Stage pratique DEA : Tatouage d'images et Cryptographie : Pour assurer confidentialité, intégrité et authentification des images médicales et insérer des données confidentielles

Romain Hérault

20 août 2004

Remerciements

Je tiens à remercier :

- Monsieur le Professeur Jean-Jacques Le Jeune, chef du service, pour son accueil au sein du laboratoire de Médecine Nucléaire et de Biophysique du CHU d'Angers.
- Madame le Docteur Christine Cavaro-Ménard, Maître de conférences et responsable de l'unité de Traitement d'Images Médicales au LISA, pour l'encadrement de ce stage, ses conseils avisés et sa disponibilité.
- Messieurs le Docteur Alain Le Duff, le Docteur Guy Plantier, le Docteur Daniel Schang, enseignants-chercheurs à l'ESEO, pour avoir su motiver mon désir d'effectuer ce DEA et de poursuivre dans le domaine de la recherche.
- Aymeric Histace et Xavier Baty, doctorants au LISA, Tangy Le Fol, Benjamin Vallet, Jean-Pierre Kinn, Fabien Lefèvre, Pierre-Yves Daniau-Clavreul, stagiaires au LISA, pour leur soutien, aide et conseils ainsi que leur bonne humeur.

Synthèse du Stage

Service de Médecine Nucléaire
Hôpital Larrey
Rue Larrey
49000 Angers

LISA FRE 2656 CNRS
Université d'Angers
62, avenue Notre Dame du Lac
49000 Angers

Hérault Romain
Université d'Angers
Février-Septembre 2004

Tatouage d'images et Cryptographie

Le déploiement des PACS et des dossiers médicaux électroniques nécessite un effort de sécurité accru en termes de confidentialité, fiabilité et disponibilité. Le tatouage de données (ou watermarking) permet de renforcer la sécurité en faveur de la fiabilité des données images par l'insertion dans l'image source d'annotations (authentification) et/ou par l'insertion d'une signature de l'image source (intégrité). Le watermarking permet également d'intégrer dans les images des données confidentielles telles que des informations physiologiques (ECG ...) et/ou diagnostiques, ce qui assure une plus grande confidentialité aux données patients. Pour renforcer cette confidentialité et obtenir une authentification rigoureuse, nous devrons insérer une étape de cryptologie dans la chaîne de tatouage.

Résultats

Notre étude a porté sur :

- la définition des caractéristiques des méthodes cryptologiques,
- la mise en œuvre des méthodes cryptologiques adaptées au domaine de l'imagerie médicale,
- l'optimisation du programme complet de tatouage et l'insertion de la cryptologie,
- la quantification des dégradations d'encodage pour des études futures du laboratoire.

Les travaux effectués durant ce stage ont permis :

- d'identifier les caractéristiques et donc les méthodes de cryptologie optimales pour notre application,
- de mettre en œuvre toutes les briques logicielles nécessaires à la réalisation de la chaîne de tatouage,
- d'implémenter les nouvelles méthodes d'évaluation de la dégradation.

Mots-Clés : Codage, tatouage réversible, confidentialité, disponibilité, fiabilité, intégrité, authentification, imagerie médicale, cryptologie, information diagnostique.

Table des matières

Chapitre 1

Présentation du sujet

L'utilisation des images en médecine et dans le domaine de la recherche est en pleine expansion. Le déploiement des PACS (Picture Archiving and Communication Systems) et des dossiers médicaux électroniques nécessitent un effort de sécurité accrue. D'après G. Coatrieux et al. [1], ces règles de sécurité reposent sur 3 principes fondamentaux : Confidentialité, Fiabilité et Disponibilité. La confidentialité assure le fait que seules les personnes autorisées peuvent accéder à l'information. La fiabilité de l'information doit être comprise comme étant l'association d'une double vérification : l'intégrité et l'authenticité des données. L'image ne doit pas avoir subi de modification (intégrité) volontaire ou non depuis son acquisition ou après tout traitement considéré faisant partie d'un protocole entièrement défini et connu. L'image doit de plus être en adéquation avec l'identité du patient (authenticité). La disponibilité du système d'information (maintenance, astreinte informatique) pour les utilisateurs est évidemment la troisième règle de sécurité. Pare-feu, contrôle d'accès, antivirus, cryptographie, signature électronique et accréditation logicielle sont les garants actuels de l'intégrité et de la confidentialité des données médicales avec les limites humaines et matérielles que l'on connaît. Le tatouage de données (ou watermarking) permet de renforcer la sécurité en faveur de la fiabilité des données images par l'insertion d'annotation dans l'image source (authentification) et/ou par l'insertion d'une signature de l'image source (intégrité). Le watermarking permet également d'intégrer dans les images des données confidentielles telles que des informations physiologiques (ECG ...) et/ou diagnostiques, ce qui assure une plus grande confidentialité aux données patients.

Une première étude [2] a abouti à la mise en œuvre d'un algorithme réversible de tatouage d'image qui exploite la corrélation spatiale des niveaux de gris d'une image. Cette méthode tatoue la marque dans l'erreur de prédiction locale des pixels de l'image. Elle s'inspire de l'approche de Tian présentée dans [3]. L'algorithme de prédiction-expansion permet d'augmenter de façon significative la capacité d'espace libéré pour l'insertion du résumé de l'image (intégrité), de la globalité de l'en-tête

DICOM (authentification) et de certaines données physiologiques et/ou diagnostiques. Mais l'algorithme prédiction-expansion ne réalise pas de contrôle d'intégrité et d'authentification, au sens cryptographique du terme, des données insérées (signatures, données diagnostiques).

L'objectif de ce stage sera donc :

- de faire l'état de l'art des méthodes de cryptologie et de leurs caractéristiques,
- d'analyser les besoins dans le domaine de l'imagerie médicale afin de définir les caractéristiques des méthodes de cryptographie à utiliser,
- d'insérer la cryptographie dans la chaîne de tatouage,
- d'améliorer la programmation actuelle de la chaîne afin de la rendre plus fonctionnelle,
- de mettre en œuvre des méthodes de quantification des dégradations engendrées par l'encodage afin de réaliser par la suite une étude des méthodes de tatouage irréversibles plus robustes aux attaques potentielles telles que le zoom, le changement de dynamique...

Chapitre 2

État de l'art

2.1 Méthodes de tatouage réversibles

Le tatouage d'une image consiste à insérer des données (ou marque) à l'intérieur de l'image.

Les techniques de tatouage peuvent êtres séparées en 2 catégories : les méthodes réversibles et les méthodes irréversibles. Après extraction et vérification de la validité de la marque, les méthodes réversibles sont capables de fournir un duplicata exact de l'image originale.

Le critère de réversibilité est bien entendu primordial pour des considérations d'ordre éthique : l'image est en partie à la source d'un diagnostic.

Suite à une recherche bibliograhique sur le tatouage réversible, nous nous sommes aperçus que seuls quatre algorithmes réversibles répondent à notre objectif: tatouer l'image avec une faible dégradation de l'image tatouée. L'image tatouée devra être globalement similaire à l'image originale afin de servir de support visuel au diagnostic réalisé sur l'image originale. Les algorithmes retenus sont :

1. l'algorithme de Fridrich, algorithme en cours d'implémentation [4][5],
2. l'algorithme de Celik [6],
3. l'algorithme différence-expansion proposé par Tian [3],
4. l'algorithme prédiction-expansion présenté dans [2].

Les publications de ces articles sont relativement récentes (1999-2003). Les deux dernières méthodes, proches dans leur conception, seront présentées dans le chapitre 3 puisque la méthode prédiction-expansion qu'il m'a fallu optimiser, a été developpée au LISA.

2.1.1 La méthode de Fridrich

- Les pixels de l'image sont assemblés en groupes de pixels (blocs...).

- Deux fonctions commutatives $f(x1,x2,...)$ et $F(X)$ sont définies :
  - la fonction f dite de discrimination cherche à mesurer la régularité des groupes de pixels. Cette fonction peut-être, par exemple, la fonction 'variation' $f(x_1,x_2,...,x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$
  - la fonction F de permutation tel que $F(F(X)) = X$. Cette fonction peut-être par exemple la fonction qui à 1 associe 0 et à 1 associe 0, à 3 associe 2 et à 2 associe 3, ...
- Pour chaque groupe de pixels est attribuée une catégorie R, S ou U suivant le schéma suivant :
  - R (Regular) si $f(F(G)) > f(G)$
  - S (Singular) si $f(F(G)) < f(G)$
  - U (Unusable) si $f(F(G)) = f(G)$
- La carte binaire de localisation contenant les positions de R et S est compressée et rajoutée aux données à embarquer.
- Le code 0 est attribué au groupe R, le code 1 au groupe S. Pour faire changer d'état un groupe de pixels, il suffit de lui appliquer la fonction F.
- Lors de la reconstruction, il suffira de réappliquer la fonction de permutation, suivant la carte de localisation, sur un pixel pour obtenir la valeur initiale du pixel.

Cet algorithme récent (2002) n'a pas encore été testé au laboratoire.

Exemple

Soit le groupe de pixel G suivant :

A	B
C	D

La fonction de discrimination est calculée sur ce groupe :

$$f1 = f(A,B,C,D)$$

La fonction de permutation est calculée pour chaque pixel :

$$A' = F(A)\ B' = F(B)\ C' = F(C)\ C' = F(C)$$

La fonction de discrimination est calculée sur le nouveau groupe :

$$f2 = f(A',B',C',D')$$

Si $f1 = f2$ le groupe est dit "unusable" et donc non retenu.
Si $f1 > f2$ le groupe est dit "singular" et équivaut à la valeur 1.
Si $f1 < f2$ le groupe est dit "regular" et équivaut à la valeur 0.

Si le groupe ne correspond pas à la valeur désirée pour le marquage, les valeurs (A,B,C,D) du groupe de pixels sont substituées par les valeurs du groupe permuté (A',B',C',D').

Lors de la reconstruction suivant la carte embarquée, il suffira de réappliquer la fonction de permutation sur (A',B',C',D') pour obtenir (A,B,C,D).

2.1.2 La méthode de Celik

Celik and al. présentent [6] un algorithme de tatouage réversible, nommé LAW (Localized Lossless Authentication Watermark). Une opération de quantification est tout d'abord effectuée afin de libérer un espace variable dans l'image originale. La compression du flux des résidus de l'opération de quantification permet d'assurer la réversibilité du marquage et d'insérer le flux de données nécessaire pour le contrôle d'intégrité. Le procesus peut être rendu hiérarchique afin d'effectuer la localisation d'éventuelles attaques (volontaires ou non). Le processus d'authentification quant à lui est réalisé par un algorithme de cryptographie combinant clés privée et publique.

Opération de quantification

Une fonction de quantification scalaire de niveau L est à la base du processus. Soit s le vecteur représantant le signal original.

$$Q_L(s) = L\lfloor\frac{s}{L}\rfloor \tag{2.1}$$

$\lfloor * \rfloor$: représente la fonction du plus grand entier inférieur ou égal à *.

Le tatouage (Equation 2.2) et le détatouage (Equation 2.3) sont effectués par le système d'équations (2.2) suivant :

$$s_w = Q_L(s) + w \tag{2.2}$$

$$w = s_w - Q_L(s_w) = s_w - Q_L(s) \tag{2.3}$$

s_w est le signal marqué. w représente la marque constituée de symboles w_i dit L-aire symboles, c'est à dire $w_i \in \{0,1,...,L-1\}$.

Opération de conversions L-air $\Leftrightarrow$ binaire

L'utilisation d'une marque constituée de caractères de type L-aire nécessite une conversion binaire vers L-aire pour les flux de bits entrant et une conversion L-aire vers binaire pour récupérer le flux binaire du message. Ces conversions peuvent entraîner des dépassements de capacité (underflow ou overflow). Dans le cas d'une image codée sur 8 bits (de 0 à 255 valeurs) par exemple, si L=6, $Q_L(s) = 252$ et

w=5 alors le pixel marqué aura la valeur $s_w = 257$. Ce résultat ne peut pas être codé entre 0 et 255, il s'agit donc d'un dépassement de capacité ou overflow. Celik propose un codage arithmétique **modifié** pour éviter ces risques de dépassement et effectuer les conversions .

Codage arithmétique modifié

Le codage arithmétique modifié est basé sur le codage arithmétique présenté en annexes (cf. A.3). Le flux de bits à tatouer est représenté par un nombre H.

$$H = 0{,}h_0h_1h_2... \; H \in [0{,}1[. \tag{2.4}$$

Au départ R représente l'intervalle [0,1[. L'image possède des pixels de valeurs comprises entre 0 et s_{max}. Pour assurer la réversibilité du système le nombre H devra être marqué sur un nombre suffisant de valeurs de s (ici les pixels correspondants sont connexes). Pour la première valeur de s, la valeur de $Q_L(s)$ est calculée ainsi qu'un nombre de niveaux N tel que :

$$N = \min(L{,}s_{max} - Q_L(s)) \tag{2.5}$$

L'équation 2.5 permet d'éviter les risques de dépassement de capacité. Le codage arithmétique modifié consiste alors à diviser R en N intervalles équivalents R_0 à R_{N-1}. Puis l'intervalle satisfaisant $H \in R_n$ ($n \in [0{,}N-1]$) est sélectionné et la marque à insérer est w=n. Le nouvel intervalle R est défini par $R = R_n$ pour la prochaine valeur de s.

