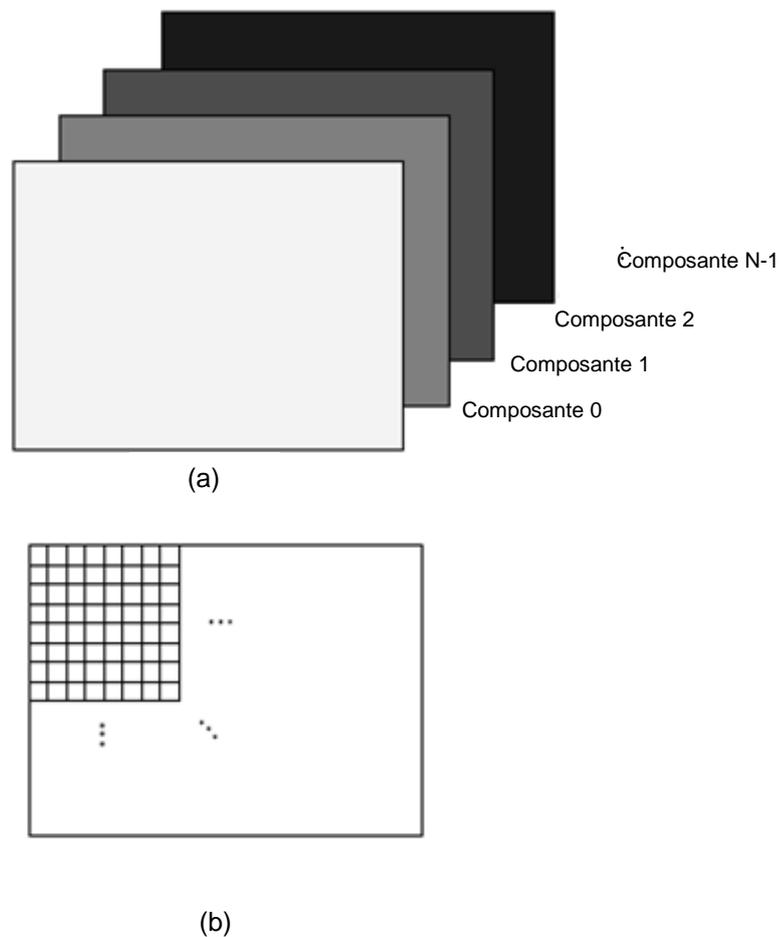


Figure 3.6 Modèle d'image utilisé (a) image à N composantes

(b) composante individuelle

### 3.4.2 La grille de référence

Le codeur JPEG2000 décrit la géométrie des différentes composantes d'une image en termes d'une grille rectangulaire appelée « grille de référence ».

La grille de référence est décrite en figure 3.7 : L'origine est le point supérieur gauche de la grille de taille  $X_{siz} \times Y_{siz}$ . La largeur et la hauteur de la grille sont limitées à  $2^{32}-1$  points, précisant ainsi la taille maximale d'une image qu'un codeur peut traiter. La zone délimitée par les deux points  $(X_{Osiz}, Y_{Osiz})$  et  $(X_{siz}-1, Y_{siz}-1)$  est appelée « zone image ». Toutes les composantes de l'image sont projetées sur la zone image de cette grille de référence [49].

Etant donné que les composantes ne sont pas toutes échantillonnées suivant la même résolution, des informations supplémentaires sont nécessaires pour leur projection dans la grille de référence.

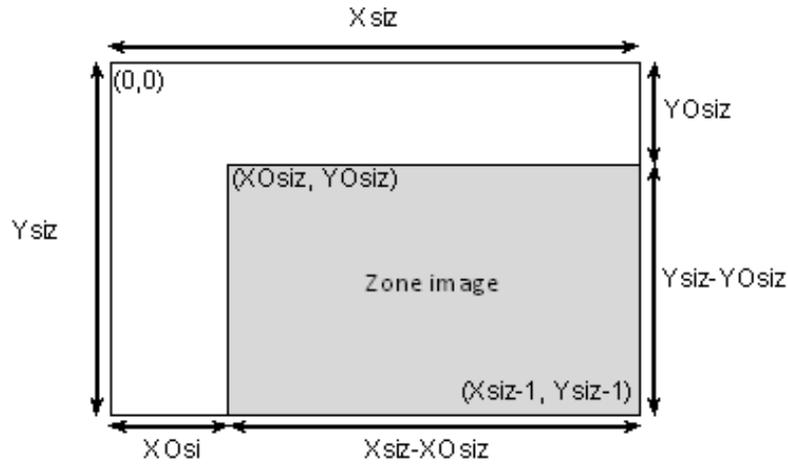


Figure 3.7 Grille de référence

Ainsi et pour chaque composante, une période d'échantillonnage horizontale «XRsiz» et verticale «YRsiz» de la grille de référence, sont exprimées en unités. Ces deux paramètres spécifient une grille d'échantillonnage rectangulaire constituée de tous les points dont les positions horizontales et verticales sont des multiples entiers de XRsiz et YRsiz respectivement. Tous les points qui tombent dans la zone image constituent des échantillons de la composante en question.

Dans un système propre des coordonnées, une composante aura la taille suivante :

$$\left( \left\lceil \frac{Xsiz}{XRsiz} \right\rceil - \left\lfloor \frac{XOsiz}{XRsiz} \right\rfloor \right) \times \left( \left\lceil \frac{Ysiz}{YRsiz} \right\rceil - \left\lfloor \frac{YOsiz}{YRsiz} \right\rfloor \right) \quad (3.1)$$

Les deux points limites d'une composante sont :

$$\left( \left\lfloor \frac{XOsiz}{XRsiz} \right\rfloor, \left\lfloor \frac{YOsiz}{YRsiz} \right\rfloor \right) \text{ et } \left( \left\lceil \frac{Xsiz}{XRsiz} \right\rceil, \left\lceil \frac{Ysiz}{YRsiz} \right\rceil \right) \quad (3.2)$$

Avec une zone image de taille:  $(Xsiz - XOsiz) * (Ysiz - YOsiz) \dots$  (3.3)

Pour une image donnée, différentes combinaisons des paramètres  $X_{siz}$ ,  $Y_{siz}$ ,  $XO_{siz}$  et  $YO_{siz}$  peuvent être choisies pour obtenir une zone image avec la même taille.

Les changements opérés sur les paramètres  $XO_{siz}$  et  $YO_{siz}$  (tout en gardant la taille de la zone image constante) permettent de réaliser un certain nombre d'opérations de base sur l'image comme un décalage, un recadrage et des rotations par un multiple de 90 degrés.

### 3.4.3 Prétraitement

Trois opérations sont réalisées au cours de cette phase. La première est optionnelle, elle consiste à partitionner l'image en rectangles de même taille appelés tuiles (tiles). La seconde opération n'a lieu que si les coefficients de l'image originale sont non signés. Elle consiste à ramener les coefficients de l'image non signés en logique signée, c'est-à-dire centrés autour de zéro. Enfin, la troisième opération appelée la transformée en couleur.

#### ➤ Le découpage en tuiles

Le découpage en tuiles signifie le partitionnement de l'image originale (source) en un ou plusieurs blocs rectangulaires non chevauchants nommés tuiles (tiles). Les tuiles sont codés de manière indépendante comme s'il s'agissait d'images entièrement distinctes [60][61][46]. Toutes les opérations, y compris la transformée en ondelettes, la quantification et le codage entropique sont réalisés de façon indépendante sur chaque tuile. La tuile d'une composante donnée «tile-component» est l'unité de base de l'image originale ou l'image reconstruite.

Le découpage en tuiles permet de limiter les ressources nécessaires pour le codage en terme de mémoire. Ainsi, comme elles sont reconstruites indépendamment, elles peuvent être utilisées pour décoder des parties spécifiques de l'image au lieu de décoder l'image entière. Les tuiles sont de taille identique excepté peut être au niveau des bordures. Néanmoins, l'étape de découpage en blocs affecte la qualité de l'image reconstruite (figure 3.8). Les tuiles de petites dimensions génèrent plus d'artefacts que les tuiles de plus grandes tailles.



(a)

(b)

(c)

Figure 3.8 Image ski reconstruite (a) sans découpage en tuiles (b découpage en tuiles 128x128 (c) découpage en tuiles 64x64

Le découpage en tuiles est effectué en superposant la grille de référence avec une grille de tuiles de dimensions  $XTsiz * YTsiz$ . Les tuiles sont numérotées de gauche à droite, et de haut en bas en commençant par zéro, sachant que la première tuile a comme point origine  $(XTOsiz, YTOsiz)$  comme le montre la figure 3.9.

En faisant correspondre la position de chaque tuile sur la grille de référence par rapport au système de coordonnées de chaque composante individuelle, un découpage de chaque composante est ainsi obtenue. Supposant qu'une tuile ait les coordonnées  $(tx0, ty0)$  et  $(tx1-1, ty1-1)$ , alors dans le système de coordonnées d'une composante particulière, cette tuile aura les positions respectives suivantes  $(tcx0, tcy0)$  et  $(tcx1-1, tcy1-1)$  avec :

$$(tcx0, tcy0) = \left( \left\lfloor \frac{tx0}{XRsiz} \right\rfloor, \left\lfloor \frac{ty0}{YRsiz} \right\rfloor \right) \dots \quad (3.4)$$

$$(tcx1, tcy1) = \left( \left\lfloor \frac{tx1}{XRsiz} \right\rfloor, \left\lfloor \frac{ty1}{YRsiz} \right\rfloor \right) \dots \quad (3.5)$$

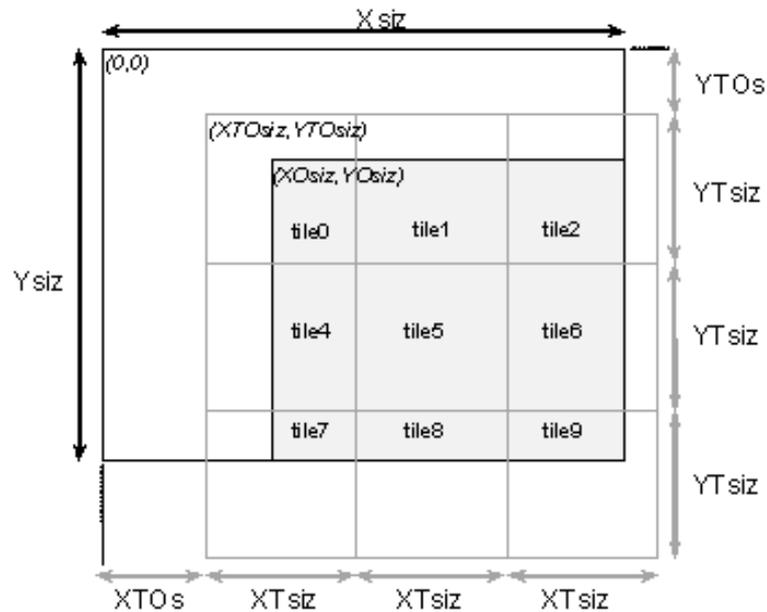


Figure 3.9 Le découpage en tuiles dans la grille de référence

➤ Décalage de la composante continue

Le décalage de la composante continue (DC level shifting) est effectuée avant la transformée en ondelettes et avant la transformée en couleur s'il s'agit d'une image couleur. Elle convertit les valeurs des composantes non signées de l'image en valeurs signées, et ce afin d'assurer que la dynamique des données soit distribuée symétriquement autour de la valeur zéro.

Soit une composante avec  $P$  bits/pixel, les valeurs peuvent être signées ou non dans une plage dynamique de  $[-2^{P-1}, 2^{P-1}-1]$  ou  $[0, 2^{P-1}]$ , respectivement. L'intervalle dynamique nominal dans le cas de valeurs non signées, n'étant pas centré autour de zéro, la valeur à retrancher à chaque pixel est de  $2^{P-1}$ .

Le décodeur réalise l'opération inverse. Si les valeurs de l'image décodée doivent être non signées, la valeur  $2^{P-1}$  doit être ajoutée. Bien évidemment, l'information permettant de savoir si les données à reconstituer en sortie de décodage sont signées ou non doit être conservée.

➤ La transformée couleur

JPEG2000 permet la compression d'images à plusieurs composantes telles que des images couleurs. La représentation chromatique de JPEG2000, utilisée pour les images couleur comprenant trois composantes, est du type (Y, Cb, Cr), bien que d'autres représentations puissent être utilisées par le standard. Selon l'espace couleur de l'image considérée, une transformation des composantes est nécessaire pour changer de représentation chromatique et ainsi passer d'une représentation type (R, G, B) à la représentation type de JPEG2000.

Cette transformation s'obtient très simplement en faisant une combinaison linéaire des composantes RGB. Dans le cas d'une compression sans perte on utilise la formule RCT (manipulation de nombres entiers), pour une compression avec perte on se sert de la formule ICT (nombres réels).

L'ICT (Irreversible Color Transform) une transformation classique du plan (R, G, B) au plan (Y, C<sub>b</sub>, C<sub>r</sub>). Elle est utilisée que lors d'un codage avec pertes avec la transformée (9,7) de Daubechies.

L'ICT est défini comme suit :

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} X \begin{bmatrix} R \\ G \\ B \end{bmatrix} \dots \quad (3.6)$$

La transformation inverse peut être obtenue à partir de l'équation suivante :

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.34413 & -0.71414 \\ 1 & -1.772 & 0 \end{bmatrix} X \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} \dots \quad (3.7)$$

La RCT (Reversible color transform) est utilisée dans le cas de codage sans pertes ou avec pertes mais seulement avec la paire réversible (5,3) en décomposition en ondelettes. Elle est une approximation réversible à la transformation YUV. La RCT présente l'avantage de décorrélation efficace des couleurs, espace de couleurs raisonnable à l'égard du système visuel humain pour la partie quantification. Aussi, une reconstruction parfaite pour une compression sans pertes.

Sa formule est donnée par :

$$\begin{pmatrix} Y \\ V \\ U \end{pmatrix} = \begin{pmatrix} \lfloor \frac{R + 2G + B}{4} \rfloor \\ R - G \\ B - G \end{pmatrix} \dots \quad (3.8)$$

La transformation inverse peut être obtenue à partir de l'équation suivante :

$$\begin{pmatrix} G \\ R \\ B \end{pmatrix} = \begin{pmatrix} Y - \lfloor \frac{U + V}{4} \rfloor \\ V + G \\ U + G \end{pmatrix} \dots \quad (3.9)$$

Le codage des trois composantes obtenues se fera alors indépendamment comme le montre la figure suivante :

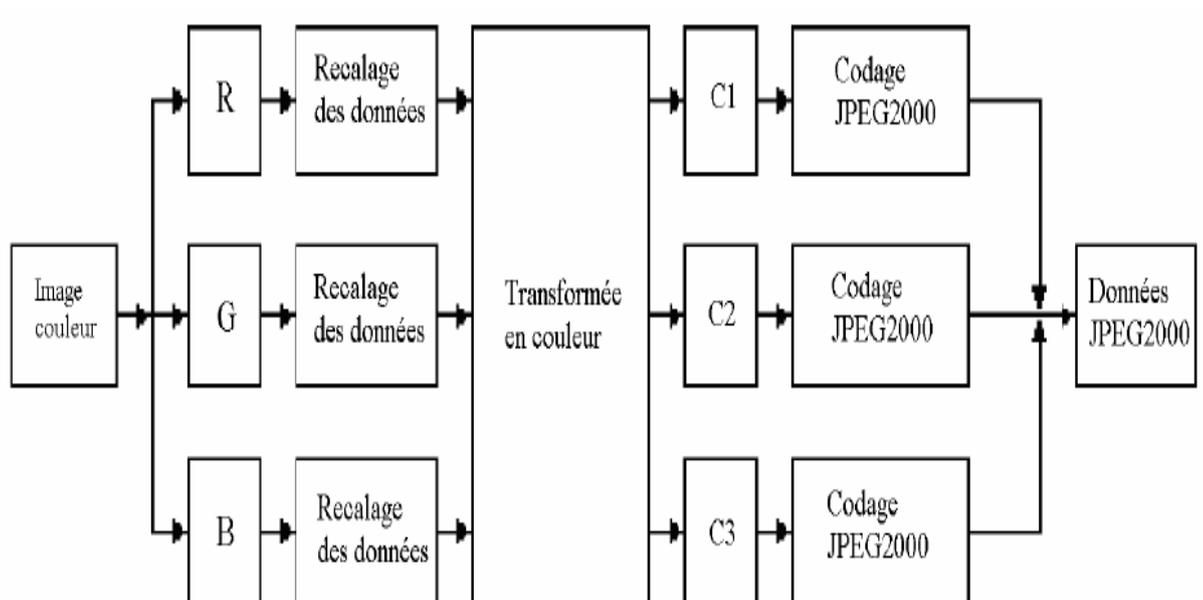


Figure 3.10 Codage des composantes couleurs

#### 3.4.4 La décomposition en ondelettes « DWT »

L'objectif de la DWT est de concentrer l'information de l'image en une zone très localisée de manière à pouvoir ensuite compresser fortement les zones ne contenant que peu d'information. Cela est réalisé en séparant les basses et hautes fréquences de l'image originale grâce à des filtres passe bas et passe haut. Sachant que les basses fréquences correspondent à une version grossière de l'image et que les hautes fréquences contiennent toute l'information sur les détails de l'image. L'énergie de l'image est dite concentrée dans les basses fréquences.

Les filtres autorisés [49] sont : la paire (9,7) **irréversible** de Daubechies qui définit un filtre passe-bas à 9 coefficients et un filtre passe-haut à 7 coefficients, permet des taux de compression plus élevés pour une qualité donnée. La paire (5,3) **réversible** de Le Gall est utilisée pour compresser sans perte.

Dans la première partie du standard JPEG2000, sont permises seulement des décompositions de puissance 2 sous la forme d'une décomposition dyadique. Décompositions successives des basses fréquences (LL) qui permettent de concentrer encore un peu plus l'énergie de l'image comme le montre la figure 3.11.

Ainsi la transformée en ondelettes est utilisée pour l'analyse des tuiles (tiles) pour différents niveaux de décomposition. Chaque niveau de décomposition génère quatre sous-bandes, HH, LH, HL et LL (L pour un filtrage passe-bas et H pour un filtrage passe-haut, la combinaison de ces indices référence le filtrage successif sur les lignes et les colonnes) composées chacune de coefficients, qui décrivent les caractéristiques horizontales et verticales de la tuile originale.

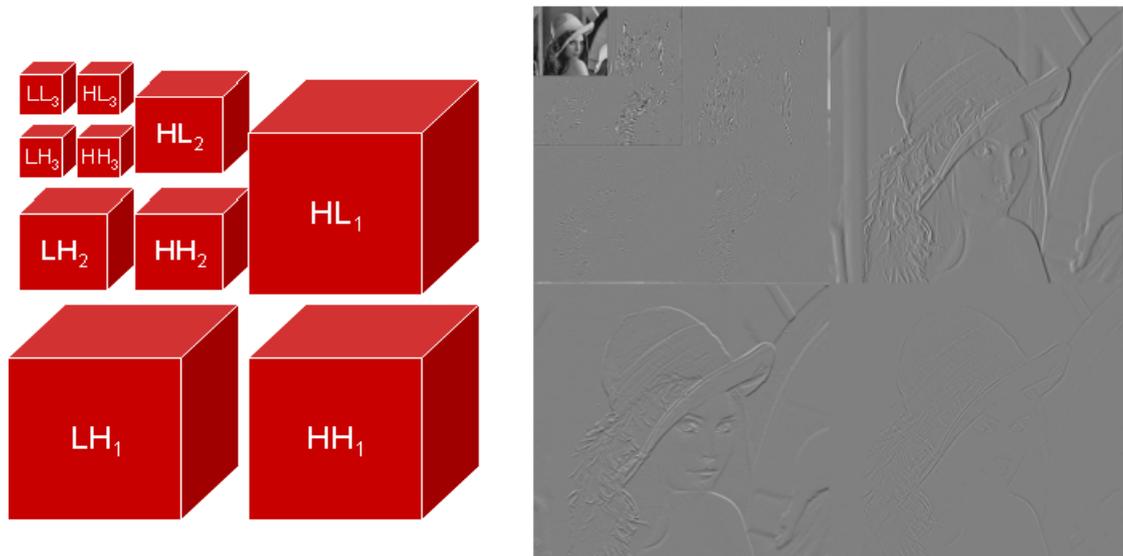


Figure 3.11 Décomposition en ondelettes à trois niveaux

### 3.4.5 La quantification

La précision sur les coefficients d'ondelettes est réduite par la quantification. Cette étape permet, d'une part, de mettre à zéro les coefficients de faibles amplitudes, influant peu sur la qualité de l'image, et fixe aussi le taux de compression minimal de l'image. Cette étape permet donc une plus grande compression par l'élimination des informations visuellement insignifiantes, c'est le codage avec perte.