Quantité d'information embarquée

L'un des avantages du codage arithmétique est que chaque caractère peut être codé sur un nombre non-entier de bits (contrairement à un algorithme du type Huffman). En effet, cet algorithme ne code pas les fichiers caractère par caractère mais par chaînes de caractères, plus ou moins longues suivant la capacité de la machine à coder des réels plus ou moins grands. Si un caractère apparaît avec une probabilité de 90%, la taille optimale du code serait de 0.15 bit [1].

Chaque signal original embarque une marque L-aire w d'après l'équation (2.2), provenant du résultat du codage arithmétique modifié. D'après la théorie de l'information chaque w représentera une capacité $C = \ln(L)$ mesurée en bps (bit per sample).

1. Quantité information = ln (probabilité)= $\ln 0{,}9 = 0{,}105 \simeq \ln 2^{0{,}15}$

Réversibilité : Méthode G-LSB sans perte

Basée sur la méthode LSB présentée en annexe (Cf. A.1), la méthode G-LSB est capable de créer une image marquée avec de faibles distorsions mais le processus reste irréversible. En effet lors de l'opération de quantification, un résidu r est créé (Equation 2.6) puis remplacé par le signal de la marque.

$$r = s - Q_L(s) \tag{2.6}$$

$$s = Q_L(s) + r = Q_L(s_w) + r \tag{2.7}$$

La réversibilité est obtenue en embarquant le résidu r comme flux de données dans la marque à insérer. Lors du détatouage, les valeurs de r seront récupérées et le signal original pourra être recalculé (Equation 2.7). Pour optimiser la taille du flux des données r, l'algorithme de compression CALIC [7] est utilisé. La compression est un élément majeur pour améliorer la capacité d'embarquement de la méthode G-LSB sans perte. Plus $C_{Résidu}$ r sera faible, meilleure sera la compression et donc plus la capacité C_{G-LSB} sans perte sera importante (Equation 2.8)

$$C_{G-LSB \text{ sans perte}} = C_{G-LSB} - C_{Résidu \text{ r}} \tag{2.8}$$

2.2 Cryptologie

2.2.1 Introduction à la cryptologie

Un peu de vocabulaire

Il existe 3 grands thèmes souvent associés à la cryptologie eux-même divisés en sous-thèmes [8] :

- Cryptographie.
  - Cryptologie : l'art de coder un message pour le rendre illisible pour les personnes auxquelles il n'est pas destiné.
  - Cryptanalyse : l'art de casser les codes.
- Stéganographie : l'art de communiquer un message sans que la transmission du message soit détectée.
  - Tatouage.
  - Image autoréparatrice (image embarquant des données qui permettent de la reconstruire en cas d'altération).
  - Evasion de fréquences (émission d'un message qui ressemble à du bruit).
  - Message subliminale.

- Empreintes : l'art de faire correspondre à un message une empreinte ou signature qui identifie le message.
  - Détection d'erreur.
  - Fonction de hachage (développée ci-dessous).

Fonction de hachage

Elle sert au calcul d'une empreinte du message. L'empreinte (ou résumé) est un chiffre identifiant, idéalement de façon unique, le message. Une fonction de hachage est considérée comme sûre si deux messages probables ne donnent pas le même résumé (collision) et si à partir de l'empreinte on ne peut présumer du contenu du message.

Une fonction de hachage permet de vérifier l'intégrité d'un message.

Il existe différentes fonctions de hachage décrites dans la littérature. Le MD5 (traditionellemnt utilisé cf table 2.2) est de plus en plus remplacé par le SHA-256 (cf table 2.2) considéré comme plus sûr : d'une part il est calculé sur 256 bits (moins de risque de collision) d'autre part il est plus difficile de contourner sa protection cryptographique.

2.2.2 Caractéristiques des méthodes de cryptage

Traditionellement, les termes "crypter" et "décrypter" un message sont utilisés, mais il y a une volontée de remplacer ces termes par "coder" et "décoder" pour ne pas choquer certaines cultures; decrypter pouvant signifier profaner.

Pour définir les principales caractéristiques des méthodes de cryptage, un ensemble de personnages aux rôles spécifiques est souvent cité dans les publications, notamment :

- Alice : l'émettrice du message,
- Bob : le récepteur du message,
- Eve : une espionne qui écoute la transmission,
- Martin : un espion qui écoute la transmission et qui peut la modifier,
- Norbert : un tiers de confiance.

Cryptographie à clé secrète (mode symétrique)

Alice et Bob possédent un secret en commun qui sert de clé de cryptage et de décryptage. Alice doit avoir confiance dans Bob et inversement. Ce mode de cryptographie est le plus ancien, le plus répandu et le plus sûr. Il est néamoins fragilisé par la distribution des clés. En effet, pour chaque couple (Alice, Bob), il doit exister une clé unique; voire pour chaque transmission (Alice et Bob doivent

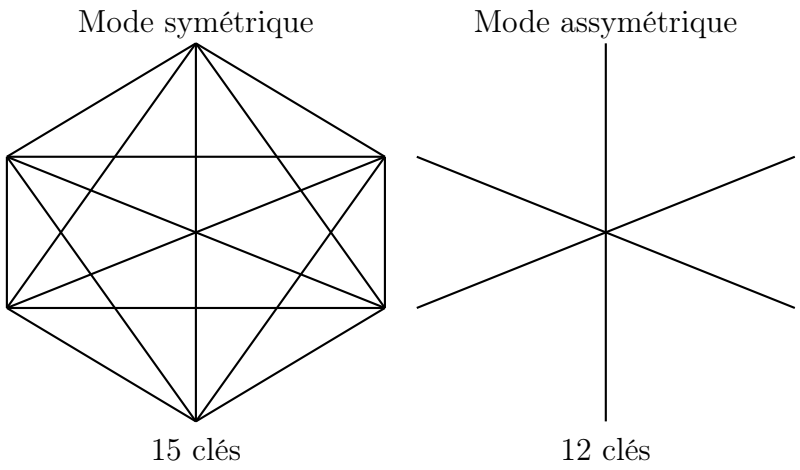

FIG. 2.1 – *Répartition des clés pour 6 intervenants*

générer une clé qui sera jetée à la fin de la communication). Le nombre de clé devient rapidement grand ce qui pose le problème de la transmission des clés par un canal sûr (cf figure 2.1)

Cryptographie à clé publique (mode assymétrique)

Ce mode de cryptographie a permis de résoudre le problème de la distribution des clés : La clé servant au codage (clé publique) est différente de la clé servant au décodage (clé privée). Alice utilise la clé publique de Bob pour coder son message; Bob utilise sa clé privée pour décoder le message. Il suffit seulement de 2 clés par intervenant.

En plus du cryptage, ce mode permet l'authentification des messages : Alice utilise sa clé privée pour coder le message ; Bob utilise la clé publique d'Alice pour décoder le message. Comme seule la clé privée d'Alice correspond à la clé publique d'Alice, si le message est intelligible, Bob est assuré de la provenance du message.

Les algorithmes à clé publique sont basés sur une difficulté de calcul (inversion d'une exponentiation par exemple) et sont donc moins sûrs que les algorithmes à clé secrète. Une combinaison des modes est le plus souvent utilisée (dans [9] ou [10] par exemple), le mode clé publique servant de canal sûr à la transmission d'un clé secrète entre Alice et Bob.

2.2.3 Tableau récapitulatif des bibliothèques cryptographiques disponibles

Les tableaux récapitulatifs (Tab. 2.1 et Tab. 2.2) présentent un bilan des bibliothèques cryptographiques disponibles sur Internet. Chaque ligne représente un programme ou une librairie, chaque colonne une fonctionalité. Les détails concernant les licences d'utilisation en vigueur pour ces bibliothèques sont données ci-dessous.

Les licences d'utilisation

Les licences MIT (Masachusets Institute of Technology) :
Il existe plusieurs types de licences d'utilisation en usage au MIT, elles permettent sous conditions strictes d'avoir accès au code source et de le modifier pour ses besoins, la redistribution du code est généralement limitée.

La licence GPL (General Public Licence) :
Issue des licences MIT, elle garantit en plus de l'accès aux sources, la liberté d'utilisation et de redistribution du logiciel à condition que tout programme issu de ce code ou utilisant ce code (même à travers une bibliothèque) soit publié sous cette même licence.

La licence LGPL (Lesser General Public Licence) :
Plus souple que la GPL, elle autorise l'utilisation par un programme non GPL ou LGPL d'une bibliothèque LGPL en liaison dynamique.

Les licences BSD (Berkeley Software Distribution) :
Licence en usage à l'université de Berkeley, très souple au niveau des conditions d'utilisation et d'accès au code source. Il n'est même pas nécessaire de préciser la provenance d'un code BSD et son auteur dans un programme tiers l'utilisant (la loi française nous y oblige quand même).

Les logiciels dits "freeware" :
Logiciels libres d'utilisation mais dont le code source est partiellement ou complétement clos. Ces logiciels sont souvent freeware pour une utilisation en recherche ou enseignement et proriétaire pour une application commerciale.

Les logiciels dits "shareware" (ou partagiciels) :
Logiciels libres d'utilisation dans un temps imparti ou dans une limitation de ces fonctionalités. Le logiciel devient complétement fonctionel après rétribution de son auteur. Code source clos.

Les logiciels dits "propriétaires" :
Logiciels qui peuvent être partiellement bridés dans leur fonctionnement et dont l'accès au code source est quasiment impossible sauf accords commerciaux très onéreux.

TAB. 2.1 – *Tableau récapitulatif : partie 1*

Appellation	url	Licence	Source
CTCjava	http://www.bifroest.demon.co.uk/ctc/	free?	oui
beecrypt	http://sourceforge.net/projects/beecrypt/ ?		
borzoi	http://www.dragongate-technologies.com/products.html	GPL	oui
certlib	?		
cryptix	http://www.cryptix.org/	BSD-like	oui
cryptlib	http://www.cs.auckland.ac.nz/ pgut001/cryptlib/	free nc	oui
crypto++	http://www.eskimo.com/ weidai/cryptlib.html	free	oui
cryptokit	http://packages.debian.org/unstable/libs/libcryptokit-ocaml	LGPL	oui
CTClib	http://www.bifroest.demon.co.uk/ctc/	free?	oui
dealib	?		
EGD	http://egd.sourceforge.net/	GPL/MIT	oui
GnuPG	http://www.gnupg.org	GPL	oui
libdes	http://linux.maruhn.com/sec/libdes.html	MIT	
libgcrypt	http://www.gnu.org/directory/security/libgcrypt.html	LGPL	oui
libmcrypt	http://mcrypt.hellug.gr/	GPL	
libTomCrypt	http://libtomcrypt.org/	free	oui
macCTC	http://www.bifroest.demon.co.uk/ctc/	free?	oui
mhash	http://mhash.sourceforge.net/	LGPL	oui
MIRACL	http://indigo.ie/ mscott/	free for education	
opencl/botan	http://botan.randombit.net/		
pycrypto	http://www.amk.ca/python/code/crypto.html	free?	oui
RSAeuro			
SSLava	http://www.phaos.com/products/category/crypto.html		
SSLeay	http://www2.psy.uq.edu.au/ ftp/Crypto/		oui
tplockbox	http://sourceforge.net/projects/tplockbox/	GPL/LGPL	oui

Appellation	Possibilités																																				
	Aléatoire	Symétrique																																				
		AES (Rijndael)	ARC2	ARC4	Blowfish	CAST5	CAST128	DES	3DES	Diamond 2	gost	IDEA	ISSAC	Lion	lubyrack	MARS	Misty1	Noekon	panama	RC2	RC4	RC5	RC6	safer	sapphire	skipjack	seal	serpent	shark	square	TEA	XTEA	Twofish	wake
CTCjava					X	X		X	X																	X				X						
beecrypt																																					
borzoi		X																																			
certlib																																					
cryptix		X			X	X		X	X			X				X				X	X		X			X		X		X			X			
cryptlib		X			X		X	X	X			X								X	X	X				X										
crypto++		X		X	X		X	X	X	X	X	X			X	X			X			X	X	X			X		X	X	X		X		X	X
cryptokit		X		X				X	X																												
CTClib					X	X		X	X			X														X				X						
dealib																																					
EGD	X																																				
GnuPG	X	X		X	X	X		X	X																								X				
libdes								X	X																												
libgcrypt	X	X		X	X	X		X	X																								X				
libmcrypt		X		X	X	X		X	X		X					X				X				X				X					X		X	X
libTomCrypt		X			X	X		X	X									X		X	X	X	X	X		X						X	X		X	
macCTC					X	X		X	X																	X				X						
mhash											X																											
MIRACL		X																																			
opencl/botan		X		X	X		X	X	X		X	X	X	X	X		X			X		X	X	X		X	X	X		X	X	X	X	X	X	X
pycrypto		X	X	X	X	X		X	X			X										X															
RSAeuro								X	X																												
SSLava								X	X																												
SSLeay					X			X	X			X										X															
tplockbox		X			X			X	X																												

TAB. 2.2 – *Tableau récapitulatif : partie 2*

Appellation	Possibilités																																		
	Asymétrique																	Hash																		
	blumgoldwassr	Diffie-Helman	DSA	ElGamal	eleptic curve	luc	lucdif	lucelg	Nyberg-Rueppel	qNew	rabin	rabin-williams	rawDSA	RSA	RSASSA-PKCS1	RSASSA-PSS	CRC	HAVAL	HAS-160	MD2	MD4	MD5	panama	RIPEMD-128	RIPE-MD160	SHA-0	SHA-1	SHA-256...	SHS	TIGER	Whirpool					
CTCjava		X																									X								
beecrypt																																			
borzoi		X			X																						X								
certlib																																			
cryptix		X		X										X	X	X				X	X	X		X	X	X	X	X	X	X					
cryptlib		X	X	X										X						X	X	X			X		X								
crypto++	X	X	X	X	X	X	X	X		X	X	X		X	X		X	X		X		X	X		X	X		X	X	X					
cryptokit		X												X								X					X								
CTClib		X																									X								
dealib																																			
EGD																																			
GnuPG			X	X										X							X	X			X		X			X					
libdes																																			
libgcrypt			X	X										X							X	X			X		X			X					
libmcrypt																																			
libTomCrypt		X	X	X										X						X	X	X		X	X		X	X		X					
macCTC		X																									X								
mhash																	X	X			X	X			X		X			X					
MIRACL		X	X											X														X							
opencl/botan		X	X	X					X			X	X				X	X	X	X	X			X	X			X		X	X				
pycrypto			X	X						X				X						X	X				X										
RSAeuro		X												X								X							X						
SSLava		X												X															X						
SSLeay		X	X											X								X			X				X						
tplockbox														X								X					X								

Chapitre 3

Algorithme prédiction-expansion

Le principe de l'algorithme de tatouage prédiction-expansion, qui a été développé au LISA, pour des images en niveaux de gris (cas des images médicales), se base sur la différence d'amplitudes entre un pixel et sa valeur prédite grâce à son voisinage (les 3 pixels supérieurs et les 3 pixels inférieurs). Si cette différence répond à un critère précis (que nous verrons par la suite), alors elle va pouvoir être tatouée. Ainsi, la valeur de la différence va être légèrement modifiée et donc la valeur du pixel reconstruit aussi. L'image sera ainsi tatouée (ou marquée); les lignes paires sont tout d'abord traitées puis les lignes impaires, exceptée la première ligne pour pouvoir initier le processus de reconstruction.

3.1 Le tatouage

Il se décompose en six étapes résumées ci-après.

3.1.1 La première étape

La différence entre la valeur x d'un pixel et sa prédiction l (équation (3.1)), obtenue à partir des 3 pixels supérieurs et des 3 pixels inférieurs, est calculée et notée h (équation (3.2)). Deux listes de valeurs h et l sont donc obtenues.
Soit x l'amplitude du pixel en cours de traitement, y_i l'ensemble des 6 pixels supérieurs et inférieurs :

y_1	y_2	y_3
	x	
y_4	y_5	y_6

$$l = \frac{\sum a_i \dot{y}_i}{\sum a_i} \tag{3.1}$$

$$h = x - l \tag{3.2}$$

La transformation inverse des équations (3.1) et (3.2) est donnée par :

$$x = l + h \tag{3.3}$$

Si l'amplitude x doit être comprise entre 0 et $2^n - 1$, n étant le nombre de bits de codage des amplitudes de l'image source (ici 8), il est clair que :

$$h \leq 255 - l \; et \; -h \leq l \tag{3.4}$$

Il est dès lors possible d'insèrer les bits de tatouage b dans la valeur h selon l'une des équations suivantes :

$$h' = 2h + b \tag{3.5}$$

$$h' = 2\lfloor \frac{h}{2} \rfloor + b \tag{3.6}$$

$\lfloor x \rfloor$: fonction plus grand entier inférieur ou égal à x.
Pour prévenir les problèmes d'overflow et d'underflow, h' doit satisfaire :

$$|h'| \leq min(255 - l,l) \tag{3.7}$$

quelque soit la valeur de b (0 ou 1)
Ainsi, deux adjectifs caractérisent h :
h est dit **"extensible"** si

$$|2h + b| \leq min(255 - l,l) \tag{3.8}$$

quelque soit la valeur de b (0 ou 1)
h est dit **changeable** si

$$|2\lfloor \frac{h}{2} \rfloor + b| \leq min(255 - l,l) \tag{3.9}$$

quelque soit la valeur de b (0 ou 1)

3.1.2 La deuxième étape

Elle permet de créer quatre groupes (EZ, EN, CN et NC) disjoints des valeurs h selon des critères indiquant si elles sont aptes à être marquées et dans quelle mesure.

EZ : contient tous les h "extensibles" dont la valeur est soit 0 soit -1.

EN : contient tous les h "extensibles" $\notin$ EZ.

CN : contient tous les h "changeables" $\notin$ (EZ $\cup$ EN).

NC : contient tous les h "non-changeables".

3.1.3 La troisième étape

Elle permet de créer une carte de localisation binaire L des valeurs h qui seront marquées. Lors de cette étape, il est possible de contrôler les dégradations de l'image en scindant le groupe EN en deux groupes EN1 (dont les valeurs h seront tatouées selon l'équation (3.5)) et EN2 (dont les valeurs h seront tatouées selon l'équation (3.6)). Le critère de sélection des h de EN1 peut se baser sur la simple valeur de h : EN1 sera constitué des plus faibles valeurs de EN, d'où l'utilisation d'un seuil T tel que :

$$EN1 = \{h \in EN : |h| \leq T\} \ et \ EN2 = \{h \in EN : |h| > T\} \tag{3.10}$$

Pour $h \in EZ \cup EN1$, la valeur 1 est assignée à la carte L. Ces valeurs seront tatouées selon l'équation (3.5). Pour $h \in EN2 \cup CN$, la valeur 0 est assignée à la carte L. Ces valeurs seront tatouées selon l'équation (3.6).

Une études complémentaire m'a été demandée afin de fixer le seuil T en fonction du nombre de bits à insérer et ainsi optimiser le rapport entre EN1 et EN2 et donc la qualité de l'image marquée en fonction du flot de bits à tatouer.

3.1.4 La quatrième étape

Puisque l'équation (3.6) substitue le bit de poids faible des valeurs h, il est necéssaire pour obtenir un tatouage réversible de transmettre pour ces valeurs les bits d'origine. Cette étape permet de récupérer les bits de poids faible des valeurs h de EN2 et CN. Le flux de bits B est alors constitué ; B sera très utile lors de la reconstruction de l'image pour retrouver les valeurs originales des pixels.

3.1.5 La cinquième étape

Elle est véritablement l'opération de tatouage de l'image. Tout d'abord, la marque M à insérer est obtenue en concaténant la carte de localisation compressée L, le flux de bits de poids faible B et un résumé noté R de l'image à tatouer. C'est aussi à ce moment que peuvent être insérées d'autres informations relatives à l'image (nom du patient, informations diagnostiques ...) si la taille de la marque M ne dépasse pas le nombre de valeurs h extensibles ou changeables. Une fois la marque créée, la valeur h est tatouée selon l'équation (3.5) ou (3.6) selon son groupe d'appartenance (cf. étape 3).

3.1.6 La sixième étape

Elle permet de construire l'image tatouée selon l'équation (3.3) à partir de la liste des valeurs l (créée à l'étape 1) et la nouvelle liste des valeurs marquées h' (créée à l'étape 5).

3.2 La reconstruction de l'image originale

Pour savoir si l'image tatouée a été altérée d'une manière volontaire (falsification) ou non (zoom...), nous devons procéder à son démarquage, et extraire le résumé de l'image originale. Le résumé de l'image restaurée peut alors être comparé au résumé de l'image originale présent dans la marque. Si les deux résumés coïncident, alors l'image n'a pas été dégradée.

L'image restaurée est construite à partir de l'image tatouée en enlevant la marque présente dans les bits de poids faible de toutes les valeurs h marquées (c'est à dire changeables) puis en restaurant les valeurs h originales puis les amplitudes des pixels. Les cinq étapes de restauration de l'image sont décrites ci-après.

La première étape de restauration est identique à la première étape du tatouage et consiste à calculer la différence h entre la valeur de chaque pixel avec la prédiction l obtenue grâce aux pixels voisins. Les listes des valeurs différence h et des prédictions l sont calculées d'après les équations (3.1) et (3.2). La valeur l calculée est alors identique à celle calculée lors du codage du fait que les valeurs des pixels du voisinage sont identiques lors de ces deux phases. En effet, les lignes paires sont tout d'abord codées puis les lignes impaires. Lors du décodage, ce sont les lignes impaires qui sont tout d'abord traitées prenant en compte les valeurs tatouées des lignes paires comme pour la phase de codage.

La deuxième étape permet de classer toutes les valeurs h dans deux groupes disjoints :
CH contient tous les h changeables selon l'équation (3.9), ces h contiennent la marque.
NC contient tous les h non changeables (ne contiennent pas la marque).

La troisième étape consiste à extraire les bits de poids faible de toutes les valeurs h appartenant au groupe CH, c'est-à-dire théoriquement celles qui ont été tatouées. On obtient alors un flux de bits correspondant à la marque M insérée lors du tatouage duquel il suffit d'extraire les flux de bits L, B et R.

La quatrième étape permet de récupérer les valeurs h originales. La carte de localisation L et le flux de données B permettent de restaurer les valeurs h originales. Les h associés à la valeur 1 de la carte de localisation L sont restaurées grâce à une simple division entière (équation (3.11)). Les autres h de CH auront à utiliser le flux de bits B pour retrouver leurs valeurs initiales (équation (3.12)).

$$h = 2\lfloor \frac{h'}{2} \rfloor \tag{3.11}$$

$$h = 2\lfloor \frac{h'}{2} \rfloor + b_r \tag{3.12}$$

b_r étant le bit récupéré.

La dernière étape consiste à reconstruire l'image initiale grâce aux valeurs h restaurées et aux valeurs l de l'étape 1 (équations 3.3). On obtient ainsi l'image restaurée à partir de l'image tatouée.

3.3 Différence-expansion

L'algorithme prédiction-expansion est une modification profonde de l'agorithme différence-expansion de Tian [3]. Cet algorithme est basé sur la décomposition en ondelettes entières de Haar. Tian travaille alors sur 2 pixels voisins. Pour les valeurs x et y de pixels voisins, nous obtenons :

$$l = \lfloor \frac{x+y}{2} \rfloor \tag{3.13}$$

$$h = x - y \tag{3.14}$$

Les tailles des matrices L et H changent. En effet, pour l'algorithme prédiction-expansion, elles ont une taille similaire à la taille de l'image originale (aux première et dernière lignes près) alors que pour Tian, elles font la moitié de la taille de l'image originale. C'est pourquoi nous pourrons insérer davantage d'informations relatives au patient ou au diagnostic dans la marque de l'image. De plus, le fait de tenir compte d'un plus grand voisinage permet une meilleure prédiction du pixel central et donc une erreur (ou différence) plus faible ce qui augmente le nombre de pixels tatouables. Par contre, l'équation (3.2) montre la modification directe de x et non des deux pixels voisins comme Tian [1]. La dégradation peut donc être plus importante car supportée par un seul pixel. Une étude des dégradations engendrées sera donc nécéssaire.

1. L'image est conservée à l'échelle 1:2

Chapitre 4

Insertion de la cryptologie dans la chaîne de tatouage

Dans le titre même du projet, il est dit que le procédé utilisé doit permettre d'assurer la confidentialité, l'intégrité, et l'authentification des images. Les méthodes de tatouage réversibles permettent en combinaison avec les fonctions de hachage d'assurer l'intégrité de l'image. Une faible confidentialité est obtenue par l'aspect stéganographique du tatouage. Aucune authentification de l'orignie du message, au sens strict, n'est assurée dans la chaîne actuelle du tatouage cependant, des informations sur le patient et l'origine de l'image sont contenus dans l'en-tête DICOM embarquée dans l'image. L'introduction de la cryptographie dans la chaîne actuelle doit apporter ces deux dernières fonctions. L'authentification par un algorithme à clé publique et la confidentialité par une combinaison clé publique clé secrète.

4.1 Données à embarquer

Nous pouvons distinguer 4 types de données à embarquer dans l'image :

- Le résumé de l'image qui doit permettre le contrôle de l'intégrité (prévu dans la chaîne de tatouage).
- L'en-tête DICOM qui contient en plus des informations patients des informations complémentaires à la prise de vue telles que définies par ce standard (T_1 ou T_2, le temps de relaxationen IRM ...)
- Les traitements subis par l'image avant tatouage : pour une meilleure lecture du diagnostic.
- Le dossier médical initial : données annexes entrées par le médecin lors du compte rendu.
- Le dossier complémentaire : données ajoutées en sus du dossier initial pour le suivi du patient (évolution du diagnostic...)

	Patient	Corps médical	Corps médical autorisé	Machine
Résumé	R	R	R	RWA
En-tête DICOM		R	R	RWA
Traitement de l'image		R	R	RWA
Dossier Initial		R	R	RWA
Dossier complémentaire		R	RW	RW

TAB. 4.1 – *Résumé des droits d'accès*

4.2 Acteurs de la communication

Dans le cadre de la transmission d'image, nous pouvons considérer qu'il y a 4 acteurs intervenant dans la communication :

- la machine qui génère l'image,
- le patient,
- le corps médical,
- le corps médical "autorisé" à ajouter des données à l'image.

4.3 Définition des droits d'accès

Le tableau 4.1 suivant résume les différents droits d'accés de chaque acteur sur chaque section (R : droit de lecture, W : Droit d'écriture , A : authentification) :

4.4 Insertion de la cryptologie

Nous préconisons l'utilisation d'un schéma à clé publique du fait que nous devons permettre à certaine personne la lecture sans leur autoriser l'écriture et que nous devons permettre l'authentification à tous. Nous pouvons dès maintenant regrouper le cas du dossier initial, des traitements subis avant tatouage et de l'en-tête DICOM, les droits d'accès de ces groupes sont identiques.