Les coefficients d'ondelettes sont quantifiés en utilisant une quantification scalaire. Pour chacune des sous-bandes, on utilise des scalaires différents. Mathématiquement, la fonction de quantification est donnée par l'équation suivante :

$$V(x, y) = \lceil |U(x, y)| / \Delta \rceil \times \text{signe } U(x, y) \quad (3.10)$$

Avec :

- $\Delta$  est le pas de quantification,
- $U(x,y)$  le coefficient d'ondelettes à quantifier.
- $V(x,y)$  représente le coefficient d'ondelettes quantifié.

Dans le cas d'un codage sans perte, le pas de quantification est égal à 1. Ce qui signifie qu'aucune quantification n'est appliquée.

### 3.4.6 Le codage entropique

Cette partie représente 70% à 80% du temps total d'exécution. Elle se compose essentiellement de trois étapes :

- Découpage en blocs appelés code-blocks.
- Modélisation des données binaires (modélisation du contexte),
- Codage arithmétique.

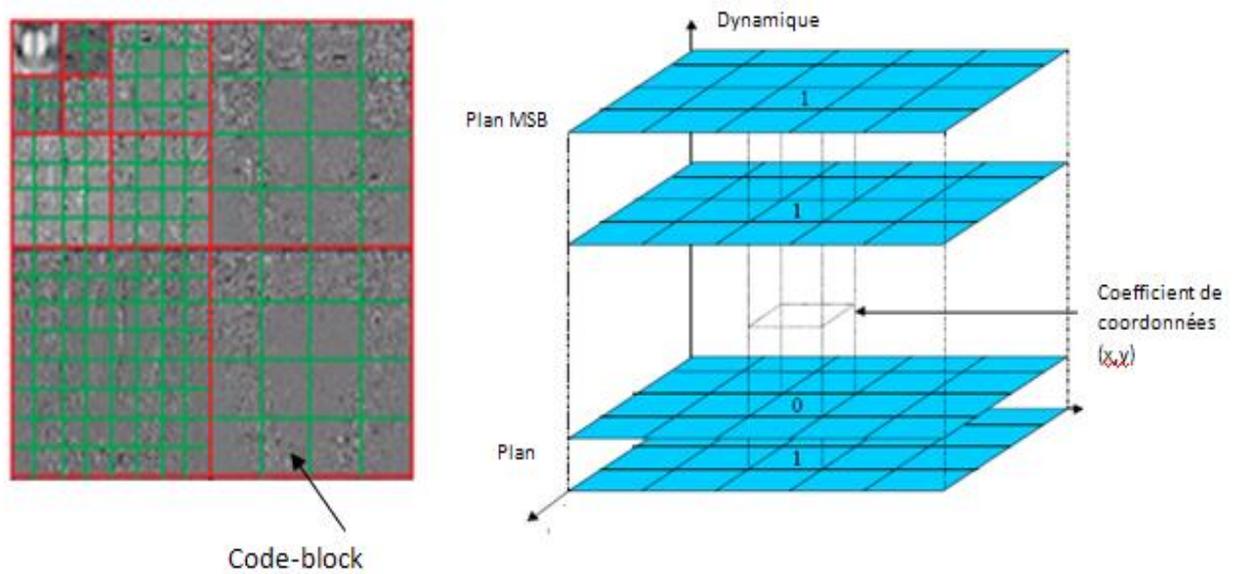
Le découpage en blocs (figure 3.13-a) consiste à grouper les coefficients quantifiés résultants de chaque sous-bande dans des blocs rectangulaires (code blocks) de même taille paramétrable (64x64 max).

Chacun des blocs obtenus est codé indépendamment sans aucune référence aux autres blocs de la même sous-bande ou d'une autre. Ce codage indépendant offre des avantages importants, comme un accès spatial aléatoire au contenu de l'image, une plus grande protection face aux erreurs et un contrôle efficace de débit.

Suite à ce découpage en blocs (code-blocks), vient l'étape de regrouper les blocs en precincts à travers chaque sous bande. Par ce regroupement, on obtient l'ensemble des coefficients d'un niveau de résolution pour une position spatiale donnée. La taille de ces precincts peut varier d'une sous bande à l'autre. Une manière de garder toute cohérence serait de prendre des precincts variant d'un facteur 2 entre deux niveaux de résolution, ce qui correspond au facteur d'échelle de la transformée. Le precinct dans JPEG2000 correspond donc à un regroupement de données compressées appartenant à une même position spatiale dans une résolution donnée (figure 3.12).



Figure 3.12 La notion de precinct dans JPEG 2000



(a)

(b)

Figure 3.13 (a) décomposition en blocs (code-blocks) (b) représentation en bit plane d'un bloc (code-block).

Les blocs (code-blocks) sont en fait des tableaux de coefficients qui peuvent être représentés par un tableau tridimensionnel binaire position-amplitude constitué de  $N_b$  bit-planes où  $N_b$  représente la dynamique de la sous-bande (figure 3.13-b).

L'étape de modélisation des données binaires se base sur le principe du codeur entropique EBCOT (Embedded Block Coding with Optimized Truncation ) [49]. Cet algorithme consiste à coder chaque bloc (code-block), bit-plane par bit-plane, en commençant par les bits de poids fort (MSB). Le codage de chaque bit-plane est quant à lui réalisé stripe par stripe. Le concept de stripe ainsi que le sens du parcours sont illustrés dans la figure 3.14.

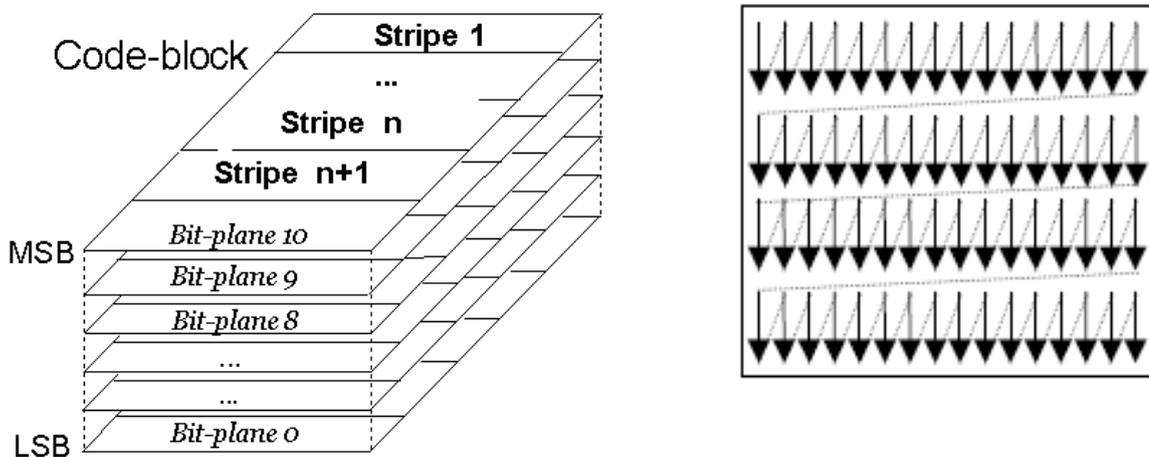


Figure 3.14 Concept de stripe et ordre de balayage des blocs (codes-blocks)

Sur chaque bit-plane sont effectuées trois passes de codage :

- **La passe de signification** : codage des bits (0 ou 1) des échantillons non significatifs ayant au moins 1 voisin significatif.
- **La passe d'affinage** : codage des bits (0 ou 1) des échantillons significatifs non encore codés.
- **La passe de nettoyage** : les bits qui n'ont pas été codés par les deux précédentes passes vont être codés dans cette passe.

Pour le premier bit-plane on applique seulement la passe de nettoyage. Chaque passe produit une paire de données qui sera envoyée au codeur arithmétique MQ afin d'encoder le bit. Cette paire est composée du contexte déterminé par les voisins du bit, ainsi que de la valeur du symbole à encoder.

L'idée fondamentale du codeur arithmétique [49] est d'associer à la séquence de «0» et de «1» à coder un chiffre rationnel appartenant à l'intervalle  $[0,1]$ . Ce chiffre est fonction de la probabilité d'occurrence de chacun des deux symboles possibles. Il est à noter que la représentation binaire du chiffre rationnel obtenu est moins longue que la séquence binaire de départ. Par conséquent, puisque l'on peut retrouver la séquence binaire originale à partir du chiffre rationnel, nous avons bien réussi à compresser un volume donné d'informations en une séquence plus courte.

#### 3.4.7 Allocation de débit

A ce stade-ci, toutes les données de l'image sont disponibles sous la forme d'une longue séquence binaire issue du MQ. L'allocation de débit consiste à trouver, dans la séquence correspondant à chaque bloc (code-block), le point de troncature optimal permettant d'atteindre le débit spécifié dans les paramètres de codage. En pratique, trois points de troncature sont possibles par bit-plane [50], correspondant aux trois passes réalisées par l'EBCOT lors de l'étape de codage entropique.

Le taux de compression échelonné est réalisé à travers plusieurs couches (**layer**). Pour une image, les données codées sont structurées suivant L couches. Les passes de codage contenant les données les plus importantes sont incluses dans les couches inférieures, alors que les passes de codage associés aux détails les plus fins sont inclus dans les couches supérieures. Plus on augmente le nombre des couches, plus la qualité de l'image reconstruite augmente.

Dans le cas d'une compression avec perte, certaines passes de codage seront écartées. Ainsi, elles ne seront pas incluses dans les couches [49]. Dans ce cas, le contrôleur du taux décide quelles passes seront incluses dans le flux de donnée final. Dans le cas de compression sans perte, toutes les passes de codage sont incluses dans les couches. La figure 3.15 représente la contribution de chaque bloc (code-block) dans chaque couche.

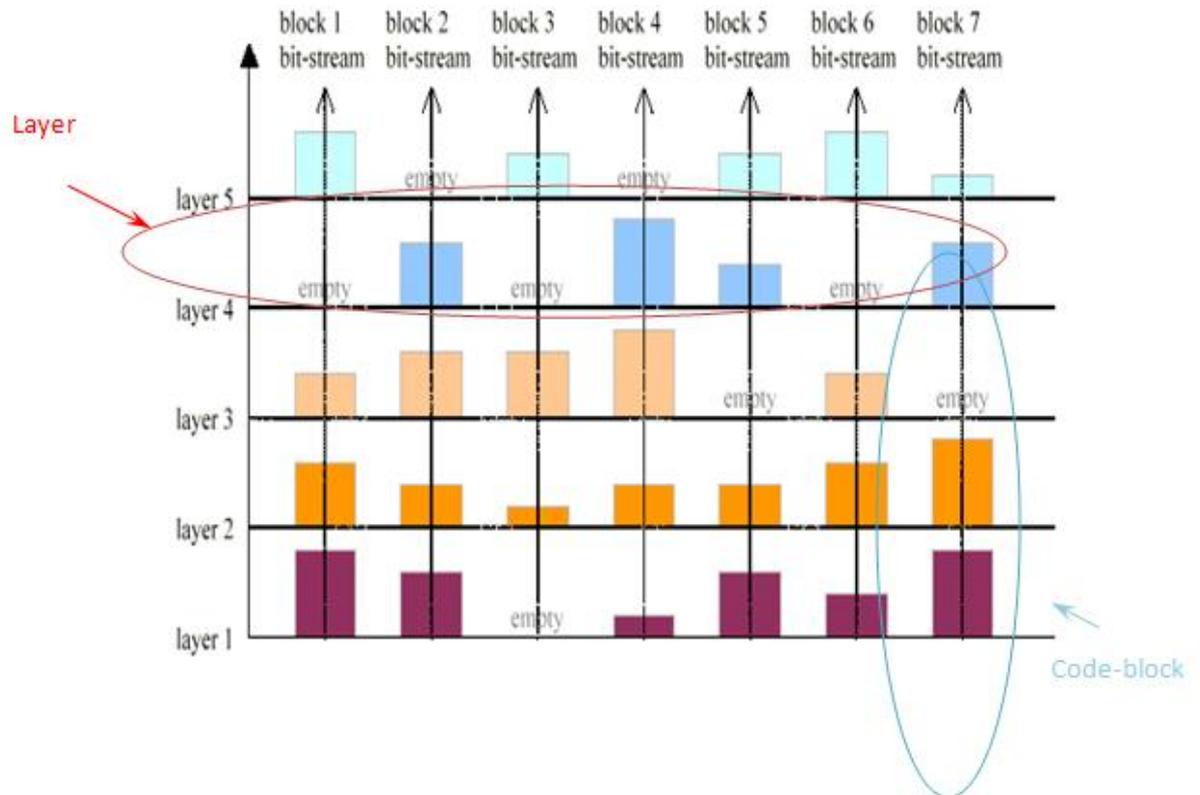


Figure 3.15 Concept de couches de qualités 'Layer'

#### 3.4.8 Organisation du flux compressé

L'aboutissement du codage est la création du flux de données compressées JPEG2000 nommé codestream. Celui-ci est constitué de marqueurs et de segments contenant la totalité de l'information nécessaire au décodage.

Le flux compressé JPEG2000 est composé comme indiqué en figure 3.16, d'une entête principale, une suite de données compressées et un marqueur indiquant la fin du flux EOC.

L'entête principale contient un ensemble de marqueurs indispensables pour le processus inverse de décodage tel que :

- SIZ marqueur qui identifie les données globales sur l'image d'origine (hauteur, largeur, type d'image, nombre de composantes,...) ainsi que les paramètres relatifs à l'étape prétraitement.
- COD marqueur englobant tous les paramètres sélectionnés à partir de l'étape de DWT.
- RGN marqueur pour la région d'intérêt 'ROI'.
- COM marqueur pour les commentaires ajoutés au flux compressé.

Les données sont une succession de tuiles (tiles), où chaque tuile est précédée d'une entête.

Le flux compressé peut être vu comme une hiérarchisation en trois niveaux principaux : tuile (tile), paquet/precinct et finalement blocs (codes-blocks).

**Niveau tuile (tile) :** contient des données compressées résultant de l'étape du découpage en tuiles (tiling). Chaque tuile est précédée d'une entête. Cette dernière contient :

- un SOT 'Start Of Tile ' qui indique le début de l'entête.
- un SOD 'Start Of Data ' indiquant la fin de l'entête et le début des données compressées
- d'autres marqueurs peuvent être inclus entre le SOT et SOD tel que le marqueur COD. Ils sont inclus si certains paramètres de compression sont spécifiques à une tuile donnée.

**Niveau paquet / precinct :** les données compressées d'une tuile sont en fait une suite de paquets ou precincts. Par défaut, le flux compressé JPEG2000 contient des paquets et non pas des precincts, le precinct étant une option dans JPEG 2000. La différence entre un paquet et un precinct réside dans le fait qu'un paquet représente toute une résolution (issue de la DWT), le precinct par contre n'est qu'un ensemble de données de mêmes positions spatiales dans une résolution. Une entête précède chaque paquet ou precinct. Elle commence toujours par '0/1' indiquant si le paquet est vide (0) ou pas (1). Elle peut contenir deux marqueurs optionnels SOP 'Start Of Paquet' et EPH 'End of Paquet Header', indiquant le début et fin de l'entête respectivement. Les deux marqueurs présents ou pas, l'entête contient toutes les informations relatives au codage entropique des blocs (codes-blocks). Ces informations sont expliquées au niveau bloc (code-block).

**Niveau bloc (code-block) :** c'est le bloc qui contient réellement les données entropiques issus du codage. Au niveau de l'entête du paquet/precinct, existe les informations concernant le codage entropique de tous les blocs (code-blocks) appartenant au paquet/precinct. Ces informations sont :

- INC : le bit inclusion
- MSB : le nombre de bits plans MSB à zéro.
- C.PASS : le nombre de passes effectuées au niveau de l'EBCOT.
- LENGTH CBLK : le nombre d'octets du bloc (code-block).

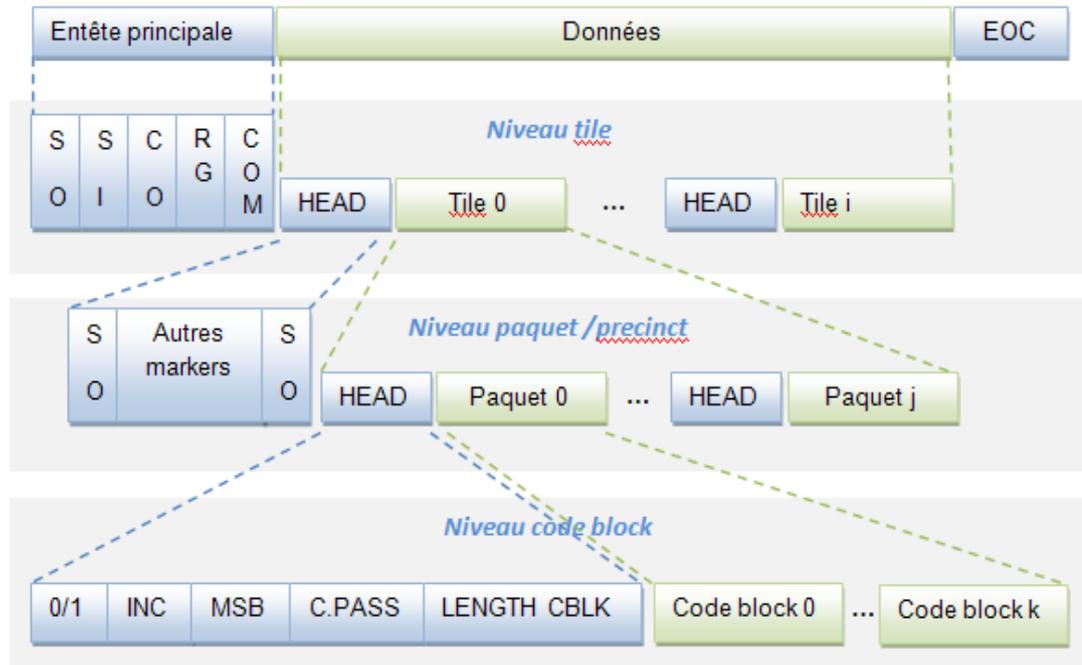


Figure 3.16. Le flux compressé dans la norme JPEG2000

### 3.5 Conclusion

JPEG2000 est une norme de compression très connue par « compress one and decompress many ». En effet, de part sa structure, le même flux compressé JPEG2000 peut être décompressé de plusieurs façons (qualité, résolution,...). JPEG2000 permet de couvrir des domaines très variés par l'exploitation des différentes fonctionnalités à travers les douze parties qu'il définit.

La notion « JPEG2000 » a donc été introduite, en précisant son principe, ses objectifs, ses domaines d'applications ainsi que son algorithme de compression.

Dans le chapitre suivant, nous allons présenter les techniques d'insertion de données cachées implémentées dans notre projet et les résultats obtenus.

## CHAPITRE 4. TECHNIQUES D'IDC IMPLÉMENTÉES:TESTS ET RÉSULTATS

### 4.1 Introduction

Il n'existe pas aujourd'hui de méthodes d'IDCs réversibles qui soient à la fois totalement réversibles, imperceptibles et qui puissent insérer une grande quantité d'information.

Logiquement, plus la technique d'insertion est inappropriée par rapport à la taille importante des informations à insérer, plus la dégradation apportée à l'image sera visible. De plus, si le processus d'insertion modifie perceptuellement l'image, il sera plus difficile de retrouver l'image originale et la méthode ne sera plus réversible.

La problématique du choix d'une méthode d'IDC réversible se dégage : il s'agit de faire un compromis entre la taille de la donnée à insérer, l'imperceptibilité et la réversibilité.

Ce choix n'est pas à priori évident et dépend beaucoup du type de support que nous désirons marquer. Ainsi, pour les images médicales, la réversibilité sera le facteur dominant. Il faudra cependant que la taille de la donnée ne soit pas trop réduite pour qu'elle puisse contenir les informations nécessaires du patient et que l'image marquée reste fidèle à l'image originale pour ne pas fausser le diagnostic. En effet, plus nous insérons d'informations, plus la visibilité risque d'être réduite, plus l'image sera dégradée.

Dans ce chapitre nous présentons les techniques d'insertion de données cachées qui ont pour objectif la vérification de l'intégrité de l'image médicale et la préservation de la confidentialité des données du patient. Cette partie, nous permettra d'évaluer leurs intérêts et leurs limites.

### 4.2 Présentation des approches développées

L'implémentation logicielle des différents algorithmes présentés dans ce mémoire sont développée sous MATLAB<sup>®</sup> 7.0, à l'exception de la technique de Katsutoshi [34] qui a été intégrée dans le codeur JPEG2000.

Les tests ont été effectués sur trois types d'images médicales (IRM, radiographie) et une image optique Lena, codées sur 256 niveaux de gris, format BMP, de taille 256 x 256. L'objectif est d'évaluer la qualité de l'image marquée, de vérifier la sensibilité de l'approche par rapport à l'attaque par compression JPEG2000 (Figure 4.1) et surtout l'aptitude de l'approche à reconstruire l'image originale à partir de l'image marquée (réversibilité).

La mesure objective utilisée pour évaluer la dégradation des images après insertion est le PSNR.