Nous pouvons donc distinguer alors 3 lots de clefs :

- CA_1 et CA_2 pour le résumé,
- CB_1 et CB_2 pour le dossier initial, l'en-tête DICOM et le processus de traitement de l'image,
- CC_1 et CC_2 pour le dossier complémentaire.

Chaque couple correspond aux 2 clés générées par un schéma à clé publique. Par contre seule la clé CA_1 est publique : CA_2 est conservée par la machine qui

seule peut accéder en écriture à cette partie. Tout le monde peut lire le résumé donc vérifier l'intégrité de l'image car la clé CA_1 est publique.

De même CB_2 est conservée par la machine mais CB_1 est seulement donnée au corps médical, restreignant ainsi l'accés en lecture mais garantissant l'authentification par la machine (CB_2 est secrète).

CC_2 est uniquement donnée au corps médical autorisé et CC_1 est donnée à tous le corps médical ce qui permet de la même façon de gérer les droits d'accès de manière cryptographique.

Des aménagements sont possibles avec un sytème hybride clé secrète, clé privée pour réduire le temps de calcul comme pour le GPG [10] : le mode assymétrique servant de canal sûr à la transmission d'une clé secrète.

Chapitre 5

Améliorations des algorithmes déjà existants

Plusieurs algorithmes ont été testés au laboratoire. Pour les algorithmes Prédiction-Expansion et Différence-Expansion la compression de l'image de localisation a été simplement réalisée par les méthodes RLE, Huffman et LZW. Les méthodes utilisées précédement donnent des résultats relativement médiocres par rapport à des algorithmes avancés, comme JBIG2. De plus il reste à finaliser l'implémentation de l'algorithme prédiction-expansion developpé par le laboratoire et à programmer l'algorithme de Fridrich.

5.1 Problèmes techniques

5.1.1 Création de DLL en C++ pour Matlab

Le mécanisme des mex-files permet de programmer en C directement dans Matlab et de compiler grâce au compilateur *LCC*. Malheuresement, le débugage est extrêmement difficile car si la dll bloque, c'est l'ensemble de Matlab qui bloque. Nous devons donc introduire de nombreux tests dans le code et imprimer un message à la moindre suspicion d'erreur.
Malheureusement, *LCC* ne peut compiler que du C et l'environement de programmation de Matlab est assez pauvre. Notamment nous ne pouvons utiliser *make* pour gérer les sources, ni *gdb* pour deboguer. Pour pouvoir programmer en C++, nous devons utiliser un autre compilateur. Pour ce faire, j'ai proposé et installé le programme *gnumex* [11] qui intègre *gcc* (GNU Compilers Collection [12]) à Matlab, ainsi que l'environement de programmation *mingw* qui fournit des outils indispensables comme make et gdb (débogage).

5.1.2 Bibliothèques cryptographiques

De nombreuses bibliothèques cryptographiques sont disponibles via l'Internet. Malheureusement la plupart des programmes pour Microsoft Windows® sont en sources closes ce qui n'est pas pertinent pour l'application médicale ; en effet la sécurité d'un système cryptographique ne peut être basée uniquement sur le secret de son algorithme car d'une part le secret peut être brisé et, d'autre part, l'algorithme ne peut être testé et adapté à l'application par la communauté scientifique. De plus, les licences sont strictes quand à l'utilisation des logiciels et nécessitent un paiement dès que l'on sort du contexte de recherche. Deux bibliothèques restent donc utilisables, *Botan* [13] (licence BSD) et *libgcrypt* [14] (license LGPL). *Botan* est disponible pour Windows® mais n'est plus maintenue. *libgcrypt* est maintenant considérée comme un standard, cette bibliothèque fait partie du projet GNU [15] ce qui lui assure une grande stabilité et périnité; néanmoins l'intégration dans Windows n'est pas encore terminée (générateur d'aléas non compatibles). Dans un premier temps, j'utiliserai *Botan* pour tester les algorithmes. *libgcrypt* sera utilisée pour finaliser l'application.

Fonctions de compression et fonctions de hachage

Plusieurs fonctions de compressions et de hachage ont été developpées grâce à la bibliothèque Botan :

- $dlllzw$ compression LZW,
- $dllrle$ compression RLE,
- $dllhuffman$ compression Huffman,
- $dllari$ compression arithmétique,
- $dllsha256$ fonction de hachage.

Dans le futur, des fonctions de cryptographies viendront se rajouter ($dllrsa$ $dllaes$).

5.2 Méthodes de compression réversible du flot de bits

Il existe de nombreuses méthodes de compression réversibles de données basées sur le codage statistique (Huffman, Arithmétique, LZW), sur le codage spatiale(RLE,...). Nous étudierons deux nouvelles méthodes de compression : la BWT basée sur un tri lexicographique et JBIG (1 et 2) basée sur un codage vectoriel plus adaptée au codage d'images binaires telles que les images de localisation.

5.2.1 La transformation de Burrow-Wheeler (BWT)

La transformation Burrow-Wheeler (BWT) est une transformation de l'espace des données vers un autre espace plus propice à la compression de données : cette transformée a pour but d'augmenter la corrélation spatiale dans le message à traiter et ainsi d'améliorer la compression par des algorithme de codage du type RLE ou LZW.

Transformée directe

Algorithme

1. Création d'un tableau contenant toutes les rotations du message.
2. Trie du tableau par ordre lexicographique.
3. La transformée du message correspond à la dernière colonne du tableau ainsi qu'à la position dans le tableau trié du message de départ.

Exemple L'exemple se fera avec le message 'bananas'.

Étape 1	Étape 2	Étape 3
bananas	ananasb	ananas**b**
ananasb	anasban	anasba**n**
nanasba	asbanan	asbana**n**
anasban	bananas	**bananas**
nasbana	nanasba	nanasb**a**
asbanan	nasbana	nasban**a**
sbanana	sbanana	sbanan**a**

La transformée du message *bananas* est donc *bnnsaaa*,4.

Transformée inverse

Algorithme

1. Une colonne est remplie avec le message à décoder.
2. Création d'un tableau contenant en première colonne, la colonne de la première étape triée et en dernière colonne, la colonne de la première étape non triée.
3. On ajoute à côté de chaque colonne, une colonne qui contient le nombre de fois que la lettre correspondante est apparue dans la dite colonne.
4. La position du message est utilisée comme point de départ de l'algorithme de reconstruction. Nous avons alors la première et la dernière lettre du message. Une lettre (colonne impaire) et un chiffre (colonne paire) forment une clé.

L'algorithme de reconstruction consiste à prendre la deuxième clé de la ligne d'étude puis à selectionner la ligne qui contient cette clé en première colonne.

Exemple Nous devons décoder le message *bnnsaaa*,4

Étape 1

b
n
n
s
a
a
a

Étape 2

a	b
a	n
a	n
b	s
n	a
n	a
s	a

Étape 3

a	1	b	1
a	2	n	1
a	3	n	2
b	1	s	1
n	1	a	1
n	2	a	2
s	1	a	3

Étape 4 : nous commençons par étudier la ligne 4, nous savons dorénavant que notre message commence par 'b' et se finit par 's'. La deuxième clé de la ligne 4 est (s, 1) nous recherchons cette clé dans la première colonne; elle se situe en ligne 7, nous obtenons alors une nouvelle clé (a, 3). Le processus est répété jusqu'à obtention du message de départ.

$$(s,1) \rightarrow (a,3) \rightarrow (n,2) \rightarrow (a,2) \rightarrow (n,1) \rightarrow (a,1) \rightarrow (b,1)$$

Partant de la dernière clé, nous retrouvons bien le message : *bananas*.

Résulats

L'algorithme de la BWT demandant le tri et la gestion en mémoire d'un grand tableau, la puissance de Matlab® n'est pas suffisante (3 heures pour une image 8 bits 512x512). J'ai donc décidé de programmer l'algorithme en C et de l'utiliser à travers une dll dans Matlab. La programmation de l'algorithme en C a permis de passer d'un temps de calcul de 3 heures à 2 minutes. L'énorme différence vient du fait que je contrôle tout le processus, notament de la gestion de la mémmoire : pour une image 512x512, Matlab doit alouer 512x512x512x512 pour faire le tri; en C, grâce à un décalage judicieux, je n'ai besoin que d'une liste 512x512. En effet, je ne dois pas caculer explicitement toutes les rotations de l'images, mais seulement un tableau contenant la position de la rotation dans le tableau trié.
Employer cette méthode sur des images à 2 ou 3 niveaux de gris ne semble pas trés concluant : le faible nombre de niveaux de gris ne permet pas de séparer correctement les données lors de la phase de tri.
J'ai contacté deux étudiants anglais qui travaillent sur une extension en 2D de cette transformée, leur travail n'a pas encore abouti. Une compression vectorielle type

JBIG est plus intéressante pour des plans de bits. La BWT est certainement à retenir pour les autres données à embarquer telles que les données diagnostiques.

5.2.2 Compression JBIG

La compression JBIG est une quantification vectorielle developpée à la base pour le fac-similé. Les parties élémentaires de l'image sont transformées puis comparées à un dictionaire. L'image compressée contient les indexs utilisés dans le dictionnaire ainsi que les transformation. La version 1 et 2 de cet algorithme diffère essentiellement par le dictionnaire utilisé et par une optimisation sur les transformations.

L'adoption de *gcc* m'a permis d'intégrer plus facilement des bibliothèques déjà existantes dans Matlab.
Une première tentative d'intégration de la bibliothèque JBIG en C issue de Linux avait échoué. J'ai donc décidé d'introduire une couche d'abstraction en C++ (compatible avec *gcc*) qui m'a permis de gérer beaucoup plus facilement les objets en mémoire et ainsi éviter les fuites de mémoire (très grande dimension des tableaux) et les erreurs de segmentation qui rendait le programme inutilisable dans sa première version.

La compression JBIG est directement utilisable dans Matlab à travers les appels *dlljbig*() pour la compression et *dllijbig*() pour la décompression.

Tian utilise JBIG2 dans ces articles; malheuresement, seul JBIG est disponible gratuitement et je n'ai trouvé aucun programme utilisable depuis Matlab qui applique la compression JBIG2. Néanmoins, les résultats obtenus sont nettement plus probant qu'avec une simple compression RLE puis LZW-HUFFMAN (gain de 20%).

Chapitre 6

Mesure de la dissimilarité

Bien que travaillant sur des méthodes réversibles, il est intéressant de quantifier la dégradation de l'image tatouée qui pourra servir de support visuel au diagnostic réalisé sur l'image original. De plus le laboratoire envisage d'étudier également une chaîne de tatouage irréversible pour satisfaire le critère de robustesse face à des modifications de l'image (zoom ou fenêtrage pour un meilleur diagnostiqc) Afin de pouvoir quantifier l'erreur introduite par la chaîne de tatouage, nous devons retenir un critère de dissimilarité entre l'image de départ et l'image tatouée. La plupart des publications utilise le PSNR comme critère mais il est peu judicieux en imagerie médical, car trop global. La perception du médecin lors d'un diagnostiqc n'étant pas la même qu'un individu non initié devant un image classique. De plus, ce critère ne met pas en avant les modifications géométriques engendrées par le codage (très visibles lors d'une compression JPEG par exemple), et pouvant être importantes dans une chaîne de tatouage si les contours supportent une grande partie de la marque.

6.1 L'EQM : l'écart quadratique moyen

C'est le critère de dissimilarité le plus connu, il est généralement associé au PSNR :

$$eqm = \frac{\sum_{i=1}^{N} x_i^2 - x'^2_i}{N} \tag{6.1}$$

6.2 Le PSNR : le rapport signal bruit

Le *psnr* n'est qu'une mise à échelle logarithmique de l'EQM :

$$psnr = 10log\left(\frac{max^2}{eqm}\right) \tag{6.2}$$

L'*eqm* et le *psnr* ne tiennent comptent que de la différence de niveau de gris pixel par pixel.

6.3 La distance de Baddeley

Pour pallier à cet inconvénient, nous avons introduit la distance de Baddeley qui prend en compte non seulement les variations de niveaux de gris mais aussi les déformations géométriques [16]. Cette mesure est basée sur une extension de l'image 2D en niveaux de gris, en une image 3D binarisée. On mesure ensuite la distance entre la surface représentée par l'image tatouée et la surface de référence représentée par l'image originale (Figure 6.1).