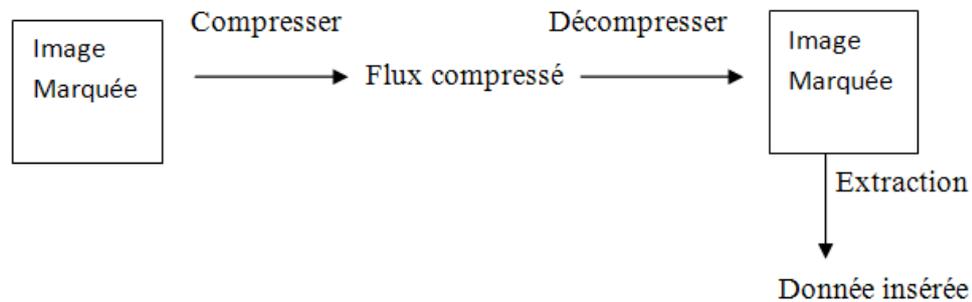


Figure 4.1. Attaque par compression JPEG2000

#### 4.2.1 L'approche de Katsutoshi

La méthode proposée par Katsutoshi [34], consiste en l'insertion de données dans la deuxième couche (layer) (voir chapitre 3.4.7) du flux compressé d'une image donnée. Elle a pour but de préserver la structure et la taille du flux compressé après insertion, et donc n'importe quel décodeur peut facilement reconstruire l'image en question. Cette technique peut tout aussi bien être appliquée dans les deux modes de compression JPEG2000, avec et sans pertes. Elle se résume en deux étapes, insertion et extraction de données cachées :

L'insertion commence par compresser l'image d'origine avec un codeur JPEG2000. La compression est faite en spécifiant deux couches (layers). Après localisation de la deuxième couche dans le flux compressé, l'insertion se fait en parcourant cette dernière de droite à gauche, et en procédant au remplacement d'un octet de la deuxième couche par un octet de la donnée à insérer jusqu'à épuisement de toute la donnée (Figure 4.2.a). Dans le cas où la taille de la donnée est plus petite que la taille de la deuxième couche, le reste des octets de la couche reste inchangé. De ce fait, il est à noter que La taille de la donnée maximale à insérer est égale à la taille de la deuxième couche dans le flux compressé.

L'extraction (Figure 4.2.b) consiste en l'analyse du flux récupéré en identifiant la deuxième couche. Le parcours de la couche se fait de droite à gauche en se limitant par la taille de la donnée insérée. La taille devant être indiquée au destinataire. La reconstruction de l'image transmise par décompression doit être précédée par la suppression de la deuxième couche du flux compressé. Ceci a pour but, d'éliminer la distorsion pouvant être induite par la donnée cachée insérée.

Cependant, cette technique ne peut être appliquée aux images de grande taille. Sachant que la deuxième couche pourrait contenir des données importantes dans le cas d'une image de grande taille, l'insertion à ce niveau altère sa texture. Par conséquent et pour le développement de la technique, nous avons varié le nombre de couches en fonction de la taille des images.

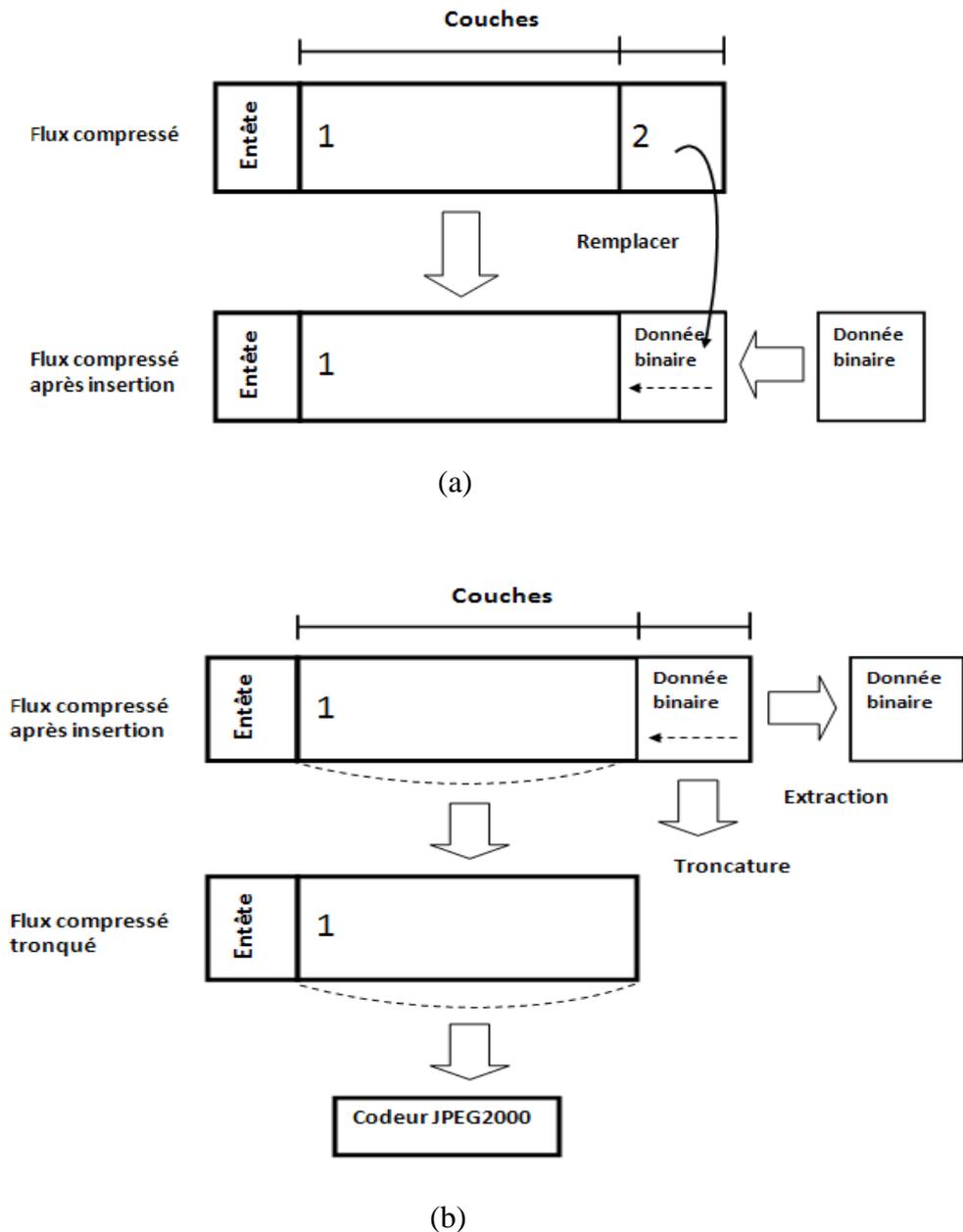


Figure 4.2 Schéma d'insertion de données cachées proposé par Katsutoshi [32] : (a) Insertion (b) Extraction

➤ Tests et Résultats

Diverses expériences ont été faites pour évaluer les performances de la technique proposée. Des images médicales, d'autres optiques ont été prises comme images originales. Les paramètres retenus lors des tests pour l'ensemble des images est résumé dans le tableau 4.1. Aussi, Le Kakadu2.2.3 a été utilisé comme Codeur JPEG 2000.

Paramètres de compression	Images médicales	Images optique
Taille	256x256x8bits	256x256x8bits
Compression	Sans pertes : filtre Gall	Avec pertes : filtre Deaubechies
Nombre de décompositions	5	5
Nombre de layers	2	2

Tableau 4.1 : Paramètres de compression

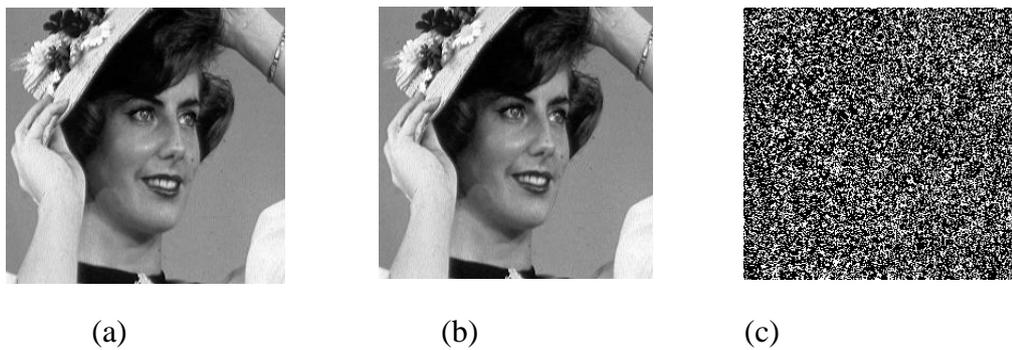


Figure 4.3 (a) Image optique originale (b) Image après insertion (c) Image différence

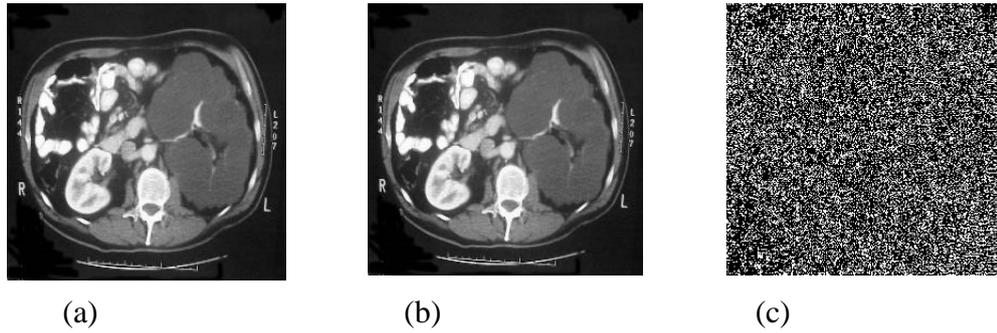


Figure 4.4 (a) Image IRM originale (b) Image après insertion (c) Image différence

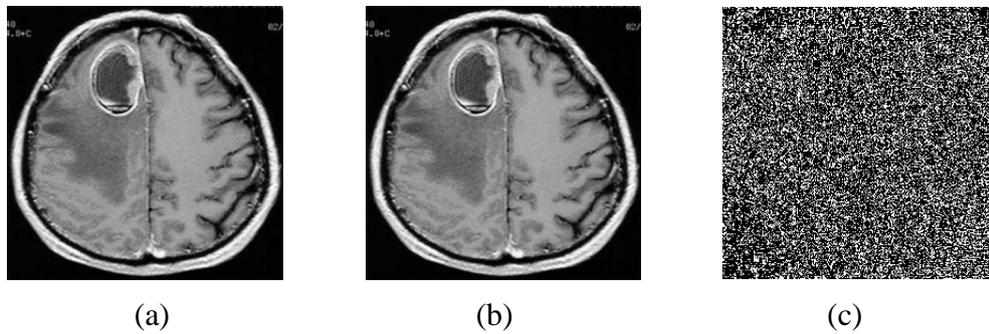


Figure 4.5 (a) Image IRM originale (b) Image après insertion (c) Image différence

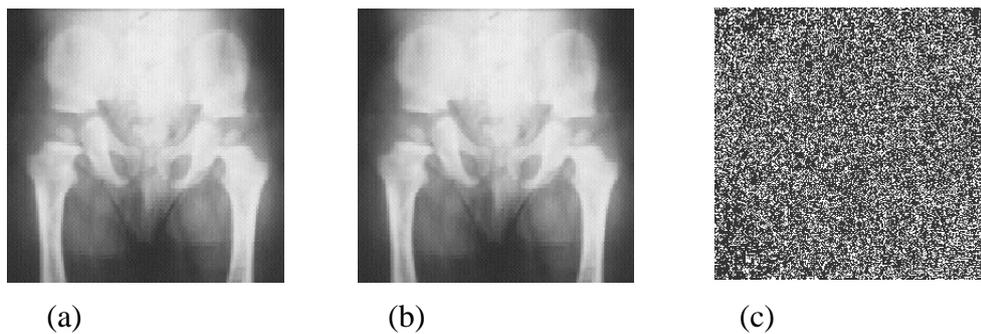


Figure 4.6 (a) Image radiologique originale (b) Image après insertion (c) Image différence

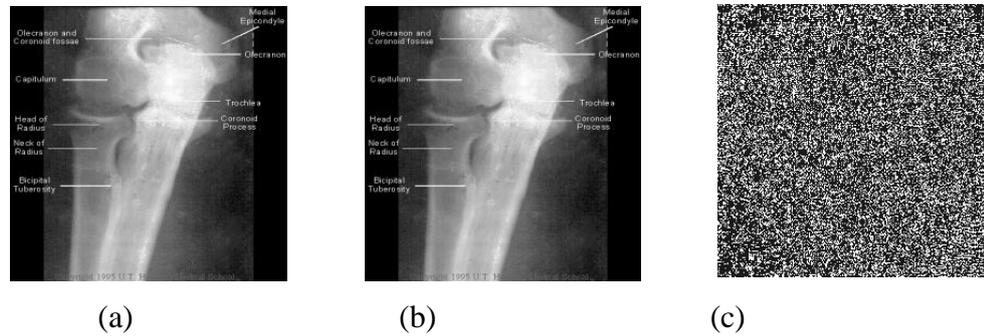


Figure 4.7 (a) Image radiologique originale (b) Image après insertion (c) Image différence

Evaluation	Image IRM	Image Lena
PSNR	46,55 dB	45,99 dB
Capacité d'insertion	10475 x8bits	11305 x8bits
Taux de compression	47,69 %	41,53%

Tableau 4.2 Résultats d'insertion

Les résultats du tableau 4.2 montrent que la technique proposée permet d'insérer un volume de données considérable tant sur les images médicales qu'optiques. De plus, la dégradation causée par l'insertion de données cachées est imperceptible (le PSNR est au dessus de 45 dB).

La quantité de données insérées dans le flux compressé de l'image optique est plus importante que celle insérée dans le flux compressé de l'image médicale. Ceci est dû au fait que la deuxième couche de l'image optique est plus volumineuse que le flux de l'image médicale. Toutefois, il est à noter que ce constat n'a aucune relation avec les paramètres de tests choisis mais est totalement lié à la texture des images tests.

Les mêmes tests ont été effectués sur des images volumineuses. Les résultats obtenus dans le tableau 4.3 montrent qu'il y a une très grande corrélation entre la taille de l'image d'origine et le nombre de couches à spécifier. En effet, deux couches dans le cas des images de grande taille ne permettent pas de préserver la qualité de l'image restituée après insertion.

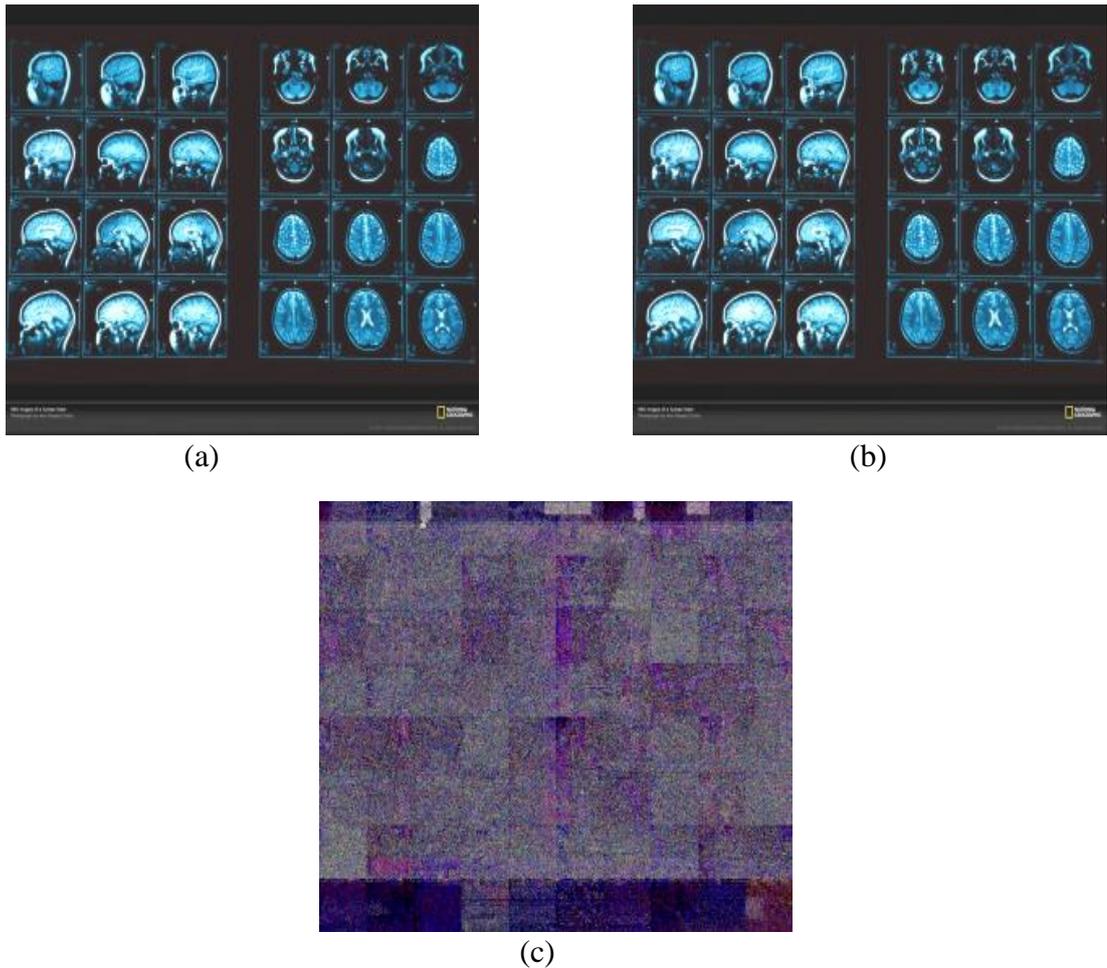


Figure 4.8 (a) Image IRM volumineuse (b) Image après insertion (c) Image différence

PSNR	39,16 dB
Capacité d'insertion	380825 x8bits
Taux de compression	66,10%

Tableau 4.3 Résultats d'insertion pour une image IRM volumineuse

A titre d'exemple, pour une image de taille 1280x1024x8bits (Figure 4.8. a) utilisant les mêmes paramètres restants du tableau 4.1, les performances obtenues pour deux couches montrent que la dégradation de l'image (Figure 4.8 .b) après insertion est sensible, (quantitativement le PSNR est de 39,16 dB). L'image différence calculée entre l'image originale et l'image après insertion montre clairement que l'image d'origine a été modifiée. De ce fait, il est nécessaire d'augmenter le nombre de couches en fonction de la taille de l'image (pour la même image il faut aller jusqu'à cinq couches).

#### 4.2.2 L'approche par Expansion de différence

Les techniques d'insertion de données cachées réversibles appelées aussi techniques d'insertion sans pertes (Lossless) insèrent de manière invisible et surtout réversible la donnée avec une dégradation très faible de l'image support.

L'approche commune des techniques d'insertion de données réversibles à hautes capacités, est de sélectionner une zone d'insertion (exemple les bits les moins significatifs de quelques pixels) dans une image, et insérer la donnée cachée ainsi que les bits originaux nécessaires pour retrouver l'image originale.

Une technique d'insertion de données cachées réversible basée sur l'expansion de différence (DE) sera présentée dans la section suivante. Comparée à d'autres techniques d'insertion de données cachées réversibles, cette méthode offre une grande capacité d'insertion avec une dégradation très faible de l'image support et une faible complexité de calcul. Nous introduisons ici, le schéma le plus représentatif de cette technique, à savoir le schéma de Tian [14] ainsi qu'une version modifiée implémentée dans le cadre de nos travaux.

##### ➤ Processus d'insertion Basé sur la DE

La méthode d'expansion de différence (DE) est basée sur l'idée de modifier l'amplitude de la différence d'intensité entre une paire de pixels. La valeur moyenne des deux pixels est maintenue constante, tandis que la différence entre eux est modifiée. La transformée entière est utilisée pour obtenir la moyenne et la différence des pixels.

Le processus d'insertion par la différence d'expansion se base sur six étapes :

- Calculer la valeur moyenne et la différence entre deux valeurs de pixels ;
- Partitionner les valeurs de différence en quatre parties ;
- Créer une carte de localisation (Location Map) ;
- Localiser les valeurs des LSBs à échanger ;
- Insertion par remplacement ;
- Appliquer la transformée inverse.

- Calcul de la valeur moyenne et la différence entre deux valeurs de pixels

L'image originale est regroupée en paires de pixels voisins. Par exemple, deux valeurs en continu dans le sens horizontal  $(i, 2j-1)$  et  $(i, 2j)$  peuvent être sélectionnées comme une paire. Il ya quatre paires différentes : horizontale, verticale, et deux directions diagonales.

Dans ce schéma, on définit d'abord une transformation entière qui opère sur deux pixels adjacents  $x$  et  $y$  en niveaux de gris,  $0 \leq x, y \leq 255$  par :

$$l = \left\lfloor \frac{x+y}{2} \right\rfloor, \quad h = x - y \quad (4.1)$$

Où le symbole  $\lfloor \ ]$  est la fonction qui signifie « le plus grand entier inférieur ou égal ».

et la transformation inverse :

$$x = l + \left\lfloor \frac{h+1}{2} \right\rfloor, \quad y = l - \left\lfloor \frac{h}{2} \right\rfloor \quad (4.2)$$

De (4.2), afin de prévenir les problèmes de débordement et dépassement, à savoir, restreindre  $x, y$ , dans l'intervalle  $[0, 255]$ , ceci est équivalent à :

$$0 \leq l + \left\lfloor \frac{h+1}{2} \right\rfloor \leq 255 \quad ; \quad 0 \leq l - \left\lfloor \frac{h}{2} \right\rfloor \leq 255 \quad (4.3)$$

De (4.3) on obtient :

$$\begin{cases} \lfloor h \rfloor \leq 2(255 - l) - 1 \\ \lfloor h \rfloor \leq 2l + 1 \\ \lfloor h \rfloor \leq 2(255 - l) \\ \lfloor h \rfloor \leq 2l \end{cases} \quad (4.4)$$

Puisque  $l$  et  $h$  sont des entiers, on peut déduire que les inégalités ci-dessus sont équivalentes aux :

$$\begin{cases} \lfloor h \rfloor \leq 2l + 1 \\ \lfloor h \rfloor \leq 2(255 - l) \end{cases} \quad (4.5)$$

Donc on déduit que :

$$\begin{cases} |h| \leq 2l + 1 & \text{si } 128 \leq l \leq 255 \\ |h| \leq 2(255 - l) & \text{si } 0 \leq l \leq 128 \end{cases} \quad (4.6)$$

- Partitionner les valeurs de différence en quatre parties

Comme nous insérons un bit  $b$  dans la valeur de différence  $h$  par la DE, la nouvelle valeur de différence élargie  $h'$  sera :

$$h' = 2 \times h + b \quad (4.7)$$

Selon la condition (1.6), et afin de prévenir le problème de débordement (restreindre  $x$ ,  $y$  dans l'intervalle  $[0, 255]$ ,  $h'$  devrait satisfaire

$$|h'| \leq \min(2(255 - l), 2l + 1) \quad (4.8)$$

L'équation peut être reformulée de la manière suivante :

- la valeur de différence  $h$  est expansible si :

$$|2h + b| \leq \min(2(255 - l), 2l + 1) \quad (4.9)$$

pour les deux valeurs  $b=0$  et  $b=1$  où  $b$  représente le LSB( $h$ ). Dans ce cas, la valeur de  $b$  est insérée en appliquant la DE.

Comme chaque entier peut être représenté par la somme d'un multiple de 2 et de son LSB, pour la nouvelle valeur de différence élargie  $h'$

$$h' = 2 \times \left\lfloor \frac{h'}{2} \right\rfloor + LSB(h')$$

Avec  $LSB(h') = 0$  ou  $1$ . Si nous modifions son LSB alors

$$g = 2 \times \left\lfloor \frac{h'}{2} \right\rfloor + b'$$

Avec  $b'=0$  ou  $1$ .

$$\begin{aligned} |g| &= \left| 2 \times \left\lfloor \frac{h'}{2} \right\rfloor + b' \right| = \left| 2 \times \left\lfloor \frac{2 \times h + b}{2} \right\rfloor + b' \right| = |2 \times h + b'| \\ &\leq \min(2(255 - l), 2l + 1) \end{aligned}$$

Ainsi, le LSB de la valeur de différence  $h'$  peut être modifiée sans causer de débordement, dans ce cas la valeur de différence est dite modifiable.

➤ la valeur de différence  $h$  est modifiable si :

$$\left| 2 \left\lfloor \frac{h}{2} \right\rfloor + b \right| \leq \min(2(255-1), 2l+1) \quad (4.10)$$

pour les deux valeurs  $b=0$  et  $b=1$ . Dans ce cas, la valeur de  $b$  peut être insérée en modifiant la valeur du LSB pour être dans l'intervalle  $[0, 255]$ .

Ainsi, quatre ensembles disjoints sont créés pour la valeur de différence : EZ, EN, CN, et NC.

- EZ : contient toutes les valeurs expansibles de  $h=0$  et de  $h=-1$
- EN : contient toutes les valeurs  $h$  expansibles qui n'appartiennent pas à EZ,  $h \neq 0$  et  $h \neq -1$
- CN : contient toutes les valeurs de  $h$  modifiables qui n'appartiennent pas à (EZ et EN)
- NC : contient toutes les valeurs de  $h$  non-modifiables

Chaque valeur de différence sera dans un seul de ces quatre ensembles. Ainsi, si toutes les valeurs de différence expansibles sont toujours modifiables, alors l'ensemble des valeurs de différence modifiables sera  $EZ \cup EN \cup CN$ .

- Création de la carte de localisation

Dans le but d'extraire les données insérées et de restaurer les valeurs d'origine, le décodeur a besoin de savoir quelles valeurs de différence ont été sélectionnées pour la DE. Ainsi, nous avons besoin d'intégrer des informations de localisation, de telle manière que le décodeur peut y accéder et s'en servir pour le décodage. Pour ce faire, une carte de localisation, qui contient les informations de localisation de toutes les valeurs de différence expansible sélectionnées est créée et insérée avec la donnée cachée.

Pour la création de la table de localisation, nous admettons que chaque valeur de différence  $h$  dans EZ sera sélectionnée pour lui appliquer la DE. Par contre, dans EN, quelques valeurs seulement seront sélectionnées pour la DE, puisque elles dépendent de la taille de la donnée (payload) à insérer. Alors, pour cette raison, l'ensemble EN sera subdivisé en deux ensembles  $EN_1$  et  $EN_2$  pour les valeurs sélectionnées et non sélectionnées respectivement. Alors, pour une différence  $h$  dans EZ et  $EN_1$  une valeur de « 1 » sera assignée dans la Table de localisation.

Par contre, si elle est dans EN<sub>2</sub>, EN et NC, une valeur « 0 » lui sera assignée. (Voir le tableau 4.4).

Catégorie	Ensemble original	Valeur originale	Table de localisation	Nouvelle valeur	Nouvel ensemble
Changeable	EZ ou EN <sub>1</sub>	h	1	$2 \times h + b$	CH
	EN <sub>2</sub> ou CN	h	0	$2 \times \left\lfloor \frac{h}{2} \right\rfloor + b$	
Non changeable	NC	h	0	h	NC

Tableau 4.4 les valeurs de différence pour l'insertion

- Localisation des valeurs LSBs

Avec la carte de localisation, l'encodeur et le décodeur partagent la même information sur laquelle des valeurs de différence ont été sélectionnées pour l'expansion de différence. Bien qu'il est simple pour le codeur d'insérer la carte de localisation, le décodeur a besoin de savoir où collecter sa valeur pour la décoder.

L'encodeur modifie toutes les valeurs changeantes, soit par rajout d'une nouvelle valeur LSB par la technique DE, ou bien par modification. Par conséquent, pour les valeurs remplacées, les valeurs de h initiales doivent être connues pour le processus de décodage, afin de garantir une récupération exacte de l'image d'origine. Dans ce cas, les valeurs LSBs d'origine sont collectées dans EN<sub>2</sub> et CN. Pour chaque h dans EN<sub>2</sub> ou CN, LSB (h) seront collectées dans un flux de bits C. Cependant, lorsque  $h = 1$  ou  $h = -2$ , aucune information n'est sauvegardée, car de même que les LSBs d'origine, ils peuvent être déterminés par la carte de localisation.

- Insertion par remplacement

La table de localisation est compressée sans pertes par le codeur JBIG2 (Joint Bi-level Image Experts Group, voir Appendice A) ou bien par le codage RLE (Run-Lenght coding, voir Appendice A). Le flux de bits résultant sera noté L. Ainsi, L, C, et P seront combinés ensemble pour former un seul flux de bits B où P est la taille de la donnée à insérer (payload).

Le flux de bits B ainsi constitué se présente comme suit:

$$B = L \cup C \cup P = b_1 b_2 b_3 \dots \dots b_m \quad (4.11)$$

où  $b_i = 0$  ou  $1$ . Le  $m$  est la longueur de B. les valeurs de B seront insérées dans la matrice des valeurs de différence.

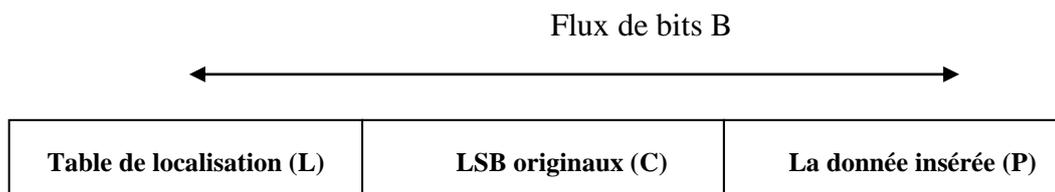


Figure 4.9 La valeur du flux de bits à insérer

Enfin, nous insérons de la manière suivante :

```

Set i=1 and j=0.
While (i ≤ m)
    j = j+1
    If  $h_j \in (EZ \cup EN1)$ 
         $h_j = 2 \times h_j + b_i$  .
        i = i+1 .
    Elsif  $h_j \in (EN2 \cup CN)$ 
         $h_j = 2 \times \left\lfloor \frac{h_j}{2} \right\rfloor + b_i$  .
        i = i+1 .
End

```

A la fin de l'insertion de tous les bits de  $B$ , nous appliquons la transformée inverse (4.2) pour obtenir l'image marquée.

Dans [14], Tian propose d'appliquer cette technique pour une insertion multicouches. En effet, l'image peut être, définie sous formes de couches (layers), ensuite insérée avec la DE dans chaque couche de manière à avoir une capacité d'insertion plus grande.

Dans ce cas, une entête  $H$  de 16 bits est insérée avant la table de localisation  $L$ , pour indiquer au décodeur dans quelle couche nous avons effectué l'insertion. Ainsi, le flux de bits  $B$  devient :

$$\mathbf{B} = H \cup L \cup C \cup P = b_1 b_2 b_3 \dots \dots b_m \quad (4.12)$$

- La transformée inverse

Suite à l'insertion des valeurs du flux  $\mathbf{B}$  dans la matrice des valeurs de différence, on calcule les nouvelles valeurs des pixels en appliquant les équations définies dans (4.2) pour obtenir enfin une nouvelle image avec les données cachées qu'on appellera image marquée. Ainsi, nous pouvons résumer le processus d'insertion de la technique DE avec toutes les étapes nécessaires pour son implémentation par l'organigramme figure 4.10.

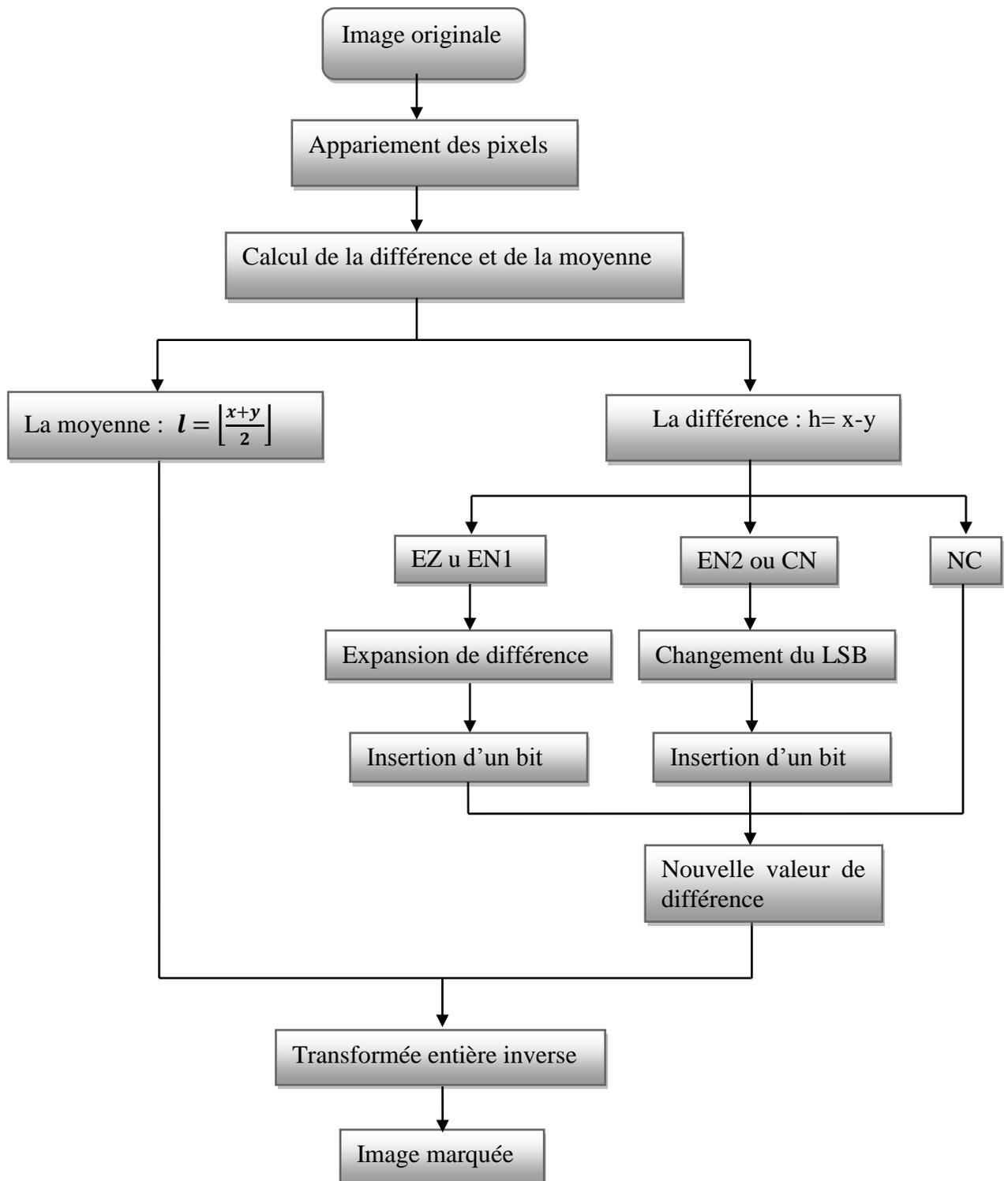


Figure 4.10 Organigramme d'Insertion

➤ Processus d'extraction

En se référant au tableau 4.4, le flux B peut être récupéré en recueillant les LSBs des valeurs de différence modifiables. Ainsi, le décodage du flux B, le flux L de la table de localisation, le flux C des valeurs des LSBs originaux, et la donnée à insérer (payload), permet de retrouver l'image originale.

Le processus de décodage suit le même principe que le processus d'insertion et se résume en cinq étapes :

- Calculer les valeurs moyennes, et de différences.

Dans cette étape, le calcul s'effectue pour l'image marquée de la même manière que pour l'insertion.

- Créer deux ensembles disjoints pour les valeurs de différences CH et NC.

CH : contient tous les h modifiés

NC : contient tous les h non modifiés.

Il faut noter que dans l'extraction, nous n'examinons pas l'expansibilité, et l'ensemble CH doit être identique à l'ensemble  $EZ \cup EN \cup CN$  (qui est un ensemble invariant contenant les informations des valeurs de différence modifiable) et qui représente la clé pour la reconstruction exacte.

- Collecter les LSBs de toutes les valeurs de h dans CH et former le bitstream B.
- Décoder la table de localisation du flux B avec JBIG2.

Comme le flux de bits compressé par JBIG2 a un symbole de fin de message à son extrémité, le décodeur peut identifier de manière exacte le dernier bit de la carte de localisation dans le flux compressé. Supposons que les premiers s bits en B est le flux de bits de la table de localisation L en incluant le dernier bit symbole, la restauration de l'image originale s'effectue alors selon l'algorithme suivant :

```

Set  $i = s + 1$ .
For  $j = 1 : n$ 
If  $h_j \in CH$ 
  If the location map value at  $h_j$  is 1
     $h_j = \lfloor h_j/2 \rfloor$ .
  Else
    If  $(0 \leq h_j \leq 1)$ 
       $h_j = 1$ .
    ElseIf  $(-2 \leq h_j \leq -1)$ 
       $h_j = -2$ .
    Else
       $h_j = 2 \times \lfloor h_j/2 \rfloor + bi$ 
       $i = i + 1$ .
End

```

– Appliquer la transformée inverse

Après avoir retrouvé les valeurs de  $h$ , nous appliquons la transformée inverse définie dans (4.2) pour retrouver l'image originale.

Ainsi, nous pouvons résumer le processus d'extraction de la technique DE avec toutes les étapes nécessaires pour son implémentation par l'organigramme figure 4.11.

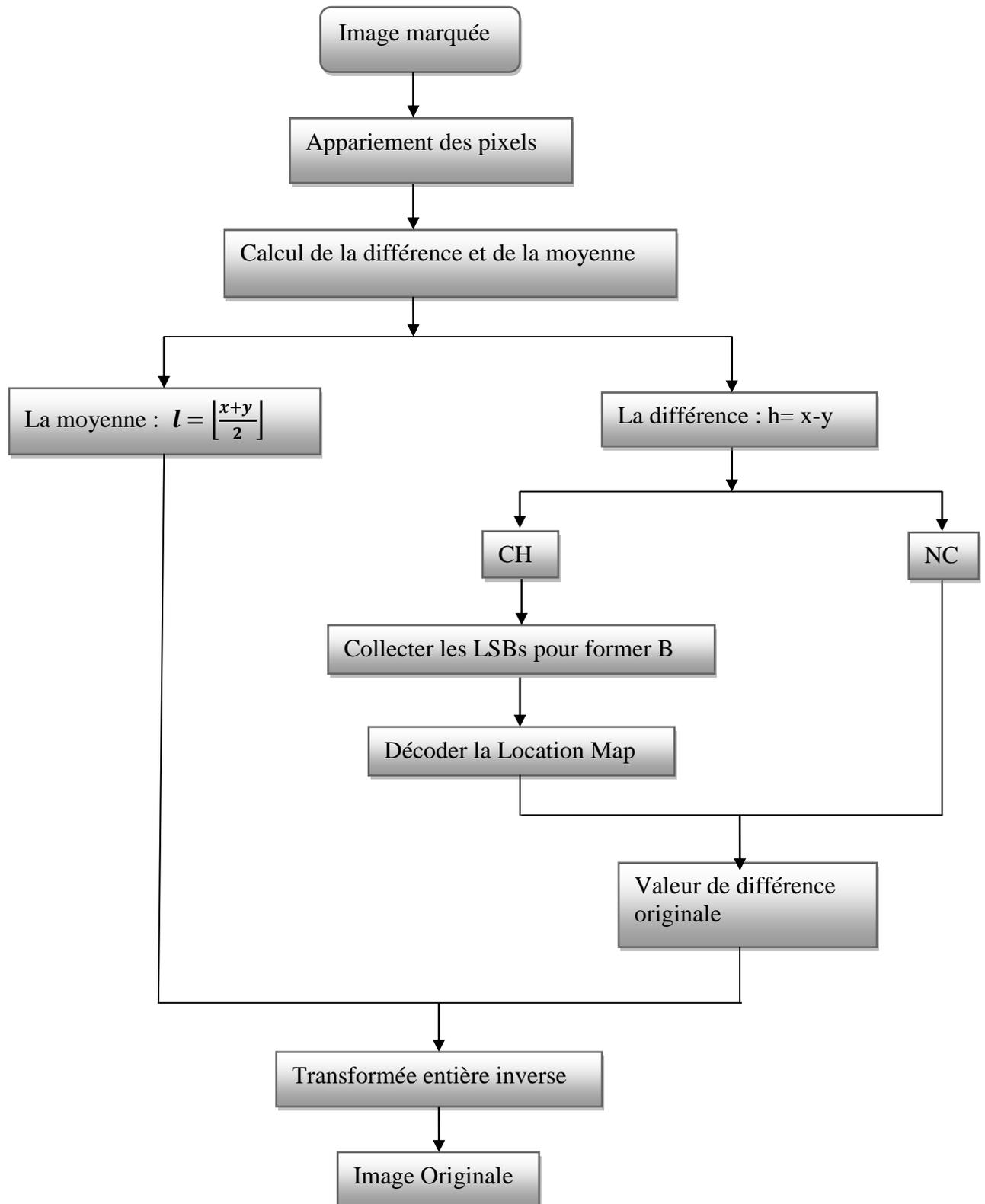


Figure 4.11 Organigramme d'Extraction

➤ Exemple d'application

Nous donnons ici un petit exemple d'application de l'insertion avec la DE, sans prendre en considération le problème de débordement.