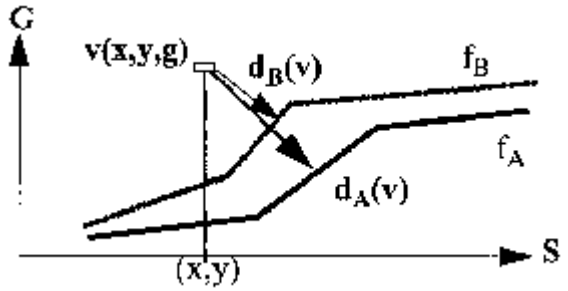

FIG. 6.1 – *Distance du pixel à la surface de référence*

6.3.1 Transformation d'une image 2D (A) en niveaux de gris en une image 3D binarisée (B)

Une troisième dimension représentant l'échelle de niveau de gris est ajoutée, l'image B est créée par la règle suivante :

$$B(i,j,z) = 1 \; si \; A(i,j) = z \; sinon \; B(i,j,z) = 0$$

6.3.2 Transformation distance

Distance de Chanfrein

On appelle une distance sur le plan discret $\mathbb{Z}^2$, toute fonction d définie sur $\mathbb{Z}^2 \times \mathbb{Z}^2$ à valeurs réelles non négatives satisfaisants les trois axiomes suivants :

- $d(p,p) = 0$ et pour $p \neq q$ $d(p,q) > 0$
- $d(p,q) = d(q,p)$
- $d(p,r) < d(p,q) + d(q,r)$

Une approche classique en géométrie discrète consiste à mettre une pondération sur l'adjacence d'un pixel à ses voisins, valant la distance entre ces 2 pixels. Ainsi, la structure donnée à $\mathbb{Z}^2$ est celle d'un graphe pondéré, dont les arêtes sont celles de la 8-adjacence. A chaque arête entre 2 pixels voisins p et q on donne un poids $P(p,q)$. La distance entre p et q correspond au poids minimum des 8-chemins pour aller de p à q. Pour une 8-adjacence, un tableau 3x3 est nécessaire à la définition des poids. Ce tableau est appelé *tableau de Chanfrein* et la distance corespondante *distance de Chanfrein.*

b	a	b
a	0	a
b	a	b

TAB. 6.1 – *Tableau de Chanfrein*

La distance euclidienne est obtenue avec $a = 1$ et $b = \sqrt{2}$. Si on considère un 8-chemin pour aller du point M au point N, le poids total du chemin correspond à la somme des poids élémentaires.

Conditions de Montanari

Pour respecter les axiomes de la distance, Montanarie a défini deux conditions sur a et b :

- $a > 0$
- $a \leq b \leq 2a$

Dans notre cas, nous nous efforcerons de nous raprocher de la distance euclidienne tout en restant dans l'ensemble des entiers naturels en prenant $a = 3$ et $b = 4$. Le masque de Chanfrein peut-être étendu à une matrice 5x5 appellée masque de Borgefors.

Transformation distance 2D

Cette transformation associe à chaque pixel une valeur correspondant à la distance de ce pixel avec un objet de référence. L'objet de référence sera défini par la valeur 1 dans l'image de départ et le fond 0. L'algorithme utilisé permet en 2 parcours de l'image d'obtenir le résultat. La matrice de Chanfrein est réduite à la matrice de précédence du parcours défini [16].

La valeur de chaque pixel P est obtenue grâce à la matrice (TAB. 6.2) de précédente M par la formule suivante :

$$P = min(P + M(i,j))\ i \in [1\ 3]\ j \in [1\ 3]$$

b	a	b
a	0	∞
∞	∞	∞

TAB. 6.2 – *Matrice de précédence du premier parcours*

Un deuxième parcours est effectué dans le sens inverse avec la matrice M retournée à 180°.

Extension à la 3D

Le principe est identique, la valeur de chaque voxel corresponds à la distance du pixel à la surface de référence (surface 3D). La matrice de Chanfrein est elle-aussi étendue à la troisième dimension (TAB. 6.3) :

c	b	c
b	a	b
c	b	c

b	a	b
a	0	a
b	a	b

c	b	c
b	a	b
c	b	c

TAB. 6.3 – *Tableau de Chanfrein étendu à la troisième dimension*

D'où la matrice de précédence suivante (TAB. 6.4) :

c	b	c
b	a	b
c	b	c

b	a	b
a	0	∞
∞	∞	∞

∞	∞	∞
∞	∞	∞
∞	∞	∞

TAB. 6.4 – *Matrice de précédence étendue à la troisième dimension*

6.3.3 Exemple de calcul de la transformée distance

Nous allons prendre un exemple simple pour illustrer cette transformée, l'image de départ est la table 6.5.

Initialisation

La première étape est de transformer l'image de départ en une matrice qui nous servira d'état initial pour le premier parcours : la valeur 0 de l'image correspond à une distance infinie, la valeur 1 de l'image (sur la surface de référence) correspond à une distance nulle (table 6.6) .

0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

TAB. 6.5 – *Image de départ*

∞	∞	∞	∞	∞	∞
∞	∞	∞	0	0	0
∞	∞	0	∞	∞	∞
∞	0	∞	∞	∞	∞
0	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞

TAB. 6.6 – *Matrice initiale*

Premier parcours

Nous allons parcourir l'image avec la matrice de précédence (table 6.2) dans l'orde décrit à la figure 6.2.

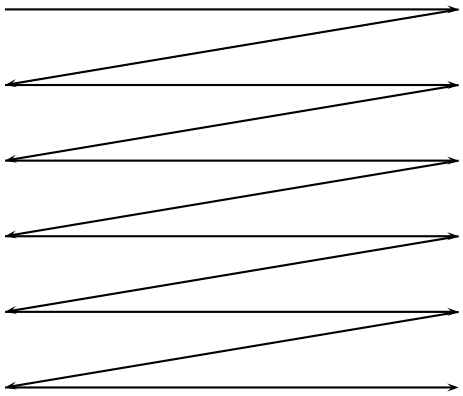

FIG. 6.2 – *Parcours de l'image*

Nous prenons le premier pixel (1,1) et son voisinage, nous les aditionnons avec la matrice de précédence puis gardons la valeur minimum, dans ce cas la valeur minimum reste ∞(table 6.7).

Le premier pixel à changer de valeur est le pixel (4,3) comme le montre la table 6.8.

Voisinage du pixel			Martrice de précédence			Somme			Minimum
	∞	∞		0	∞		∞	∞	∞
	∞	∞		∞	∞		∞	∞	

TAB. 6.7 – *Calcul du Pixel (1,1)*

Voisinage du pixel				Martrice de précédence				Somme			Minimum
	∞	0	0		4	3	4		∞	3	4	
	0	∞	∞		3	0	∞		3	∞	∞	3
	∞	∞	∞		∞	∞	∞		∞	∞	∞	

TAB. 6.8 – *Calcul du Pixel (4,3)*

L'algorithme est le même pour chaque pixel, le résultat du premier parcours se trouve en Table 6.9.

∞	∞	∞	∞	∞	∞
∞	∞	∞	0	0	0
∞	∞	0	3	3	3
∞	0	3	4	6	6
0	3	4	7	8	9
3	4	7	8	11	12

TAB. 6.9 – *Matrice obtenue à la fin du premier parcours*

Second parcours

La démarche est identique pour le second parcours, seul change le sens du parcours (figure 6.3) et la matrice de précédence (table 6.10).

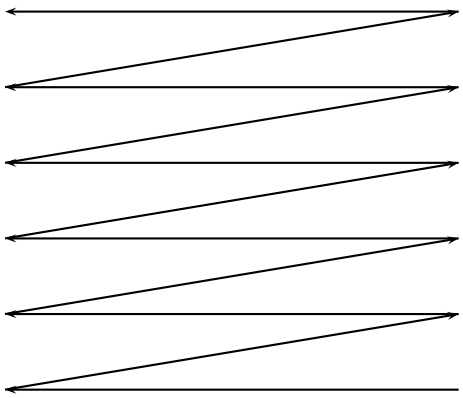

FIG. 6.3 – *Parcours de l'image*

∞	∞	∞
∞	0	3
4	3	4

TAB. 6.10 – *Matrice de précédence retournée à 180°*

Le résultat du second parcours donne la matrice transformée distance (table 6.11).

8	7	4	3	3	3
7	4	3	0	0	0
4	3	0	3	3	3
3	0	3	4	6	6
0	3	4	7	8	9
3	4	7	8	11	12

TAB. 6.11 – *Matrice transformée distance*

6.3.4 Calcul de la distance de Baddeley

La transfomée distance est calculée pour les deux images à comparer. On obtient l'espace S pour l'image source et l'espace I pour l'image à tester. La distance D est calculée comme suit :

$$D = \frac{\left[\sum_{v \in S}(S(v)^d - I(v)^d)\right]^{\frac{1}{d}}}{card(S)}$$

On prend généralement $d = 2$ pour être conforme à la distance Euclidienne. Cette distance prend en compte à la fois les déformations géométriques et les variations de niveaux de gris.

Ici, le voxel v est de forme cubique. En modifiant sa forme pour obtenir un parallélépipède, on peut jouer sur la dédection des dissemblances et défavoriser les déformations géométriques au profit des variations de niveaux de gris ou inversement.

Chapitre 7

Phase de test

7.1 Le panel d'images tests

La difficulté en imagerie médicale est que nous ne disposons pas de données standards comme les clichés de Lenna ou de Barbara. De plus chaque modalité réagit différemment au traitement, ce qui accroît considérablement le nombre d'images nécéssaires à la constitution du panel représentatif. Nous avons retenu 133 images réparties comme suit:

- Résonance magnétique: 38 clichés.
- Médecine nucléaire: 25 clichés dont 22 PETs.
- Scanner X: 59 clichés.
- Non médicales: 11 clichés.

7.2 Quels sont les besoins ?

Le protocole que nous devons mettre en place doit permettre de répondre à ces deux questions, pour chaque chaîne de tatouage :

- Quelle est la taille de données qu'il est possible d'embarquer?
- A quantité embarquée constante, quelle est la qualité des images marquées?

Nous ne pouvons savoir exactement a priori ni la qualité ni la quantité d'informations que nous pourrons embarquer. Voilà pourquoi, nous chercherons à mesurer différents paramètres annexes qui nous donnerons des indices pour répondre aux questions posées :

- Le maximum d'informations embarquées.
- Les dégradations engendrées.
- L'influence du seuil (différent pour chaque algorithme) sur la quantité embarquée et sur la qualité des images.

D'autres part, la phase de test devra permettre une comparaison entre les différentes méthodes de mesure des dégradations et dissimilarités entre les images. La comparaison des résultats pour les différentes méthodes de tatouages sur un seul cliché montre la difficulté de trouver une méthode d'évaluation unique.

7.3 Le protocole de test

7.3.1 Méthodes de tatouages

Pour chaque image, nous allons appliquer les méthodes de marquages de Différences-Expansions et de Prédictions-Expansions suivant sept niveaux de seuils (0%, 10%, 25%, 50%, 75%, 90%, 100%) avec trois algorithmes de compressions différents (RLE non modifié utilisé lors du précedent stage, RLE amélioré et JBIG). Par image, il y a donc 2x7x3=42 tests effectués.
Pour chaque test, nous notons la quantité de bits libérés, la capacité par pixel libérée (i.e. la quantité divisée par la taille de l'image), le PSNR et l'écart quadratique moyen.
La distance de Baddeley demandant un nombre important de calculs, elle sera réservée à un nombre limité de séquence de test.

7.3.2 Algorithmes de compression et cryptages

Les algorithmes de compressions et de cryptages seront testés sur les fichiers de rapport 4Dwrite issus de la base de données 4D du service de médecine nucléaire de l'hôpital Larrey.

Chapitre 8

Résultats et interprétations

8.1 Méthodes de tatouage

8.1.1 Algorithme de Cellik

Cette méthode de tatouage est uniquement disponible, pour l'instant, sous forme de programme binaire (sans source) et n'est capable de tatouer que des images au format 8 bits. Nous ne pouvons donc l'utiliser sur l'ensemble du panel d'images (uniquement les images non-médicales). Nous ne l'étudierons donc pas pour l'instant.

8.1.2 Algorithme de Fridrich

Résultats

Taille des images	Nombre de bits libérés	Capacité
128x128	-400	-0.024
256x256	240	0.0037
512x512	400	0.0015

TAB. 8.1 – *Tatouage par la méthode de Fridrich*

Observations et interprétations

Cette méthode ne libère pas assez de place pour insérer la marque sur des images médicales. Elle ne pourra donc pas être utilisée telle quelle. Les fonctions de permutations et de discriminations ne semblent pas adaptées à l'imagerie médiacle: cet algorithme constitue un brevet qui a été déposé par Kodak aux États-Unis, les fonctions publiées dans les articles de Fridrich [5] ne sont donc pas optimisées.

8.1.3 Algorithme de Tian

Rappels des travaux précédents

Lorsque le seuil augmente, la capacité embarquée augmente jusqu'à saturation (figure 8.1). En effet, plus le seuil augmente plus le nombre de pixels marqués par extension augmente et donc le flux de bits conservé après le marquage par modification du bit de poids faible diminue, la capacité embarquée augmente alors. La saturation est due à la limite de la compression de la carte de localisation par l'algorithme de compression.

Néanmoins, nous pouvons noter que plus le seuil augmente plus la dégradation est importante (Le PSNR diminue) (figure 8.2) La méthode de marquage par expansion est plus dégradante que la méthode de marquage par modification du bit de poid faible. En conséquence, plus la capacité augmente plus l'image est dégradée (Figure 8.3).

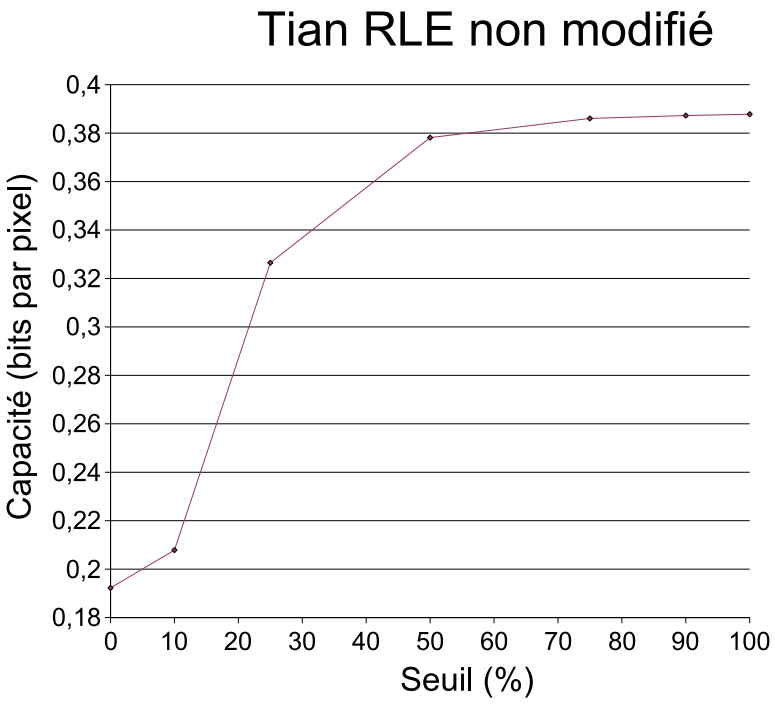

FIG. 8.1 – *Capacité libérée en fonction du seuil*

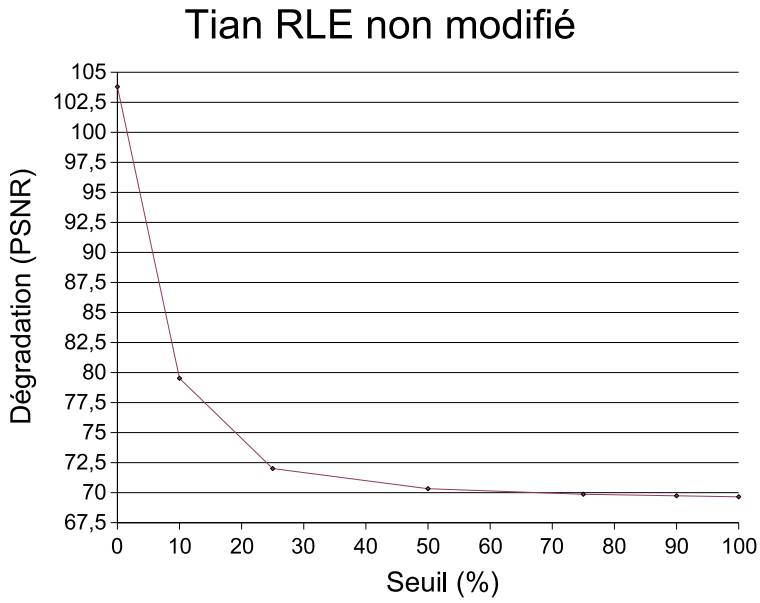

FIG. 8.2 – *Dégradation engendrée en fonction du seuil*

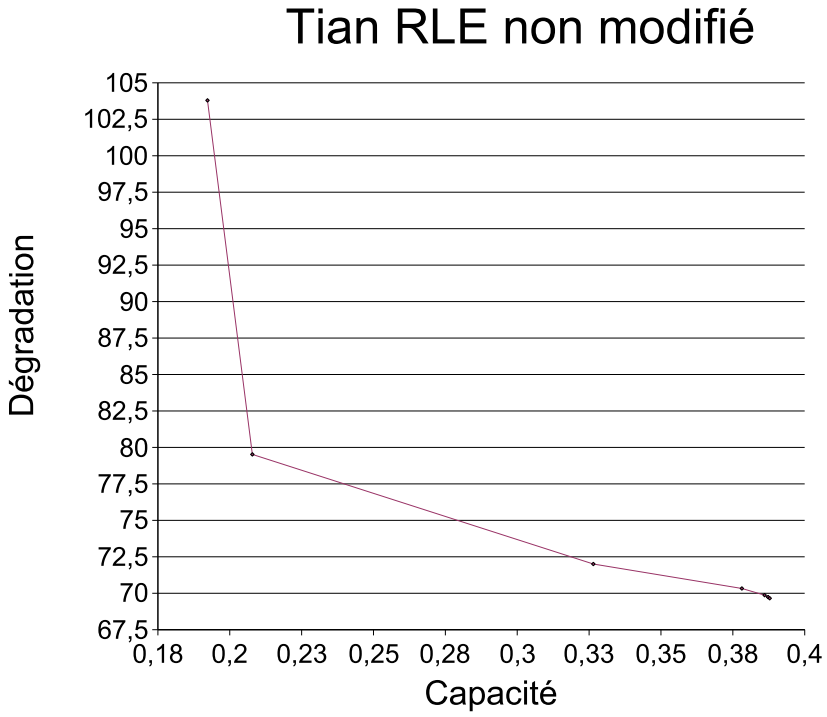

FIG. 8.3 – *Dégradation engendrée en fonction de la capacité*

Améliorations par modification des algorithmes de compression

L'étude précédente utilisait une méthode de compression spatiale RLE (Run Length Encoding) simple. Nous allons comparer cette méthode de compression avec deux nouvelles méthodes RLE modifié (utilisation de compression statitisque et adaptation aux données à embarquer) et JBIG.

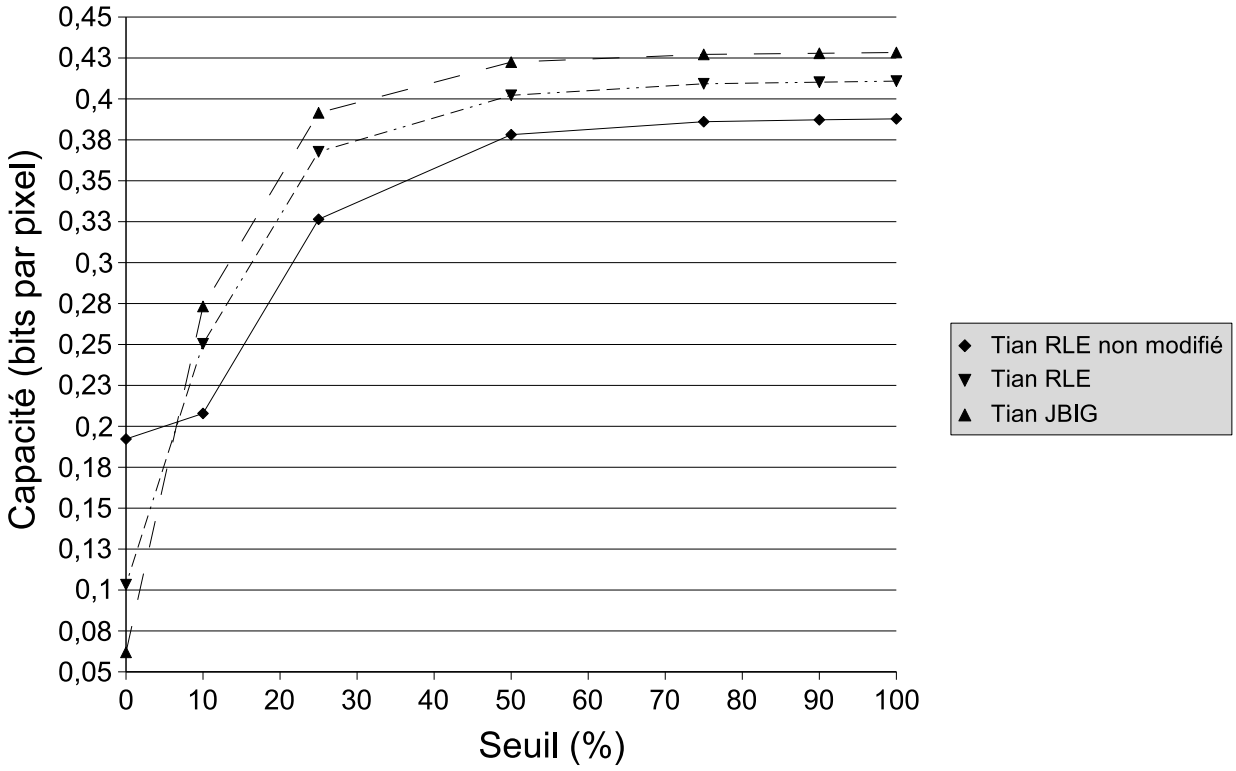

FIG. 8.4 – *Comparaison de la capacité libérée pour différents algorithmes de compression*

Étude sur la capacité Les courbes obtenues (figure 8.4) montrent que JBIG apporte une meilleure capacité que RLE qui est lui même supérieur pour cette même capacité au RLE non modifié. Ils permettent de compresser d'avantage la carte et le flux de bits et, donc,pour un seuil donné, d'augmenter la place libérée.

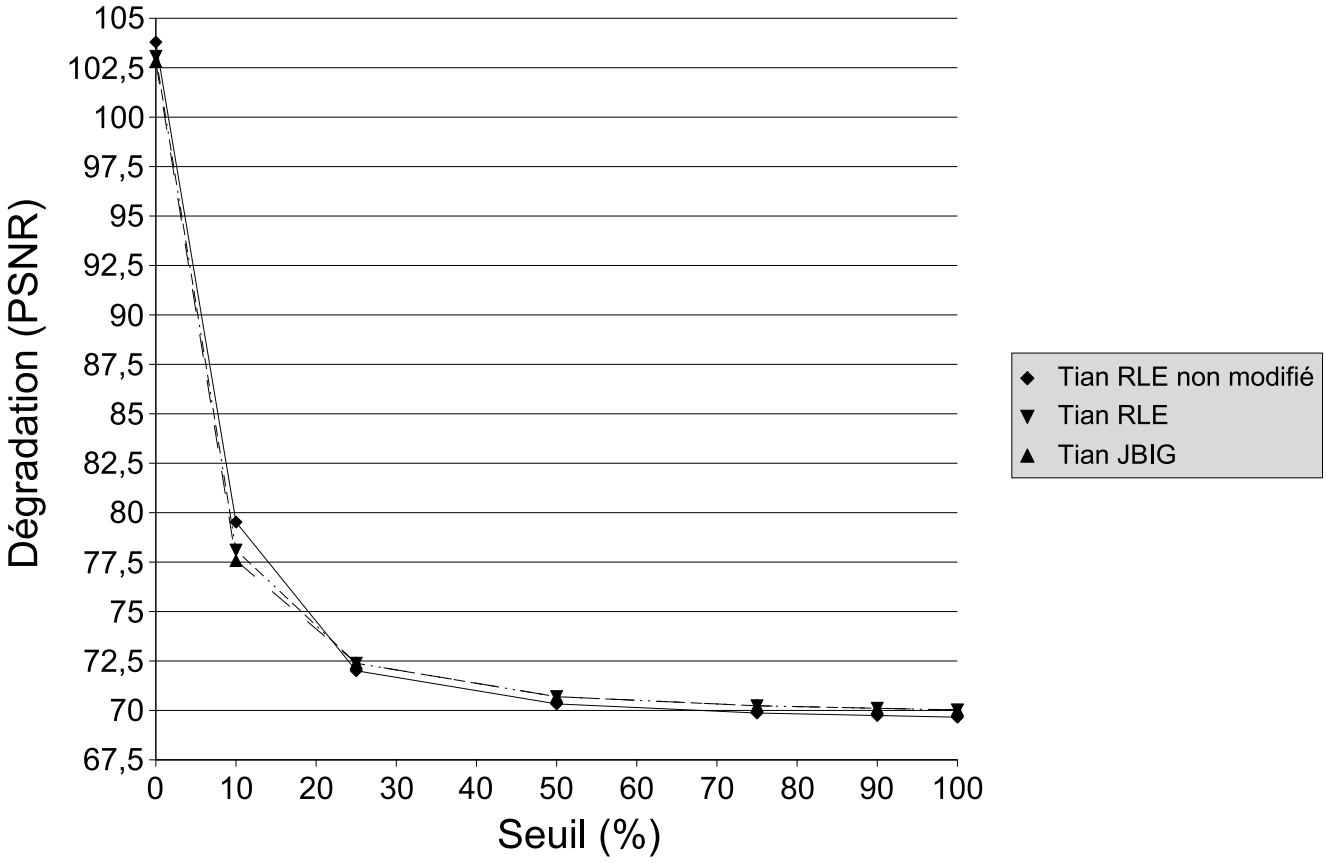

FIG. 8.5 – *Comparaison de la dégradation engendrée pour différents algorithmes de compression*

Étude de la dégradation Les courbes de dégradations (figure 8.5) restent très proches pour les différents algorithme néanmoins JBIG semble être un peu moins destructif.

Conclusion JBIG devient donc la méthode la plus intéressante de compression de la carte de localisation pour le tatouage de Tian. Outre son amélioration de la capacité, elle apporte aussi une diminution de la dégradation. JBIG doit donc être considérée comme la méthode à adopter pour le marquage de Tian.

8.1.4 Algorithme Prédiction-Expansion

Comme l'algorithme de Prédiction-Expansion marque chaque pixel et non un groupe de 2 pixels comme le fait Tian, il devrait être plus performant. Avant de comparer les deux méthodes, nous allons étudier seul l'algorithme de Prédiction-Expansion pour savoir si son comportement est identique à l'algorithme de Tian.