Le processus d'insertion s'effectue alors de la manière suivante :

1. Pour chaque paire de pixels adjacents  $\{x_i, y_i\}$ , calculer  $l_i$  et  $h_i$ .

Par exemple,  $x_0 = 106$  et  $y_0 = 100$ , nous avons  $l_0 = \left\lfloor \frac{106+100}{2} \right\rfloor$

et  $h_0 = 106 - 100 = 6$

2. Calculer  $h' = 2 \times h + w_i$  ou  $w_i$  est le  $i^{\text{ème}}$  bit de la donnée à insérer.

Pour le même exemple si  $w_0 = 0$  alors  $h'_0 = 6 \times 2 + 0 = 12$ , si  $w_0 = 1$  alors

$h'_0 = 6 \times 2 + 1 = 13$

1. Obtenir  $x'$  et  $y'$  par substitution des paramètres  $l_i$  et  $h'_i$  dans la transformation entière

Pour notre exemple on aura :

- Si  $w_0 = 1$ ,  $\begin{cases} x'_0 = l_0 + \left\lfloor \frac{h'_0+1}{2} \right\rfloor = 103 + \left\lfloor \frac{13+1}{2} \right\rfloor = 110 \\ y'_0 = l_0 - \left\lfloor \frac{h'_0}{2} \right\rfloor = 103 - \left\lfloor \frac{13}{2} \right\rfloor = 97 \end{cases}$  et
- Si  $w_0 = 0$ ,  $\begin{cases} x'_0 = l_0 + \left\lfloor \frac{h'_0+1}{2} \right\rfloor = 103 + \left\lfloor \frac{12+1}{2} \right\rfloor = 109 \\ y'_0 = l_0 - \left\lfloor \frac{h'_0}{2} \right\rfloor = 103 - \left\lfloor \frac{12}{2} \right\rfloor = 97 \end{cases}$

2. Répéter ces étapes pour toutes les paires de pixels.
3. L'image marquée peut être obtenue par substitution de tous les  $x'_i, y'_i$  aux  $x_i, y_i$

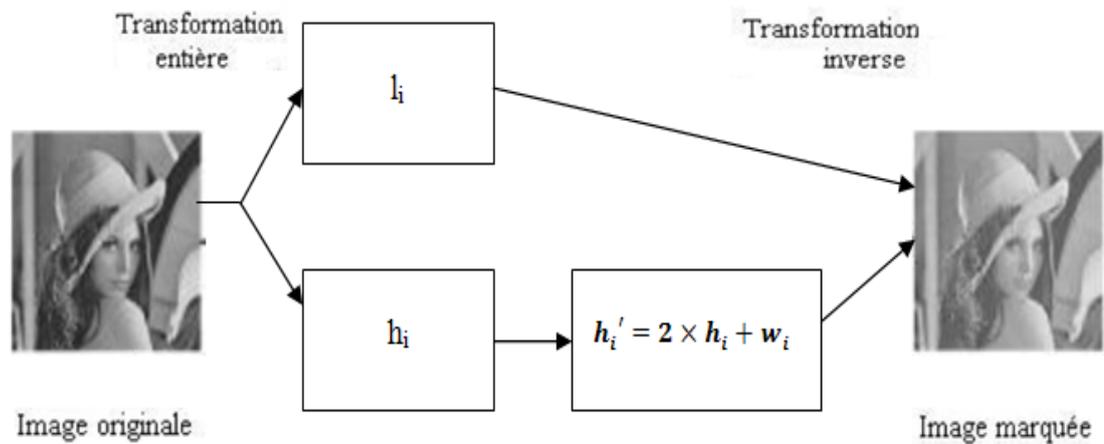


Figure 4.12 Insertion par le schéma de Tian [14]

Dans la phase d'extraction, les paires de pixels sont une nouvelle fois formées et  $l_i$  et  $h_i$  recalculés. Donc si  $x'_0=110$  et  $y'_0=97$  alors :

$$l_0 = \left\lfloor \frac{x'_0 + y'_0}{2} \right\rfloor = \left\lfloor \frac{110 + 97}{2} \right\rfloor = 103 \text{ et } h'_0 = x'_0 - y'_0 = 110 - 97 = 13 = \{1101\}_2$$

En ôtant le bit le moins significatif de  $h'_0$ , nous obtenons le bit de la marque  $w_0=1$  et

$h_0 = \left\lfloor \frac{13}{2} \right\rfloor = 6$ . Finalement, les pixels originaux sont reconstitués en calculant :

$$x_0 = l_0 + \left\lfloor \frac{h_0 + 1}{2} \right\rfloor = 103 + \left\lfloor \frac{6 + 1}{2} \right\rfloor = 106 \text{ et } y_0 = l_0 - \left\lfloor \frac{h_0}{2} \right\rfloor = 103 - \left\lfloor \frac{6}{2} \right\rfloor = 100.$$

#### 4.2.3 La DE modifiée

Dans la technique de différence d'expansion de Tian, le problème de dépassement de capacité a été résolu par l'insertion d'une table de localisation qui indique la position des valeurs qui ont été marquées après compression. Il faudrait noter que cette manière de procéder va diminuer la capacité d'insertion tout en augmentant le temps de calcul, à cause du processus de compression.

Ainsi, nous proposons de résoudre ce problème sans recourir à la table de localisation, ou à la compression. Nous allons intervenir dans la phase d'expansion de la technique de Tian, en représentant la valeur de la différence  $h$  dans le code binaire réfléchi de gray (CBRG) [67]. Une propriété fondamentale de ce code est que les bits les moins significatifs des mots de codes adjacents sont alternativement 11 00 11 00 11 00.....

contrairement au binaire pur pour lequel les mots de codes se terminent alternativement par 0 ou 1.

Cette alternance est exploitée ici pour élargir la valeur de différence sans induire de débordement aux pixels modifiés. Dans notre cas, la phase d'expansion de la différence devient (figure 4.13) :

- Calcul du code binaire réfléchi de  $h$  que l'on note  $hg$
- Calcul de  $h'$  selon la valeur de la donnée à insérer  $w_i$  ainsi :
  - Si  $w_i = \text{LSB}(hg)$  alors  $h' = hg$
  - Si  $w_i \neq \text{LSB}(hg)$  et  $hg$  est pair alors  $h' = hg + 1$
  - Si  $w_i \neq \text{LSB}(hg)$  et  $hg$  est impair alors  $h' = hg - 1$

Ainsi l'insertion par expansion de différence s'effectue dans ce cas par les deux équations suivantes :

$$x' = 1 - h' \quad \text{et} \quad y' = 1 + h'.$$

De cette manière, le LSB de  $h'$  est forcé à contenir  $w_i$  tout en restant dans l'intervalle [0-255], contrairement à la méthode initiale où  $h$  était multiplié par deux pour pouvoir insérer  $w_i$ , ce qui pouvait provoquer un débordement.

Dans la phase d'extraction, le processus inverse est opéré (figure 4.14) :

- Recalculer  $l'$  et  $h'$
- Transformer  $h'$  dans le code CBRG.
- Extraire le LSB de  $h'$  représentant la donnée insérée  $w_i$ .
- Recalculer la valeur de différence initiale de la manière suivante :
  - Si  $w_i = 0$  alors  $h = h' - 1$ .
  - Si  $w_i = 1$  alors  $h = h' + 1$ .

Par ailleurs, la méthode reste inchangée. Il est à noter que la capacité de la méthode va augmenter pour atteindre un bit pour une paire de pixels.

Nous proposons aussi d'ajouter une sécurisation en chiffrant la donnée avant de l'insérer en utilisant l'algorithme AES développé au sein de l'équipe [68].

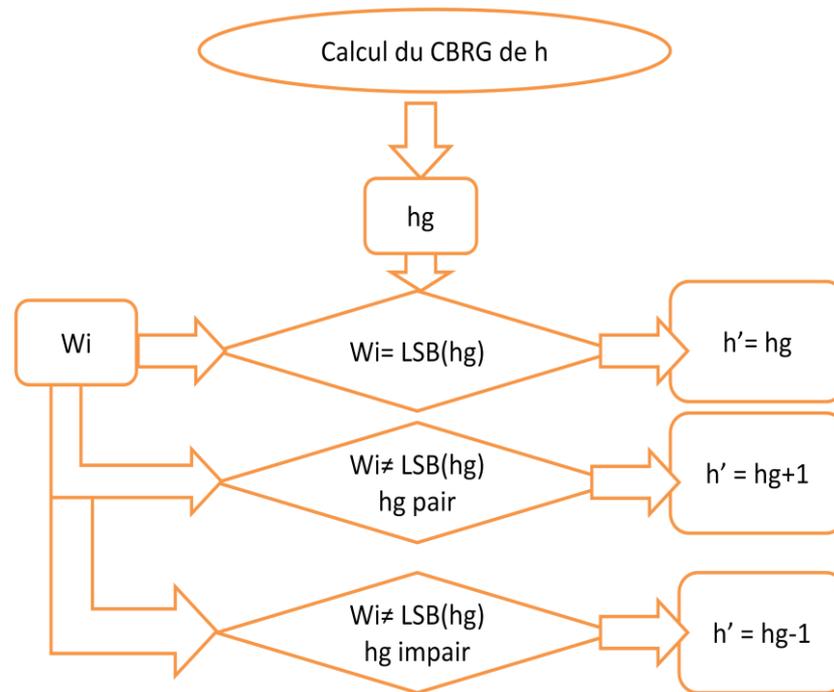


Figure 4.13 Insertion par la DE modifiée

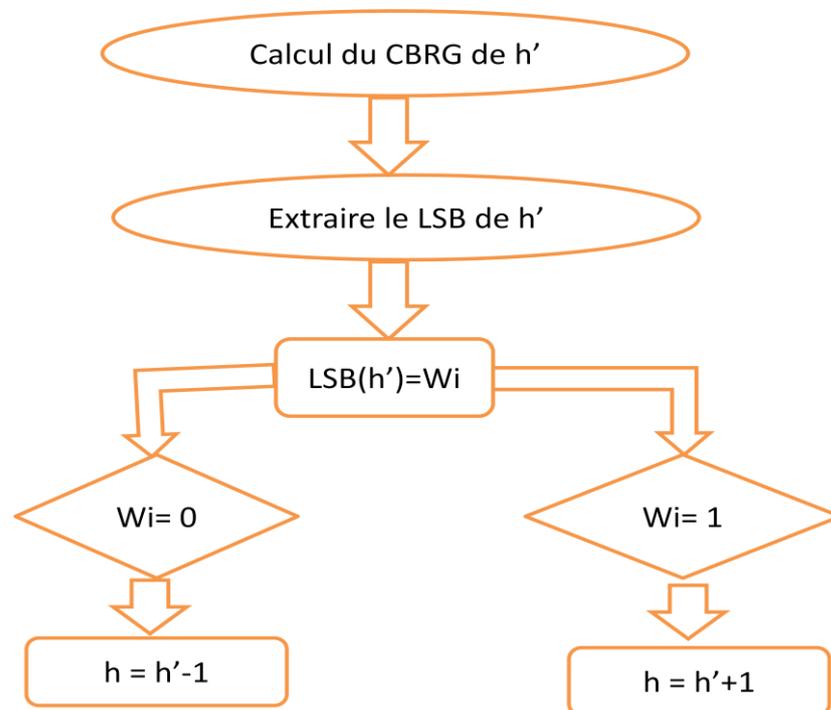


Figure 4.14 Extraction par la DE modifiée

➤ Tests et Résultats

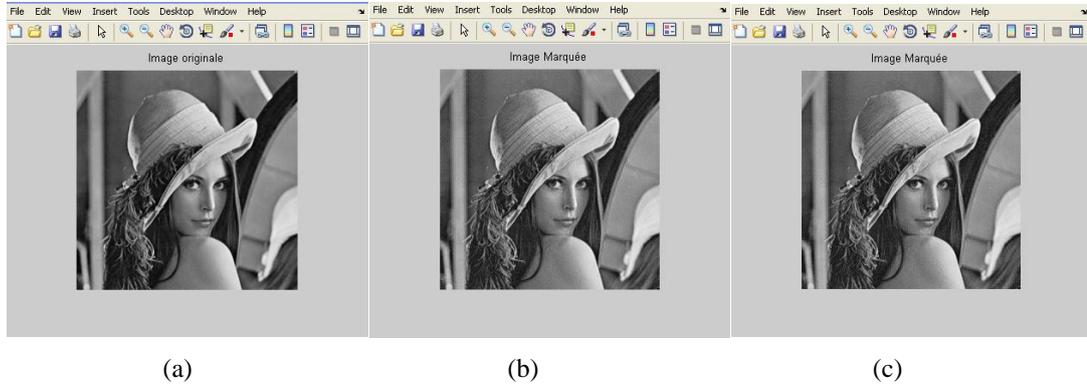


Figure 4.15 (a) Image originale (b) DE: capacité d'insertion =26269 bits (0.4008 bpp), PSNR=37.24 dB (c) DE modifiée: capacité d'insertion =32768 bits (0.5 bpp), PSNR=39.01 dB

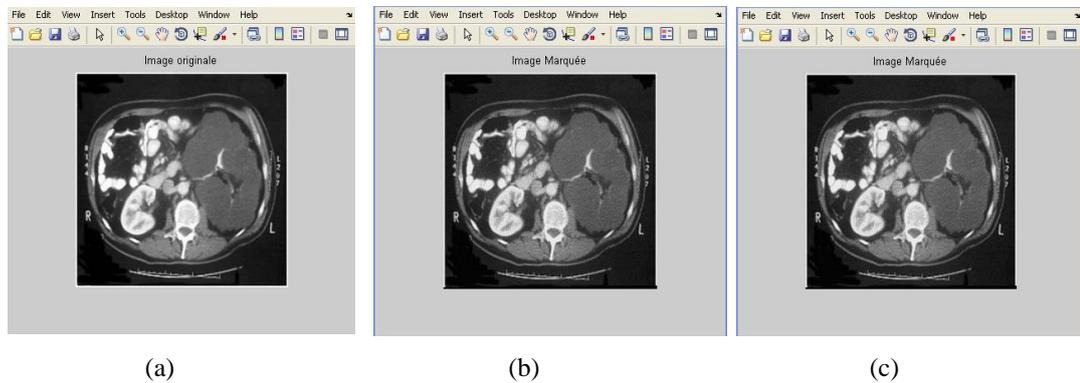


Figure 4.16 (a) Image IRM originale (b) DE: capacité d'insertion =23864 bits (0.3641 bpp), PSNR=38,05 dB (c) DE modifiée : capacité d'insertion =32768 bits (0.5 bpp), PSNR=40,15 dB

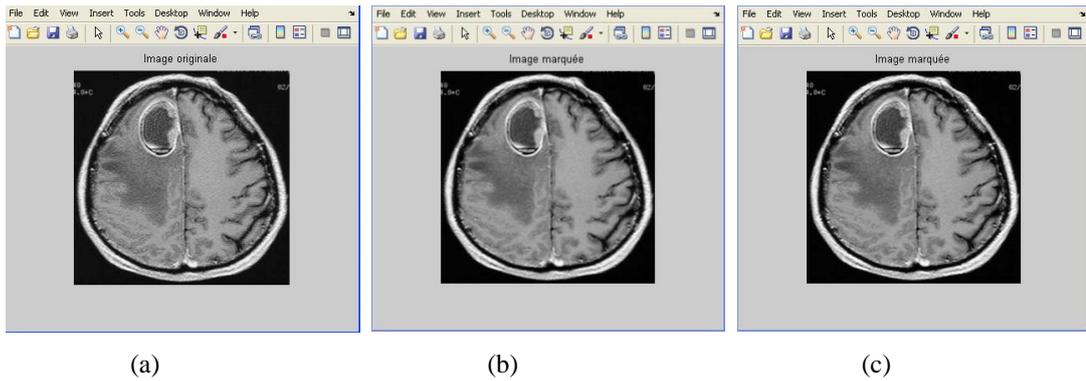


Figure 4.17 (a) Image IRM originale (b) DE: capacité d'insertion =23864 bits (0.3641 bpp), PSNR=38,24 dB (c) DE modifiée : capacité d'insertion =32768 bits (0.5 bpp), PSNR=40,11 dB

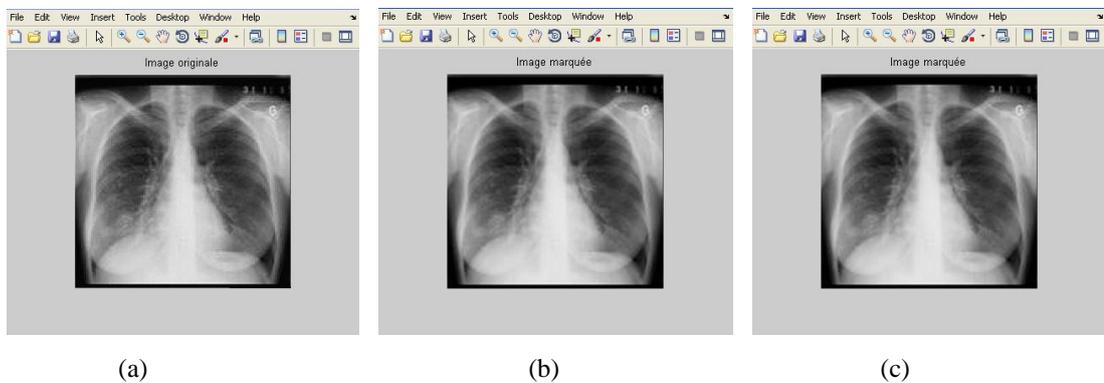


Figure 4.18 (a) Image Radiologique originale (b) DE: capacité d'insertion =23864 bits (0.3641 bpp), PSNR=38,59 dB (c) DE modifiée : capacité d'insertion =32768 bits (0.5 bpp), PSNR=40,29 dB

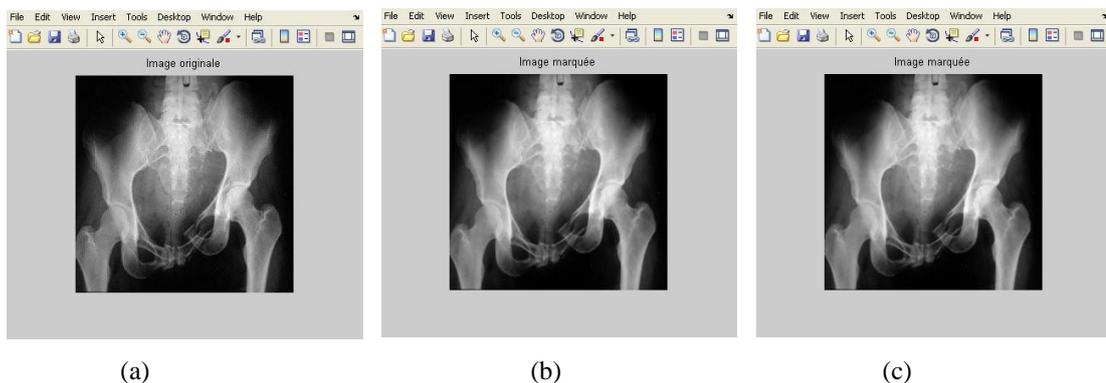


Figure 4.19 (a) Image Radiologique originale (b) DE: capacité d'insertion =23864 bits (0.3641 bpp), PSNR=38,65 dB (c) DE modifiée : capacité d'insertion =32768 bits (0.5 bpp), PSNR=40,32 dB



Figure 4.20 (a) Image optique originale (b) Image marquée après compression jpeg2000 / décompression et après extraction de la donnée (c) Image de différence

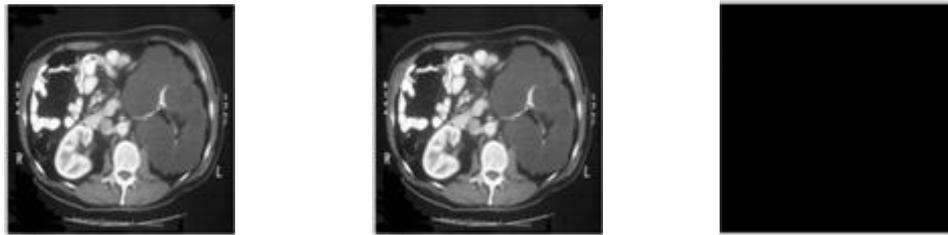


Figure 4.21 (a) Image médicale originale (b) Image marquée après compression jpeg2000/ décompression et après extraction de la donnée (c) Image de différence

Capacité d'insertion (bits)	32768	34288	60791	83519	98173
Débit (bpp )	0.50	0.52	0.93	1.3	1.49
PSNR (dB)	39.01	38,16	36.11	34.11	32.50

Tableau 4.5 DE modifiée pour Lena

Capacité d'insertion (bits)	32768	34288	60791	83519	98173
Débit (bpp )	0.50	0.52	0.93	1.3	1.49
PSNR (dB)	40.15	39,23	37.09	35.10	33.50

Tableau 4.6 DE modifiée pour l'image IRM (figure 4.16)

Nous présentons ici les résultats de nos tests sur les images Lena et médicale IRM de taille 256 x 256 x 8 bits. Nous avons observé les effets induits par une variation de la capacité d'insertion comprise entre 0.5 et 1.49 bpp sur la qualité de l'image comme le montre les tableaux 4.5 et 4.6 Au-delà de 1.49 bpp, il a été constaté que l'image est trop dégradée pour que la persistance de l'insertion soit nécessaire pour la taille de l'image

choisit. Néanmoins, la modification de la technique de DE proposée dans nos travaux, qui supprime la contrainte d'insertion de la table de localisation, permet d'améliorer la capacité d'insertion jusqu'à 50% de la capacité d'insertion de l'image contre 40% seulement avec la technique originale.

La capacité d'insertion dépend largement de la taille de l'image. En effet, la technique de DE permet une insertion maximale de un bit par paire de pixels, et jusqu'à 50% de l'image. Ainsi, plus la taille de l'image est grande, plus la capacité d'insertion est plus grande.