Capacité en fonction du seuil

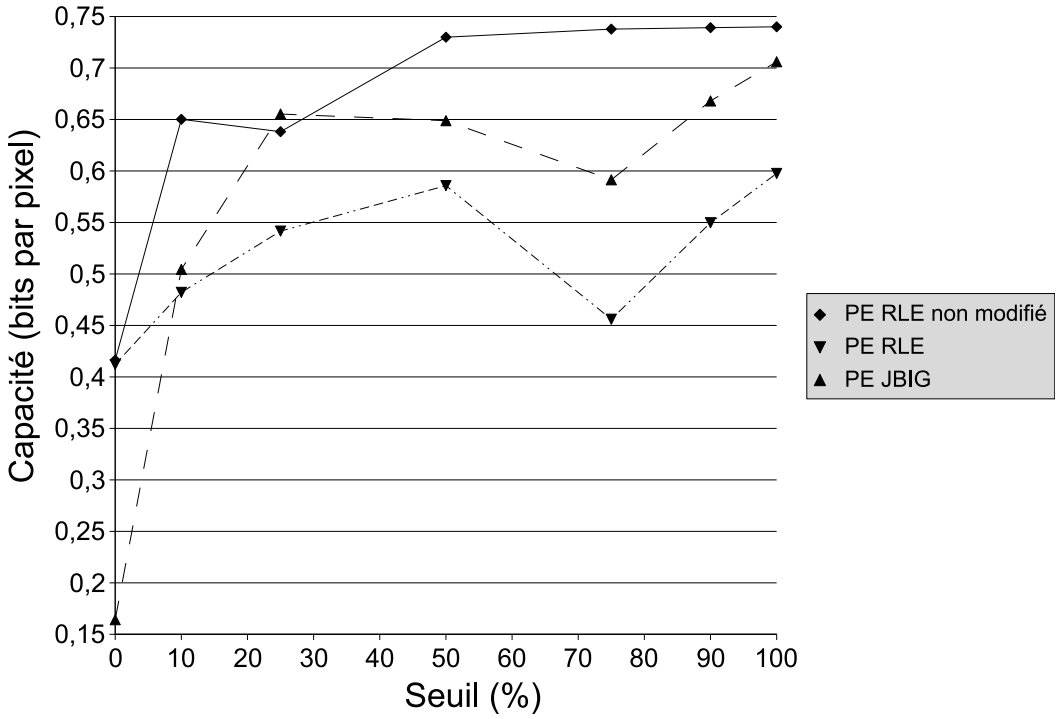

FIG. 8.6 – *Comparaison de la capacité libérée pour différents algorithmes de compression*

L'algorithme RLE non modifié semble plus performant suivi par JBIG et par RLE (figure 8.6). JBIG est censé améliorer la compression donc devrait être meilleur que RLE non modifié or l'expérience nous montre le contraire. JBIG et RLE peuvent être moins performants dans le cas de la compression de très grandes plages, provenant de grandes images, qui peuvent entrainer des dépassements de capacités.

Nous allons recommencer l'étude en nous limitant à des images plus petites que 512 par 512 pixels (figure 8.7). Cette fois-ci nous obtenons bien le meilleur résultat par JBIG puis RLE et enfin RLE non modifié.

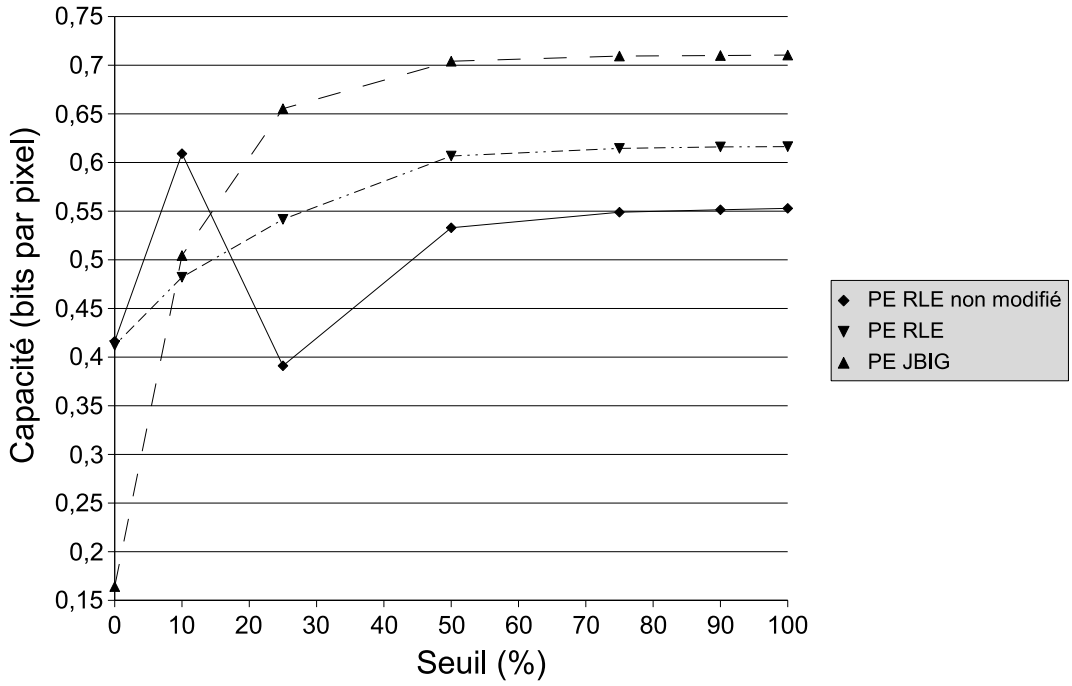

FIG. 8.7 – *Comparaison de la capacité libérée sur l'ensemble réduit*

Dégradation en fonction du seuil

Les méthodes JBIG et RLE se détachent très nettement de la méthode RLE non modifiée. En effet, les deux méthodes les moins dégradantes compressent les données à embarqer de façon plus bruitée. Le marquage de l'image en est moins destructif.

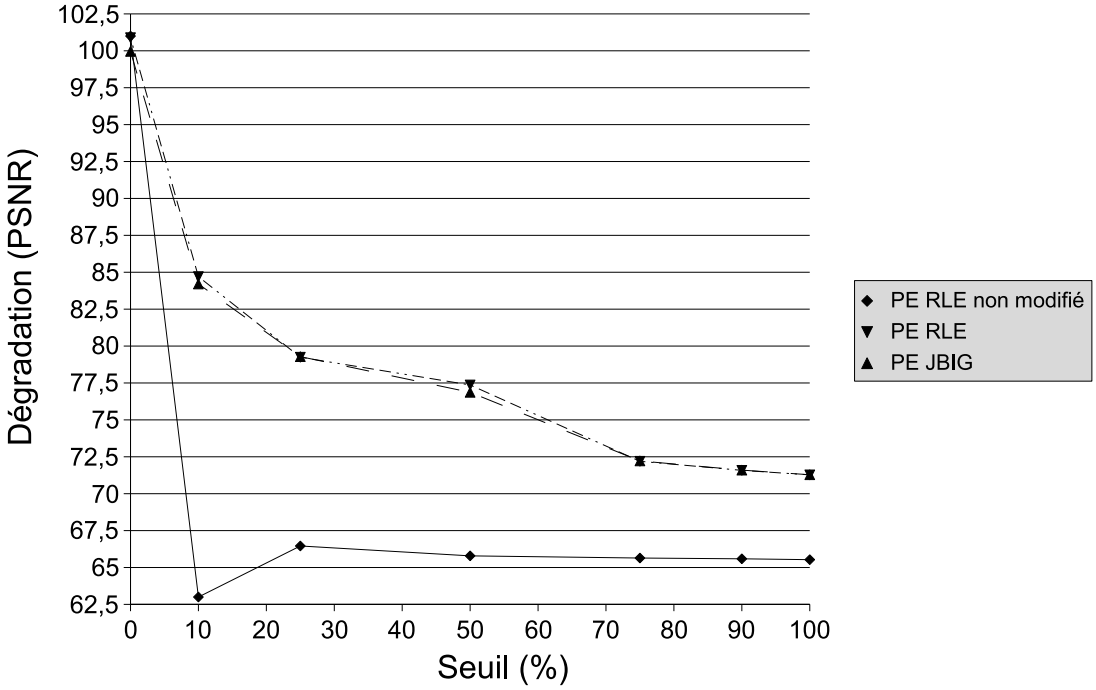

FIG. 8.8 – *Comparaison de la dégradation engendrée*

Conclusion

Sur des images de taille inférieure ou égale à 512x512, l'algorithme se comporte comme Tian. Il faudra développer de nouvelles méthodes de compression de données pour que cette méthode se comporte idéalement avec des images plus grandes.

8.1.5 Comparaison de Tian et Prédiction-Expansion

Capacité embarquée

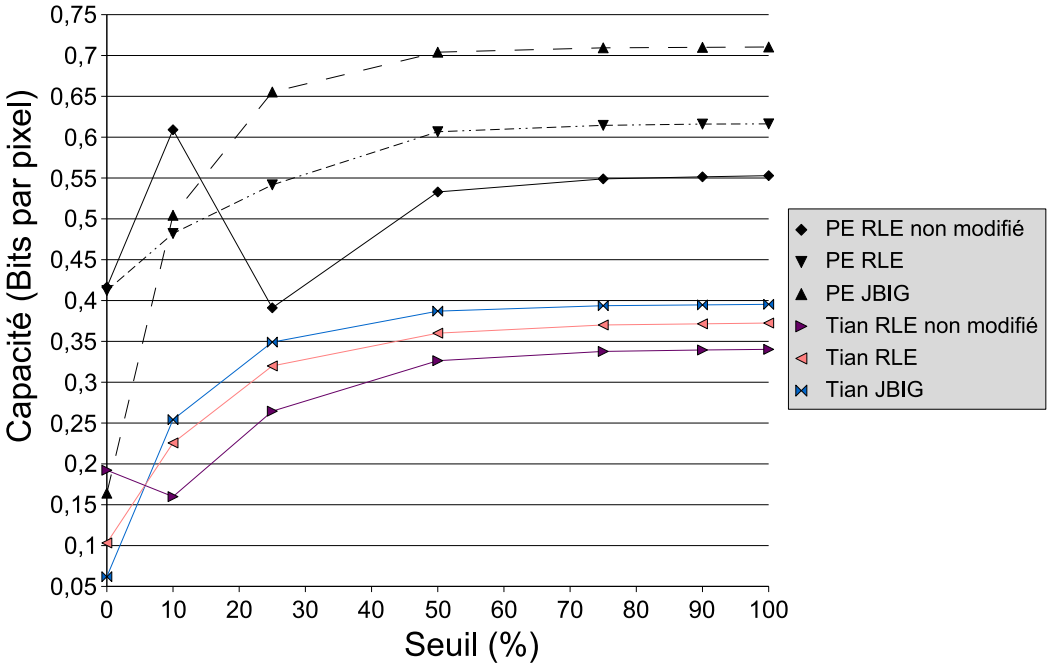

FIG. 8.9 – *Comparaison de la capacité libérée*

L'algorithme de Prédiction-Expension marquant l'image sur l'ensemble des pixels, et non sur la moitié des pixels comme Tian, permet une augmentation de la capacité. L'algorithme de Prédiction-Expension est donc le plus efficace.

Dégradation apportée à l'image

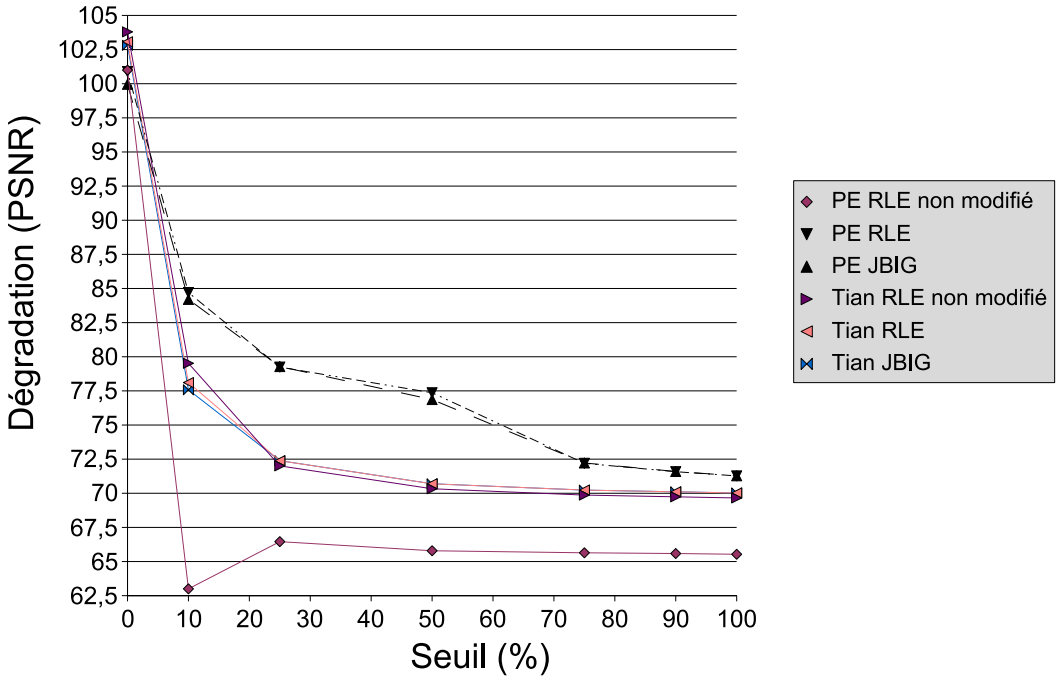

FIG. 8.10 – *Comparaison de la dégradation engendrée*

Algorithme de compression original Du fait même que Tian garde l'image à l'échelle 1:2 et que l'algorithme de prédiction-Expension modifie l'ensemble des pixels, Tian dégrade moins les images.

JBIG et RLE modifié JBIG et RLE modifié apportent du bruit dans la marque, l'algorihtme de Préduction-Expension surpasse alors Tian. En effet, plus la marque ressemble à du bruit, mieux elle est intégrée à l'image et les dégradations sont donc moins perceptibles. La méthode de Prédiction-Expansion avec l'algorithme de compression JBIG surpasse alors toutes les autres méthodes.