#### 4.2.4 L'approche par BPCS

La technique du Bit Plan Complexity Segmentation (BPCS) a été introduite en 1998 par Eiji Kawaguchi et Richard O.Eason [19] pour améliorer les performances de la technique traditionnelle du Bit le Moins Significatif (LSB – Least Significant Bit) du point de vue capacité d'insertion. La technique du LSB offre une capacité d'insertion de 10 à 15% de l'image alors que la BPCS peut aller jusqu'à une capacité d'insertion de 50%. La BPCS est basée sur l'idée de l'insertion de la donnée cachée dans les régions complexes des plans de bits de l'image déterminées selon les critères définis dans le paragraphe suivant.

##### ➤ Principe de base de la BPCS

- Complexité de mesure

L'étape la plus importante dans la BPCS consiste à trouver des régions "complexes" dans l'image où les données peuvent être cachées imperceptiblement. Il n'existe pas de définition universelle de la complexité d'une image (ou une région d'une image). Kawaguchi et Niimi [69] se sont penchés sur deux mesures de complexité différentes, l'une basée sur la longueur de la bordure noire et blanche et une autre basée sur le nombre de zones liées qui pourraient être utilisées pour trouver les régions dans une image complexe.

- Complexité de mesure basée sur la longueur de la bordure noire et blanche

Cette mesure est définie sur les quatre voisinages connectés d'un pixel. La longueur totale de la frontière en noir et blanc est définie comme la somme des variations de couleur le long des rangées et des colonnes dans l'image. Par exemple, un seul pixel blanc entouré de 4 pixels noirs, aura une longueur de bordure de 4 (2 changements de couleur le long de

chaque rangée et colonne). En étendant cette idée à une image binaire carrée de taille  $2^N \times 2^N$ , la longueur minimale possible des bordures est 0, obtenue pour une image blanche ou une image noire, et la longueur maximale possible de la bordure est  $2 * 2^N * (2^N - 1)$ , pour le motif de damier noir et blanc de  $((2^N - 1)$  changements le long de chacune des deux rangées lignes et colonnes.

Ainsi, la mesure de complexité d'une image est alors définie comme la valeur normalisée de la longueur totale de la bordure en noir et blanc dans l'image, notée  $\alpha$  :

$$\alpha = \frac{k}{2 \times 2^N \times (2^N - 1)} , 0 \leq k \leq (2 \times 2^N \times (2^N - 1)) \quad (4.13)$$

où  $k$  est la valeur de la longueur de bordure de l'image. Il est évident que  $\alpha$  est dans l'intervalle  $[0,1]$ .

- Complexité de mesure basée sur le nombre de zones liées

Cette mesure est basée aussi sur les quatre voisinages connectés du pixel, notée  $\beta$  définie comme suit :

$$\beta = \frac{m}{2^N \times 2^N} \quad (4.14)$$

$m$  est le nombre de zones connectées dans une image binaire carré de  $2^N \times 2^N$  pixels.

$\beta$  est dans l'intervalle  $(1/ (2^N \times 2^N), 1)$  avec un maximum obtenu pour une image à motif damier noir et blanc et le minimum pour une image noire ou blanche.

Les figures 4.22 (a) et 4.22 (b) montrent des valeurs  $\alpha$  et  $\beta$  pour deux types de blocs 8x8. Dans la pratique, il est constaté que, pour des blocs de 8x8, la mesure  $\alpha$  est plus ou moins uniformément distribuée dans  $[0, 1]$  tandis que la mesure  $\beta$  tend à avoir un pic définie à  $\beta = 0,2$  [2]. Ainsi, Eiji Kawaguchi choisit de mesurer  $\alpha$  pour trouver la région complexe dans la technique BPCS. Comme déjà mentionné, les valeurs maximales de  $\alpha$  et  $\beta$  sont obtenues pour le damier noir et blanc, voir la figure 4.22 (c).

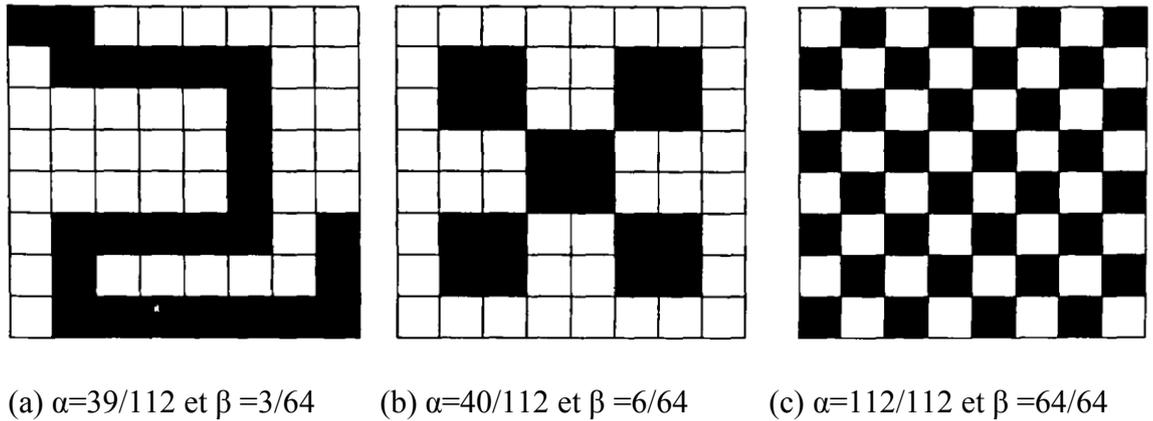


Figure 4.22 Valeurs  $\alpha$  et  $\beta$  pour un bloc 8x8

- Représentation en bits plans et Codage Gray

Dans la technique de la BPCS (Figure 4.23), l'image est segmentée en blocs de 8 bits, et chaque bloc est représenté en 8 bits plans pour une image en niveaux de gris, et 24 bits plans pour une image couleur (8 plans pour chaque composante rouge, vert et bleu).

Ainsi, soit une image en niveaux de gris  $P$  constituée de pixels codés sur  $n$  bits, sa représentation en bits plans sera [68]:

$$P = (P_1, P_2, \dots, P_n) \quad (4.15)$$

Pour obtenir ces différents plans, il faut appliquer un "et" (AND) avec un masque suivant le plan que l'on veut, par exemple pour une image en niveaux de gris de 8-bits si on applique AND "10000000" à l'image (donc à chaque valeur du pixel), nous obtenons le septième plan de bits, et ainsi de suite.

Dans le cas où l'image est une image couleur, elle sera décomposée comme ceci.

$$P = (PR_1, PR_2, \dots, PR_n, PG_1, PG_2, \dots, PG_n, PB_1, PB_2, \dots, PB_n),$$

Où  $PR_1, PG_1, PB_1$  sont les plans de bit les plus significatifs et  $PR_n, PG_n, PB_n$  sont les plans les moins significatifs.

L'objectif principal de la BPCS est d'insérer le maximum de données sans dégrader d'une manière très significatif l'aspect visuel de l'image originale.

Le codage binaire (PBC, Pur Binary Coding) est le code le plus utilisé dans le domaine de l'insertion de données cachées, l'inconvénient majeur du code binaire étant que deux valeurs proches dans l'espace des variables (voir le Tableau 4.7) ne sont pas

nécessairement codées par deux chaînes de bits voisines. On remédie en général à ce problème en utilisant le codage de Gray (CGC, Canonical Gray Coding) qui conserve une distance de Hamming de « 1 » entre deux chaînes (voir le Tableau 4.8). La distance de Hamming entre deux chaînes de bits est le nombre de bits qui diffère de l'une à l'autre. Pour les deux chaînes suivantes : 111 et 100, la distance est de 2.

<b>Nombre</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>
Format PBC initial	000	001	010	011
Format PBC après insertion	<b>100</b>	<b>101</b>	<b>110</b>	<b>111</b>

Tableau 4.7 Code binaire avant et après insertion

<b>Nombre</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>
Format CGC initial	000	001	011	010
Format CGC après insertion	<b>100</b>	<b>101</b>	<b>111</b>	<b>110</b>

Tableau 4.8 Code Gray avant et après insertion

Considérons deux pixels consécutifs dans une image ayant des valeurs d'intensité 127 et 128 respectivement. En PBC, 127 est représenté comme 01111111 et 128 est représenté comme 10000000. Les pixels apparaissent identiques à l'œil humain, mais diffèrent grandement dans la représentation en bits. C'est ce qu'on appelle le concept "Cliff Hamming". Après insertion, 01111111 devient 11111111 et 10000000 devient 00000000.

Avant insertion, la différence de niveau de gris est négligeable à l'œil humain. Après insertion, la différence de niveau de gris est de 255 soit un pixel noir foncé pour l'un un blanc pur pour l'autre. Ce changement est facilement perceptible à l'œil humain. Dans le code CGC, 127 qui est représenté sous forme binaire comme 01111111 devient 01000000. De même, 128 est représenté dans le CCG comme 11000000. En effet, les deux pixels semblent similaires, mais ne diffèrent que par un seul bit. Après insertion, 01000000 peut éventuellement devenir 11000000 et 11000000 devenir 01000000. Ainsi, la différence en

niveau d'intensité des pixels est imperceptible. Par conséquent, le CCG est préféré au PBC dans la technique du BPCS [70] [71].

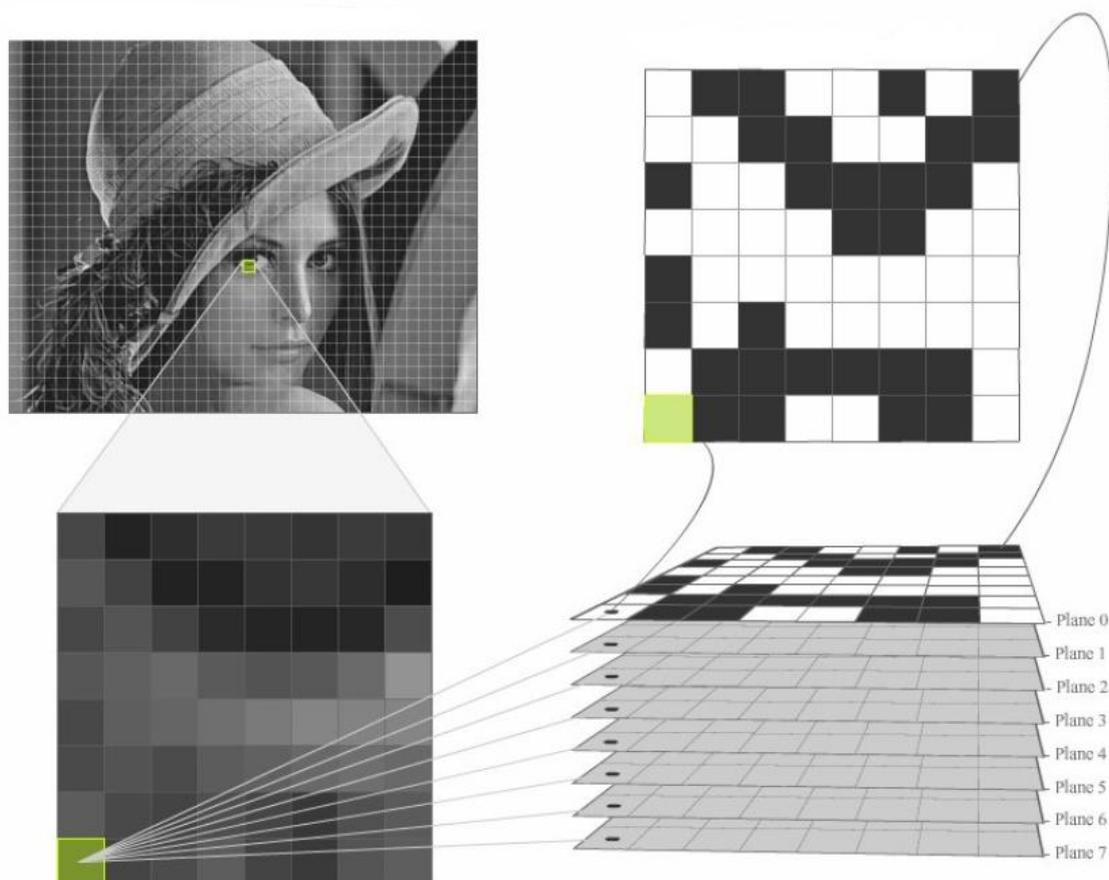


Figure 4.23 Représentation en bits plans et Codage Gray

- Opération de conjugaison

Soit une image 24 bits (image d'origine) transformée en 24 bits plans. La complexité  $\alpha$  de chaque bloc 8x8 dans chacun des 24 bits plans est calculée, et la complexité de chaque bloc est comparée à un seuil  $\alpha_0$ . Si  $\alpha > \alpha_0$ , le bloc est considéré comme complexe et peut être remplacé par le bloc de données à insérer.

La valeur standard utilisée pour  $\alpha_0$  est d'environ 0,3[19]. Les blocs de données à insérer sont des blocs de données obtenues à partir de n'importe quel fichier ASCII codé (ou fichier image qui peut être lue comme une chaîne de caractères ASCII) appelés fichier de ressources.

Chaque bloc de 8 octets d'un fichier de ressources constitue un bloc de ressource de 8x8 avec la représentation binaire de 8 bits de chaque octet formant une rangée de bloc 8x8. Par exemple, une séquence de 8 caractères à partir d'un MS Word Document « This one» (8 caractères y compris l'espace vide) formerait un bloc de ressource 8x8 comme le montre la figure 4.24 (a) et un bloc de 8 espaces vides consécutifs formeraient un bloc comme le montre la figure 4.24 (b), puisque, «T» est 01010100, 'h' est 01101000, 00100000 espace vide (32 en ASCII) et ainsi de suite [72].

0	1	0	1	0	1	0	0	T		0	0	1	0	0	0	0	0
0	1	1	0	1	0	0	0	h		0	0	1	0	0	0	0	0
0	1	1	0	1	0	0	1	i		0	0	1	0	0	0	0	0
0	1	1	1	0	0	1	1	s		0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0			0	0	1	0	0	0	0	0
0	1	1	0	1	1	1	1	o		0	0	1	0	0	0	0	0
0	1	1	0	1	1	1	0	n		0	0	1	0	0	0	0	0
0	1	1	0	0	1	0	1	e		0	0	1	0	0	0	0	0
(a)									(b)								

Figure 4.24 Blocs ressources de deux séquences différentes de 8 caractères

Un bloc complexe représente une séquence qui apparaît bruyante, et la modification d'un tel bloc ne serait pas perceptible, à moins que, la modification aboutit à un bloc moins complexe que la valeur seuil  $\alpha_0$ . Considérons le bloc représenté dans la figure 4.24 (b). Ceci est un bloc fréquemment rencontré dans les documents Word contenant de nombreux tronçons avec un espace vide.

La complexité de ce bloc est 0,1429, donc inférieure à  $\alpha_0$ . Si ce bloc remplace un bloc complexe dans le plan de bits de l'image, la modification sera visible, en particulier si elle est dans un ordre supérieur des plans de bits. En outre, le module de décodage ne reconnaîtra pas le bloc, car elle suppose que seuls les blocs complexes ont été remplacés et, seuls les blocs complexes ont été remplacés par des informations valides. Pour remédier à ce problème, l'opération de conjugaison est introduite.

Figure 4.25 (b) montre le bloc 8x8 le plus complexe avec une complexité de 1. Ce bloc est noté  $W_C$ . Le bloc  $W_C$  a une propriété particulière quand il est XORé (opération de OU exclusif) avec un bloc non-complexe P par exemple, de complexité  $\alpha_n < \alpha_0$ . Le bloc qui en résulte,  $P^*$ , a une complexité de  $(1-\alpha_n) > \alpha_0$ . Comme pour toute opération XOR, le bloc

P peut être facilement retrouvé par XOR à nouveau avec  $W_c$ . Cette opération de changement de complexité d'un bloc par XOR avec  $W_c$  est appelée l'opération de conjugaison et est notée par '\*'. Figure 4.25 (a) montre un exemple de bloc non complexe P, figure 4.25 (b) un bloc parfaitement complexe  $W_c$ , et figure 4.25 (c) un bloc conjugué  $P^*$ , obtenu en appliquant un XOR entre le bloc de la figure 4.25 (a) et le bloc de la figure 4.25 (b).

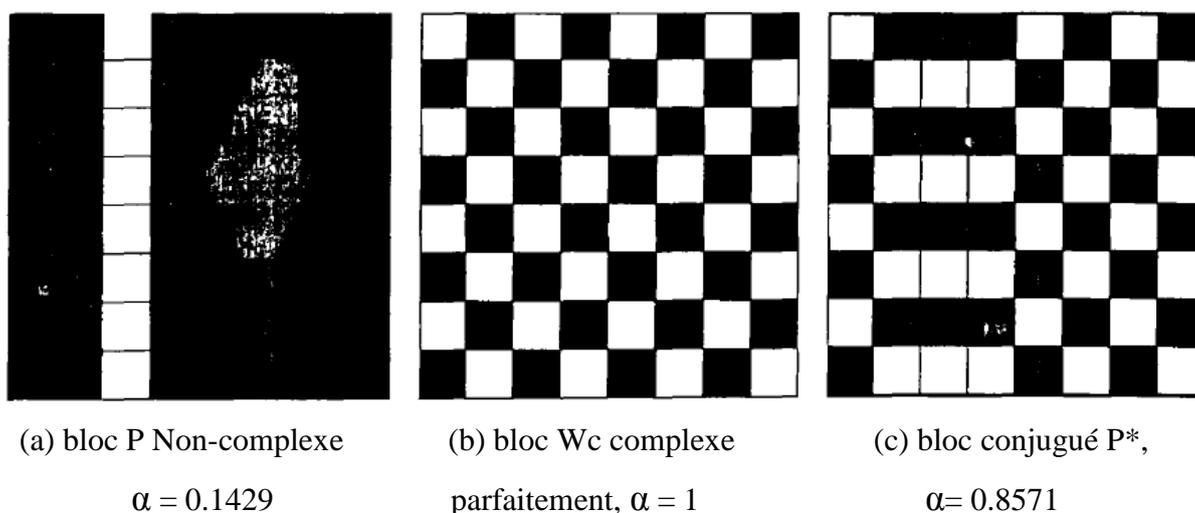


Figure 4.25 Opération de conjugaison (exemple)

Les deux propriétés importantes de l'opération de conjugaison peuvent être résumées comme suit:

- $\alpha(P^*) = 1 - \alpha(P)$
- $(P^*)^* = P$

La première propriété est utilisée au niveau du codeur pour rendre complexes les blocs de ressource non-complexes, tandis que la seconde propriété est utilisée au niveau du décodeur pour récupérer le bloc d'origine.

Cependant, il arrive souvent que quelques blocs de ressources soient complexes initialement sans avoir recours à l'étape de conjugaison. Il devient donc important de garder une trace de ceux qui ont été conjugués. Ceci est possible en utilisant une carte Conjugaison. Pour chaque bloc de 8 octets du fichier de ressources, un bit est ajouté à la carte de conjugaison pour indiquer si le bloc a été conjugué ou pas. Un '1' indique que le bloc de ressources a été conjugué, tandis qu'un '0' indique que le bloc de ressources a été

inséré sans conjugaison. La carte de conjugaison est finalement intégrée après l'insertion de tous les blocs de ressources.

➤ Principe de base de la BPCS

• Procédure d'insertion

Les étapes de codage se résument comme suit (voir figure 4.26, 4.27, et 4.28) :

1. Convertir les valeurs d'intensité de l'image originale du binaire en code gray et la décomposer en bits plans.
2. Calculer la complexité de mesure  $\alpha$  pour chaque bloc  $8 \times 8$  des bits plans.  
Si  $\alpha > \alpha_0$ , marquer le bloc  $8 \times 8$  comme complexe dans la carte de conjugaison par 1. Pour une image de  $512 \times 512$ , la carte de conjugaison est de  $64 \times 64$  bits pour chaque bit plan.
3. Identifier le nombre de fichiers de ressources à insérer  $n$ . Préparer le premier octet de l'en-tête global (OH), et l'intégrer dans le premier bloc complexe de l'image originale, le conjuguer, si nécessaire. Répétez les étapes 4 à 7,  $n$  fois jusqu'à l'insertion totale de l'information
4. Lire le fichier de ressources et le transformer en une séquence (ou vecteur) de valeurs ASCII. Faire en sorte que le nombre d'octets dans la séquence est un multiple de 8. Cela est nécessaire parce que le codeur intègre des blocs de 8 octets à la fois. La taille de l'en-tête contenant le nom du fichier et sa taille est fixé à 24 octets.
5. Lire l'entête du fichier, 8 octets à la fois, et le transformer en blocs binaires  $8 \times 8$ . Calculer  $\alpha$  pour chaque bloc et effectuez l'une des options suivantes:
  - Si  $\alpha > \alpha_0$ , le bloc ressource est inséré sans conjugaison et dans la carte de conjugaison, nous le marquons par 0.
  - $\alpha < \alpha_0$ , le bloc ressource doit être conjugué et la carte de conjugaisons est marquée par 1.
6. Subdiviser la carte de conjugaison en blocs de 63 bits chacun, remplir avec des zéros pour le dernier bloc, le cas échéant. Rendre le premier bit (bit supérieur gauche) d'un bloc  $8 \times 8$  '0' et ajouter le bloc 63 bits, par lignes. Si le bloc est complexe, l'intégrer sans conjugaison dans le bloc suivant complexe disponible des

plans de bits de l'image originale. Si le bloc n'est pas complexe, alors le conjuguer avant de l'insérer. Pour l'entête du fichier, il n'y aura qu'un seul bloc.