8.2 Mesure de dissimilarité

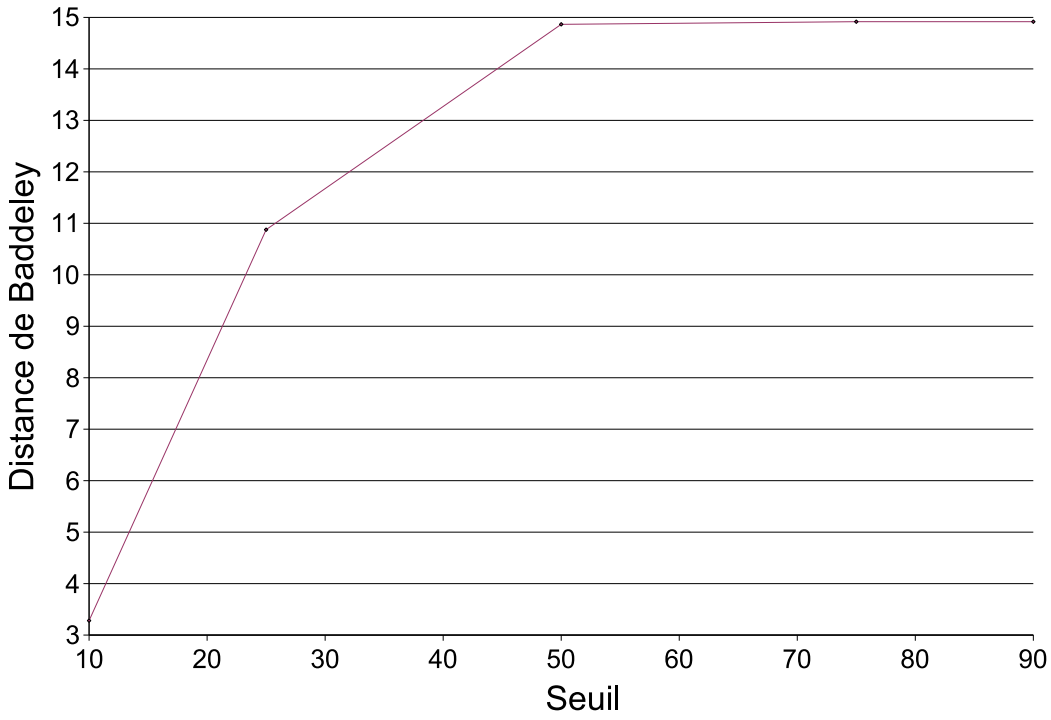

FIG. 8.11 – *Distance de Baddeley en fonction du seuil*

Les résultats obtenus par la distance de Baddeley sont similaires aux résultats obtenus par le PSNR: il existe une relation quasiment linéaire entre la distance et le rapport signal/bruit (figure 8.12). Le même effet de saturation apparaît à partir d'un seuil de 50% (figure 8.11).

Le calcul de la distance étant très long, car écrit en Matlab, nous conseillons d'utiliser le PSNR pour explorer un large éventail d'image puis Baddeley pour affiner les résultats.

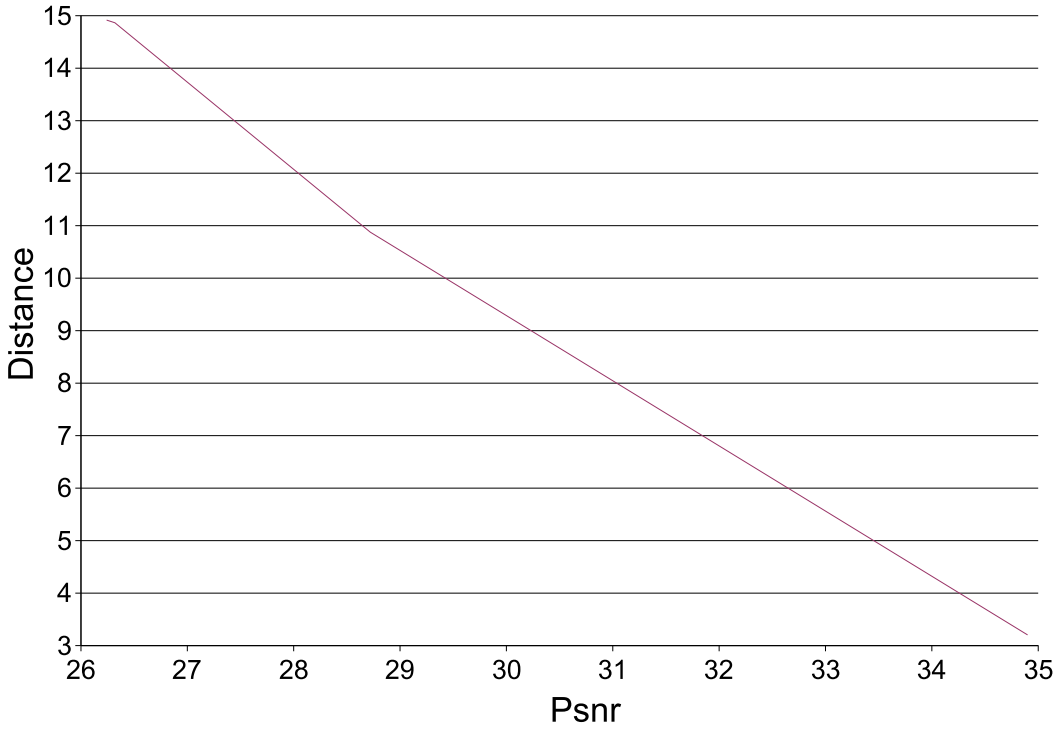

FIG. 8.12 – *Distance de Baddeley en fonction du rapport signal/bruit*

8.3 Méthode de cryptage et de compression

Taille du fichier original (Ko)	Taille du fichier crypté et compressé (Ko)	Taille du fichier crypté et compressé avec BWT (Ko)
5,67	3,56	3,02
5,92	3,60	3,05

TAB. 8.2 – *Résultats de la compression et du cryptage*

Les fichiers compressés affiche une taille inférieur à 55% des fichiers originaux. L'introduction de la Burrow Wheeler Transform, quand à elle, permet un gain de l'ordre de 18 % sur un fichier compressé uniquement par RLE puis Huffman. Il est important d'utiliser une bonne compression avant le cryptage car le fichier une fois crypté ressemblant fortement à du bruit, il est dificille de le compresser. Cette ressemblance au bruit à néamoins l'avantage de réduire la dégradation de l'image.

Chapitre 9

Perspectives

Il serait intéressant de réfléchir à une méthode de tatouage réversible et robuste à certaines attaques. Cette méthode pourrait reconstruire l'image seulement si l'image n'a pas été dégradée. Dans la cas contraire, la méthode ne pourrait pas reconstruire l'image mais pourrait en extraire des informations pertinentes comme le dossier médical.

La robustesse peut venir par exemple de la redondance des données embarquées ou de l'utilisation de codes de correction d'erreurs.

D'autres part, certaines modalités utilisent une représentation en trois dimensions des informations, une généralisation de l'algorithme de prédiction-expansion à ce nombre de dimension serait un moyen de gagner encore en qualité et en quantité de données embarquées.

Chapitre 10

Conclusion

Ce stage a conforté ma volonté de poursuivre mon parcours professionnel dans le domaine de la recherche. En effet, ces six derniers mois ont confirmé ce que j'avais déjà pressenti lors de mon stage technique. Je suis actuellement en recherche de financement pour une allocation de recherche en vue de la préparation d'une thèse.

L'apport personnel de ce stage a été très important, il m'a permis d'aborder les contraintes spécifiques du milieu médical qui demandent l'utilisation de nouvelles techniques de mesure de qualité et de dissimilarité. Il m'a permis d'acquérir une bonne connaissance technique du domaine de la cryptologie, domaine que j'avais déjà abordé lors de mon stage de première année mais dans une approche plus législative.

Enfin, j'ai été marqué par la bonne ambiance de travail et de convivialité au sein du laboratoire d'autant plus enrichissante que l'unité se trouve à l'interface de deux mondes : le millieu scientifique et le milieu hospitalier.

Chapitre 11

Références

Bibliographie

[1] G. Coatrieux, H. Maître, B. Sankur, Y. Rolland, and R. Collorec. Relevance of watermarking in medical imaging. In *Information Technology Applications in Biomedicine*, EMBS International Conference, pages 250–255. IEEE, November 2000.

[2] C. Cavaro-Ménard and S. Amiard. Reversible data embedding for integrity control and authentication of medical images. July 2004.

[3] J. Tian. Reversible data embedding and content authentication using difference expansion. *IEEE Transaction on Circuits and systems for Video Technology*, February 2003.

[4] J.Fridrich, M. Goljan, and R.Du. Invertible authentication. In *Security and Watermarking of Multimedia contents III*, volume 3971, pages 197–208. SPIE Photonics West, January 2001.

[5] J.Fridrich, M. Goljan, and R.Du. Lossless data embedding - new paradigm in digital watermarking. *Specials Issue on Emerging Applictions of Multimedia Data Hiding*, 2002(2):185–196, February 2002.

[6] Mehmet Utku Celik, Gaurav Sharma, Ahmet Murat Tekalp, and Eli Saber. Lossless generalized-lsb data embedding. http://citeseer.ist.psu.edu/586093.html.

[7] X. Wu. Lossless compression of continous-tone images via context selection, quantization, and modelling. *IEEE Transactions on Image Processing*, 6:656:664, May 1997.

[8] Bruce Schneier. Applied cryptography, second edition. *John Wiley and Sons*, 1998.

[9] http://www.pgp.com/.

[10] http://www.gnupg.org/.

[11] http://gnumex.sourceforge.net/.

[12] http://gcc.gnu.org/.

[13] http://botan.randombit.net/.

[14] http://www.gnu.org/directory/libgcrypt.html.

[15] http://www.gnu.org/.

[16] Yousra CHEHADEH. *Opérateurs locaux de distance en maillages rectangulaires et parallelepipediques: application à l'analyse d'image*. PhD thesis, Université de Savoie, 1997.

Table des figures

Liste des tableaux

Annexe A

Algorithmes

A.1 Méthode des LSB

La méthode des LSB (Low State Binary ou bits de poids faible) est une méthode simple et bien connue. Elle consiste à remplacer les bits de poids faibles pour y insérer de nouvelles données. Etant donné que l'on ne modifie qu'un seul bit, la capacité C en bit par pixel (bpp) est de 1. Cette méthode est invisible,peu robuste mais surtout totalement irréversible. En effet la suppression des bits de poids faibles de l'image originale entraîne certes une faible variation du niveau de gris (1/256 pour une image 8 bits, 1/65536 pour une image 16 bits) mais une perte d'information définitive. L'augmentation de la taille des données à insérer et donc de la capacité C (C= 2 ...bpp) est possible en changeant le nombre de bits de poids faibles modifiés (2 et plus). En contrepartie, la dégradation de l'image augmente et la méthode reste toujours irréversible.

A.2 Méthode des LSB généralisée : G-LSB

La capacité C de la méthode LSB est C=n avec $n \in \mathbb{N}$. La méthode G-LSB se propose de rendre C granulaire de telle manière que C=n avec $n \in \mathbb{Z}^+$.

A.3 Le Codage Arithmétique

Le codage arithmétique attribue une valeur réelle à une suite de symboles. Le principe de base consiste à diviser l'intervalle des réels [0 1[en sous-intervalles dont les longueurs sont fonction des probabilités des symboles. Au fur et à mesure du codage des symboles, la longueur de l'intervalle diminue en tenant compte du modèle

et du sous-intervalle précédent.

Symboles	Probabilité	partition Initiale
a	0,2	[0 0,2[
e	0,3	[0,2 0,5[
i	0,1	[0,5 0,6[
o	0,2	[0,6 0,8[
u	0,1	[0,8 0,9[
!	0,1	[0,9 1,0[

TAB. A.1 – *Partition de l'intervalle [0 1[pour l'alphabet (a,e,i,o,u,!)*

Intevalle de travail : [0 1,0[
Symbole à traiter : e

$$(1{,}0 - 0) * [0{,}2\ 0{,}5[+ 0 = [0{,}2\ 0{,}5[$$

Symbole à traiter : a

$$(0{,}5 - 0{,}2) * [0\ 0{,}2[+ 0{,}2 = [0{,}2\ 0{,}26[$$

Symbole à traiter : i

$$(0{,}26 - 0{,}2) * [0{,}5\ 0{,}6[+ 0{,}2 = [0{,}230\ 0{,}236[$$

Symbole à traiter : i

$$(0{,}236 - 0{,}230) * [0{,}5\ 0{,}6[+ 0{,}230 = [0{,}233\ 0{,}2336[$$

Symbole à traiter : !

$$(0{,}2336 - 0{,}233) * [0{,}9\ 1{,}0[+ 0{,}233 = [0{,}23354\ 0{,}2336[$$

Tout réel appartenant à l'intervalle [0,23354 0,2336[code le message eaii!

FIG. A.1 – *Construction du code du message eaii!*

Annexe B

Introduction à la programmation d'une DLL en C

Ce document a été réalisé dans un but pédagogique pour faciliter l'écriture de DLL en C pour tout autre personne du laboratoire. Ce document est à mettre à jour pour expliquer l'installation et la configuration de gcc et de l'outil gnumex pour Matlab®. Gcc permet de compiler en C, en Cpp, en Fortran et Java ...