7. Répéter les étapes 5 et 6 pour l'entête du fichier. Reconvertir l'image du CGC au PBC pour obtenir l'image marquée.

- Procédure d'extraction

Le travail du décodeur est de systématiquement inverser les opérations au niveau du codeur. La procédure de décodage peut être résumée comme suit:

1. Lire l'image codée, la convertir en code de Gray et effectuer la décomposition en bits plans.
2. Utiliser le seuil fixé pour la complexité,  $\alpha_0$ , et marquer chaque bloc 8x8 dans le plan de bits d'image avec un «1» s'il est complexe.
3. Récupérer la valeur du nombre de fichiers incorporés dans l'image codée,  $n$ , à partir du premier octet du premier bloc 8x8 complexe. Répétez les étapes 4 à 5,  $n$  fois.
4. Récupérer les 4 prochains blocs complexes. Les trois premiers blocs contiennent l'entête du fichier (3 bits d'une séquence 63 bits) et le quatrième bloc contient la carte de conjugaison. Si le bit en haut à gauche de ce quatrième bloc est «1», cela signifie que le bloc a été conjugué et doit être conjugué à nouveau pour récupérer les informations d'origine (propriété 2 de l'opérateur de conjugaison). Convertir ce bloc 8x8 en une séquence de 63 bits (à l'exception du bit en haut à gauche) et d'utiliser les 3 premiers bits de cette séquence pour reconstruire les 3 premiers blocs. Si le bit est un 1, cela signifie que le bloc correspondant doit être conjugué pour récupérer les informations d'origine. Ces blocs reconstruits formeront les 24 octets du fichier d'en-tête (18 octets pour le nom du fichier et 6 octets pour sa taille).
5. Le nombre de blocs complexes 8x8 du fichier ressource étant égal à  $T/8$  ( $T$ =la taille du fichier). Retrouver ces blocs complexes et leur carte de conjugaison correspondante et reconstruire le fichier ressource original de la même manière que l'entête du fichier.

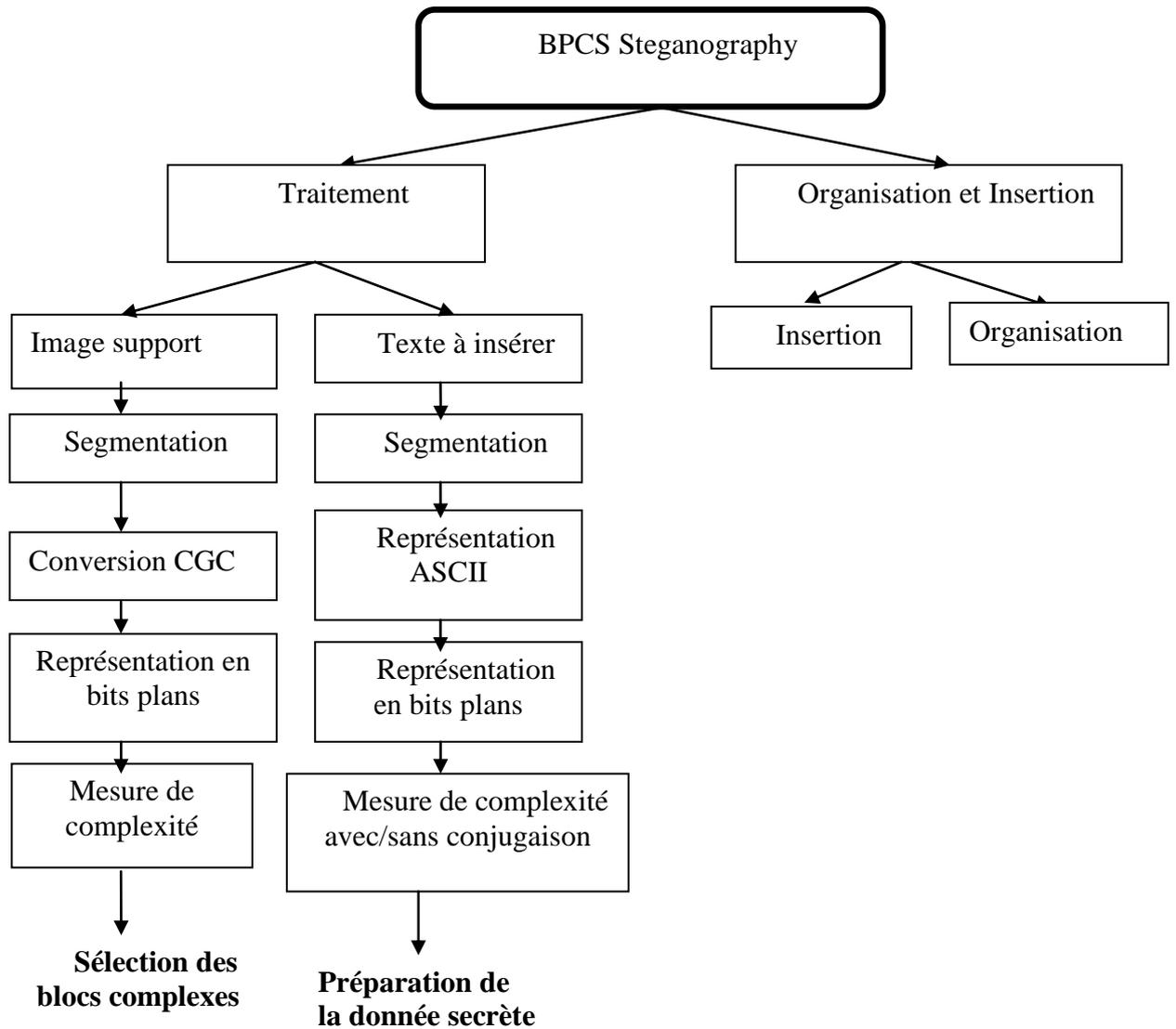


Figure 4.26 La technique BPCS

Répéter l'insertion au nombre de fichiers insérés



- 1 Premier bloc : nombre de fichiers insérés (8 octets)
- 2 Trois blocs: 24 octets ( 18 pour le nom du fichier + 6 pour la taille)
- 3 Cinquième bloc: Carte de conjugaison du FH de 8 octets (03 bits sont utilisés)
- 4 Données secrètes insérées
- 5 Carte de conjugaison

Figure 4.27 Schéma d'organisation de la donnée insérée

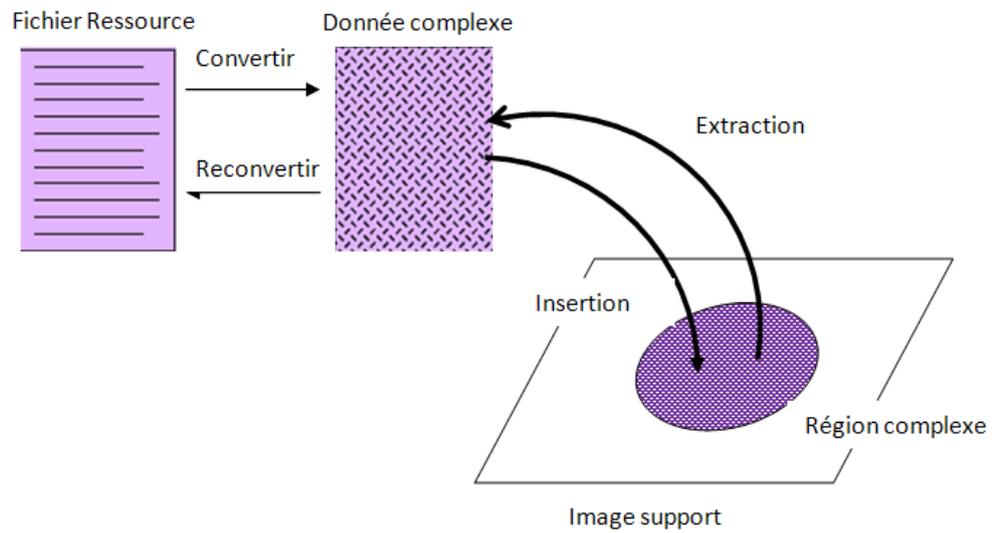


Figure 4.28 Schéma d'insertion et d'extraction

#### 4.2.5 L'approche proposée

Dans cette section, nous présentons notre approche qui consiste à prendre l'esprit de la technique de BPCS pour la préparation de l'image au marquage, et lui associer la technique de la DE modifiée pour effectuer l'insertion.

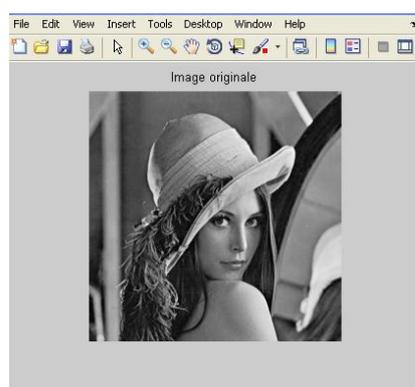
- **Schéma d'insertion**

- Transformer l'image originale du PBC au CGC
- La décomposer en bits plans
- Insérer avec la DE modifiée dans les quatre dernier bits plans de l'image
- Reconstruire l'image
- La transformer du CGC au PBC

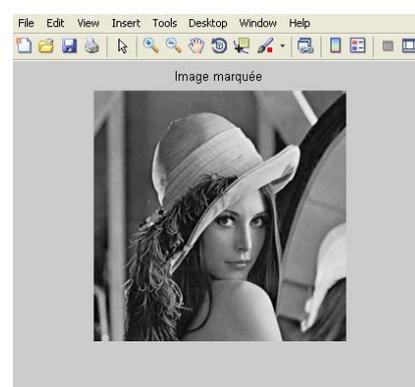
- **Schéma d'Extraction**

- Transformer l'image originale du PBC au CGC
- La décomposer en bits plans
- Extraire avec la DE modifiée dans les quatre dernier bits plans de l'image
- Reconstruire l'image
- La transformer du CGC au PBC

➤ Tests et Résultats

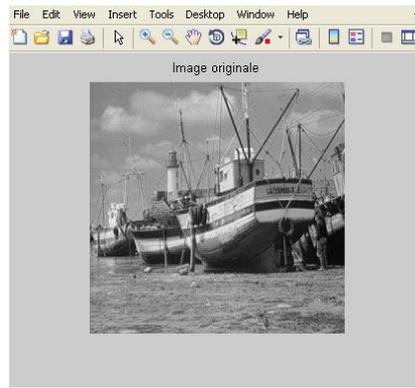


(a)

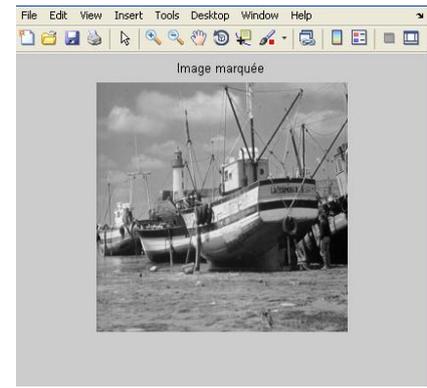


(b)

Figure 4.29 (a) Image optique originale (b) BPCS+DE: capacité d'insertion =32768 bits (0.5 bpp), PSNR=45.11 dB

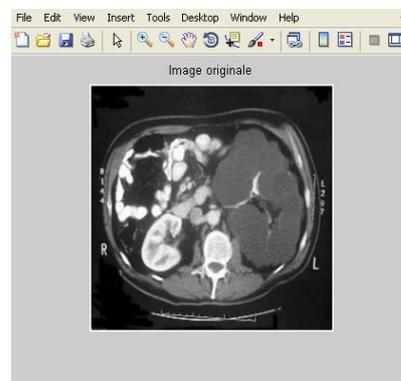


(a)

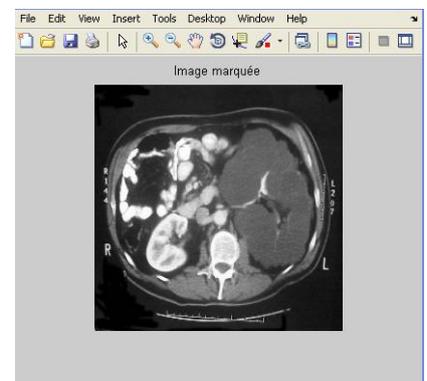


(b)

Figure 4.30 (a) Image optique originale (b) BPCS+DE: capacité d'insertion =32768 bits (0.5 bpp), PSNR=45.20 dB

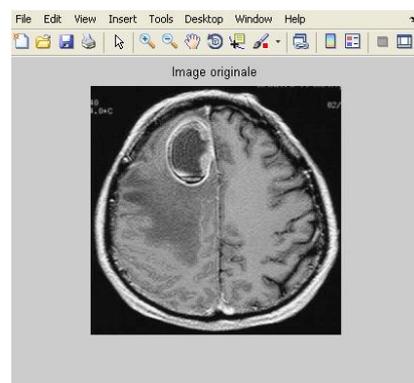


(a)

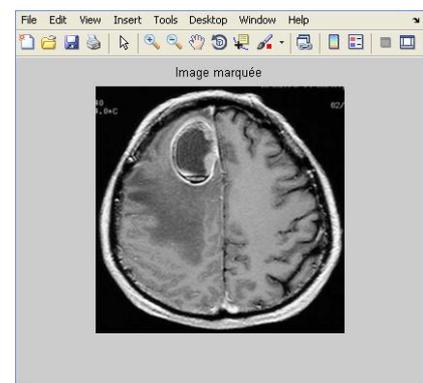


(b)

Figure 4.31 (a) Image IRM originale (b) BPCS+DE : capacité d'insertion =32768 bits (0.5 bpp), PSNR=46,30dB



(a)



(b)

Figure 4.32 (a) Image IRM originale (b) BPCS+DE : capacité d'insertion =32768 bits (0.5 bpp), PSNR=46,42dB

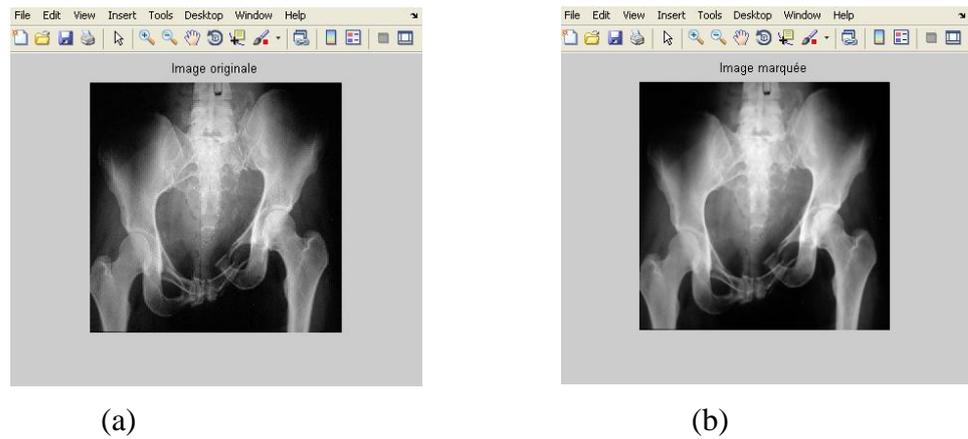


Figure 4.33 (a) Image Radiologique originale (b) BPCS+DE : capacité d'insertion =32768 bits (0.5 bpp), PSNR=46,55dB

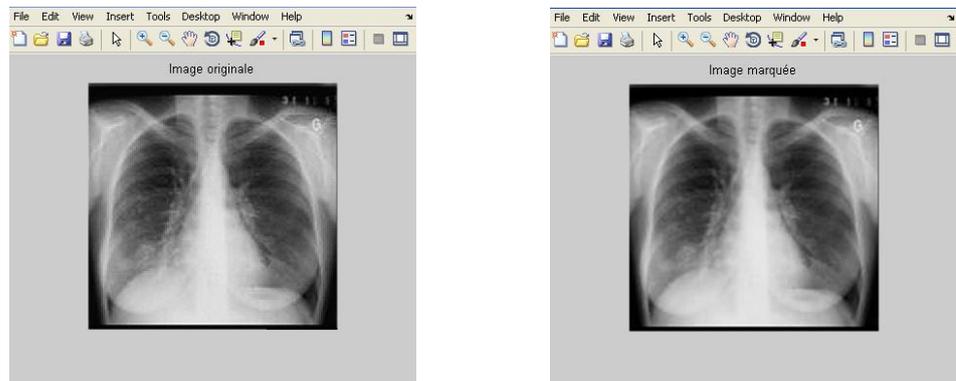


Figure 4.34 (a) Image Radiologique originale (b) BPCS+DE : capacité d'insertion =32768 bits (0.5 bpp), PSNR=46,52dB



Figure 4.35 (a) Image optique originale (b) Image marquée après compression jpeg2000 / décompression et après extraction de la donnée (c) Image de différence

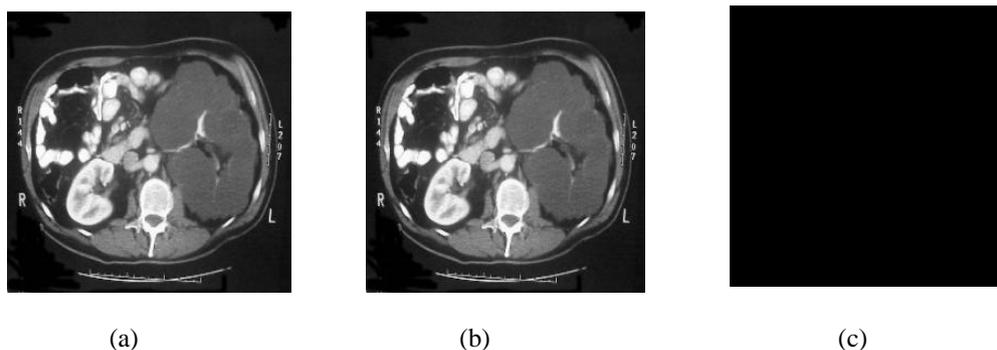


Figure 4.36 (a) Image IRM originale (b) Image marquée après compression jpeg2000/ décompression et après extraction de la donnée (c) Image de différence

Capacité d'insertion (bits)	32768	34288	60791	83519	98173
Débit (bpp )	0.50	0.52	0.93	1.3	1.49
PSNR (dB) (notre approche)	45.11	44.50	42.10	40.69	39.50

Tableau 4.9 BPCS+DE modifiée pour Lena

Capacité d'insertion (bits)	32768	34288	60791	83519	98173
Débit (bpp )	0.50	0.52	0.93	1.3	1.49
PSNR (dB)	46.30	45,23	43.08	41.75	40.12

Tableau 4.10 BPCS+DE modifiée pour l'image médicale IRM (figure 4.31)

Débit (bpp )	0.46	0.54	0.67	0.74	0.85
PSNR (dB)	37.66	36.15	34.80	33.05	32.54

Tableau 4.11 Tian pour Lena 512×512×8 bits

Les tests effectués pour valider notre nouvelle approche sont sur une image optique et une image médicale de taille 256 x 256 en niveaux de gris. Une variation de la capacité d'insertion comprise entre 0.5 et 1.49 bpp est présentée dans les tableaux 4.9 et 4.10.

Nous avons comparé nos résultats avec ceux de Tian [14]. Les tableaux 4.9, 4.10 et 4.11 montrent les résultats de l'image optique Lena, et l'image médicale, respectivement. A travers ces tableaux, nous pouvons comparer les performances de notre approche avec celle de Tian par la capacité d'insertion (mesurée à l'aide du débit par pixel (bpp)) et la dégradation de l'image (mesurée à l'aide du PSNR (Peak Signal to Noise Ratio)).

Dans les tableaux 4.9, et 4.11, on peut voir que le nouveau schéma est plus performant que celui de Tian. Pour des valeurs à peu près égales des débits, nous obtenons des valeurs de PSNR bien meilleurs que celles obtenues par Tian.

Cependant, les résultats obtenus en termes de qualité d'image, évaluée à l'aide du PSNR (Tableau 4.10), prouvent que notre schéma fournit de très bonnes performances sur l'image médicale.

D'autre part, les résultats obtenus (figure 4.35 et figure 4.36) montrent que notre nouveau schéma d'insertion résiste à l'attaque Jpeg2000 et permet une reconstruction parfaite de l'image originale.

### 4.3 Conclusion

Dans ce chapitre, nous avons présenté une technique d'insertion de données basée sur l'expansion de différence proposée par Tian [14]. Une technique réversible avec possibilité d'insérer une capacité importante de données. Nous avons proposé une amélioration de ce schéma d'insertion par la suppression de l'insertion de la table de localisation et ainsi remédier au problème de débordement. Enfin, nous avons proposé un nouveau schéma d'insertion de données cachées conjoint basé sur la BPCS et la DE, une technique d'IDC réversible qui offre une capacité d'insertion importante avec une dégradation minimales des images. Les travaux futurs porteront sur la possibilité, d'insertion de ce schéma dans le codeur Jpeg2000, l'objectif pour lequel la BPCS a été choisie pour pouvoir insérer dans les bits plans générés en cours du processus de compression.

## CONCLUSION GÉNÉRALE

Au terme de ce mémoire nous tentons de tirer des conclusions des travaux présentés et de proposer des perspectives d'amélioration de nos propositions.

L'objectif principal de cette étude était de créer et développer des algorithmes d'insertion de données dans le flux compressé JPEG2000 de l'image médicale. Il s'agit d'insérer les données relatives aux patients dans le flux compressé de l'image sans la dégrader.

Après une étude bibliographique des techniques d'insertion de données cachées (IDC) existantes, nous avons commencé par étudier ses origines, à savoir la stéganographie. Nous avons aussi mis l'accent sur ses caractéristiques et les contraintes auxquelles il est soumis pour passer par la suite à démontrer la nécessité de l'IDC réversible. Nous avons présenté en détails ses caractéristiques, les schémas d'insertion et de détection pour ensuite énumérer les techniques d'IDC les plus connues.

D'autre part, nous avons présenté le codeur JPEG2000. L'algorithme de codage JPEG 2000 repose sur une Transformée en Ondelettes Discrète (TOF), une quantification scalaire et un codage arithmétique adaptatif. Ainsi le terme de domaine compressé/codé peut signifier, suivant les cas, soit le domaine des Ondelettes, quantifié ou non, soit les flux de bits résultants du codage entropique de groupes de coefficients quantifiés.

Dans le cadre de ce mémoire, nous avons abouti à une technique d'insertion de données cachées (IDC) réversible qui permet d'assurer une grande capacité d'insertion tout en offrant aux utilisateurs une copie intacte de l'image originale.

Nous avons développé dans un premier temps une technique d'IDC basée sur l'insertion dans les LSBs du flux compressé JPEG2000 proposée par Katsutoshi [34] afin d'exploiter les différentes capacités du système de codage JPEG2000, tels que la structure de couches (Layers). Nous avons appliqué cette technique sur des images médicales et nous avons montré qu'il est nécessaire d'augmenter le nombre de couches en fonction de la taille de l'image pour que la dégradation causée par l'insertion de données cachées soit imperceptible.

Dans un deuxième temps, Nous avons développé dans le domaine spatial, une nouvelle approche basée sur l'expansion de différence entre deux pixels voisins. Puis nous l'avons appliquée sur des images optiques et médicales de 256 niveaux de gris de taille  $256 \times 256$  pixels. Nous avons pu montrer l'efficacité de cette technique pour insérer une grande quantité d'information dans des images médicales tout en assurant une bonne qualité

visuelle. Nous avons aussi montré que les performances de cette technique point de vue capacité d'insertion peuvent être améliorées, en contournant le problème de dépassement.

Dans un troisième temps, afin de réaliser l'objectif de notre projet, à savoir développer un algorithme d'insertion de données dans le flux compressé JPEG2000 de l'image médicale, nous avons combiné l'esprit d'insertion de données cachées dans les bits plans de l'image d'une technique nommée Bit Plan Complexity Segmentation (BPCS) avec celle basée sur l'expansion de différence, afin d'obtenir une technique très performante permettant son application au flux compressé JPEG2000.

Enfin, nous avons terminé cette partie par une comparaison de notre méthode avec celle proposée par Tian [14] pour mettre en évidence les résultats très prometteurs issus de ce travail.

Les principales contributions développées dans ce rapport concernent :

- Le développement d'un schéma d'IDC réversible utilisant le domaine compressé JPEG2000.
- Le développement d'un schéma d'IDC réversible utilisant le domaine spatial avec le contournement du problème de débordement.
- Combinaison de deux techniques d'IDC afin d'obtenir une technique d'IDC réversible permettant d'insérer une grande quantité d'information dans les bits plans de l'image.

L'objectif de ce mémoire est de mettre en place une nouvelle approche d'IDC réversible permettant une insertion de données dans le domaine compressé JPEG2000. Le domaine spatial offrant une multitude d'avantages, présente aussi des limitations et inconvénients pouvant affecter l'efficacité du schéma d'IDC. D'où la nécessité d'utiliser le domaine compressé. JPEG2000 présente une haute flexibilité qui découle dans le choix de zones d'insertion, à savoir, les coefficients d'ondelettes, les bits plans ou le système de couches (Layers). De ce fait, la principale perspective à court terme consiste à intégrer la technique proposée au codeur JPEG2000.

La perspective à moyen et long terme est de poursuivre les travaux dans le domaine en développant de nouvelles approches avec une extension à la vidéo dans ses nouvelles normes notamment la H264. Cette extension paraît naturelle dans la mesure où l'on peut considérer une vidéo comme une succession d'images fixes.

## APPENDICE A. PRESENTATION DES CODEURS IBIG2 ET RLE

### A.1. Le codeur JBIG2

C'est une évolution de JBIG (Joint Bi-level Image Experts Group) développé pour la compression sans perte des images binaires (bi-level). Il peut aussi coder des images en niveaux de gris ou en couleur avec un nombre limite de bits par pixel. Il est utilisé essentiellement pour les fax. JBIG2 est venu améliorer les performances de ces prédécesseurs pour la compression des images binaires. Il supporte aussi bien la compression avec et sans pertes contrairement au JBIG1 qui ne supporte que la compression sans pertes. La capacité de compression a été multipliée par un facteur allant de 3 à 10. Il supporte, aussi, la compression d'un document multi-pages [49].

### A.2. Codage Run-Length (RLE)

Le principe de la compression Run-Length (RLE) repose sur le fait que dans une image, il existe de nombreuses répétitions d'un même pixel, ou d'une même séquence de pixel, tous juxtaposés. Ainsi, au lieu de coder chaque pixel d'une image, le RLE propose de coder d'une part le nombre de répétitions, et d'autre part la séquence ou l'octet à répéter. Lorsqu'il s'agit d'une répétition de séquence, on ajoute entre le nombre de répétitions et la séquence, un octet représentant le nombre d'octets de la séquence. (voir tableau 2). Si la taille de celle-ci est impaire, alors on ajoute un octet nul (00) à la fin.

10 10 10 10 10	est codé 05 10
0A	est codé 0A 0A
23 65 55 34 22	est codé 00 05 23 65 34 22 00
10 10 89 23	est codé 00 04 10 10 89 23

Tableau A.1 : Exemple de codage du format RLE

En plus des pixels, le code RLE doit aussi coder un saut de ligne (sur deux octets : 00 00), et la fin de l'image (sur deux octets aussi : 00 01). Ce codage présente une autre particularité puisqu'il peut aussi coder le déplacement du « pointer » (lors du traitement de l'image), et ainsi omettre volontairement des pixels. La syntaxe est la suivante : 00 02 XX YY, avec XX le nombre de pixel sur une ligne et YY le nombre de pixel sur une colonne (en hexadécimale) [73].

## REFERENCES BIBLIOGRAPHIQUES

- [1] Audio Steg : <http://www.snotmonkey.com/work/school/405/overview.html>
- [2] Barbier, J., « La stéganographie moderne: d'Hérodote à nos jours. » In Computer & Electronics Security Application, CESAR 2007, novembre 2007, Rennes, France.
- [3] Barbier, J., « Analyse de canaux de communication dans un contexte non coopératif », thèse en informatique (2007), Ecole Polytechnique, Rennes, France.
- [4] Dentand, J., «Stéganographie. ». Travail de diplôme 2005. Sous la direction de Peter DAEHNE, professeur HES. Haute École de Gestion de Genève (HEG-GE).
- [5] Diskin, P., Lau, S., Parlett, R., « Steganography and digital watermarking », copyright©2004, School of Computer Science University of Birmingham.
- [6] Nicolove, L., « Stéganographie, Détection de messages cachés », Travail de diplôme (décembre 2008), Haute Ecole Spécialisée de Suisse occidentale.
- [7] Petitcolas, F.A.P., Anderson, R. J., Kuhn, M. G., « Information Hiding|- A Survey » Proceedings of the IEEE, special issue on protection of multimédia content, (July 1999) 87(7):1062{1078.
- [8] Bouchama, S., « Le tatouage des images appliqué à l'imagerie médicale », Mémoire de magister en électronique (2007), Ecole nationale polytechnique, Algérie.
- [9] Grover, D., « Steganography and Watermarking of Digital Data » Information Hiding. IH'98, Portland, Oregon, USA.
- [10] Bender, W., Gruhl, D., Morimoto, N., Lu, A., "Techniques for Data hiding," IBM System Journal, (1996), VOL.35, 3&4.
- [11] Nikolaidis N.; Pitas I, "Robust Image Watermarking in the Spatial Domain," Signal Processing, (1998), Volume 66, Number 3, pp 385-403.
- [12] Maity, S.P., Kundu, M.K., « Robust and Blind Spatial Watermarking in Digital Image », Proceedings of the Third Indian Conference on Computer Vision, Graphics & Image Processing, Ahmadabad, India, (December 16-18. 2002).
- [13] Dugelay, J.L., Rey, C., Roche, S., « Introduction au tatouage d'images » CORESA99, (14-15 juin 1999) Compression et représentation des signaux audiovisuels", Sophia-Antipolis.
- [14] Tian, J., "Reversible data embedding using a difference expansion," IEEE Trans. Circuits Syst. Video Technol , (Aug. 2003), vol.13, no. 8, pp. 890-896.
- [15] Wu, D., Tsai, W., « A steganographic method for images by pixel-value differencing », Pattern Recognition Letters, (2003), Elsevier.

- [16] Shi, Y.Q., Kim, H.-J., Katzenbeisser, S., "An Improved Reversible Difference Expansion Watermarking Algorithm" IWDW (2007), LNCS 5041, pp. 254–263, (2008). Springer-Verlag Berlin Heidelberg (2008).
- [17] Kim, H.J., Sachnev, V., Shi, Y.Q., Nam, J., Choo, H.G., "A novel difference expansion transform for reversible data embedding" IEEE Transaction Information Forensics and Security 3 (3), (2008), 456–465.
- [18] Hu, Y., Lee, H-K., Li, J.," DE-Based Reversible Data Hiding With Improved Overflow Location Map" IEEE transactions on circuits and systems for video technology, (february 2009), vol. 19,no 2.
- [19] Kawaguchi, E., Eason, R.O., « Principle and applications of BPCS- Steganography » SPIE's International Symposium on Voice, Video, and Data Communications, (1998).
- [20] Mohanty, S.P., Ramakrishnan, K.R., « A DCT Domain Visible Watermarking Technique for Images », IEEE International Conference on Multimedia and Expo, (2000).
- [21]: Wong, P.H.W., Au, O.C., « A Capacity Estimation Technique for JPEG-to-JPEG Image Watermarking », IEEE trans. circuits syst. video technol, (2003), vol. 13, no 8 (37 ref.), pp. 746-752.
- [22] Piva, A., Barni, M., Bartolini, F., Cappellini, V., « DCT-based Watermark Recovering without Resorting to the Uncorrupted Original Image », International Conference on Image Processing, ICIP (1997)
- [23] Barni, M., Bartolini, F., Cappellini, V., Piva, A., «A DCT- domain system for robust image watermarking»,signal processing, , (1998), 66 : 375-372.
- [24] Ashourian, M., Ho, Y-S., « Blind Image Data Hiding in the Wavelet Domain », Springer: Advances in Multimedia Information Processing - PCM, (2004), pp. 747-754.
- [25] Piva, A., Bartolini, F., Caldelli, R., « Self Recovery Authentication Of Images In The Dwt Domain ». Int Journal of Image and Graphics , (2005),vol 5 N 1 pp. 149-166.
- [26] Liang, X., Wu, X., Huang, J., « Reversible Data Hiding for Image Based on Histogram Modification of Wavelet Coefficients », CIS (2) (2005), Volume 3802/2005,pp. 573-580.
- [27] Xia, X., Boncelet, C., Arce, G., «A Multiresolution Watermark for Digital Images» Proc. IEEE International Conference on Image Processing, (Oct.1997), vol.3, pp. 548-551.
- [28] Grosbois, R., Ebrahimi, T. «Watermarking in the JPEG 2000 domain », Proceedings of MMSP'01 - Image/Video watermarking, (2001), p. 339-344.
- [29] Toutant, J.L., Puech, W., Fiorio, C., « Amélioration de l'Invisibilité par Adaptation de la Quantification aux Données à Insérer ». GRETSI'05,(September 2005), Louvain-la-Neuve, Belgium.
- [30] Chang, C., Lin, C., Tseng, C., Tai, W., « Reversible hiding in DCT-based compressed images », Information Sciences: an International Journal, (July 2007), v.177 n.13, p.2768-2786.

- [31] Mobasseri, B.G., « A Foundation for Watermarking in Compressed Domain », IEEE Signal Processing Letters, (May 2005), Vol. 12, No. 5.
- [32] Lee, H., Song, G., Kim, M., Yoo, K., Lee, W., « A Digital Watermarking Scheme in JPEG-2000 Using the Properties of Wavelet Coefficient Sign ». ICCSA(1) (2004), LNCS 3043, pp. 159-166.
- [33] Tung-Shou, C., Jeanne, C., Jian-Guo, C., « A simple and efficient watermark technique based on JPEG2000 Codec », Springer: Multimedia systems, ISSN 0942-4962, (2004) , vol. 10, no 1, pp. 16-26 .
- [34] Katsutoshi, A., Hiroyuki, K., Hitoshi, K., « The method for embedding binary data into JPEG2000 bitstreams based on the layer structure », journal: IEIC Technical Report (Institute of Electronics, Information and Communication Engineers, (2001), VOL.101, NO.387 (IE2001 84-93);PAGE.17-24.
- [35] Chen, J., Chen, T.S., Lin, C.N., Cheng, C.Y. « A bitrate controlled data hiding scheme for JPEG2000 ». International Journal of Computers and Applications , (2010), 32(2) ,238–241.
- [36] Su, P.C., Kuo, C.J., « Steganography in JPEG2000 compressed images ».IEEE transaction on Consumer Electronics, (2003), 49(4) :824–832.
- [37] Thomos, N., Boulgouris, N.V., Kokkinou, E., Strintzis, M.G., « Efficient data hiding in JPEG2000 images using sequential decoding of convolutional codes ». In International Conference in Digital Signal Processing (2002), volume 2, pages 710–720.
- [38] Noda, H., Tsukamizu, Y., Niimi, M., « JPEG2000 Steganography Possibly Secure Against Histogram-Based Attack » springer, Lecture Notes in Computer Science, Volume 4261, pp. 80-87, 2006
- [39] Vassaux, B., Nguyen, P., Bas, P., Chassery, J-M., « A survey on attacks in image and video watermarking », SPIE, Seattle, (juillet 2002).
- [40] Petitcolas, P., Anderson, J., Kuhn, G., « Attacks on copyright marking systems ». In Proceedings Second workshop on information hiding, Portland, Oregon, (April 1998).
- [41] B.Vassaux. Technique multicouches pour le tatouage d'images et adaptation aux flux vidéo MPEG-2 et MPEG-4.PhD thesis, INPG de Grenoble, France, Novembre 2003
- [42] Derraz, F., Beladgham, M., Khelif, M., « Mesure Objective de la Qualité d'Image Médicale Dérivée de l'Index de Similarité Structurale », MAJECSTIC04, Calais, (octobre 2004).
- [43] Autrusseau, F., Le Callet, P., Campisi, P., "Métriques de qualité pour le benchmarking d'algorithmes de tatouage", journée GDR-ISIS - Codage et qualité : outils et méthodologies d'évaluation, May, 3rd (2007).
- [44] Watson, A. B., « DCT quantization matrices visually optimized for individual images ». In Proc. SPIE, volume 1913, page 202\_216, 1993.

- [45] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, (Avril 2004).
- [46] Skodra. A.N., Christopoulos. C.A., Ebrahimi. T., « The JPEG 2000 Still Image Compression Standard », IEEE signal processing magazine, (2001), Vol. 18, 36-58.
- [47] Santa-Cruz. D., Ebrahimi. T., Askelof. J., Larsson. M., Christopoulos. C.A., « JPEG 2000 still image coding versus other standards », in proc (SPIE 2000), California, Vol 4115, 446-454.
- [48] Santa-Cruz. D., Ebrahimi. T., « An analytical study of JPEG 2000 functionalities », in proc of IEEE (ICIP 2000), Canada, Vol 2, 49-52.
- [49]. Aouadi. I., « Optimisation de JPEG 2000 sur système sur puce programmable », doctorat sciences appliquées, électronique, ENSTA, (2005)
- [50] Caruso. A., « La norme JPEG 2000 – une nouvelle venue dans la production et le transport de contenu », Revue des technologies de radio canada-numéro 2, (2006). [http : //www.cbc.radio-canada.ca](http://www.cbc.radio-canada.ca)
- [51] Taubman. D., « High Performance Scalable Image Compression with EBCOT », IEEE Transactions on Image Processing, (2000), vol. 9, N°7.
- [52] Boliek. M., Mmajani. E., Houchin. J.S., Kasner. J., Larsson. M., « JPEG 2000 part II final Committee draft ». ISO/IEC JTC1/SC29 WG1 N2000, (2000), 160–163.
- [53] Fukuhara. T., Singer. D., « Motion JPEG 2000 final draft international standard 1.1 (including COR1) », (2002).
- [54] Buckley. R., Sharpe. L., Mcpartlin. S., « JPEG 2000 Part 6. Final Draft International Standard (ISO FORMAT) ». ISO/IEC JTC1/SC29 WG1 N2813, (2003).
- [55] Ebrahimi. T., Rollin. C., « JPSEC working draft ». ISO/IEC JTC1/SC29 WG1 N2923, (2003).
- [56] Colyer. G.J., Clark. R.A., « JPEG 2000 and interactive applications ». IEEE transactions on consumer Electronics, Vol 49, N° 4, 2003.
- [57] Omno, P., Lambert-Nebout. C., Nicholson. D., « JPEG 2000: present and future of the new standard », in proc of EUSIPCO 2002, (2002), Vol 3, 47-50.
- [58] Brislawn. C., Schelkens. P., « JP3D: Scope and Requirements document, version 4.0 ». ISO/IEC JTC1/SC29 WG1 N2919, (2003).
- [59] Nicholson. D., Frescur. F., « JPWL working draft ». ISO/IEC JTC1/SC29 WG1 N2847, (2003).
- [60] Christopoulos. C.A., Skodas. A.N., Ebrahimi. T., « The JPEG 2000 still image coding system: an overview », IEEE transaction on consumer electronics, (2000), Vol.46, N° 4, 1103-1127,.

- [61] Skodas. A.N., Christopoulos. C.A., Ebrahimi. T, «The JPEG 2000 still image coding system: an overview», in proc of RECA00D, (2000), 359-366.
- [62] Anastassopoulos. G.K., Skodas. A.N., «JPEG 2000 ROI coding in medical imaging applications», Democritus university of Greece, university of Patras, Greece.
- [63] Christopoulos. E.B., Skodas. A.N., Reed. T.R., Christopoulos. A., «On the JPEG 2000 Implementation on different computer platforms», in proc of SPIE 2000, (2000), Vol 4115, 561-569.
- [64] Taubman. D., Marcellin. M., «JPEG 2000: Image Compression Fundamentals, Standards and Practice», The Springer International Series in Engineering and Computer Science,(2002) Vol. 642, ISBN: 978-0-7923-7519-7.
- [65] Adams. M.D., «the JPEG 2000 still image compression standard» (2002), ISO/IEC JTC 1/SC 29/WG 1(ITU-T SG 16), N° 2412.
- [66] [www.jpeg.org](http://www.jpeg.org)
- [67] <http://www.datahide.com/BPCSe/abc-vs-cgc-e.html>
- [68] Mostefai, F., Bessalah, H., Brahim, Z., "FPGA implementation of the AES encryption and decryption algorithms", International Arab Conference on Information Technology: ACIT'2007, held at the Arab Academy for Science & Technology and Maritime Transports (AAST) branch in collaboration with Tishreen University, Syria on (November 26-28, 2007).
- [69] Kawaguchi, E. Niimi, M., "Modeling Digital Image into Informative and Noise-Like Regions by Complexity Measure," Information Modeling & Knowledge Bases IX, IOS Press, pp.255-265, (April 1998).
- [70] Kawaguchi, E. Eason, O., « Principle and applications of BPCS Steganography ». [www.datahide.com](http://www.datahide.com)
- [71] Khaire and al. « Review: Steganography – Bit Plane Complexity Segmentation (BPCS) Technique ». International Journal of Engineering Science and Technology, (2010) Vol. 2(9), 4860-4868.
- [72] Srinivasan, Y., « High capacity data hiding system using BPCS steganography » Thèse en ingénierie électrique, université Texas Tech, décembre 2003.
- [73] Michot, J., « les formats de compression d'images » Rapport, IUT de Tours. (2004).